Recent Advances of Computational Intelligence Techniques for Composing Music

Yu-Wei Wen, *Member, IEEE,* and Chuan-Kang Ting, *Senior Member, IEEE*

***Abstract*—Music exerts a ubiquitous influence on human cultures and daily lives. Composing music is deemed rather complicated because it involves various factors (e.g., instruments, melodies, percussions, and chords) needed to be well coordinated for creating harmony, tension, and emotions. Computational intelligence (CI) has shown its effectiveness in solving complex problems, such as optimization, data modeling, and reasoning. In light of the advantages of CI, a considerable amount of research has been proposed to incorporate CI techniques into music composition applications. The literature shows that evolutionary computation and neural network are very popular in this research area. The present survey reviews the recent studies on music composition using CI techniques, to reflect the methodological advances in the past decade. Particularly, this survey stresses two trends: 1) an increasing interest in deep learning for music composition and 2) the deepened engagement of synergizing domain knowledge, music data, and human interaction. In addition, we provide a taxonomy to classify these studies and discuss the research challenges and future directions.**

***Index Terms*—Computational intelligence, music composition, neural network, deep learning, evolutionary computation, genetic algorithm.**

I. Introduction

Automatic music composition has been investigated since the 1980s. The studies demonstrated that music composition can be automated through computational intelligence (CI) techniques. The three pillars of CI, i.e., evolutionary computation (EC), neural network (NN), and fuzzy system (FS), have been adopted in various musical tasks. The CI-based music composition approaches are shown to be competent at diverse music composition tasks and continue to thrive nowadays. More specifically, the existing CI studies pertain to different types of composition tasks, such as monophonic melodies, polyphonic melodies, chords, rhythm, accompaniment, and multitrack music. Monophonic melody composition plays a major role in music composition studies, which usually concerns a genre-specific or general-purpose composition task. Polyphonic melodies composition is commonly associated with four-part harmonization and counterpoint, featuring the exquisite coordination of multiple parallel melodies. Most of the melody composition tasks rely on predefined chords. By contrast, chords composition deals with the selection, progression, and transition of chords that are suitable for a given melody or sometimes a desired vibe [1]–[6]. Rhythm generation involves composing for the percussive instruments [7]–[11] and assigning durations to the notes of a melody [12]–[14]. Accompaniment composition focuses on the addition of backing parts, which may consist of harmonization and percussion, for a given melody [15], [16]. Finally, multitrack music composition attempts to compose melody, chord, and percussion, where these music components can be composed track-by-track or all at once.

Regarding the methodologies for music composition, EC and NN are usually used as the protagonist, whereas FS is seldom used. Composing music is commonly formulated as an optimization problem in EC studies; by contrast, it is mainly treated as a data modeling problem in NN studies. From the perspective of evolutionism, the task of music composition can be formulated as

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}\in\boldsymbol{X}} f(\boldsymbol{x}), \tag{1}$$

where $\boldsymbol{x}$ denotes a composition and f is the music quality evaluator, also known as the fitness function in evolutionary algorithms (EAs) [17]. In the formulation, the aim of composing music is to search the solution space $\boldsymbol{X}$ for an arrangement of musical events $\boldsymbol{x}^*$ (e.g., music notes) which maximizes the response of f. Here the evaluator f is a prerequisite for composing music and needs to be carefully designed in order to satisfy human's sensation or resemble esthetics. Designing such a fitness function f to quantify and approximate human esthetics poses a formidable challenge to evolutionary composition approaches. The strategies for fitness evaluation can be categorized into three types: 1) interactive evaluation, 2) rule-based evaluation, and 3) learning-based evaluation [18]. First, interaction-based fitness evaluation requires a human evaluator to listen and judge the quality of candidate music in the evolution cycles. Interaction is a direct way to acquire human feedback; however, this type of systems commonly suffer from human fatigue caused by repeated evaluation and listening to poor-quality candidates [18], [19]. Second, rule-based evaluation usually considers music knowledge to automate evaluation of compositions. A recent trend in this category is the increasing sophistication of evaluation rules [20]–[23]. The meticulously crafted rules have enabled EAs to compose music of different genres. Third, learning-based evaluation uses an evaluation model to measure the fitness value of a candidate. This type of evaluation approaches embraces various musical features [24]–[27].

Yu-Wei Wen is with the Department of Artificial Intelligence, Tamkang University; the Department of Power Mechanical Engineering, National Tsing Hua University; and the Department of Computer Science and Information Engineering, National Chung Cheng University, Taiwan.

Chuan-Kang Ting is with the Department of Power Mechanical Engineering, National Tsing Hua University, Taiwan (e-mail: ckting@pme.nthu.edu.tw).

Corresponding author: Chuan-Kang Ting.

From the perspective of connectionism, a music composition $\boldsymbol{x}$ can be generated by

$$\boldsymbol{x} = f^*(\boldsymbol{z}) \quad \text{with } f^* = \arg\min_{f \in F} L(f), \tag{2}$$

where L is the loss function and f represents a neural network model that transforms an arbitrary vector $\boldsymbol{z}$ to a composition $\boldsymbol{x}$. Based on this formulation, a prerequisite for composing music is to obtain the model f^*, aiming to minimize the loss, e.g., the discrepancy of probability distribution between model and training data. After model training, f^* is then used to transform a randomly (or heuristically) sampled vector $\boldsymbol{z}$ into a music composition. A major challenge in these NN-based approaches is to design and train the model f^*. The existing studies commonly predetermine the neural architecture and confine the search space F as a parameter search to be solved by the training algorithm. The early NN-based composition systems adopted vanilla and shallow models, such as long short-term memory (LSTM) network with a single hidden layer [28]. Recently, the research on deep learning (DL) exhibited the potential of complex NNs for modeling music data. The advances of NN for composing music are highly related to the innovations in model architecture and data representation. The advanced model architectures, such as generative adversarial network (GAN) [29]–[32], are used to learn the mapping from a point in the input space to a piece of music. Moreover, variational autoencoder (VAE) is adopted to learn the transition among music pieces [10], [33]–[35]. Bidirectional recurrent network [36]–[38] and transformer [39]–[41] can enhance the capability of NNs in processing long music sequences. Novel data representations are further proposed to encode musical events more efficiently. In contrast to conventional MIDI-like representations, the newly proposed data representations benefit from music domain knowledge [40], [41] and hierarchical structure [35]. These representations facilitate capturing the patterns of interest for NNs.

This survey aims to provide a collective view of up-to-date EC and NN techniques and advances in automatic music composition. The FS techniques are excluded because they are less used as the major role but an auxiliary tool for music composition, such as expressiveness control [42], sub-function of fitness evaluation in evolutionary composition [43], and hyper-heuristics in memetic composition [44], [45]. The survey is focused on the systems for symbolic music composition proposed in the past decade. There have been some surveys on computer music, algorithmic composition [46]–[48], artificial intelligence-based composition [49]–[52], and CI-based composition [18]. In addition, some articles introduce one or few techniques for composing music, such as DL [53]–[56] and EC [57]–[61], or compare these techniques [22], [62]–[64]. Liu and Ting [18] presented a comprehensive review on CI music composition techniques from a musical perspective, mainly on the role of the music content that a system produces. Nonetheless, the research profile of CI-based music composition has been considerably changed since then. First, due to the successes of DL, the interest in using it for music composition is significantly increased (see Fig. 1). Second, synergy of music knowledge, data, and human interaction turns to be a tendency. As Fig. 2 indicates, modern composition systems usually involve more than one factor. To reflect the recent advances, the present paper conducts a review and investigation into various types of EC and NN/DL, research challenges and progress in music composition. In addition, this survey presents a broader-level taxonomy to categorize the NN-based music composition systems based on their sampling strategies: 1) sampling from input space, 2) sampling from latent space, and 3) sampling from output space.

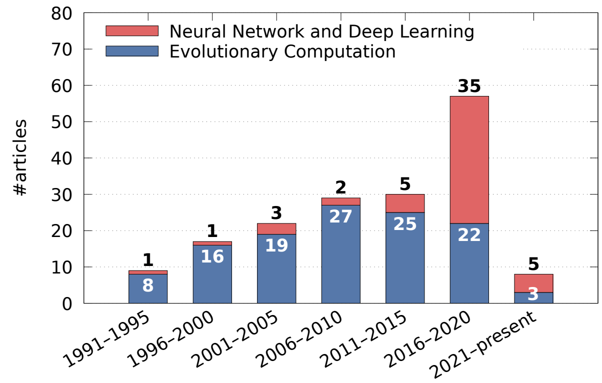


Fig. 1. Number of papers on EC and NN for music composition reviewed in this survey.

The remainder of this paper is organized as follows. Section II introduces the existing data representations for music composition. Section III recapitulates the recent composition methods based on EC. Section IV reviews the NN and DL related music composition methods. Section V elaborates on our analysis and discussion for future research directions. Finally, concluding remarks are given in Section VI.

II. Data Representation

Music content can be saved in audio or symbolic form. The former records the continuous variations of audio states, while the latter encodes music as a sequence of events. Audio representations are often used to handle the tasks related to sound texture, e.g., audio synthesis [65] and timbre modulation [66]. On the other hand, symbolic representations, especially MIDI-based encodings and pianoroll, are adopted largely in automatic music generation studies [18], [50], [55], [64].

Musical instrument digital interface (MIDI) provides a standard of storing and processing musical data in a symbolic manner. Various MIDI-like representations are used to encode musical events in a sequential order. As the musical events are encoded linearly to temporal progression, the MIDI-like representations are also known as linear representations in EC and event-based representations in NN. A common strategy in EC studies is to represent a monophonic melody as an integer string, in which a note's pitch and duration form a melodic unit. Figure 3 exemplifies a music excerpt and its corresponding linear representation, where note pitches can be encoded with absolute values, relative values [67], or harmonic roles [68]. On the other hand, the event-based representation

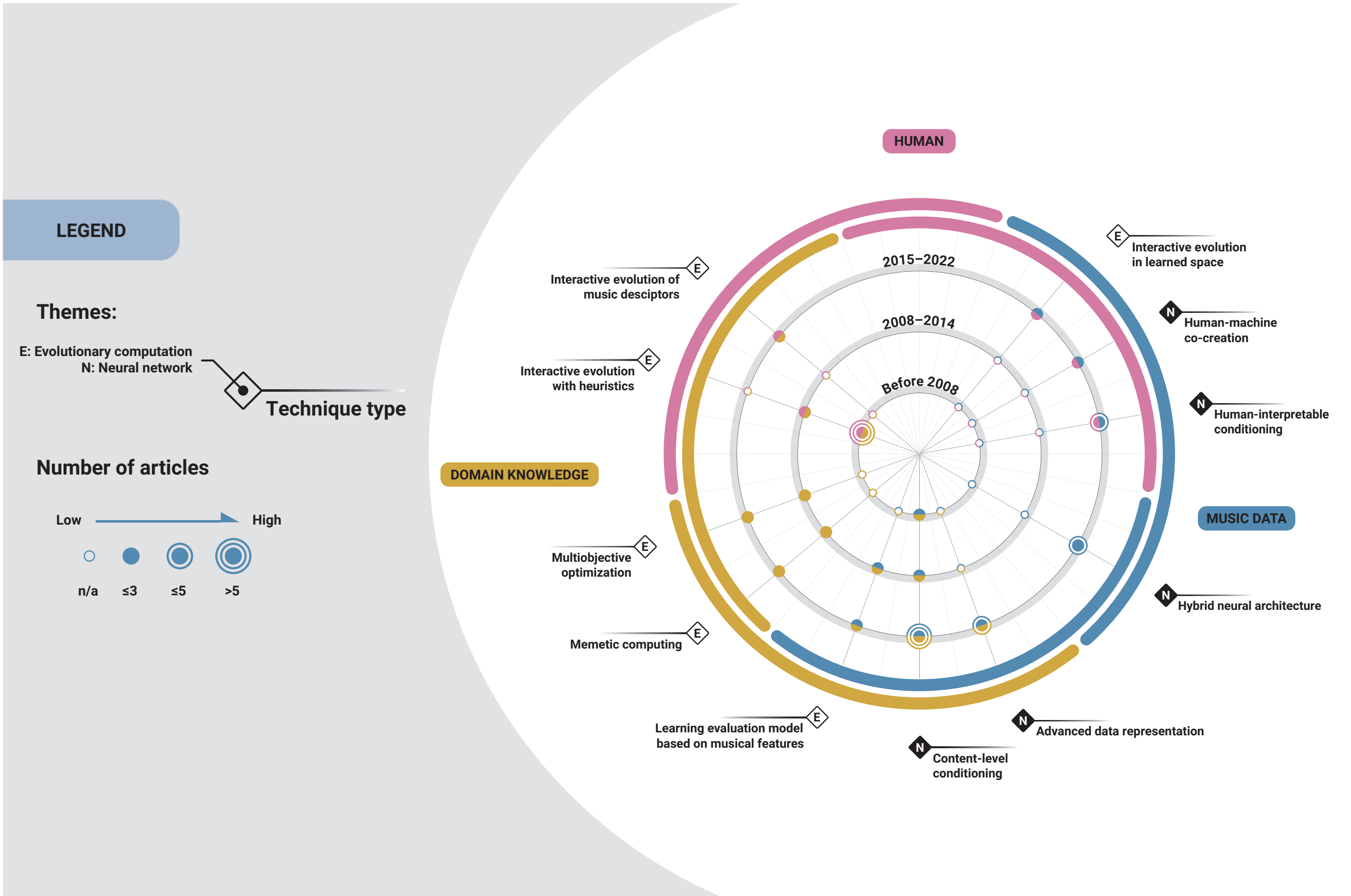


Fig. 2. A map of EC and NN techniques engaged with human, knowledge, and data in music composition.

resembles the concept of linguistic data encoding by pre-defining an event set as the vocabulary. The size of an event set varies with the pitch range, velocity range, and temporal resolution. Using a feasibly defined event set, a music excerpt can be represented as a series of musical events. Figure 4b illustrates a piece of sheet music and its corresponding event-based form. There have been some event-based representations proposed to enhance long sequence generation; for example, compound word representation wraps multiple elementary events into a single token [41].

Pianoroll is another typical representation for music data. A pianoroll is usually represented as a two-dimensional matrix in which one dimension is aligned with the pitch range and the other dimension is associated with temporal progression. A binary matrix is capable of encoding the presence of onset events, while an integer or real matrix can further indicate the velocity of onset events. Figure 4c gives an example of pianoroll representation. A pianoroll can be extracted and disassembled into multiple matrices to encode music information with respect to channels [69], analogous to the RGB channels in images.

Tree-based representations are developed to further specify the hierarchical structure of music. These representations encode musical information by recursively dividing it into low-level structures, e.g., phrases, bars, and notes, until the temporal unit is reached. Figure 5 illustrates a tree-based representation for a short excerpt. This type of representation is a generic strategy to represent a piece of candidate music in genetic programming (GP) [13], [70]–[72]. Wang et al. [35] demonstrated that NN is also able to cooperate with a tree-based data representation.

