Teaching AI to Play Chess Like People

Austin A. Robinson
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
robi1467@morris.umn.edu

ABSTRACT

Current chess engines have far exceeded the skill of human players. Though chess is far from being considered a "solved" game, like tic-tac-toe or checkers, a new goal is to create a chess engine that plays chess like humans, making moves that a person in a specific skill range would make and making the same types of mistakes a player would make. In 2020 a new chess engine called Maia was released where through training on thousand of online chess games, Maia was able to capture the specific play styles of players in certain skill ranges. Maia was also able to outperform two of the top chess engines available in human-move prediction, showing that Maia can play more like a human than other chess engines.

Keywords

Chess, Maia, Human-AI Interaction, Monte Carlo Tree Search, Neural Network, Deep Learning

1. INTRODUCTION

Chess is a problem that computer scientists have been attempting to solve for nearly 70 years now. The first goal was to develop an algorithm that was able to play chess. The first chess engine that was able to accomplish playing was created in the late 1950's [15]. Over the next 40 years, chess engines would gradually get better until a significant milestone was surpassed; a chess engine was able to defeat the world's best chess player at the time. In 1997 Deep Blue, a chess engine developed by IBM was able to defeat Garry Kasparov [16]. Since Garry Kasparov's loss to Deep Blue, computers have been considered better than people at chess.

Since then, chess engines have become much stronger; they are finding moves that the world's best chess players would never be able to find. This is an issue in the chess community since many people use chess engines to further improve their game, through analysis, or through practice. Recently researchers have been attempting to create a chess engine that will play like a person. This chess engine is called Maia. The goal of Maia is not to play chess in the best way possible, but to play chess in the most human way possible.

In this paper, Section 2 will discuss terminology regarding chess and concepts related to Deep Learning that will be necessary to understand for this paper. Section 3 will be discussing the impact of a chess engine that utilized Deep Learning to become a powerful chess engine. Section 4 will be discussing the result of training AI with human games to create a more human-like AI. Lastly, Section 5 will discuss research on human and AI interaction alongside with possible application of a chess engine that plays more human-like.

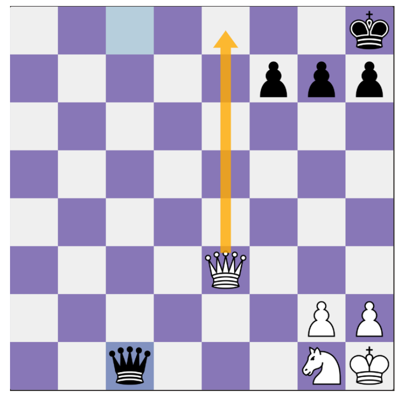

Figure 1: An example of a blunder made by black when they moved their queen from the top row to the bottom row indicated by the blue tinted square. This move that allows white to win in one move. If white moves their queen to the top row, than that is checkmate and black loses the game [2].

2. BACKGROUND

In this section, I will cover key definitions and concepts that will be important in Section 3 and Section 4. Section 2.1 will cover basic chess terminology, Section 2.2 will cover concepts surrounding Deep Learning, and 2.3 will cover two commonly used algorithms related to chess engines.

2.1 Chess Terminology

There are hundreds of different terms used in chess, but for the scope of this paper, the terms listed in this section will be the only ones needed to know to understand discussions surrounding chess.

In chess, a person's skill is quantified by an Elo rating. The name Elo is from the last name of the physics professor who developed this system, Arpad Elo. At an Elo of 2000, a person is considered to be a chess expert. In a game of chess, the player with a higher Elo rating is usually expected to win. For example, a person with an Elo rating of 100 points

UMM CSci Senior Seminar Conference, April 2021 Morris, MN.

