Artificial Intelligence Techniques for Pop Music Creation: a Real Music Production Perspective

Ning Zhang, Junchi Yan, Jean-Pierre Briot

Artificial Intelligence Techniques for Pop Music Creation: a Real Music Production Perspective

Ning Zhang[a], Junchi Yan[a], Jean-Pierre Briot[b]

[a]*Department of Computer Science and Engineering,Shanghai Jiao Tong University, Shang Hai, China*
[b]*LIP6, Sorbonne Université - CNRS, France*

Abstract

The recent surge of interest in computational music creation has been greatly influenced by the advent of large generative models such as ChatGPT and Stable Diffusion. These powerful generative AI models have demonstrated remarkable capabilities, especially in the domain of text and image generation. Spurred by these developments, the music industry has also begun to explore the deployment of large models for music creation, such as MusicLM and MusicGen. However, it's noteworthy that the performance and capabilities of these music-focused generative models have not yet reached the same level of sophistication as their counterparts in text and image generation. The generation of music presents unique challenges, such as capturing the intricate temporal structures, orchestrating an emotional progression, painting a sonic landscape, and managing the sophisticated interplay between various musical elements. The controllability and interactivity of the current AI-based music generation systems is unsatisfactory. In light of these considerations, a critical examination on the evolution of AI-based pop music creation techniques is both timely and essential, particularly from an industry perspective.

This paper, drawn from the authors' extensive experience as senior researchers in both industry and academia, provides a comprehensive overview of AI-based music creation techniques and their practical applications in real-world music production. It examines multiple aspects including lyrics generation, melody creation, lyrics-melody matching, arranging, and audio synthesis. The review offers an insight into the evolution and application of AI techniques in actual music production, critically evaluating their advantages, limitations. Furthermore, this paper identifies challenges and potential future directions for the field, with the aspiration of contributing to the de-

velopment of more intelligent and versatile AI tools that can serve the music industry more effectively.

Keywords: Algorithmic Composition, Pop Music Creation, Intelligent Creation

1. Introduction

Algorithmic composition has a history of over half a century, dating back to the early stochastic composition system 'Atree' [1]. Since its inception, relentless efforts have been devoted to developing systems capable of automated music creation. Traditional approaches [2, 3] relied on expert systems employing handcrafted grammars and rules to create music of different styles and structure. However, crafting these rules is a nontrivial task as music theory is inherently complex, with numerous rules that can vary depending on genres, composers' styles and music forms. For example, a system for generating four-part J.S. Bach chorale was developed using 350 handcrafted rules [2]. Some works [4] extended this by learning from the score corpus of a specific composer to creates custom grammars and rules.

In recent years, deep learning techniques have ventured into the music creation field. Different from the rule-based models, deep learning models can automatically learn the distributions of the training samples, thereby generating music samples that bear resemblance to the training set. Symbolic music can be represented mainly in two forms: piano roll and the event sequences. The piano roll represents the symbolic music as images of shape $P \times T \times I$, where P, T, I denote the number of pitches, time steps, and instruments, respectively [5]. On the other hand, event sequences represent symbolic music as note-based or frame-based sequences [6]. While a line of studies [5, 7, 8] aim to generate piano roll music, [9, 10, 11, 12] construct event sequence models like RNNs and Transformers to generate sequences music. *Flow Composer* [13] is an AI-assisted music composition tool, which has been iteratively improved over past years. This system has found extensive use among musicians. With the assistance of the *Flow Composer*, artists have even produced a music album: *Hello World* [14]. The prevailing GPT4 [15] can generate simple music scores in several formats. MusicLM [16] and MusicGen [17] are large models which are capable of generating music audio directly from given text. However, the fine-grained control over the music generation process is still unsatisfactory, leading to a gap between

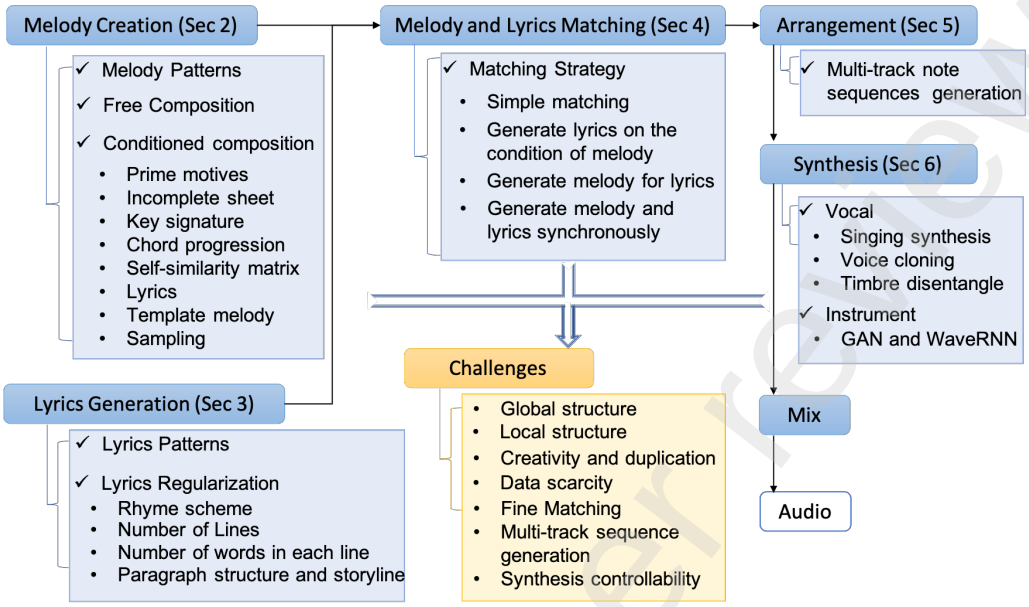

Figure 1: Flowchart of industrial music production, and the various techniques covered in this survey.

AI-generated compositions and human-crafted music.

A line of works [18, 19, 20] have provided thorough reviews on the deep learning based music generation techniques, mainly focusing on monophonic and polyphony symbolic music, accompaniment and counterpoint. They offer in-depth discussions on representations, architectures and challenges in deep learning based music generation techniques. The term 'music' in these reviews refers broadly, including various kinds of pieces like drum beats, melodies, violin quartets, piano concertos and so on. For foundational knowledge on deep learning based music generation, these surveys are highly recommended. However to the best of our knowledge, there is no review that focuses on creating complete pop music from scratch using artificial intelligence techniques, which is both challenging and practical. This process not only involves symbolic music generation, but also includes text generation, matching skills, vocal and instrument synthesis, mixing, and more.

