nature human behaviour

Article

https://doi.org/10.1038/s41562-023-01696-5

Self-orienting in human and machine learning

Received: 5 September 2022

Accepted: 7 August 2023

Published online: 31 August 2023

Check for updates

Julian De Freitas[1] ✉, Ahmet Kaan Uğuralp[2], Zeliha Oğuz-Uğuralp[3], L. A. Paul[4], Joshua Tenenbaum[5] & Tomer D. Ullman[6]

A current proposal for a computational notion of self is a representation of one's body in a specific time and place, which includes the recognition of that representation as the agent. This turns self-representation into a process of self-orientation, a challenging computational problem for any human-like agent. Here, to examine this process, we created several 'self-finding' tasks based on simple video games, in which players ($N = 124$) had to identify themselves out of a set of candidates in order to play effectively. Quantitative and qualitative testing showed that human players are nearly optimal at self-orienting. In contrast, well-known deep reinforcement learning algorithms, which excel at learning much more complex video games, are far from optimal. We suggest that self-orienting allows humans to flexibly navigate new settings.

A current proposal for a computational notion of self is a representation of one's body in a specific time and place, which includes the recognition of that representation as the agent. This turns self-representation into a process of self-orientation, a challenging computational problem for any human-like agent. While there has been much work on 'the self' in philosophy, psychology and neuroscience[1–13], few that we know of show how the self can be concretely represented in artificial intelligence (AI) algorithms[14,15]. In our theoretical paper[16], we introduced the notion of a 'computational self' and explained its role in solving a basic problem faced by any intelligent agent—human or artificial—that learns, thinks and acts for itself. In that paper, we argue for a computational notion of 'self', a representation that points to an entity in the world and tags it as the agent that is doing the representing and taking actions in the world, propose that existing AI probably do not have a computational self-representation in this sense, and explore the case for how such a self can be concretely represented in AI algorithms. Paul et al.[16] suggest this representation and process is crucial for flexible learning and action in humans, that many new environments require humans to first solve this process, and that the computational challenge that humans can solve in a general-purpose way is linking to the correct self-representing entity across situations and environments.

In this Article, building on this theoretical work, we test for a computational notion of 'self'. We refer to the process of identifying this entity as 'self-orienting'. We suggest this representation and process is crucial for flexible learning and action in humans, and that many new environments require humans to first solve this process before achieving other goals. By building on past work on reinforcement learning (RL)[17–21], game playing[22,23] and cognitive science work on RL and game playing[24–26], we also propose that existing AI probably do not have a computational self-representation in our sense, and that, while some algorithms may be trained to carry out self-orienting in particular environments, they do so through particular brittle cues rather than a general-purpose process. The computational challenge that humans can solve in a general-purpose way is linking to the correct self-representing entity across situations and environments.

Related to the current work, Hierarchical Attentive Multiple Models for Execution and Recognition (HAMMER) uses perceptual perspective-taking to learn both an egocentric sensory space and a similar space for another agent, including the possible goals and actions of that agent[15]. The notion of needing to identify with an avatar that matches one's actions and orient to the perspective of that avatar to successfully play the game is akin to the notion of self-orienting explored in the current manuscript. HAMMER has since been incorporated in the Distributed Action Control architecture[14], where its perspective-taking capabilities were integrated with other AI models related to the sense of self, and it is also being leveraged to develop

[1]Marketing Unit, Harvard Business School, Boston, MA, USA. [2]Department of Computer Engineering, Bilkent University, Ankara, Turkey. [3]Department of Psychology, Bilkent University, Ankara, Turkey. [4]Department of Philosophy, Yale University, New Haven, CT, USA. [5]Brain & Cognitive Sciences Department, MIT, Boston, MA, USA. [6]Psychology Department, Harvard University, Boston, MA, USA. ✉e-mail: jdefreitas@hbs.edu

AI systems that can reason about the activities of other agents[27] and develop a self-other distinction[28]. This work is inspired by the development of perspective-taking and imitation in humans[29], in which children initially have a limited ability to engage in perspective-taking that hampers their environment[30].

Also, related to the current work, Simultaneous Localization and Mapping addresses the problem of constructing or updating a map of an unknown environment while simultaneously estimating the agent's position within it[31], a critical ability that allows robots such as autonomous vehicles and drones to navigate in diverse environments. In contrast, the current work stress tests the ability to flexibly self-orient via dedicated self-orienting tasks involving multiple possible selves (rather than just a single given self), and asks whether such environments can be navigated by popular RL agents.

To appreciate the importance of self-orienting, it is helpful to imagine what it is like to 'not' have a correct computational self-representation. Often, humans experience this feeling for only a few moments before resolving the issue. This happens, for instance, when we do not know where we are in a library without windows, or when we wake up in a cold sweat and forget that we are in a hotel in Paris, or when we start a completely new video game, and have no sense yet which entity in the game is us, or what we can do with it. Games often single out a particular entity in the game as the player's 'Avatar', and give it particular action affordances and a point of view and in general centre the game around it.

In Paul et al.[16], we argue that finding out which avatar you are in a new game is a particularly useful way to explore our sense of a computational self-representation. This process is fast, and automatic: often we do not even think of how quickly we resolve the fact that of all the entities in the game the giant pink cat is 'us'. In both games and life there are many possible cues that can help us in self-orientation. In a game, we might futz with a controller, or simply be told 'you are the giant cat'. In the library, we may consult a map to spatially orient ourselves. In the hotel room, the tactile touch of the bedsheets may inform us we are in a hotel, and the clock face may orient us in time. The point is not that humans are good at exploiting any single one of these cues, but that they do so in service of a larger unified goal: to orient themselves in terms of space, time and identity.

In everyday life, we are constantly achieving the computational feat of self-orienting, the equivalent of identifying an avatar in a game, but with our body and its whereabouts as the avatar. The ease with which we perform self-orienting belies its complexity and importance: the ability to accurately self-represent ourselves is a crucial part of how we flexibly navigate different environments. When confronted with a new setting, we do not need to learn everything from scratch. Rather, we efficiently self-orient, and proceed to plan from there.

In the current work, we extend the proposal of self-orienting[16] to investigate the extent to which humans and well-known AI algorithms from RL are capable of self-orienting. To do this, we created a set of increasingly complex 'self-finding' tasks that deliberately make it challenging to find one's self. The tasks act as litmus tests of whether an agent is capable of flexible self-orienting. We compare both humans and RL algorithms to optimal play, as well as to each other, asking whether they can solve these tests, how quickly they can do so and how they do so. We find that humans exhibit near-optimal play across a variety of tasks. By contrast, well-known RL algorithms are not able to generalize across multiple settings, nor when a given setting is perturbed. We note that the algorithms that we use are not the cutting edge of contemporary AI, and cannot be representative of the latest frontiers in deep RL. Rather, we chose these algorithms because they are well known, well studied and among the most popular and influential baselines for building agents that operate autonomously in some environment, and there was no principled a priori argument for why they definitely should not be able to pass our tasks, given their human or super-human performance on many challenging environments. These algorithms also embody the thesis of RL that some prominent AI researchers[32–34] have suggested is a scaling route to building fully general AI with human-level intelligence or beyond.

Self-finding games

All four games featured four agents (also known as 'possible selves') indicated by red squares. Crucially, only one of these agents (the 'digital self') was controlled by the player's keypresses. To complete a level, the player had to navigate their digital self to a goal (indicated in green) by moving through unimpeded spaces (black) and avoiding wall boundaries (grey). In each of the four games, there were four basic moves: Left, Right, Up or Down, which human players enacted by pressing the arrow keys, although the arrows did not necessarily correspond to the resulting action.

In principle, the games could be solved without self-orientation. However, our hypothesis was that for humans each game level naturally consisted of two phases: (1) 'self-orienting', in which the player figures out which of several possible selves is their real digital self (their avatar), and (2) 'navigation', in which the player moves the digital self to a rewarding goal.

Each game consisted of 100 levels (except for study 4c). The levels of each game obeyed the overall rules of the game, while varying the starting position of the different entities (agents, avatar and walls).

To measure general self-orienting, the different games were designed such that agents had to exploit different kinds of cues to successfully self-orient.

Human and artificial players

Human participants had little instruction or no feedback of any kind, to not give them an advantage over the AI algorithms. To test if they played optimally, we compared their performance with that of a 'self-class' that we hard coded to solve each game optimally: first, it found the digital self by taking informative actions that disambiguated the most possible selves simultaneously; second, after identifying the digital self, it navigated it to the goal. Finally, to assess the abilities of well-known game-playing RL algorithms, each game was played for 2,000 levels by the following RL algorithms: DQN, TRPO, PPO2, A2C, ACER and OC. These algorithms (also known as pixel-based RL baselines) use a combination of convolutional and fully connected neural network layers to learn from frame-by-frame images of the game. They received a reward of 1 for completing a level of the game. As a control, we also ran the games through a random policy that took random actions.

Study 1: The Logic Game

In the Logic Game (Fig. 1), we predicted that human players would rapidly learn the optimal strategy (disambiguate which agent was their avatar, then navigate to the goal), whereas RL baselines would not. We expected that this would be because humans already had the correct goal of eliminating options for the digital self, whereas RL agents did not have this goal, but were simply learning state–action pairs. RL agents should thus be inefficient at self-orienting, because they would be unable to learn from 'non-events' in which actions led to no visible effects. More generally, it is possible that the reason these agents cannot learn from such an event is because they suffer from a more basic problem: they do not have a notion of self-orienting at all.

In this and in the other games, our measure of performance is the number of steps taken to complete each level. How did human players compare with the self-class and artificial agents on this measure? Figure 2a,b shows that humans rapidly reached optimal play after approximately just one level, performing indistinguishably from the optimal self-class thereafter. Bayes factor *t*-tests between human players and the self-class revealed that humans begin to perform indistinguishably from the self-class after just five levels.

