Musicality-Novelty Generative Adversarial Nets for Algorithmic Composition

Gong Chen
Department of Computing
The Hong Kong Polytechnic University
csgchen@comp.polyu.edu.hk

Yan Liu
Department of Computing
The Hong Kong Polytechnic University
csyliu@comp.polyu.edu.hk

Sheng-hua Zhong
College of Computer Science and Software Engineering
Shenzhen University
csshzhong@szu.edu.cn

Xiang Zhang
Department of Computing
The Hong Kong Polytechnic University
csxgzhang@comp.polyu.edu.hk

ABSTRACT

Algorithmic composition, which enables computer to generate music like human composers, has lasting charm because it intends to approximate artistic creation, most mysterious part of human intelligence. To deliver both melodious and refreshing music, this paper proposes the Musicality-Novelty Generative Adversarial Nets for algorithmic composition. With the same generator, two adversarial nets alternately optimize the musicality and novelty of the machine-composed music. A new model called novelty game is presented to maximize the minimal distance between the machine-composed music sample and any human-composed music sample in the novelty space, where all well-known human composed music products are far from each other. We implement the proposed framework using three supervised CNNs with one for generator, one for musicality critic and one for novelty critic on the time-pitch feature space. Specifically, the novelty critic is implemented by Siamese neural networks with temporal alignment using dynamic time warping. We provide empirical validations by generating the music samples under various scenarios.

CCS CONCEPTS

• **Applied computing** → *Sound and music computing*;

KEYWORDS

music; algorithmic composition; generative adversarial nets

ACM Reference Format:
Gong Chen, Yan Liu, Sheng-hua Zhong, and Xiang Zhang. 2018. Musicality-Novelty Generative Adversarial Nets for Algorithmic Composition. In *MM '18: 2018 ACM Multimedia Conference, Oct. 22–26, 2018, Seoul, Republic of Korea.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3240508.3240604

MM '18, October 22–26, 2018, Seoul, Republic of Korea

ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240604

1 INTRODUCTION

Music is an artistic form of auditory communication incorporating instrumental or vocal tones in a structured and continuous manner [29]. As a classical research topic in multimedia, the study of music has shown both great academic and commercial value. For example, music emotion analysis is a typical interdisciplinary research problem using the knowledge and techniques from multimedia, data mining, musicology, and psychology [2, 21, 43]. The research outcomes from music recommendation has shown large commercial potentials [23].

Algorithmic composition, designing the computational models to generate the music like human composer [34], attracts researchers from many different areas for centuries. Moreover, recent enormous advance in artificial intelligence makes algorithmic composition a topic of great concern again [10, 16, 19]. The magic of algorithmic composition lies in that it targets to reproduce the real intelligence of human beings, rather than a sophisticated imitation. Obviously, it is very challenging to build the artistic creativity of machines. Although researchers have made a lot of effort for many years, algorithmic composition still stays in the research labs while other artificial intelligence techniques, such as search agents and fingerprint recognition, have been widely used in the industry.

A new trend in algorithmic composition is using Generative Adversarial Nets (GANs) [13], which cast generative modeling as a game between two competing networks: a generator network produces synthetic data and a discriminator network discriminates between the generator's output and true data. GANs can generate visually appealing samples, but are often hard to train. To address this problem, Arjovsky et al. propose the Wasserstein GAN algorithm with a novel discriminator called critic, which improves the stability of learning, get rid of problems like mode collapse, and provide meaningful learning curves useful for debugging and hyperparameter searching [3]. Most recently, Gulrajani et al. propose to improve the training of Wasserstein GAN via gradient penalty [14].

It is no accident that GANs-based methods are studied in this paper because of irreversibility of musical rules in composition. Music is an artistic combination of artificial sound, hence there is no absolute standard to identify a correct music or incorrect one. It mainly depends on the subjective experience of audiences to indicate if the composition is successful. And even if we find the

complete rules from human-composed music, such as the composition theory, we can only conclude that those rules are used by some individual music works. But we cannot generate the music directly from these rules. Simply speaking, rules don't make the music; it is the music that makes the rules [37]. The advantage of using GANs in algorithmic composition is that the model targets to make the machine-composed music undifferentiated from the human-composed music by iteratively updates the weight spaces of generator and discriminator via an adversary approach.

Some existing works have validated the effectiveness of GANs in algorithmic composition [10, 44]. Unlike most existing works that focus on the imitation of musicality, i.e., the property of sounding like music [29], this paper emphasizes the imitation of artistic creation, i.e., the novelty of the generated sample. Unfortunately, existing GANs-based algorithmic composition cannot output the product with enough novelty naturally. Think of an extreme example. Suppose that the GANs generate n music samples by training m human-composed samples while n is smaller than m. If n generated samples are exactly the same with n real samples in the training data set, GANs will converge perfectly. However, for artistic creation like music composition, this perfect result means that the generated music is actually a copy of existing work. Although the randomness of GANs may avoid this extreme situation, the novelty in artistic creation using GANs relies more on luck.

To address this problem, we propose a new framework named Musicality-Novelty Generative Adversarial Nets (GANs), which optimize the musicality and novelty of machine-composed music alternately via two generative adversarial nets. Similar with existing works, the first generative adversarial nets aim at good musicality. The generator in the musicality game is trained to minimize the divergence between the distribution of machine-composed music and the distribution of human-composed music. The discriminator in the musicality game, which is called musicality critic, is trained to distinguish between the generated music samples and human-composed music samples.