Pop music is much different from classical or other kinds of music. It typically comprises a melody line, lyrics and arrangements, with the melody line often led by one or more vocals in recorded music. The basic form of a pop song is the verse-chorus structure, which may be repeated several times.

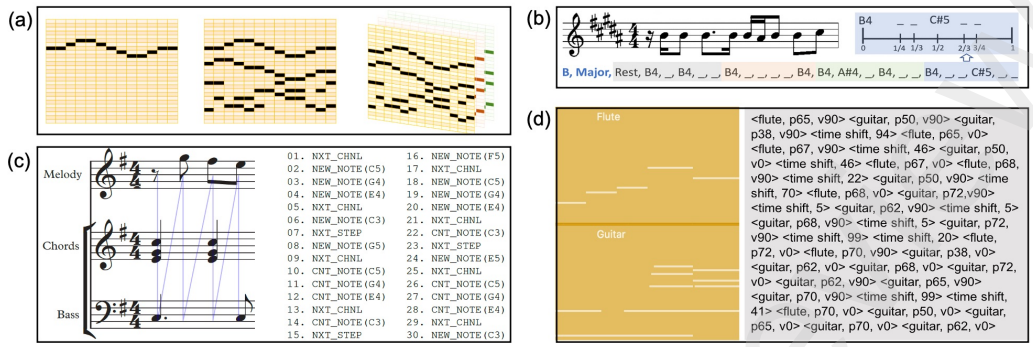

Figure 2: Representations for symbolic music. (a) From left to right: piano roll for single track monophonic music, single track polyphonic music, multi-track polyphonic music. (b) Uneven frame-based event sequence representation for melody. (c) Even frame-based event sequence representation for multi-track polyphonic music (quoted from [23]). (d) Note-based event sequence representation for multi-track polyphonic music.

Verse parts or chorus parts usually have similar lyrics and melodies, while chorus parts often feature different and 'bigger' lyrics, melodies, chord progressions, textures than verse parts. From a production view, several steps are needed to create complete pop music. First, composers create the lyrics and melodies; Next, the arranger adds textures to the melodies; Finally, the vocal and instrumental sounds are recorded or synthesized in a recording studio, and then the vocal and the instrumental sounds are mixed into audio. Various deep learning techniques have been studied to assist or mimic human work in all these steps. Increasingly more researchers have tried to apply deep learning models to monophonic melody composition, multi-track arrangement creation, lyrics conditioned melody generation, voice synthesis, instrumental synthesis, etc [21, 22].

In this review, we will review artificial intelligence techniques for each part of the pop music creation process, including melody creation, lyrics generation, melody and lyrics matching, arranging, voice and instrumental synthesis. Fig. 1 presents the flowchart of industrial music production and the corresponding sections in this paper. This review serves as a guideline for AI-based pop music creation and summarizes research progresses and challenges, which will be beneficial for both newcomers and experienced researchers in this field.

2. Melody Creation

2.1. Problem Statement

Symbolic music is commonly represented in tow formats: the piano roll and the event sequence, which are shown in Figure 2. The piano roll representation treats the symbolic music as a matrix, analogous to an image, where each element corresponds to a particular note at a specific time. Consequently, image generation techniques can be applied to the piano roll representation. Mathematically, this can be expressed as:

$$p(M) = \sum_{c} p(I_m|c)p(c) \tag{1}$$

where M represents the symbolic music, I_m denotes the piano roll. The variable c represents given conditions, which could include elements such as a melody line, beat structure, chord progression, style notation or other constraints. The event sequence representation, on the other hand, serializes symbolic music into event sequences. There are two primary approaches to achieve this serialization: the note-based sequences and the frame-based sequences. The note-based sequence representation captures the music score with events that are note-wise, including *note on*, *note off* and *time shift*. These events are then arranged chronologically, as illustrated in Fig. 2(b). The frame-based sequence representation quantizes time into equal or unequal intervals, using the smallest sub-division, such as a 16^{th} or 32^{th} note. For each sub-division or tick, the notes that start on that tick are represented by tokens corresponding to the note names. Fig. 2(c) and (d) show examples of frame-based sequence representations. For melody generation, we recommend using the uneven frame-based sequence representation [24], due to its simplicity, efficiency, and conciseness. This representation is amenable to various sequence models for generating event sequence-based music. The joint probability of an event sequence can be expressed as the production of a series of conditional probabilities:

$$p(M) = \prod_{i=1}^{n} p(m_i|m_{0,\ldots,i-1}, c) \tag{2}$$

where m_i represents an element of the sequence, and n is the length of the sequence.

Typical Pop Song Form: ABA'B'A''B'' OR ABA'B'CB''		
A	First verse	Solo singer with quiet instrumental backup
B	First chorus	Different melody, different chord progression, often a 'bigger', more complex texture than verse.
A'	Second verse	Different words, but the music is very similar to the first verse (usually with small differences)
B'	Second chorus	Same as the first chorus (no noticeable differences)
A''	Third verse	Same comments as second verse
C	Bridge	New melody with new chord progression
B''	Final chorus	May add more vocal or instrumental parts

Figure 3: Typical pop song forms and the characteristics of each part. (Figure is adapted from: https://courses.lumenlearning.com/musicappreciation_with_theory/chapter/binary-form/).

2.2. Techniques and Models

In practice, the melody typically comprises a motif or a fundamental idea that spans one or two bars in length. The motif is then elaborated into phrases, sections and eventually a complete melody piece through various composition techniques. As per the composition theory, there exist over ten kinds of techniques such as repetition, transposition, compression, expansion, and inversion for developing a motif into melody [1].

Melody creation techniques can be broadly categorized based on whether the creation is conditioned. In the unconditioned case, termed as free composition, the melody is generated from scratch. Various models, like the performance RNN [9], Music Transformers [10, 25, 11], and Music Diffusion Model [26], are capable of operating in the free composition mode. These models represent music as a sequence of note-based events, and employ sequence models to learn the joint distribution of training sequences.

