Playing Gomoku Like a Human

Ziyi Xie, Yuqi Wei, Shuwen Shen

I. INTRODUCTION

Board games have been entertaining human ever since their inventions. Nowadays, with the advancement in areas of machine learning and deep learning, machines become capable of playing board games with human and some have exceptionally well performance. The prominent appearance of AlphaGo during 2015 has brought such models to the spotlight of the general public for the first time. The AlphaGo model played the board game Go with the reigning European Champion Fan Hui and won with a dominating score of 5-0. Ever since then, it became clearer how machine models could beat human in complex games such as Go. In this sense, we think it would be interesting to study whether those board game models are just emulating human behaviors or they are fundamentally different from the way human plays. Also, we want to study the influence some engineered features would bring to the models for playing board games. Since the game Go is too complex and takes too long to train, we will go for the alternative game GoBang, which is also called 5-in-a-row. In GoBang, the board is a grid and players with black and white stones take turns to put their stones on grid intersections. Whoever first reaches 5 pieces in a row horizontally, vertically or diagonally wins the game.

In this project, we adopt a GoBang model structure developed by junxiaosong[1]: An implementation of the AlphaZero algorithm for GoBang (also called Gomoku or Five in a Row) on a 11-by-11 board as a baseline structure for our study. Then we collect human data for their behaviors playing GoBang and compare them to model behaviors to check how correlated they are. Next, we update the baseline by adding newly engineered features to see if the new model outputs correlate with human results more closely. Those engineered features contain information about patterns on the board that the model should pay special attention to, and those additional features to model are like tips given to human players when they first learn GoBang. We evaluate the two models by comparing model outputs in different board game setups to corresponding human results we collected. Three different metrics are used to compute similarity scores. Finally, we analyze model performances by looking at some particularly interesting game setups.

II. METHOD

A. Model

The model we use is a combination of Neural Networks(NN) and Monte Carlo Tree Search (MCTS), and training this model for GoBang would be an unsupervised problem.

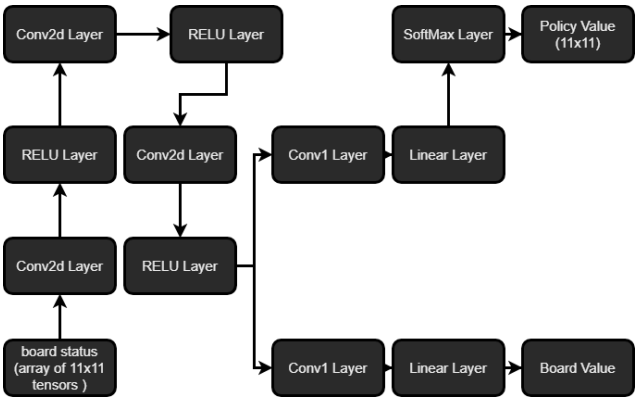


Fig. 1: Structure of the Neural Networks used for GoBang model

Neural Networks is a multi-layer deep learning structure that works like a black box, and it is very generalizable because of its large number of neurons (computing components). The design for the NN used here is to have input = 'board state' that contains information of the current board, and output = 'policy value' and 'board score'. The format of the input depends on the number of features used to train the NN. Each feature is represented by a 11-by-11 tensor. The output policy value is a 11-by-11 tensor with probability values of the next move ranked by projected win rates. The board score is a float number that should relate to the winning probability at current state. The overall structure of the NN used in the model is demonstrated in Figure 1. We have also designed the NN convolutional layers to scale with the number of features.

MCTS, a Monte Carlo method useful for generating training data, is a process also called self-play. In MCTS, one would start at a node with a current board state. The node then diverges into derivative board state nodes using some policy for sampling the next steps of placing a stone by the current player. This is called a selection process. The nodes further diverge to the next set of derivative nodes using selection, just like a tree structure. This is called an expansion process. This diverging process continues only until the winner of the game appears. Finally, the backup step allows us to evaluate the quality of each node using the number of games won after this node divided by the total number of games played after this node. This searching process is demonstrated in Figure 2. During this process, the selection/expansion can incorporate exploration by introducing randomness. The randomness, governed by some ϵ threshold (a real number between 0 and 1), is generated by

