

Article

The Synergy of Double Neural Networks for Bridge Bidding

Xiaoyu Zhang ^{1,2,3} , Rongheng Lin ^{2,3,*} , Yuchang Bo ^{2,3} and Fangchun Yang ^{2,3}¹ School of Information Science and Technology, Beijing Forestry University, Beijing 100083, China² School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China³ State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

* Correspondence: rhlin@bupt.edu.cn

Abstract: Artificial intelligence (AI) has made many breakthroughs in the perfect information game. Nevertheless, Bridge, a multiplayer imperfect information game, is still quite challenging. Bridge consists of two parts: bidding and playing. Bidding accounts for about 75% of the game and playing for about 25%. Expert-level teams are generally indistinguishable at the playing level, so bidding is the more decisive factor in winning or losing. The two teams can communicate using different systems during the bidding phase. However, existing bridge bidding models focus on at most one bidding system, which does not conform to the real game rules. This paper proposes a deep reinforcement learning model that supports multiple bidding systems, which can compete with players using different bidding systems and exchange hand information normally. The model mainly comprises two deep neural networks: a bid selection network and a state evaluation network. The bid selection network can predict the probabilities of all bids, and the state evaluation network can directly evaluate the optional bids and make decisions based on the evaluation results. Experiments show that the bidding model is not limited by a single bidding system and has superior bidding performance.

Keywords: bridge bidding; imperfect information; deep neural network; deep reinforcement learning

MSC: 68T20



Citation: Zhang, X.; Lin, R.; Bo, Y.; Yang, F. The Synergy of Double Neural Networks for Bridge Bidding. *Mathematics* **2022**, *10*, 3187. <https://doi.org/10.3390/math10173187>

Academic Editors: Zibin Zheng, Ruoxi Jia, Dan Li, Yuxun Zhou and Liang Xu

Received: 30 June 2022

Accepted: 27 July 2022

Published: 3 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A computer game uses the advantages of fast and accurate calculations so that computer can participate in games under some given rules instead of human beings. The computer game is divided into a perfect or an imperfect information game according to whether players have complete information about other participants during the game.

In the perfect information game, the game state of players is open to the others, who also know the information of other players. Chess [1,2], international checkers, and Go [3,4] belong to the perfect information game. However, in the imperfect information game, players only have part of the information shown to other players and will not accurately know all the information of other players. Fight the Landlords, Texas Hold'em [5,6], and Bridge belong to the imperfect information game. Compared with a perfect information game, the uncertainty of information in an imperfect game directly affects decision-making accuracy. So far, AI in most chess and card games has reached top players and has even defeated top human players in some games. At the same time, AI has a large number of applications in many engineering fields [7–11]. However, there is little research on Bridge.

Bridge is a poker game that belongs to an imperfect information game. It is divided into two stages: bidding and playing. Some software such as Bridge Baron, GIB [12,13], Snyrey, and Wbridge5 have proven more effective in the playing stage. In the bidding stage, the same bid has different meanings under different bidding systems, making bidding more challenging. In the comparative research of bridge bidding, there are three main

implementation methods of bidding system, Monte Carlo sampling algorithm, Imitation learning algorithm based on expert experience, and a Reinforcement learning algorithm based on the self game.

We believe that there are two significant challenges. Firstly, the game state space is enormous due to the uncertainty of unknown information. Exploring all possible unknown states is a severe test for computer hardware performance and game algorithm design. Secondly, the research route of these three bidding models has the same problem. That is, they only support the use of a single bidding system. When opponents use the bidding system consistent with AI in the game, they can get results. However, when opponents use an inconsistent bidding system, their understanding of what opponents are bidding for will cause the error, significantly affecting the results.

The current bidding research is based on the same system, violating the situation that both sides can use different bidding systems in the real game. Therefore, in this paper, we design an AI that can support multiple bidding systems. It can use a natural bidding system to fight against players using any bidding system. It combines two deep neural networks to process imperfect information. One selects the candidate bid, and the other takes the first output as the input to evaluate the situation to choose an appropriate bid. At the same time, due to the particularity of the rules of the bridge bidding system, combined with expert experience, this paper translates the bidding sequence information into a variety of practical knowledge. It solves the multi-systems confrontation of Bridge bidding for the first time. The main contributions of our work are as follows:

- The extraction of useful information solves the inconsistent understanding of bidding sequence under different bidding systems. Combined with expert experience, the bidding sequence is transformed into general bridge characteristic data as the input of the model, which solves the problem of understanding the historical bidding sequence of the bidding model under the multi-bidding system for the first time.
- To ensure that the output candidate bidding products comply with the natural bidding method, this paper uses a deep neural network to fit the natural bidding method so that the output of the model complies with the bidding constraints and takes the output of the network as one of the inputs of the evaluation network for situation evaluation.
- Using a deep neural network to fit the action search algorithm will speed up the calculation and get the legal bidding in the current bidding state faster. The state evaluation algorithm based on a neural network will directly give the value evaluation after receiving the input of the current bidding state. The information completion process of random sampling is no longer carried out, and the evaluation time is considerably shortened.

2. Related Works

The general bridge bidding problem can be divided into two subproblems: bidding without competition and bidding with the competition. The bidding without competition assumes that the opponents always call PASS when bidding, so the information exchange between teammates will not be blocked. The bidding with competition means that both teams want to communicate through bidding. In this paper, as the existing works [14–16], we focus on the subproblem of bidding with competition.

The existing works on bridge are mainly divided into two categories: human-based and non-human-based bidding systems. The method not based on the human bidding system allows the machine to learn the original bridge data directly and gradually form a bidding system of the machine itself in the mutual self game. Chun yen Ho and Hsuan-Tien Lin [17] proposed a learning framework based on UCB (Upper Confidence Bound) algorithm so that the model did not rely on the bidding system with humans and learned directly according to random bidding sequences so that AI could finally reach a better contract with team friends with no competitions. Chin Kuan Yeh and Hsuan Tien Lin [18] first proposed using DQN arithmetic to let machines learn their bidding system through deep reinforcement learning. However, due to the rules of bridge competition, players

need to be able to explain the system they use and can not hide information. Therefore, the application of the model of the non-human bidding system is narrow, and more research is still based on the human bidding system. The World Computer Bridge Championship (Wbridge5) and the silver medal winner (Synrey) adopt the Monte Carlo search method based on the human bidding system. Delooze and Downey [19] generated a large number of training data with the human bidding model, clustered the hand cards and bidding process using a self-organizing map (SOM) and learned the human bidding system using unsupervised learning. Amit and Markovitch designed a decision tree model, which uses the tree hierarchy to store the rules in the human bidding system [20]. The nodes of each tree store a situation state and possible actions of the bidding system in the current situation. The model’s performance and accuracy can be further improved through expert data. Finally, through the information provided by the decision tree model, the Monte Carlo method was used to complete the fuzzy information and reduce the search space of decision-making. However, the above research focuses on bridge bidding with no competition, which is inconsistent with the real game. Our previous research [14] used a recurrent neural network to simulate the contention problem of bridge for the first time and achieved good results under the natural bidding system. Rong. J and Qin. T used two estimation and policy neural networks to simulate the bridge competing bidding problem [16].

However, the above research based on the human bidding system was aimed at a single bidding system; that is, players and opponents must use the same one. When opponents in the game use the bidding system consistent with AI, they can get results. However, when opponents use an inconsistent bidding system, their understanding of what opponents are bidding for will cause errors, which greatly affects bidding decision-making. Therefore, we designed a bridge bidding AI that supports multiple bidding systems. It can use Chinese Bridge Association (CCBA) system against players using any bidding system.