The second generative adversarial nets aim to guarantee the novelty of the generated music, which is the main contribution of this paper. Novelty measure is very challenging due to the difficulty to define the novelty. People with different backgrounds hold different viewpoints, and so far, there is no widely accepted criteria of novelty in artistic creation. Moreover, it is also very difficult to quantify novelty. Tracking the history, inheritance and absorption promote the prosperous of art while the copyright of artists is violated by plagiarism and replication, which impede the development of music. Hence, how to distinguish different degrees of reference is a controversial topic in the music field. Inspired by the rationale of GANs, this paper proposes a new model to improve the novelty of music samples by utilizing the adversarial relationship of GANs based on the assumption that all well-known human-composed music samples have enough novelty. This is a reasonable assumption, which fortunately makes the problem of novelty modeling much easier. Unlike the musicality critic for the musicality game, the input to the novelty critic for novelty game is a set of pair-wise music samples and the output is a set of novelty labels. We propose a new critical model to construct the novelty space, which guarantees that all the human-composed music samples are far from each other there. The generator of the novelty game is shared with the musicality game, adjusted by a new objective function based on a simple idea that a generated music with good novelty is not similar to any existing human composed music works. Therefore, in the novelty game, the generator is trained to generate the music which maximizes the minimal distance to human-composed music in the novelty space.

In summary, by training the GANs for musicality and novelty simultaneously, the computer generates the music, which brings refreshing experience with the common characters similar to some well-known human-composed music samples. The rest of the paper is organized as follows. In section 2, we review the history of algorithmic composition.

In section 3, we propose a general framework of Musicality-Novelty Generative Adversarial Nets and its application in algorithmic composition. Section 4 provides the empirical validation of the proposed model. The paper is closed with the conclusion and the discussion of the future work.

2 RELATED WORK

In this part, we brief the important works of algorithmic composition in time order. The best-known pioneering work of modern algorithmic composition is the string quartet Illiac Suite by Hiller and Isaacson in 1958 [17]. Markov chain is used to generate notes in their composition system. Based on this seminal work, many studies focus on the Markov probability transitions and achieve some encouraging results. But the melodic quality of these works is not good enough and most of them do not consider the expressiveness and emotional content in music. Different to the Markov chain-based methods, rule-based composition method is proposed by Koenig [22] to generate music with better structure. Besides, Rader [36] proposes a rule-based artificial intelligence program to generate melody and harmony. The rules used in this program describes how notes or chords can be put together. Artificial intelligence techniques are used to determine the applicability of these melody and chord-generation rules, and the likelihood of application of an applicable rule.

Marvin Minsky, a superstar in artificial intelligence, hints that it is a possible approach to generate music using intelligent agents [30]. Using searching agent, Cope's Experiments in Musical Intelligence (EMI) program successfully composes music in the styles of many famous composers such as Chopin, Bach, and Rachmaninoff [9]. With the advances in machine learning algorithms and computer hardware, many studies focus on the learning-based methods. Early attempts using learning-based method are reported in [39], where monophonic melodies are encoded in pitch and duration and a Recurrent Neural Network (RNN) is trained to predict upcoming events. Bharucha trains a neural network to model the musical harmony. Eck and Schmidhuber trained a Long Short-Term Memory (LSTM) network jointly on a single chord sequence along with several different melodies [11]. Besides neural networks, learning methods such as random field have also been used to model the polyphonic music [25]. Some other studies focus on musical structure generation. Hörnel and Degenhardt uses a neural network to learn and reproduce higher-level structure in melodic sequences. Markov chains and evolutionary algorithms are also used to generate repetition structure [18]. Another study proposes to generate repetitive

harmonic sequences by a formal representation for the semiotic structure of chord sequences and a learned statistical model for ranking the instances of a semiotic pattern [8].

In recent years, neural network models become deeper. Boulanger-Lewandowski et al. proposes the RNN Restricted Boltzmann Machine (RNN-RBM) model for polyphonic music generation [5]. To create accurate yet flexible music models, Lyu et al. proposes the LSTM Recurrent Temporal RBM (LSTM-RTRBM) method that achieves significant performance improvement [28]. To improve the efficiency and efficacy of LSTM, Liu et al. [27] propose to use resilient propagation (RPROP) instead of standard back propagation to train the RNN in music composition. A generative model of Convolutional Restricted Boltzmann Machine (C-RBM) is proposed in [24] to impose higher-level structure of the generated polyphonic music. To generate natural-sounding music conforming to music theory, some tenets from music theory are encoded as filters in [42]. A hierarchical model that incorporates knowledge from music theory is proposed in [7] to generate multi-track pop music. This work shows some interesting applications such as neural dancing and neural story singing. Using reinforcement learning, a novel RNN-based method is proposed in [19] for improving the structure and quality of the generated sequences, while maintaining information originally learned from data as well as sample diversity. This work demonstrates good performance in generating musically pleasing melodies. Another interesting work [20] uses raw audio, instead of MIDI file, to train LSTM networks, which requires less manual effort for data representation. The industry also makes some of their deep learning composition projects open source [12, 15], which provides practical tools for the research in this field.

Triggered by the boom of GANs in image and video generation [35, 38, 41]. Some researchers have tried using GANs in algorithmic composition. The SeqGAN proposed in [44] enables the framework of GANs to generate sequences tokens such as monophonic music. Using SeqGAN, [26] generates musically coherent sequences and reports that careful tuning of reinforcement learning signals is crucial for music generation. C-RNN-GAN proposed in [32] trains a sequential model with continuous data to generate classical music. Different to most deep learning music generation methods which use the architecture of RNNs, the MidiNet in [42] uses Convolutional Neural Networks (CNNs) as the generator and discriminator, and designs a conditional CNN to "look back" without a recurrent unit. Most recently, Dong et al. [10] proposes the MuseGAN. Different from most of previous works, MuseGAN can generate symbolic multi-track music. In this composition system, three different models, namely, jamming model, composer model, and hybrid model, are designed in this work for different compositional approaches. Two methods are also provided to model the temporal structure in music. The music pieces generated by these GAN-based methods show good progress in many different prospects, such as chord formation, overall structure, and melodiousness. The impressive performance evidences the potential of GANs in music creation and encourages our further exploration of algorithmic composition.

