바둑학연구, 제17권 제2호, pp. 61-88
Journal of Go Studies
2023. Vol.17, No.2. pp. 61-88

# Go players should not trust AI win rate

Quentin Rendu*)
Hamburg University of Technology (TUHH), Germany

**Abstract:** The advent of artificial intelligence (AI) has transformed the landscape of various strategic games, including Go. In 2016, the AIpowered engine AlphaGo defeated one of the world's strongest players. Since then, Go engines have routinely been used by amateur and professional Go players to analyse theirgames. In the early stages of AI analysis, Go players relied solely on the AI win rate, the only availableindicator. However, the AI win rate does not accurately reflect the win rate of human Go players and might be misleading.

Katago, first released in 2019, is the first engine to provide score predictions in addition to win rates. While it is now possible to evaluate board positions with a score, it remains unclear how this score translates into human win rates. In this work, a large database of online and professional games is analysed to extract the win rate of a human player based on their strength and the stage of the game. As expected, the human win rate is significantly lower than the AI win rate, even for 9dan professionalplayers. A general for-

*) quentin.rendu@gmail.com

mula is provided to compute the win rate based on player strength and move number. This feature offers new insights into the relative importance of mistakes and can assist players in makingimproved decisions during games.

**Keywords:** Go, Baduk, Weiqi, AI, Katago, Win rate, Statistics.

# Ⅰ. Introduction

Spoiler alert: humans are no longer the strongest Go players. Go is an abstract strategy board gamethat has been played for thousands of years. Nowadays, it is predominantly played on a 19x19 grid wheretwo players alternately place Black or White stones. The grid starts empty, with the stones not moving during the game, and the objective is to encircle a larger area than the opponent. At the game's conclusion, each player receives one point for each stone on the board and one point per intersection in controlled areas. An example of a finished game on a smaller board is shown in Figure 1.

Slightly before computers were a thing, artificial intelligence (AI) was born. Alan Turing, widely recognised as the father of modern computer science, designed an algorithm for a Chess engine as early as 1948 (Kasparov and Friedel, 2017). With the rapid increase in computational power, it soon became possible to explore millions of positions and determine the move leading to the best result. This tree search algorithm was an important part of Deep Blue (Campbell et al., 2002), the first Chess engine to defeat a World Chess Champion in 1997.

Go, on the other hand, is renowned among both players and computer scientists for its sheer num ber of possible moves. The branching factor in Go is significantly larger than that in Checkers, Chess, or Shogi, rendering pure tree search algorithms inefficient. A similar complexity in branching is also found in Backgammon, due to the numerous possible outcomes of dice rolls. To address this challenge, Tesauro et al. (1995) developed a Backgammon engine using artificial neural networks (ANN) trained through reinforcement

learning. Starting with minimal knowledge, the engine played games against itself and adopted successful strategies. This approach enabled TDGammon to attain a worldclass level in Backgammon.

A similar concept found success in Go. The first Go engine to defeat a worldclass champion was AlphaGo (Silver et al., 2016), which relies on a database of human games and two ANN known as policy and value networks, refined through reinforcement learning. The policy network examines each move and provides the corresponding probability of winning for that move. Initially trained with human knowl

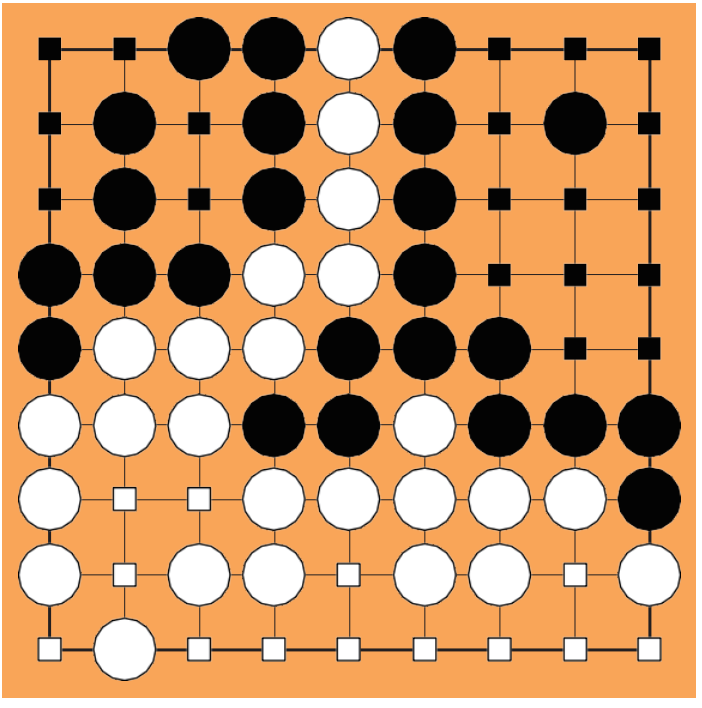

**Figure 1 Finished game on a 9x9 Go board. Black player scores 43 points (24 stones and 19 controlledintersections) whereas White player scores 38 points (25 stones and 13 controlled intersections). Withoutkomi, Black wins the game by 5 points.**

