International Journal for Housing Science and Its Applications
Publish August 3, 2025. Volume 46, Issue 3 Pages 910-919

https://doi.org/10.70517/ijhsa46368

Innovative Research on Algorithmic Model of Music Composition Based on Neural Network Optimization Algorithm

Xiaohu Du[1,*]
[1] Composition Department at Wuhan Conservatory of Music, Wuhan, Hubei, 430060, China
Corresponding authors: (e-mail: 40708472@qq.com).

Abstract This paper proposes a music composition model based on neural network optimization algorithm, which integrates genetic algorithm and improved BP neural network to realize the intelligence and efficiency of music composition. The problem of insufficient diversity of traditional methods is solved by a multidimensional coding strategy (12-bit octal, quadratic and octal coding for scales, registers and beats, respectively), combined with genetic operators to dynamically optimize melodic segments. For polyphonic music counterpoint vocal part generation, the elastic gradient descent method is introduced to improve the BP network, which effectively overcomes the defects of the traditional algorithm that is slow to converge and prone to trap local extremes. The experiments use LakhMIDI and MUT datasets, and compare with RNN, LSTM, Seq2Seq and other models, and the results show that the similarity between the generated music and the database waveforms reaches 86.74%, and the chord rule matching is significantly consistent. In the manual evaluation, the model is fully ahead in fluency of 4.23, consistency of 4.49, and rhythm of 4.57, with an average score of 4.34. The evaluation of the music theory features shows that the note repetition degree is 18.77% and the style matching degree is 91.48%, which are better than the benchmark model. The study shows that the model significantly improves the automation level and artistry of music generation by synergistically optimizing the coding strategy and network structure.

Index Terms neural network optimization algorithm, BP neural network, music composition, counterpoint vocal part generation

I. Introduction

In today's world, music as a form of art is more and more needed by people in their daily life. However, due to individual differences, people have their own standards and requirements for different types of music, and at the same time, music creation has a strong professionalism since its emergence, which requires the creator to have a deep insight into the knowledge of music theory [1]. It is because of this high threshold, non-professional personnel want to create a pleasant music is basically impossible. But with the development of Artificial Intelligence (AI) and Deep Learning, scientists are realizing that people can compose music through computer technology [2], [3]. This is a completely different path from composing music by musicians, and people who compose music through this method often do not need to have strong musical expertise, yet they can create satisfying musical works through computers, and this technique related to creating music using music creation algorithms is known as algorithmic composition [4], [5].

Algorithmic composition in general can be divided into two broad categories, one is to develop a series of rules for music composition through specific musical knowledge, and the other is to learn the relevant composition rules for music composition through machine learning, or deep learning algorithms [6]. In the early years, since the related technologies of machine learning and artificial intelligence have not yet emerged, as most researchers tend to use the first approach for algorithmic composition of music. For example, literature [7] derives a customized music knowledge rule set based on the analysis of genre musicology, which is used to automatically compose folk music in the style of the Galician Sota genre. Literature [8] trains an automatic composition program on a corpus of musical theatre songs to generate new musical material and outputs from the program a score of vocal melodies and chords based on user-supplied lyrics. Literature [9] utilizes a sequential memory subsystem and a knowledge subsystem to form an algorithmic composition method, where the knowledge subsystem, is used to learn information about genres, composers, and titles of works, and the sequential subsystem is used to encode note pitches and durations. It can be seen that the first method has the advantage of being very logically organized, as well as being able to explain clearly the reasons for the behavior associated with music [10].

In contrast, the second class of algorithmic compositions uses machine learning or deep learning methods to create music, which are more adaptable as well as generalizable [11], [12]. For example, literature [13] trained traditional music and English Renaissance vocal compositions with recurrent neural networks to generate piano melodies. Literature [14] used Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) in a deep learning model to capture post-tonal and post-metrical features of music, and the model enabled the creation of different styles of music. Literature [15] proposes a computer-aided polyphonic music generation framework (NeuralPMG) that uses machine learning, Leap Motion devices and brain-computer interfaces to enable the generative creation of polyphonic music compositions. Literature [16] utilizes gated recurrent units (GRUs) in recurrent neural networks to simulate patriotic melodies from original musical compositions, a model that has been successful in creating complex musical compositions. The second class of methods greatly reduces the threshold of music composition because, in contrast to the first class of methods, this class of methods does not require you to formulate the appropriate musical rules, which are supported by a large amount of music theory knowledge [17]. The second type of method directly omits this process, it only requires the creator to understand the basic knowledge of music theory to be able to create music, which greatly reduces the threshold required for music creation [18], [19]. The research conducted in this paper is also based on the second type of algorithmic composition.

