Review

AI-Enabled Text-to-Music Generation: A Comprehensive Review of Methods, Frameworks, and Future Directions

Yujia Zhao , Mingzhi Yang , Yujia Lin , Xiaohong Zhang , Feifei Shi , Zongjie Wang * , Jianguo Ding , Huansheng Ning *

Posted Date: 25 February 2025

doi: 10.20944/preprints202502.1791.v1

Keywords: Music Generation; Text-to-music Generation; Artificial Intelligence; Large Language Model

Review

AI-Enabled Text-to-Music Generation: A Comprehensive Review of Methods, Frameworks, and Future Directions

Yujia Zhao [1,†], Mingzhi Yang [2,†], Yujia Lin [3], Xiaohong Zhang [4], Feifei Shi [1], Zongjie Wang [1,*], Jianguo Ding [5] and Huansheng Ning [1,*]

[1] School of Computer and Communications Engineering, University of Science and Technology Beijing, Beijing,100083, China
[2] Guangxi Tourism Development One-Click Tour Digital Cultural Tourism Industry Co., Ltd, Guangxi 530012, China
[3] Chunan Academy of Governance, Zhejiang, 311700, China
[4] Jinzhong University, Shanxi, 030606, China
[5] Department of Computer Science, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden
* Correspondence: wangzj@ustb.edu.cn (Z.W.); ninghuansheng@ustb.edu.cn (H.N.)
† These authors contributed equally to this work.

Abstract: Text-to-music generation integrates natural language processing and music generation, enabling artificial intelligence (AI) to compose music from textual descriptions. While AI enabled music generation has advanced, challenges in aligning text with musical structures remain under explored. This paper systematically reviews text-to-music generation across symbolic and audio domains, covering melody composition, polyphony, instrumental synthesis, and singing voice generation. It categorizes existing methods into traditional, hybrid, and LLM-centric frameworks according to the usage of large language model (LLM), highlighting the growing role of LLMs in improving controllability and expressiveness. Despite progress, challenges such as data scarcity, representation limitations, and long-term coherence persist. Future work should enhance multi-modal integration, improve model generalization, and develop more user-controllable frameworks to advance AI-enabled music composition.

Keywords: Music Generation; Text-to-music Generation; Artificial Intelligence; Large Language Model

Introduction

1.1. Background

Music, as a "universal language," [1] bridges different cultures and historical periods, playing a significant role in expressing human emotions and creativity. Traditional music composition often relies on musicians applying their knowledge of music theory to create works using real instruments. In contrast, computers have gradually become tools for music creation, utilizing algorithms and models to replicate the composition process. This evolution has led to the emergence of music generation, a field that originally relied heavily on music theory as prior knowledge to design algorithms. However, recent advancements have shifted the focus from knowledge-driven approaches to data-driven methods, leveraging large datasets of musical compositions to enhance generative capabilities.

Music generation is a typically multi-modal task involving the transformation of symbols, audio, text, images, and other modalities [2]. Among these, text-to-music generation stands out as a uniquely promising area due to its ability to interpret natural language descriptions and transform them into

music. Unlike other modalities such as images or videos, text provides a more intuitive, user-friendly, and accessible medium for expressing musical intent, allowing users to articulate emotions, styles, or themes with precision and simplicity. This accessibility significantly lowers the barriers to music creation, enabling broader participation from individuals without formal musical training. Furthermore, the potential of text-to-music generation extends beyond user convenience—it offers a transformative tool for diverse applications such as music therapy, dynamic video soundtracks, and immersive experiences in the metaverse. By bridging natural language processing (NLP) with music generation, this field can redefine how music is created and experienced, making it a critical area of study. This review is thus essential for providing a comprehensive understanding of the technological advancements, challenges, and future opportunities in text-to-music generation, setting the stage for continued innovation in this emerging domain.

1.2. Motivation

Text-to-music generation is an emerging research area at the intersection of artificial intelligence, music generation, and natural language processing. While existing reviews have extensively explored general music generation, they have primarily focused on traditional composition tasks, single-modality generation, or deep learning-based music synthesis, often overlooking the unique cross-modal challenges of text-to-music generation. This gap is particularly significant given the increasing integration of large language models (LLMs) in creative AI, which has opened new possibilities for translating textual descriptions into structured, meaningful, and emotionally resonant musical compositions. Despite the transformative potential of LLMs in aligning textual inputs with complex musical outputs, their role in text-to-music generation remains underexplored, highlighting the need for a comprehensive review of this rapidly growing field.

Existing reviews have extensively explored the broader landscape of music generation tasks, synthesizing representational levels, compositional processes, and single-modality tasks. Several surveys have comprehensively reviewed the broader field of music generation, offering valuable insights into its methodologies and applications. A significant portion of these studies emphasizes the role of deep learning. For instance, Ji et al. (2020) [1] provide an overview of various compositional tasks at different levels of music generation, while Briot et al. (2020) [3] explore deep learning cases from perspectives such as musical structure, creativity, and interactivity. Another study by Ji et al. (2023) [4] delves into the applications of deep learning in symbolic music generation. Additionally, Hernandez-Olivan and Beltran (2021) [5] examine research advancements by aligning them with the stages and methods involved in the human creative process of composing music.

Other reviews have examined the field from alternative perspectives. Civit et al. (2022) [6] employed bibliometric methods to analyze the development of artificial intelligence in music generation. Herremans et al. (2017) [7] categorized music generation systems based on their functionality. Zhu et al. (2023) [8] introduced various tools for music generation, and Wen and Ting (2023) [9] discussed the evolution of computational intelligence techniques in this domain. Ma et al.(2024) [2] give a quite comprehensive survey on foundation models for music. These comprehensive reviews highlight the diverse approaches and significant progress in music generation.

However, the unique challenges and opportunities of cross-modal text-to-music generation remain underexplored. Most existing reviews focus on broader music generation tasks or single-modal approaches, often classifying studies based on network architectures or technical methodologies. This makes it difficult for researchers to gain a precise understanding of specific generation tasks, such as generating melodies from lyrics. Furthermore, the transformative potential of LLMs in text-to-music generation has been largely overlooked. While LLMs have revolutionized other fields, their integration into text-to-music generation—particularly in aligning textual inputs with complex musical outputs—remains an emerging area of research. This paper aims to address these gaps by providing a comprehensive review of text-to-music generation, with a focus on the integration of LLMs and the unique challenges of cross-modal generation.

1.3. Objectives

This paper aims to provide a comprehensive and task-oriented review of advancements in text-to-music generation, addressing key gaps in the field and proposing actionable insights for future research. The main objectives of this work are as follows:

- **To systematically classify and analyze text-to-music generation tasks:** By categorizing tasks into symbolic and audio domains, the paper examines subtasks such as melody generation, polyphony generation, singing voice synthesis, and complete song composition. This taxonomy offers a clear aspect for understanding the distinct challenges and opportunities within each domain. This framework supports modular method development by providing researchers with a structured reference for locating domain-specific innovations.
- **To emphasize the potential of LLMs through framework comparison:** The study focuses on the traditional methods, hybrid approaches, and end-to-end LLM systems, providing a detailed analysis of their strengths, limitations, and applicability. The analysis highlights the progressive improvements introduced by LLMs, demonstrating their ability to enhance user controllability, generalization capability, etc., offering a clearer perspective on the role of LLMs in advancing AI-enabled music composition.
- **To identify challenges and propose future directions:** This objective is crucial because addressing unresolved challenges—such as data scarcity, model generalization, emotion modeling, and user interactivity—is the foundation for advancing text-to-music generation. By systematically analyzing these barriers, the paper provides a roadmap for overcoming limitations that currently hinder the effectiveness and creativity of such systems. This exploration advances text-to-music generation, establishing it as a key direction for creative industries.

This paper is organized as follows. Section 2 provides an overview of the evolution of text-to-music generation, tracing its development from rule-based systems to the integration of LLMs. Section 3 discusses the representation forms of text and music, as well as their roles in aligning textual semantics with musical outputs. Section 4 critically reviews text-to-music generation methods, categorizing them into symbolic domain methods and audio domain methods and providing a comparative analysis of existing techniques. Section 5 sorts out three mainstream research frameworks based on the LLMs integration, highlighting the potential of LLMs in enhancing end-to-end generation and multi-modal integration. Section 6 outlines the challenges and future directions, identifies unresolved issues such as data scarcity, emotion modeling, and interactive systems. Finally, Section 7 concludes the paper by summarizing key insights and proposing actionable recommendations for advancing research in this emerging field.

2. Evolution

2.1. Early Rule-Based Systems

Music generation research dates back to the mid-20th century, initially focusing on using programming languages and mathematical algorithms to simulate the process of music creation. Early works such as Iannis Xenakis' use of probability theory [10] as well as Lejaren Hiller and Leonard Isaacson's work Illiac Suite [11] marked the birth of automated music generation. These efforts were primarily concerned with generating melodies, harmonic progressions, and rhythmic patterns.

Early approaches to text-to-music generation also relied heavily on predefined rules and templates to create music. These methods included lyric-based melody generation, where algorithms analyzed the content of lyrics to produce melodies that aligned with their emotional and rhythmic structures. For example, systems mapped syllables to notes based on rhythmic patterns and harmonic rules [12,13]. Additionally, textual instruction sequences, often based on music theory, guided melody generation. Such systems translated harmonic progressions (e.g., I-IV-V-I) [14,15] and other theoretical constructs into melodies by algorithmically processing these instructions. While these

methods ensured compliance with musical structures, they were limited by their reliance on rigid rules and templates, often resulting in a lack of diversity and creativity in the generated outputs.

2.2. Emergence of Machine Learning

The late 20th and early 21st centuries brought significant shifts with the introduction of machine learning techniques into music generation. In the machine learning era, traditional techniques for music generation focus on learning patterns and structures from large datasets of existing compositions. Early methods often employed Hidden Markov Models (HMMs), which excel at modeling sequential data by capturing probabilistic transitions between states [16]. HMMs were used to generate melodies or harmonies by determining the likelihood of note sequences, though their capacity to handle complex musical structures was limited by their reliance on fixed state-transition probabilities.

Building on these early approaches, more advanced models such as Recurrent Neural Networks (RNNs) [17,18], which are suited for sequential data like music. These networks generate melodies or chord progressions by predicting the next note based on prior information. An improvement to RNNs, Long Short-Term Memory (LSTM) networks [19], address the challenge of remembering long-term dependencies, allowing for more coherent and extended music sequences.

The same techniques began being applied to text-to-music generation, where systems began to link textual data with musical outputs. In text-to-music generation, this shift enabled the field to move beyond lyric-to-melody mapping. Researchers began exploring models that not only mapped text to melody but also incorporated additional musical elements such as harmony, accompaniment, and vocals. However, these systems still had limitations. They were heavily dependent on the patterns present in the training data, and the generated music often lacked true creativity and innovation.

2.3. The Rise of Deep Learning and Cross-Modal Approaches

With the advent of deep learning, the capabilities of general music and text-to-music generation greatly expanded. Deep learning models such as Generative Adversarial Networks (GANs) [20] and Transformers [21] began to offer more realistic and diverse music compositions by capturing complex dependencies in both symbolic music and raw audio data.

The emergence of text-to-audio models has opened a new direction for music generation. Models like AudioLM [22] and Suno's bark[1] combine audio representation with text representation, allowing them to understand textual content and generate corresponding audio. Building on these innovations, researchers began developing more comprehensive text-to-music generation models, which go beyond simple lyric-to-melody mappings. These models now aim to capture emotion, themes, and other non-musical elements from texts to guide the music generation process.

The success of diffusion models in image generation tasks [23] has led researchers to apply these models to music generation [24]. This approach has proven effective in creating richer, more expressive outputs and has laid a strong foundation for the continued development of contemporary text-to-music generation techniques.

2.4. The Integration of LLMs

Recent breakthroughs in music generation have been driven by multi-modal and cross-modal learning techniques, which integrate various data types such as text, audio, and symbolic representations. These models utilize advanced deep learning frameworks to capture the intricate relationships between these diverse data types, enabling the generation of music that is not only structurally coherent but also emotionally expressive and contextually rich.

In parallel, advancements in large-scale models, particularly LLMs, have paved the way for end-to-end text-to-music generation. These models, trained on vast datasets of text and music, are capable

[1] https://github.com/suno-ai/bark

of directly mapping textual descriptions to musical outputs. For example, modern LLMs-based systems [25] can interpret detailed textual prompts, including emotional expressions, scene descriptions, or stylistic preferences, and generate highly consistent compositions in style, rhythm, and harmony. This end-to-end paradigm significantly lowers the barrier to music creation, allowing users without formal music training to create complex and expressive musical works. Furthermore, the adaptability of these models opens new possibilities for personalized music creation, soundtrack generation, and other multi-modal applications.

Music generation technologies have evolved from rule-based methods to data-driven approaches, with deep learning and large language models significantly advancing general music generation and text-to-music generation. The adoption of multi-modal techniques has broadened the capabilities of these fields, enabling the creation of music that is both contextually relevant and emotionally nuanced. The text-to-music generation now integrates natural language processing, deep learning, and creative music technology, opening new avenues for research and development.

3. Representation Forms of Text and Music

3.1. Text Types

In a text-to-music generation, the common text types are mainly categorized into three types, which serve as input for the generation process and provide guidance for music creation. The common text types and their roles in text-to-music generation are shown in **Table 1.**

Table 1. Text types and their characteristics.

Category	Description	Application Example	Generation Characteristics	Challenges
Lyrics	The singing words of songs.	"Let it be, let it be..."	● Lyrics-melody matching ● Singing voice synthesis ● Emotion- and Rhythm-based	● Multi-language ● Cultural context ● Appropriate rhythm
Musical Attributes	Describes musical rules like chords.	I-IV-V-I, 120 bpm	● Music theory-based ● Using attribute templates	● Complex music theory ● Lacking of creativity
Natural Language Description	Describes emotion or scene.	"Create a melody filled with hope..."	● Flexible description ● Diverse music features.	● Abstract concepts ● understanding ● Converting consistency

3.2. Musical Representation

3.2.1. Event Representation: Midi-like

MIDI [2] (Musical Instrument Digital Interface) is an industry-standard protocol for communicating between electronic musical instruments and exchanging data between an instrument and a computer. MIDI files record information about a player's actions, such as which key was pressed, how hard it was pressed, and how long it lasted. These messages are called “events”, which are binary data such as Note On, Velocity, Note Of, Aftertouch, Pitch Bend, etc. **Table 2** lists the common events used in symbolic music generation research.

[2] https://en.wikipedia.org/wiki/MIDI

Table 2. Common events in music generation research.

