Modelling Recognition in Human Puzzle Solving

Ben Prystawski[1] (ben.prystawski@mail.utoronto.ca),
Rebekah Gelpi[1] (rebekah.gelpi@mail.utoronto.ca),
Christopher G. Lucas[2] (clucas2@inf.ed.ac.uk),
and Daphna Buchsbaum[3] (daphna@brown.edu)

[1]Department of Psychology, University of Toronto, Canada
[2]School of Informatics, University of Edinburgh, United Kingdom
[3]Department of Cognitive, Linguistic, and Psychological Sciences, Brown University, USA

Abstract

Our ability to play games like chess and Go relies on both planning several moves ahead and on recognition or gist–intuitively assessing the quality of possible game states without explicit planning. In this paper, we investigate the role of recognition in puzzle solving. We introduce a simple puzzle game to study planning and recognition in a non-adversarial context and a reinforcement learning agent which solves these puzzles relying purely on recognition. The agent relies on a neural network to capture intuitions about which game states are promising. We find that our model effectively predicts the relative difficulty of the puzzles for humans and shows similar qualitative patterns of success and initial moves to humans. Our task and model provide a basis for the study of planning and intuitive notions of fit in puzzle solving that is simple enough for use in developmental studies.

Keywords: decision-making; games; planning; deep learning; neural networks

Introduction

When playing a game like chess or Go, we decide what moves to take through a mixture of planning several moves into the future and assessing the gist of a game state through heuristics and intuition, without explicit planning. Relying on intuition to some degree is necessary due to the vast state spaces of these games, as it is not feasible to evaluate the potential outcomes of all possible moves using brute-force search techniques. In chess, it is typically believed that the primary difference distinguishing grandmasters from amateurs is their superior ability to interpret the gist of possible game states rather than an ability to plan more moves into the future (De Groot, 1978; Gobet & Simon, 1996). In contrast, the difference between skilled and unskilled amateurs is more often due to skilled amateurs thinking further ahead. This pattern has been found in games with much smaller state spaces as well; van Opheusden et al. (2021) found that players who performed better at a simple kind of game planned more moves ahead than those who achieved worse performances.

There are several possible explanations as to why chess grandmasters tend to rely more on recognition than planning. When we play a game like chess, we tend to imagine the outcomes of different moves in light of how the other player might respond. Simulating or searching through the space of possibilities only goes so far, however; in most games we can only consider a tiny fraction of possible outcomes, so we must also rely on a sense of what intermediate outcomes are better or worse. Sometimes this involves concrete events – a move might lead to taking or losing a piece – but experienced players also develop a holistic sense of how good or bad a particular game state or intermediate outcome is. There is evidence that people rely on both mental simulation and a holistic sense of the "gist" of an outcome (Gobet & Simon, 1996) when making decisions in general. In understanding how people learn to play games, we can shed light on the foundations of human planning, learning, and decision-making.

Computational models are valuable tools for understanding how people play games and solve puzzles. Given models that successfully play games such as chess and Go, we can probe their internal representations to provide hypotheses as to how humans might represent game states and approach game play. For example, algorithms like AlphaZero use a mixture of planning and recognition via Monte Carlo Tree Search to achieve human-level or super-human performance in a wide variety of games (Silver et al., 2018).

In this paper, we investigate how people use planning and recognition when playing games using a puzzle-solving task. While adversarial games have been used to study facets of human cognition (Charness, 1992, for example), and are well-studied testbeds for reinforcement learning methods, a single-player puzzle-solving task has some advantages.

First, in an adversarial game, a player must reason about both their own moves and how their opponent might respond. This engages a mixture of problem-solving and theory of mind, as the player must assess their opponent's play style and tailor their moves accordingly. While theory of mind in competitive games is a rich area of research (Pynadath et al., 2013; Oey et al., 2019; Brockbank & Vul, 2020), we often want to focus exclusively on the problem-solving component in order to isolate specific problem-solving strategies. Differences in how people apply theory of mind might interfere with attempts to study problem-solving exclusively.

