Cognitively-plausible Monte-Carlo Tree Search

Aloïs Rautureau[1][0009−0007−8507−2167] and Éric Piette[2][0000−0001−8355−636X]

[1] École Normale Supérieure de Rennes, Bruz, France
alois.rautureau@ens-rennes.fr
[2] Université Catholique de Louvain, Louvain-La-Neuve, Belgium
eric.piette@uclouvain.be

Abstract. While AI systems have surpassed human performance in a wide variety of games such as chess, go or shogi, it would be far-fetched to describe these systems as "human-like". Despite their exceptional performance, these systems fail to replicate the pattern-based, intuitive characteristics of human decision-making observed in the literature on human cognition. We introduce and evaluate a model inspired by the Double-Process Theory of Cognition (DPTC), and prior research, aiming to create agents that mimick human behavior rather than optimizing playing strength. Unlike traditional models that rely on exhaustive search strategies, our approach focuses on pattern recognition to aggressively narrow the search space. Experimental evaluation of the model is done in the context of Renju, a Five-In-A-Row game, by measuring the capability of our model to predict human moves. Our experiments demonstrate that the proposed model achieves a higher degree of accuracy in the move matching task compared to other methods, indicating a closer alignment with human behaviour in the game of Renju. These results validate our approach as a preliminary step towards developing human-like game-playing agents, with the goal of applying these concepts to General Game Playing (GGP).

Keywords: Human-like AI · Double-process Theory of Cognition · Cognitive modelling · Pattern Recognition · Game playing · Monte-Carlo Tree Search.

1 Introduction

Although AI systems have surpassed humans in games like chess [5] or Go [24], describing them as "human-like" would be an overstatement. Despite their exceptional performance, these systems fail to replicate the intuitive, pattern-based decision-making that defines human cognition [10] [8]. To address this gap, we propose and evaluate a model inspired by cognitive science, particularly the Double-Process Theory of Cognition (DPTC) [16], and prior research on human-like agents across various domains [6] [11] [15] [18] [19] [20] [21] [27]. Unlike state-of-the-art models that prioritize exhaustive search strategies for maximizing performance, our model focuses on pattern recognition of spatial features and best-first biased search algorithms like Monte-Carlo Tree Search (MCTS)

[4]. The aim is to emulate the adaptive, intuitive decision-making of humans, rather than purely optimizing for gameplay strength. To measure this, we assess the model's "human-likeness" through move matching—its ability to predict human moves in specific game states.

Our experiments show that the proposed model achieves greater accuracy in move matching compared to other methods, demonstrating closer alignment with human behavior in the game of Renju. This variant of Gomoku balances the game by restricting certain moves for the first player to counter their inherent advantage. Both games are played on a 15x15 board, where players aim to form an unbroken line of five stones. Renju also introduces opening rules, requiring initial moves in the center and allowing players to swap colors, leading to 26 possible opening patterns before regular play starts. While Renju programs have only recently reached human-level strength—Yixin achieved a draw against 2018 world champion Qi Guan—Monte-Carlo Tree Search (MCTS) struggles with "trap states," where only one move wins or loses, due to its random sampling. Humans, however, easily recognize patterns like double-four, double-three, five-in-a-row, or overlines, making these situations simpler to resolve.

2 Related work

2.1 Cognitive science foundations

The approach proposed in this article, as well as previous research on artificial agents discussed in Section 2.2, take inspiration from cognitive science. Akin to Biologically Inspired Cognitive Architectures [13], they focus on high-level features of human cognition such as pattern recognition and spatial reasoning, rather than trying to replicate low-level neurophysiological phenomenons.

Seminal studies by de Groot [10] and Chase and Simon [8] on chess perception showed that expertise is more about efficiently representing the board as spatially connected "chunks" of pieces than conducting deep searches. Experts analyze more complex chunks than novices, allowing them to identify advantageous moves more accurately. Unlike modern chess engines, which perform exhaustive searches, human players typically focus on just two moves on average [10], despite chess's branching factor of around 35. This supports the idea that experts focus on specific spatial relationships between pieces, a skill refined over years of play, further confirmed by studies on saccadic eye movements [7] [14].

