International Journal of Networking and Computing – www.ijnc.org, ISSN 2185-2847
Volume 13, Number 2, pages 258–272, July 2023

The cost of passing – using deep learning AIs to expand our understanding of the ancient game of Go[1]

Attila Egri-Nagy
Akita International University
Department of Mathematics and Natural Sciences
Yuwa, Akita-City 010-1292, Japan
Email: egri-nagy@aiu.ac.jp

Antti Törmänen
Nihon Ki-in – Japan Go Association
7-2 Gobancho, Chiyoda City,
Tokyo 102-0076
Email: tormanen.antti@gmail.com

Received: February 14, 2023
Accepted: May 12, 2023
Communicated by Hiroyuki Sato

Abstract

In addition to their playing skills, AI engines utilizing deep learning neural networks provide excellent tools for analyzing traditional board games if we define new measures based on their raw output. For the ancient game of Go, we develop a numerical tool for context-sensitive move-by-move performance evaluation and for automating the recognition of game features. We measure the urgency of a move by the cost of passing, which is the score value difference between the current configuration of stones and after a hypothetical pass in the same board position. In this paper, we investigate the properties of this measure and describe some applications for developing learning tools and analyzing a large number of games. As we use AI tools to gain new insights into the ancient game of Go and develop a more precise, quantified understanding, this work fits into the more significant and general project of utilizing superhuman AI engines for deepening human understanding and growing human knowledge.

1 Introduction

Board games have been the focus of artificial intelligence research since the beginning of the field as convenient testbed applications. They are complex enough to be challenging, but with their well-defined rules, they are relatively simple, closed worlds. This tradition continues in the current phase of artificial intelligence, where we are more interested in human-centered applications than in developing ever more powerful AI software.

Our primary interest is human learning. Thus we focus on deepening our understanding of board games using the newly developed AI software tools. We specialize in the game of Go due to

[1] This paper is an improved and extended version of the work presented at CANDAR 2022, The Tenth International Symposium on Computing and Networking, published as 10.1109/CANDAR57322.2022.00019.

its complexity and popularity. Here we briefly describe the rules of Go, distilled from the description in [1], the motivations leading to this research, and the idea of the cost of passing.

1.1 The Game of Go

Go is a two-person perfect-information game with no random factor. It is played on a grid with black and white stones, with the players taking turns to place one stone at a time on an empty intersection of the grid. The goal in the game is to conquer a larger part of the game board than the opponent by surrounding it with one's stones. Stones fully surrounded (that have no empty neighboring intersections) are captured and removed from the board. When capturing is possible by one move, we say that the group in danger is in *atari*.

The same whole-board position cannot be repeated (*ko* rule), so games are guaranteed to finish in finite time. Although the stones do not move, the game is very dynamic: any played stone can tip the balance of the forces on the board. Having the initiative is called *sente*, while the opposite, i.e., mere reacting to opponent moves is called *gote*.

Stones of the same color are solidly connected if they are vertically or horizontally adjacent, and loosely connected if they are not solidly connected but nonetheless cannot be cut apart by the opponent. Connected stones of the same color form *groups*, which are crucial to the game's strategy. Important fights are often played out between groups of the two players, and capturing a key group from the opponent often results in winning the game.

The winner can be determined by either counting the number of stones, or the sum of a player's stones and the empty intersections ('territory') they surround, or the sum of the stones a player has captured and the territories their stones surround. While these scoring methods are considerably different, they describe the same game and (with a few exceptions) lead to the same result. This paper uses the third method, which is commonly referred to as Japanese scoring.

The *komi* is a compensation given to White for not making the first move. The komi shows that ultimately we want to define Go as a game in which perfect play gives a draw. In practice, komi is usually not set to an integer value to prevent tied games: this is more about game management and tournament organization than the nature of the game.

For players of different skill levels, using *handicap* stones can balance the winning chances. These are free moves for the black player at the beginning of the game to provide tentative territorial frameworks.

1.2 Motivation

AlphaGo [2] made history by being the first Go program capable of winning against a top human professional player. The event also changed the goals of artificial intelligence research projects. Producing ever stronger and more general engines is still a worthy pursuit [3–5] since the game is not solved yet [1]. Our exact knowledge about the best possible lines of play is limited to 30 intersections (e.g., a 6×5 board) [6]. Current research in that direction works in the deep learning paradigm and tries to optimize the process by changing the neural network architecture, the activation function, or the training algorithm (see, e.g., [7]). However, there are now new possible applications. Superhuman Go engines can be used to deepen human understanding of the game [8, 9]. There is no evidence that the current AIs are close to perfect play. Thus, developing playing skills is still a practically open-ended activity. Our goal is to facilitate the human learning process, and we develop tools for context-sensitive evaluation of the value of moves and automated feature detection of games.

1.3 The general idea of characterizing game positions

Traditionally, we divide a game of Go into three main parts: the opening, the middle game, and the endgame. These all have different characteristics, and we apply different types of thinking and skills in each stage. Of course, these distinctions may not be apparent in a particular game. For instance, the stages can overlap. A middle-game fight can erupt in a corner and reach its conclusion, while the other corners remain in the opening stage. Still, the stages provide a natural framework for understanding games.