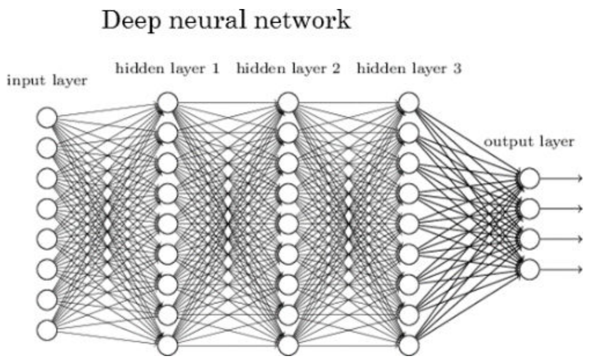

Figure 2: An example of a deep neural network [4]

greater than their opponent is expected to win 64% of the matches against their opponent [18]. Magnus Carlsen, the current number one player, has an Elo rating of 2847 as of the time this paper was written [3]. Elo rating is also used as a predictor for who will likely win a match.

Time control, where time control refers to how much time a player has to make all of their moves. An example of this would be the time control 5 + 3. The 5 indicates that each player starts with 5 minutes, and the 3 indicates that each player gets an additional 3 seconds added to their time after each move they make.

A blunder refers to an error made in a game that causes the player to lose a piece or causes the player to lose the game. A mistake in chess refers to a move that worsens their position [14]. Figure 1 demonstrates a blunder in a game that causes the black pieces to lose immediately.

2.2 Deep Learning

There are many fields of Machine Learning; one field is Deep Learning. Deep learning is allowing computers to learn from experience. The computer learns more complicated concepts by building on simpler concepts to reach the more complex concept. In the book *Deep Learning*, by Ian Goodfellow, Yoshua Bengio, and Aaron Courville, the authors state

> The hierarchy of concepts enables the computer to learn complicated concepts by building them out of simpler ones. If we draw a graph showing how these concepts are built on top of each other, the graph is deep, with many layers. For this reason, we call this approach to AI deep learning

In Deep Learning a deep neural network is used; a deep neural network is a network with many layers between the input and output layers [17]. Figure 2 demonstrates what a neural network looks like. As shown in Figure 2 there is an output layer to a neural network; the output for the output layer is decided with the parameters of the neural network. The parameters of a neural network are essentially weights that the neural network decides are important through various methods of deep learning. A method of Deep learning that is will be discussed in this paper is reinforcement learning; reinforcement learning is where an autonomous agent must learn a task through trial and error without any guidance of a human operator. The chess engines AlphaZero and Leela used this technique to develop a strong playstyle (further details in Section 3).

2.3 Algorithms in Chess engines

When developing chess engines there are two main camps. The first is classical evaluations; a classical evaluation is from an algorithm that is hand crafted by chess experts. The second camp is using Deep Learning. In this camp developers only develop a way for a chess engine to learn. In both of these camps some form a tree search will be needed to efficiently search and evaluate moves. A tree search will utilize the common data structure of a tree data structure. The tree data structure is a set of nodes that branch off of a parent node. There are several ways to traverse a tree which is the tree search. In the context of this paper, there will only be two tree searches discussed. The first is Alpha-Beta pruning. The second is a Monte Carlo Tree Search

2.3.1 Alpha-Beta pruning

The goal of Alpha-Beta pruning is to significantly decrease the number of nodes that are evaluated. Alpha-Beta significantly reduces the computation time of a mini-max algorithm by stopping an evaluation for a move if it is discovered that it could lead to a worse situation than a move previously discovered. An example using chess would be is if the algorithm is evaluating a move and then discovers right away that, that move can lead to the opponent getting a checkmate, the algorithm will stop evaluating and move on to a different move to evaluate. Saving time with the search since all other moves from the parent move would be redundant.

2.3.2 Monte Carlo Tree Search

Monte Carlo Tree Search (MCTS) is different than alpha-beta pruning given that MCTS combines a tree search alongside principles of reinforcement learning. Through reinforcement learning, the tree for the MCTS will adjust its weights for a move based on previous games that the MCTS has been a part of. A more traditional chess engine like Stockfish uses alpha-beta pruning for move selection, while chess engines like Leela and AlphaZero will utilize MCTS in the game to determine the best move. An important thing to note with a MCTS is that the strength of move selection is entirely determined by the training the chess engine had. So a chess engine utilizing a MCTS with little training will likely perform poorly while a chess engine with a significant amount of training is more likely to perform well.

