IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 22 October 2022, accepted 25 November 2022, date of publication 30 November 2022,
date of current version 6 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3225689

SURVEY

Toward Interactive Music Generation: A Position Paper

SHAYAN DADMAN, BERNT ARILD BREMDAL, BØRRE BANG, AND RUNE DALMO
Department of Computer Science, UiT The Arctic University of Norway, 8514 Narvik, Norway

Corresponding author: Shayan Dadman (shayan.dadman@uit.no)

This work was supported in part by the UiT The Arctic University of Norway Ph.D. Scholarship Program funded by the Norwegian Government, and in part by a grant from the publication fund of the UiT The Arctic University of Norway.

ABSTRACT Music generation using deep learning has received considerable attention in recent years. Researchers have developed various generative models capable of imitating musical conventions, comprehending the musical corpora, and generating new samples based on the learning outcome. Although the samples generated by these models are persuasive, they often lack musical structure and creativity. For instance, a vanilla end-to-end approach, which deals with all levels of music representation at once, does not offer human-level control and interaction during the learning process, leading to constrained results. Indeed, music creation is a recurrent process that follows some principles by a musician, where various musical features are reused or adapted. On the other hand, a musical piece adheres to a musical style, breaking down into precise concepts of timbre style, performance style, composition style, and the coherency between these aspects. Here, we study and analyze the current advances in music generation using deep learning models through different criteria. We discuss the shortcomings and limitations of these models regarding interactivity and adaptability. Finally, we draw the potential future research direction addressing multi-agent systems and reinforcement learning algorithms to alleviate these shortcomings and limitations.

INDEX TERMS Deep learning, multi-agent systems, music composition, music creativity, music generation, music information retrieval, neural networks, reinforcement learning.

I. INTRODUCTION

Computers have introduced a new way of approaching music composition to create an elaborate piece of music. There are several approaches for the algorithmic composition of music [1], such as mathematical models [2], knowledge-based systems and grammars [3], evolutionary methods [4], and Markov models [5]. Although these models have shown the ability to create melodies in various styles such as [6] and [7], they lack generalization [8] and, in some cases, require manual preparation of rule-based definitions for different types of music. In contrast to handcrafted models, machine learning models, and particularly deep learning (DL) models, can learn from large distribution of musical examples and generate new content. Besides, deep learning models exhibit strength in processing raw unstructured data by extracting higher-level features associated with the task. Mainly, music generation comprises subtasks like melody and multi-instrument generation, style transfer, and audio synthesis. Models such as DeepJ [9], DeepBach [10], and BachBot [11] can mimic a particular musical style with plausible results. JukeBox [12] can generate complete high-quality songs with singing in raw audio in an end-to-end approach.

Despite this promising progress, there are challenges in using end-to-end deep generative models for music generation. These models often suffer from the scarcity of musical structure, expressiveness, and creativity. Besides, there is no unified music evaluation method for deep learning models [13]. Furthermore, these models are primarily limited in interactivity and controllability. It is demanding for artists to generate creative and genuine content using end-to-end models [14]. Consequently, it is essential to have a clear perspective of challenges and problems to improve the performance and ability of these models.

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Jiang.

In this study, we provide an overview of the advances in deep learning methods for music generation in the symbolic domain. We further outline different evaluation techniques and the challenges and limitations of these models in music generation tasks. Additionally, we summarise these models' characteristics and the challenges they addressed, including 73 deep-learning models. Accordingly, we describe a potential approach to overcome these issues. Our study concentrates explicitly on adaptability and interactivity issues by demonstrating a better approach using multi-agent systems and reinforcement learning algorithms.

This paper is organized as follows. Section II briefly introduces various aspects of the music generation task. Section III presents different domains of music representation. Section IV sorts out the common deep learning architectures of generative models. Section V deals with the methods for music generation, categorized based on the architectures in Section IV. Section VI presents different music evaluation methods from objective and subjective points of view. Section VII points out some shortcomings of current methods and challenges in the music generation task. Section VIII exposes the potential future research direction. Finally, Section IX concludes this paper.

II. ASPECTS OF MUSIC GENERATION TASK

The objective of the music generation task refers to the musical content to be generated. Reference [15] determines the music generation objectives with five aspects: type, destination, use, mode, and style. The most important factor among these five aspects is the type, which defines the nature of the music generation model. In this context, we can classify the main musical types as single-track monophony, single-track polyphony, multi-track polyphony, and accompaniment. The single-track monophony represents the sequence of notes with at most one note at a time for a single instrument or vocal.

In comparison, single-track polyphony represents more than one note at a time. Examples of single-track polyphony instruments are the piano and guitar. While single-track monophony and polyphony are for a single instrument, multi-track polyphony is intended for more than one voice or instrument. Multi-track polyphony can capture a complete band, such as a Jazz trio with piano, bass, and drums, and it constitutes the traditional recording format. Additionally, the accompaniment can be rhythmic or harmonic support (or both) to a given melody, like chord progression and counterpoint. Note that this is only one of several ways to classify musical types, but useful in discussing music generation tasks in this study.

The mode aspect defines whether humans can intervene in the music generation process or if it is fully automated. The interactive ability of a musical system provides some degree of control over the content generation. Based on the mode, we can determine the destination and use of the generated content. For instance, the generated musical content can be played by an audio system (waveform), processed by sequencer software (Musical Instrument Digital Interface (MIDI)), or performed by a human (score). Moreover, the generated musical content can be influenced by the style of certain musicians such as for example Bach. Indeed, the choice of training examples directly affects the model's learning outcome regarding the musical style.

III. REPRESENTATION OF MUSIC

Musicians work with many levels of inference, ranging from abstract symbolic representation like the lead sheet to the continuous and concrete representation of audio signals. We can divide music into symbolic and audio domains [16]. Mainly, the symbolic domain consists of discrete variables, while the audio domain is continuous. Additionally, the symbolic domain includes a representation referred to as performance control. Considering the multi-level and multi-modal characteristics of music representation:

- The high level is the score representation, including the structure and symbolic features (like note, pitch, and chord). It is an abstract representation of music that enables musicians to develop and communicate musical ideas seamlessly.
- The middle level is the performance representation consisting of detailed timing and dynamics for the musical expression. The performance representation conveys the changes in emotion and information, which are not marked in the score but performed by the musician.
- The bottom level is the audio representation related to acoustic features, such as timbre, that can be determined as a sound.

The music generation can be addressed relative to each of these levels. Deep learning models and computer programs generally solicit a precise definition of input representation. In this study, we concentrate on deep learning models for the symbolic representation of music.

IV. DEEP LEARNING

In recent years, deep learning has seen many advancements in the architecture of generative models. The most utilized generative models in the music generation tasks are Recurrent Neural Networks (RNNs), Variational AutoEncoders (VAEs), Generative Adversarial Networks (GANs), and Transformers. Additionally, researchers have investigated the potential of reinforcement learning algorithms for music generation. This section outlined the architecture of these methods.

A. RECURRENT NEURAL NETWORKS

Recurrent Neural Networks (RNNs) are neural network architectures suitable for learning the sequence of data. They can capture the time dependencies between input sequences by sampling from the neuron's output and feeding in the

sample as input in the next time step. However, due to the gradient vanishing problem, RNNs struggle to learn long-term dependencies within the input sequences. The Long Short Term Memory (LSTM) network [17] is an advanced type of RNN that comprises layers of neurons with recurrent connections. LSTM contains a computational unit called a memory cell or memory block, consisting of weights and gates connected recurrently. The network can interact with memory cells through the gates that increase the number of parameters to be estimated during training. In this manner, the network can control the flow of information in detail for each cell, resulting in faster convergence.

B. GENERATIVE ADVERSARIAL NETWORKS

Generative adversarial networks (GANs) [18] are another family of deep generative models. The main idea is to train two neural networks at the same time. The GAN's architecture includes the generator G and the discriminator D. The generator learns a distribution of the input data during the training process to resemble the actual samples. At the same time, the discriminator takes examples of the real (input examples) and generated data (output examples by generator) and attempts to maximize the probability of assigning the correct label to real and synthetic (generated) data. Indeed, the training process of GAN forms a two-player MiniMax game in which the models are trained until the discriminator is fooled half the time.

C. VARIATIONAL AUTO-ENCODERS

The variational autoencoders (VAEs) [19] are powerful deep generative models. They have shown an excellent capacity to produce various high-quality content such as images, texts, and sounds. VAE is an autoencoder (AE) with constraints on encoded representation (latent variables), denoted by the variable z. The applied constraints ensure that the encoder produces latent variables with a predefined structure and properties.

To elaborate, AE is a neural network with one hidden layer in which the output layer (decoder) reflects the input layer (encoder). In other words, the encoder compresses each example in the dataset into a vector of numbers (latent variables) to create the latent space of the dataset. The decoder reconstructs the same examples using the latent variables. However, it is difficult to ensure the regularity of the latent space organized (encoded) by the encoder. The training regime in AE results in encoding and decoding with no information loss, which indicates the overfitting problem. Therefore, the decoder prunes to generate poor quality content caused by a lack of structure in the latent space.

The VAEs architecture alleviates the latent space irregularity issue by encoding the examples following a probability distribution $P(z)$ like the Gaussian distribution. In this manner, VAEs ensure a better structure of latent space by forcing the encoder to return a distribution over the latent space instead of a single point.

D. TRANSFORMERS

Transformers [20] have been used widely in natural language processing (NLP) [21], and computer vision [22] tasks with outstanding performance. Transformers architecture relies on an attention mechanism that computes the representation of its input and output by concentrating on some specific elements of the input sequences. Particularly, the transformers belong to the family of sequence-to-sequence models. Their architecture includes an encoder and decoder, yet recognizable to AE models and backpropagation-based learning. The given inputs are prepared as tokens to train the transformer model, which is a structured representation. In this manner, the positional information is preserved, which enables the model to determine temporal dependencies within the input sequences.

E. REINFORCEMENT LEARNING

In Reinforcement Learning (RL), an agent learns to interact with an environment through trial and error. The agent selects and performs actions sequentially within the environment. Each action takes the agent into a new state, where the agent receives a reward. The given reward relies on the fitness of the action to the current state (environment). The agent's goal is to learn an optimal policy to maximize its cumulative rewards (gain) through the learning process. Indeed, the agent maximizes its gain by knowing when to explore to learn more and when to exploit what it has learned.

For instance, Q-learning is a model-free reinforcement learning algorithm in which the agent learns to estimate the value of an action in a particular state. Q-learning is model-free as it does not require assessing the dynamics of the environment as in the case of transition and reward function. Indeed, Q-learning is a value-based learning algorithm that updates the value function based on an equation (Bellman equation). The agent maintains the estimated values in Q-table and updates the table's values during its interaction with the environment.

For an overview of approaches and algorithms for reinforcement learning, we refer to [23] and [24]. In this study, we mainly concentrate on using deep learning in reinforcement learning algorithms, known as deep reinforcement learning (DRL). Reinforcement learning emerges to be a promising approach to the music generation task. It can enhance the interactivity issue of the deep learning architecture through control methods like the reward mechanism. Furthermore, DRL algorithms an process large input examples, which is important in the case of music generation tasks.