Second, typical adversarial games have extremely large state spaces – even versions of these common games that are modified to be simpler. For example, 9x9 Go still has over 10^{22} possible states (Tromp & Farnebäck, 2006).

Third, players often memorize explicit sequences of moves in games like chess, such as common openings and endgames. Human play in these games might then reflect the common sequences that players have memorized rather than general-purpose game-playing strategies.

Finally, the rules of games such as chess and Go tend to be

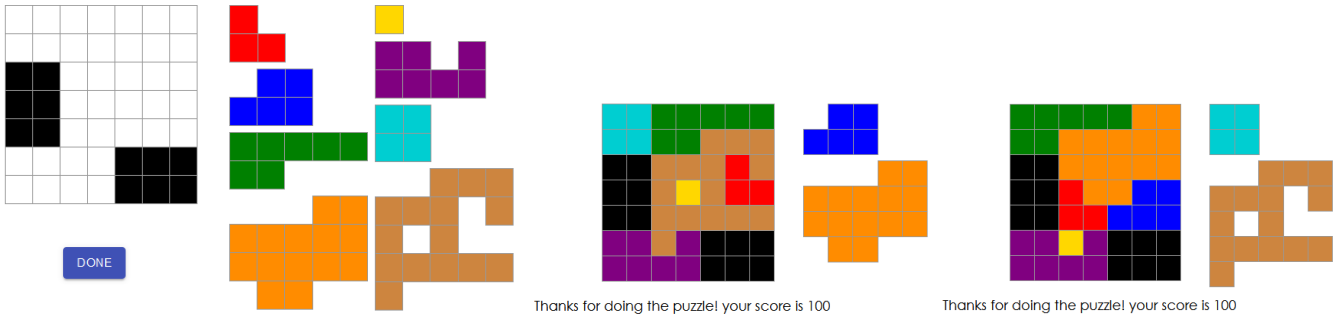

Figure 1: An example of a puzzle in our digital experiment. The leftmost panel shows the unsolved puzzle and the centre and right panels show the two possible solutions.

too complex for developmental studies. It is rare for children to learn to play chess before the age of 6. These games, then, are not suitable for the study of cognitive development. Children and adults have previously been shown to differ in how they learn and explore (Lucas et al., 2014; Blanco & Sloutsky, 2020) so understanding developmental differences in game playing and puzzle solving is potentially valuable.

We introduce a simple puzzle game for the study of decision-making and problem-solving which addresses these problems. Our puzzle-solving task is non-adversarial, appropriate for developmental studies, and it is amenable to customization, e.g., changing the number of possible moves, maximum game length, and other factors that influence difficulty. We also present a computational model of the game that allows us to explore the limits of recognition-based strategies and predict the performance of human players.

Related Work

While recognition is useful and even necessary in many games, it can be a double-edged sword. A line of work in developmental psychology has shown that adults' strong expectations about the causal structure of the world can inhibit their ability to learn unintuitive causal rules. In contrast, children tend to have weaker expectations and be more successful in learning true causal rules (Lucas et al., 2014). Kosoy et al. (2020) compared reinforcement learning agents to children directly by having children explore mazes in the DeepMind Lab environment. Both children and reinforcement learning agents learn about their environments by exploring, so reinforcement learning can be an appropriate framework for modelling children's active exploration.

Many game-playing algorithms in reinforcement learning achieve human-level performance or greater through a combination of planning and recognizing good game states. Methods such as Monte Carlo Tree Search (Coulom, 2006) and SAVE (Hamrick et al., 2019) use value estimates to plan several states ahead when choosing actions. These algorithms plan a few steps into the future, but the number of steps forward on each possible path is determined by recognition. Deep neural networks have previously succeeded at recognition, learning which states tend to lead to high and low rewards based on previous experience (e.g., Mnih et al., 2015).