Here we augment the traditional division of the stages of the game with a quantitative and fine-grained measure. We observe that different stages of a game have different costs for mistakes. For example, multiple inaccuracies in the opening can be compensated by the opponent's single larger mistake during a middle-game fight. It seems that we can characterize board positions by the cost of a mistake. However, a mistake is context-dependent, and so the question arises: how can we define a bad move precisely and independently from the situation? In Go, every move is an investment towards some end. In the vast majority of game situations, playing any stone is better than not, and therefore, *passing a move can represent a maximally bad move*. Thus, the actual value of the point loss caused by a passed move can be used to characterize a given board position. Then, this value can be compared with the point loss or gain of the actual game move to get a more nuanced evaluation. It is important to note that the cost of passing is not a way to evaluate the player's moves directly. The loss induced by a hypothetical pass provides the background information required to judge the game's values more precisely.

Passing during the game is not universally bad: for example, when a game is over, playing a stone inside one's own or the opponent's territory costs a point, while passing avoids losing a point. Also, in general, there can be bigger mistakes on the board (such as killing one's own group) than passing itself. However, for game analysis purposes, the cost of passing is a natural and practical tool to assess performance move by move.

1.4 The structure of the paper

In Section 2, we give a definition for the cost of passing, then in Section 3 we describe the core idea for implementing its calculation using the existing AI engines. In Section 4, we give a general description of how the cost of passing changes in different stages and in different situations of a game. In Section 5, we discuss the concept of sente (having the initiative) in terms of the cost of passing. We briefly mention a few possible applications of this measure in Section 6 and describe our research methods including software tools in Section 7. Finally, we conclude with the plan for future research based on the cost of passing.

2 Defining the cost of passing

We denote the board position after i moves by s_i, and s_0 refers to the empty board. We use function notation for moves: $a_{i+1}(s_i) = s_{i+1}$, i.e. the $i+1$th move produces the $i+1$th board position. In particular, $a_1(s_0) = s_1$, the first move produces the second board position, indexed by 1, as for stone configurations on the board, the indices always refer to the number of moves made.

For a given board position, deep learning Go-playing AIs can provide information about the probabilities for winning, the *win rate*, $V(s)$, and the final result score estimate, the score lead, or *score mean* m_s. As a convention, we always compute the score mean from the perspective of the black player. Thus, a positive score mean predicts a black win, while negative values show white's advantage.

The win rate is a fundamental internal parameter of the deep learning Go AIs. It is calculated as the result of the tree search, which relies on the neural network's evaluation [2]. The neural network is also trained for estimating the final score in more advanced architectures [5]. These are statistical measures, and due to the probabilistic nature of the tree search and the neural networks, they are subject to random fluctuations.

The effect of a move is the difference between the score lead estimate before and after the move [10]. This describes the efficiency of a move compared to the AI's best move candidate. However, board positions and their available moves can have very different characteristics: in some board positions there is only a single good move available, while in other positions dozens of moves can be similarly good. Furthermore, mistakes can be more costly in some board positions than others. The objective value of a move therefore depends on the game context, and it would be beneficial to have a way to measure the value of a move while taking the whole-board situation into account.

Definition (Informal)**.** The *cost of passing* in a board position for the player in turn is the difference between the score means before making a move and after passing.

In other words, the cost of passing is the price to pay for a missed move.

Definition. $c_i = \mu_{s_i} - \mu_{\text{pass}(s_i)}$

As we adopted the convention to report the score mean always from the perspective of the black player, we need to change the sign for the white player. This way, the cost of passing is in general a positive value. The main exceptions to this are when a game is finished and neutral points are being filled out, during when the cost of passing will be zero, and when a game is completely finished, during when passing in fact saves a point. It should be noted that in the former case, the probabilistic nature of the AI analysis might in fact also output minuscule negative values.

Technically, we could define the cost of passing in terms of the win rate. However, it is more informative to evaluate moves in terms of expected scores. It is a single bit of information to know who won the game, but a lot more informative to know by how much.

3 Implementation

For playing the game, the AI engines do not need to calculate the cost of passing. They evaluate board positions in terms of winning probabilities and estimated scores. These are already derived measures, as choosing a move is usually decided by the visit count of the corresponding board position. The engine chooses the move most often considered by the Monte-Carlo tree search. Therefore, we need to calculate the cost of passing externally.

The idea is simple: we carry out a hypothetical pass move. We gain useful information (about an actual board position) by thinking about an event (pass) that has not happened. This is a form of *counterfactual reasoning*, which is a crucial tool in causal inference [11]. Counterfactuals are considered to be important for building future AI systems, and here they are already important in current applications.

Let's say we want to find the cost of passing for black. We evaluate the board position and record the score mean value. Then, we leave the same configuration of the board but evaluate it as if it was white's turn. This direct manipulation of the turn may not work well with the engine, as they take into account the history leading to the board position (to avoid repeating an earlier board configuration). Thus we need to add a pass to the sequence of moves explicitly. The cost of passing is the difference between the two score estimates. This simple trick works for all engines that can do score estimates.

The drawback of this method is that we need to double the number of evaluations. Game analyses are computation-intensive and consequently time-consuming. Therefore we have a trade-off: the cost of passing gives more information but requires more resources. In practice, one needs to estimate the required strength of the analysis and set the number of visits per move accordingly.

4 The dynamics of cost of passing in top-level games

For a given game, we calculate the value of the cost of passing for each board position. We claim that this sequence of values contains descriptive information about the game. Here we describe how the values and their changes can characterize the different situations in the game.

4.1 Linear descent - the background dynamics

A game of Go finishes when neither player can change the size of territories on the board. Two consecutive passes end the game. Thus, the cost of passing is zero when a game is finished.