3 PROPOSED METHOD

The general framework of Musicality-Novelty Generative Adversarial Nets is shown in Figure 1. The proposed framework utilize

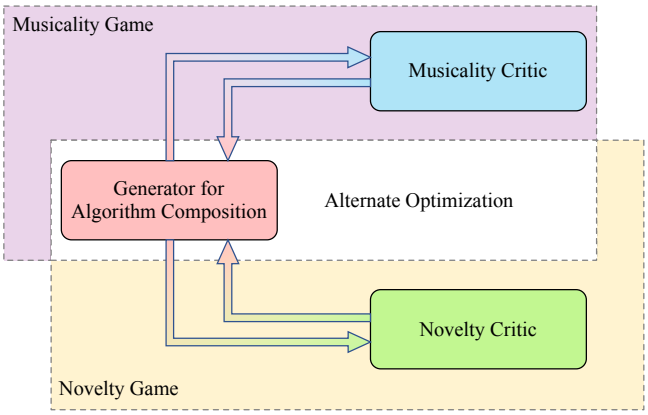

Figure 1: The general framework of the proposed Musicality-Novelty Generative Adversarial Nets.

two games to optimize the musicality and the novelty alternately by sharing the same generator. In the musicality game, we simultaneously train two models: a generator and a musicality critic. The generator captures the real music distribution and outputs the generated music instances. The musicality critic estimates the distance between the generator's distribution and the real music distribution. In the novelty game, a generator and a novelty critic are also trained simultaneously. The generator captures the machine-composed music that is far from any human-composed music in the novelty space, while the novelty critic seeks the novelty space where all human-composed music samples are far from each other.

From the global view, the machine-composed samples and human-composed samples should have similar distributions, while from the local view, a machine-composed music should guarantee enough distance to the nearest human-composed neighbor. With this framework, we expect to generate music with both good musicality and good novelty.

3.1 Musicality Game

Let $\mathbf{x} = (x_1, ..., x_T)$ be a music instance, where T denotes the number of time steps in the instance. We denote the generator as $G_\theta(\mathbf{z})$, where G is a differentiable function represented by a multilayer perceptron with parameters θ. The input $\mathbf{z}$ to the generator is sampled from some simple noise distribution implicitly defined by $\mathbf{z} \sim p_\mathbf{z}(\mathbf{z})$. In addition, we define a second multilayer perceptron, the musicality critic, as $D_w(\mathbf{x})$ with parameter w. We denote the generator's distribution as $\mathbb{P}_g$ and the real music distribution as $\mathbb{P}_r$. To learn the generator's distribution $\mathbb{P}_g$ over the real music, G and D play the following two-player minimax game:

$$\min_\theta \max_w \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathbb{P}_r} \big(D_w(\mathbf{x})\big) - \mathop{\mathbb{E}}_{\tilde{\mathbf{x}}\sim\mathbb{P}_g} \big(D_w(\tilde{\mathbf{x}})\big), \tag{1}$$

where $\tilde{\mathbf{x}} = G_\theta(\mathbf{z})$ denotes the generated music instances. Under an optimal musicality critic, minimizing the objective function in Eq. (1) with respect to θ minimizes the Wasserstein distance between $\mathbb{P}_g$ and $\mathbb{P}_r$ [14]. In this paper, the game in Eq. (1) is named musicality game.

3.2 Novelty Game

We begin with the definition of the pairwise novelty:

Definition 1 *Pairwise novelty*: Given two music instances $\mathbf{a}$ and $\mathbf{b}$, the pairwise novelty is a function $H(\mathbf{a}, \mathbf{b})$ representing the degree of dissimilarity between $\mathbf{a}$ and $\mathbf{b}$.

Based on the pairwise novelty, we propose an assumption:

Assumption 1 Given a set of real music instances $\{\mathbf{x}_n\}_{n=1}^N$ and a generated music instance $\tilde{\mathbf{x}}$, the higher the pairwise novelty $H(\tilde{\mathbf{x}}, \mathbf{x}_n)$ is for all $n = 1, ..., N$, the higher the novelty of $\tilde{\mathbf{x}}$ is.

With this assumption, we design a novelty critic to model the pairwise novelty. Similarly to the generator and musicality critic, the novelty critic is represented by a third multilayer perceptron. Based on the pairwise novelty, the novelty critic and the generator optimize their parameters via a maximin game.

We denote the novelty critic as H_υ with pairwise input and scalar output. Similarly to the generator and musicality critic, the novelty critic is represented by a third multilayer perceptron with parameters υ. We train H_υ to be an ideal adversary of the generator, which maximize the pairwise novelty between real music instances, and minimize the pairwise novelty between generated and real music instances. We simultaneously train the generator G_θ to maximize the the pairwise novelty between generated and real ones. In other words, H_υ and G_θ play the following two-player maximin game:

$$\max_\theta \min_\upsilon \inf_{\tilde{\mathbf{x}} \sim \mathbb{P}_g, \mathbf{x} \sim \mathbb{P}_r} \big(H_\upsilon(\tilde{\mathbf{x}}, \mathbf{x})\big) - \inf_{\mathbf{x}_1, \mathbf{x}_2 \sim \mathbb{P}_r} \big(H_\upsilon(\mathbf{x}_1, \mathbf{x}_2)\big). \quad (2)$$

Note that we use the infimum rather than expectation as our optimization objective, because a large average pairwise novelty would still tolerate small individual pairwise novelty, but we aim to generate music which is novel compared with any one of the real music instances.

3.3 Algorithm

The Musicality-Novelty Generative Adversarial Nets are presented in Algorithm 1. The musicality game to optimize Eq. (1) is stated from line 2 to line 12. The novelty game to optimize Eq. (2) is stated from line 13 to line 23.