This paper proposes an innovative framework based on neural network optimization algorithm, which realizes the intelligence and efficiency of music creation by fusing genetic algorithm and BP neural network. In the system modeling, firstly, multi-dimensional encoding of musical attributes is carried out: 12-bit octal code is used for scale vector, 12-bit quadratic code is used for range vector, and octal code is also used for beat vector, which ensures the uniqueness and operability of the encoding space. The melodic segments are dynamically optimized by genetic operator crossover and mutation, which solves the problem of insufficient diversity in traditional methods. Further, BP neural network is introduced as the core tool for the generation of counterpoint in polyphonic music, which effectively overcomes the shortcomings of traditional BP algorithms, such as slow convergence and easy to be trapped in local extremes, by learning the nonlinear mapping relationship between the input tunes and the counterpoints, and combining with the elastic gradient descent method to improve the training process. Through the synergistic design of coding strategy and network structure, the system is able to generate the counterpart vocal parts that conform to the polyphony rules based on the motivic fragment, which significantly improves the automation level of music creation.

II. Research on music composition model based on BP neural network

II. A.Music Composition Algorithm System Modeling

II. A. 1) Coding rules

In this system, we refine and illustrate three properties.

(1) Scale Vector

Within the same city of tones, the scale variations are “do re mi fa so la xi” and a rest, which corresponds to the elemental representation of “C D E F G A B” and “z” in the previous section. We denote each of them by a number, mapped as “1 2 3 4 5 6 7 0”, so that for a scale vector there are 8-12 = 2-36 = 6.8719e+ 10 possibilities, and thus the scale property can be uniquely determined by an integer between (0, 6.8719e+ 10-1), and a 36-digit integer. Accordingly, it can be determined either by a 36-bit binary code or a 12-bit octal code, i.e., a tone scale attribute code. The latter, a 12-bit octal code, is used in this system.

When coding, the first step is to determine if there are enough twelve notes in each measure, and if not enough, they need to be supplemented to twelve with “[ ]”. “[ ]” is the code for the rest z's, each of which is represented by a corresponding [ ].

(2) Range Vector

The sound range of the simple music studied in this paper is generally within four octaves, and it is assumed that from low to high, there are four parts: bass, mid-bass, mid-high, and treble. It is mapped to the four numbers 0, 1, 2, and 3 respectively: then for a sound city vector, it has a total of four to the twelfth power, that is, 4-12 = 2-24 = 16777216 possibilities, so the vocal domain property can be uniquely determined by an integer between 0 and 16777216-1, and correspondingly can be uniquely determined by a 24-bit binary code or a 12-bit quadruple code, that is, the vocal domain attribute code. The latter is the case, a 12-bit quadruple code.

Note that the default range is bass 0 for rest z and for spaces encountered in the encoding.

(3) Beat Vector

In simple music, the common beat range is generally 1/16~1 with seven beats, which are mapped as 1, 2, 3, 4, 5, 6, and 7, which include the rest z in the consideration of the Fan Circle, which is mapped as 0: for a beat vector, there are 8-12=2-36=6.8719e+10 possibilities, and thus the beat attribute can be uniquely determined by an integer between 1 and 6.8719e+10, and accordingly, a 36-bit integer can be used to uniquely determine the beat attribute.

An integer between 1 and 6.8719e+10 can be uniquely determined, and accordingly, a 36-bit binary code can be uniquely determined with a 12-bit octal code determination, i.e., the beat attribute code. The latter, a 12-bit octal code, is used in this system.

Note that for the space portion of the code, the default beat is 0.

II. A. 2) Genetic operators

(1) The crossover operator

In the system studied in this paper, there are two main types of crossover methods: crossover in terms of subsections, or crossover in terms of parts in a single subsection. It should be noted that these two crossing methods in the actual application of the process need to pay attention to the cross part of the time value must be equal to carry out the cross operation, otherwise, if the cross part of the time value is not the same, the cross will result in the subsection of the time value of the total sum is not equal.

(2) Variation operator

In the system studied in this paper, in order to reduce the realization complexity. As well as for the characteristics of the simple music, we have selected four types of mutation: randomly generating a new note, randomly replacing an existing note, mutation in the case of excessive interval spacing, and mutation in the case of suspended tones. All four types of mutations are possible only if certain trigger conditions are met, and the specific rules will be determined on a case-by-case basis.

II. B.Principles of BP Neural Networks for Creating Counterpoint Combined Voices

After completing the construction of the system model of the music composition algorithm, how to further optimize the generation of counterpoint vocal parts becomes critical. In this section, we will combine BP neural networks to explore the principles and methods of its generation of counterpoint combined vocal parts.

BP network is a multilayer network that utilizes nonlinear differentiable functions for weight training. Due to its simple structure and plasticity, it is widely used in the fields of function approximation, pattern recognition, information classification and data compression. Since the counterpoint always corresponds to a fixed tune in polyphonic music composition, and the composition of counterpoint must be limited to follow the rules of polyphonic music composition, the counterpoint can be regarded as the feedback of the fixed tune to a certain extent, so the fixed tune can be described as the input, and the counterpoint can be taken as the output, and the BP network can be applied to train the generation of the counterpoint. In this type of attempt, how to utilize the rules to design the training samples becomes the key to the creation of neural networks.

