The Playout Paradox: Balancing Accuracy and Computational Power in AI-Driven Go Game Analysis

Quentin Rendu

Hamburg University of Technology (TUHH), Hamburg, Germany

quentin.rendu@gmail.com

Abstract

Analysing Go games with AI-powered engines is now common practice among amateur and professional Go players. These engines rely on deep artificial neural networks (ANN) and Monte-Carlo tree search (MCTS) to evaluate board positions. For a given ANN, the engine's accuracy increases with the number of positions evaluated during the tree search, or number of visits (often called the number of playouts). Although more playouts lead to more accurate analyses, the relationship between playouts and engine accuracy has not been fully explored. In this study, we analysed thousands of games from amateur and professional players, using different numbers of playouts per move, ranging from 5 to 15,000. A statistical analysis of the results shows that a constant number of playouts per move leads to varying accuracy during the game (high accuracy during the opening and endgame, low accuracy during the middle game). Quantitative guidelines to choose the number of playouts depending on one's rank are derived. Looking at AI prediction of human win rate, the playouts needed to reach 99% accuracy are 140 for 1d, 235 for 5d, and 1,500 for 9d Fox players and 5,500 for a 9p professional player.

Keywords

Go, Baduk, Weiqi, AI, Katago, Playouts.

I Introduction

Go is an abstract strategy game played on a board where two opponents place stones alternately to claim the largest territory. Most of Go's complexity emerges from the size of the board, a 19x19 grid, and its sheer number of possibilities. The branching factor, or the number of legal moves at each turn, is about 35 in Chess and 250 in Go. This makes classical alpha-beta search algorithms, like those used in DeepBlue to defeat the world Chess champion in 1997 (Campbell et al., 2002), unsuitable for Go engines. In 2016, AlphaGo defeated Lee Sedol by combining Artificial Neural Networks (ANN) and Monte Carlo Tree Search (MCTS) (Silver et al., 2016).

Monte Carlo methods are a broad class of algorithms that rely on repeated random sampling instead of exploring every possibility. The first successful implementation of such methods for computer Go was achieved by Coulom (2006). Using a classical tree structure, a position is evaluated by repeatedly running random Go game simulations and averaging their outcomes. In AlphaGo, the score evaluation is given by the value network, an ANN trained through reinforcement learning (AlphaGo playing games against itself). When playing, or analysing, a Go game, AlphaGo then relies on MCTS with a certain number of visits, also called playouts. The number of playouts is the number of board positions that AlphaGo will simulate to accurately predict the current score (for a complete description of the algorithm, see Silver et al. (2017)). More playouts mean that AlphaGo is able to simulate deeper lines and a larger choice of moves before giving its evaluation.

Today, AI-driven Go game analysis is common among both amateur and professional players. AI is now accessible to everyone thanks to open-source software such as Katago (Wu, 2019). Training with AI analysis has since been shown to improve Go players' performance (Shin et al., 2021). Companies offering Go game analyses usually rely on pricing options proportional to the number of playouts. For instance, AI-sensei offers options ranging from 50 to 10,000 playouts per move (Teuber et al., 2023), and OGS from 400 to 12,000 playouts per move (OGS, 2013).

This study aims at understanding the link between playouts and AI accuracy. A statistical analysis is performed on thousands of games to derive quantitative guidelines. Several ranks are evaluated, yielding insights useful to all Go players from amateur (1d Fox) to high-level (9p) professional ranks.

Section II will delve into the methodology, including AI settings and game datasets. In Section III, the link between playouts and AI accuracy will be thoroughly investigated. Particular attention will be brought to the game stage and to the player's rank. In Section IV, solutions to balance accuracy and computational power are discussed. Guidelines for choosing a playout number depending on one's rank are also presented.

II Methods

II.1 Go games datasets

Katago v1.12.4 (Wu, 2019) was used with the neural net ”b18c384nbt-uec-20221121b” for game analysis. The computations were performed on a single laptop equipped with an NVIDIA RTX A3000 Laptop GPU. Games were analysed with a number of playouts ranging from 5 to 15,000. A total of 18 billion board positions were evaluated by Katago, requiring approximately 6,000 hours of computational time ($\approx$ 8 months).