Event Type	Description	Example Format
Note On	Starts a note	Note On, Channel 1, Pitch 60, Velocity 100
Note Off	Ends a note	Note Off, Channel 1, Pitch 60, Velocity 0
Program Change	Changes in instrument or sound	Program Change, Channel 1, Program 32
Control Change	Adjusts control parameters (e.g., volume, sustain pedal)	Control Change, Channel 1, Controller 64, Value 127
Pitch Bend	Bends pitch slightly or continuously	Pitch Bend, Channel 1, Value 8192
Aftertouch	Pressure applied after pressing a note	Aftertouch, Channel 1, Pressure 60
Tempo Change	Sets playback speed in beats per minute (BPM)	Tempo Change, 120 BPM
Time Signature	Defines beat structure (e.g., 4/4, 3/4 time)	Time Signature, 4/4
Key Signature	Sets the song's key (e.g., C Major, G Minor)	Key Signature, C Major

MIDI is a highly compatible and easy-to-edit file format with a small file size that facilitates communication between devices and music creation, as shown in **Figure 1**. In music generation research, an algorithm or model first slices a melody into sequences of notes and then establishes a mapping relationship between musical elements and numbers through quantization and encoding to obtain a data representation of the music. Native MIDI representations have representational limitations, such as the inability to express the concepts of quarter notes or rest, not being able to represent the musical onset time, etc. Therefore, some studies have improved MIDI representations for music generation by proposing REMI [26], REMI+ to represent more information.

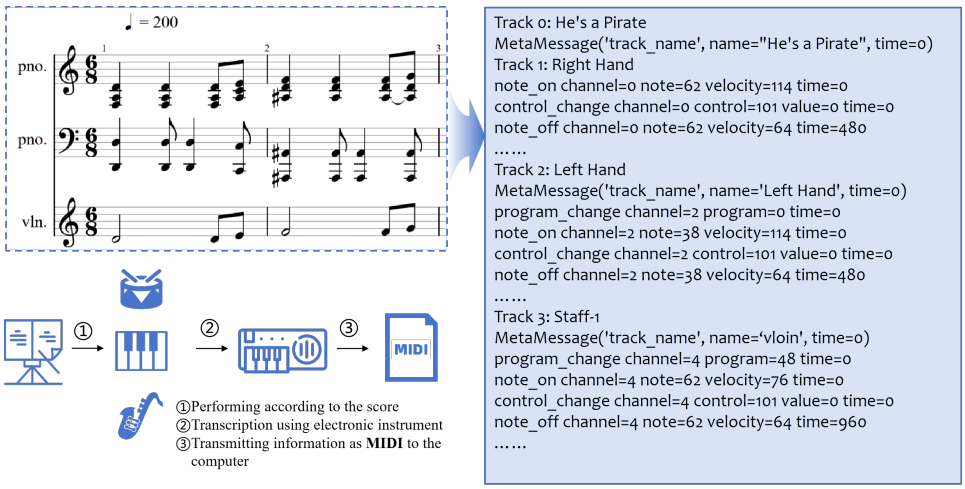

Figure 1. MIDI-based event representation.

3.2.2. Audio Representation: Waveform and Spectrogram

Audio representation is a continuous form, typically categorized into one-dimensional and two-dimensional forms. One-dimensional representations, usually in the time domain, are the simplest type. In this form, the audio signal is represented as a time series, often visualized as a waveform. Each point in the waveform corresponds to the amplitude value at a specific time, and the entire sequence shows how the audio signal changes over time. In contrast, two-dimensional representations, such as spectrograms, transform the audio signal from the time domain to the frequency domain. These representations break down the audio signal into various frequency components using methods like the Short-Time Fourier Transform (STFT), as shown in **Figure 2.**

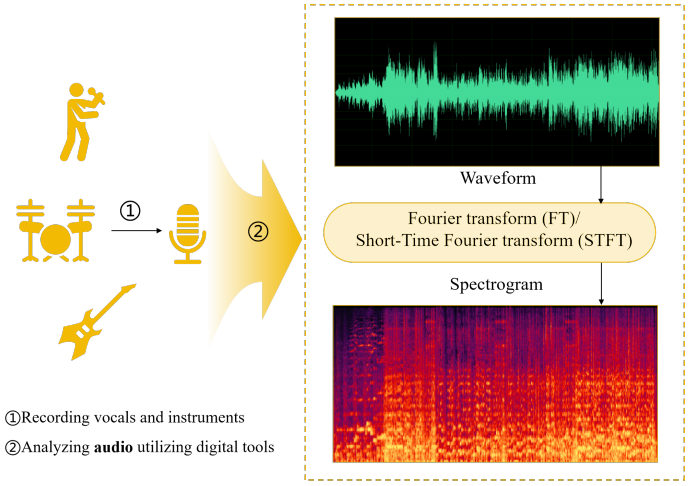

Figure 2. Waveform and Spectrogram Representation.

Compared to MIDI representation, audio waveform, and spectrogram retain more details, enabling the style, timbre, emotion, and vocal performance to be modeled. As a result, they offer a greater advantage in creating natural and expressive musical compositions.

3.2.3. Text Representation: ABC Notation

ABC notation is an ASCII-based text format for representing music[3], using simple letters and symbols to encode information such as notes, rhythms, and key signatures. It consists of two parts: the header fields and the tune body. The header fields typically include track number (X), title (T), meter (M), note length unit (L), tempo (Q), key (K), and others. The tuning body represents the sequence of notes, in which the letters A to G correspond to musical notes, and the numbers 1 to 8 indicate pitch variations. For example, "C" represents the C note, and "C2" represents the second octave of C. The symbol "|" indicates bar divisions, while numbers specify the duration of notes. For instance, "C/2" indicates that the C note lasts for two eighth notes. Additional symbols are used to represent note lifts, sustains, rests, and other musical elements.

After being encoded, the text files of ABC notation can extract information such as notes, rhythms, and chords. Based on the extracted information, such as note start, note end, and note strength, they can eventually be interconverted with MIDI files, as shown in **Figure 3**. Therefore, in this paper, we also categorize the research that generates the form of ABC notation into the symbolic domain.

[3] https://abcnotation.com

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: Posted: 25 February 2025 doi:10.20944/preprints202502.1791.v1

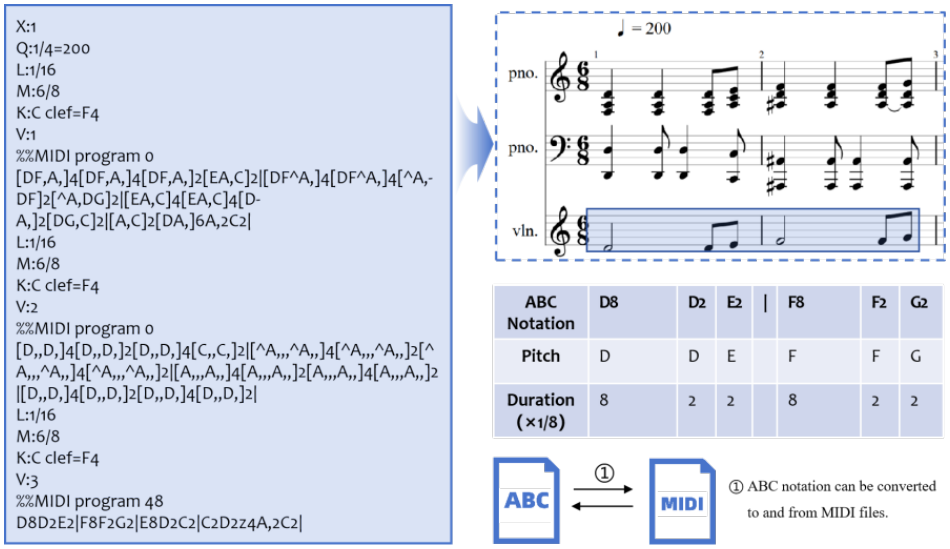

Figure 3. ABC Notation.

4. Methods

In a text-to-music generation, methods are broadly categorized into symbolic domain and audio domain based on data representation formats as shown in **Figure 4**. The textual inputs—comprising lyrics, musical instructions, and natural language descriptions—serve as semantic drivers for generation tasks. The symbolic domain, anchored in structured representations such as MIDI and ABC notation, facilitates melody generation and polyphony generation. In contrast, the audio domain operates on raw waveform and spectrogram data to achieve instrumental music synthesis, singing voice generation, and complete song composition. The evolution of text-to-music systems reflects a clear trajectory: advancing from single-track to multi-track generation, from simplistic structures to intricate compositions, and from localized musical fragments to holistic, contextually coherent pieces.

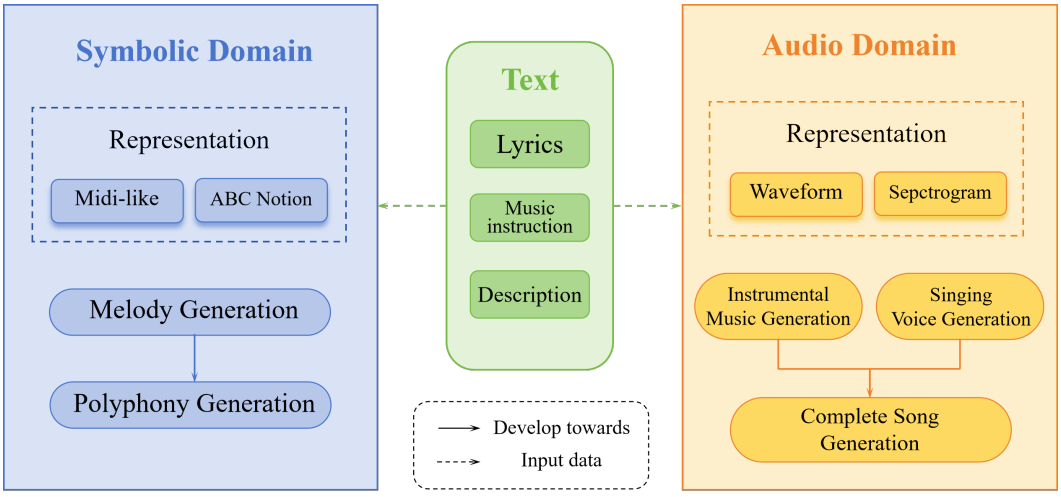

Figure 4. Overview of Text-to-Music Generation Based on Representation and Task Domains.

4.1. Symbolic Domain Methods

Symbolic-domain music generation is the task of automatically generating symbolic music representations by using computational models. In this process, algorithms create new musical sequences with coherent and creative characteristics based on previously learned patterns or rules. Symbolic music representations usually refer to discrete musical information structures, such as

MIDI files or digitized sheet music (e.g., ABC notation), which decompose music into a series of discrete time and frequency units, such as notes, rhythms, pitches, and intensities.

4.1.1. Melody Generation

Melody is one of the fundamental elements of music, which consists of a series of notes arranged in a specific rhythm and pitch. The melody generation task is the process of automatically generating new melodic lines through algorithms or models. This task is a core component of music generation. The melody generation task aims to create a new melody that conforms to the rules of music theory and is artistically pleasing. The text-based melody generation task is mainly divided into lyrics-based melody generation and text description-based melody generation.

1. **Lyric-based Melody Generation**

The earliest attempt at lyric-based melody generation was based on rule-based systems. Fukayama et al. (2010) [12] developed an algorithm for generating melodies when specific Japanese lyrics, rhythmic patterns, and harmonic sequences were provided. The algorithm treats composition as an optimal solution search problem under the constraints of lyrics' rhymes, and searches for the optimal composition through dynamic programming. In addition, the algorithm innovatively integrates text with melody, which is also considered to be the beginning of the task of generating melodies from lyrics.

As statistical methods began to gain traction, researchers like Monteith et al. (2012) [27] moved away from rule-based systems by applying probabilistic models, such as n-gram models, to generate melodies. The system produced hundreds of rhythm and pitch combinations for given lyrics, and the best result was selected using metrics to evaluate the generated melody. This approach shifted from strict rule-based generation to probabilistic modeling, allowing for more variety in melody generation, though it still depended heavily on predefined patterns and lacked the complexity needed for capturing the full depth of melody generation. Similarly, Scirea et al. (2015) [13] expanded this idea by constructing Markov chains over note sequences using lyric syllables, showcasing how statistical models could link lyrics with melody generation through probabilistic transitions.

The next major shift came with machine learning and neural network algorithms. Ackerman et al. (2017) [28] applied random forests to predict note durations and scales, marking an early attempt at using machine learning to model melodic structures. While this approach improved the generation of rhythmic and melodic patterns, it still required handcrafted features and could not fully capture the complex relationships between lyrics and melodies. Using neural networks further explores the intrinsic connection between lyrics and melody. Bao et al. (2019) [29] developed SongWriter, a sequence-to-sequence (seq2seq) model built on RNNs, which generates melodies from lyrics while precisely aligning them. The model used two encoders: one to encode the lyrics and the other to encode the melody context. The hierarchical decoder generated the musical notes and their corresponding alignments with the lyrics. The use of seq2seq models marked a significant improvement, as they could learn complex mappings between lyrics and melodies, resulting in more cohesive and flexible melodies. This approach outperformed earlier machine learning methods, allowing for better alignment between the generated melodies and the input lyrics.

Building on the success of RNN-based models like SongWriter, Long Short-Term Memory (LSTM) networks have been used in music generation due to their ability to capture long-term dependencies in sequential data. Unlike standard RNNs, LSTMs address the vanishing gradient problem, making them particularly effective for modeling the complex temporal structures inherent in music. In parallel, GANs have emerged as a powerful framework for music generation, particularly in creating high-quality and diverse musical outputs. The research combining the above two structures has become a hot topic. Yu et al. (2021) [30] used a conditional LSTM-GAN for the lyrics of generation-based melodies. They combined syllable embedding vectors converted from text lyrics with noise vectors and input them into the generator. A deep discriminator was also trained to distinguish the generated MIDI note sequences from the real ones. This approach demonstrates both LSTM's ability to capture long-term dependencies of melodies and GAN's advantage to enhance the

realism and naturalness of melodies through an adversarial learning mechanism. To further improve generation quality, a three-branch conditional LSTM-GAN network is used in Srivastava et al.(2022) [31]. Research utilizing a single structure independently is also investigating the potential of LSTMs and GANs. Yu et al.(2022) [32] also proposed a three-branch structure for modeling three independent melodic attributes. The difference is that they did not use LSTM but used a conditional hybrid GAN. Zhang et al. (2023) [33] introduced inter-branch memory fusion (Memofu), which facilitates information flow between multi-branch stacked LSTM networks. This allows for better modeling of dependencies across multiple musical attributes and sequences, improving the overall coherence of the generated melodies.