| Level | Main time | AI visits | Games number |
|---|---|---|---|
| 12k Fox | 20 min | 5 | 28,595 |
| 10k Fox | 5 min | 5 | 27,830 |
| 8k Fox | 5 min | 5 | 25,590 |
| 6k Fox | 5 min | 5 | 28,723 |
| 4k Fox | 5 min | 5 | 31,264 |
| 2k Fox | 5 min | 5 | 26,002 |
| 1d Fox | 20 min | 5 | 51,079 |
| 1d Fox | 20 min | 500 | 11,938 |
| 1d Fox | 5 min | 5 | 52,004 |
| 1d Fox | 1 min | 5 | 41,456 |
| 3d Fox | 5 min | 5 | 37,736 |
| 5d Fox | 5 min | 5 | 25,991 |
| 7d Fox | 5 min | 5 | 35,207 |
| 8d Fox | 1 min | 5 | 24,920 |
| 9d Fox | 1 min | 5 | 28,990 |
| 1p - 5p | - | 5 | 23,159 |
| 9p | - | 5 | 14,842 |
| Total | | | 515,326 |

**Table 1 Details of the analysed kifu database**

edge, it later underwent reinforcement learning. The value network reads the current board position and produces the probability of winning (AI win rate). A year later, a new engine named AlphaZero (Silver et al., 2017) was introduced, surpassing AlphaGo's performance with fewer computational resources and without using any human knowledge. AlphaZero was then extended to Chess and Shogi (Silver et al., 2018), once again outperforming the top AIpowered engines.

Since then, Go engines entered the daily routine of amateurs and professional Go players. While quite strong to serve as sparring partners, these

engines aid in analysing games and positions. Shin et al. (2021) showed that players who utilise AI for reviewing their games exhibit improved performance during gameplay. AlphaZero even revisited fundamental sequences and principles taught to every Go player (Baker and Hui, 2017). Similar advancements occurred in Chess (Sadler and Regan, 2019), underscoring AI's potential to enhance game understanding.

Post game analysis is widely employed to improve at strategy games. The main idea involves review ing games and seeking feedback from opponents or stronger players to identify errors. Leela Zero (Pas cutto, 2017), an open-source implementation of the AlphaZero algorithm, attained superhuman strength in 2017, becoming a staple for Go players to analyse their games. Leela Zero exclusively provides AI win rates for evaluating board positions. In 2019, a novel engine called Katago (Wu, 2019) was relased. For a given position, Katago provides both AI win rate and score, a feature that quickly gained popularity. While AI win rates broadly represent the Go engine's probability of winning against itself, an abstract concept, scores offer a more tangible metric for Go players.

In games like Chess and Shogi, evaluating positions necessitates considering factors such as material advantage, piece activity, and king safety. These factors can be combined into a score, which is then transformed into a win rate using an evaluation curve (Takeuchi et al., 2007). The creation of an evaluation function based on heuristic features has also been assessed in the early stages of computer Go (Bouzy and Cazenave, 2001; Müller, 2002).

In Go, players directly evaluate score differences during games. This involves estimating the final state of the board and counting the intersections belonging to each player. However, knowing the score of the current position, even perfectly, is not enough. Predicting the game's outcome based on

score is a complex task, depending on the strength of the players as well as the current stage of the game. Some strong amateur players might assert that overturning a 20point lead in the endgame is virtually impossi ble. Others might emphasise that resigning is the only move that guarantees a 100% chance of losing the game.

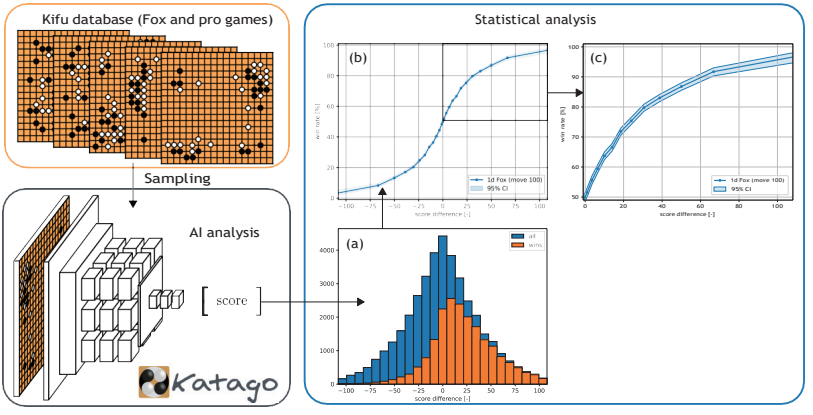


**Figure 2 Summary of the developped method. A large set of games is selected and analysed with Katago, outputting score at each move. Distributions of total and won games are plotted against score (a). Ratio of wins on total games gives the average win rate at each score (b). 95% confidence interval is computed using Beta distribution (b). Symmetry is enforced (c). Plots at move 100 for a 1d Fox player.**

In order to design a useful metric for analysing Go games, this study aims to compute win rates using an extensive database of Go games, encompassing varying player strengths.