To run the statistical analyses, four datasets were selected, consisting of 3,500 different games in total. Three datasets were randomly sampled from games played on the Fox Go server by 1-dan, 5-dan, and 9-dan players (Featurecat, 2019). Only non-handicap games featuring players of equal strength were selected. Each dataset contains games with the same time settings. Games terminated by a draw, by connection loss, or by time were excluded, considering only games won by score or resignation. The fourth dataset consists of 1,000 games of 9-dan professional players. The games were obtained thanks to the Go4Go professional games database (`https://www.go4go.net/go/`). All the games and their corresponding AI analyses (playouts per move ranging from 5 to 15,000) are available online under an open-source license (Rendu, 2023a).

II.2 Evaluation of Katago's accuracy

In order to evaluate the accuracy of Katago for different numbers of playouts, one would ideally need the true score for each board position. In this work, the true score is approximated by the score obtained with 15,000 playouts. For comparison, the highest tier plans from AI-sensei and OGS use 10,000 and 12,000 playouts respectively. 15,000 playouts is thus sufficiently large to provide a very good approximation of the true score while keeping the computational costs reasonable.

Win rate is another metric that can be used to assess Katago's accuracy. The AI win rate is directly obtained by Katago's value network. It represents, from a given position, the winning probability of Katago when playing against itself. Similarly, the human win rate is defined as the winning probability of a human when playing against an opponent of the same rank (Rendu, 2023b).

To evaluate the influence of playouts, the same games are analysed with different numbers of playouts, namely 5, 50, 500, and 5,000 playouts per move. The score and win rate errors are then computed as the difference between the current analysis and the 15,000 playouts analysis.

III Results

III.1 From Opening to Endgame

Using one dataset for each rank, one can compute the average score of the winner as a function of the move number. The 15,000 playouts datasets were used, and the resulting plot is shown in Figure 1. All the curves gradually increase, starting from 0 and reaching their maximal value between moves 150 and 200. The score lead then decreases until the end of the game. This surprising behaviour is due to resigned games being included in the datasets along with games that finished by counting. Resigned games often have larger score leads and do not contribute to the average score lead beyond the resignation move. Due to the resigned games, the averaging is done over a small sample of games for high move numbers, leading to noisier curves.

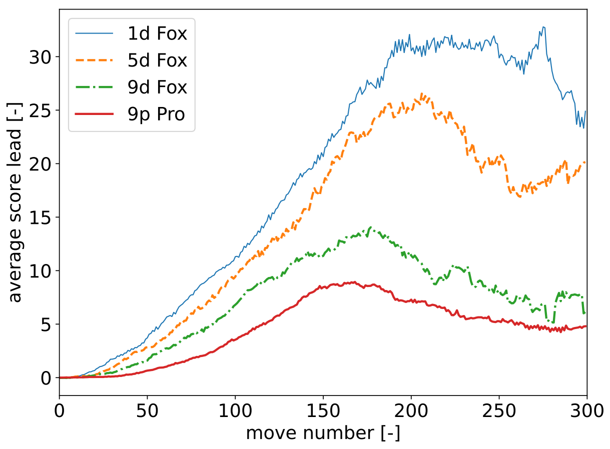


Figure 1 Average score lead of the winner plotted along move number for different player's rank - 1d, 5d and 9d Fox correspond to amateur games played on Fox Go server, 9p to games between 9-dan professional players

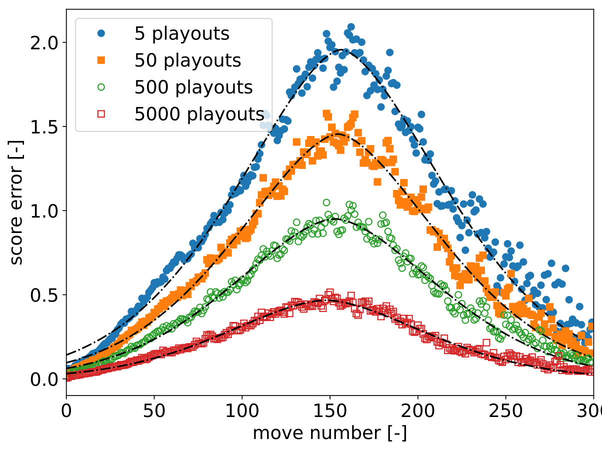


Figure 2 Average score error against move number for different numbers of playouts - data (9p professional games) is represented by scatter plots, dashed lines show a generalised (symmetric) normal distribution fit

By comparing the different curves, one can see that the score lead is higher for the players of lower rank, at any given move number. The curves for 1d and 5d Fox players are the most similar, although the maximal average score lead is around 30 points for 1d players and 25 points for 5d players. 9d Fox players show a great gap with 5d players, with a maximal average score lead slightly below 15 points. The 9p professional players show the lowest average score lead with a maximum between 5 and 10 points. This link between rank and score lead shows that stronger players make fewer mistakes and tend to play tighter games.