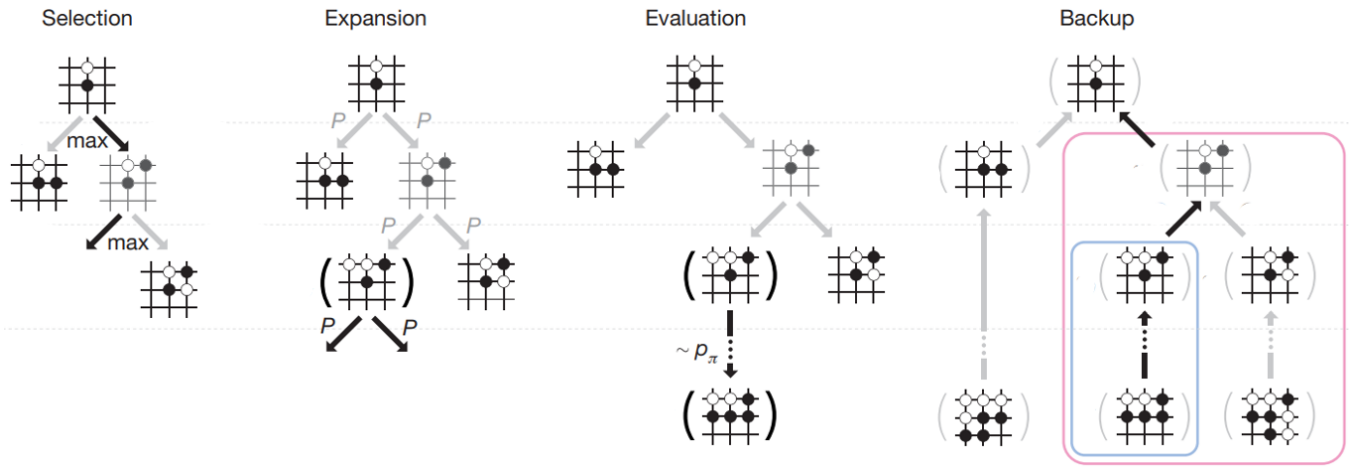


Fig. 2: MCTS Process demonstration. Selection is done using policy value from the Neural Network. Then expansion follows a tree structure of several selection process. Evaluation process finds if there is a game winner at the nodes, if there is, expansion stops. Finally backup/backpropogation creates counts for calculating the win rate at each node.

randomly choosing a number. If the random number is larger than ϵ, then the expansion would follow the policy, otherwise the expansion would randomly choose a position for the next move. In the realization of MCTS in this project, randomness is only incorporated after the first few rounds of expansion. Keeping sampling as any other Monte Carlo Methods, at each node of a board state, a policy derived from win rate is generated. We call this policy the MCTS policy. In addition, another factor we care about is whether, for a certain board state, the current player actually wins the game using the current policy. We represent this using a number between 0 and 1 as the win rate of current player given the current board status. This win rate corresponds to the board score output of the NN. In this way, MCTS creates training data of the form (board state, MCTS policy, win rate).

B. Training Pipeline

TABLE I: symbols in loss function

Letter	Meaning
s	board state
z	win rate from MCTS sampling result
π	win rate based MCTS policy for next steps
p	policy value from NN model
v	board value from NN model
θ	set of model parameters

The two concepts could be combined to build a training pipeline. The policy that guides the MCTS process is the policy output of the NN model. Using the policy from the NN, MCTS sampling process generates a win rate and a win-rate-based policy for each board state. The policy and win rate serve as a guide to the NN model since they results from sampling, a process that is known to converge to true values. Therefore, the training data generated by MCTS is then used to train the Neural Network model with some loss. To better explain the training process we use some symbol abbreviations in Table 1. To begin with, after one stone is placed, we have s0 with a 1 at some position and 0 at all other positions on a 11-by-11 board. Then the NN is initialized with random weights and given inputs as features. The NN outputs the Policy Value, which are used for MCTS to develop the game as a so-called self-play. In this way MCTS is able to generate training points of (s, pi, z). Those training points are put into the NN for training with loss $\ell = (z-v)^2 - \pi^T log(p) + c||\theta||^2$ with corresponding symbol meanings in Table 1. The first term is a loss on the model's decision if it is going to lose the game given current board and current policy value. The second term computes the cosine similarity between policy value from the model and the next step win rates from MCTS. The third term is for regularization purposes, and the model uses L2 regularization here.