In contrast to the above representations that directly encode a music work, EC can evolve indirect representations to compose music. Specifically, GP evolves different types of programs to produce music: tree-based programs [73]–[77], graph-based programs [78], and sequential programs [24], [79], [80]. Grammatical evolution (GE) was used to evolve grammar rules [81]–[83] for generation of music pieces. Evolving recurrent neural networks (RNNs), from which music is sampled, exerts the synergy of EC and NN [84], [85].

Audio representations such as spectrogram are sometimes used in music composition systems. Wang and Yang [86] proposed the PerformanceNet model which generates the spectrogram of its input pianoroll. Wei et al. [87] presented a drumming system that generates drum patterns suitable for an input spectrogram. Additionally, Huang et al. [88] trained a CNN to identify the compatibility of candidate excerpts given their spectrograms.

III. Evolutionary Computation Based Composition

Evolutionary algorithms (EAs) are population-based, quality driven, and stochastic optimization algorithms that follow the

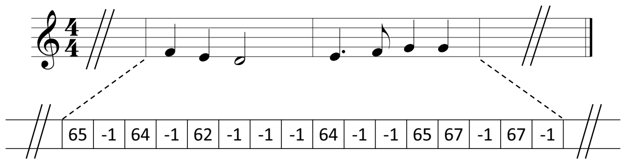
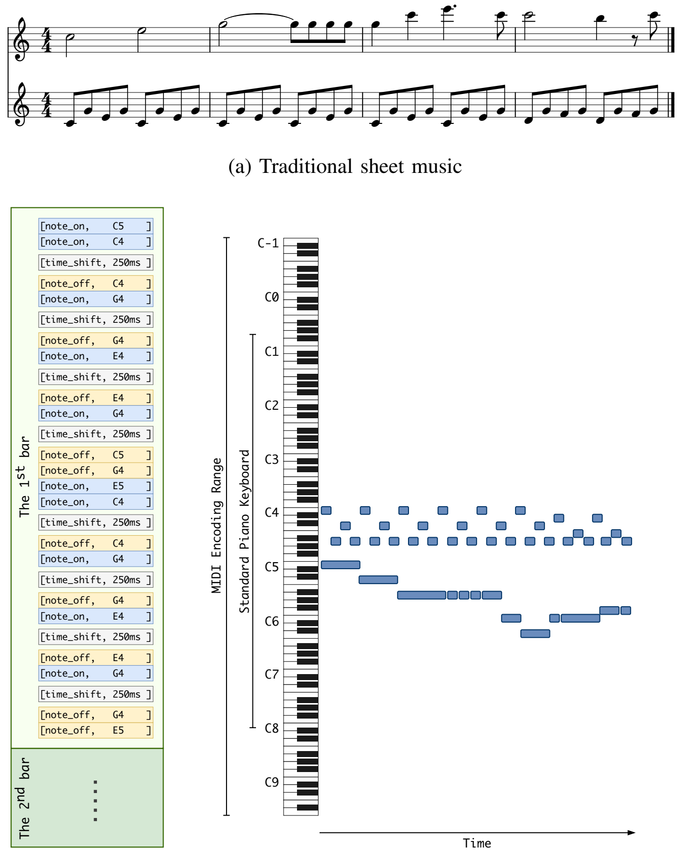


Fig. 3. Example linear representation for music pieces.



(a) Traditional sheet music

(b) Event-based encoding

(c) Pianoroll

Fig. 4. Pianoroll and event-based representations of the beginning excerpt of Sonatina Op.20 No.1 Friedrich Kuhlau.

notion of natural selection and evolution theory. The powerful searching ability of EAs has received many successes in diverse types of optimization problems. In the EC studies, music composition is ordinarily formulated as a combinatorial optimization problem aiming for a combination of music notes that achieves satisfactory quality based on the given evaluation measure. The solution representation, reproduction operators, and evaluation function are three essential parts that form an eligible EA. The two former components are generally designed jointly, while the evaluation function can be designed separately. This section reviews music representations and evolutionary mechanisms, followed by the fitness evaluation strategies.

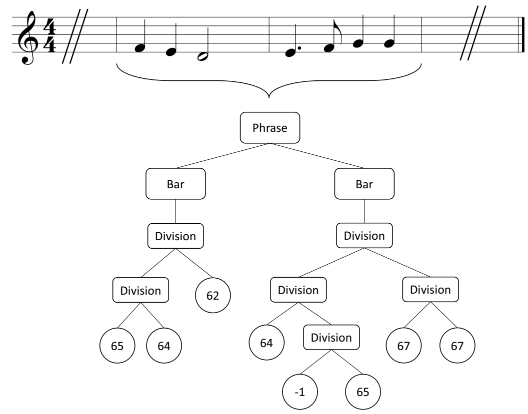


Fig. 5. Example tree representation for music pieces.

A. Algorithms

There have been dozens of EAs, mainly genetic algorithm (GA), presented for music composition [18]. In addition to GA, particle swarm optimization (PSO) [89], [90], ant colony optimization (ACO) [91]–[94], and differential evolution (DE) [95]–[97] have been developed for generating compositions. In light of the targets of evolutionary manipulation, PSO and DE are similar to GA in that each individual in the population represents a music piece. By contrast, the ACO systems generate music pieces through a constructive procedure. The GP and GE systems evolve either music pieces or systems that generate music pieces.

In addition to the canonical EAs inspired from Darwinian evolution, memetic algorithms (MAs) are found useful for music composition. MAs implement Lamarckian or Baldwinian theory of biological evolution, which evaluate candidate solutions according to their genetic information or traits explored by local search, heuristic search, or metaheuristic search [98]. Acampora et al. [44] proposed a hybrid intelligence model which combines fuzzy control and several metaheuristic algorithms for automatic harmonization of figured bass. The problem considers composition for a four-voice setting, i.e., soprano, alto, tenor, and bass. Given a bass line and its corresponding figures (chord information), the problem-solver is asked to complete the other voices in compliance with harmonization rules, e.g., the guidelines in Johann Joseph Fux's Gradus Ad Parnassum [99]. The hybrid model employs fuzzy control as the conductor that triggers metaheuristic-based optimizers in different stages of evolution. A follow-up study was proposed by Muñoz et al. [45] using a hybridization model for solving the unfigured bass problem, which is similar to the figured bass problem but provides only the bass without figures. Mańdziuk et al. [100] applied an MA that integrates a modified GA and four music-related local heuristics for composing romantic classical piano solos.

B. Fitness Evaluation

Music quality assessment is a challenging task in evolutionary music composition. As a quality-driven optimization method, EAs manipulate the population to pursue improvement of fitness values, where the search direction is guided by the results of fitness evaluation. A precise evaluation function is advantageous to the efficiency and effectiveness of EAs. Hence, various fitness evaluation methods have been proposed for assessment of music. These methods can be categorized into three types [18]: interactive evaluation, rule-based evaluation, and learning-based evaluation. Figure 6 presents a general framework of existing evolutionary composition systems. The following subsections will describe and discuss these three types of fitness evaluation. Tables I–III further summarize the reviewed studies as per the three categories.

1) Interactive Fitness Evaluation: The general goal of evolutionary music composition is to automatically generate music that maximizes the satisfaction of audience. To achieve this goal, one intuitive way is to introduce human's feedback into fitness evaluation. Interactive evolutionary composition enables this composition method and is known for its ability of composing customized music. Based on the feedback from the user, interactive evolutionary algorithms search for the melodies that fit the user's preference. In general, the human evaluator is asked to rate, score, or compare the generated music pieces. Different from the questionnaire-like feedback, Moroni et al. [101] developed a system that allows users to interact with the evolutionary composing system through a graphic pad. With novel interaction media, Unehara et al. [102] adopted electroencephalography (EEG) to measure human responses for evaluation of candidate music.

The interactive genetic algorithms (IGAs) provide a way hinged on human aesthetics to evaluate the generated compositions; however, they are plagued with the issues of high evaluation latency, user fatigue, and inconsistent evaluation. More specifically, the evaluation process based on interaction can be much slower than automatic evaluation because human evaluators must listen through all the candidates enquired. The high evaluation latency further lengthens the time of evolutionary process. Evaluator's patience can be worn down by the long period of search and the numerous evaluation requests. Moreover, human fatigue and loss of attention are inevitable and hardly tractable in the interactive scenario, causing inconsistent and low-fidelity evaluation.

Some studies propose reducing the workload at evaluation to mitigate the human fatigue issue (see Table I). One way is to cluster the population into several groups according to a similarity metric [103], [104] and then evaluate only the centroid of each cluster. This approach can substantially decrease the amount of music works for evaluation. Nonetheless, designing a similarity metric conforming to human perception can be challenging as well. Evaluating only fragments instead of a full-length composition can also reduce the evaluation time and help to alleviate human fatigue [105]–[108]. After evolution, the short pieces with highest fitness values may be concatenated into a complete piece of music.

Another issue of interactive evaluation occurs at the beginning of evolutionary process: the low-quality and nearly random music pieces are highly likely to harass the evaluators and waste their time and effort. The techniques for automatic preliminary quality check involve heuristics [76], [78], [109], surrogate models [76], [110]–[112], and their combination [13]. These methods filter out low-quality candidates using heuristic functions or surrogate models. Preventing the system to generate low-quality music can also save human evaluation. The methods based on this notion include smart initialization [2], [113], [114], heuristic operators [105], [115]–[117], and searching in high-level feature spaces [11], [90].

Table I summarizes the studies on interactive evolutionary composition, in which the approaches for addressing the high evaluation cost and human fatigue problem can be categorized into four major techniques:

1) clustering candidates and evaluating only their representatives;
2) evolving merely subcomponents and assembling good ones to complete the whole music piece;
3) evaluating or filtering out low-quality candidates by heuristics or surrogate models;
4) incorporating smart initialization or heuristic operators to avoid generating low-quality candidates.

2) Rule-Based Fitness Evaluation: Over a long period of time, musicians have accumulated extensive knowledge and experience in music composition. Inclusion of such professional opinions into EAs can improve the efficiency and quality of fitness evaluation. A significant approach is to transform music knowledge into computable functions or rules. In this regard, the guidelines in music theory suggest the dos and don'ts for composition of a wide range of music types. However, only a few guidelines are well-defined because music knowledge is delivered from generation to generation in natural languages, which might be recorded or explained imprecisely.

Counterpoint and four-part harmonization, two music forms prevailing in the pre-Baroque and Baroque periods, strictly follow the regulated rules. Harmonization is a main task in counterpoint and four-part harmonization. It focuses on composition of polyphony[1]. In algorithmic composition, harmonization is generally formulated as a constraint satisfaction problem: given the melody of one or multiple parts, the remaining parts must be completed in compliance with composition rules. The counterpoint [99] and voice leading [128] are the disciplines related to the composition rules. The principle of counterpoint is to maintain consonance among parts; for instance, the notes of different parts that occur at the same moment ordinarily need to be perfect consonant or imperfect consonant to the others. Voice leading states a fundamental idea that every part needs to be reasonably singable to vocalists. For example, a rule may define that, within a single voice, the pitch range is confined and the pitch interval of consecutive notes must be small. However, some of the rules could be mutually incompatible. To address this issue, several recent studies employ multiobjective EAs to search for the trade-off solutions [72], [129], [130]. Instead

[1]The term polyphony is interpreted differently in music taxonomy and music information retrieval. Regarding music genres, polyphony typically refers to the type of music that features the existence of parallel melodies. Each of the melodies is self-contained and yet harmonic to the others. In music information retrieval, polyphony usually describes the music data in which at least one moment, multiple music notes are played together or overlap with one another.

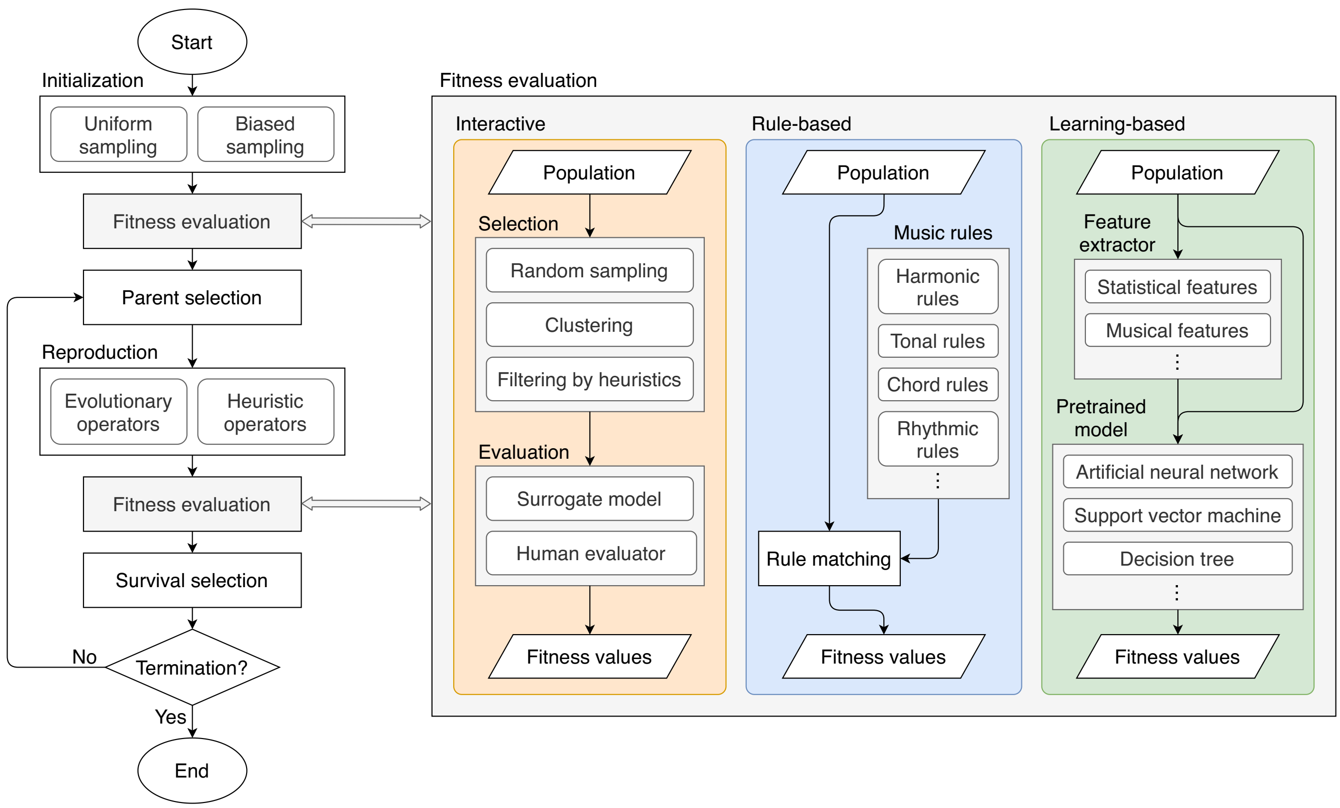


Fig. 6. General framework of existing evolutionary composition systems.

of codifying explicit evaluation rules, Chang and Chen [92]–[94] proposed AntsOMG which embeds music knowledge in the edge costs to manipulate the transition probabilities of compositional actions. Such a system demonstrates a novel way to incorporate music knowledge in EA.