In most instances, melodies are generated with specific conditions or priors, such as motifs, tonic and style. We enumerate common conditions and corresponding models below.

Prime Motifs. Many RNN or transformer based models [9, 10, 25] can be adept at working in continuation mode, where an initial motif is provided and the model autoregressively continues the generation process.

Incomplete Sheet. At times, only portions of the sheet music are available, and models are tasked with filling in the gaps. CocoNet [5] addresses this scenario as an image inpainting problem, generating piano rolls through an iterative erasure and inpainting process. Although designed for Bach-style chorales, CocoNet can be adapted for melody generation. InpaintNet [24] combines MeasureVAE with LatentRNN to consider past and future musical contexts, generating a sequence that connects them. By incorporating a diffusion model into MusicVAE, the model in [26] generates consecutive music phrases in parallel, interpolating in latent space to fill in vacancy measures.

Chord Progressions. The models in [11] and [10] can operate in a sequence-to-sequence mode, where the input to the encoder is a chord sequence and the decoder's target is the melody sequence. The work in [27] constructs a convolutional GAN (generative adversarial network) to generate melody piano rolls conditioned on chord progressions. Additionally, [28] employs a two-phase training process involving a transformer-based rhythm decoder and pitch decoder for chord conditioned melody generation. A Harmony-Aware Hierarchical Music Transformer (HAT) is introduced in [29], comprising transformers for texture, form and song individually. It tries to learn the mutual dependency between the textures and chord progressions.

Key Signatures, Instruments, Styles, etc. Key signatures, instruments, and musical styles, among other aspects, can be integrated into the model as conditioning tokens, which are placed ahead of the note sequences (as depicted in Fig. 2(c)). During the inference phase, these conditioning tokens are fed into the model, guiding it to generate the subsequent elements in accordance with desired attributes. For example, MuseNet[11] employs additional tokens to specify instruments and styles, thereby allowing users to control the feature of the generatured music. Another notable model, FIGARO [30], operates in a self-supervised description-to-sequence mode. Uniquely, FIGARO has the capacity to create music based on a descriptive conditioning sequence, encompassing an array of parameters such as time signature, note density, average pitch, velocity, duration, as well as the instruments and chords to be used throughout the composition. Futher, the work in [31] introduces a CMT (Controllable Music Transformer) that encodes both rhythm-related and note-related attributes. These encoded attributes are subsequently utilized as conditions during the inference phase, granting uses more control over generated music's characteristics. This control is especially valuable in

aligning the generated content with the artistic vision or specific requirements of a project.

Self-Similarity Matrix. In [32], the self-similarity matrix (SSM) is generated by a designed GAN and then melodies are generated on the condition of this SSM. [33] takes the SSM as a constraints in the model design. The SSM contains the music structure information, so the generated melody is imposed a given structure.

Lyrics. Some works focus on generating melodies that are conditioned on lyrics. For instance, the study by Bao et al. [34] introduces a model that predicts the pitch and duration for each word in the provided lyrics. Another interesting approach is proposed by Yu et al. [35], where they employ a conditional LSTM-GAN for generating melodic phrases based on lyrics sentences. TeleMelody [36] presents a two-stage lyric-to-melody generation system. This innovative method employs a template, which serves as an intermediary between lyrics and melodies. Consequently, two models can be trained: one to map lyrics to the template, and another to convert the template into a melody. These models are trained using self-supervision techniques, and interestingly, they do not explicitly rely on paired lyric-melody data, circumventing the issue of data scarcity.

Template Melody. Another avenue pursued by researchers is generating melodies based on a template melody. The objective here is to adapt or transfer characteristics of template melodies to craft new compositions. For instance, MusicVAE [37] has the ability to sample a 'middle point' between two given melody segments. effectively creating a new melody conditioned on two templates. The resulting melody can be seen as a synthesis of the two input melodies. Dai et al. [38] take a different approach by extracting musical frameworks from existing songs, which are then used as a foundations for generating new melodies. These music frameworks encode multi-level musical structures including sections, phrases, rhythm structures and melodic contours. By blending multi-level structures from different songs, their approach facilitates the creation of new music that inherits elements from the original templates.

Sampling Condition. The conditions discussed above are primarily applied during the inference stage. However, conditions can also be implemented during the sampling stage. For example, the samples at each time step could

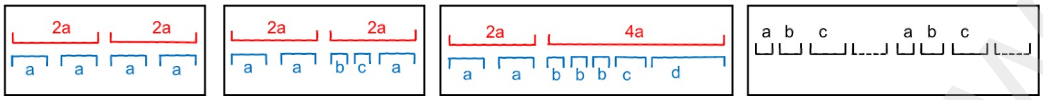

Figure 4: Typical phrase structures of melodies. 'a, b, c' represent different phrase lengths. The red lines denote phrase level, and the blue lines represent sub-phrase level. The black lines denote either the phrase level or the sub-phrase level.

be constrained to align with a specific tonic or metric. Additionally, the sampling probability of specific tokens can be manipulated based on certain conditions. [38] introduces a sampling method with dynamic time warpping control, which is capable of selecting melodies with high contour similarity to a reference melody.

Incorporating condition into generative models offers a level of user-control, enabling the user to represent music elements as conditions that guide the generation of melodies. Moreover, these conditional techniques can be combined to achieve more sophisticated results. For instance, one can employ both SSM and tonic conditions during the inference stage, and simultaneously utilize sampling conditions. [39] integrates several constraints with Markov chains, effectively addressing the constraint problem to produce music.

2.3. Challenges

Global Structure.. The fundamental structure of a pop song typically follows the AABA form, with various adaptions derived from this base structure. Fig. 3 illustrates two common pop song structures. The melody in each section possesses distinct characteristics and must be properly correlated with other sections. A comprehensive pop song comprises a lengthy note sequence, usually spanning at least 32 bars. Modeling such extensive sequences with a well-defined hierarchical structure presents significant challenges.

Some research attempts to leverage SSM to control the global structure of generated melodies [33]. However, obtaining an effective SSM becomes a challenge in itself. The approach taken in [32] employs adversarial learning to generate SSM, while in [33], SSM is derived from existing music.