V. RELATED WORK

This section studies the current advances and state-of-the-art approaches to music generation using deep learning techniques in the symbolic domain. We discuss the approaches based on the architectures mentioned in Section IV. Note

that some of the mentioned papers are under peer review or preprint.

For a comprehensive overview of the deep learning models for music generation, we refer to [15], [16], [25], and [26]. References [1], [27], [28], [29], [30], [31], and [32] provide an overview of the music generation systems, and algorithmic composition of music. Additionally, the authors in [33] survey the application of robotics in music generation tasks.

A. RECURRENT NEURAL NETWORKS

Following the success of LSTM architecture [17], Eck and Schmidhuber [34] used LSTM to address the lack of global coherence in algorithmic composition since they are better for learning temporal dependencies than vanilla-RNN. Their work demonstrated the LSTM network's ability to learn the local and global structures and reproduce long-term conventions. However, the network's tendency to bind with the training set conventions stop exploring and producing new musical forms. To further improve the performance of the LSTM model in the music generation task, Eck and Lapalme [35] proposed a music-specific sequence learner that can capture long-timescale structure in the musical piece. They introduced a bias toward the metrical structure to confront the network's problem to learn repetitive musical sequences by providing time-delayed copies of input.

Sturm et al. [36] used a character-based approach that works with a vocabulary of single characters, with textual transcriptions of folk music, to train a deep LSTM model. The training examples contain 24,000 high-level transcriptions of folk tunes in "ABC" notation with a vocabulary size of 134. The input representation carries the one-hot encoded input vectors similar to [35] with the softmax output layer providing the distribution over the vocabulary adapted to the input. They developed two models, charRNN trained on a consecutive text file, and folkRNN trained on single complete transcriptions. Similarly, Choi et al. [37] used word-based learning (wordRNN) in addition to character-based learning (charRNN) for automatic music composition. They utilized textual data representation to generate Jazz chord progressions and Rock music drum tracks. In the preprocessing step, the start and end flags indicate the score's beginning and end. They transposed all the scores to the key of C. For drum tracks, they used a binary representation of pitches to encode drum components where only nine components were included for training efficiency.

Li et al. [38] proposed a novel technique to improve the performance of LSTM RNN models to learn long-temporal dependencies. The proposed model is named Enhanced Memory Network (EMN), which consists of several recurrent units known as Enhanced Memory Units (EMU). EMN incorporates musical beat information and historical hidden states to improve the learning ability of LSTM RNN. Medeot et al. [39] proposed StructureNet that learns musical structure space to generate melody. StructureNet includes two networks: structure and melody model. The structure model induces the musical structure within the given training examples (melodies) and encodes them as a sequence of binary vectors. They used the trained structure model to steer the melodies generated by the melody model during the generation process. The melody model is a probabilistic model that predicts the probability distribution of musical events.

Similarly, Dai et al. [40] introduced MusicFrameworks for controllable melody generation using hierarchical music structures. Their system composes a melody by arranging a musical piece into sections and phrase-level structures. Then, it generates rhythm and basic melody parts using two transformer-based models. Finally, the system generates the final melody by conditioning on various musical attributes. Keerti et al. [41] utilized Bi-directional LSTM RNN to compose polyphonic Jazz pieces. Their model employs the attention mechanism to identify the parts of the input sequences with salient musical features. For similar approaches to address the structure in music using LSTM RNN, we refer to [42], [43], and [44].

Although the above systems can generate musical content, their generations lack musical expressions. Oore et al [45] proposed PerformanceRNN to address the expressiveness in music. They utilized a dataset of recorded human performances, including notes' exact timing and dynamics. Hadjeres and Nielsen [46] proposed AnticipationRNN to implement positional constraints on model's generation. Their architecture and method provide interactivity to the RNN-based model, enabling the users to perform positional constraints on notes.

We often desire to generate music based on sentiment or a specific music style. Ferreira and Whitehead [47] proposed a method to control the deep learning model and generate musical pieces using a specific sentiment. Their generative model includes an LSTM network paired with a Logistic Regression model. Their model also shows the potential to perform sentiment analysis of symbolic music. Furthermore, they provided a labeled dataset of symbolic music annotated according to sentiment for future research. Cífka et al [48] presented a style transfer method to generate polyphonic accompaniment styles for Jazz. They trained neural networks with encoder-decoder architecture in a supervised manner on synthetic parallel training data labeled by the styles of music [49]. The training data includes chord charts from a chord language model of the Jazz music standards and rhythmic variations. The sampled chord charts are prepared as a token of the chord's root, quality, and duration. To evaluate the model's performance, they used the content preservation technique to estimate how well the model captured the harmonic structure and the style fit technique to measure how well the output matched the desired style.

Chen et al. [50] used the chord progression as constraints to generate melody using the WaveNet. Their work compares temporal-CNN and LSTM RNN models systematically.

Furthermore, they propose a technique to encode chords and melodies in a staggering representation. They used the Information Dynamics method to analyze and evaluate the content generated by the model through pattern identification. Lu et al. [51] proposed MeloForm, an expert system to compose music according to a musical form. Their system consists of two modules: expert systems and a transformer model. MeloForm can generate different forms of music, such as verse and chorus, rondo, and sonata forms. The expert systems module utilizes the handcraft rules (music theory) for melody generation. The transformer model refines the generated melody using various strategies, such as refining phrase by phrase, conditioning on harmony and rhythm, and others.

Ziegler and Rush [52] utilized the normalizing flow architectures for generative models to compose the melody and polyphonic music. In their approach, they considered character-level language modeling and polyphonic music generation, where the normalizing flow method models the continuous representation of the input sequences.

B. VARIATIONAL AUTO-ENCODERS

An example of a VAE-based model for music composition tasks is MusicVAE [53] for monophonic and polyphonic music. The architecture of the proposed model includes an encoder and a decoder with a two-level hierarchical RNN structure. They utilized a corpus of MIDI files collected from the web to extract monophonic melodies, drum patterns, and trio sequences (drum, bass, melody (piano or guitar)). They trained the model on 2 or 16 measures long for monophonic melodies and drum patterns, and 16 measures long for trio sequences. Furthermore, by utilizing the latent space of the VAE architecture, the model can generate musical content through different operations like translation and interpolation.

Later, Simon et al. [54] proposed an extension of MusicVAE, called multi-track MusicVAE, to generate musical pieces with an arbitrary number of instruments. In their model, both the encoder and decoder adopted the hierarchical architecture. Similarly, by benefiting from VAE latent space, the model has the capacity to generate samples by chord conditioning. Dinculescu et al. [55] proposed a new method to learn the latent space of the MusicVAE model to enhance conditional sampling. They achieved this by employing the latent constraints [56] to lower the dimension of the latent space, concentrating on the portions that are similar to a particular style or genre. Wang et al. [57] proposed a novel tree-structure model called PianoTreeVAE by addressing the hierarchical structure of music. The architecture of the network resembles the tree structure, where each node represents the embeddings of musical elements with bidirectional edges.

Liang et al. [58] proposed MIDI-Sandwich2, a hierarchical VAE-based model for polyphonic music generation. In contrast to other hierarchical VAE-based models, they used RNN instead of CNN models to build the generative model. Their model utilizes Binary VAE (BVAE) method to handle various multi-track music information. Mittal et al. [59] presented a new approach to utilizing probabilistic diffusion models for melody generation. Their training regime includes first training the VAE model on input sequences and then training the diffusion model to learn the VAE latent space. Indeed, the diffusion model is trained to learn long-term dependencies and expand the ability of the VAE model to generate long sequences, in this case, 64 bars.

Chen et al. [60] introduced Music SketchNet, a novel guided music generation framework. Their model is intended to complete the missing parts of musical measures, given the musical piece and related parameters as input. The input parameters are pitch contours and rhythm patterns defined by the user. The model's architecture consists of three components: SketchVAE, SketchInpainter, and SketchConnector. SketchVAE is a VAE model that encodes and decodes the training examples into high-dimensional latent variables, while SketchInpainitng is a stacked RNN model that handles the prediction of musical ideas by utilizing the latent variables. SketchConnector combines the predictions from SketchInpainitng and musical ideas given by the user to carry out the final latent variables. The decoder of the SketchVAE receives these latent variables to generate music output.

Akbari and Liang [61] proposed a semi-recurrent CNN-based VAE-GAN model for melody generation. The model includes the encoder, generator/decoder, and discriminator. They put the VAE decoder and the GAN generator under one hood, where they shared the parameters and trained together. The encoder encodes the input sequences and constructs the latent representation. The generator/decoder utilizes the latent variables to carry out the output. Then, the discriminator module receives the real (original training examples) and fake (generated output) data. They trained their model for piano music generation. Similarly, Brunner et al. [62] proposed MIDI-VAE for polyphonic music generation and modeling the dynamics of music. Their model includes a VAE model paired with a style classifier which navigates the encoder in VAE to construct the latent space based on the style information. Their model can perform style transfer by changing the attributes such as pitch, velocity, and instrument of a musical piece.

Wang et al. [63] introduced hierarchical variational recurrent auto-encoders (VRAE) to model polyphonic music. They used normalized note representation proposed by BachProp BachProp [44] and multiple embedding layers to project each melodic feature. For the encoder, they utilized four GRU layers to construct the latent representation of the melodic features given by the embedding layers. The decoder has a similar architecture with 7 GRU layers for modeling attribute-specific context, combining multiple attributes, and generating corresponding note attributes. The architecture of their model represents the capability to generate dynamic music with various time signatures.

Tan and Herremans [64] proposed Music FaderNets, a framework that utilizes latent variable models to learn high-level musical features through the low-level representation of music. They used Gaussian Mixture Variational Autoencoders (GM-VAEs) as their model architecture to capture low-level musical attributes latent space. Indeed, by employing such hierarchical latent space architecture, they could derive high-level musical attributes from low-level representations. Music FaderNets provide an interactive and controllable generation by tweaking the low-level musical features. This possibility is appeared as sliding knobs and is inspired by visual controllers in Fader Networks Fader Networks [65].

Pati et al. [66] proposed music inpainting, a technique to traverse the latent space of VAE models. Inpainting is a task in which the purpose is to refine or complete the missing parts of a media [67]. Their model can generate content based on past and future musical contexts in an interactive manner.

C. GENERATIVE ADVERSARIAL NETWORKS

Mogren [68] represented one of the earliest use of GAN-based music generation models. Their model, C-RNN-GAN, is an RNN model with adversarial training using a continuous sequence of data. They used real-valued continuous quadruplets of frequency, length, intensity, and timing as musical features to model the musical signals. Later, Guimaraes et al [69] proposed ORGAN, a new GAN-based approach to compose polyphonic music. ORGAN architecture includes an LSTM RNN for the generator and CNN for the discriminator. It uses a reinforcement learning (RL) based reward function representing domain-specific metrics to train the generator model.