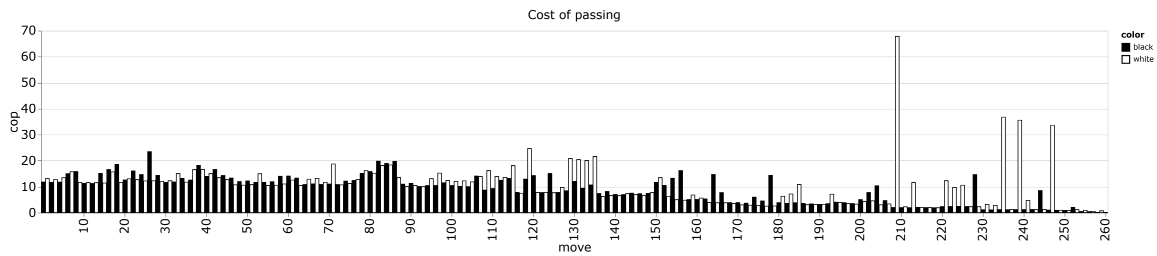


Figure 1: The cost of passing for a relatively peaceful professional game (see Game 1 in the Appendix for the game record). The linear descent can be clearly seen. There are still individual forcing moves (large spikes), one busy exchange for both players (at around 75–88), and one one-sided forcing sequence (130–136). The total cost of passing for the 261 moves is 2516, averaging to 9.64 per move.

Traditionally, the value of a single handicap stone has been estimated at roughly 10 points. Giving the black player an extra handicap stone is identical to the white player passing an extra time, so we can use the cost of passing to approximate numerical values for handicap stones. On an empty board, this value comes to roughly 12 points (see the methods section for the details of the engine and the network).

The game starts from an empty board and proceeds to a 'crystallized' state, the final configuration. On the one hand, in the early stages of a game there are more open possibilities than towards the end; and, on the other, because groups on the board need to secure two eyes in order to survive, the fewer there are stones on the board, the larger the effect that a single stone has on its surroundings. Therefore, in general, the cost of passing follows a linear descent. This theoretical line can be clearly visible from the cost of passing graphs resulting from AI analysis (see Fig. 1).

Parts of the game, however, deviate from the linear descent: these are when 'forcing moves' are played. A forcing move is a move that threatens to make a larger gain if the opponent does not respond. As far as the rules of Go are concerned, a player is always free to play on any open intersection on the board; but if a player wants to win the game, there are situations when it is strictly necessary to respond to the opponent's threat. For the side that plays the forcing move, it can be strategically beneficial to force the opponent to respond in a particular way. When the whole-board situation stabilises and no forcing moves are being played, the cost-of-passing graph returns to the linear descent, which is its fundamental shape.

All games of Go do not follow quite the same linear descent graph. This is because games have variable lengths, and the length of stages of games also differ. Some fighting-oriented games may only enter the endgame after move 200, in which case the cost of passing remains high for a long time. Some peaceful games may enter the endgame before move 100, after which the cost of passing may stay at below seven points. A player can resign any time in the game in case it becomes decidedly one-sided. Obviously, winning by resignation implies that at the end of game it is possible to have cost of passing values significantly different from zero.

4.2 Cost of passing as 'temperature'

The idea of a measure that is high in the beginning and drops to zero by the end of the game is not new. This is often called 'temperature,' roughly described as the urgency of making a move and often described as the value of the biggest possible move on the board. The term can be used with mathematical precision in combinatorial game theory [12,13] or in a more intuitive general sense. The cost of passing may simply seem like a way to calculate this temperature precisely, but there are some decisive differences. In Go, we often distinguish between the local and ambient temperature, while the cost of passing is calculated for the whole board. Due to the local-versus-ambient distinction, the temperature only goes down as the game progresses, while the cost of passing can also increase. Therefore, it is justified to keep the two concepts separate.

4.3 The stages of the game characterized numerically

Generally, the 'baseline' cost of passing in the early game ranges from about 13 to 10 points. In the middle game, values are generally between 12 and 7 points, and values below 7 points usually indicate the endgame. These values are estimates based on dozens of cost-of-passing analyses by the second author relying on his professional player experience and the work done in cheat detection [10]. In the future it may be useful to conduct a statistical research on the values by a bulk analysis of a large number of games.

4.4 The 'heated' parts: fighting sequences and threats

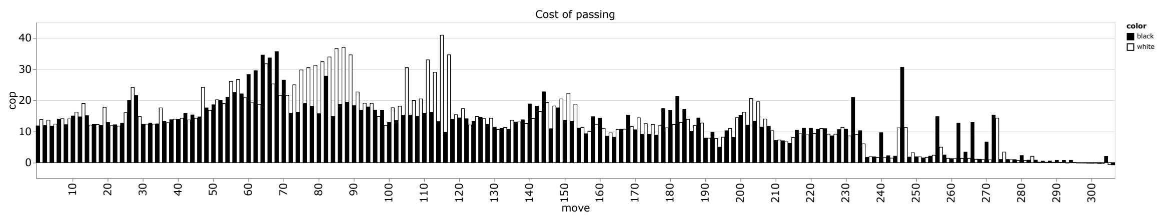


Figure 2: AI versus AI game, KataGo playing itself but with different networks (40-block versus 60-block). It is a fighting game indicated by the elevated level of cost of passing. The one-sided black peak (centered at move 65) and two subsequent white peaks (centered at moves 80 and 105) show first white and then the black player dominating the fighting (see Game 2 in the Appendix for the game record). The total cost of passing for the 307 moves is 3937.93, averaging to 12.83 per move.