The location of the maximum is also affected by the players' rank. It is around move 160 for professional players, around 175 for 9d Fox players and close to 200 for 5d and 1d Fox players. As a possible explanation, it is harder to overturn a game against a stronger player. This might lead to earlier resignation (hence shorter games) for higher-ranked players when compared to lower-ranked players.

Focusing now on the 9-dan professional games, the error on score and win rate are evaluated as described in Section II.2. The average of the score error over the 1,000 games is plotted against move number in Figure 2. The scatter plot corresponds to the raw data, while the dashed lines show a generalised (symmetric) normal distribution fit. This plot is a direct measure of Katago's accuracy.

First of all, it can be observed that the score error decreases when the number of playouts increases. With 5 playouts, a very low value, the maximal error reaches 2 points, whereas with 5,000 playouts the maximal error is only 0.5 points (both around move 150). In other words, more playouts yield more accuracy, which should be familiar to all AI Go engine users.

More interestingly, all the curves show a bell pattern. The score error is very low during the opening (before move 50), it increases and reaches a maximum in the middle game (around move 150) and decreases

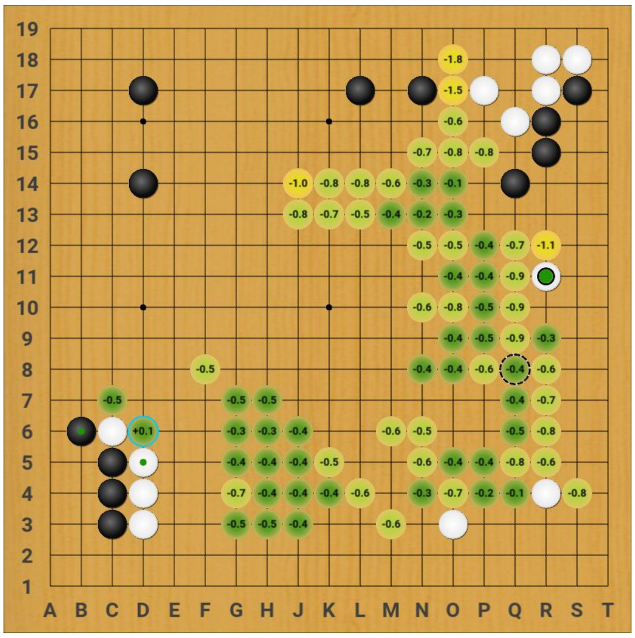


Figure 3 Katago evaluation (10,000 playouts) at move 24 of a professional game between Park Junghwan 9p and Weon Seongjin 9p - each candidate move is labelled with the score, the best move is circled in blue (D6)

until reaching very low values during the endgame (after move 250). It shows that Katago's accuracy is highest during the opening and endgame, and lowest during the middle game.

During the Go endgame, the main objective is to count the value of the remaining moves in order to prioritise them. On top of this, the number of possibilities is greatly reduced compared to the opening or the middle game. It is thus not surprising that Katago is very accurate in the endgame.

The great accuracy of Katago in the opening might be less intuitive, as it is regarded as a very strategic stage where the move choice is driven by intuition rather than calculation. However, lots of options are available at this early stage, and many moves will have a similar value, very close to the best one. As an illustration, a Katago evaluation at move 24 of a professional game played between Park Junghwan 9p and Weon Seongjin 9p is shown in Figure 3. The last move was played by White at R11, Black (Weon Seongjin 9p) plays next at Q8 (dashed circle). 10,000 playouts are used to evaluate evenly 76 different moves. 95% of the candidate moves are within 1 point of the best move, and 50% within 0.5 points of the best move. This large number of excellent moves in the opening stage explains Katago's great accuracy.