Algorithm 1 Musicality-Novelty Generative Adversarial Nets

Require: α_D and α_H, the learning rates in musicality game and novelty game, respectively. Similarly, c_D and c_H, the clipping parameters. m_D and m_H, the batch sizes. U_D and U_H, the numbers of iterations.
Require: θ_0, initial generator parameters. w_0, initial musicality critic parameters. υ_0, initial novelty critic parameters.
1: **while** θ has not converged **do**
2: /* Musicality Game */
3: **for** $u_D = 1, ..., U_D$ **do**
4: Sample a batch $\{\mathbf{x}_n\}_{n=1}^{m_D} \sim \mathbb{P}_r$ from the real music instances.
5: Sample a batch $\{\mathbf{z}_n\}_{n=1}^{m_D} \sim p_\mathbf{z}(\mathbf{z})$ from a noise distribution.
6: $g_w \leftarrow \nabla_w[\frac{1}{m_D}\sum_{n=1}^{m_D} D_w(\mathbf{x}_n) - \frac{1}{m_D}\sum_{n=1}^{m_D} D_w(G_\theta(\mathbf{z}_n))]$
7: $w \leftarrow w + \alpha_D \cdot \mathrm{RMSProp}(w, g_w)$
8: $w \leftarrow \mathrm{clip}(w, -c_D, c_D)$
9: **end for**
10: Sample a batch $\{\mathbf{z}_n\}_{n=1}^{m_D} \sim p_\mathbf{z}(\mathbf{z})$ from a noise distribution.
11: $g_\theta \leftarrow -\nabla_\theta \frac{1}{m_D}\sum_{n=1}^{m_D} D_w(G_\theta(\mathbf{z}_n))$
12: $\theta \leftarrow \theta - \alpha_D \cdot \mathrm{RMSProp}(\theta, g_\theta)$
13: /* Novelty Game */
14: **for** $u_H = 1, ..., U_H$ **do**
15: Sample a batch $\{\mathbf{x}_n\}_{n=1}^{m_H} \sim \mathbb{P}_r$ from the real music instances.
16: Sample a batch $\{\mathbf{z}_n\}_{n=1}^{m_H} \sim p_\mathbf{z}(\mathbf{z})$ from a noise distribution.
17: $g_\upsilon \leftarrow \nabla_\upsilon[\inf\{H_\upsilon(\tilde{\mathbf{x}}_{n_1}, \mathbf{x}_{n_2})\} - \inf\{H_\upsilon(\mathbf{x}_{n_3}, \mathbf{x}_{n_4})\}]$, where $n_1, n_2, n_3, n_4 \in \{1, ..., m_H\}$ and $n_3 \neq n_4$
18: $\upsilon \leftarrow \upsilon - \alpha_H \cdot \mathrm{RMSProp}(\upsilon, g_\upsilon)$
19: $\upsilon \leftarrow \mathrm{clip}(\upsilon, -c_H, c_H)$
20: **end for**
21: Sample a batch $\{\mathbf{z}_n\}_{n=1}^{m_H} \sim p_\mathbf{z}(\mathbf{z})$ from a noise distribution.
22: $g_\theta \leftarrow \nabla_\theta \inf\{H_\upsilon(\tilde{\mathbf{x}}_{n_1}, \mathbf{x}_{n_2})\}$, where $n_1, n_2 \in \{1, ..., m_H\}$
23: $\theta \leftarrow \theta + \alpha_H \cdot \mathrm{RMSProp}(\theta, g_\theta)$
24: **end while**

3.4 Implementation

This paper proposes a general framework of MNGANs, which can support various implementations. In this part, we present our implementation in detail and brief some other possible implementations.

3.4.1 Feature Space. Referencing the latest work in algorithmic composition [10], we utilize the third-order tensor to represent the music sequence for both the generator and the critic. The first order is the time axis, the second order indicates the pitch, and the third order represents bars. The consecutive data points have half overlap in the temporal axis. It is fine to utilize other feature space, such as pitch sequence as the input for the generator and the critics.

3.4.2 Learning Method. We can utilize either supervised manner [13] or reinforcement manner [44] for the learning of GANs. Followed the classical GANs [13], supervised manner is used in this paper. For the musicality game, human composed music works are defined as the positive examples. For the novelty game, a pairwise human composed music is defined as one positive example. Reinforcement manner and the learning combining both the supervised manner and reinforcement manner are also promising, which have been listed as our future work.

3.4.3 Generator. Under the proposed framework, the generator can be implemented by any neural network based method, no matter in the shallow structure or deep one. Consistent with the feature space of second-order tensor structure, this paper utilizes the CNNs as the generator. No doubt, RNNs [44] and long-short term memory (LSTM) [11], are also good choices for the proposed framework.

3.4.4 Critic. The musicality critic holds the similar setting with the existing works of GANs-based algorithmic composition. The

novelty critic is a little different because the input is the pairwise data from two instances. A naive way to form the pairwise data is to stack two instances $\mathbf{a}$ and $\mathbf{b}$ up into a tensor with higher order, namely, $input = \{\mathbf{a}^T, \mathbf{b}^T\}^T$. Besides, we recommend to use siamese neural networks [6] to learn the pairwise novelty. Both of the siamese RNNs [33] and siamese CNNs [40] are designed specifically for pairwise data and have shown good performance.