Transformer-based models have been widely applied in music generation. However, limited by the quality of the melody-lyrics pairing dataset and the diversity of variations in real melodies, a large number of studies are still highly dependent on the dataset and are not highly transferable. SongMASS proposed by Sheng et al. (2021) [34] effectively alleviates data dependency by using separate lyrics and melody encoder-decoder structures within a Transformer-based framework. It also enhances the matching accuracy of lyrics and melodies by introducing sentence-level and word-level alignment constraints. However, the complex alignment mechanism may increase the difficulty of training and reduce the controllability of melody generation. To overcome data scarcity and improve generation controllability, Ju et al. (2022) [35] proposed TeleMelody, a two-stage generation pipeline based on music templates. The system first converts lyrics to templates and then generates melodies based on the templates. The music templates include tonality, chord progressions, rhythmic patterns, and terminations, and they use a self-supervised approach to realize template-generated melodies, which solves data dependency.

In a recent study, thanks to the development of LLMs, Ding et al. (2024) [36] proposed SongComposer, a LLM for lyrics and melody composition. It employs a single language model architecture instead of the separate encoders and decoders of traditional approaches, and uses the next-token prediction technique. This approach allows the model to predict subsequent notes based on the current lyrics and a portion of the melody until the entire song is generated, outperforming SongMASS and TeleMelody. In contrast to traditional methods, SongComposer does not require complex rules or preset musical templates but rather learns patterns from large amounts of data to make predictions that can generate high-quality melodies without the guidance of explicit music theory. In addition to generating melodies from lyrics, the model can also perform the tasks of generating lyrics from melodies, song continuation, and songs from text. The text-to-song task in this context refers to a task pipeline consisting of textual cues to generate lyrics, lyrics to generate melodies, and artificially produced vocals and accompaniment (as demonstrated by the authors in the project demo4), and is therefore distinct from the text-to-song task mentioned later.

The field of lyric-based melody generation has evolved from conditional constraints to deep learning, and has reached new heights driven by large language modeling. Despite its theoretical appeal, this approach still faces several challenges in practice. First, the mapping relationship between lyrics and melodies is not one-to-one. The same lyrics can correspond to multiple plausible melodic configurations, making it more difficult for the model to learn the correct mapping relationship. Second, high-quality, diverse, and representative datasets of lyric-melody pairings are relatively scarce, which further limits the learning effectiveness and generalization ability of the model.

2. Musical attribute-based Melody Generation

Earlier studies, limited by the mapping of textual semantics to music, have very limited generative capabilities. TransProse, proposed by Davis et al. (2014) [37], contains several mapping rules for sentiment labels to musical elements. TransProse generates music based on the density of sentiment words in a given text. However, TransProse does not reflect non-emotional information in the text, and its creativity is limited by manually formulated mapping rules. Rangarajan et al. (2015)

[4] https://pjlab-songcomposer.github.io/

[38] devised three strategies for mapping text to music: using all letters, using only vowels, and using vowels in conjunction with the part of speech (POS) of words. However, since it is based on character-level mapping, the generated music is very random and does not reflect the semantic information in the text. In order to optimize the mapping between text and music, Zhang et al. (2020) [39] proposed a framework called Butter, which is a multimodal representation learning system for bidirectional music and text retrieval and generation. The system learns music, keyword descriptions, and their cross-modal representations based on a Variational Auto Encoder (VAE). Butter can generate three types of potential representations: music representations, keyword embeddings, and cross-modal representations, and can generate ABC notation representations of music from text containing three musical keywords (e.g., key, beat, and style). However, the method is limited by the fact that the three keywords must be specified precisely, and the generated music is restricted to the Chinese folk song dataset.

3. **Description-based Melody Generation**

To escape the limitations of manually formulated rules, Wu et al. (2023) [40] developed a tranformer-based model. The model achieved for the first time the generation of complete and semantically consistent musical scores directly from text-based natural language descriptions and also demonstrated the effectiveness of using publicly available pre-trained BERT, GPT-2, and BART checkpoints on music generation tasks.

With the development of LLMs, such large language models not specifically designed for music as GPT-4 [25], and LLaMA-2 [41] have shown some level of music comprehension, but they still perform poorly in music generation. However, ChatMusician, introduced by Yuan et al. (2024) [42], represents a significant progress. ChatMusician uses music as a second language for large language models. This new approach, based on the continuously pre-trained and fine-tuned LLaMA2 model, is trained on a 4B dataset and utilizes the ABC notation to seamlessly fuse music and text. In so doing, ChatMusician enabled in-house music composition and analysis without relying on external multi-modal frameworks. Compared to traditional LLMs, ChatMusician can understand music, generate structured, full-length musical compositions, and condition text, chords, melodies, motifs, musical forms, etc., beyond the GPT-4 baseline.

Early research on description-based melody generation was limited by the ability to generate effective mappings from textual semantics to melodies. In recent years, with technological advances, researchers have been able to generate semantically consistent melodies from natural language descriptions. In the latest progress, models such as ChatMusician are not only able to understand music, but also to generate structurally complete and moderate-length musical compositions, which significantly improves the ability of text-generated music. The generation of independent melodic lines lays a foundation for polyphony generation. Relevant studies are summarized in **Table 3.**

4.1.2. Polyphony Generation

Polyphony is a style of musical composition employing two or more simultaneous but relatively independent melodic lines. These melodic lines intertwine and support each other harmonically, creating a rich musical texture. Each instrument or voice can have its Midi track that flows and unfolds in a harmonious manner. Typical polyphonic music is polyphonic pieces (e.g. classical music) as well as contemporary musical accompaniment. Creating polyphonic music is, therefore, more complex than creating a single melody. Generating polyphonic music from text requires the model to correctly extract or understand the music-theoretic knowledge or semantic information contained in the text and to generate harmonized polyphonic music.

1. **Musical Attribute-based Polyphony Generation**

Early studies used strict attribute templates as textual input to accurately generate conforming multi-track symbolic music. Evolving from attribute-conditional controlled generation, this type of research creatively replaced attribute labels with text input. Rütte et al. (2023) [14]proposed FiGARO. This system is based on a Transformer and can generate multi-track symbolic music by combining

expert and learning features. They introduce a self-supervised description-to-sequence learning method. The method automatically extracts fine-grained, human-interpretable features from music sequences and trains a sequence-to-sequence model, reconstructing the original music sequence from the description. However, the descriptions are complex attribute templates, “expert description”, including three types of musical attributes, namely instrument, harmony, and meta-information. These “high-level control codes” raise the bar for users while allowing them to precisely control the generation.

 doi:10.20944/preprints202502.1791.v1

Table 3. Melody generation tasks.

Task Type	Model Name	Year	Music Representation	Model Architecture	Description	Large Model Relevance	Dataset Name	Generated Music Length	Accessed Link
Lyric-based Melody Generation	SongComposer [36]	2024	MIDI	Transformer	LLM, Instruction Following, Next Token Prediction	A LLM designed for composing songs	SongCompose-PT	Multiple minutes	https://pjlab-songcomposer.github.io/
	TeleMelody [35]	2022	MIDI	Transformer	Transformer-based Encoder-Decoder, Template-Based Two-Stage Method	/	lyric-rhythm data (9,761 samples in English and 74,328 samples in Chinese)	Not mentioned	https://ai-muzic.github.io/telemelody/
	SongMass [34]	2021	MIDI	Transformer	Pre-training, the sentence-level and token-level alignment constraints.	/	380,000+ lyrics from MetroLyrics; The Lakh MIDI Dataset	Not mentioned	https://musicgeneration.github.io/SongMASS/
	[30]	2021	MIDI	LSTM-GAN	Conditional LSTM-GAN, Synchronized Lyrics - Note Alignment	/	12,197 MIDI songs	Not mentioned	https://drive.google.com/file/d/1ugOwfBsURax1VQ4jHmI8P3ldE5xdDj0l/view?usp=sharing
	SongWriter [29]	2019	MIDI	RNN	Seq-to-Seq, Lyric-Melody Alignment	/	18,451 Chinese pop songs	Not mentioned	/
	ALYSIA [28]	2017	MusicXML &MIDI	Random Forests	Co-creative Songwriting Partner, Rhythm Model, Melody Model	/	/	Not mentioned	http://bit.ly/2eQHado
	Orpheus [12]	2010	MIDI	/	Dynamic Programming	/	Japanese prosody dataset	Not mentioned	http://orpheus.hil.t.u-tokyo.ac.jp
Musical attribute-based Melody Generation	BUTTER [39]	2020	MIDI, ABC Notation	VAE	Representation Learning, Bi-directional Music-Text Retrieval	/	16,257 Chinese folk songs	Short Music Fragment	https://github.com/ldzhangyx/BUTTER
	[38]	2015	MIDI	/	Full parse tree, POS Tag	/	/	Not mentioned	/
	TransProse [37]	2014	MIDI	Markov Chains	Generate music from Literature, Emotion Density	/	Emotional words from literature	Not mentioned	http://transprose.weebly.com/final-pieces.html
Description-based Melody Generation	ChatMusician [42]	2024	ABC Notation	Transformer	Music Reasoning, Repetition Structure	An LLM of symbolic music understanding and generation	MusicPile (4B tokens)	Full Score of ABC Notation	https://shanghaicannon.github.io/ChatMusician/
	[40]	2023	ABC Notation	Transformer	Exploring the Efficacy of Pre-trained Checkpoints.	Using pre-trained checkpoints	Textune (282,870 text-tune pairs)	Full Score of ABC Notation	/

In order to lower the threshold for users, human natural language is used to describe target generated music, enabling the re-understanding and re-generalization of natural language to attribute templates to become a new development direction. MuseCoco, proposed by Lu et al. (2023) [15], is a typical representative. Unlike FiGARO, this system extends the set of musical attributes, and its attribute templates cover 12 musical attributes such as instrument, tempo, time, and pitch range. In addition, the system allows natural language input instead of complex templates. This system also adopts a two-stage framework, consisting of text-to-attribute understanding and attribute-to-music generation. MuseCoco leverages ChatGPT's superior performance in text understanding to convert text descriptions into attributes, which allows users to use natural language to generate music. On top of that, A richer set of attribute templates also improves the accuracy of the music generated to meet users' requirements. A richer set of attribute templates also improves the accuracy of the music generated to meet the user's requirements.

2. **Description-based Polyphony Generation**

In a recent study, Liang et al. (2024) [43] proposed ByteComposer, which utilizes LLM to simulate mankind's music-composing process. The system adopts a modular design that includes four stages: conceptual analysis, draft generation, self-evaluation and revision, as well as aesthetic selection. Unlike MuseCoco, which only uses ChatGPT to extract attribute information from textual descriptions, ByteComposer embeds LLM as an expert module that not only extracts attribute information, but also provides guidance based on a library of music theory knowledge. As a result, ByteComposer allows LLM to play the role of "melody composer". At the same time, a voting module and a memory module are added to ByteComposer, enabling users to subjectively judge the generation results and store the evolution trajectory and interaction data. The system combines the interactive and knowledge understanding properties of LLMs with existing symbolic music generation models to achieve a melodic composition agent comparable to human creators. In addition, ComposerX proposed by Deng et al. (2024) [44], adopts a multi-agent approach to significantly improve the quality of music composition for large language models (e.g., GPT-4). ConposerX can generate coherent polyphonic musical compositions while following users' instructions. This has shown that the multi-agent approach boasts enormous potential in generative tasks. The division of labor of the agents is shown in the following **Table 4**:

Table 4. Division of Labor.

Agent Name	Task Description
Group Leader Agent	Responsible for **analyzing user input** and breaking it down into **specific tasks** to be assigned to other agents.
Melody Agent	**Generates a monophonic melody** under the guidance of the Group Leader.
Harmony Agent	Adds **harmony and counterpoint elements** to the composition to enrich its structure.
Instrument Agent	Selects **appropriate instruments** for each voice part.
Reviewer Agent	**Evaluates and provides feedback** on the melody, harmony, and instrument choices.
Arrangement Agent	Standardizes the final output into **ABC notation format**.

At the same time, ComposerX significantly reduces training costs. The quality of works generated by ComposerX is comparable to polyphonic compositions generated by specialized notated music generation systems [15,40] that require substantial computing resources and data. It is also worth noting that ChatMusician [42] can generate polyphonic music that meets the requirements and maintains good quality. However, it cannot select instruments and only generates polyphonic ABC notation for a single instrument.

Polyphonic music generation technology has evolved from using structured attribute templates to natural language descriptions, aiming to improve user-friendliness and enhance the diversity and

expressiveness of the generated music. The advantage of early structured attribute templates is their ability to ensure that the generated music adheres to certain musical theory standards. However, these templates have several limitations. First, the forms of text input are constrained by structured templates, requiring the selection of fixed attributes, which makes the generation process less flexible. Second, since the attribute templates essentially define the labels, the generated results often lack personalization. Additionally, the multitrack melodies generated are relatively independent, lacking coordination. With the development of deep learning and large language models, modern music generation systems now employ natural language processing to interpret text descriptions and model the collaborative relationships between multiple tracks, making the input process more intuitive and universal, while the output sounds more harmonious and pleasant. Relevant studies are summarized in **Table 5.**

Table 5. Polyphony Generation Tasks.

Task Type	Model Name	Year	Music Representation	Model Architecture	Description	Large Model Relevance	Dataset Name	Generated Music Length	Accessed Link
Musical Attribute-based Polyphony Generation	FIGARO [14]	2023	REMI+	Transformer	Human-interpretable, Expert Description, Multi-Track	/	LakhMIDI Dataset	Not mentioned	https://tinyurl.com/28etxz27
	MuseCoco [15]	2023	MIDI	Transformer	Text-to-attribute understanding and attribute-to-music generation	Textual synthesis and template refinement	MMD, EMOPIA, MetaMidi, POP909, Symphony, Emotion-gen	<= 16 bars	https://ai-muzic.github.io/musecoco/
Description-based Polyphony Generation	ByteCompo [43]	2024	MIDI	Transformer	Imitate the human creative process, Multi-step Reasoning, Procedural Control	A melody composition LLM agent	the Irish Massive ABC Notation dataset	Not mentioned	/
	ComposerX [44]	2024	ABC Notation	Transformer	Significantly improve the music generation quality of GPT-4 through a multi-agent approach	Multi-agent LLM-based framework	the Irish Massive ABC Notation dataset, KernScores	Varied Lengths	https://lllindsey0615.github.io/ComposerX_demo/

4.2. Audio Domain Methods

The audio domain text-to-music generation task is a task that automatically generates music segments directly at the audio signal level from input text. The output is usually in the form of time-series sound waveform data rather than a symbolic representation. The research challenge in audio-domain text-to-music generation tasks is establishing a good mapping between text and audio signals. To overcome this challenge, the model must efficiently capture the dependencies between text and music audio and generate high-quality sound outputs. By doing so, the generated music segments will not only follow the textual instructions but also sound smooth and pleasing to the ear.