Table II lists the rule-based evaluation associated with harmonization in the first category. These studies aim to generate the remaining parts in harmony with the given parts. Development of figured bass and unfigured bass belongs to this kind of problem, where the bass voice is given and the other three parts (soprano, alto, and tenor) need to be generated. A difference lies in that figured bass problem provides complementary information of chords in problem definition, whereas the unfigured bass does not. Existing studies have shown that EAs are capable of handling harmonization, e.g., four-part harmonization [22], [131]–[134], species counterpoint [75], [93], [94], [112], fugue [135], with the rule-based fitness evaluation functions.

In contrast with polyphony, monophony in music taxonomy represents the type of music that has a single dominant melody, while the other parts together with harmonic chords form its accompaniment. Algorithmic monophony composition usually involves one or some of the following goals:

1) composing single or multiple melodies given chord progressions [21], [43], [68], [136], [137] or reversely [3], [138];
2) composing melody of a specific genre [21], [68], [139]–[141];
3) composing melody of fused genres [20], [142];
4) composing accompaniment that fits a chord progression or melody [7], [15], [21].

In these tasks, compiling the compositional convention and music knowledge into the evaluation rules has always been a major challenge. As the second and third categories of Table II indicate, although the evaluation rules vary a lot from task to task, most of the evolutionary systems consider consonance and dissonance between the melody and chords in order to manipulate the musical texture of tension and resolution.

As for the implementation of rule-based fitness evaluation, both unweighted and weighted rules are considered in the studies [8], [143]. The fitness function using unweighted rules evaluates individuals by matching the individuals with the rules, and its output fitness value is usually related to the number of rules matched. By contrast, the fitness function using weighted rules additionally considers the weights of matched rules. The fitness function using unweighted rules is easier to set because determining weights for the rules itself turns out to be a professional task. On the other hand, the systems using weighted rules allow users to specify the importance of each rule and guide the optimizer to focus more on the highly weighted ones. The disadvantage of rule matching is the tendency of forming tier-structured fitness landscape. A rule that discourages huge pitch interval between notes, for instance, may be unable to distinguish a huge interval from a very huge interval. Therefore, designing the rules that can express the degree of compliance beyond binary

TABLE I
INTERACTIVE EVOLUTIONARY ALGORITHMS IN MUSIC COMPOSITION. THE TERM SUB. TEST IS SHORT FOR SUBJECTIVE TEST, REF. FOR REFERENCE, ACCOMP. FOR ACCOMPANIMENT, CHORD PROG. FOR CHORD PROGRESSION, REL. FOR RELATIVE, AND ABS. FOR ABSOLUTE.

Technique		Task	Style	Algorithm	Representation	Sub. Test	Year	Ref.
Workload reduction	Clustering	melody	variation	GA	linear (rel.)	-	'95	[103]
	Sampling	melody	-	GA	linear (abs.)	scoring	'00	[118]
		melody, accomp.	-	GA	linear (abs.)	scoring	'02	[119]
		melody, accomp.	-	GA	linear (abs.)	scoring	'03	[120]
		melody, accomp.	-	GA	linear (abs.)	scoring	'05	[121]
	Evolving segments	melody	jazz	GA	linear (abs.)	-	'94	[105]
		melody	jazz	GA	linear (abs.)	-	'98	[115]
		melody	jazz	GA	linear (abs.)	-	'99	[116]
		melody	jazz	GA	linear (abs.)	-	'02	[117]
		melody, accomp.	-	GA	linear (abs.)	-	'01	[106]
		melody, accomp.	-	GA	linear (abs.)	-	'01	[122]
		melody, accomp.	-	GA	linear (abs.)	-	'01	[123]
		melody, accomp.	-	GA	linear (abs.)	-	'02	[124]
		melody	-	GA	-	-	'06	[107]
		melody	-	GA	linear (abs.)	-	'07	[108]
	Evolving music features	melody	-	GA, PSO	linear (abs.)	-	'16	[90]
		melody; drum	-	GA	latent vector of VAE	scoring	'18	[11]
	Pairwise comparison	melody	sign sound	DE	linear (abs.)	scoring	'11	[95]
Quality assurance	Surrogate	melody	-	GP	tree (program)	-	'98	[76]
		rhythm	-	GP	tree (hierarchical)	-	'00	[13]
		melody	-	GA	linear (abs.)	-	'06	[125]
	Heuristics	melody, accomp.	-	GA	linear (abs.)	scoring	'03	[126]
		melody, accomp.	-	GA	linear (abs.)	scoring	'04	[109]
		chord prog.	-	GA	binary	scoring	'10	[2]
		melody, accomp.	-	GP	executable graph	-	'11	[78]
	Smart Initialization	melody	-	GA	linear (abs.)	-	'04	[113]
		chord prog.	-	GA	binary	scoring	'10	[2]
		melody	-	GA	linear (abs.)	scoring	'11	[114]
	Heuristic operators	melody	jazz	GA	linear (abs.)	-	'94	[105]
		melody	jazz	GA	linear (abs.)	-	'98	[115]
		melody	jazz	GA	linear (abs.)	-	'99	[116]
		melody	jazz	GA	linear (abs.)	-	'01	[117]
Others / unspecified	EEG device	melody	-	GA	linear (abs.)	-	'14	[102]
	Graphic pad	melody, accomp.	-	GA	linear (abs.)	-	'99	[101]
	Evolving rule weights	expression	-	GA	weights of rules	A/B test	'08	[127]
	-	chord prog.	-	GA	linear (abs.)	scoring	'10	[1]
	-	drum fill-in	-	GA	linear (abs.)	-	'11	[9]
	-	chord prog.	-	GA	linear (musical role)	scoring	'14	[4]

level is generally beneficial to the search efficiency of rule-based evaluation systems.

3) Learning-Based Fitness Evaluation: Hybridization of CI techniques has gained considerable encouraging results in various areas. In music composition, hybrid systems commonly adopt machine learning techniques such as NN to construct the fitness evaluation function and then apply an EA to generate music pieces [12], [111], [162]–[164]. Table III lists the studies using such learning-based fitness evaluation. In these hybrid systems, NNs are trained on symbolic music corpuses that serve as the archetypes of music to be composed by the systems. Other machine learning algorithms [25], [125], e.g., decision tree, can also be applied. Furthermore, EAs are used to search for the music pieces that maximize or minimize the model's response.

Some challenges arise from the hybrid systems. First, directly training an evaluation model with raw music data may not work due to diverse aspects of information in the data: melody line, bass line, chord accompaniment, chord progression, rhythm patterns, phrasing, forming, harmonization, and so on. In addition, transposing a melody to another key, for example, alters the represented data, but the transposed melodies may still sound similar or even be the same as the original. Data processing is therefore needed for machine leaning algorithms to learn meaningful and desired evaluation from the music data, whereas music knowledge and expertise is required for processing the data. Instead of training models from music data, some studies propose training the evaluation model using selected musical features [25], [70], [149] or statistical metrics (e.g., Zipf's law [70], [146] and Bayesian surprise [165]). These methods also require music knowledge and understanding of the music corpuses for constructing the musical features of interest. The second category in Table III presents the methods related to learning from statistical features.

Learning-based systems have been studied on several composition tasks. Their applicability is currently subject to the presence of music corpuses, which must be symbolic-coded,

TABLE II
RULE-BASED EVOLUTIONARY ALGORITHMS IN MUSIC COMPOSITION.

Category	Task	Style	Algorithm	Representation	Sub. Test	Year	Ref.
Harmonization	polyphony	four-part harmonization	GA	linear (abs.)	-	’94	[131]
	polyphony	counterpoint	GP	tree (program)	-	’97	[75]
	polyphony	four-part harmonization	GA	linear (abs.)	-	’98	[61]
	polyphony	four-part harmonization	GA	linear (abs.)	-	’99	[63]
	polyphony	four-part harmonization	GA	linear (abs.)	-	’99	[132]
	polyphony	-	GA	linear (abs.)	-	’00	[144]
	polyphony	four-part harmonization	GA	linear (abs.)	-	’00	[133]
	melody	counterpoint	GA	linear (abs.)	-	’02	[112]
	melody	fugue	GA	linear (rel.)	-	’04	[135]
	polyphony	harmonization	ACO	linear (rel.)	-	’07	[91]
	polyphony	four-part harmonization	GA	linear (abs.)	-	’09	[145]
	polyphony	four-part harmonization	GA	linear (abs.)	-	’10	[130]
	polyphony	four-part harmonization	MA	linear (abs.)	comments	’11	[44]
	polyphony	four-part harmonization	GA	linear (abs.)	-	’11	[134]
	chord prog.	harmonization	GA	linear (abs.)	-	’11	[3]
	polyphony	four-part harmonization	MA	linear (abs.)	-	’16	[45]
	melody	counterpoint	GA	linear (abs.)	-	’17	[146]
	polyphony	four-part harmonization	GA	linear (abs.)	-	’17	[22]
	polyphony	four-part harmonization	GA	tree (hierarchical)	-	’20	[129]
	polyphony	counterpoint	ACO	linear (abs.)	-	’21	[93]
	polyphony	counterpoint	ACO	linear (abs.)	-	’21	[94]
Genre	melody	jazz	GP	tree (program)	-	’94	[73]
	melody	jazz	GA	linear (abs.)	-	’98	[68]
	style modulation	J. S. Bach to jazz	GA	linear (abs.)	-	’06	[142]
	melody, accomp.	rock	GA	linear (abs.)	-	’08	[147]
	melody	jazz	GA	rule (binary)	-	’08	[148]
	melody	Chinese pop	GA	linear (abs.)	-	’10	[149]
	melody	bossa nova	EA	linear (abs.)	-	’12	[139]
	melody, accomp.	Romantic era	MA	linear (abs.)	scoring	’13	[100]
	melody	jazz	GA	linear (rel.)	-	’15	[150]
	melody	Jay Chou	GA	pattern templates	-	’15	[137]
	melody	Chinese	GA	linear (abs.)	-	’17	[140]
	melody, accomp.	western, Chinese	GA	linear (abs.)	-	’17	[21]
	melody, accomp.	flamenco×tango	GA	linear (abs.)	-	’17	[20]
	melody	unaccompanied cello	GA	linear (abs.)	-	’19	[151]
	melody	cantus firmus	ACO	linear (abs.)	-	’20	[92]
	melody, accomp.	bossa nova	GA	linear (abs.)	-	’20	[141]
General	melody, accomp.	-	GA	linear (abs.)	-	’99	[43]
	melody	-	GA	linear (abs.)	-	’01	[152]
	melody	-	GA	linear (abs.)	-	’07	[153]
	melody	-	GA	linear (rel.)	-	’08	[8]
	melody	-	GA	linear (rel.)	-	’09	[143]
	melody	-	GE	integer string	scoring	’09	[82]
	melody	-	GA	linear (rel.)	-	’10	[67]
	melody, accomp.	-	GP	executable graph	-	’11	[78]
	motif	-	GA	linear (abs.)	-	’11	[154]
	accomp.	-	GA	linear (abs.)	-	’12	[15]
	melody	-	GA	linear (abs.)	-	’13	[136]
	melody	-	GA	linear (abs.)	-	’14	[155]
	melody	-	GE	integer string	-	’15	[83]
	melody	-	GA	linear (abs.)	-	’17	[156]
Others	melody	thematic bridging	GA	linear (abs.)	-	’91	[157]
	melody	thematic bridging	GA	linear (abs.)	-	’93	[158]
	percussion	drum set	GA	linear (abs.)	-	’05	[7]
	arrangement	guitar	GA	binary	-	’06	[159]
	motif	twelve-tone technique	GA	linear (abs.)	-	’09	[160]
	melody	twelve-tone technique	GA	linear (abs.)	-	’10	[14]
	melody	jamming	GA	linear (abs.)	-	’11	[161]

TABLE III
LEARNING-BASED EVOLUTIONARY ALGORITHMS IN MUSIC COMPOSITION.

Evaluation model		Task	Style	Algorithm	Representation	Sub. Test	Year	Ref.
NN	SLP	polyphony	four-part harmonization	GA	binary	-	’91	[162]
	MLP	melody	jazz	GP	tree (program)	-	’95	[74]
		melody	jazz	GA	linear (abs.)	-	’96	[111]
		percussion	drum set	GA	linear (abs.)	-	’98	[12]
		melody	-	GA	linear (abs.)	-	’05	[163]
		melody, accomp.	classical	GP	tree (hierarchical)	scoring	’07	[70]
		melody	Chinese pop	GA	linear (abs.)	-	’10	[149]
		melody	-	GA	linear (abs.)	-	’20	[166]
	SOM	melody	digital game	GP	tree (program)	-	’09	[77]
	RNN	melody	pop	GA	linear (abs.)	-	’11	[167]
Musical & statistical features		melody	-	GE	integer string	-	’02	[81]
		melody	imitation	GA	linear (abs.)	-	’07	[168]
		melody	-	GA	linear (abs.)	-	’08	[169]
		melody	-	GA	linear (abs.)	-	’08	[164]
		melody	-	GA	linear (abs.)	-	’08	[170]
		melody	-	GA	linear (abs.)	-	’11	[171]
		melody	-	GA	linear (abs.)	-	’12	[27]
		melody	Japanese pop	GA	linear (abs.)	-	’14	[26]
		melody	classical, classical rock	GA	linear (abs.)	-	’16	[25]
		melody	-	GP	tree (hierarchical)	-	’16	[71]
		melody	digital game	GA	linear (abs.)	scoring	’17	[165]
		polyphony	J. S. Bach	LGP	instructions	-	’17	[79]
		polyphony	J. S. Bach	LGP	instructions	-	’18	[80]
		melody	Hungarian folk	LGP	instructions	-	’19	[24]
		melody	-	GP	tree (hierarchical)	-	’19	[72]
Decision tree		melody	classical + classical rock	GA	linear (abs.)	-	’16	[25]

well-organized, open to public, and large in volume.

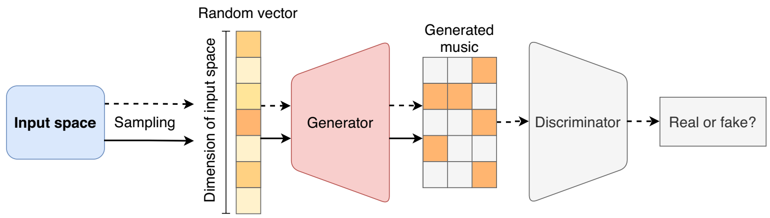
IV. NEURAL NETWORK BASED COMPOSITION

Machine learning techniques have been applied to model musical features for handling music-related tasks, such as music genre classification, recommendation, transcription, style transfer, and music generation. NNs are computational models inspired from biological neural networks. An NN consists of mutually connected neurons, which act as the processing units. Generally, neurons are arranged layer by layer. Perceptron is a simple NN model that computes the weighted sum of its input neurons and forwards the result from an activation function to its output neurons. Considering the sequential and temporal nature of music, RNNs comprise not only current input but also previous states. This type of NNs are found more eligible for modeling musical features than MLP. For example, long short-term memory (LSTM) [172], featuring the ability of modeling long-term structure of temporal data, has become a popular model for music generation tasks. In addition, DL has recently received much attention owing to its promising performance, e.g., deep RNN, variational autoencoder (VAE), generative adversarial networks (GAN), and transformer.

