Modeling Human Reasoning in Combinatorial Strategy Games

Gianni Galbiati

New York University

September 2016

A thesis in the Department of Psychology submitted to the faculty of the Graduate School of Arts & Science in partial fulfillment of the requirements for the degree of Master of Arts at New York University

Sponsor:_____

Wei Ji Ma, Ph.D.

Reader:_____

Pascal Wallisch, Ph.D.

Abstract

People have played strategy games recreationally for millennia, yet the process by which they make decisions in these complex, sequentially contingent environments remains underexplored. Psychological work has suggested two important components: "chunking" by decomposing a game position into relevant features, and "thinking ahead" by iteratively simulating a move and evaluating the expected outcome (Miller, 1956; Chase & Simon, 1973; de Groot, 1978). However, this body of research has not yet produced a generative model capable of directly predicting the move a given person will make in a given position. In the meantime, artificial intelligence researchers have produced sophisticated algorithms to play games with performance beyond human expertise. We adapt some of these algorithms as generative models for human gameplay by freeing and then fitting parameters to individual subjects playing an unfamiliar combinatorial game. In doing so, we quantitatively corroborate the qualitative findings of past psychological research and establish a new experimental paradigm for research on human gameplay and procedural rationality.

Keywords: combinatorial games, computational models, procedural rationality

Computational Modelling of Human Cognition in Combinatorial Strategy Games

For millennia, humankind has played a variety of strategy games for sport (Romain, 2000). Some well-known examples include chess, checkers, go, and backgammon. A few of these games, such as nim, even have formal solutions; for others, including chess and go, there now exist sophisticated computer algorithms that outplay even the strongest human players (van den Herik et al., 2002; Silver et al., 2016). However, despite the success of artificial intelligence researchers at developing such powerful programs, there has been relatively little progress on a complementary question: how do people play strategy games?

Chess and Psychology

For psychologists, chess has been the paradigmatic strategy game for the last 50 years. Some foundational studies were performed by Adriaan de Groot, who impelled chess players of varying skill level to narrate their thoughts aloud as they considered chess positions (de Groot, 1946/1978). He found that players talked explicitly about considering individual moves and planned several moves ahead. While it may seem intuitive that the more moves a player considers, the better their chances of finding the best available move, de Groot concluded that chess experts actually consider a smaller number of candidate moves in greater detail than novices.

Chase and Simon later showed players preconfigured arrangements of chess pieces (Chase & Simon, 1973). Some of these arrangements were positions taken from real chess games; others were pieces randomly scattered on the board. The researchers asked players of varying skill to recreate the most recently seen board from memory. Novice players showed little difference in their ability to remember the configuration of pieces of real or random positions,

and on random positions expert chess players performed no better than novices. However, expert chess players were substantially better than novices at recreating the configurations from real positions. Chase and Simon concluded that the development of skill at chess involved improved ability to recognize functionally relevant arrangements of pieces, which they call "chunks" following Miller (1956), that arise regularly during chess play.

Chase and Simon's finding can be used to tell a deeper story about de Groot's observations (Simon & Schaeffer, 1990). Expert chess players use relevant chunks or features to quickly evaluate and sort candidate moves, freeing them to plan ahead more efficiently (Miller, 1956; Chase & Simon, 1973; Gobet & Simon, 1998a). Because they can plan ahead more efficiently, they can spend more time and resources considering the most promising candidates in greater detail. Other work has explored the effect of various manipulations on game-related performance, importantly demonstrating that in addition to the benefits of learning good features, spending more time deliberating also improves performance for experts and novices alike (Moxley et al., 2012).

Chess and Artificial Intelligence

Similar pre-experimental intuitions contributed to the development of computer algorithms for chess. Claude Shannon, inspired by de Groot's work, presented one of the earliest designs for a chess-playing computer, or agent (de Groot, 1946/1978; Shannon, 1950). Shannon begins the development of his agent by describing an algorithm that builds a game tree by iteratively exploring all available moves, the resulting moves available to the opponent, and so on until each branch of the constructed tree terminates in a win, draw, or loss. Once every

possible sequence of moves has been explored, the agent can work backward from game ends to determine whether a forced win or draw is available and the appropriate moves to make.

As Shannon observes, following chess statistics from de Groot (1946/1978), the number of possible legal games of chess from the starting position is on the order of 10^{120}, which would require over 10^{90} years to calculate at a rate of 1 position per millisecond (Shannon, 1950). Even on modern hardware, such an algorithm would require multiple lifetimes of the universe just to make the first move. Shannon also suggests an alternative method that stores the optimal move for each possible position in an associative array and observes that because the number of possible positions is on the order of 10^{43}, which exceeds estimates of the number of molecules in the universe, such a method is likewise infeasible (Simon & Schaeffer, 1990).[1]

Shannon circumvents these problems by describing what is now called a heuristic search algorithm (Shannon, 1950; Hart, Nilsson, & Raphael, 1968; Allis, 1994; Russell & Norvig, 2009). Heuristic search algorithms have two main components: a heuristic function, and a tree search procedure. The heuristic function takes a game position as an argument and returns a value. The higher the value, the better the position is estimated to be. For the tree search algorithm, Shannon uses a "minimax" procedure that finds the sequence of a small, fixed number moves resulting in a position with the maximum heuristic value given that the opponent selects moves that minimize the value of the resulting position. Shannon's algorithm does so by first simulating all possible moves for the player, all possible moves for the opponent in each of the

[1] For reference, in artificial intelligence this sort of algorithm is known as a pattern database (Culberson & Schaeffer, 1998).

resulting positions, and so on for a limited number of turns. It then works backwards from the most distant moves by selecting moves for the opponent that minimize the heuristic value and moves for the current player that maximize the heuristic value. With a good heuristic function, many fewer moves need be explored explicitly to achieve good performance. However, finding a good heuristic function, particularly for games with complicated rules like chess, can be quite difficult.

There is a clear, intuitive analogy between heuristic search and the psychological findings of de Groot and Chase and Simon. The way in which experts have learned the roles of functionally relevant features can be represented by the heuristic function, and the way in which experts think ahead can likewise be represented by the building of game trees by a computer algorithm. Heuristic search, a product of artificial intelligence research, is naturally interpretable as a computational model of human cognition (de Groot, 1978; Simon & Schaeffer, 1990).

Simon and Schaeffer's synthesis of early chess psychology research into a "chunking theory" constitutes a model description of the family of heuristic search algorithms: they present a verbal description of a computational procedure that picks out a relational structure between its inputs and outputs (Simon & Schaeffer, 1990; Weisberg, 2013). However, psychologists have yet to cache out Simon and Schaeffer's model descriptions and actually test a full heuristic search algorithm as a generative model by fitting parameters to data from human gameplay. Subsequent research has instead been focused on the relationship between chess knowledge qua memorized chunks, or features, and expertise. Despite the well-established theoretical relationship between search and chunking, research on tree search has by and large been quiescent, excepting a handful of simulation studies (Miller, 1956; Simon & Gilmartin, 1973; de

Groot, 1978; Berliner & Ebeling, 1988; Gobet & Charness, 2006). Essentially, researchers have treated the chunking-search model description as a syntactic view theory and attempted to investigate chunking indirectly without testing predictions from a generative model (da Costa & French, 1990; Suppe, 2000).

Advantages of Modeling

Traditionally, psychologists have adopted a syntactic view of theorization emphasizing deductive systems of propositions. However, philosophers of science overwhelmingly favor semantic views, which focus on the evaluation of models as representational structures, for theoretical and practical advantages (Cartwright, 1983; da Costa & French, 1990; Suppe, 2000; Weisberg, 2013). Importantly, explicit computational modelling allows graded quantitative evaluations of theoretical quality. The core normative question of a model on a semantic view is not whether the model is falsified by observations, but rather how well the model represents the target phenomena (van Fraassen, 1980; van Fraassen, 2008; Weisberg, 2013).[2] While richer evaluations are inherently desirable, they have two further advantages. First, following Weisberg (2013), models are interpreted structures, and as such have individual components, substructures, and variables. These components can be evaluated individually in two ways. First, they can be removed from the model one at a time to determine their contribution to the overall predictive success of the model; this is sometimes called model lesioning, analogous to the study of the

[2] As a familiar case, typical between-groups statistical significance testing is an exemplar of hypothetico-deductive testing for whether a hypothetical proposition derived from a syntactic theory is falsified by evidence. Effect sizes and other statistical measures are typically not directly predicted from a theory and therefore become secondary concerns in theory evaluation.

effects of brain lesions. Second, model components can also be evaluated by interpreting them as or relating them to measures of secondary observable phenomena.[3]

These two approaches to component evaluation can be thought of as roughly corresponding to Weisberg's distinction between dynamical and representational fidelity criteria. Dynamical (predictive) fidelity criteria "deal only with the [model's] predictions about how a real-world phenomenon will behave," while representational fidelity criteria measure "how well the structure of the model maps onto the target system of interest" (Weisberg, 2013).[4] The first approach to analyzing model components, lesioning, evaluates the importance of components using predictive accuracy, for example as measured by log-likelihood, as the predictive fidelity criterion. The second approach, using inferred component variables to predict secondary empirical measures, is best understood as a representational fidelity measure because it directly evaluates similarity between a real phenomenon and a model component construed as representing that phenomenon.

The quantifiability of model quality allows for the side-by-side relative comparison of multiple models. In syntactic accounts of theorization, models are understood as mere illustrations or demonstrations and are not of direct theoretical interest. As a result, side-by-side comparison of theories is not possible because theories and deduced hypotheses themselves are

---

[3] For example, the main phenomena of interest in a forced choice task might be the stimulus and the response, but response times may also be recorded and compared to model components.

[4] Weisberg develops his account of mathematical modeling primarily with reference to dynamical models, but it applies equally to other types of mathematical and computational models. Our models are not dynamic, so for clarity I will be using the term predictive fidelity instead of dynamical fidelity.

evaluated as propositions, which can only take one of two values: true or false. Comparing models' predictive performance directly thus allows for a more thorough, comprehensive evaluation of competing theories.

Misunderstanding of Modeling

The semantic view, widely accepted by contemporary philosophers of science, is poorly understood by many practicing scientists. Their lack of exposure is made abundantly clear by the recent disagreements over the increasingly popular practice of Bayesian modelling. Due to their commitment to Popperian hypothetico-deductivism and falsifiability, critics of Bayesian modelling in psychology accidentally target all of generative modelling (Marcus & Davis, 2013; Bowers & Davis, 2012a). Fortunately, their criticisms are defused by correctly rejecting hypothetico-deductivism as a metatheory appropriate to modelling. Unfortunately, proponents of Bayesian modelling appear incapable of making this response, due to the same acceptance of Popperian science or a naive Bayesian inductivist picture (Suppe, 2002; Jones & Love, 2011; Bowers & Davis, 2012b; Gelman & Shalizi, 2012; Griffiths et al., 2012; Gelman & Shalizi, 2013). Tellingly, papers on both sides in this discourse fail to cite any of the highly relevant philosophical work produced over the last 50 years.[5]

[5] The lone exception appears to be Gelman and Shalizi (2012), who at least attempted to engage with contemporary philosophers. While they correctly identify that the common folk-scientist account of Bayesian inference as induction is troubled, they fail to recognize and consider alternatives to hypothetico-deductivism.

It is thus important to set the record straight by rejecting the shoehorning of scientific modelling into the outmoded Popperian picture.[6] The most visible anti-Bayesian mistakes can be largely understood as variants of two criticisms: models are not falsifiable, and models are unavoidably configured post-hoc (Bowers & Davis, 2012a; Marcus & Davis, 2013).[7,8] Both mistakes directly result from attempts to treat models as though they are hypothetico-deductive theories.

The first criticism is a *non sequitur*: of course models are not falsifiable. Models are not propositions; they are not in the business of being true or false (Popper, 1959; Suppes, 1961; van Fraassen, 1980; Cartwright, 1983; da Costa & French, 1990; Brading & Landry 2006).[9] If a

---

[6] A thorough defense of the semantic view and a full accounting of the syntactic view's inadequacies is well beyond the scope of this paper and more adequately treated elsewhere by philosophers. The body of this literature is enormous, but some excellent examples can be found in Quine (1951), Suppes (1961, 1967), Suppe (1977, 1989, 2000), van Fraassen (1980, 2008), Cartwright (1983), da Costa & French (1990), Frigg (2006), and Weisberg (2013).

[7] Many Bayesians have further discursive problems, as observed by Bowers and Davis (2012b), that result from equivocation over the relationship between Bayesian models, rationality, and whether or not the brain literally implements Bayesian inference.

[8] I am again bracketing the concerns of Gelman and Shalizi (2012, 2013), who make important observations about the apparent adoption of a form of inductivism by some contemporary Bayesian modelers and raise appropriate objections.

[9] To elaborate a little, models are more like maps than they are like propositions. If we are using a map to navigate, we might evaluate it on its accuracy in portraying the relevant aspects of the landscape, but we would not describe the map itself as being "true" or "false". A map is a physical object that represents the physical world as

researcher says a model is false, either they are making an overly casual statement about how well the model captures what it is intended to capture, or they are deeply confused. When Popper originally introduced falsifiability as a demarcation criterion, it was in the context of the positivist received view, or syntactic view, of theories, in which theories are understood strictly as deductive systems of theoretical axioms and hypothetical propositions (Popper, 1959; Suppe, 2000). However, while Popperian theory continues to be taught in introductory philosophy of science courses, it is no longer widely accepted among professional theoreticians (Hempel, 1974; Suppe, 2000; Frigg, 2006). Falsifiability is not a suitable criterion for evaluating models, and the only reason to attempt to apply it to models is the reflexive presupposition of Popper's account of scientific theorization.

The second criticism is likewise addressed if we correctly understand models as distinct from the hypotheses used in null-hypothesis significance testing (NHST) paradigms. In NHST, post-hoc hypothesizing is largely problematic in multidimensional data because researchers can "fish" or "dredge" for significant test statistics across many variables, increasing the probability of false positives. While fitting free parameters in a computational model is in some sense likewise sifting through a large hypothesis space, the model structure, or the uninstantiated model description in Weisberg's terminology, is typically the object of interest, not individual parameter settings and their corresponding instantiated model.