4.2.1. Instrumental Music Generation

In recent years, the task of text-to-audio generation has gained significant attention as an important branch of cross-modal tasks. AudioLM [22], as a pioneering work, has made significant breakthroughs in audio modeling. This model maps audio signals to a series of discrete audio representations, transforming the audio generation task into a language modeling problem within this representation space. AudioLM can generate natural and smooth audio from brief prompts,

covering human speech, environmental sound effects, and basic piano melodies, while maintaining consistency and coherence across long-time sequences. Following the success of AudioLM, researchers have further explored how to use text input to precisely control the audio generation process, leading to models like AudioLDM, Tango, and Tango2 [45–47]. These models combine the advantages of language models and diffusion models, enabling efficient and expressive text-to-audio generation. These advancements have laid a crucial theoretical and technical foundation for music audio generation.

Early research generated new music by combining audio retrieval with textual labels. A typical example of this is Mubert[5], which constructs a music database with labels and assigns appropriate labels based on the user's text input. The appropriate music clips are then selected from the database and combined to create a new piece of music. This approach allows Mubert to respond quickly to user input prompts and to generate musical compositions with some degree of editing. However, Mubert has some limitations regarding creativity and flexibility because it relies on combining existing music fragments rather than creating entirely new ones.

The launch of Riffusion marked the beginning of the use of diffusion models for music generation tasks. Riffusion, developed by Forsgren et al. (2022) [24], is a real-time music generation system based on the stable diffusion model. It features direct noise diffusion on a spectrogram. Riffusion is suitable for live performance or real-time composition as it can rapidly generate short music clips (usually no more than a few seconds) when specific textual description or lyrics is given. Although Riffusion had significant limitations in terms of music length and complexity, it creatively migrated "text-to-image" technology to the audio domain. Since then, diffusion has become one of the most widely used models for music generation tasks. Huang et al. (2023) [48] proposed a model called Noise2Music. The model uses a two-stage diffusion modeling framework that includes a generator model and a cascade model. The study explored two intermediate representations, i.e., spectrograms and low-fidelity audio (3.2 kHz waveforms). Experimental results show that when low-fidelity audio is used as an intermediate representation, the results are better than when spectrograms are used. Nevertheless, the audio generated by Noise2Music can last for 30 seconds or less and the sampling rate is 24kHz. Schneider et al. (2023) [49] proposed Moûsai. This model also uses a two-stage diffusion modeling framework and is capable of generating stereo music at 48kHz lasting up to several minutes. The first stage of the model compresses the audio signal by using a diffusion amplitude self-encoder, and the second stage generates music using a text-conditional latent space diffusion model. In addition, Moûsai achieves a significant breakthrough in computational efficiency, enabling real-time extrapolation on a consumer-grade graphics processor while maintaining high sound quality and long temporal structural integrity. Recently, Li et al. (2024) [50] proposed JEN-1. Based on a diffusion model, JEN-1 can handle multiple types of tasks (music generation, music repair, music continuation, etc.), improving the multitask generalization of music generation models. Also, similar to Noise2Music, JEN-1 processes raw waveform data directly, avoiding the loss of fidelity associated with conversion to spectral formats, and generating 48 kHz stereo music. JEN-1 incorporates both autoregressive and non-autoregressive structures. The autoregressive mode helps to capture the time-series dependence of music, while the non-autoregressive mode accelerates the process of sequence generation. This hybrid mode overcomes the limitations of a single mode.

Another part of the research explored the application of language models to generative tasks. Almost simultaneously, Agostinelli et al. (2023) [51] proposed MusicLM, which introduces a hierarchical sequence-to-sequence autoregressive modelling approach. This model extends AudioLM to include three levels of language models, namely semantic, coarse acoustic, and fine acoustic and is able to generate musical audio at 24kHz. MusicLM addresses the problem of paired audio-text data scarcity by combining MuLan [52] (an embedding model that unites music and text). In addition, MusicLM demonstrates its potential for melodic transformation, being able to stylize a

[5] https://mubert.com/

hummed or whistled melody based on a prompt. Considering the advantages of language modeling and diffusion modeling, Lam et al. (2023) [53] proposed MeLoDy, an approach that combines the language model with the diffusion model. MeLoDy is an LM-guided diffusion model. It uses the Dual Path Diffusion (DPD) model and an audio VAE-GAN to decode semantic tokens for the fast generation of musical waveforms. The DPD model effectively incorporates semantic information into the underlying representation passages in the denoising step while handling coarse-grained and fine-grained acoustic features. While MeLoDy continues to use the top-level language model in MusicLM for semantic modeling, it significantly reduces the number of forward passes in MusicLM. As well as improving generation efficiency, MeLoDy maintains musicality and text relevance comparable to MusicLM and Noise2Music, and exceeds the baseline model in terms of audio quality.

Previous models suffer from the limited size of the music dataset, copyright infringement, and plagiarism. To address these problems, Chen et al. (2023) [54] proposed MusicLDM, which aims to address this challenge. Based on the AudioLDM architecture, MusicLDM introduces a beat-synchronized Mixup strategy to enhance the novelty of text-to-music generation. The mixup strategy is a method that restructures existing training samples through linear interpolation, whereby it can augment the training dataset. This approach facilitates MusicLDM to learn via interpolation among training samples rather than simply memorizing a single training instance. Consequently, it helps to reduce overfitting resulting from the limited size of the dataset and reduces the risk of plagiarism in the generated content.

Traditional multi-stage music generation methods usually rely on cascading of multiple models or upsampling steps. This not only increases the complexity of the system, but also imposes a high computational overhead. Copet et al. (2023) [55] proposed MusicGen. MusicGen moves from the traditional multi-stage generation approach to a single autoregressive Transformer decoder, which can simultaneously operate multiple parallel streams of music representations by efficiently interleaving compressed discrete music representations (i.e., music tokens). This approach not only simplifies the music generation process, but also significantly reduces the computational costs while maintaining high quality music output. Notably, unlike MusicLM [51] which relies on supervised data, MusicGen can control melody through unsupervised data.

From the above studies, diffusion models and language models have demonstrated their powerful capabilities in music generation. Diffusion models, with their unique noise diffusion and denoising process, have achieved remarkable results in music generation tasks and can generate high-quality and diverse musical works. Language models, based on their mature application in natural language processing, enable effective modeling and generation of music signals by mapping audio signals to discrete representations. Both models can generate music based on textual descriptions, and control such musical attributes as style, melody, rhythm, etc., demonstrating a high degree of controllability and flexibility in music generation. Relevant studies are summarized in **Table 6**.

4.2.2. Singing Voice Synthesis

Singing voice synthesis (SVS) refers to the synthesis of a singing voice according to lyrics and musical scores with the help of speech synthesis techniques. Compared with traditional music generation tasks, SVS is a relatively independent research field because it involves more digital signal processing techniques and audio sampling synthesis techniques. Text-based singing voice generation mainly refers to providing lyrics to generate singing voices. The technical basis of this task is Text-to-speech. Like Text-to-speech, the mainstream task is divided into three types: splicing synthesis, statistical parameter synthesis, and the current popular neural network synthesis method.

1. **Splicing Synthesis**

Splicing synthesis first requires creating a sound inventory containing a large number of short vocal units. Then, based on the features of the target vocal, such as pitch, duration, and timbre, the unit with the smallest distance from the target unit is selected for splicing. The duration and pitch of the selected units are then adjusted to match the melody and tempo of the target voice. As early as

1997, Macon et al. (1997) [56] proposed Lyricos, a song synthesizer extended from a text-to-speech synthesizer based on the unit concatenation. Since then, a large number of studies have been modeled on this framework, which has developed song synthesis systems of various languages. A successful business case is the Vocaloid [57] software released by Yamaha in 2003. This software uses this Splicing synthesis method. Since then, many companies have used Vocaloid as the engine to launch a series of virtual singers, such as Hatsune Miku and LuoTianyi.

Since splicing synthesis synthesizes a song by recording, arranging, and splicing different pronunciations, it has the advantages of a wide range of sounds and a high degree of editorial freedom. However, the method relies heavily on pre-recorded sound libraries, which are expensive to acquire, label, and train; secondly, when splicing different audio segments, the transition between neighboring segments can lead to artifacts at the splices if not handled properly; and finally, It is difficult for the model to generate pitch variations or articulation styles beyond the range of the training data, which limits the effectiveness of the generated singing voice.

2. **Statistical Parameter Synthesis**

Hence, statistical parameter-based synthesis methods have come into being. Saino et al. (2006) [58] extended the application of HMMs in speech synthesis research to song synthesis. In speech synthesis, HMM attaches importance to precisely quantizing time-series variations of speech features into specific statistical parameters. The model treats the textual information as an observable outcome with the acoustic features as its hidden states. The model aims to accurately map from textual to acoustic information through these statistical parameters. When applied to vocal synthesis, HMM needs to record a large number of vocal clips of the same singer and then refine the acoustic feature parameters (e.g., pitch, duration, resonance peaks, etc.) for vocal synthesis through its modeling; finally, the sequence of acoustic features is converted into an audio signal through a vocoder to realize the synthesis of the vocals.

 doi:10.20944/preprints202502.1791.v1

Table 6. Instrumental Music Generation Tasks.

Task Type	Model Name	Year	Music Representation	Model Architecture	Description	Large Model Relevance	Dataset Name	Generated Music Length	Accessed Link
Commercial instrumental music Generation	Mubert	/	Waveform	/	Tag-based control, Music segment combination	/	/	Varied lengths	https://mubert.com/
Description-based instrumental music Generation	JEN-1 [50]	2024	Waveform	Diffusion	Omnidirectional Diffusion Models,Hybrid AR and NAR Architecture, Masked Noise Robust Autoencoder	/	Pond5, MusicCaps	Varied lengths	https://jenmusic.ai/audio-demos
	MusicLDM [54]	2024	Mel-Spectrogram	Diffusion	Beat-synchronous mixup,Latent Diffusion,CLAP,AudioLDM	Trained on Broad Data at Scale	Audiostock dataset, 2.8 Million text-audio pairs	Varied lengths	https://musicldm.github.io/
	Moûsai [49]	2023	Waveform (48kHz@2)	Diffusion	Latent Diffusion, 64x compression	/	TEXT2MUSIC Dataset	Multiple minutes	https://bit.ly/audio-diffusion
	MusicGen [55]	2023	Discrete tokens (32kHz)	Transformer	Transformer LM,Codebook Interleaving Strategy	Trained on Broad Data at Scale	Internal Dataset,ShutterStock,Pond5,MusicCaps	<= 5 minutes	https://github.com/facebookresearch/audiocraft
	MeLoDy [53]	2023	Waveform (32kHz)	Diffusion& VAE-GAN	Dual-path diffusion, language model,Audio VAE-GAN	Trained LLaMA for semantic modeling	257k hours of music	10s - 30s	https://efficient-melody.github.io/
	MusicLM [51]	2023	Waveform (24kHz)	Transformer	Based on AudioLM,multi-stage modeling, MuLan	Optimize using pre-trained models Mulan and w2v-BERT	MusicCaps (280k hours)	Multiple minutes	https://google-research.github.io/seanet/musiclm/examples/
	Noise2Music [48]	2023	Spectrogram and Waveform(better)	Diffusion	Cascading diffusion,1D Efficient U-Net	Using for Description for Training Generation and Text Embedding Extraction	MuLaMCap (150k hours)	30 seconds	https://google-research.github.io/noise2music
	Riffusion [24]	2022	Spectrogram	Diffusion	Tag-based control, Music segment combination	/	/	≈10 seconds	https://www.riffusion.com/

The statistical parameter synthesis technique significantly reduces the labor cost in singing voice synthesis compared to the traditional sample splicing method providing more stable and consistent results. This technology has become the basic framework of the current research on singing voice synthesis. However, constrained by statistical laws, statistical models have limitations in capturing complex pitch and rhythmic variations.

3. Neural Network Synthesis

With the development of neural networks, some studies have begun to apply neural networks to singing voice synthesis. Nishimura et al. (2016) [59] proposed a DNN-based singing voice synthesis method. Since singing voice synthesis considers more contextual factors than standard TTS synthesis, DNN is used to represent the mapping function from contextual features to acoustic features. Compared to HMM, DNN can better handle complex contextual factors. To address the problem of pitch context sparsity, singing voice synthesis employs note-level pitch normalization and linear interpolation techniques to improve the accuracy of F0[6] prediction. In the subjective listening test, this system significantly outperforms the HMM-based system. Based on similar neural network frameworks, song synthesis techniques based on various types of neural networks, such as CNN [60], LSTM [61], GAN [62], etc., have been born since then.

XiaoiceSing [63] is one of the earliest commercially deployed SVS systems driven by deep learning. The system is built on the main architecture of FastSpeech [64] and makes specific adjustments to adapt to singing synthesis tasks. To avoid the out-of-tune issue, XiaoiceSing adds a residual join to the F0 prediction to make the predicted pitch more accurate. In addition, to improve rhythm, XiaoiceSing, in addition to the duration loss of each phoneme, calculates the total durations all phonemes make up a note to take. Using WORLD [65] as a vocoder, XiaoiceSing is able to ensure that the input F0 contour is consistent with the F0 contour in the generated vocals, ensuring a high level of quality and consistency. During this period, research on using Transformers and WORLD vocoders has been springing up [66–68]. In order to overcome the limitation of the sampling rate, Chen et al. (2020) [69] proposed HiFiSinger. It replaces WORLD with a parallel WaveGAN [70] to generate waveforms at a high-fidelity 48kHz sampling rate, although it utilizes the same FastSpeech-based acoustic model as XiaoiceSing. WaveGAN, unlike WaveRNN, can generate a more realistic audio waveform through a discriminator.

In addition to the FastSpeech architecture, Tacotron [71] is also widely used for vocal synthesis tasks, with a greater focus on generative detail and expressiveness. Gu et al. (2020) [72] proposed ByteSing, which combines the advantages of a Tacotron-like architecture with the neural vocoder of WaveRNN. Neurovocoders are capable of capturing and reproducing more complex acoustic features. This high-fidelity generative capability is much better than the generative ability of conventional vocoders. ByteSing employs an autoregressive decoder to convert the input features (extended by duration information) directly into Mel spectrograms which are synthesized into waveforms by the vocoder. By using attention-based alignment and the encoder-decoder framework, ByteSing effectively manages long-range dependencies and detailed acoustic feature modeling. Auxiliary phoneme duration prediction models are added to enhance ByteSing's ability to handle the complex temporal nuances inherent in singing. The system is capable of a guaranteed sampling rate of 24kHz.

As diffusion models are demonstrating enormous potential in generative tasks, Diffsinger proposed by Liu et al. (2021) [73], also based on FastSpeech, employs a denoising diffusion probabilistic model to transform generative tasks into parametric Markov chains conditioned on musical scores. The model adds noise to the Mel spectrogram through a diffusion process until it becomes Gaussian, and gradually restores the Mel spectrogram during denoising. In order to improve sound quality and inference speed, Diffsinger introduces a shallow diffusion mechanism and utilizes prior knowledge acquired from simple loss to reduce inference steps, allowing the model

[6] F0: The fundamental frequency of pitch, commonly used to describe the pitch of a sound.

to close to a real-time generation.The techniques mentioned above all rely on large databases, so studies aiming to reduce data consumption are cropping up, such as LiteSing [67], Sinsy [74], etc.