Section II will delve into the methodology, AI settings, and game databases. In Section III, the human win rate will be compared to AI win rate, with an exploration of the game stage's influence. A general formula will also be proposed to calculate human win rates based on player strength and game stage (formally move number). In Section IV, we examine how the human

win rate can enhance both the learning and decisionmaking processes of Go players.

# Ⅱ. Methods

## 1. Building the Database

**AIpowered Go engine** Katago v1.12.4 (Wu, 2019) was used with the neural net ”b18c384nbtuec 20221121b” for game analysis. Most of the games underwent analysis with 5 visits, meaning that at each position, the tree search explored 5 additional positions. A smaller sample of games was analysed with 500 visits to validate the methodology. The computations were performed on a single laptop equipped with NVIDIA RTX A3000 Laptop GPU. Half a million games were analysed, requiring approximately 800 hours of computational time (≈ 33 days).

**Go games database** The statistical analysis conducted in this study requires an extensive collection of analysed Go games. To construct such a database, online Go games played on the Fox Go server between 2015 and 2019 were utilised (Featurecat, 2019). Only nonhandicap games featuring players of equal strength were selected. The players’ strengths range from 12kyu to 9dan. The majority of games have a main time of 5 minutes per player. Games terminated by a draw, by connection loss or by time were excluded, considering only games won by score or resignation.

A total of 477,325 analysed kifu based on online games were compiled. This database has been made available online under an opensource license (Rendu, 2023). The quantity of games per skill level, along with the time set-

tings and the number of visits, is provided in Table 1.

To evaluate the difference between online and real games, the Go4Go (https://www.go4go.net/

go/) database of professional games was employed. A total of 38,001 games were analysed using 5 visits, with player strengths either equal to 9p (14,842 games) or falling within the range 1p5p (23,159 games). These games were downloaded through a commercial license and could not be made available online.

## 2. From Go Games to Human Win Rate

In this section, the method developed to compute the human win rate is presented. A visual summary of the method is provided in Figure 2.

First, games that meet the criteria (no handicap, equal strength, etc.) are chosen from professional (Go4Go) and online games (Fox Go server) to create a kifu database. The games are then analysed by Katago, which produces the AI win rate and the score at each move.

Using the database of analysed games, one can select all the games for a given player strength (e.g., 1d) and generate a histogram of the score difference at a specified move number (e.g., 100). The resulting plot is shown in Figure 2a. Within each score bin, the histogram displays the number of won games ($N_{wins}$ in orange) and the total number of games ($N_{games}$ in blue).

The probability of a twooutcome event (winning and losing) can be estimated using the Beta distri bution, based on the count of previous successes ($N_{wins}$, the number of games won) and the total number of games ($N_{games}$). The expected win rate is calculated by the formula:

$$\text{win rate} = \frac{N_{wins}}{N_{games}} \tag{1}$$

The uncertainty in the win rate primarily depends on the number of games ($N_{games}$) and can be readily calculated through the Beta distribution. Unless otherwise specified, the results presented in this work are derived from an average of 2000 games per bin, resulting in a 95% confidence interval of approximately

±2%. A typical win rate curve against the score is plotted along with its corresponding 95% confidence interval in Figure 2b.

**Enforcing symmetry** The win rates computed from data slightly differ between Black and White players. This disparity may be attributed to statistical noise or unaccounted factors, such as the correct komi

value for a fair game, the matching algorithm potentially favoring White for the stronger player, or even psychological effects. However, this kind of analysis is beyond the scope of this study.

To calculate the win rate without regard to the player's color, the win rate of the leading player in the game is sought. For each game, the absolute value of the score is calculated, as well as a boolean set to true if the leader won the game. The range of positive scores is then divided into bins, where the counts of wins ($N_{wins}$) and the total number of games ($N_{games}$) are obtained. For symmetric win rate curves, all the information is contained in the upper right quadrant as shown in Figure 2c.

**Dataset size and bins number** Once the dataset is sufficiently large, the computed win rate should not depend on the dataset size. To assess the convergence of our statistical analysis, the win rate is plotted against the score for varying dataset sizes in Figure 3. The score range has been divided into 12 bins for this analysis. Using only 6,000 games (500 games per bin), the win rate curve appears noisy and does not follow a regular sigmoid curve. With 24,000 games (2,000 games per bin), a smooth win rate curve is obtained, nearly identical to the win rate derived from twice as much data (48,000 games). Unless otherwise specified, a minimum dataset size of 24,000 games will be employed in this study, ensuring the statistical convergence of the analyses.

To avoid binning the data, alternative methods for computing the win rate were explored. One ap proach involves fitting a parametrised probability density function to the score distribution of won games. Using Bayes' theorem, the win rate can then be computed. This approach yielded favorable outcomes for a limited range of move numbers and player strengths, yet failed to generalise across the entire range of investigation.

Relying on discrete bins to compute the win rate is not problematic, as long as the number of bins does not impact the results. Given a dataset of 24,000 games, the win rate is calculated for three different bin numbers: 6 (4,000 games per bin), 12 (2,000 games per bin), and 24 (1,000 games per bin). The resulting curves are displayed in Figure 4. Notably, employing 1,000 games per bin produces a win rate curve with significant noise and a wide confidence interval. With 4,000 games per bin, the curve is smoother, but the data points are relatively distant from each other. Consequently, a value of 2,000 games per bin was selected, corresponding to 12 bins for our minimal dataset of 24,000 games.