The cost of passing can detect different game situations based on the deviations from the baseline linear descent. Fighting sequences are characterized by values that are high for both players. Threats and forcing moves can be recognized when the elevation is one-sided, where it is high only for the defending player. See Fig. 2 for a fighting game exhibiting continuously elevated levels of cost of passing.

4.5 Efficiency

What can be lost by missing a move is what can be ensured by optimal play. Therefore, the cost of passing in a way defines the maximum possible effect of a move. Consequently, we can evaluate the achievement of a move by stacking it against the cost of passing. This defines an efficiency measure for a move that can be expressed in percentages. An optimal move has 100% efficiency, a pass has 0%, and other moves can take any values between these limits. We can then use this efficiency to evaluate a player's performance compared to the optimal play or at least to the superhuman playing skills of current AI engines.

However, the efficiency measure cannot be used in all stages of a game. As discussed, there is a natural decreasing dynamic for the cost of passing, and in the endgame, it can become negligibly small. These small values, coupled with the stochastic nature of the game analysis, can produce meaningless efficiency values outside the 0-100% range.

As a practical solution, one can restrict the maximum and minimum efficiency values since negative values can also appear due to random noise in the analysis in the endgame. The limits prevent scaling issues with diagrams; see Fig. 3.

5 On sente

Sente, or having the initiative, is a fundamental concept in the game. It can roughly be translated into having the freedom of deciding where to play. The opposite is gote, having no choice but to

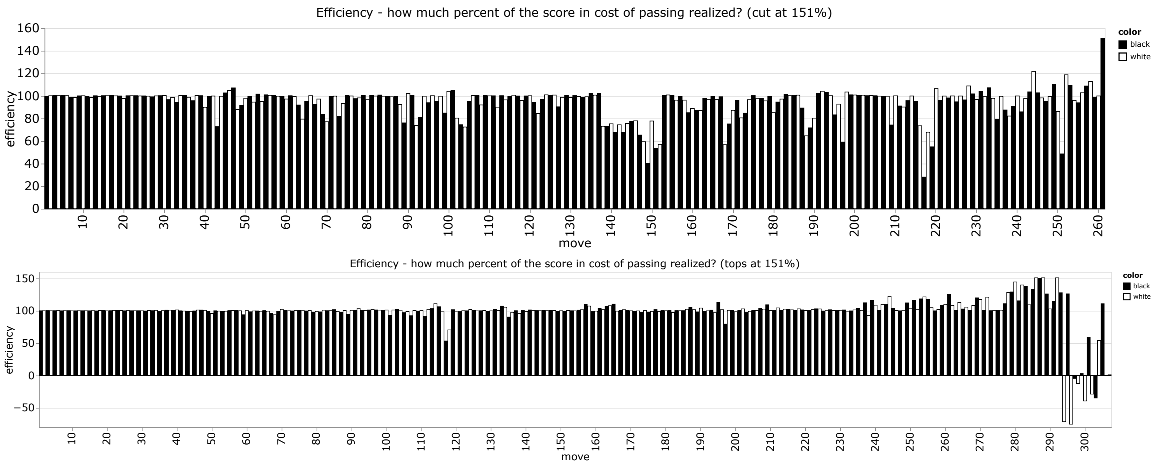


Figure 3: Efficiency of moves the human played Game 1 and AI self-play Game 2. The efficiency values are capped at 151% as the unstable values of the endgame could hide the valid signal in the preceding moves. The trouble caused by the small cost of passing values is especially bad at in the second game, showing even negative values. Note, that AI self-play's near perfect efficiency could be partially due to the fact that the analysis was done by the same engine.

make a forced move to avoid a larger loss, in effect merely reacting to the opponent's actions. Often but not always, the player with sente has the strategical upper hand in a game.

5.1 On detecting sente

The cost of passing makes it possible to define mathematically whether a player has sente or not. When the cost of passing is elevated beyond the baseline suggested by the linear descent graph, there is something 'urgent' on the board and a player is not strategically free to choose their action; in this case, the player has gote. If the cost of passing is instead at the baseline, then there are multiple similarly good moves available, and the player's action is not 'forced'; in this case, the player has sente. When a fight is finished, there is a drop from an elevated value to a baseline value of the cost of passing, and the player who has the baseline value has sente. See Fig. 4 for an example.

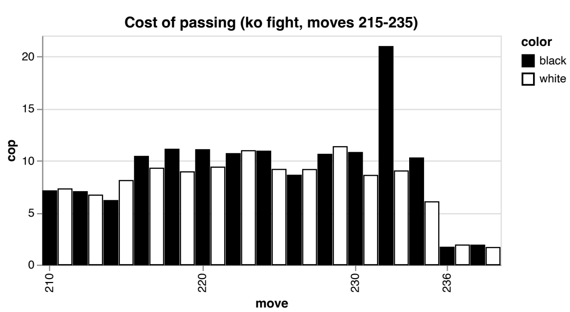


Figure 4: Black provokes a fight that develops into a ko fight, hence the elevated cost of passing for the sequence. White wins the ko, but the ko needs to be finished in gote, thus after 236 moves, Black has a clear sente move. (see Game 2 in the Appendix for the game record)

5.2 On the value of having sente

When professional players estimate the score in a game, they usually count the values of secure territories and then weigh unfinished, potential territories against each other. Finally, to finish this process, the player has to take into account whose turn it is, as having the move turn (i.e., sente) is also worth something.