---

being a certain way, and is evaluable with respect to our intentions for it, such as successful navigation. Likewise, models in science, even when not concrete objects, are intentional representations, not propositions (van Fraassen, 2008; Weisberg, 2013).

A variant of this objection targets the comparison of multiple models on a single data set. However, when comparing multiple models, the primary objective is not to find a model with the highest raw performance, but rather to select the model with highest relative performance. In this case, each additional model actually reduces the even-odds probability of a given model being selected.

Problems with Chess as an Experimental Paradigm

The desirability of a model to cash out the model description suggested by Simon and Schaeffer (1990) is clear; nonetheless, contrary to Chase and Simon, we find that chess and other natural games present several problems as "model environments" (Chase & Simon, 1973). First, chess is too familiar to the general population. Chess experts have typically played thousands of hours, many novices already have a general sense of some important patterns, and almost every subject accessible to experimenters will have played the game previously. Second, paradoxically, chess is also too difficult for both human and computer players. The difficulty has three sources: heterogenous pieces, complex rules, and a relatively large board. Together, the first two entail that subjects must remember a large ruleset and potentially learn a large number of complex features, and that a heuristic search algorithm requires a complicated, burdensome heuristic function. The third source of difficulty results in the previously discussed immensity of the game tree in chess.

However, understanding how people play chess specifically is not the real prize; like Simon, we are interested in the more general question of how people think ahead, idealized as the play of strategy games (Simon & Schaeffer, 1990). To this end, we selected a more suitable combinatorial game of perfect information, recorded human subjects playing, and used a variety

of artificial intelligence algorithms as generative models of their decisions. Combinatorial games are deterministic, and in games of perfect information, all players have exactly the same knowledge about the state of the game (Berlekamp & Conway, et al., 1982). These properties help isolate the cognitive processes involved in thinking ahead by eliminating the need for players to consider uncertainty about the outcomes of their moves or the moves available to their opponent. Additionally, combinatorial games can in principle be played perfectly, though often not so in practice, which supports the use of psychometric techniques, such as estimating a player's general strength (Neumann & Morgenstern, 1944; Shannon, 1950).

An Alternative To Chess: (*m, n, k*) Games

To avoid the problems with chess as an experimental paradigm, we selected an (m, n, k) game as our fundamental task. (m, n, k) games are a broad class of games, previously studied by mathematicians and computer scientists, in which players take turns placing pieces on an m by n grid, and the first player to place k of their own pieces in a row in any orientation wins the game (van den Herik et al., 2002; Wu & Huang 2005). A well-known example of an (m, n, k) game is tic-tac-toe: players take turns placing pieces on a 3 by 3 grid, and the first player to place 3 pieces in a row wins. Tic-tac-toe, however, is far too simple a game to be of experimental interest: the game is guaranteed to be tied if the both players play perfectly, and most people can learn perfect play with relative ease if they have not already (Uiterwijk & van den Herik, 2000).

We chose to use the game (4, 9, 4), which has a 4 by 9 grid on which players must place 4 in a row to win (Figure 1). As a combinatorial game of perfect information, (4, 9, 4) retains many of chess's useful properties with a much more manageable level of theoretical and

practical complexity. Unlike chess, the number of available moves in a position is exactly equal to the number of unoccupied squares, and unlike tic-tac-toe, strategy is nontrivial and subjects cannot easily learn perfect play. The state space of (4, 9, 4) is about 10^{20}, many orders of magnitude smaller than chess's 10^{43} and larger than tic-tac-toe's 765. Furthermore, (4, 9, 4) is an unfamiliar game to most people, so subjects are reliably novices, moderating the influence of past experience as occurs in the study of chess. The simplicity of the rules means subjects can nevertheless learn to play with minimal instruction. Because of these virtues, (4, 9, 4) is well suited as an experimental game for the study of cognitive processes underlying human play.

Using (4, 9, 4) To Study Human Gameplay

To determine the quality of heuristic search as a representation of how people think ahead in strategy games, we recorded human subjects playing (4, 9, 4) against each other in pairs and used a heuristic search algorithm as a generative model by freeing and fitting parameters in the tree building procedure and the heuristic function. We additionally employ several alternative models that are not consistent with the model description explored in prior research. Each of our models takes a representation of a game board that a player saw as input and returns a prediction for the corresponding move made by that player. We fit each model's parameters to individual subjects and use the average negative log-likelihood of the model's predictions on the test set as our predictive fidelity measure.

Models. Our first model is a heuristic search algorithm, HS, which approximately corresponds to Chase and Simon and de Groot's qualitative characterization of reasoning in chess by using a heuristic function to evaluate positions and explore only the most promising branches of the decision tree. The second model is a Monte Carlo tree search algorithm, MCTS,

which uses random simulations of game tree branches to find the most promising moves at each depth (Abramson & Korf, 1987; Kocsis & Szepesvári, 2006; Enzenberger et al., 2010). The third model, NT, makes a decision using the heuristic function from HS, but does not build a game tree. Fourth, NT + C_{opp} is NT with the addition of an "opponent scaling" parameter that gives a relative boost or discount to the aggregate value of the opponent's position in HS's heuristic function. Fifth, CN is a convolutional neural network, a popular class of algorithms in image recognition that has recently seen some success in artificial intelligence gameplay (Clark & Stokey, 2014; LeCun, Bengio, & Hinton, 2015; Silver et al., 2016). Finally, we use a fitted softmax function, SM, and an optimal-random mixture model, OR, as unstructured control models. All models treat trials as independent and identically distributed; we do not model dependence between trials.[10] More extensive descriptions of each model and theoretical discussion are available in the supplementary information.

Based on the similarities between of heuristic search algorithms and successful cognitive model descriptions, we expected that HS would have the highest performance as measured by cross-validated log-likelihood of the model's predictions of moves in subject gameplay data. Our other models are alternative hypotheses that are inconsistent with the model description suggested by past researchers, despite Monte Carlo tree search and convolutional neural networks having been demonstrated as powerful game playing algorithms in their own right. We additionally predict that the response time of subjects will correlate with internal properties of HS, such as the number of positions simulated by the model before it makes a response. The

[10] To be completely clear, in reality the trials are neither identically nor independently distributed.

ability to make secondary predictions using internal model features is critical to establishing the success of heuristic search as a model of decision making in games by quantifying the representational fidelity of at least one internal component.

Methods

Participants

We recruited 40 participants aged 20 to 28 in pairs from the New York University student community. Participants were only required to have corrected-to-normal vision and be proficient enough in spoken English to understand the experiment instructions. Participants were each compensated $12 for approximately one hour of participation.

Research Design

The study is within-subjects and observational. The researchers lack control over the stimuli seen by the subjects or the number of times subjects see identical stimuli, which instead result directly from the decisions of subjects. Thus, the “independent” measure is not completely controlled and randomized by the experimenter. We construct an environment with certain rules and then passively observe subjects’ behavior in response to particular states of that environment. As we lack control over the particularities of stimuli within that environment, we do not believe our study is properly called experimental. As we do not analyze any preexisting differences between subjects, our study is likewise not properly called quasi-experimental. We recorded behavior of subjects in a relatively uncontrolled environment, so by these delineators our project is fundamentally observational. However, because we use maximum likelihood estimation in probabilistic models instead of conventional hypothesis testing, we perform similar

analyses to those we would should we have chosen to use a controlled repeated-measures design with independent randomized trials.

Procedures

After signing consent forms and receiving written and verbal instructions, each participant in a pair was seated in a separate room at a computer terminal. The experiment, coded as a web application, was launched in Google's Chrome web browser. The two browsers communicated and recorded data through an Apache-PHP server with a MySQL database. Participants were allowed to play against each other for one hour plus the amount of time required to finish their final game. There was no time limit on individual moves; participants were free to deliberate as long as they wished. Participants were permitted to take breaks as desired, but once they began playing they were not allowed to interact directly until the end of the experiment.

Each game began with a blank nine by four playing board (Figure 1). The player with black pieces went first. Each player took turns placing one of their playing pieces on an empty square. The first player to place four of their own pieces in a row won the game. Games could also end in draws, either when the entire board was full with neither player achieving a four-in-a-row, or when both players agreed to a draw by clicking a button. Participants took turns as Black and White in alternating games.

Measures

We recorded each position each participant saw, the color they were currently playing, the move they made in response to that position, when they requested a draw, and the time it took for them to respond after their opponent's move updated on their screen.

Results

Predictive fidelity measure

As measured by cross validated log-likelihood, the heuristic search model HS outperforms all other models we tested, with an average negative log likelihood of 2.05 ± .04 (Figure 2), a 35% improvement over the negative log likelihood of random guessing at 3.20. All models are significantly better than both chance and the relatively unstructured control models SM and OR. Figure 3 shows the distribution of average negative log likelihoods per subject; figures 4 and 5 show more detail for the relative performance of each model on different subjects. The lesioning comparison demonstrated that tree building, the three-in-a-row feature, random feature dropping, and the stop condition were all significant contributors to performance (Figure 6). The remaining components in the analysis may have contributed, but their contributions were not significant. Figure 7 shows that CN significantly outperforms HS on the first two moves, but consistently underperforms from the ninth move on. Additionally, Figure 7 illustrates that the width of the Bayesian credible interval for the mean of model performance increases as the number of moves in the game increases.

Representational fidelity measure: Response time

Players' response times are roughly exponentially distributed across the population, and median response times were strongly correlated within pairs of players with $R^2 = .42$, $p < .001$ (Figure 8). Additionally, average response time is lower when the board is either very empty or very full, and response times increase as play continues until about five moves before the end of a game, where response time drops off precipitously (Figure 9). Finally, to a varying degree,

response times also correlate with the number of positions explored during search by HS after fitting (Figure 10).

Discussion

Findings

Scientific models are evaluated against two kinds of fidelity criteria: dynamical, or predictive, fidelity criteria, and representational fidelity criteria (Weisberg, 2013). Our measure for the predictive fidelity criterion is the average negative log-likelihood per subject. Because we are primarily interested in the relative performance of multiple models, we are not deeply concerned with a specific criterion. Instead, we use a "chance" model, which makes a move at random, as well as two minimally structured models, SM and OR, to establish several performance baselines that we correctly expect all competitive models to beat.

The heuristic search model HS was effectively tied for best performance with the treeless model augmented with opponent scaling, NT + C_{opp}, on the gameplay data we collected. While the mean negative log likelihood for NT + C_{opp} is strictly less than that for HS, its 95% confidence interval contains the mean performance of the HS model, so we cannot confidently say that their performance is distinguishable on our chosen predictive fidelity criterion alone.

However, using additional measures for representational fidelity allows us to further distinguish between these two models. In this case, we are able to use HS to attempt predictions for other empirical measures, allowing for the evaluation of performance against representational fidelity criteria. Specifically, we can use the number of moves explored by HS during tree building as a representation of how much thought a subject gives to a particular position, allowing us to predict response times for some subjects (Figure 10).

However, the successfulness of the prediction is at best mixed - for more than half the subjects, there is no significant correlation at $p < .001$, and for at least one, the correlation is at $p < .05$ and negative. Because there are 40 correlations being performed, $p < .001$ is the most appropriate threshold for accepting the statistical hypothesis that log tree size is correlated with log response times. In Figure 10, we report bootstrapped means ($n = 10{,}000$) for the Pearson correlation coefficient between log tree size and log response times. Given the poor-to-nonexistent correlation for most of our subjects, it is readily apparent that despite the weak average correlation across the population, we cannot claim this analysis as evidence that HS faithfully represents the unobservable cognitive process that results in observable response times.

This failure presents an opportunity to raise another criticism of HS. While we take the model description advanced by past research as a starting point, HS diverges from the classical account in critical ways. Most importantly, as contrary to the description of the chunking theory in Gobet and Simon (1998) and Simon and Gilmartin's (1973) simulation work on their EPAM model, the heuristic function in HS has a very small number of fixed features. Thus, HS is unable to do one of the important things suggested by past research, which is add high level features to its heuristic function to make search more efficient. For example, in (4, 9, 4) there are some complex patterns that guarantee a win two moves in advance. A model that could learn such representations nonlinearly would be able to search two moves fewer to find that positions with these features had high heuristic values. If human cognition is more similar to classical chunking descriptions of chess, then HS may not build trees in the same way as people, which is consistent with the general lack of relationship between tree size and response times.

Regardless, because NT + C_{opp} lacks the tree search procedure, it also entirely lacks the components we interpret as the amount of deliberation in which a subject engages. We are therefore unable even to attempt to use NT + C_{opp} to make the same response time prediction; there is no component to NT + C_{opp} that could similarly be interpreted as a predictor of thinking time. One important caveat is that we could similarly take the total operations performed by each algorithm as the predictor for response time, but do so is too literal in principle and too burdensome in practice. We therefore count the additional representational capacity of HS as a strong reason to prefer it over NT + C_{opp}, even though its representation is largely unsuccessful.[11]

Furthermore, despite its departures from chunking theory's details, HS is the only model we created that is consistent with past research on chess. The remaining models were developed from other artificial intelligence algorithms without much attention to human cognition. For example, we regard MCTS to be cognitively implausible. Humans almost certainly do not simulate many hundreds of random moves. Past work on chess expertise is consistent with this intuition: even chess experts do not recount considering hundreds of moves (de Groot, 1978;

[11] There is a third possible predictor for response time that could be developed, for example, from the entropy of the likelihood distribution output from the models. However, it is not clear that this response time prediction should be counted as a representational fidelity measure, but rather as the predictive fidelity measure for a different model that incorporates likelihood from the first model to make response time predictions. In other words, to do so would be to invert the current measures: the correct prediction of response time becomes a predictive measure, and the correct prediction of subject moves becomes a representational measure. We are primarily interested in how humans make moves, not how much time they require to think, and so inverting the criteria does not serve our purposes. Appropriate alternatives must be components measures of the process that produces the model predictions, not a property of the model predictions.

Chase & Simon, 1973). CN and HS are more psychologically plausible than MCTS, but regardless, both are worse than HS in predictive performance.

