Improving the Human-Likeness of Game AI's Moves by Combining Multiple Prediction Models

Tatsuyoshi Ogawa, Chu-Hsuan Hsueh and Kokolo Ikeda
Japan Advanced Institute of Science and Technology, Nomi, Ishikawa, Japan

Keywords: Human-Likeness, Player Modeling, Move Prediction, AlphaZero, Shogi.

Abstract: Strong game AI's moves are sometimes strange or difficult for humans to understand. To achieve better human-computer interaction, researchers try to create human-like game AI. For chess and Go, supervised learning with deep neural networks is one of the most effective methods to predict human moves. In this study, we first show that supervised learning is also effective in Shogi (Japanese chess) to predict human moves. We also find that the AlphaZero-based model more accurately predicted moves of players with higher skill. We then investigate two evaluation metrics for measuring human-likeness, where one is move-matching accuracy that is often used in existing works, and the other is likelihood (the geometric mean of human moves' probabilities predicted by the model). To create game AI that is more human-like, we propose two methods to combine multiple move prediction models. One uses a Classifier to select a suitable prediction model according to different situations, and the other is Blend that mixes probabilities from different prediction models because we observe that each model is good at some situations where other models cannot predict well. We show that the Classifier method increases the move-matching accuracy by 1%-3% but fails to improve the likelihood. The Blend method increases the move-matching accuracy by 3%-4% and the likelihood by 2%-5%.

1 INTRODUCTION

Game AI has beaten human champions in two-player perfect information games such as chess, Go and Shogi (Japanese chess) (Silver et al., 2018). However, strong game AI's policies (probability distributions over moves given states) often differ from human players' (McIlroy-Young et al., 2020), which may cause problem in human-computer interactions. For example, players often do not enjoy playing against game AI when the game AI's moves may look strange or hard to understand.

Human-like policies are needed to solve this problem and can be used not only as an opponent AI. For example, human-like policies can be used to assess the difficulty of game problems such as chess mating problems according to the move prediction. With many possible applications of human-like policies', creating human-like game AI is one of the most important topics in the field of game research.

To predict human players' moves, McIlroy-Young et al. (2020) introduced a chess AI called Maia. They divided human players' records into 9 groups according to the players' ratings (e.g., 1100–1199, 1200–1299). They used 12 million games in each group to train a model to predict human players' moves. Their results showed that Maia could predict human players' moves better than other chess AI.

Despite the promising results, we find two issues that are worth further discussing: the evaluation metric and room for improvement in human-likeness when using small amounts of data. As for the evaluation metric, related studies often used move-matching accuracy (McIlroy-Young et al., 2020) (Jacob et al., 2022), i.e., whether the predicted moves matched human players' moves. This metric is useful to some extent in evaluating the ability to imitate humans.

However, it has a weakness where moves different from human moves are equally evaluated as mismatches, no matter whether the moves are natural or impossibly unnatural to humans. In other words, even if game AI has averagely high move-matching accuracy, it may still play moves that are not human-like.

As an alternative metric to evaluate human-likeness, we consider it reasonable to use likelihood, which is human moves' probabilities predicted by the model. A product of likelihoods is maximal if and only if the model's policy is equal to human moves. Hence likelihood is an evaluation metric that is consistent with the goal of imitating human policies.

Ogawa, T., Hsueh, C. and Ikeda, K.
Improving the Human-Likeness of Game AI's Moves by Combining Multiple Prediction Models.
DOI: 10.5220/0011804200003393
In *Proceedings of the 15th International Conference on Agents and Artificial Intelligence (ICAART 2023) - Volume 3*, pages 931-939
ISBN: 978-989-758-623-1; ISSN: 2184-433X

Regarding the room for improvement in human-likeness when using small amounts of data, Maia's method requires a large amount of data and cannot be used in many games. Generally, supervised learning does not work well when there is little training data. In such cases, the move-matching accuracy and the likelihood may be improved by search (Jacob et al., 2022), or by combining human-like models with different move selection mechanisms such as the policy of strong game AI like AlphaZero (Silver et al., 2018).

In this paper, we first confirm whether supervised learning like Maia is also effective in Shogi. We then propose methods that combine multiple policies to further improve human-likeness. To our knowledge, we are the first to combine multiple policies to imitate human policies. The combined policies include those from Maia-like models and AlphaZero-like AI, where the latter has been shown to be less human-like (McIlroy-Young et al., 2020). Interestingly, our results show that the combination improves move-matching accuracy and likelihood.