II. B. 1) Steps and Methods of BP Neural Network for Generating Counterpart Vocal Parts

In using BP neural network to realize the automatic generation of counterpoint combination voice parts the process is as follows:

(1) Select the coding strategy to encode the collected sample music and motivic segments; (2) Construct the BP neural network and set up the necessary parameters.

(2) Construct the BP neural network and set up the necessary parameters.

(3) Input the training samples, and finally get the trained network; (4) Input the motivation clips, and finally get the trained network; (5) Input the training samples.

(4) Input the motivic fragments and train the network to obtain the desired counterpoint combination of vocal parts.

Here, when the samples are used to train the network, the BP algorithm can be improved due to the shortcomings of slow convergence, local extremes, and difficulty in determining the number of hidden layers and hidden layer nodes. We improve it by using the BP training method that can be reset. This method is also called elastic gradient descent method, which can eliminate the slope of the output function is close to 0 when the input is very large or very small. then when the gradient descent method is applied to train the multilayer network, the gradient order of magnitude will be very small, which will make the adjustment range of the weights and thresholds reduce, that is to say, even if it does not reach the optimal value, it will also form a phenomenon such as the result of the training stops.

II. B. 2) Coding of notes

In Bach's two-voice creative compositions, the counterpoint union voices are directed to the motivic fragment. Therefore, it is necessary to analyze the motivic fragments first. There are several forms of diatonic counterpoint compositions: one-tone to one-tone, two-tone, and four-tone. These three forms should be analyzed separately. First of all, we need to take a fixed tune as the input, and when coding, we regard the input tune as a set of n-dimensional vectors, and each note corresponds to a number in the vector, which is more intuitive and concise than the numerical form of coding, and the interval gaps of the notes in the notes can be correspondingly expressed by the gaps between the numbers, which will be much easier when coding the notes. This makes it much easier to

encode notes. For example, if a note is notated as [1, 3, 5], and the pitch is more than an octave higher, but the pitch does not change, it can still be encoded in this way, e.g., by notating 1 as an octave higher than 8, where we are not allowed to encode a negative number, and therefore 1 is the lowest note in the measure. If we make the input vector 2x3. Then, for example, if we take a note to a diatonic, what we want to get from this 2-dimensional vector is a 1-dimensional vector of 1x3, so when designing this neural network we should see that we are aiming for a 1-dimensional output from an n-dimensional input, so that when designing the neural network the hidden layer can be increased by a few more neurons, or by increasing the number of layers of the neural network, because music doesn't have any rigid regularity. The nonlinearity of the music is often very strong when it is written as an encoding, so designing the network in this way can increase the ability of the network to represent nonlinear mappings.

III. Experimental design and performance evaluation of music generation model based on multi-dimensional feature verification

In Chapter 2, the systematic design of the music composition model and the construction of the counterpoint vocal part generation mechanism are completed by fusing the genetic algorithm and the improved BP neural network. In order to further validate the model's generation effect and rule adaptability in real music scenes, Chapter 3 will focus on the systematic analysis of experimental dataset construction, multi-dimensional feature validation and manual evaluation to quantitatively assess the model's innovativeness and practicality.

III. A. Experimental dataset construction and comparative model selection

III. A. 1) Data pre-processing

In terms of dataset selection, LakhMIDI dataset and MUT dataset, which are large in data volume and meet the experimental requirements, are selected in this chapter. The former contains 203,783 pieces of multi-track music in MIDI format with labeled instrumental tracks, while the MUT dataset consists of a wide range of styles of music in MIDI format provided by Tencent, collected on the web and various platforms, and selected. The MUT music dataset consists of a series of subsets categorized according to different genres, e.g., number of instrumental tracks, genre, label type, etc. The dataset can be applied to music understanding. The dataset can be applied to music understanding tasks such as melody generation, accompaniment generation, style migration, music recognition, etc.

There are 98 different annotation types in the database, so each note can be represented using a tensor of length 95. For music in 4/4 beat, the corresponding time step is 16, so the note information of each measure in this beat can be converted to a 16* 95 matrix. It should also be noted that after the conversion, if the total duration of some measures is longer than 16, in this case, scaling and rounding will be done for each note in these measures until the duration is less than 16. If the number of notes in two different neighboring measures is greater than 16, then even if the length of each note is 1, the overall length of the entire measure will be greater than 16, in this case, the first 16 notes in the measure will be used directly as the note information of the measure, which is generally very rare and will not affect the overall training of the model.

At the same time, we also chunked the data, divided each piece of music into blocks of 100 characters in length for processing, i.e., batch_size is set to 100, and the first block is passed to the model as an input, and at the same time, the block with the same target is passed (shifted to the left by 1), and the weights of the model are updated through back propagation, and we need to pay attention to the fact that, in the recursive neural network, the We should note that in recurrent neural networks, “batch size” and “block size” are two different hyperparameters.