The policy that guides the MCTS process is the policy output of the NN model. Using the policy from the NN, MCTS process is able to conduct a tree search to generate training data. This training data is used to train an improved NN, that is then used for the following MCTS. This back and forth process is very important in training the model by using MCTS data.

C. Features

There are four features we use for the basic model of Neural Network:

- Board status of **current** player. The 11-by-11 grid has value 1 for positions on which the current player places a stone and 0 elsewhere.
- Board status of **opponent** player. The 11-by-11 grid has value 1 for positions on which the opponent player places a stone and 0 elsewhere.
- The most recent position where the **current** player places a stone. The 11-by-11 grid has value 1 for that position and 0 elsewhere.

- The most recent position where the **opponent** player places a stone. The 11-by-11 grid has value 1 for that position and 0 elsewhere.

As for a human player, these four features are the basic things to look at when playing a GoBang game. Therefore, we would like to feed those four features in the baseline model and examine the performance of such model in comparison to human.

In addition, we want to add some features that further help the performance of the model and see if the model with more features would be more similar to how human plays. To determine the additional features, we think about the aspects a human player would consider when playing GoBang. The first feature we add is a tensor representing the distance of a player's last move to the edge of the board. The logic of this feature is that a player should not form a 4-in-a-row following a 3-in-a-row if the next step to form a five exceeds the board edge. This is an issue occurred when we play GoBang against the baseline model. The second feature we want to add is a tensor that is all 1 if the current board contains certain patterns and 0 otherwise. The patterns considered to be included in this feature are shown in Figure 3.

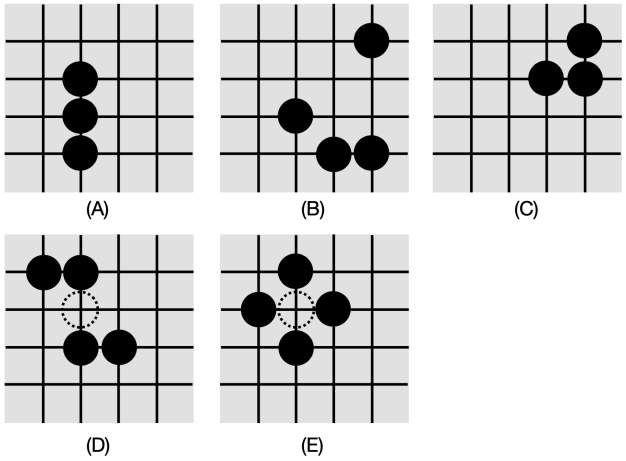


Fig. 3: Particular patterns indicated by solid black stones that players should pay special attention to while playing GoBang. (A) is a three in a row pattern. (B) (C) are structures that give the player advantages. (D) (E) are patterns that are one step away from securing a win. The position to secure a win is indicated by the dotted circle.

Those patterns could easily lead to a win or a loss by placing the next stone correctly. Therefore this feature to model is like a tip to human, reminding the player of potential win or loss opportunities for certain special board structures.

D. Comparison with Human Cognitive

In order to learn the relationship between model and human cognition for playing GoBang, we need human data. Therefore we conduct a survey to obtain human results about playing GoBang. We select 20 distinct GoBang game setups. Participants are given the color of stone they should place and are asked to choose an optimal position for the next step. We use this survey results as human results to compare with model outputs.

III. Result

A. Training Result

There are two models to be trained: the **baseline model** with four features, and the **improved model** with 2 additional features. The models are trained using the GPU environment on Google Colab. For the baseline model, it is trained approximately 20 hours where it hits a minimum loss of 2.188 as shown in Figure 4. The training for the baseline was terminated within the 24 hour GPU running time limit of Google Colab since the loss no longer improves. For the improved model, the training loss reaches as low as 2.315 as shown in Figure 5. The training process was forced to stop on Google Colab at around 2250 self-plays, and was then restarted, causing s spike in loss due to the re-initialization of board tree in MCTS.

Figure 6 shows a comparison of the training loss of the two models. The improved model only reaches a minimum loss close to the loss minimum of the baseline model for about 3x the training time. This could be due to the more complex convolutional layers of the improved model. It is also interesting that the best loss of the baseline model is about 2.5% lower than the best loss of the improved model.

