Article

Expertise increases planning depth in human gameplay

https://doi.org/10.1038/s41586-023-06124-2

Received: 3 June 2021

Accepted: 24 April 2023

Published online: 31 May 2023

Check for updates

Bas van Opheusden[1,2]✉, Ionatan Kuperwajs[1], Gianni Galbiati[1,3], Zahy Bnaya[1], Yunqi Li[1] & Wei Ji Ma[1]

A hallmark of human intelligence is the ability to plan multiple steps into the future[1,2]. Despite decades of research[3–5], it is still debated whether skilled decision-makers plan more steps ahead than novices[6–8]. Traditionally, the study of expertise in planning has used board games such as chess, but the complexity of these games poses a barrier to quantitative estimates of planning depth. Conversely, common planning tasks in cognitive science often have a lower complexity[9,10] and impose a ceiling for the depth to which any player can plan. Here we investigate expertise in a complex board game that offers ample opportunity for skilled players to plan deeply. We use model fitting methods to show that human behaviour can be captured using a computational cognitive model based on heuristic search. To validate this model, we predict human choices, response times and eye movements. We also perform a Turing test and a reconstruction experiment. Using the model, we find robust evidence for increased planning depth with expertise in both laboratory and large-scale mobile data. Experts memorize and reconstruct board features more accurately. Using complex tasks combined with precise behavioural modelling might expand our understanding of human planning and help to bridge the gap with progress in artificial intelligence.

Real-world decision-making often involves sequences of actions with multiple alternatives at each stage. Such decisions require people to mentally simulate the consequences of candidate actions multiple steps into the future using an internal model of the environment—a process known as planning. Examples of ecologically relevant planning tasks are navigation, preparing a meal, making career decisions and playing strategy games. Given the importance of planning, a natural hypothesis is that skilled decision-makers are more successful because they plan further into the future. Following seminal work[3,11], a growing body of literature has investigated the nature of expertise in planning by studying how expert chess players differ from less-skilled counterparts[7,12,13]. This literature has explained the superior performance of experts as being due to better pattern recognition[8,14,15] and/or deeper search[4,7,16–18]. However, developing computational cognitive models that accurately predict the moves of individual chess players has proven to be difficult[6,19] and, instead, studies rely on clever experimental manipulations[5,20] or verbal reports [21].

By contrast, cognitive and neuroscience studies often use simpler tasks so that behaviour and neural activity can be precisely modelled. These studies provide strong evidence that humans and animals engage in forward planning at decision time and suggest candidates for the underlying neural substrates[1,2]. Human choices in the classic two-step task[9] reveal a goal-directed planning component to their decision-making. In a more complex goal-directed decision-making task, it was previously found[10,22] that people plan along multiple branches in a decision tree, but eliminate unpromising branches by pruning. Planning was studied[23] in a fast-paced, dynamic environment, finding human behaviour consistent with planning several steps into the future. It was previously demonstrated[24] that people use prospective information to guide current choices, and located the representation of prospective information to cingulate and prefrontal cortices.

In animals, hippocampal place cells display signatures of prospective activity along candidate trajectories[25], particularly when an animal stops at a choice point[26]. Hippocampal neural activity has been associated with both planning at decision time and planning in the background[27]. Moreover, evidence for planning in animals has been found in adaptations of the two-step task for rodents[28–30].

These human and animal studies rely on planning tasks of limited complexity, which imposes a ceiling for the depth of planning and makes them less suitable to study the nature of expertise. The perfect task for studying expertise in planning needs to be complex enough that strong play requires thinking multiple steps ahead, but tractable for computational modelling. Furthermore, to encourage learning, it should be novel, have simple rules and be engaging. We introduce a task that satisfies these competing desiderata, develop a computational cognitive model for human decision-making, validate it using choice, response time and eye movement data, and finally use the model to investigate the nature of expertise in planning.

The four-in-a-row behavioural task

Our task is a generalization of tic-tac-toe, in which two players alternate placing pieces on a 4-by-9 board (Fig. 1a), aiming to get four pieces in a row horizontally, vertically or diagonally. The game can be played online (https://weijimalab.github.io/). With approximately 1.2×10^{16}

[1]Center for Neural Science and Department of Psychology, New York University, New York, NY, USA. [2]Department of Computer Science, Princeton University, Princeton, NJ, USA. [3]Present address: Vidrovr, New York, NY, USA. ✉e-mail: basvanopheusden@gmail.com

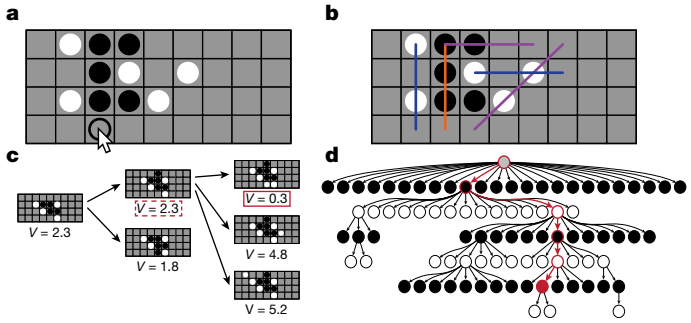


Fig. 1 | Task and computational model. a, Example board position in the four-in-a-row game. Two players, black and white, alternate placing pieces on the board, and the first player to achieve four-in-a-row wins the game. In this position, black is about to win by moving on the third square in the bottom row (open circle, mouse cursor). **b**, The features used in the heuristic function. Features with identical colours are constrained to have identical weights. The model also includes a central tendency feature and a four-in-a-row feature. **c**, Illustration of the heuristic search algorithm. In the root position (left), black is to move. After expanding the root node with two candidate moves for black and evaluating the resulting positions using $V(s)$, the algorithm selects the highest-value node ($V = 2.3$) on the second iteration and expands it with three candidate moves for white. The algorithm evaluates the resulting positions, and backpropagates the lowest value ($V = 0.3$), as white is the opponent. It then compares that value against its alternatives in each intermediate node of the tree to decide in which direction to expand the tree in the algorithm's next iteration. **d**, The decision tree built by the model with fitted parameters on an example board. The red nodes indicate the principal variation—the sequence of highest-value moves for both players. Note that different branches are evaluated to different depths.

non-terminal states (Supplementary Information 3), four-in-a-row has a state space complexity[31] far beyond common cognitive science tasks[32].

Computational cognitive model

We adapted our model of human planning from the artificial intelligence literature, in particular, heuristic search[33,34]. The core of a heuristic search algorithm is a heuristic function, which maps a board state to a value estimate, often as a weighted linear combination of board features. For example, a common chess heuristic is to count pieces for both players, with different point values for different pieces (pawns, knights, rooks and so on). Similarly, our heuristic function counts how often particular features (Fig. 1b) appear on the board. It weighs those counts by feature weights, resulting in a quick-to-compute but approximate value estimate.

To refine the value estimate, the model explores a decision tree of possible continuations (Fig. 1c,d). We based our model on best-first search[35]. This algorithm iteratively expands nodes on the principal variation, the sequence of best moves for both players given the current decision tree. Best-first search is well-suited for human planning, as it preferentially allocates computational resources to relevant branches of the decision tree[36].

Our other model choices derive from cognitive science. Inspired by previous studies[10,22], the model prunes branches in the decision tree with low heuristic value. This improves the efficiency of search, but the model may not spot winning sequences. Moreover, to enable the model to capture variability and make human-like mistakes, we added Gaussian noise to the heuristic function and included feature dropout. For each move that the model makes, it randomly omits feature instances from the heuristic function before performing search. We interpret these feature omissions cognitively as lapses of selective attention[37].

Model validation

We conducted several experiments and analyses to validate our computational model for human decision-making in four-in-a-row. In our first experiment, 40 human participants played games against other human players without any time pressure. For each participant, we estimated model parameters (feature weights, feature drop rate, decision tree size, pruning threshold and noise level) using fivefold cross-validation. The model predicts out-of-sample choices with 40.8 ± 1.4% accuracy (mean ± s.e.m. across participants; two-sample t-test against chance: $t_{39} = 26$, $P < 0.001$). Figure 2a shows an example model prediction, and Fig. 2b shows the model accuracy for each participant. In Supplementary Information 4–7 (Extended Data Figs. 1–4), we validate our model specification by comparing against 22 alternative models, including ones that lesion model components, and we show that the model's parameters can be reliably estimated using custom fitting methods[38,39] and that it predicts multiple summary statistics. We found that a feature-based value function, tree search and a mechanism for attentional oversights are essential for predicting human choices (Supplementary Information 8 and Extended Data Fig. 5), and selected the main model as a parsimonious representative of that model class.

Next, we performed a generalization experiment in which 40 participants performed three tasks: playing against computer opponents, a two-alternative forced-choice (2AFC) between moves in a given position and a board-evaluation task. We optimized the positions to make these decisions challenging, both to our participants and for the model to predict. For each player, we estimated model parameters from their choices during human-versus-computer games, and predicted their 2AFC and evaluation decisions. For both tasks, the model predicted people's choices above chance (percentage of correct 2AFC = 58.6 ± 1.0%, $t_{39} = 8.3$, $P < 0.001$; Fig. 2c; correlation predicted-observed evaluations: $\rho = 0.377 \pm 0.039$, $t_{39} = 9.6$, $P < 0.001$; Fig. 2d). These results suggest that the model can generalize between different choice tasks in the four-in-a-row domain. In Supplementary Information 9, we show that the model outperforms an oracle model (which makes objectively correct moves with random tie-breaking), suggesting that the model captures the subjective preferences of individual participants.

Moreover, we conducted a Turing test experiment[40], in which 30 observers, familiar with the game, decided whether sequences of moves (9.38 on average) were generated by the model or by human players. Human observers achieved only 55.4% discrimination accuracy (Extended Data Fig. 6), suggesting that the model makes human-like decisions.

We tested the main model's ability to predict process data by analysing response times and eye movements. To predict response times, we estimate model parameters from choice data, and we extend the best-first search algorithm using an early-stopping rule, which terminates the search when the model's decision is unlikely to change with more iterations (Supplementary Information 10). We then use the decision tree built by the model on each trial as a predictor for response time. Figure 2e shows the predicted and observed response times (in logarithmic space) across all of the participants in the human-versus-human experiment, and Fig. 2f shows the Pearson correlation for each participant ($\rho = 0.351 \pm 0.029$, $t_{39} = 12$, $P < 0.001$).

To analyse eye movements, we conducted an experiment in which ten participants played against computer opponents while we tracked their eye movements with an infrared video-based eye tracker (Supplementary Information 11). Figure 2g shows one participant's fixation trajectory in an example board position. We estimated the distribution of squares that a participant overtly attends to on an individual trial by convolving their fixation trajectory with a Gaussian filter, truncating to unoccupied squares and averaging in time. Figure 2h,i shows that the distribution of squares visited by the cognitive model during its search process resembles this distribution of attention (mean correlation across participants: $\rho = 0.535 \pm 0.024$, $t_9 = 21$, $P < 0.001$). In Extended Data Fig. 7, we show that this correlation is driven by branches in the decision tree occasionally reaching up to seven moves deep.

The ability of the model to predict both response times and eye movements on individual trials suggests that people plan their moves by

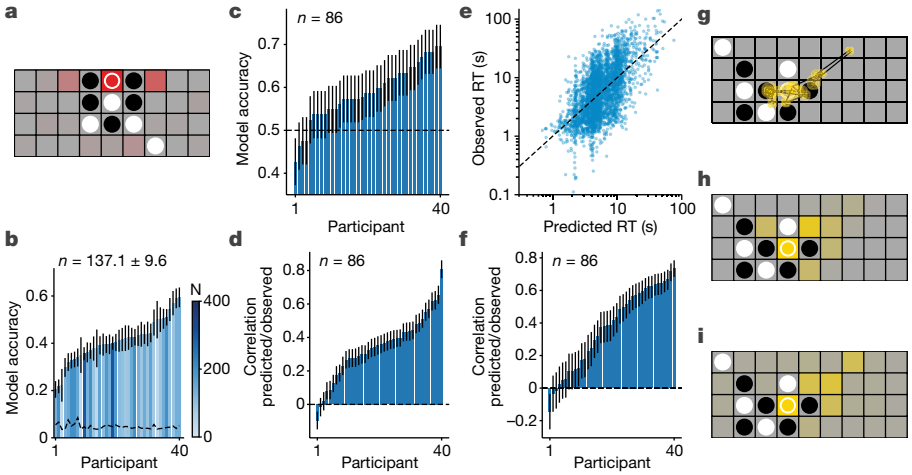


Fig. 2 | The model accounts for multivariate data and generalizes to unseen data. a, Example board position from a human-versus-human game. The open circle indicates the move that the active player (white) chose. The red shading indicates the probability distribution of that participant's next move, as predicted by the model with parameters inferred for that participant using fivefold cross-validation. **b**, Model accuracy (percentage of correctly predicted moves) for each participant in human-versus-human games, ranked from worst to best predicted. Data are mean ± s.e.m. *n* denotes the number of trials per participant. The dashed line represents the accuracy of a 'chance' model, which assumes that people move onto a randomly selected unoccupied square. **c**, Model accuracy for 2AFC decisions in the generalization experiment. For each participant, we estimated model parameters from that participant's moves in games against computer opponents. **d**, The same as in **c**, but for the correlation between predicted and observed responses on the board-evaluation task. **e**, Predicted and observed response times (RT) across all participants in the human-versus-human data. We exclude any positions with fewer than 6 or more than 30 pieces on the board. **f**, The correlation between predicted and observed response times for each participant. **g**, The trajectory of eye movements on one example trial. The black lines represent saccades and the yellow circles represent fixations. The circle area indicates the duration of fixation. **h**, The estimated distribution of overt attention across unoccupied squares, obtained by convolving the eye trajectory with a Gaussian filter. **i**, The distribution of squares visited by the model's search algorithm, with parameters estimated from the participant's choices.

building decision trees, using an algorithm similar to that in our computational cognitive model.

The effect of expertise on planning

The model enables us to investigate how expert players differ from novices. To do so, we performed a learning experiment in which 30 participants played against computer opponents for five sessions, spaced no more than 2 days apart. We measured the task performance of the participants using Elo ratings[41], with a common baseline across all experimental data (see the 'Playing strength estimation using Bayeselo' section in the Methods).