The framework presented in [38] offers an alternative by harnessing the multi-level structures found in songs. Nevertheless, analyzing structures at the section and phrase levels introduces another challenge. The accuracy of structure extraction is critical, as any inadequacy directly impacts the

structure of the music generated. Furthermore, the similarities in structures and contours of generated melodies may point to a lack of creativity.

Local Structure.. Fig. 4 depicts common phrase structures in melodies. In pop music, phrase structure is highly regularized and concise, even a slight deviation in beat can disrupt the phrase structure. Historically, only a handful of studies have attempted to address the intricate phrase structure in generated melodies. Recently, developments focus on the nuanced phrase and rhythm structure. For instance, [40] annotates the phrase-level structures in the POP909 dataset [41]. Building on this, [29] extracts phrase patterns from chord progressions to guide melody generation. However, questions remain regarding controllability and the regularity of phrase structures. Currently human invention or rules-based systems are required to ensure that the phrase structure.

Creativity and Duplication.. Since generative models are trained using the existing data, there is a possibility that the generated samples may replicate or closely mimic the training data. [42] proposes a method for constraining the order of Markov chains-based generation in order. While this method may not be directly applicable to deep learning techniques, it is presented here for consideration.

3. Lyrics Generation

3.1. Problem Statement

Lyrics play a crucial roles in pop music creation as they convey messages and emotions. The task of lyrics generation falls under the domain of text generation, which has a wealth of literature. There are numerous text generation models based on self-attention architectures such as GPTs [15, 43]. Generally, these models generate text autoregressively, step by step, similar to the generation of melody sequences.

$$p(s) = \prod_{i}^{n} p(t_i | t_{0,...,i-1}, c) \tag{3}$$

where $s = (t_0, t_1, ..., t_n)$ is a sentence comprised of n tokens.

However, there are some differences between the general text generation and lyrics generation. Lyrics, akin to poetry, are intended for singing and exhibit distinct line and paragraph characters. Firstly, lyrics typically adhere

to a specific rhyme scheme, with each line often ending in a rhyming word. Below is an example of AABB rhyme scheme (*Couplet*) from the 'Cleanup Song':

Over here and over there	A
Up and down and everywhere	A
There's paper, brushes, paints and glue	B
And lots of pictures that we drew	B

Numerous rhyme schemes exist, such as ABABCDCD (*Alternate rhyme*), ABABBCBC (*Ballade*), AAABBB (*Triplet*), AABBA (*Lime rick*), among others. Secondly, lyrics have structured paragraph akin to the verse-bridge-chorus structure in melodies, as seen in Fig. 3. Each paragraph has unique features and together they form a cohesive storyline. Typically, the verse sets the stage, the bridge recalls past events, and the chorus expresses emotions [44]. These paragraphs often transition between themes [45].

Lyrics data is relative abundant, and large volumes of lyrics text can be easily acquired from the internet. The genre of generated lyrics can be steered by preparing training sets from different genres. Next we will discuss how to incorporate the two characteristics of lyrics.

3.2. The Speciality of Lyrics Generation

Rhyme Scheme, Number of Lines/Words for Each Line.. When using transformer-based language models, the generated lyrics may naturally contain rhymes, albeit unintentional. There is a straightforward way to control the rhyme of each line in the lyrics: firstly, reverse the words in each line of the data; secondly, train a language model using the reversed corpus; and thirdly, during the generation phase, mask out words that do not conform to the rhyme scheme. The number of lines can be controlled by halting the language model at the desired position. Controlling the number of words in each line can be achieved by inserting the word number control token between lines, plus a post-processing procedure.

The Paragraph Structure and Storyline.. Various models have been developed to generate lyrics according to given themes or keywords. Tra-la-Lyrics 2.0 [46] combines the original rhythm lyrics model Tra-la-Lyrics with the poetry generation platform PoeTryMe [47], enabling the creation of lyrics within a semantic domain. Rapformer [48] is a transformer-based denoising autoencoder capable of synthesizing a rap verse from the content of any text. [49]

introduces a hierarchical attention based Seq2Seq model for context-aware generation of Chinese song lyrics. The *Youling* [50] is an AI-assist lyrics creation system designed for collaboration with human lyricists. However, most existing works generate only a single paragraph of lyrics and do not address the verse-bridge-chorus structure found in actual pop songs. Capturing long-term consistency and topic transitions across paragraphs is challenging [51]. Efforts have been made to model the topic transition structures from lyrics data without supervision [44, 45], but these do not demonstrate performance on lyrics generation.

4. Melody and Lyrics Matching

4.1. Matching Strategies

As highlighted in the previous section, lyrics are created for songs and are therefore inherently tied to rhythms or melodies. It is vital for the melody and lyrics to be coherent in terms of rhythm, phrase structure, paragraph structure, and even semantic emotions. For instance, the boundaries of words in lyrics should align with the rests in a melody [52], and stressed syllables in the lyrics should coincides with the strong beats in the melody [46]. In this section, we discuss several strategies for matching melody and lyrics.

Simple Matching.. A straightforward approach is to pair a melody with lyrics by ensuring that the number of notes in melody is equal to the number of syllables in the lyrics, and then match each note with a syllable. This method aligns the melody and lyrics in terms of length but does not take into account other aspects such as rhythm and stress, structures.

Generate Lyrics On the Condition of Melody.. Tra-la-lyrics [53] is an approach that generates lyrics based on the rhythm of melodies, although this method relies on hand-crafted strategies rather than deep learning techniques. MC-SeqGAN [54] offers an end-to-end system for generating lyrics conditioned on a melody, but it only generates a line of lyrics given the corresponding melody as the input, and does not consider syllable matching. [52] considers the syllable structure and employs a two-channel Seq2Seq (sequence-to-sequence) model for Chinese lyrics generation. This model utilizes two different Bi-LSTMs to process the structural token sequence and preceding sentences, while text generation is handled by a forward LSTM. The goal is to generate lyrics that harmonize with the original melody by

effectively learning the syllable structure. Additionally, [55] introduces a dataset of lyrics-melody pairs and suggests a melody conditional RNNLM (RNN Language Model). This model uses the combination of a context melody vector and word embedding as input, simultaneously predicting the words and the syllable counts of those words. Experimental results indicate that this approach is capable of maintaining compatibility between the boundaries of generated lyrics and melody structures.