Multi-track polyphony music includes multiple voices independent in terms of time. Each of these voices has its temporal dynamics, layered on top of each other to shape the desired sound. Dong et al. [70] proposed MuseGAN to generate multi-track polyphonic music. MuseGAN is the integration and extension of generative and temporal models. The generative models are forward multi-track music generators based on WGAN-GP [71], including composer, jamming, and hybrid models. Each generative model can generate multi-track music bar by bar, following a specific scenario. Therefore, they proposed temporal models to generate multiple bars with temporal structure and coherency. Nevertheless, the music generated by MuseGAN is inconsistent in musical segments and harmony and contains fragmented notes [16]. The instrument set in MuseGAN is a fixed quintet composed of bass, drum, guitar, piano, and string.

The instability of GAN-based models for music generation is mainly due to the use of convolutional layers in their architecture to extract features [16]. Indeed, the CNNs are not effective in capturing the temporal dependencies. Therefore, Guan et al. [72] proposed Dual Multi-branches GAN (DBM-GAN) to overcome the lack of consistency. DBM-GAN integrates the self-attention mechanism in its architecture to learn temporal dependencies and extract spatial features. Besides, the model's multi-branch architecture enables the arrangement of various instruments across time. Similarly, Valenti et al. [73] proposed the first music adversarial autoencoder called MusAE. MusAE uses adversarial regularization instead of the Kullback–Leibler (KL) divergence in VAEs. It can reconstruct new phrases and interpolate between latent representations to change specific musical attributes.

Liu and Yang [74] defined a new music generation task called lead sheet arrangement for multi-instrument music generation. The proposed model takes the lead sheet as input and generates accompaniment for the given melody with instruments such as guitar, bass, piano, strings, and drum. The model architecture includes a recurrent convolutional network with adversarial training composed of three stages: lead sheet generation to generate lead sheets of eight bars from scratch, feature extraction to extract harmonic features, and arrangement generation stage to generate five-track piano-rolls of one bar, respectively.

Angioloni et al. [75] introduced CONLON to generate polyphonic and multi-instrument music. Their work presented a Wasserstein autoencoder (WAE) model trained on lossless input representation, including the velocity and duration information from MIDI data in two separate channels. The proposed generative process includes exploring the WAE model's latent space based on interpolation to maintain consistency between transitions and variations within the generated musical piece.

One of the exciting tasks within the music generation field is the ability to transfer a musical piece from one domain to another. Notably, we like to obtain a mapping function that learns and underlines the attributes and characteristics of musical structure. Accordingly, Chen et al. [76] proposed a GAN-based model with a dual learning method to combine music across multiple domains. They utilized the Wasserstein-based metric to approximate the distance between the target and existing domains and represent the model's learning progress. Furthermore, Brunner et al. [77] explored the ability of the CycleGAN-based model [78] for music genre transfer in the symbolic domain of music. The CycleGAN architecture includes two GANs arranged in a cyclic manner and trained together, in which one generator transfers data from domain A to B and the other from B to A. One discriminator is tied to each generator's output to identify the fake and real outputs. Later, Brunner et al. [79] further analyzed the influence of spectral normalization and self-attention on GAN training using the proposed model in [77].

Tokui [80] proposed an extended GAN model to compose genre-conditioned music rhythm patterns. To do so, they added a second discriminator model with genre ambiguity loss to classify the genre of the generated musical piece. Particularly, the genre ambiguity loss is a cross-entropy loss [81]. In this manner, the generator is encouraged to

generate new content in a new musical genre. Similarly, Lattner and Grachten [82] proposed a convolutional variant of the gated autoencoder (GAE) to generate music rhythm patterns. Their model encodes the rhythmic interactions of the kick drum against bass and snare patterns and captures the local relations between them.

D. TRANSFORMERS

The attention mechanism facilitates the extraction of spatial and temporal dependencies but depends on absolute positions in its inputs. Therefore, it struggles to track the dependencies in music, such as regularities, event orderings, and periodicity. To alleviate this issue, Shaw et al. [83] proposed the relative attention mechanism, which focuses on relational features by approximating the distance between two tokens. Huang et al. [84] proposed Music Transformer that exhibits the relative attention mechanism to generate polyphonic music. The model can learn the long-term musical structure to develop long melodies or continue a given motif. Similarly, Payne [85] created MuseNet based on GPT-2 that can generate a long musical piece with ten different instruments in various styles. Nevertheless, Music Transformer and MuseNet lean to generate random notes and harmonies after a few bars [16].

Many attempts have been made to overcome the issue of randomness and generate pieces with a high musical structure. Zhang [86] proposed a novel adversarial transformer, which combines generative adversarial learning with the attention mechanism. The adversarial objectives facilitate the transformer to concentrate on temporal dependencies within the musical structure. Compared to Music Transformer and MuseNet, their model depicts advancement in musical quality for a monophonic and polyphonic generation. Similarly, Jiang et al. [87] proposed TransformerVAE, a combination of VAEs and transformers. Their approach benefits from MusicVAE hierarchical structure and attention mechanism in transformer models for representation learning. Huang and Yang [88] expands the learning ability of the generative models by introducing a new approach for discrete representation of music. They proposed revamped MIDI-derived events (REMI), an explicit metrical grid that extracts the hierarchical structure of music using events such as Chord, Bar, and Position. Their study experimented with transformer-based models, where they examined various musical features to capture higher-level characteristics of music.

Peracha [89] concentrated on the sequential modeling of polyphonic music instead of the network architecture. Their study experimented with a multi-layer transformer encoder and a GRU-based model named TonicNet using the JSB chorales dataset.[1] Their results depict improvement in both models' performance by introducing new salient musical features in the form of chords and intra-voice token repetition. Dai et al. [40] presented Music Frameworks to generate customizable full-length melodies. Music Frameworks inherits a hierarchical architecture to represent high-level musical features such as repeated sections and phrases, and low-level features such as rhythm structure and melodic contour. Music Frameworks can generate long-term music structures conditioned on the basic melody and rhythm structures. Wu and Yang [90] proposed MuseMorphose to generate full song and perform style transfer. Their model represents an ability to generate long sequences with fine-grained controllability and conditioning over musical attributes such as rhythmic intensity and polyphony.

Zhang et al. [91] proposed a transformer-based model that learns and captures the harmonic attributes of the musical structure, such as form and texture. Rütte et al. [92] proposed FIGARO, a novel self-supervised task called description-to-sequence, that can generate music based on the defined descriptions with global and fine-grained control. Their model includes two distinct description functions: *learned* and *expert* modules. The *learned* module extracts the salient musical features using the constructed low-fidelity, human-interpretable sequences by the *expert* module. For music generation, they utilized a transformer-based model that receives the extracted features by *learned* and *expert* modules. For similar approaches to address the structure and control in music using Transformer, we refer to [93], [94], [95], [96], [97], [98], and [99].

Zou et al. [100] introduced MELONS, a full-song melody generation framework using a graph representation of music and transformers model. MELONS generation process includes structure and conditional melody generation. Their work concentrates on the generation of pop music by constructing eight types of bar-level relations to represent the musical structure. Furthermore, they used a directed graph to describe the melody structure of a song using bar-level relations. MELONS architecture includes two transformer-based generation models: structure and melody generation. The structure generation models and generates the structure graph as a sequence of relations. The melody generation uses event-based music representation to compose conditional or unconditional structured melodies. The unconditional generator is trained on the original training data, while the training data for the conditional generator is organized according to the specified condition.

Liu et al. [101] introduced a novel approach to composing symphony music. Their study presented Multi-track Multi-instrument Repeatable (MMR) and Music Byte Pair Encoding (BPE) methods to model and represent symphony music. MMR models symphony music by separating and capturing repeated instruments within a single track. On the other hand, Music BPE is a BPE-based algorithm to tokenize and preprocess the musical examples by considering the concurrence of the notes. Their model inherits transformer-based model architecture with 3-D positional embedding that compresses the spatial and structural details of the input sequences. Furthermore, they gathered and processed a large-scale corpus of symphonic music, which is made publicly available.

[1] http://www-ens.iro.umontreal.ca/ boulanni/icml2012

Furthermore, Shih et al. [102] introduced a theme-based method to condition the generative model. Their model uses contrastive learning [103] and density-based [104] methods to cluster similar fragments of a musical piece to form a latent space. In this manner, they formed an augmentation strategy to generate various variations of musical examples for each cluster to train the transformer-based model. Besides, they utilized the same clustering approach to generate new test examples and evaluate the model. Hawthorne et al. [105] proposed TransformerNADE, a transformer-based model for expressive piano performances. To generate meaningful piano performances, they proposed a new representation using NADE [106]. Their model architecture is inspired by RNN-NADE [107].

Training deep learning models often requires a large amount of data. Researchers have used methods such as transfer learning to solve problems in case of data scarcity [108], [109]. For music generation, Donahue et al. [110] presented the benefit of transfer learning to improve transformer-based model performance. They also employed data augmentation methods in their study. Similarly, Hung et al. [111] examined the outcome of two transfer learning methods for the Jazz music generation. Their work studied *model fine-tuning* and *multitask learning* methods for unconditioned melody generation.

E. REINFORCEMENT LEARNING

Although the automatic music generation can inspire human creation, it is limited to certain musical examples such as Bach. Interactive music generation can help enhance the sample generations by incorporating human objectives and preferences in the music creation process. Jaques et al. [112] proposed RL-Tuner, a reinforcement learning model to generate music using user-defined constraints. The RL-Tuner architecture includes two deep Q networks and two RNN models. One RNN model, called NoteRNN, is trained on the dataset of melodies. The second RNN model is a copy of NoteRNN, called RewardRNN. The Q network goal is to learn to select the following note (action) based on the generated melody so far (state). The second Q network is called the Target Q network in parallel to the Q network. The Target Q network is trained to estimate the accumulated rewards (gain) achieved by NoteRNN. The Q network's reward combines RewardRNN output and adherence to music theory constraints. Kumar and Ravindran [113] used LSTM RNN with RL to compose melody and basic chords. They processed the polyphonic pieces by dividing them into a stream of monophonic examples. They trained the LSTM model on these examples and created an RL agent to find a suitable combination of songs.

Later, Jiang et al. [114] proposed RL-Duet for online accompaniment using reinforcement learning. It can generate melodic and harmonic music responses to the human part. RL-Duet uses actor-critic with a generalized advantage estimator (GAE) for the reinforcement learning architecture. They introduce a reward function that considers the fittingness of the inter-part and intra-part of the generated notes in horizontal and vertical perspectives. The reward model is learned from monophonic and polyphonic examples instead of hand-crafted composition rules and criteria utilized in RL-Tuner.

Subsequently, Liu et al. [115] proposed RE-RLTuner, an extension to RL-Tuner that uses the Latent Dirichlet Allocation (LDA) as a musical feature extractor. The LDA extractor represents the musical structure characteristics by clustering music at different scales (musical segments) and extracting the musical features into three aspects called topics. The topic models maintain different music structure information. The architecture of the model is similar to RL-Tuner. The network's reward combines the reward model (RewardRNN) and topic models extracted by the LDA extractor.

F. OTHERS

This study mainly focuses on deep learning methods for music generation. However, researchers investigated and examined other approaches along the deep learning methods to tackle music generation tasks. For instance, Moulieras and Pachet [116] introduced a new approach for melody generation using the maximum entropy statistical model [117]. In this approach, the melodies are considered a network of interacting notes. The model assigns a probability distribution to this network and learns the statistical dependencies of the pitch sequences. Later, Hadjeres et al. [118] and Moulieras and Pachet [116] extended the model to handle polyphonic music with multiple voices and generate expressive music, respectively.