Research shows that humans rely on recognizing spatial patterns to intuitively identify advantageous moves, though deeper analysis of positions remains limited. Recent work by He et al. [15] applied the Dual-Process Theory of Cognition [16] within a game-theoretic framework for conversational agents, with promising results. This theory suggests that human decision-making operates through two interconnected systems, offering a more human-like approach to planning:

– *System 1* is the *intuitive brain*, making heavy use of pattern-recognition capabilities and memory to identify potentially rewarding actions. This system

can be linked to the research of de Groot [10] and Chase and Simon [8] on the thought process of chess players. Possible actions are filtered at this point, with only a few intuitively good moves remaining.

– *System 2* is the *analytical brain*. It serves as a fallback in case System 1 is not capable of making a decision, which happens when met with a situation which has not been encountered before or when doubt arises between two or more actions of seemingly almost equal value. This system can be linked to usual search algorithms used for game playing agents.

2.2 Previous work on human-like agents

Human-like AI has been a topic of interest for many fields of research in which artificial agents interact with humans i.e. *cooperative tasks*, such as Human-Robot Interaction (HRI) [27] [1], video games [21] [6] [26] or conversational agents [15]. Previous literature on human-like tabletop game playing agents also exists in the form of multiple human-like chess playing programs such as Maia Chess [20], Morph [18] or SYLPH [11], as well as attempts to generalize evaluations based on pattern-recognition to games in general [19].

While these works have put forth many methods to design and evaluate human-like agents such as Inverse Reinforcement Learning (IRL) [23] [3] and other imitation learning methods, we will focus on the methods used and expanded upon in this paper.

Research on pattern recognition, as discussed in section 2.1, has inspired several models. Levinson and Snyder's Morph [18] and SYLPH [11] used chess engines that assigned weights to attack-defend patterns. A key challenge was how to aggregate weights for complex, repeating patterns within a position. With neural networks, this approach was revisited. Mandziuk [19] applied it to Connect-Four, using atomic patterns from the game's rules as inputs for a feed-forward neural network. This enabled the network to learn complex features, introducing non-linearity and improving decision-making, demonstrating the effectiveness of pattern-recognition models in emulating human-like decision-making.

The Maia Chess engine, as described by [20], aimed to narrow the behavioral gap between superhuman AI and human players by introducing move matching to evaluate AI human-likeness, as outlined in subsection **??**. Traditional methods to weaken AI, such as depth limiting in $\alpha\beta$ searches or stopping self-play early, did not improve move matching. However, an AlphaZero-based model trained on human game databases outperformed engines like Leela Chess Zero and Stockfish in this task. Despite its success, critics argue this approach treats gameplay as a classification task, failing to replicate higher-level human cognition. Additionally, its reliance on extensive human game data limits its application to General Game Playing (GGP) and less-documented games, lacking ongoing self-learning capabilities.

3 Human-like model for Renju

The proposed model (Figure 1) draws from the Double-Process Theory of Cognition, recently evaluated by He et al. [15] to improve the decision-making of conversational agents, to create an intuitive policy for game playing. It employs two interconnected systems: an intuitive "System 1" based on pattern recognition and an analytical "System 2" using Monte-Carlo Tree Search (MCTS) [4]. System 1 filters out intuitively poor options (Section 3.1), while System 2 performs deeper analysis via MCTS when multiple options have similar estimated values (Section 3.2). Section 3.3 will explain how the model learns through self-play.