III. A. 2) Contrasting models

In the automatic evaluation, the method of this paper is compared with two benchmark methods. The two benchmark methods are RNN, LSTM, and Seq2Seq (S2S), Seq2Seq plus Skip-thought (S2SS), Seq2Seq plus Attention (S2SA), Seq2Seq plus BeamSearch (S2SB), Seq2Seq plus Teacher Forcing (S2ST). All model codes are implemented through the deep learning framework Tensorflow.

III. B. Multi-dimensional characterization validation experiments for generating music

After completing the construction of the experimental dataset and the selection of the comparison model, in order to further validate the quality of the generated music, this section develops a multi-feature analysis in terms of similarity, pitch distribution and other dimensions, and quantifies the correlation between the model generation effect and the real music through the comparison of waveform plots and statistical distributions.

III. B. 1) Comparison of similarity

We compared the similarity between the generated music and the database music. The specific comparison process is as follows: firstly, the music file in mid format is converted to MP3 format, meanwhile we get the waveform graphs of the two music as shown in Fig. 1.

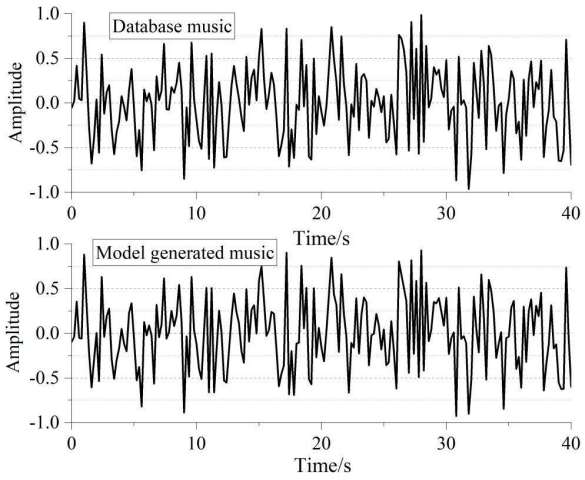

Figure 1: Database and model generated music waveform diagram comparison

As can be seen from Figure 1, the overall similarity between the music waveforms generated based on the BP neural network creation algorithm model and the music waveforms in the database reaches 86.74%, reflecting the high similarity between the music creation algorithm model constructed in this paper and the actual music.

III. B. 2) Comparison of pitch distribution

While conducting the experiments, we found that the model in this paper is able to learn the chord generation rules in the database to some extent. For example, in real compositions, when the notes C,E,G appear in a certain measure, the C major chord is usually considered. Figure 2 shows the pitch distribution of the music generated by the database and the model in this paper in C major.

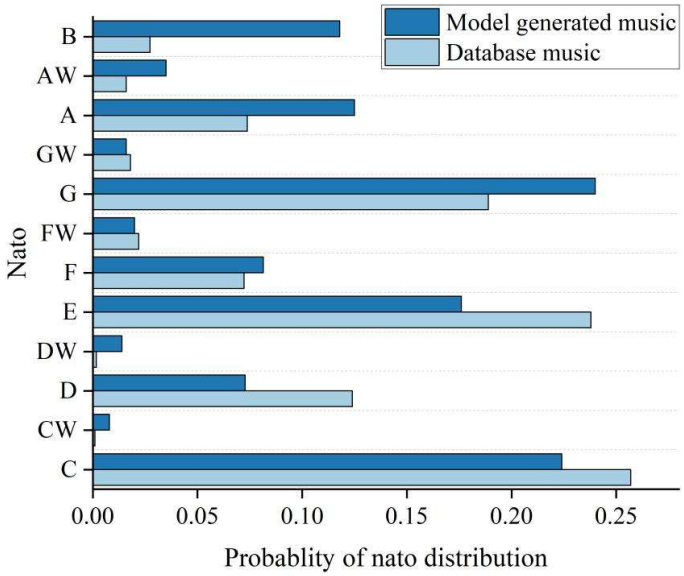

Figure 2: The database and model generate the pitch distribution of music in C major

From Fig. 2 we can see that for C (C major) chords, the distribution probabilities of the notes generated using the model are roughly the same as those in the actual database, especially for the chord constituent notes, in the case of C chords, for example, C,E,G are the constituent notes, which occupy the top three positions in the test dataset,

and at the same time, they also occupy the top three positions under the C chord in the generated data, which indicates that the model in this paper can learn the chord generation rules in the database to some extent.

III. C. Model Training Process

Based on the validation results of the multidimensional features, this section further analyzes the training process of the model, reveals the convergence dynamics of the model through the iterative curves of the loss function, and evaluates the effectiveness of the network structure and the optimization algorithm by combining the training efficiency with the final loss value. Figure 3 demonstrates the change of loss during model training.