3. Problem Setup

The subproblem of bidding with competition is defined as follows. Bridge is played by four players. The set of four players is defined as $X, X = \{N, E, S, W\}$. $N, E, S,$ and W represent north, east, south, and west players, respectively. Each player holds 13 cards out of the standard deck of 52 cards. We use h_i to represent the cards of player i . Hence, a standard deck of 52 cards is represented as

$$H = \bigcup_{i \in X} h_i. \tag{1}$$

In the bidding phase, the dealer decides on the bid first (referred to as the ‘opening bid’) and then the others bid in clockwise order. Each player chooses a bid from 38 bids, of which 35 ordered real bids ($1\clubsuit, 1\diamond, 1\heartsuit, 1\spadesuit, 1NT, 2\clubsuit, \dots, 7NT$) and 3 flexible bids (Pass (P), Double (X), Double (XX)). B represents the set of all bids. During the bidding process, it must be ensured that the real bid is higher than the last real bid. When three consecutive ‘Passes’ appear, the bidding process ends, and the last real bid is the final contract. We use a length- t sequence $L(t)$ to represent the bidding. Let

$$V = \{v | None, NS, EW, ALL\} \tag{2}$$

be the set of vulnerability.

4. Model

The bridge bidding decision model based on double neural network synergy comprises two core networks: bidding selection and evaluation. The decision model framework is shown in Figure 1. The left side of the figure shows the process of multi-system bidding. The upper right side is the training process of the bid selection network, and the bottom is the reinforcement learning training process of the evaluation network. This decision model

uses expert experience to assist in learning historical bidding sequences and extracts effective general feature information. According to the analysis of general feature information, all optional bids are output through the bid selection model. Then the trained evaluation model is used to evaluate all optional bids to make the final decision directly. Therefore, this section first describes how to extract historical bidding sequences, encode them, and then introduces the bid selection and evaluation models in detail.

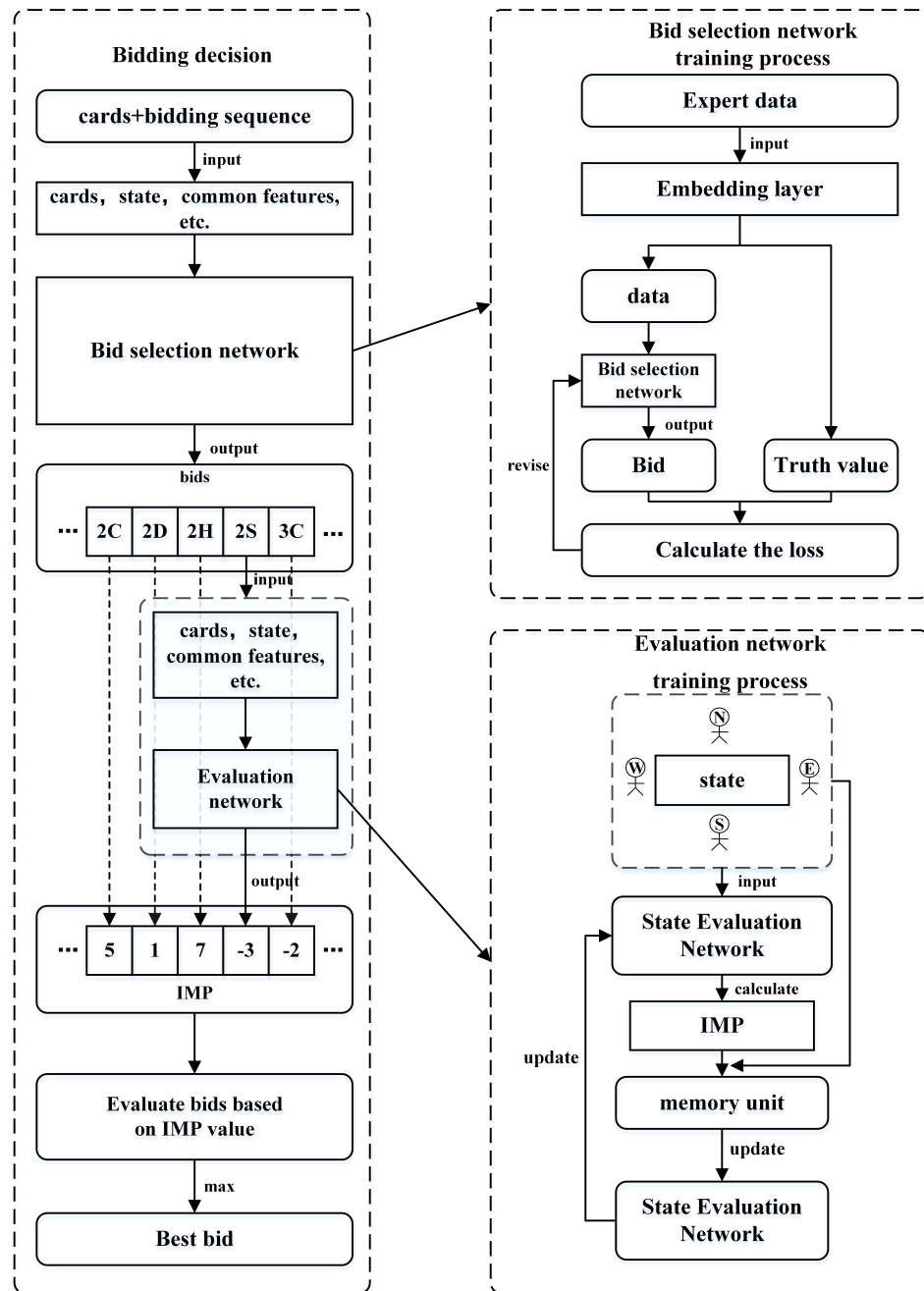


Figure 1. An overview of the proposed framework. It includes two phases: bid selection model and evaluation model.

4.1. Feature Extraction

The game information required for bidding decisions mainly includes three parts: hand information, situation information, and historical bidding sequences. The situation information includes the vulnerability, the opener, and which player’s turn it is to bid. Among them, the most difficult to deal with is the bidding sequences. Many previous

studies usually directly use deep neural networks to learn the meanings of the bidding sequences. The same bidding sequence has different meanings under different bidding systems, so previous studies can only make bidding decisions for a single system. Therefore, we no longer use the deep neural networks to learn the bidding sequences directly but convert them into general features that are not limited by bidding systems based on expert experience, thereby reducing the direct impact of the bidding sequences on the bidding decision model. No matter what bidding system is used, the model can give a general interpretation from the same dimensions. It is possible to make effective bidding decisions by learning these general features.

First, we selected 30 features that have received extensive attention in real-world games through an analysis of 2.5 million instances of bidding, combined with input from bridge experts. The bidding rules under any system can be converted into general features through fixed logic. These general features can be divided into deterministic and range-valued information. Candidate values for deterministic information are deterministic, while candidates for range value information lie within a restricted range. The specific meaning of the general features is shown in Table 1.

Table 1. Valid information and its meaning and candidate values.

Features	Meaning	Candidate Values
CUESUIT	suit	NT/S/H/D/C
BHCP	High Card Point	0–40
BTP	Total Point	0–40
BDP	Adjusted Point	0–40
BSUIT[S/H/D/C]	Number of each suit	0–13
STRENGTH[S/H/D/C]	Strength of each suit	0–7
CUECTRL[S/H/D/C]	Control of each suit	unknow/1/2/3
KINGEXIST[S/H/D/C]	Whether own K of each suit	unknow /yes/no
ST	Current bidding status	Force, PASS etc.
NT	Current bidding type	Natural Bid, Double, etc.
SLAM	Slam	Slam
AGREEDSUIT	Confirm trump	unknow/S/H/D/C
BNACE/BNKING	Number of Ace	unknow /0/1/2/3/4
BNKING	Number of King	unknow /0/1/2/3/4
NKEYCRD	Number of control	unknow/0 or 3/1 or 4/2 or 5
KSUIT	King’s suit	unknow /S/H/D/C
QUEENFLG	Whether own Queen	unknow /yes/no
POWER	Extra power	unknow /yes/no

Although the selection of these 30 general features cannot fully cover the meaning of the bidding sequences due to the complexity of the rules in the bidding systems, 30 general features are still more accurate than the neural networks directly learning the bidding sequences. With the help of Beijing Synrey Bridge Technology Limited, we design two feature extraction algorithms of the bidding sequence, F , which are suitable for extracting features of the precision bidding system and CCBA bidding system, respectively. The useful features of the bidding sequence L are represented by I_L , that is, $I_L = F(L)$.