Zhao and Xia [119] proposed a hybrid model that can generate piano accompaniment based on a lead sheet. Their model includes phrase selection and neural transfer models to generate content. Phrase selection is a rule-based model that carries out the phrase montages from the database. The neural transfer model receives the phrase montages and manipulates them to match the corresponding style of the given lead sheet. Furthermore, the model's output can be conditioned on rhythm density and voice number.

VI. EVALUATION

Researchers use diverse methods to evaluate deep learning models for music generation. These methods mainly depend on the model's output, which can be subjective or objective. Often it is viable to perform the subjective evaluation in music generation tasks as they involve creativity. However, a thorough subjective evaluation requires an appropriate experimental design and resources to produce reliable, valid, and replicable results [120]. Consequently, the objective evaluation methods facilitate the evaluation of the generative models by providing comparable and relevant results. Indeed, by utilizing objective methods, it is easier to control the variables entangled in the test and reduce bias. The final evaluation results are obtained from both subjective and objective approaches for a better model assessment and a

reliable scientific benchmark. This section covers the current evaluation methods for music generation tasks. We refer to [16] and [121] for complete review of music evaluation methods.

A. SUBJECTIVE EVALUATION

The subjective methods evaluate the model's generated content in terms of creativity and novelty. It is essential to evaluate the music from a subjective stance, as a musical piece consists of perceptual qualities that numerical metrics can not measure. Among the available listening tests [16], the Turing test is a standard method for subjective evaluation [122]. This model was introduced by Alan Turing [123] to answer the question: ''Can a machine think?''. In the case of music generation tasks, the questions often include whether the generated content is aesthetically pleasing and whether it is composed by a human. During the Turing test, the human listener tries to differentiate the machine-generated from the human-created piece. Two examples of models of the Turing test for music generation systems are the musical directive toy test (MDtT), and the musical output toy test (MOtT) [124]. The MDtT depends on musical directives such as genre, style, or melodic or rhythmic fragments, while MotT is free from musical directives. Both of these models are only dependent on the human listener's judgments.

Overall, to obtain a valid listening test, [13] specifies some requirements:

- A sufficient number of listening subjects with diverse musical knowledge to obtain meaningful statistical results;
- The subjects are evenly distributed based on their musical knowledge, including the amateurs with no or basic music knowledge and experts in the field;
- Experiments are performed in a controlled environment under specific acoustic characteristics and equipment;
- Each subject receives the exact instructions and stimuli.

Note that each of these requirements confines a study's degree of accuracy and repeatability. Furthermore, it is possible to utilize online platforms to conduct listening tests. For example, crowdMOS [13] is a platform for subjective listening tests using Amazon Mechanical Turk. CrowdMOS contains a set of freely distributable and open-source tools that delivers quality results by detecting and discarding inaccurate or malicious submissions. Défossez et al. [125] used crowdMOS in their study to obtain Mean Opinion Score for the ground truth samples.

Another method of subjective assessment of music is the visual analysis that is conducted by a human expert. The methods in visual analysis utilize visual representations like score, waveform, and spectrogram instead of the auditory form of music. For instance, the authors in MuseGAN Engel et al. [70] performed score analysis on different aspects of generated melodies, such as stability and smoothness analysis of the chord and rhythm patterns. Engel et al. [126] performed spectrogram analysis by employing the Rainbowgram to compare the reconstructed notes of different instruments with the original audio.

B. OBJECTIVE EVALUATION

The objective evaluation methods measure the model's performance and generated content. We can measure the model's performance using numerical metrics such as loss and accuracy. While for evaluation of the generated content, we use statistical descriptors derived from musical concepts. In the following, we explain each of these measurement methods.

Numerical metrics do not contain music domain knowledge and only represent the model's ability to process the data. It is common to use numerical metrics like loss and perplexity during the training process. They mainly consider the statistical distribution of the generated samples or classification accuracy. For instance, loss indicates the difference between inputs and outputs from a mathematical perspective, while perplexity evaluates the model's generalization capability [127]. Additionally, Jeong et al. [128] used mean squared error (MSE) and correlation metrics to assess the model's performance ability using the generated performance and human performance characteristics. Similarly, Gillick et al. [129] proposed metrics such as Timing mean absolute error (MAE), Timing MSE, Velocity and Timing Kullback–Leibler (KL) divergence to measure the model's performance.

Besides the numerical metrics, we can evaluate the generated music by utilizing methods such as log-likelihood and density estimation [130], [70], [131], [132]. For instance, Huang et al. [131] proposed a frame-wise evaluation of the generated content by calculating the negative log-likelihood between the model's output and the ground truth. However, based on the observations of the Theis et al. [133], the probabilistic measure is not always consistent, as generative models can produce irrelevant samples and represent a perfect probabilistic measurement. Other techniques such as chord classification [134], style classification [77], style likelihood [77], and reconstruction accuracy [53] are examples of metrics for specific tasks.

To improve the interpretability of the generative system's outcome, researchers proposed musical metrics by integrating the musical domain knowledge. These metrics provide a detailed evaluation concerning specific music characteristics. Ji et al. [16] categorizes these metrics into pitch-related, rhythm-related, chord/harmony-related, and style transfer and provides a comprehensive overview of these methods. As an example, Sabathé et al. [135] proposed a novel evaluation method using the Mahalanobis distance [136] by using high-level symbolic music descriptors to describe the musical samples. Yang and Lerch [121] introduced a musical metric using absolute and relative metrics. They represent a practical and reproducible approach to evaluating the model's performance and generated content. Their evaluation framework has been used by [111], [137], [138], and [114].

Furthermore, there are evaluation methods to assess specific musical aspects using other theories or algorithms. Variable Markov Oracle (VMO) [139] is a method to evaluate the repetitive patterns in a musical piece. [10] introduced a technique to assess the originality and creativity of a piece and avoid plagiarism. Minimum Distance Classifier (MDC) [140] is a method to determine the style similarity of the generated content with the expectation style. Lattner et al. [141] utilized Humdrum toolkit [142] to evaluate the tonality of the generated musical piece. Wu and Yang [93] used the Scape plot [143] to capture, visualize, and compare the repetitive structure of the generated piece with the original examples.

VII. CHALLENGES

Compared to traditional approaches, deep learning methods have shown great capabilities in the music generation task. However, there are still many difficulties and challenges in using deep learning to generate music. Indeed, the multi-modal nature of music makes the field of music generation with deep learning even more challenging. On the other hand, the black-box nature of deep learning models makes it hard to diagnose their learning process. Here, we address some challenges deep learning models face in music generation tasks.

A. STRUCTURE

A musical piece evolves over time through the development of musical ideas. The musical structure refers to the arrangement of these musical ideas as a whole. Particularly, the musical structure consists of local and global structures. Global structure relates to the long patterns, extended multiple bars like AABA. On the other hand, local structure relates to each musical idea repeated or developed to create themes and variations. Although much work has been done to model and generate music, making a complete musical piece is still challenging. In most cases, the generated content by deep learning models gradually becomes tedious as there is no clear sense of direction, and it may end unexpectedly.

Researchers have investigated various methods for better structure representation. Models such as [100] used graph representation of melody with eight types of bar-level relations such as repetition, transposition, rhythmic sequence, and harmonious cadence. Other models, such as [53], [54], and [58], utilized hierarchical architectures to address this issue. The template-based method proposed by Zhou et al. [42] has shown the ability to generate a specific overall structure. The harmony-Aware Hierchical model proposed by Zhang et al. [91] improved the issue further, possessing the ability to imitate the outline structure of real music. Nonetheless, the generated content by these models still lacks musical details and requires refinement to present an actual musical piece.

B. REPRESENTATION

The representation in nearly all of the current deep learning models involves the pitch and duration of notes, and primarily triads for chords [16]. This simplification restricts the musical understanding of the deep learning models to generate quality musical content. Furthermore, the current methods use relatively simple mechanisms to model instrument characteristics. For instance, it is challenging to model the piano's sustain pedal, which influences the duration of all notes until the pedal is released [105]. Indeed, it is necessary to utilize a better form of representation that can convey musical intricacies, such as the performance of instruments, harmonic content, and ornaments.

Some efforts have been made to ameliorate this issue. Revamped MIDI-derived events (REMI) [88] is an enhanced representation of music that denotes an explicit metrical grid to model music. Specifically, REMI has been shown effective for pop piano music. Wu and Yang [93] and Chen et al. [94] expand REMI further for other scenarios such as guitar tabulator and Jazz music. Compound Words [98] is another technique that utilizes REMI to generate musical tokens and group them into super tokens. Nevertheless, these methods are primarily tailored and applied to a specific genre like pop music. Therefore, further investigation is required to determine their effectiveness for other scenarios.

C. CREATIVITY

Another issue that comes to the scene with the deep learning music generation is the shortcoming of creative musical ideas. The deep learning models are data-driven, and the learning outcome of the models relies heavily on the given training examples. Even with a good learning outcome, the generations can be marked as inaccurate, inconsistent, or monotonous when studied by human listeners.

We can define creativity as an innovative combination of two or more variations in a meaningful manner. Therefore, a generative model requires first understanding the underlying dynamics of musical compositions and second learning how to compile that knowledge into a new meaningful composition. Models like MusicVAE [53] can generate variations by interpolating motifs and sampling from latent space. However, we can encounter a lack of quality in harmonic content and understanding of rhythmic patterns by analyzing the generated content. In other words, the current models can mainly exploit the learning outcome rather than explore and extrapolate to create new variations.

Models such as [77] and [79] attempted to create new musical styles by compelling the model to diverge from the existing styles. Other models, such as [90] and [40], utilized conditioning techniques as a strategy to address creativity. However, the lack of evaluation methods to measure the creativity aspect of a musical piece makes creativity an arduous and open challenge.

D. STYLE

Currently, there are some deep learning models which can generate music with specific styles, like DeepJ [9] and DeepBach [10]. However, these models are limited to the

style of classical music extracted from the training examples. Indeed, the main challenge lies in the ability of the model to extract the musical features according to the musical style. Other models such as [77] and [79] can perform style transfer from Jazz to classic music genres. However, the generated content lacks musical details, although it sounds plausible. In fact, different musical styles require distinct definitions, making it challenging to obtain an adaptable framework for diverse musical styles. To achieve this, we need a better representation of music. As we have discussed previously, there are challenges tied to music representation, limiting the generative models' ability.

E. INTERACTIVITY

The algorithmic composition systems are desired to achieve the ability to create musical pieces inspired by human compositions rather than pure imitation. However, the black-box nature of neural networks makes it demanding to interact with and control the output of the deep learning models for human users.

It is necessary to differentiate control from interactivity in generative models. To elaborate, control refers to the possibility of defining a set of parameters to achieve an objective and generate a specific context. While models such as Markov Chains allow the definition of constraints during the generation process [6], [7], deep learning models do not possess such possibilities. Therefore, some techniques are introduced to alleviate this issue, such as the unary constraints [146], positional constraints [46], and conditioning [54]. Although these methods provide some degree of control, they are still insufficient to control the model generation in an arbitrary direction.