2 RELATED RESEARCH

To create strong Go programs, Coulom (2007) proposed a new Bayesian technique for supervised learning for training a model to predict the probability distribution of human players' moves. He used strong human players' games to train the prediction model and then combined the model into a Go program based on Monte-Carlo tree search (MCTS). The Go program's strength was greatly improved. Similarly, some other researchers strengthened their game AI by incorporating move prediction models (Tsuruoka et al., 2002) or evaluation functions trained using human players' games (Hoki and Kaneko, 2014).

Obata et al. (2010) proposed a consultation algorithm that selects a move from moves of multiple Shogi AI and succeeded in making Shogi AI significantly stronger than each AI alone. A possible reason their method worked well was that the majority vote can compensate for each other's shortcomings.

AlphaZero (Silver et al., 2018) is a reinforcement model trained using self-play games instead of human games. Silver et al. (2018) used a policy network to predict probabilities of moves from positions and a value network to predict the win rates of positions. The training data of the networks came from self-play games played by a variant of MCTS that incorporates the networks. AlphaZero beated world champion game AI in chess, Go, and Shogi.

Maia (McIlroy-Young et al., 2020) is known for one of the most effective chess AI in predicting human moves. This chess AI used deep neural networks for supervised learning. Human players were divided into 9 groups according to their ratings. Each neural network corresponded to a rating range and was trained using 12 million games from the players in the rating range. Their results showed that moves in a rating range was best predicted by the neural network of the corresponding rating range, where the move-matching accuracy was about 50%. McIlroy-Young et al. (2020) claimed that using neural networks alone obtained higher move-matching accuracy than combining the neural networks into tree search as AlphaZero did. However, Jacob et al. (2022) showed that even with the same training model as Maia, the model with search was stronger and had higher move-matching moves if parameter was adjusted properly.

With respect to human-likeness, Togelius et al. (2013) introduced the concept of "believability". Believability refers to the ability to make a character or bot seem as if it were controlled by a human being. Various approaches were then proposed to achieve humanlike characteristics (Fujii et al., 2013) (Hingston, 2010).

As another approach to create human-like AI, Kinebuchi and Ito (2015) proposed to improve move-matching accuracy of Shogi AI by considering the flow of preceding moves. They also targeted players in a wide range of skill levels. They represented the flow by combining a search-based value function (Hoki and Kaneko, 2014) using a transition probability function (Tsuruoka et al., 2002). Linear combination was used and the weight was trained with human moves. Their proposed method predicted human moves significantly better than each function alone.

3 PROPOSED METHOD

The overview of this study is as follows. First, in the case of Shogi, we confirm whether supervised learning like Maia can well predict human moves in two metrics, move-matching accuracy and likelihood. We also compare AlphaZero-like policy with supervised learning policy to identify the strengths and weaknesses of each policy. We then propose two approaches to improve move-matching accuracy and likelihood by combining the supervised learning policy and the AlphaZero-like policy.

We follow Maia's method and use neural networks for supervised learning. Consider a neural network used for multiclass classification with K classes, and let x be the input and $u_k(-\infty < u_k < \infty)$ be one of the output of the neural network. The probability $p(C_k|x)$

that x belongs to class C_k is often expressed as

$$p(C_k|x) = \frac{exp(u_k)}{\sum_{j=1}^{K} exp(u_j)}, \quad (1)$$

and x is classified into the class of $\arg\max_k p(C_k|x)$. In the case of Shogi, for a position x, if there are K legal moves, a policy function can be made from the probabilities $p(C_k|x)$ of moves C_k ($k = 1$ to K).

Move-matching accuracy is an evaluation metric that $\arg\max_k p(C_k|x)$ matches the actual move and is often used in studies when predicting human moves. However, move-matching accuracy has a weakness of not being able to properly evaluate the probability distribution. As an example, consider a position in which 60% of human players play move A, 30% play move B, and 10% play move C. Assuming that we have two prediction models, one with the probability distribution of A: B: C = 90%: 5%: 5% and the other with 34%: 33%: 33%, both models' move-matching accuracy is 60% because move A has the highest probability. When imitating humans to improve the strength of game AI, there were no problems using either prediction model. However, when imitating humans for human-likeness, the shape of the distribution becomes important. The ideal probability distribution is to give high probabilities to moves that human players often play and low probabilities to moves rarely played, i.e., a distribution with the same shape as humans.