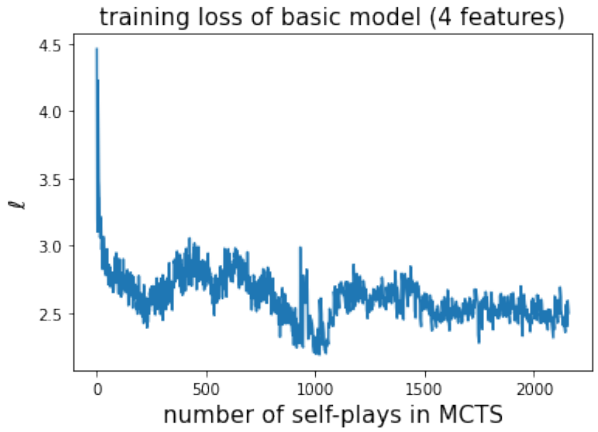


Fig. 4: Training loss using Neural Networks and Monte Carlo Tree Search for the baseline model of 4 features in game GoBang.

B. Survey Result

We received 37 survey responses from volunteers in total. For each response, it contains one subsequent stone position for the corresponding game. After removing some invalid input data, 34 responses are adopted for the evaluation process. We attach the survey link in appendix.1 and the raw survey results could be found in appendix.2.

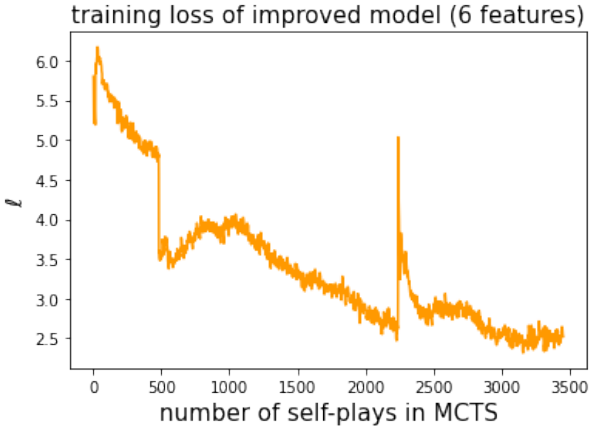


Fig. 5: Training loss using Neural Networks and Monte Carlo Tree Search for the improved model of 6 features in game GoBang.

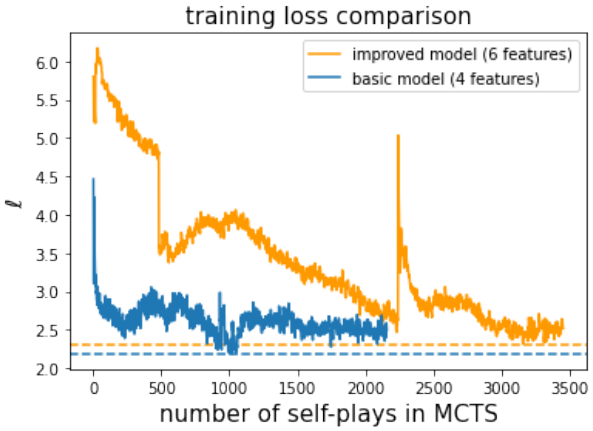


Fig. 6: Comparison of training loss using Neural Networks and Monte Carlo Tree Search between the baseline model of 4 features and the improved model of 6 features in game GoBang.

IV. Evaluation

A. Model Comparison

An important goal of this project is to evaluate our models using similarity between the policies generated by GoBang models and those by human in different board setups. The two models here are the baseline model with 4 features and the improved model with 6 features in total. We aggregate survey results from all participants for each of the 20 games to reduce individual bias.

TABLE II: Three Evaluation Metrics

	Baseline Model	Improved Model
Accuracy	0.3 (6/20)	0.3 (6/20)
Overlap Coefficient	0.3851	0.4847
Mean Average Precision	1.9987	2.8509

We use three different metrics to evaluate model performance and the similarities between model outputs and human results (see Table II). The first one is Accuracy. Similar to the traditional accuracy we know, the score here is defined as:

Accuracy = (number of games with model best policy = human best position) / 20

Since the best policy is the action actually taken, this is the first thing we evaluate. Simple as it is, this metric provides an intuitive way to compare the baseline and the improved model in terms of the degree to which the next move taken by models matches human decision. The two models both achieve an accuracy score of 0.3, although the compositions of ”correct” games are different.