Contrasting with human players, the AI agents played for several hundred levels before their performance plateaued. Notably, even

https://doi.org/10.1038/s41562-023-01696-5

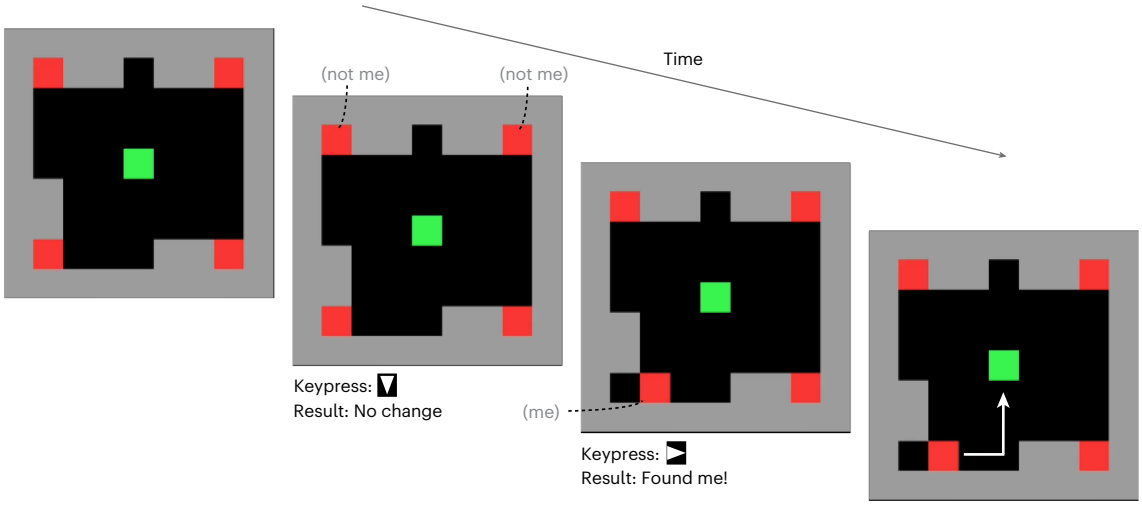

Fig. 1 | The Logic Game. There are four agents (red blocks), one of which is your avatar. The level ends when the avatar reaches the goal (green block). In this example, moving down disambiguates the most possible selves (red)—the top two. If moving down produces no visible change, then you must be one of the bottom two agents. To disambiguate which of these bottom agents is your digital self, it is now equally informative to move right or up. Moving right reveals that the digital self was in the bottom left corner. Knowing this, you navigate it to the reward (green).

after 2,000 levels, most AI players (ACER, TRPO, A2C and OC) did not reach human-level performance (Fig. 2c and Supplementary Table 1). In short, although the artificial agents improved in their performance, humans learned far more quickly and played more optimally.

What explains this difference between human and artificial players? One possibility is that it arises from the self-orienting phase of the game. Since human players reach optimal play, they must be able to eliminate candidates for their digital self, even when their actions produce no visible displacements (which occurs when their keypress leads the digital self to try moving through an immovable wall). It is possible that artificial players are not able to learn from such 'non-events'. To investigate this possibility, we also plotted the number of steps taken until the first visible displacement occurred, which we treat as the moment when players successfully self-oriented (Supplementary Fig. 1). We find that none of the algorithms shows a noticeable improvement in how quickly they self-orient; in other words, although algorithms learned how to navigate to the reward (Fig. 2a), they do not learn how to optimally self-orient (Supplementary Fig. 1), whereas human players rapidly reach optimal levels of self-orienting (Fig. 2b and Supplementary Fig. 1). Notably, the average correlation between the number of steps taken until self-orienting (the first visible displacement) and the total number of steps was higher than the correlation of the random agent ($M_{r_{\text{Human}}} = 0.85$, $M_{r_{\text{Random}}} = 0.07$; $t(20.2) = 27.6$; $P < 0.001$; Cohen's $d = 8.72$; 95% confidence interval (CI) 0.73 to 0.85; Fig. 2c), suggesting that performance depended on successful self-orientation. As for the last hundred levels of the artificial agents, all RL agents also had correlations significantly higher than the random agent (all P values <0.001; $M_{r_{\text{DQN}}} = 0.99$, $M_{r_{\text{A2C}}} = 0.97$, $M_{r_{\text{PPO2}}} = 0.93$, $M_{r_{\text{TRPO}}} = 0.89$, $M_{r_{\text{ACER}}} = 0.56$, $M_{r_{\text{OC}}} = 0.24$), supporting the idea that all of these agents were able to optimally navigate to the reward after the first visible displacement occurred, even though they did not learn how to optimally self-orient (Supplementary Fig. 1). As a sanity check, we confirmed that the self-class showed a perfect correlation between self-orienting and performance ($M_r = 1.0$). A final way to compare human and artificial players is to examine their behavioural patterns within the gaming environment over time. The heat maps in Fig. 2d show the patterns of each player across time, broken down for the first and last hundred levels. For the first hundred levels, we see that only the self-class resembles human players, with clear horizontal moves near the reward. In contrast, the artificial players move in a more dispersed fashion, and spend more time in the corners. By the last hundred levels, however, one of the AI players, DQN, begins to resemble human players. In Supplementary Information, we compare human and AI players, and find that most artificial players are not similar to humans even after 2,000 levels of training. We also speculate on why only DQN resembles human players, and why artificial agents have suboptimal behaviour.

In sum, the behaviour of human players was consistent with a strategy that first disambiguates which of several possible selves the player is meant to identify with (self-orienting), and only then pursues more explicit goals such as navigation. Even when not receiving (1) any description of how the game works or (2) any explicit instruction to navigate to the goal in as few moves as possible, and even when (3) their actions led to no visible change, human players took informative actions to optimally rule out candidates for their digital self.

While several well-known pixel-based RL algorithms learned to play the game more efficiently over time, they never played optimally, because they did not optimally self-orient in a game where some actions have no observable effects. Even at the end of a long learning process, most RL agents' movement patterns did not indicate a self-orienting phase. These results do not mean that no state-of-the-art algorithm could solve the game as efficiently as humans did. In fact, the self-class is a very simple such algorithm. But to the extent that humans outperformed all our standard well-known game-playing algorithms, this underscores the efficiency with which they localized their digital selves in new digital settings, and points to a missing representation and process in well-known RL agents. In fact, the results are consistent with, and support, the view that the RL agents never learn to self-orient.

Study 2: Contingency Game

The Logic Game shows the most bare-bones dynamic of self-orienting: a single step or two is sufficient for the first part of the task, and only one entity moves (at most). However, to bring the task closer to some of the opening examples such as a four-player split-screen games, the Contingency Game explores another way in which human players might self-orient: by exploiting informative contingencies. This time, whenever the player pressed a key, all possible agents moved, even though only one agent was truly controlled by the player. To orient on their digital self, players needed to eliminate from consideration those avatars that moved in unexpected directions after a given keypress (Fig. 3).

Again, we predicted that human players would go through a two-step process, first self-orienting (figuring out which entity is their

https://doi.org/10.1038/s41562-023-01696-5

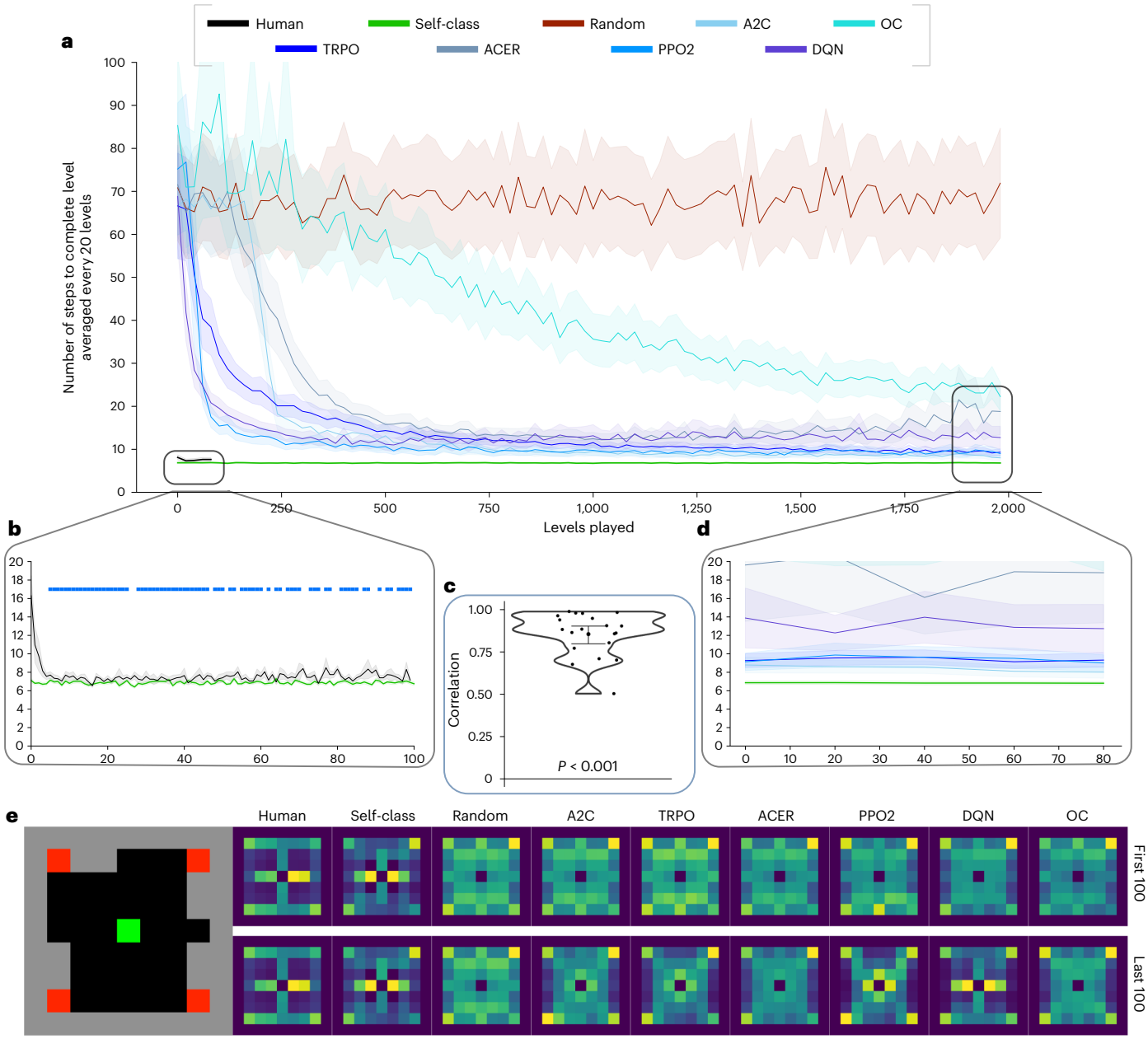

Fig. 2 | Results of study 1 (Logic Game). a, Number of steps taken by all agents, averaged every 20 levels. Error shading reflects standard error of the mean. **b**, Zooming in on level-by-level human players and the self-class for the first hundred levels. Blue horizontal lines above the plot indicate levels where human performance is indistinguishable from optimal (that is, levels where the line is visible have a Bayes factor (BF_{01}) above 1.0). **c**, Violin plot showing the correlations between the number of steps until the first visible displacement occurred and the total step count, for each participant. Error bars reflect 95% CIs, centre of the error bar reflects the mean, and the *P* value beneath the plot shows the significance resulting from two-sided *t*-test comparing the correlation values with that of a random agent. **d**, Zooming in on artificial players for the last hundred levels, averaged every 20 levels. **e**, Heat maps of normalized action patterns for the first hundred levels (top row) and last hundred levels (bottom row), with human performance for first hundred levels included for comparison. Yellow shows the most visited, and purple shows the least visited, locations by the digital self.

avatar) and then navigating towards a goal. In contrast to the Logic Game, we expected that this time the pixel-based RL algorithms would eventually learn to play the game and be close to optimal, given that every action in the game leads to an observable result (unlike in study 1). Even so, we expected that the algorithms would require more levels of learning than human players before reaching optimal play, and would not have a notion of self-orienting.