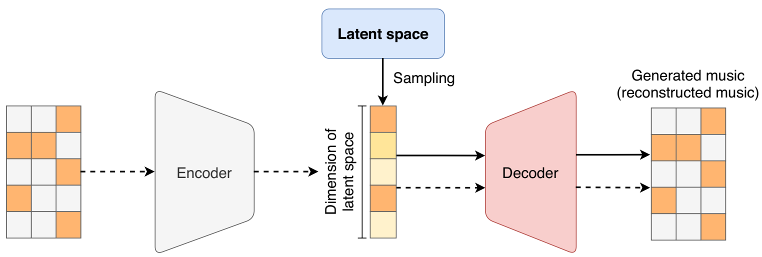
The recent development of NNs has turned their roles in music composition from an auxiliary tool to a stand-alone predominant technique. A variety of NN and DL based composition systems have been proposed. The two reviews [54], [173] categorized them according to the model architecture, neuron types, and data representations. However, in music composition, the NN models are found polymorphic and hybridized; thus, clear classification becomes unrealistic. For example, GAN may comprise convolutional neurons [31], [32], [174], recurrent neurons [29], and even both [175]. The convolutional and recurrent neurons were also used together to process the pianoroll data [30], [176]. As another example, attention-based network has been implanted in GAN [30], [177], [178] and autoencoder [179]. To deal with this difficulty in classification, this study presents a new taxonomy that categorizes NN-based composition systems according to their sampling strategy for generating music from the model: 1) sampling from input space, 2) sampling from latent space, and 3) sampling from output space. First, a system that samples music from its input space aims to learn a mapping of an arbitrary point from the input space to a piece of music. Second, a system that samples music from its latent space learns to transform a piece of music into a point in the latent space and transform an arbitrary point in the latent space back into a piece of music. Third, a system that samples music from its output space learns to output a probability vector from which music excerpts are sampled. Figure 7 presents three simplified yet characteristic frameworks with respect to the three sampling strategies. Tables IV–VI further summarize their corresponding studies.

A. *Sampling from Input Space*

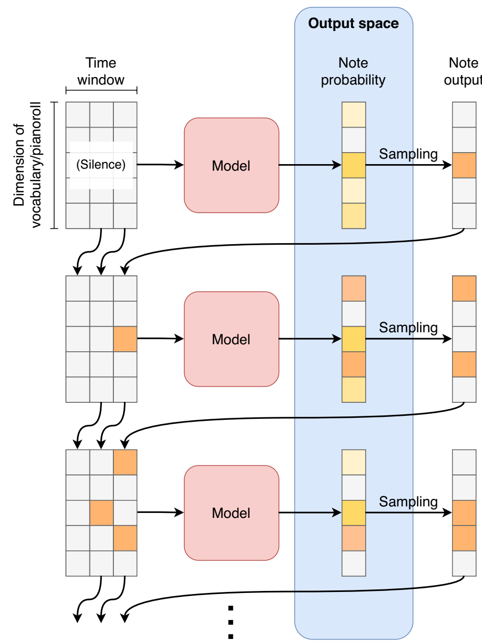
This sampling strategy aims to train a model for transforming a given or random input vector into a music piece. The most famous framework in this category is the generative adversarial network (GAN) [180]. A GAN is composed of a generator and a discriminator. In the training phase, the generator is fed with random vectors and then outputs its



(a) Sampling-from-input-space strategy.



(b) Sampling-from-latent-space strategy.



(c) Sampling-from-output-space strategy.

Fig. 7. Frameworks of NN-based music composing systems that sample music from the input, latent, or output space. In (a) and (b), the dashed and the solid arrows indicate the data flow in the training and generating phases, respectively. In (c), the solid arrows indicate the data flow in both training and generating phases.

generated data. The discriminator learns to distinguish between generated data and real data. Restated, the generator is trained to generate data similar enough to fool the discriminator, whereas the discriminator is trained to accurately separate the generated data and real data apart. After the training phase, the resultant generator is used to compose music by mapping arbitrary random vectors to music sequences. The GAN-based music generation systems typically include RNNs, CNNs, or both.

By regarding music as time series data, recurrent units have been integrated into GAN for music generation and discrimination. Mogren [29] combined RNNs and adversarial training as the continuous RNN-GAN for music generation. The generator is a unidirectional deep LSTM network and the discriminator is a bidirectional deep LSTM network. After adversarial training, the generator composes music with the random vectors sampled from input space by a common autoregressive prediction. Yu et al. [181] proposed a conditional LSTM-GAN that composes melody for given lyrics. To this end, the conditioning mechanism is introduced to both of the generator and discriminator based on LSTMs, allowing the two networks being conditioned on the same lyrics embedding.

Through separating music data with sliding window, the extracted fragments provide another viewpoint to music content: each extracted fragment represents a sparse matrix that encodes the activated pitches in the corresponding time window. For a specific moment in the window, the activated pitches form a one-dimensional one-hot or multi-hot vector. A sequence of one-hot vectors can represent a monophonic melody; furthermore, a sequence of multi-hot vectors can represent a polyphonic music fragment. Based on this idea, several studies treat the extracted matrices as images and employ CNNs to model music content beat by beat, bar by bar, or phrase by phrase. The combination of CNN and GAN presents a feasible implementation. Yang et al. [30] proposed a CNN-based GAN system, called MidiNet, which generates monophonic melody in a bar-by-bar manner. By including the conditioner CNN, MidiNet accepts conditioning through taking multiple preceding bars as the priming melody to be continued. Dong et al. [31] developed MuseGAN, a deep CNN-based GAN system for multitrack pianoroll generation. MuseGAN employs a divide-and-conquer approach generating music hierarchically. Specifically, MuseGAN aggregates multiple sub-CNNs, each of which is responsible for a part of music modeling, including modeling temporal dependency, inter-track dependency, and intra-bar notes dependency. A follow-up study [32] refined the process of sampling actually activated notes from the real-valued matrix output by the generator network. Based on MuseGAN, an additional network is appended to the generator for discretizing the output matrix. Liu et al. [174], [182] adapted MuseGAN for generating the lead sheet, which is further taken as a sampling condition of the other GAN to compose arrangement for five instruments.

In light of the individual advantages of different network types, hybridization has been a viable way to boost DL. Li et al. [175] proposed a GAN system with two RNN and CNN based discriminators for interbar analysis and intrabar analysis, respectively. Wei et al. [87] presented a GAN system to generate drum patterns given a melodic excerpt. This system is innovative in two aspects: 1) the ability to accept both audio and symbolic data as input and 2) the implantation of a VAE into the GAN framework.

Some early studies investigated restricted Boltzmann machine (RBM) and deep belief network (DBN) for modeling and generating music data. A standard RBM consists of only two layers, namely the visible and hidden layers, where a DBN extends RBM by stacking multiple layers to deepen its structure. The goals of RBM and DBN are to transform input data in the hidden space and to reconstruct the data back from the hidden vectors. The visible layer is used to receive as input the original data and emit as output the reconstructed data; accordingly, the number of nodes in the visible layer is practically identical to the dimension of input data. A narrow hidden layer is treated as an information bottleneck for dimension reduction; by contrast, a wider hidden layer allows the model to yield a richer latent representation. In RBM and DBN, the data encoding and reconstruction processes are done by the same network architecture and weights but in the opposite directions. To generate new data from a trained model, one can sample a random input vector from the visible space and then perform block Gibbs sampling until convergence. An intuitive implementation of RBM and DBN for music generation is to set the dimension of visible units to be the scale (88 notes) of a piano keyboard. Boulanger-Lewandowsk et al. [183] proposed the RBM using recurrent units (RNN-RBM) and tested it on reconstruction of the polyphonic pianorolls. Similarly, Goel et al. [184] attempted to model and generate music with the recurrent DBN. Applying the random sampling method, users are not involved in the generation process. Instead, Lattner et al. [185] proposed the constrained sampling strategy to gradually adjust the intermediate result for complying with the user-defined constraints (e.g., tonality and music form) during the sampling iterations.

The major attention of the community falls on designing a system that can compose music by arranging music notes. Differently, Deep Composer [186] presents an LSTM system that learns a hash-based representation for existing music segments so that the successiveness of these segments can be measured by the Hamming distance. Rather than directly producing music notes, Deep Composer serves as a coordinator retrieving suitable segments from its database to continue a query segment.

B. Sampling from Latent Space

The latent space refers to the input space of a hidden layer in a neural network. It is usually the bottleneck of stacked autoencoders. An autoencoder typically consists of two subnetworks, i.e., the encoder and the decoder. The encoder is trained to extract characteristic information from the input data and encodes the information into a latent vector, whereas the decoder is trained to reconstruct the input data with the information contained in the latent vector. Variational autoencoders (VAEs) extend the autoencoder by introducing an extra constraint to the latent representation. In VAEs, the distribution of latent representation is trained to approximate a prior probability distribution, which is usually diagonal-covariance Gaussian. This constraint is harnessed to guide VAE to model semantically meaningful transition from a data point to another in the latent representation. After model training, new data can be generated by sampling random vectors from the latent space, and then VAE utilizes the decoder to map the sampled latent vectors to new data. In addition to generating by random sampling, the approaches such as interpolation, extrapolation, and user-guided sampling [11] are also usable in VAEs. By sampling the latent vectors that are linear to certain reference latent vectors, VAE can transform the sampled vectors to the data that retain or fuse the characteristics of its reference points.

Roberts et al. [10], [33] proposed MusicVAE featuring a hierarchical decoding mechanism for modeling monophonic music. The encoder of MusicVAE adopts bidirectional LSTM and follows a common stacked structure of VAE. The decoder of MusicVAE consists of two decoding subnetworks, i.e., the conductor RNN and the decoder RNN, both applying unidirectional LSTM. Instead of directly decoding a latent vector to music notes, the hierarchical decoding uses the conductor RNN to map a latent vector to a sequence of embedding vectors, and uses the decoder RNN to generate a sequence of music notes autoregressively from each embedding vector. Simon et al. [195] extends MusicVAE for handling multitrack (up to eight tracks) polyphonic music. The extended system first encodes each track into an embedding and then converts the multiple embeddings into a single latent vector. The decoding process is done reversely. The study further investigated chord conditioning in MusicVAE, which allows the model to produce music sequences over a chord progression. Pati and Lerch [198] introduced rhythmic complexity and pitch range to regularize the loss function of MusicVAE for improving the controllability of music generation. Gillick et al. [69] modified MusicVAE as the GrooveVAE model for generating expressive drum performance. Guo et al. [199] proposed using VAE to encode the tonal tension of music pieces in the latent vectors. By manipulating the latent vector of a music piece, this system changes the pitches of the piece and produces tenser or looser pieces. Based on a similar concept, VAE can be used to model musical transition in accord with the arousal level [205]. In addition, Hung et al. [197] demonstrated with VAE that simple transfer learning techniques, e.g., parameter fine-tuning, can benefit the generative models for under-resourced music genres.

Koh et al. [196] proposed a convolutional-variational RNN (CVRNN), which integrates VAE, CNN and gated recurrent unit (GRU). CVRNN has a novel asymmetric structure, which is a two-layered CNN followed by a GRU-based VAE. The convolutional layers of CVRNN serve as a feature extractor that transforms the input framed pianorolls into feature vectors before forwarding to the GRU-based VAE. Wang et al. [201] presented a GRU-based VAE system learning two latent representations, i.e., the chord vector and the texture vector, which encode chord information and music notes, respectively. By combining the chord vector from one music piece and the texture vector from another piece, the system can generate a new piece in between the reference pieces. Choi et al. [179] also demonstrated that autoencoder can be applied to learn a latent representation for music performance, showing the potential for performance modulation in music.

TABLE IV
NEURAL NETWORK SYSTEMS THAT SAMPLE THEIR PRODUCTS FROM INPUT SPACE FOR CNN (C), GRU (G), LSTM (L), AND RNN (R).

Model	Task	Style	Condition	Representation	Dataset	Sub. Test	Year	Ref.
RNN	melody	-	-	-	-	-	’12	[85]
LSTM	melody	Mozart	melody	pianoroll	Mozart K. 545	-	’13	[187]
LSTM	block retrieval	folk	prime	hash-based	NMD [188]	scoring	’20	[186]
GAN (L)	general	classical	-	MIDI (vector)	manual	-	’16	[29]
GAN (C)	melody	pop	chord	pianoroll	TheoryTab [189]	-	’17	[30]
GAN (G)	general	alternative	-	pianoroll	LPD [31]	A/B test	’18	[32]
GAN (C)	pop band	rock	prime	pianoroll	LPD [31]	scoring	’18	[31]
GAN (C+R)	lead sheet, arrangement	mixed	lead sheet	pianoroll	LPD [31], TheoryTab [189]	voting, scoring	’18	[174]
GAN (C+L)	melody	Korean pop	-	MIDI (vector)	manual	-	’19	[175]
GAN (C; VAE)	drum	-	spectrogram	pianoroll	LPD [31]	A/B test	’19	[87]
GAN (L)	melody	-	lyrics	event	LMD† [181]	scoring	’21	[181]
RBM (R)	general	classical, folk	-	pianoroll	JSB [190], MuseData [191], NMD [188], piano-midi [192]	-	’12	[183]
DBN (R)	general	classical, folk	-	pianoroll	JSB [190], MuseData [191], NMD [188], piano-midi [192]	-	’14	[184]
RBM (C)	piano	Mozart	music form	pianoroll	manual	-	’18	[185]

TABLE V
NEURAL NETWORK SYSTEMS THAT SAMPLE THEIR PRODUCTS FROM LATENT SPACE FOR CNN (C), GRU (G), LSTM (L), RNN (R), AND TRANSFORMER (T).