The middle game is undoubtedly the most complex stage of a Go game. It requires both tactical and strategic skills, as well as good timing abilities. The status of the different areas of the board might swing from one side to another, groups might be sacrificed in order to get a bigger advantage elsewhere, ko will appear, or threaten to appear, leaving most groups' status as unclear. At a time when humans were still stronger than AI, Stern et al. (2006) and Coulom (2007) showed that the middle game was the hardest to predict. Figure 2 shows that it's the stage at which Katago is the least accurate for a given amount of playouts. In other words,

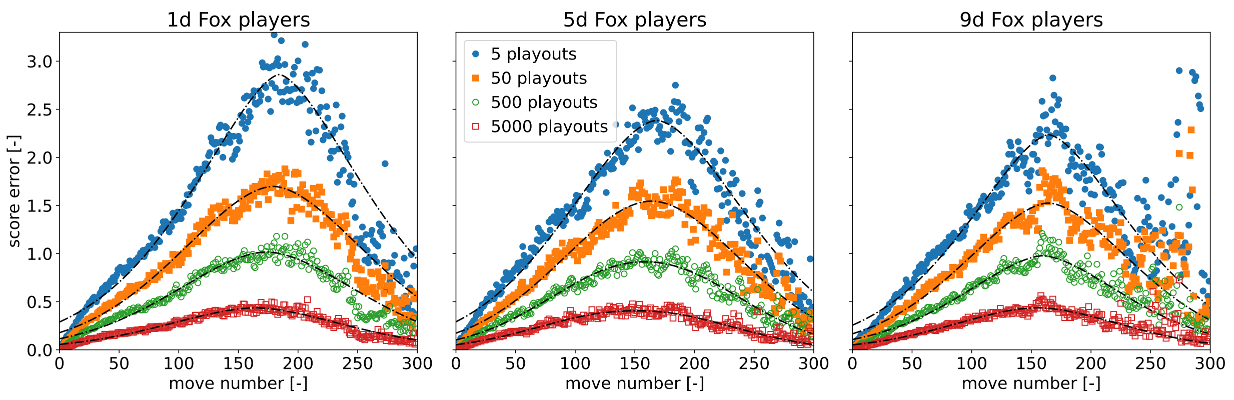


Figure 4 Average score error against move number for different number of playouts - data (amateur Fox games) is represented by scatter plots, dashed lines show a generalised (symmetric) normal distribution fit

Katago needs to evaluate more board positions to give an accurate score prediction during the middle game than during the opening. Looking again at Figure 2, a score estimation with a 0.5 point error requires 5 playouts during the opening (move 50) but 5,000 playouts during the middle game (move 150).

III.2 Impact of Player's Rank - an AI perspective

In this section, the influence of the player's rank on the accuracy of Katago evaluation is studied. There are two ways of addressing this question. The first one, called the AI perspective, is to see if the error on the score prediction is affected by the player's rank. This would be the case if games from strong amateurs are more complex and hence harder to analyse. To test this hypothesis, Figure 4 shows the score error against move number for 1d, 5d, and 9d Fox players. For 5 playouts data, the score error is larger for 1d Fox players than for 5d and 9d Fox players. The maximum score error is 2.8 points for 1d compared to 2.2 points for 5d and 9d. For 50 playouts data, the maximum score error is slightly larger for 1d, namely 1.7 points compared to 1.5 points for 5d and 9d. The curves corresponding to 500 and 5,000 playouts have the same maximum score error and almost identical shape. Overall, there is not a significant difference between the curves, indicating that the players' rank does not impact Katago accuracy.

The link between the score error and the number of playouts can thus be derived without taking into account the player's rank. Such a correlation is plotted in Figure 5. Notice that the y-axis is in logarithmic scale. The curves corresponding to the different ranks are not significantly different from each other and show a similar behaviour. The score error is found to be equal to 2.5 points when running an analysis with 1 playout, and to decrease by 0.5 points each time the number of playouts is multiplied by 10. According to this plot, 100,000 playouts would be needed to reach perfect accuracy.