On the other hand, interactivity refers to the model's ability to be utilized in a fine-grained manner. Music creation is a concurrent iterative process. Artists adapt various strategies to develop a musical idea and create a musical piece. An example of musical strategies in music generation is *incremental variable instantiation* that has been used by [11] and [10]. Comparably, models such as [64] provide interactive and controllable generation through the captured latent space. Indeed, interactivity allows artists to perform local modifications and regenerate specific musical parts incrementally. This functionality is essential for the music generation systems to be practical and assist artists in composing music.

F. EVALUATION

Often, it is the case that a musical piece performs well in the objective evaluation and poorly in the subjective evaluation. On the other hand, the subjective assessment is only conducted on the generated content, not during the training process. Moreover, the current deep learning models lack automatic content evaluation, and there is no direct objective method to evaluate attributes such as creativity. Furthermore, a good subjective evaluation lacks a clear explanation of quantitative metrics. Auditory fatigue must also be considered in the case of subjective evaluation, which can cause bias in the listeners if they listen to similar samples for an extended period. Consequently, it is demanding to define an evaluation metric for performance generations similar to human experts to obtain a meaningful assessment based on musical attributes. Indeed, the challenge of music evaluation portrays a complex task that is hard to automate using computational models. Therefore, the development of a universal evaluation system facilitates maintaining an accurate benchmark of the model's performance subjectively and objectively.

VIII. FUTURE DIRECTION

Table 1 summarises the characteristics of the models overviewed in this work. Music production is an iterative process where a musician or composer as an artist creates and develops musical ideas. Indeed, it is a complex task that involves multiple levels of processing. Although these models can generate novel, innovative and pleasant music, they cannot handle various musical objectives. Therefore, they fail to model the process of music composition.

Mainly, music production is a complex and hierarchical process divided into five main stages: composition, arrangement, sound design, mixing, and mastering. The composition stage includes creating and developing new melodic, harmonic, and rhythmic ideas. The arrangement is a stage of organizing the created musical ideas in the form of a timeline to make a complete piece. The sound design stage consists of sampling, synthesizing, and manipulating sounds. The mixing stage involves instrument arrangement, combining, and balancing the audio layers. Finally, the mastering stage includes the post-production process to balance all the audio elements and ensure the final mix is ready. Note that a musician may step into these stages concurrently by following a particular strategy or approach to create a complete song. Indeed, the creative process in music production involves a complex relationship between each of the music production stages [147].

A cooperative system like Multi-agent systems (MAS) [148] can be a suitable approach for music generation. MAS are distributed artificial intelligence systems consisting of multiple autonomous agents that work together and make independent decisions. The MAS architecture allows the utilization of various computational intelligence methods like deep learning, which is advantageous for modeling music production and musical creativity. The action abilities and perception of MAS agents enable them to cooperate and coordinate with each other to satisfy the objectives of the task [148].

The main challenge of using deep generative models is performing the creative and technological processes while conserving the balance between these two processes. These models involve a series of processing decisions that can significantly influence how artists think about music when they collaborate with these models. Indeed, the shortage of interpretability makes it hard to understand

TABLE 1. Summary of deep learning models for music generation tasks.

	Model	Year	Musical Aspect	Architecture	Hierarchical Structure	Standalone	Modular	Interpretability	Interactivity	Creativity	Control	Structure	Style
1	[34]	2002	Melody	LSTM-RNN		✓						✓	
2	[35]	2008	Melody	LSTM-RNN		✓						✓	
3	FolkRNN [36]	2015	Melody	LSTM-RNN		✓						✓	
4	WordRNN [37]	2016	Melody	LSTM-RNN		✓						✓	
5	PerformanceRNN [45]	2018	Single-track Polyphonic	LSTM-RNN		✓						✓	
6	AnticipationRNN [46]	2017	Single-track Polyphonic	LSTM-RNN		✓			✓		✓	✓	
7	StructureNet [39]	2018	Melody	LSTM-RNN			✓			✓	✓	✓	✓
8	BachProp [44]	2018	Single-track Polyphonic	LSTM-RNN		✓						✓	
9	BandNet [42]	2018	Multi-track Polyphonic	LSTM-RNN		✓				✓		✓	✓
10	[52]	2019	Single-track Polyphonic	Normalizing Flows/LSTM-RNN		✓						✓	
11	[50]	2019	Melody	LSTM-RNN		✓						✓	
12	Two-stageRNN [43]	2019	Melody	LSTM-RNN		✓					✓	✓	
13	[40]	2021	Melody	LSTM-RNN	✓		✓		✓	✓	✓	✓	
14	[38]	2021	Melody	LSTM-RNN		✓						✓	
15	[47]	2021	Single-track Polyphonic	LSTM-RNN		✓					✓	✓	✓
16	MeloForm [51]	2022	Melody	LSTM-RNN	✓		✓	✓	✓	✓	✓	✓	✓
17	[41]	2022	Single-track Polyphonic	LSTM-RNN		✓						✓	✓
18	Groove2Groove [48]	2020	Multi-track Polyphonic	AE		✓				✓	✓		✓
19	MusicVAE [53]	2018	Single-track Polyphonic	VAE	✓	✓			✓	✓	✓	✓	✓
20	Multi-track MuiscVAE [54]	2018	Multi-track Polyphonic	VAE	✓	✓			✓	✓	✓	✓	✓
21	[61]	2018	Melody	VAE/GAN		✓					✓	✓	
22	MIDI-VAE [62]	2018	Multi-track Polyphonic	VAE	✓	✓				✓	✓	✓	✓
23	MidiMe [55]	2019	Multi-track Polyphonic	VAE	✓	✓			✓	✓	✓	✓	✓
24	MIDI-Sandwich2 [58]	2019	Multi-track Polyphonic	VAE/LSTM-RNN	✓	✓				✓	✓	✓	✓
25	[66]	2019	In-painting	VAE	✓	✓			✓	✓	✓	✓	✓
26	[63]	2019	Single-track Polyphonic	VAE	✓	✓					✓	✓	
27	PianoTreeVAE [57]	2020	Single-track Polyphonic	Tree-structure/VAE	✓	✓		✓	✓	✓	✓	✓	
28	MusAE [73]	2020	Multi-track Polyphonic	Adversarial AE	✓	✓			✓	✓	✓	✓	✓
29	TonicNet [89]	2020	Multi-track Polyphonic	Transformer/GRU-RNN		✓						✓	
30	Music SketchNet [60]	2020	Melody	VAE/RNN	✓	✓				✓	✓	✓	
31	[59]	2021	Melody	VAE/Diffusion	✓	✓				✓	✓	✓	
32	Music FaderNets [64]	2021	Multi-track Polyphonic	GM-VAE	✓	✓			✓	✓	✓	✓	✓
33	C-RNN-GAN [144]	2016	Melody	GAN/LSTM-RNN		✓				✓		✓	
34	[69]	2017	Multi-track Polyphonic	GAN/RL		✓				✓		✓	
35	FusionGAN [76]	2017	Style Transfer	GAN		✓				✓	✓	✓	✓
36	[74]	2018	Multi-track Polyphonic	GAN/RNN/CNN		✓				✓	✓	✓	✓
37	CycleGAN [77]	2018	Style Transfer	GAN		✓				✓	✓	✓	✓
38	CycleGAN2 [79]	2019	Style Transfer	GAN		✓				✓	✓	✓	✓
39	MuseGAN [70]	2018	Multi-track Polyphonic	GAN	✓	✓			✓	✓	✓	✓	✓
40	DBM-GAN [72]	2019	Multi-track Polyphonic	GAN	✓	✓			✓	✓	✓	✓	✓
41	[82]	2019	Rhythm	GAN	✓	✓				✓	✓	✓	✓
42	[80]	2020	Rhythm	GAN		✓				✓		✓	✓
43	CONLON [75]	2020	Multi-track Polyphonic	WAE		✓				✓	✓	✓	
44	TransformerNADE [105]	2018	Melody	Transformer		✓				✓		✓	✓
45	MusicTransformer [84]	2018	Melody	Transformer	✓	✓						✓	✓
46	MuseNet [85]	2019	Multi-track Polyphonic	Transformer	✓	✓				✓		✓	✓
47	[99]	2020	Single-track Polyphonic	Transformer		✓						✓	✓
48	[86]	2020	Multi-track Polyphonic	Adversarial Transformer	✓	✓						✓	✓
49	TransformerVAE [87]	2020	Multi-track Polyphonic	Transformer/VAE	✓	✓			✓	✓	✓	✓	✓
50	MMM [96]	2020	Multi-track Polyphonic	Transformer	✓	✓				✓	✓	✓	✓
51	Jazz Transformer [93]	2020	Melody	Transformer	✓	✓						✓	✓
52	Guitar Transformer [94]	2020	Single-track Polyphonic	Transformer	✓	✓						✓	✓
53	Pop Music Transformer [88]	2020	Single-track Polyphonic	Transformer		✓				✓	✓	✓	✓
54	[95]	2020	Single-track Polyphonic	Transformer		✓				✓	✓	✓	✓
55	MusicFrameworks [40]	2021	Melody	Transformer/LSTM-RNN	✓	✓				✓	✓	✓	✓
56	[97]	2021	Multi-track Polyphonic	Transformer		✓				✓	✓	✓	
57	MuseMorphose [90]	2021	Multi-track Polyphonic	Transformer/VAE	✓	✓				✓	✓	✓	✓
58	[98]	2021	Single-track Polyphonic	Transformer		✓				✓		✓	
59	[91]	2021	Multi-track Polyphonic	Transformer	✓	✓				✓	✓	✓	✓
60	SymphonyNet [101]	2022	Multi-track Polyphonic	Transformer		✓				✓	✓	✓	
61	FIGARO [92]	2022	Multi-track Polyphonic	Transformer/VQ-VAE		✓				✓	✓	✓	✓
62	MELONS [100]	2022	Melody	Transformer	✓	✓				✓	✓	✓	✓
63	Theme Transformer [102]	2022	Single-track Polyphonic	Transformer		✓				✓	✓	✓	✓
64	RL-Tuner [112]	2017	Melody	RL		✓			✓	✓	✓	✓	✓
65	Amadeus [113]	2019	Single-track Polyphonic	RL		✓				✓		✓	
66	RL-Duet [114]	2020	Melody	RL		✓			✓	✓	✓	✓	✓
67	RE-RL-Tuner [115]	2021	Melody	RL		✓			✓	✓	✓	✓	✓
68	[145]	2016	Melody	Maximum Entropy		✓						✓	
69	[116]	2016	Melody	Maximum Entropy		✓				✓		✓	✓
70	[118]	2016	Multi-track Polyphonic	Maximum Entropy		✓				✓		✓	✓
71	LakhNES [110]	2019	Transfer Learning	Transformer		✓						✓	
72	[111]	2019	Transfer Learning	VAE/GRU		✓						✓	
73	Accomontage [119]	2021	Multi-track Polyphonic	Hybrid	✓	✓				✓	✓	✓	✓

the decision-making process behind the generated content. The interpretability of the system allows us to locate and correct causes of undesired results. However, the lack of interpretability influences the extensibility of AI systems. Extensibility is important to interact and extend the behaviors and features of a system. Notably, for the algorithmic composition of music, artists often like to create music in a specific style to comply with their desires and musical ideas. Indeed, the extensibility allows human users to be creative and experiment with the system differently. Models such as PianoTreeVAE [57], and MeloForm [51] alleviate the interpretability issue and can provide a better framework. Nevertheless, this is still an open issue for deep learning models.