In another investigation of how humans play games, Lindstedt & Gray (2020) found that world champion Tetris players exhibit a "cognitive speed bump", taking slightly more time to make their first rotation compared to novices, but pressing fewer buttons overall in rotation. This suggests that the highly skilled Tetris players planned where to place a piece by mentally rotating and moving it, then executed a sequence of rotations to bring the piece to its planned locations. In contrast, novice Tetris players tend to favor one rotation direction and make more rotations overall. This suggests that they do less planning at the outset than the expert players, instead rotating pieces to help search through the state space. Tetris has important similarities to our puzzle task, such as being non-adversarial, so skilled players' reliance on search might suggest that people also rely heavily on search in puzzle solving. However, Tetris players only know the next piece that will appear on the board and have limited time to plan, so it is possible that recognition also plays an important role.

Here, our focus is on understanding the role of intuitive assessments of how promising a state is in human puzzle solving. Do people search several steps using uninformed methods, or do they rely mostly on the gist of possible states? To investigate this question, we develop a computational model of recognition-based puzzle solving. We then compare the model's behaviour to that of humans in an experiment, focusing on overall success at solving puzzles, which kinds of puzzles humans and our model find challenging, and qualitative aspects of puzzle-solving strategy.

Puzzle Task

In designing a puzzle task for humans and reinforcement learning agents to solve, we sought to create a task that is simple enough that humans can quickly learn the task and solve it, but challenging enough that people do not always succeed. We also wanted people to be able to use existing spatial intuitions to estimate the values of game states with minimal prior experience. To meet these criteria, we designed a game where users try to fit pieces together to fill a grid.

The puzzle game consists of a square grid, walls, and pieces. The goal is to place the pieces on the grid such that they cover all of the empty squares. Some of the squares on the grid are walls where pieces cannot be placed. Once a

piece is placed on the grid, it cannot be moved or removed. This makes the analysis cleaner as it eliminates the risk of models getting stuck in loops of repeatedly adding and removing pieces. A puzzle might also have some extra pieces which are not necessary to solve it, making the game more difficult. Puzzles can have one solution or multiple. Figure 1 shows an example of a puzzle. The empty squares are white, the walls are black, and each piece is a different colour.

Automatic Generation of Puzzles

Algorithm 1: Puzzle generation algorithm

```
piece_index ← 0
while empty squares > 0 do
    piece_index ← piece_index + 1
    Select a random empty tile (i, j) on the grid
    while true do
        board[i, j] ← piece_index
        with probability p_t, break
        if there are no empty neighbours, break
        i, j ← the indices of a random empty
          neighbouring tile
    end
end
return board
```

In order to create a large amount of training data for our models, we developed an algorithm to automatically generate puzzles. Puzzles were situated on a 7-by-7 grid. We generated three types of puzzle. First, we created standard single-solution puzzles. These were made using Algorithm 1. In brief, the board was filled by a series of random walks, with the tiles covered by each walk turned into a piece. Next, two of the pieces were converted into walls. We generated puzzles repeatedly and selected those that had a total of 5 pieces, which we decided created a puzzle that was neither too easy nor too complex. While this algorithm can sometimes generate puzzles with multiple solutions, we inspected many puzzles and found that solutions were never meaningfully different. For example, the algorithm might generate a puzzle with two interchangeable single-tile pieces, but the random walk tends to generate oddly shaped pieces that can only fit into a complete solution in one place.

We also generated multi-solution puzzles to test whether people tend to arrive at the same solution to multi-solution puzzles or whether they find different solutions. To generate these puzzles, we took the single-solution puzzles described above and then randomly placed the existing pieces on the board until no more could be placed. Next, all contiguous regions of remaining empty space were turned into new pieces, creating a second solution. Thus, multi-solution puzzles have more pieces than are necessary to solve the puzzle. This can be seen in Figure 1 (right), where the puzzle is complete but there are still remaining pieces on the right. We repeated this process, starting from the same single-solution puzzle until we created a new puzzle with 7 pieces in total.

Finally, we created single-solution puzzles with extra pieces in order to distinguish between the effects of having multiple possible solutions and having extra pieces. To generate these puzzles, we started with a single-solution puzzle, then did two more random walks on the board, keeping the walls and replacing the previous pieces with empty spaces. This results in a puzzle with 7 pieces.