Figure 4a,b shows that human players quickly plateaued, reaching close to optimal play on the first level, and then fluctuating in and out of optimal play thereafter. On average, humans took significantly more steps than the self-class (M_{human} = 14.33, $M_{self\text{-}class}$ = 9.46; t(19.5) = 6.4; P < 0.001; Cohen's d = 2.04; 95% CI 3.29 to 6.44). One likely reason that they played suboptimally is that they took the 'lazy' strategy of initially finding the digital self by repeatedly hitting the same key in one direction, for example, pressing right until one agent was clearly more displaced than the others. This is suggested by the human heat map (Fig. 4e), where we see horizontal lines emanating outwards from the starting location (indicated in blueish green). This behaviour is not strictly suboptimal since players were never explicitly instructed to complete the game in as few moves as possible. The strategy can even be considered optimal from the standpoint of saving cognitive effort, because it is easier to just hit one key until one avatar clearly pops out

https://doi.org/10.1038/s41562-023-01696-5

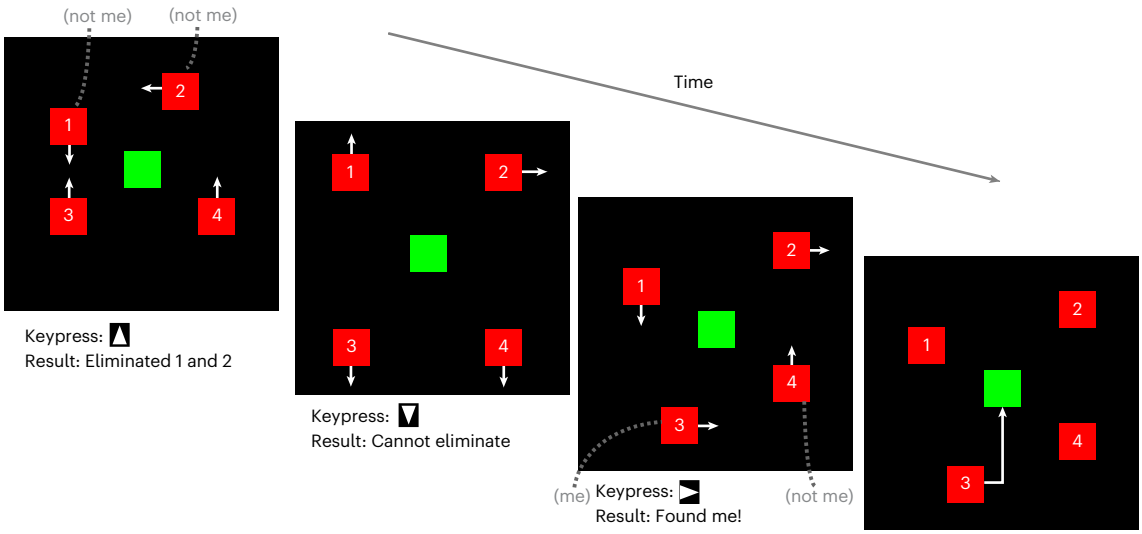

Fig. 3 | The Contingency Game. In this example, moving up eliminates the top two candidate selves (1 and 2), which do not move in the direction of the keypress. In frame 2, moving down does not help you find your digital self, since by chance both the remaining possible selves (that is, 3 and 4) move down. In frame 3, moving right eliminates another candidate self (4), disambiguating your digital self. Going forward, you can navigate the digital self (3) to the reward.

than to attend to which of four avatars is consistently responding contingently to changes in your keypress. In other words, the cost of moving away from the goal to find the avatar is much less than waiting to think about the optimal move[35].

In contrast to human players, the artificial players required several hundred levels before their performance plateaued (Fig. 4a). Unlike in study 1, some artificial players (DQN, A2C and PPO2) achieved optimal performance by the end of training (Fig. 4d and Supplementary Table 2). This is likely because seeing an observable consequence for each action enabled the algorithms to learn. DQN was the most reliable AI player, consistently staying close to optimal performance, while A2C sometimes achieved optimality but also experienced periods of suboptimal performance. DQN's success as the top artificial player could be attributed to its suitability for discrete environments and its exploration strategy ('RL algorithms' in Supplementary Information). In short, human players learned quicker than algorithms in this game, yet they fluctuated in and out of optimal performance, whereas some AI players eventually played optimally.

Did human and artificial players follow similar behavioural patterns? The heat maps in Fig. 4e show that, for the first hundred levels, artificial players exhibited more dispersed behavioural patterns than humans and some RL algorithms—such as TRPO, ACER and PPO2—appear to have gotten stuck in the corners. By the last 100 levels, the paths of the artificial agents were clearer ('Heat map results' in Supplementary Information).

As in study 1, another way to compare human and algorithmic players is to see how many moves they spend in the self-orienting and navigation phases of the game, relative to optimal play. While the Contingency Game does not allow us to definitively isolate the point when human players found their digital selves, we can still get a sense of when this occurred by plotting the average distance of the digital self from the reward across the steps taken within a given level, and comparing this with the self-class. Supplementary Fig. 2 plots this distance for each subsequent move in the first level and last level for human (level 100) and artificial players (level 2,000). We find that human players take a few extra steps before they begin navigating to the reward on level 1, but by the last level they are clearly optimal.

To more definitively establish the relationship between self-orienting and performance, supplemental replication study 2b tested whether participants who localize their digital selves earlier finish the game in fewer steps. In this version of the game, participants not only completed game levels but for each level were also asked to click on the square they believed they were controlling as soon as they found it (those who neglected to do so, or who clicked a square more than once, were shown error messages). This design discloses the self-finding aspect of the game, yet we found no difference in the average number of steps taken for the 100 levels of this game versus study 2a ($M_{original}$ = 14.33, M_{new} = 14.32; t(168.3) < 0.1; P = 0.994; Cohen's d < 0.01; 95% CI −1.81 to 1.82); BF_{01} 6.50), suggesting that participants played no differently. Notably, the average correlation for participants between the number of steps taken until self-orientation across levels and the total number of steps taken in those levels was significantly higher than this correlation for the random agent ($M_{r_{Human}}$ = 0.55, $M_{r_{Random}}$ = −0.01; t(30.8) = 10.6; P < 0.001; Cohen's d = 3.35; 95% CI 0.46 to 0.67; Fig. 4c), suggesting that performance depended on successful self-orientation (Methods).

As for the artificial agents, self-orienting started at random on level 1, but by the end of training some of the algorithms (DQN and A2C) were close to optimal (Supplementary Fig. 2). This suggests that some algorithms learned to behave in a way that is similar to self-orienting, while humans learned to optimally self-orient. If the artificial players truly learn how to efficiently self-orient, then they should be robust to environmental changes that affect the self-orienting task. To explore this, after the artificial agents learned for 2,000 levels, we added an additional 'mock possible self' to the game, which was coloured red like the other possible selves; in reality, the mock possible self was never controllable by the player's keypresses. After this mock agent was added, all AI players exhibited a decrement in efficiency, requiring ~700 more levels to recover pre-perturbation performance levels; although some algorithms, such as TRPO and PPO2, never do (Supplementary Fig. 3a), perhaps because these methods avoid using large policy updates, making them less flexible. This pattern suggests that the algorithms did not learn a robust self-orienting strategy.

In supplemental replication study 2c, human participants again played the first hundred levels of the game followed by the same mock agent perturbation shown to the algorithms, after which they played for 50 more levels. Although performance was slightly worse after the perturbation ($M_{original}$ = 13.98, $M_{perturbated}$ = 15.35; t(147.8) = −3.0; P = 0.003; Cohen's d = -0.43; 95% CI −2.25 to −0.48), it remained near optimal levels (Supplementary Fig. 3c). To compare post-perturbation