We can formulate the strategic part of the model exploration, exploitation, and selection processes by emerging

effective model combinations. Through MAS, we can combine the flexibility of smaller models with the benefit of global structure awareness of end-to-end models in a modular manner. Indeed, this approach represents a more dynamic behavior as it divides the main task into sub-tasks and distributes them among the multiple agents. Consequently, MAS can further improve the extensibility and interpretability of the system. Hutchings and McCormack [149] is an example of MAS using deep learning models. It consists of harmonic and melodic agents working cooperatively. The harmonic agent is an RNN-based model, while the melodic agent is a rule-based system. Additionally, Tatar and Pasquier [150] surveys the typology and state-of-the-art agent-based learning in music generation tasks.

Moreover, some of the deep learning models provide some degree of control and interactivity. However, they still lack human participation during content generation. As presented in Table 1, these models are primarily standalone systems. Based on a study conducted by Huang et al. [14], artists mainly achieve their musical goals by leveraging and incorporating a wide range of generative models in a modular way. Indeed, it is challenging to control end-to-end deep learning models to produce high-quality songs in one shot. Artists would like to retain a certain amount of control and freedom to navigate deep learning models to generate samples creatively.

Furthermore, artists may desire to generate musical content strictly coherent with their style. Herein, the system requires to be adaptable and flexible. In Section V we studied the reinforcement learning models for music generation. These models show the ability to learn and adapt to changes by observing the modifications in the environment. Based on the observations, the agent takes action and receives a reward for the action's suitability. The reward function can combine objective and subjective evaluation methods to preserve a balance in performance and creativeness. Therefore, the combination of RL and MAS could provide a more flexible workflow and building blocks through a dynamic learning process. For instance, the agents can cooperate and share their progress using the Blackboard [151] communication approach to fulfill the task (music generation).

Besides, users can work on new ideas efficiently by benefiting from past experiences, utilizing the system to get inspired, and broadening the creative process to various extents. Additionally, the flexibility of the RL framework regarding the models' learnability lets the artists adapt the system to their needs. Therefore, we can enhance the human and AI interaction in the context of music generation.

IX. CONCLUSION

In this paper, we have investigated and studied the generative models in symbolic music generation using deep learning techniques. We have underlined the current state-of-the-art methods and provided an overview of their architectures and strategies to generate musical content. We have outlined the main criteria to model, generate and evaluate musical content. We have discussed the current challenges in music generation and emphasized the essential aspects of these challenges in deep generative models. Notably, we have concentrated on the interactivity and adaptability of these models and proposed a potential research direction to alleviate these challenges and strengthen AI and human interaction.

Almost all of the studies of deep learning models are concentrated on developing the algorithms and specific methods in an end-to-end manner. Indeed, these models are mainly autonomous music-making systems. This type of system is more intended for purposes such as commercial use or entertainment. Notably, artists are more interested in assisted composition systems, where the system is intended, for instance, to provide a glimpse into possible musical variations and inspire the artists to develop new musical ideas. Besides, it is essential to note that music creation is a concurrent process involving many stages of pre-processing and post-processing of musical ideas and materials.

Multi-agent systems have shown great potential in music generation tasks, particularly modeling the music creation process. They can provide a framework in which a combination of multiple approaches can be used to fulfill the desired goal and present a system capable of processing various tasks and inputs. Indeed, its modular and hybrid characteristics can help to alleviate the shortcomings and challenges of the music generation tasks. For example, each instrument consists of specific nuances and characteristics that distinguish their representation of music and musical style. By utilizing MAS architecture, we can simplify the representation of music by concentrating on one instrument at a time, where different agents can be assigned to a specific instrument. This is analogous to how musicians work in a band.

In RL algorithms, the reward function plays an important role, where it assesses the agent's action suitability to the current state of the environment. Therefore, we can formulate the model's evaluation using the RL reward function by combining objective and subjective techniques. The objective evaluation can involve one or multiple agents assessing the sample consistency according to the musical goal using the combination of methods provided in Section VI. On the other hand, the subjective evaluation can be performed by the human listener (agent) who interacts with the musical system. For instance, we can formulate this with a thumbs-up or thumbs-down approach, where the agent receives a reward accordingly. Consequently, the agents within the system incorporate the provided feedback to adapt and adjust their behavior, strategy, or musical goals.

REFERENCES

[1] F. Carnovalini and A. Rodà, "Computational creativity and music generation systems: An introduction to the state of the art," *Frontiers Artif. Intell.*, vol. 3, p. 14, Apr. 2020.

[2] P. Doornbusch, "Gerhard nierhaus: Algorithmic composition: Paradigms of automated music generation," *Comput. Music J.*, vol. 34, no. 3, pp. 70–74, Sep. 2010, doi: 10.1162/COMJ_r_00008.

[3] D. Cope, "Experiments in musical intelligence (EMI): Non-linear linguistic-based composition," *Interface*, vol. 18, nos. 1–2, pp. 117–139, Jan. 1989, doi: 10.1080/09298218908570541.
[4] G. A. Wiggins, G. Papadopoulos, S. Phon-Amnuaisuk, and A. Tuson, "Evolutionary methods for musical composition," *CASYS, Int. J. Comput. Anticipatory Syst.*, 1999. [Online]. Available: http://www.soi.city.ac.uk/~geraint/papers/CASYS98a.pdf
[5] C. Ames, "The Markov process as a compositional model: A survey and tutorial," *Leonardo*, vol. 22, no. 2, pp. 175–187, 1989. [Online]. Available: http://www.jstor.org/stable/1575226
[6] F. Pachet and P. Roy, "Markov constraints: Steerable generation of Markov sequences," *Constraints*, vol. 16, no. 2, pp. 148–172, Apr. 2011.
[7] F. Pachet, A. Papadopoulos, and P. Roy, "Sampling variations of sequences for structured music generation," in *Proc. ISMIR*, 2017, pp. 167–173.
[8] A. Papadopoulos, P. Roy, and F. Pachet, "Avoiding plagiarism in Markov sequence generation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 28, 2014, pp. 1–7.
[9] H. H. Mao, T. Shin, and G. Cottrell, "DeepJ: Style-specific music generation," in *Proc. IEEE 12th Int. Conf. Semantic Comput. (ICSC)*, Jan. 2018, pp. 377–382.
[10] G. Hadjeres, F. Pachet, and F. Nielsen, "Deepbach: A steerable model for bach chorales generation," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1362–1371.
[11] F. Liang, "BachBot: Automatic composition in the style of bach chorales," *Univ. Cambridge*, vol. 8, pp. 19–48, Aug. 2016.
[12] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," 2020, *arXiv:2005.00341*.
[13] F. Ribeiro, D. Florencio, C. Zhang, and M. Seltzer, "CROWDMOS: An approach for crowdsourcing mean opinion score studies," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2011, pp. 2416–2419.
[14] C.-Z. A. Huang, H. V. Koops, E. Newton-Rex, M. Dinculescu, and C. J. Cai, "AI song contest: Human-AI co-creation in songwriting," 2020, *arXiv:2010.05388*.
[15] J.-P. Briot, G. Hadjeres, and F.-D. Pachet," *Deep Learning Techniques for Music Generation*, 1st ed. New York, NY, USA: Springer, 2019.
[16] S. Ji, J. Luo, and X. Yang, "A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions," 2020, *arXiv:2011.06801*.
[17] S. Hochreiter and J. J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 80–1735, 1997, doi: 10.1162/neco.1997.9.8.1735.
[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 139–144.
[19] P. D. Kingma and M. Welling, "Auto-encoding variational Bayes," 2014, *arXiv:1312.6114*.
[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.
[21] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, and M. Funtowicz, "Transformers: State-of-the-art natural language processing," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*, 2020, pp. 38–45.
[22] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM Comput. Surveys*, vol. 54, no. 10s, pp. 1–41, Jan. 2022.
[23] Y. Li, "Deep reinforcement learning: An overview," 2017, *arXiv:1701.07274*.
[24] K. Arulkumaran, M. Peter Deisenroth, M. Brundage, and A. A. Bharath, "A brief survey of deep reinforcement learning," 2017, *arXiv:1708.05866*.
[25] C. Hernandez-Olivan and R. J. Beltran, "Music composition with deep learning: A review," 2021, *arXiv:2108.12290*.
[26] J.-P. Briot and F. Pachet, "Deep learning for music generation: Challenges and directions," *Neural Comput. Appl.*, vol. 32, no. 4, pp. 981–993, Feb. 2020.
[27] D. Herremans, C.-H. Chuan, and E. Chew, "A functional taxonomy of music generation systems," *ACM Comput. Surveys*, vol. 50, no. 5, pp. 1–30, Sep. 2018, doi: 10.1145/3108242.
[28] D. Williams, A. Kirke, E. R. Miranda, E. Roesch, I. Daly, and S. Nasuto, "Investigating affect in algorithmic composition systems," *Psychol. Music*, vol. 43, no. 6, pp. 831–854, Nov. 2015, doi: 10.1177/0305735614543282.
[29] T. Gifford, S. Knotts, J. McCormack, S. Kalonaris, M. Yee-King, and M. D'Inverno, "Computational systems for music improvisation," *Digit. Creativity*, vol. 29, no. 1, pp. 19–36, Jan. 2018, doi: 10.1080/14626268.2018.1426613.
[30] J. D. Fernandez and F. Vico, "AI methods in algorithmic composition: A comprehensive survey," *J. Artif. Intell. Res.*, vol. 48, pp. 513–582, Nov. 2013.
[31] G. Nierhaus, *Algorithmic Composition: Paradigms of Automated Music Generation*. Berlin, Germany: Springer, 2009, doi: 10.1007/978-3-211-75540-2.
[32] O. Lopez-Rincon, O. Starostenko, and G. A.-S. Martin, "Algoritmic music composition based on artificial intelligence: A survey," in *Proc. Int. Conf. Electron., Commun. Comput. (CONIELECOMP)*, Feb. 2018, pp. 187–193, doi: 10.1109/CONIELECOMP.2018.8327197
[33] M. Bretan and G. Weinberg, "A survey of robotic musicianship," *Commun. ACM*, vol. 59, no. 5, pp. 100–109, Apr. 2016, doi: 10.1145/2818994.
[34] D. Eck and J. Schmidhuber, "Finding temporal structure in music: Blues improvisation with LSTM recurrent networks," in *Proc. 12th IEEE Workshop Neural Netw. Signal Process.*, Sep. 2002, pp. 747–756, doi: 10.1109/NNSP.2002.1030094.
[35] D. Eck and J. Lapalme, "Learning musical structure directly from sequences of music," Dept. Comput. Sci., CP, Univ. Montreal, Montreal, QC, Canada, 2008.
[36] B. Sturm, J. F. Santos, and I. Korshunova, "Folk music style modelling by recurrent neural networks with long short term memory units," in *Proc. 16th Int. Soc. Music Inf. Retr. Conf.*, 2015.
[37] K. Choi, G. Fazekas, and M. Sandler, "Text-based LSTM networks for automatic music composition," 2016, *arXiv:1604.05358*.
[38] J. Li, H. Liu, N. Yan, and L. Wang, "Enhanced memory network: The novel network structure for symbolic music generation," 2021, *arXiv:2110.03392*.
[39] G. Medeot, S. Cherla, K. Kosta, M. McVicar, S. Abdallah, M. Selvi, E. Newton-Rex, and K. Webster, "Structurenet: Inducing structure in generated melodies," in *Proc. ISMIR*, 2018, pp. 725–731.
[40] S. Dai, Z. Jin, C. Gomes, and R. B. Dannenberg, "Controllable deep melody generation via hierarchical music structure representation," 2021, *arXiv:2109.00663*.
[41] G. Keerti, A. Vaishnavi, P. Mukherjee, A. S. Vidya, G. S. Sreenithya, and D. Nayab, "Attentional networks for music generation," *Multimedia Tools Appl.*, vol. 81, no. 4, pp. 5179–5189, 2022.
[42] Y. Zhou, W. Chu, S. Young, and X. Chen, "BandNet: A neural network-based, multi-instrument beatles-style MIDI music composition machine," 2018. [Online]. Available: https://arxiv.org/abs/1812.07126, doi: 10.48550/ARXIV.1812.07126.
[43] C. D. Boom, S. V. Laere, T. Verbelen, and B. Dhoedt, "Rhythm, chord and melody generation for lead sheets using recurrent neural networks," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Springer, 2019, pp. 454–461.
[44] F. Colombo, J. Brea, and W. Gerstner, "Learning to generate music with BachProp," 2018, *arXiv:1812.06669*.
[45] S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan, "This time with feeling: Learning expressive musical performance," *Neural Comput. Appl.*, vol. 32, no. 4, pp. 955–967, Feb. 2020.
[46] G. Hadjeres and F. Nielsen, "Interactive music generation with positional constraints using anticipation-RNNs," 2017, *arXiv:1709.06404*.
[47] L. N. Ferreira and J. Whitehead, "Learning to generate music with sentiment," 2021, *arXiv:2103.06125*.
[48] O. Cifka, U. Simsekli, and G. Richard, "Groove2Groove: One-shot music style transfer with supervision from synthetic data," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 2638–2650, 2020.
[49] Y. Broze and D. Shanahan, "Diachronic changes in jazz harmony: A cognitive perspective," *Music Perception, Interdiscipl. J.*, vol. 31, no. 1, pp. 32–45, 2013.
[50] K. Chen, W. Zhang, S. Dubnov, G. Xia, and W. Li, "The effect of explicit structure encoding of deep neural networks for symbolic music generation," in *Proc. Int. Workshop Multilayer Music Represent. Process. (MMRP)*, Jan. 2019, pp. 77–84.
[51] P. Lu, X. Tan, B. Yu, T. Qin, S. Zhao, and T.-Y. Liu, "MeloForm: Generating melody with musical form based on expert systems and neural networks," 2022, *arXiv:2208.14345*.
[52] Z. Ziegler and A. Rush, "Latent normalizing flows for discrete sequences," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7673–7682.
[53] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4364–4373.

