

# Neuro-cognitive Model of Move Location in The Game of Go

Terry Bossomaier\*, Jason Traish\*, Fernand Gobet<sup>†</sup> and Peter C. R. Lane<sup>‡</sup>

\*Centre for Research in Complex Systems

Charles Sturt University

Bathurst, Australia.

Email: {tbossomaier,jtraish}@csu.edu.au

<sup>†</sup>School of Social Sciences

Brunel University

London, UK

Email: fernand.gobet@brunel.ac.uk

<sup>‡</sup>School of Computer Science

University of Hertfordshire

Hertfordshire, UK

Email: peter.lane@bcs.org.uk

**Abstract**—Although computer Go players are now better than humans on small board sizes, they are still a fair way from the top human players on standard board sizes. Thus the nature of human expertise is of great interest to artificial intelligence. Human play relies much more on pattern memory and has been extensively explored in chess. The big challenge in Go is local-global interaction – local search is good but global integration is weak. We used techniques based on the cognitive neuroscience of chess to predict optimal areas to move using perceptual chunks, which we cross-validated against game records comprising upwards of five million positions. Prediction to within a small window was about 50%, a remarkable result.

## I. INTRODUCTION

In many complex systems human expertise has dominated computer software. In some it still does, but the number is dwindling. One of the landmark victories was the fall of human chess champion Gary Kasparov to IBM computer Deep Blue in 1997, after he had won the first such match in 1996. More recently IBM computer Watson has beaten the best human players at Jeopardy, a television quiz game requiring general knowledge and natural language understanding.

Yet despite the gradual encroaching of computers on human skill levels, there are still fundamental differences in the algorithmic approaches and a full procedural description of how human expertise works and is acquired is still lacking. Computers such as Deep Blue operate primarily by *search*. Human chess players rely much more on memory for patterns and their experience of similar positions encountered in the past.

Herbert Simon [2] proposed that around 50,000 patterns or chunks are needed to reach a high level of expertise in *any* domain. Chase, Gobet and Simon [1], [2], [12] subsequently explored the nature of these chunks for chess and Gobet and colleagues subsequently created models for identifying

chunks, the CHREST model, discussed in section III. The CHUMP model then uses the chunks to predict moves.

A full solution to the expertise problem requires a model whose performance grows at a similar rate to human performance as a function of exposure. Some successful computer programs, such as *neurogammon* for Backgammon [22], have surpassed the top human players, but use much greater exposure, in this case an order of magnitude more games. The Simon argument was that around 10,000 hours of exposure, feedback and analysis are needed to reach the top.

Although it is difficult to compare the speed and memory of the human brain with a computer because of their radically different computational behavior, it seems to be generally accepted that the human brain achieves around 1 petaflop, about 100 times the top computers in 2011. Thus the best one can hope for is to get the scaling behavior right. Games provide the ideal framework to do this since they have precise metrics for performance and exposure (positions encountered, or games played). In the case of the old oriental game, Go, computers still do not approach human professional levels. Thus the game is doubly interesting, since not only is it a good domain to study human cognition, but it also holds out the possibility of a computer using human strategy to outperform human players.

Competition Go is played on a 19x19 grid and two players, black and white, alternately place *stones* on the intersections, referred to as *points*. The stones do not move, but are killed and removed when completely surrounded by opposing pieces. Unlike chess the branching factor of successive moves is not amenable to pruning in any way so far discovered, which makes search impractical.

However, the latest technique, Monte Carlo tree search, has enabled a leap forward in computer Go, and computers now win on smaller boards, such as 9x9 (in 2011). On the larger

boards, global factors influence local decisions. Anecdotal evidence suggests that top players very rapidly, in less than a second, identify the area of board in which to move, with their first eye fixation on this area. Subsequent fixations may be made elsewhere, but usually this first fixation is where the move is made.

