

Article

AI Enabled Bridge Bidding Supporting Interactive Visualization

Xiaoyu Zhang ^{1,2} , Wei Liu ^{1,*}, Linhui Lou ^{1,2} and Fangchun Yang ^{1,2}

¹ School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China; xy_zhang@bupt.edu.cn (X.Z.); loulinhui.shey@bytedance.com (L.L.); fcyang@bupt.edu.cn (F.Y.)

² State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

* Correspondence: liuw@bupt.edu.cn

Abstract: With the fast progress in perfect information game problems such as AI chess and AI Go, researchers have turned to imperfect information game problems, including Texas Hold'em and Bridge. Bridge is one of the most challenging card games that have significant research value. Bridge playing is divided into two phases: bidding and playing. This paper focuses on bridge bidding and proposes a bridge bidding service framework using deep neural networks, and supports bidding visualization for the first time. The framework consists of two parts: the bidding model (BM) with a multilayer neural network, and a visualization system. The framework predicts not only reasonable bids from the existing bidding system of humans, but also provides intuitive explanations for decisions to enable human–computer information interaction. Experimental results show that this bidding AI outperforms majority of existing systems.

Keywords: bridge bidding; imperfect information; deep neural network; interactive visualization



Citation: Zhang, X.; Liu, W.; Lou, L.; Yang, F. AI Enabled Bridge Bidding Supporting Interactive Visualization. *Sensors* **2022**, *22*, 1877. <https://doi.org/10.3390/s22051877>

Academic Editor: Joaquin Ordieres Meré

Received: 12 January 2022

Accepted: 25 February 2022

Published: 27 February 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

Computational games have been important research to verify the development level of artificial intelligence. At present, Artificial Intelligence has seen several breakthroughs in perfect information games. Chess [1] and Go [2–4] are typical representatives in which players can obtain the whole state of games. The search space of chess can realize a global search under the current computing resources and find the perfect solution with computing power to defeat human beings. DEEPMIND has made a breakthrough in the research of Go. The AlphaGo-Zero algorithm does not require inputting human knowledge and can update the neural network through self-play, achieving the effect of defeating human experts [4]. Recently, computer programs defeated humans in 2-player and 6-player no-limit Texas Hold'em [5–9].

Nevertheless, imperfect information games such as Mahjong [10], DeltaDou [11,12], and Bridge, where players cannot obtain the whole state of games, are still very challenging. Existing methods for resolving perfect information games do not directly apply to imperfect information games. However, most decision-making problems in the real world are under imperfect information, so it is necessary to study the imperfect information games. Hence, this paper focuses on a typical imperfect information game—Bridge.

Bridge uses standard 52 cards to compete, and the four players are usually considered East, West, South, and North. East–West and South–North teams confront, and finally get high scores. Bridge is mainly divided into two phases: bidding and playing. In the bidding phase, four players from two teams use intra-team collaboration and inter-team confrontation to transmit hand information through bidding to determine the final contract, which influences the final score. The four players each hold a random 13 cards and can only see their own 13 cards. There are 6.35×10^{11} possible hand holdings [13]. Then, players used 38 bids to bid in clockwise order. All bids are divided into three non-bids

(PASS, DOUBLE, and REDOUBLE) and a set of orderly bids: { 1♣, 1♦, 1♥, 1♠, 1NT, 2♣, ..., 7NT }. The PASS means that the player skips this round of bidding; the DOUBLE or REDOUBLE means that the player's score needs to be doubled. The bidding proceeds until it is terminated by three consecutive PASS, at which time the last bid is regarded as the contract of the bidding. The contract consists of two parts: a number and a symbol. The number indicates that the winning team needs to win at least additional rounds to complete the contract during the subsequent playing stage, and the symbol indicates the trump suit.

In the playing phase, the four players play a card in clockwise order. A total of 13 rounds of cards will be played. The players use certain skills to win as many rounds as possible, trying to complete their contract or prevent the opponent from completing the contract. Although several algorithmic techniques have been successfully applied to the playing phase, bidding needs a different approach. This paper focuses on Bridge bidding.

Bridge bidding can be divided into two sub-problems: the bidding without competition [14–16] and the bidding with competition [17]. The sub-problem of the bidding without competition assumes that the opponent always calls PASS when bidding, so the exchange of information between teammates will not be prevented. Nevertheless, this situation does not conform to the real competition. Hence, we focus on the sub-problem of the bidding with competition, which means that each AI player may bid and exchange information as in real bridge games. The main contributions of our work are summarized as follows:

- We trained a bidding model (BM) based on deep neural networks, which can learn the human bidding system with only expert data and no manual input of rules. It can achieve human-machine competition.
- We designed a visualization system that can help beginners learn more rules in the bidding system.

The rest of this paper is structured as follows. Section 2 discusses related work. Section 3 proposes the model framework and describes the visualization system in detail. This whole project is hosted at <https://youtu.be/x0VVyg8C2i4> (accessed on 1 October 2020). Then we introduce experimental setups and present the results with evaluation in Section 4. Finally, we conclude our work in Section 5.

2. Related Work

The research on Bridge began in the 1980s. In the first World Computer Bridge Championship in 1997, the program Bridge Baron [18], written by Tom Throop, won the championship. In the second year, the GIB program developed by Matthew Ginsberg won the championship [19]. In the following years, Jack and Wbridge5 <http://www.wbridge5.com/> (accessed on 1 October 2020) took turns to win the championship [20]. These traditional Bridge AIs have manually input complicated bidding rules into computer programs. However, since the number of the possible bidding sequences (see the analysis at <http://tedmuller.us/Bridge/Esoterica/CountingBridgeAuctions.htm> (accessed on 1 October 2020)) is as high as 10^{47} , it is hard for these human-designed rules to cover all the situations, which leads to ambiguous bidding rules. In addition, there may be multiple conflicting interpretations for a bid. Human players typically spend much time practicing with their peers to improve mutual understanding and minimize bids' ambiguity. However, it is highly challenging for AI players to resolve ambiguity by replicating human bidding systems to achieve the same mutual understanding that human players can achieve, which puts them at a disadvantage in the bidding phase.