To solve this problem, we use likelihood as another metric to evaluate how well a policy predicts human moves. Given a set of positions $x \in X$ and the corresponding human moves C_{human}, we calculate likelihood as follows,

$$(\prod_{x \in X} p(C_{human}|x))^{\frac{1}{|X|}}, \quad (2)$$

where $|X|$ is the size of X and $p(C_{human}|x)$ is prbability of C_{human} from the policy. In other words, it is the geometric mean of the predicted probablities by the policy. likelihood is the maximal only when the human move distribution and the model policies are equal. Therefore, likelihood is a reasonable measure of the imitation of human policies. In this paper, we use both move-matching accuracy and likelihood to evaluate models' human-likeness.

3.1 Classifier Model

The first method combining multiple policies is a model using a classifier. Figure 1 shows an overview of the Classifier model. Assume that we have two different policies, P_1 and P_2, each with their own strengths and weaknesses to predict human moves. We want P_1 to predict when the positon is suitable for P_1 and P_2 to predict when the position is suitable for P_2. To achieve this, we use a classifier to determine whether the positions are suitable for P_1 or P_2.

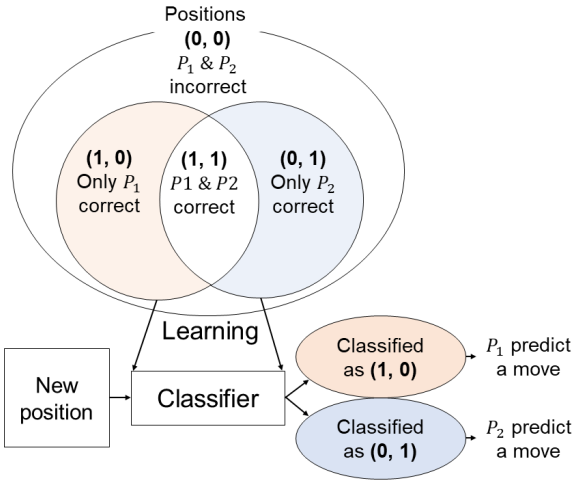


Figure 1: An overview of the Classifier model, where (0, 0) means positions that P_1 and P_2 cannot correctly predict human moves, (1, 0) means positions that only P_1 correctly predict human moves, (0, 1) means positions that only P_2 correctly predict human moves, (1, 1) means positions that P_1 and P_2 correctly predict human moves.

To create the training data for the classifier, we prepare a set of positions with human moves and let P_1 and P_2 predict moves for each position. A position is labeled as (1, 0) if only P_1 correctly predicts the corresponding human move and as (0, 1) if only P_2 correctly predicts the move. We then use these (position, label) pairs to train the classifier by supervised learning.

When predicting human moves, P_1 is used if the position is classified as (1, 0), and P_2 is used if the position is classified as (0, 1). With the classifier, we can use the relatively proper policy for each position. This method can be further extended from binary classification to multi-class or multi-label classification.

3.2 Blend Model

The second method combining multiple policies is a model that blends the values of policies. Inspired by ensemble learning, a set of machine learning algorithms that obtain better accuracy by integrating the estimation results of multiple learners, we propose to blend probabilities from different policies.

As in the Classifier model, assume that we have two policies that have different strengths and weaknesses. Let p_{1k} be the probability of move k in P_1, p_{2k} be that in P_2, and $\alpha (0 \leq \alpha \leq 1)$ be a parameter deciding the importance of P_1. The new probability p_k is

calculated as follows.

$$p_{new\,k} = p_{1k}^{\alpha} \times p_{2k}^{(1-\alpha)} \tag{3}$$

$$p_k = \frac{p_{new\,k}}{\sum_{j=1}^{K} p_{new\,j}} \tag{4}$$

While this formula blends the two probabilities nonlinearly, it is possible to use a linear blend like

$$p_{newk} = \alpha \times p_{1k} + (1-\alpha) \times p_{2k} \tag{5}$$

or a more general form from both (3) and (5) like

$$p_{newk} = (\alpha \times p_{1k}^{\beta} + (1-\alpha) \times p_{2k}^{\beta})^{1/\beta}. \tag{6}$$

In this paper, we use formula (3) because preliminary experiments showed that (3) is superior to (5) and almost equal to (6) in terms of move-matching accuracy and likelihood.

4 HUMAN MOVE PREDICTION IN SHOGI

4.1 Experiment Settings

In this section, we conducted experiments to confirm whether supervised learning can predict human moves with high accuracy in Shogi, as Maia does in chess, and how well AlphaZero-like policies can predict human moves. When evaluating how well human moves are predicted, we used two metrics, move-matching accuracy and likelihood.