As shown in Figure 3, MCTS has four primary steps: Selection, Expansion, Simulation and Backpropagation. In the Selection phase, the algorithm for the MCTS will traverse the tree, and eventually, a node at the end of the tree will be returned based on various parameters. In the Expansion phase, a new node is added to the node chosen in the Selection phase. This then leads to the Simulation phase. In the Simulation phase, moves or strategies are chosen until a result is achieved. In this paper, the Simulation phase is the most important to understand. In the simulation phase, the MCTS will simulate a set number of games to determine the best move in a given position. Figure 4 demonstrates how a move is selected based on game simulation. The last phase is Backpropagation. After a node is added to the tree, the tree is updated with values determined by the simulation phase [9].

2.4 Chess Engines

A chess engine is software that is used to analyze chess positions, and also generate moves that it determines to be

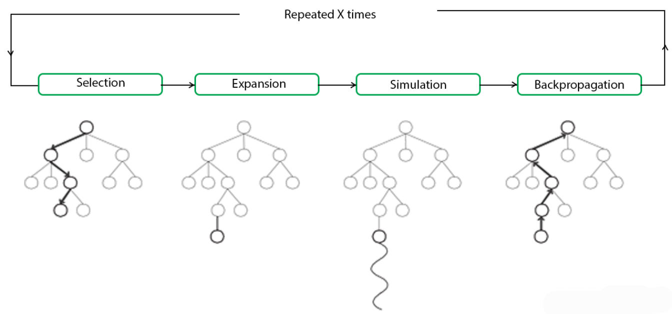

Figure 3: An example of a Monte Carlo Tree Search. In this figure, the four main steps can be repeated X amount of times [9].

the best [13]. In this section, the chess engines Stockfish, Leela, and AlphaZero will be introduced.

2.4.1 *Stockfish*

Stockfish is considered one of the best chess engines in the world as of November 2020 [13]. Often Stockfish will be used to evaluate whether a move is a perfect move, a good move, a mistake, or a blunder. The classification of a move is determined by the evaluation of a position. Stockfish is able to determine which player is winning. If a player plays a move that causes them to be in a position that Stockfish deems to be a losing position then that move would be considered a mistake or a blunder depending on the severity of the move. A good move is a move that helps positively influences their position, but is not the best move according to Stockfish. A perfect move is the move that Stockfish plays in a position. The developers of the Maia chess engine use Stockfish to evaluate moves [8].

2.4.2 *Leela and AlphaZero*

AlphaZero is a unique chess engine developed by Google. Unlike many other chess engines where they usually use classical evaluations, AlphaZero utilizes Deep Learning (Explanation on Deep Learning in Section 2.2). AlphaZero played millions of games by itself while slowly improving until it reached a point where AlphaZero was able to defeat Stockfish in a match-up of 1000 games back in 2017. The final outcome was AlphaZero won 155 games, drew 839 games, and lost 6 games [10].

Leela, formally known as Leela Chess Zero, is an open-source implementation of AlphaZero. Leela used a similar process that AlphaZero used, such as utilizing reinforcement learning through self-play to become a strong chess engine. In 2017, Leela was able to beat the version of Stockfish that AlphaZero originally defeated.

3. DEEP LEARNING IN CHESS

As stated in Section 2.4.2, both AlphaZero and Leela utilize Deep Learning to be powerful chess engines. This section will go into further detail on how Deep Learning in chess is accomplished and the results of the chess engines using Deep Learning.