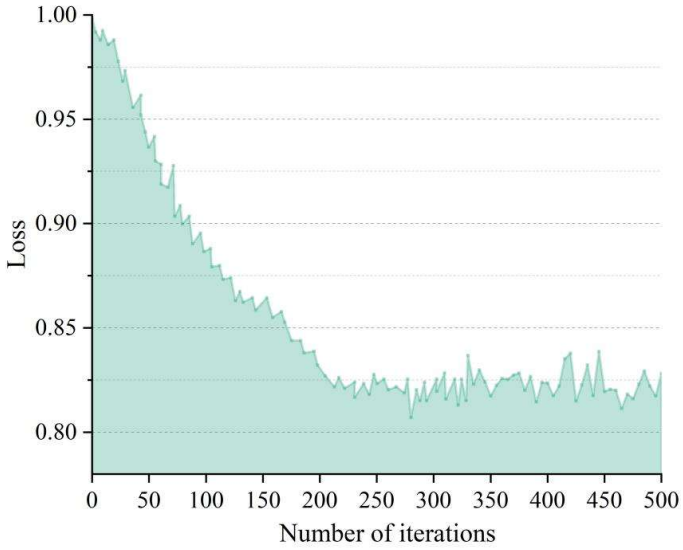

Figure 3: Model loss iteration

As can be seen from the figure, the overall decline of the model is faster at the beginning of the model training, when the whole model converges to about 200, the model starts to show a slow decline state, and after the model has gone through up to 500 iterations, the model finally converges to around 0.828 loss. The final loss value of the model is small, indicating that the model has converged correctly. The experiments illustrate that a variety of different model structures have completed convergence in the music generation application, the designed network structure is adapted to the music generation problem, and the overall model can better achieve the generation goal.

III. D. Analysis of the effectiveness of manual assessment

After the model completes the training, in order to comprehensively assess the practicality and artistry of the generated music, this section introduces an artificial evaluation mechanism to score from multiple perspectives in terms of the dimensions of fluency, rhythm, and emotional expression, and to compare the differences in the performance of different models at the creative level.

The music lyrics generated by the model not only need to meet the basic requirements such as content fluency, but also need to have higher requirements such as musical beauty. Thus, we divide the evaluation criteria of lyrics into a series of criteria, the first series of criteria is the basic text generation criteria, which mainly includes the criteria of smooth content and consistent theme. The second series of criteria are high-level requirements, which mainly assess whether the lyrics have musical beauty, including the degree of rhyming, the length of the generated text feeling and other criteria.

III. D. 1) Scoring criteria

In the manual evaluation, the effect of lyrics generation is evaluated from the perspective of rhyming angle, full text fluency angle, content consistency angle, theme relevance, readability angle, length of the generated utterance, differentiation between the generated lyrics and the real lyrics, and emotionality score.

The evaluation standard adopts a five-point system, each aspect of the total score of 5 points, the higher the score indicates that the generation effect is more excellent, the more people recognize the current lyrics, of which 2 points indicates that the lyrics can not meet the basic requirements, the lyrics as a whole there is a certain lack of 3 points indicates that the lyrics basically meet the requirements of the generation can be used for the creation of the lyrics, 4 points indicates that it is excellent, the generation effect has been recognized by human beings adequately, according to this criterion is considered to be given a certain number of points. Artificial evaluation

mainly judges two parts, the first part is to evaluate the overall lyrics to bring human feelings; the second part of the details of the evaluation, judging the score of a single sentence, and finally take the average of the score of a single sentence as the final score of a single sentence.

The judges were 50 fellow graduate students majoring in music, ranging in age from 22-28, with an average age of 24.7. All graduate students gathered in a separate classroom, and each scored the lyrics independently, without being allowed to talk or communicate with each other, and then left the classroom on their own after the scoring process was completed. There was no time limit for the scoring process, and the evaluation process ended when everyone had finished scoring.

III. D. 2) Comparison of basic evaluation results

Table 1 shows the basic expression requirements of the models in this paper and the eight models of RNN, LSTM, S2S, S2SS, S2SA, S2SB, and S2ST for their generation of composed music lyrics.

Table 1: Different models generate the basic expressive results of the lyrics

Model	Fluency	Consistency	Correlation	Readability
RNN	3.01	3.92	3.39	3.70
LSTM	3.37	3.26	3.40	3.19
S2S	3.55	3.38	3.94	3.26
S2SS	3.52	3.13	3.37	3.68
S2SA	3.57	3.14	4.01	3.19
S2SB	3.92	3.61	3.78	3.58
S2ST	3.41	3.91	3.94	3.10
OURS	4.23	4.49	4.36	4.28

The comparison of basic expression requirements shows that the model proposed in this paper significantly outperforms other benchmark models RNN, LSTM, and Seq2Seq series in four metrics: fluency 4.23, consistency 4.49, relevance 4.36, and readability 4.28. Among them, fluency and consistency scores are the highest, indicating that the lyrics generated by the model are close to the level of manual creation in terms of grammatical coherence and thematic unity. Comparing with other models, Seq2Seq plus BeamSearch performs next best on fluency 3.92 and consistency 3.61, but still has a significant gap with this paper's model, with a difference of 0.31 and 0.88, respectively.The LSTM model scores lower on consistency 3.26 and readability 3.19, which may be due to the lack of its long time-dependent modeling ability resulting in thematic deviation or redundant utterances.