Despite the lower predictive performance of CN, we do not believe we have exhaustively explored the possibilities provided by neural networks. One important concern is that CN is handicapped in this analysis due to a lack of data. Even with data augmentation, attempting to fit CN's weights to individual subjects results in extreme overfitting and low test set performance of 3.36, or close to random, indicating that more data is necessary to achieve good test set performance on individual subjects. The superior early game fit and the response of CN to an empty board (Figures 7 and 11) further suggest that the network is capable of even better performance, particularly in the earliest stages of the game. The reason for the particular shape is that some subjects prefer to play the first move on a corner, and some prefer to play closer to the center (Figure 12). In general, because CN is close to HS in performance and is prone to overfitting (Figure 13), we believe that with more data, CN may have a greater capacity to capture individual differences. Collectively, these observations imply that simply collecting more data might enable CN's performance to equal or surpass HS. We plan to explore this potential in future experiments with much more data per subject.

The fact that convolutional neural networks require so much training data is itself an important shortcoming. Because CN has no built-in game-specific knowledge, it must learn how to play from trial and error, as corresponds to the gradient descent fitting procedure. However, subjects are able to immediately grasp the rules and do not require thousands of trials to learn to play successfully. Additionally, the features learned by CN (Figure S3) are 4 by 4 by 2 tensors with non-binary elements. It is plausible but unintuitive that subjects learn such large, flexible

feature representations; even though CN most clearly matches our intuitions about neural implementations, it fails to correspond to the self-reports of chess players in past research (de Groot, 1946/1978).

Somewhat paradoxically, even though CN's lack of game knowledge is troubling from an experimental perspective, its ability to perform well also provides reason to take CN's predictive performance seriously and continue to develop neural networks as models for our task. Unlike HS, which has hard-coded features selected according to a game-related criterion, CN learns to imitate subjects entirely from scratch. This task is nontrivial, and when we added to CN the nonlinearity that prevents illegal moves, we were able to halve the number of required features (Figure S4) while significantly improving prediction quality in the best-fitting model. Thus we understand that "batteries included" task knowledge is an advantage both intuitively and empirically. In fact, we speculate that learning game rules may occupy the bulk of the network's training data requirements; in the future, we may examine networks with partially hard-coded features and weights and attempt pretraining networks using reinforcement learning.

Another concern for HS is the indistinguishability of the predictive performance of HS and HS_{agg} (Figure 2). If fitting parameters to individual subjects produces little improvement over fitting to the data aggregated across subjects, there are three possible circumstances. First, HS may also be suffering from insufficient data, and individual fits per subject are below potential as a result. However, we find no correlation between the number of observations and the fit of HS for each subject ($R^2 = .01$, $p = .47$). Second, it may be that HS does not have the capacity to capture substantial individual differences. With 10 free parameters, however, HS should be sufficiently flexible to capture a variety of behaviors. Finally, there might not be substantial

individual differences in the first place, as (4, 9, 4) is easy enough for subjects to learn to play quite well. Because the game has neither complicated rules nor a large state space, there is comparatively little variety in successful strategies and winning patterns. As a result, subject behavior may converge as skill increases such that there is relatively little individual difference for HS to learn.

Our component lesioning analysis provides some useful insights. Primarily, we can look at components for different model subprocesses - the heuristic function, the tree construction procedure, and noise - to evaluate their relative importance. One important takeaway is that all the components except the four-in-a-row feature appear to be essential. It might appear surprising that the feature indicating a win is inessential, but it is actually redundant with the tree building algorithm, which returns fixed values for terminal nodes, or the ends of games. The fact that virtually all components are necessary is further surprising in light of the success of NT + C_{opp}, which replaces all the tree building components in lesioning analysis as well as the tree itself and achieves very similar performance. This circumstance might indicate that the extra complexity of HS results in a less successful fitting procedure, that the additional parameters in HS contribute to overfitting beyond improved performance, or that the scaling parameter NT + C_{opp} captures a feature of behavior not included in HS. The former case has no simple solution, and the second case could be ameliorated by collecting more data, as is the case for CN. We plan to address the third case in future work by modifying the heuristic function of HS to include a scaling term for the opposing player at each step in the tree building procedure.

Additionally, we do not know the generality of the model's representation of cognition for this game; to establish this, we will be pursuing extensions of the game (4, 9, 4) to forced

choice, puzzles, qualitative judgments of positions, and other related tasks. In future work, we will pay particularly close attention to qualitative judgments as a new measure of representational fidelity; these judgments can be directly related to the heuristic function value in HS. Finally, we also do not know the generality of the model class defined broadly as heuristic search as suitable representations for other strategy games. Parallel future work demonstrating whether heuristic search is the best performing model for a variety of games with varying rules and geometries will be essential to fully understand how heuristic search relates to human cognition.

Studying cognition in strategy games might appear abstract and esoteric, but combinatorial gameplay cleanly exposes some of the essential elements for more general reasoning and decision making, such as value judgements and the anticipation of consequences. In general, on any given day a person will plan sequences of actions many times, whether for mundane activities like ordering a set of errands or for sophisticated decisions like military strategy. Strategy games are an idealized environment in which the processes underlying complicated decision making in natural environments are isolated without being reduced away entirely. Consequently, our work applies not only to replicating human gameplay, but to furthering our understanding of the fundamentals of human reasoning by establishing a new experimental paradigm for the study of sequential decision making and procedural rationality.

Our heuristic search model is tied for the best performance on our predictive fidelity measure, and we can generate secondary predictions from its components. We therefore take heuristic search to be the best supported general family of models for combinatorial gameplay. Importantly, our heuristic search model captures critical features of, and by its success

corroborates, the cognitive story told by Chase and Simon and de Groot's pioneering work, building on the past success of heuristic search model descriptions. Most importantly, heuristic search's success as a model of reasoning in combinatorial games provides evidence that heuristic search plays an important role in more general decision making.

Acknowledgements

I would like to sincerely thank Dr. Wei Ji Ma, Bas van Opheusden, and Dr. Zahy Bnaya for their support and respective contributions. Particularly, the implementations of various heuristic search, treeless, and Monte Carlo tree search models are the work of Bas and Zahy. Bas additionally contributed Figures S1 and S2 describing the heuristic search model.

References

Abramson, B., & Korf, R. E. (1987). A model of two player evaluation functions. Proceedings from AAAI '87 : *The Sixth National Conference on Artificial Intelligence*. (Vol. 1, pp. 90-94). AAAI Press.

Al-Rfou, R., Alain, G., Almahairi, A., & Angermueller, C., Bahdanau, D., Ballas, N., Bastien, F….Zhang, Y. (2016). Theano: A Python framework for fast computation of mathematical expressions. *arXiv eprints abs/1605.02688 (May 2016). url: http://arxiv. org/abs/1605.02688*.

Allis, L. V. (1994). *Searching for solutions in games and artificial intelligence* (Doctoral dissertation). Maastricht University, Maastrict, Netherlands.

Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3), 235-256.

Berlekamp, E., J. H. Conway, et al. (1982). *Winning Ways for your Mathematical Plays.* (Vol. 1). Cambridge, MA: Academic Press.

Berliner, H., & Ebeling, C. (1989). Pattern knowledge and search: The SUPREM architecture. *Artificial Intelligence*, 38(2), 161-198.

Brading, K., & Landry, E. (2006). Scientific structuralism: Presentation and representation. *Philosophy of Science*, 73(5), 571-581.

Bowers, J. S., & Davis, C. J. (2012). Bayesian just-so stories in psychology and neuroscience. *Psychological Bulletin*, 138(3), 389.

Bowers, J. S., & Davis, C. J. (2012). Is that what Bayesians believe? Reply to Griffiths, Chater, Norris, and Pouget (2012). *Psychological Bulletin*, 138(3), 423-426.

Cartwright, N. (1983). *How the laws of physics lie*. Oxford, United Kingdom: Oxford University Press.

Charness, N., Reingold, E. M., Pomplun, M., & Stampe, D. M. (2001). The perceptual aspect of skilled performance in chess: Evidence from eye movements. *Memory & cognition*, *29*(8), 1146-1152.

Chase, W. G., & Simon, H. A. (1973). Perception in chess. *Cognitive psychology*, *4*(1), 55-81.

Clark, C., & Storkey, A. (2014). Teaching deep convolutional neural networks to play go. *arXiv preprint arXiv:1412.3409*.

Culberson, J. C., & Schaeffer, J. (1998). Pattern databases. *Computational Intelligence*, *14*(3), 318-334.

da Costa, N. C., & French, S. (1990). The model-theoretic approach in the philosophy of science. *Philosophy of Science*, 248-265.

de Groot, A. D. (1978). *Thought and choice in chess (Revised translation of de Groot, 1946)*. The Hague: Mouton Publishers.

Devijver, P. A., & Kittler, J. (1982). *Pattern recognition: A statistical approach.* Upper Saddle River, NJ: Prentice Hall.

Enzenberger, M., Müller, M., Arneson, B., & Segal, R. (2010). Fuego—an open-source framework for board games and Go engine based on Monte Carlo tree search. *IEEE Transactions on Computational Intelligence and AI in Games, 2*(4), 259-270.

Frigg, R. (2006). Scientific representation and the semantic view of theories. *Theoria*, 21(55), 49-65.

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2014). *Bayesian data analysis* (Vol. 2). Boca Raton, FL: Chapman & Hall/CRC.

Gelman, A., Hwang, J., & Vehtari, A. (2014). Understanding predictive information criteria for Bayesian models. *Statistics and Computing*, 24(6), 997-1016.

Gelman, A., & Shalizi, C. (2012). Philosophy and the practice of Bayesian statistics in the social sciences. In Kincaid, H. (Eds.), *Oxford handbook of the philosophy of the social sciences* (259-273). Oxford, United Kingdom: Oxford University Press.

Gelman, A., & Shalizi, C. R. (2013). Philosophy and the practice of Bayesian statistics. T*he British Journal of Mathematical and Statistical Psychology*, 66(1), 8–38. http://doi.org/10.1111/j.2044-8317.2011.02037.x

Giles, R. C. S. L. L. (2001). Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping. Proceedings from: *Advances in Neural Information Processing Systems 13* (Vol. 13, pp. 402). Cambridge, MA: MIT Press.

Gobet, F., & Charness, N. (2006). Chess and games. *Cambridge handbook on expertise and expert performance* (pp. 523-538). Cambridge, United Kingdom: Cambridge University Press.

Gobet, F., & Simon, H. A. (1998). Pattern recognition makes search possible: Comments on Holding (1992). *Psychological Research*, *61*(3), 204-208.

Gobet, F., & Simon, H. A. (1998). Expert chess memory: Revisiting the chunking hypothesis. *Memory,* 6(3), 225-255.

Goodman, N. D., Frank, M. C., Griffiths, T. L., Tenenbaum, J. B., Battaglia, P. W., & Hamrick, J. B. (2015). Relevant and robust: a response to Marcus and Davis (2013). *Psychological Science*, 26(4), 539-541.

Griffiths, T. L., Chater, N., Norris, D., & Pouget, A. (2012). How the Bayesians got their beliefs (and what those beliefs actually are): Comment on Bowers and Davis (2012). Psychological Bulletin, 138(3), 415-422.

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2), 100-107.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Huyer, W., & Neumaier, A. (1999). Global optimization by multilevel coordinate search. *Journal of Global Optimization*, *14*(4), 331-355.

Jones, M., & Love, B. C. (2011). Bayesian fundamentalism or enlightenment? On the explanatory status and theoretical contributions of Bayesian models of cognition. *Behavioral and Brain Sciences*, 34(04), 169-188.

Kocsis, L., & Szepesvári, C. (2006). Bandit based Monte-Carlo planning. Proceedings from: *European Conference on Machine Learning* (pp. 282-293). Heidelberg, Germany: Springer Berlin Heidelberg.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.

Maddison, C. J., Huang, A., Sutskever, I., & Silver, D. (2014). Move evaluation in Go using deep convolutional neural networks. *arXiv preprint arXiv:1412.6564*.

Marcus, G. F., & Davis, E. (2013). How robust are probabilistic models of higher-level cognition?. *Psychological Science*, 24(12), 2351-2360.

Marcus, G. F., & Davis, E. (2015). Still searching for principles: A response to Goodman et al. (2015). *Psychological Science*, 26(4), 542-544.

Mech, L. (2007). Possible use of foresight, understanding, and planning by wolves hunting muskoxen. *Arctic*, 60(2), 145-149.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2), 81.

Moxley, J. H., Ericsson, K. A., Charness, N., & Krampe, R. T. (2012). The role of intuition and deliberative thinking in experts' superior tactical decision-making. *Cognition*, *124*(1), 72-78.

Murphy, K. P. (2012). Machine learning: A probabilistic perspective. Cambridge, MA: MIT Press.

Popper, K. (1959). *The logic of scientific discovery*. Abingdon-On-Thames, United Kingdom: Routledge.

Romain, P. (2000). "Les représentations des jeux de pions dans le Proche-Orient ancien et leur signification." *Board Game Studies*, 3: 11-38.

Russell, S., & Norvig, P. (2009). *Artificial intelligence: A modern approach* (3rd ed.). New York City, NY: Pearson.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1988). Learning representations by back-propagating errors. *Cognitive Modeling*, 5(3), 1.

Shannon, C. E. (1950). "Programming a computer for playing chess." *Philosophical Magazine*, 41(314).

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Dieleman, S. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529*(7587), 484-489.

Simon, H. A., & Gilmartin, K. (1973). A simulation of memory for chess positions. *Cognitive Psychology*, 5(1), 29-46.

Simon, H. A., & Schaeffer, J. (1990). The game of chess. In Aumann, R.J, & Hart, S. (Eds.) *Handbook of game theory with economic applications* (Vol. 1, pp. 1-17). Amsterdam, Netherlands: Elsevier.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1), 1929-1958.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society*. *Series B (Methodological)*, 111-147.

Stone, M. (1977). An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion. *Journal of the Royal Statistical Society. Series B (Methodological)*, 44-47.

Sutskever, I., Martens, J., Dahl, G. E., & Hinton, G. E. (2013). On the importance of initialization and momentum in deep learning. *International Conference on Machine Learning*, (3), 28, 1139-1147.

Suppe, F. (1989). *The semantic conception of theories and scientific realism*. Champaign, IL: University of Illinois Press.

Suppe, F. (2000). Understanding scientific theories: An assessment of developments, 1969-1998. *Philosophy of Science*, S102-S115.

Suppes, P. (1961). A comparison of the meaning and uses of models in mathematics and the empirical sciences. In Freudenthal, H. (Eds.), *The concept and the role of the model in mathematics and natural and social sciences: Proceedings of the colloquium sponsored by the Division of Philosophy of Sciences of the International Union of History and Philosophy of Sciences organized at Utrecht, January 1960* (Vol. 3, pp. 163-177). Amsterdam, Netherlands: Springer Netherlands.