https://doi.org/10.1038/s41562-023-01696-5

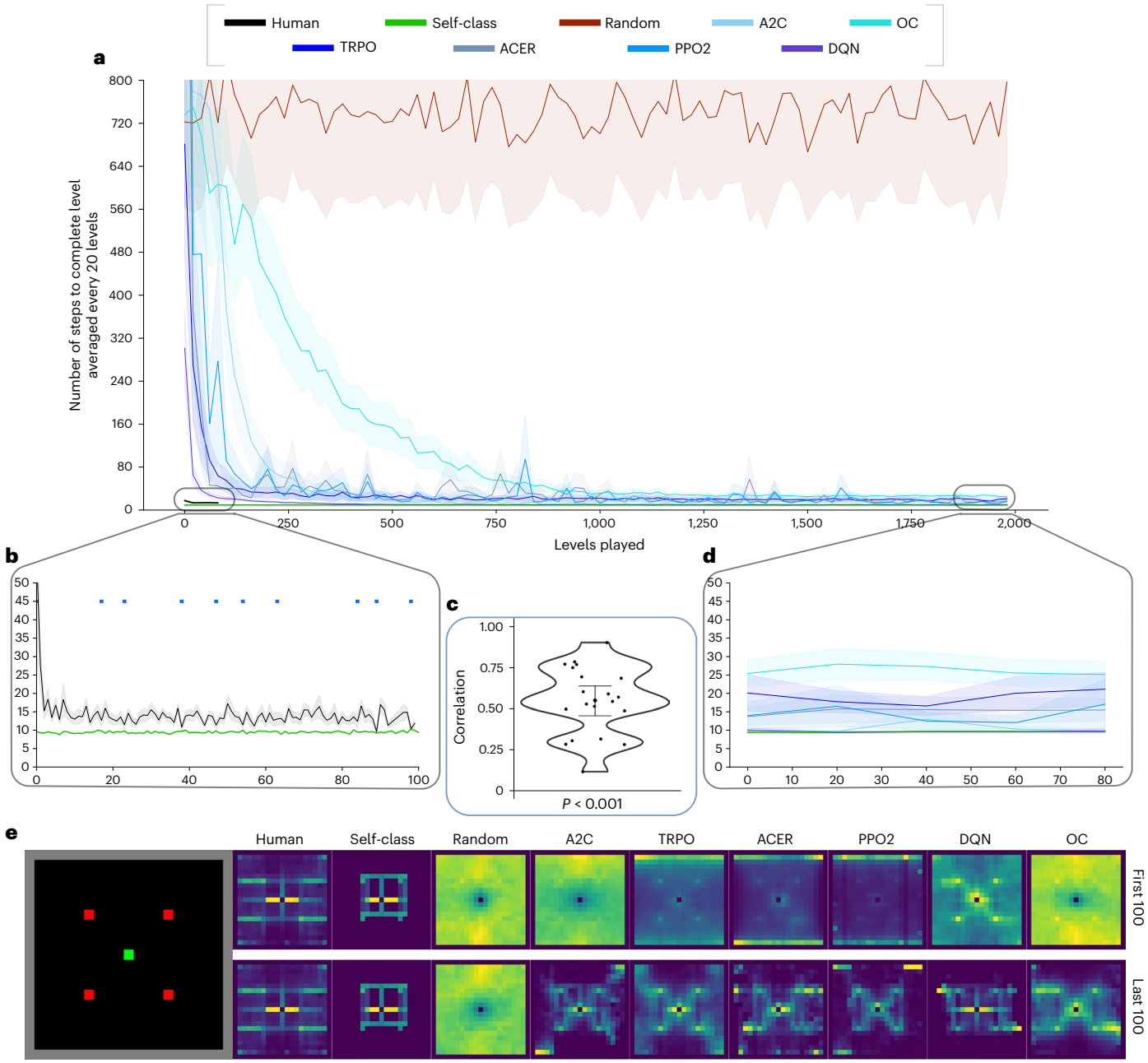

Fig. 4 | Results of study 2 (Contingency Game). a, Number of steps taken by all agents, averaged every 20 levels. Error shading reflects standard error of the mean. **b**, Zooming in on level-by-level human players and the self-class for the first hundred levels. Blue horizontal lines above the plot indicate levels where human performance was indistinguishable from optimal play (that is, levels where the line is visible have a Bayes factor (BF_{01}) above 1.0). **c**, Violin plot showing the correlations between the number of steps until self-orientation and the total step count, for each participant (study 2b). Error bars reflect 95% confidence intervals, centre of the error bar reflects the mean, and the *P* value beneath the plot shows the significance resulting from two-sided *t*-test comparing the correlation values to that of a random agent. **d**, Zooming in on artificial players for the last hundred levels, averaged every 20 levels. **e**, Heat maps of normalized action patterns for the first hundred levels (top row) and last hundred levels (bottom row, with human performance for first hundred levels included for comparison). Yellow shows the most visited, and purple shows the least visited, locations by the digital self.

performance of human players for 50 levels against that of algorithmic players for 2,000 levels, we averaged algorithmic performance at every 40-level interval. Average human performance exceeded all artificial models ($P < 0.05$) except for A2C, which was similar to human players ($M_{A2C} = 20.07$, $M_{Human} = 15.35$; $t(49.7) = 1.5$; $P = 0.130$; Cohen's $d = 0.31$; 95% CI −1.44 to 10.88; BF_{01} 1.66).

In short, study 2 presents the opposite challenge of study 1: how to self-orient when your actions are correlated with several (as opposed to no) changes in the environment. The solution is to focus on informative changes—moves that are consistent with one's keypresses—then narrow down candidates from there. Human players were able to quickly solve this problem at near-optimal levels, albeit by first taking 'lazy' steps to effortlessly disambiguate the digital self. The artificial agents were also able to solve the levels and reach optimal play—presumably because this time actions always led to observable consequences. Even so, these agent failed on the robustness test, suggesting they did not actually learn to self-orient and they did not learn and behave in the same way that people do on these tasks.

https://doi.org/10.1038/s41562-023-01696-5

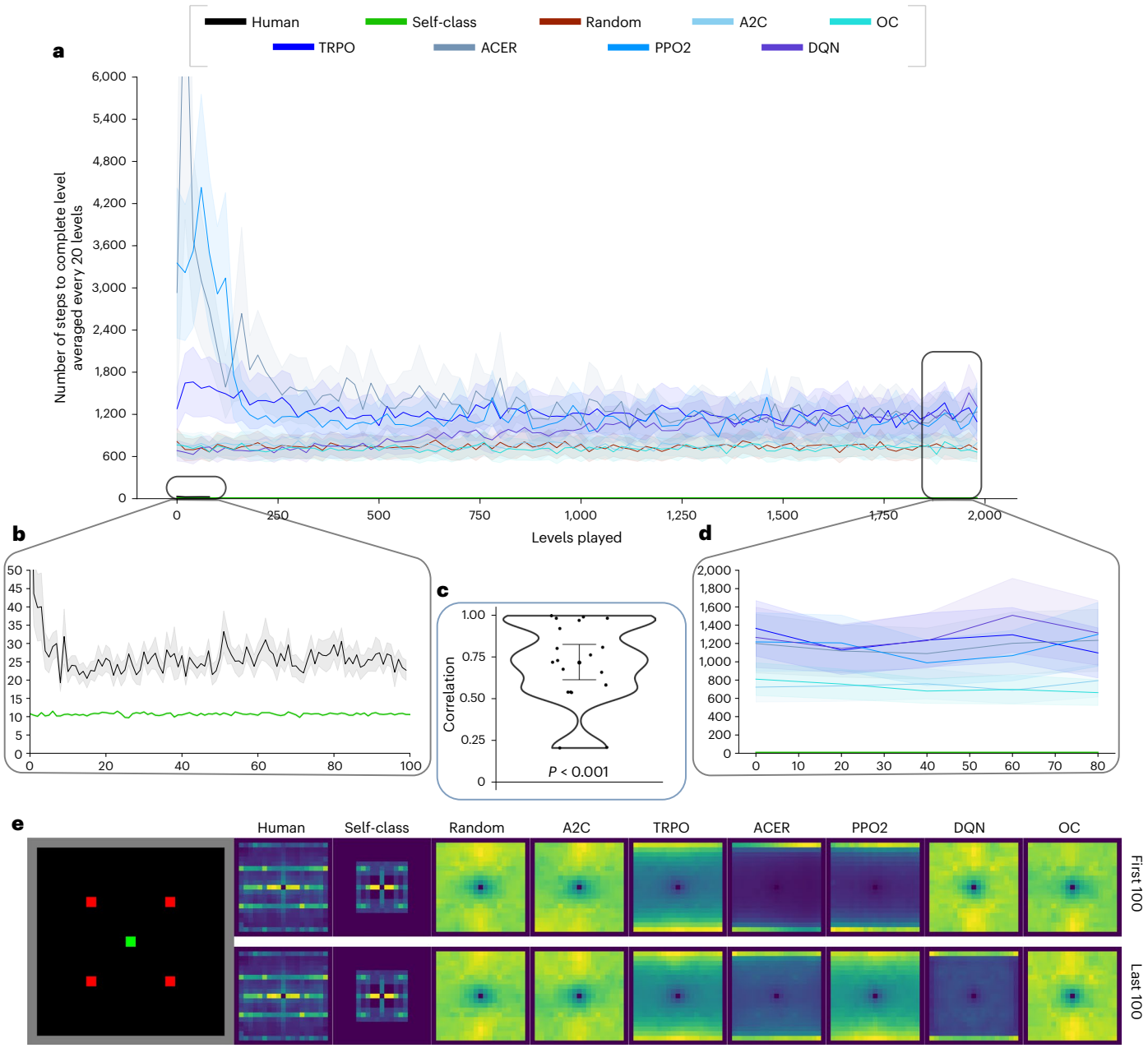

Fig. 5 | Results of study 3 (Switching Mappings Game). a, Number of steps taken by all agents, averaged every 20 levels. Error shading reflects standard error of the mean. **b**, Zooming in on level-by-level human players and the self-class for the first hundred levels. **c**, Violin plot showing the correlations between the number of steps until self-orientation and the total step count, for each participant (study 3b). Error bars reflect 95% confidence intervals, centre of the error bar reflects the mean, and the *P* value beneath the plot shows the significance resulting from two-sided *t*-test comparing the correlation values with that of a random agent. **d**, Zooming in on artificial players for the last hundred levels, averaged every 20 levels. **e**, Heat maps of normalized action patterns for the first hundred levels (top row) and last hundred levels (bottom row), with human performance for first hundred levels included for comparison. Yellow shows the most visited, and purple shows the least visited, locations by the digital self.

Studies 3 and 4 test the boundaries of human and algorithmic capabilities, by deliberately exploring more challenging variants of the Contingency Game in which key mappings change or a player is induced to lose track of the digital self.

Study 3: Switching Mappings Game

Study 3 explored a variant of the Contingency Game in which the mappings between keypresses and actions are randomly switched each level, requiring players to be more flexible than in studies 1 and 2. The switched mappings manipulation can be likened to controlling a new remote control, or figuring out the rules of a new game. Self-orienting in such contexts entails figuring out how your actions relate to the environment, rather than simply assuming that there is a predictable mapping between your keypresses and the observable consequences (as in study 2).