The goal of this paper is to determine the extent to which low level chunks are able to predict the local area where a move might be made. One would not expect predictions to be very accurate, since any given move will require precise local search [7]. Section II provides an overview of research on human expertise in Go. Section III summarises the CHREST and CHUMP models. Section IV describes the use of online game data to measure the success of CHUMP and section V gives the results. Section VI discusses the findings.

## II. THE PSYCHOLOGY OF GO EXPERTISE

Board games, given their complexity and strategic nature, have offered crucial insight for understanding the cognitive processes underpinning experts' thinking. Simon's estimates of 50,000 chunks and 10,000 hours of practice were based on his seminal research with Chase [1] on chess. They used a recall task, where a chess position is briefly presented, and a copy task, where one has to reconstruct a position on a new board whilst the original position stays in sight. They then defined the boundary between chunks as a pair of pieces replaced (a) with an inter-piece latency longer than 2 seconds in the recall task, and (b) with a glance at the stimulus board in the copy task. An analysis of the chess relations between pair of pieces (e.g. attack, proximity) showed that the two definitions led to similar results. Considerable research on chess has largely confirmed but also extended these results (for an overview, see Gobet et al. [10]).

As search trees are much larger in Go than in chess, perception and knowledge would be expected to be even more important in Go than in chess. Empirical research has confirmed the central role of these in Go. In a recall task, the skill effect with game positions was found in Go [18], where a 11x11 version of Go was used. No skill effect was observed in a control task independent from Go. Reitman [20] was interested in seeing whether Chase and Simon's [1] results supporting the presence of chunks with chess players would lead to similar results with Go. She used both a recall task (brief presentation of positions) and a copy task, where one has to reconstruct a position on a new board whilst the original position stays in sight. The results, although limited to one master and one beginner, are interesting. Just as in the chess data, she found that the latency between the placements of two stones was longer when there was a glance at the stimulus position between the placement of the stones than when the two stones were placed in succession. However, an important difference with the chess data was that there was only a poor match between the chunks found in the recall task and in the copy task. Reitman also carried out a partitioning task. This task consists in asking participants to draw boundaries around the clusters of stones that they perceive as meaningful.

While Chase and Simon's results with chess suggested that chunks were organized hierarchically in long-term memory, the results with Go suggested that the Go master's knowledge was organized as overlapping clusters. However, given the very small number of participants in Reitman's study, more data is necessary to reach firm conclusions.

A natural way to study the role of perception is to record eye movements. Unfortunately, very little data are available on Go using this methodology. Yoshikawa and Saito [24], [25] recorded eye movements to understand how players generate candidate moves in Go. Just as in chess [4], players tended to fixate between stones and not on the stones themselves. Players examined only a portion of the board before selecting a move but looked at a larger area after having played their move. These authors also studied how players solved tsume-Go problems (local 'life-and-death' problems used for practising look-ahead skills) when only limited thinking time (4 seconds) was allowed. For the problems solved correctly, the eye fixations of their stronger player (6-dan) were fast (between 200 and 260 ms), which suggests the use of pattern recognition. By contrast, the weaker players had to carry out search to be able to find the solution.

The most comprehensive study on the cognitive mechanisms underpinning expertise in Go was carried out by Masunaga and Horn [18], [19]. They used a remarkably large sample for this kind of study (263 players, from beginners to professionals, spanning 48 levels of expertise). They had a number of Go tasks, and each task was matched by a similar task using material different from Go. These control tasks made it possible to investigate the possibility of transfer. Here, we provide only a sample of their results, focusing on the importance of perception and pattern recognition. In a pattern-recognition task, players were presented with 'atari' (important stone configurations) and were requested to identify them as quickly as they could from foils. Better players were quicker, and the difference was particularly pronounced between professionals and the rest of the players. As expected, there was no skill difference in the corresponding control task, which consisted of finding a target letter amongst Japanese letters. In a pattern-matching task, players had to compare a pair of Go configurations and decide as quickly as possible whether the two configurations were the same. As in the first task, the professionals were quicker than the remainder of the players, but there was no skill effect in the corresponding control task in which pairs of strings of Japanese letters were compared.