In order to overcome the shortcomings mentioned above, some researchers have made relevant attempts to improve the human bidding system. For example, Amit et al. [13] proposed a Monte Carlo sampling approach based on the human bidding system to solve the ambiguity of bidding by building a decision tree model. DeLooze and Downey [21] used a human bidding model to generate a large amount of training data and clustered the bidding process utilizing self-organizing mapping for ambiguity resolution. Nevertheless,

this model can only be used to bid no trump hands effectively. Human bidding systems play a central role in AI players' bidding in those works. Ho and Lin [14] proposed a learning framework based on the UCB algorithm that did not rely on the human bidding system, which demonstrated the possibility to learn to bid directly in a data-driven manner. Unfortunately, the study is somewhat limited by the decision tree model with a limit of up to five choices per bid, which does not correspond to the actual bidding situation. Yeh and Lin [15] first tried using deep reinforcement learning to bid without depending on any human expert knowledge but did not consider the bidding with the competition. In [17] and [22], they all begin to focus on the sub-problem of the bidding with competition.

We find that decision making has been applied in various fields like economics [23] and robot applications [24]. There are many papers using different techniques, such as machine learning and deep learning, for decision making. Most of the above Bridge bidding algorithms and models assist players in making decisions, but they are difficult to explain the basis behind the decision. Therefore, we extend some ideas from [22] and propose an AI bidding framework that can not only predict bids but also provide intuitive explanations for decisions made to help human–computer information interaction.

3. Framework

In this section, we introduce the details of our proposed framework, shown in Figure 1. The framework includes the bidding model (BM) and a visualization system. BM can predict the probability distribution of bids under the natural bidding system and output the characteristics conveyed by other players' bids in the current bidding process. The visualization system can more intuitively display the prediction results of the characteristics of other players' hands.

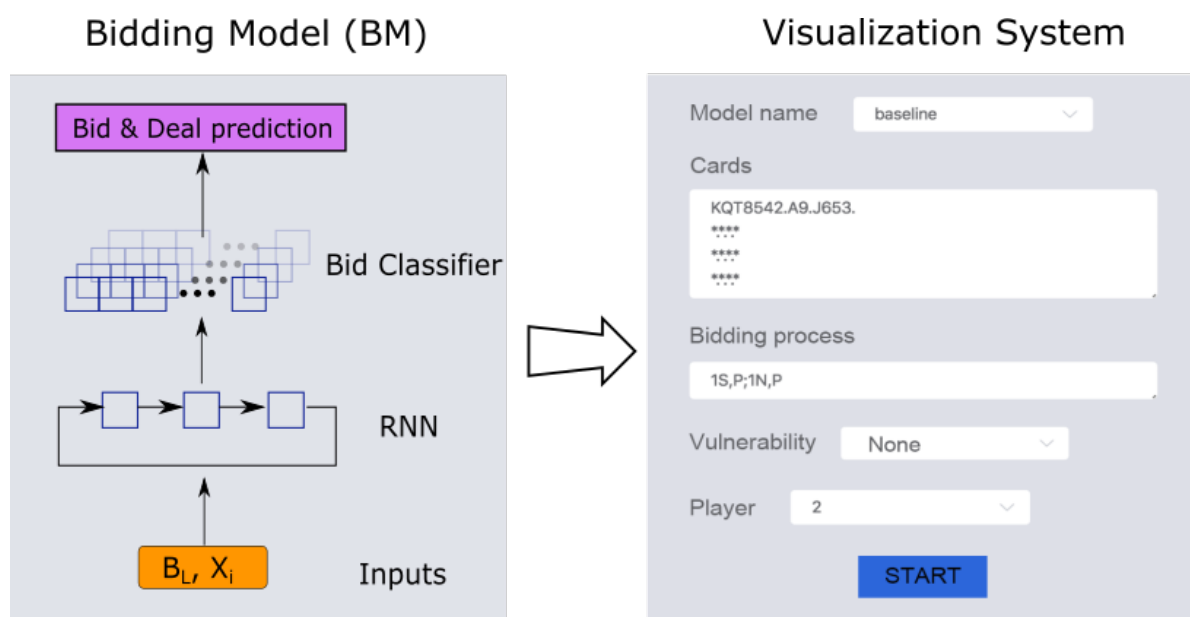


Figure 1. An overview of the proposed framework. It includes two parts: the bidding model and the visualization system.

3.1. Bidding Model (BM)

Bridge is a game of teamwork in which two players of a partnership use the same bidding system to exchange information. The bidding system is composed of many rules that give each player the meaning of every possible bid, which is equivalent to providing a communication language that allows players to use bids to exchange information about the cards they hold. Therefore, we use 4 million pairs of real bidding data provided by the Synrey platform for human–machine confrontation under the natural bidding system to train a bridge bidding model. The model is based on expert experience that supports the

natural bidding system. Obviously, a single bid cannot predict the next bid in the bidding phase. The current historical bidding sequence can directly affect the next bidding. Hence, we choose Recurrent Neural Network (RNN) as the main body of our model. The specific structure of the BM is shown in Figure 2, which consists of two parts, one is a bidding prediction system based on imitation learning, and the other is a visualization system for hands understanding. This visualization system can help players transfer hand information to each other and evaluate the value of the hands to help predict bidding more accurately.

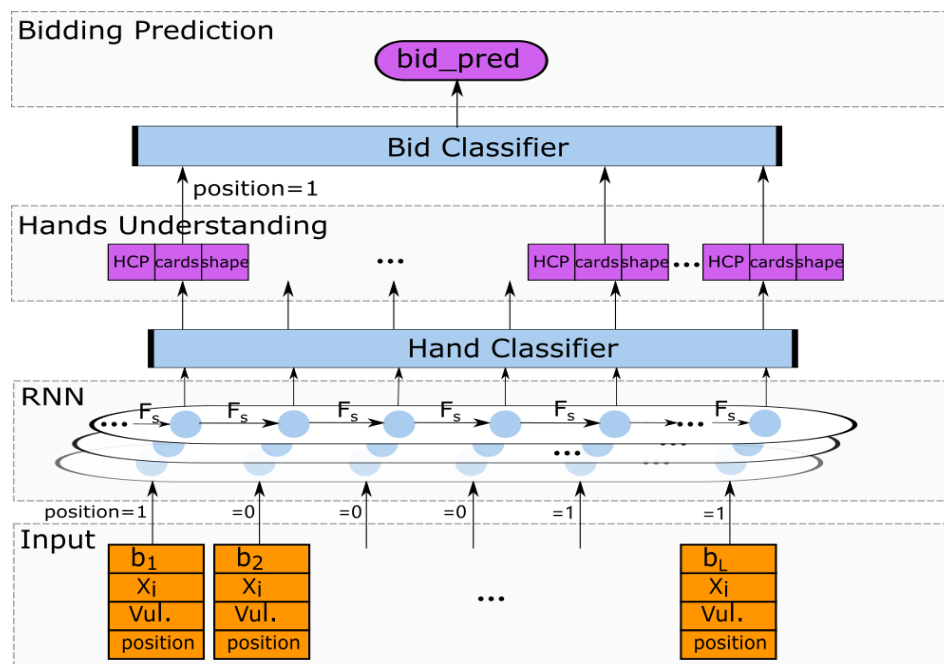


Figure 2. The structure for BM.