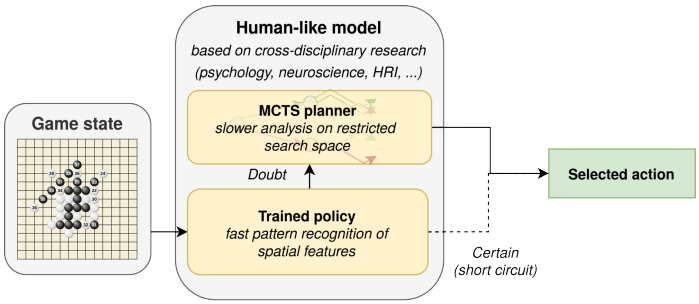

Fig. 1. Proposed model inspired by the dual-process theory of cognition [16]. This model integrates a fast pattern-recognition system based on a feed-forward neural network taking as input high-level spatial features of Renju (System 1) that defaults to Monte-Carlo Tree Search (System 2) when it encounters a state of uncertainty regarding the optimal action.

3.1 Pattern-based policy (System 1)

Our model's pattern-based policy is inspired by the work of Mańdziuk [19] on Connect-Four. Our patterns are represented as 5-element vectors spanning part of a single line in any direction. Each value can be set to 1 if the intersection contains a stone of the current player, 0 if empty or -1 for stones of the opponent. The intersections are ordered bottom-up and left-to-right. Each pattern also has associated features:

- **(F1)** The trivial feature of being a pattern.
- **(F2)** The pattern's direction, being one of ascending, descending, horizontal or vertical.
- **(F3)** The number of the leftmost column spanned by the pattern.
- **(F4)** The number of the lowest row spanned by the pattern.

And 3 attributes that track more intricate information about the pattern:

- **(A1)** Assigned only to the empty pattern i.e. $\{0, 0, 0, 0, 0\}$

- **(A2)** The pattern's potential. This is 0 if the pattern contains more than one type of stone (no player can make a five-in-a-row), or the number of stones of one color that are present, negative if they are stones of the opponent e.g. -4 if the opponent has four stones.
- **(A3)** Assigned to patterns with no empty intersections.

These patterns are fed into a feed-forward neural network with shared weights across attributes and features, producing a policy that estimates the value of each possible action. This is represented as a 225-element vector, with each element corresponding to a board intersection. Unlike conventional methods, this policy is not used for preliminary evaluation in MCTS but to filter and retain only intuitively good actions. This is done by partitioning the action space, as detailed in Algorithm 1. If the filtering process identifies a single "good" action, the deliberative System 2 (Subsection 3.2) is bypassed. Otherwise, the search is limited to the refined subset of actions.

Algorithm 1 The algorithm partitions the action space A by iteratively selecting the best actions until the entropy of the policy of remaining non-selected actions $H(P_A)$ is high enough with regards to an ϵ parameter. This divides the actions into two subsets: "good" and "bad", both with high entropy. If all actions are equally viable at the start, resulting in no "good" subset, the algorithm returns the full set of legal actions.

$P_A \leftarrow$ policy over A
$A' \leftarrow \{\}$
$H \leftarrow H(P_A)$
if $2^{|A|} - \epsilon \leq H$ or $|A| = 0$ **then**
  return A
end if
repeat
  $a^* \leftarrow \arg\max p_a$
  $A' \leftarrow A' \cup \{a^*\}$
  $A \leftarrow A \setminus \{a^*\}$
  $P_A \leftarrow \{\frac{p_a}{1-p_{a^*}}, a \in A\}$
  $H \leftarrow H(P_A)$
until $\frac{2^{|A|}}{H} \leq \epsilon$ or $|A| = 0$
return A'

This method more accurately emulates human decision-making by focusing on a subset of intuitively good actions, significantly reducing the search space for deeper analysis.

3.2 Search (System 2)

Actions filtered by System 1 are searched deeper using a tweaked MCTS framework. The choice of this algorithm instead of depth-first alternatives such as

Minimax is due to the fact that best-first algorithm tend to more closely emulate human decision-making [10] [8] [20]. The original MCTS algorithm operates in four phases:

1. **Selection**: Starting at the root, child nodes are recursively selected until an unexpanded leaf node is reached, often using a strategy like Upper Confidence Bound applied to Trees (UCT) [17], which balances exploration and exploitation.
2. **Expansion**: A new child node is added to the tree.
3. **Rollout**: A simulated game is played from the expanded node, using random or heuristic-based decisions to predict an outcome.
4. **Backpropagation**: The result is backpropagated up the tree, updating the values of nodes visited during selection.