Shogi is a Japanese chess-like game. The main difference between Shogi and chess is allowing captured pieces to be returned to the board by the capturing player. We use Shogi games played by humans on Shogi-Quest[1]. Shogi-Quest is a popular Shogi platform that adopts the Elo rating system to evaluate players' skill levels. On this platform, players can choose 2-minute, 5-minute, or 10-minute games. These minutes are the thinking time per player, and when a player runs out of this time limit, he or she loses the game immediately.

To predict human moves in Shogi, we performed supervised learning of policy functions and value functions like Maia's study. We used 3 million 10-minute games and filtered out improper data of the following three types. First, we eliminated games where players lost due to running out of the time. The reason for this was that there may be noisy behaviors specific to be losing the game by out-of-time, such as moving the piece that was easiest to operate. Second, we eliminated games with a player rating difference of 50 or more. The reason for this was that rating difference could adversely affect the learning of the value function as well as the policy function, as the stronger player may win from an extremely disadvantageous situation, making the data noisy. Third, we used the positions in which the number of moves was after the 50th move. The reason for excluding the early positions was that there are many similar positions in the early stages of the game, and having many similar data may harm the learning process.

The remaining 760 thousand games were divided into six groups of equal number of games according to the average rating of the players. The rating range for each group was as follows.

- Group 1: R1433 - R1591
- Group 2: R1592 - R1655
- Group 3: R1656 - R1708
- Group 4: R1709 - R1768
- Group 5: R1769 - R1855
- Group 6: R1856 - R2140

As a result, we used 127 thousand games for training each model. This is about one-hundredth the number of datasets compared to Maia's 12 million.

90% of data in each group were training data, 5% were validation data, and the remaining 5% were test data for evaluation. We performed multi-task learning similarly to AlphaZero's network architecture, in which the policy network and the value network were simultaneously learned as a single network. We referred to the python-dlshogi2 library[2] for the network structure and learning options. The major difference from the library was that we included past positions in the network's input instead of only inputting the current position. This is because in Maia's study, move-matching accuracy was significantly improved after including the recent history of 12 ply (6 moves for each player). In our preliminary experiment, a model that included the last 12 positions improved move-matching accuracy compared to a model based only on the current positions. Thus, we adopted the model that includes the last 12 positions.

We performed 10 epochs of training for each group. This is because we observed that the loss often converged at around 10 epochs. The training took about 4 hours for each group on a PC with an RTX-3070 GPU.

To simplify discussions, these Maia-like models are denoted by Maia-S (S stands for the initials of "small data" and "Shogi"), and the model trained using group 1 is denoted by Maia-S-1, the model trained using group 2 is denoted by Maia-S-2, and the remaining is similar.

[1] http://questgames.net/

[2] https://github.com/TadaoYamaoka/python-dlshogi2

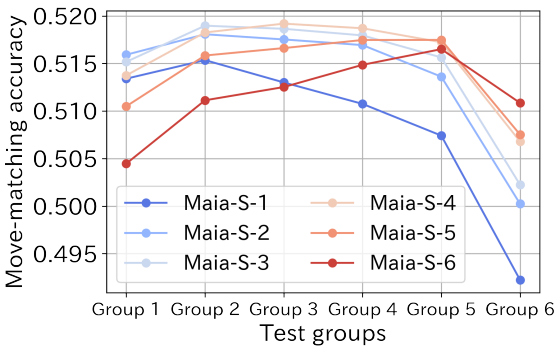


Figure 2: Move-matching accuracy of Maia-S models.

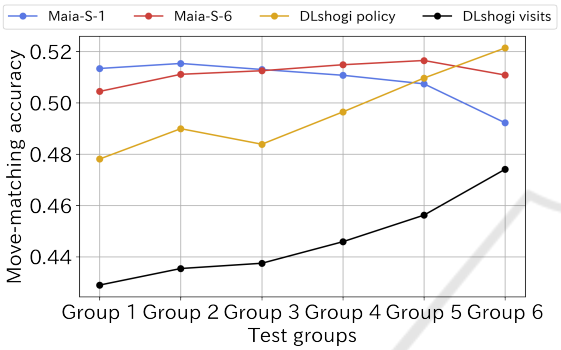


Figure 3: Move-matching accuracy of Maia-S models and DLshogi.

In addition to Maia-S models, we employed an AlphaZero-based program, DLshogi[3], for comparison. DLshogi was also the winner of the 32nd World Computer Shogi Championship held in 2022. In this paper, "DLshogi policy" means DLshogi prior (policy without search) and "DLShogi visits" means the probability distribution over moves proportionally with respect to the visit counts obtained by MCTS of DLshogi. In our experiments, we limited the number of nodes in DLshogi's MCTS to 10,000.