In a memory-recognition task under distraction, players were presented with configurations for around 10 seconds. Their task consisted in both counting the number of black and/or white stones and memorizing the position. Once the position was removed from sight, players had to select the position from six configurations. A skill effect was found (particularly pronounced with the professionals), but there was no skill difference in a control task using material different from Go. A similar pattern was found in other tasks measuring memory, intelligence and reasoning, either with Go or with

non-Go material.

Knowledge plays a central role in Go expertise. This can be seen in the rich terminology used in Go books, with concepts such as *kosumi* (a diagonal extension) or *shinogi* (saving a group of stones that was in difficulty) (see Shirayanagi [21]). Yoshikawa et al. [23] found that Go experts, when seeing a new position, first used concepts to provide a global evaluation of their possible moves and their opponent's possibilities. The generation of candidate moves and effective analysis of variations only came in a second stage. This behavior is reminiscent of chess players' behavior [3]. Due to their limited knowledge, novices tended to create their own terms to designate concepts. While intermediate players are able to understand when concepts are relevant in a given position, these concepts are not linked to pertinent evaluations or plans. Only advanced players have practiced and studied enough to acquire this kind of knowledge, where perceptual and conceptual information is linked.

### III. CHREST AND CHUMP

#### A. Overview

CHREST (Chunk Hierarchy and REtrieval STRuctures) [8], [11] consists of four main components: (a) a long-term memory, where chunks are stored, (b) visual and auditory short-term memories (STMs) with limited capacity, (c) modules dealing with visual and auditory mental images, and (d) attention mechanisms. CHREST is a self-organizing, dynamic system, in which chunks are accessed by traversing a discrimination net, which is a treelike structure consisting of a set of nodes (chunks) connected by links. Learning occurs through creating new nodes, adding information to these nodes, creating links between nodes, creating *templates* (nodes with schema-like properties) and creating micro-productions (links between a perceptual chunk and a possible action or sequence of actions). In the literature, when the focus is on the micro-productions, as in this paper, the model is usually known as CHUMP [10].

STM is dynamic, in the sense that older chunks are continuously updated by new incoming information. The largest chunk recognized so far is used to direct eye movements; the rationale is that eye movements that were useful in the past are likely to be useful in the future if a similar constellation of pieces is present on the board. Other heuristics used by the eye movements include a preference to fixate on 'novel' information (pieces on the periphery that have not been considered so far), and also to scan parts of the board so far not observed.

When the focus is on simulating human data, the model uses time parameters, such as the time to create a new chunk (8 seconds) and the time to encode a chunk into STM (50 milliseconds). Recently, CHREST's decision making ability has been extended so that it can use attentional and problem-solving heuristics to supplement the information provided by pattern recognition. CHREST learns from complex input data, including realistic data reflecting the statistical structure and complexity of the environment. CHREST can be seen as

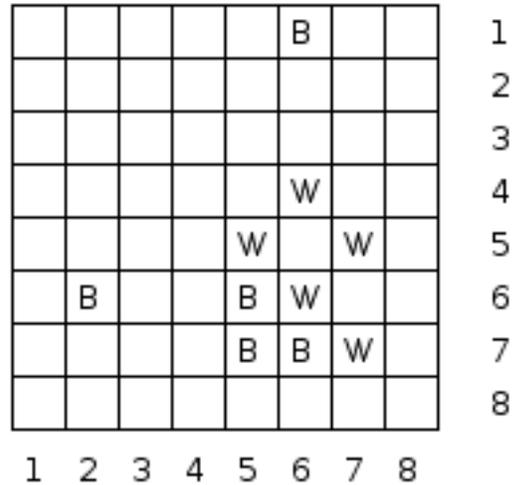


Fig. 1. Example Go position, on a reduced 8x8 board.

using incremental case-based reasoning, where the cases are built up piece-meal with new information, rather than being constructed fully on the single exposure of a case.