Several modifications were made to align the base MCTS algorithm with human cognitive processes:

- The **simulation phase** is omitted, as humans don't randomly simulate outcomes. Instead, the value of the expanded leaf node is set by the highest value of its potential children, as evaluated by System 1.
- Children are defined not by the entire action space but **by a filtered subset of actions through System 1**, reflecting how humans focus on a subset of "good" actions based on experience.
- The search tree has **bounded growth** to mimic human cognitive limits. Even with time to deliberate, humans can retain only so much strategic information, so focus is on refining exploitable lines. A strict cap limits the number of nodes stored.

This approach emphasizes the intuitive brain by heavily biasing the search algorithm, determining leaf node values during expansions and narrowing the search space to focus on promising lines identified through pattern matching.

To prevent unrealistic deep searches due to the reduced search space, a hard limit is imposed on the number of nodes stored in the game tree. We propose using a fixed-size table to map states to nodes, with a replacement strategy for handling collisions. Replacement strategies could prioritize nodes based on criteria such as:

- **high amount of visits** as they are likely important to the search. This strategy will be used in our first experiments.
- **high exploitation values** i.e. nodes that are known to be good.

3.3 Learning (System 1 and 2)

System 2 operates purely based on the game's rules and requires no prior preparation to function effectively. In contrast, System 1 relies entirely on the recognition of learned patterns, utilizing the expert iteration framework [2] as shown in Figure 2.

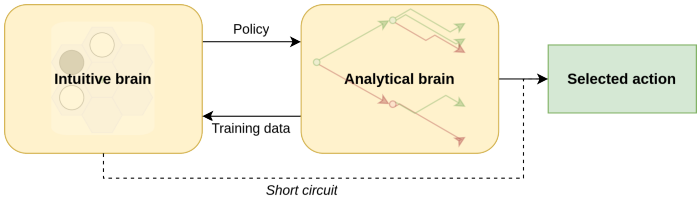

Fig. 2. Co-dependency between the two systems of DPTC is illustrated as a back-and-forth, the intuitive brain guiding the analytical MCTS algorithm during its iterations by reducing the search space drastically, while the results of this deeper analysis are learnt from to sharpen the intuition of the model.

When learning a new game, the intuitive brain starts as a blank slate, much like a child encountering a game for the first time. It has no insights to guide its decisions, relying entirely on the MCTS algorithm for analysis. Early strategies are basic or arbitrary, as planning ahead without understanding long-term consequences is ineffective. Our goal is twofold:

- We want to avoid a scenario where the agent's learning curve is slow at the start of training due to non-existent heuristics and the removal of the simulation phase of the MCTS algorithm.
- We want to allow the agent to infer *long-term rewards* rather than forcing it to rely only on the relatively short-sighted analytical brain.

Initially, strategies may appear suboptimal or random, as effective planning is impossible without intuition about the long-term consequences of actions. Only after the game concludes does the intuitive brain begin processing events and forming insights about pattern values. Critical learning happens between games, allowing the brain to refine its understanding. This method aligns with reinforcement learning principles: after each game, the agent reflects on the outcomes, assigning game-theoretic values to all actions taken, deepening its understanding of strategic moves.

4 Evaluation framework

The model is evaluated in the context of the move matching task [20] on the game of Renju and compared to four other methods:

- Rapfi [3], currently the best rated Renju agent on Gomocup.[4], based on $\alpha\beta$ search with an NNUE-based [22] evaluation of states.
- A pure UCT algorithm without bias.
- A UCT algorithm biased using System 1 as a policy, but without narrowing down the search space by filtering actions.

[3] Rapfi engine repository: `https://github.com/dhbloo/rapfi`
[4] Gomocup website: `https://gomocup.org/`

– A random player, choosing actions uniformly to form a baseline.