To further quantify similarity, we introduce the second metric, which is called the Overlap Coefficient, or Szymkiewicz–Simpson Coefficient. It is defined as the size of the top policies sets intersection divided by the size of the set of top policies by human. For example, for Game 1, we have 8 possible positions in total provided by participants, ranked by frequency. For comparison purpose, we take the same-sized top-ranked positions from model outputs to compute the overlap coefficient. The higher the coefficient, the more overlap there is between the top possible policies of models and those of human, and thus the more similar they are. The improved model turns out to have a higher average overlap coefficient than the baseline model.

Although Overlap Coefficient provides additional information about similarities between models and human by taking into accounts the next best policies, it does not consider ranking. In other words, as long as the model set contains all positions present in the human set, the coefficient would be maximized even if the ranking is reversed. Hence, we introduce a modified version of Mean Average Precision (MAP). Average Precision is defined as, for each matching position, the fraction of higher-ranked positions that were also matching. MAP is therefore the mean of the 20 AP scores. We can see that the improved model has a higher MAP score than the basic model.

Hence, based on the three metrics, the improved model seems to make more human-like decisions on average than the basic model. The two additional features do capture some human decision-making characteristics in playing the board game.

B. Individual Game Analysis

Apart from model comparison from an aggregated perspective, we are also interested in studying specific game setups where the models and human differ (match) the most. To better visualize the strategies of machine and human, we adopt the idea of human attention and model visit distribution plots from Opheusden’s 2021 paper [2]. More specifically, for each model, we choose two game setups, one most similar to human using the three measures, the other one least similar to human.

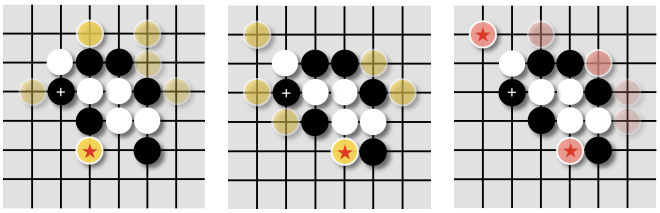
Figure 7 shows the machine distribution of the two models and human attention for Game 19, the setup where the improved model provide strategies most similar to human strategies according to the three evaluation metrics among all games. We include the baseline result for comparison purpose. An interesting observation is that although in this setting the decision making process of human is similar to that of our improved NN model, human tends to put more emphasis on forming a 4 stone in a row while machine tends to put more emphasis on stopping its opponent from forming patterns that could potentially facilitate opponent winning.

Another game worth studying is Game 5 shown in Figure 8, where models and human provide almost completely different strategies. We realize that although we have added to the improved model a feature identifying patterns that are one step away from securing a win, the improved model output is not what we expect. The pattern formed by black stones in Game 5 corresponds to the one in Fig.3E. As supported by survey result, most people would choose to place the next white stone at the center of the four black stone. The improved model's first choice is to place the white stone two positions right to the four black stones with about 37% confidence. If the next white stone is placed more than two positions away from the black stones, black is guaranteed to win the game. Hence, although the model's choice is different from what most people would do, it is not necessarily bad.

We then look at Game 17, the setup where baseline model output matches human results the most. Again, the improved model output is included for comparison. Both the baseline and human choose their next move to achieve a 3-in-a-row diagonally while locating close to other black stones, while the improved model chooses not to form an obvious attack.

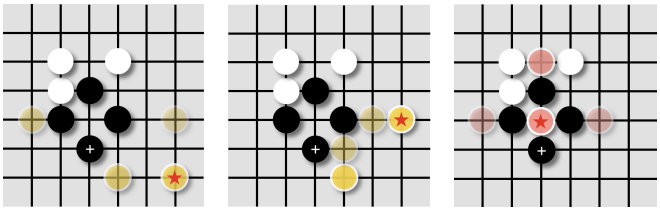
For Game 10, the top choices of the baseline are completely different from those of human. Similar to the situation in 19, human tend to form a 4-in-a-row, but models prefer exploring new reign that may be beneficial for future steps.

When evaluating the model results versus human results above, we observe that models tend to play overall more conservatively than human players. This is characterized by human players having preference in forming 3-in-a-row with two end unobstructed or 4-in-a-row, while models tend not to do so. Many more factors could cause discrepancies between model results and human results. One is that the training data is biased. Since we are generating training data using MCTS, we could fail to balance between exploration and exploitation even though the model incorporates randomness while following a policy. The biased data could cause the model to learn a set of biased parameters that in the end cause the model result to match human results less, assuming that the model should be a good cognitive model for playing GoBang. Another possible explanation is that the model *does* learn the pattern, but it decides to take another move that maximizes the probability of win based on its computation along the whole MCTS tree. In this case, the model becomes more considerate than human in the depth of thinking about next steps, because average person would not try to think as many steps ahead as there could be.