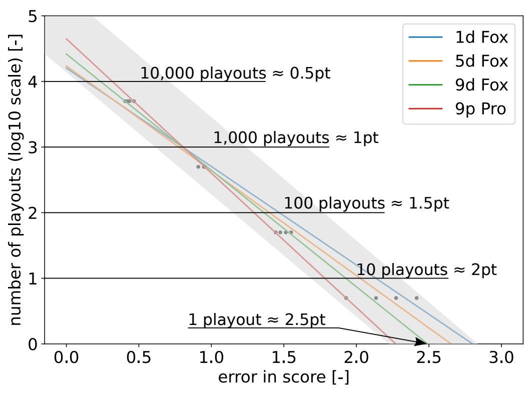


Figure 5 Playouts vs maximal score error (move 150) for different ranks - the ranks all behave similarly, following an exponential regression represented by the grey area (y-axis in logarithmic scale)

III.3 Impact of Player's Rank - a Human perspective

Now, let us ask the same question from a human perspective: as a Go player, how many playouts should I use to analyse my games accurately? Well, this time, it depends on your rank! So far, Katago's accuracy has been evaluated using the score prediction. The same conclusions could have been drawn by looking at AI win rate, another commonly used output of AI Go engines. However, a 1 point mistake will not have the same consequences for a 1d and a 9d player. To predict how a mistake will affect a player's winning probability depending on their rank is exactly what the human win rate has been developed for (Rendu, 2023b).

The human win rate generic formula has been derived for move numbers between 50 and 200. It is used in this work for move numbers ranging from 0 to 300, sometimes yielding noisy data, especially at high move numbers. Katago's error in human win rate prediction is plotted in Figure 6. Like the score error curves, one can observe that the error is smaller for a higher number of playouts. The curves' shape is also similar, starting at a low value, reaching a maximum around move 150 and then decreasing. The large values found at move numbers higher than 200 are most certainly extrapolation errors (using the human win rate formula outside of its designed range). Interestingly, the curves are much flatter near their maximum value than score error curves. This behaviour is observed for all ranks and is thus likely significant. Last, but not least, the value of the error scales with the player's rank, especially for 5d, 9d and 9p players. Looking at 1d and 5d players, the maximal human win rate error when using 5 playouts is 2%, whereas it is 4.5% for 9p professional players, who would need 500 playouts (100 times more!) to reach the same error of 2%.

To draw definite conclusions, the number of playouts is plotted against the human win rate error in Figure 7a for middle game (move 150) and in Figure 7b for opening and endgame (move < 50 or move > 250). Once again, the y-axis is in logarithmic scale. The curves for 1d and 5d players are almost identical for middle game, and very close to each other for openging and endgame, indicating that the effect of playouts

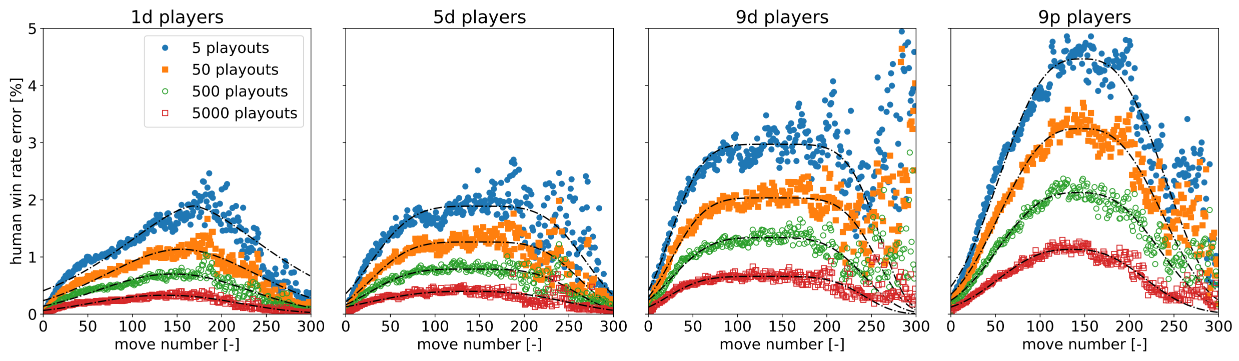
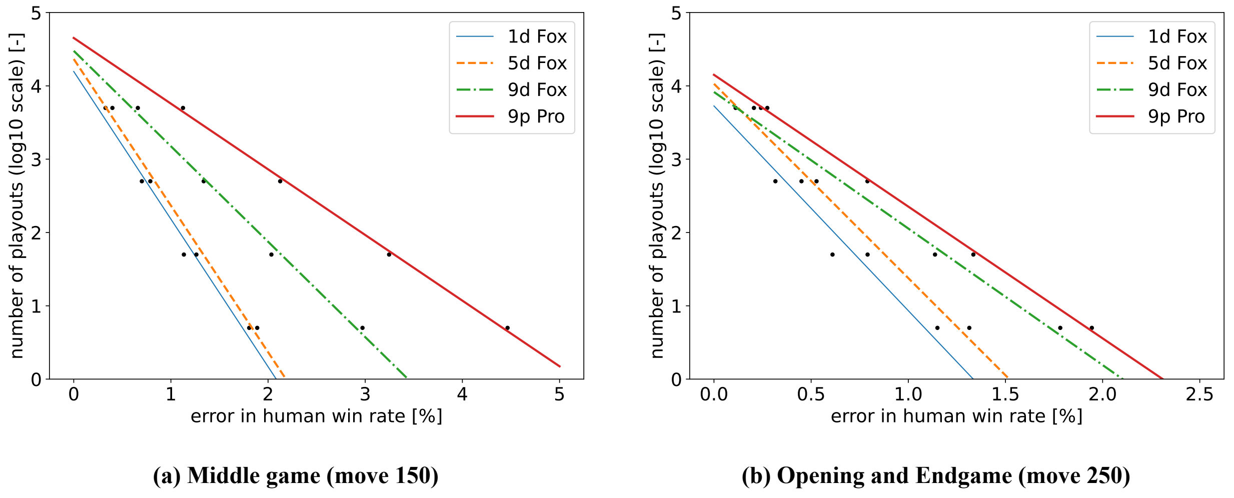