Model	Task	Style	Condition	Representation	Dataset	Sub. Test	Year	Ref.
VAE (L)	piano	Beethoven	latent vector	pianoroll	manual	-	’17	[193]
VAE (L)	melody, drum, trio	-	latent vector	MIDI (vector)	manual	-	’17	[33]
VAE (L)	melody, drum, trio	-	latent vector	MIDI (vector)	manual	A/B test	’18	[10]
VAE (L)	general	folk	chord prog.	event	LMD [194]	-	’18	[195]
VAE (C+G)	general	folk	latent vector	pianoroll	NMD [188]	-	’18	[196]
VAE (G)	melody	jazz	-	pianoroll	TheoryTab [189]	scoring	’19	[197]
VAE (L)	melody, drum, trio	-	music attr.	MIDI (vector)	manual	-	’19	[198]
VAE (L)	drum performance	-	drum pattern	pianoroll	GMD [69]	A/B test	’19	[69]
VAE (G)	melody, accomp.	pop	prime	pianoroll	LMD [194]	-	’20	[199]
VAE (G)	piano	pop	chord prog., melody	pianoroll	POP909 [200]	scoring	’20	[201]
VAE (L)	general	Bach, folk	style	MIDI (vector)	JSB [190], NMD [188]	-	’20	[202]
VAE (G)	piano	-	arousal		e-piano [203], VGMIDI [204]	scoring	’20	[205]
VAE (G)	piano	-	latent vector	tree (hierarchical)	manual, POP909 [200]	voting	’20	[35]
AE (T)	performance	-	excerpt, melody	event	MAESTRO [206], YouTube	scoring	’20	[179]

C. Sampling from Output Space

Numerous systems of sampling from output space rely on autoregressive sampling to generate music from a model. Autoregressive sampling can be combined with different types of NNs. In the early years, MLPs and RNNs were used to model the temporal relation of monophonic melodies. Given an input sequence, a model is trained to predict its succeeding sequences. The generation process repeats the procedure: predicting the probability distribution of the next value on the input sequence, sampling from the output probability distribution, and then appending the sampled value to the input. The initial input can be a null sequence, a random sequence, or a short music sequence as motif. Recently, researchers have paid much attention to enlarging the scale of neural networks for addressing complex problems. In the light of many successes in natural language processing (NLP), deep recurrent-based models have become popular in dealing with music-related tasks because both NLP and music emphasize modeling sequential and temporal relationship of data. Regarding the types of neurons, standard recurrent units hardly model long-term information. Alternatively, LSTM and GRU are shown to better capture both the short and long-term relationship in data. As for neural architectures, the transformer [223], which is a sequence model featuring the attention mechanism, achieves state-of-the-art performance in linguistic translation tasks and has also been modified for music generation [39]. Despite the transformer being an encoder-decoder model, it is considered to be a sampling-from-output-space system because the music generation relies on autoregressive sampling from the output probabilities of music notes, instead of mapping latent vectors to music pieces.

As one of the earliest attempts of RNN for music generation, Mozer [207] proposed modeling and generating melodies with an RNN using the autoregressive strategy. Eck and Schmidhuber [28], [209] considered using LSTM in music generation to improve the modeling of long-term relationship. Franklin [210] used a similar strategy for jazz music generation. Later on, Liu and Ramakrishnan [176] attempted

TABLE VI
NEURAL NETWORK SYSTEMS THAT SAMPLE THEIR PRODUCTS FROM OUTPUT SPACE.

Model	Task	Style	Condition	Representation	Dataset	Sub. Test	Year	Ref.
RNN	melody	Bach, folk	melody	psychoacoustics	JSB [190], manual	-	’94	[207]
RNN	melody	Bach, folk	melody	psychoacoustics	JSB [190], manual	-	’99	[208]
LSTM	melody, chord	blue	-	pianoroll	manual	-	’02	[209]
LSTM	melody, chord	blue	-	pianoroll	manual	-	’02	[28]
LSTM	chord prog.	jazz	-	psychoacoustics	manual	-	’06	[210]
TDNN	melody	classical, folk	-	ABC notation	manual	-	’08	[211]
LSTM	harmonization	Bach	prime	pianoroll	JSB [190]	-	’14	[176]
LSTM	general	Bach, classical	-	event, pianoroll	MuseData [191]	scoring	’16	[212]
LSTM	melody		prime	pianoroll	manual	-	’16	[213]
LSTM	melody, accomp.	pop	scale, melody	mixed	midi_man [214]	A/B test	’16	[215]
LSTM	chord prog.	western	melody	event	Wikifonia [5]	scoring	’17	[5]
LSTM	harmonization	Bach	notes, rhythm, cadence	event	JSB [190]	A/B test	’17	[36]
LSTM	piano	-	scale	event	e-piano [203]	-	’17	[216]
LSTM	general	classical	composer	pianoroll	piano-midi [192]	A/B test	’18	[37]
Transformer	chorales, piano	Bach, piano	prime	event	JSB [190], MAESTRO [206]	A/B test	’18	[39]
GRU	pop band	pop	chord prog.	pianoroll	manual	-	’18	[217]
LSTM	piano	classical	valence-arousal	pianoroll	piano-midi [192]	-	’19	[38]
LSTM	chord prog.	-	melody	lead sheet	Wikifonia [5]	-	’19	[6]
LSTM	melody	Persian	-	MIDI (vector)	manual	-	’19	[218]
LSTM	chord prog.	mixed	-	one-hot vector	NMD [188], McGill [219], Wikifonia [5]	-	’19	[220]
Transformer	guitar	classical	prime	event	manual	scoring	’20	[177]
GRU	pop band	pop	chord prog.	pianoroll	manual	-	’20	[221]
Transformer	piano	pop	-	event	manual	scoring	’20	[40]
GRU+Conv.	accomp.	counterpoint	melody	event	JSB [190]	A/B test	’20	[16]
Transformer	piano	pop	lead sheet	event set	manual	scoring	’21	[41]
Transformer	piano	-	priming	event	MAESTRO [206]	scoring	’21	[178]
Transformer	melody, lyrics	-	melody, lyrics	event	LMD† [181]	scoring	’21	[222]

to model four-part harmonization with LSTM. With training on Bach chorales, the LSTM model is able to continue a chorale given a fragment. Jaques et al. [213] designed an LSTM model that learns to predict the next note of a melody. Notably, the note-predicting model is further refined by deep Q-learning with a reward function based on music theory rules. The later studies tend to use more complex and deeper neural architectures. Chu et al. [215] proposed an LSTM-based hierarchical system that sequentially generates pitch and duration, chord, and percussion in an autoregressive manner. The hierarchical model consists of four two-layered LSTM models with 512-dimensional hidden states. DeepJ [37] is a deep RNN designed for modeling multiple genres and generating music with a specific or mixed genre. RL-Duet [16] presents a human-machine interactive actor-critic model that generates counterpoint accompaniment for a given melody. In addition to music generation, LSTM has been used in generation of chord progression and chord-to-note interpretation [220].

Recently, Huang et al. [39] employed the transformer to music generation task and proposed a music transformer model. They designed the relative self-attention in place of the standard attention mechanism for improving memory efficiency and inference time. Using autoregressive sampling, the music transformer can generate music from scratch (by feeding a null sequence), or continue a given motif. The music transformer can generate longer music pieces (few minutes) considering expressiveness and long-range coherence. Although the model considers long-term relationship at generation, its produced music is sometimes metrically inconsistent. Huang and Yang [40] addressed this issue through a modified event-based data representation that further encodes metrical information. Hsiao et al. [41] proposed aggregating multiple events into one token. The improved representations allow the model to generate locally and globally structured music. Nevertheless, a few studies attempted to incorporate transformer with adversarial training and pretraining techniques. Zhang [177] applied an encoder-only transformer as the discriminator to analyze the local and global quality of a generated sequence. Muhamed et al. [178] introduced the SpanBERT pretraining technique [224] to strengthen the transformer-based discriminator. Sheng et al. [222] proposed a song writing dual transformer system in which one transformer composes melody and the other writes lyrics. The MASS pretraining technique [225] is applied to both the melody and lyrics transformers to cope with the limited paired data issue.

The autoregressive sampling generally allows conditioning with a prime. In addition, Zhao et al. [38] introduced the 2-dimensional valence-arousal conditioning to LSTM for influencing the emotion of generated music. Zhu et al. proposed XiaoIce Band [217] and its extension [221], both of which are attention-based recurrent models that allow conditioning on chord progression for generating multitrack music in a pop music arrangement.

Gibbs sampling is another option for sampling music. Hadjeres et al. [36] presented the DeepBach system, a two-way LSTM architecture for generating Bach-like chorales. To sample music from the DeepBach model, the authors used a pseudo-Gibbs sampling that iteratively samples each time step of a voice, while the others are fixed. Ebrahimi et al. [218] further trained the DeepBach model on the Persian music dataset, of which the MIDI files are obtained by applying optical music recognition software on scanned music sheets. This data conversion framework shows a potential for partially solving the labor-intensive data processing problem.

V. Future Research Directions

As the above survey shows, CI techniques have been developed for music composition tasks in recent decades. In spite of the many successful designs and remarkable music works that have been achieved, several issues remain. We believe the challenges and opportunities for future research include five major directions: customization and interaction, instrumental performability and arrangement, music structure, evaluation metrics, and CI technologies. The following describes and discusses these directions in more detail.

A. Customization and Interaction

Automatic music composition is beneficial to real-world applications, such as generating royalty free music for videos, digital games, retail stores, and restaurants. Different genres of music may be required to fit the atmosphere of a specific scenario. The vibes in music are affected by several factors: key, mode, melody, chord, tempo, rhythm, instrumental arrangement, etc. In the evolutionary systems, music composition can be customized by allowing users to indicate some of the aforementioned properties of music [21]. In NN and DL systems, customization can be attained by modeling the conditional probability of music and sampling with the conditioning mechanisms. However, in the above-reviewed composition systems, the customizability is limited to one or few music factors or to certain forms of interaction [36], [215], [226]. These issues detract from the practical utility of the CI systems [227]. Extending the controllable factors can improve the customizability of automatic music composition; on the other hand, enabling more interaction forms facilitates more practical applications. Considering that music knowledge is required to set these factors, a possible solution to enhance user friendliness of the systems is to enable high-level control or fuzzy control of the music factors. Development of autofill or writing suggestion in music renders another direction for expediting AI-assisted music composition and for promoting human-machine interaction.

B. Instrumental Performability and Arrangement

Nowadays, assorted instruments are available to composers, while each instrument has its characteristics and limitations. Performability concerns whether a music composition is playable with the off-the-shelf real-world instruments by human musicians. A system lacking of performability awareness may generate a composition that is too hard to play or beyond the pitch range of an instrument. In addition, arrangement considers and makes use of the characteristics of instruments. For a system to be arrangement aware, it has to be performability aware of each instrument and must coordinate the instruments to exert their strengths.

Several early studies aim to compose monophonic melody for general purpose and thus neglects instruments. For the systems that compose four-part harmony, the arrangement is ordinarily based on a standard four-part choir[2]. In such a scenario, the role of each voice has been conventionally regulated with limited flexibility. Recently, some studies consider composition for a single instrument, such as a piano [40], [100], guitar [159], [177], and drum kit [7], [12], [87]. Some further investigate composing music for a multi-instrument band, including a pop band [174], [182], [215], [217], [221] and a rock band [31], [32], [147]. Despite that these studies focus on particular instruments, performability awareness is seldom considered and thus the resultant compositions might be unplayable. As for arrangement, it is observed that the coverage of the investigated instrumental arrangements corresponds closely to the coverage of the publicly accessible music corpuses. Therefore, contributing open and well-coded music datasets is a way to nourish this research topic. Some existent systems can compose multi-track music through indirect intertrack interaction based on a preset or condition. However, explicit intertrack interaction, e.g., intertrack melodic imitation and role exchange, receives less attention. To achieve explicit intertrack interaction, a composition system needs to ponder the instrumental characteristics and keep trace of the role of each track.

C. Music Structure

A considerable number of human-composed music works are organized in structure systematically; by contrast, only a few studies concern producing structural music in automatic composition. Music structure involves meter (e.g., the consistent periods of beats and bars), phrase, and form. Most of the human-composed music works are in static meters. Restated, a single work may include multiple meters but does not change the meter frequently. The steady progression of beats and bars intensifies the groove of the music. In evolutionary systems, this can be achieved by defining the mapping between genotypes and phenotypes based on user-defined or predefined meters. In NNs, modeling meter with the time-based data has been found to be non-trivial [39], [216]. Lack of perceptible meters causes the machine-generated music sounds random, unintentional, and even awkward. Modeling meter-based data is able to enforce steady meters in generated compositions [40].

On the other hand, music phrase and form are important roles in managing sensory recurrence in human-composed works. For instance, in classical music, sonate and sonatine follow the sonata form, comprising the exposition, development, and recapitulation. In addition, the similar melodies or rhythmic patterns often engage with multiple phrases. Recurrence of the motifs and thematic melodies further strengthens the audience's impression on the music. Several studies in EC impose music form templates and then compose each section individually. However, this method cannot link sections to a

[2]Four-part harmony can also be performed with other four-part instrumental arrangements (i.e., soprano, alto, tenor, and bass) or with just a single keyboard.

theme because they are composed without considering the context [136]. The early studies in NNs, by contrast, often suffer from the difficulty of modeling long-term dependency. Recent studies have shown significant improvement on long-term relation modeling with the aid of attention mechanism [39], advanced data representations [35], [40], [41], and self-similarity analysis [87]. A few NN studies attempted to improve music structure, such as by imposing a music form during sampling [185]. At present, the techniques for modeling structural information in music are still under development.

D. Evaluation Metrics

A common assessment method is to conduct a survey, i.e., a subjective test, to evaluate the generated music pieces according to the human responses collected. Its major drawback is the inefficiency and irreplicability of evaluation; for instance, it may require days or weeks to collect an acceptable number of responses. Another issue lies in the potential bias in the questionnaires. These issues may hinder the subjective tests from reflecting the genuine opinions of the majority. For some cases such as the studies on ethnic genres, soliciting qualified subjects can be even more difficult and expensive.

As for objective evaluation, rule-based EAs present the means of evaluating music quality by hand-crafted rules. These methods are genre-specific [20], [22], [150] and usually involve personal opinions, which may be hard to generalize to other cases. Plenty of NN studies adopt statistical analysis to evaluate the belongingness of generated music to the training corpus [55]. Specifically, Yang and Lerch [228] presented an open-source toolbox with a set of frequently used statistical features, including the pitch-based and rhythm-based statistics. These statistical features can measure, to certain extent, how well the model resembles the probability distribution of the training data. However, the features might fail to reflect human sensation because humans listen to music, instead of counting music. For example, the pitch-based features are unable to distinguish between various arpeggio forms of a chord although they sound quite different. The same issue also occurs in the vertically mirrored palindromic melodies (e.g., the C major scale steps of C4–C3–C4 and C3–C4–C3). On the other hand, the rhythm-based features are vulnerable to time stretching. A potential direction to address this issue is to promote the statistics from note level to pattern level. Designing such features can be music knowledge demanding and challenging, but the resulting evaluation metric aligned with human perception will be efficient and beneficial to automatic music composition as well as music information retrieval.

E. CI Technologies

There is ample room for improving the designs of CI technologies in music composition. First, as the above reviewed, most of the studies on EC-based music composition systems use simple GA or canonical EAs as the optimizer. For the EAs using interaction-based evaluation, the evaluation resources are considered rare and costly, which forms a computational expensive optimization problem in essence. Modern surrogate-assisted EAs (SAEAs) have shown to be effective in reducing evaluation requests [229], [230]; these approaches are promising for further alleviating human fatigue. Second, multiplicity and diversity in the generated compositions is another key consideration. In this regard, novelty search [231] may facilitate the EA-based systems composing music in batch rather than a single piece of music. Lastly, the models used in learning-based evaluation are relatively shallow and simple. An EA with DL-based evaluation is therefore promising for enhancing the evolutionary composition systems.