In Fig. 3a, we show that participants played stronger in later sessions (linear regression: $\beta = 21.6 \pm 4.6$, $P < 0.001$). To investigate which aspects of people's decision-making process underlie this increase, we converted the set of parameters inferred for each participant in each session to three metrics: planning depth, feature drop rate and heuristic quality (Supplementary Information 2.7). We define planning depth as the length of the principal variation in the tree—an approximate measure of the number of steps that an individual thinks ahead. The feature drop rate is defined as the attentional lapse probability, a model parameter. Finally, we define the heuristic quality as the correlation between heuristic and objective value—measuring the 'correctness' of the feature weights. These metrics map to different hypotheses on the nature of expertise (Discussion). In Extended Data Fig. 2, we show that these metrics, and planning depth in particular, can be reliably inferred from choice data, and together explain 56.7% of the variance in playing strength (Supplementary Information 12).

Figure 3b–d shows that planning depth increases across sessions ($\beta = 0.255 \pm 0.061$, $P < 0.001$) while feature drop rate decreases ($\beta = -0.0119 \pm 0.0028$, $P < 0.001$). Heuristic quality does not increase, and even decreases slightly ($\beta = -0.0067 \pm 0.0020$, $P = 0.0012$). In Extended Data Fig. 8, we show that individual differences in playing strength are also correlated with planning depth and feature drop rate, and not heuristic quality. Finally, in Supplementary Information 12, we break down the overall gain of 90 ± 26 Elo points between sessions 1 and 5 as a gain of 36 ± 11 points due to increased planning, a gain of 46 ± 12 due to attention and a loss of 6.6 ± 3.5 points due to heuristic quality. These results suggest that stronger players plan deeper and have fewer lapses of attention. We find no evidence for improvements in feature weights.

In Extended Data Fig. 8, we show that the participants play faster in later sessions, verifying that the planning depth increase cannot be due to slower play. Another potential concern is that all parameter estimates and metrics depend on the model specification, which may not match human planning algorithms. Specifically, people may use features that are not present in our heuristic function, and there may be a worry that the model confuses increases in the weights of those features across sessions with increased planning. However, we note that planning depth and feature weights are not confusable, at least for features in our model (Extended Data Fig. 2). Moreover, Extended Data Table 1 shows that our correlations between expertise and metrics are robust across alternative model specifications. Although the existence of additional features in people's heuristic functions is theoretically impossible to rule out, we have no evidence suggesting that adding features to the model will change the main result of deeper planning and improved attention with expertise.

The effect of time pressure on planning

To experimentally validate the planning depth metric, we conducted a time-pressure experiment in which 30 participants played against computer opponents, with a time limit of 5, 10 or 20 s per move, randomly sampled for each game. In Extended Data Fig. 8, we show that this manipulation is effective at changing the response time of the participants. We predicted that, if planning depth approximately measures the amount of computations that a participant performs while making a move, it should scale with time used for that move[12,42,43]. Figure 3f shows that planning depth is overall lower in the time-pressure experiment compared with in the learning experiment and indeed increases with longer time limits ($\beta = 0.042 \pm 0.018$, $P = 0.019$). However, despite this increase, we found no improvement in the participants' playing

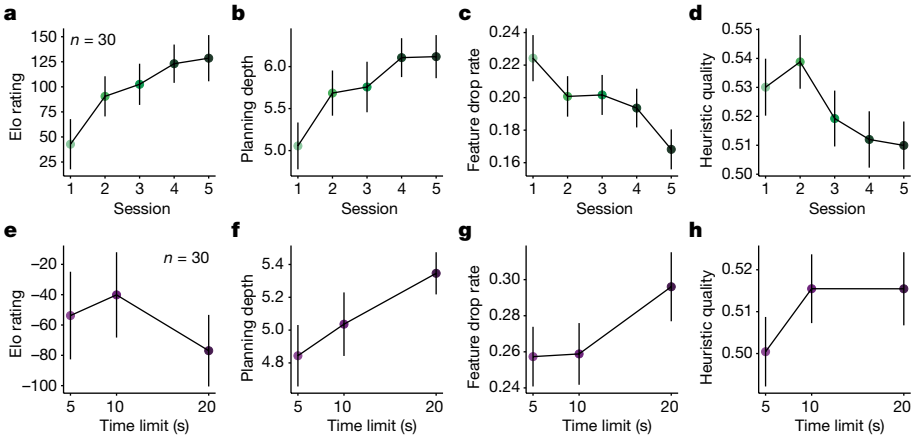


Fig. 3 | The effects of expertise and time pressure on planning. a, The average Elo rating of participants in the learning experiment as a function of session number. Data are mean ± s.e.m. across participants. **b**, The average depth to which participants plan, as estimated by the behavioural model. **c**, The same as in **b**, for feature drop rate. **d**, The same as in **b**, for heuristic quality. **e**, The average Elo rating of the participants in the time-pressure experiment, as a function of the time limit. **f**–**h**, The planning depth (**f**), feature drop rate (**g**) and heuristic quality (**h**) as in **b**–**d**, respectively, but for the time-pressure experiment.

strength ($\beta = -2.0 \pm 1.6$, $P = 0.21$; Fig. 3e). The model suggests a potential explanation for the lack of performance—at the most relaxed time limit, people overlook features more often ($\beta = 0.0027 \pm 0.0010$, $P = 0.009$; Fig. 3g and Supplementary Fig. 10), and the dropped features cancel out the benefit of increased search. Finally, in this experiment, the heuristic quality does not change with time pressure ($\beta = 0.00086 \pm 0.00056$, $P = 0.13$; Fig. 3h).

Generalization to large-scale mobile data

In all of these experiments, we investigated expertise in participants recruited to perform a psychology experiment in a laboratory context. It is not clear whether our expertise results will generalize to a more natural context for acquiring expertise. To address this issue, we collaborated with Peak, a mobile app company (https://www.peak.net), to collect a large-scale dataset of users playing a visually enriched version (see the 'Large-scale mobile data' section of the Methods) of four-in-a-row at their leisure in their daily environment.

We analysed data from 1,000 randomly selected users who played at least 100 games; this number of games approximately matches the total experience of participants in our learning experiment. For each user, we grouped their experience into 5 blocks of 20 games, and estimated model parameters for each block (see the 'Large-scale mobile data' section of the Methods). As before, playing strength ($\beta = 1.13 \pm 0.04$, $P < 0.001$; Fig. 4a) and the depth of planning ($\beta = 0.0108 \pm 0.0010$, $P < 0.001$; Fig. 4b) increase with experience, whereas the feature drop rate decreases ($\beta = -2.58 \times 10^{-4} \pm 4.7 \times 10^{-5}$, $P < 0.001$; Fig. 4c). We validated that the increase in the planning depth of the user was not a result of slower play (Supplementary Fig. 11), replicating the results from the laboratory experiment. In this experiment, we also observed a reliable increase in heuristic quality ($6.12 \times 10^{-4} \pm 4.2 \times 10^{-5}$, $P < 0.001$; Fig. 4d). However, the heuristic quality in the first 20 games of the mobile app data was much lower than that in the first session of the laboratory data (0.5301 ± 0.0098 versus 0.4788 ± 0.0044, $t_{999} = 4.7$, $P < 0.001$). Thus, the users have more opportunity to improve their feature weights, whereas heuristic quality in the laboratory data might already start at ceiling.

Contextualizing planning depth magnitude

Our model best matches the choices of individual participants with a planning depth of 4 to 6, which contradicts participants' anecdotal responses as well as previous studies that found lower numbers[23,44]. We first note that these planning depth estimates do not imply that a person's plan is equally concrete for each of their next 4–6 moves. Our model contains value noise that is summed along branches of the decision tree. Effectively, the model forms a concrete plan for the first few moves, and later moves are planned more loosely.

The model also contains a sophisticated algorithm for deciding which nodes in the tree to explore, but its decision as to when to terminate this search is random. In practice, this leads the model to often continue the search without changing its final decision. By contrast, people's termination rules are more strategical and approach optimality[36]. In Supplementary Information 10, we show that, with the early-stopping rule, the model estimates lower planning depth. Although the stopping threshold cannot be identified from choice data, we find that the effect of expertise on planning depth is robust across a range of thresholds.

Planning depth or pattern recognition

Previous chess literature has framed the superior performance of experts in terms of pattern recognition[4], often operationally defined through reconstruction experiments[3,8,11]. We conducted a memory and

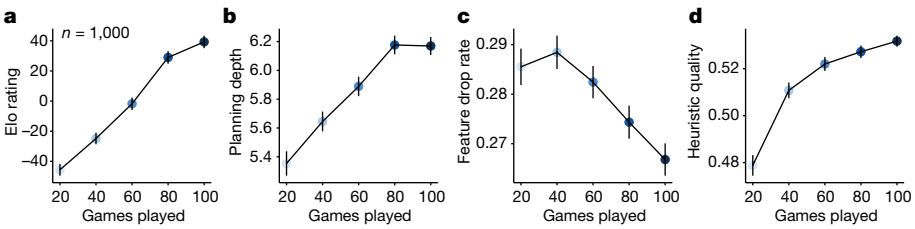


Fig. 4 | The effects of expertise on planning in mobile data. a, The average Elo rating of users of the mobile app as a function of number of games played. Data are mean ± s.e.m. across participants. **b**, The average depth to which users plan, as estimated by the behavioural model. **c**, The same as in **b**, but for the feature drop rate. **d**, The same as in **b**, but for heuristic quality.

reconstruction experiment with participants in the expertise experiment (Supplementary Information 15), which shows that experts are better at reconstructing specifically those features that our model relies on for evaluations. Parallel work[45] has replicated this result for memory and reconstruction of game sequences rather than individual positions. These data suggest a mechanistic explanation for the observed effect of expertise on planning depth: experts sharpen their representation of game-relevant features (Extended Data Table 1), allowing for more position evaluations per unit of time and therefore deeper planning. Thus, our results are consistent with improved pattern recognition in experts, but highlight the underappreciated role of processing speed.

Discussion

Regarding whether our results on the nature of expertise generalize to more complex games or natural planning tasks, we speculate that, in more complex games, expertise will also improve attention and search. For the heuristic quality effect, we note that, in the laboratory data, the participants already start with approximately correct inductive biases[46] about the relevant features and their relative values, and we observed no increase in heuristic quality with expertise. In the mobile app data, people's feature weights are initially worse and we did observe an increase. Thus, the model reveals a difference between laboratory and mobile data that was not obvious from playing strength alone.

Complex games such as chess or Go contain many non-obvious features that people can learn only through extensive experience or explicit instruction[47]. We therefore speculate that, in such games, the superior performance of experts also involves domain-specific feature knowledge. We can straightforwardly adapt our model to test this hypothesis, given a procedure to generate sophisticated candidate features. For four-in-a-row, we found a small set of simple features that enable the model to explain people's choices through manual exploration and model comparison. A promising feature-discovery approach for complex games would be to examine internal representations of neural networks trained to either play these games or predict human choices. Finally, our model applies only to deterministic two-player games; human behaviour in stochastic or multiplayer games[48,49] might involve additional computational mechanisms.

Our modelling results show how experts differ from novice players, but do not shed light on how those differences are shaped by their specific experience. A promising candidate for modelling the learning process is deep reinforcement learning, specifically algorithms such as AlphaZero[50] and SAVE[51], which combine learning from experience with forward planning at decision time. In future research, we aim to test these theories by analysing games from all 1.2 million users in the mobile dataset.

Our study opens the door to a precise understanding of human planning across development[52] and in patient populations. It also raises the question of how the components of the model are represented neurally. A specific hypothesis is that the value of future states is correlated with the activity of neurons associated with reward-based decision-making, such as those in orbitofrontal cortex[53]. Moreover, we predict that the time course of neural activity while a player contemplates their move reflects the dynamics of the value of the root node over iterations of the search algorithm.

In this Article, we introduced a two-player game of intermediate complexity that provides rich human behaviour, but for which computational cognitive modelling is still tractable. We demonstrated that a computational model based on a heuristic value function and forward search algorithm predicts human choices, response times and eye movements. Using this task and model, we showed robust evidence for increased planning and improved attention with expertise in both laboratory experiments and large-scale mobile data.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41586-023-06124-2.