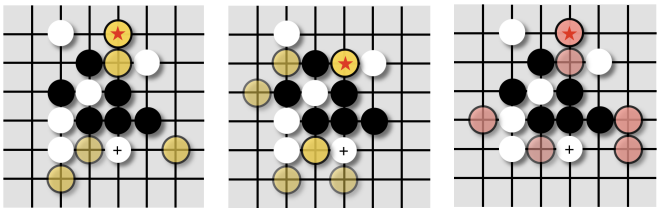
(a) Baseline Model Distribution (b) Improved Model Distribution (c) Human Attention

Fig. 7: Game 19



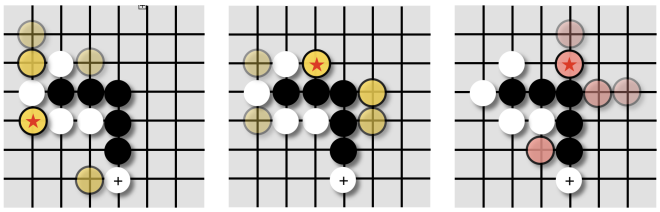
(a) Baseline Model Distribution (b) Improved Model Distribution (c) Human Attention

Fig. 8: Game 5



(a) Baseline Model Distribution (b) Improved Model Distribution (c) Human Attention

Fig. 9: Game 17



(a) Baseline Model Distribution (b) Improved Model Distribution (c) Human Attention

Fig. 10: Game 10

In Fig. 7 - Fig. 10, we summarized the top machine output policy value distribution of the baseline model and improved model for next stone placement together with the top human survey result distribution for next stone placement for Game 19, 5, 17 and 10. In each plot, stone labeled with star have the highest probability score from model or highest survey distribution of human. The opacity of the stone corresponds to model output policy value or survey result distribution from human. The black ring or white ring around stones denotes the color of stone it should be considered to be. Stone with "+" denotes the last stone taken by the opponent.

C. Individual Participant Analysis

We also analyze how human results correlates to our models at individual level. For each individual we have one response for 20 distinct games from the survey, and we compare the moves done by each individual with the two sets of model results using the accuracy score. The accuracy score for an individual is defined as the sum of 0-1 match score for all games. In figure 11 we plot the accuracy score of human results versus baseline model and in figure 12 we plot the accuracy score of human results versus improved model. We see that the average accuracy score among human players is higher for the improved model. This indicates that the improved model with the two additional features matches how human thinks to a greater degree.

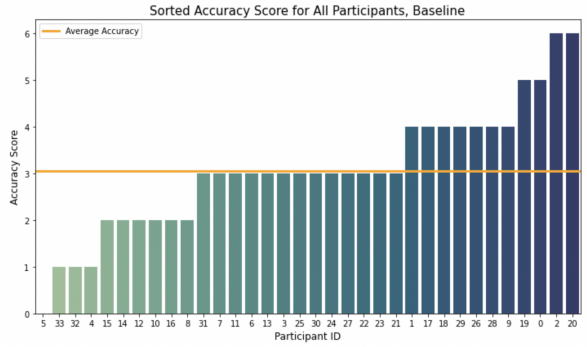


Fig. 11: Participant accuracy scores using the baseline model, defined as the total number of matching results of each participant with the baseline model.

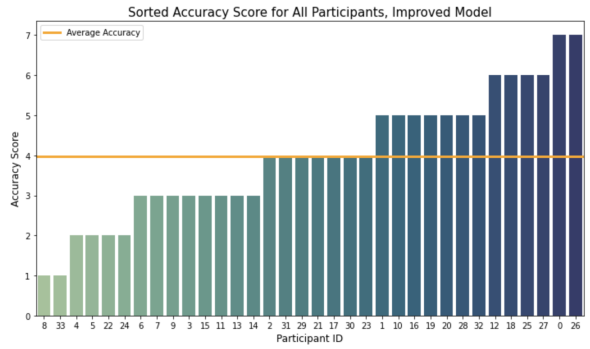


Fig. 12: Participant accuracy scores using the improved model, defined as the total number of matching results of each participant with the improved model.