III. D. 3) Comparison of high-level evaluation results

Table 2 shows the results of the high-level evaluation of the lyrics.

Table 2: Comparison of high-level evaluation results of different models

Model	Rhyme	Length feeling	Emotion	Discrimination	Average
RNN	3.47	3.21	2.95	3.25	3.22
LSTM	3.62	3.50	3.08	3.28	3.37
S2S	3.14	3.58	3.14	2.98	3.21
S2SS	3.61	3.71	3.84	3.98	3.79
S2SA	3.72	3.60	3.73	3.98	3.76
S2SB	3.19	3.35	3.46	3.59	3.40
S2ST	3.70	3.15	3.70	3.53	3.52
OURS	4.57	4.32	4.28	4.17	4.34

The high-level evaluation comparison further verifies the comprehensive advantage of this paper's model. In the four indexes of Rhyme 4.57, Length Feeling 4.32, Emotional Expression 4.28 and Differentiation 4.17, this paper's model is ranked first with a significant advantage, and the average score reaches 4.34, which is much better than the other models. In comparison, Seq2Seq plus Attention (S2SA) performs better in emotion expression 3.73 and differentiation 3.98, but its rhyme score 3.72 and length perception 3.60 are still far away from this paper's model with a difference of 0.85 and 0.72, respectively.The RNN model scores the lowest in rhyme 3.47 and emotion expression 2.95, which may be due to its simple structure is difficult to capture the complex nonlinear relationships

in music. In addition, the high score of this paper's model on differentiation indicates that the difference between generated lyrics and real lyrics is small and close to the authenticity of human creation.

III. E. Comparative analysis of model performance

The manual evaluation results reveal the subjective advantages of the model. In this section, we further combine the music theory rules with the evaluation of professional composers, and systematically analyze the comprehensive performance of the model and its innovativeness in music generation tasks from the levels of objective metrics (e.g., note repetition, chord matching) and subjective rule fitness.

III. E. 1) Comparison of Musical Characteristics

In order to verify whether the music rules and their rewards and punishments in the models in this paper play a guiding role for the models in generating music, this section quantifies the music rules into objective indicators based on the set music rules and conducts a series of comparison experiments for the music rules. This experiment is conducted for the comparison of 500 music pieces generated by seven models of RNN, LSTM and S2S series. In order to ensure the fairness of the experiment, the parameters of the five models, the number of bars and the starting notes of the generated music are the same when generating the samples. From the music samples of the above eight models, the series of effective feature information was selected, and seven corresponding evaluation indexes were proposed, including P1 note repetition, P2 average autocorrelation, P3 note out of key, P4 interval difference greater than eight, P5 with unique maximum note, P6 with unique minimum note, and P7 pop style, and then summarized according to the scores, and the comparison of the music theory features is shown in Table 3 shows.

Table 3: Comparison of music theory characteristics

Model	P1	P2	P3	P4	P5	P6	P7
RNN	53.53%	0.23	15.30%	54.12%	46.99%	42.16%	60.38%
LSTM	43.08%	0.43	12.77%	43.97%	49.96%	44.53%	67.21%
S2S	43.60%	0.21	7.31%	39.04%	50.22%	46.33%	70.17%
S2SS	33.62%	0.14	7.96%	36.21%	52.79%	48.87%	76.04%
S2SA	30.38%	0.18	8.94%	37.10%	53.81%	50.84%	78.16%
S2SB	32.33%	0.15	6.57%	34.54%	51.48%	50.13%	74.32%
S2ST	29.41%	0.16	6.24%	35.95%	54.14%	52.09%	78.98%
OURS	18.77%	0.08	5.53%	32.58%	56.63%	58.36%	91.48%

Table 3 Comparison of music theory features shows that this paper's model is comprehensively ahead of other comparison models in seven music theory indicators. Specifically, in note repetition, this paper's model is only 18.77%, which is much lower than the 53.53% of traditional RNN and 43.08% of LSTM, indicating that its generated music is more diversified; in average autocorrelation, this paper's model scores 0.08, which is significantly better than all the models, such as the 0.18 of S2SA, indicating that the generated notes are more independent. In addition, this paper's model performs best in interval reasonableness of 32.58% and tonal accuracy of 5.53%, indicating that the model can effectively circumvent the errors of excessive intervals or notes deviating from the tonality. Notably, the model scores the highest on style characterization of 91.48%, verifying that its generated music is closer to the popular style of real works. These results are attributed to the synergistic optimization of the genetic algorithm and the improved BP neural network, which makes the music generation both satisfy the rule constraints and innovative.