We predicted that switching key–action mappings on every level would prevent human players from playing optimally, because they would struggle to remember the new mappings. Even so, we expected that they would perform at near-optimal levels, because they would still employ an otherwise efficient self-orienting strategy. We also expected that, as before, playing this game requires a two-step process: self-orienting, followed by navigation. In contrast, we expected that

https://doi.org/10.1038/s41562-023-01696-5

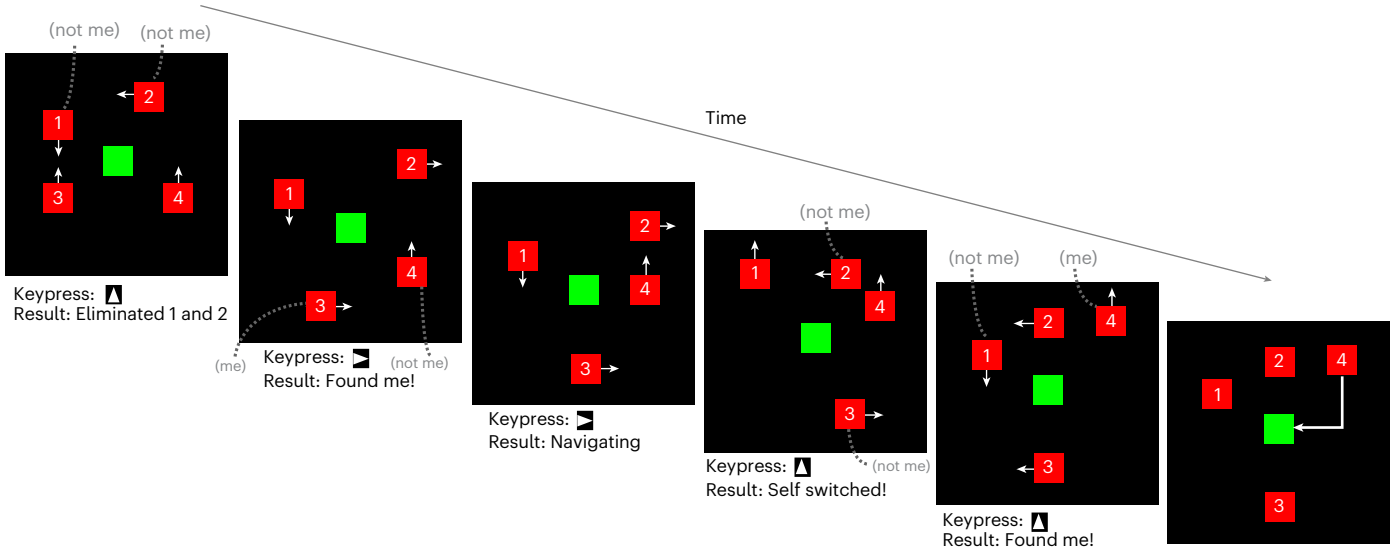

Fig. 6 | The Switching Embodiments Game. In this example, moving up on the first frame eliminates two candidate selves (1 and 2), because they do not move in the direction of the keypress. Moving right on the second frame eliminates the last possible self (4), revealing the digital self (3). On frame 3, you move right again to navigate to the reward. On frame 4, you attempt to navigate the digital self up, but avatar 3 moves in an unexpected direction (rightwards)—your digital self has switched embodiments. Meanwhile, agents 1 and 4 did move up, so they are the new candidate digital selves. In the next step, you try to disambiguate the digital self by moving up again, and notice that only avatar 4 moves up. This is your new digital self, so you navigate it to the reward.

the RL algorithms would not be able to learn the general strategy of self-orienting followed by navigation, but rather only locally learn key mappings to reward. Such local learning is particularly susceptible to key switches, and so we would expect the RL agents to mostly fail on this task, further underscoring the flexibility of human players.

Figure 5a,b shows that human players learned quickly at the beginning, then consistently underperformed the self-class. On average, human players took significantly more steps than the self-class during their 100 levels of gameplay ($M_{human} = 26.66$, $M_{self\text{-}class} = 10.70$; $t(19.0) = 7.2$; $P < 0.001$; Cohen's $d = 2.27$; 95% CI 11.31 to 20.62). A likely reason for this performance gap is that human players struggled to remember constantly switching key mappings and battled an inconsistency with strong prior expectations about how arrow keys relate to actions, for example, expecting that '←' means left. However, this gap does not necessarily mean that humans could not have performed better had they been instructed to play as efficiently as possible. Also, human players still performed at near-optimal levels, suggesting that they otherwise employed an effective self-orienting strategy (Fig. 5).

Strikingly, the artificial agents were not able to learn the game at all (Fig. 5a–d and Supplementary Table 3). Supplemental analyses suggest this is because the learning that occurred for these algorithms was specific to a given key mapping, rather than generalizable across key mappings.

Did human and artificial players follow similar behavioural patterns? When we shuffled the key–action mappings after each level, the behaviours of the artificial agents were dispersed and random-looking (Fig. 5e), appearing different from human play. Furthermore, as in study 2b, study 3b found that participants who localize their digital selves earlier finish the game in fewer steps, suggesting that performance depended on successful self-orientation ('Supplementary results' in Supplementary Information).

It is important to emphasize the qualitative difference between human and RL agent behaviour on this task. Human players seem to follow the general strategy of self-orientation, by figuring out which keys map to which actions and disambiguating their avatar, then navigating to the goal. They also seem to learn the overall patterns and rules of the game: 'in each level, the arrows are scrambled, and I need to re-figure them out, then I can navigate'. By contrast, the RL agents seem to do nothing of this sort, and are unable to learn general patterns and strategies even following thousands of levels. In sum, study 3 presented another challenge beyond study 2: self-orienting when key–action mappings are unpredictable. Humans were able to solve the game, albeit by performing consistently at near-optimal levels. In contrast, the added challenge rendered the pixel-based RL algorithms incapable of solving the game altogether, suggesting that they did not learn a general policy for self-orienting. Instead, their policies were overspecialized to specific key–action mappings.

Study 4: Switching Embodiments Game

Study 4 explored a variant of the Contingency Game in which the digital self periodically switched embodiments within each level, causing players to temporarily lose track of their digital selves (Fig. 6). This disruption required players to be even more flexible than in the previous games, repeatedly self-orienting and navigating. The manipulation can be likened to getting lost, as when your digital self crosses paths with another avatar in a crowded virtual setting.

We predicted that human players would find the game challenging but would eventually learn to play optimally, because not doing so would aversively increase how long it took them to complete the game. Unlike previous games, we expected that behaviour would not follow a two-step process of self-orientation followed by navigation, but rather interleaved bursts of the two. As for the pixel-based RL algorithms, we did not have strong predictions about whether they would be able to learn the game. On the one hand, actions always had observable consequences, which was useful for RL agent learning in study 2. On the other hand, RL agents might not be able to handle the embodiment switch, because this would only occur after several steps (although the switches did always occur after a consistent number of steps). Either way, we expected that the algorithms would be less efficient than human players.

Results

Figure 7a,b shows that human players learned quickly at the beginning, then reached optimal performance after ~30 levels. Why did human players reach optimal performance in the Switching Embodiments Game, but not the Switching Mappings Game? Most likely, the Switching

https://doi.org/10.1038/s41562-023-01696-5

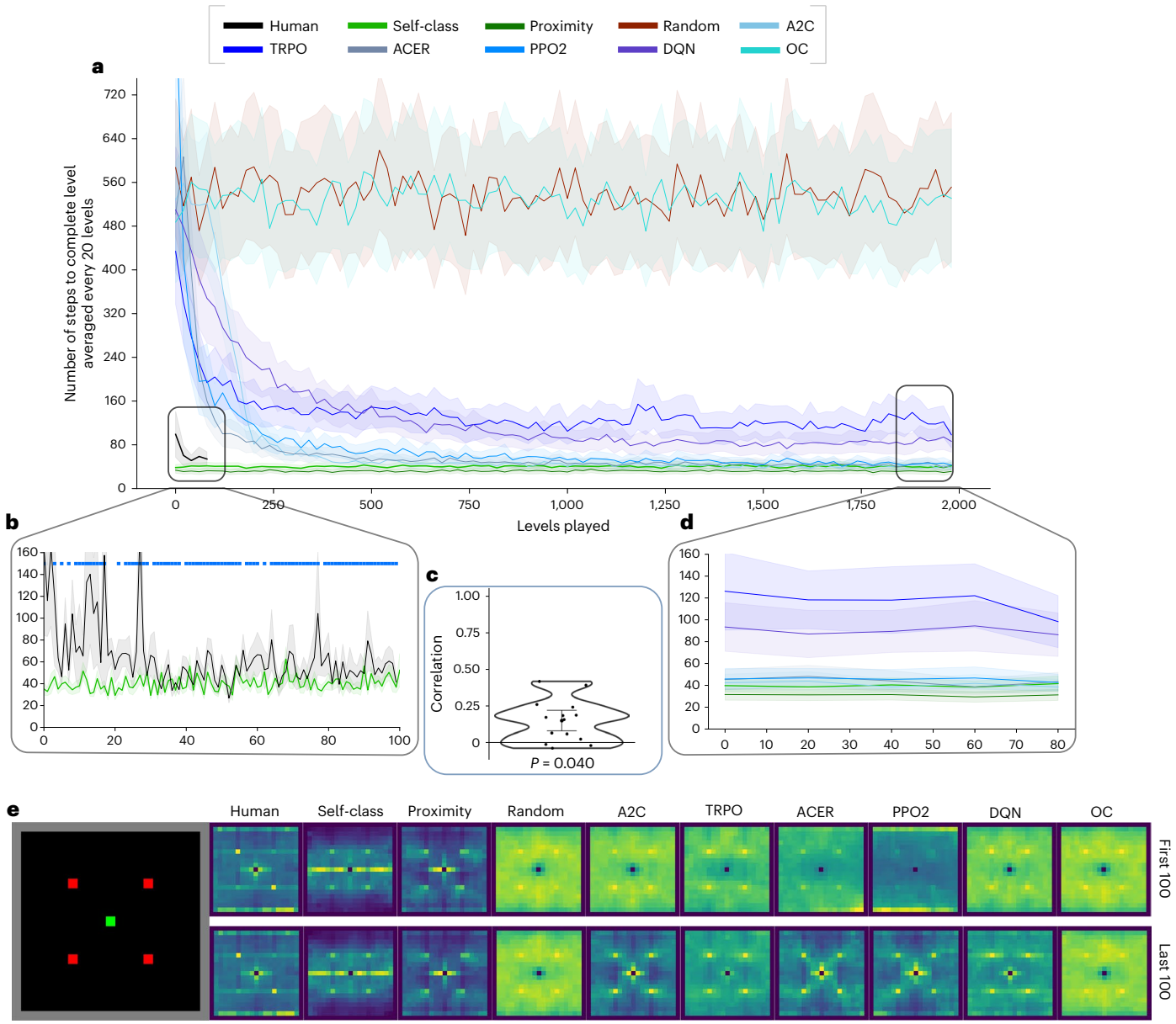

Fig. 7 | Results for study 4 (Switching Embodiments Game). a, Number of steps taken by all agents, averaged every 20 levels. Error shading reflects standard error of the mean. **b**, Zooming in on level-by-level human players and the self-class for the first hundred levels. Blue horizontal lines above the plot indicate levels where human performance is indistinguishable from optimal play (that is, levels where the line is visible have a Bayes factor (BF_{01}) above 1.0). **c**, Violin plots showing the correlations between total step count and the mean steps until self-orientation (study 4c). Error bars reflect 95% confidence intervals, centre of the error bar reflects the mean, and the *P* value beneath the plot shows the significance resulting from two-sided *t*-test comparing the correlation values with that of a random agent. **d**, Zooming in on artificial players for the last hundred levels, averaged every 20 levels. **e**, Heat maps of normalized action patterns for the first hundred levels (top row) and last hundred levels (bottom row), with human performance for first hundred levels included for comparison. Yellow shows the most visited, and purple shows the least visited, locations by the digital self.