Subsequent training of the bid selection network and evaluation network will use the extracted general features as one of the inputs. It is no longer necessary to input the bidding sequences directly.

4.2. Bid Selection Model (BM)

The bidding selection model based on deep neural networks learns the relevant information of the bidding and makes decisions. Usually, there are three types of information worth paying attention to in the bidding stage: the players’ hand cards, the bidding sequences, and other situational information. Hand cards are the most intuitive evaluation index for players’ strength information, the basis for intra-team communication and inter-

team competition. The bidding sequences are converted into general features through the feature extraction algorithm F . The situation information mainly covers the vulnerability, the opener, and which player bid. Therefore, by learning the above information, the bidding selection model uses the Mini-batch Gradient Descent (MBGD) algorithm for training, which the basic structure is shown in Figure 2. Take the three types of information encoding as input, perform feature fusion through the feature layer concatenate to obtain a 1×1284 feature vector, and then input the feature vector into the multi-layer fully connected neural network. The output layer of the model is a fully connected neural network composed of 38 neurons. After normalization, the probability distribution of 38 bids can be obtained. The definition of the bidding selection model is shown in Equation (3), which means that in the current bidding state, the probability of choosing bid b is p . When it is player i 's turn to bid, the bidding selection model receives the vulnerability v , opener d , player i , hand cards h_i and the general features of the bidding sequence L , and outputs the probability P of 38 bids.

$$P = bid_{selector}(v, d, i, h_i, L) = \{b \rightarrow p | b \in B \text{ and } p = 1\} \tag{3}$$

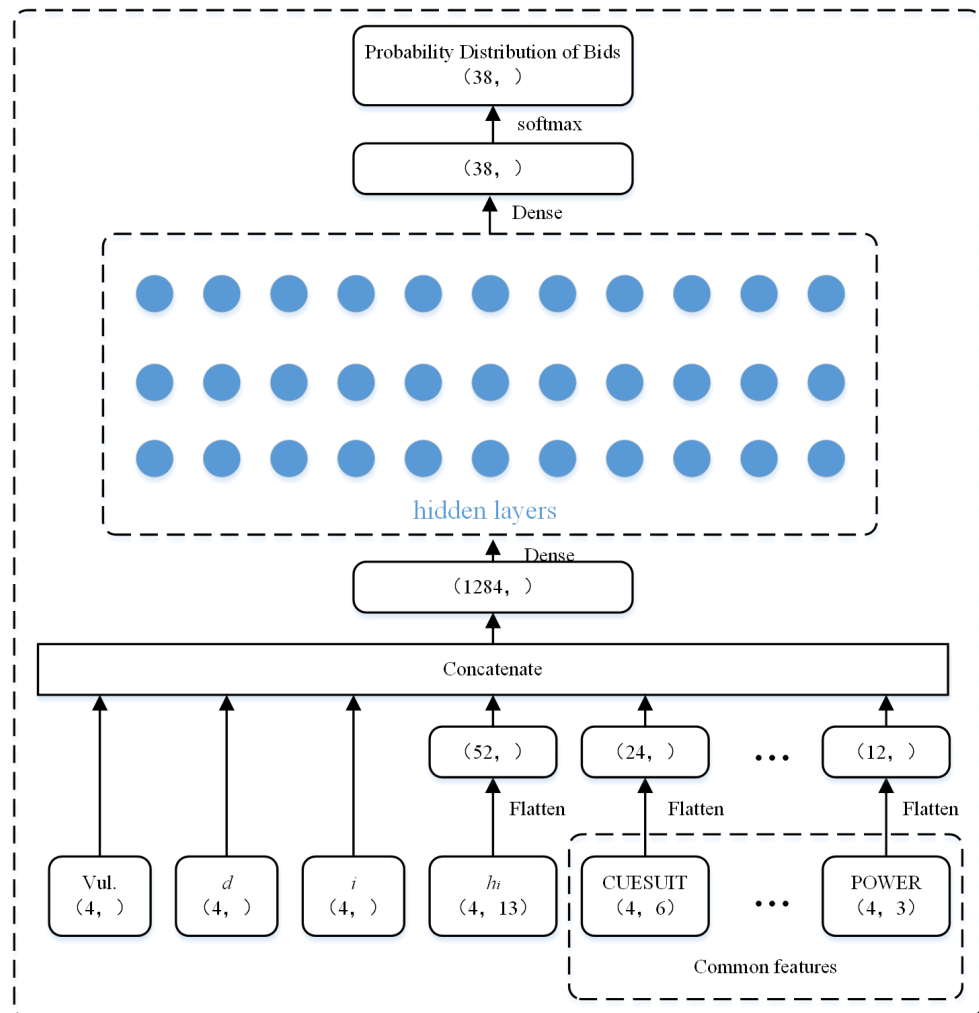


Figure 2. The structure for bid selection model.

Next, the encoding methods of several feature data are introduced in detail. Next, we introduce the encoding methods of several feature data in detail. Combined with the bidding rules of bridge and the training data features, the data needs to be preprocessed. The number of small cards has a greater impact on the bidding decision than the value of the small cards. In this paper, cards with values of 2 to 9 are classified as small cards, and the rest are high cards. High cards are coded in placeholder mode; that is, the card’s position is marked with 1; for small cards, the placeholder no longer indicates the position of the small card but the number of small cards. For example, among the 13 cards in a player’s hand, there are three small cards of the club suit, then the positions 1, 2, and 3 are marked with one, respectively, which means that the player has three small cards. Figure 3 is an example of hand “K32.AK95.T874.A6” encoding. For the range value information in the general features, we propose an averaging scheme in which the position of the data in the range is marked with *R*. It follows that:

$$R = \frac{1}{F_{max} - F_{min} + 1} \tag{4}$$

where F_{max} represents the maximum value in the range information, and F_{min} represents the minimum value. For example, a player’s high card point is 12~15, then the value of the four positions of 13, 14, 15, and 16 in the encoding vector is 0.25. Figure 4 is an example.

The bid selection model can predict available bids, which must be evaluated to make optimal decisions. In the Section 4.3, we will show how to evaluate and select these output bids reasonably.

	Small cards								High cards				
	2	3	4	5	6	7	8	9	10	J	Q	K	A
S	1	1	0	0	0	0	0	0	0	0	0	1	0
H	1	1	0	0	0	0	0	0	0	0	0	1	1
D	1	1	1	0	0	0	0	0	1	0	0	0	0
C	1	0	0	0	0	0	0	0	0	0	0	0	1

Figure 3. An example of ‘K32.AK95.T874.A6’ encoding.

0	1	...	9	10	11	12	...	15	16	17	18	...	39	40
0	0	...	0	0	0	0.25	...	0.25	0	0	0	...	0	0

Figure 4. An example of ‘HCP = 12~15’ encoding.

4.3. Evaluation Model (EM)

Bridge bidding requires the synergy of double neural networks, in which the bid selection networks select the optional bids. The evaluation networks evaluate these bids and select the one with the highest reward as the final decision. How to comprehensively evaluate a bid is an important research content of the evaluation model. Unlike chess games, a bid cannot get a real-time reward. The reward can be obtained after the bidding and playing are over. Which team win is determined by the reward. However, using the score as the reward is greatly affected by the randomness of the cards. For example, when a team’s cards are super good, it is guaranteed to win. This situation is extremely unfair to the other team. Therefore, in real bridge games, people often take duplicate bridge

tournaments to avoid the effect of the randomness of cards, that is, two teams, each of which has four players, and two tables use the same deck of cards. The scores of each team at the table are added together to get the total score, which is then converted into International Match Point (IMP). Therefore, the state evaluation model selects IMP as the evaluation standard of bids for fairness, as defined in Equation (5).

$$IMP = situation_{evaluator}(v, d, i, h_i, I_L), \tag{5}$$

where v is the vulnerability; d is the opener; h_i is hand cards; and I_L is the features of biddings sequences. It should be noted that the I_L in the bid selection model is the features generated by the bidding sequence L_{t-1} in which the player does not select the bid, while the I_L in the evaluation model is the features generated by the bidding sequence L_t after the player selects a bid.