1. Miller, K. J. & Venditto, S. J. C. Multi-step planning in the brain. *Curr. Opin. Behav. Sci.* **38**, 29–39 (2021).
2. Mattar, M. G. & Lengyel, M. Planning in the brain. *Neuron* **110**, 914–934 (2022).
3. de Groot, A. D. *Het Denken van den Sckaken* (Noord-Holland. Uitgev. Maatschappij, 1946).
4. Charness, N. in *Toward a General Theory of Expertise: Prospects and Limits* (eds Anders, E. K. & Smith, J.) 39–63 (Cambridge University Press, 1991).
5. Holding, D. H. Theories of chess skill. *Psychol. Res.* **54**, 10–16 (1992).
6. Gobet, F. A pattern-recognition theory of search in expert problem solving. *Think. Reasoning* **3**, 291–313 (1997).
7. Campitelli, G. & Gobet, F. Adaptive expert decision making: Skilled chess players search more and deeper. *J. Int. Comput. Games Assoc.* **27**, 209–216 (2004).
8. Linhares, A., Freitas, A. E. T., Mendes, A. & Silva, J. S. Entanglement of perception and reasoning in the combinatorial game of chess: differential errors of strategic reconstruction. *Cogn. Syst. Res.* **13**, 72–86 (2012).
9. Daw, N. D., Gershman, S. J., Seymour, B., Dayan, P. & Dolan, R. J. Model-based influences on humans' choices and striatal prediction errors. *Neuron* **69**, 1204–1215 (2011).
10. Huys, Q. J. et al. Bonsai trees in your head: how the Pavlovian system sculpts goal-directed choices by pruning decision trees. *PLoS Comput. Biol.* **8**, e1002410 (2012).
11. Chase, W. G. & Simon, H. A. Perception in chess. *Cogn. Psychol.* **4**, 55–81 (1973).
12. Van Harreveld, F., Wagenmakers, E.-J. & Van Der Maas, H. L. The effects of time pressure on chess skill: an investigation into fast and slow processes underlying expert performance. *Psychol. Res.* **71**, 591–597 (2007).
13. Sheridan, H. & Reingold, E. M. Chess players' eye movements reveal rapid recognition of complex visual patterns: evidence from a chess-related visual search task. *J. Vis.* **17**, 4 (2017).
14. Gobet, F. & Simon, H. A. Expert chess memory: revisiting the chunking hypothesis. *Memory* **6**, 225–255 (1998).
15. Bilalić, M., Langner, R., Erb, M. & Grodd, W. Mechanisms and neural basis of object and pattern recognition: a study with chess experts. *J. Exp. Psychol. Gen.* **139**, 728–742 (2010).
16. Saariluoma, P. Visuospatial and articulatory interference in chess players' information intake. *Appl. Cogn. Psychol.* **6**, 77–89 (1992).
17. Holding, D. H. *The Psychology of Chess Skill* (Lawrence Erlbaum, 1985).
18. Holding, D. H. Evaluation factors in human tree search. *Am. J. Psychol.* **102**, 103–108 (1989).
19. Gobet, F. & Jansen, P. Towards a chess program based on a model of human memory. *Adv. Comput. Chess* **7**, 35–60 (1994).
20. Holding, D. H. Counting backward during chess move choice. *Bull. Psychon. Soc.* **27**, 421–424 (1989).
21. Charness, N. in *Complex Information Processing* 203–228 (Psychology Press, 2013).
22. Huys, Q. J. et al. Interplay of approximate planning strategies. *Proc. Natl Acad. Sci. USA* **112**, 3098–3103 (2015).
23. Snider, J., Lee, D., Poizner, H. & Gepshtein, S. Prospective optimization with limited resources. *PLoS Comput. Biol.* **11**, e1004501 (2015).
24. Kolling, N., Scholl, J., Chekroud, A., Trier, H. A. & Rushworth, M. F. Prospection, perseverance, and insight in sequential behavior. *Neuron* **99**, 1069–1082 (2018).
25. Pfeiffer, B. E. & Foster, D. J. Hippocampal place-cell sequences depict future paths to remembered goals. *Nature* **497**, 74–79 (2013).
26. Redish, A. D. Vicarious trial and error. *Nat. Rev. Neurosci.* **17**, 147–159 (2016).
27. Pezzulo, G., Donnarumma, F., Maisto, D. & Stoianov, I. Planning at decision time and in the background during spatial navigation. *Curr. Opin. Behav. Sci.* **29**, 69–76 (2019).
28. Miller, K. J., Botvinick, M. M. & Brody, C. D. Dorsal hippocampus contributes to model-based planning. *Nat. Neurosci.* **20**, 1269 (2017).
29. Groman, S. M., Rich, K. M., Smith, N. J., Lee, D. & Taylor, J. R. Chronic exposure to methamphetamine disrupts reinforcement-based decision making in rats. *Neuropsychopharmacology* **43**, 770–780 (2018).
30. Akam, T. et al. The anterior cingulate cortex predicts future states to mediate model-based action selection. *Neuron* **109**, 149–163 (2020).
31. Beck, J. *Combinatorial Games: Tic-Tac-Toe Theory* Vol. 114 (Cambridge Univ. Press, 2008).
32. van Opheusden, B. & Ma, W. J. Tasks for aligning human and machine planning. *Curr. Opin. Behav. Sci.* **29**, 127–133 (2019).
33. Pearl, J. *Heuristics: Intelligent Search Strategies for Computer Problem Solving* (Addison-Wesley Longman Publishing Co., Inc., 1984).
34. Bonet, B. & Geffner, H. Planning as heuristic search. *Artif. Int.* **129**, 5–33 (2001).
35. Dechter, R. & Pearl, J. Generalized best-first search strategies and the optimality of A*. *J. ACM* **32**, 505–536 (1985).
36. Callaway, F. et al. Rational use of cognitive resources in human planning. *Nat. Hum. Behav.* **6**, 1112–1125 (2022).
37. Treisman, A. M. & Gelade, G. A feature-integration theory of attention. *Cogn. Psychol.* **12**, 97–136 (1980).
38. van Opheusden, B., Acerbi, L. & Ma, W. J. Unbiased and efficient log-likelihood estimation with inverse binomial sampling. *PLOS Comput. Biol.* **16**, e1008483 (2020).
39. Acerbi, L. & Ma, W. J. Practical Bayesian optimization for model fitting with Bayesian adaptive direct search. *Proceedings of the 31st International Conference on Neural Information Processing Systems* 1834–1844 (2017).
40. Turing, A. Computing machinery and intelligence. *Mind* **59**, 433–460 (1950).
41. Elo, A. E. *The Rating of Chessplayers, Past and Present* (Arco Pub., 1978).

42. Chabris, C. F. & Hearst, E. S. Visualization, pattern recognition, and forward search: Effects of playing speed and sight of the position on grandmaster chess errors. *Cogn. Sci.* **27**, 637–648 (2003).
43. Calderwood, R., Klein, G. A. & Crandall, B. W. Time pressure, skill, and move quality in chess. *Am. J. Psychol.* **101**, 481–493 (1988).
44. Krusche, M. J., Schulz, E., Guez, A. & Speekenbrink, M. Adaptive planning in human search. Preprint at *BioRxiv* https://doi.org/10.1101/268938 (2018).
45. Huang, J., Velarde, I., Ma, W. J. & Baldassano, C. Schema-based predictive eye movements support sequential memory encoding. *eLife* **12**, e82599 (2023).
46. Dubey, R., Agrawal, P., Pathak, D., Griffiths, T. L. & Efros, A. A. Investigating human priors for playing video games. In *Proc. Intennational Conference of Machine Learning (ICML)* (2018).
47. Charness, N., Tuffiash, M., Krampe, R., Reingold, E. & Vasyukova, E. The role of deliberate practice in chess expertise. *Appl. Cogn. Psychol.* **19**, 151–165 (2005).
48. Brown, N. & Sandholm, T. Superhuman AI for multiplayer poker. *Science* **365**, 885–890 (2019).
49. Meta Fundamental AI Research Diplomacy Team (FAIR) et al.Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science* **378**, 1067–1074 (2022).
50. Silver, D. et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science* **362**, 1140–1144 (2018).
51. Hamrick, J. B. et al. Combining q-learning and search with amortized value estimates. In *Proc. International Conference on Learning Representations (ICLR)* (2020).
52. Ma, I., Phaneuf, C., van Opheusden, B., Ma, W. J. & Hartley, C. The component processes of complex planning follow distinct developmental trajectories. Preprint at *PsyArXiv* https://doi.org/10.31234/osf.io/d62rw (2022).
53. Padoa-Schioppa, C. & Assad, J. A. Neurons in the orbitofrontal cortex encode economic value. *Nature* **441**, 223–226 (2006).

Methods

We conducted seven laboratory experiments: human-versus-human (n = 40 participants), generalization (n = 40), eye tracking (n = 10), learning (n = 30), time pressure (n = 30), a Turing test (n = 30) and a memory and reconstruction experiment (n = 38). The experiments can be played online (https://weijimalab.github.io/), except for human-versus-human and eye tracking.

Participants

We recruited participants through the NYU Psychology research participant system, flyers, a sign-up link on our laboratory webpage and personal communication. We compensated participants with US$12 per hour independent of task performance. The participants provided informed consent and our experiments were approved by the Institutional Review Board of New York University.

Human-versus-human experiment

For our human-versus-human experiment, we recruited 40 participants in pairs. For each pair, we provided consent forms and instructed participants on the task together, after which we separated them into different rooms, from which they played games against each other through an online interface. After 50 mins had expired and they finished their last game, the participants completed a post-task questionnaire, during which we provided them with compensation (US$12 in cash). Only after completing the survey and receiving compensation did the participants leave their respective rooms. Thus, the participants interacted socially before and after the experiment, but not during the games.

The participants played games against each other, switching colours after every game. After each game, we presented both participants with a pop-up showing both players' names, the current score and a button to continue to the next game. The interface proceeded only after both players had clicked the 'continue' button. Every time the participant or their opponent moved, the interface made a faint clicking noise. During the games, instead of making a move, participants could offer a draw to their opponent, which caused a pop-up prompt to appear on the other participant's screen to accept or reject the offer. If the opponent accepted the draw, the game ended immediately, otherwise the pop-up disappeared and the player who made the offer could make a move instead. We did not restrict how many draw offers participants could make (including multiple offers on the same move), but participants made relatively few draw offers. In this experiment, we never imposed any time limits.

Generalization experiment

In this experiment and all following ones, the participants performed the task individually. Each session started with the participant providing informed consent, after which we instructed them on the details of the task. We always compensated participants US$12 at the end of their session.

In the generalization experiment, the participants played against computer opponents for 30 min, after which they completed 84 trials each of a 2AFC between moves in given board positions, and 84 board-evaluation trials, in which they rated their winning chances in given board positions on a seven-point scale. Afterwards, we debriefed the participants and provided payment. The interface for the play-against-computer task was identical to the human-versus-human experiment, except for two modifications: the between-game pop-up did not display any names or score, and we removed the 'offer draw' button.

In all human-versus-computer games, the computer's algorithm is similar to the behavioural model (see the 'Detailed model specification' section) with three modifications: we used the pruning rule from the fixed branching model, and we included scale factors for weights of features belonging to the opponent (as in the opponent scaling model) and for features of different orientation (the orientation-dependent weights model) but not between active and passive feature weights (as in the no active scaling model). Finally, the algorithm used a slightly different feature set. We artificially added a thinking time to each computer move, which monotonically increased with the number of search iterations that the computer performed on each move. This ensured that the computer played faster in easy positions than in harder ones.

We created 30 computer opponents, all using the same algorithm but with different parameters. We started by fitting the behavioural model to individual participants in the human-versus-human experiment. For each parameter vector for a human participant, we created eight additional vectors by either doubling the mean tree size, halving the value noise, halving the feature drop rate or any combination thereof. We then ran an all-versus-all tournament between agents using these parameter vectors, and ranked their performance using the Elo system (see the 'Playing strength estimation using Bayeselo' section). Finally, we selected 30 agents such that their Elo ratings uniformly cover an interval ranging from slightly weaker than the worst human players to slightly stronger than the best. We divided the set of 30 agents into 6 levels with 5 agents per level, and matched the participants with computer opponents using a one-up, one-down staircase, starting at level 3. For each game, we randomly selected an opponent from the five agents on the current level.

On a 2AFC trial, we presented a participant with a board position and two candidate options, and they indicated their preference by clicking on the corresponding candidate move (Supplementary Fig. 1a). We did not impose any time limits on the participants' choices. To present participants with interesting choices and to ensure that participants' choices constrain model parameters, we selected board positions that maximized mutual information between the chosen move and model parameters, within the set of parameter vectors inferred by the model for human participants. We also computed the objective value of each move, and ensured that each trial type (both moves winning, one winning and one drawn, both drawn and so on) is represented equally (14 times). We presented the same positions to each participant, in shuffled order.

In the evaluation experiment, we presented participants with prearranged board positions and instructed them to indicate their expected winning chances by clicking one of seven buttons (Supplementary Fig. 1b). We labelled the first, middle and last button with losing, equal and winning, respectively. For the evaluation experiment, we selected positions using the same procedure as in the 2AFC experiment, except that, in the final selection stage, we ensured that the game-theoretic values (Supplementary Information 2.4) of the presented positions were equally distributed across winning, losing or drawn (28 each).

Turing test experiment

The Turing test experiment consisted of two sessions on consecutive days. On the first session, the participants played against computer opponents for 60 min. In this experiment, each computer opponent followed the main model with parameters inferred for an individual participant in the human-versus-human experiment.

On the second session, the participants performed 180 trials of a classification task. On each trial, we presented participants with a video of a segment of a game played either by two players in the human-versus-human experiment, or two computers following the main model with parameters inferred for those players. The participants could start the video at any time by pressing a 'play' button, and the video played at a constant speed of 1.8 seconds per move. After the video, the participants judged the video using a slider labelled "Certainly computers" on the left, "No clue" in the middle and "Certainly humans" on the right. After each trial, we provided the participants with feedback on whether their classification judgement was correct ("Correct!") or not ("Incorrect").

We selected game segments to use for human-versus-human videos from games played in human-versus-human experiment. For each game, we sampled the starting number from a geometric distribution with rate 0.15, and discarded everything up to that move number. We then drew a maximum length for the segment from another geometric distribution with rate 0.1, and added moves from the game until the segment exceeded that maximum length or until the end of the game. For each game, we also generated a computer-versus-computer segment using a similar sampling method. We started from the same position, and added moves from a simulated computer-versus-computer game until the segment reached the same maximum length or the game ended. Thus, all computer-versus-computer video segments start from a position that occurred in a human-versus-human game, but all moves are made by the computational cognitive model. Owing to this sampling method and the constant playback speed of the videos, the only cues available to participants are the moves played and not the starting position or response times. Finally, we selected a random subset of 90 games to use for human-versus-human videos and 90 others for computer-versus-computer videos.

To instruct the participants on the task, we used the following text: "Today, you will be shown 180 short videos, either from games between two human players or between two computers. Half of the videos are from games between humans, the other half between computers. The videos may start from any point in a game, so the starting position is not necessarily an empty board. Your task is to identify if the video is from a human-vs-human game or a computer-vs-computer game. You will also be asked to report how confident you are about your choice. There is no time limit to this task."

Eye tracking experiment

In the eye tracking experiment, the participants played against computer opponents for 40 min and performed 84 trials of the 2AFC experiment, with settings for both experiments identical to the generalization experiment above. For the entire experiment, we recorded the eye movements of the participants using a remote infrared video-oculographic system (EyeLink 1000; SR Research[54]) with a 1 kHz sampling rate and around a 0.01° precision. We acquired eye position data with the EyeLink software using the 'Heuristic filter ON' option. We displayed stimuli on a 21 in Sony GDMF520 CRT monitor (resolution: 1,280 × 960 px; refresh rate, 100 Hz). The participants used a headrest located approximately 57 cm from the screen. We configured the eye tracker to record events only, so that our dataset consists of a time series of fixations, saccades and blinks.

For each session, we first calibrated the eye tracker with the built-in nine-point calibration method, but we also added a calibration condition directly before and after the play-against-computer component of our experiment. In this calibration procedure, we presented an empty board with a white piece with a fixation cross on top of it on the bottom left square (Supplementary Fig. 1d). We instructed the participants to fixate on the cross and press the space bar when they believed that their fixation was steady. After they pressed the space bar, the piece and the cross moved one square to the right, instructing the participant to fixate on the next square, which they again indicated with a space bar press. We continued moving the cross accordingly across all 4 rows, obtaining 9 fixation coordinates and time stamps for the space bar presses at each square in each row.