Suppes, P. (1967). What is a scientific theory?. In Morgenbesser, S. (Eds.), *Philosophy of science today* (pp. 55-67). New York City, NY: Basic Books

Uiterwijk, J. W. H. M., & Van den Herik, H. J. (2000). The advantage of the initiative. *Information Sciences*, *122*(1), 43-58.

van Fraassen, B. C. (1980). *The scientific image*. Oxford: Oxford University Press.

van Fraassen, B. C. (2008). *Scientific representation: Paradoxes of perspective*. Oxford: Oxford University Press.

van den Herik, H. J., Uiterwijk, J. W., & Van Rijswijck, J. (2002). Games solved: Now and in the future. *Artificial Intelligence*, *134*(1), 277-311.

von Neumann, J. and O. Morgenstern (1944). *Theory of games and economic behavior*. Princeton, NJ: Princeton University Press.

Quine, W. V. (1951). Main trends in recent philosophy: Two dogmas of empiricism. *The Philosophical Review*, 20-43.

Wu, I. C., & Huang, D. Y. (2005). A New Family of k-in-a-row Games. *Advances in Computer Games* (pp. 180-194). Heidelberg, Germany: Springer Berlin Heidelberg.

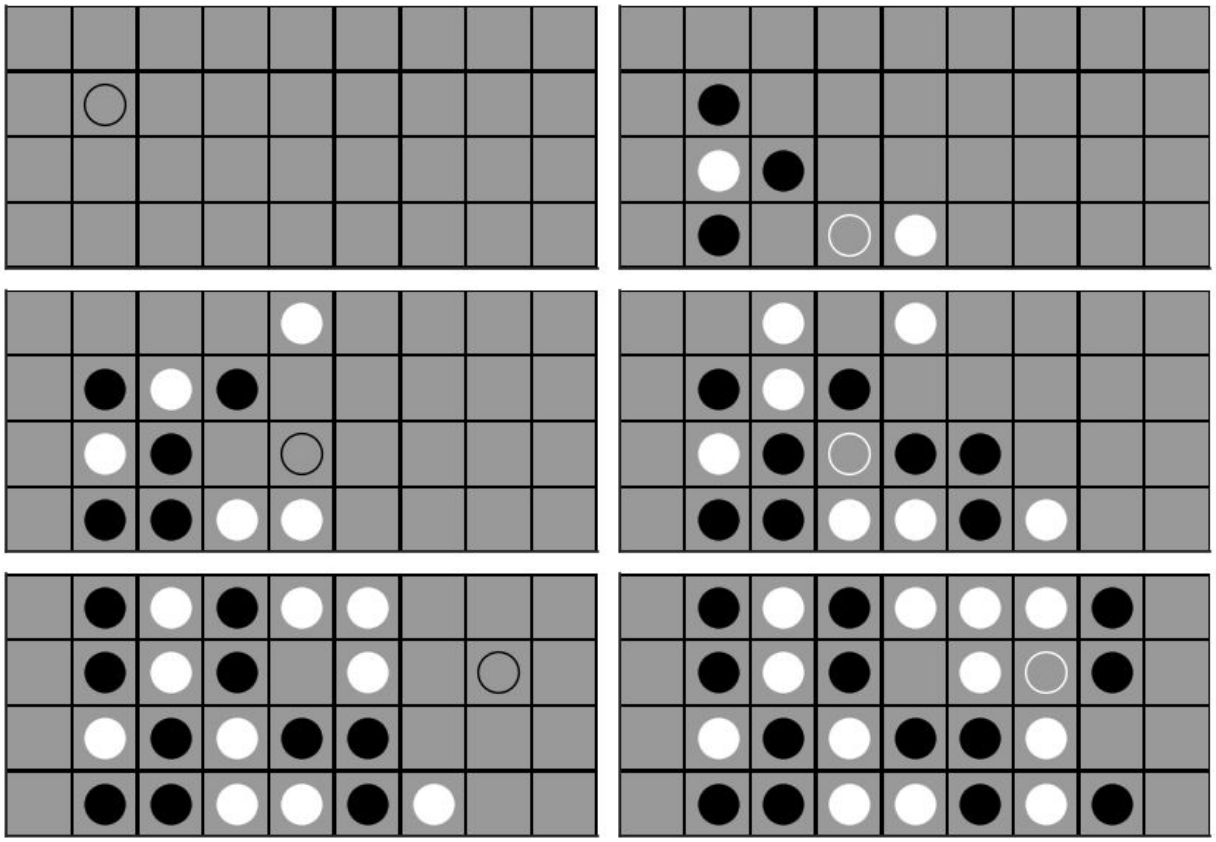

Figure 1: Example positions from a game of (4, 9, 4). The move made by the current player is indicated by an outline of the corresponding color. In the last board at the bottom right, White wins by making a vertical four-in-a-row.

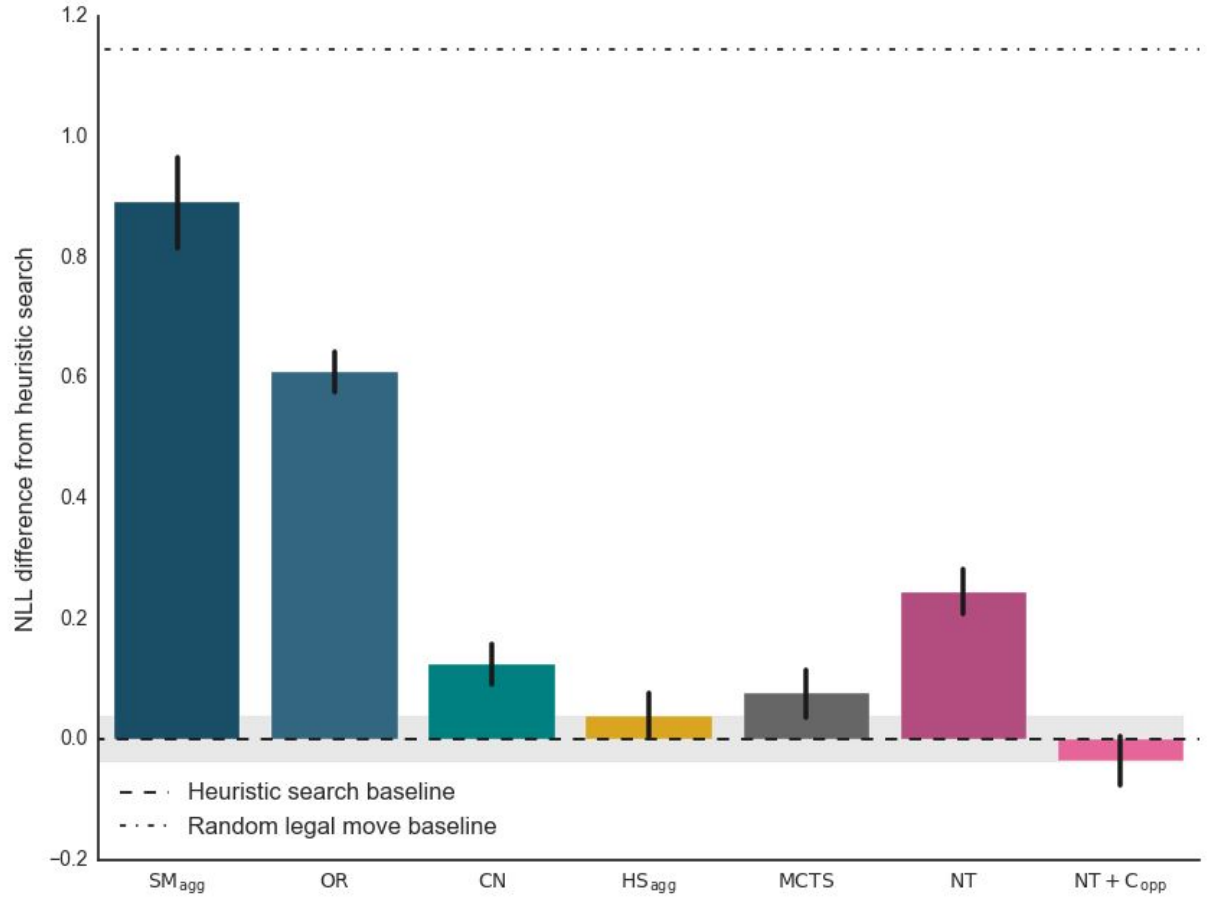

Figure 2: Comparison of difference in model performance between HS and other models. SM_{agg} is the softmax model trained on aggregate data. OR is the optimal-random mixture model. CN is the convolutional neural network. MCTS is the Monte Carlo tree search model. NT is the heuristic function from the heuristic search model. NT + C_{opp} is NT with an additional free parameter that scales the value of the opponent's pieces in the heuristic function. HS_{agg} is HS trained on data aggregated across subjects.

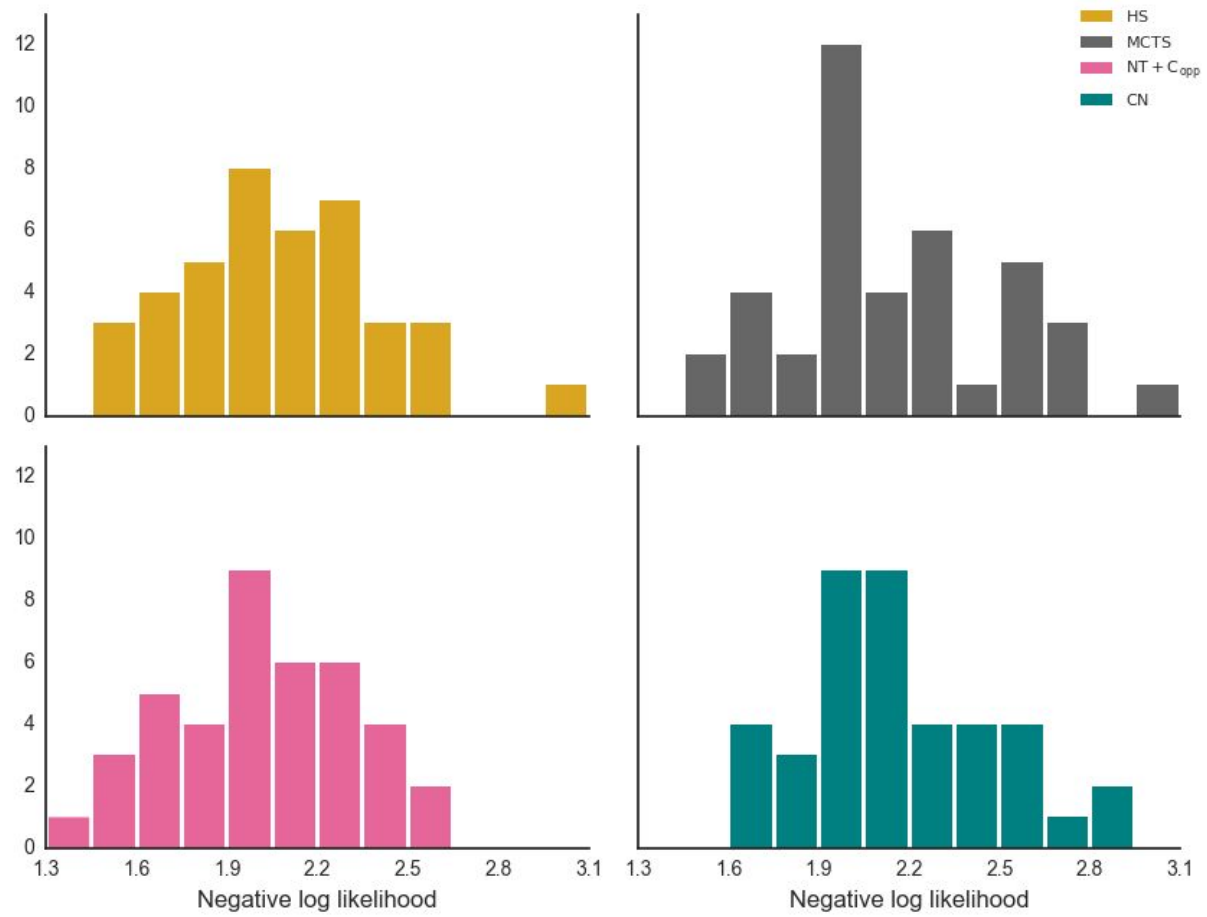

Figure 3: Histograms of the average negative log likelihood per subject for the four competitive models.