Generate Melody for Lyrics.. In Section 2.2, we have discussed various works that focus on generating melodies based on lyrics. Moreover, iComposer [56] employs three Seq2Seq models to generate pitches for lyrics, duration for lyrics, and lyrics for melodies, respectively. It features a user interface and has made the source code available.

Synchronous Generation of Melody and Lyrics.. AutoNLMC [57] is engineered to simultaneously generate lyrics and the corresponding melodies. It includes a lyrics prediction module, an encoder and a melody decoder. The melody decoder is further composed of a duration decoder, pitch decoder, and a dedicated rest decoder. SongMASS [21] incorporates a cross melody MASS (masked seq2seq transformer) and a lyrics MASS. This model supports supervised learning for both melody-to-lyrics and lyrics-to-melody mappings, and also facilitates unsupervised learning of melodies and lyrics separately.

4.2. Challenges

Data Efficiency.. While [55] has managed to collect a dataset of 1,000 Japanese lyrics-melody pairs, the volume of data remains insufficient for a comprehensive study of lyrics-melody matching. [35] constructs a subset containing 7,998 aligned melodies and English lyrics from the LAKH dataset [58]. Nonetheless, the alignment of melodies and lyrics in this dataset is relatively coarse, and the syllables in midi files exhibit irregularities. This dataset requires additional annotations to be effectively utilized for supervised learning. Consequently, there is a need to explore self-supervised or unsupervised methods in this domain to enhance data efficiency. The work of [36] represents a promising direction in reducing dependence on paired data.

Fine-grained Matching between Melody and Lyrics.. The field of melody and lyrics matching is still in its nascent stage. Much of the existing research focuses on preliminary alignment, such as ensuring that lyric sentences and melody phrases are matching in length. However, achieving fine-grained

matching, which includes alignment in terms of rhythm and paragraph structure, is significantly more challenging and requires additional effort. Currently, employing hand-crafted strategies as demonstrated by [46], could facilitate finer matching between melody and lyrics.

Coherence. Achieving coherence is a vital aspect of matching melodies and lyrics. This entails ensuring that the melodies and lyrics are harmonious and create a unified musical experience. While technical alignment such as rhythm and structure is important, it is also imperative to consider the emotional and semantic dimensions of the song. This is because the emotional resonance and meaningfulness of the lyrics can significantly impact the overall musical experience. For example, a somber melody may not be well-suited to upbeat or cheerful lyrics. Developing models that are sensitive to the emotional content of both the lyrics and the melody, and can generate or match content that is thematically and emotionally coherent, is an area that requires further exploration

Evaluation Metrics.. Evaluating the quality of matched melodies and lyrics is also challenging. While it is possible to use objective metrics such as rhythm and structure alignment, evaluating the artistic and emotional coherence is more subjective and may require human assessment. Developing evaluation metrics that can effectively measure both the technical alignment and the artistic quality of matched melodies and lyrics is a critical area for future research.

5. Arrangement

In practice, a piece of pop music typically comprises multiple tracks in its arrangement, including elements such as chords, beats, rhythm patterns, etc. These elements are performed with a variety of instruments, such as piano, guitar, bass, drums, strings, winds, and others. Arrangement plays a pivotal role in defining the music's style and emotion. By utilizing different accompaniments, a melody can be adapted into various styles and convey a wide range of emotions. Arrangement involves harnessing the characteristics of different instruments, allocating different instruments to separate tracks, and orchestrating their roles to create a harmonious piece of music.

5.1. Arranging Models

Currently, deep learning-based music arrangement techniques have limitations in handling a large number of tracks. Most existing works typically deal with up to 6 tracks. In [59], a hierarchical RNN is constructed for multi-track pop music generation, where the network is structured to mirror the real-world pop music creation process. The lower layers generate the melody, while the upper levels produce accompaniments such as drums and chords. MuseGAN [7] incorporates a set of GANs for multi-track music generation. In this work, symbolic music is represented as piano-rolls of 5 tracks: bass, drums, guitar, piano and strings. The model accounts for intra-track structure, inter-track dependencies and temporal fluidity. Their training dataset LPD is derived from the LAKH dataset via selecting, merging, and clipping. The performances on the generation of 4 bars music of 5 tracks are presented.

The Microsoft XiaoIce Band [60] generates melody and arrangement within a unified framework. Initially, a melody is produced based on a given chord progression through a Chord-based Rhythm and Melody Cross-Generation Model (CRMCG). Subsequently, multi-track arrangements are generated by the Multi-Instrument Co-Arrangement Model (MICA), with an information-sharing strategy employed to enhance inter-track harmony. The resultant compositions consist of a melody played on the piano, accompanied by drum, bass, and violin. Further advancement in XiaoIce Band [61] allow for multi-style, multi-track arrangement, employing MICA as a generator with two discriminators for multi-style and harmony discrimination. This facilitates style control and improves harmony in generated music. Additionally, a guitar track has been introduced.

[62] introduces the POP909 dataset, encompassing multiple versions of piano arrangements for 909 popular songs curated by professional musicians, with annotations for tempo, beat, key, and chords. This work also proposes a baseline Transformer model for polyphonic music generation and piano arrangement generation. Models in [11, 63, 64] utilize Transformer-based architectures with note-based event sequence representation for multi-track music generation.

5.2. Challenges

Despite these advancements, learning-based arrangement generation remains in the early stages of development.

Complexity of Arrangement. There is a pressing need for further research in handling a more extensive array of tracks and increasing the complexity of the arrangement.

Style and Emotion Control.. Additionally, making the generation process more controllable is an important research direction. A vital aspect of arrangement is the ability to adapt music in distinct styles and emotional expression. Techniques such as style transfer, where the stylistic elements of one piece are applied to another, and conditioning arranging models on style or emotional labels, could be used to enable control over the style and emotional expression of the arrangement.