Learning experiment

The learning experiment consisted of five sessions. We required the participants to schedule consecutive sessions no more than 2 days apart. On the first, third and fifth session, the participants played against computer opponents for 30 min and completed 60 trials each of the 2AFC and evaluation conditions. On the second and fourth sessions, the participants played against computer opponents for the entire 60 min session. In all these sessions, the computer opponents were identical to those of the generalization experiment and the positions were selected using the same information criteria, with one difference. We selected 180 positions that we divided into 3 groups of 60, and ensured that the order of the days on which we presented these positions was counterbalanced across participants. We compensated participants US$12 per session, with a US$12 completion bonus at the end.

Time-pressure experiment

In the time-pressure experiment, the participants played against computer opponents for 50 min, again with the identical procedure as in the generalization experiment. However, in each game, both the human participant and the computer opponent had to obey a time limit of 5, 10 or 20 s per move. The time constraint was constant within each game and varied randomly between games. If a participant exceeded this time limit, the game ended immediately and counted as a loss. We also amended the thinking time for the computer, to ensure that it never used more than 80% of its allotted time. However, we emphasize that this did not change the computer opponent's decisions, as those take only a fraction of a second to compute. We control the opponent's thinking time by simply pausing the interface for the appropriate amount of time.

To inform the participants of the time constraint, we indicated the time limit for each game in a pop-up before the start of that game. Furthermore, directly to the right of the board, we displayed a timer—a coloured bar that shrunk gradually while participants were contemplating their move. Directly below the timer, we displayed a text-based count-down with the remaining thinking time in seconds (Supplementary Fig. 1c). In the 20 s condition, at the start of each move, the colour bar was equally high as the board and linearly decreased to zero in 20 s. Initially, the bar was green, but when the participant had 10 s left, it changed colour to blue and, 5 s before the end, it changed colour to red. To warn the participants even more of the passage of time, we played three warning sounds (short beeps) when the participant had 2, 1 or 0 s left, with an increasingly higher pitch each time. In the 5 or 10 s condition, we started the timer in the same state that it would be in the 20 s condition after 10 or 15 s had elapsed (10 s, blue colour bar, half as high as the board; 5 s, red bar, quarter board height). When the computer was 'thinking', we displayed a timer to the left of the board, with the identical behaviour. The time warnings were largely effective, and the participants lost on time in only 1.87% (33 out of 1,766) of their games.

Large-scale mobile data

In collaboration with the mobile app company Peak (https://www.peak.net), we collected a dataset of people playing four-in-a-row. When signing up for the app, users consented to a privacy policy, which included a provision that aggregated and anonymized data might be shared with third parties such as universities. The Institutional Review Board of New York University determined that no further consent was required and approved the research protocol as 'exempt'.

We collected 10,874,547 games from 1,234,844 unique users. Users always play first, and the game board is oriented vertically and visually enriched (Supplementary Fig. 1e). Moreover, users play at will against a computer opponent implementing a version of our main model, with the parameters adapted from fits on data collected in the laboratory experiments (human-versus-human, generalization, eye tracking, learning and time pressure). The procedure for generating the computer opponents is identical to the one in the generalization, learning and time-pressure experiments, but we recalibrated the computer opponents as they always play second. We created seven classes of computer opponents of varying strength, and matched users with an opponent based on their track record of game results. For analysis, we randomly selected 1,000 participants from this dataset that had each played at least 100 games. We grouped their experience into 5 blocks of 20 games to approximately match the total experience level of participants in the learning experiment.

Memory and reconstruction experiment

In this experiment, the participants memorized and reconstructed board positions. We recruited 2 groups of 19 participants. The first group consisted of participants who had previously completed the learning experiment, no more than 4 weeks before. The second group had no previous experience with the game and were informed that the task involved memorizing patterns of squares and circles.

On each trial, we presented the participants with board positions for 10 s followed by a blank board for 1 s. We then prompted them to reconstruct the original position without a time limit. The reconstruction interface allowed the participants to right-click on any square to place or remove a black piece and to left-click to place or remove a white piece. At any time, the participants could click a 'submit' button to indicate that they had finished their reconstruction, after which they received feedback indicating the fraction of the 36 squares correctly reconstructed, including empty squares.

Each participant reconstructed the same set of 96 positions in a random order, in an approximately 1 h session. We generated 2 sets of 48 positions. The first set contained positions from human-versus-human games. To generate this set, we varied the number of pieces in each position from 11 to 18, and randomly selected 6 positions from human-versus-human games with that number of pieces. The second set consisted of procedurally generated positions, constrained to exactly match the distribution of the number of pieces, and approximately match the marginal distribution of occupied squares.

Analysis methods

Playing strength estimation using Bayeselo. To estimate a player's playing strength from games against computer opponents, we use Elo ratings[55,41], implemented using the publicly available program Bayeselo[56]. To measure Elo ratings of all players in all experiments against a common baseline, we run Bayeselo on a database containing all human-versus-computer games and a simulated computer-versus-computer tournament, in which each computer plays once against every other computer, including itself. In the computer-versus-computer tournament, we include all agents used in the generalization, learning and time-pressure experiment as well as the agents used in the mobile app.

Model specification. We assume that people's choices on each move are independent and generated by the same decision-making process with the same parameters within a single session. We first describe the model broadly, then in more detail. Our model is based on heuristic search [57], and consists of a value function and a tree search algorithm. Furthermore, we include sources of noise to capture variability in human play and human-like mistakes.

Value function. The core of our model is a value function $V(s, \mathbf{w})$, which assigns a value to a board state s. The higher this value, the more likely the black player is to win from that state. We assume that people use value function approximation[58], and that people's value function is a weighted sum of features

$$V(s, \mathbf{w}) = \sum_{i=1}^{5} w_i \phi_i(s, \text{self}) - \sum_{i=1}^{5} w_i \phi_i(s, \text{opponent}) \quad (1)$$

where ϕ_i denotes the features and w_i the weights. In the following, and in the main text, we omit the dependence of $V(s, \mathbf{w})$ on $\mathbf{w}$ for brevity. The value function uses five features: centre, connected two-in-a-row, unconnected two-in-a-row, three-in-a-row and four-in-a-row. The centre feature assigns a higher value to squares near the centre of the board. The other features count how often their corresponding patterns occur on the board (horizontally, vertically or diagonally).

Whenever the model evaluates a state, the weights of features belonging to the active player are multiplied by a scaling constant C. This captures value differences between active and passive features. For example, a three-in-a-row feature signals an immediate win on the active player's move but not the opponent's (it can be blocked). We do not scale the centre feature.

Tree search. The value function guides the construction of a decision tree with an iterative best-first search algorithm[35]. Each iteration, the algorithm chooses a board position to explore further, evaluates the positions resulting from each legal move and prunes all moves with value below that of the best move minus a threshold. After each iteration, the algorithm stops with a probability γ, resulting in a geometric distribution over the total number of iterations.

Noise. To account for variability in people's choices, we add three sources of noise. We model selective attention by randomly dropping features (at specific locations and orientations) before constructing the decision tree, which are then omitted during the calculation of $V(s)$ anywhere in the tree. During the tree search, we add Gaussian noise to $V(s)$ in each node. Finally, we include a lapse rate λ.

Detailed model specification. Value function. The value function consists of two terms, the first of which measures whose pieces are closer to the board centre:

$$V_{\text{centre}}(s) = \sum_{\mathbf{x} \in \text{Pieces(s,black)}} \frac{1}{\|\mathbf{x} - \mathbf{x}_{\text{centre}}\|} - \sum_{\mathbf{x} \in \text{Pieces(s,white)}} \frac{1}{\|\mathbf{x} - \mathbf{x}_{\text{centre}}\|} \quad (2)$$

where $\text{Pieces}(s, p)$ enumerates the locations of all pieces that player p owns, $\mathbf{x}_{\text{center}}$ denotes the coordinate of the board centre, and $\|\cdot\|$ is the Euclidean distance.

The second term counts how often particular patterns occur on the board (horizontally, vertically or diagonally). A feature is a binary function $f_{t,x,y,o}(s)$ that returns 1 if a pattern of type t occurs at location (x, y) with orientation o, and 0 otherwise. We use the following four patterns. (1) Connected two-in-a-row: two adjacent pieces with enough empty squares around them to complete four-in-a-row. (2) Unconnected two-in-a-row: two non-adjacent pieces that lie on a line of four contiguous squares, with the remaining two squares empty. (3) Three-in-a-row: three pieces that lie on a line of four contiguous squares, with the remaining square empty. This pattern represents an immediate winning threat. (4) Four-in-a-row: four pieces in a row. This pattern appears only in board states where a player has already won the game.

We define F to be the set of all such features (one for each type, orientation and board location), and associate a weight w to each feature in this set. The feature weight depends only on its type, and not on the orientation or location. Finally, we write the value function as:

$$\begin{aligned} V_F(s) = w_{\text{centre}} V_{\text{centre}}(s) + c_{\text{black}} \sum_{i \in F} w_i f_i(s, \text{black}) \\ - c_{\text{white}} \sum_{i \in F} w_i f_i(s, \text{white}) + \mathcal{N}(0, 1) \end{aligned} \quad (3)$$

where $c_{\text{black}} = C$ and $c_{\text{white}} = 1$ whenever black is to move in state s, and $c_{\text{black}} = 1$ and $c_{\text{white}} = C$ when it is white's move. The final term $\mathcal{N}(0, 1)$ represents additive Gaussian noise with mean zero and unit variance.

Search algorithm. The search algorithm constructs a decision tree, consisting of nodes that contain a state s, the colour of the active player in that state and a value associated to the state. The algorithm initializes the value of each new node by calling the feature-based evaluation function $V(s)$. However, this value changes as the algorithm investigates the consequences of future play from that state. The algorithm starts with a single-node decision tree and gradually grows the tree. Each iteration, the algorithm selects a leaf node, expands it by adding one child node each for a number of candidate moves and backpropagates the value of these new nodes recursively into the leaf node as well as its parents. Specifically, to make a move in a given position, the model (Supplementary Algorithm 1) follows the following steps. (1) Decide whether to lapse, with probability λ. If it does lapse, the model makes

a random move. (2) Randomly drop features from the value function, each instance independently with probability δ. (3) Iteratively select a node, expand it and backpropagate the resulting value (see below for details of these three functions) until the value of the root node (winning, losing or drawn) has been determined with certainty. Moreover, after each iteration, the algorithm has a stopping probability γ to terminate. (4) Finally, make the move that maximizes value in the root node: $\underset{c\in\text{children(root)}}{\arg\max}\ c.\text{val}$.

To select a node, we use best-first search (Supplementary Algorithm 2). This search procedure relies on the principal variation, the sequence in which both players always make the best moves according to the currently estimated values starting from the root to a leaf node. The model selects this leaf node for expansion. As the value of nodes in the tree change after each iteration, so does the principal variation, and the search algorithm therefore dynamically switches between different branches of the tree.

To expand the selected node, the model adds one child node for each legal move in the associated state (Supplementary Algorithm 3). As it initializes the children, it automatically evaluates their states using $V(s)$ as defined above. The algorithm does not yet check whether either of these states is terminal (that is, either player has achieved four-in-a-row or the board is full), but it effectively does so if $w_{\text{four-in-a-row}}$ is high enough. Next, the algorithm prunes unpromising children; those of which the value difference with the best candidate move exceeds a threshold θ. Only afterwards does it assign $V = 10{,}000$ to each child state in which black has won, $V = -10{,}000$ if white has won and $V = 0$ for draws. It is therefore possible that, if $w_{\text{four-in-a-row}}$ is too low, or if the algorithm has dropped a four-in-a-row feature in a relevant location, it will prune away an immediately winning move, which can result in bad (but human-like) blunders.

To backpropagate, the search algorithm incorporates the value of the newly created nodes into the decision tree using the minimax rule, which sets each node's value to the maximum of its children's values if black is to move or minimum if white is to move. It achieves this efficiently by updating only the nodes on the principal variation, in backwards order (Supplementary Algorithm 4). Thus, after backpropagation, the value of each state reflects the search algorithm's best estimate of the result of a game starting in that state with perfect play from both sides.

The search algorithm continues to run until the root node is determined, or the random stopping occurs. If the value of the root node is never determined with certainty, the stopping probability is constant and independently drawn each iteration, and the total number of iterations is geometrically distributed, with parameter γ. When implementing our model as an AI algorithm to play against human opponents, we convert the number of iterations N into a 'thinking time' for the AI by $t = a\sqrt{N\gamma} + b$, where $a = 4$ s and $b = 0.5$ s.

The main model has 10 parameters: the pruning threshold θ, the stopping probability γ, the lapse rate λ, the feature drop rate δ, the active scaling constant C and the feature weights w_{centre}, $w_{\text{connected-two-in-a-row}}$, $w_{\text{unconnected-two-in-a-row}}$, $w_{\text{three-in-a-row}}$ and $w_{\text{four-in-a-row}}$. We do not add a parameter for the variance of the value noise, as changing the noise distribution from $\mathcal{N}(0, 1)$ to $\mathcal{N}(0, \sigma^2)$ has the same effect as changing $\theta \to \frac{\theta}{\sigma}$ and $w \to \frac{w}{\sigma}$ for each feature. Thus, adding σ would over-parametrize the model and cause σ, θ and $\{w_i\}$ to be unidentifiable from data.

Computing game-theoretic values. We can use the model to calculate the game-theoretic value $\tilde{V}(s)$ of a position s, that is, the outcome of a game starting from position s with perfect play from both sides. To compute the game-theoretic value, we execute the best-first search algorithm with default feature weights, no sources of noise and no pruning. In the limit of infinitely many iterations, the value of the root node in the decision tree of best-first search is guaranteed to converge to the game-theoretic value. In practice we found that 200,000 search iterations was sufficient for almost all positions. For positions in which 200,000 iterations did not yield a determined result, we set the game-theoretical value to $\tilde{V}(s) = 0$, in other words, a draw.