[54] I. Simon, A. Roberts, C. Raffel, J. Engel, C. Hawthorne, and D. Eck, "Learning a latent space of multitrack measures," 2018. [Online]. Available: https://arxiv.org/abs/1806.00195, doi: 10.48550/ARXIV.1806.00195.

[55] M. Dinculescu, J. Engel, and A. Roberts, "MidiMe: Personalizing a musicvae model with user data," Tech. Rep., 2019.

[56] J. Engel, M. Hoffman, and A. Roberts, "Latent constraints: Learning to generate conditionally from unconditional generative models," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–22. [Online]. Available: https://openreview.net/forum?id=Sy8XvGb0-

[57] Z. Wang, Y. Zhang, Y. Zhang, J. Jiang, R. Yang, J. Zhao, and G. Xia, "PIANOTREE VAE: Structured representation learning for polyphonic music," 2020, *arXiv:2008.07118*.

[58] X. Liang, J. Wu, and J. Cao, "MIDI-sandwich2: RNN-based hierarchical multi-modal fusion generation VAE networks for multi-track symbolic music generation," 2019, *arXiv:1909.03522*.

[59] G. Mittal, J. Engel, C. Hawthorne, and I. Simon, "Symbolic music generation with diffusion models," 2021, *arXiv:2103.16091*.

[60] K. Chen, C.-I. Wang, T. Berg-Kirkpatrick, and S. Dubnov, "Music SketchNet: Controllable music generation via factorized representations of pitch and rhythm," 2020, *arXiv:2008.01291*.

[61] M. Akbari and J. Liang, "Semi-recurrent CNN-based VAE-GAN for sequential data generation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 2321–2325.

[62] G. Brunner, A. Konrad, Y. Wang, and R. Wattenhofer, "MIDI-VAE: Modeling dynamics and instrumentation of music with applications to style transfer," 2018, *arXiv:1809.07600*.

[63] Y.-A. Wang, Y.-K. Huang, T.-C. Lin, S.-Y. Su, and Y.-N. Chen, "Modeling melodic feature dependency with modularized variational auto-encoder," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 191–195.

[64] H. H. Tan and D. Herremans, "Music FaderNets: Controllable music generation based on high-level features via low-level feature modelling," 2020, *arXiv:2007.15474*.

[65] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. Denoyer, and M. Ranzato, "Fader networks: Manipulating images by sliding attributes," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–10.

[66] A. Pati, A. Lerch, and G. Hadjeres, "Learning to traverse latent spaces for musical score inpainting," 2019, *arXiv:1907.01164*.

[67] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. 27th Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*. Boston, FL, USA: Addison-Wesley, 2000, pp. 417–424, doi: 10.1145/344779.344972.

[68] O. Mogren, "C-RNN-GAN: A continuous recurrent neural network with adversarial training," in *Proc. Constructive Mach. Learn. Workshop (CML) NIPS*, 2016, pp. 1–6.

[69] G. Lima Guimaraes, B. Sanchez-Lengeling, C. Outeiral, P. Luis Cunha Farias, and A. Aspuru-Guzik, "Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models," 2017, *arXiv:1705.10843*.

[70] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, 2018, pp. 1–8.

[71] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[72] F. Guan, C. Yu, and S. Yang, "A GAN model with self-attention mechanism to generate multi-instruments symbolic music," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–6.

[73] A. Valenti, A. Carta, and D. Bacciu, "Learning style-aware symbolic music representations by adversarial autoencoders," 2020, *arXiv:2001.05494*.

[74] H.-M. Liu and Y.-H. Yang, "Lead sheet generation and arrangement by conditional generative adversarial network," in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2018, pp. 722–727.

[75] L. Angioloni, V. Borghuis, L. Brusci, and P. Frasconi, "CONLON: A pseudo-song generator based on a new pianoroll, Wasserstein autoencoders, and optimal interpolations," in *Proc. ISMIR*, 2020, pp. 876–883.

[76] Z. Chen, C.-W. Wu, Y.-C. Lu, A. Lerch, and C.-T. Lu, "Learning to fuse music genres with generative adversarial dual learning," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2017, pp. 817–822.

[77] G. Brunner, Y. Wang, R. Wattenhofer, and S. Zhao, "Symbolic music genre transfer with CycleGAN," in *Proc. IEEE 30th Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2018, pp. 786–793.

[78] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," 2017. [Online]. Available: https://arxiv.org/abs/1703.10593, doi: 10.48550/ARXIV.1703.10593.

[79] G. Brunner, M. Moayeri, O. Richter, R. Wattenhofer, and C. Zhang, "Neural symbolic music genre transfer insights," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Springer, 2019, pp. 437–445.

[80] N. Tokui, "Can GAN originate new electronic dance music genres?—Generating novel rhythm patterns using GAN with genre ambiguity loss," 2020, *arXiv:2011.13062*.

[81] A. Elgammal, B. Liu, M. Elhoseiny, and M. Mazzone, "CAN: Creative adversarial networks, generating 'art' by learning about styles and deviating from style norms," 2017, *arXiv:1706.07068*.

[82] S. Lattner and M. Grachten, "High-level control of drum track generation using learned patterns of rhythmic interaction," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*, Oct. 2019, pp. 35–39.

[83] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," 2018, *arXiv:1803.02155*.

[84] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer," 2018, *arXiv:1809.04281*.

[85] C. Payne. (2019). *Musenet*.[Online]. Available: https://openai.com/blog/musenet

[86] N. Zhang, "Learning adversarial transformer for symbolic music generation," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 2, 2020, doi: 10.1109/TNNLS.2020.2990746.

[87] J. Jiang, G. G. Xia, D. B. Carlton, C. N. Anderson, and R. H. Miyakawa, "Transformer VAE: A hierarchical model for structure-aware and interpretable music representation learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 516–520, doi: 10.1109/ICASSP40776.2020.9054554.

[88] Y.-S. Huang and Y.-H. Yang, "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 1180–1188.

[89] O. Peracha, "Improving polyphonic music models with feature-rich encoding," in *Proc. ISMIR*, 2020, pp. 1–7.

[90] S.-L. Wu and Y.-H. Yang, "MuseMorphose: Full-song and fine-grained music style transfer with one transformer VAE," 2021, *arXiv:2105.04090*.

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

[92] D. von Rütte, L. Biggio, Y. Kilcher, and T. Hofmann, "FIGARO: Generating symbolic music with fine-grained artistic control," 2022, *arXiv:2201.10936*.

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

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

[95] K. Choi, C. Hawthorne, I. Simon, M. Dinculescu, and J. Engel, "Encoding musical style with transformer autoencoders," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1899–1908.

[96] J. Ens and P. Pasquier, "MMM: Exploring conditional multi-track music generation with the transformer," 2020, *arXiv:2008.06048*.

[97] S. Di, Z. Jiang, S. Liu, Z. Wang, L. Zhu, Z. He, H. Liu, and S. Yan, "Video background music generation with controllable music transformer," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 2037–2045.

[98] W.-Y. Hsiao, J.-Y. Liu, Y.-C. Yeh, and Y.-H. Yang, "Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 178–186.

[99] E. P. Nichols, S. Kalonaris, G. Micchi, and A. Aljanaki, "Modeling baroque two-part counterpoint with neural machine translation," 2020, *arXiv:2006.14221*.

[100] Y. Zou, P. Zou, Y. Zhao, K. Zhang, R. Zhang, and X. Wang, "Melons: Generating melody with long-term structure using transformers and structure graph," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 191–195.