BM contains a three-layer stacked Recurrent Neural Network(RNN) and multi-layer fully connected neural network. To facilitate the description, we first define some symbols shown in Table 1. The input layer accepts the player i 's hands X_i , bidding action b_L , vulnerability v and position p . After getting the embedding vectors, we concatenate them to be the input of a three-layer stacked RNN. The hidden state s is finally encoded by the RNN, which is as follows:

$$\begin{aligned}
 s &= F_s(X_i, B_L), \\
 B_L &= \{b_L, v, p\}, \\
 i &= 1, 2, 3, 4; L = \{1, 2, \dots, n\},
 \end{aligned}
 \tag{1}$$

where $F_s(\bullet)$ is a recursively called nonlinear activation function of the recurrent neural network. B_L represents the L th bid in a bidding sequence.

Simultaneously, the outputs of the hidden states encoded by RNN are fed into the multi-layer fully connected neural network, i.e., MLP (Multi-Layer Perception). We then define function $F_{mlp}^H(\bullet)$, $F_{mlp}^{Csuit}(\bullet)$, $F_{mlp}^Y(\bullet)$ and $F_{mlp}^G(\bullet)$ for four types of MLP units. The overall formula in features of external information are as follows.

$$\begin{aligned}
 s_H &= F_{mlp}^H(s), \\
 s_{Csuit} &= F_{mlp}^{Csuit}(s), \\
 s_Y &= F_{mlp}^Y(s), \\
 s_G &= F_{mlp}^G(s, s_H, s_{Csuit}, s_Y) \times P_{mask}^B(L),
 \end{aligned}
 \tag{2}$$

where $suit = \{\clubsuit, \diamondsuit, \heartsuit, \spadesuit\}$.

Table 1. Parameters for BSM.

	Type	Length
Input	The hands' distribution of Player i : X_i	one-hot
	The bid in L round: b_L	(1*256)
	Vulnerable: v	(1*4)
	Is the player to bid: p	(1)
	check rules	(1*41)
Output	Bidding prediction	(1*41)
	The distribution of HCP: P^H	(1*38)
	The distribution of teammate's hands: P^C	(4*13)
	The distribution of cards kept by players: P^Y	(3*52)

In order to prevent the violation of bridge rules in the bidding prediction, we set up a rule filter vector P_{mask}^B . When a bid violates the bridge rules in a certain state, we define the weight of such bid to 0 so that the rule of Bridge can be gradually learned. We define $P_{mask}^B(L)$ as follows:

$$P_{mask}^B(L) = \begin{cases} 1 & \text{the } L\text{th bid is legal,} \\ 0 & \text{the } L\text{th bid is illegal.} \end{cases} \quad (3)$$

We will verify the importance of the rule vector in Section 4.2.

The output layer of BM consists of the hands understanding and the bidding prediction. The hands understanding system has three outputs: the probability distribution of the HCPs of the teammate $P(H = h|X_i, B_L)$, the probability distribution of the number of cards of each suit $P(C^{suit} = c|X_i, B_L)$ and the probability distribution of the hands of the other three players $P(Y_i^p = 1|X_i, B_L)$. High Card Points (HCPs) are total points of a full deck of cards. A = 4, K = 3, Q = 2, J = 1. The total number of HCPs is 40, and the expected number of HCPs in the given hands is 10.

$$\begin{aligned} P(H = h|X_i, B_L) &= P(H = h|s_H) \\ &= \text{softmax}(w_h, s_H) \\ &= \frac{\exp(w_h, s_H)}{\sum_{h=0}^{37} \exp(w_h, s_H)}, \\ P(C^{suit} = c|X_i, B_L) &= P(C^{suit} = c|s_{csuit}) \\ &= \text{softmax}(w_{csuit}, s_{csuit}), \\ &= \frac{\exp(w_{csuit}, s_{csuit})}{\sum \exp((w_{csuit}, s_{csuit}))}, \\ P(Y_i^p = 1|X_i, B_L) &= P(Y_i^p = 1|s_Y) \\ &= \frac{\exp(w_i^p, s_Y)}{1 + \exp(w_i^p, s_Y)}. \end{aligned} \quad (4)$$

By combining Equations (1), (2) and (4), we get the objective function I of the hand understanding as the manually specified feature.

$$\begin{aligned} I(X_i, B_L) &= P(H, C, Y|X_i, B_L) \\ &= \{P(H|X_i, B_L), P(C|X_i, B_L), P(Y|X_i, B_L)\} \end{aligned} \quad (5)$$

Note that the primary purpose of bridge bidding is to find the best possible contract and convey some information to partners. So, the bidding prediction system describes a bid probability distribution that conforms to the rules of the system based on the features

coded by the model and the manually specified feature, in which the objective function G is as follows:

$$\begin{aligned}
 G(X_i, B_L, I) &= P(b_{L+1} = \alpha | X_i, B_L, I) \\
 &= P(b_{L+1}) \\
 &= \text{softmax}(w_G, s_G) \\
 &= \frac{\exp(w_G, s_G)}{\sum_{h=0}^{37} \exp(w_G, s_G)}.
 \end{aligned}
 \tag{6}$$

Figures 3 and 4 respectively show examples of model input and output. We can see that the model’s input is the player’s hands, bidding action, check rules, vulnerability, and position, followed by three-layer RNN. A 52-dimensional vector represents the player’s hands, and a 41-dimensional vector is used to check rules that can ensure the bidding sequence’s rationality. We also add the vulnerability to the input for better score calculation. Besides this, one of the most important factors is the position, which means whether one’s turn to bid. The importance of this factor will be demonstrated in the experiment later.

	52 bits	41 bits	41 bits	4 bits	1 bit
input	one’s own 13 cards	bidding action	check rules	vulnerability	position (whether one’s turn to bid)
	A.92.AQ862.KQ842	1D		None	
input	10000...01000101	00010...0 00 000	11100...1 10 011	1000	1
	52 bits for cards	41 bits for bidding action	41 bits for check rules	4 bits for vulnerability	

Figure 3. An example of model input.