The value of having sente is generally half of the cost of passing. This is mathematically self-evident: if a player passes, they give the sente to the opponent. The player first had whatever was the value of having the sente, and after the pass it is the opponent who gets the value, so the total difference is twice the value.

At the start of the game, the value of sente is equal to correct komi, or the white player's compensation for the fact that black goes first. Under the Japanese rules of Go, currently the komi is 6.5 points.

For example, a professional player might estimate that, in a given game position, black has 40 points and white 30 points of secure territory and the players' territorial potentials are roughly equal. If the game has just entered the endgame, the cost of passing might be about seven points. If it is white's turn, half of seven points and 6.5 from the komi would get added to his score, so the estimation suggests that the game is roughly even.

6 Applications

Here we sketch a few possible applications of the cost of passing. Our focus is on the human side, i.e., we would like to facilitate the learning process for human players.

6.1 Danger level indicator

Beginning players often have issues with recognizing immediate danger. This can easily lead to a lost game, even if the player would have known how to defend. As a direct solution, we can implement the advice directly in a teaching software application by highlighting the critical area and possibly the required defensive moves. Some Go clients have the option to warn about atari, as a basic and very special case of this idea. However, it could be more beneficial for the learner to merely indicate the presence of the danger, but not the location, and let her find out what to do.

The cost of passing is suitable for implementing such a danger level indicator. The higher the price for passing a move, the higher the level of danger. Seeing the warning, the player still has to scan the board for the threat, which can lead to good playing habits.

6.2 Visual identification and automated selection of points of interests

One way to improve board game skills is to study as many game records as possible. When working with a large collection of game records we may want to find examples of certain situations like fighting sequences, or threats and defences. Possibly the biggest advantage of deep learning AIs over humans is the speed that they can learn from previous games. Therefore, automating the task of finding the points of interest in a game record can improve the human learning process.

One obvious way to do this is to have an expert player going through game records. This time-consuming process can be made more efficient by automatically picking the points of interest. However, recognizing the interesting events is difficult by using the existing measures. Win rate can detect decisive blunders in balanced games, but once a game is one-sided, further blunders go unnoticed. Using score mean is better and with the derived measure of effect we can have a move-by-move analysis of performance. Still, the effect does not give information about the context of a move. The cost of passing is designed to give the contextual information. As we saw, it can recognize features of the game. Diagrams like Fig. 1 and 2 can act as a fingerprint for quick visual identification of game features. Alternatively, the interesting game events can be automatically discovered. Therefore, adding the cost of passing to the analysis measures can improve cataloging and data mining on game record databases.

In Fig. 5 we show this application in action. An important question about a game is whether it reached the scoring phase or one of the players resigned the game before the endgame concluded. This can be guessed from the game's length alone, but the cost of passing diagram gives more information. Three games from 1927, 2009, and 2011 reached their natural conclusions in scoring