**Impact of number of visits** When analysing Go games with AI, one of the most crucial parameters is the number of visits, also referred to as playouts. The number of visits represents the maximum count of board positions evaluated during the Monte Carlo tree search. A higher number of visits ensures greater accuracy in score and AI win rate estimation, at the expense of increased computational costs. While a substantial number of visits is typically necessary for postgame analysis, it may not be essential for statistical analysis. If a lower number of visits augments the variance of AI predictions without introducing bias, it can be anticipated that errors in score predictions will offset one another. To examine this hypothesis, 12,000 games were analysed using 500 visits, requiring 288 hours.

Merely 12 hours (24 times less) are needed to analyse the same database with 5 visits. The calculated win rate is depicted in Figure 5 for both visit numbers. Only 12,000 games with 5 visits are used for a fair comparison, and the number of bins is set to 8 to ensure a sufficient number of games per bin. The outcomes are identical, aligning with expectations that the number

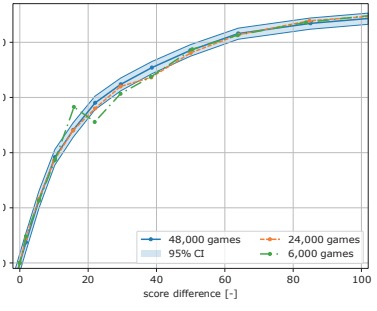


**Figure 3 Influence of dataset size on win rate using12 bins (move 150, 1d fox player)**

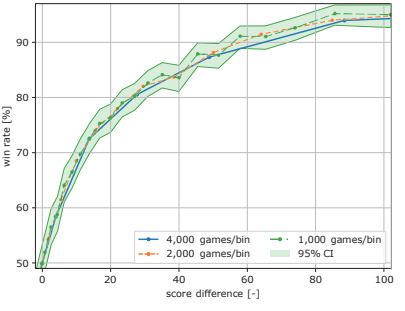


**Figure 4 Influence of games per bin on win rate using24,000 games (move 150, 1d fox player)**

of visits solely influences the variance of score predictions, without introducing bias. The results presented further in this study are obtained using 5 visits.

**Impact of time settings** Most of the collected games use a main time of 5 minutes. Nevertheless, for certain player strengths, insufficient data is available at this time setting. For 8dan and 9dan players, more data was accessible from shorter games with a main time of 1 minute. For 12kyu players, most games are played with longer time settings, including a main time of 20 minutes.

To assess the impact of time settings on the win rate, the win rate is plotted in Figure 6 for the threedistinct settings, considering a player strength of 1dan. A total of 50,000 games are collected for each time setting, ensuring a low level of uncertainty. It can be observed that the three curves are in strong agreement, signifying no influence of time settings on the win rate.

# Ⅲ. Results

## 1. AI Win Rate or Score?

To evaluate a board position, Katago uses a neural network known as the value network. This network generates various scalar values, two of which are relevant for game analysis: win rate and score. From a given position, the AI win rate can be approximated as the probability that the AI will win the game when playing against itself.

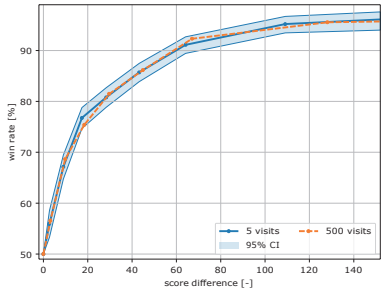


**Figure 5 Impact of number of visits on win rate (move 150, 1d fox player)**

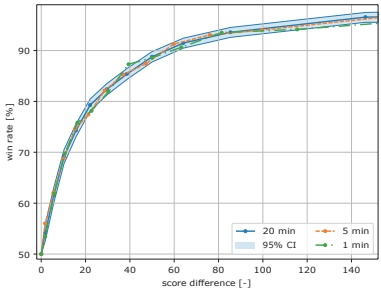


**Figure 6 Impact of main time on win rate (move 150, 1d fox player)**

On the other hand, the score is an estimation of the score at the end of the game. AlphaGo, Leela Zero and Katago utilise the AI win rate to train their neural networks. SAI used a two scalars output for its value network, and could be trained to maximise the score difference, but it was only assessed on small boards (Morandin et al., 2019). It is generally accepted that the win rate is a more reliable metric to train such neural networks than the score.

By considering all the analysed games, the AI win rate is plotted against score in Figure 7. It is apparent that the relationship between AI win rate and score is nonlinear. As anticipated, a 1point difference holds significant impact on the game's outcome when the score difference is near 0, but it has a minor effect on the win rate if the score difference is already substantial (e.g., 30).

Furthermore, the relationship between AI win rate and score is influenced by the move number. A 5point lead corresponds to an AI win rate of 85% at move 50 and 95% at move 200. This aligns with expectations, as overturning the game is easier during the opening and middle game phases,

where numerous possible moves exist, compared to the endgame, where the range of viable moves is more limited.