The input and network structure of the evaluation model are the same as those of the bid selection model and will not be repeated here. Figure 5 is the schematic representation of the evaluation model.

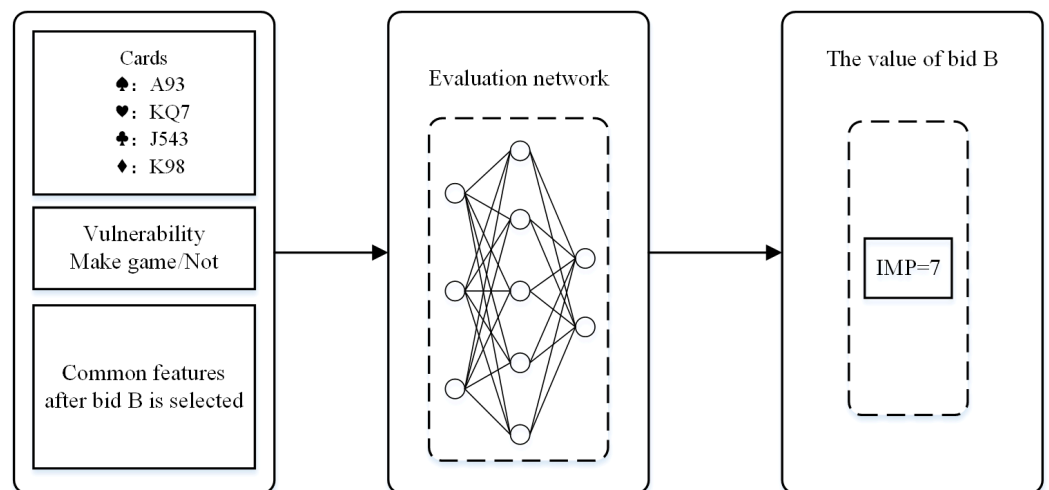


Figure 5. Schematic representation of the evaluation model.

The range of IMP is $(-24, +24)$, so this paper uses a one-hot vector with a length of 49 to encode IMP. Figure 6 is a coding example when IMP is 2.

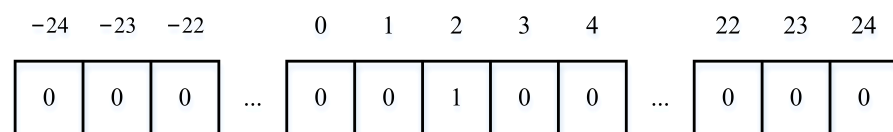


Figure 6. A coding example when IMP is 2.

Next, we introduce the calculation process of IMP in detail. Since there is no IMP in the training data for the evaluation model, the IMP needs to be calculated before training the model. The calculation of IMP requires two scores; one is the actual score after the game is over, and the other that we choose is the score of the best contract. The score of the best contract is not the highest, but the balance point where the winner cannot get a higher score and the loser cannot get a lower score. This is fair to both teams. The Beijing Synrey Bridge Technology Limited’s program can output 20 double-dummy results (ddr) after inputting four hand cards. Different declarers have different best contracts. The best contract is defined as

$$\langle contract | declarer \rangle = B_{best\ contract}(ddr, v), \tag{6}$$

where ddr represents the double dummy result; v represents the vulnerability; and $B_{best\ contract}$ represents the program that calculates the best contract by double dummy analysis.

Since this paper only focuses on bidding, it does not play cards but directly obtains the score under the contract based on the double-dummy analysis in the playing stage. Let

$$score = S_{calculate\ score}(wr, c, v', dw) \tag{7}$$

be the function of the declarer's score. The wr represents the number of tricks won by the declarer according to analysis; c represents the contract; v' represents the vulnerability of the contracting party; and dw represents whether it is doubled or not. $S_{calculate\ score}$ represents the program that calculates the declarer's score. Figure 7 shows the calculation flow of IMP. The left side of the figure shows the score calculation after the actual bid. The number of tricks won can be calculated from the double-dummy result. The right side shows the score calculation for the best contract. The two scores are finally converted into the IMP using the program–Score2IMP.

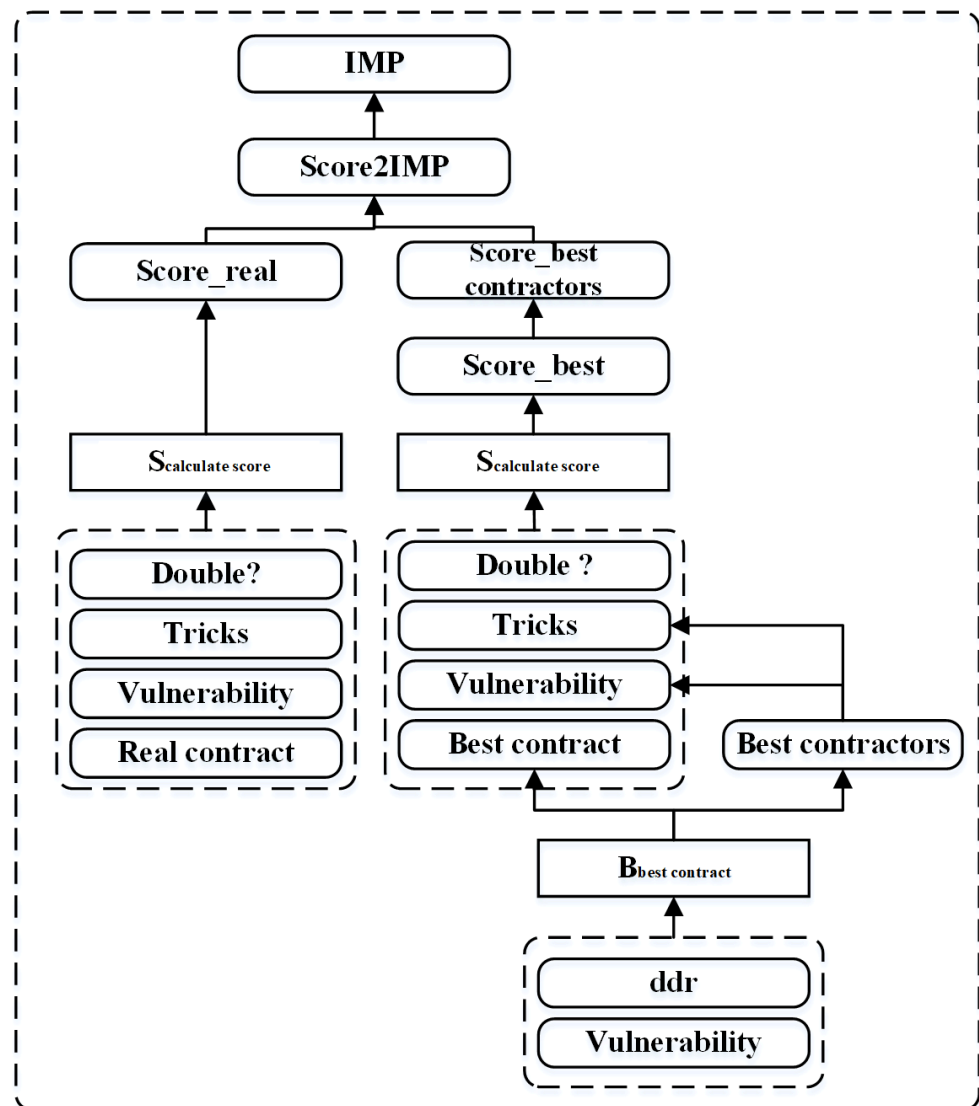


Figure 7. A flowchart to compute IMP.