CHREST currently provides state-of-the-art models in multiple psychological domains including: perception and memory in expert behavior in chess and Awele [4], [9], [12], concept formation [17], problem solving with diagrams [16] and acquisition of syntax and vocabulary [5],[6], [13], [14] In all these cases, it uses naturalistic input (e.g. masters' games in chess) for training the model.

In this paper, we use the micro-productions to predict moves in Go. As CHUMP is trained, it associates the move in a given position with the chunks within that position. In the test phase, CHUMP recognizes a set of chunks in the test position, and retrieves the associated moves. CHUMP then predicts the most frequently occurring legal move from this set. (In future experiments, we plan to use an extension of CHUMP, called SEARCH, which provides lookahead search routines; see [7], [15] for a description.)

#### B. Learning Chunks

Perceptual chunks are learnt in an unsupervised manner from each game position in turn. The eye is guided across the grid and perceives the pieces in a part of the board, depending on the size of its field of view. For example, looking at square (5, 5) of Figure 1, the model would retrieve the pieces [W 5 5] [B 5 6] [B 5 7] [W 6 4] [W 6 6] [B 6 7] [W 7 5] [B 7 7].

The representation used for the pieces is an *item-on-square* representation: the name of the piece and its row/column coordinates are given. The chunks are thus tied to specific board positions. This representation has been found to be effective for generalising patterns across a larger sector of the board; discussion of this representation may be found in [15].

We can illustrate the two key learning mechanisms used within CHREST using two simple Go patterns. The learning

mechanisms are *familiarisation* and *discrimination*. The learning process generates a discrimination network, using tests based on specific pieces or groups of pieces to sort an input pattern to an internal node within LTM. This internal node contains a *familiar pattern*, otherwise known as a *chunk*. Thus, chunks are simply patterns which have become familiar.

The discrimination process uses mismatches between the currently input pattern and the retrieved node to extend the network with new test links and new nodes. The familiarisation process adds information from the current input pattern to a retrieved node, when the two are compatible. Figure 2 illustrates the process. The greyed nodes represent test links, and the clear nodes contain chunks. In (a), the model has learnt about one pattern, with two pieces. When presented with the new pattern, the model retrieves node 3 but there is a mismatch because the colour of the piece on square (5, 6) is incorrect. So in (b), the model has discriminated between the two patterns adding a new node for the new chunk (single-piece chunks are also learnt incrementally, to hold the ‘alphabet’ of the patterns); this illustrates *discrimination*. As learning proceeds, the new node is filled out with information from the current pattern; this is *familiarisation*. Note that chunks may hold more, less or equal amounts of information as the set of tests required to retrieve them. Further details of the learning mechanisms, including the way learning can proceed faster, a pattern at a time instead of a piece at a time, and also how subsets of the network become associated into board-covering *templates*, may be found in [11].

### C. Eye Fixations

The perceptual chunks are learnt in an unsupervised manner from the database of games. Each position is scanned for 100 fixations, using the heuristics described above to guide the eye. New chunks are created and familiarised. As an illustration of what has been learnt, Figure 4 shows a trace of the model’s eye fixations on the position in Figure 1. A model of Go positions was trained from a set of 21,000 positions, training continuing until the model’s LTM contained 10,000 chunks. An indication of the fixations is shown in Figure 3.

The eye fixations show the range of heuristics used in retrieving information: the eye initially is located at one of the central squares, and then proceeds using one of a set of generic heuristics. In this example, three heuristics are illustrated.

- 1) *Random item* fixates a previously unseen item within the field of view, but not at the centre.
- 2) *Random place* fixates a previously unseen square within the field of view, but not the centre.
- 3) *LTM* uses the largest chunk currently in STM to guide the next fixation. The heuristic proposes the location of information required to pass one of its test links.

At the end of its cycle of fixations, the model’s STM contains pointers to nodes 3940, 9, 7087 and 5087. The trace above shows the *images* of these retrieved nodes. This information, and information about the moves associated with each chunk, is used by CHUMP for further learning and to generate moves.