Figure 6 Average human win rate error against move number for different number of playouts and different ranks - data is represented by scatter plots, dashed lines show a generalised (symmetric) normal distribution fit

is not very significant below 5d Fox rank. When comparing 5d, 9d and 9p players, the curves are strikingly different from each other. Looking at middle game, with 1 playout, the expected error in human win rate for 5d, 9d and 9p players is 2%, 3.5% and 5% respectively. Whereas 5d players get a 2% error almost for free (1 playout), 9d players need 100 playouts and professional players 1,000 playouts. Reaching a 1% error would require 500 playouts for a 5d player, 5,000 for a 9d player and more than 10,000 for a professional player. Figure 7b show a similar behaviour with a smaller difference between 9d and 9p players.

IV Balancing AI Accuracy and Computational Power

The results presented in the previous section can be used to improve AI-driven Go game analysis. An interesting finding in Section III.1 is the variation of AI accuracy from the opening to the endgame. AI requires fewer playouts to achieve good accuracy in the opening and endgame but needs significantly more during the middle game, the most complex part. State-of-the-art Go engines use a fixed number of playouts per move to analyse a game, causing AI accuracy to vary with each move. This is represented on the left of Figure 8 under the label ”Current AI analysis”. A more satisfying use of AI would be to spend few playouts



(a) Middle game (move 150)

(b) Opening and Endgame (move 250)

Figure 7 Playouts vs human win rate error for different ranks - dots represent data, lines show an exponential fit

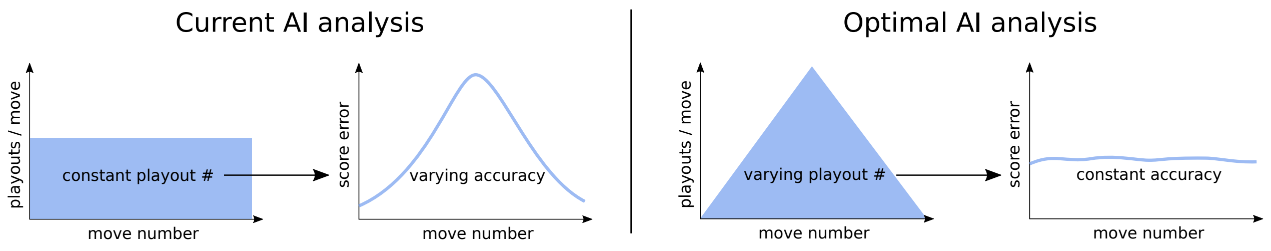


Figure 8 Current versus optimal use of AI Go engine for a constant distribution of accuracy with the same computational cost

in the opening and the endgame, and a large amount of playouts during the middle game. If the distribution of playouts per move along move number is correctly set up, it would result in a constant AI accuracy throughout all the stages of the game. This is presented on the right of Figure 8 under the label ”Optimal AI analysis”. The area under the curve playouts/move, shaded in blue in Figure 8, represents the total number of playouts, and is directly proportional to the computational cost. Both areas are equal, meaning that the optimal AI analysis has the same computational cost as the current AI analysis. In other words, it is not about increasing power to be more accurate, it is about balancing computational costs to reach constant AI accuracy.