3.4.5 Temporal Alignment. For a better performance, temporal alignment for pairwise music samples is considered. In this paper, we use the Dynamic Time Warping (DTW) method to process the music instance pair before feeding it to the novelty critic. Let $\mathbf{a}$ and $\mathbf{b}$ be any two different music instances from real music or generated music:

$$\mathbf{a}, \mathbf{b} \in \mathcal{X}_m \cup \widetilde{\mathcal{X}}_m, \mathbf{a} \neq \mathbf{b}, \tag{3}$$

where $\mathcal{X}_m = \{\mathbf{x}\}_{n=1}^m \subseteq \{\mathbf{x}_n\}_{n=1}^N$ is a subset of real music data with cardinal number m, and $\widetilde{\mathcal{X}}_m = \{\tilde{\mathbf{x}}\}_{n=1}^m$ is the set of generated music with cardinal number m. $\mathcal{X}_m$ and $\widetilde{\mathcal{X}}_m$ represent the batch data used in one training iteration, and m represents the batch size.

For two time steps t_1 and t_2, we consider a differentiable substitution-cost function $\delta(\mathbf{a}_{t_1}, \mathbf{b}_{t_2})$, which will be the quadratic Euclidean distance between two vectors in most cases. Then, we can define the cost matrix $\Delta(\mathbf{a}, \mathbf{b}) \in \mathbb{R}^{T \times T}$ as follows:

$$\Delta(\mathbf{a}, \mathbf{b}) = [\delta(\mathbf{a}_{t_1}, \mathbf{b}_{t_2})]_{(t_1, t_2)}. \tag{4}$$

Besides, we write $\mathcal{A}_{T,T} \subset \{0,1\}^{T \times T}$ for the set of alignment matrices, which represents the paths on a $T \times T$ matrix that connect the upper-left $(1,1)$ matrix entry to the lower-right (T,T) one using only $\downarrow, \rightarrow, \searrow$ moves. The optimal alignment matrix $\mathbf{A}^*$ is given by solving the Dynamic Time Warping (DTW) problem as follows:

$$\min_{\mathbf{A} \in \mathcal{A}_{T,T}} \langle \mathbf{A}, \Delta(\mathbf{a}, \mathbf{b}) \rangle, \tag{5}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. The problem in Eq. (5) can be solved using the Bellman's recursion [4]. Let $\mathbf{p}^* = (p_1, ..., p_l, ..., p_L)$ be the L-step path determined by the optimal alignment matrix $\mathbf{A}^*$, where the index pair $p_l = (i_l, j_l), i_l \leq T, j_l \leq T$ represents that step l of the optimal path is at row i_l and column j_l. We can write the aligned music instances $\mathbf{a}'$ and $\mathbf{b}'$ as follows:

$$\begin{aligned} \mathbf{a}' &= (a_{i_1}, ..., a_{i_l}, ..., a_{i_L}), \\ \mathbf{b}' &= (b_{i_1}, ..., b_{i_l}, ..., b_{i_L}). \end{aligned} \tag{6}$$

The alignment in Eq. (6) can be represented by a two-layer perceptron mapping T neurons into L neurons. For each $t = i_l, t = 1, ..., T, l = 1, ..., L$, we use a synapse with fixed weight 1 to connect the neuron t of input layer and neuron l of output layer. Thus, in backpropagation, the gradient can be backpropagated to the generator. However, because this two-layer perceptron is alignment operation in nature, rather than a learning machine, we do not update these fixed weights.

3.4.6 Applications. The proposed framework can generate the music in different levels and various scenarios. The simplest case is to generate a short motif from random noise or a piece of humming, which can inspire the human composers to write a complete work. Given the motif, it is also applicable to generate its variations, canon, and fugue by the proposed framework. A much more complex case is to generate polyphonic music with multiple tracks directly if corresponding training data has been fed [10]. In the experiment part, we demonstrate the empirical validation by generating the music with the length of several bars, several phrases, and several paragraphs. Both the monophonic and polyphonic music are generated.

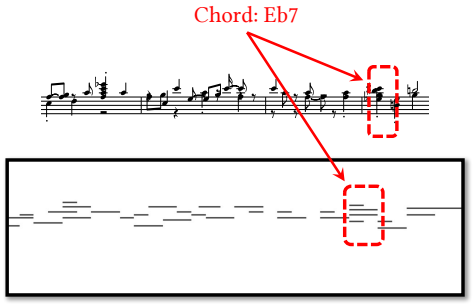

Figure 2: The chord in the sample generated by MNGANs.

4 EXPERIMENTS

We conduct three experiments to validate the proposed MNGANs. In the first experiment, we demonstrate the musicality of the music generated by our proposed model. The second experiment validates the novelty of the generated music samples by a well-designed recall task. The last experiment compares the performance of the proposed model with some representative algorithmic composition methods.

4.1 Musicality

4.1.1 Setting. We train the Musicality-Novelty Generative Adversarial Nets on the piano-roll dataset Lakh MIDI dataset. Following the data reprocessing operation in [10], we use 50,266 music samples of four bars as training data, and implement the generators as CNNs at two level, namely, temporal structure generator with two convolutional layers and bar generator with six convolutional layers. The input of the generator is random vector with length 128. The output of the generator is samples of four bars. The musicality critic is also implemented as CNNs with six convolutional layers. We train the model using Adam optimizer. For the novelty critic, we use convolutional Siamese networks [40] to learn the pairwise novelty. Following the best setting in [40], we use two convolutional layers in the novelty critic.

4.1.2 Results. Figure 2 to Figure 4 shows the scores and piano-rolls of three generated samples. There are some interesting observations about musicality in these examples. First, chords are generated in

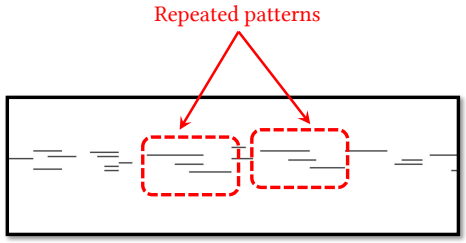

Figure 3: The repeated pattern in the sample generated by MNGANs.