As traditional SVS techniques employ a two-stage generation approach, independent training of the acoustic model and vocoder may result in mismatches. However, VIsinger [75] and VISinger 2 [76] proposed by Zhang et al. have significantly reduced these mismatches. They have successfully applied end-to-end speech synthesis techniques to song synthesis and generated song audio directly from lyrics and music scores.

This method operates on the main architecture of VITS (Variational Inference with adversarial learning for end-to-end Text-to-Speech) [77]. It means that VITS uses a combined end-to-end speech synthesis model that incorporates VAE, normalizing flow, and GAN to improve the encoder following singing characteristics. While modeling the acoustic variations in singing, VITS introduces an F0 predictor to obtain stable singing performance. This system also optimizes rhythm, modifying the traditional duration prediction to the duration ratio of phonemes to notes. Introducing the VIsinger series takes singing synthesis to a new end-to-end model.

The neural network synthesis approach simplifies the system architecture. Firstly, it generates high-quality singing audio from text through an end-to-end modeling approach. Secondly, deep learning models enable the learning of complex acoustic feature representations, generating high-fidelity singing voices. Finally, the improved architectural and training technique can improve computation efficiency, making real-time generation possible and supporting multi-modal information fusion.

Singing voice synthesis technology has undergone three stages splicing synthesis, statistical parameter synthesis, and neural network synthesis. Great Changes have occurred in the representation of acoustic features, model structure, and other aspects. These techniques have improved the naturalness of synthesized singing and enabled the system to better capture pitch changes and rhythms, generating more vivid and realistic singing audio. Future research will continue to explore new ways to reduce data requirements, increase synthesis speed, and enhance model generalization capabilities. Relevant studies are summarized in **Table 7**.

 doi:10.20944/preprints202502.1791.v1

Table 7. Singing Voice Synthesis Tasks.

Task Type	Model Name	Year	Music Representation	Model Architecture	Keywords	Dataset Name	Accessed Link
Commercial Singing Voice Engine	ACE Studio	2021	/	/	AI synthesis,Auto pitch	/	https://acestudio.ai/
	Synthesizer V Studio	2018	/	/	WaveNet vocoder,DNN,AI synthesis	/	https://dreamtonics.com/synthesizerv/
	Vocaloid	2004	Waveform&Spectrum	/	sample concatenation	/	https://www.vocaloid.com/
Singing Voice Synthesis	VISinger 2 [76]	2022	Mel-Spectrogram	VAE+DSP	conditional VAE,Improved Decoder, Parameter Optimization, Higher Sampling Rate(Considering to VISinger)	OpenCpop	https://zhangyongmao.github.io/VISinger2/
	VISinger [75]	2022	Mel-Spectrogram	VAE	end-to-end solution, F0 predictor, normalizing flow based prior encoder and adversarial decoder	4.7 hours singing dataset with 100 songs	https://zhangyongmao.github.io/VISinger/
	DiffSinger [73]	2021	Mel-Spectrogram	Diffusion+Neural Vocoder	Shallow diffusion mechanism, parameterized Markov chain, Denoising Diffusion Probabilistic Model, FastSpeech	PopCS	https://diffsinger.github.io
	HiFiSinger [69]	2020	Mel-Spectrogram	Transformer + Neural Vocoder	Parallel WaveGAN (sub-frequency GAN+multi-length GAN), FastSpeech	Chinese Mandarin pop songs	https://speechresearch.github.io/hifisinger/
	ByteSing [72]	2020	Mel-Spectrogram	Transformer+Neural Vocoder	WaveRNN, Auxiliary Phoneme Duration Prediction model, Tacotron	90 Chinese songs (MusicXML[1])	https://ByteSings.github.io
	XiaoiceSing [63]	2020	Acoustic parameters	Transformer + WORLD	integrated network, Residual F0, syllable duration modeling, FastSpeech	Mandarin pop songs	https://xiaoicesing.github.io/
	[59]	2016	Acoustic parameters	DNN	musical-note-level pitch normalization, linear-interpolation	70 Japanese children's songs (female)	/
	[58]	2006	Acoustic parameters	HMM	Context-dependent HMMs, duration models, and time-lag models	60 Japanese children's songs (male)	https://www.sp.nitech.ac.jp/~k-saino/music/
	Lyricos [56]	1997	Waveform	Sinusoidal model	ABS/OLA sinusoidal model, vibrato, phonetic modeling	ten minutes of continuous singing data	/

[1] MusicXML is a standard XML-based file format for representing sheet music and music information.

4.2.3. Complete Song Generation

A song is a combination of vocals and accompaniment. Complete song generation synthesizes research on pure music generation and song synthesis with the goal of automating the creation of complete songs. This generation task is a multi-modal one involving multiple types of generation tasks. It requires not only synthesizing the corresponding accompaniment based on the textual content but also generating matching lyrics, vocals, etc.

Jukebox [78], proposed by the OpenAI team, is among the first to explore complete song generation. Rather than generating complete songs based entirely on text, Jukebox generates complete songs by modeling the raw audio domain while providing a way to use lyrics to control the generated content. It uses a multi-layered VQ-VAE architecture capable of compressing audio into discrete spaces while retaining as much musical information as possible. Jukebox uses an encoder-decoder model to implement conditional control of lyrics and uses the NUS AutoLyricsAlign tool to align lyrics and music. In addition to lyrics, Jukebox also allows users to control artists and genres.

Hong et al. (2024) [79], for the first time, proposed Text-to-song which incorporates both vocal and accompaniment generation. They developed Melodist, a two-stage text-to-song method. Melodist generates singing voice synthesis (SVS) first, and then vocal-to-accompaniment (V2A) synthesis based on SVS. Finally, Melodist mixes SVS and V2A together to form a complete song. In the vocal-to-accompaniment synthesis stage, the Melodist adopts the tri-tower contrastive pre-trained framework to learn more efficient text representations and jointly embeds text, vocals, and accompaniment into an aligned space, which enables the model to control accompaniment generation by using natural language cues. The experiment shows that the outputs generated by the Medodist model achieve better performance in terms of subjective and objective metrics assessment, as well as text consistency. However, as the results generated rely on the quality of the source separation, the method still has limitations—it cannot achieve an end-to-end generation. On top of this, this method also sees the accompaniment as a piece of music, ignoring the complex combinations between instrumental tracks.

As a representative of business projects for text-to-song tasks, Suno[7] is currently one of the most influential software. It is capable of generating complete songs with lyrics end-to-end via natural language descriptions, or it can use natural language descriptions to control the generation of accompaniment on the condition that lyrics are provided. It uses heuristics for audio tokenization and the transformer architecture , but it is an unofficial open source project now[8]. The team's other open source project is a text-to-audio generation model called Bark[9], which is capable of generating near-human-level speech and can be used to generate music by adding tokens. This project's excellence in text-to-audio generation also laid the groundwork for the creation of Suno.

Recently, the ByteDance team proposed Seed-Music [80], a multi-modal music generation large model. This is a comprehensive framework designed to generate high-quality music through fine-grained style control. It integrates autoregressive language modeling and diffusion methods to support two key workflows: controlled music generation and post-editing. The controlled generation workflow harmonically unified vocals and accompaniment (accompaniment in midi format) to be created through multimodal inputs (e.g., lyrics, stylistic descriptions, audio references, scores, and voice cues), providing a high degree of customization and adaptability. For another thing, post-production editing features enable users to interactively modify elements of existing music tracks, including vocal lyrics, melody, and timbre.

Currently, the research on text-to-music generation has expanded from pure audio generation to more complex and comprehensive tasks, and complete song generation is a great challenge with a

[7] https://suno.com/

[8] Relevant content referenced from the podcast: https://www.latent.space/p/suno

[9] https://github.com/suno-ai/bark

generation process that integrates various types of tasks. In the future, with the development of multimodal large models as well as generative models, it is possible to provide richer contexts and details for song generation, and to further improve the quality as well as the diversity of generation. Relevant studies are summarized in **Table 8**.

Table 8. Complete Song Generation Methods.

Task Type	Model Name	Year	Music Representation	Model Architecture	Keywords	Large Model Relevance	Dataset Name	Generated Music Length	Accessed Link
Text-to-Song Generation	Seed-Music [80]	2024	Waveform & MIDI	Transformer & Diffusion	Multi-modal Inputs, Auto-regressive Language Modeling,Vocoder Latents,Zero-shot Singing Voice Conversion	Large multi-modal language models for understanding and generation	Not mentioned	Varied Length	https://team.doubao.com/en/special/seed-music
	Melodist [79]	2024	Waveform	Transformer	Tri-Tower Contrastive Pre-training, Cross-Modality Information Matching, Lyrics and Prompt-based	Using LLM to generate natural language prompts	Chinese song datasets and Open-Source Datasets	Not mentioned	https://text2songMelodist.github.io/Sample/
	Jukebox [78]	2020	Waveform	VQ-VAE+ Transformer	Multiscale VQ-VAE, Autoregressive Transformer, Conditional Generation, Hierarchical Modeling	Trained on Broad Data at Scale	1.2 million songs with lyrics	Multiple minutes	https://jukebox.openai.com/
Commercial Complete Song Generation	Suno AI	2023	Waveform	Transformer	Heuristic method, Audio Tokenization, Zero threshold for use	/	Singing audio and non-singing audio	<= 4 minutes	https://alpha.suno.ai/

4.3. Comments on Existing Techniques

The field of text-to-music generation has made great progress over the years, with advancements in rule-based systems, statistical models, generative approaches, and large language models (LLMs). However, each method has its strengths and limitations. Challenges such as data dependency, model controllability, and generalization remain significant. This section reviews these techniques and highlights the key issues that require attention.

1. **Rule-Based and Template Methods**

Rule-based and template-driven methods are among the earliest approaches in text-to-music generation. These methods follow predefined musical rules, such as chord progressions or rhythmic patterns, to generate melodies. Their simplicity and reliability make them highly interpretable and consistent, useful in structured applications like educational tools or composition guides. However, their deterministic nature severely limits their ability to adapt to the diversity and complexity of real-world music. For example, such methods struggle to create dynamic and expressive outputs when handling lyrics with varying emotional tones. While they are useful for tasks requiring fixed structures, their lack of creativity makes them unsuitable for tasks that demand innovation and diversity.

2. **Statistical Models**

Statistical approaches, such as Markov chains and n-gram models, introduced randomness that improved over rule-based systems. By analyzing patterns in training data, these models can generate melodies that exhibit some variability and complexity. However, they cannot capture long-term dependencies, which are crucial for creating coherent music. For instance, while a Markov chain might produce locally plausible note sequences, it often fails to generate melodies with meaningful global structures. Additionally, statistical models are highly dependent on the quality of the training data, often overfitting to specific patterns and failing to generalize to new musical styles or datasets. This limits their ability to produce diverse and innovative outputs across broader applications.

3. **Generative Models**

Generative models represent a significant leap in text-to-music generation, offering powerful tools for creating realistic, diverse, and dynamic outputs. GANs are one of the most prominent methods, leveraging an adversarial framework where a generator produces melodies and a discriminator evaluates their quality. Models like LSTM-GANs combine sequential modeling with GANs to improve the coherence of generated music. However, GANs often face issues such as mode collapse, where the generator produces limited variations, and instability during training, making it challenging for them to optimize effectively. Furthermore, GANs generally lack fine control over specific musical attributes, which limits their use in tasks requiring precise alignment with textual inputs.

In addition to GANs, Variational Autoencoders (VAEs) have gained attention for their ability to learn structured latent representations of music. VAEs map input data (e.g., melodies) to a latent space, allowing for smooth interpolation between musical features and generating new, coherent outputs. Their probabilistic framework ensures stable training and enables control over the diversity of generated melodies. However, VAEs tend to produce less sharp or vivid outputs than GANs, which can affect the perceptual quality of the music.

More recently, diffusion models have emerged as a powerful alternative in generative tasks, including music generation. These models progressively transform noise into structured outputs through a reverse diffusion process, guided by a learned probability distribution. Diffusion models excel at generating high-quality outputs with precise control over attributes, making them suitable for tasks that require both diversity and coherence. For example, text-to-music diffusion models can effectively map lyrics to melodies by capturing nuanced relationships in a step-by-step generation process. While these models are computationally intensive and require careful tuning, they have demonstrated significant potential in addressing the limitations of earlier generative approaches.

4. **Transformer-Based Architectures**

The introduction of Transformer-based models has significantly advanced music generation by addressing many of the limitations of earlier methods. Transformers excel in modeling long-range dependencies and capturing intricate relationships between musical elements, such as aligning lyrics with melodies. Models like SongMASS leverage these capabilities by using separate encoders and decoders for lyrics and melody, combined with pre-training techniques to improve generation quality. These models effectively handle the complexity of musical structures by focusing on parallel processing and self-attention mechanisms, enabling more coherent and contextually aligned outputs.

In addition to task-specific Transformer models, language models based on the Transformer architecture have been adapted for music generation tasks. These models learn representations of sequences, whether text or symbolic music, and can generate music by treating it as a language. For instance, in some approaches, musical notes and rhythms are tokenized into sequences akin to words in natural language, allowing the models to predict the next note or phrase based on the preceding context. This adaptation of language models provides a flexible and scalable framework for melody generation, where the system can leverage transfer learning from vast datasets of natural language or symbolic music.

Despite these strengths, Transformer-based architectures and language models share several limitations. They are computationally intensive, requiring large amounts of memory and processing power. Their performance is also highly sensitive to the quality and diversity of training data, which can restrict their ability to handle underrepresented musical styles or genres. Additionally, self-attention mechanisms, while powerful, may struggle with extremely long sequences, a common challenge in music generation tasks. To address these issues, some researchers have introduced hierarchical Transformer architectures that split sequences into smaller, more manageable chunks, improving computational efficiency without sacrificing output quality.

Integrating language models into music generation represents a bridge between natural language processing and musical creativity. While these models effectively handle sequential data and enable creative outputs, they often require fine-tuning and additional preprocessing to adapt to the unique characteristics of music data, such as time signatures and harmonic progressions. Despite these challenges, language models have opened new possibilities for using textual descriptions to guide music generation, providing a foundation for more advanced systems.