Alternative model specifications. Lesions. Our first set of alternative models are lesion models, obtained by removing components from the main model. Each lesion can be implemented by fixing a parameter to a constant. The no centre, no connected two-in-a-row, no unconnected two-in-a-row, no three-in-a-row and no four-in-a-row models are obtained by setting the respective feature weight to zero. The no feature drop model is obtained by fixing δ to zero, and the no active scaling model results from fixing C to 1. To obtain the no pruning model, we fix θ to 20,000, which is larger than any value difference that occurs in search and causes the model to never prune. Note that the model cannot compensate by increasing feature weights as their order of magnitude is yoked by fixing the value noise to have unit variance. Finally, the no tree model is achieved by fixing γ to 1. This causes the algorithm to stop after 1 iteration, in which case it will have expanded only the root node, and its choice will be the highest-value child. Pruning lower-value children does not affect this choice, so θ is not a parameter in this model.
Modifications. In our first modified model, fixed iterations, we change the stopping criterion to occur at a fixed iteration number N. In the fixed depth model, we amend the search process to explore every branch up to a fixed depth D. In the fixed branching model, we amend the pruning rule to keep the K highest-value children in each node (lowest value when white is to move). If the expanded node has less than K children, the algorithm prunes nothing. Next, we consider removing the feature drop mechanism and instead applying a function in which each child is pruned with a probability ε while expanding a node before the value-based pruning, resulting in the square dropping model. For the optimal weights model, we restrict the feature weights $\{w_i\}$ to a constant vector, which we chose by maximizing the Pearson correlation between $\tanh(V(s)/20)$ and the game-theoretic value $\tilde{V}(s)$ across all states s that occurred in the human-versus-human experiment (Supplementary Information 2.4).

Finally, we consider Monte Carlo tree search (MCTS). In this algorithm, instead of evaluating a state with $V(s)$, we perform a rollout—a simulated game starting from state s between two agents that follow a myopic policy. That is, in state s', the agent chooses the move m that maximizes $V(s' + m)$, or the one that minimizes it when white is to move. We then assign a value of 1 to state s if the rollout results in a win for black, 0 for white wins, and 1/2 if the game is a draw. Note that, as the evaluation function contains noise, the myopic policy and the outcome of the rollout are also stochastic. Note also that we perform only a single rollout when evaluating a state.

After performing a rollout, MCTS backpropagates by averaging rather than minimax, ensuring that the value of each intermediate node of the tree is equal to the average outcome of the rollouts conducted in all descendants of that node. We also amend the best-first selection rule

$$m = \underset{c\in\text{children(root)}}{\arg\max}\ c.\text{val} \tag{4}$$

to the UCB formula

$$m = \underset{c\in\text{children(root)}}{\arg\max}\ c.\text{val} + C_{\text{exp}} \times \sqrt{\frac{\log(n.N_{\text{rollouts}})}{c.N_{\text{rollouts}}}} \tag{5}$$

where $n.N_{\text{rollouts}}$ counts the number of rollouts that have been conducted in node n or any of its descendants, and C_{exp} is a parameter that controls the balance between exploitation (investigating high-value children) and exploration (investigating children that have not been investigated much). Finally, after the tree search terminates, the algorithm makes a move by maximizing N_{rollouts} across all children of the root node.
Extensions. We create the orientation-dependent weights by multiplying the weight of vertically or diagonally oriented features by scaling

constants c_{vert} and c_{diag}, respectively. For the orientation-dependent dropping model, we allow the feature drop rate for horizontally, vertically or diagonally oriented features to vary, whereas, in the type-dependent dropping, we let the drop rate depend on the feature type. In the triangle model, we include a feature that counts the number of times that any of a set of three-piece patterns occurs on the board. Finally, the opponent scaling model extends the main model by adding a scaling constant c_{opp} that multiplies weights of features belonging to the opponent. Note that opponent scaling and active scaling are dissociated as the former multiplies weights of the opponent's features regardless of whose move it is, whereas the latter is adaptive.

Model fitting. The main model has 10 parameters: the 5 feature weights, the active-passive scaling constant C, the pruning threshold θ, stopping probability γ, feature drop rate δ and the lapse rate λ. We infer these parameters for individual participants and individual learning sessions or time limit conditions with maximum-likelihood estimation. Unfortunately, deriving the log-likelihood analytically requires marginalization of all latent variables (which features are dropped, the number of iterations in the search algorithm and the value noise at each node), which is intractable, restricting ourselves to only models with analytical likelihoods would limit the types of models that one can consider, particularly in regards to the noise structure. Instead, we estimate the log-likelihood with inverse binomial sampling[38,59,60], a method that estimates the log-likelihood by comparing the data to simulated data generated from the model. Inverse binomial sampling is unbiased but its estimates are noisy. Moreover, we cannot calculate gradients of the log-likelihood, so we optimize the log-likelihood with multilevel coordinate search [61], a gradient-free algorithm. To reduce overfitting, we compare models using fivefold cross-validation.

This pipeline is computationally expensive, and fitting one participant's data for a single model requires approximately 10^{14} floating-point operations. We perform the model fits on the NYU high-performance cluster (Intel Xeon E5-2690v2 CPUs 3.0 GHz) with a parallel implementation of inverse binomial sampling, which uses 20 cores. On our hardware, fitting takes approximately 1 h for one participant and one model.

Derived metrics. To analyse the nature of expertise and the effect of time pressure, we convert the set of ten parameters from the main model to three derived metrics: planning depth, feature drop rate and heuristic quality.

We define the planning depth as the length of the principal variation in the model's decision tree, averaged across simulations of the model with a given a parameter vector in a fixed set of probe positions, specifically, all positions that occurred in the human-versus-human experiment (5,482 positions). As in Supplementary Algorithm 4, the principal variation is the sequence in which both players make the best move according to the values in the decision tree, from the root to a leaf. The length of this sequence is equal to the depth of that leaf node, and reflects how far into the future the model plans. We average this depth across ten simulated moves, and across all probe positions.

The feature drop rate is simply the parameter δ. To define the heuristic quality, we evaluate $V(s, \mathbf{w})$ in all of the probe positions, and compute the Pearson correlation between $\tanh(V(s, \mathbf{w})/20)$ and the game-theoretic value $\widetilde{V}(s)$ (Supplementary Information 2.4). Note that the heuristic quality depends only on the feature weights $\mathbf{w}$ and the active scaling constant C.

As the probe positions are fixed in the definition of planning depth, it is purely a function of the model parameters. Planning depth depends primarily on the stopping probability (Spearman correlation: $\rho = -0.87$, $P < 0.001$), and there is a minor dependence on the pruning threshold ($\rho = -0.21$, $P < 0.001$). These correlations are computed across a range of parameter vectors taken from model fits to human data. The heuristic quality is a more complicated function of the feature weights and active scaling constant. For example, the heuristic quality correlates with $w_{three\text{-}in\text{-}a\text{-}row}/w_{connected\text{-}two\text{-}in\text{-}a\text{-}row}$, but the correlation is relatively weak ($\rho = 0.55$, $P < 0.001$), and other feature weights influence the heuristic quality too.

In other words, the derived metrics carve up the set of ten parameters: planning depth primarily depends on pruning threshold and stopping probability, feature drop rate depends on the feature drop rate and heuristic quality solely depends on feature weights. Together, the three metrics provide a reduced representation of the model parameters that is more interpretable, more reliably inferred (Extended Data Fig. 2) and sufficient to capture the increase in performance across sessions (Supplementary Fig. 9).

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

Data supporting the findings of this study are publicly available at the Open Science Framework (https://osf.io/n2xjm/).

Code availability

Code used in this study is publicly available at the Open Science Framework (https://osf.io/n2xjm/).

54. Cornelissen, F. W., Peters, E. M. & Palmer, J. The eyelink toolbox: eye tracking with MATLAB and the psychophysics toolbox. *Behav. Res. Methods Instr. Comput.* **34**, 613–617 (2002).
55. Zermelo, E. Die berechnung der turnier-ergebnisse als ein maximumproblem der wahrscheinlichkeitsrechnung. *Math. Z.* **29**, 436–460 (1929).
56. Hunter, D. R. MM algorithms for generalized Bradley-Terry models. *Ann. Stat.* **32**, 384–406 (2004).
57. Sutton, R. S. & Barto, A. G. *Reinforcement Learning: An Introduction* Vol. 1 (MIT Press, 1998).
58. Sutton, R. S., McAllester, D. A., Singh, S. P. & Mansour, Y. in *Advances in Neural Information Processing Systems* 1057–1063 (2000).
59. Dawson, R. *Unbiased Tests, Unbiased Estimators, and Randomized Similar Regions*. PhD thesis, Harvard Univ. (1953).
60. de Groot, M. H. Unbiased sequential estimation for binomial populations. *Ann. Math. Stat.* **30**, 80–101 (1959).
61. Huyer, W. & Neumaier, A. Global optimization by multilevel coordinate search. *J. Glob. Optim.* **14**, 331–355 (1999).

Acknowledgements We thank Z. Shu for piloting an early version of the experiment; F. Khalidi for assistance with data collection; and A. Mihali, A. Yoo, M. Honig, L. Acerbi, W. Adler, F. Callaway, T. Griffiths and M. Mattar, and the other current members and alumni of the Ma laboratory for discussions. This work was supported by grant number IIS-1344256 to W.J.M. and by Graduate Research Fellowship number DGE1839302 to I.K. from the National Science Foundation.

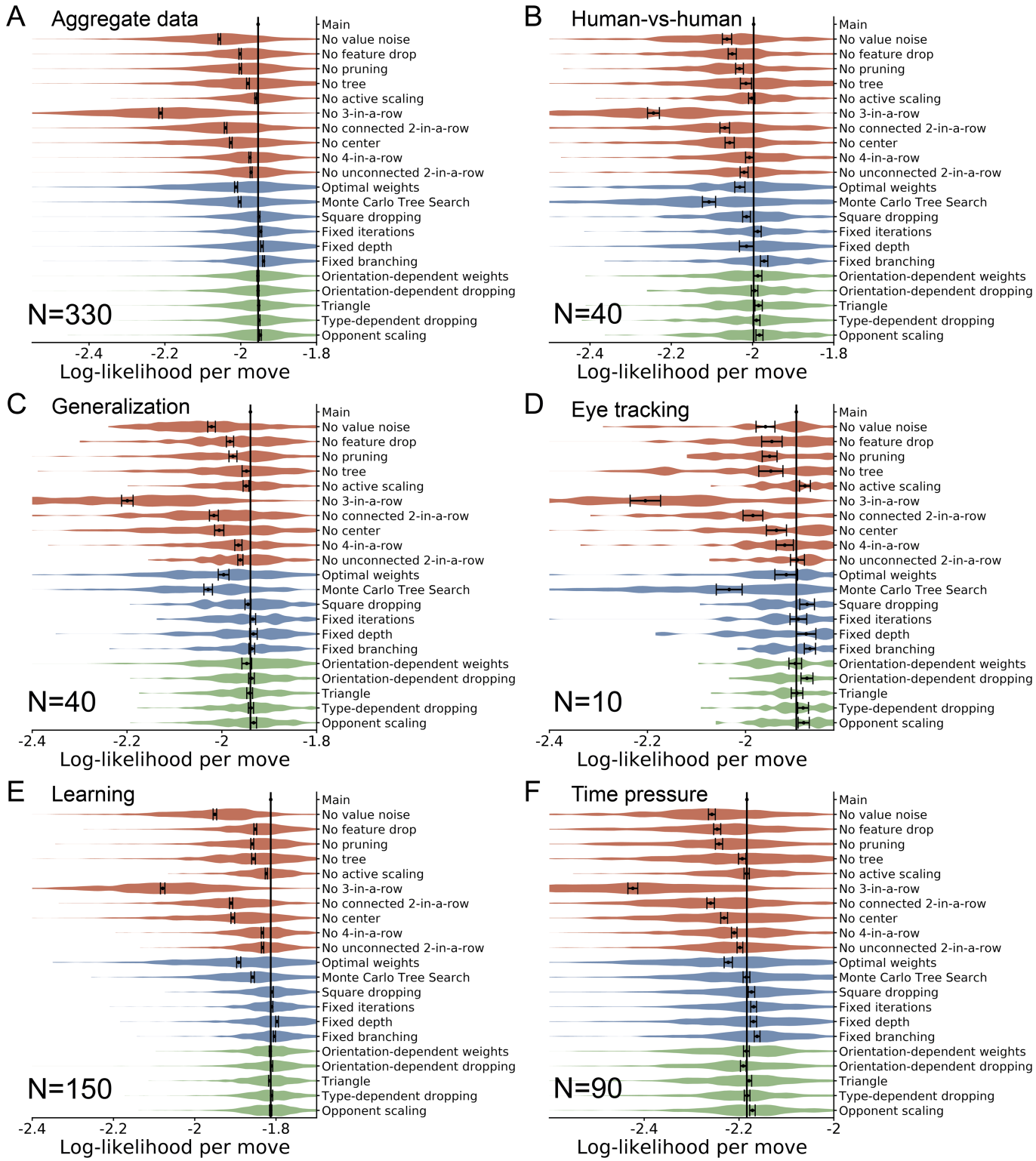
Author contributions All of the authors contributed to conceptualization of the research. B.v.O., G.G., I.K. and Y.L. collected data. B.v.O., I.K., G.G., Y.L. and Z.B. developed software, methodology and performed analysis. B.v.O., I.K. and W.J.M. wrote the paper. W.J.M. supervised the project and acquired funding.

Competing interests The authors declare no competing interests.