Finally, we introduce reinforcement learning for the evaluation model, which first uses supervised learning for pre-training and then enhances the exploration ability through reinforcement learning. Instead of initializing the value function, we assigned the pre-trained parameters to the reinforcement learning model. Figure 8 shows the training process of reinforcement learning. Reinforcement learning models are trained using a gradient descent algorithm. First, it is necessary to output all bids with a probability distribution greater than 1% as the optional bids in a certain state through the bid selection model. Then use the greedy strategy to select the action a . If the random number is less than the set ϵ , select one of the optional bids at random; otherwise, use the pre-training model to calculate the IMP of the optional bids and select the bid with the highest IMP. Next, calculate the reward R for the selected action. We set the reward for non-contract to 0 and calculated the reward for the contract as shown in Figure 7. Store the previous state s , action a , and reward R in replay memory D . Randomly sample several pieces of data from replay memory D , and calculate the true value of the value network output according to the final state.

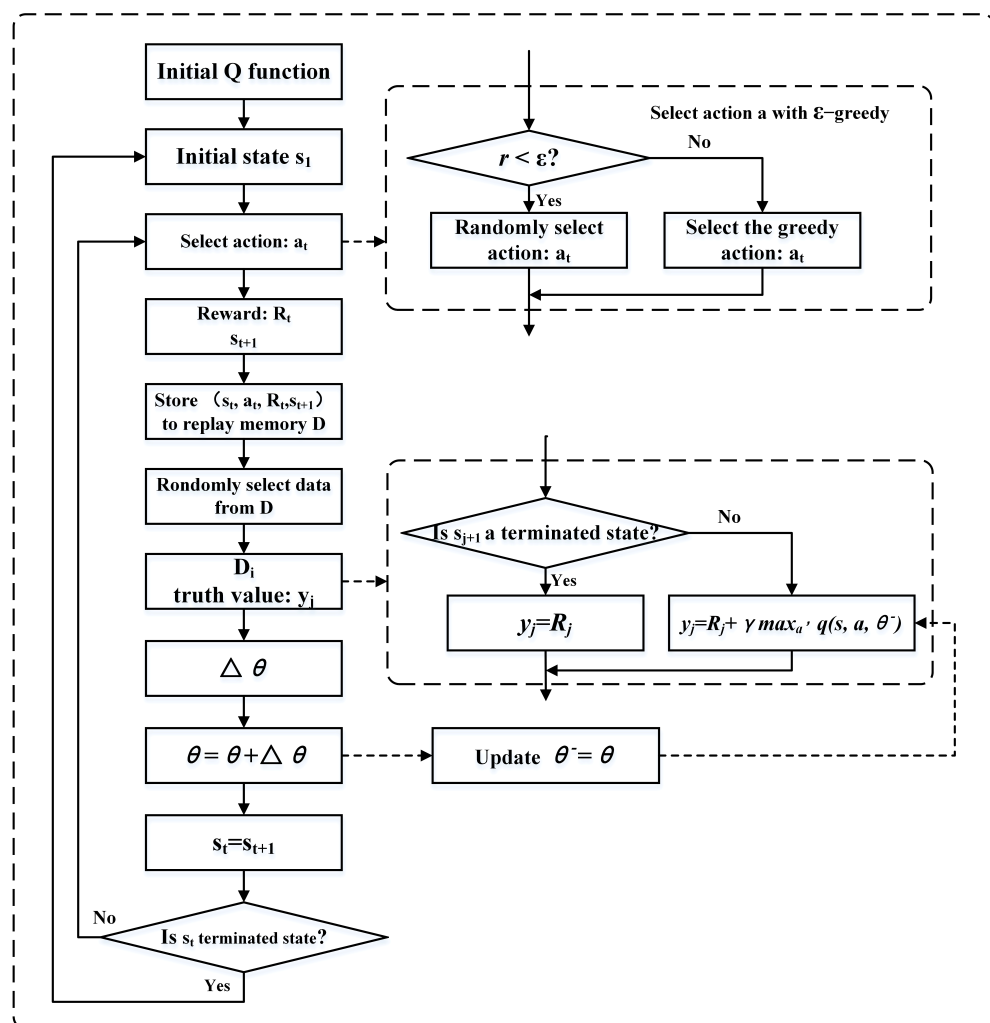


Figure 8. The training process of reinforcement learning.

5. Experiment

In this section, we conducted a set of experiments to evaluate our bidding decision system based on the synergy of neural networks. We used Python to prepare and analyze the data. We first introduced the dataset. Next, we presented the performance evaluation of BM and EM. Finally, we tested the multi-system capabilities of the bidding system.

5.1. Dataset

The dataset used in the experiments was provided by the Snyrey platform, which has accumulated a historical record of nearly 500 million bidding sequences and doubles results. We randomly selected nearly 8,000,000 real bidding instances as our dataset, 80% of which were used as training data and the rest as testing data. Tables 2 and 3 are examples of the data formats used by the two models. Each training data consist of four parts: vulnerability, opener, hand cards, and historical bidding sequence. However, the data used by EM adds one more double-dummy result than the BM. Each player holds 13 cards of four suits representing ♠, ♥, ♦, ♣. The letter ‘T’ represents 10. The double-dummy result is represented by a string of length 20, consisting of the tricks that North, South, East, and West can win when NT, ♠, ♥, ♦, ♣ are the contracted suits. A maximum of 13 tricks can be won in each round, so the numbers 10, 11, 12, and 13 are replaced by the letters a, b, c, and d in the double-dummy result.

Table 2. An example of training data in BM.

Vulnerability	None	
Opener	N	
Cards	North	Q94.KJ97.52.9763
	East	J732.2.AKQ43.AQJ
	South	K6.T53.J876.K542
	West	AT85.AQ864.T9.T8
Bid Sequence	P 1D P 1H P 1S P 3S P 4S P P P	

Table 3. An example of training data in EM.

Vulnerability	None	
Opener	N	
Cards	North	Q94.KJ97.52.9763
	East	J732.2.AKQ43.AQJ
	South	K6.T53.J876.K542
	West	AT85.AQ864.T9.T8
Bid Sequence	P 1D P 1H P 1S P 3S P 4S P P P	
Double dummy analysis	42435424359b8a89b8a8	

Next, we verified the rationality of the dataset from three aspects: opener, vulnerability, and the length of the bidding sequence. Figure 9a,b shows that the distribution of opener and vulnerability in the dataset was relatively balanced, which is in line with the real situation of bridge games. Figure 9c shows that the length of the bidding sequence in the dataset was approximately normally distributed. The bidding sequence length of the 10 had the most data, and the distribution on both sides of the 10 gradually decreases. The shortest bidding sequence was 4, which was in line with the real situation; there will not be a situation where the bidding sequence is less than 4 in the bidding stage. Overall, the relevant information in the dataset conforms to the probability distribution in the bidding, so the dataset is valid.