4.2 Results

In this section, we first discussed Maia-S results. Figure 2 showed the move-matching accuracy of Maia-S models tested on different groups. For all groups, the move-matching accuracy was between 51.0% to 52.0% by the model that best predicted the group. For example, for Group 1, Maia-S-2's move-matching accuracy, 0.515, was the best among Maia-S models. We confirmed that Maia's method had reasonably high move-matching accuracy in Shogi. The accuracy was about the same level as Maia's results in chess, though it is less meaningful to compare results in different games. As a general tendency, if the rating of the training data for the prediction model is closer to the rating of the test data, the prediction performance were better, and if the ratings were further, the prediction performance got worse.

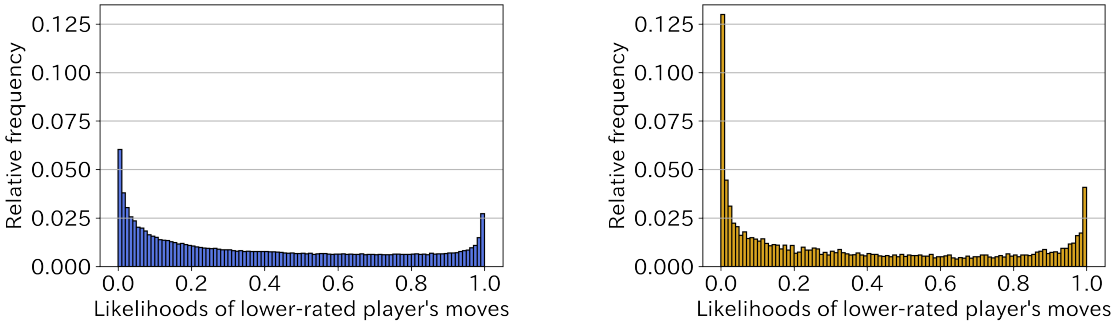
Next, we analyzed whether AlphaZero-like policies can successfully predict human moves. The results of DLshogi policy and DLshogi visits are depicted as the yellow and black curves in Figure 3, respectively. Both policies tended to be able to predict the higher rated human moves more accurately. This tendency is consistent with the results of Jacob et al. (2022). They also claimed that the effect of search depended on the rating when using search with Maia's models.

Compared to the results of DLshogi, the move-matching accuracy of Maia-S models varied only about 1 % across different groups. In addition, for Group 6 (high-rated players), DLshogi policy had higher move-matching accuracy than Maia-S models. From these results, we conclude that Maia-S models can predict human moves with high accuracy, independent of the rating, and DLshogi policy can predict the higher-rated human moves more accurately.

In addition to move-matching accuracy, we analyzed the models' likelihoods of playing human moves (2). We focused on the data of Group 1 (low-rated players) for the following reason. We considered it was worth investigating low-rated players' moves because the gap between DLshogi policy and the best-performed Maia-S model was the biggest. Figure 4 shows the histograms of the likelihoods of low-rated players' data for Maia-S-1 model and DLshogi policy. The x-axes are likelihoods divided into 100 bins (i.e., 0.00–0.01, 0.01–0.02, ..., and 0.99–1.00), and the y-axes are the relative frequency of each bin. Both distributions were bimodal and had two peaks at the two ends. In more detail, one peak was at the bin of 0.00–0.01, which means that the models were unlikely to select the human moves in the data. The other peak was at the bin of 0.99–1.00, which means that the models were likely to select the human moves. When comparing DLshogi policy and the Maia-S-1 model, DLshogi policy predicted human moves to have very low probabilities (the left peak) much more frequently than the Maia-S-1 model.

We further looked into the positions where the corresponding human moves received low probabilities from Maia-S models and/or DLshogi policy. Some moves were good moves but requiring looking ahead (search) to find these moves to be good. Maia-S models sometimes predicted these good moves to have low probabilities, resulting in low likelihood. In addition, both Maia-S models and DLshogi policy often obtained low likelihood due to humans' misun-