5. **Large Language Models**

Large language models, such as SongComposer, represent the latest advancements in text-to-music generation. LLMs are highly flexible and capable of learning complex patterns from vast datasets, enabling them to produce high-quality music that aligns closely with textual inputs. Unlike traditional methods that rely on fixed rules or templates, LLMs learn to infer relationships between lyrics and melodies in a data-driven manner. This allows them to handle tasks like melody generation, lyric-melody alignment, and even song continuation with impressive results. However, LLMs also face challenges. Their reliance on massive computational resources makes them less accessible for smaller-scale applications. Additionally, they lack fine-grained control over specific musical attributes, which can lead to outputs that deviate from user expectations. The “black-box” nature of LLMs also makes it difficult to interpret their decisions, which can be problematic in applications requiring transparency or adherence to strict musical guidelines.

A key challenge across all methods is balancing creativity and control. Rule-based and statistical methods excel in providing structure and interpretability but fail to produce diverse and expressive music. On the other hand, generative models, such as GANs, VAEs, and diffusion models, along with LLMs, offer unparalleled creativity and flexibility but often lack fine control over the outputs. Another significant issue is the dependency on high-quality datasets. Many models require extensive, diverse, and well-annotated training data to perform well, yet such datasets are often scarce, especially for underrepresented musical styles or languages. This limitation hinders the generalization of these models to broader and more diverse applications.

To address these challenges, future research should focus on hybrid approaches that combine the strengths of different methods. For example, integrating rule-based templates with Transformer-based architectures could provide better control over specific musical features while retaining the flexibility of deep learning models. Similarly, LLMs could be enhanced with interpretable mechanisms or user-guided controls to improve alignment with specific requirements. Exploring self-supervised learning and transfer learning techniques could also help mitigate the dependency on large labeled datasets, making models more versatile and adaptable across diverse scenarios.

5. Frameworks

Text-to-music generation frameworks are evolving rapidly, driven by advancements in large language models (LLMs). This chapter categorizes existing approaches into three paradigms based on their integration of LLMs: Traditional Rule-Driven Frameworks, Hybrid LLM-Augmented Frameworks, and End-to-End LLM-Centric Frameworks. Each paradigm addresses distinct challenges in semantic-text-to-music alignment, controllability, and scalability, offering unique trade-offs between interpretability and generative flexibility.

5.1. Traditional Learning-Based Frameworks

Traditional learning-based frameworks in text-to-music generation typically rely on machine learning or deep learning models designed for sequence generation. These models treat music and text as sequences, using neural networks to capture relationships between the two modalities. By training on paired datasets of text and music, they aim to generate musical outputs that align with textual inputs. These methods usually employ encoder-decoder architectures or recurrent structures (e.g., LSTM, RNN) to model dependencies within and across the modalities. The general pipeline for traditional methods in text-to-music generation can be divided into three main stages:

- **Text Encoding:** The input text (e.g., lyrics) is converted into numerical representations using embedding layers, capturing semantic and rhythmic information.
- **Sequence Generation:** Deep learning models (e.g. LSTM, RNN) generate musical sequences (e.g., melody, chords, or rhythm) based on the encoded text.
- **Output Synthesis:** The generated musical sequences are converted into symbolic music formats (e.g., MIDI) or synthesized into audio.

Case 1: LSTM-GAN

Yu et al. (2021) [30] proposed a conditional LSTM-GAN model for lyric-to-melody generation, enhancing the traditional pipeline with adversarial training. The model encodes lyrics into syllable-level embeddings, capturing both semantics and rhythm. These embeddings serve as inputs for an LSTM-based generator, which creates musical sequences, introducing variability through noise vectors for increased creativity. A discriminator then evaluates the generated melodies, guiding the generator to produce more realistic and contextually aligned outputs. This combination of sequence generation and adversarial training improves the diversity and quality of the music, retaining the core structure of text-to-music generation.

Traditional text-to-music generation methods, including models like Yu's LSTM-GAN, have several key characteristics. One notable strength is their ability to capture long-range musical dependencies, ensuring that the generated melodies maintain coherence over time. The use of adversarial training further improves the diversity and realism of the outputs, moving beyond the deterministic nature of early models. However, these improvements come with certain trade-offs. For example, the reliance on large, high-quality paired datasets makes the model highly data-dependent, limiting its ability to generalize across different musical styles or genres. Additionally, while adversarial training encourages creativity, it may limit control over specific musical elements like rhythm or harmony, making it difficult to fine-tune outputs for professional use.

5.2. Hybrid LLM-Augmented Frameworks

Hybrid approaches in text-to-music generation integrate Large Language Models (LLMs) as a core module alongside traditional sequence generation models. In this framework, the LLM plays a versatile role by processing text in various ways, such as extracting musical attributes, generating lyrics, or reconstructing descriptions. The LLM enriches the input text, which is fed into a subsequent music generation model (e.g., LSTM, Transformer) to produce musical outputs. By acting as a powerful intermediary, the LLM helps bridge the gap between complex textual input and the generated music, ensuring better alignment and context preservation. The general pipeline for hybrid approaches can be summarized in the following stages:

- **Text Encoding:** With traditional methods.
- **LLM Module:** Extracts key semantic features and contextual information from the input text and generates new content, such as lyrics or expanded descriptions, based on the input.
- **Sequence Generation:** With traditional methods.
- **Output Synthesis:** With traditional methods. Sometimes LLM is used to give feedback.

Case 2: MuseCoCo

MuseCoCo [15] is an innovative hybrid model for generating music from text descriptions. It combines the power of pre-trained language models (LLMs) with traditional sequence generation models to enhance text-to-music generation. In the MuseCoCo system, templates are pre-prepared

for the LLM. For example, a template could be: "The music is imbued with [EMOTION]". When the input prompt is "write a happy four-beat pop song", the LLM extracts the relevant attributes, such as emotion and time signature, and refines the template by filling in values. This results in a description like: "The music is imbued with [happiness] and the [4/4] time signature is used in the music. The genre of the music is [pop]." These templates guide the generation process, ensuring the music aligns with the user's description.

While integrating the LLM provides greater flexibility and control over the music generation process, it also introduces an additional layer of complexity. The model's performance depends heavily on the quality of the LLM's text processing and its ability to accurately extract or generate relevant musical attributes. Moreover, the system's effectiveness is contingent on the quality of the attribute templates used to guide the generation process. Poorly defined or overly rigid templates can limit the system's creativity and adaptability, preventing the generation of truly innovative or diverse music.

5.3. End-to-End LLM-Centric Frameworks

End-to-end large language model (LLM) systems treat music as a second language, applying sequence-based models, typically used in natural language processing (NLP), to generate music directly from text. In this approach, the LLM processes textual input (e.g., lyrics, prompts, or descriptions) and generates corresponding musical elements (such as melody, rhythm, and harmony), considering these elements as analogous to linguistic structures like words and sentences. This eliminates the need for separate music theory modules or templates, offering a unified framework for text-to-music generation. The general pipeline for end-to-end LLM-based systems in text-to-music generation consists of the following stages:

- **Text Encoding:** With traditional methods.
- **LLM Processing:** The encoded text is processed by the language model, which treats music as a sequence similar to text. The model predicts the next musical element (e.g., note, rhythm, or harmony) based on the current context, generating a complete musical sequence in an iterative manner. In this stage, the LLM is able to use its extensive pre-trained knowledge of language and patterns to generate musically coherent sequences that align with the input description.
- **Output Synthesis:** Extract symbol information from textual music sequences and synthesize them.

Case 3: SongComposer

SongComposer [36] is a specialized LLM for generating lyrics and melodies directly from textual input. It is trained using a high-quality lyrics-melody pairing dataset, which fine-tunes the LLM to understand the relationship between lyrics and melody more effectively. This fine-tuning step, along with the introduction of innovative encoding rules, enables the model to process melody sequences, ensuring that the generated music is both contextually appropriate and musically coherent. SongComposer operates by accepting a text description, generating both the lyrics and matching melody, including information like note pitch, duration, and rest duration. This dual output facilitates the generation of complete musical pieces and allows the extracted information to be used for further music creation.

LLM-based systems for text-to-music generation offer creativity and flexibility, as they can generate diverse and contextually relevant music across a wide range of genres, styles, and emotional tones. By integrating text processing and music generation into a single framework, these systems ensure a seamless alignment between the input text and the generated music. However, there are limitations, particularly in the lack of fine-grained control over musical features such as tempo, dynamics, and instrumentation. While LLMs can create highly creative and musically coherent compositions, achieving precise control over these elements is difficult. Additionally, these models are computationally intensive, requiring significant resources for both training and inference. Their performance is also highly dependent on the quality and diversity of the training data, making them less effective for underrepresented genres or musical styles.

5.4. Comparative Analysis and Limitations

Table 9 provides a comparison and analysis of the three mainstream frameworks. The metrics used for evaluation are explained as follows.

- **Creativity:** The ability to generate unique and diverse outputs.
- **Control over Output:** The level of control a user has over specific aspects of the generated music (e.g., tempo, harmony).
- **Data Dependency:** The reliance on high-quality labeled datasets for training the model. (More stars mean low dependency.)
- **Generalization:** The ability to adapt across different genres and tasks.
- **Training Complexity:** The computational cost and difficulty of training the model. (More stars mean low Complexity.)
- **Output Quality:** The coherence, relevance, and alignment of generated music with the text input.

Table 9. Comparison of the three mainstream frameworks.

	Creativity	Control over Output	Data Dependency	Generalization	Training Complexity	Output Quality
Traditional Methods	★★	★★★	★★	★★	★★★★★	★★★
Hybrid Approaches	★★★★	★★★★	★★★	★★★	★★★	★★★★
End-to-End LLM Systems	★★★★★	★★★★★	★★★★	★★★★★	★★	★★★★★

The table compares three mainstream frameworks for text-to-music generation—Traditional Methods, Hybrid Approaches, and End-to-End LLM Systems—across six key aspects, highlighting their strengths and limitations.

Traditional Methods primarily rely on models like LSTMs, RNNs, and VAEs, which follow a linear pipeline from text encoding to sequence generation and music synthesis. While these methods provide high output quality and control through high-quality paired datasets, their creativity and generalization capabilities are moderate, constrained by pre-defined patterns and rules. Additionally, traditional methods are less adaptable to varied tasks and rely heavily on labeled data for effective training, though their training complexity remains relatively low.

Overall, the table illustrates the trade-offs between these frameworks, with traditional methods excelling in simplicity and control, hybrid approaches balancing flexibility and complexity, and end-to-end LLM systems pushing the boundaries of creativity and generalization at the expense of control and resource efficiency.

6. Challenges and Future Directions

6.1. Challenges

6.1.1. Technical Level

Although breakthroughs have been made, text-to-music generation tasks still face the following technical challenges.

1. **Dataset Scarcity and Representation Limitations**

High-quality datasets are the foundation for training effective text-to-music generation models. However, current datasets often lack diversity in musical styles and emotional expressions, resulting in generated outputs that are overly homogeneous. Furthermore, the accuracy of dataset labeling directly impacts model training, as incorrect labels may mislead models into learning faulty musical patterns. Large-scale datasets, essential for training complex models, also pose significant challenges in terms of data collection, processing, and representation. Symbolic datasets (e.g., MIDI) may fail to capture the expressive nuances of music, while audio-based datasets are computationally demanding

and challenging to align with textual semantics. Addressing these limitations requires innovative approaches to dataset design, multi-modal alignment, and data augmentation.

2. **Model Training and Generalization**

The generalization ability of current models remains a key limitation, especially when dealing with unseen data. Many existing systems struggle to produce coherent and contextually appropriate music outside their training data distribution. Moreover, training large-scale models demands extensive computational resources, which limits accessibility for researchers and developers. Additionally, model interpretability is a significant concern; understanding how models make decisions during music generation is crucial for improving their performance and providing more guided outputs. Enhancing interpretability can also aid in debugging and refining models to better align with the intended tasks.

3. **Evaluation Metrics for Creativity**

The limitation in model generalization is a key factor restricting the creativity of generated outputs. Creativity inherently involves successful extrapolation beyond the dataset distribution, whereas current machine learning methods mainly address interpolation rather than extrapolation [81,82]. Models need to strike a balance between imitating existing musical styles and generating novel music. Additionally, the lack of effective methods for quantifying and evaluating musical creativity limits objective assessments of innovation in generated music.

4. **Song Structure and Long-Term Coherence**

Music often relies on complex short-term and long-term structures, such as the verse-bridge-chorus format in popular music or the thematic development in classical compositions. Capturing and generating such structures poses a significant challenge, particularly for long-sequence modeling tasks. Current models struggle to simultaneously manage local coherence (e.g., smooth transitions between notes or measures) and global structure (e.g., thematic development across an entire piece). Achieving this balance requires advanced techniques that can effectively handle hierarchical dependencies in musical compositions [18].

5. **Emotion Representation and Modeling**

Although emotion is a vital component of music, representing and modeling emotion poses a complex challenge. The limitations of existing models lie in how to effectively analyze emotional representations in text and model emotional features. Furthermore, the relationship between emotion and musical elements is a complex issue that involves both psychology and musicology, requiring models to understand and leverage these associations to generate music with specific emotional qualities. Only a few studies have addressed the emotional aspect of music [83–88].

6. **Interactivity between Human and Computer**

While end-to-end modeling has enabled systems to generate complete musical compositions seamlessly, there is a growing demand for interactive generation systems. Users often prefer to engage with AI as a "musical partner," adjusting outputs dynamically during the generation process. Existing interactive systems [89,90] have demonstrated promise, but they are far from widespread adoption. Key challenges include designing interfaces that allow intuitive user interaction, enabling real-time feedback without compromising the coherence of the generated music, and addressing the balance between user input and model autonomy. Further exploration of human-AI interaction in the context of music generation is essential to create systems that are not only functional but also user-friendly and adaptable to diverse creative workflows.

6.1.2. Social Level

Due to the unique nature of artistic works, the text-to-music generation faces several social challenges:

1. **Copyright Issues**

The music generation task now faces three main challenges, including the legality of training datasets, originality of generated content, and copyright ownership. The music industry is most concerned that AI learning from songs to generate new content could infringe on the copyrights of original artists [91,92]. Major music companies, such as Universal Music Group, have begun taking steps and demanding that streaming platforms prevent AI tools from scraping lyrics and melodies from copyrighted songs. The Recording Industry Association of America has submitted a list of AI developers to the U.S. government, and filed a lawsuit against AI music companies, aiming to prevent the unauthorized use of copyrighted recordings to "train" generative AI models10. Additionally, the "deep fake" of generated content also deserves attention [93,94], as it poses a serious threat to the originality and personal style of artists.

2. **Privacy Concerns**

Despite years of research in artificial intelligence, privacy concerns remain unresolved(Zhang et al. 2022). Privacy is especially pronounced in singing voice generation. Bai et al. (2024) [80] noted that they have recognized "the singing voice evokes one of the strongest expressions of individual identity". Therefore, it becomes a burning issue to ensure data collection and usage do not infringe on personal privacy in training.