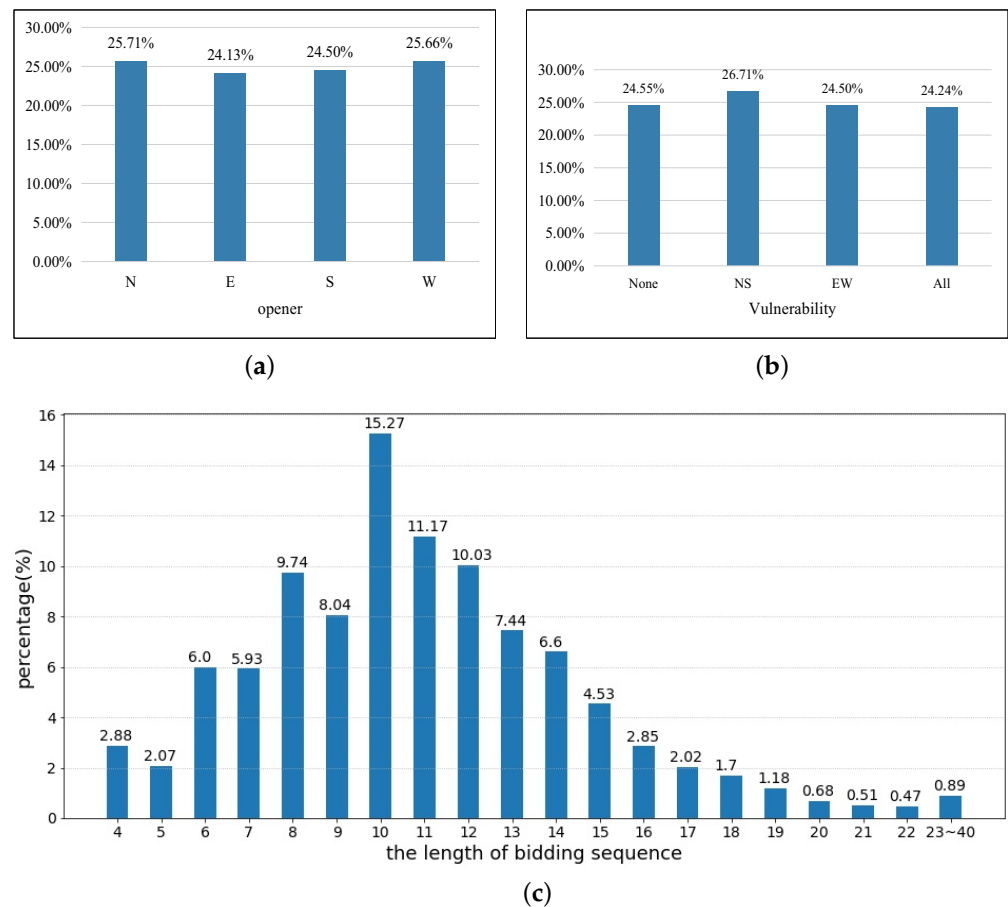


Figure 9. Dataset Evaluation. (a) The distribution of opener in dataset. (b) The distribution of vulnerability in dataset. (c) The distribution of lengths of bidding sequences in the dataset.

5.2. Evaluation of BM and EM

Next, we evaluated the performance of BM and EM separately.

5.2.1. Evaluation of BM

We first analyzed BM from the network structure, activation function, and optimizer, and then we selected the optimal parameters to evaluate the performance. We first analyzed BM from three aspects: network structure, activation function, and optimizer, and then conducted subsequent experiments after selecting the optimal parameters. In the experiment, an epoch was set to 30, and the batch size was set to 512. We selected three metrics: Total-Precision (overall precision), Real-Precision (precision of the ordered bids), and Unreal-Precision (precision of no-bid). Table 4 shows the effect of different network structures on the results. When the model selects a 4-layer fully connected neural network with 2048 neurons in each layer, the precision reaches 92%. After that, the effect of increasing the number of neural network layers and neurons is weakened.

The activation functions after full connection layers also greatly influence the convergence of the model. We selected three common activation functions in the training process, Sigmoid, Tanh, ReLU, and Leaky ReLU. Figure 10 shows the effect of different activation functions on model performance. It is easy to see that when the tanh was used, Total Precision, Real Precision, and Unreal Precision were all lower; the Sigmoid and Relu had similar effects, and the overall precision in the 1% fluctuated up and down; the effect of the Leaky ReLU worked best. Meanwhile, we compared three common optimizers. Figure 11 shows that different optimizers had different effects on model performance. Adam had improved performance over the other two.

Table 4. Comparison of experiment results with different network structures.

Structures	Total Precision	Real Precision	Unreal Precision
(512, 512, 512)	83.5%	72.3%	86.7%
(512, 1024, 512)	84.8%	72.3%	87.4%
(1024, 1024, 1024)	87.9%	72.6%	88.0%
(1024, 2048, 1024)	89.7%	73.5%	90.4%
(2048, 2048, 2048)	90.3%	73.7%	92.2%
(512, 512, 512, 512)	84.2%	72.6%	87.3%
(512, 1024, 1024, 512)	87.0%	72.7%	87.5%
(1024, 1024, 1024, 1024)	89.5%	73.0%	91.4%
(1024, 2048, 2048, 1024)	90.1%	72.6%	92.1%
(2048, 2048, 2048, 2048)	92.9%	75.3%	94.1%
(512, 512, 512, 512, 512)	85.1%	72.3%	87.3%
(512, 1024, 1024, 1024, 512)	88.6%	73.7%	88.3%
(1024, 1024, 1024, 1024, 1024)	89.6%	74.4%	92.0%
(1024, 2048, 2048, 2048, 1024)	92.1%	74.9%	93.3%
(2048, 2048, 2048, 2048, 2048)	92.6%	75.2%	94.0%

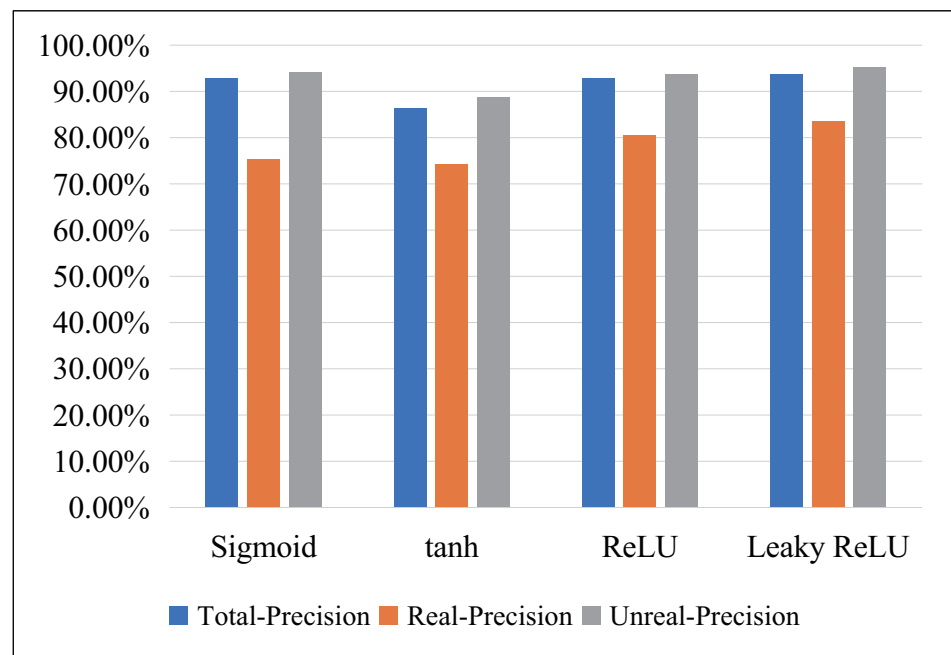


Figure 10. Comparison of experimental results with different activation functions.

Therefore, we used Leaky ReLU and Adam for the training process. The bidding performance of BM is shown in Figure 12. We divided the length of bidding sequences into five types (4 steps, 5–9, 9–12, 13–16, >16) for analysis. It is not difficult to find that the Unreal Precision in each type was higher than Real Precision because the training data of non-bids were more than ordered bids. When the length of the bidding sequence was 4–8, the overall precision can basically reach 95%. In general, the training effect of BM reached the expectation, which assisted the reinforcement training of EM.