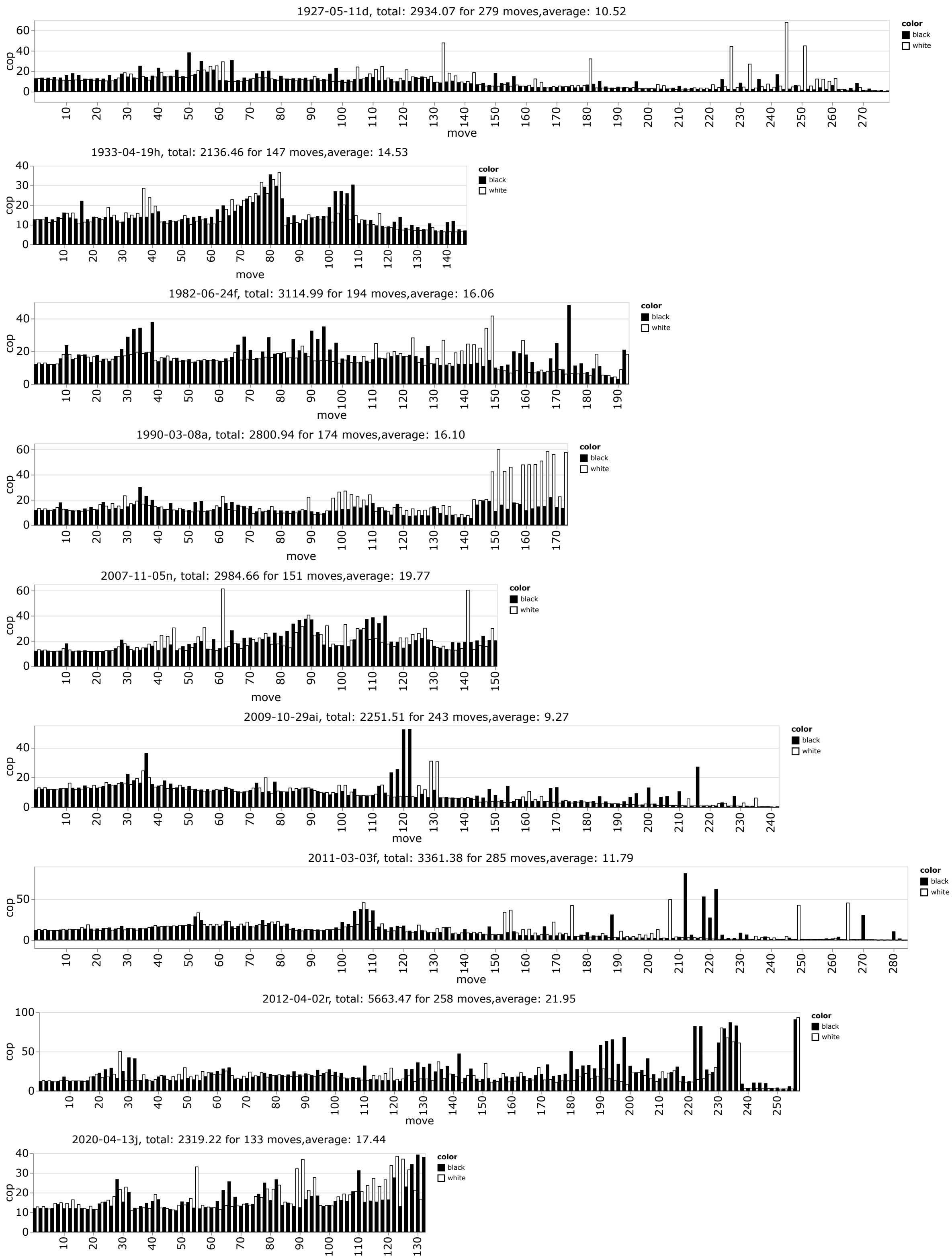


Figure 5: Fingerprinting by cost of passing diagrams of 9 randomly chosen games from a large historical collection (the July 2022 GoGoD database).

territories, as seen from the linear descent. However, the game from 2012 is long enough to reach scoring, but the final two bars with unusually high values indicate a more abrupt ending. Another interesting point is the curious 'growing stake' in the game from 1933, between moves 70 and 85. One would expect a fierce fight during these moves. Indeed, it is a situation where a white group gets surrounded and tries to break out. From the drop in the cost of passing, we can see that white gets sente, so just by looking at the diagram, we can conjecture the favorable outcome for white, which happens to be correct. These examples demonstrate the easiness of picking the points of interest by a quick visual check of the diagrams.

6.3 Game stage indicator

As mentioned in 1.3, a game of Go is traditionally divided into three main parts: the opening, the middle game, and the endgame. This division is made to reflect the differences in efficient tactics and strategies as a game progresses. It is said that, in the opening, one should aim to spread one's stones efficiently around the board and build flexible shapes and influence; in the middle game, one should focus on reinforcing one's weak groups and attack the opponent's; and, in the endgame, one should optimise their territorial profit. If a player could know which of these stages a game is in, this would provide useful information regarding what kinds of moves they should be considering. Cost of passing values could be used to create a tool to give the player such information.

Traditionally, the middle game is considered to start when a game-altering fight between the players' groups erupts; or, if there is no fighting, when the corners and sides of the board have been claimed and play moves into the centre of the board. The former would be reflected by temporarily elevated cost of passing values due to the new urgency of the board position, while in the latter case the cost of passing values would have dropped below a certain threshold to reflect the lower value of centre-oriented moves. Most games involve some level of middle-game fighting, so the former case is by far more common.

The endgame is considered to start when all of the groups on the board have been settled (i.e., are clearly either alive or dead) and there is no room for new groups on the board. After this point, moves played only have territorial value. Even after this point, forcing moves may be played, so there can be 'spikes' in the cost of passing graph reflecting the opponent's need to respond; but because purely territorial moves are generally smaller than moves that affect the well-being of groups on the board, in general endgame moves have smaller costs of passing than middle-game moves.

In the games in Figure 5, human players would probably more or less agree on this division of game stages:

1. 1927-05-11d: middle game starts at move 27 (12.2, immediate elevation); endgame starts at move 140 (7.9)
2. 1933-04-19h: middle game starts at move 47 (12.2, reduced values); endgame starts at move 119 (9.2)
3. 1982-06-24f: middle game starts at move 50 (14.1, not obvious from graph); endgame starts at move 151 (9.8)
4. 1990-03-08a: middle game starts at move 28 (15.2, immediate elevation); endgame starts at move 115 (10.5)
5. 2007-11-05n: middle game starts at move 32 (13.2, some elevation); no endgame (values stay high)
6. 2009-10-29ai: middle game starts at move 63 (13.5, not obvious from graph); endgame starts at move 103 (8.0)
7. 2011-03-03f: middle game starts at move 34 (12.2, immediate elevation); endgame starts at move 142 (8.1)

8. 2012-04-02r: middle game starts at move 16 (13.0, immediate elevation); endgame starts at move 239 (8.8)

9. 2020-04-13j: middle game starts at move 29 (15.2, not obvious from graph); no endgame (values stay high)

In games 1, 4, 5, 7, and 8, there is elevation in the cost of passing value following the middle-game threshold move. To eliminate false positives from forcing moves, the player could be informed 3-4 moves after the fact that middle game has started. Game 2 instead follows the reduced-values pattern, although some players could argue that the middle game only started at move 63, after which there is also a clear elevation in the values. In games 3, 6, and 9, the cost of passing values do not follow the 'traditional' understanding of when the middle game started, and could lead to somewhat misleading information.