	4×52 floats		41 floats
deal output	prediction of probability distribution of all players’ cards	partner’s HCP output	prediction of probability distribution of partner’s HCP
	4×14 floats		41 floats
partner’s card amount output	prediction of probability distribution of partner’s card amounts in different suits	bid output	prediction of bidding action
	A.92.AQ862.KQ842 KT74.KQ754.743.9 Q9865.J3.KT.AJT7 J32.AT86.J95.653		
deal output	0.9 0.1 0.2 ... 0.8 0.2 0.9 0.1 0.95 0.1 ... 0.2 0.05 0.1 0 0.1 0.85 ... 0.3 0.25 0.05 0.2 0.1 0.2 ... 0 0.9 0.1	partner’s HCP output	11 HCP 0 0 ... 0.1 0.6 0.2 ... 0 0
	5 cards in S, 2 cards in H, 2 cards in D, 4 cards in C		
partner’s card amount output	0 0 0 0 0.2 0.5 ... 0 0 0 0 0.1 0.6 0.3 0 0 ... 0 0 0 0 0.3 0.4 0.3 0 0 ... 0 0 0 0 0 0.1 0.2 0.5 0.2 ... 0 0 0	bid output	1S 0.1 0.8 0 0 0 ... 0 0 0

Figure 4. An example of model output.

3.2. Visualization System

According to the rules of Bridge, players have the right to let their opponents explain the meaning of their current bids. For human players, explaining their decisions is very simple, but in a human–computer game, how do the robots explain the meaning of their decisions to humans. Figure 5 is a visualization system framework we designed, which is mainly divided into three parts: web front-end server, web back-end server, and GPU server. The user first accesses the visual page in the web front-end server, enters the player’s own hands and the current bidding process, and sends the task to the web back-end server. The web back-end server sends the task to the task queue, and all GPU nodes will process the task after competitively obtaining the task. Then, after the GPU node obtains the prediction result, it stores the result in the result queue, and finally, the web back-end server will get the corresponding result from the result queue and send it to the front-end server to present it to the user.

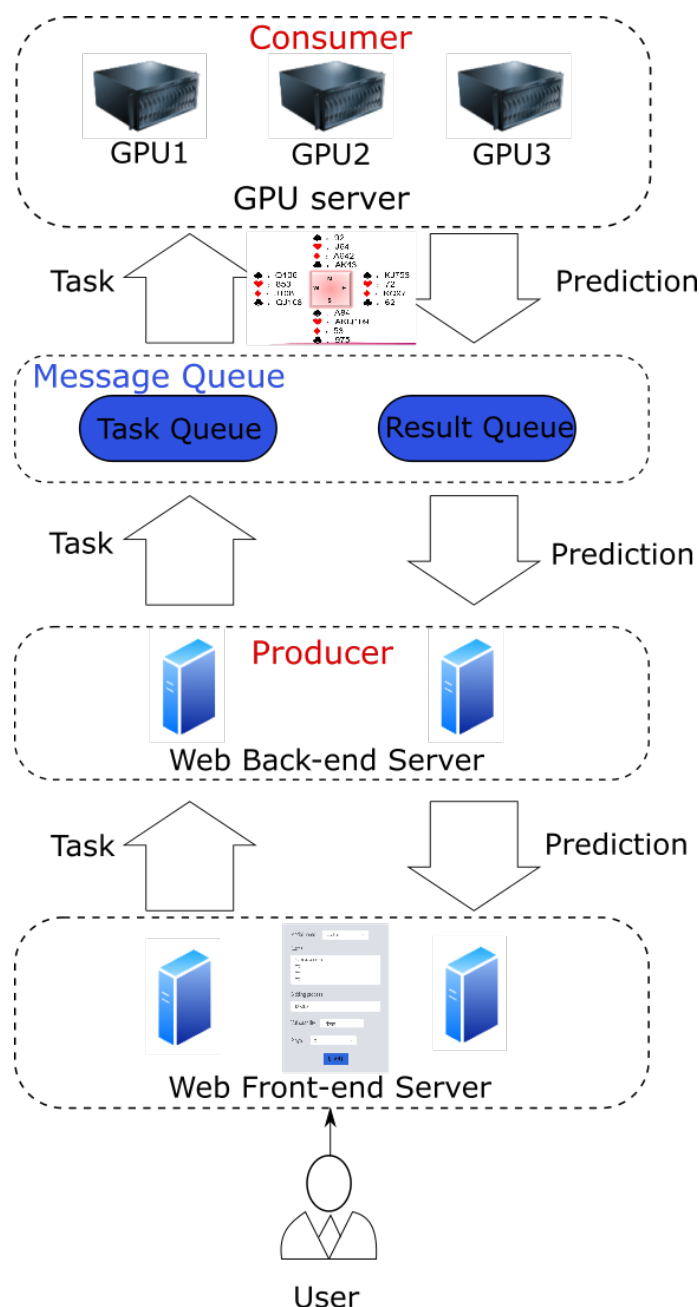


Figure 5. General architecture of visualization system.

The web front-end provides a user interface to facilitate sending user tasks to the web back-end server, and at the same time, it can display the final returned results to the user. The web back-end server needs to provide APIs for the web front-end to call. It is responsible for interacting with the GPU server and sending tasks input by the user to the GPU server. The bidding model on the GPU server makes predictions and returns the prediction results to the web back-end. The web back-end server not only needs to be able to perform a simple verification of user input but also has a specific error handling mechanism to deal with the situation where the GPU server does not respond. The specific functions of each part are shown in Figure 6.

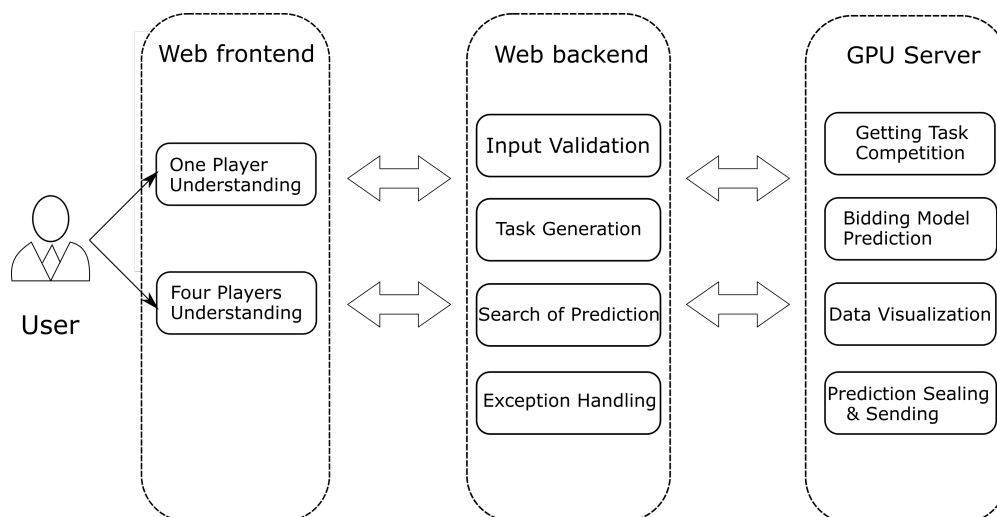


Figure 6. The specific functions of each part.