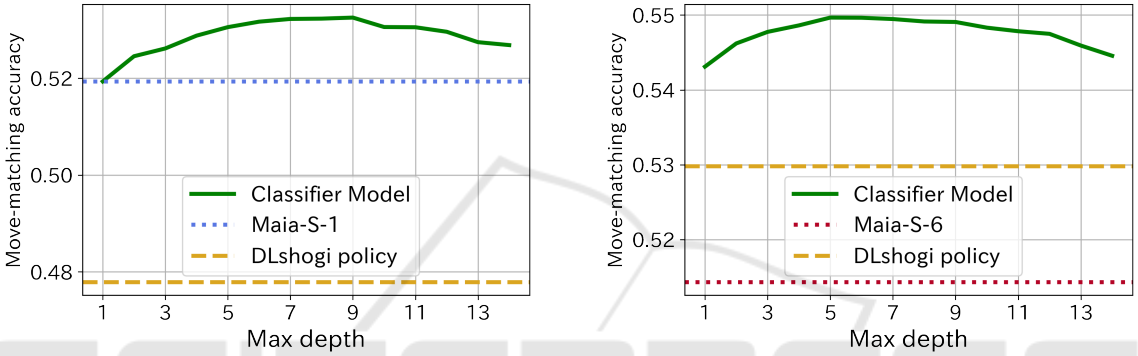
[3]https://github.com/TadaoYamaoka/DeepLearningShogi



(a) Maia-S-1 model

(b) DLshogi policy

Figure 4: Likelihood histograms of low-rated players' data.



(a) Low-rated players' data.

(b) High-rated players' data.

Figure 5: Move-matching accuracy of the Classifier model, Maia-S models, and DLshogi policy.

derstandings such as oversights of moves.

In this chapter, we showed that Maia-S models and DLshogi policy had reasonably high move-matching accuracy in predicting human moves, and that each has its own strengths. We also showed that there were some human moves with low prediction probabilities for both Maia-S and DLshogi policies. In Chapters 5 and 6, we will combine these models to utilize their strengths and show that the combinations can predict human move better.

5 CLASSIFIER MODEL

5.1 Data and Model Settings

In this chapter, we conduct experiments to evaluate the Classifier model proposed in section 3.1 for improving move-matching accuracy and likelihood.

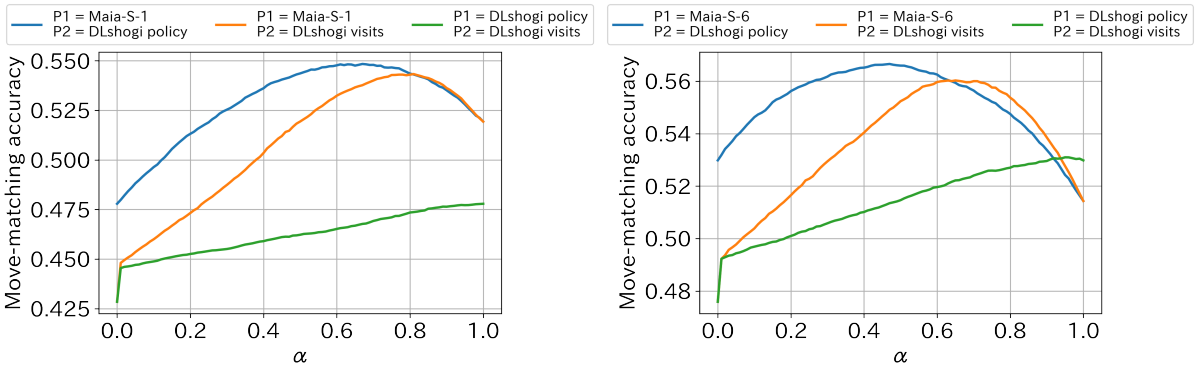
We randomly sampled 45,000 positions from group 1 and group 6 of the experiments in chapter 4. We used sklearn.ensemble.RandomForestClassifier[4] with parameters as the default settings except that we tried different settings for max depth. Regarding the input to the classifier, we used three features: $\max_k p(C_k|x)$ for the given position x from Maia-S policy, that from DLshogi policy, and the KL divergence (like the distance between probability distributions) between Maia-S and DLshogi policy. The Classifier model outputs (1, 0) or (0, 1), deciding whether the Maia-S policy or the DLshogi policy were more suitable for the given position. 10-fold cross-validation was used for evaluation, and the mean of the move-matching accuracy for each test data was calculated.

5.2 Results

Figure 5 shows the move-matching accuracy of the Classifier model with different max depth settings. We also included the results of the Maia-S policy and the DLshogi policy for comparison.