The endgame threshold values suggest that the endgame generally starts when the cost of passing is at around 8-9 points. Games 3 and 4 have somewhat higher values, but setting the value of endgame moves this high would lead to some middle-game positions getting mistakenly labelled as being in the endgame. The main challenge in identifying when the endgame starts is that games sometimes have late middle-game fights when the game seemed to be in the endgame. To reduce the number of mistaken early calls, the threshold value should be set somewhat lower than what the above values suggest.

As mentioned in 4.3, these values are only estimates based on the second author's professional player experience. A larger statistical research should be made to verify whether an automated model can accurately identify how the stages of a game are divided, and how well this matches the human consensus.

6.4 Game quality quantifier

Just the effect of moves (i.e., counting differences between chosen moves and the AI's top candidates) is not a sufficient indicator to calculate the quality of a game. This is because the nature of a game – how peaceful or fighting-oriented it is – directly affects the result. Fighting-oriented and therefore more complicated professional games can have average effects of -0.6 points a move or more, while peaceful, 'simpler' games by intermediate-level amateurs can have average effects of only -0.4 points a move. In reality, however, there will still be a huge level gap between the players of these games.

Instead of merely looking at the effects of moves, it can be more accurate to quantify how big of a percentage of a player's cost of passing throughout a game was 'realised' through their moves. If a player's cumulative cost of passing in a game was for example 1,000 points, and the realised value of the player's moves was 950 points, this gives a performance value of 95 percent. This method is strictly fairer than looking at the effects of moves, since the cumulative cost of passing is adjusted for the 'complexity' of the game.

More precise profiling of a player's performance is important in developing tools for cheat-detection. Measuring the cost of passing can help make the existing tools more efficient [10].

7 Methods

The development of the cost of passing measure was a natural continuation of our previous project for cheat detection [10]. We have three main software components in this project:

1. a deep learning Go engine,

2. data processing and

3. interactive visualization tools.

We used the KataGo engine for doing the game analysis. It is a leading open-source implementation of the self-play-trained Go engine model introduced by AlphaGo Zero [3]. It incorporates many improvements [5]. Establishing the strength of an engine precisely is difficult since it depends on many factors, e.g., the game settings and the underlying hardware. Rankings of Go engines are often heavily debated. Therefore our choice is based on secondary pieces of evidence. KataGo is used in many analysis tools and serves as a base for several engines in computer Go competitions. We used version v1.11.0 with networks kata1-b40c256-s11101799168-d2715431527 and kata1-b60c320-s6321537280-d2951683615. These are available at `https://katagotraining.org/`. Similarly, the number of visits (corresponding to the length and strength of the AI analysis) was set to 12000. At that time, this was the strongest available AI analysis on the popular online Go server OGS (`www.online-go.com`).

We used our purpose-built LambdaGo software package written in Clojure [14] to process the output of the analysis engine (available at `https://github.com/egri-nagy/lambdago`). This data-oriented general purpose language is well-suited for exploratory analysis of the AI engine output and preparing visualizations of the data.

For reporting the analysis results we use a browser window to render the diagrams. The graphs in this paper are taken from those reports. For rendering we used the Vega and Vega-Lite visualization grammars [15].

8 Conclusion

Here we made the next step in utilizing Go-playing AI engines to improve human understanding of the game. With the development of the cost of passing measure, we could provide a graph of a complete game that serves as a visual summary of the events. It indicates the different stages of the game and the different types of plays (e.g., critical moves, threats, and defenses) that cannot be picked up by the win rate and the score mean graphs.

We expect that the existing analysis software tools will adopt the cost of passing measure to provide more information for human learners. Also, as a next step, we plan a more thorough bulk analysis of historical and current games, similar to the analysis [16] done by ELF OpenGo [17]. Beyond utilizing the now available komi setting ability of analysis engines, we will use the cost of passing fingerprinting to make the analysis results more accessible and accurate.

The work presented here is a necessary technical step in using Go AIs to deepen the human understanding of the game. Developing measures and tools based on AIs' raw output could also serve as a general model for human-centered applications in other domains.

Acknowledgements

This project was funded by the Kakenhi grant 22K00015, titled 'On progressing human understanding in the shadow of superhuman deep learning artificial intelligence entities' (Grant-in-Aid for Scientific Research type C, `https://kaken.nii.ac.jp/grant/KAKENHI-PROJECT-22K00015/`). We also thank the six anonymous reviewers from the CANDAR 2022 conference. Their useful comments helped us to express our ideas more efficiently.

References

[1] A. Egri-Nagy and A. Törmänen, "The game is not over yet—go in the post-alphago era," *Philosophies*, vol. 5, no. 4, pp. 37–0, 2020. [Online]. Available: https://www.mdpi.com/2409-9287/5/4/37

[2] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, January 2016.

[3] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, pp. 354–359, October 2017. [Online]. Available: http://dx.doi.org/10.1038/nature24270

[4] 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*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[5] D. J. Wu, "Accelerating self-play learning in Go," https://arxiv.org/abs/1902.10565, 2019.

[6] E. Werf and M. Winands, "Solving Go for rectangular boards," *ICGA Journal*, vol. 32, pp. 77–88, June 2009.

[7] T. Cazenave, J. Sentuc, and M. Videau, "Cosine annealing, mixnet and swish activation for computer Go," in *Advances in Computer Games: 17th International Conference, ACG 2021*. Berlin, Heidelberg: Springer-Verlag, 2021, pp. 53–60.

[8] M. Shin, J. Kim, and M. Kim, "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, no. 43, 2021.