3. **Music Industry Impact**

Music generation has shifted traditional creation methods, posing a potential threat to music producers' livelihoods. The proliferation of low-cost or even free music generation tools is encroaching upon the traditional music market, impacting the structure of the music industry [96].

6.2. Future Directions

Text-to-music generation represents a revolutionary advancement in music creation, applying natural language processing and machine learning techniques to the composition process and opening new pathways for music creation. From early rule-based methods to today's deep learning models, music generation technology has made great strides, enabling researchers to produce music with considerable artistic value and emotional depth. Given the challenges outlined above, several future directions for text-to-music generation are proposed.

1. **Enhancing Data Quality and Diversity**

Future developments will prioritize enhancing data quality and diversity. Building comprehensive music datasets that cover a wider range of styles, genres, and cultures will enable models to learn broader musical characteristics and improve generalization. High-quality datasets enriched with detailed annotations—such as emotional content, structural markers, and performance techniques—will be essential for refining models' learning capabilities. The incorporation of synthetic data generation using LLMs could also serve as a supplementary approach to address data scarcity by creating realistic textual and musical annotations.

2. **Optimizing Training Efficiency**

Another key direction is the optimization of model training methods, which can reduce training time and costs through the application of distributed computing platforms and improve algorithm efficiency to facilitate a more efficient learning process.

3. **Improving the Quality and Personalization**

Future models will increasingly adopt innovative mechanisms to improve the quality and personalization of generated music. Techniques such as attention mechanisms and style transfer will make compositions more adaptable to specific user requirements, producing works that are highly artistic and personalized. For instance, LLMs can play a pivotal role in capturing nuanced textual

[10] News Link: https://www.riaa.com/record-companies-bring-landmark-cases-for-responsible-ai-against suno-and-udio-in-boston-and-new-york-federal-courts-respectively/

inputs—such as emotional tones or thematic details—and translating them into corresponding musical elements, such as melodic contours or harmonic progressions.

4. **Deepening Understanding of Musical Structures**

Model designs will focus on better understanding musical structures, such as segment divisions, motif development, and long-term coherence. With Transformer-based architectures and the use of pre-trained LLMs, future systems will be capable of generating compositions that exhibit complex structures and rich variations, bridging the gap between algorithmic generation and human creativity.

5. **Bridging Music and Emotion**

Emotion modeling will also become a central area of focus, enabling models to generate music that evokes emotional resonance. By integrating LLMs trained on multi-modal data, such as text and audio, future systems will achieve a deeper understanding of the relationships between linguistic expressions and musical emotions. These improvements will empower models to create emotionally expressive compositions that resonate with listeners on a profound level.

6. **Advancing Cross-Modal Music Generation**

The rise of large-scale models will push advancements toward model integration and cross-modal capabilities. LLMs with multi-modal inputs—such as text, images, and video—will pave the way for generating music inspired by diverse input types, significantly expanding the application scenarios of music generation. For example, a model could generate soundtracks for a video or a painting, seamlessly bridging artistic domains and enriching creative workflows.

7. **Establishing Clear Copyright Ownership**

From a social perspective, it is imperative to set forth more clear-cut rules of copyright ownership and record the process of creation with the help of blockchain technology in the future. This initiative will ensure the legality of creations and promote the development of open copyright music databases, encouraging data sharing while protecting artists' rights and interests.

8. **Strengthening Privacy Protection**

In terms of privacy protection, stronger encryption and anonymization of user data will become a defining trend. This means that users' privacy will not be invaded and that specific, transparent service terms will be rolled out to build users’ trust.

9. **Fostering Collaboration Between Technology and Artists**

In the music industry, there will be a greater focus on collaboration between technology platforms and artists, exploring innovative applications, and developing fairer revenue distribution mechanisms to balance the conflicting interests between automated generation and traditional manual creation.

7. Conclusion

This paper provides a comprehensive overview of recent advancements in text-to-music generation, focusing on the classification and methodologies of symbolic and audio domains. It systematically introduces key improvements in text-to-music generation across various tasks (melody generation, polyphony generation, instrumental music generation, singing voice synthesis, and complete song generation). By introducing a taxonomy of text types (lyrics, musical attributes, and natural descriptions) and musical representations (MIDI, spectrograms, ABC notation), we establish a structured framework for evaluating cross-modal alignment challenges.

The primary contribution of this work lies in its critical review and classification of existing frameworks for text-to-music generation. By categorizing approaches into traditional methods, hybrid techniques, and end-to-end large language model (LLM) systems, the paper provides a detailed comparison of their strengths, limitations, and applicability to different tasks. Unlike prior surveys focused on single-modality generation, we highlight how LLMs enhance controllability and

generalization by integrating semantic understanding with musical structure modeling, addressing limitations of rule-based systems in creativity and data-driven models in interpretability.

Key technical challenges are identified, including dataset scarcity for underrepresented genres, long-term coherence in multi-track compositions, and the need for emotion-aware generation. Social challenges, such as copyright ambiguity and AI-generated content originality, are also discussed. Future advancements are expected to improve the quality and diversity of generated music while simplifying the generation process to make it more intuitive and accessible. Key areas for exploration include developing more sophisticated algorithms to better interpret textual semantics, reducing dependence on large labeled datasets through innovative data processing techniques, and enhancing model generalization to produce more creative and personalized outputs. Additionally, integrating multi-modal large models will enable systems to incorporate diverse information sources, such as images, videos, and environmental sounds, fostering the creation of richer and more multidimensional musical experiences.

This work provides a critical roadmap for advancing text-to-music generation by systematizing methodologies, clarifying cross-modal alignment challenges, and highlighting the transformative role of LLMs in enhancing controllability and interpretability. By bridging gaps between semantic understanding and structural modeling, and prioritizing ethical and technical challenges, the paper lays the groundwork for future innovations that balance creativity, technical rigor, and societal impact in AI-generated music, empowering researchers to develop more accessible, diverse, and socially responsible systems.

Author Contributions: Conceptualization, Y.Z and M.Y.; methodology, Y.Z, J.D and M.Y.; formal analysis, Y.L and X.Z.; investigation, F.S and Y.Z.; resources, Y.Z, Z.W and H.N..; data curation, M.Y and F.S.; writing—original draft preparation, Y.Z and Y.L.; writing—review and editing, Y.Z and M.Y.; visualization, Y.Z and M.Y..; supervision, Z.W and H.N.; project administration, Z.W, H.N and J.D.. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable.

Acknowledgments: The authors thank the Cyberspace Data and Intelligence Lab at the School of Computer and Communications Engineering, University of Science and Technology, Beijing, for providing the research facilities.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ji, S.; Luo, J.; Yang, X. A Comprehensive Survey on Deep Music Generation: Multi-Level Representations, Algorithms, Evaluations, and Future Directions 2020.
2. Ma, Y.; Øland, A.; Ragni, A.; Sette, B.M.D.; Saitis, C.; Donahue, C.; Lin, C.; Plachouras, C.; Benetos, E.; Shatri, E.; et al. Foundation Models for Music: A Survey 2024.
3. Briot, J.-P.; Pachet, F. Music Generation by Deep Learning - Challenges and Directions. *Neural Computing and Applications* **2020**, *32*, 981–993, doi:10.1007/s00521-018-3813-6.
4. Ji, S.; Yang, X.; Luo, J. A Survey on Deep Learning for Symbolic Music Generation: Representations, Algorithms, Evaluations, and Challenges. *ACM Computing Surveys* **2023**, *56*, doi:10.1145/3597493.
5. Hernandez-Olivan, C.; Beltran, J.R. Music Composition with Deep Learning: A Review 2021.
6. Civit, M.; Civit-Masot, J.; Cuadrado, F.; Escalona, M.J. A Systematic Review of Artificial Intelligence-Based Music Generation: Scope, Applications, and Future Trends. *Expert Systems with Applications* **2022**, *209*, 118190, doi:10.1016/j.eswa.2022.118190.
7. Herremans, D.; Chuan, C.-H.; Chew, E. A Functional Taxonomy of Music Generation Systems. *ACM Comput. Surv.* **2017**, *50*, 69:1-69:30, doi:10.1145/3108242.
8. Zhu, Y.; Baca, J.; Rekabdar, B.; Rawassizadeh, R. A Survey of AI Music Generation Tools and Models. **2023**, doi:10.48550/ARXIV.2308.12982.

9. Wen, Y.-W.; Ting, C.-K. Recent Advances of Computational Intelligence Techniques for Composing Music. *IEEE Trans. Emerg. Top. Comput. Intell.* **2023**, *7*, 578–597, doi:10.1109/TETCI.2022.3221126.
10. Xenakis, I. *Formalized Music: Thought and Mathematics in Composition*; Pendragon Press, 1992;
11. Schot, J.W.; Hiller, L.; Isaacson, L.M. Experimental Music Composition with an Electronic Computer. In Proceedings of the Mathematics of Computation; October 1962; Vol. 16, p. 507.
12. Fukayama, S.; Nakatsuma, K.; Sako, S.; Nishimoto, T.; Sagayama, S. Automatic Song Composition From The Lyrics Exploiting Prosody Of Japanese Language.; Zenodo, July 21 2010.
13. Scirea, M.; Barros, G.A.B.; Shaker, N.; Togelius, J. SMUG: Scientific Music Generator.; 2015.
14. Rütte, D. von; Biggio, L.; Kilcher, Y.; Hofmann, T. FIGARO: Controllable Music Generation Using Learned and Expert Features.; September 29 2022.
15. Lu, P.; Xu, X.; Kang, C.; Yu, B.; Xing, C.; Tan, X.; Bian, J. MuseCoco: Generating Symbolic Music from Text. **2023**, doi:10.48550/ARXIV.2306.00110.
16. Herremans, D.; Weisser, S.; Sörensen, K.; Conklin, D. Generating Structured Music for Bagana Using Quality Metrics Based on Markov Models. *Expert Syst. Appl.* **2015**, *42*, 7424–7435, doi:10.1016/j.eswa.2015.05.043.
17. Wu, J.; Hu, C.; Wang, Y.; Hu, X.; Zhu, J. A Hierarchical Recurrent Neural Network for Symbolic Melody Generation. *IEEE Transactions on Cybernetics* **2020**, *50*, 2749–2757, doi:10.1109/TCYB.2019.2953194.
18. Guo, Z.; Dimos, M.; Dorien, H. Hierarchical Recurrent Neural Networks for Conditional Melody Generation with Long-Term Structure 2021.
19. Choi, K.; Fazekas, G.; Sandler, M. Text-Based LSTM Networks for Automatic Music Composition. *arXiv preprint arXiv:1604.05358* **2016**.
20. Dong, H.-W.; Hsiao, W.-Y.; Yang, L.-C.; Yang, Y.-H. MuseGAN: Multi-Track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment. *Proceedings of the AAAI Conference on Artificial Intelligence* **2018**, *32*, doi:10.1609/aaai.v32i1.11312.
21. Huang, C.-Z.A.; Vaswani, A.; Uszkoreit, J.; Shazeer, N.M.; Simon, I.; Hawthorne, C.; Dai, A.M.; Hoffman, M.; Dinculescu, M.; Eck, D. Music Transformer: Generating Music with Long-Term Structure.; September 12 2018.
22. Borsos, Z.; Marinier, R.; Vincent, D.; Kharitonov, E.; Pietquin, O.; Sharifi, M.; Roblek, D.; Teboul, O.; Grangier, D.; Tagliasacchi, M.; et al. AudioLM: A Language Modeling Approach to Audio Generation 2023.
23. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-Resolution Image Synthesis with Latent Diffusion Models 2022.
24. Forsgren, S. Riffusion Stable Diffusion for Real-Time Music Generation. 2022.
25. OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; et al. GPT-4 Technical Report 2024.
26. Huang, Y.-S.; Yang, Y.-H. Pop Music Transformer: Beat-Based Modeling and Generation of Expressive Pop Piano Compositions. *Proceedings of the 28th ACM International Conference on Multimedia* **2020**, 1180–1188, doi:10.1145/3394171.3413671.
27. Monteith, K.; Martinez, T.; Ventura, D. Automatic Generation of Melodic Accompaniments for Lyrics.; 2012.
28. Ackerman, M.; Loker, D. Algorithmic Songwriting with ALYSIA. In Proceedings of the Computational Intelligence in Music, Sound, Art and Design; Correia, J., Ciesielski, V., Liapis, A., Eds.; Springer International Publishing: Cham, 2017; pp. 1–16.
29. Bao, H.; Huang, S.; Wei, F.; Cui, L.; Wu, Y.; Tan, C.; Piao, S.; Zhou, M. Neural Melody Composition from Lyrics. **2019**, *11838*, 499–511, doi:10.1007/978-3-030-32233-5_39.
30. Yu, Y.; Srivastava, A.; Canales, S. Conditional LSTM-GAN for Melody Generation from Lyrics. *ACM Trans. Multimedia Comput. Commun. Appl.* **2021**, *17*, 1–20, doi:10.1145/3424116.
31. Srivastava, A.; Duan, W.; Shah, R.R.; Wu, J.; Tang, S.; Li, W.; Yu, Y. Melody Generation from Lyrics Using Three Branch Conditional LSTM-GAN. In Proceedings of the MultiMedia Modeling; Þór Jónsson, B., Gurrin, C., Tran, M.-T., Dang-Nguyen, D.-T., Hu, A.M.-C., Huynh Thi Thanh, B., Huet, B., Eds.; Springer International Publishing: Cham, 2022; pp. 569–581.
32. Yu, Y.; Zhang, Z.; Duan, W.; Srivastava, A.; Shah, R.; Ren, Y. Conditional Hybrid GAN for Melody Generation from Lyrics. *Neural Comput & Applic* **2023**, *35*, 3191–3202, doi:10.1007/s00521-022-07863-5.