6. Audio Synthesis

After obtaining the music score, it can be converted into audio through synthesis and mixing. The synthesis process from score to audio is also known as sound rendering. In real practice, a melody is typically sung by a professional singer, and is then mixed with instrumental sound. The sound rendering and mixing are performed in a recording studio with a DAW (Digital Audio Workstation). Reaper and Logic Pro are examples of DAW software. These can be connected with sound libraries, such as KONTAKT, to render instrument scores into audios. FluidSynth coupled Soundfont2 is also a popular scheme for instrument rendering. Nonetheless, traditional rendering schemes tend to be time consuming and lack diversity. For a given score, different human performers imbue their own expressiveness. Conversely, the aforementioned rendering scheme consistently produce the same audio if using the same sound source. Consequently, some researchers have endeavoured to construct AI performer capable of generating music audio from scores.

6.1. Singing Synthesis

There are numerous public available vocal synthesizers, such as VOCALOID, Synthesizer V, which operate by splicing phonemes, rather than utilizing deep learning. Recently, however, there has been an influx of work centered around deep learning-based vocal synthesis. Most of these works adapt models from the TTS (text-to-speech) technology. Typically, a front-end model ingests text or musical scores as input and generates word or note embedding. These embeddings are then passed to a Seq2Seq module, and the decoder within this module generates acoustic features. Finally, a vocoder synthesizes the

sound audio. For example, the method in [65] constructs a singing synthesis model using the above process. Furthermore, [66] extracts the F0 feature and builds a duration model to facilitate the decoder's learning of acoustic features. [67] employs a denoising diffusion model to augment the acoustic model's prediction accuracy.

Training TTS models requires large amount of high quality data, and the labelling of TTS corpus is intricate and costly. The same challenges apply to singing synthesis, as singing corpus are rare. To alleviate dependence on singing corpora, some researchers have investigated voice conversion techniques. For instance, [68] adapts the voice cloning technique from the TTS to the singing synthesis, enabling the creation of a multi-speaker model using data from various speakers. This model can be efficiently adapted to new voices using a small amount of target data. [69] introduces a singing-from-speech model capable of synthesizing a target speaker's singing voice using only their speech samples. [70] expands a single-singer model to support multiple singers by decoupling the timbre embedding from the system. Sinsy [71] and LiteSing [72] are examples of data-efficient singing synthesis models that operate with around 1 hour data.

6.2. Instrument Synthesis

Creating expressive instrument synthesis is challenging due to the consideration variation in timbre among different instruments. Additionally, instrument synthesis data is scarce. GANSynth [73] is a neural instrument synthesis model based on GANs. The GAN is used to model log-magnitudes and instantaneous frequencies, which are subsequently converted to audio in time domain. A more streamlined WaveRNN model is presented in [74], which explores a variety of domain-specific conditioning features and architectures. This model is time-efficient and capable of fine-grained control over different dimensions. MIDI-DDSP (Differentiable Digital Signal Processing) [75] is a hierarchical music audio generative model, which not only provides realistic neural audio synthesis but also allows for detailed user control for 13 different kinds of instruments.

6.3. Challenges

Expressiveness and Variability. Adding expressiveness and variability to synthesized audio is an important challenge. Current sound rendering schemes often result in static and monotonous outputs. AI models can be trained to mimic the nuances and expressiveness of human performance. Techniques

such as human performance modeling, where AI algorithms are trained on performances by human musicians, can be employed. These models can learn to add natural variations, such as subtle timing deviations, dynamics changes, and articulations, to make the synthesized audio more expressive and human-like. Additionally, enabling users to control the level of expressiveness through user-defined parameters or real-time performance data (e.g., MIDI controllers) could further enhance the applicability and creativity of audio synthesis systems.

7. Conclusion and Outlook

This paper provides an overview of artificial intelligence techniques employed in various stages of real-world industrial pop music production. We have discussed the characteristics of these techniques in detail and she light on the challenges that researchers and practitioners face. It's important to note that we did not include the audio mixing and remixing related techniques in this survey, primarily because deep learning applications in this sub-field are still relatively scarce.

For future works, establishing robust evaluation methodologies for music generation techniques is critical. Current approaches often rely on subjective human evaluation, which is not scalable and can be biased. Objective metrics that can effectively measure the quality, creativity, and diversity of generated music would be invaluable in comparing and advancing different techniques. Furthermore, the availability of more public, standardized music datasets is essential for the progress of the field. Such datasets can facilitate better benchmarking and promote reproducibility, which in turn can accelerate the development of new models and algorithms. Finally, incorporating the ability to have more control and expressiveness in the music generation process is an important aspect. This involves developing models that allow for user-defined parameters such as style, emotion, progression, and more, enabling creators to have more agency over the music they generate.

In conclusion, artificial intelligent technology holds immense potential for revolutionizing the way pop music is produced. By continuing to innovate and address the challenges head-on, the convergence of technology and creativity can herald a new era for music production.

Acknowledgement

This research was supported by Science and Technology Commission of Shanghai Municipality (STCSM) No.22ZR1434900, for which the authors are profoundly grateful.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used GPT4 in order to improve language and readability, with caution. After using this tool/service, the authors reviewed and edited the content as needed and takes full responsibility for the content of the publication.

References

[1] I. Xenakis, Formalized music: Thought and mathematics in composition (sharon kanach, compilation and edition) (1963).

[2] K. Ebcioğlu, An expert system for harmonizing four-part chorales, Computer Music Journal 12 (3) (1988) 43–51.

[3] G. Nierhaus, Algorithmic composition: paradigms of automated music generation, Springer Science & Business Media, 2009.

[4] D. Cope, The algorithmic composer, Vol. 16, AR Editions, Inc., 2000.

[5] C. Huang, T. Cooijmans, A. Roberts, A. Courville, D. Eck, Counterpoint by convolution, in: Proc. of 18th ISMIR, 2017.

[6] G. Hadjeres, F. Pachet, F. Nielsen, Deepbach: A steerable model for bach chorales generation, in: Proc. of 34th ICML, 2017.

[7] H. Dong, W. Hsiao, L. Yang, Y. Yang, Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment, in: Proc. of 32nd AAAI, 2018.

[8] H. Dong, Y. Yang, Convolutional generative adversarial networks with binary neurons for polyphonic music generation, arXiv preprint arXiv:1804.09399 (2018).

[9] I. Simon, S. Oore, Performance rnn: Generating music with expressive timing and dynamics, in: Proc. of JMLR, Vol. 80, 2017, p. 116.