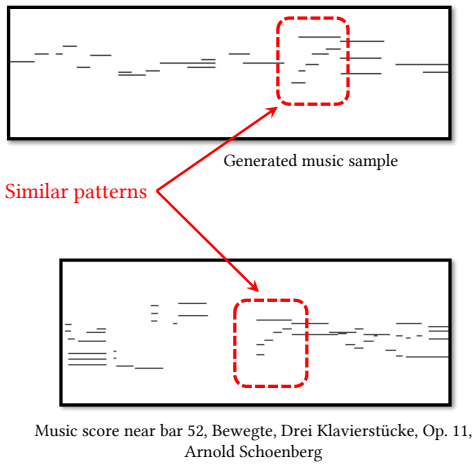

Figure 4: The pattern generated by MNGANs similar with atonal music composed by Arnold Schoenberg.

some samples (see Figure 2). A chord is a combination of different pitches played simultaneously. Only the pitch combinations consistent with rules about intervals in the music theory can be formed as a chord. Hence, the generation of chords suggests the generator has learned the rules of musicality via adversarial training. Second, some repeated patterns are observed (see Figure 3). By repetition with appropriate variations, simple motives can be formed into complex and informative music pieces. The occurrence of the repeated patterns implies that the structural information is generated. Third, no obvious tonality is found. This could result from the inconsistency of the training dataset, which contains music pieces with many different tonalities. However, we report that such multiple tonalities training data may be a good "teacher" of atonality. For example, in Figure 4 the generated sample even shows a pattern very similar with the work Drei Klavierstucke of Arnold Schoenberg, the musician famous for atonal compositions. Besides, following many previous works, we perform an expert validation with three professional musicians on the listening experience of five randomly selected pieces we generate. Almost no dissonance is found in the pieces.

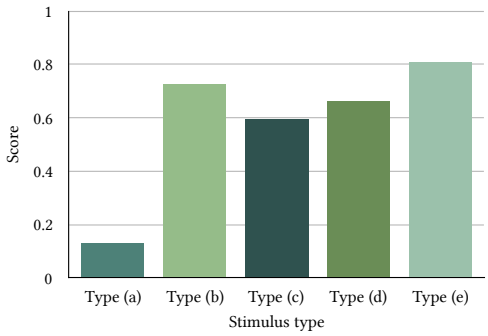

Figure 5: The samples generated by Musicality-Novelty Generative Adversarial Nets. Type (a) is the repeated human-composed music samples. Type (b) is the new human-composed music samples. Type (c) is the new part of human-composed music. Type (d) is the machine-composed music generated only by the musicality game. Type (e) is the machine-composed music generated by both musicality game and novelty game.

4.2 Novelty

To demonstrate the novelty of our generated samples, we design a human experiment paradigm with a recall task.

4.2.1 Subject. Fifty-two undergraduate students take part in our study. Using a self-report questionnaire, 47 of them indicate no professional musical training and they are remained as the subjects. None of the subjects reports neurological or hearing dysfunctions. None of the subjects has any prior knowledge of our study. Besides, none of them reports that he/she knows the stimuli used in our study.

4.2.2 Stimuli. The experiment contains four sessions. Each session contains two blocks. In the first block, eight human-composed music samples are played to the subjects. In the second block, eight music samples in five types are played :

Type (a): four from the eight human-composed music samples used in the first block;

Type (b): one new human-composed music sample;

Type (c): one human-composed sample from a new part in an original music piece corresponding to one of the other four samples except (a) used in the first block;

Type (d): one sample generated only by the musicality game;

Type (e): one sample generated by the proposed Musicality-Novelty Genarative Adversarial nets with both musicality and novelty game.

The length of all samples played in this experiment is four bars.

4.2.3 Procedure. The subjects are seated in a comfortable chair in a sound-shielded room. A session for practice is provided before the four main sessions. For each session, in the first block, subjects only need to listen to the music samples. The first block contains eight trials and each trial is followed by an inter-trial period, which contains 5 seconds silence, 5 seconds white noise, and 5 seconds silence. The inter-trial period is provided to minimize contamination from the previous stimuli. A 60 seconds break is provided between two blocks. The second block also contains eight trials. In the second block, subjects are required to recall whether the current music sample has been played in the first block by questionnaires. Accordingly, the inter-trial period of the second block contains 5 seconds silence for the judgement, 5 seconds silence, 5 seconds white noise, and 5 seconds silence. The music samples are played in the tempo of 120 BPM, which is a typical setting [31]. In each block, the music samples are played in a random order. The samples used in different sessions are randomly selected without replacement.

4.2.4 Results. We calculate the novelty score of each music sample in the second session in a simple way:

$$\text{novelty}(k) = \frac{\sum_{s=1}^{S} \text{judgement}(s,k)}{S}, \tag{7}$$

where judgement(s,k) equals to 1 if the subject s makes the judgement that the music sample k has never been played, and 0 in other case. S=47 denotes the number of subjects.

Figure 5 shows the results. The average novelty score of type (a) is the lowest, which indicates the subjects can well detect the old music samples. Type (b) gets a relatively high score. Which indicates that the subjects can also detect the novel music samples. Type (c) gets an average score around 0.5, which implies the variance between different temporal samples within the same piece of music can impact subjects' judgement, while such an effect is limited. The average performance of type (d) is also good, but still not sufficient compared with type (b). Type (e) achieves best performance, which indicates that our proposed MNGANs can generate music samples with good novelty.

4.3 Statistical Comparison

In this part, we compare the experience of the audiences when they listen to the music samples generated by different ways.

4.3.1 Setting. The 47 subjects in subsection 4.2 participate this performance evaluation experiment after finishing the recall task. The stimuli are the music samples generated by five representative methods ranging from classical algorithmic composition methods such as EMI to our new proposed MNGANs. Human-composed music samples and pure random note sequence are also employed as baselines. All of the samples used here are different with those in subsection 4.2. We clip the samples into 20 seconds segments, and play them to the subjects in a random order. Subjects rate the appreciation experience for each music sample in a 7-point Likert scale during the 20 seconds inter-trial periods.