The game representation remains consistent across all evaluated methods to ensure that any observed differences in performance are not due to variations in the representation itself. All relevant code is made available in a public repository [5] to facilitate transparency and make the experiments reproducible.

The three models are evaluated using the **move matching task**, where each Renju position and human move is tested for accuracy. A correct prediction scores 1, and an incorrect one scores 0. This provides a statistical measure of human-likeness by quantifying how closely the agent's decisions align with actual human gameplay.

Our dataset is sourced from the complete archive of tournament games available on the Renju International Federation's (RIF) website [6]. Player ratings, computed using the Whole History Rating (WHR) algorithm [9], were not originally included. To account for the advantage of playing black [28], we excluded the first three moves from the 26 standard opening patterns, as they are based on established theory and less relevant to our evaluation. This resulted in a dataset of nearly 2 million positions, with a normal rating distribution centered around 2200 points. From this, approximately $50,000$ positions were selected for a final fixed dataset with the same rating distribution. Each agent was allocated 5 seconds of thinking time and one thread, with benchmarks running on an AMD Ryzen 5 3600 CPU.

5 Results

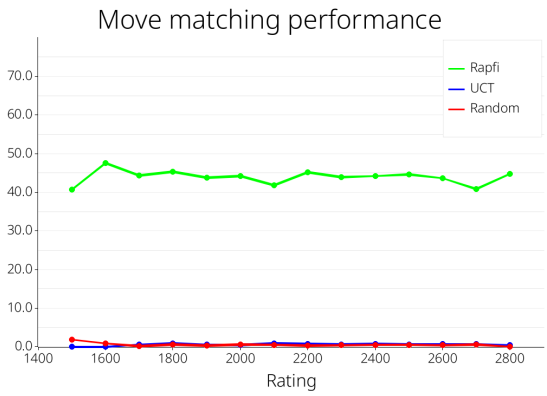

Fig. 3. Comparison of the move matching performance of 3 agents.

[5] Romoku: `https://github.com/aloisrtr/romoku` and move matching software: `https://github.com/aloisrtr/renju_move_matching`

[6] RIF Website: `https://www.renju.net/`, database dating of the 21/08/2024

Our results follow what was observed by the Maia Chess team for chess, in that the Rapfi engine's accuracy in move matching stagnates at an average accuracy of 44%. However, the accuracy does not increase monotonically with players' ratings, instead staying relatively consistent across the whole rating range.

The unbiased UCT agent's results reflected the difficulty of Renju for MCTS-based approaches. It rarely matched human moves, regardless of skill level.

6 Conclusion and future work

These results validate our approach as an important first step toward developing human-like game-playing agents. We now plan to extend these concepts to General Game Playing (GGP) [12], using generalized State-Action Features [25] as a replacement for the Neural Network architecture currently used for System 1.

References

1. Alami, R., Clodic, A., Montreuil, V., Sisbot, E.A., Chatila, R.: Task planning for human-robot interaction. In: Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-Aware Services: Usages and Technologies. p. 81–85. Association for Computing Machinery (2005)
2. Anthony, T., Tian, Z., Barber, D.: Thinking fast and slow with deep learning and tree search. CoRR **abs/1705.08439** (2017)
3. Arora, S., Doshi, P.: A survey of inverse reinforcement learning: Challenges, methods and progress. Artificial Intelligence **297**, 103500 (2021)
4. Browne, C.B., Powley, E., Whitehouse, D., Lucas, S.M., Cowling, P.I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., Colton, S.: A survey of monte carlo tree search methods. IEEE Transactions on Computational Intelligence and AI in Games **4**(1), 1–43 (2012)
5. Campbell, M., Hoane, A., hsiung Hsu, F.: Deep blue. Artificial Intelligence **134**(1), 57–83 (2002)
6. Carroll, M., Shah, R., Ho, M.K., Griffiths, T., Seshia, S., Abbeel, P., Dragan, A.: On the utility of learning about humans for human-ai coordination. In: Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019)
7. Charness, N., Reingold, E.M., Pomplun, M., Stampe, D.M.: The perceptual aspect of skilled performance in chess: Evidence from eye movements. Memory & Cognition **29**(8), 1146–1152 (2001)
8. Chase, W.G., Simon, H.A.: Perception in chess. Cognitive Psychology **4**(1), 55–81 (1973)
9. Coulom, R.: Whole-history rating: A bayesian rating system for players of time-varying strength. In: International conference on computers and games. pp. 113–124. Springer (2008)
10. De Groot, A.D., De Groot, A.D.: Thought and choice in chess, vol. 4. Walter de Gruyter (1978)
11. Finkelstein, L., Markovitch, S.: Learning to play chess selectively by acquiring move patterns. ICCA Journal **21**(2), 100–119 (1998)