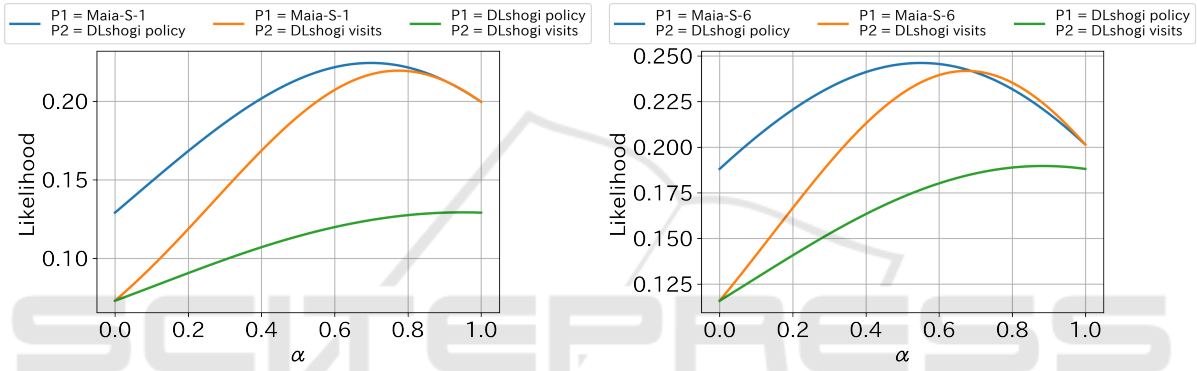
For the low-rated players' data, move-matching accuracy increased by 1% when max depth was 9 compared to Maia-S-1 policy. For the high-rated

[4]https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html



(a) Low-rated players' data. (b) High-rated players' data.

Figure 6: The move-matching accuracy under different α and P1/P2 combinations in the 2-Blend models.



(a) Low-rated players' data. (b) High-rated players' data.

Figure 7: The likelihoods under different α and P1/P2 combinations in the 2-Blend models.

players' data, move-matching accuracy increased by 2% when max depth was 5 compared to the DLshogi policy. The Classifier model obtained higher move-matching accuracy than using single models for both low-rated and high-rated players' data. The results showed that the accuracy was improved by selecting more suitable policy using the Classifier model.

We also compared the models' likelihoods. For low-rated players, Maia-S-1 model's likelihood was 0.196, DLshogi policy's likelihood was 0.129, and the Classifier model's likelihood was 0.169. For the high-rated players, Maia-S-6 model's likelihood was 0.198, DLshogi policy's likelihood was 0.188, and the Classifier model's likelihood was 0.190. In summary, the Classifier model improves the move-matching accuracy but not the likelihood. This is one example where move-matching accuracy alone is not a perfect measure of human-likeness.

6 BLEND MODEL

6.1 Data and Model Settings

In this chapter, we conduct experiments to evaluate the Blend model proposed in section 3.2 for improving move-matching accuracy and likelihood.

We used the same sets of 45,000 positions for group 1 and group 6 as section 5.1 to test the Blend model. The model is $p_k = p_{1k}^{\alpha} \times p_{2k}^{(1-\alpha)}$ for the 2-Blend model. Candidates for P are Maia-S, DLshogi policy, and DLshogi visits. Including DLshogi visits was the main reason that we used 45,000 positions instead of all games in a group, where DLshogi visits cost about 4 seconds to obtain the probability distribution per position.

6.2 Results

Figure 6 plots the move-matching accuracy for the 2-Blend model with the blend parameter α from 0.0 to

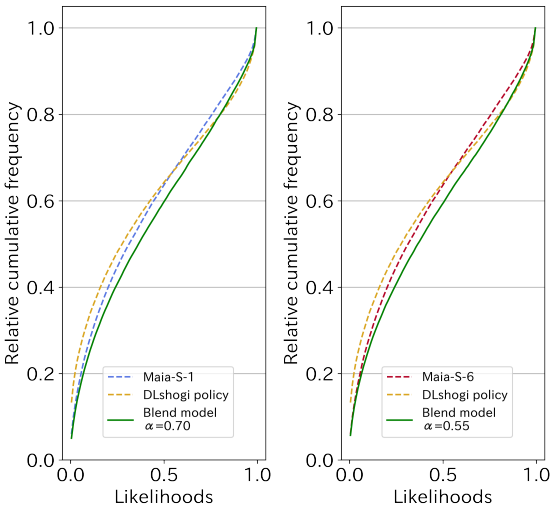
1.0, and Figure 7 plots the likelihood.

The combination of Maia-S and DLshogi policy (the blue curves) obtained better move-matching accuracy and likelihood than the other combinations for both low-rated players' data and high-rated data. The best alpha was around 0.4-0.7, which was better than using the policy alone. As for the combination of DLshogi policy and DLshogi visits (the green curves), the curves show a steadily increasing tendency, but the move-matching accuracy and likelihood were as best as using DLshogi policy alone ($\alpha = 1.0$). The results showed that this combination was less valuable.