## 2. Human Win Rate

As stated in the preceding section, the AI win rate is an estimate of the winning probability when the AI competes against itself. Similarly, the human win rate is defined as the probability of winning when playing against oneself or against an opponent of equivalent strength. Since AI strength greatly surpasses that of top professional players, we anticipate that the human win rate will significantly differ from the AI win rate, particularly for amateur Go players.

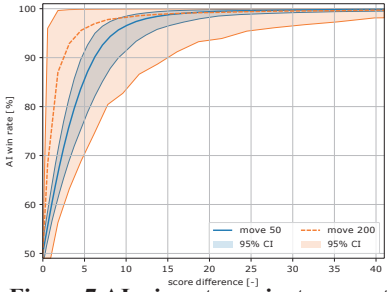


**Figure 7 AI win rate against score at different move number (50 ≈ end of the opening, 200 ≈ endgame)**

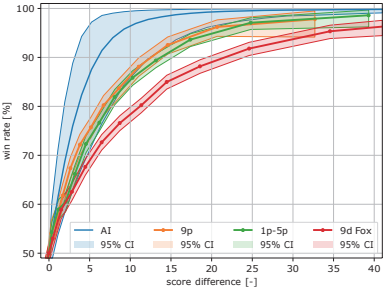


**Figure 8 Comparison of AI and human win rate (9p, 1p5p are from professional games, 9d Fox are from online games, win rate is shown at move 100)**

The comparison between AI win rate and human win rate is presented in Figure 8. As expected, the AI win rate considerably exceeds the human win rate, even when examining 9dan professional players. The win rate for professional players ranging from 1p to 5p is very similar to that of 9p players,

althoughconsistently slightly lower. This suggests that our methodology and the number of analysed games suffice to capture the difference between 9p and 1p to 5p professional players, but this win rate distinction remains relatively small. The win rate is notably lower for 9d Fox players, whose strength is expected to be closeto that of professional players.

Several hypotheses can explain this disparity. The average skill level of 9dan Fox players might be notably lower than that of professional players. Time settings could potentially influence the win rate professional games span several hours, while the 9dan Fox games analysed here feature a mere oneminute main time. Moreover, it's plausible that players approach official professional games moreseriously compared to online games. Finally, players might adopt distinct playing styles during online games, exhibiting more aggressive or unconventional moves. Further studies would be necessary to assess these hypotheses.

## 3. Impact of Game Stage

Go games are typically divided into three stages: the opening (*fuseki*), middle game (*chuban*), and endgame. One of the authors of Li et al. (2019) analysed 500 Go games and extracted the move numbers at which the middle game and the endgame begin. Both distributions were found to be normal. Their results reveal that the middle game starts around move 49 ± 6, and the endgame at move 162 ± 19.

It is widely understood that as a game progresses toward its conclusion, it becomes easier to securevictory given a fixed score lead. For instance, with a 10point lead in score, the win rate is anticipated to

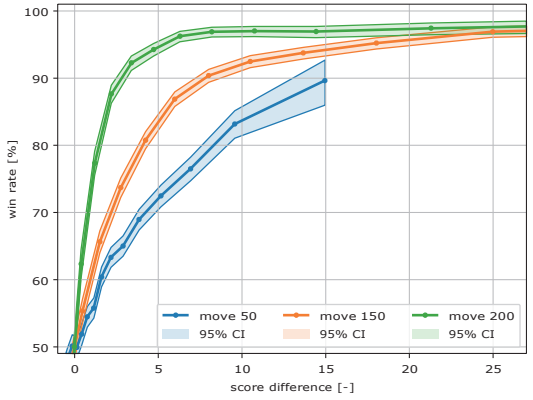


**Figure 9 Win rate against score at different stages of the game for professional players**

be higher during the endgame compared to the opening. To evaluate this effect, the win rate is plotted atmoves 50, 150 and 200 in Figure 9 for professional players (1p to 5p and 9p altogether). With a 10pointlead, the win rate stands at 84% at the end of the opening (move 50), rises to 92% at the end of the middlegame (move 150) and reaches 97% during the endgame. As a consequence, the shape of the win rate curves evolves with move number. It exhibits nearly linear behavior at move 50, becoming steeper as themove number rises, and culminating in an almost square step function by move 200.

## 4. One Fit to Rule Them All

In order to make win rates accessible to a wide audience of players, one would ideally need a simple formula. Win rate curves exhibit a distinctive sigmoid shape that can be characterised by the two parameters algebraic function:

$$f(x) = \frac{1}{2} * \frac{\gamma x}{(1+|\gamma x|^k)^{1/k}} + \frac{1}{2} \quad (2)$$

where x is the score, f (x) the win rate, and γ and k are real parameters. The parameter k governs the slope of the sigmoid function, enabling the modeling of both steep sigmoids (for higher move numbers)and quasilinear sigmoids (for lower move numbers). The value of k is displayed against move numberin Figure 10 for various player strengths. A consistent decreasing trend is observed across all player strengths, indicating that k is mostly influenced by move number. A linear regression using the methodof least squares on the range $n_{move} \in [50; 200]$ yields the following formula for k:

$$k(n_{move}) = 1.99 - 0.00557 * n_{move} \quad (3)$$

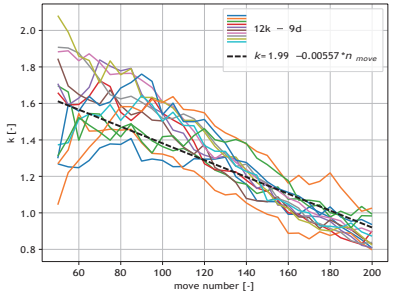


**Figure 10 Value of parameter k againt move number for different player strengths**

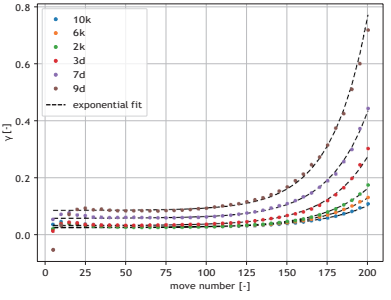


**Figure 11 Value of parameter γ againt move number for different player strengths**

The value of γ is plotted against move number for different player strengths in Figure 11. It can be observed that γ is dependent on both the number of moves and player strength. Each player strength curve can be ad-

equately represented by the exponential fit:

$$\gamma\,(n_{move}, L) = 0.0001 * e^{w(L)*n_{move}} + \delta\,(L) \tag{4}$$

were $w\,(L)$ and $\delta\,(L)$ are real parameters depending only on player strength. Applying least square minimization within the range $L \in [-11; 9]$ every 2 levels (with 11 corresponding to 12kyu and 9 to 9dan) yields the following coefficients:

$$w\,(L) = 0.0375 + 0.000543 * L \tag{5}$$

$$\delta\,(L) = 0.00292 * e^{0.354*L} + 0.025 \tag{6}$$

The general formula is derived by substituting the fitted parameters from Equations 3 and 4 into Equation 2. The resulting win rate is compared to the data in Figure 12 for kyu players and Figure 13 for dan players. Four game stages were selected: moves 50, 100, 150, and 200. The comparison reveals a strong agreement between data and the formula, with an average absolute error of 1.2%. This suggests that the formula serves as a solid interpolation for win rates across player strengths ranging from 12kyu to 9dan and move numbers between 50 and 200. However, the validity of the formula outside these bounds has not been assessed.