12. Genesereth, M., Love, N., Pell, B.: General game playing: Overview of the aaai competition. AI magazine **26**(2), 62–62 (2005)
13. Goertzel, B., Lian, R., Arel, I., De Garis, H., Chen, S.: A world survey of artificial brain projects, part ii: Biologically inspired cognitive architectures. Neurocomputing **74**(1-3), 30–49 (2010)
14. de Groot, A.D., Gobet, F.R., Jongman, R.W.: Perception and memory in chess. J. Int. Comput. Games Assoc. **19**, 183–185 (1996)
15. He, T., Liao, L., Cao, Y., Liu, Y., Liu, M., Chen, Z., Qin, B.: Planning like human: A dual-process framework for dialogue planning (2024)
16. Kahneman, D.: A perspective on judgment and choice: mapping bounded rationality. Am Psychol **58**(9), 697–720 (2003)
17. Kocsis, L., Szepesvári, C.: Bandit based monte-carlo planning. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) Machine Learning: ECML 2006. pp. 282–293. Springer Berlin Heidelberg (2006)
18. Levinson, R., Snyder, R.: Adaptive pattern-oriented chess. In: Birnbaum, L.A., Collins, G.C. (eds.) Machine Learning Proceedings 1991, pp. 85–89. Morgan Kaufmann (1991)
19. Mańdziuk, J.: Towards cognitively plausible game playing systems. IEEE Computational Intelligence Magazine **6**(2), 38–51 (2011)
20. McIlroy-Young, R., Sen, S., Kleinberg, J., Anderson, A.: Aligning superhuman ai with human behavior: Chess as a model system. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining. KDD '20, ACM (2020)
21. Milani, S., Juliani, A., Momennejad, I., Georgescu, R., Rzepecki, J., Shaw, A., Costello, G., Fang, F., Devlin, S., Hofmann, K.: Navigates like me: Understanding how people evaluate human-like ai in video games. In: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems. CHI '23, ACM (2023)
22. Nasu, Y.: Efficiently updatable neural-network-based evaluation functions for computer shogi. The 28th World Computer Shogi Championship Appeal Document **185** (2018)
23. Russell, S.: Learning agents for uncertain environments (extended abstract). In: Proceedings of the Eleventh Annual Conference on Computational Learning Theory. p. 101–103. COLT' 98, Association for Computing Machinery (1998)
24. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., Hassabis, D.: Mastering the game of go with deep neural networks and tree search. Nature **529**(7587), 484–489 (Jan 2016). `https://doi.org/10.1038/nature16961`, `https://doi.org/10.1038/nature16961`
25. Soemers, D.J.N.J., Piette, É., Stephenson, M., Browne, C.: Spatial state-action features for general games. CoRR **abs/2201.06401** (2022), `https://arxiv.org/abs/2201.06401`
26. Tucker, A., Gleave, A., Russell, S.: Inverse reinforcement learning for video games. CoRR **abs/1810.10593** (2018)
27. Turnwald, A., Wollherr, D.: Human-like motion planning based on game theoretic decision making. International Journal of Social Robotics **11**(1), 151–170 (2019)
28. Wágner, J., Virág, I.: Solving renju. ICGA journal **24**(1), 30–35 (2001)