Embodiments game did not impose the same memory requirements (players only needed to remember their embodiment, not how each key mapped to an action). Also, unlike in the Switching Mappings Game, if players did not efficiently navigate the Switching Embodiments Game, then the embodiments would switch indefinitely, preventing them from completing a given level and prolonging the experiment.

OC was unable to learn the game, even after we tried numerous hyperparameters (Supplementary Table 14), perhaps because the agent required more training to learn (given that it was slower to learn the Contingency Game compared with other agents), or because the options became too complex to solve for the agent. All other artificial agents were able to learn the game, with some (A2C, ACER and PPO2) even reaching optimal play after ~600 levels (Fig. 7a–d and Supplementary Table 4). A possible reason that the algorithms learned is that the key mappings remained useful; it is just that they were enacted through different agents. Presumably, the algorithms learned to manoeuvre whichever agent was correctly responding to key mappings closest to the reward. Of course, a corollary of this interpretation is that the artificial agents would not have solved the game in a human-like manner by orienting on a specific self. To test this, we added an additional simple algorithm (Proximity) that found the closest agent to the goal in each step and tried to navigate it to the goal. Surprisingly, this agent was slightly better than the self-class even though it did not implement self-orienting and thus had an obvious drawback, that is, it wasted steps

https://doi.org/10.1038/s41562-023-01696-5

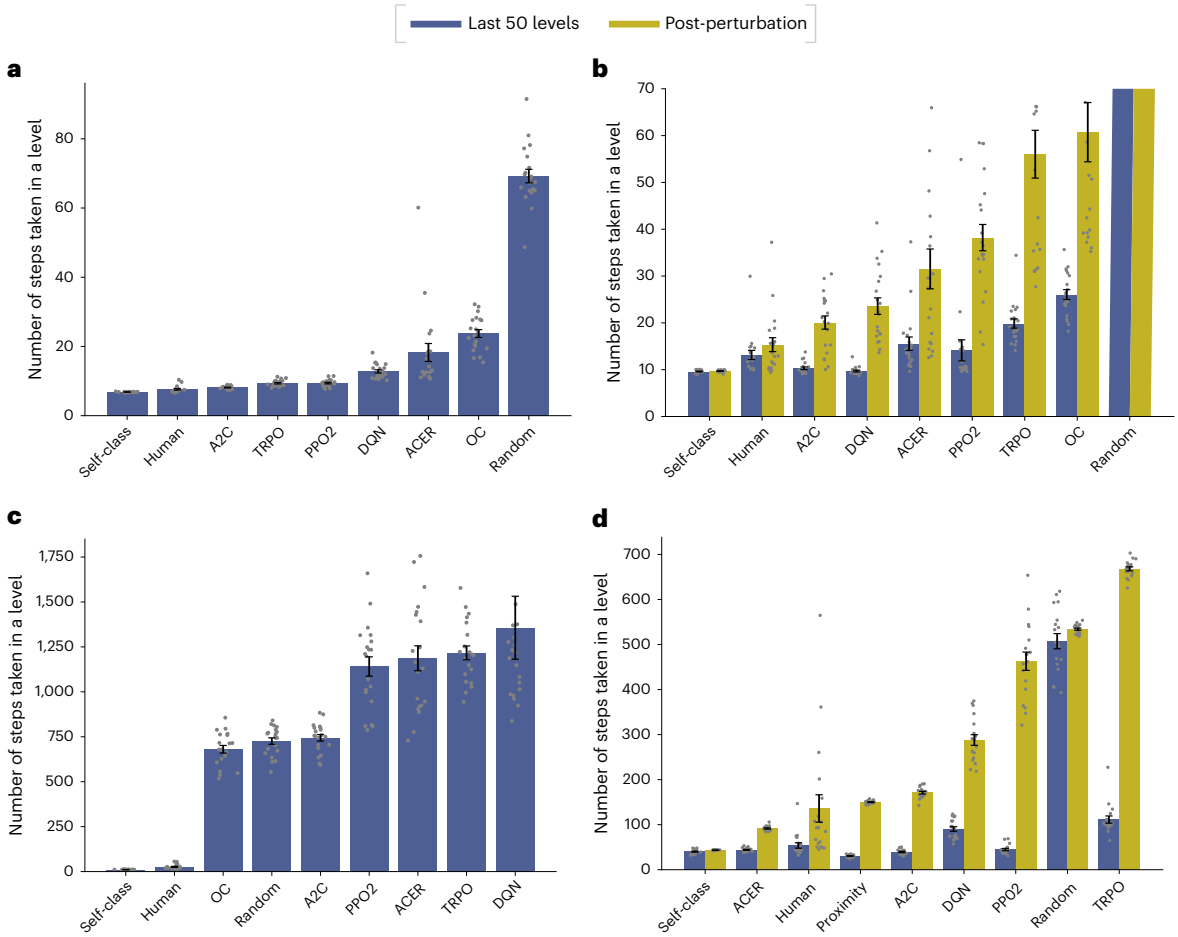

Fig. 8 | Results for the mean number of steps during the last 50 levels and (where relevant) all post-perturbation levels. a, Logic Game ($N = 20$). **b**, Contingency Game ($N = 20$). **c**, Switching Mappings Game ($N = 20$). **d**, Switching Embodiments Game ($N = 18$ for artificial players and $N = 19$ for human players). **a** and **c** are sorted by the mean number of steps taken in the last 50 levels, whereas **b** and **d** are sorted by the mean number of steps taken in the post-perturbation levels. Error bars indicate standard error of the mean, and centre of the error bar reflects the mean. Points on the bars represent different seeds/participants. We note that humans mostly recovered from perturbations in cases where the perturbation was not meant to have an effect on the self-orientation strategy, but that for the fourth game humans did perform worse, which is expected. There was no RL algorithm that matched the pattern of human performance post perturbation, suggesting they did not employ a human-like self-orientation strategy.

if the closest agent was not the real self. Even so, the Proximity algorithm outperformed the self-class in this game due to one key advantage: it effectively utilized the additional steps that would have otherwise been expensively spent on self-orienting (which needed to happen at least twice in the game, given the frequent embodiment switches). In general, self-orientation before planning is not always guaranteed to be optimal in this game, in terms of winning the game in the fewest number of steps.

Did humans and RL agents follow similar behavioural patterns? By plotting the average distance of the digital self from the reward on the first and last levels of play (Supplementary Fig. 7), we see that both human and artificial agents improved. The exploration heat maps show that, unlike the optimal self-class, humans spent more time in the lower and upper edges, maybe when they were struggling to adapt to the embodiment switch (Fig. 7d). Otherwise, the exploration patterns look similar. As in studies 1 and 2, we also tested how artificial and human players responded to a 'mock possible self' perturbation (studies 4b and 4c), again finding that human players used self-orientation whereas the algorithms did not. Post-perturbation performance for human players also exceeded all artificial players except for ACER, which employed a proximity-like heuristic strategy other than self-orienting that worked well in this game ('Supplementary results' in Supplementary Information).

In short, study 4 presented a constant challenge by switching the embodiment of the digital self within a level. This required players to be highly flexible, repeatedly self-orienting after losing their digital selves. Even so, artificial agents learned to play at optimal levels, but did so using a strategy that is different from self-orienting, suggesting that they did not actually learn to self-orient. In contrast, human players solved the game by self-orienting; when they performed worse during post-perturbation, this was because they self-oriented less consistently. Figure 8 summarizes the performance in all games for each agent before and after perturbation, by showing the average number of steps taken in a level for the final 50 levels during the training phase, as well as for all levels in the post-perturbation levels.

General discussion

This paper develops an account of how and why a computational notion of a self-orienting agent should be represented in machines that are meant to capture human-like intelligence. Such a representation is needed to solve basic problems continually confronted by any intelligent agent that learns, thinks and acts for itself. We tasked current game-playing algorithms with solving games in which the spatiotemporal location of the self's avatar was ambiguous, and found them to be structurally unsuited for learning the notion that the game

contains an avatar entity, and self-orienting towards the avatar. Even when RL algorithms did learn to play the games we studied, stress tests showed that these algorithms did not learn a robust and general self-orienting strategy.

When we minimally perturbed the games in ways that maintained the basic structure, the RL algorithms were generally off the mark, and there was no RL algorithm that consistently matched human behaviour post-perturbation: people were able to recover from the perturbations, except in cases where the perturbations are expected to make self-orientation more difficult. In general, we do not expect self-orientation to be robust to all perturbations or to guarantee optimal performance in all cases in the sense of getting to a goal as quickly as possible. Rather, we suggest self-orientation is an automatic process that people use in planning that usually allows them to operate with near-optimal efficiency across situations, but also that such a process may locally lose out to a simpler strategy in tailored situations, as in our fourth game.

Although we have studied an ability that happens very quickly in humans, we also see how consequential it is—when human players struggle to self-orient, this leads to large increases in time spent playing a game, and when AI are incapable of self-orienting, they need many hundreds of more levels of gameplay before playing optimally. Here we identified the structural limitation in some well-known AI algorithms that leads to this difference, bringing some concreteness to an ancient topic of 'self-representation' that previously escaped computational rigour. While the very latest AI can even play multiple games[36,37], our results predict that they will learn these games in a non-human-like way. Our contributions also include a test bed that can be expanded to examine whether algorithms have a computational notion of self-orienting. Specifically, we created a series of litmus tests of whether agents have a computational self-representation, and metrics of optimality and AI comparisons by which to assess the extent to which humans are effective at self-orienting. Potential objections to our findings are addressed in 'Supplementary discussion' in Supplementary Information.

Methods

All data were collected under approval by the Harvard University-Area Committee on the Use of Human Subjects (IRB-16-0158). Informed consent was obtained from all participants.

We aimed to recruit 20 participants to play each game. All games were created by modifying an existing gridworld gaming environment[38] compatible with the OpenAI Gym Toolkit[39]. Readers can play the games at https://eilexp.xyz/.