As for NN-based music composition, current studies and proposed approaches are usually restricted to the availability of massive training corpuses. This issue has impeded the NN systems to compose music for the music styles that have only small data, e.g., tribal music, folk music, and modern music[3]. Transfer learning has shown its effectiveness in learning with low resources. For example, a lot of research in computer vision uses the pretrained VGG or ResNet as the backbone model for low-resource problems [232], [233]. In linguistic tasks, transfer based on bidirectional encoder representations from transformers (BERT) [234] has been a widely accepted pretraining techniques to improve the model performance on the downstream tasks. Using transfer learning in music composition merits in-depth investigation but has only a few initiative studies exploring its potential [178], [197], [222].

VI. Conclusions

Computational intelligence has been broadly studied and applied in music composition tasks. In particular, various types of systems are proposed for music composition in recent years, including EAs, MAs, NNs, DL, and hybrid intelligence. In this survey, we comprehensively reviewed and discussed the studies on CI techniques for music composition from technical perspectives. Specifically, the first part introduces the data representations for music composition. The second part reviews the EC approaches and classifies them according to their fitness evaluation criteria, i.e., interactive evaluation, rule-based evaluation, and learning-based evaluation. The third part examines the NN and DL studies as per their music sampling strategies, i.e., sampling from input space, latent space, and output space. After the thematic reviews, the fourth part suggests the challenges and future research directions: customization and interaction, arrangement and performability, music structure, evaluation metrics, and CI technologies.

This survey provides a new taxonomy to classify music composing systems, delves into the CI techniques for music composition, and analyzes their strengths and weaknesses. The present reviews and discussions show that CI possesses high capability and great potentiality for music composition tasks; in addition, they reveal the limitations and issues of current research, such as a lack of robust quality and diversity assessment of the outcome music.

[3]A huge amount of modern music is available in audio form. However, their audio and symbolic content are mostly subject to copyright. Unauthorized use in data modeling is still a controversial issue.

Acknowledgment

The authors would like to thank Dr. Li Su for the early discussion on classification of sampling strategies.

References

[1] S. Nakashima, Y. Imamura, S. Ogawa, and M. Fukumoto, "Generation of appropriate user chord development based on interactive genetic algorithm," in *Proceedings of International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, 2010, pp. 450–453.

[2] B. Xu, S. Wang, and X. Li, "An emotional harmony generation system," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, 2010, pp. 1–7.

[3] A. Freitas and F. Guimaraes, "Melody harmonization in evolutionary music using multiobjective genetic algorithms," in *Proceedings of Sound and Music Computing Conference*, 2011.

[4] M. Fukumoto, "Creation of music chord progression suited for user's feelings based on interactive genetic algorithm," in *Proceedings of IIAI 3rd International Conference on Advanced Applied Informatics*, 2014, pp. 757–762.

[5] H. Lim, S. Rhyu, and K. Lee, "Chord generation from symbolic melody using BLSTM networks," in *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 621–627.

[6] W. Yang, P. Sun, Y. Zhang, and Y. Zhang, "CLSTMS: A combination of two LSTM models to generate chords accompaniment for symbolic melody," in *Proceedings of International Conference on High Performance Big Data and Intelligent Systems*, 2019, pp. 176–180.

[7] M. Dostál, "Genetic algorithms as a model of musical creativity–on generating of a human-like rhythmic accompaniment." *Computers and Artificial Intelligence*, vol. 24, pp. 321–340, 2005.

[8] T. Dimitrios and M. Eleni, "A dynamic GA-based rhythm generator," in *Trends in Intelligent Systems and Computer Engineering*. Springer, 2008, pp. 57–73.

[9] R. Yamamoto, S. Nakashima, S. Ogawa, and M. Fukumoto, "Proposal for automated creation of drum's fill-in pattern using interactive genetic algorithm," in *Proceedings of International Conference on Biometrics and Kansei Engineering*, 2011, pp. 238–241.

[10] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," in *Proceedings of International Conference on Machine Learning (ICML)*, 2018, pp. 4364–4373.

[11] N. Masuda and H. Iba, "Musical composition by interactive evolutionary computation and latent space modeling," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2018, pp. 2792–2797.

[12] A. R. Burton and T. Vladimirova, "Genetic algorithm utilising neural network fitness evaluation for musical composition," in *Proceedings of International Conference on Artificial Neural Nets and Genetic Algorithms*, 1998, pp. 219–223.

[13] N. Tokui and H. Iba, "Music composition with interactive evolutionary computation," in *Proceedings of 3rd International Conference on Generative Art*, vol. 17, no. 2, 2000, pp. 215–226.

[14] Y. Maeda and Y. Kajihara, "Rhythm generation method for automatic musical composition using genetic algorithm," in *Proceedings of IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2010, pp. 1–7.

[15] C.-H. Liu and C.-K. Ting, "Polyphonic accompaniment using genetic algorithm with music theory," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, 2012, pp. 1–7.

[16] N. Jiang, S. Jin, Z. Duan, and C. Zhang, "RL-Duet: Online music accompaniment generation using deep reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 710–718.

[17] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. Springer, 2003.

[18] C.-H. Liu and C.-K. Ting, "Computational intelligence in music composition: A survey," *IEEE Transactions on Emerging Topics in Computational Intelligence (TETCI)*, vol. 1, pp. 2–15, 2017.

[19] E. R. Miranda and J. A. Biles, *Evolutionary Computer Music*. Springer, 2007.

[20] C.-H. Liu and C.-K. Ting, "Fusing Flamenco and Argentine Tango by evolutionary composition," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, 2017, pp. 2645–2652.

[21] C.-H. Liu, "Automatic music composition using evolutionary algorithms," Ph.D. dissertation, National Chung Cheng University, 2017.

[22] F. F. de Vega, "Revisiting the 4-part harmonization problem with GAs: A critical review and proposals for improving," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, 2017, pp. 1271–1278.

[23] Y.-W. Wen, "Memetic computing for automatic music composition," Ph.D. dissertation, National Chung Cheng University, 2022.

[24] C. Sulyok, C. Harte, and Z. Bodó, "On the impact of domain-specific knowledge in evolutionary music composition," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 2019, pp. 188–197.

[25] A. V. Samson and A. D. Coronel, "Evolutionary algorithm-based composition of hybrid-genre melodies using selected feature sets," in *Proceedings of IEEE International Workshop on Computational Intelligence and Applications (IWCIA)*, 2016, pp. 51–56.

[26] M. Kikuchi and Y. Osana, "Automatic melody generation considering chord progression by genetic algorithm," in *Proceedings of 6th World Congress on Nature and Biologically Inspired Computing (NaBIC)*, 2014, pp. 190–195.

[27] M. Takano and Y. Osana, "Automatic composition system using genetic algorithm and N-gram model considering melody blocks," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, 2012, pp. 1–8.

[28] D. Eck and J. Schmidhuber, "A first look at music composition using LSTM recurrent neural networks," Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale, Tech. Rep., 2002.

[29] O. Mogren, "C-RNN-GAN: Continuous recurrent neural networks with adversarial training," in *NIPS Workshop on Constructive Machine Learning*, 2016.

[30] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang, "MidiNet: A convolutional generative adversarial network for symbolic-domain music generation," in *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 324–331.

[31] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018, pp. 34–41.

[32] H.-W. Dong and Y.-H. Yang, "Convolutional generative adversarial networks with binary neurons for polyphonic music generation," in *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 190–196.

[33] A. Roberts, J. Engel, and D. Eck, "Hierarchical variational autoencoders for music," in *NIPS Workshop on Machine Learning for Creativity and Design*, 2017.

[34] A. Roberts, J. Engel, S. Oore, and D. Eck, "Learning latent representations of music to generate interactive musical palettes," in *ACM Workshop on Intelligent Music Interfaces for Listening and Creation*, 2018.

[35] Z. Wang, Y. Zhang, Y. Zhang, J. Jiang, R. Yang, J. Zhao, and G. Xia, "Pianotree VAE: Structured representation learning for polyphonic music," in *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*, 2020, pp. 368–375.

[36] G. Hadjeres, F. Pachet, and F. Nielsen, "DeepBach: A steerable model for Bach chorales generation," in *Proceedings of International Conference on Machine Learning (ICML)*, 2017, pp. 1362–1371.

[37] H. H. Mao, T. Shin, and G. Cottrell, "DeepJ: Style-specific music generation," in *Proceedings of IEEE International Conference on Semantic Computing (ICSC)*, 2018, pp. 377–382.

[38] K. Zhao, S. Li, J. Cai, H. Wang, and J. Wang, "An emotional symbolic music generation system based on LSTM networks," in *Proceedings of IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 2019, pp. 2039–2043.

[39] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer: Generating music with long-term structure," arXiv preprint arXiv:1809.04281, 2018.

[40] Y.-S. Huang and Y.-H. Yang, "Pop music transformer: Generating music with rhythm and harmony," in *Proceedings of ACM Multimedia*, 2020, pp. 1180–1188.

[41] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, 2021, pp. 178–186.

[42] M. Suzuki and T. Onisawa, "Musical expression generation in several music genres by fuzzy rules," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2011, pp. 319–326.

[43] A. Moroni, J. Manzolli, F. von Zuben, and R. Gudwin, "Evolutionary computation applied to algorithmic composition," in *Proceedings of*

IEEE Congress on Evolutionary Computation (CEC), vol. 2, 1999, pp. 807–811.

[44] G. Acampora, J. M. Cadenas, R. D. Prisco, V. Loia, E. M. noz, and R. Zaccagnino, "A hybrid computational intelligence approach for automatic music composition," in *Proceedings of IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2011, pp. 202–209.

[45] E. Muñoz, J. M. Cadenas, Y.-S. Ong, and G. Acampora, "Memetic music composition," *IEEE Transactions on Evolutionary Computation (TEVC)*, vol. 20, pp. 1–15, 2016.

[46] D. Herremans, C.-H. Chuan, and E. Chew, "A functional taxonomy of music generation systems," *ACM Computing Surveys*, vol. 50, no. 5, p. 69, 2017.

[47] H. Järveläinen, "Algorithmic musical composition," in *Proceedings of Seminar on Content Creation Art*. University of Technology, Laboratory of Acoustics and Audio Signal, 2000.

[48] A. Alpern. (1995) Techniques for algorithmic composition of music. [Online]. Available: http://hamp.hampshire.edu/adaF92/algocomp/algocomp

[49] O. Lopez-Rincon, O. Starostenko, and G. A. Martín, "Algorithmic music composition based on artificial intelligence: Survey," in *Proceedings of International Conference on Electronics, Communications and Computers (CONIELECOMP)*, 2018, pp. 187–193.

[50] J. D. Fernández and F. Vico, "AI methods in algorithmic composition: a comprehensive survey," *Journal of Artificial Intelligence Research*, vol. 48, no. 1, pp. 513–582, 2013.

[51] G. Papadopoulos and G. Wiggins, "AI methods for algorithmic composition: A survey, a critical view and future prospects," in *Proceedings of AISB Symposium on Musical Creativity*, vol. 124. Edinburgh, UK, 1999, pp. 110–117.

[52] R. L. de Mantaras and J. L. Arcos, "AI and music from composition to expressive performance," *AI Magazine*, vol. 23, no. 3, pp. 43–57, 2002.

[53] J.-P. Briot, "From artificial neural networks to deep learning for music generation: History, concepts and trends," *Neural Computing and Applications*, vol. 33, no. 1, pp. 39–65, 2021.

[54] S. Ji, J. Luo, and X. Yang, "A comprehensive survey on deep music generation: multi-level representations, algorithms, evaluations, and future directions," arXiv preprint arXiv:2011.06801, 2020.

[55] J.-P. Briot, G. Hadjeres, and F.-D. Pachet, *Deep Learning Techniques for Music Generation*. Springer, 2020.

[56] ——, "Deep learning techniques for music generation–a survey," arXiv preprint arXiv:1709.01620, 2017.

[57] R. Loughran and M. O'Neill, "Evolutionary music: applying evolutionary computation to the art of creating music," *Genetic Programming and Evolvable Machines*, pp. 1–31, 2020.

[58] M. Dostál, "Evolutionary music composition," in *Handbook of Optimization: From Classical to Modern Approach*. Springer, 2013, pp. 935–964.

[59] E. R. Miranda, "At the crossroads of evolutionary computation and music: Self-programming synthesizers, swarm orchestras and the origins of melody," *Evolutionary Computation*, vol. 12, no. 2, pp. 137–158, 2004.

[60] A. Gartland-Jones and P. Copley, "The suitability of genetic algorithms for musical composition," *Contemporary Music Review*, vol. 22, no. 3, pp. 43–55, 2003.

[61] G. A. Wiggins, G. Papadopoulos, S. Phon-Amnuaisuk, and A. Tuson, *Evolutionary methods for musical composition*. University of Edinburgh, Department of Artificial Intelligence, 1998.

[62] P. Wiriyachaiporn, K. Chanasit, A. Suchato, P. Punyabukkana, and E. Chuangsuwanich, "Algorithmic music composition comparison," in *Proceedings of 5th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 2018, pp. 1–6.

[63] S. Phon-Amnuaisuk, A. Tuson, and G. Wiggins, "Evolving musical harmonisation," in *Proceedings of Artificial Neural Nets and Genetic Algorithms*. Springer, 1999, pp. 229–234.

[64] N. Carvalho and G. Bernardes, "Towards balanced tunes: a review of symbolic music representations and their hierarchical modeling." in *Proceedings of International Conference on Computational Creativity (ICCC)*, 2020, pp. 236–242.

[65] 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 *Proceedings of International Conference on Machine Learning (ICML)*, 2017, pp. 1068–1077.

[66] L. Gabrielli, C. E. Cella, F. Vesperini, D. Droghini, E. Principi, and S. Squartini, "Deep learning for timbre modification and transfer: An evaluation study," in *Proceedings of Audio Engineering Society Convention*, 2018.

[67] D. Matić, "A genetic algorithm for composing music," *Yugoslav Journal of Operations Research*, vol. 20, no. 1, pp. 157–177, 2010.

[68] G. Papadopoulos and G. Wiggins, "A genetic algorithm for the generation of jazz melodies," *Proceedings of the Finnish Conference on Artificial Intelligence (STeP)*, vol. 98, 1998.

[69] J. Gillick, A. Roberts, J. Engel, D. Eck, and D. Bamman, "Learning to groove with inverse sequence transformations," in *Proceedings of International Conference on Machine Learning (ICML)*, 2019, pp. 2269–2279.

[70] B. Manaris, P. Roos, P. Machado, D. Krehbiel, L. Pellicoro, and J. Romero, "A corpus-based hybrid approach to music analysis and composition," in *Proceedings of 22nd National Conference on Artificial Intelligence*, 2007, pp. 839–845.

[71] P. J. P. de León, J. M. Iñesta, J. Calvo-Zaragoza, and D. Rizo, "Data-based melody generation through multi-objective evolutionary computation," *Journal of Mathematics and Music*, vol. 10, no. 2, pp. 173–192, 2016.

[72] E. Rodríguez Pastor, "Automatic music composition by genetic programming," University of Alicante, Tech. Rep., 2019.

[73] L. Spector and A. Alpern, "Criticism, culture, and the automatic generation of artworks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 1994, pp. 3–8.