V. Conclusion

In conclusion, we train two models constructed using Neural Networks and Monte Carlo Tree Search method to play the game GoBang. The baseline model contains four features; the improved model contains two additional engineered features and has a slightly larger convolutional layer. One of the feature captures the distance to the edge of the board and the other certain patterns signaling potential wins. The baseline model has slightly lower training loss and trains faster than the improved model. We then compare the model performances with human results using survey on how people play certain GoBang games. The improved model has a higher mean overlap coefficient and a higher mean average precision score than the baseline model when looking at the human next-step distribution and model next-step distribution. The models do not necessarily have a very similar distribution for playing the next stone as comparing to human players, but they do show resemblance for some game setups. We also find that, by analyzing the accuracy score for each individual participant, the improved neural network better models human behavior because it matches individual human results better.

There are some improvements that could be made without the time pressure provided for this project. The survey we conducted has 37 responses, and not all of them are valid data. It is obvious that in order to characterize human behavior in playing GoBang, we may need more survey results to reduce variance. As for the model, the better choice would be to apply model ensemble techniques. This is because the weights in the Neural Networks are randomly initialized and sometimes we find that the model training just does not converge to the same level as other training instances. Therefore, there are still many improvements that could be made to this project.

References

[1] AlphaZero-Gomoku, `https://github.com/junxiaosong/AlphaZero_Gomoku`

[2] Opheusden, Bas Van, et al. *Revealing the impact of expertise on human planning with a two-player board game*, `https://psyarxiv.com/rhq5j`

Contribution

Member	Contribution
Ziyi Xie	Model Understanding, Model Training, Feature Engineering, Visualization
Yuqi Wei	Model Evaluation, Model vs. Human Analysis, Visualization
Shuwen Shen	Survey Preparation, Model Evaluation, Data Visualization

Appendix

1) Survey Link: https://forms.gle/Jje8pVjHZJvQZHzu7

2) Survey Result:

Timestamp	1. You are: White	2. You are: Black	3. You are: Black	4. You are: Black	5. You are: White	6. You are: Black
4/24/2021 22:54:46	E5	H7	D6	F4	D5	F7
4/24/2021 22:59:19	E5	H7	D6	F4	D5	F9
4/24/2021 23:00:38	E5	H7	D6	F4	D5	G6
4/24/2021 23:04:04	E5	F4	D6	F4	D5	G6
4/24/2021 23:26:52	H9	H7	C8	H4	D3	D8
4/24/2021 23:30:06	F4					
4/24/2021 23:30:14	F9	H4	C8	I5	D5	G6
4/24/2021 23:31:09	E5	C8	D9	H4	D5	G6
4/24/2021 23:31:35	E5	H4	C8	F4	D5	D8
4/24/2021 23:37:45	F4	F5	F8	H5	D3	I6
4/24/2021 23:39:16	E5	C8	C8	F7	D5	G6
4/24/2021 23:41:13	E5	H7	D6	G4	D5	G6
4/24/2021 23:45:05	D9	H7	H6	F4	D3	F7
4/24/2021 23:48:20	E5	H4	D6	F4	D5	F9
4/24/2021 23:54:48	E9	H7	D6	I8	D5	F9
4/25/2021 0:15:41	E5	H7	C8	F4	D3	F9
4/25/2021 0:20:47	H9	H7	D6	F4	B5	F9
4/25/2021 0:24:59	D9	H4	H7	E7	D5	F9
4/25/2021 0:31:26	E5	H5	D6	F5	D5	F9
4/25/2021 0:34:50	E5	H7	C8	F4	D5	F9
4/25/2021 0:35:25	E5	H7	C8	F4	D5	F9
4/25/2021 0:36:56	H9	H7	D6	F5	D3	F5
4/25/2021 0:41:05	E5	D6	D6	F7	D5	G6
4/25/2021 0:47:11	E5	H8	E6	H4	F5	D8
4/25/2021 2:21:42	H9	H4	D6	F4	D5	F9
4/25/2021 3:37:19						
4/25/2021 6:12:34	E5	H7	D9	J6	D5	H5
4/25/2021 6:32:55	E5	D6	D6	E7	D3	F9
4/25/2021 7:42:50	E5	I7	D5	H4	D5	F9
4/25/2021 16:14:59	E5	H7	D6	F4	D5	F7
4/25/2021 18:53:17	E5	F5	C8	F5	D5	G6
4/25/2021 21:30:14	E5	C8	D6	F5	D5	F9
4/26/2021 1:13:14	E5	F5	D6	F4	D5	F9
4/30/2021 3:10:04	E5	H7	D6	H4	D3	F9
4/30/2021 17:35:15	G8	F5	D6	F4	D3	D8
5/3/2021 14:08:11	D6	H7	E6	F5	D3	F9
5/3/2021 17:00:10	H9	H7	H7	F4	D3	D8