The other way to optimally use computational power is to link AI accuracy with the number of playouts for different players’ rank. It is intuitively accepted that professional players need deeper AI analysis than amateur players. The results presented in Section III.3 allow quantitative conclusions. The number of playouts needed to reach different maximal average error in human win rate are presented in Table 1. This Table can be used as a guideline when choosing the number of playouts. For players up to 5d Fox rank, 1% error is a very good compromise. It is twice as accurate as 2% error for a small extra cost of 200 playouts. 0.5% is twice as accurate but at the extra cost of thousands of playouts. The cost for 9d Fox players is already significantly higher. 300 playouts yield a 1.5% error, and reaching 1% needs 1,500 playouts, five times more. A good compromise might be to screen the game with a small number of playouts (typically 300) and to put more computational power into a few complex board positions. Professional players need in any case thousands of playouts to reach satisfactory accuracy. 5,000 playouts, yielding 99% AI accuracy, seems a good compromise. Note that these guidelines are obtained using the maximal average error, which corresponds to middle game board positions. In the opening and the endgame, much fewer playouts are needed to reach great AI accuracy (99% AI accuracy using 250 playouts for 9p professional players, see Figure 7b).

Rank	3% error	2% error	1.5% error	1% error	0.5% error
1d Fox	0	2	15	140	1,500
5d Fox	0	2	23	235	2,300
9d Fox	4	70	300	1,500	7,000
9p Pro	90	700	2,000	5,500	16,000

Table 1 Number of playouts needed to reach maximal (middle game) average error in human win rate for players of different ranks (1d Fox to 9p professional)

V Conclusions

The link between playouts and AI accuracy has been investigated through statistical analysis. All the games and their corresponding AI analysis are available online under open source license.

An important finding is the impact of the game stage. AI Go engines are very accurate in the opening and the endgame, and less accurate during the middle game. This means that using a constant number of playouts per move results in varying accuracy throughout the game. To obtain a constant accuracy, it is recommended to use fewer playouts during the opening and the endgame, and more during the middle game (this can be done without affecting the total computational cost).

Another important result is the quantitative link between player's rank, playouts and AI accuracy. When looking at human win rate, a 99% accuracy is obtained using 140/235/1,500/5,500 for 1d/5d/9d/9p players. It confirms the intuitive idea that stronger players need a higher number of playouts. It also gives a quantitative guideline to help players choose the best compromise between accuracy and computational power.

References

(2013). Online go server. https://online-go.com/.

Campbell, M., Hoane, A., and hsiung Hsu, F. (2002). Deep blue. *Artificial Intelligence*, 134(1):57–83.

Coulom, R. (2006). Efficient selectivity and backup operators in monte-carlo tree search. In *International conference on computers and games*, pages 72–83. Springer.

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

Featurecat (2019). Go dataset. https://github.com/featurecat/go-dataset.

Rendu, Q. (2023a). Analysed kifu database. https://gitlab.com/qrendu/analysed-kifu-database.

Rendu, Q. (2023b). Go players should not trust AI win rate. *Journal of Go Studies*, 17(2):61–88.

Shin, M., Kim, J., and Kim, M. (2021). Human learning from artificial intelligence: evidence from human go players' decisions after alphago. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 43(43).

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359.

Stern, D., Herbrich, R., and Graepel, T. (2006). Bayesian pattern ranking for move prediction in the game of go. In *Proceedings of the 23rd international conference on Machine learning*, pages 873–880.

Teuber, B., Ouchterlony, E., and Dohme, M. (2023). Ai sensei. https://ai-sensei.com/.

Wu, D. J. (2019). Accelerating self-play learning in go. *arXiv preprint arXiv:1902.10565*.

Received: 30, Apr, 2025
Accepted: 15, May, 2025