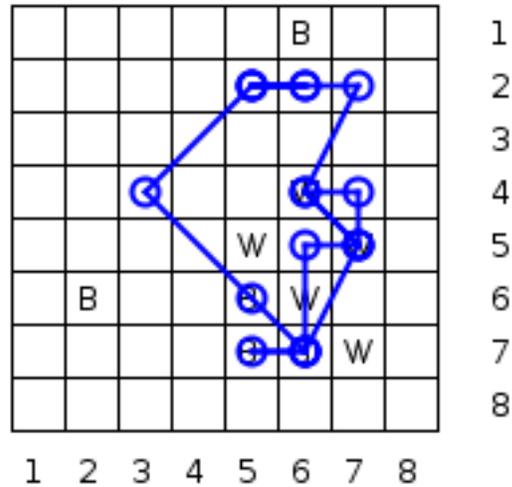


Fig. 3. Illustration of fixations on example Go position.

#### Fixations:

- (6, 7) Random item heuristic
- (6, 5) Random place heuristic
- (7, 5) Random item heuristic
- (6, 4) LTM heuristic
- (7, 2) Random place heuristic
- (5, 2) LTM heuristic
- (6, 2) Random place heuristic
- (5, 2) LTM heuristic
- (6, 2) Random place heuristic
- (5, 2) LTM heuristic
- (3, 4) Random place heuristic
- (5, 6) Random place heuristic
- (6, 7) Random item heuristic
- (5, 7) Random item heuristic
- (6, 7) Random place heuristic
- (6, 7) Random item heuristic
- (7, 5) Random place heuristic
- (6, 4) LTM heuristic
- (7, 4) Random place heuristic
- (7, 5) Random item heuristic

#### Chunks retrieved:

- Node: 3940 < [W 5 5] [B 5 6] [B 5 7]  
  [W 6 6] [B 6 7] >
- Node: 9 < [B 2 6] [W 5 5] [B 5 6] >
- Node: 7087 < [W 5 5] >
- Node: 5087 < [B 5 6] >