We also created versions of the multi-solution and single-solution extra-pieces puzzles with fewer possible paths through the state space to test whether simpler puzzles were easier for humans. We generated puzzles as described above but kept only those with at most 5000 paths possible paths.

Computational Methods

Our puzzle-solving task is an instance of a Markov Decision Process (MDP). At each time step, the agent can place any of the unplaced pieces in any of the locations on the grid where that piece fits. It repeats this until it can no longer place any pieces and receives a reward of $100(2^{-n})$, where n is the number of empty squares remaining on the board. We chose this reward structure to create a strong incentive to find correct and complete solutions to puzzles, while still providing partial credit for almost complete ones.

We use a neural network to assess state values. Our model uses an ε-greedy policy with $\varepsilon = 0.1$. At each time step, it places a random piece in a random legal location with probability ε. With probability $1-\varepsilon$, it assesses the value of each possible future state using the neural network and takes the action which the network predicts is best.

Our reinforcement learning agent models one extreme of the spectrum between planning and recognition: purely recognition-based puzzle solving. At every step, the agent either selects an action randomly with a small probability or considers every possible action available and the state that would result from taking it, then takes the action whose resulting state has the highest estimated value. The state values are based on which states led to high rewards during training, so the neural network's estimates can be viewed as approximate planning toward a solution.

Neural Value Approximation for Assessing Fit

We model people's intuitions about the goodness of a given puzzle piece's fit using a neural network. Neural networks have previously succeeded at capturing intuitive knowledge that is not easily expressible in words, such as determining whether a sonar signal resembles that of a mine or rock (Gorman & Sejnowski, 1988), which suggests that a neural network is appropriate for recognizing good puzzle states.

We designed state representations to assess three aspects of the state: the piece that is newly placed, the previously-placed pieces, and the walls of the puzzle. These are each represented as layers, which are two-dimensional grids of the same size as the board that capture one aspect of the game state. The first layer represents the shape and location of the most recently placed piece, with a 1 in locations occupied by

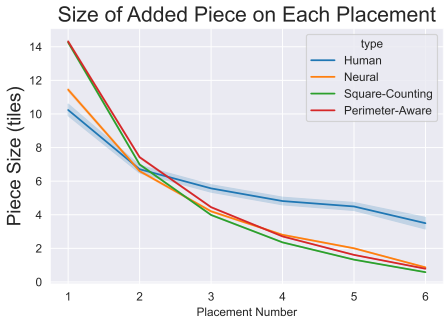

Figure 2: Mean size of the pieces added by humans (blue), the neural model (orange), the square-counting model (green), and the perimeter-aware model (red) on each placement.

that piece and a 0 elsewhere. The second layer consists of 1s representing the tiles covered by previously-placed puzzle pieces, and the third layer consists of 1s representing walls.

This state representation is fed into the neural network, which first applies two two-dimensional convolutional layers, then three fully-connected layers. All layers use ReLU activation functions. The neural network was trained using the deep Q-learning algorithm (Mnih et al., 2013). For each puzzle type without a restricted state space, we generated training, validation, and test sets of 1000 puzzles each. The model was trained on a single type of puzzle but run across all test puzzles. We trained 34 instances of this neural network on the single-solution puzzles and 33 on each of the multi-solution puzzles and single-solution puzzles with extra pieces, then selected the trained model that performed best across all puzzle types on the validation sets for further use. The puzzles with restricted state spaces are computationally more intensive to generate, so we only generated 100 test puzzles of each type.

Model Behaviour

On the entire training set, the model[1] solved the puzzle correctly 28% of the time on the single-solution puzzles, 29% of the time on the multi-solution puzzles, and 21% of the time on the single-solution puzzles with extra pieces. It achieved a mean reward of 30.7, 35.2, and 26.8 on the single-solution, multi-solution, and single-solution extra-pieces puzzles respectively. It performed better on the puzzles with restricted state spaces, getting the multi-solution and single-solution extra-pieces puzzles correct 62.2% and 36.6% of the time and achieving mean rewards of 63.9 and 43.6, respectively. Model scores on each condition are shown in Figure 3.