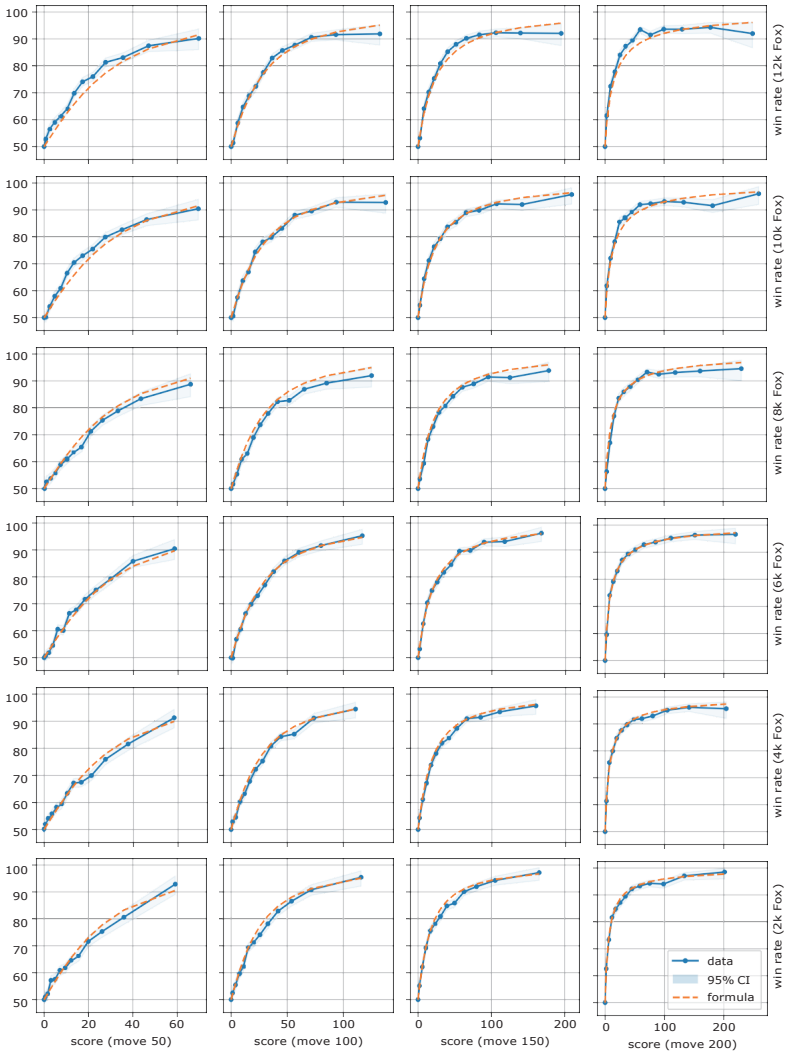


**Figure 12 Win rate against score with respect to move number and player strength (kyu level)**

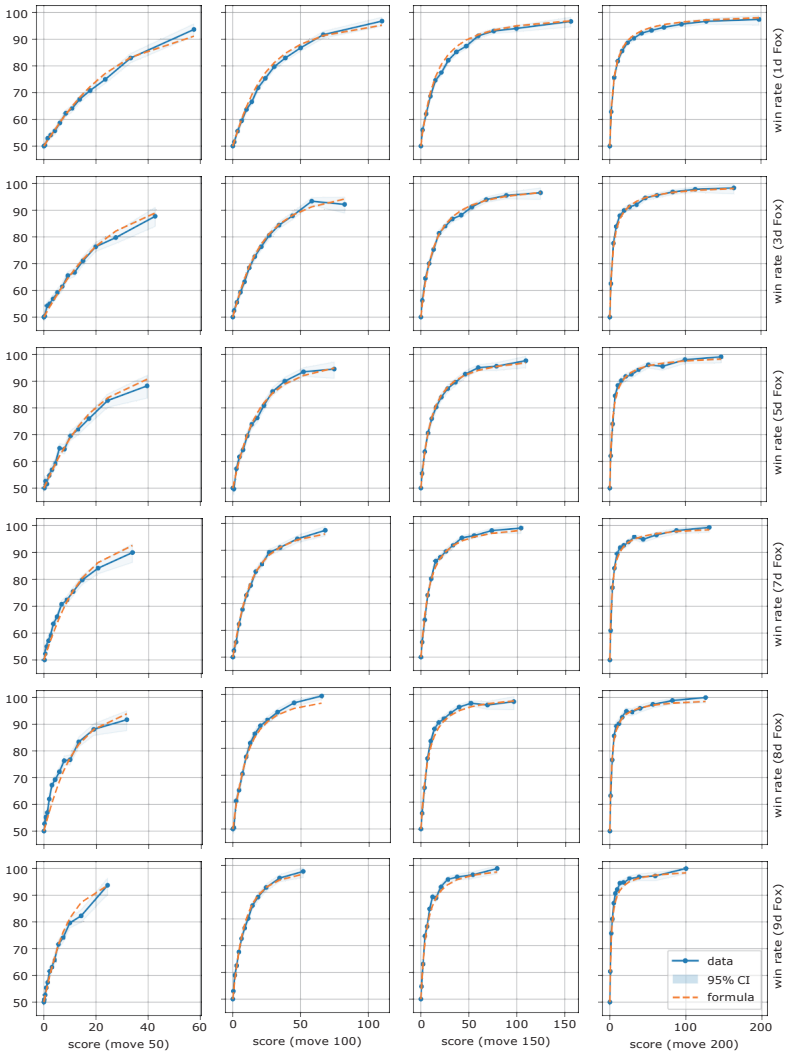


**Figure 13 Win rate against score with respect to move number and player strength (dan level)**

# Ⅳ. Discussion

As pointed out by EgriNagy and Törmänen (2020), AI win rates can be misleading in certain situa tions. For instance, in a handicap game, Black might initially hold a score lead yet maintain a win rate of 50% if the handicap is wellchosen. Furthermore, the AI win rate is only meaningful for players of equiv alent strength. Following the methodology developped within this study, games involving handicaps or participants of differing strengths could be analysed. Such an approach would yield a novel metric that factors in the skilllevel disparity between players.

Even in balanced games, the AI win rate can lead to wrong conclusions, particularly for amateur players. A 5point lead at move 200 yields a 97% AI win rate, implying that the game is already decided according to Go engines. For a 5dan Fox player, the win rate is about 75%, suggesting a significant chance of turning the tide in the endgame. For a 8kyu Fox player, the win rate drops to 60%, indicating a more evenly matched game.

The strategic choices made during a game of Go hinge on the game's stage and score evaluation. When slightly behind, players should choose optimal moves and exploit every chance to secure a few points. Conversely, if a player is significantly losing, they might initiate difficult fights to create com plex scenarios where the odds could shift. Such strategies often involve suboptimal moves (colloquially termed trick plays) which might incur point losses on average but harbor a small possibility of overturning the game. The ability to estimate one's win rate is thus of crucial importance.

The human win rate derived in this study can be read directly on Figures 12 and 13. It can also be computed using the formula given by Equation 2. A

small sample of human win rates associated to a score lead of 10, 20 and 30 points, are listed in Table 2 for illustration. They span a large range of player strengths and various stages of the game (moves 50, 150 and 200).