3.2.1. Web Back-End Server

The web back-end server is an intermediate module that interacts with the GPU prediction node and the web front-end server. After getting a task, the server will first verify the validity of the data. After the verification is completed, the web back-end server will generate a universally unique identifier for each task to mark it. It is mainly responsible for verifying and encapsulating the user input data. The data with the wrong format will directly return the error message and then send the data to the task queue. The data in the result queue is sent to the web front end for display and then put in the task queue. After being put into the task queue, the server will immediately start polling the result queue to find whether the prediction result corresponding to the task ID appears in the result queue until it times out or gets the task's predicted result. If it times out, it will return the task timeout error message. If the prediction result is obtained, the corresponding prediction result is returned.

Since the web server needs to obtain the predicted results from the result queue through polling when querying, each query is synchronized and time-consuming. Because the Django framework is single-threaded, it does not support simultaneous processing of multiple requests, so a single synchronous request takes too long to cause other requests to accumulate, and even requests are accumulated on the web server, causing a large number of GPU nodes to become idle. This article chooses to use uwsgi [25] to deploy the Django project so that Django supports multiple processes to process multiple requests at the same time so that the web server can process multiple requests at the same time and improve the resource utilization of GPU nodes.

3.2.2. GPU Server

The GPU server is the system's core module that uses the deep learning model to make predictions. It consists of model loading, processing prediction tasks, and data visualization. Because of the enormous computing power required, it needs to support

a flexible horizontal expansion method. Since most deep learning frameworks do not support multithreading and multi-process, in order to ensure the flexibility of deployment, the system adopts the following two design methods:

1. The GPU server comprises GPU prediction nodes one by one. Each node corresponds to a unique GPU determined when the node is started. It will not change until the node is shut down. If the GPU memory capacity allows, each GPU is bound to any number of nodes. The above model (BM) is loaded on each GPU node to perform bid prediction tasks. This one-to-many relationship between GPU and GPU nodes has the following two advantages:
 - This allows a single GPU to run prediction tasks on multiple nodes simultaneously, which effectively improves GPU computing performance.
 - The one-to-one architecture only executes one prediction task at a time. If this prediction task takes too much time, it is likely to block the subsequent prediction tasks, causing many tasks to accumulate in the message queue. Therefore, the one-to-many architecture alleviates the blocking mentioned above.
2. The web server and GPU nodes do not interact directly through APIs, but they are decoupled through message queues to form a producer-consumer model, which brings the following advantages:
 - The GPU nodes and the web back-end server are completely decoupled, and the services on both sides can be deployed and expanded freely.
 - The distributed nature of the system is stronger and effectively reduces the accumulation of tasks in the message queue by increasing the number of consumers.
 - Ensuring the reliability of the system. If the consumer nodes restart, the failed messages will stay in the message queue instead of disappearing directly, which effectively enhances the system's reliability.

4. Experiment

In this part, we conducted a series of experiments to evaluate our AI. We first verified the reliability of the experimental data. Next, we evaluated the performance of BM in detail and showed the results of the visualization system with examples. Finally, we compared the proposed model with a state-of-the-art model and a well-known computer bridge software, Wbridge5, which won the computer bridge championship for several years.

4.1. Dataset

In the experiment, we used the Snyrey platform robot to generate a large amount of club-level data in the natural bidding system. Currently, the Snyrey platform has accumulated nearly 500 million historical bidding records and playing with double-dummy hands. We randomly generated nearly 4,000,000 instances of real bidding as our dataset, 70% of which are used as training data for BM, 20% of which as a validation set, and the remaining data as testing data. Each instance contains the real hands of four players, the historical bidding sequence, and the double-dummy hands' results. Figure 7a shows that the probability of each card in each player is roughly the same. The distribution of lengths of bidding sequences in the data set are depicted in Figure 7b, which presents a normal distribution. Most of the bids are concentrated at the end of 8~13, and the probability of too long or short bidding sequences is very low.