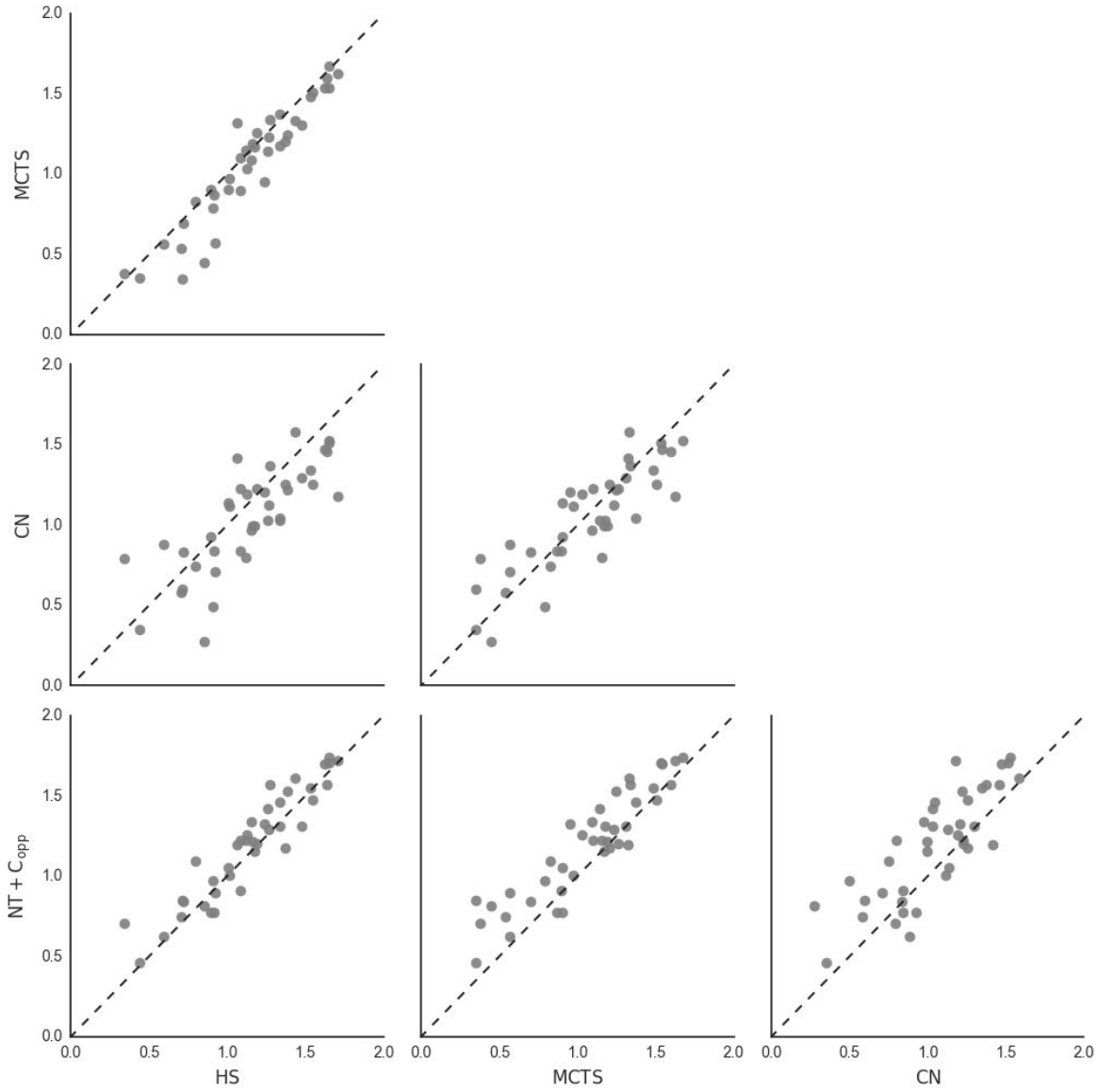

Figure 4: Pairwise comparison for all competitive models. The axes are negative log likelihood relative to chance, with higher values indicating a better fit. Each dot is one subject's average fit.

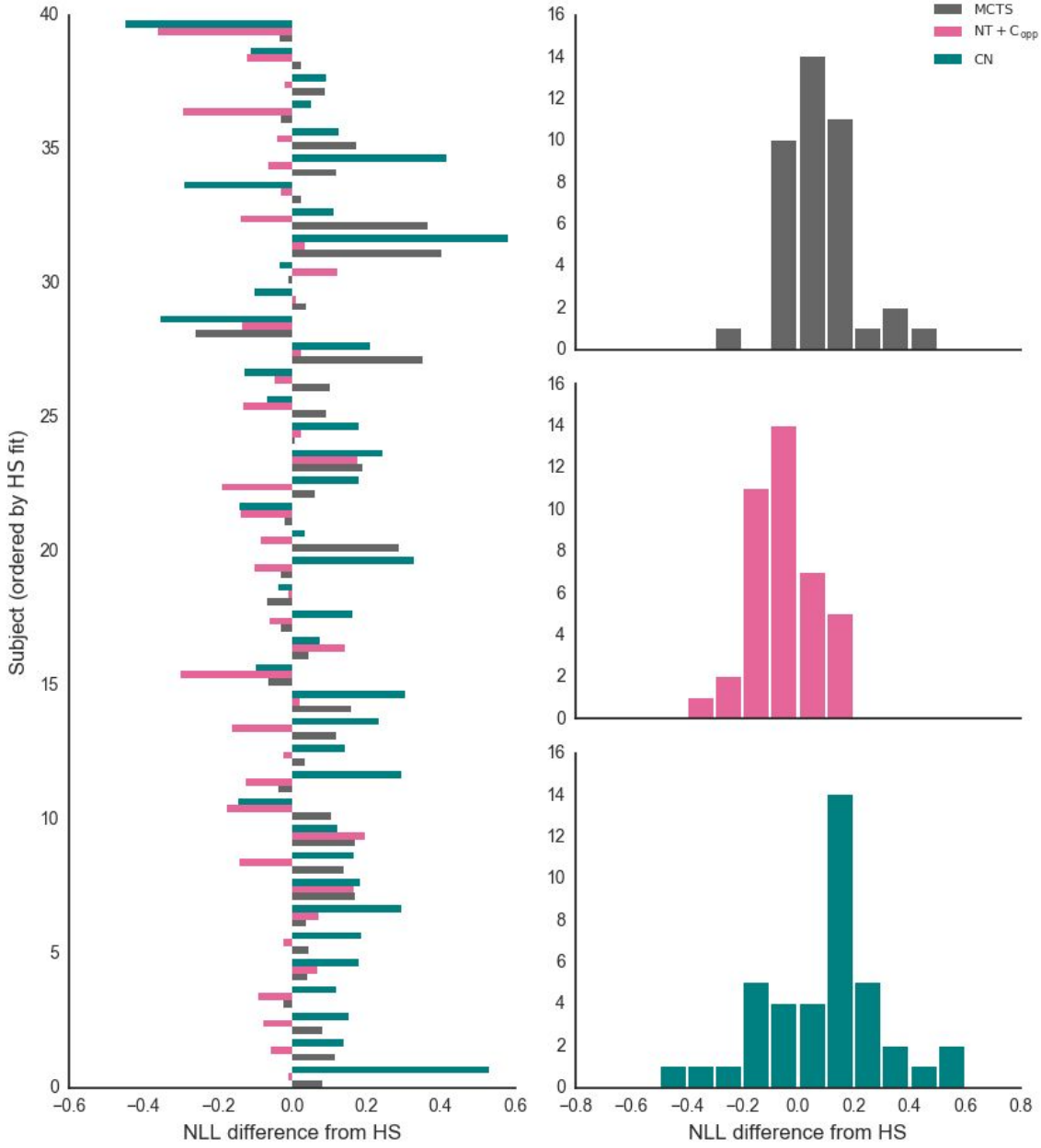

Figure 5: Distributions of negative log likelihood difference from HS for MCTS, and NT + C_{opp}, and CN. On the left is the negative log likelihood (NLL) difference from HS for each combination of subject and model, ordered by HS NLL. On the right are histograms of the difference in NLL from HS. Lower values indicate better performance relative to HS.

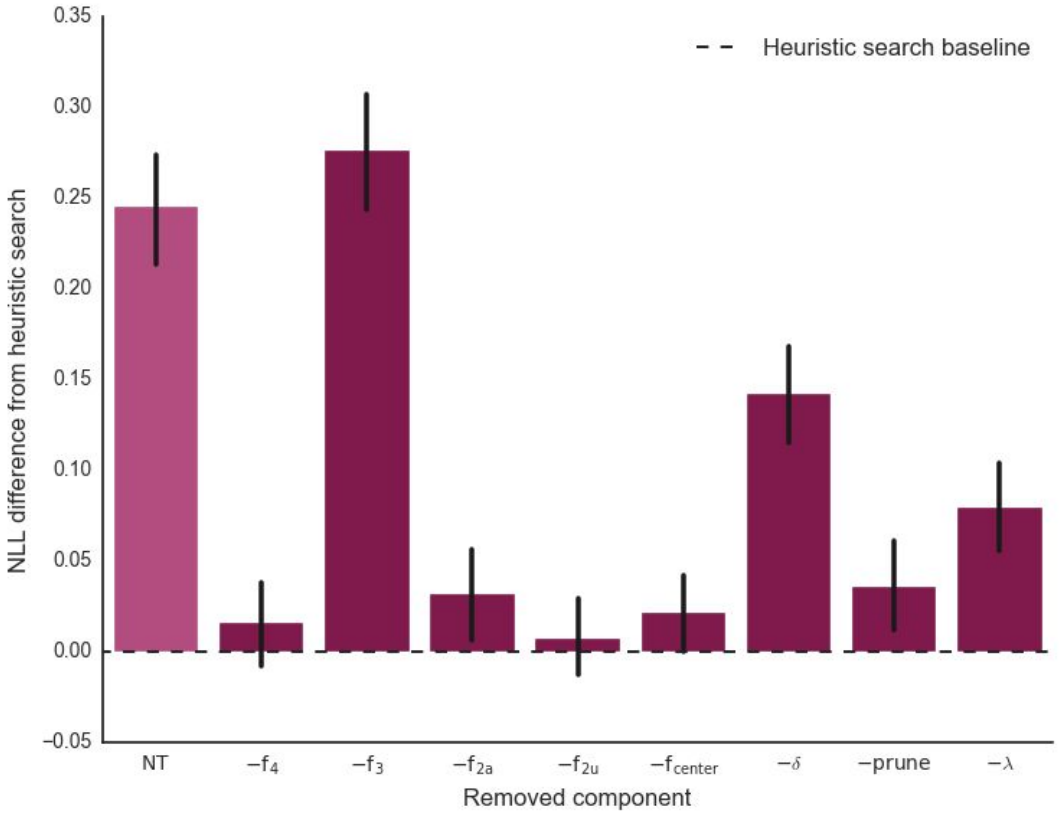

Figure 6: Lesioning analysis results from removing each HS component. NT is HS's heuristic function with no tree building. The -f_x models are removals of features from the heuristic function: 4-in-a-row, 3-in-a-row, 2 adjacent, distance to center, feature drop rate, tree branch pruning factor, and lapse rate, respectively.

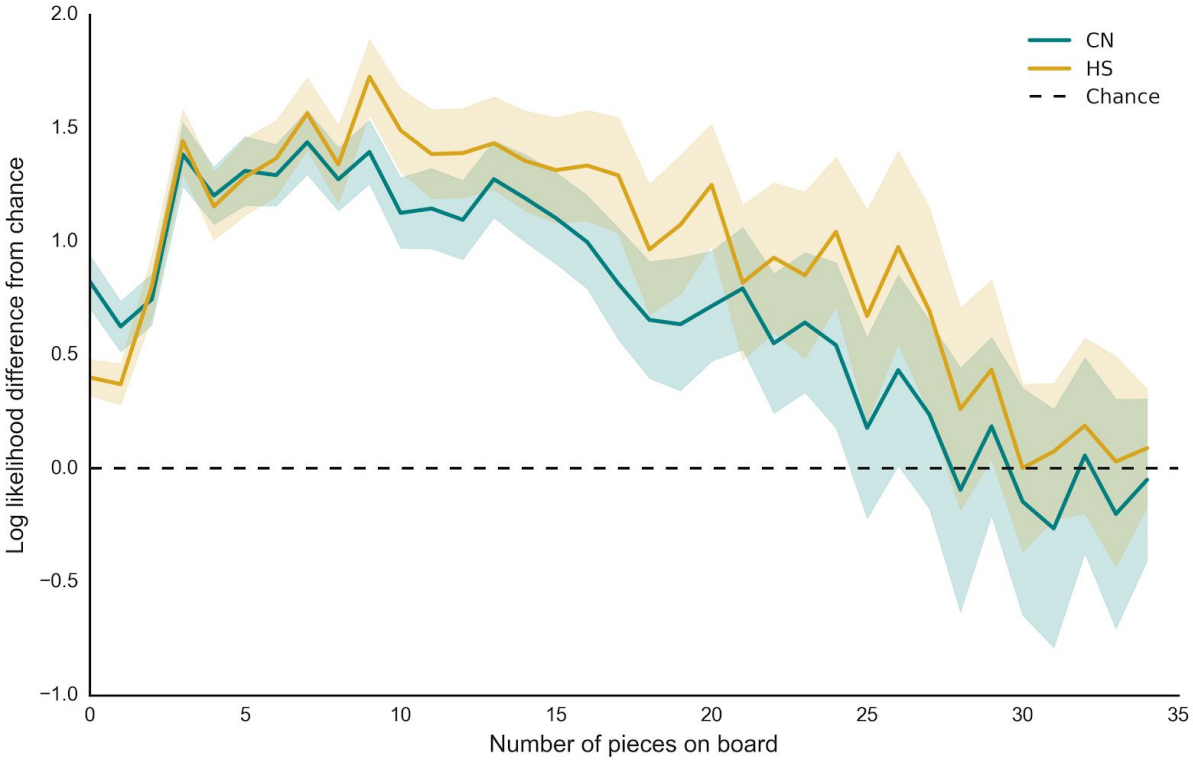

Figure 7: Mean log-likelihood relative to chance by number of moves played. A higher value indicates a better performance.