3.1 Architecture of AlphaZero

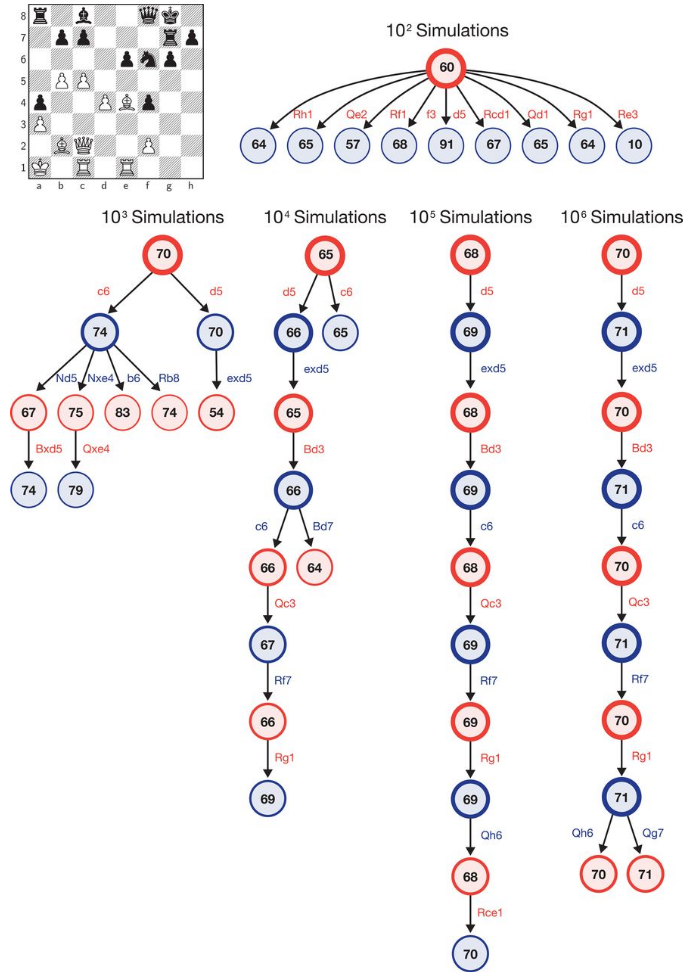

Figure 4: This figure shows a Monte Carlo Tree Search being used for a given chess position. Each simulation summary shows the top 10 most visited states and the estimated value for a move that was simulated[11].

A chess engine like Stockfish utilizes a human-made algorithm and uses techniques like Alpha-Beta pruning (see Section 2.3.1) while AlphaZero utilizes a Monte Carlo Tree Search and a deep neural network. For the deep neural network, the board positions are the inputs. The output is a vector of move probabilities. The move probability is given a value based on the expected outcome of a game. According to the authors of the report *A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play*, AlphaZero learns these move probabilities and value estimates entirely from self play [11].

Figure 4 shows how AlphaZero would utilize a MCTS algorithm with a given position. In the figure, each state has a value associated with the state. This is determined by the simulations, where at the end of a game, the ending position is scored: -1 for a loss, +1 for a win, and 0 for a draw [11].

3.2 Training

At the beginning of the training, the deep neural network knows nothing about chess other than the rules. Every move at the start of the training was completely random. After each game, the parameters (see Section 2.2 for more details) of the neural network are adjusted as AlphaZero learned from wins, losses, and draw. The training for AlphaZero lasted approximately 9 hours [10].

3.3 Results

AlphaZero was first able to beat the version of Stockfish that won the 2016 Top Chess Engine Championship. Similarly, Leela was also able to beat the same version of Stockfish. An observation that was made by the researcher from analyzing games played by AlphaZero was that AlphaZero was able to learn and discover common strategies and opening moves that humans commonly played. Some of these strategies including relatively elementary ideas like emphasizing protection for the king and some advanced strategies like focus on the position of the pawns. AlphaZero also played strategies that are not common among human players; one of these strategies that AlphaZero used commonly was sacrificing pieces earlier in the game. Sacrificing a piece here refers to willing to give up a piece with the hopes of gaining an advantage from it later on. Human players will often sacrifice pieces as well, but usually, with human players, the player will only sacrifice a piece when an immediate advantage can be seen. AlphaZero was sacrificing pieces when there was no clear advantage to be seen.