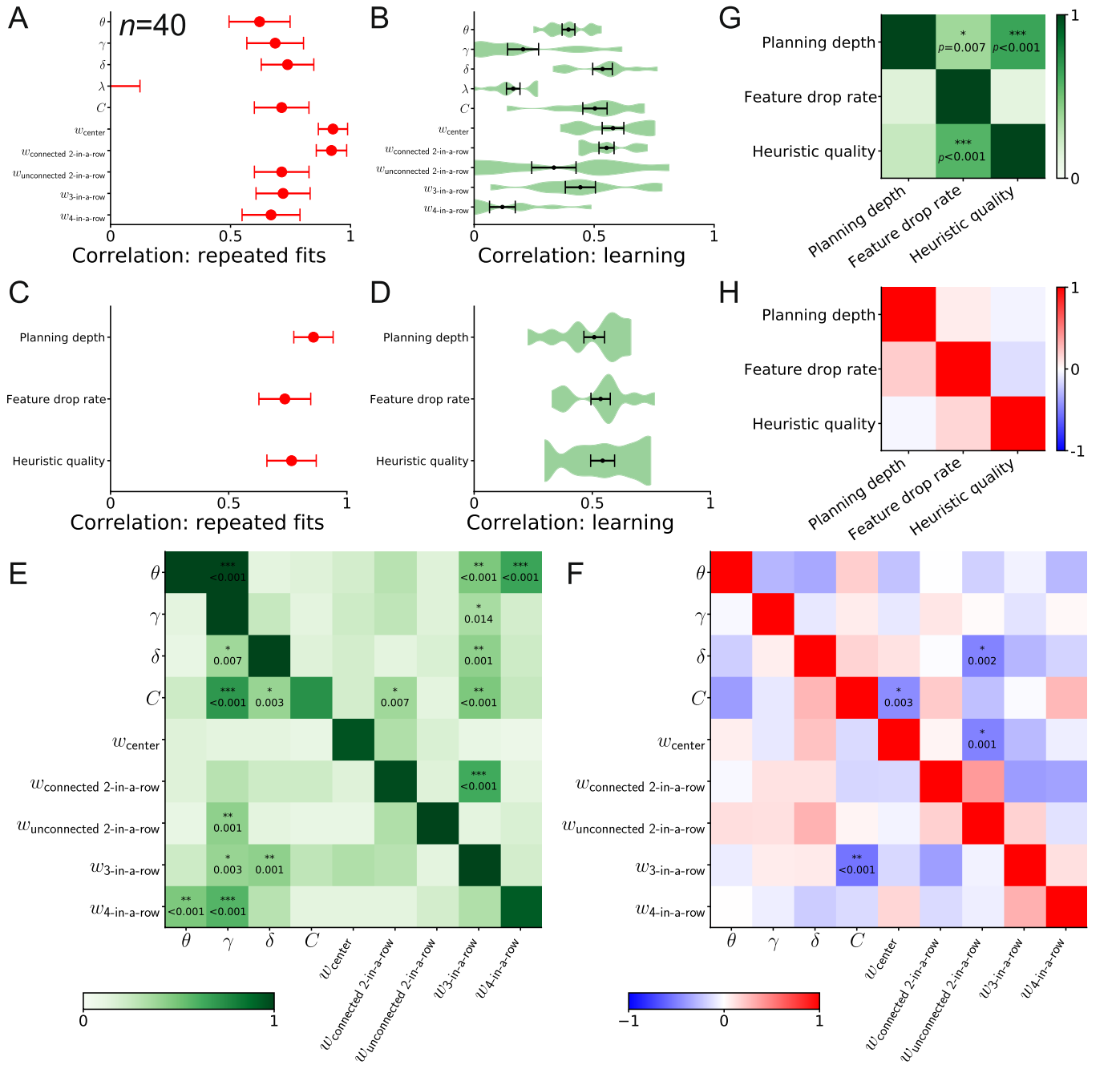
Additional information
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41586-023-06124-2.

Correspondence and requests for materials should be addressed to Bas van Opheusden.

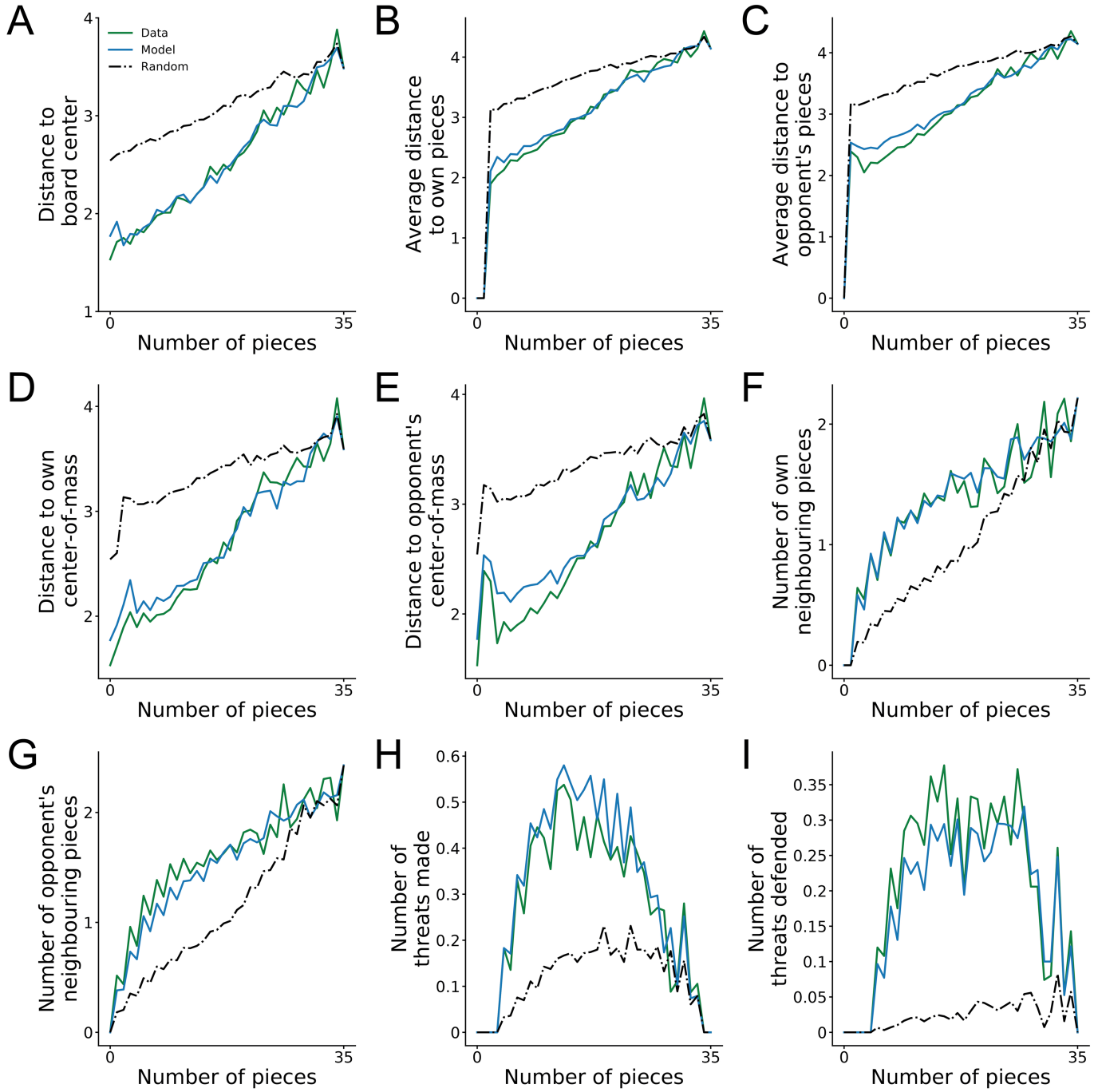
Peer review information *Nature* thanks Quentin Huys and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.



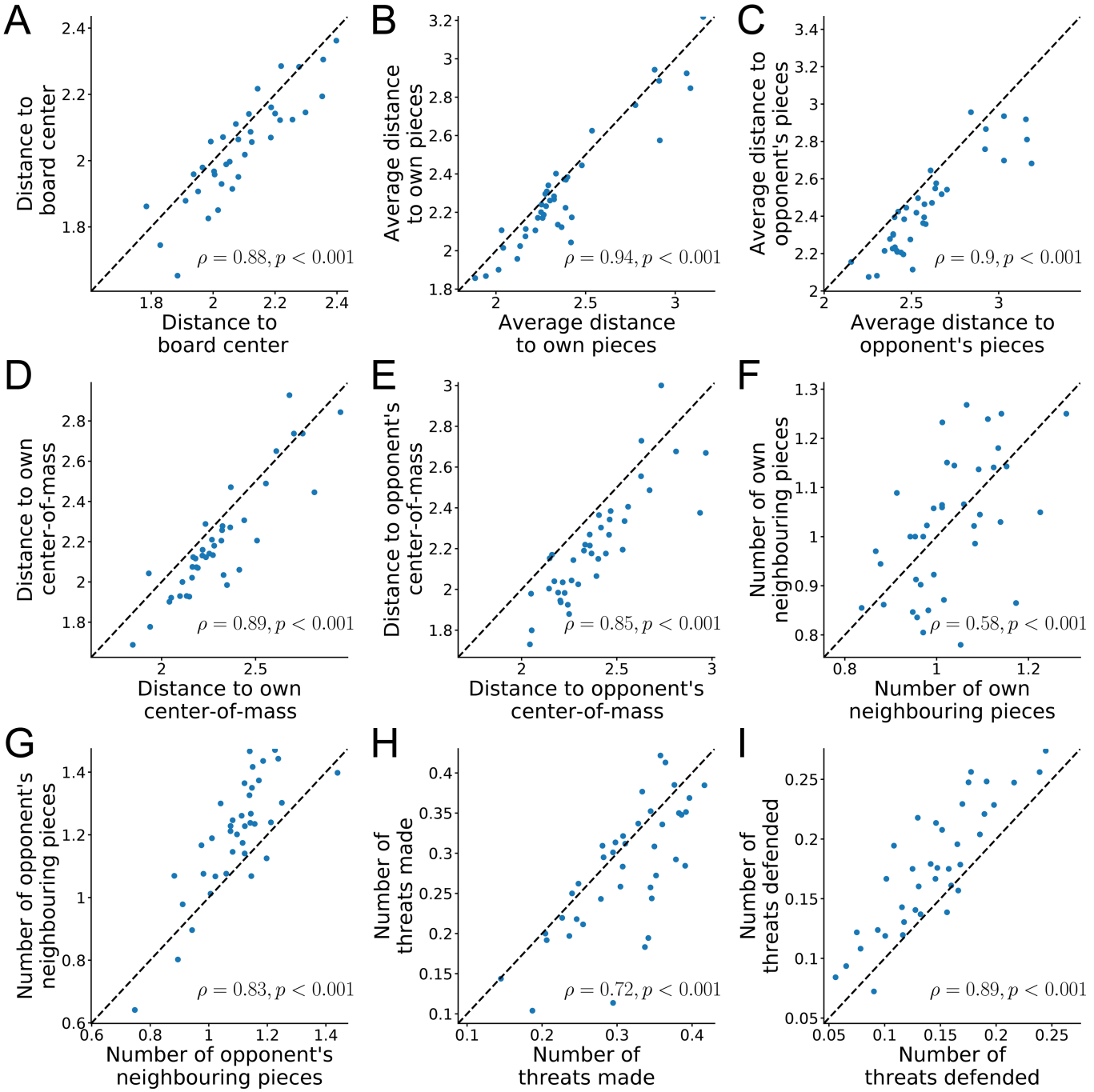
Extended Data Fig. 1 | Model comparison. We validate our main model specification by comparing to alternatives in three categories: lesions generated by removing model components (red), extensions generated by adding new model components (blue) and modifications generated by replacing a model component with a similar implementation (green). **A**. Cross-validated log-likelihood per move, across all participants in the laboratory experiments. Error bars indicate mean and s.e.m. of the difference in log-likelihood with the main model. **B**–**F**. Same as **A**., for participants in the human-vs-human, generalization, eye tracking, learning and time pressure experiments.



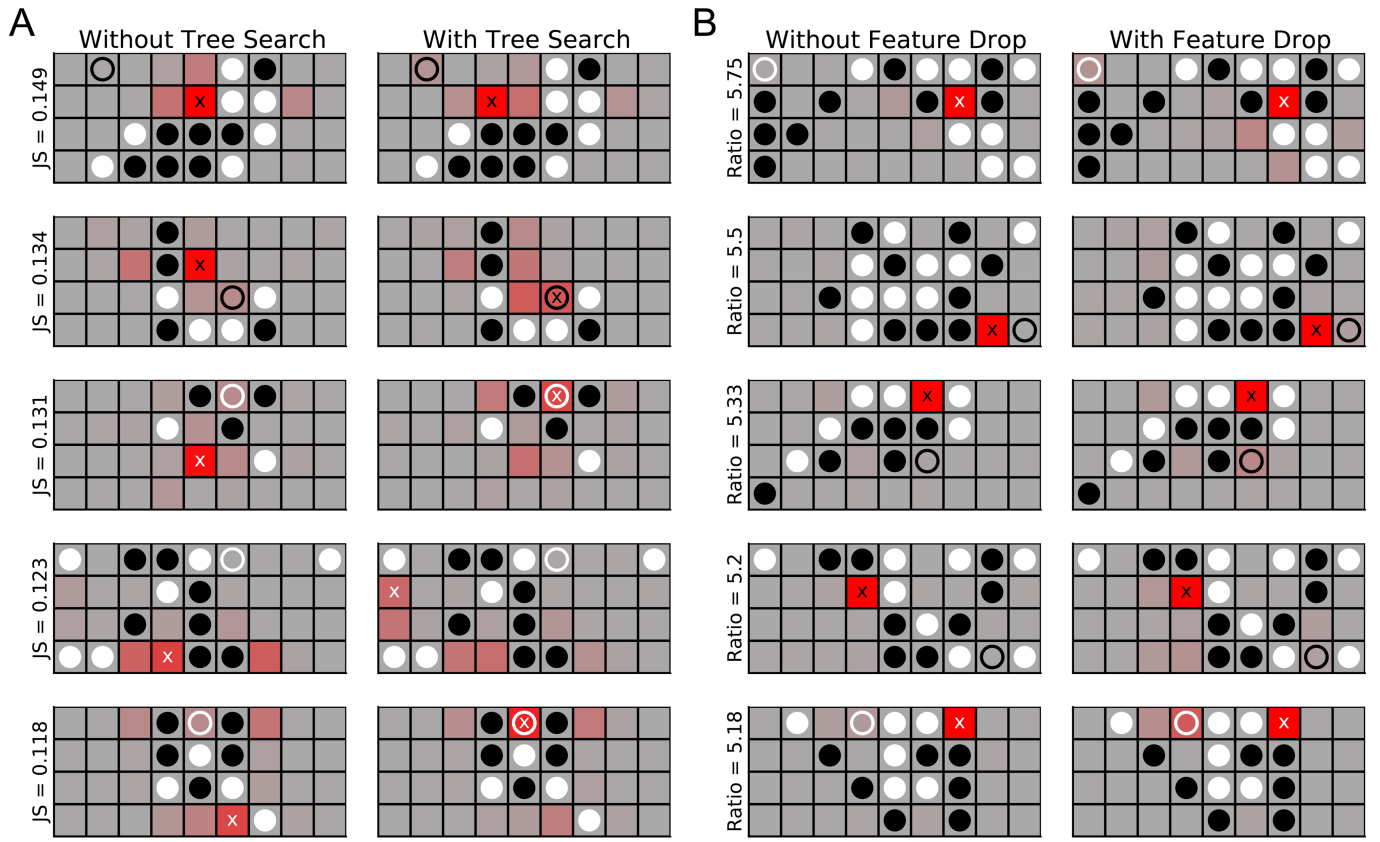
Extended Data Fig. 2 | Parameter validation. Because model fitting is too computationally expense for parameter recovery, we assess the reliability of the parameter estimates using less computationally expensive methods. **A**. Pearson correlation across participants between model parameters estimated in two independent fits. Error bars indicate the confidence interval. **B**. Same as **A**., for different sessions in the learning experiment. Error bars indicate s.e.m. across participants **C-D**. Same as **A-B**., for the derived metrics. **E**. 2-sample Kolmogorov-Smirnov test statistic between the distribution of $\hat{\theta}_j^{\text{lesion } i}$ and $\hat{\theta}_j^{\text{full}}$ for each pair of parameters. In all panels, we indicate tests that are significant after correcting for multiple comparisons using false discovery rate by *: $\alpha = 0.05$, **: $\alpha = 0.01$, ***: $\alpha = 0.001$. For significant tests, we additionally report uncorrected two-sided p-values. **F**. Trade-offs between model parameters using a Pearson correlation between $\hat{\theta}_i^{\text{full}}$ and $\hat{\theta}_j^{\text{full}} - \hat{\theta}_j^{\text{lesion } i}$ for each pair of model parameters. **G-H**. Same as **E-F**., for the derived metrics.