Before each game, participants were asked to answer a consent form and two attention check questions. Those who passed the checks entered our Qualtrics survey, where they were provided a link to start playing the game and simply instructed to 'use the arrow keys to play the game'. After the game ended, we asked participants to fill in their random ID and complete the remaining questions in the Qualtrics survey, including comprehension checks and demographics. We used Cloud Research to publish our studies on Amazon's Mechanical Turk.

All artificial agents were run twenty independent times except in study 4 where they were run 18 times to match the number of human participants included in that study, using randomly initialized seeds per run.

In comparing human performance to optimal play we use both *t*-tests and Bayes factors[40], which can be used to quantify evidence for the null hypothesis. For example, BF_{01} of 5.0 means that the data would be five times more likely under the null hypothesis than under the alternative hypothesis. Our Bayes factor analysis assumes a default medium prior of sqrt(2)/2, and is conducted using the BayesFactor library in R. We looked for evidence in favour of the null hypothesis of no difference in performance between human players and the optimal self-class (that is, BF_{01}), which would be evinced by a BF_{01} >1. The results of these tests are depicted with horizontal lines in Fig. 2b. Because we conduct 100 Bayes tests, one for each level, full test results for each game are reported in Supplementary Information (Supplementary Tables 15–20).

Study 1: Logic Game

Participants. Our sample size for all experiments was determined by a pilot study that found clean results with just 10 participants. Since we wanted to run several seeds of each artificial algorithm, we doubled this number for the sake of fair comparison. We recruited 20 participants (40% female, M_{age} = 42), paying them US$1 each. All participants passed attention checks, and at least one of the two comprehension checks. Participants completed the game in 8.9 min on average.

Game and agents. In the Logic Game, an optimal player should leverage logic to take the most efficient action to disambiguate the digital self from all possible selves, even when an action leads to no visible movement from any avatar. By design, there was only one correct move, else the digital self did not physically displace. The Logic Game consisted of a 9 × 9 grid space. Each possible self was neighboured by three walls. On each level we varied two factors: the positions of the walls neighbouring each possible self, and the starting location of the digital self (Supplementary Fig. 16). The game is visualized in Fig. 1, and hyperparameters for each of the game-playing RL agents are provided in Supplementary Information (Supplementary Tables 9–14). To optimally self-orient and navigate the digital self to the reward, the hard-coded self-class employed the logic in Supplementary Fig. 17.

Study 2: Contingency Game

Participants. We recruited 20 participants (42% female, M_{age} = 44), paying them US$1 each. They completed the game in 11.6 min on average. All passed attention checks and at least one of the two comprehension checks.

Game and agents. The Contingency Game consisted of a 21 × 21 grid space. Each possible self was located in the middle of each quarter of the grid space. On each level, we varied the following: the oscillation direction of each possible self and the starting location of the digital self. Whenever a player pressed a key, all agents moved. For every move, all possible selves oscillated in one of two directions sampled at random: up–down and left–right. The only constraint was that the possible selves remained within a designated 9 × 9 space centred at their starting locations. To optimally orient on the digital self and navigate it to the reward, the self-class employed the logic in Supplementary Fig. 18.

Analyses. Since it is less clearcut to measure the point of self-orientation for a random agent in all games except the logic game, in this study onwards we measured the correlation of the random agent by identifying the point at which the self-class would find the self, using the steps taken by the random agent. In other words, for each level in the game, we iterated through each action and compared the locations of each possible self after each action, eliminating the selves that moved in a direction different from the provided action. The step at which there was only a single remaining possible self was marked as the step at which the self was found. We then correlated this value with the total number of steps the agent required to complete the level, to compare with this same correlation for human players.

Study 2b: Contingency Game with self-finding task

Participants. We recruited 20 participants (40% female, M_{age} = 43), paying them US$2 each. They completed the game in 20.7 min on average. All passed attention checks and at least one of the two comprehension checks.

Game and agents. The environment was the same as in the Contingency Game, but in this version of the game, participants were told: 'In each level, as soon as you find which square you are controlling, please click on that square. You can only click one time per level'. Those who neglected to do so, or who clicked a square more than once, were shown error messages. When participants clicked on an agent, the agent that was clicked turned blue for half a second, to give feedback that they selected that agent.

Study 2c: Contingency Game with perturbation task

Participants. We recruited 20 participants (35% female, M_{age} = 38), paying them US$2 each. They completed the game in 18.1 min on average. All passed attention checks and at least one of the two comprehension checks.

Game and agents. Participants played the first hundred levels of the game followed by 50 levels of the same perturbation shown to the algorithms in study 2.

Study 3: Switching Mappings Game

Participants. We recruited 20 participants (35% female, M_{age} = 37), paying them US$3 each. They completed the game in 20.1 minutes on average (almost twice as long as in the Contingency Game). All passed attention checks and at least one of the two comprehension checks.

Game and agents. The game environment was the same as the Contingency Game, except that key–action mappings were shuffled at the start of each level. To optimally self-orient and navigate to the reward, the self-class employed the logic in Supplementary Fig. 19.

Study 3b: Switching Mappings Game with self-finding task

Participants. We recruited 20 participants (60% female, M_{age} = 43), paying them US$2 each. They completed the game in 46.0 min on average. All passed attention checks and at least one of the two comprehension checks.

Game and agents. The environment was the same as in the Switching Mappings Game, but in this version of the game, participants were told: 'In each level, as soon as you find which square you are controlling, please click on that square. You can only click one time per level'. Those who neglected to do so, or who clicked a square more than once, were shown error messages. When participants clicked on an agent, the agent that was clicked turned blue for half a second, to give feedback that they selected that agent.

Study 4: Switching Embodiments Game

Participants. We recruited 18 participants (56% female, M_{age} = 38), paying them US$4 each. They completed the game in 33.3 min on average. The study took approximately 20 min longer to complete than the Contingency Game (study 2) and 12 min longer to complete than the Switching Mappings Game (study 3), probably because the embodiment switching disrupted play. All passed attention checks and at least one of the two comprehension checks.

Game and agents. The environment was the same as the Contingency Game. The digital self switched embodiments every seven moves—exactly one move before when an optimal player in studies 2 and 3 would have finished the game. To optimally self-orient and navigate to the reward, the self-class employed the logic in Supplementary Fig. 20.

Study 4b: Switching Embodiments Game with perturbation task

Participants. We recruited 19 participants (37% female, M_{age} = 44), paying them US$6 each. They completed the game in 70.4 min on average. All passed attention checks and at least one of the two comprehension checks.

Game and agents. Participants played the first hundred levels of the game followed by 50 levels of the same challenging perturbation shown to the algorithms in study 4.

Study 4c: Switching Embodiments Game with self-finding task

Participants. We recruited 19 participants (63% female, M_{age} = 38), paying them US$8 each. They completed the game in 59.7 min on average. All passed attention checks and at least one of the two comprehension checks.

Game and agents. Participants played the first 34 levels of the original game, which was the point at which their performance plateaued, as observed in study 4a. After that, they played 20 levels of the challenging perturbation task, which was the same as the one presented to the algorithms in study 4. The performance plateau was identified using the elbow method applied to a third-degree polynomial that was fitted to the human data. This polynomial had the lowest root mean square error when tested on a 30/70 test–train split, which was replicated 100 times to ensure optimal fitting.

In this version of the game, participants were told: 'In each level, as soon as you find which square you are controlling, please click on that square. Please do this every time when you find which square you are controlling'. Those who neglected to do so in a given level, or who clicked a square more than once within a second, were shown error messages. When participants clicked on an agent, the agent that was clicked turned blue for half a second, to provide feedback that the agent was selected.

We calculated the mean steps until self-orientation following an embodiment switch by dividing the step at which a participant clicked on an agent by 7, and taking the remainder (since embodiment switched at each of the seven steps). For example, if a participant identified themselves at step 10, we assume they took three steps to do so, as the last embodiment switch occurred at step 7. In cases where the player clicked on agents multiple times within the period of seven steps after the latest embodiment switch, we only considered the last selection in our analyses. If the participant selected the incorrect self and did not correct this before the next embodiment switch, we heuristically treated the self-orienting steps as 7, since this is the worst measurable self-orienting performance.

Artificial agents

All RL algorithms except OC were drawn from a public repository called 'stable baselines'[41], a set of improved implementations of RL algorithms based on the original OpenAI Baselines repository[42]. The OC algorithm was drawn from a separate GitHub repository[43]. Hyperparameters of the algorithms were tuned to maximize performance for each game.

To map the environment pixel grid (for example, the 21 × 21 matrix in the Contingency Game) to the model's input observation array (which is 128 × 128 × 3) for model training, we first calculate the size of each cell by dividing the width and height dimensions of the observation array (128 × 128) by that of the environment (21 × 21), producing approximately 6 × 6 pixel cells for the Contingency Game. We then iterate over each of these cells to assign the corresponding colour to each cell, for example, we assign the colour red if the current cell is a possible self, and grey if the current cell is a wall. At the end, we obtain a 128 × 128 × 3 matrix that represents the environment as an image, where the third dimension stores the colour values for each cell, as the colours are represented by three values (red, green and blue).

Reporting summary

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

Data availability

The data that support the findings of this study are available in the Open Science Framework at https://osf.io/bwzth/.

Code availability

All code for data analysis and reproducing the plots is available at https://github.com/Ethical-Intelligence-Lab/probabilisticSelf.