7. You are: White	8. You are: Black	9. You are: Black	10. You are: Black	11. You are: White	12. You are: Black
G7	G6	E8	I6	F4	G5
E6	F7	D7	H5	H6	G7
H7	F7	D7	G8	G8	E7
H7	F7	D7	G8	C7	E7
G7	F7	D7	H5	F4	H3
H7	F7	H6	H5	H5	G7
H7	F7	H9	G8	G8	G7
H7	F7	D7	G8	F4	G7
D7	E7	E5	H5	D4	E5
H7	F7	D7	G8	G8	E7
H7	F7	D7	I6	F4	H3
F4	F7	D7	G8	G8	E7
H7	F7	D7	G8	F4	G7
H7	F7	D7	H4	H6	H3
H7	F7	D7	H5	F4	G5
H7	F7	G4	G8	G8	E7
H7	F7	D7	G4	G8	F3
H7	F7	D7	J6	G8	I5
H7	F7	H6	I6	G8	G5
H7	F7	F4	G8	G5	H3
H7	E5	H6	H5	G5	E7
E6	E7	G4	H5	E4	G5
H7	F7	D7	H4	G8	E7
H7	F7	H6	I6	F4	G5
H7	F7	D7	H5	E7	E7
H7	G8	G4	G5	C7	G5
H7	F7	I3	I6	F4	G5
F4	F7	D7	G8	G8	G7
F4	F7	D7	G8	E7	E7
H7	F7	H6	J6	F4	G5
H7	F7	D7	H5	F4	H3
G7	E7	H6	H5	H5	H3
G7	G6	E5	H5	F4	H5
H7	B3	I3	H5	H6	H3

13. You are: White	14. You are: White	15. You are: White	16. You are: Black	17. You are: Black	18. You are: Black	19. You are: White	20. You are: White
D4	E8	E7	E5	G3	F4	G7	E5
F3	E6	G6	E5	G3	F4	C3	E5
E8	E8	G6	D5	G3	D4	E3	E5
D7	E6	G6	E7	D6	G6	G4	E5
E8	D8	H3	D3	D6	G6	H7	H9
C6	H8	G6	E5	D6	F4	C3	H9
C6	E8	G6	E7	D6	G6	F4	E5
E8	E8	G6	D3	I6	D4	C3	E5
D4	H8	E3	I6	F3	F4	H6	H8
D8	E8	G6	D5	F7	H4	E3	E5
E8	E8	G6	F4	G4	F4	F7	H9
E8	E6	G6	D3	D7	G6	C3	E5
E8	F7	G6	E5	I7	D4	F7	E5
E8	E8	G6	D3	I6	D4	C6	H5
G8	E8	G6	D3	I7	G6	C3	H9
C7	E6	G6	D4	I7	D4	F7	E5
C7	E6	G8	B3	G4	D6	F7	E5
E8	E8	G6	D4	H7	D4	C3	E5
G8	E8	G6	E7	G4	F4	G4	E5
E8	E8	D4	D3	G3	F4	F7	E5
D7	E8	G6	H5	G3	F4	G4	E5
F3	F5	D4	E5	I7	F4	G4	E5
F3	E8	G6	E7	G3	G6	F7	H9
C7	H6	G3	D3	G3	D4	F7	H9
F3	E8	G6	E7	I6	F4	C3	H9
D7	E8	G6	D4	G3	E8	F7	F9
F3	H8	G6	E5	G3	D5	C3	E5
E8	E8	G6	E5	F7	G6	F7	E5
E8	E6	G6	E7	G3	G6	F7	E5
G8	E8	G6	H4	G3	D4	H5	E5
C6	E6	G6	F4	I6	F4	C3	E5
D7	F7	G6	D3	I6	C7	C3	E5
D4	H6	D4	H5	I7	D8	F7	G9
F5	H6	E3	D3	I6	G6	C3	H9