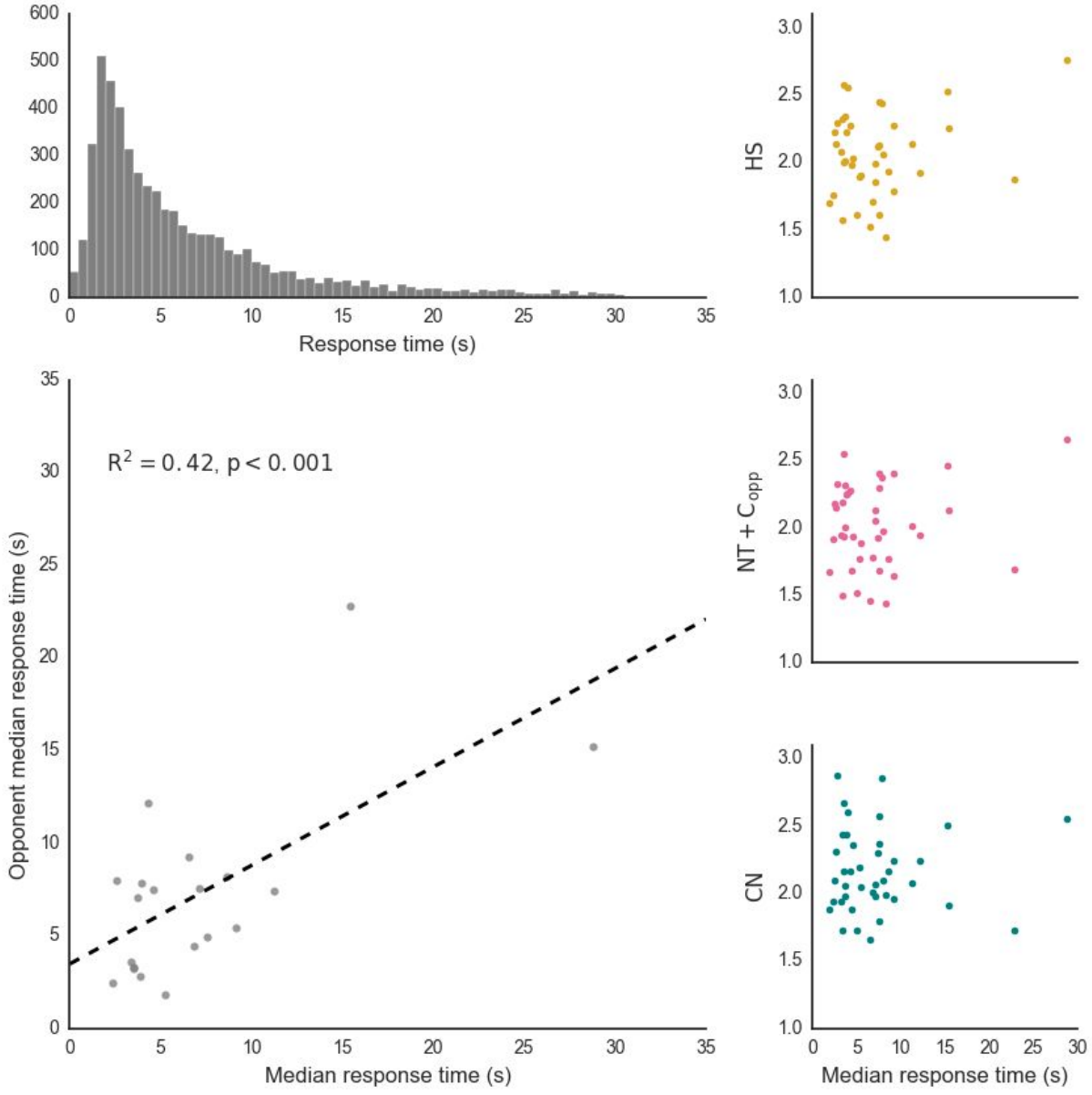

Figure 8: Top left: histogram of response times for entire population. Bottom left: correlation between median response times for pairs of players. Right: scatter plots of median response times and model log likelihoods for each player for CN, NT + C_{opp}, and HS.

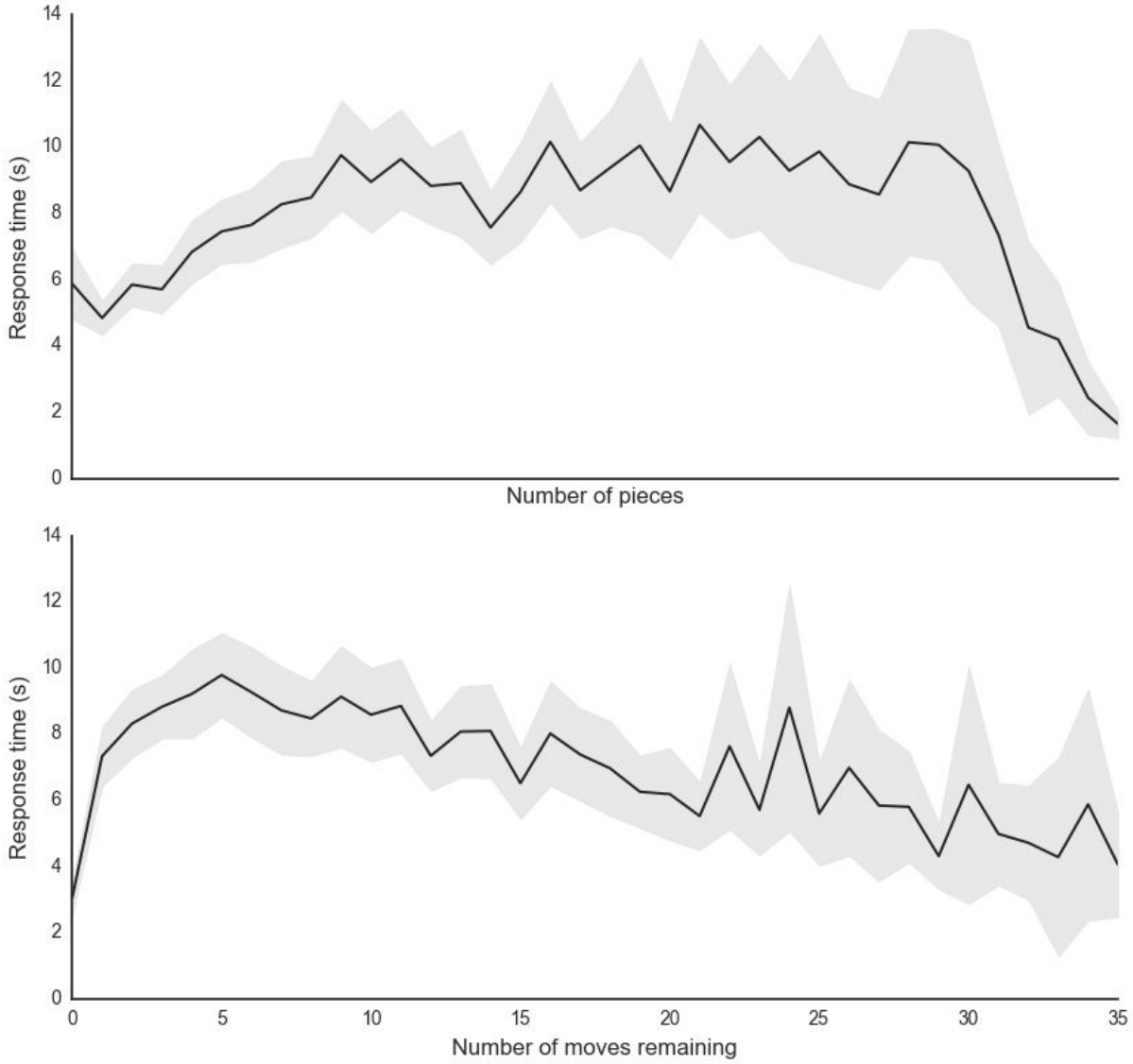

Figure 9: Mean response time as a function of number of pieces on the board (top) and number of moves left in the current game (bottom).

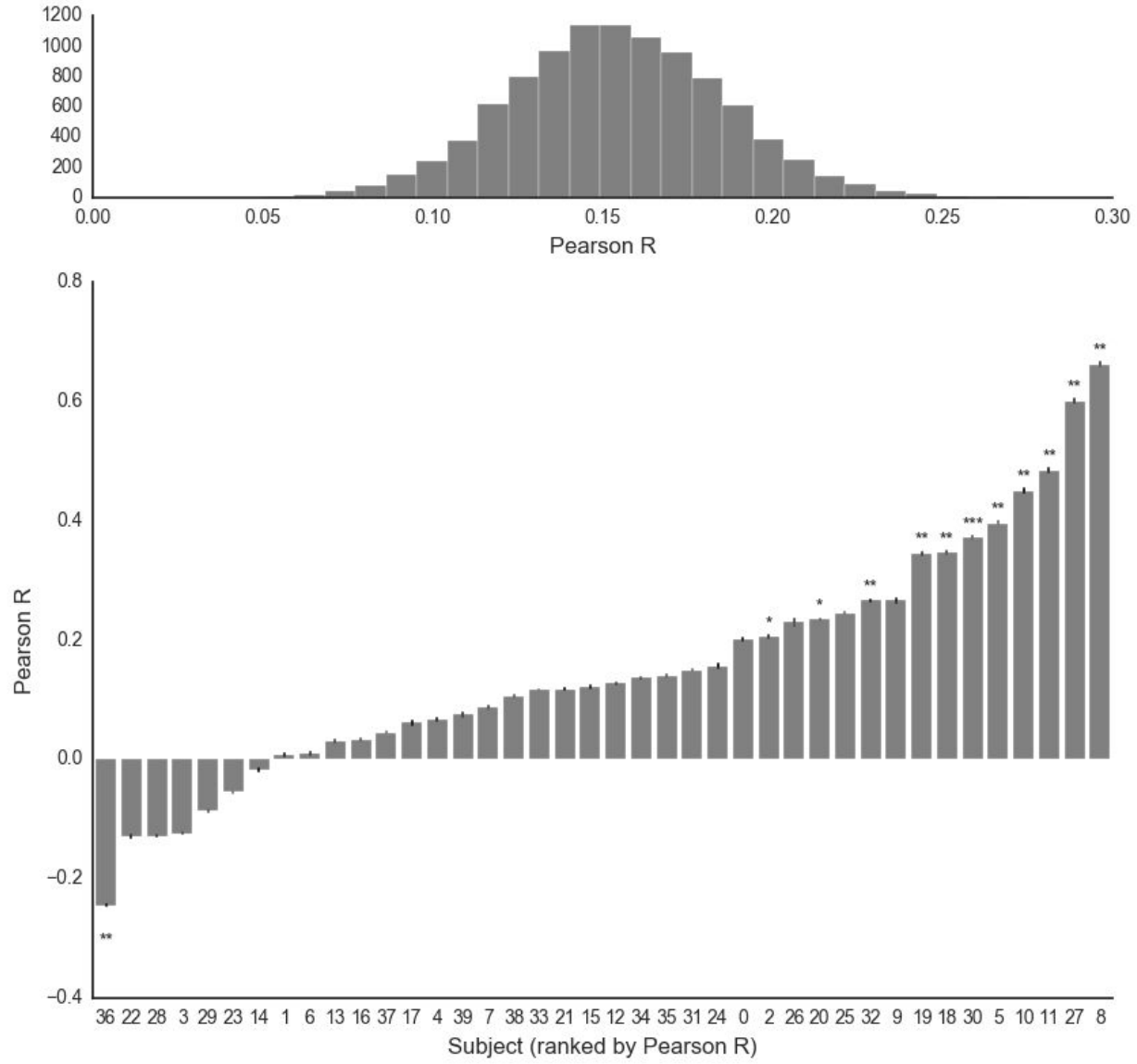

Figure 10: Correlation between the log tree size built by HS and the response time of subjects. Top, the histogram of the bootstrapped mean of subject-by-subject correlations between the log of response time and the log of tree size built by HS. Bottom, the mean of the bootstrapped correlation per subject. Asterisks indicate an original correlation p value < .05, .01, and .001 respectively.

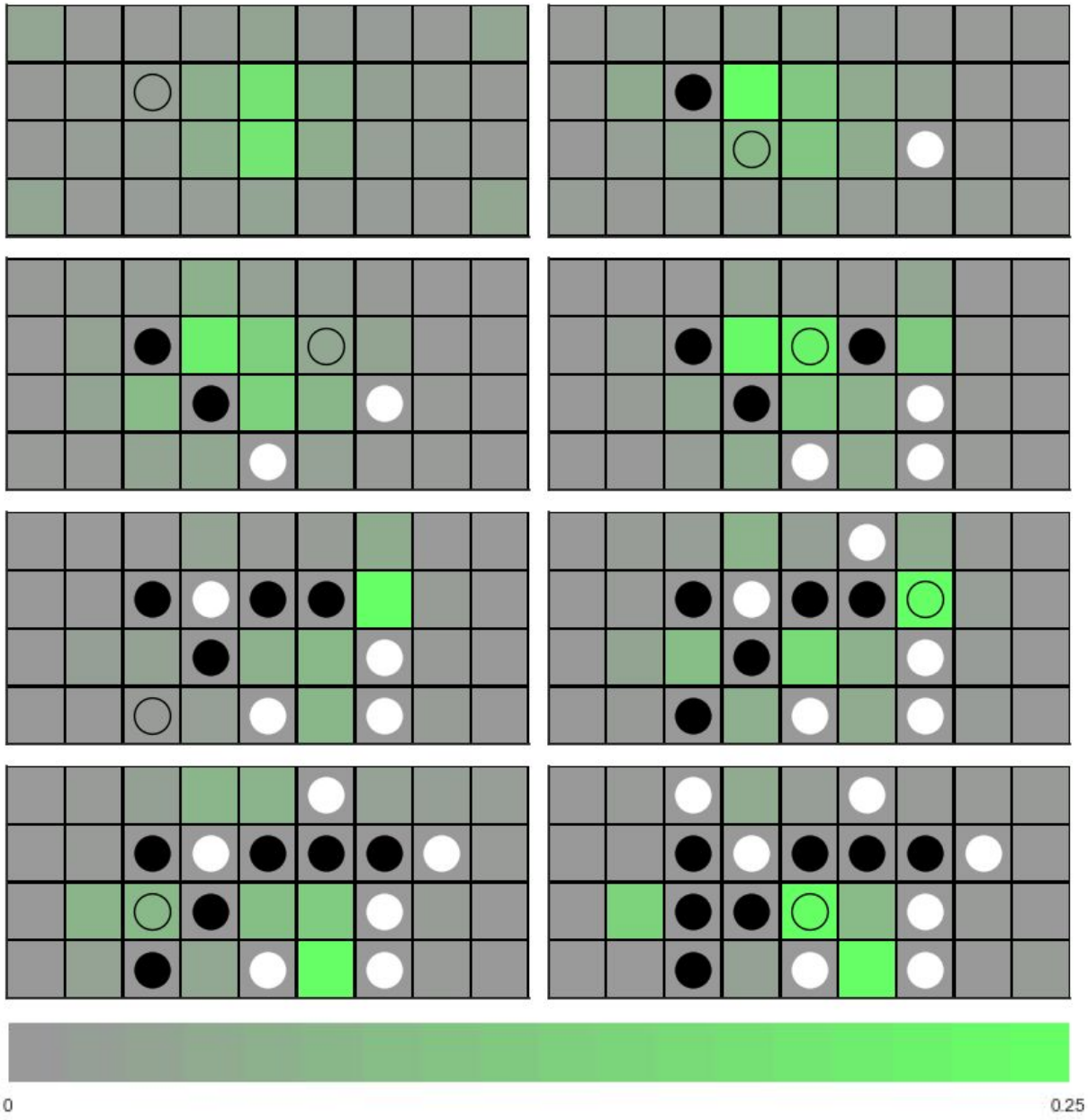

Figure 11: Example predictions for Black by the convolutional network model. The move the subject actually makes is indicated by an outline. The more probability assigned by the model to a location, the brighter the green.