The neural model tended to place a very large piece first and decrease the size of the placed piece in a concave fashion. Figure 2 shows the precise curve of the average added piece size for each placement. This suggests that the neural network has learned to evaluate game states where more of the board is filled as more promising than emptier states since they are

[1]Reported model results use the version trained on single-solution puzzles which overall performed best on the validation puzzles of all types, but results are similar for the versions trained on the different puzzle types.

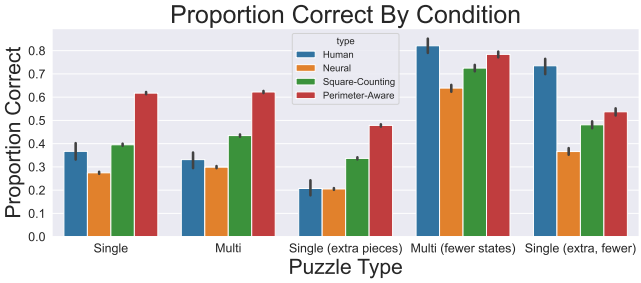

Figure 3: Proportion of puzzles solved correctly by humans (blue), the neural model (orange), the square-counting model (green), and the perimeter-aware model (red) per condition. Model scores are across all test puzzles while human scores are across the 30 puzzles in the experiment. Error bars denote standard error of the mean.

fewer tiles away from being complete.

Alternative Models

We introduce two alternative models to represent different approaches to puzzle solving that humans could be taking: a *square-counting* heuristic model that simply places large pieces first and a *perimeter-aware* model that uses knowledge of the existing perimeter in addition to piece size.

The square-counting model captures the basic fact that filling more of the board by placing larger pieces brings the puzzle closer to completion. This model simply places the pieces in descending order of their size, where size is the number of squares they occupy, choosing randomly between pieces in case of ties. The model does not make any judgments about where to place the chosen piece, so it simply places it randomly. If the largest piece does not fit anywhere on the board, it skips it and places the next largest piece that fits somewhere.

The perimeter-aware model also places larger pieces first, but it decides where to place each piece by maximizing contact between the piece's perimeter and the edges of the existing board. Here "edges" includes the walls of the puzzle, the boundaries of the board, and squares covered by previously-placed pieces. This model favours placements that fit snugly within the existing board. If multiple locations have the same perimeter contact, it chooses randomly between them. This corresponds to the intuitive strategy of placing the largest available piece wherever it fits most snugly on the board.

Experiment: How do People Solve Puzzles?

We tested the predictions of our model in an online behavioural experiment where humans solve some of the same puzzles presented to our model. There were three main hypotheses we tested in this experiment.

First, we wanted to test whether our model can effectively predict which puzzles humans find easy and hard. Secondly, we wanted to test our model's qualitative prediction that people will place larger pieces earlier. If people plan their action sequences beforehand via brute-force methods, the order in which they add pieces should be roughly random. If they rely

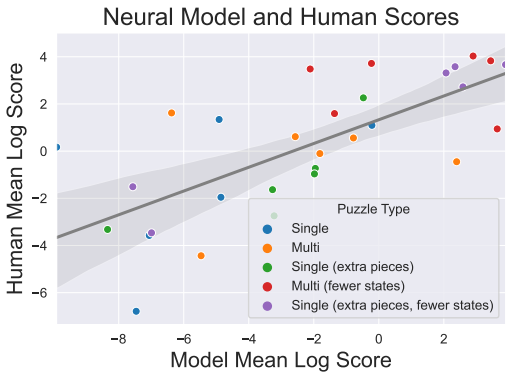

Figure 4: Scatterplot of mean score for humans and the neural model ($\rho = 0.71$. Scores in this plot were calculated using the model trained on single-solution puzzles; results are similar when trained on the other conditions. Log model scores were averaged over 100 runs on each puzzle.

more on recognition, they would likely place larger pieces first, as our model did. Finally, we wanted to test how often people discover alternative solutions when they exist. When a puzzle has multiple solutions, do people tend to find the same solution, or do they uncover multiple solutions?