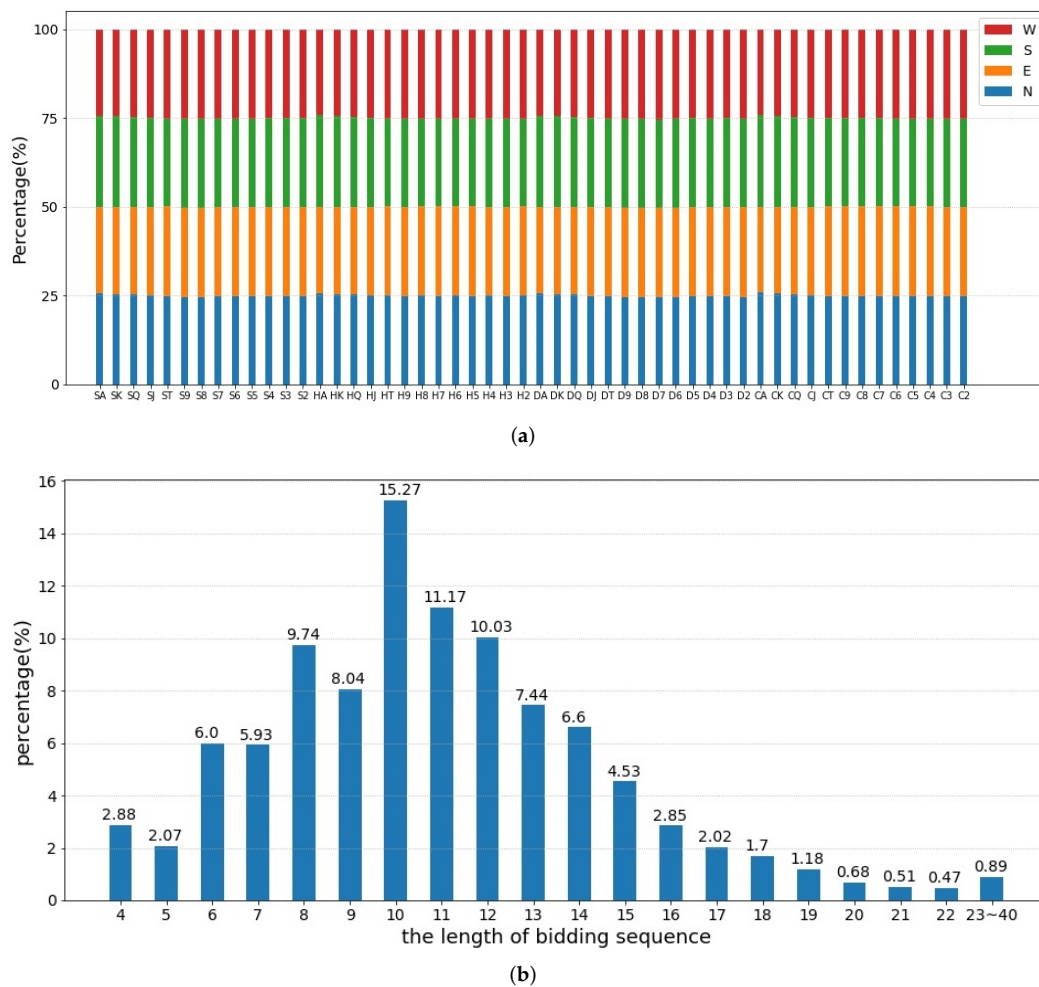


Figure 7. Dataset Evaluation. (a) The percentage of each card among four players. (b) The distribution of lengths of bidding sequences.

4.2. Performance Analysis of BM

We designed BM based on deep neural networks. Our models were run on a single GPU. We run all the models up to 200,000 epochs. We selected accuracy, precision, and recall as the evaluation criteria to analyze the predicted results of the bid and the non-bid. As shown in Figure 8, the precision of the bid is higher than the recall, and the recall of the non-bid is higher than the precision. It can be seen that the predicted results of the model are more biased towards predicting non-bid. This is because there are fewer types of non-bid in the training dataset while there are more types of bids.

In Section 3.1, we designed a rule vector $P_{mask}^B(L)$ to ensure the legitimacy of the output. Next, we verify whether the rule vector affects all bids and bids under the condition that the training data, testing data, and other parts of the model remain unchanged. The experimental results are shown in Figure 9. Since PASS (one of the non-bid) accounts for a large proportion of all bids, in addition to the overall accuracy of all bids, the accuracy of bids (except non-bid) is also added to the evaluation criteria at this stage. It is not difficult to see that, whether it is the overall accuracy of bids or the accuracy of bids (except non-bid), the results of the models with the rule vector are all better than the results of the models without the rule vector. Meanwhile, the model that adds the rule vector converges faster. In a word, the overall accuracy of our model reached 89%.

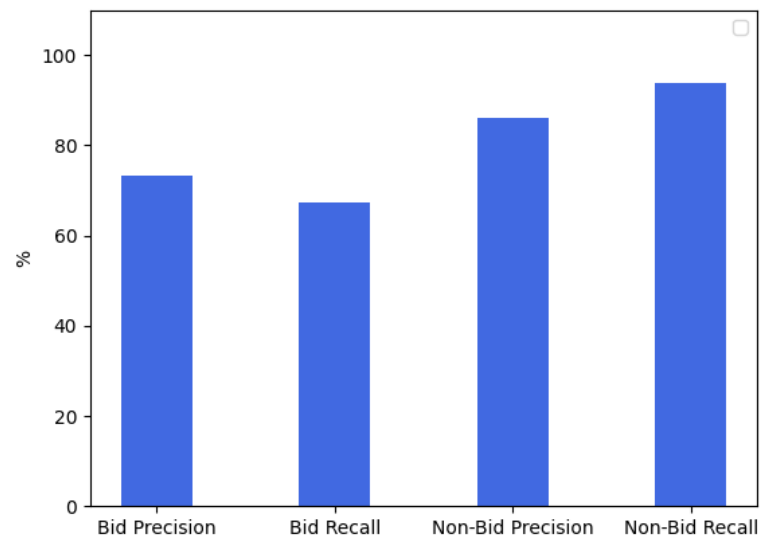


Figure 8. Performance Analysis of BM.

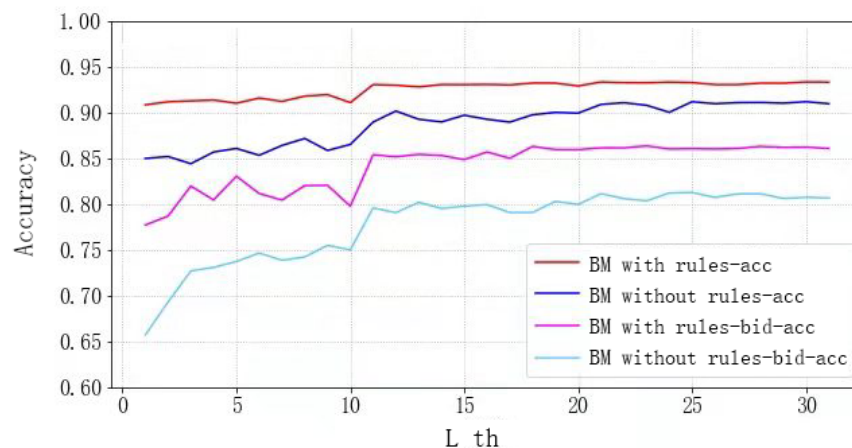


Figure 9. Accuracy of different structures.

4.3. The Evaluation of Visualization System

The existing bridge AI programs cannot explain the meaning of their bids. In order to better match the real bridge game and realize the real human–computer interaction, we have made a visualization system that can reflect the distribution of bid prediction, player hands, HCPs, and the number of the card—taking an example to illustrate it. We select a representative bidding process to analyze and verify the visualized results and evaluate whether the visualized results can correctly express BM’s understanding. For ease of presentation, in the analysis of experimental results, C, D, H, S are uniformly used to represent suits ♣, ♦, ♥, ♠, and N represents no-trump card. The hands of each player in the test case were represented by a string separated by dots representing the cards of the suits ♠, ♥, ♦, ♣ held by a player.

Table 2 shows that the North player first opened 1D. In the natural system, 1D is an open-bid, which means that the HCPs exceed 12 points and have four cards with ♦. According to the natural bidding system, we can see from the hands that the North player has 13 HCPs. Thus, 1D should be opened. When the teammate of the North player, the South player, receives this open-bid, he or she should understand the above information. Figure 10a shows the probability distribution of average HCPs of the North player predicted by the South player after getting the open-bid 1D. It is not difficult to see that the South player indicates that the probability of the North player’s HCPs is between 12–14, which conforms to the rules of the natural system.