Fig. 4. An illustration of the eye fixations made by the model

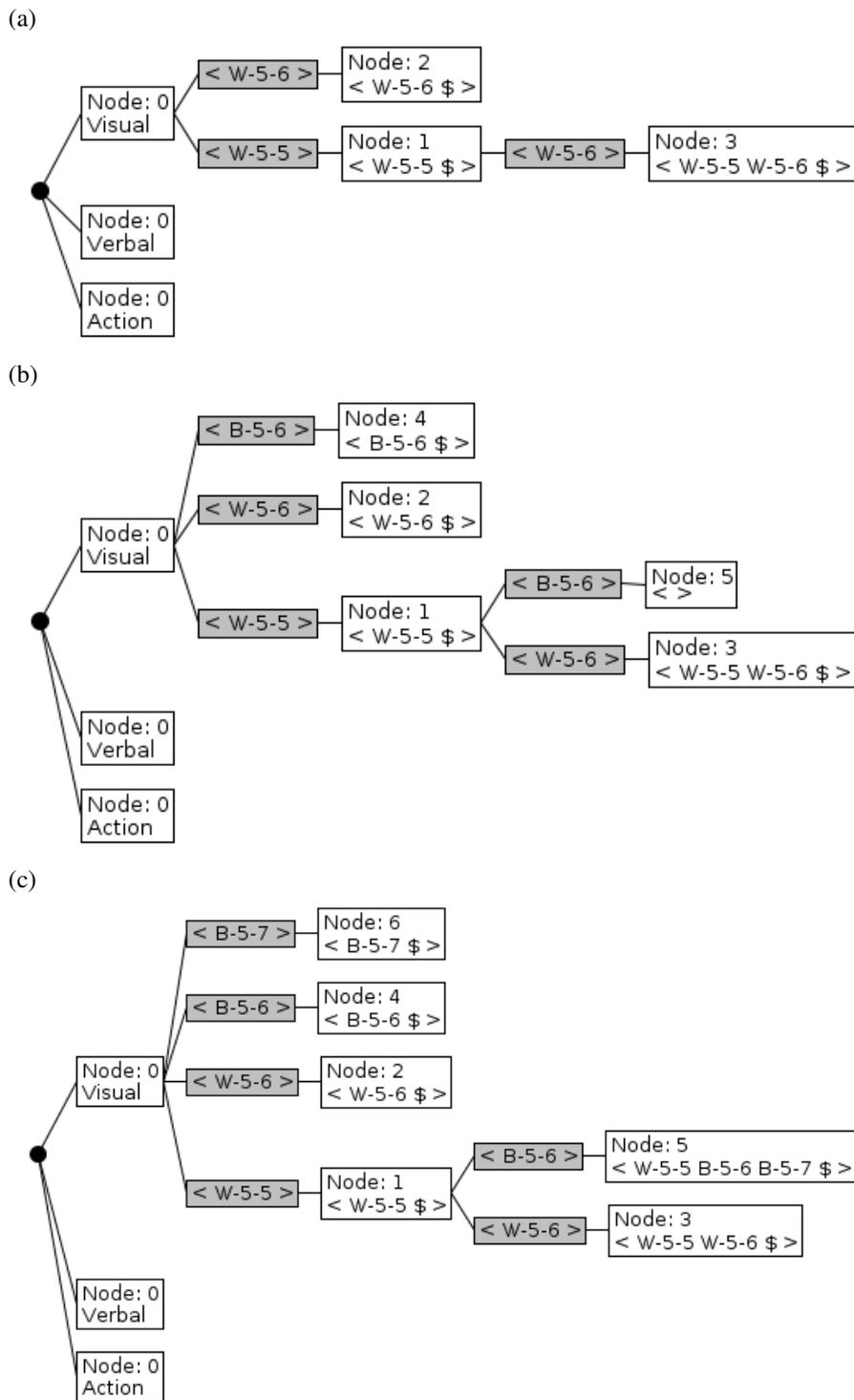


Fig. 2. A trace of three learning steps. In (a) the model has learnt pattern [W 5 5] [W 5 6]. In (b), the model has seen [W 5 5] [B 5 6] [B 5 7], and discriminated a new node. In (c), the model has completely learnt the new pattern.

#### IV. METHODOLOGY

CHUMP was run on a series of games provided by GoGoD<sup>1</sup> to predict the actual move and the move within a small, 5x5 window. The average errors in both directions between the prediction and the exact move were also calculated.

For reference a random estimate of the move was calculated. A random move could be illegal, since the point may be occupied or have no liberties. Thus random moves were generated iteratively until a legal move was found.

The 5,000,000 positions were partitioned into lots of 750,000 to work on different date ranges of games: as indicated by the Table 1's titles A–G.

#### V. RESULTS

Tables 1A-G show the performance of CHUMP for predicting the exact move and the 5x5 window in which the move occurs. Surprisingly, the prediction depends only weakly on the number of positions, wherein table G only had 500,000 positions and still maintained around 50% accuracy. This suggests that the most important chunks have been found early in the learning cycle.

Tables 1A-G show the results of tenfold cross validation using 100 fixations and a visual short term memory of 90. The number correct within the 5x5 window is around 50%. Both the exact number correct and the average error are considerably better than random.

#### VI. DISCUSSION

The prediction of the location of a move to within a small window is surprisingly good, considering that there are *no* specifically Go heuristics used in the position estimation. The implication is that the low level building blocks, which are the key to expert position memory go a substantial way to predicting which areas of the board have the highest priority for action.

There has been little eye movement work for Go, hence the eye movement protocols used herein are at best intuitive (see section II). Thus the fixations are noisy and more accurate eye movement data would undoubtedly improve the accuracy and reduce the number of fixations required.

The results improve as the short term memory increases, with values much higher than that of humans. This might result from the poor eye movement heuristics, making the chunks themselves noisy also. Thus a larger number of chunks is required to get an adequate prediction.