References

1. James, W., Burkhardt, F., Bowers, F. & Skrupskelis, I. K. *The Principles of Psychology* Vol. 1 (Macmillan London, 1890).
2. Belk, R. W. Extended self in a digital world. *J. Consum. Res.* **40**, 477–500 (2013).
3. Buckner, R. L. & Carroll, D. C. Self-projection and the brain. *Trends Cogn. Sci.* **11**, 49–57 (2007).
4. Dennett, D. C. in *Self and Consciousness* 111–123 (Psychology Press, 2014).
5. Sui, J. & Humphreys, G. W. The integrative self: how self-reference integrates perception and memory. *Trends Cogn. Sci.* **19**, 719–728 (2015).
6. Blanke, O. & Metzinger, T. Full-body illusions and minimal phenomenal selfhood. *Trends Cogn. Sci.* **13**, 7–13 (2009).
7. Bem, D. J. Self-perception: an alternative interpretation of cognitive dissonance phenomena. *Psychol. Rev.* **74**, 183 (1967).
8. McConnell, A. R. The multiple self-aspects framework: self-concept representation and its implications. *Personal. Soc. Psychol. Rev.* **15**, 3–27 (2011).
9. Sanchez-Vives, M. V. & Slater, M. From presence to consciousness through virtual reality. *Nat. Rev. Neurosci.* **6**, 332–339 (2005).
10. Strawson, G. The sense of the self. *Lond. Rev. Books* **18**, 126–152 (1996).
11. Dennett, D. C. in *Science Fiction and Philosophy: From Time Travel to Superintelligence* (ed. Schneider, S.) 55–68 (John Wiley & Sons, 2016).
12. Nozick, R. *Philosophical Explanations* (Harvard Univ. Press, 1981).
13. Perry, J. Can the self divide? *J. Philos.* **69**, 463–488 (1972).
14. Moulin-Frier, C. et al. DAC-h3: a proactive robot cognitive architecture to acquire and express knowledge about the world and the self. *IEEE Trans. Cogn. Dev. Syst.* **10**, 1005–1022 (2017).
15. Johnson, M. & Demiris, Y. Perceptual perspective taking and action recognition. *Int. J. Adv. Rob. Syst.* **2**, 32 (2005).
16. Paul, L., Ullman, T. E., De Freitas, J. & Tenenbaum, J. Reverse-engineering the self. Preprint at https://doi.org/10.31234/osf.io/vzwrn (2023).
17. Andrychowicz, M. et al. Hindsight experience replay. *Adv. Neural Inform. Process. Syst.* **30**, 5048–5058 (2017).
18. Hausknecht, M. & Stone, P. in *2015 AAAI Fall Symposium Series* 29–37 (AAAI, 2015).
19. Schaul, T., Quan, J., Antonoglou, I. & Silver, D. Prioritized experience replay. Preprint at https://doi.org/10.48550/arXiv.1511.05952 (2015).
20. Van Hasselt, H., Guez, A. & Silver, D. in *Proc. AAAI Conference on Artificial Intelligence* 2094–2100 (AAAI, 2016).
21. Wang, Z. et al. in *International Conference on Machine Learning 1995–2003* (PMLR, 2016).
22. Mnih, V. et al. Playing Atari with deep reinforcement learning. Preprint at https://doi.org/10.48550/arXiv.1312.5602 (2013).
23. Kaiser, L. et al. Model-based reinforcement learning for Atari. Preprint at https://doi.org/10.48550/arXiv.1903.00374 (2019).
24. Dubey, R., Agrawal, P., Pathak, D., Griffiths, T. L. & Efros, A. A. Investigating human priors for playing video games. Preprint at https://doi.org/10.48550/arXiv.1802.10217 (2018).
25. Tsividis, P. A. et al. Human-level reinforcement learning through theory-based modeling, exploration, and planning. Preprint at https://doi.org/10.48550/arXiv.2107.12544 (2021).
26. Tsividis, P. A., Pouncy, T., Xu, J. L., Tenenbaum, J. B. & Gershman, S. J. in *2017 AAAI Spring Symposium Series* 643–646 (AAAI, 2017).
27. Uhde, C., Berberich, N., Ramirez-Amaro, K. & Cheng, G. in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* 8081–8086 (IEEE, 2020).
28. Lanillos, P. & Cheng, G. Robot self/other distinction: active inference meets neural networks learning in a mirror. Preprint at https://doi.org/10.48550/arXiv.2004.05473 (2020).
29. Demiris, Y. & Meltzoff, A. The robot in the crib: a developmental analysis of imitation skills in infants and robots. *Infant Child Dev. Int. J. Res. Pract.* **17**, 43–53 (2008).
30. Piaget, J. The construction of reality in the child. *J. Consult. Psychol.* **19**, 77 (1955).
31. Thrun, S. in *Robotics and Cognitive Approaches to Spatial Mapping* 13–41 (Springer, 2008).
32. Silver, D., Singh, S., Precup, D. & Sutton, R. S. Reward is enough. *Artif. Intell.* **299**, 103535 (2021).
33. Botvinick, M. et al. Building machines that learn and think for themselves. *Behav. Brain Sci.* **40**, E255 (2017).
34. Botvinick, M. et al. Building machines that learn and think for themselves: Commentary on Lake et al., Behavioral and Brain Sciences, 2017. Preprint at https://doi.org/10.48550/arXiv.1711.08378 (2017).
35. Vul, E., Goodman, N., Griffiths, T. L. & Tenenbaum, J. B. One and done? Optimal decisions from very few samples. *Cogn. Sci.* **38**, 599–637 (2014).
36. Reed, S. et al. A generalist agent. *Trans. Mach. Learn. Res.* https://openreview.net/forum?id=1ikK0kHjvj (2022).
37. Schrittwieser, J. et al. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature* **588**, 604–609 (2020).
38. Pan, X. et al. How you act tells a lot: privacy-leakage attack on deep reinforcement learning. Preprint at https://doi.org/10.48550/arXiv.1904.11082 (2019).
39. Brockman, G. et al. OpenAI Gym. Preprint at https://doi.org/10.48550/arXiv.1606.01540 (2016).
40. Rouder, J. N., Speckman, P. L., Sun, D., Morey, R. D. & Iverson, G. Bayesian *t* tests for accepting and rejecting the null hypothesis. *Psychon. Bull. Rev.* **16**, 225–237 (2009).
41. Hill, A., Raffin, A., Ernestus, M., Gleave, A. & Kanervisto, A. stable-baselines. *GitHub* https://github.com/Stable-Baselines-Team/stable-baselines (2018).
42. Dhariwal, P. et al. Openai baselines. *GitHub* https://github.com/openai/baselines (2017).
43. Weitkamp, L. option-critic-pytorch. *GitHub* https://github.com/lweitkamp/option-critic-pytorch (2019).

Acknowledgements

For running the artificial models, we used the Harvard Business School compute cluster. This research was funded by Harvard Business School. The funders had no role in study design, data collection and analysis, decision to publish or preparation of the manuscript.

Author contributions

J.D.F. initiated the research, J.D.F., A.K.U. and Z.O.-U. put together the data and conducted the analyses, and J.D.F., A.K.U., Z.O.-U., L.A.P., J.T. and T.D.U. wrote the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41562-023-01696-5.

Correspondence and requests for materials should be addressed to Julian De Freitas.

Peer review information *Nature Human Behaviour* thanks Nathan Faivre, Tony Prescott and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

nature portfolio

Corresponding author(s): Julian De Freitas

Last updated by author(s): Aug 1, 2023

Reporting Summary

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

Statistics

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

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

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

Software and code

Policy information about availability of computer code

Data collection	All the required code for the data collection of artificial agents are in the following GitHub repository: https://github.com/Ethical-Intelligence-Lab/probabilisticSelf. Code used for data collection of human subjects are available in the following GitHub repository: https://github.com/Ethical-Intelligence-Lab/prob_self_app_js. Both repositories are publicly accessible. For running artificial models we used Python 3.9 (for OC algorithm) and Python 3.6 (for other RL models). We used Python 3.8 for the human data collection app.
Data analysis	All the required code for data analysis are in the following GitHub repository: https://github.com/Ethical-Intelligence-Lab/probabilisticSelf. All analyses of human and artificial agent data were perfomed using R (version 4.3.1) and python 3.10.12. For the Bayes Factor analysis, we used BayesFactor package (v. 0.9.12-4.4), for calculating Cohen's d values we used effsize package (v. 0.8.1) in R language. We used python for generating the plots and for heatmap analyses, and used matplotlib (v. 3.7.1), numpy (v. 1.23.5), pandas (v. 2.0.1), matplotlib (v. 3.7.1), scikit-learn (v. 1.2.2), scikit-image (v. 0.20.0), rpy2 (v. 3.5.11), pillow (v. 9.5.0), kneed (v. 0.8.5).

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

Data

Policy information about availability of data

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

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

The datasets generated and analyzed during the current study are available in the OSF repository, https://osf.io/bwzth/.

Human research participants

Policy information about studies involving human research participants and Sex and Gender in Research.

Reporting on sex and gender	There was no sex- and gender- based analysis, since we were interested in overall human self-orientation abilities.
Population characteristics	See above.
Recruitment	We used Cloud Research to publish our studies on Amazon's Mechanical Turk. We limited recruitment to participants with > 95% approval rates who had previously participated in at least 100 studies. We believe that there is no self-selection bias that would differ these participants from other participants in the Mechanical Turk platform.
Ethics oversight	All data were collected under approval by the Harvard University-Area Committee on the Use of Human Subjects.

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

Field-specific reporting

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

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

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

Behavioural & social sciences study design

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

Study description	Studies involved quantitative data.
Research sample	We recruited a total 124 participants from Amazon's Mechanical Turk including non-participations. Study 1: 20 participants (40% female, Mean age = 42), Study 2: 20 participants (42% female, Mean age = 44), Study 2b: 20 participants (40% female, Mean age = 43), Study 2c: 20 participants (35% female, Mean age = 38), Study 3: 20 participants (35% female, Mean age = 37), Study 3b: 20 participants (60% female, Mean age = 43), Study 4: 18 participants (56% female, Mean age = 38), Study 4b: 19 participants (37% female, Mean age = 44), Study 4c: 19 participants (63% female, Mean age = 38). Our sample size for all experiments was determined by a pilot study that found clean results with just 10 participants. We consider our sample of participants to be representative of users within Amazon's Mechanical Turk platform, but not necessarily representative of all humans.
Sampling strategy	We randomly sampled from the available Mturk participation pool. Sample size was determined based on a pilot study, which found that 10 participants provided sufficient statistical power to observe our effects. In each of the reported studies 1-4, we conservatively aimed to double that number of participants. This decision is in alignment with the findings of Julious (2005), where he suggested a minimum of 12 participants when there is no prior information to determine the sample size. Julious, S. A. (2005). Sample size of 12 per group rule of thumb for a pilot study. Pharmaceutical Statistics: The Journal of Applied Statistics in the Pharmaceutical Industry, 4(4), 287-291.
Data collection	We used Cloud Research to publish our studies online on Amazon's Mechanical Turk. No one else was present besides the participants as the experiment was done online.
Timing	Study 1: 8-10 December 2021 Study 2: 10-14 December 2021 Study 2b: 18 January 2023 Study 2c: 20 January 2023 Study 3: 15 December 2021

	Study 3b: 19 December 2023 Study 4: 17-27 December 2021 Study 4b: 29 March 2023 Study 4c: 28 April 2023
Data exclusions	No exclusions: All passed attention and at least one of two comprehension checks.
Non-participation	All participants accepted the consent form. 28 participants did not finish the survey, and out of the remaining 96 participants, 18 did not finish the game.
Randomization	Participants were randomly allocated.

Reporting for specific materials, systems and methods

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

Materials & experimental systems

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

Methods

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