[74] ——, "Induction and recapitulation of deep musical structure," in *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 95, 1995, pp. 20–25.

[75] J. Polito, J. M. Daida, and T. F. Bersano-Begey, "Musica ex machina: Composing 16th-century counterpoint with genetic programming and symbiosis," in *Proceedings of 6th International Conference on Evolutionary Programming*, 1997, pp. 113–124.

[76] B. Johanson and R. Poli, "GP-music: An interactive genetic programming system for music generation with automated fitness raters," University of Birmingham, Tech. Rep., 1998.

[77] S. Phon-Amnuaisuk, E. H. Law, and H. C. Kuan, "Evolving music generation with SOM-fitness genetic programming," in *Proceedings of EvoWorkshops on Applications of Evolutionary Computing*, 2009, pp. 557–566.

[78] J. McDermott and U.-M. O'Reilly, "An executable graph representation for evolutionary generative music," in *Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO)*, 2011, pp. 403–410.

[79] C. Sulyok and C. Harte, "On virtual machine architectures for evolutionary music composition," in *Artificial Life Conference Proceedings*, 2017, pp. 577–584.

[80] C. Sulyok, "Genetic operators in evolutionary music composition," in *Proceedings of 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, 2018, pp. 253–259.

[81] A. O. de la Puente, R. S. Alfonso, and M. A. Moreno, "Automatic composition of music by means of grammatical evolution," in *Proceedings of Conference on APL: Array Processing Languages: Lore, Problems, and Applications*, 2002, pp. 148–155.

[82] J. Reddin, J. Mcdermott, and M. O'Neill, "Elevated pitch: Automated grammatical evolution of short compositions," in *Workshops on Applications of Evolutionary Computation*, vol. 5484, 2009, pp. 579–584.

[83] R. Loughran, J. McDermott, and M. O'Neill, "Tonality driven piano compositions with grammatical evolution," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, 2015, pp. 2168–2175.

[84] C.-C. J. Chen and R. Miikkulainen, "Creating melodies with evolving recurrent neural networks," in *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, vol. 3, 2001, pp. 2241–2246.

[85] I. D. Matić, A. P. Oliveira, and A. Cardoso, "Automatic melody generation using neural networks and cellular automata," in *Proceedings of 11th Symposium on Neural Network Applications in Electrical Engineering*, 2012, pp. 89–94.

[86] B. Wang and Y.-H. Yang, "PerformanceNet: Score-to-audio music generation with multi-band convolutional residual network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 1174–1181.

[87] I.-C. Wei, C.-W. Wu, and L. Su, "Generating structured drum pattern using variational autoencoder and self-similarity matrix," in *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*, 2019, pp. 847–854.

[88] J. Huang, J.-C. Wang, J. B. L. Smith, X. Song, and Y. Wang, "Modeling the compatibility of stem tracks to generate music mashups," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, 2021, pp. 187–195.

[89] T. M. Blackwell and P. Bentley, "Improvised music with swarms," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, vol. 2, 2002, pp. 1462–1467.

[90] M. A. Kaliakatsos-Papakostas, A. Floros, and M. N. Vrahatis, "Interactive music composition driven by feature evolution," in *SpringerPlus*, vol. 5, no. 1, 2016, pp. 1–38.

[91] M. Geis and M. Middendorf, "An ant colony optimizer for melody creation with baroque harmony," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, 2007, pp. 461–468.

[92] C.-Y. Chang and Y.-P. Chen, "AntsOMG: a framework aiming to automate creativity and intelligent behavior with a showcase on cantus firmus composition and style development," *Electronics*, vol. 9, no. 8, p. 1212, 2020.

[93] ——, "Contrapuntal composition and autonomous style development of organum motets by using AntsOMG," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2021, pp. 2023–2030.

[94] ——, "In the name of creativity: en route to inspiring machines," in *Proceedings of International Conference on Computational Creativity (ICCC)*, 2021, pp. 400–404.

[95] M. Fukumoto and S. Ogawa, "A proposal for optimization of sign sound using interactive differential evolution," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2011, pp. 1370–1375.

[96] M. Fukumoto, R. Yamamoto, and S. Ogawa, "The efficiency of interactive differential evolution in creation of sound contents: In comparison with interactive genetic algorithm," in *ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, 2012, pp. 25–30.

[97] ——, "The efficiency of interactive differential evolution in creation of sound contents: In comparison with interactive genetic algorithm," *International Journal of Software Innovation (IJSI)*, vol. 1, no. 2, pp. 16–27, 2013.

[98] Y.-S. Ong, M. H. Lim, and X.-S. Chen, "Memetic computation—Past, present & future," *IEEE Computational Intelligence Magazine (CIM)*, vol. 5, pp. 24–31, 2010.

[99] J. J. Fux, *The Study of Counterpoint: From Johann Joseph Fux's Gradus Ad Parnassum*. W. W. Norton & Company, 1965.

[100] J. Mańdziuk, M. Goss, and A. Woźniczko, "Chopin or not? A memetic approach to music composition," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, 2013, pp. 546–553.

[101] A. Moroni, J. Manzolli, and F. Von Zuben, "Composing with interactive genetic algorithms," in *Proceedings of Brazilian Symposium on Computer Music*, 1999.

[102] M. Unehara, K. Yamada, and T. Shimada, "Subjective evaluation of music with brain wave analysis for interactive music composition by IEC," in *Proceedings of Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS)*, 2014, pp. 66–70.

[103] D. Ralley, "Genetic algorithms as a tool for melodic development," in *Proceedings of International Computer Music Conference (ICMC)*, 1995, pp. 501–502.

[104] H. Takagi, "Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation," in *Proceedings of the IEEE*, vol. 89, 2001, pp. 1275–1296.

[105] J. A. Biles, "GenJam: A genetic algorithm for generating jazz solos," in *Proceedings of International Computer Music Conference (ICMC)*, vol. 94, 1994, pp. 131–137.

[106] T. Unemi and E. Nakada, "A tool for composing short music pieces by means of breeding," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC)*, vol. 5, 2001, pp. 3458–3463.

[107] T.-Y. Fu, T.-Y. Wu, C.-Y. Chen, K.-C. Wu, and Y.-P. Chen, "Evolutionary interactive music composition," in *Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO)*, 2006, pp. 1863–1864.

[108] Y.-P. Chen, "Interactive music composition with the CFE framework," *ACM SIGEVOlution*, vol. 2, no. 1, pp. 9–16, 2007.

[109] M. Unehara and T. Onisawa, "Interactive music composition system–composition of 16-bars musical work with a melody part and backing parts," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC)*, vol. 6, 2004, pp. 5736–5741.

[110] B. Jacob, "Composing with genetic algorithms," in *Proceedings of International Computer Music Conference (ICMC)*, 1995, pp. 452–455.

[111] J. A. Biles, P. G. Anderson, and L. Loggi, "Neural network fitness functions for a musical IGA," in *Proceedings of International ICSC Symposium on Intelligent Industrial Automation and Soft Computing*, 1996, pp. B39–B44.

[112] N. Gwee, "Complexity and heuristics in ruled-based algorithmic music composition," Ph.D. dissertation, Louisiana State University and Agricultural & Mechanical College, 2002.

[113] J. Yoon, "An interactive evolutionary system IFGAM," in *Proceedings of Annual Conference of IEEE Industrial Electronics Society (IECON)*, vol. 3, 2004, pp. 3166–3171.

[114] N. Fortier and M. Van Dyne, "A genetic algorithm approach to improve automated music composition," *International Journal of Computers*, vol. 5, pp. 525–532, 2011.

[115] J. A. Biles, "Interactive GenJam: Integrating real-time performance with a genetic algorithm," in *Proceedings of International Computer Music Conference (ICMC)*, 1998, pp. 232–235.

[116] ——, "Life with GenJam: Interacting with a musical IGA," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, vol. 3, 1999, pp. 652–656.

[117] ——, "GenJam: Evolution of a Jazz improviser," in *Creative Evolutionary Systems*. Morgan Kaufmann, 2002, pp. 165–187.

[118] T. Onisawa, W. Takizawa, and M. Unehara, "Composition of melody reflecting user's feeling," in *Proceedings of 26th Annual Conference of IEEE Industrial Electronics Society (IECON)*, vol. 3, 2000, pp. 1620–1625.

[119] M. Unehara and T. Onisawa, "Interactive music composition system," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC)*, vol. 6, 2002, p. 6.

[120] ——, "Music composition system with human evaluation as human centered system," *Soft Computing*, vol. 7, no. 3, pp. 167–178, 2003.

[121] ——, "Music composition by interaction between human and computer," *New Generation Computing*, vol. 23, no. 2, pp. 181–191, 2005.

[122] ——, "Composition of music using human evaluation," in *Proceedings of IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, vol. 3, 2001, pp. 1203–1206.

[123] T. Unemi and M. Senda, "A new musical tool for composition and play based on simulated breeding," in *Second Iteration - Emergence: Proceedings of the Second International Conference on Generative Systems in the Electronic Arts*. Citeseer, 2001, pp. 100–109.

[124] T. Unemi, "A design of genetic encoding for breeding short musical pieces," in *ALife VIII Workshop Proceedings*, 2002, pp. 25–30.

[125] R. Klinger and G. Rudolph, "Evolutionary composition of music with learned melody evaluation," in *Proceedings of 5th WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics*, 2006, pp. 234–239.

[126] M. Unehara and T. Onisawa, "Music composition system based on subjective evaluation," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC)*, vol. 1, 2003, pp. 980–986.

[127] H. Zhu, S. Wang, and Z. Wang, "Emotional music generation using interactive genetic algorithm," in *Proceedings of International Conference on Computer Science and Software Engineering*, vol. 1, 2008, pp. 345–348.

[128] E. Aldwell and A. Cadwallader, *Harmony & voice leading*. Cengage Learning, 2018.

[129] R. De Prisco, G. Zaccagnino, and R. Zaccagnino, "EvoComposer: an evolutionary algorithm for 4-voice music compositions," *Evolutionary Computation*, vol. 28, no. 3, pp. 489–530, 2020.

[130] ——, "EvoBassComposer: a multi-objective genetic algorithm for 4-voice compositions," in *Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO)*, 2010, pp. 817–818.

[131] R. A. McIntyre, "Bach in a box: The evolution of four part Baroque harmony using the genetic algorithm," in *Proceedings of IEEE Conference on Evolutionary Computation (CEC)*, 1994, pp. 852–857.

[132] S. Phon-Amnuaisuk and G. Wiggins, "The four-part harmonisation problem: A comparison between genetic algorithms and a rule-based system," in *Proceedings of AISB Symposium on Musical Creativity*, 1999, pp. 28–34.

[133] T. Maddox and J. Otten, "Using an evolutionary algorithm to generate four-part 18th century harmony," in *Proceedings of the WSEAS International Conference on Mathematics and Computers in Business and Economics*, 2000, pp. 83–89.

[134] P. Donnelly and J. Sheppard, "Evolving four-part harmony using genetic algorithms," in *Applications of Evolutionary Computation*, 2011, pp. 273–282.

[135] A. Garay Acevedo, "Fugue composition with counterpoint melody generation using genetic algorithms," in *Proceedings of Second International Conference on Computer Music Modeling and Retrieval*, 2004, pp. 96–106.

[136] C.-H. Liu and C.-K. Ting, "Evolutionary composition using music theory and charts," in *Proceedings of IEEE Symposium on Computational*

Intelligence for Creativity and Affective Computing (CICAC), 2013, pp. 63–70.

[137] ——, "Music pattern mining for chromosome representation in evolutionary composition," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, 2015, pp. 2145–2152.

[138] C.-W. Chang and H. C. Jiau, "An improved music representation method by using harmonic-based chord decision algorithm," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, vol. 1, 2004, pp. 615–618.

[139] A. Freitas, F. Guimarães, and R. Barbosa, "Automatic evaluation methods in evolutionary music: An example with Bossa melodies," in *International Conference on Parallel Problem Solving from Nature (PPSN)*, 2012, pp. 458–467.

[140] X. Zheng, D. Li, L. Wang, Y. Zhu, L. Shen, and Y. Gao, "Chinese folk music composition based on genetic algorithm," in *Proceedings of 3rd International Conference on Computational Intelligence & Communication Technology (CICT)*, 2017, pp. 1–6.

[141] Y.-W. Wen and C.-K. Ting, "Composing Bossa Nova by evolutionary computation," in *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–8.

[142] D. Tzimeas and E. Mangina, "Jazz Sebastian Bach: A GA system for music style modification," in *Proceedings of International Conference on Systems and Networks Communications (ICSNC)*, 2006, pp. 36–36.

[143] ——, "Dynamic techniques for genetic algorithm-based music systems," *Computer Music Journal*, vol. 33, pp. 45–60, 2009.

[144] M. Marques, V. Oliveira, S. Vieira, and A. C. Rosa, "Music composition using genetic evolutionary algorithms," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, vol. 1, 2000, pp. 714–719.

[145] R. Prisco and R. Zaccagnino, "An evolutionary music composer algorithm for bass harmonization," in *Proceedings of Workshops on Applications of Evolutionary Computation*, 2009, pp. 567–572.

[146] H. B. Lopes, F. V. C. Martins, R. T. N. Cardoso, and V. F. dos Santos, "Combining rules and proportions: A multiobjective approach to algorithmic composition," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, 2017, pp. 2282–2289.

[147] T. M. Oliwa, "Genetic algorithms and the abc music notation language for rock music composition," in *Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO)*, 2008, pp. 1603–1610.

[148] R. Ramirez, A. Hazan, E. Maestre, and X. Serra, "A genetic rule-based model of expressive performance for Jazz saxophone," *Computer Music Journal*, vol. 32, pp. 38–50, 2008.

[149] M. Jiang and C. Zhou, "Automated composition system based on GA," in *Proceedings of IEEE International Conference on Intelligent Systems and Knowledge Engineering*, 2010, pp. 380–383.

[150] F. V. Vargas, J. A. Fuster, and C. B. Castañón, "Artificial musical pattern generation with genetic algorithms," in *Proceedings of Latin America Congress on Computational Intelligence (LA-CCI)*, 2015, pp. 1–5.

[151] C. Chang and Y. Chen, "Fusing creative operations into evolutionary computation for composition: from a composer's perspective," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, 2019, pp. 2113–2120.

[152] M. Towsey, A. Brown, S. Wright, and J. Diederich, "Towards melodic extension using genetic algorithms," *Journal of Educational Technology & Society*, vol. 4, no. 2, pp. 54–65, 2001.

[153] E. Özcan and T. Erçal, "A genetic algorithm for generating improvised music," in *Proceedings of International Conference on Artificial Evolution (Evolution Artificielle)*, 2007, pp. 266–277.

[154] A. Calvo and J. Seitzer, "MELEC: Meta-level evolutionary composer," *Journal of Systemics, Cybernetics and Informatics*, vol. 9, no. 1, pp. 45–50, 2011.

[155] C.-L. Wu, C.-H. Liu, and C.-K. Ting, "A novel genetic algorithm considering measures and phrases for generating melody," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, 2014, pp. 2101–2107.