III. E. 2) Rule measurement

Rule metrics is a crucial method in subjective evaluation, which evaluates whether the generated music meets the experts' criteria for music. A dataset containing 50 pieces of music was composed by professional composers in accordance with the rule metrics for the 8 model-generated music pieces, and professional composers from the Central Conservatory of Music, the Communication University of China and Zhengzhou University were asked to conduct aural evaluation of the music in this dataset, and the invited participants were all professional composers with relevant compositional educational backgrounds, and the scoring statistics were conducted according to the rule metrics scoring criteria in Section 3.4.1. The scoring statistics were performed and averaged, and the results are shown in Table 4 below.

Table 4: Rule test score

Model	Chord progression	Interval	Concordance degree	Pitch	Style
RNN	3.20	3.27	3.44	4.00	3.58
LSTM	3.38	3.23	3.5	4.01	3.86
S2S	3.92	3.62	3.71	4.04	4.02
S2SS	3.66	3.39	3.45	4.12	3.09
S2SA	3.83	3.28	3.93	4.13	3.15
S2SB	3.86	3.39	4.01	4.15	3.45
S2ST	3.36	3.08	3.16	3.99	3.14
OURS	4.52	3.91	4.12	4.52	4.18

The rule-based evaluation further validates the superiority of this paper's model from the perspective of professional composers. In the five indicators of chord progressions, intervals, harmony, pitch and style, the model of this paper ranks first with significant advantages. For example, in terms of harmony, the model scores 4.12, which is much higher than the 4.01 of S2SB and the 3.93 of S2SA, indicating that the counterpoint voices generated by the model are more in line with the harmonic rules of polyphonic music. In addition, the leading pitch score of 4.52 and style score of 4.18 indicate that the model is able to accurately capture the pitch distribution and style features by learning the multidimensional coding strategy. In contrast, the lower pitch score of 3.27 for RNN and style score of 3.14 for S2ST expose the limitations of traditional models in modeling complex rules. In conclusion, the BP neural network-based model proposed in this paper generates music that is most similar to the style of real music in the models mentioned above, has the highest degree of harmony, most closely matches the pitch and chord progressions set in this paper, and most closely matches the intervals used in human compositions. It reflects that the model proposed in this paper has certain advantages in generating effects.

III. E. 3) Comparison of music effects

In order to verify the effectiveness of the model in terms of generative network and discriminative network, the model of this paper is compared with seven other comparative models in terms of both chord matching and musical harmony, and experiments are carried out using two datasets, LakhMIDI and MUT. This experiment not only validates the effectiveness of the generative and discriminative network improvements, but also validates the effectiveness of the model compared to the other models. In order to compare the fairness of the experiments, this experiment uses the same training dataset and test training set for all the models, and the length of the generation is the same. The comparison of different models in terms of the overall effectiveness of music generation is shown in Figure 4.

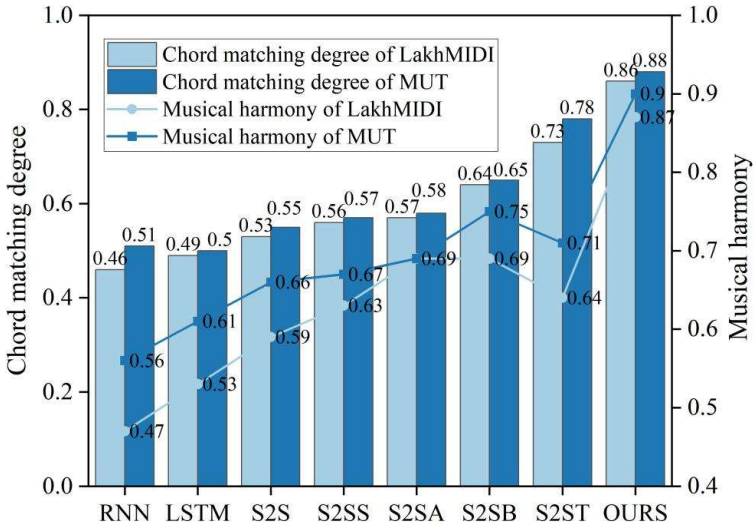

Figure 4: Comparison of the overall effect of different models on music generation

The music effect comparison quantifies the model performance from the perspective of real datasets. In terms of chord matching, this paper's model reaches 0.86 and 0.88 on the LakhMIDI and MUT datasets, respectively, which are significantly higher than the 0.73 and 0.78 of S2ST and the 0.64 and 0.65 of S2SB, verifying the guiding effect

of its encoding strategy on chord generation. In terms of musical harmony, the model in this paper also performs outstandingly, with 0.87 on the LakhMIDI dataset and 0.90 on the MUT dataset, and especially breaks through the 0.9 threshold on the MUT dataset, which indicates that the model possesses strong robustness in cross-stylistic music generation. Comparing with other models, although S2SA performs moderately well with a LakhMIDI harmony of 0.69, its chord matching is 0.57, which is still significantly different from the model in this paper, with a difference of 0.29. This result confirms the improved effect of elastic gradient descent method on network training and the accurate mapping of multidimensional coding rules to music theory features.