[10] C. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. Dai, M. Hoffman, M. Dinculescu, D. Eck, Music transformer: Generating music with long-term structure, in: ICLR, 2018.

[11] C. Payne, "musenet.", openai.com/blog/musenet (2019).

[12] Y.-S. Huang, Y.-H. Yang, Pop music transformer: Generating music with rhythm and harmony, arXiv:2002.00212 (2020).

[13] F. Pachet, A. Papadopoulos, P. Roy, Comments on "assisted lead sheet composition using flowcomposer", 25th anniversary of the CP Conference (2019).

[14] Skygge, et al., The site of the hello world album, https://www.helloworldalbum.net/ (2020).

[15] OpenAI, Gpt-4 technical report (2023). arXiv:2303.08774.

[16] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, C. Frank, Musiclm: Generating music from text (2023). arXiv:2301.11325.

[17] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, A. Défossez, Simple and controllable music generation (2023). arXiv:2306.05284.

[18] J. Briot, G. Hadjeres, F. Pachet, Deep learning techniques for music generation–a survey, arXiv preprint arXiv:1709.01620 (2017).

[19] J. Briot, F. Pachet, Deep learning for music generation: challenges and directions, Neural Computing and Applications (2018) 1–13.

[20] E. R. Miranda, Handbook of Artificial Intelligence for Music: Foundations, Advanced Approaches, and Developments for Creativity, Springer Nature, 2021.

[21] Z. Sheng, K. Song, X. Tan, Y. Ren, W. Ye, S. Zhang, T. Qin, Songmass: Automatic song writing with pre-training and alignment constraint, in: AAAI, 2021.

[22] K. Watcharasupat, Controllable music: supervised learning of disentangled representations for music generation, Nanyang Technological University (2021).

[23] Y. Zhou, W. Chu, S. Young, X. Chen, Bandnet: A neural network-based, multi-instrument beatles-style midi music composition machine, arXiv preprint arXiv:1812.07126 (2018).

[24] A. Pati, A. Lerch, G. Hadjeres, Learning to traverse latent spaces for musical score inpainting, in: Proc. of 20th ISMIR, 2019.

[25] W.-Y. Hsiao, J.-Y. Liu, Y.-C. Yeh, Y.-H. Yang, Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs, in: Proc. of AAAI, Vol. 35, 2021, pp. 178–186.

[26] G. Mittal, J. Engel, C. Hawthorne, I. Simon, Symbolic music generation with diffusion models, in: Proc. of the 22nd ISMIR, 2021.
URL https://archives.ismir.net/ismir2021/paper/000058.pdf

[27] L. Yang, S. Chou, Y. Yang, Midinet: A convolutional generative adversarial network for symbolic-domain music generation, arXiv preprint arXiv:1703.10847 (2017).

[28] K. Choi, J. Park, W. Heo, S. Jeon, J. Park, Chord conditioned melody generation with transformer based decoders, IEEE Access 9 (2021) 42071–42080. doi:10.1109/ACCESS.2021.3065831.

[29] X. Zhang, J. Zhang, Y. Qiu, L. Wang, J. Zhou, Structure-enhanced pop music generation via harmony-aware learning, arXiv preprint arXiv:2109.06441 (2021).

[30] D. von Rütte, L. Biggio, Y. Kilcher, T. Hoffman, Figaro: Generating symbolic music with fine-grained artistic control, arXiv preprint arXiv:2201.10936 (2022).

[31] S. Di, Z. Jiang, S. Liu, Z. Wang, L. Zhu, Z. He, H. Liu, S. Yan, Video background music generation with controllable music transformer, in: 29th ACM Multimedia, 2021.

[32] H. Jhamtani, T. Berg-Kirkpatrick, Modeling self-repetition in music generation using generative adversarial networks, in: Machine Learning for Music Discovery Workshop, ICML, 2019.

[33] S. Lattner, M. Grachten, G. Widmer, Imposing higher-level structure in polyphonic music generation using convolutional restricted boltzmann machines and constraints, Journal of Creative Music Systems 2 (2018) 1–31.

[34] H. Bao, S. Huang, F. Wei, L. Cui, Y. Wu, C. Tan, S. Piao, M. Zhou, Neural melody composition from lyrics, in: CCF International Conference on Natural Language Processing and Chinese Computing, Springer, 2019, pp. 499–511.

[35] Y. Yu, A. Srivastava, S. Canales, Conditional lstm-gan for melody generation from lyrics, ACM TOMM 17 (1) (2021) 1–20.

[36] Z. Ju, P. Lu, X. Tan, R. Wang, C. Zhang, S. Wu, K. Zhang, X.-Y. Li, T. Qin, T.-Y. Liu, Telemelody: Lyric-to-melody generation with a template-based two-stage method, ArXiv abs/2109.09617 (2021).

[37] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, D. Eck, A hierarchical latent vector model for learning long-term structure in music, in: ICML, 2018.

[38] S. Dai, Z. Jin, C. Gomes, R. B. Dannenberg, Controllable deep melody generation via hierarchical music structure representation, in: ISMIR, 2021.

[39] F. Pachet, P. Roy, B. Carr'e, Assisted music creation with flow machines: towards new categories of new, ArXiv abs/2006.09232 (2020).

[40] S. Dai, H. Zhang, R. B. Dannenberg, Automatic analysis and influence of hierarchical structure on melody, rhythm and harmony in popular music, in: Proc. of the 2020 CSMC+MUME, 2020.

[41] Z. Wang, K. Chen, J. Jiang, Y. Zhang, M. Xu, S. Dai, X. Gu, G. Xia, Pop909: A pop-song dataset for music arrangement generation, in: Proc. of 21st ISMIR, 2020.

[42] A. Papadopoulos, P. Roy, F. Pachet, Avoiding plagiarism in markov sequence generation, in: Proc. of the AAAI, Vol. 28, 2014.

[43] B. Wang, A. Komatsuzaki, GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model, https://github.com/kingoflolz/mesh-transformer-jax (May 2021).

[44] K. Watanabe, Y. Matsubayashi, K. Inui, S. Fukayama, T. Nakano, M. Goto, Modeling storylines in lyrics, IEICE Transactions on Information and Systems 101 (4) (2018) 1167–1179.

[45] K. Watanabe, Modeling discourse structure of lyrics, Ph.D. thesis, Tohoku University (2018).