| Rank (Fox) | win rate at move 50 | win rate at move 150 | win rate at move 200 |
|---|---|---|---|
| 12k | 62 / 72 / 79% | 64 / 73 / 79% | 70 / 78 / 82% |
| 8k | 62 / 72 / 79% | 66 / 75 / 80% | 74 / 81 / 85% |
| 4k | 63 / 72 / 79% | 68 / 77 / 82% | 78 / 85 / 88% |
| 2k | 63 / 73 / 80% | 69 / 78 / 83% | 80 / 87 / 90% |
| 1d | 64 / 74 / 81% | 70 / 80 / 85% | 82 / 88 / 91% |
| 3d | 66 / 77 / 84% | 72 / 82 / 87% | 84 / 90 / 92% |
| 5d | 69 / 80 / 87% | 75 / 84 / 89% | 86 / 91 / 94% |
| 7d | 74 / 86 / 91% | 78 / 87 / 91% | 88 / 93 / 95% |
| 8d | 77 / 89 / 93% | 80 / 89 / 92% | 89 / 93 / 95% |
| 9d | 82 / 92 / 95% | 83 / 90 / 93% | 90 / 94 / 96% |

**Table 2 Human win rates for 10 / 20 / 30point lead for different player strengths at various stages of the game**

Human win rates not only aid players in making informed decisions during gameplay but also in post game analysis for error review. Tools like AI Sensei (Teuber et al., 2023) already categorise moves as 'Good,' 'Inaccuracy,' 'Mistake,' or 'Blunder,' based on point drop and player strength. By incorporating the developed formula, the associated drop in human win rate could be provided to complete the analysis. The point drop offers retrospective insight into a move's absolute value, whereas the win rate illustrates its impact on the game's outcome. These metrics are synergistic and could be employed together to enhance the learning of Go players.

# V. Conclusions

This study involves the analysis of a substantial collection of online games played on the Fox Go server, as well as professional games, using the Go engine Katago. The resulting database of analysed games has been made available online under an opensource license.

Within this study, a novel methodology has been developed for computing win rates using analysed Go games. The findings reveal a consistent trend: human win rates are notably lower than AI win rates, which applies to both professional and amateur players. This suggests that one should not rely blindly on AI win rates for game analysis.

A general formula has been derived to predict win rate curves at specific move numbers and player strength. The formula's accuracy has been validated across move numbers ranging from 50 to 200, as well as player strengths from 12kyu to 9dan. This innovative metric can serve to assess the importance of mistakes during game analysis, depending on the game stage and player strength. Furthermore, it can provide guidance for estimating one's probability of winning during a game of Go, leading to improved strategic choices.

# References


Baker, L. and Hui, F. (2017), 'Innovations of alphago'.
URL: *https://github.com/featurecat/godataset*

Bouzy, B. and Cazenave, T. (2001), 'Computer go: an ai oriented survey', *Artificial Intelligence* 132(1), 39–103.

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

EgriNagy, A. and Törmänen, A. (2020), Derived metrics for the game of go–intrinsic network strength assessment and cheatdetection, *in* '2020 Eighth International Symposium on Computing and Net working (CANDAR)', IEEE, pp. 9–18.

Featurecat (2019), 'Go dataset'.
URL: *https://github.com/featurecat/godataset*

Kasparov, G. and Friedel, F. (2017), 'Reconstructing turing's "paper machine"', *EasyChair Preprint* 3.

Li, X., Lv, Z., Wang, S., Wei, Z., Zhang, X. and Wu, L. (2019), 'A middle game search algorithm appli cable to lowcost personal computer for go', *IEEE Access* 7, 121719–121727.

Morandin, F., Amato, G., Gini, R., Metta, C., Parton, M. and Pascutto, G.C. (2019), Sai a sensible artifi cial intelligence that plays go, *in* '2019 International Joint Conference on Neural Networks (IJCNN)', pp. 1–8.

Müller, M. (2002), 'Position evaluation in computer go', *ICGA Journal* 25(4), 219–228.

Pascutto, G.C. (2017), 'Leela zero'.
URL: *https://github.com/leelazero/leelazero*

Rendu, Q. (2023), 'Analysed kifu database'.

URL: *https://gitlab.com/qrendu/analysedkifudatabase*

Sadler, M. and Regan, N. (2019), 'Game changer', *AlphaZero's Groundbreaking Chess Strategies and the Promise of AI. The Netherlands. New in Chess* .

Shin, M., Kim, J. and Kim, M. (2021), Human learning from artificial intelligence: evidence from human go players' decisions after alphago, *in* 'Proceedings of the Annual Meeting of the Cognitive Science Society', Vol. 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., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Ku maran, D., Graepel, T. et al. (2018), 'A general reinforcement learning algorithm that masters chess, shogi, and go through selfplay', *Science* 362(6419), 1140–1144.

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.

Takeuchi, S., Kaneko, T., Yamaguchi, K. and Kawai, S. (2007), Visualization and adjustment of evalu ation functions based on evaluation values and win probability, in 'Proceedings of the national con ference on Artificial Intelligence', Vol. 22, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, p. 858.

Tesauro, G. et al. (1995), 'Temporal difference learning and tdgammon', *Communications of the ACM* 38(3), 58–68.

Teuber, B., Ouchterlony, E. and Dohme, M. (2023), 'Ai sensei'.

URL: *https://aisensei.com/*

Wu, D. J. (2019), 'Accelerating selfplay learning in go', *arXiv preprint arXiv:1902.10565* .

Received: 14, Sep, 2023

Accepted: 30, Oct, 2023