[101] J. Liu, Y. Dong, Z. Cheng, X. Zhang, X. Li, F. Yu, and M. Sun, "Symphony generation with permutation invariant language model," 2022, *arXiv:2205.05448*.
[102] Y.-J. Shih, S.-L. Wu, F. Zalkow, M. Müller, and Y.-H. Yang, "Theme transformer: Symbolic music generation with theme-conditioned transformer," *IEEE Trans. Multimedia*, early access, Mar. 23, 2022, doi: 10.1109/TMM.2022.3161851.
[103] D. Zhang, F. Nan, X. Wei, S. Li, H. Zhu, K. McKeown, R. Nallapati, A. Arnold, and B. Xiang, "Supporting clustering with contrastive learning," 2021, *arXiv:2103.12953*.
[104] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, vol. 96, 1996, pp. 226–231.
[105] C. Hawthorne, A. Huang, D. Ippolito, and D. Eck, "Transformer-NADE for piano performances," in *Proc. NIPS 2nd Workshop Mach. Learn. Creativity Design*, 2018, pp. 1–3.
[106] H. Larochelle and I. Murray, "The neural autoregressive distribution estimator," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 29–37.
[107] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription," 2012, *arXiv:1206.6392*.
[108] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[109] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, pp. 1–24, 2019.
[110] C. Donahue, H. Henry Mao, Y. Ethan Li, G. W. Cottrell, and J. McAuley, "LakhNES: Improving multi-instrumental music generation with cross-domain pre-training," 2019, *arXiv:1907.04868*.
[111] H.-T. Hung, C.-Y. Wang, Y.-H. Yang, and H.-M. Wang, "Improving automatic jazz melody generation by transfer learning techniques," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Nov. 2019, pp. 339–346.
[112] N. Jaques, S. Gu, R. E. Turner, and D. Eck, "Tuning recurrent neural networks with reinforcement learning," 2017.
[113] H. Kumar and B. Ravindran, "Polyphonic music composition with LSTM neural networks and reinforcement learning," 2019, *arXiv:1902.01973*.
[114] N. Jiang, S. Jin, Z. Duan, and C. Zhang, "RL-duet: Online music accompaniment generation using deep reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 710–718.
[115] H. Liu, X. Xie, R. Ruzi, L. Wang, and N. Yan, "RE-RLTuner: A topic-based music generation method," in *Proc. IEEE Int. Conf. Real-time Comput. Robot. (RCAR)*, Jul. 2021, pp. 1139–1142.
[116] S. Moulieras and F. Pachet, "Maximum entropy models for generation of expressive music," 2016, *arXiv:1610.03606*.
[117] T. Mora, A. M. Walczak, W. Bialek, and C. G. Callan, Jr., "Maximum entropy models for antibody diversity," *Proc. Nat. Acad. Sci. USA*, vol. 107, no. 12, pp. 5405–5410, 2010.
[118] G. Hadjeres, J. Sakellariou, and F. Pachet, "Style imitation and chord invention in polyphonic music with exponential families," 2016, *arXiv:1609.05152*.
[119] J. Zhao and G. Xia, "AccoMontage: Accompaniment arrangement via phrase selection and style transfer," 2021, *arXiv:2108.11213*.
[120] A. Jordanous, "A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative," *Cogn. Comput.*, vol. 4, no. 3, pp. 246–279, 2012.
[121] L.-C. Yang and A. Lerch, "On the evaluation of generative models in music," *Neural Comput. Appl.*, vol. 32, no. 9, pp. 4773–4784, 2020.
[122] E. P. Asmus, "Music assessment concepts," *Music Educators J.*, vol. 86, no. 2, pp. 19–24, 1999.
[123] A. M. Turing, "Computing machinery and intelligence," in *Parsing the Turing Test*. Springer, 2009, pp. 23–65.
[124] C. Ariza, "The interrogator as critic: The Turing test and the evaluation of generative music systems," *Comput. Music J.*, vol. 33, no. 2, pp. 48–70, Jun. 2009.
[125] A. Défossez, N. Zeghidour, N. Usunier, L. Bottou, and F. Bach, "SING: Symbol-to-instrument neural generator," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.
[126] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, "Neural audio synthesis of musical notes with WaveNet autoencoders," in *Proc. 34th Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, vol. 70, D. Precup and Y. W. Teh, Eds. PMLR, Aug. 2017, pp. 1068–1077. [Online]. Available: https://proceedings.mlr.press/v70/engel17a.html
[127] S. F. Chen, D. Beeferman, and R. Rosenfeld, "Evaluation metrics for language models," Tech. Rep., 1998.
[128] D. Jeong, T. Kwon, Y. Kim, and J. Nam, "Graph neural network for music score data and modeling expressive piano performance," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3060–3070.
[129] J. Gillick, A. Roberts, and J. Engel. (2019). *Groovae: Generating and Controlling Expressive Drum Performances*. Magenta Blog. [Online]. Available: https://magenta.tensorflow.org/groovae
[130] B. L. Sturm and O. Ben-Tal, "Taking the models back to music practice: Evaluating generative transcription models built using deep learning," *J. Creative Music Syst.*, vol. 2, no. 1, pp. 32–60, Sep. 2017.
[131] C.-Z. A. Huang, T. Cooijmans, A. Roberts, A. Courville, and D. Eck, "Counterpoint by convolution," 2019. [Online]. Available: https://arxiv.org/abs/1903.07227, doi: 10.48550/ARXIV.1903.07227.
[132] D. D. Johnson, "Generating polyphonic music using tied parallel networks," in *Proc. Int. Conf. Evol. Biologically Inspired Music Art*. Springer, 2017, pp. 128–143.
[133] L. Theis, A. van den Oord, and M. Bethge, "A note on the evaluation of generative models," 2015, *arXiv:1511.01844*.
[134] H. Lim, S. Rhyu, and K. Lee, "Chord generation from symbolic melody using BLSTM networks," 2017, *arXiv:1712.01011*.
[135] R. Sabathé, E. Coutinho, and B. Schuller, "Deep recurrent music writer: Memory-enhanced variational autoencoder-based musical score composition and an objective measure," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*. IEEE, 2017, pp. 3467–3474, doi: 10.1109/IJCNN.2017.7966292.
[136] P. C. Mahalanobis, "On the generalized distance in statistics," Nat. Inst. Sci. India, Tech. Rep., 1936.
[137] Y. Ren, J. He, X. Tan, T. Qin, Z. Zhao, and T.-Y. Liu, "PopMAG: Pop music accompaniment generation," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 1198–1206.
[138] B. Genchel, A. Pati, and A. Lerch, "Explicitly conditioned melody generation: A case study with interdependent RNNs," 2019, *arXiv:1907.05208*.
[139] C.-I. Wang and S. Dubnov, "Guided music synthesis with variable Markov Oracle," in *Proc. 10th Artif. Intell. Interact. Digit. Entertainment Conf.*, 2014, pp. 1–8.
[140] C. Jin, Y. Tie, Y. Bai, X. Lv, and S. Liu, "A style-specific music composition neural network," *Neural Process. Lett.*, vol. 52, no. 3, pp. 1893–1912, Dec. 2020.
[141] G. Widmer, M. Grachten, and S. Lattner, "Imposing higher-level structure in polyphonic music generation using convolutional restricted Boltzmann machines and constraints," *J. Creative Music Syst.*, vol. 2, no. 2, pp. 1–31, Mar. 2018.
[142] D. Huron, "Music information processing using the humdrum toolkit: Concepts, examples, and lessons," *Comput. Music J.*, vol. 26, no. 2, pp. 11–26, Jun. 2002.
[143] M. Müller and N. Jiang, "A scape plot representation for visualizing repetitive structures of music recordings," in *Proc. ISMIR*, 2012, pp. 97–102.
[144] O. Mogren, "C-RNN-GAN: Continuous recurrent neural networks with adversarial training," 2016, *arXiv:1611.09904*.
[145] J. Sakellariou, F. Tria, V. Loreto, and F. Pachet, "Maximum entropy model for melodic patterns," in *Proc. ICML Workshop Constructive Mach. Learn.*, 2015, pp. 1–4.
[146] G. Hadjeres and F. Nielsen, "Anticipation-RNN: Enforcing unary constraints in sequence generation, with application to interactive music generation," *Neural Comput. Appl.*, vol. 32, no. 4, pp. 995–1005, 2020.
[147] C. Roads, *Composing Electronic Music: A New Aesthetic*. New York, NY, USA: Oxford Univ. Press, 2015.
[148] M. J. Wooldridge, *An Introduction to Multiagent Systems*, 2nd ed. Chichester, U.K.: Wiley, 2009.
[149] P. Hutchings and J. McCormack, "Using autonomous agents to improvise music compositions in real-time," in *Proc. Int. Conf. Evol. Biologically Inspired Music Art*. Springer, 2017, pp. 114–127.
[150] K. Tatar and P. Pasquier, "Musical agents: A typology and state of the art towards musical metacreation," *J. New Music Res.*, vol. 48, no. 1, pp. 56–105, Jan. 2019, doi: 10.1080/09298215.2018.1511736.
[151] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, vol. 6, pp. 28573–28593, 2018.

SHAYAN DADMAN received the B.S. degree in software engineering from Azad University North Tehran Branch, Tehran, Iran, in 2017, and the M.S. degree in computer science and geometric modeling from the UiT The Arctic University of Norway, Narvik, in 2020, where he is currently pursuing the Ph.D. degree in artificial intelligence, and the application of reinforcement learning and multi-agent systems.

From 2020 to 2021, he was a Research Assistant with the Department of Computer Science and Computational Engineering, UiT The Arctic University of Norway. His research interests include computational creativity, the algorithmic composition of music, music information retrieval, and human–computer interaction.

BERNT ARILD BREMDAL received the Ph.D. degree in applied AI and the M.S. degree in mechanical engineering from the NTNU, Norway. He has a 25 year background from academia and business, as a researcher, a entrepreneur, and the director. He worked with AI and machine learning in media, energy and manufacturing. He was the co-responsible for the creation of the first GeoX system for hydrocarbon resource estimation, an AI supported decision support system that has become an industry standard in the oil and gas business. He was also the co-inventor of the ASAP system, which has been maintained and in daily use in the offshore construction business, since 1989. He also invented the ground breaking CORPORUM Summarizer that became a commercial success, in 2000. In the past decade, he has been pre-occupied with creative AI and smart energy related topics.

BØRRE BANG received the Ph.D. degree.

He has a background in civil engineering and computer science. He was introduced to AI, knowledge-based systems, and constraint logic programming in the domain of reals as a Ph.D. candidate, under Prof. S. Zeuthen, in 1990. Since 2016, he has been a Full Professor with the Department of Computer Science and Computational Engineering, Faculty of Engineering and Technology, UiT The Arctic University of Norway, Narvik, Norway, where he is currently working as the Head of the Department. His research interests include simulation, geometric modeling, and computational methods.

RUNE DALMO received the B.S. and M.S. degrees in computer science from Narvik University College, Norway, in 2001 and 2003, respectively, and the Ph.D. degree in computer science from the University of Oslo, in 2016.

From 2003 to 2011, he worked as a Software Developer, Narvik and Oslo, Norway. From 2011 to 2015, he was an Assistant Professor with Narvik University College. Since 2016, he has been an Associate Professor with the Department of Computer Science and Computational Engineering, Faculty of Engineering Science and Technology, UiT The Arctic University of Norway. He is currently leading the research and development group simulations. He is the author of a number of research papers. His research interests include geometric modeling, data fitting and visualization, splines, interpolation and approximation, and computational geometry.