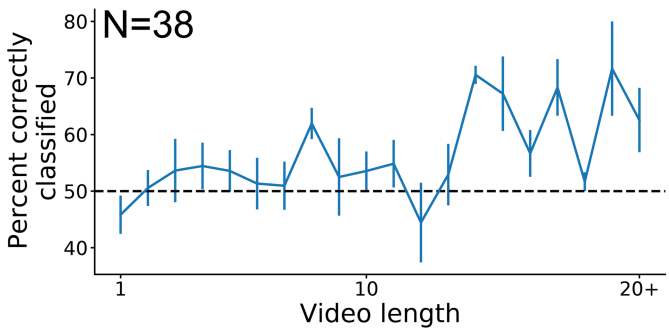
Extended Data Fig. 3 | Summary statistics. Comparing our main model directly to human choices is challenging because the data is high-dimensional and discrete. Instead, we compute summary statistics as a function of number of pieces on the board, to probe for systematic patterns in the time course of people's games, such as a tendency to start playing near the centre of the board and gradually expand outwards. We compare moves made in human-vs-human games (green solid lines), the behavioural model with inferred parameters on the same positions (blue solid lines) or random moves (black dashed lines). For all summary statistics, people deviate considerably from random, and the main model closely matches the human data. All panels depict cross-validated predictions.

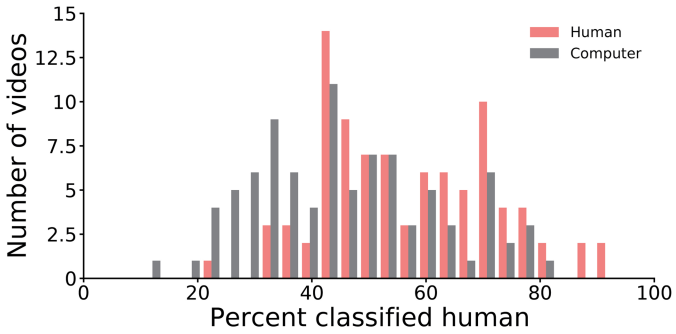


Extended Data Fig. 4 | Individual differences across summary statistics. Each panel shows a scatterplot for the same set of summary statistics as in Extended Data Fig. 3, where each point represents a participant in the human-vs-human experiment, the horizontal coordinate the statistic computed on that participant's moves, and the vertical coordinate the statistic computed on moves made by the model, with parameters inferred for that participant on out-of-sample choices. The Pearson correlation coefficient and two-sided *p*-value are reported within each panel. The model accurately predicts individual differences between participants.

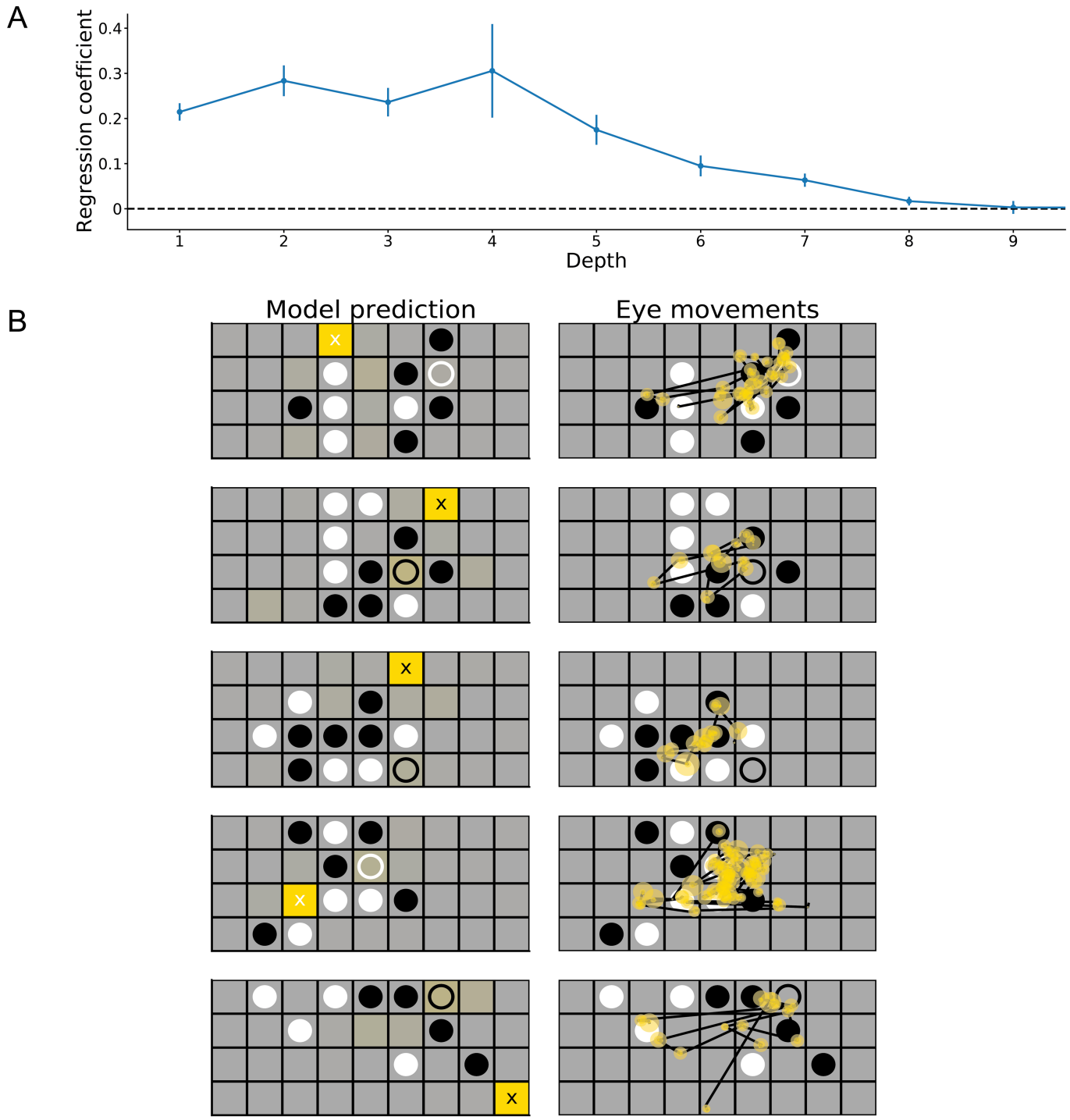


Extended Data Fig. 5 | Example board positions illustrating model components. To investigate which patterns in the data are explained by tree search and feature dropping, we compare the distribution of choices predicted by the main model against lesion models. **A**. Example positions from human-vs-human games in which the model with (right column) and without tree search (left column) make highly different predictions (red shade), as quantified by Jensen-Shannon divergence. In each position, we also show the models' preferred move (with an x) and the move made by the human participant (open circle). These predictions are averaged across simulations with 200 different parameter vectors from fits to human data, to capture positions with robust differences between planning and no planning. Upon inspection, we recognize these positions as ones where the player to move has multiple reasonable options, but to evaluate their quality one has to calculate many moves ahead. For example, in the second position, the move preferred by the **No tree** model is losing and the one by the main model is drawn, but this relies on a specific 10-move forced sequence that can only be found through explicit search. **B**. Same as **A**., but lesioning the feature drop metric, and using the ratio of the predicted probability of the human move as metric for selecting positions. The feature drop mechanism is primarily necessary to account for people's tendency to overlook possibilities to immediately make four-in-a-row, or block immediate four-in-a-row threats by the opponent.

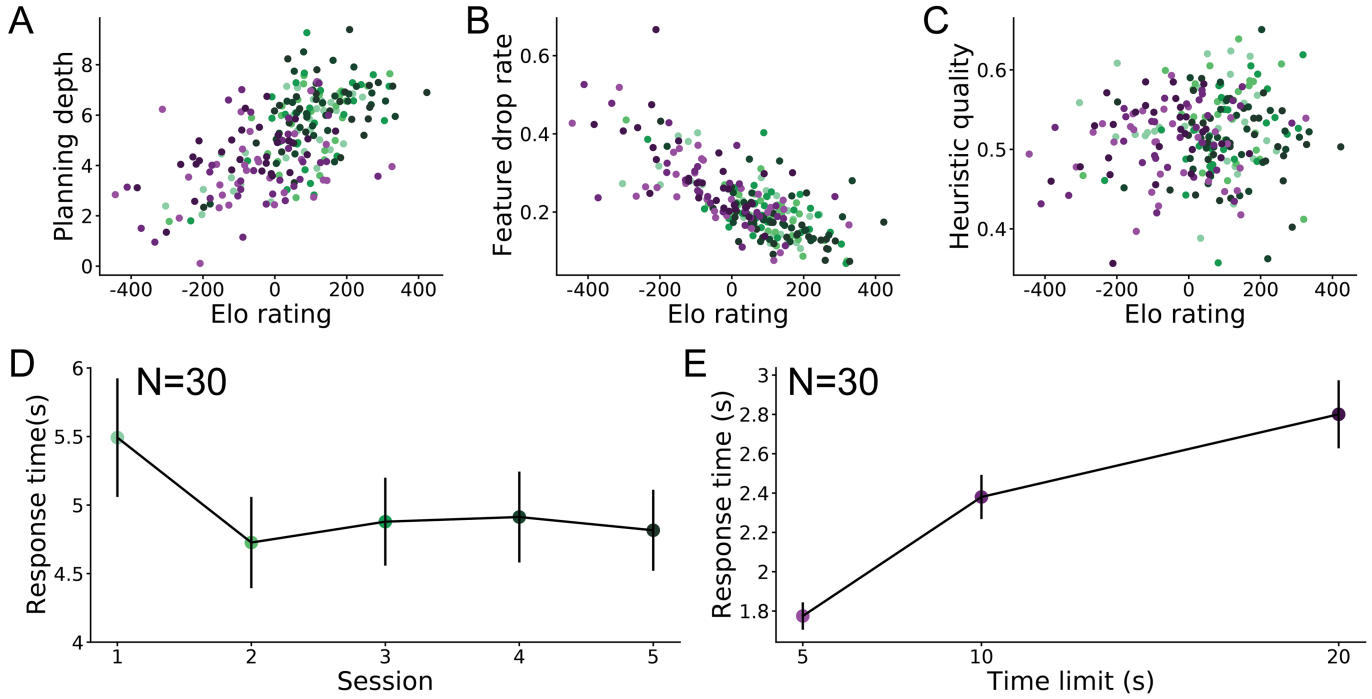




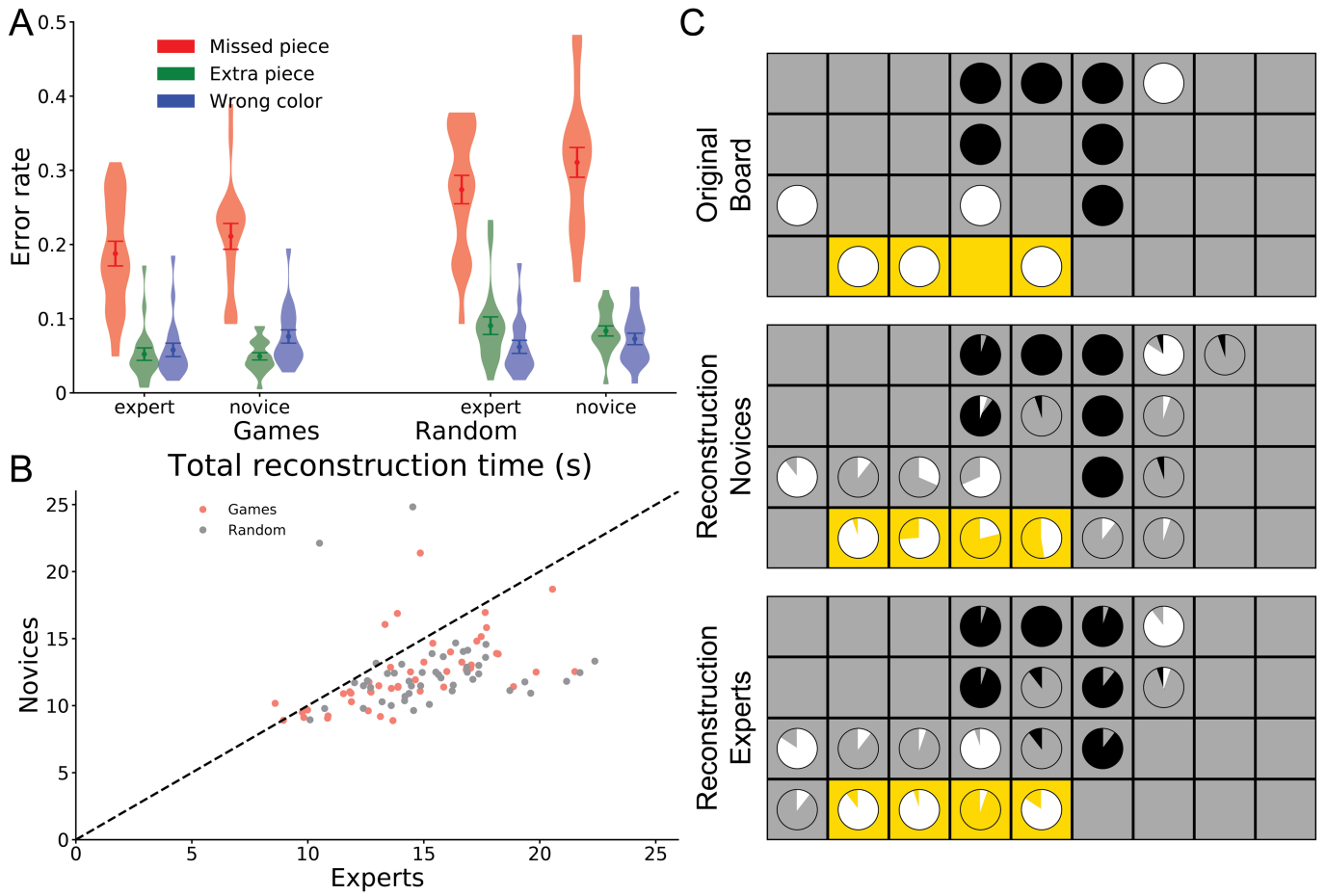
Extended Data Fig. 6 | Turing test. In the Turing test, we showed participants video segments of sequences of moves, on average 9.38 moves long. **A**. Classification accuracy in the Turing test as a function of video length. Error bars indicate s.e.m. Participants are at chance level for classification of one-move videos (of which there were 8), and their accuracy only substantially exceeds 50% for sequences longer than 10 moves. A mixed effects linear regression with accuracy as dependent variable and observer-specific random intercepts estimates the increase in accuracy per observed move as only $0.33 \pm 0.10\%$. **B**. Histogram of the percentage of observers classifying a given video as human-vs-human or computer-vs-computer, for either human games (pink), or computer-generated games (grey). While human games are on average more likely to be classified as human and computer games as computers, there are no videos for which all 30 observers agree, and there is a considerable fraction of videos (63 out of 180) for which a majority of observers respond incorrectly.