Methods

Participants and Design We recruited 100 adult participants on Amazon Mechanical Turk and paid $0.50 for completing the task. Ninety-two participants' data were analyzed; eight were excluded due to not completing an attention check.

Materials and Procedure Participants completed an online puzzle-solving task. In this task, they first read an instructions page, then saw a sequence of ten puzzles with a grid in the centre of the screen and a collection of pieces on the right. Participants could click the pieces and drag them onto the board. Once pieces were placed on the board, they could not be moved or removed. Participants could click a "Done" button if they could not place any more pieces.

Participants were assigned to one of three possible sets of ten puzzles, with two puzzles from each of the five types. The puzzles were displayed to participants in random order. After completing a puzzle, participants saw a score computed identically to the reward function for the model. However, these scores did not correspond to any monetary bonus.

The puzzles shown to participants were generated using the automatic puzzle generation method outlined above, then hand-selected to be interesting and challenging. For each single-solution puzzle that was included, the multi-solution and single-solution with extra pieces puzzles based on that puzzle were also included, while the restricted state space puzzles were each generated and chosen independently. Figure 1 shows examples of the puzzle-solving interface.

Results and Discussion

Humans solved the puzzles 37%, 33%, and 21% of the time in the single-solution, multi-solution, and single-solution with

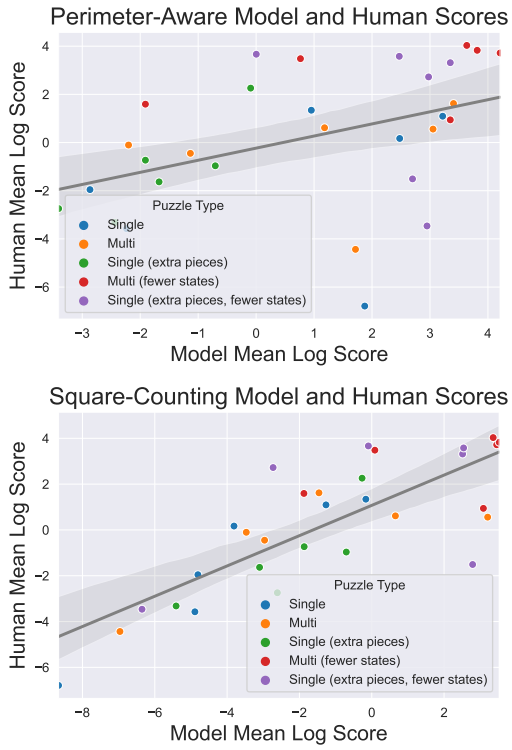

Figure 5: Scatterplot comparing mean log score for humans and the perimeter-aware model (top, $\rho = 0.43$) and the square-counting model (bottom, $\rho = 0.79$). Log model scores were averaged over 100 runs on each puzzle.

extra pieces conditions respectively, achieving mean scores of 36.9, 36.1, and 25.0. They solved the puzzles 82% and 73% of the time in the multi-solution and single-solution extra pieces puzzles with restricted state spaces, achieving mean scores of 83.1 and 74.0. Figure 3 shows the mean score achieved by participants in each condition.

Humans performed much better on the puzzles with restricted state spaces compared to those without restricted state spaces. A two-way mixed effects ANOVA on state space (restricted vs. unrestricted) and puzzle type with participant as a random effect (multi-solution vs. single-solution with extra pieces) revealed significant effects of state space ($F(1,273) = 233.3$, $p < 0.0001$) and puzzle type ($F(1,273) = 9.7$, $p = 0.002$), but no significant interaction ($F(1,273) = 0.16$, $p = 0.69$). This suggests that humans find puzzles with fewer possible paths through the state space easier. The neural model also performed better on puzzles with restricted state spaces, but the increase was less dramatic.