[9] A. Egri-Nagy and A. Törmänen, "Advancing human understanding with deep learning Go AI engines," *Proceedings*, vol. 81, no. 1, 2022. [Online]. Available: https://www.mdpi.com/2504-3900/81/1/22

[10] ——, "Derived metrics for the game of Go – intrinsic network strength assessment and cheat-detection," in *2020 Eighth International Symposium on Computing and Networking (CANDAR)*, 2020, pp. 9–18, 10.1109/CANDAR51075.2020.00010, preprint: https://arxiv.org/abs/2009.01606.

[11] J. Pearl and D. Mackenzie, *The Book of Why: The New Science of Cause and Effect.* Penguin Books, 2018.

[12] E. Berlekamp, J. Conway, and R. Guy, *Winning ways for your mathematical plays Volume 1.* A.K. Peters, 2001.

[13] E. Berlekamp and D. Wolfe, *Mathematical Go: Chilling Gets the Last Point.* CRC Press, 1994.

[14] R. Hickey, "A history of Clojure," *Proc. ACM Program. Lang.*, vol. 4, no. HOPL, Jun. 2020. [Online]. Available: https://doi.org/10.1145/3386321

[15] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-Lite: A Grammar of Interactive Graphics," *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2017.

[16] "Human professional game analysis," Meta AI website, accessed on July 9th, 2022. [Online]. Available: https://ai.facebook.com/tools/elf-opengo/

[17] Y. Tian, J. Ma, Q. Gong, S. Sengupta, Z. Chen, J. Pinkerton, and C. L. Zitnick, "ELF OpenGo: An analysis and open reimplementation of AlphaZero," 2019.

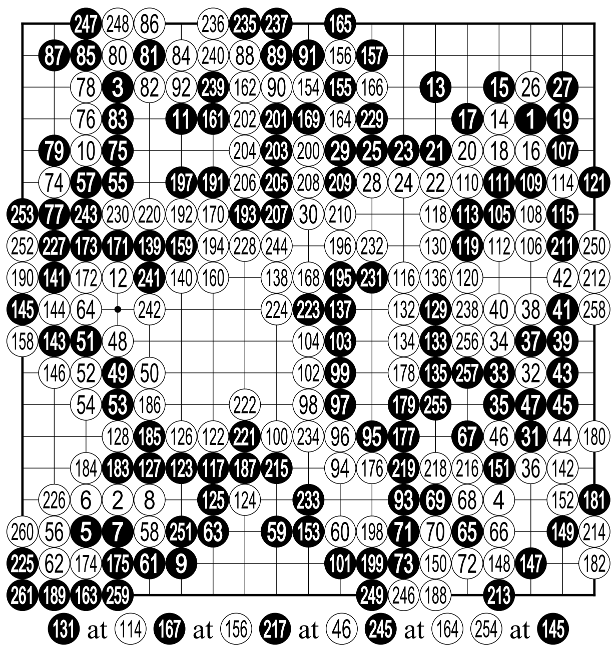
A Game Records

The following game records describe the games analyzed in the paper. Go game records (kifus) provide a visual picture, as the stones do not move. Therefore, it is enough to indicate the move number on the stone. However, we need to include additional information when several captures happen throughout the game: we say 'x at y' if move x is made at the intersection where move y was made previously. For convenience, we also include links to the game record files that any Go software tool can view.

Game 1

Antti Törmänen 1p versus Shuto Shun 8p, 2022.03.14, W+2.5

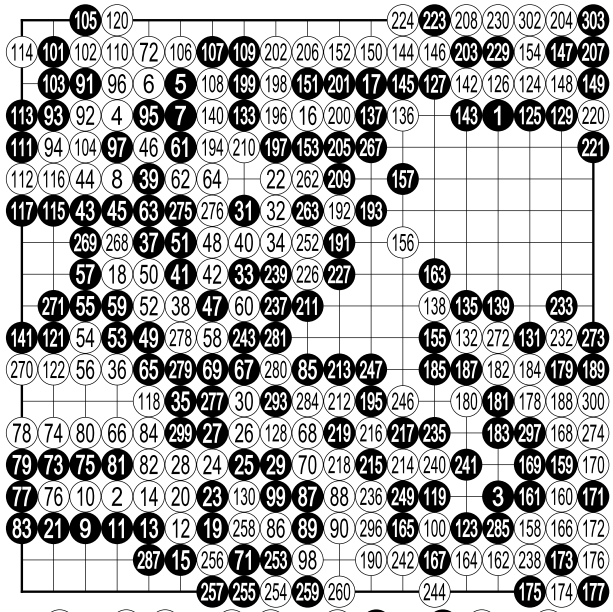
https://github.com/egri-nagy/lambdago/blob/master/resources/SGF/2022_cop/antti-shuto.sgf



Game 2

kata1-b40c256-s11101799168-d2715431527 versus kata1-b60c320-s6321537280-d2951683615 using KataGo v1.11.0

https://github.com/egri-nagy/lambdago/blob/master/resources/SGF/2022_cop/20220810_40bvs60b.sgf



134 at 46 186 at 174 222 at 216 225 at 219 228 at 216
231 at 219 234 at 216 245 at 47 248 at 60 250 at 47
251 at 215 261 at 214 264 at 256 265 at 19 266 at 192
282 at 100 283 at 219 286 at 216 288 at 254 289 at 263
290 at 259 291 at 192 292 at 167 294 at 219 295 at 280
298 at 171 301 at 220 304 at 23 305 at 256 306 307 pass