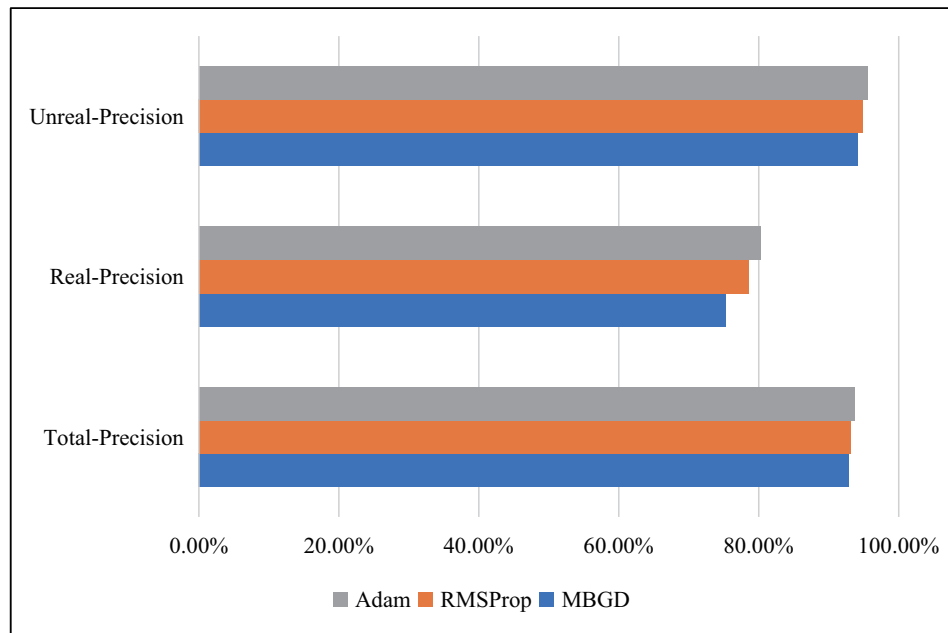


Figure 11. Comparison of experimental results with different optimizers.

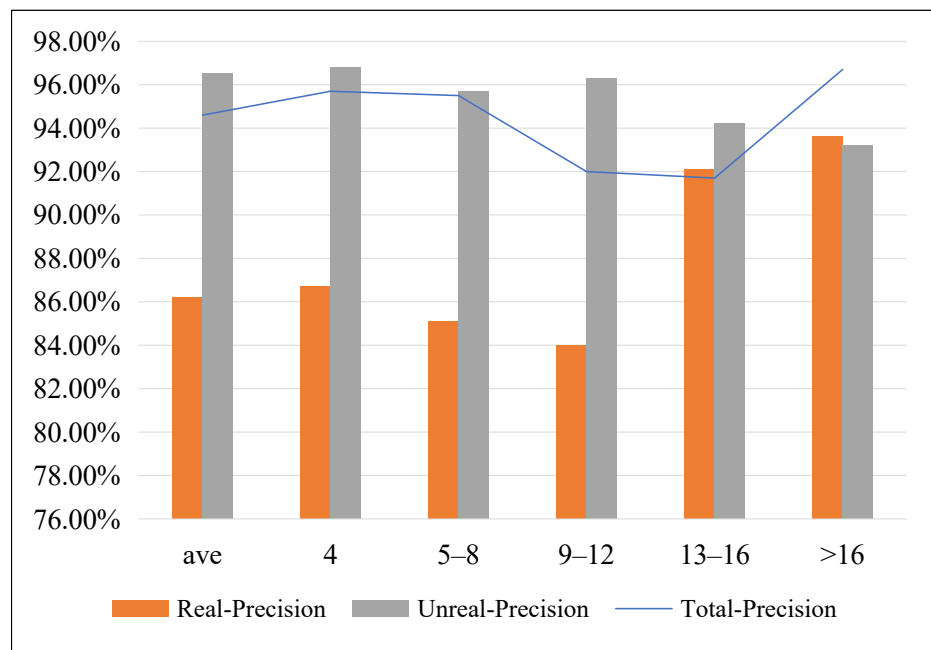


Figure 12. The performance of BM.

5.2.2. Evaluation of EM

Before the EM reinforcement training, this paper uses 2,500,000 training data as the pre-training initial parameters in the reinforcement learning stage. Figure 13 shows the effect of the pre-trained model. Except for the high accuracy and precision within the error range of 2, the other effects are not ideal. Therefore, it is necessary to continue intensive training to improve the performance of the evaluation network. The model was obtained after different reinforcement times, the pre-trained model was used for 10,000 rounds of double bridge games, and the average IMP value was calculated. The results are shown in Figure 14. The average IMP values are all greater than zero, indicating that the performance of the enhanced model is improved compared to the pre-trained model. With the increase in the number of iterations, the larger the average IMP, the better the effect, and an effect in the early stage is better than one in the later stage. After 1,000,000 rounds of iterations,

the average IMP of 10,000 decks of cards reached over 0.7, and the performance was greatly improved after strengthening.

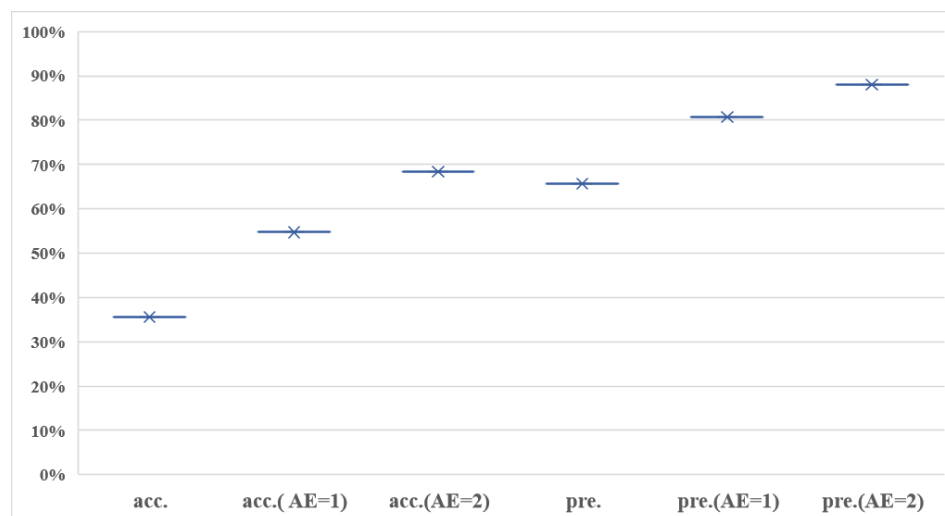


Figure 13. The performance of pre-trained EM.

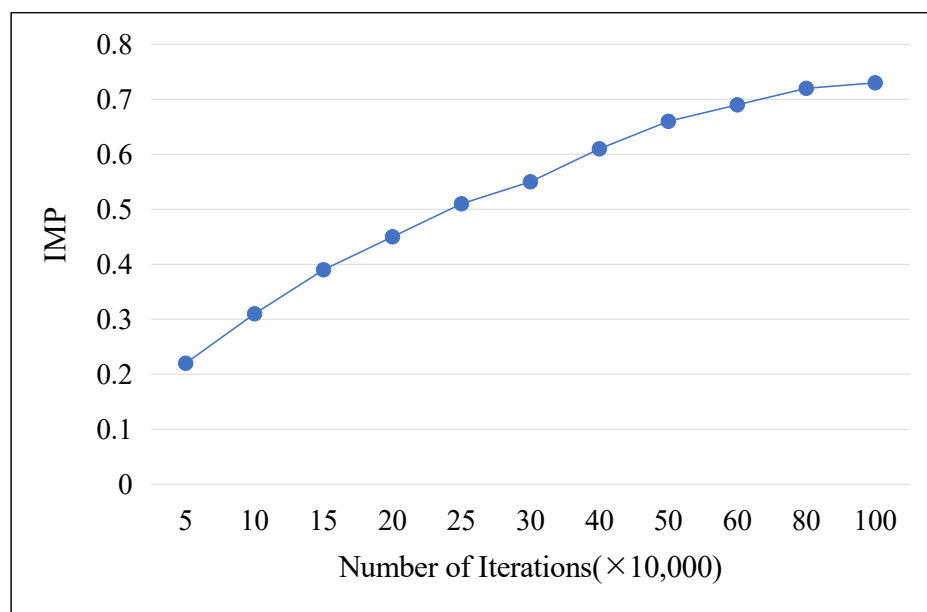


Figure 14. Average IMP in duplicate bridge tournaments.