4. MAIA CHESS ENGINE

Maia is a new chess engine that was released in 2020. As stated earlier Maia is a chess engine designed to play like a person. The developers of Maia quantify this via the question, "What is the probability that the engine plays the move that the human players in the game?" [6]

4.1 Development

Maia utilizes much of the code from the open-source chess engine, Leela. Maia also utilizes the chess engine Stockfish to find the win probability of human in a given position and to also quantify a blunder or mistake made by a human player. So unlike AlphaZero, Leela, or Stockfish, where they are tasked with finding the best move possible, Maia is tasked with attempting to predict the next move that human in a given skill range would make. Specific skill ranges are vital in order for Maia to be able to accurately predict human moves. People tend to find better or the best move more often when they have a higher Elo rating; the inverse tends to be true with lower-rated players tending to make mistakes more often than the higher-rated players.

Maia repurposed the deep neural network to be trained on human games instead of games with itself (more on the training data in 4.1.1) [8]. One key difference this leads to is a large deviation in skill compared to other top-tier chess engines. Other chess engines like Stockfish or Leela are able to outperform the world's best chess player; these chess engines would carry an Elo rating of above 2800. While Maia will have a rating similar to the data that is given to it. Maia can be separated into several different models categorized by skill ranges in blocks of 100 Elo rating points. These models start at 1100 and go to 1900.

Though Maia repurposes a large amount of code from Leela, Maia makes a key difference when it comes to the architecture of the project. Maia does not utilize a Monte Carlo Tree Search. Rather than a tree search, Maia is given the last 6 moves that each player makes.

Maia Model	Best Elo Range	Accuracy
1100	1200	50.8%
1300	1400	51.8%
1500	1700	52.2%
1700	1800	52.7%
1900	1900	52.9%

Table 1: In this table, the Maia model column represents which skill range a specific version of Maia was trained at. The Best column shows at what specific Elo range a given model of Maia performed the best at in terms of human-move prediction accuracy. The Accuracy column shows the accuracy of the Best column [8]. With the data provided from the developers of Maia only exact numbers for the Maia Models of 1100, 1300, 1500, 1700 and 1900 were provided.

4.1.1 Data Sets

The data for Maia to train on is from lichess.org. Lichess provided millions of games from a wide variety of player skill levels. Some of the parameters for the data that would be used was that the first 10 moves of the game were excluded since often the openings of games of chess are memorized. Games with time controls of 3 minutes or less were also excluded from the data since the players often do not have time to make rational decisions in very little time. After these parameters, the games were then split up into 9 blocks from 1100 to 1900. Each of the blocks had 10,000 games with players in the given skill range. Each of the blocks had roughly 500,000 positions for the computer to learn from.

4.1.2 Testing

As stated earlier to quantify how much like a person Maia plays chess-like, the probability that Maia plays a move that a human play is needed. So with each of the positions in the skill ranges, a percent was gathered from whether or not Maia played the same move as a person. Furthermore, Stockfish was used to determine if the move the human made was a good move or a mistake. It is difficult to predict the move for every person since Maia is essentially the average of every player in a specific Elo range. Another cause is that sometime a player does make a move that can be considered random, but that tends to be less of an issue as a player's Elo rating increases.

4.2 Results

After testing, the results for Maia are impressive in all models. As Table 1 and Figure 5 show, each model of Maia was able to score an accuracy of over 50% in at least one Elo range. Accuracy here refers to how often Maia or any chess engine for the matter is able to play the move a human made. In every test case, there was only one human move that was made for Maia to be tested against. In Figure 5 the two lowest accuracies are held by Maia 1100 and Maia 1900. This indicates that there are unique styles of playing at 1100 Elo rating and a 1900 Elo rating. The idea that 1100 rated players are just playing a combination of random good moves and bad moves is ruled out since it would be near impossible to capture that kind of playstyle of an 1100 rated player and get a move predicting accuracy of over 50% at the