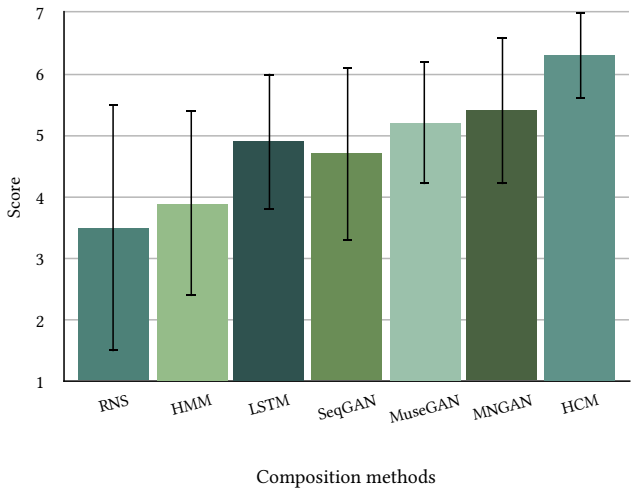

Figure 6: The experience of the audiences when they listen to the music samples generated by different ways. Music samples are composed by seven composition methods including: Random Note Sequence (RNS), Hidden Markov Model (HMM), Long Short-Term Memory (LSTM), Sequence Generative Adversarial Nets (SeqGAN), Multi-track sequential Generative Adversarial Network (MuseGAN), Musicality-Novelty Generative Adversarial Net (MNGAN), Human-Composed Music (HCM). 7-point Likert scale [1] is used in the subjective rating. High score indicates better audience experience.

4.3.2 Results. Figure 6 shows the average scores and standard deviations of audience experience. Three professional musicians validate the results. Random note sequence has the lowest score while human-compose music has the highest score. The large standard deviation caused by listening to random note sequence indicates a significant inter-subject difference. The audiences have similar experience when listening to the human-composed music. Compared with algorithmic composition using Hidden Markov Model, neural network-based algorithms achieve better performance in general. Moreover, MuseGAN and MNGAN outperform other methods. The proposed MNGAN gets the highest score while the MuseGAN performs more stably.

5 CONCLUSION AND FUTURE WORK

This paper proposes a new algorithmic composition model using Musicality-Novelty Genarative Adversarial Nets. By iteratively optimizing musicality game and novelty game, the machine-generated music evokes the artistic and refreshing experience of the audiences. A set of well-designed experiments validate the performance of the proposed model. As the general framework, the proposed model can support various implementations although this paper only provides one of them. The future work will be explored from the following aspects:

- The musicality and the novelty of the computer composed music are jointly optimized by sharing the same generator, which is updated in musicality game and novelty game in turn. The further work will be explored to trying different ways for joint optimization.
- In this paper, the generator, the musicality critic and the novelty critic use the same feature space although it is not necessary. The further work will be explored to try different feature spaces.
- For music generation, CNNs are used because the feature space is second order tensor. With different feature spaces, different neural networks will be explored, such as RNNs and LSTM.
- This is the first work that models novelty of the music using GANs. We will keep exploring other models to study the nature of artistic creation and to simulate the generation of different types of music.
- Supervised manner has been selected as learning method in this paper. In recent year, reinforcement learning has demonstrated very good performance in many applications, especially for the tasks that need long-term learning. Reinforcement manner and the learning method including both supervised manner and reinforcement manner will be explored.
- Theoretically speaking, the proposed framework can support the generation of music for different lengths. However, in the practical usage of the proposed model for empirical validation, we have noticed that it is difficult to learn both the local structures and global structure of the music if the music is longer than several minutes. This is the common challenges faced by many researchers in algorithmic composition. We will study the possibility to solve this problem by exploring hierarchical GANs.
- The proposed framework can support both monophonic and polyphonic music generation. The changes of arrangements will bring very different experiences of music appreciation even if the main melody is the same. The future work will focus on the study of the relationship between monophonic music generation and polyphonic music generation under the proposed framework.
- We validate the effectiveness of the proposed framework in typical experimental settings. It would be interesting to try some other settings in future work. For example, we will use different tempo such as 60 BPM, 100 BPM, and 180 BPM, besides 120 BPM, for the same piece of music.

ACKNOWLEDGMENTS

The author would like to thank the reviewers for their constructive comments. The work is supported by The Hong Kong Polytechnic University under Grant No.: PolyU 152101/14E and Grant No.: G-UAEU.