As Figure 5 shows, there is a strong correlation between the log-scores achieved by our model and human participants ($\rho = 0.71$, $p < 0.0001$). The square-counting model's performance also correlated strongly with human performance ($\rho = 0.79$, $p < 0.0001$) while the perimeter-aware model correlated fairly weakly with human performance ($\rho = 0.42, p = 0.020$). Human intuitions might incorporate a piece's perimeter to some extent, but relying exclusively on the perimeter in

deciding where to place a piece results in different behaviour than humans. There was also a significant negative correlation between the logarithm of the number of possible paths in the state space, as computed through depth-first search, and the mean log scores achieved by humans ($\rho = -0.49$, $p = 0.006$), the neural model ($\rho = -0.36$, $p = 0.048$), the square-counting model ($\rho = -0.39$, $p = 0.031$), and the perimeter-aware model ($\rho = -0.52$, $p = 0.003$). Humans and all of the models do better on simpler puzzles.

Among puzzles with multiple solutions, we find that participants arrive at the most common solution for a given puzzle on 67% of trials. A brute-force strategy has no reason to favour any solutions over others, so we would expect to see an approximately uniform distribution over solutions. Instead, people cluster around one solution, which suggests that shared intuitions about which states are good and bad lead them through similar paths. The neural model did not solve the experimental puzzles frequently enough to compare, but across all 1000 multi-solution test puzzles it arrived at the most common solution 85% of the time.

Finally, we compared the size of pieces placed at each time step between humans and the models. Figure 2 shows the mean size of the piece added at each time step for humans, the neural model, and the square-counting model. The same general pattern of a sharp decrease for the first two placements, then gradual levelling off occurs in all three. However, the trend becomes flatter around placement 3 for humans compared to the models, with the square-counting model exhibiting the sharpest decrease. This might be due to people using some amount of planning rather than relying on recognition entirely. In particular, it is possible that people place one or two large pieces first to reduce the complexity of the remaining puzzle, then either apply brute-force search or partially gist-informed strategies for the remainder of the puzzle.

Figure 3 also shows that the improvement in performance on puzzles with restricted state spaces is greater for humans than for the neural model. Humans might rely more on gist when puzzles are more complex and more on brute-force planning when puzzles are simpler. The neural model does not plan, so it cannot benefit from simpler puzzles in this way.

General Discussion

In this paper, we presented a puzzle game that we used to study the relationship between planning and recognition in games. We developed a computational model of recognition-based puzzle solving and compared its behaviour to that of humans in a behavioural experiment, finding that humans do better on puzzles with smaller state spaces, their scores correlate strongly with those of the model, and humans and the model both show a pattern of decreasing placed piece sizes.

Our experiment provided evidence that a purely recognition-based model of puzzle solving effectively captured the difficulty of a puzzle. The correlation between the neural model's and humans' performance across puzzles indicates that how straightforward it is to recognize good and bad states in a particular puzzle is a major determinant of the difficulty of the puzzle for humans. The scores of the neural and square-counting models both correlated strongly with human scores, which suggests that humans might rely on a combination of recognizing promising states and the simple, effective heuristic of placing the biggest possible piece first wherever it fits. Placing pieces in descending order of size appears to be a fairly effective strategy, particularly on simpler puzzles with relatively small state spaces. In fact, learning this heuristic might even be a consequence of recognition. The neural model also learns to place larger pieces first, indicating that it tends to recognize states resulting from placing larger pieces as more promising.

Furthermore, our models show qualitative similarities to human puzzle solving. The size of placed pieces starts large, decreases sharply, and gradually levels off. The decrease is less steep for humans compared to any of the models after the second placement. This might indicate that people rely on recognition for the first one or two piece placements, shrinking the remaining state space significantly, which likely makes subsequent planning much easier.

The finding that on average 67% of participants arrived at the most common solution for each puzzle indicates that while people may have some shared notions of promising and unpromising puzzle states, these intuitions may not be universal, or may not be the only consideration when placing a piece. For example, if people switch between search and intuition at different points in a puzzle, different solutions might become more or less available under different strategies.

In future work, we hope to investigate the relationship between planning and recognition developmentally. If adults fail to solve some puzzles because they have unhelpful prior expectations about which states are good, children might perform better on those puzzles or discover different solutions. Previous studies have suggested that children tend to have weaker priors in tasks like causal inference and search hypothesis spaces more widely than adults, which can lead them to uncover counter-intuitive patterns that adults miss (Lucas et al., 2014). Determining whether a similar effect exists in puzzle solving could be a valuable step toward understanding other developmental phenomena, such as changes in susceptibility to functional fixedness (German & Defeyter, 2000).