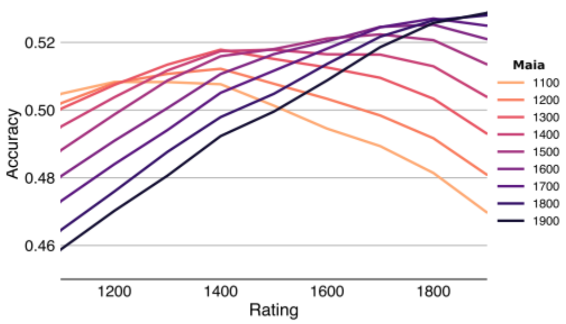

Figure 5: This figure shows Maia's accuracy of predicting human moves through all models of Maia. These results were achieved through testing each model of Maia at each test set. Each test set only contains position from games played in specific Elo ranges [8].

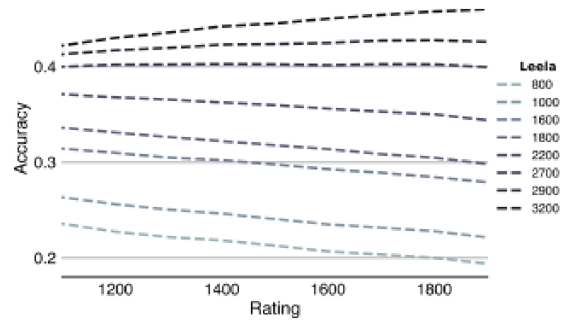

Figure 6: Leela results from human move prediction [8].

1100 Elo rating. Since it has been shown that the model of Maia is able to predict human moves at the 1100 Elo rating this indicates that there is a unique playstyle at the 1100 Elo range [6]. The phrase somewhat unique is used here since the 1100 model of Maia is also able to predict human moves at the 1200 Elo range. An observation that can be made with the data in Figure 5 is that a large portion of the models of Maia has a higher accuracy at predicting human moves at an Elo range higher than the Elo range that the specific model of Maia was trained on. The developers did not give a clear answer on why this trend appeared.

The data in Table 1 and Figure 5 show that as the Elo model of Maia gets higher, and when it is playing at a higher Elo range, the accuracy also gets higher. One reason that the developers gave for the higher accuracy of Maia is that Maia is learning to recognize specific board patterns. Maia is also learning how to solve problems the way humans do in specific skill ranges. The developers also stated that the accuracy of Maia's move prediction depends on the data that Maia has seen. Earlier in a game, Maia is more likely to have seen a position, whereas later in a game Maia is more likely to not have seen a position. The first two moves made by the players would have been seen in the training data 100% of the time, while by move 15 for both players was only seen 0.02% of the time. A consequence of this is that the stage of the game directly influences the accuracy of move prediction [7].

Maia has been able to show that it is able to predict the moves a person would play with an accuracy greater than 50%; Maia is essentially the average player at a given skill level. So it could be inferred that having a much higher accuracy for all players at a skill level would be near impossible especially at lower Elo ranges. The developers did state in a separate article for Microsoft, based on their research, for some personalized models of Maia, they were able to get prediction accuracies upwards of 75% [6].

4.3 Comparison to other Chess engines

Leela and Stockfish were also run through the same test as Maia to see how they perform at predicting human moves. Leela and Stockfish are some of the highest performing chess engines in terms of Elo rating available for testing. Neither of these chess engines was explicitly designed to play chess like a human, but rather to perform at the highest ability possible. This lead to Maia significantly outperforming Leela and Stockfish in human move prediction in the Elo range 1100-1900. Testing was not done outside of this Elo range.