33. Zhang, Z.; Yu, Y.; Takasu, A. Controllable Lyrics-to-Melody Generation. *Neural Comput & Applic* **2023**, *35*, 19805–19819, doi:10.1007/s00521-023-08728-1.
34. Sheng, Z.; Song, K.; Tan, X.; Ren, Y.; Ye, W.; Zhang, S.; Qin, T. SongMASS: Automatic Song Writing with Pre-Training and Alignment Constraint. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence; May 18 2021; Vol. 35, pp. 13798–13805.
35. Ju, Z.; Lu, P.; Tan, X.; Wang, R.; Zhang, C.; Wu, S.; Zhang, K.; Li, X.-Y.; Qin, T.; Liu, T.-Y. TeleMelody: Lyric-to-Melody Generation with a Template-Based Two-Stage Method. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* **2022**, 5426–5437, doi:10.18653/v1/2022.emnlp-main.364.
36. Ding, S.; Liu, Z.; Dong, X.; Zhang, P.; Qian, R.; He, C.; Lin, D.; Wang, J. SongComposer: A Large Language Model for Lyric and Melody Composition in Song Generation. **2024**, doi:10.48550/ARXIV.2402.17645.
37. Davis, H.; Mohammad, S. Generating Music from Literature. In Proceedings of the Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLFL); Association for Computational Linguistics: Gothenburg, Sweden, 2014; pp. 1–10.
38. Rangarajan, R. Generating Music from Natural Language Text. *2015 Tenth International Conference on Digital Information Management (ICDIM)* **2015**, 85–88, doi:10.1109/ICDIM.2015.7381853.
39. Zhang, Y.; Wang, Z.; Wang, D.; Xia, G. BUTTER: A Representation Learning Framework for Bi-Directional Music-Sentence Retrieval and Generation. In Proceedings of the Proceedings of the 1st Workshop on NLP for Music and Audio (NLP4MusA); Oramas, S., Espinosa-Anke, L., Epure, E., Jones, R., Sordo, M., Quadrana, M., Watanabe, K., Eds.; Association for Computational Linguistics: Online, 2020; pp. 54–58.
40. Wu, S.; Sun, M. Exploring the Efficacy of Pre-Trained Checkpoints in Text-to-Music Generation Task 2023.
41. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. LLaMA: Open and Efficient Foundation Language Models 2023.
42. Yuan, R.; Lin, H.; Wang, Y.; Tian, Z.; Wu, S.; Shen, T.; Zhang, G.; Wu, Y.; Liu, C.; Zhou, Z.; et al. ChatMusician: Understanding and Generating Music Intrinsically with LLM. **2024**, doi:10.48550/ARXIV.2402.16153.
43. Liang, X.; Du, X.; Lin, J.; Zou, P.; Wan, Y.; Zhu, B. ByteComposer: A Human-like Melody Composition Method Based on Language Model Agent 2024.
44. Deng, Q.; Yang, Q.; Yuan, R.; Huang, Y.; Wang, Y.; Liu, X.; Tian, Z.; Pan, J.; Zhang, G.; Lin, H.; et al. ComposerX: Multi-Agent Symbolic Music Composition with LLMs 2024.
45. Liu, H.; Chen, Z.; Yuan, Y.; Mei, X.; Liu, X.; Mandic, D.; Wang, W.; Plumbley, M.D. AudioLDM: Text-to-Audio Generation with Latent Diffusion Models 2023.
46. Ghosal, D.; Majumder, N.; Mehrish, A.; Poria, S. Text-to-Audio Generation Using Instruction-Tuned LLM and Latent Diffusion Model 2023.
47. Majumder, N.; Hung, C.-Y.; Ghosal, D.; Hsu, W.-N.; Mihalcea, R.; Poria, S. Tango 2: Aligning Diffusion-Based Text-to-Audio Generations through Direct Preference Optimization. In Proceedings of the Proceedings of the 32nd ACM International Conference on Multimedia; Association for Computing Machinery: New York, NY, USA, October 28 2024; pp. 564–572.
48. Huang, Q.; Park, D.S.; Wang, T.; Denk, T.I.; Ly, A.; Chen, N.; Zhang, Z.; Zhang, Z.; Yu, J.; Frank, C.; et al. Noise2Music: Text-Conditioned Music Generation with Diffusion Models 2023.
49. Schneider, F.; Kamal, O.; Jin, Z.; Schölkopf, B. Moûsai: Text-to-Music Generation with Long-Context Latent Diffusion. **2023**, doi:10.48550/ARXIV.2301.11757.
50. Li, P.P.; Chen, B.; Yao, Y.; Wang, Y.; Wang, A.; Wang, A. JEN-1: Text-Guided Universal Music Generation with Omnidirectional Diffusion Models. *2024 IEEE Conference on Artificial Intelligence (CAI)* **2024**, 762–769, doi:10.1109/CAI59869.2024.00146.
51. Agostinelli, A.; Denk, T.I.; Borsos, Z.; Engel, J.; Verzetti, M.; Caillon, A.; Huang, Q.; Jansen, A.; Roberts, A.; Tagliasacchi, M.; et al. MusicLM: Generating Music From Text 2023.
52. Huang, Q.; Jansen, A.; Lee, J.; Ganti, R.; Li, J.Y.; Ellis, D.P.W. MuLan: A Joint Embedding of Music Audio and Natural Language 2022.
53. Lam, M.W.Y.; Tian, Q.; Li, T.; Yin, Z.; Feng, S.; Tu, M.; Ji, Y.; Xia, R.; Ma, M.; Song, X.; et al. Efficient Neural Music Generation. *Advances in Neural Information Processing Systems* **2023**, *36*, 17450–17463.

54. Chen, K.; Wu, Y.; Liu, H.; Nezhurina, M.; Berg-Kirkpatrick, T.; Dubnov, S. MusicLDM: Enhancing Novelty in Text-to-Music Generation Using Beat-Synchronous Mixup Strategies. In Proceedings of the ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); April 2024; pp. 1206–1210.
55. Copet, J.; Kreuk, F.; Gat, I.; Remez, T.; Kant, D.; Synnaeve, G.; Adi, Y.; Defossez, A. Simple and Controllable Music Generation. *Advances in Neural Information Processing Systems* **2023**, *36*, 47704–47720.
56. Macon, M.W.; Jensen-Link, L.; George, E.; Oliverio, J.C.; Clements, M. Concatenation-Based MIDI-to-Singing Voice Synthesis. *Journal of The Audio Engineering Society* **1997**.
57. Kenmochi, H.; Ohshita, H. VOCALOID-Commercial Singing Synthesizer Based on Sample Concatenation. In Proceedings of the Interspeech; 2007; Vol. 2007, pp. 4009–4010.
58. Saino, K.; Zen, H.; Nankaku, Y.; Lee, A.; Tokuda, K. An HMM-Based Singing Voice Synthesis System. In Proceedings of the Interspeech 2006; ISCA, September 17 2006; p. paper 2077-Thu1BuP.7-0.
59. Nishimura, M.; Hashimoto, K.; Oura, K.; Nankaku, Y.; Tokuda, K. Singing Voice Synthesis Based on Deep Neural Networks. In Proceedings of the Interspeech 2016; ISCA, September 8 2016; pp. 2478–2482.
60. Nakamura, K.; Hashimoto, K.; Oura, K.; Nankaku, Y.; Tokuda, K. Singing Voice Synthesis Based on Convolutional Neural Networks 2019.
61. Kim, J.; Choi, H.; Park, J.; Kim, S.; Kim, J.; Hahn, M. Korean Singing Voice Synthesis System Based on an LSTM Recurrent Neural Network. In Proceedings of the Proc. Interspeech; 2018; pp. 1551–1555.
62. Hono, Y.; Hashimoto, K.; Oura, K.; Nankaku, Y.; Tokuda, K. Singing Voice Synthesis Based on Generative Adversarial Networks. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* **2019**, 6955–6959, doi:10.1109/ICASSP.2019.8683154.
63. Lu, P.; Wu, J.; Luan, J.; Tan, X.; Zhou, L. XiaoiceSing: A High-Quality and Integrated Singing Voice Synthesis System 2020.
64. Ren, Y.; Ruan, Y.; Tan, X.; Qin, T.; Zhao, S.; Zhao, Z.; Liu, T.-Y. FastSpeech: Fast, Robust and Controllable Text to Speech 2019.
65. Morise, M.; Yokomori, F.; Ozawa, K. World: A Vocoder-Based High-Quality Speech Synthesis System for Real-Time Applications. *IEICE TRANSACTIONS on Information and Systems* **2016**, *99*, 1877–1884.
66. Blaauw, M.; Bonada, J. Sequence-to-Sequence Singing Synthesis Using the Feed-Forward Transformer. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* **2020**, 7229–7233, doi:10.1109/ICASSP40776.2020.9053944.
67. Zhuang, X.; Jiang, T.; Chou, S.-Y.; Wu, B.; Hu, P.; Lui, S. *Litesing: Towards Fast, Lightweight and Expressive Singing Voice Synthesis*; 2021; p. 7082;.
68. Lee, G.-H.; Kim, T.-W.; Bae, H.; Lee, M.-J.; Kim, Y.-I.; Cho, H.-Y. N-Singer: A Non-Autoregressive Korean Singing Voice Synthesis System for Pronunciation Enhancement 2022.
69. Chen, J.; Tan, X.; Luan, J.; Qin, T.; Liu, T.-Y. HiFiSinger: Towards High-Fidelity Neural Singing Voice Synthesis 2020.
70. Yamamoto, R.; Song, E.; Kim, J.-M. Parallel WaveGAN: A Fast Waveform Generation Model Based on Generative Adversarial Networks with Multi-Resolution Spectrogram 2020.
71. Wang, Y.; Skerry-Ryan, R.J.; Stanton, D.; Wu, Y.; Weiss, R.J.; Jaitly, N.; Yang, Z.; Xiao, Y.; Chen, Z.; Bengio, S.; et al. Tacotron: Towards End-to-End Speech Synthesis 2017.
72. Gu, Y.; Yin, X.; Rao, Y.; Wan, Y.; Tang, B.; Zhang, Y.; Chen, J.; Wang, Y.; Ma, Z. ByteSing: A Chinese Singing Voice Synthesis System Using Duration Allocated Encoder-Decoder Acoustic Models and WaveRNN Vocoders.; January 24 2021; pp. 1–5.
73. Liu, J.; Li, C.; Ren, Y.; Chen, F.; Zhao, Z. DiffSinger: Singing Voice Synthesis via Shallow Diffusion Mechanism 2022.
74. Hono, Y.; Hashimoto, K.; Oura, K.; Nankaku, Y.; Tokuda, K. Sinsy: A Deep Neural Network-Based Singing Voice Synthesis System. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **2021**, *PP*, 1–1, doi:10.1109/TASLP.2021.3104165.
75. Zhang, Y.; Cong, J.; Xue, H.; Xie, L.; Zhu, P.; Bi, M. VISinger: Variational Inference with Adversarial Learning for End-to-End Singing Voice Synthesis 2022.

76. Zhang, Y.; Xue, H.; Li, H.; Xie, L.; Guo, T.; Zhang, R.; Gong, C. VISinger 2: High-Fidelity End-to-End Singing Voice Synthesis Enhanced by Digital Signal Processing Synthesizer 2022.
77. Kim, J.; Kong, J.; Son, J. Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech. In Proceedings of the Proceedings of the 38th International Conference on Machine Learning; PMLR, July 1 2021; pp. 5530–5540.
78. Dhariwal, P.; Jun, H.; Payne, C.; Kim, J.W.; Radford, A.; Sutskever, I. Jukebox: A Generative Model for Music. *arXiv preprint arXiv:2005.00341* **2020**.
79. Hong, Z.; Huang, R.; Cheng, X.; Wang, Y.; Li, R.; You, F.; Zhao, Z.; Zhang, Z. Text-to-Song: Towards Controllable Music Generation Incorporating Vocals and Accompaniment 2024.
80. Bai, Y.; Chen, H.; Chen, J.; Chen, Z.; Deng, Y.; Dong, X.; Hantrakul, L.; Hao, W.; Huang, Q.; Huang, Z.; et al. Seed-Music: A Unified Framework for High Quality and Controlled Music Generation 2024.
81. Chen, G.; Liu, Y.; Zhong, S.; Zhang, X. Musicality-Novelty Generative Adversarial Nets for Algorithmic Composition. *Proceedings of the 26th ACM international conference on Multimedia* **2018**, 1607–1615, doi:10.1145/3240508.3240604.
82. Hakimi, S.H.; Bhonker, N.; El-Yaniv, R. BebopNet: Deep Neural Models for Personalized Jazz Improvisations. In Proceedings of the ISMIR; 2020; pp. 828–836.
83. Martinez, T.R.; Monteith, K.; Ventura, D.A. Automatic Generation of Music for Inducing Emotive Response. **2010**.
84. Hung, H.-T.; Ching, J.; Doh, S.; Kim, N.; Nam, J.; Yang, Y.-H. EMOPIA: A Multi-Modal Pop Piano Dataset For Emotion Recognition and Emotion-Based Music Generation 2021.
85. Zheng, K.; Meng, R.; Zheng, C.; Li, X.; Sang, J.; Cai, J.; Wang, J.; Wang, X. EmotionBox: A Music-Element-Driven Emotional Music Generation System Based on Music Psychology. *Front. Psychol.* **2022**, *13*, doi:10.3389/fpsyg.2022.841926.
86. Neves, P.; Fornari, J.; Florindo, J. Generating Music with Sentiment Using Transformer-GANs 2022.
87. Dash, A.; Agres, K.R. AI-Based Affective Music Generation Systems: A Review of Methods, and Challenges 2023.
88. Ji, S.; Yang, X. EmoMusicTV: Emotion-Conditioned Symbolic Music Generation With Hierarchical Transformer VAE. *IEEE Trans. Multimedia* **2024**, *26*, 1076–1088, doi:10.1109/TMM.2023.3276177.
89. Jiang, N.; Jin, S.; Duan, Z.; Zhang, C. RL-Duet: Online Music Accompaniment Generation Using Deep Reinforcement Learning. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence; April 3 2020; Vol. 34, pp. 710–718.
90. Louie, R.; Coenen, A.; Huang, C.Z.; Terry, M.; Cai, C.J. Novice-AI Music Co-Creation via AI-Steering Tools for Deep Generative Models. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* **2020**, 1–13, doi:10.1145/3313831.3376739.
91. Surbhi, A.; Roy, D. Tunes of Tomorrow: Copyright and AI-Generated Music in the Digital Age. *AIP Conference Proceedings* **2024**, *3220*, 050003, doi:10.1063/5.0234946.
92. Bulayenko, O.; Quintais, J.P.; Gervais, D.J.; Poort, J. AI Music Outputs: Challenges to the Copyright Legal Framework 2022.
93. Hsiao, W.-Y.; Liu, J.-Y.; Yeh, Y.-C.; Yang, Y.-H. Compound Word Transformer: Learning to Compose Full-Song Music over Dynamic Directed Hypergraphs. *Proceedings of the AAAI Conference on Artificial Intelligence* **2021**, *35*, 178–186, doi:10.1609/aaai.v35i1.16091.
94. Josan, H.H.S. AI and Deepfake Voice Cloning: Innovation, Copyright and Artists' Rights. *Artificial Intelligence* **2024**.
95. Zhang, Z.; Ning, H.; Shi, F.; Farha, F.; Xu, Y.; Xu, J.; Zhang, F.; Choo, K.-K.R. Artificial Intelligence in Cyber Security: Research Advances, Challenges, and Opportunities. *Artif Intell Rev* **2022**, *55*, 1029–1053, doi:10.1007/s10462-021-09976-0.
96. Fox, M.; Vaidyanathan, G.; Breese, J.L. The Impact of Artificial Intelligence on Musicians. *Issues in Information Systems* **2024**, *25*.