Table 2. Test Case. The actual situation is that the North and South players have nine cards with ♠, nine cards with ♦ and the East and West have 11 cards with ♥. The hands of both sides are very obvious. The North and South players prefer the contract called ♠ or ♦. It is biased towards the contract called ♥. After fierce bidding, the West player is the declarer, and the contract is 7H.

CardS	North East South West	63.8.KJT932.AK63 Q2.AJT92.Q854.JT AKJT985.6.A76.84 74.KQ7543..Q9752
VUL	NS	
Bid Sequence	N:1D, 1H, 1S, 4H, P, P, 4S, 5H, P, P, 5S, 6H, P, P, 6S, 7H, P, P, X, P, P, P, END Declare:West; The final contract:7H	

Next, let us analyze the visualization results generated when the contract is finally obtained. Figure 10b shows that the West player has 0–2 ♠ cards, 6–7 ♥ cards, 0–2 ♦ cards and 3–5 ♣ cards, which is close to the real cards of the West player (2 ♠ cards, 6 ♥ card, 0 ♦ cards, 5 ♣ cards). Judging from the prediction results of the distribution of the West player (Figure 10c), there is a high probability of having a lot of ♥ cards. Furthermore, the West player might have the feasibility of a K ♥ card. In fact, the West player does own the K ♥ card. In brief, the above examples can prove that the visualization system can predict the information transmitted through the bidding and reasonably display hands understanding for players to communicate.

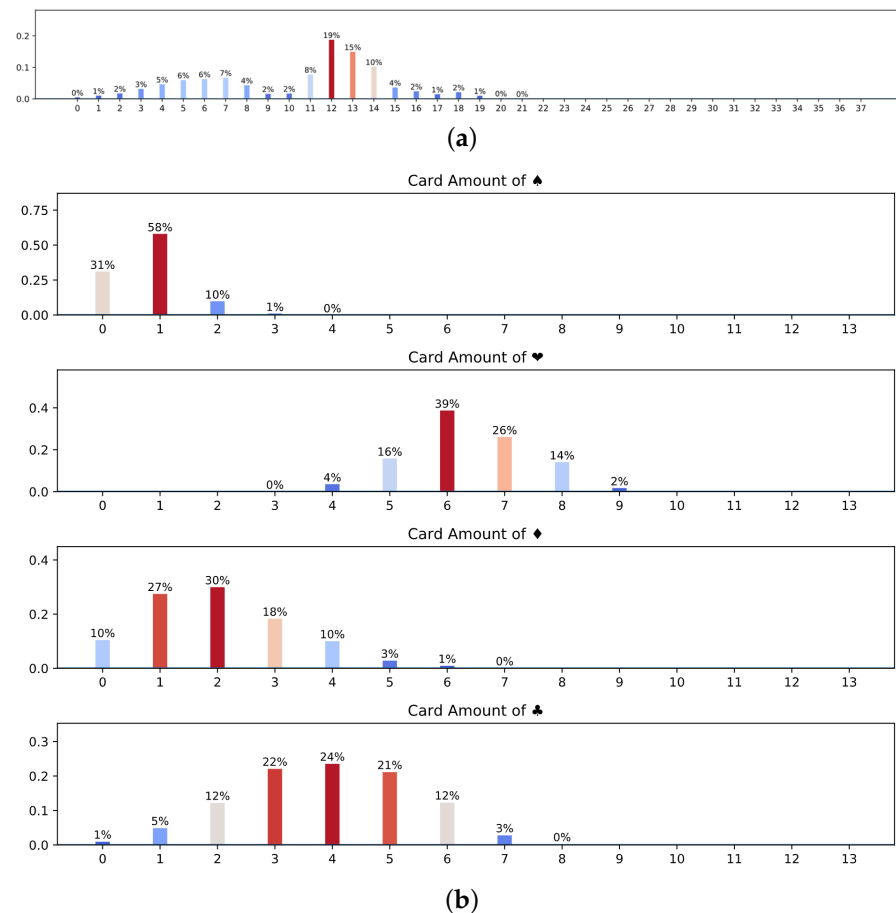


Figure 10. Cont.

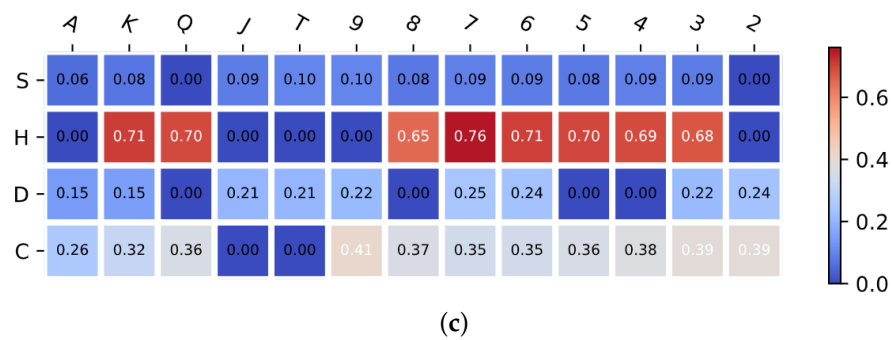


Figure 10. The visualization system. (a) Average HCPs of North player; (b) The distribution of the number of cards in different suits of West players; (c) The distribution of West player's hands. We show the hands distribution in the form of a heat map. The horizontal axis of the heat map represents the size of the cards, and the vertical axis represents the suit of the cards. The darker the color, the higher the probability of owning the card in hand. It is evident that the West player has more cards in the suit of H(♥).

4.4. Comparison with State-of-the-Art

We compared the bridge bidding result with that in the work of [15]. We chose their different deep reinforcement learning models with layer = 2, 3, 4, 5, and ran the same 20,000, 140,000, 200,000 data. Since they studied the problem of bidding without competition, we provided a method based on the difference between the predicted IMP https://www.acbl.org/learn_page/how-to-play-bridge/how-to-keep-score/teams/ (accessed on 1 October 2020) and the best contracted IMP, which can be used to evaluate the bidding strength. Negative numbers indicate that the predicted results are inferior to the best results. The smaller the absolute value, the more accurate the prediction. The result in Table 3 demonstrates that our model has the best performance among all models, although our model is simpler without using reinforcement learning.

Table 3. Comparisons of the average difference of IMP between our model and the different models in [15].