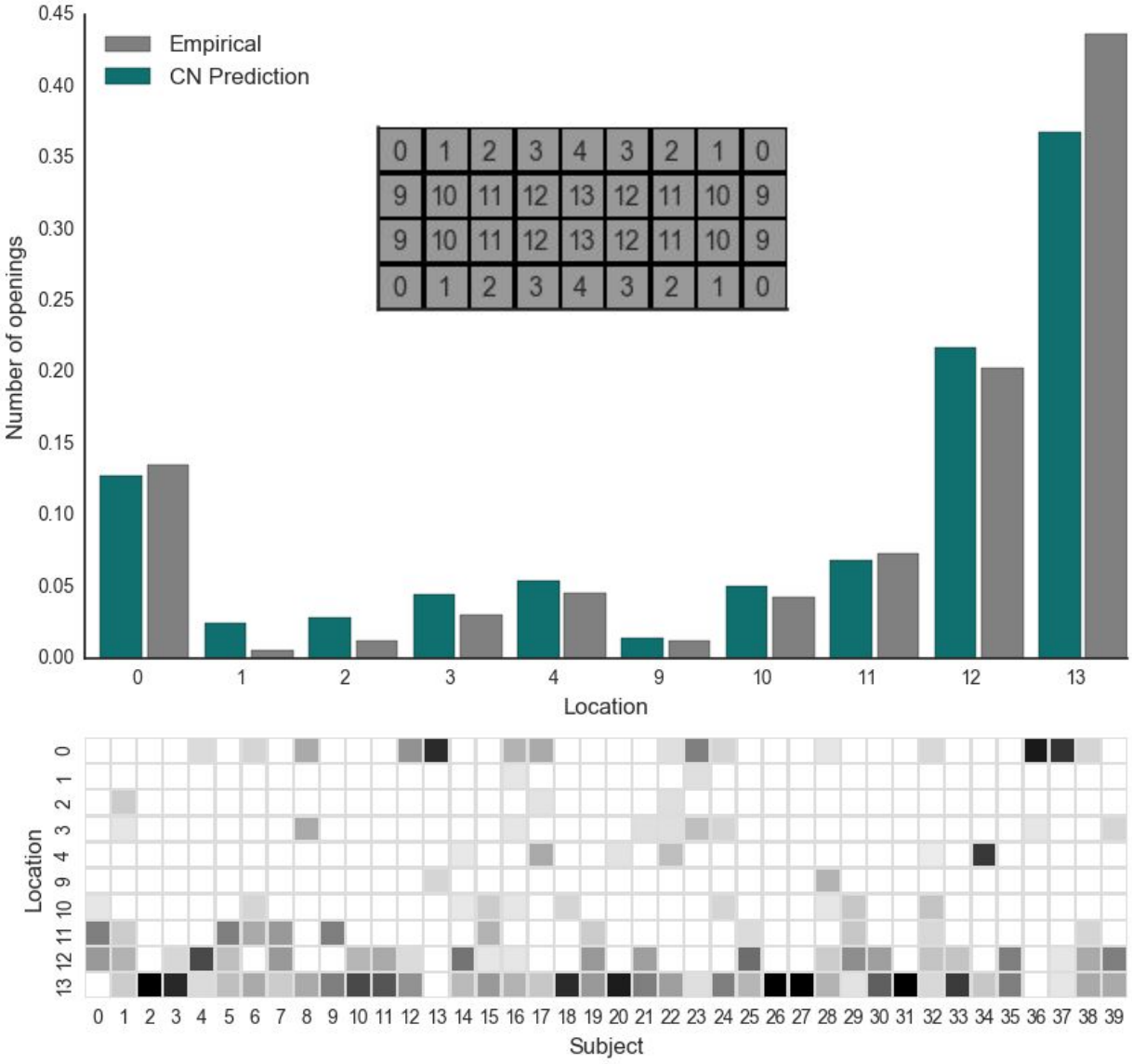

Figure 12: Above is empirical distribution of symmetrized opening moves, compared with the prediction from CN (see inset). Location 0 includes the four corners, and location 13 includes the two centermost squares. Below is a normalized 2D histogram of each human player's opening preference; black indicates 100% of openings in the corresponding location.

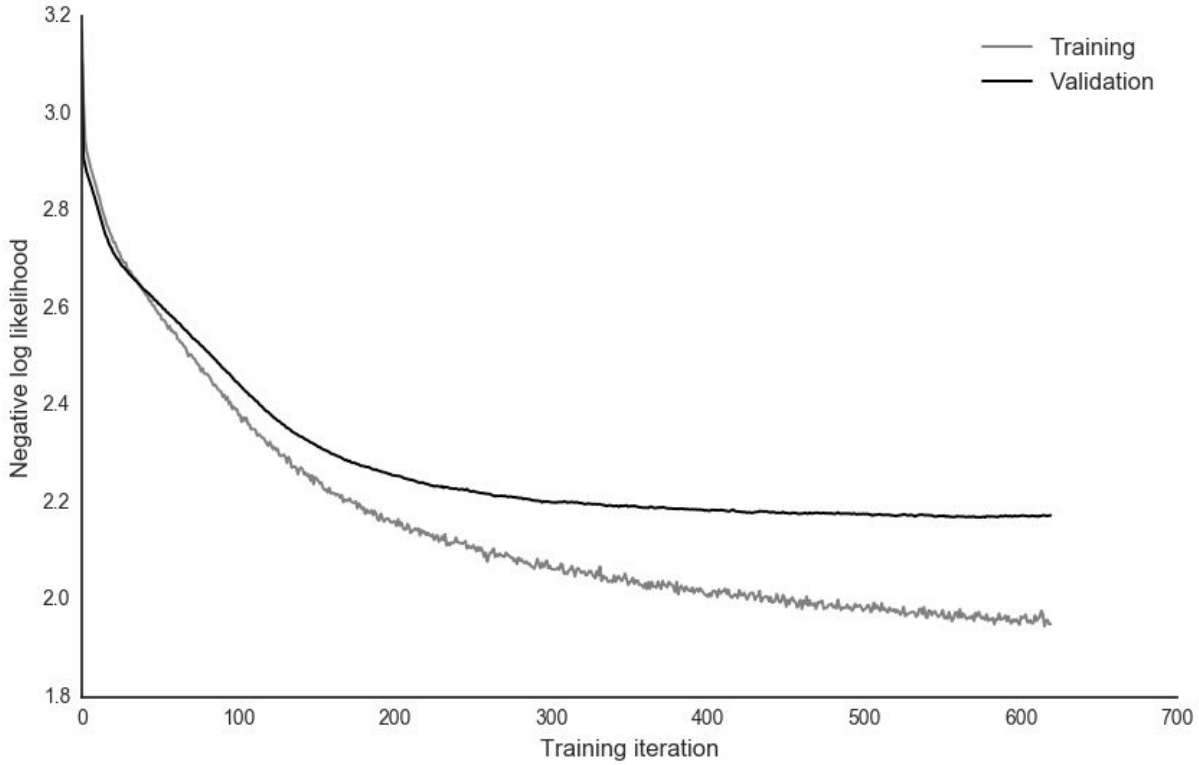

Figure 13: The training set and validation set performance traces from training CN on the fourth cross-validation split.

Supplement

Models

Heuristic Search. The heuristic search model (HS) works by iteratively selecting moves, simulating their independent addition to the board ("expanding a node"), evaluating the resulting position, and "backpropagating" the values by updating the values of previously visited positions accordingly (Figure S1). The evaluation step calls a heuristic function that takes a weighted sum of features, or configurations of pieces present on the board, and returns a single value:

$$H(s) = \sum_i w_i f_i(s_{self}) - \sum_i w_i f_i(s_{opponent}) + N(0, 1)$$

where s is a game position, s_{self} and $s_{opponent}$ respectively are the arrangement of a player's own pieces and that of their opponent, f_i is the number of occurrences of feature i in s, and w_i is the corresponding weight for that feature. The features are four in a row (a win), three in a row of four with one open space (commonly called a "threat"), two adjacent in a row of four with two open spaces, two nonadjacent in a row of four with two open spaces, and a "center" feature that counts the number of pieces in the three centermost columns of the game board (Figure S2). The first four features can be in any orientation (horizontal, vertical, or diagonal). Together they are an exhaustive combination of $n \in [2, 4]$ of a player's pieces in a four by one section of the game board.

The heuristic value is then backpropagated to the previous position according to a minimax rule. If the position was the player's own position, then the maximum value of all direct children of its parent is backpropagated; if the position was the opponent's, then the minimum value is backpropagated (Figure S1). The model then selects a new unexplored position by

following a path of maximum value for the current player and minimum value for the opponent to a previously unexpanded node.

There are a total of 10 free parameters in the heuristic search model. As previously discussed, there is one parameter w_i for each feature weight, for 5 free parameters in the heuristic function. Another parameter in the heuristic function is the feature drop rate δ. The feature drop rate is the probability with which an instance of a feature f_i is not counted and represents subjects failing to notice the presence of of a feature at some location.

There is an additional lapse rate parameter λ, which is a probability with which the model builds no tree and instead randomly selects a move from a uniform distribution over all legal moves. After expanding a node, a "pruning" module removes further branches of the decision tree for which the heuristic value is below the maximum value less a threshold parameter t_p. Pruning is representative of subjects ignoring some options when those options appear to be relatively unpromising.

Finally, there are two parameters governing stopping conditions, or when the model stops iteratively adding and pruning nodes from a tree and makes a move based on the maximum value of explored moves. The first is a random early stop parameter γ which is the probability with which the model stops updating the game tree after each backpropagation. The second is a threshold t_s; when the difference between the values of the best and second best move exceeds the stopping threshold, the model stops iterating and selects the best move.

Monte Carlo Tree Search. The second model implements a parameterized Monte Carlo tree search (MCTS) algorithm, based on the UCT algorithm (Kocsis & Szepesvári, 2006). The MCTS model simulates a number of random chains of moves through the end of a game,

backpropagates the number of wins and the number of visits to each node, then selects a new move to explore based on the ratio of wins to visits for each available move. The exact expression for the selection is the UCB1 policy (Auer, Cesa-Bianchi, & Fischer, 2002):

$$argmax_{m \in M} V(m) + c\sqrt{\frac{log(N(p))}{N(p_m)}}$$

where m is a move from the set of legal moves M at position p, $V(m)$ is the average value of game ends reached from move m, p_m is the position that results from move m, $N(p)$ is the number of visits to position p, and c is a free parameter called the exploration coefficient. The exploration coefficient loosely represents how inclined a player is to explore new moves that they had not previously considered. After selecting a previously unexplored node, the algorithm performs a random "rollout" by simulating random moves until it reaches a won, drawn, or lost position. It then backpropagates the game theoretical value from that position (1, 0, and -1, respectively) through all parent nodes. After a fixed number of rollouts, the model makes the move corresponding the highest move with the highest value.

Convolutional Neural Network. A typical convolutional neural network is composed of several layers: an input layer, which takes in image-like data with each pixel being represented by one node, or element in a tensor; hidden layers, which perform some operation on the value of each node in the previous layer, and an output layer, which typically has one node for each of a number of possible classification label (LeCun, Bengio, & Hinton, 2015). Hidden layers are typically of one of two primary types: a convolutional layer, or a fully connected layer. Convolutional layers iteratively multiply sections of their input with each of a number of filters, producing a vector of responses for each filter at each location. Fully connected layers perform a

vector product between their weight parameters and their inputs. A fairly minimal structure for a convolutional neural network is then an input layer, a single convolutional layer, and a single fully connected output layer. This minimal layer structure is what we use for our model CN.

We treat our inputs as a 9 by 4 image (the game board), which we represent as a rank 3 tensor with dimensionality (2, 4, 9) - there are two channels in the 4 by 9 image, one for one's own pieces, and one for the opponent's pieces. The convolutional layer we use contains 32 filters of dimensionality (2, 4, 4). Each filter is iteratively moved across the input image, producing a tensor of filter responses. We then apply a rectified linear function to the filter responses, and then pass to a fully connected output layer of 36 units, one for each location on the game board. We subsequently apply a softmax function to the output layer to convert the class label prediction into a probability distribution over possible moves. Finally, we filter the output by the input image and renormalize the distribution to prevent assignment of any probability to illegal moves.

After exploring a number of different layer structures, which we do not report, we eventually settled on the above architecture. To decide the number of filters to use in our architecture and to measure the effect of including the legal move filter, we trained each of the two architectures with 9 different numbers of filters, from 1 to 256 by powers of 2 (Figure S4). We chose our architecture first according to best fit and second according to lowest number of filters.

Control Models. A fourth model uses the heuristic function from the heuristic search model but does not build a game tree at all and just selects the most promising move according to the heuristic function's valuations. This "no tree" model (NT) allows us to comparatively

evaluate the importance of building a game tree to our HS model. The fifth model, NT + C_{opp}, is NT with an additional parameter that scales the value of the heuristic function for the opponent's position. The sixth model is a mixture model (OR) that either makes the best available move or a random move drawn from a uniform distribution over legal moves. The final model is a softmax model (SM) that estimates a weight parameter for every location on the board, returning a normalized distribution over available moves. SM is entirely a control model that essentially represents subjects as having independent relative preferences for every location regardless of any other features present on the board, and we do not expect it to perform well at all; its presence is to establish a baseline at which a minimally structured model performs for comparison.

Fitting Procedures

HS, MCTS, NT, NT + C_{opp}, and the lesion models are fitted using multilevel coordinate search (Huyer & Neumaier, 1999). In the five-fold cross-validation scheme, the data is divided into five groups. Each group is set aside once as a test set while the remaining four are used for training. CN is trained using gradient descent with Nesterov momentum (Sutskever, Martens, Dahl, & Hinton, 2013). SM is trained using gradient descent, and OR is computed analytically. In our analyses, we report the test set log likelihood for every data point.

In the five-fold cross-validation scheme for CN. we use five rotations each with three groups as a training set, one group as a validation set, and one group as a test set. The purpose of the validation set is for early stopping, a method that prevents overfitting by halting network training when the error on the independent validation set stops decreasing. Because early stopping essentially minimizes error on the validation set as well as the training set, the test set

remains important as a fully independent measure of the model's ability to generalize to unseen data. We use random dropout with p =.75 between the convolution and output layer during training to prevent filter coadaptation and encourage independent features (Srivistava et al., 2014).