The approach here is thus a good starting point for the goal of building an AI along the lines of human expertise. The major problem of selecting the area to move within a large board is partially solved by use of chunk data harvested from expert games.

Further work will attempt to extend this approach to attach meaningful information to the chunks and templates during learning. The template theory, which CHREST and CHUMP implement, argues that templates are associated with strategic

##### A. 750,000 positions, 0196 – 1951

Predictions CHUMP	Predictions Correct	Predictions Window	xDiff	yDiff
Average	723	32350	3.97	3.98
Std Dev	29.2	181	0.0462	0.0644
<b>Random</b>				
Average	195	17627	5.80	5.80
Std Dev	19.6	109	0.0120	0.0202

##### B. 750,000 positions, 1951 – 1976

Predictions CHUMP	Predictions Correct	Predictions Window	xDiff	yDiff
Average	710	31432	4.05	4.10
Std Dev	19.7	263	0.0850	0.0694
<b>Random</b>				
Average	194	17728	5.79	5.80
Std Dev	10.1	127	0.0177	0.0190

##### C. 750,000 positions, 1976 – 1988

Predictions CHUMP	Predictions Correct	Predictions Window	xDiff	yDiff
Average	665	31770	4.02	4.09
Std Dev	34.0	222	0.0774	0.0722
<b>Random</b>				
Average	190	17613	5.79	5.82
Std Dev	11.4	122	0.0226	0.0196

##### D. 750,000 positions, 1988 – 1993

Predictions CHUMP	Predictions Correct	Predictions Window	xDiff	yDiff
Average	769	32357	4.00	3.99
Std Dev	38.6	213	0.0584	0.0653
<b>Random</b>				
Average	198	17621	5.81	5.81
Std Dev	8.3	129	0.0143	0.0155

##### E. 750,000 positions, 1993 – 1997

Predictions CHUMP	Predictions Correct	Predictions Window	xDiff	yDiff
Average	572	22411	3.98	3.98
Std Dev	18.2	255	0.069	0.061
<b>Random</b>				
Average	135	12193	5.79	5.82
Std Dev	13.1	89.5	0.0241	0.0188

##### F. 750,000 positions, 1997 – 2002

Predictions CHUMP	Predictions Correct	Predictions Window	xDiff	yDiff
Average	784	32271	3.97	3.97
Std Dev	24.6	457	0.0722	0.0627
<b>Random</b>				
Average	195	17645	5.80	5.80
Std Dev	12.8	78.1	0.0177	0.0132

##### G. 518,219 positions, 2002 – 2003

Predictions CHUMP	Predictions Correct	Predictions Window	xDiff	yDiff
Average	572	22411	3.98	3.98
Std Dev	18.2	255	0.069	0.061
<b>Random</b>				
Average	135	12193	5.79	5.82
Std Dev	13.1	89.5	0.0241	0.0188

TABLE I

<sup>1</sup><http://www.gogod.co.uk>

information, memory of previous games, etc. Examples of this association in the game of chess may be seen in [12], [15]; we expect these ideas to generalise naturally to Go.

#### ACKNOWLEDGEMENTS

This work was supported by the Australian Research Council under Discovery Project DP0881829