Data Size	Our	[15] Layer = 2	[15] Layer = 3	[15] Layer = 4	[15] Layer = 5
20,000	−0.208	−2.213	−2.679	−5.213	−6.312
140,000	−0.163	−2.120	−2.558	−5.312	−6.136
200,000	−40.174	−1.986	−2.636	−5.232	−6.154

Meanwhile, we compare our bridge AI program with Wbridge5, the champions of the World Computer Bridge Championship 2016–2018. We can see that in Table 4, our program outperforms Wbridge5 by 0.41 IMP.

Table 4. Bidding strength comparison.

Model	Avg IMP	Max IMP	Min IMP
OURS-Wbridge5	0.41	0.8	−1.1

5. Conclusions

In this paper, we used the real bidding data provided by the Synrey platform to train the bidding model, which contains three-layer stacked RNN and multi-layer fully connected neural networks. Experiments showed that the accuracy of bid prediction reached 89%. At the same time, we have made a visualization system to intuitively explain the meaning of the bid, which is helpful to realize human–computer information interaction and make beginners learn the bidding rules better.

In future work, we would like to use reinforcement learning techniques to study the self-learning of bridge bidding. At present, there is still a big gap between bridge AIs and top human players. Nevertheless, we believe that subsequent research will better help bridge AIs achieve superhuman strength.

Author Contributions: Conceptualization and methodology, X.Z. and W.L.; software, X.Z. and L.L.; validation, X.Z. and L.L.; writing—original draft preparation, X.Z.; writing—review and editing, X.Z. and W.L.; supervision, F.Y.; project administration, W.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.

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

- Campbell, M.; Hoane, A.J., Jr.; Hsu, F.H. Deep blue. *Artif. Intell.* **2002**, *134*, 57–83. [[CrossRef](#)]
- 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](#)]
- 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](#)]
- Schrittwieser, J.; Antonoglou, I.; Hubert, T.; Simonyan, K.; Sifre, L.; Schmitt, S.; Guez, A.; Lockhart, E.; Hassabis, D.; Graepel, T.; et al. Mastering atari, go, chess and shogi by planning with a learned model. *arXiv* **2019**, arXiv:1911.08265.
- Blair, A.; Saffidine, A. AI surpasses humans at six-player poker. *Science* **2019**, *365*, 864–865. [[CrossRef](#)] [[PubMed](#)]
- Moravčík, M.; Schmid, M.; Burch, N.; Lisy, V.; Morrill, D.; Bard, N.; Davis, T.; Waugh, K.; Johanson, M.; Bowling, M. Deepstack: Expert-level artificial intelligence in no-limit poker. *arXiv* **2017**, arXiv:1701.01724.
- Brown, N.; Sandholm, T.; Amos, B. Depth-Limited Solving for Imperfect-Information Games. *arXiv* **2018**, arXiv:1805.08195.
- Tammelinn, O.; Burch, N.; Johanson, M.; Bowling, M. Solving Heads-Up Limit Texas Hold'em. In Proceedings of the IJCAI, Buenos Aires, Argentina, 25–31 July 2015; pp. 645–652.
- Brown, N.; Sandholm, T.; Machine, S. Libratus: The Superhuman AI for No-Limit Poker. In Proceedings of the IJCAI, Melbourne, Australia, 19–25 August 2017; pp. 5226–5228.
- Li, J.; Koyamada, S.; Ye, Q.; Liu, G.; Wang, C.; Yang, R.; Zhao, L.; Qin, T.; Liu, T.Y.; Hon, H.W. Suphx: Mastering mahjong with deep reinforcement learning. *arXiv* **2020**, arXiv:2003.13590.
- Jiang, Q.; Li, K.; Du, B.; Chen, H.; Fang, H. DeltaDou: Expert-level Doudizhu AI through Self-play. In Proceedings of the IJCAI, Macao, China, 10–16 August 2019; pp. 1265–1271.
- Luo, G.; Zhang, H.; He, H.; Li, J.; Wang, F.Y. Multiagent Adversarial Collaborative Learning via Mean-Field Theory. *IEEE Trans. Cybern.* **2020**, *51*, 4994–5007. [[CrossRef](#)] [[PubMed](#)]
- Amit, A.; Markovitch, S. Learning to bid in bridge. *Mach. Learn.* **2006**, *63*, 287–327. [[CrossRef](#)]
- Ho, C.Y.; Lin, H.T. Contract Bridge Bidding by Learning. In *AAAI Workshop: Computer Poker and Imperfect Information*; 2015. Available online: <https://www.semanticscholar.org/paper/Contract-Bridge-Bidding-by-Learning-Ho-Lin/b9977337612267f7f950ecacb8b0d945f8b6d2ff> (accessed on 11 January 2012).
- Yeh, C.K.; Hsieh, C.Y.; Lin, H.T. Automatic bridge bidding using deep reinforcement learning. *IEEE Trans. Games* **2018**, *10*, 365–377. [[CrossRef](#)]
- Tian, Z.; Zou, S.; Warr, T.; Wu, L.; Wang, J. Learning multi-agent implicit communication through actions: A case study in contract bridge, a collaborative imperfect information game. *arXiv* **2018**, arXiv:1810.04444.
- Rong, J.; Qin, T.; An, B. Competitive bridge bidding with deep neural networks. *arXiv* **2019**, arXiv:1903.00900.
- Smith, S.J.; Nau, D.; Throop, T. Computer bridge: A big win for AI planning. *AI Mag.* **1998**, *19*, 93.
- 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.
- Jack, W.; Wbridge, J.W.J. JACK triumphed over WBRIDGE5, 150-118, to retain the championship it won in 2009. *ICGA J.* **2010**, *33*, 250.

21. DeLooze, L.L.; Downey, J. Bridge bidding with imperfect information. In Proceedings of the Computational Intelligence and Games, Honolulu, HI, USA, 1–5 April 2007; pp. 368–373.
22. 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, IEEE 18th International Conference on Smart City, IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Yanuca Island, Cuvu, Fiji, 14–16 December 2020; pp. 999–1005.
23. Kou, G.; Peng, Y.; Wang, G. Evaluation of clustering algorithms for financial risk analysis using MCDM methods. *Inf. Sci.* **2014**, *275*, 1–12. [[CrossRef](#)]
24. Hwang, C.L.; Lan, C.W.; Chou, Y.J. “Search, Track, and Kick to Virtual Target Point” of Humanoid Robots by a Neural-Network-Based Active Embedded Vision System. *IEEE Syst. J.* **2013**, *9*, 107–118. [[CrossRef](#)]
25. Baumgartner, P.; Malet, Y. *High Performance Django*; CreateSpace Independent Publishing Platform: Scotts Valley, CA, USA, 2015.