[46] H. Oliveira, Tra-la-lyrics 2.0: Automatic generation of song lyrics on a semantic domain, Journal of Artificial General Intelligence 6 (1) (2015) 87–110.

[47] H. Oliveira, T. Mendes, A. Boavida, A. Nakamura, M. Ackerman, Co-poetryme: interactive poetry generation, Cognitive Systems Research 54 (2019) 199–216.

[48] N. Nikolov, E. Malmi, C. Northcutt, L. Parisi, Rapformer: Conditional rap lyrics generation with denoising autoencoders, in: Proc. of the 13th INLG, 2020, pp. 360–373.

[49] H. Fan, J. Wang, B. Zhuang, S. Wang, J. Xiao, A hierarchical attention based seq2seq model for chinese lyrics generation, in: PRICAI, Springer, 2019, pp. 279–288.

[50] R. Zhang, X. Mao, L. Li, L. Jiang, L. Chen, Z. Hu, Y. Xi, C. Fan, M. Huang, Youling: an ai-assisted lyrics creation system, in: Proc. of EMNLP, 2020.

[51] K. Watanabe, M. Goto, Lyrics information processing: Analysis, generation, and applications, in: Proc. of the 1st NLP4MusA, 2020, pp. 6–12.

[52] X. Lu, J. Wang, B. Zhuang, S. Wang, J. Xiao, A syllable-structured, contextually-based conditionally generation of chinese lyrics, in: PRICAI, Springer, 2019, pp. 257–265.

[53] H. Oliveira, F. Cardoso, F. Pereira, Tra-la-lyrics: An approach to generate text based on rhythm, in: Proc. of the 4th. IJWCC, 2007.

[54] Y. Chen, A. Lerch, Melody-conditioned lyrics generation with seqgans, in: 2020 IEEE International Symposium on Multimedia (ISM), 2020, pp. 189–196. doi:10.1109/ISM.2020.00040.

[55] K. Watanabe, Y. Matsubayashi, S. Fukayama, M. Goto, K. Inui, T. Nakano, A melody-conditioned lyrics language model, in: NAACL-HLT, 2018.

[56] H. Lee, J. Fang, W. Ma, icomposer: An automatic songwriting system for chinese popular music, in: Proc. of the NAACL (Demonstrations), 2019, pp. 84–88.

[57] G. Madhumani, Y. Yu, F. Harscoët, S. Canales, S. Tang, Automatic neural lyrics and melody composition, arXiv preprint arXiv:2011.06380 (2020).

[58] C. Raffel, Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching, Ph.D. thesis, Columbia University (2016).

[59] H. Chu, R. Urtasun, S. Fidler, Song from pi: A musically plausible network for pop music generation, arXiv:1611.03477 (2016).

[60] H. Zhu, Q. Liu, N. Yuan, C. Qin, J. Li, K. Zhang, G. Zhou, F. Wei, Y. Xu, E. Chen, Xiaoice band: A melody and arrangement generation framework for pop music, in: Proc. of the 24th ACM SIGKDD, 2018, pp. 2837–2846.

[61] H. Zhu, Q. Liu, N. Yuan, K. Zhang, G. Zhou, E. Chen, Pop music generation: From melody to multi-style arrangement, ACM TKDD 14 (5) (2020) 1–31.

[62] Z. Wang, K. Chen, J. Jiang, Y. Zhang, M. Xu, S. Dai, X. Gu, G. Xia, Pop909: A pop-song dataset for music arrangement generation, in: Proc. of 21st ISMIR, 2020.

[63] Y. Ren, J. He, X. Tan, T. Qin, Z. Zhao, T. Liu, Popmag: Pop music accompaniment generation, in: Proc. of the 28th ACM Multimedia, 2020, pp. 1198–1206.

[64] J. Ens, P. Pasquier, Mmm: Exploring conditional multi-track music generation with the transformer, arXiv preprint arXiv:2008.06048 (2020).

[65] O. Angelini, A. Moinet, K. Yanagisawa, T. Drugman, Singing Synthesis: With a Little Help from my Attention, in: Proc. Interspeech 2020, 2020, pp. 1221–1225. doi:10.21437/Interspeech.2020-1399.

[66] M. Blaauw, J. Bonada, Sequence-to-sequence singing synthesis using the feed-forward transformer, in: ICASSP, 2020.

[67] J. Liu, C. Li, Y. Ren, F. Chen, P. Liu, Z. Zhao, Diffsinger: Diffusion acoustic model for singing voice synthesis, arXiv preprint arXiv:2105.02446 (2021).

[68] M. Blaauw, J. Bonada, R. Daido, Data efficient voice cloning for neural singing synthesis, in: ICASSP, 2019, pp. 6840–6844.

[69] L. Zhang, C. Yu, H. Lu, C. Weng, Y. Wu, X. Xie, Z. Li, D. Yu, Learning singing from speech, arXiv preprint arXiv:1912.10128 (2019).

[70] J. Lee, H. Choi, J. Koo, K. Lee, Disentangling timbre and singing style with multi-singer singing synthesis system, in: ICASSP, 2020, pp. 7224–7228.

[71] Y. Hono, K. Hashimoto, K. Oura, Y. Nankaku, K. Tokuda, Sinsy: A deep neural network-based singing voice synthesis system, IEEE/ACM TASLP 29 (2021) 2803–2815.

[72] X. Zhuang, T. Jiang, S.-Y. Chou, B. Wu, P. Hu, S. Lui, Litesing: Towards fast, lightweight and expressive singing voice synthesis, in: ICASSP, IEEE, 2021, pp. 7078–7082.

[73] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, A. Roberts, GANSynth: Adversarial neural audio synthesis, in: ICLR, 2019.
URL https://openreview.net/forum?id=H1xQVn09FX

[74] L. Hantrakul, J. Engel, A. Roberts, C. Gu, Fast and flexible neural audio synthesis., in: Proc. of 20th ISMIR, 2019.

[75] Y. Wu, E. Manilow, Y. Deng, R. Swavely, K. Kastner, T. Cooijmans, A. Courville, C.-Z. A. Huang, J. Engel, Midi-ddsp: Detailed control of musical performance via hierarchical modeling, in: ICLR, 2022.