With the combination of Maia-S and DLshogi policy with the best α settings, the 2-Blend model improved the move-matching accuracy from 0.519 to 0.548 for the low-rated players' data and from 0.530 to 0.567 for the high-rated players' data. The model also increased the likelihood from 0.200 to 0.224 for the low-rated players' data and from 0.201 to 0.246 for the high-rated players' data.

When comparing Figures 6(a) and 6(b), we observed that the best value of α tends to be higher for the low-rated players' data than the high-rated players' data. α represents how much P1 was blended, for example, the amount of Maia-S model in the blue curves. Section 4.2 has shown that Maia-S model was better at predicting relatively low-rated players' data, while DLshogi policy was better at predicting relatively high-rated players' data. The tendency of α was consistent with this result.

Figure 8 shows the relative cumulative frequency of the likelihoods for the 2-Blend model, Maia-S model, and DLshogi policy. The closer the likelihood is to 0, the more unlikely the model selected human moves, and the closer likelihood is to 1, the more likely the model selected human moves. The 2-Blend model's curves (green curves) are generally below Maia-S models' and DLshogi policy' curves, which means that the 2-Blend model gave higher probabilities to human moves than Maia-S models and DLshogi policy.



(a) Low-rated players' data. (b) High-rated players' data.

Figure 8: The relative cumulative frequencies of likelihoods.

7 CONCLUSION

Maia is known for a chess AI that learns from human games and is the most effective chess AI in predicting human moves. In this paper, we first showed that supervised learning like Maia, which we named Maia-S, was effective in predicting human moves in Shogi. We also analyzed how well AlphaZero-based models predicted human moves, where AlphaZero learned from self-play games instead of human games. We found that the AlphaZero-based model more accurately predicted moves of players with higher skill. Based on the analyses, we proposed two approaches to improve the prediction performance on human moves by combining multiple policies. The first approach uses a classifier to predict human moves with a policy more suited to each position. The second approach is to blend the probabilities output by different policies. The former method increased the move-matching accuracy by 1%-3%. The latter method the move-matching accuracy by 2%-5%.

There are several directions for future work. Currently, we do not use the search with Maia-S model. We will combine these Maia-S models with tree search as Jacob et al. (2022) did and analyze whether this helps improve predicting human moves in Shogi. As another direction, improvement can be expected by analyzing positions and human moves with low likelihood and incorporating new approaches that can reproduce these moves (e.g., approaches with oversights of moves like humans do). It would also be important to investigate to what extent the likelihood can relects whether a move is likely to be selected by.

ACKNOWLEDGEMENTS

This work was supported by JSPS KAKENHI Grant Numbers JP18H03347 and JP20K12121. We would also like to thank Mindwalk Corp. for providing the Shogi game records on Shogi-Quest.

REFERENCES

Coulom, R. (2007). Computing "Elo ratings" of move patterns in the game of Go. *ICGA journal*, 30(4):198–208.

Fujii, N., Sato, Y., Wakama, H., Kazai, K., and Katayose, H. (2013). Evaluating human-like behaviors of video-game agents autonomously acquired with biological constraints. In *International Conference on Advances in Computer Entertainment Technology*, pages 61–76. Springer.

Hingston, P. (2010). A new design for a turing test for bots. In *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*, pages 345–350.

Hoki, K. and Kaneko, T. (2014). Large-scale optimization for evaluation functions with minimax search. *Journal of Artificial Intelligence Research*, 49:527–568.

Jacob, A. P., Wu, D. J., Farina, G., Lerer, A., Hu, H., Bakhtin, A., Andreas, J., and Brown, N. (2022). Modeling strong and human-like gameplay with KL-regularized search. In *International Conference on Machine Learning*, pages 9695–9728. PMLR.

Kinebuchi, T. and Ito, T. (2015). Shogi program that selects natural moves by considering the flow of preceding moves. In *2015 3rd International Conference on Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence*, pages 79–84. IEEE.

McIlroy-Young, R., Sen, S., Kleinberg, J., and Anderson, A. (2020). Aligning superhuman AI with human behavior: Chess as a model system. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1677–1687.

Obata, T., Sugiyama, T., Hoki, K., and Ito, T. (2010). Consultation algorithm for computer shogi: Move decisions by majority. In *International Conference on Computers and Games*, pages 156–165. Springer.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., and Hassabis, D. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144.

Togelius, J., Yannakakis, G. N., Karakovskiy, S., and Shaker, N. (2013). Assessing believability. In *Believable bots*, pages 215–230. Springer.

Tsuruoka, Y., Yokoyama, D., and Chikayama, T. (2002). Game-tree search algorithm based on realization probability. *ICGA Journal*, 25(3):145–152.