[156] C.-K. Ting, C.-L. Wu, and C.-H. Liu, "A novel automatic composition system using evolutionary algorithm and phrase imitation," *IEEE Systems Journal*, vol. 11, pp. 1284–1295, 2017.

[157] A. Horner and D. E. Goldberg, "Genetic algorithms and computer-assisted music composition." in *Proceedings of International Conference on Genetic Algorithm (ICGA)*, 1991, pp. 437–441.

[158] K. Ricanek, A. Homaifar, and G. Lebby, "Genetic algorithm composes music," in *Proceedings of Southeastern Symposium on System Theory*, 1993, pp. 223–227.

[159] D. R. Tuohy and W. D. Potter, "GA-based music arranging for guitar," in *Proceedings of IEEE International Conference on Evolutionary Computation (CEC)*, 2006, pp. 1065–1070.

[160] Y. Maeda and Y. Kajihara, "Automatic generation method of twelve tone row for musical composition used genetic algorithm," in *Proceedings of IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2009, pp. 963–968.

[161] B. Manaris, D. Hughes, and Y. Vassilandonakis, "Monterey Mirror: Combining markov models, genetic algorithms, and power laws," in *Proceedings of 1st Workshop in Evolutionary Music, 2011 IEEE Congress on Evolutionary Computation (CEC)*, 2011, pp. 33–40.

[162] P. M. Gibson and J. A. Byrne, "NEUROGEN, musical composition using genetic algorithms and cooperating neural networks," in *Proceedings of International Conference on Artificial Neural Networks (ICANN)*, 1991, pp. 309–313.

[163] H. Göksu, P. Pigg, and V. Dixit, "Music composition using genetic algorithms and multilayer perceptrons," in *Proceedings of 1st International Conference on Advances in Natural Computation*, 2005, pp. 1242–1250.

[164] P. Sheikholharam and M. Teshnehlab, "Music composition using combination of genetic algorithms and recurrent neural networks," in *Proceedings of 8th International Conference on Hybrid Intelligent Systems*, 2008, pp. 350–355.

[165] H. A. Batista, L. F. W. Goes, C. França, and W. C. A. Batista, "Evolutionary music composition for digital games using regent-dependent creativity metric," in *Proceedings of 16th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, 2017, pp. 88–95.

[166] I. A. Doush and A. Sawalha, "Automatic music composition using genetic algorithm and artificial neural networks," *Malaysian Journal of Computer Science*, vol. 33, no. 1, pp. 35–51, 2020.

[167] A. Ç. Yiiksel, M. M. Karci, and A. Ş. Uyar, "Automatic music generation using evolutionary algorithms and neural networks," in *Proceedings of International Symposium on Innovations in Intelligent Systems and Applications*, 2011, pp. 354–358.

[168] M. Alfonseca, M. Cebrian, and A. Ortega, "A simple genetic algorithm for music generation by means of algorithmic information theory," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, 2007, pp. 3035–3042.

[169] M. Tomari, M. Sato, and Y. Osana, "Automatic composition system based on genetic algorithm and N-gram model," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2008, pp. 202–207.

[170] P. Sheikholharam and M. Teshnehlab, "Music composition using combination of genetic algorithms and Kohonen grammar," in *Proceedings of International Symposium on Computational Intelligence and Design*, vol. 1, 2008, pp. 255–260.

[171] K. Fujii, M. Takeshita, and Y. Osana, "Automatic composition system using genetic algorithm and N-gram model–influence of N in N-gram model," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2011, pp. 1388–1393.

[172] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[173] C. Hernandez-Olivan and J. R. Beltran, "Music composition with deep learning: A review," arXiv preprint arXiv:2108.12290, 2021.

[174] H.-M. Liu and Y.-H. Yang, "Lead sheet generation and arrangement by conditional generative adversarial network," in *Proceedings of IEEE International Conference on Machine Learning and Applications*, 2018, pp. 722–727.

[175] S. Li, S. Jang, and Y. Sung, "Automatic melody composition using enhanced GAN," *Mathematics*, vol. 7, p. 883, 2019.

[176] I.-T. Liu and B. Ramakrishnan, "Bach in 2014: Music composition with recurrent neural network," arXiv preprint arXiv:1412.3191, 2014.

[177] N. Zhang, "Learning adversarial transformer for symbolic music generation," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–10, 2020.

[178] A. Muhamed, L. Li, X. Shi, S. Yaddanapudi, W. Chi, D. Jackson, R. Suresh, Z. C. Lipton, and A. J. Smola, "Symbolic music generation with Transformer-GANs," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, 2021, pp. 408–417.

[179] K. Choi, C. Hawthorne, I. Simon, M. Dinculescu, and J. Engel, "Encoding musical style with transformer autoencoders," in *Proceedings of International Conference on Machine Learning (ICML)*, 2020, pp. 1899–1908.

[180] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[181] Y. Yu, A. Srivastava, and S. Canales, "Conditional LSTM-GAN for melody generation from lyrics," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 17, no. 1, 2021.

[182] H.-M. Liu, M.-H. Wu, and Y.-H. Yang, "Lead sheet generation and arrangement via a hybrid generative model," in *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 23–27.

[183] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription," in *Proceedings of International Conference on Machine Learning (ICML)*, 2012, pp. 1881–1888.

[184] K. Goel, R. Vohra, and J. K. Sahoo, "Polyphonic music generation by modeling temporal dependencies using a RNN-DBN," in *Lecture Notes in Computer Science*. Springer, 2014, vol. 8681, pp. 217–224.

[185] S. Lattner, M. Grachten, and G. Widmer, "Imposing higher-level structure in polyphonic music generation using convolutional restricted Boltzmann machines and constraints," *Journal of Creative Music Systems*, vol. 2, pp. 1–31, 2018.

[186] B. Royal, K. Hua, and B. Zhang, "Deep composer: Deep neural hashing and retrieval approach to automatic music generation," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, 2020, pp. 1–6.

[187] A. E. Coca, D. C. Corrêa, and L. Zhao, "Computer-aided music composition with LSTM neural network and chaotic inspiration," in *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, 2013, pp. 1–7.

[188] Nottingham database. [Online]. Available: http://ifdo.ca/~seymour/nottingham/nottingham.html

[189] TheoryTab Database. [Online]. Available: https://www.hooktheory.com/theorytab

[190] M. Allan and C. K. I. Williams, "Harmonising chorales by probabilistic inference," *Advances in Neural Information Processing Systems*, vol. 17, pp. 25–32, 2005.

[191] Musedata. [Online]. Available: http://www.musedata.com/

[192] The piano-midi.de dataset. [Online]. Available: http://piano-midi.de/

[193] R. Sabathé, E. Coutinho, and B. Schuller, "Deep recurrent music writer: Memory-enhanced variational autoencoder-based musical score composition and an objective measure," in *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 3467–3474.

[194] C. Raffel, "Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching," Ph.D. dissertation, Columbia University, 2016.

[195] I. Simon, A. Roberts, C. Raffel, J. Engel, C. Hawthorne, and D. Eck, "Learning a latent space of multitrack measures," in *NIPS Workshop on Machine Learning and Creativity*, 2018.

[196] E. S. Koh, S. Dubnov, and D. Wright, "Rethinking recurrent latent variable model for music composition," in *Proceedings of IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2018, pp. 1–6.

[197] H.-T. Hung, C.-Y. Wang, Y.-H. Yang, and H.-M. Wang, "Improving automatic jazz melody generation by transfer learning techniques," in *Proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2019, pp. 339–346.

[198] A. Pati and A. Lerch, "Latent space regularization for explicit control of musical attributes," in *ICML Workshop on Machine Learning for Music Discovery*, 2019.

[199] R. Guo, I. Simpson, T. Magnusson, C. Kiefer, and D. Herremans, "A variational autoencoder for music generation controlled by tonal tension," in *Proceedings of Joint Conference on AI Music Creativity*, 2020.

[200] Z. Wang, K. Chen, J. Jiang, Y. Zhang, M. Xu, S. Dai, X. Gu, and G. Xia, "POP909: A pop-song dataset for music arrangement generation," in *Proceedings of International Conference on Music Information Retrieval (ISMIR)*, 2020, pp. 38–45.

[201] Z. Wang, D. Wang, Y. Zhang, and G. Xia, "Learning interpretable representation for controllable polyphonic music generation," in *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*, 2020, pp. 662–669.

[202] Y.-Q. Lim, C.-S. Chan, and F.-Y. Loo, "Style-conditioned music generation," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, 2020, pp. 1–6.

[203] International Piano-e-Competition. [Online]. Available: http://www.piano-e-competition.com/

[204] L. N. Ferreira and J. Whitehead, "Learning to generate music with sentiment," in *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*, 2019.

[205] H. H. Tan and D. Herremans, "Music FaderNets: controllable music generation based on high-level features via low-level feature modelling," in *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*, 2020, pp. 109–116.

[206] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the MAESTRO dataset," arXiv preprint arXiv:1810.12247, 2018.

[207] M. C. Mozer, "Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multi-scale processing," *Connection Science*, vol. 6, no. 2-3, pp. 247–280, 1994.

[208] ——, "Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multi-scale processing," *Musical Networks: Parallel Distributed Perception and Performance*, p. 227, 1999.

[209] D. Eck and J. Schmidhuber, "Finding temporal structure in music: Blues improvisation with LSTM recurrent networks," in *Proceedings of IEEE Workshop on Neural Networks for Signal Processing*, 2002, pp. 747–756.

[210] J. A. Franklin, "Recurrent neural networks for music computation," *INFORMS Journal on Computing*, vol. 18, no. 3, pp. 321–338, 2006.

[211] T. Oliwa and M. Wagner, "Composing music with neural networks and probabilistic finite-state machines," in *Proceedings of Workshops on Applications of Evolutionary Computing*, 2008, pp. 503–508.

[212] A. Huang and R. Wu, "Deep learning for music," arXiv preprint arXiv:1606.04930, 2016.

[213] N. Jaques, S. Gu, R. E. Turner, and D. Eck, "Generating music by fine-tuning recurrent neural networks with reinforcement learning," in *NIPS Workshop on Deep Reinforcement Learning*, 2016.

[214] The midi_man dataset. [Online]. Available: https://www.reddit.com/r/WeAreTheMusicMakers/comments/3ajwe4/the_largest_midi_collection_on_the_internet/

[215] H. Chu, R. Urtasun, and S. Fidler, "Song from PI: A musically plausible network for pop music generation," arXiv preprint arXiv:1611.03477, 2016.

[216] S. Oore, I. Simon, S. Dieleman, and D. Eck, "Learning to create piano performances," in *NIPS Workshop on Machine Learning and Creativity*, 2017.

[217] H. Zhu, Q. Liu, N. J. Yuan, C. Qin, J. Li, K. Zhang, G. Zhou, F. Wei, Y. Xu, and E. Chen, "XiaoIce Band: A melody and arrangement generation framework for pop music," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2837–2846.

[218] M. Ebrahimi, B. Majidi, and M. Eshghi, "Procedural composition of traditional Persian music using deep neural networks," in *Proceedings of 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*, 2019, pp. 521–525.

[219] J. B. L. Smith, J. A. Burgoyne, I. Fujinaga, D. De Roure, and J. S. Downie, "Design and creation of a large-scale database of structural annotations," in *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*, 2011, pp. 555–560.

[220] H.-H. Tan, "ChordAL: a chord-based approach for music generation using Bi-LSTMs," in *Proceedings of International Conference on Computational Creativity (ICCC)*, 2019, pp. 364–365.

[221] H. Zhu, Q. Liu, N. J. Yuan, K. Zhang, G. Zhou, and E. Chen, "Pop music generation: From melody to multi-style arrangement," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 14, no. 5, pp. 1–31, 2020.

[222] Z. Sheng, K. Song, X. Tan, Y. Ren, W. Ye, S. Zhang, and T. Qin, "SongMASS: automatic song writing with pre-training and alignment constraint," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 15, 2021, pp. 13 798–13 805.

[223] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 5998–6008.

[224] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "SpanBERT: improving pre-training by representing and predicting spans," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 64–77, 2020.

[225] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, "MASS: masked sequence to sequence pre-training for language generation," in *Proceedings of International Conference on Machine Learning (ICML)*, 2019, pp. 5926–5936.

[226] N. Meade, N. Barreyre, S. C. Lowe, and S. Oore, "Exploring conditioning for generative music systems with human-interpretable controls," in *Proceedings of International Conference on Computational Creativity (ICCC)*, 2019, pp. 148–155.
[227] 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," in *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*, 2020, pp. 708–716.
[228] L.-C. Yang and A. Lerch, "On the evaluation of generative models in music," *Neural Computing and Applications*, vol. 32, no. 9, pp. 4773–4784, 2020.
[229] B. Liu, Q. Zhang, and G. G. E. Gielen, "A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Transactions on Evolutionary Computation (TEVC)*, vol. 18, no. 2, pp. 180–192, 2014.
[230] L. Pan, C. He, Y. Tian, H. Wang, X. Zhang, and Y. Jin, "A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization," *IEEE Transactions on Evolutionary Computation (TEVC)*, vol. 23, no. 1, pp. 74–88, 2019.
[231] J. Lehman and K. O. Stanley, "Evolving a diversity of virtual creatures through novelty search and local competition," in *Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO)*, 2011, pp. 211–218.
[232] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, vol. 27, 2014, pp. 3320–3328.
[233] Y.-W. Wen, S.-H. Peng, and C.-K. Ting, "Two-stage evolutionary neural architecture search for transfer learning," *IEEE Transactions on Evolutionary Computation (TEVC)*, vol. 25, no. 5, pp. 928–940, 2021.
[234] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

Yu-Wei Wen (Member, IEEE) received the B.S. degree and the Ph.D. degree in computer science and information engineering from National Chung Cheng University, Taiwan, in 2015 and 2022, respectively.

He is currently an Assistant Professor of Department of Artificial Intelligence, Tamkang University, Taiwan. His research interests include evolutionary computation, memetic algorithm, machine learning, and computer music composition.

Chuan-Kang Ting (Senior Member, IEEE) received the B.S. degree from National Chiao Tung University, Taiwan, in 1994, the M.S. degree from National Tsing Hua University, Taiwan, in 1996, and the Dr. rer. nat. degree in Computer Science from Paderborn University, Germany, in 2005.

He is currently a Professor and the Chair of Department of Power Mechanical Engineering, National Tsing Hua University, Taiwan. His research interests include evolutionary computation, computational intelligence, memetic computing, machine learning, and their applications in manufacturing, ethics, music and arts.

Dr. Ting is the Editor-in-Chief of *IEEE Computational Intelligence Magazine*, the Editor-in-Chief of *Memetic Computing*, an Associate Editor of IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, and an Editorial Board Member of *Soft Computing*. He served as the IEEE Computational Intelligence Society (CIS) Newsletter Editor, the IEEE CIS Webmaster, the Chair of IEEE CIS Chapters Committee, and the Chair of IEEE CIS Creative Intelligence Task Force. He is an Executive Board Member of Taiwanese Association for Artificial Intelligence.