We use the negative log likelihood instead of accuracy because it is more informative. For example, players see and respond to a blank board many times over the course of an experiment, but they may not always make the same choice. The negative log likelihood provides a principled measure of the match between the empirical distribution of a player's choices. In other words, using predictive accuracy does not allow us to answer questions about to what degree a model's prediction is incorrect for a given position. To maximize the accuracy of this prediction for a given move, the probability distribution of the model be exactly equal to the empirical distribution of a player's choices.

We report cross-validated values instead of information criteria because they are the most natural way of protecting analyses from overfitted models (Stone, 1974). Popular information-criterion methods, like Bayes or Aikake information criteria, are best used when cross-validation is not feasible. Rather than directly measuring overfitting, information criteria heuristically regularize by the number of free parameters; as such, when a model is prohibitively difficult to fit or data is excessively sparse, information criteria can be useful (Stone, 1977; Gelman, Hwang, & Vehtari, 2013). However, the number of free parameters is not the fundamental problem in constraining computational models; overfitting is, and free parameters are only *prima facie* problematic insofar as they allow for overfitting. Cross-validation directly

addresses the problem that information criteria propose to ameliorate and is the more principled choice when feasible.

The biggest practical obstacle in using a neural network as a model with this experimental data set is a lack of data. We have an average of only 137 data points per subject with a standard deviation of about 61. For perspective, most training sets for neural networks, especially in image classification, contain 10,000 to 100,000 image-label pairs. As a result, when we attempted to train the network on individual subjects, we found an average negative log-likelihood of 3.36, or close to random guessing, on the test set examples. We attempted an increase in the random dropout rate from .75 to .95, but this only reduces test error to 3.04. Because the training set error with dropout at .75 drops below 1.80 during fitting, we can conclude confidently that the mismatch is overwhelming due to overfitting, and that regularization via dropout is insufficient to move past the problem with this little data.

Statistics

The NumPy and SciPy packages for Python were used for all analyses and results figures. Error bars and bands for figures are 95% Bayesian credibility regions for the reported statistic unless otherwise indicated. Statistics for the entire subject population are calculated from the raw negative log likelihoods for every data point for HS, HS_{agg}, CN, NT, NT + C_{opp}, and lesioned models. Statistics for SM, SM_{agg}, and MCTS are reported by averaging first across subjects; log likelihoods and predictive distributions for individual observations are not currently available.

Implementations

HS, MCTS, NT, and OR, and their respective variants are all implemented in C++. SM is implemented in Python with NumPy and SciPy. CN is implemented in Python with Theano and Lasagne (Al-Rfou et al., 2012).

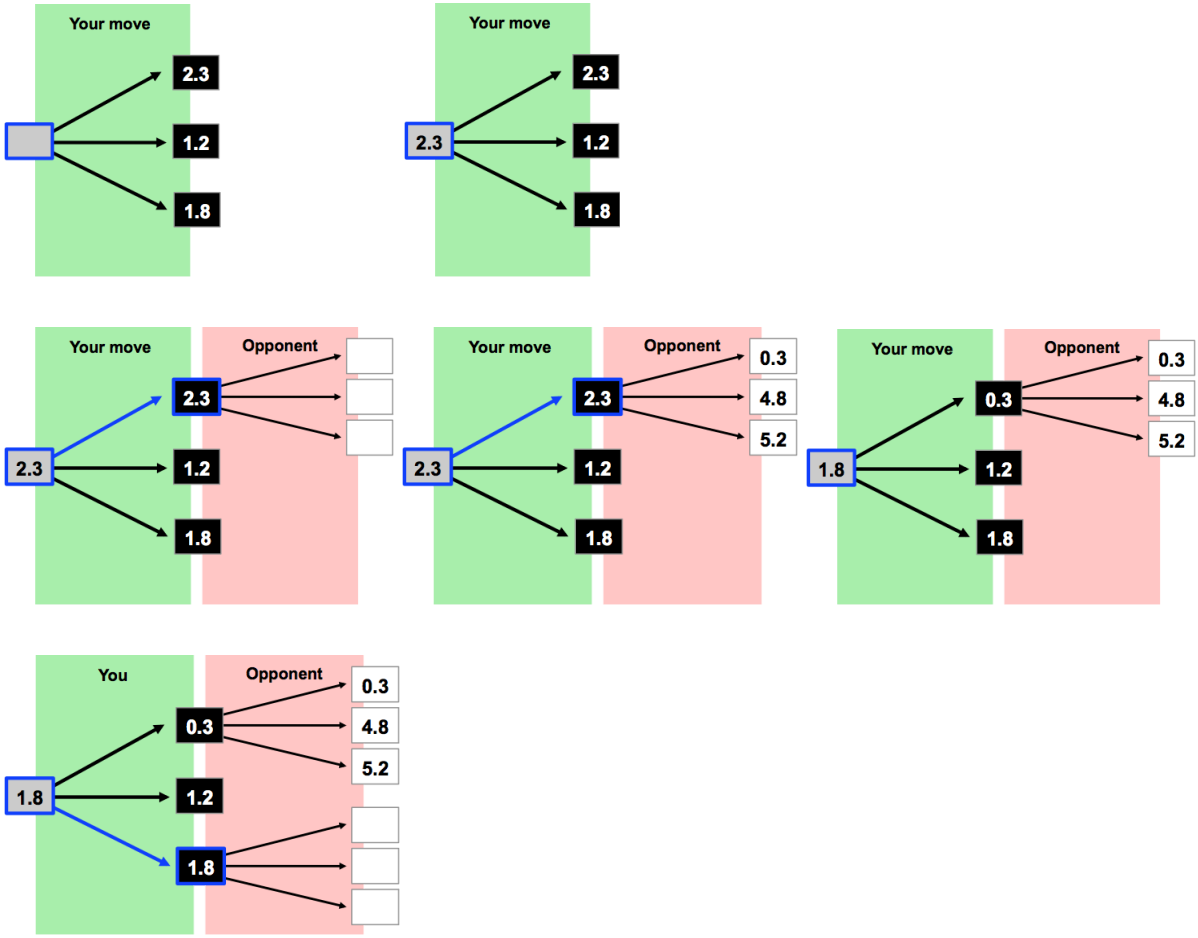

Figure S1: Demonstration of heuristic search algorithm. Each square represents a game position, and each edge represents the addition of a game piece to the board at a specific location. The grey square is the starting position (what the subject sees and the input to the model); the black squares are positions that result from a player's own moves, and the white squares are positions that result from the opponent's moves. Each row is an iteration; from left to right, the steps are *select and expand*, *evaluate*, and *backpropagate*. Note that it is the *minimum* value of an opponent's moves and the *maximum* value of one's own moves that is backpropagated.

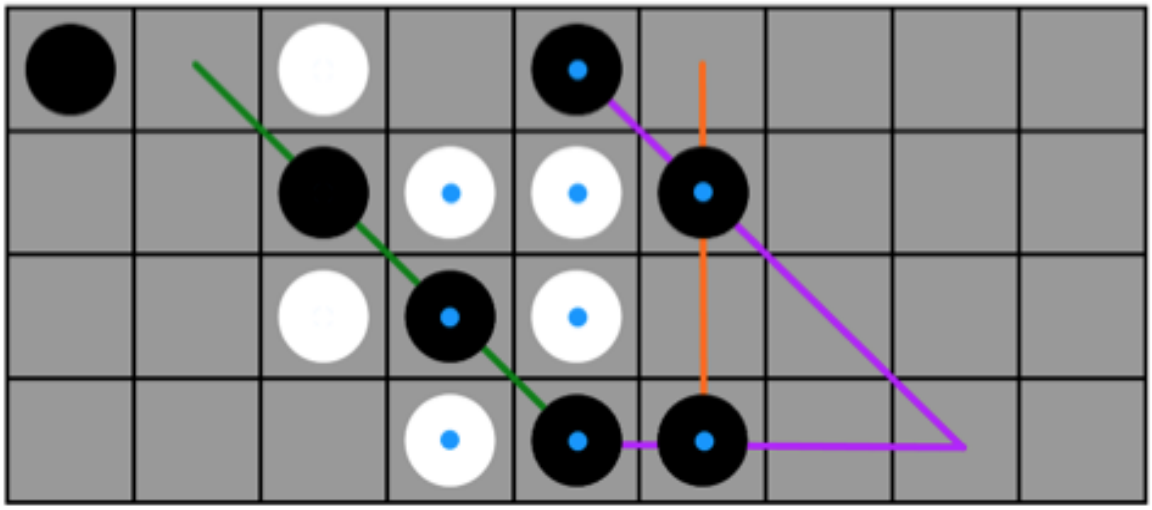

Figure S2: Demonstration of feature presence. Purple is f_{2a}, green is f_3, and orange is f_{2u}. Pieces with the blue dots count towards the "center" feature. Not shown is f_4, the four-in-a-row feature that results in a win.

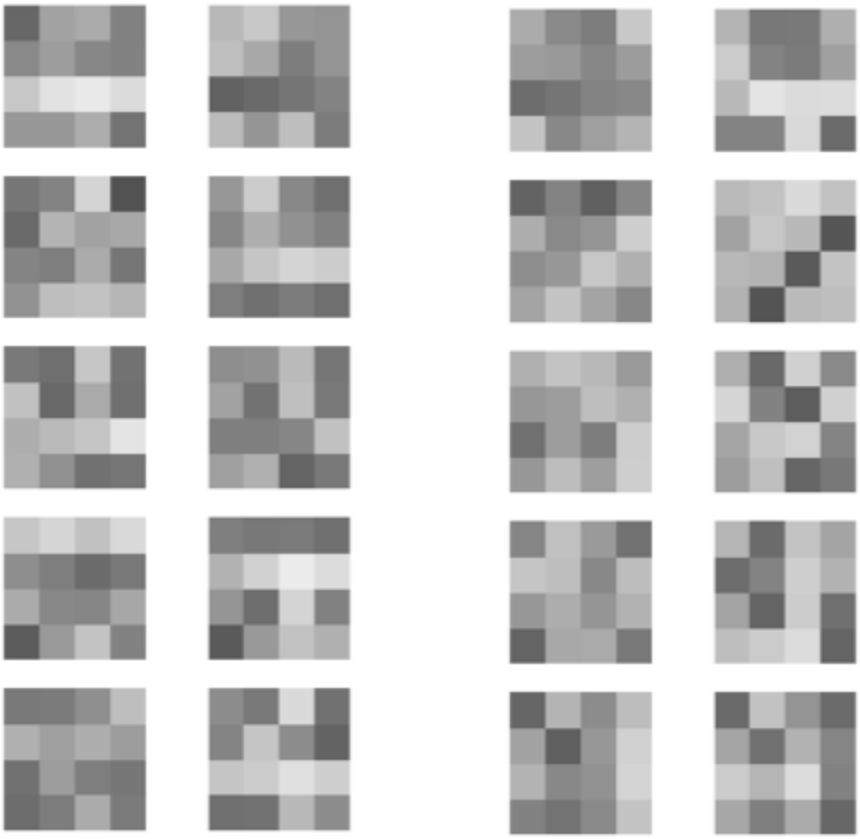

Figure S3: Some example filters learned by model CN. Each filter is a (2, 4, 4) volume, with two channels for the current player's pieces and their opponent's pieces. Each pair of patches in this image is a single filter; the left patch in each pair is for the player's own pieces, and the right patch is for their opponent's. Darker elements in a patch indicates that the filter responds *less* strongly to the presence of a piece at that location in its receptive field.

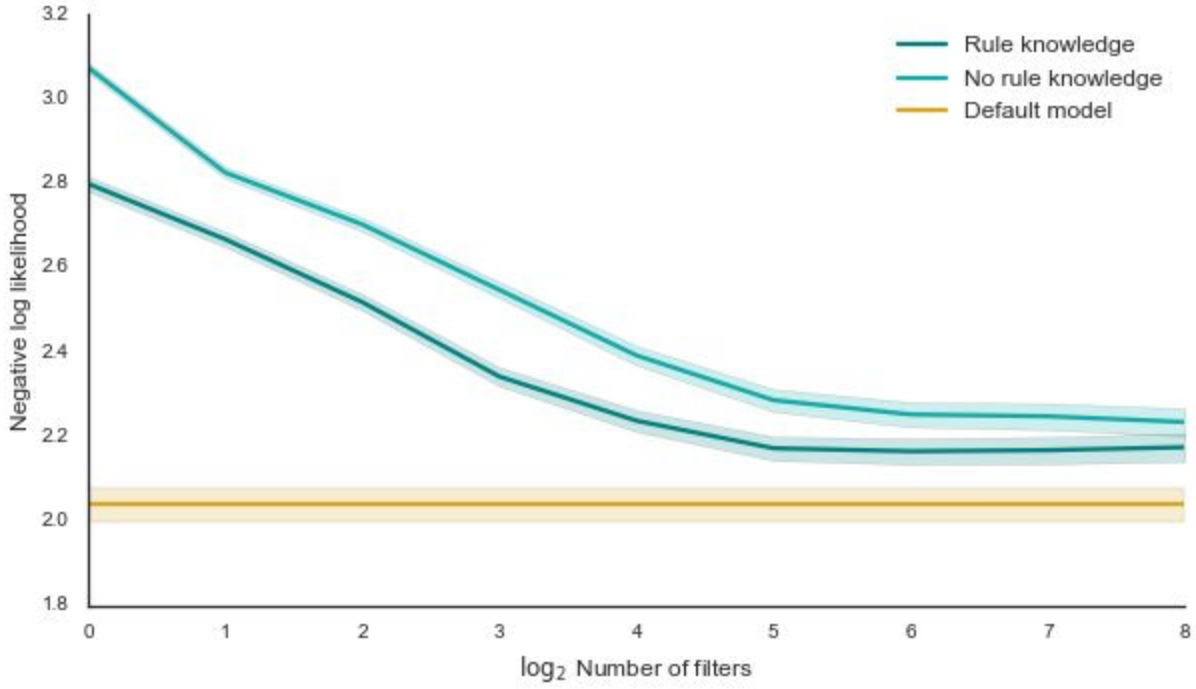

Figure S4: The negative log likelihood of the trained model for each combination of neural network architecture (with or without the legal move filter) and each number of filters on a $\log_2$ scale.