4.3.1 *Leela*

Leela tended to have the most consistent results. Figure 6 shows that all models of Leela tended to stay around the same accuracy range. It is important to note that Leela has its own rating system that is not comparable to the rating systems of Maia. Leela's rating system still does indicate the strength of play though, with higher ratings corresponding to higher skill levels. Essentially the way that Leela gets better is to play more games against itself, so earlier versions of Leela will be closer to random moves (Leela rating of around 800), and later versions will be making moves closer to being the best (Leela rating of around 3200). With earlier versions of Leela playing more random moves, this is likely the low accuracy for the earlier version of Leela. Leela does perform better than Stockfish, with Leela having the best accuracy of around 46%. As Figure 6 show the results of all versions of Leela stay very consistent at all Elo ranges; this shows that Leela is not able to differentiate human skill levels or styles of play. As stated in Section 4.2 there are definitively different play styles across Elo ranges. Leela predicts human moves in the same fashion for all skill levels which indicates that Leela is not able to find the difference in human play styles.

4.3.2 *Stockfish*

Stockfish is one of the world's top chess engines, but in the context of this study, the best performance of Stockfish occurred at an Elo rating of 1900 and with a depth of search of 15. Figure 7 compares various depths of search with Stockfish from an Elo rating of 1100 to 1900. On all depths of search Stockfish's accuracy for prediction improves as the player's Elo rating improves. Stockfish attempts to find the best move possible according to its algorithm. So it is logical that as a player gets better they find the best move more frequently than a player with a lower Elo rating. Lowering the depth of search does cause the chess engine to perform more at the skill level of a person, but as shown in Figure 7, lowering the depth of search to cause Stockfish to be at a specific skill level does not cause Stockfish to capture the style of play of a human at the desired skill level [8]. The algorithm behind Stockfish is not meant to imitate that of a person, but to play the game in the best way possible. So

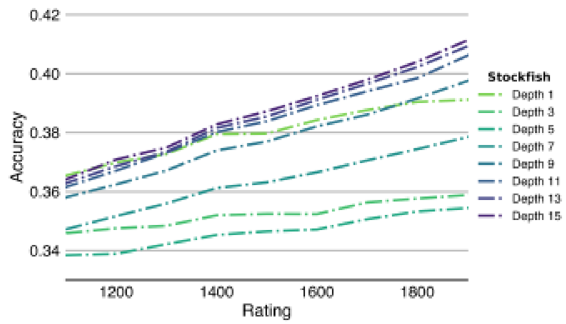

Figure 7: Stockfish results from human move prediction [8].

limiting the depth of search to fewer searches just causes the algorithm to perform more poorly, miss better moves, and just simply make more seemingly random blunders. Maia utilizes Stockfish to determine whether a move is a blunder or a good move, but Stockfish is designed to play the best move possible, thus it is not able to play like a human, making mistakes that humans make and mediocre moves that humans make. Though statistically, it could be at the same level as a person, it will not capture the play style of a human.

5. APPLICATIONS

The application of Maia would seem to be only beneficial to chess, but outside of chess, there has been a growing need for more human-like interactions between humans and AI. AI is becoming ever more prevalent in the life of people every day. With the rise of AI cooperation between humans and AI is becoming more important as well. A chess engine like Maia will be able to support the need of human and AI interactions

5.1 Chess

Chess is one of the most popular games in the world, platforms like Lichess.org or Chess.com see hundreds of thousands of users daily [5]. Chess.com has over 20 million members [1]. On Lichess.org the median Elo rating was around 1500 [8]; this means that most players are not expert and have room for improvement. A chess engine like Maia is designed to play like a person; playing like a person is not having the same skill as a person, but being able to make similar moves as a person. This also includes having the chess engine make the same types of mistakes as a person in a given skill range. Having a more human-like chess engine would be able to help a player to train more effectively. As described earlier in Section 4.2, Maia has been able to have a move prediction accuracy of upwards of 75%. This could help players train by possibly showing where a specific player is often making blunders or mistakes in strategy.

5.2 Human and AI cooperation

A study conducted by Harvard researchers looked at whether reinforcement learning can be used to create a helpful behavior in AI. Through the application of reinforcement learning in the environment of two-player Atari games, the researchers were able to show success with the AI improving the performance of humans [12]. Furthermore, the Harvard researcher were able to develop several different type of AI that each served a different purpose. Some were used to work with the human partner to achieve the high score possible and others were used to simply improve the humans skill level.