Extended Data Fig. 7 | Eye tracking. A. Coefficients in a linear regression predicting participants' attentional distribution from the distribution of squares that the model includes in its principal variation at each depth. The regression coefficients are significantly greater than zero (one-sample T-test across participants) for depth up to 7, and highest for depth closer to 1. Error bars indicate s.e.m. across participants. **B**. Example positions from the eye tracking data in which the **No feature drop** model assigns low probability to the participant's move. The right column shows the eye movements while the participant contemplates their move. In most positions, the participant spends no time whatsoever looking at the square preferred by the model, suggesting they indeed dropped the relevant four-in-a-row feature.



Extended Data Fig. 8 | Playing strength correlations and response times. **A**. Planning depth vs Elo rating of all participants in the learning (green) and time pressure experiments (purple). Playing strength correlates with planning depth ($\rho = 0.62$, $p < 0.001$). **B**. Same as **A**., for feature drop rate ($\rho = -0.73$, $p < 0.001$). **C**. Same as **A**., for heuristic quality, which does correlate with playing strength ($\rho = 0.11$, $p = 0.088$). **C**. Response times for participants in each session of the learning experiment. Error bars indicate s.e.m. across participants. Participants play slightly faster in later sessions. Therefore, our finding of increased planning in later sessions is not confounded by an increase in thinking time. Instead, people plan more while using less time. **D**. Same as **C**., for the time pressure experiment. The time limit manipulation is effective at increasing participants' response times, even though they use only a fraction of the available time on average.



Extended Data Fig. 9 | Memory and reconstruction experiment. A. Error rates in the memory and reconstruction experiment. Although experts are slightly worse than novices in the extra piece error rate ($\beta = 0.0071 \pm 0.0031$, $p = 0.049$), experts substantially outperform novices in the missed piece ($\beta = 0.037 \pm 0.006$, $p < 0.001$) and the wrong colour rate ($\beta = 0.019 \pm 0.003$, $p < 0.001$). **B.** Scatterplot of total reconstruction time for experts and novices. Each point represents a board position in the memory in reconstruction experiment, the x-coordinate the average time that experts take to finish their reconstruction, and the y-coordinate the same but for novices. Positions from games are coloured pink, randomly scrambled positions in grey. Experts take more time to reconstruct pieces ($\beta = 2.73 \pm 0.57$, $p < 0.001$), meaning that the error rate result could reflect a speed-accuracy trade-off as opposed to an overall improvement. However, experts reconstruct game-relevant features such as 3-in-a-row more accurately in the same amount of time. **C.** Example position of the memory and reconstruction experiment. The original board contains a 3-in-a-row feature on the bottom row (yellow shading). In the reconstructions, each circle indicates the distribution of pieces placed by different observers, with the angles of the grey, black and white wedges indicating the probability for that square to be empty, contain a black or contain a white piece, respectively. Novices correctly reconstruct the 3-in-a-row feature 42.1% of the time, but experts 84.2%. Together, these results suggest that players represent boards in memory in terms of game-relevant features.

Extended Data Table 1 | Robustness analysis

	Loglik per move	Planning depth		Feature drop rate		Heuristic quality	
		ρ	p	ρ	p	ρ	p
Main	−1.95	0.61	< 0.001	−0.66	< 0.001	0.02	0.85
No value noise	−2.06	0.31	< 0.001	−0.38	< 0.001	0.35	< 0.001
No feature drop	−2.0	0.53	< 0.001	Not applicable		−0.02	0.82
No pruning	−2.0	0.71	< 0.001	−0.59	< 0.001	−0.02	0.78
No tree	−1.98	Not applicable		−0.59	< 0.001	−0.10	0.20
No active scaling	−1.96	0.58	< 0.001	−0.63	< 0.001	−0.05	0.56
No 3-in-a-row	−2.21	0.67	< 0.001	−0.39	< 0.001	0.19	0.02
No connected 2-in-a-row	−2.04	0.67	< 0.001	−0.51	< 0.001	−0.15	0.06
No center	−2.03	0.58	< 0.001	−0.66	< 0.001	−0.04	0.64
No 4-in-a-row	−1.98	0.65	< 0.001	−0.64	< 0.001	0.02	0.79
No unconnected 2-in-a-row	−1.97	0.59	< 0.001	−0.65	< 0.001	0.07	0.39
Optimal weights	−2.01	0.44	< 0.001	−0.61	< 0.001	Not applicable	
Square dropping	−1.95	0.62	< 0.001	−0.67	< 0.001	0.02	0.85
Fixed iterations	−1.95	0.64	< 0.001	−0.31	< 0.001	0.02	0.80
Fixed depth	−1.94	0.59	< 0.001	−0.70	< 0.001	0.01	0.95
Fixed branching	−1.94	0.61	< 0.001	−0.36	< 0.001	−0.04	0.65
Orientation-dep. weights	−1.95	0.56	< 0.001	−0.64	< 0.001	−0.01	0.92
Orientation-dep. dropping	−1.95	0.61	< 0.001	−0.59	< 0.001	0.08	0.32
Triangle	−1.95	0.39	< 0.001	−0.72	< 0.001	−0.21	0.01
Type-dep. dropping	−1.95	0.24	0.0026	−0.70	< 0.001	−0.08	0.32
Opponent scaling	−1.95	0.59	< 0.001	−0.60	< 0.001	−0.26	0.0014

To demonstrate that correlation between Elo rating and derived metrics are robust to choices in the model specification, we report outcomes of a two-sided Pearson correlation test between Elo rating and derived metrics across all participants and sessions in the learning experiment (analogous to Extended Data Fig. 8A-C), with derived metrics computed for each alternative model specification. For the **Orientation-dependent dropping** and **Type-dependent dropping** models, we define the feature drop rate as the drop rate of the horizontal 3-in-a-row feature. For all other models and metrics, the extension is straightforward. Note that the **Fixed depth** model explores every branch of the decision tree up to the same depth, hence the planning depth is not just the length of the principal variation, but also the length of every other variation. Across all 22 models for which it is applicable, participants' Elo rating correlates strongly with planning depth and feature drop rate, confirming that our main result on the nature of expertise is robust to the choice of model specification.

nature research

Corresponding author(s): Bas van Opheusden

Last updated by author(s): Mar 27, 2023

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
☐	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☐	☒	A description of all covariates tested
☐	☒	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☐	☒	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	Data for all laboratory experiments was collected using custom code, available on through the Open Science Framework, https://osf.io/n2xjm/. For eye tracking, we used an EyeLink 1000 Plus Camera and Workstation with built-in software.
Data analysis	We used to edf2asc.exe program for SR Research to convert raw data to fixations and saccades. We used bayeselo.exe from Remi Coulom to estimate Elo ratings for both human participants and AI opponents. All other data analysis was performed using custom code, available on through the Open Science Framework, https://osf.io/n2xjm/

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

Data for all laboratory experiments is publicly available on through the Open Science Framework, https://osf.io/n2xjm/

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☐ Life sciences ☒ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Behavioural & social sciences study design

All studies must disclose on these points even when the disclosure is negative.

Study description	All our data is quantitative: actions taken by participants in a board game, as well as response times and eye movements. All analyses involve either raw data or quantities derived from raw data by model fitting.
Research sample	For laboratory experiments, participants were members of the NYU community, or participants who signed up through online flyers and the NYU research participant pool (Sona). This study sample was based on convenience and reflects the diversity of NYU students. For the large-scale mobile experiment, participants were anyone who downloaded the mobile app through the iOS App Store. The study sample was not controlled. It consisted of those who downloaded the Peak app (on iOS) and played at least 100 games of "Connect 'Em Up". We do not have access to age and gender data, but participants are primarily based in the United States of America (45%) and the United Kingdom (21%).
Sampling strategy	For laboratory experiments with a single condition for all participants (human-vs-human & generalization), we decided to collect N=40 participants. For laboratory experiments with multiple conditions (learning & time pressure), we recruited N=30 participants. For laboratory experiments with within-participant, across-trial analyses (eye tracking), we collected N=10 participants. For the Turing test, we recruited N=30 participants. For the memory and reconstruction experiment, we collected N=38 participants. For the large-scale mobile experiment, we recorded data for all users who downloaded the app, which was N=1,234,844, and analyzed data from N=1,000 participants. We chose these sample sizes to reflect best practices in cognitive science for laboratory experiments (e.g., see Drugowitsch et al, Neuron 2016; Liu et al, Science 2021 and Polanía et al., Nature Neuroscience 2018) to ensure sufficient power to detect typical effect sizes. However, we did not conduct a formal power analysis. For experiments with across-participant intended analyses (human-vs-human and generalization), we recruited additional participants, resulting in N=40. For the mobile dataset, our sample size was determined by the maximum number of participants we could analyze given our computational time budget.
Data collection	All laboratory experiments were conducted with a 21-inch Sony GDMF520 CRT monitor (resolution:1280×960 pixels, refresh rate:100Hz). The eye tracking experiment was performed using an EyeLink 1000 eye tracker (SR Research, Ltd., Mississauga, Ontario, Canada). A researcher was present during data collection in the eye tracking experiment, but only monitored the eye tracker calibration and did, to the best of our ability, not interfere with the participant's game play. In all other laboratory experiments, no one besides the participant was present during data collection. Researchers were not blinded to the experimental condition or study hypothesis. For the large-scale mobile experiment, we did not ask participants what environment they played the game in.
Timing	The human-vs-human experiment was conducted between 06/2014 and 09/2014. The eye tracking experiment was conducted from 04/2015 to 05/2015. The generalization experiment was conducted from 05/2015 to 02/2016. The learning experiment was conducted from 09/2015 to 04/2016. The time pressure experiment was conducted from 09/2016 to 11/2016. The Turing test experiment was conducted between 11/2016 and 02/2017. The memory and reconstruction experiment was conducted from 02/2016 and 03/2017. In the large-scale mobile experiment, we analyze data collected from 09/2018 to 04/2019.
Data exclusions	We excluded no data from participants in laboratory experiments. For the large-scale mobile experiment, we randomly sampled 1,000 out of 1,234,844 participants for analysis. This subsampling was necessary for computational time constraints.
Non-participation	In laboratory experiments, 1 participant decided to quit the experiment early because they had another unforeseen engagement. They still received payment for the session, as outlined in the consent form. In the large-scale mobile experiment, participants provided informed consent by clicking a notification in the app; we did not record how many participants declined.
Randomization	In all experiments with human-vs-computer play, we randomized the settings of the computer opponents on each game. In the time pressure experiment, we also randomized the time limit per-game. In the 2AFC, evaluation and memory and reconstruction experiments, we selected the pre-generated board positions using block randomization.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☐	☒ Human research participants
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging

Human research participants

Policy information about studies involving human research participants

Population characteristics	For laboratory experiments, participants were members of the NYU community, or participants who signed up through online flyers and the NYU research participant pool (Sona). This study sample was based on convenience and reflects the diversity of NYU students. For the large-scale mobile experiment, participants were anyone who downloaded the mobile app through the iOS App Store. The study sample was not controlled. It consisted of those who downloaded the Peak app (on iOS) and played at least 100 games of "Connect 'Em Up". We do not have access to age and gender data, but participants are primarily based in the United States of America (45%) and the United Kingdom (21%).
Recruitment	Laboratory participants were recruited through mailing lists, flyers, the NYU Sona system, and our personal networks. Our participant population reflects the demographic diversity of New York City, with a bias towards people within NYU's academic network. Additionally, participants might be self-selected for an interest in cognitive tasks and may have above-average motivation to improve. Participants in the mobile dataset were a pseudo-random subset of Peak app users. No special recruitment took place beyond how Peak recruited its users. Participants hailed from across the globe. Our recruitment sample reflects a self-selection bias since people need to download the brain training app, and we additionally select participants who play at least 100 games. These selection biases might limit our results to people who are sufficiently motivated to participate in behavioral experiments or use brain training apps. Less motivated participants might show a lesser amount of learning on average, making our analytical approach more difficult to execute. We do not have reason to believe that any of our conclusions would change.
Ethics oversight	Our experiments were approved by the Institutional Review Board of New York University

Note that full information on the approval of the study protocol must also be provided in the manuscript.