IV. Conclusion

In this study, a music composition model based on genetic algorithm and improved BP neural network is proposed, which realizes the efficient generation of polyphonic music counterpoint vocal parts through the synergistic design of multidimensional coding strategy and elastic gradient descent method. The experimental validation shows that.

(1) The music generated by the model is 86.74% similar to the real data waveform, and the pitch distribution and chord rule matching are close to the level of artificial composition.

(2) In the manual evaluation, the model exceeds the benchmark model in fluency, consistency, rhythm and other indicators, and the average score is improved by about 22%.

(3) The evaluation of music theory rules shows that the model is significantly ahead in the objective indicators such as note repetition degree of 18.77% and style matching degree of 91.48%, which verifies the diversity and professionalism of the music it generates.

(4) The evaluation by professional composers further confirms that the music generated by the model is close to the human composition standard in terms of subjective indicators such as chord progression score of 4.52 and harmony score of 4.12.

References

[1] Yung, B. (2019). Exploring creativity in traditional music. Yearbook for Traditional Music, 51, 1-15.

[2] Zhao, B., Zhan, D., Zhang, C., & Su, M. (2023). Computer-aided digital media art creation based on artificial intelligence. Neural Computing and Applications, 35(35), 24565-24574.

[3] Peng, W., Tang, Y., & Ouyang, Y. (2023, June). Design of Computer-Aided Music Generation Model Based on Artificial Intelligence Algorithm. In International Conference on Computational Finance and Business Analytics (pp. 229-237). Cham: Springer Nature Switzerland.

[4] Atanacković, D. (2024). Artificial Intelligence: Duality in Applications of Generative AI and Assistive AI in Music. INSAM Journal of Contemporary Music, Art and Technology, (12), 12-31.

[5] Kwiecień, J., Skrzyński, P., Chmiel, W., Dąbrowski, A., Szadkowski, B., & Pluta, M. (2024). Technical, Musical, and Legal Aspects of an AI-Aided Algorithmic Music Production System. Applied Sciences, 14(9), 3541.

[6] López-Montes, J., Molina-Solana, M., & Fajardo, W. (2022). GenoMus: Representing Procedural Musical Structures with an Encoded Functional Grammar Optimized for Metaprogramming and Machine Learning. Applied Sciences, 12(16), 8322.

[7] Mira, R., Coutinho, E., Parada-Cabaleiro, E., & Schuller, B. W. (2023). Automated composition of Galician Xota—tuning RNN-based composers for specific musical styles using deep Q-learning. PeerJ Computer Science, 9, e1356.

[8] Collins, N. (2016). A funny thing happened on the way to the formula: Algorithmic composition for musical theater. Computer music journal, 40(3), 41-57.

[9] Liang, Q., & Zeng, Y. (2021). Stylistic composition of melodies based on a brain-inspired spiking neural network. Frontiers in systems neuroscience, 15, 639484.

[10] Liu, C. H., & Ting, C. K. (2016). Computational intelligence in music composition: A survey. IEEE Transactions on Emerging Topics in Computational Intelligence, 1(1), 2-15.

[11] He, J. (2022). Algorithm composition and emotion recognition based on machine learning. Computational Intelligence and Neuroscience, 2022(1), 1092383.

[12] Hernandez-Olivan, C., & Beltran, J. R. (2022). Music composition with deep learning: A review. Advances in speech and music technology: computational aspects and applications, 25-50.

[13] Hagan's, K. (2017). Sound Anthology: Program Notes. Computer Music Journal, 41(1), 106-113.

[14] Dean, R. T., & Forth, J. (2020). Towards a deep improviser: a prototype deep learning post-tonal free music generator. Neural Computing and Applications, 32, 969-979.

[15] Colafiglio, T., Ardito, C., Sorino, P., Lofù, D., Festa, F., Di Noia, T., & Di Sciascio, E. (2024). Neuralpmg: a neural polyphonic music generation system based on machine learning algorithms. Cognitive Computation, 16(5), 2779-2802.

[16] HASTUTI, K., & HIDAYAT, E. Y. (2024). ALGORITHMIC COMPOSITION USING GATED RECURRENT UNIT FOR NATIONALISTIC MUSIC. Journal of Theoretical and Applied Information Technology, 102(1).

[17] Briot, J. P., & Pachet, F. (2020). Deep learning for music generation: challenges and directions. Neural Computing and Applications, 32(4), 981-993.

[18] Civit, M., Civit-Masot, J., Cuadrado, F., & Escalona, M. J. (2022). A systematic review of artificial intelligence-based music generation: Scope, applications, and future trends. Expert Systems with Applications, 209, 118190.

[19] Yang, L. C., Chou, S. Y., & Yang, Y. H. (2017). MidiNet: A convolutional generative adversarial network for symbolic-domain music generation. arXiv preprint arXiv:1703.10847.