5.3. Multi-System Bidding

The ultimate goal of the bidding system based on double neural network synergy is to realize multi-system bidding. We used this system to play separately against natural and precise bidding system players. Figure 15 shows the bidding process and finally the complete normal bidding, which showed that our bidding system realized the multi-system bidding.

```

Initialization...
Load BM...
Load EM...
Select position: 1. NS 2. EW
2
Please select the opponent's bidding system: 1. Natural... 2. Precision...
1
Select opener: 1.N 2.E 3.S 4.W
1
Select vulnerability: 1.None 2.NS 3.EW 4.ALL
1
Please input the hand cards for E: 72.J75.Q3.AJT632
Please input the hand cards for W: T94.AKT32.AJ52.5

START BIDDING
N: input N's bid: 1D
E: output: 1H
S: input S's bid: 1S
W: output: 2H
N: input N's bid: 2S
E: output: 3H
S: input S's bid: P
W: output: P
N: input N's bid: P
Bidding sequence(1D; 1H; 1S; 2H; 2S; 3H; P; P; P)
END

Initialization...
Load BM...
Load EM...
Select position: 1. NS 2. EW
2
Please select the opponent's bidding system: 1. Natural... 2. Precision...
2
Select opener: 1.N 2.E 3.S 4.W
1
Select vulnerability: 1.None 2.NS 3.EW 4.ALL
4
Please input the hand cards for E: 763.KJ873.A3.T94
Please input the hand cards for W: QJ94.AQT2.KQ54.8

START BIDDING
N: input N's bid: 2C
E: output: X
S: input S's bid: 3C
W: output: 3H
N: input N's bid: P
E: output: 4H
S: input S's bid: P
W: output: P
N: input N's bid: P
Bidding sequence(2C; X; 3C; 3H; P; 4H; P; P; P)
END

```

(a)
(b)

Figure 15. Example of System Testing. (a) Bidding with Natural System. (b) Bidding with Precision System.

6. Conclusions

This paper designed a bidding model that supports multi-system bidding based on the synergy of double neural networks. To reduce the influence of the historical bidding sequence on the unity of the system, this paper used a logic algorithm to convert the historical bidding sequences into 30 kinds of general feature information in advance. Then directly learned these features to obtain a bidding model that was not constrained by a single system. The model mainly consisted of two parts, the bid selection network and the state evaluation network. The bid selection network can obtain the optional probabilities of all bids in a certain state. On this basis, a state evaluation network was used to directly evaluate the influence of these bids on the whole situation and make a final decision according to the evaluation effect. In the end, our bidding model was able to play against different players' systems and worked as expected.

In future research, we will further consider the representation of historical bidding sequences to improve the strength of our bidding model. The feature extraction algorithm in this paper may not have been able to extract some bidding systems correctly. For this problem, we will try to use the coding network for research. Besides, in the reinforcement learning of EM, the learning rate of supervised learning had not been compared and optimized in detail. We will optimize the neural network update algorithm in reinforcement learning first to ensure enough update times.

Author Contributions: Conceptualization and methodology, X.Z. and R.L.; software, X.Z. and Y.B.; validation, X.Z. and Y.B.; writing—original draft preparation, X.Z.; writing—review and editing, X.Z. and R.L.; supervision, F.Y.; project administration, R.L. and F.Y.; funding acquisition, F.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Funds for Creative Research Groups of China under No. 61921003 and the key research and development in Jiangxi Province under 20212BBE51002.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work is supported by Snyrey Bridge Company. We thank the company's researchers.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hsu, F.-H. IBM's deep blue chess grandmaster chips. *IEEE Micro* **1999**, *19*, 70–81.
2. Shannon, C.E. Programming a computer for playing chess. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **1950**, *41*, 256–275. [[CrossRef](#)]
3. Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, *529*, 484. [[CrossRef](#)] [[PubMed](#)]
4. Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. Mastering the game of Go without human knowledge. *Nature* **2017**, *550*, 354. [[CrossRef](#)] [[PubMed](#)]
5. Moravčík, M.; Schmid, M.; Burch, N.; Lisý, V.; Morrill, D.; Bard, N.; Davis, T.; Waugh, K.; Johanson, M.; Bowling, M. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science* **2017**, *356*, 508–513. [[CrossRef](#)] [[PubMed](#)]
6. Brown, N.; Sandholm, T.; Amos, B. Depth-limited solving for imperfect-information games. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 7674–7685.
7. Charandabi, S.E.; Kamyar, K. Using a feed forward neural network algorithm to predict prices of multiple cryptocurrencies. *Eur. J. Bus. Manag. Res.* **2021**, *6*, 15–19. [[CrossRef](#)]
8. Jafari Gukeh, M.; Moitra, S.; Ibrahim, A.N.; Derrible, S.; Megaridis, C.M. Machine learning prediction of TiO₂-coating wettability tuned via UV exposure. *ACS Appl. Mater. Interfaces* **2021**, *13*, 46171–46179. [[CrossRef](#)] [[PubMed](#)]
9. Moayedi, H.; Aghel, B.; Vaferi, B.; Foong, L.K.; Bui, D.T. The feasibility of Levenberg–Marquardt algorithm combined with imperialist competitive computational method predicting drag reduction in crude oil pipelines. *J. Pet. Sci. Eng.* **2020**, *185*, 106634. [[CrossRef](#)]
10. Nasr, A.K.; Tavana, M.; Alavi, B.; Mina, H. A novel fuzzy multi-objective circular supplier selection and order allocation model for sustainable closed-loop supply chains. *J. Clean. Prod.* **2021**, *287*, 124994. [[CrossRef](#)]
11. Karimi, M.; Vaferi, B.; Hosseini, S.H.; Rasteh, M. Designing an efficient artificial intelligent approach for estimation of hydrodynamic characteristics of tapered fluidized bed from its design and operating parameters. *Ind. Eng. Chem. Res.* **2018**, *57*, 259–267. [[CrossRef](#)]
12. Ginsberg, M.L. GIB: Steps toward an expert-level bridge-playing program. In Proceedings of the IJCAI, Stockholm, Sweden, 31 July–6 August 1999; pp. 584–593.
13. Ginsberg, M.L. GIB: Imperfect information in a computationally challenging game. *J. Artif. Intell. Res.* **2001**, *14*, 303–358. [[CrossRef](#)]
14. Zhang, X.; Liu, W.; Yang, F. A Neural Model for Automatic Bidding of Contract Bridge. In Proceedings of the 2020 IEEE 22nd International Conference on High Performance Computing and Communications, Yanuca Island, Fiji, 14–16 December 2020; pp. 999–1005.
15. Zhang, X.; Liu, W.; Lou, L.; Yang, F. AI Enabled Bridge Bidding Supporting Interactive Visualization. *Sensors* **2022**, *22*, 1877. [[CrossRef](#)] [[PubMed](#)]
16. Rong, J.; Qin, T.; An, B. Competitive Bridge Bidding with Deep Neural Networks. In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, Montreal, QC, Canada, 13–17 May 2019; pp. 16–24.
17. Ho, C.Y.; Lin, H.T. Contract Bridge Bidding by Learning. In Proceedings of the AAAI Workshop: Computer Poker and Imperfect Information, Austin, TX, USA, 26 January 2015.
18. Yeh, C.K.; Hsieh, C.Y.; Lin, H.T. Automatic bridge bidding using deep reinforcement learning. *IEEE Trans. Games* **2018**, *10*, 365–377. [[CrossRef](#)]
19. DeLooze, L.L.; Downey, J. Bridge bidding with imperfect information. In Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Games, Honolulu, HI, USA, 1–5 April 2007; pp. 368–373.
20. Amit, A.; Markovitch, S. Learning to bid in bridge. *Mach. Learn.* **2006**, *63*, 287–327. [[CrossRef](#)]