REFERENCES

[1] I Elaine Allen and Christopher A Seaman. 2007. Likert scales and data analyses. *Quality progress* 40, 7 (2007), 64–65.
[2] Eckart O Altenmüller. 2001. How many music centers are in the brain? *Annals of the New York Academy of Sciences* 930, 1 (2001), 273–280.
[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875* (2017).
[4] Richard Bellman. 1952. On the theory of dynamic programming. *Proceedings of the National Academy of Sciences* 38, 8 (1952), 716–719.
[5] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. 2012. Modeling temporal dependencies in high-dimensional sequences: application to polyphonic music generation and transcription. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*. Omnipress, 1881–1888.
[6] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1994. Signature verification using a" siamese" time delay neural network. In *Advances in Neural Information Processing Systems*. 737–744.
[7] Hang Chu, Raquel Urtasun, and Sanja Fidler. 2017. Song From PI: A Musically Plausible Network for Pop Music Generation. In *workshop on International Conference on Learning Representations, 2017. ICLR workshop 2017.*
[8] Darrell Conklin. 2016. Chord sequence generation with semiotic patterns. *Journal of Mathematics and Music* 10, 2 (2016), 92–106.
[9] David Cope. 1987. Experiments in Musical Intelligence. http://artsites.ucsc.edu/faculty/cope/experiments.htm. (1987).
[10] Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. 2018. MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment. In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*.
[11] Douglas Eck and Juergen Schmidhuber. 2002. A first look at music composition using lstm recurrent neural networks. *Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale* 103 (2002).
[12] Adam Roberts Dan Abolafia Elliot Waite, Douglas Eck. 2016. Project Magenta. https://magenta.tensorflow.org/.
[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.
[14] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*. 5769–5779.
[15] Gaëtan Hadjeres and François Pachet. 2016. DeepBach: a Steerable Model for Bach chorales generation. *arXiv preprint arXiv:1612.01010* (2016).
[16] Dorien Herremans and Elaine Chew. 2017. MorpheuS: generating structured music with constrained patterns and tension. *IEEE Transactions on Affective Computing* (2017).
[17] Lejaren A Hiller Jr and Leonard M Isaacson. 1957. Musical composition with a high speed digital computer. In *Audio Engineering Society Convention 9*. Audio Engineering Society.
[18] Dominik Hörnel and Peter Degenhardt. 1997. A Neural Organist Improvising Baroque-Style Melodic Variations. (1997).
[19] Natasha Jaques, Shixiang Gu, Dzmitry Bahdanau, José Miguel Hernández-Lobato, Richard E Turner, and Douglas Eck. 2016. Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control. *arXiv preprint arXiv:1611.02796* (2016).
[20] Vasanth Kalingeri and Srikanth Grandhe. 2016. Music generation with deep learning. *arXiv preprint arXiv:1612.04928* (2016).
[21] Youngmoo E Kim, Erik M Schmidt, Raymond Migneco, Brandon G Morton, Patrick Richardson, Jeffrey Scott, Jacquelin A Speck, and Douglas Turnbull. 2010. Music emotion recognition: A state of the art review. In *Proc. ISMIR*. Citeseer, 255–266.
[22] G Koenig. 1993. Aesthetische Praxis/Texte zur Music, Band 3, 1968-1991. *Quellentexte zur Musik im* 20 (1993).
[23] Noam Koenigstein, Gideon Dror, and Yehuda Koren. 2011. Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy. In *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 165–172.
[24] Stefan Lattner, Maarten Grachten, and Gerhard Widmer. 2016. Imposing higher-level Structure in Polyphonic Music Generation using Convolutional Restricted Boltzmann Machines and Constraints. *arXiv preprint arXiv:1612.04742* (2016).
[25] Victor Lavrenko and Jeremy Pickens. 2003. Polyphonic music modeling with random fields. In *Proceedings of the eleventh ACM international conference on Multimedia*. ACM, 120–129.
[26] Sang-gil Lee, Uiwon Hwang, Seonwoo Min, and Sungroh Yoon. 2017. A seqgan for polyphonic music generation. *arXiv preprint arXiv:1710.11418* (2017).
[27] I Liu, Bhiksha Ramakrishnan, et al. 2014. Bach in 2014: Music composition with recurrent neural network. *arXiv preprint arXiv:1412.3191* (2014).

[28] Qi Lyu, Zhiyong Wu, and Jun Zhu. 2015. Polyphonic music modelling with LSTM-RTRBM. In *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 991–994.
[29] George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.
[30] Marvin Minsky. 1982. Music, mind, and meaning. In *Music, mind, and brain*. Springer, 1–19.
[31] Dirk Moelants. 2002. Preferred tempo reconsidered. In *Proceedings of the 7th international conference on music perception and cognition*, Vol. 2002. Sydney, 1–4.
[32] Olof Mogren. 2016. C-RNN-GAN: Continuous recurrent neural networks with adversarial training. *arXiv preprint arXiv:1611.09904* (2016).
[33] Jonas Mueller and Aditya Thyagarajan. 2016. Siamese Recurrent Architectures for Learning Sentence Similarity.. In *AAAI*. 2786–2792.
[34] Gerhard Nierhaus. 2009. *Algorithmic composition: paradigms of automated music generation*. Springer Science & Business Media.
[35] Kyle Olszewski, Zimo Li, Chao Yang, Yi Zhou, Ronald Yu, Zeng Huang, Sitao Xiang, Shunsuke Saito, Pushmeet Kohli, and Hao Li. 2017. Realistic dynamic facial textures from a single image using gans. In *IEEE International Conference on Computer Vision (ICCV)*. 5429–5438.
[36] Gary M Rader. 1992. A method for composing simple traditional music by computer. In *Machine models of music*. MIT Press, 243–260.
[37] Jordi Sabater, Josep Lluis Arcos, and R López de Mántaras. 1998. Using rules to support case-based reasoning for harmonizing melodies. In *Multimodal Reasoning: Papers from the 1998 AAAI Spring Symposium*. Citeseer, 147–151.
[38] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. 2017. Temporal generative adversarial nets with singular value clipping. In *IEEE International Conference on Computer Vision (ICCV)*. 2830–2839.
[39] Peter M Todd. 1989. A connectionist approach to algorithmic composition. *Computer Music Journal* 13, 4 (1989), 27–43.
[40] Jack Valmadre, Luca Bertinetto, João Henriques, Andrea Vedaldi, and Philip HS Torr. 2017. End-to-end representation learning for correlation filter based tracking. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 5000–5008.
[41] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*. 613–621.
[42] Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang. 2017. MidiNet: A convolutional generative adversarial network for symbolic-domain music generation. In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR?2017), Suzhou, China*.
[43] Yi-Hsuan Yang and Homer H Chen. 2012. Machine recognition of music emotion: A review. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3, 3 (2012), 40.
[44] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient.. In *AAAI*. 2852–2858.