#### REFERENCES

- [1] W. Chase and H. Simon, "The mind's eye in chess," in *Visual Information Processing*, C. W.G., Ed. Academic Press, NY, 1973, pp. 215–281.
- [2] —, "Perception in chess," *Cognitive Psychology*, vol. 4, pp. 55–81, 1973.
- [3] A. De Groot, *Thought and choice in chess*. The Hague: Mouton Publishers, 1965.
- [4] A. De Groot and F. Gobet, *Perception and memory in chess*. Assen: Van Gorcum, 1996.
- [5] D. Freudenthal, J. M. Pine, J. Aguado-Orea, and F. Gobet, "Modelling the developmental patterning of finiteness marking in English, Dutch, German and Spanish using MOSAIC," *Cognitive Science*, vol. 31, pp. 311–341, 2007.
- [6] D. Freudenthal, J. M. Pine, and F. Gobet, "Simulating the referential properties of dutch, german and english root infinitives in mosaic," *Language Learning and Development*, vol. 5, pp. 1–29, 2009.
- [7] F. Gobet, "A pattern-recognition theory of search in expert problem solving," *Thinking and Reasoning*, vol. 3, pp. 291–313, 1997.
- [8] —, "Réseaux de discrimination en psychologie: L'exemple de chrest (discrimination networks in psychology: The example of chrest)," *Swiss journal of Psychology*, vol. 60, pp. 264–277, 2001.
- [9] —, "Using a cognitive architecture for addressing the question of cognitive universals in cross-cultural psychology," *Journal of Cross-Cultural Psychology*, vol. 40, pp. 627–648, 2009.
- [10] F. Gobet, A. de Voogt, and J. Retschitzki, *Moves in mind*. Hove, UK: Psychology Press, 2004.
- [11] F. Gobet, P. Lane, S. Croker, P. Cheng, G. Jones, I. Oliver, and J. Pine, "Chunking mechanisms in human learning," *Trends in Cognitive Sciences*, vol. 5, pp. 236–243, 2001.
- [12] F. Gobet and H. Simon, "Five seconds or sixty? presentation time in expert memory," *Cognitive Science*, vol. 24, pp. 651–682, 2000.
- [13] G. Jones, F. Gobet, and J. M. Pine, "Linking working memory and long-term memory: A computational model of the learning of new words," *Developmental Science*, vol. 10, pp. 853–873, 2007.
- [14] —, "Computer simulations of developmental change: The contributions of working memory capacity and long-term knowledge," *Cognitive Science*, vol. 32, pp. 1148–1176, 2008.
- [15] P. C. R. Lane and F. Gobet, "Perception in chess and beyond: Commentary on Linhares and Freitas (2010)," *New Ideas in Psychology*, vol. 29, pp. 156–61, 2011.
- [16] P. Lane, P. Cheng, and F. Gobet, "Chrest: Investigating how humans learn to solve problems using diagrams," *AISB Quarterly*, vol. 103, pp. 24–30, 2000.
- [17] P. Lane and F. Gobet, "Discovering predictive variables when evolving cognitive models," in *Pattern Recognition and Data Mining, Pt 1, Proceedings. Lecture Notes in Computer Science*. New York, NY: Springer, 2005, vol. 3686, pp. 108–117.
- [18] H. Masunaga and J. Horn, "Characterizing mature human intelligence - expertise development," *Learning and Individual Differences*, vol. 12, no. 1, pp. 5–33, 2000.
- [19] —, "Expertise and age-related changes in components of intelligence," *Psychology and Aging*, vol. 16, no. 2, pp. 293–311, 2001.
- [20] J. S. Reitman, "Skilled perception in go: Deducing memory structures from inter-response times," *Cognitive Psychology*, vol. 8, pp. 336–356, 1976.
- [21] K. Shirayanagi, "Basic study of knowledge processing for go," NTT International Technical Report, Tech. Rep. 12967, 1986.
- [22] G. Tesauro, "Temporal difference learning and td-gammon," *Commun. ACM*, vol. 38, no. 3, pp. 58–68, 1995.
- [23] A. Yoshikawa, T. Kojima, and Y. Saito, "Relation between skill and the use of terms: An analysis of protocols of the game of go," in *Computers and Games*, H. J. van den Herik and H. Iida, Eds. Berlin: Springer, 1998, pp. 282–299.
- [24] A. Yoshikawa and Y. Saito, "The difference of the knowledge for solving tsume-go problem according to skill," in *Proceedings of the Fourth Game Programming Workshop in Japan 97*, H. Matsubara, Ed. Kanagawa: Computer Shogi Association, 1997, pp. 87–95.
- [25] —, "Hybrid pattern knowledge: Go players knowledge representation for solving tsume-go problems," in *Proceedings of the First International Conference on Cognitive Science in Korea*. Seoul: Seoul National University, 1997, pp. 34–139.