6. CONCLUSION

Chess engines have widely been able to outperform human players for the last 15 years. Though the goal has been shifting recently with there being a new chess engine that rather than focus on the skill its focuses on playing more human like. The chess engine Maia has shown that it is possible to capture the play style of chess players in a specific skill range and that it is possible to play similar moves to humans at higher rates than other chess engines. The development of a more human-like chess engine could lead to better, more efficient training for chess players given that in other environments AI is being used in cooperation with humans to improve the skill of humans.

Acknowledgments

I would like to thank Nic McPhee and Elena Machkasova for their advice and feedback. I would also like to thank my friends and family who constantly showed me their support with this paper.

7. REFERENCES

[1] Chess.com. How Many Chess Players Are There In The World?, 2017. https://www.chess.com/article/view/how-many-chess-players-are-there-in-the-world [Online; accessed 12-March-2021].

[2] B. Fischer, S. Margulies, and D. Mosenfelder. *Bobby Fischer Teaches Chess*. Bantam, 1999.

[3] International Chess Federation. Carlsen, Magnus, 2021. https://ratings.fide.com/profile/1503014 [Online; accessed 29-March-2021].

[4] A. Kyrykovych. Deep Neural Networks, 2020. https://www.kdnuggets.com/2020/02/deep-neural-networks.html [Online; accessed 1-April-2021].

[5] Lichess.org. Lichess End of Year Update 2020, 2020. https://lichess.org/blog/X-2TABUAANCqhnH5/lichess-end-of-year-update-2020 [Online; accessed 12-March-2021].

[6] R. McIlroy-Young, A. Anderson, J. Kleinberg, and S. Sen. The human side of AI for chess, 2020. https://www.microsoft.com/en-us/research/blog/the-human-side-of-ai-for-chess/?OCID=$msr_{b}log_{M}aiaChess_{t}w[Online; accessed7-March-2021]$.

[7] R. McIlroy-Young, A. Anderson, J. Kleinberg, and S. Sen. Learning Personalized Models of Human Behavior in Chess, 2021. https://arxiv.org/pdf/2008.10086.pdf [Pre-published; accessed 7-March-2021].

[8] R. McIlroy-Young, S. Sen, J. Kleinberg, and A. Anderson. 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*, KDD '20, page 1677–1687, New York, NY, USA, 2020. Association for Computing Machinery.

[9] R. Roy. ML | Monte Carlo Tree Search (MCTS), 2019. https://www.geeksforgeeks.org/ml-monte-carlo-tree-search-mcts/; online; accessed 7-March-2021].

[10] D. Silver, T. Hubert, S. Julian, and D. Hassabis. AlphaZero: Shedding new light on chess, shogi and Go, 2018. https://deepmind.com/blog/article/alphazero-shedding-new-light-grand-games-chess-shogi-and-go [Online; accessed 30-March-2021].

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

[12] P. Tylkin, G. Radanovic, and D. C. Parkes. Learning Robust Helpful Behaviors in Two-Player Cooperative Atari Environments. *NeurIPS 2020*, 2020.

[13] Wikipedia contributors. Chess engine — Wikipedia, the free encyclopedia, 2020. [Online; accessed 13-March-2021].

[14] Wikipedia contributors. Blunder (chess) — Wikipedia, the free encyclopedia, 2021. [Online; accessed 13-March-2021].

[15] Wikipedia contributors. Computer chess — Wikipedia, the free encyclopedia, 2021. [Online; accessed 24-February-2021].

[16] Wikipedia contributors. Deep Blue versus Garry Kasparov — Wikipedia, the free encyclopedia, 2021. [Online; accessed 22-February-2021].

[17] Wikipedia contributors. Deep learning — Wikipedia, the free encyclopedia, 2021. [Online; accessed 2-April-2021].

[18] Wikipedia contributors. Elo rating system — Wikipedia, the free encyclopedia, 2021. [Online; accessed 13-March-2021].