Furthermore, it is likely that planning still plays a significant part in our puzzle-solving strategies, which our model does not capture. We plan to improve the model by incorporating variable levels of planning. Reinforcement learning agents often select actions by planning several states into the future via methods like Monte Carlo Tree Search (Coulom, 2006). These methods often use recognition to estimate which possible future states are most promising and allocate more planning resources there. Developing a model that can interpolate between fully brute-force planning and pure reliance on recognition could be valuable in determining how people combine recognition and planning more precisely.

Acknowledgements

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC, 2016- 05552).

References

Blanco, N. J., & Sloutsky, V. M. (2020). Systematic exploration and uncertainty dominate young children's choices. *Developmental Science*, e13026.

Brockbank, E., & Vul, E. (2020). Recursive adversarial reasoning in the rock, paper, scissors game. In *Proceedings of the 42nd Annual Meeting of the Cognitive Science Society* (pp. 1015–1021).

Charness, N. (1992). The impact of chess research on cognitive science. *Psychological Research*, *54*(1), 4–9.

Coulom, R. (2006). Efficient selectivity and backup operators in monte-carlo tree search. In *International Conference on Computers and Games* (pp. 72–83).

De Groot, A. D. (1978). *Thought and choice in chess* (Vol. 4). Walter de Gruyter GmbH & Co KG.

German, T. P., & Defeyter, M. A. (2000). Immunity to functional fixedness in young children. *Psychonomic Bulletin & Review*, *7*(4), 707–712.

Gobet, F., & Simon, H. A. (1996). The roles of recognition processes and look-ahead search in time-constrained expert problem solving: Evidence from grand-master-level chess. *Psychological Science*, *7*(1), 52-55. doi: 10.1111/j.1467-9280.1996.tb00666.x

Gorman, R. P., & Sejnowski, T. J. (1988). Learned classification of sonar targets using a massively parallel network. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *36*(7), 1135–1140.

Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Pfaff, T., Weber, T., Buesing, L., & Battaglia, P. W. (2019). Combining Q-learning and search with amortized value estimates. *arXiv preprint arXiv:1912.02807*.

Kosoy, E., Collins, J., Chan, D. M., Hamrick, J. B., Huang, S., Gopnik, A., & Canny, J. (2020). Exploring exploration: Comparing children with RL agents in unified environments. *arXiv preprint arXiv:2005.02880*.

Lindstedt, J. K., & Gray, W. D. (2020). The "cognitive speedbump": How world champion Tetris players trade milliseconds for seconds. In *Proceedings of the 42nd Annual Meeting of the Cognitive Science Society* (pp. 66–71).

Lucas, C. G., Bridgers, S., Griffiths, T. L., & Gopnik, A. (2014). When children are better (or at least more open-minded) learners than adults: Developmental differences in learning the forms of causal relationships. *Cognition*, *131*(2), 284–299.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing Atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... others (2015). Human-level control through deep reinforcement learning. *nature*, *518*(7540), 529–533.

Oey, L. A., Schachner, A., & Vul, E. (2019). Designing good deception: Recursive theory of mind in lying and lie detection. In *Proceedings of the 41st Annual Meeting of the Cognitive Science Society* (pp. 897–903).

Pynadath, D. V., Wang, N., & Marsella, S. C. (2013). Are you thinking what i'm thinking? An evaluation of a simplified theory of mind. In *International Workshop on Intelligent Virtual Agents* (pp. 44–57).

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., ... others (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, *362*(6419), 1140–1144.

Tromp, J., & Farnebäck, G. (2006). Combinatorics of go. In *International Conference on Computers and Games* (pp. 84–99).

van Opheusden, B., Galbiati, G., Kuperwajs, I., Bnaya, Z., li, Y., & Ma, W.-J. (2021). *Revealing the impact of expertise on human planning with a two-player board game.* PsyArXiv. Retrieved from psyarxiv.com/rhq5j