Working Paper 23-017

What Would It Mean for a Machine to Have a Self?

Julian De Freitas
Ahmet Kaan Uğuralp
Zeliha Uğuralp
Laurie Paul
Joshua B. Tenenbaum
Tomer Ullman

What Would It Mean for a Machine to Have a Self?

Julian De Freitas
Harvard Business School

Ahmet Kaan Uğuralp
Bilkent University

Zeliha Uğuralp
Bilkent University

Laurie Paul
Yale University

Joshua B. Tenenbaum
MIT

Tomer Ullman
Harvard University

Working Paper 23-017

Funding for this research was provided in part by Harvard Business School.

What Would It Mean For a Machine to Have a Self?

Julian De Freitas[1], Ahmet Kaan Uğuralp[2], Zeliha Uğuralp[3], Laurie Paul[4], Joshua Tenenbaum[5], Tomer D. Ullman[6]

1 – Marketing Unit, Harvard Business School

2 – Computer Science Department, Bilkent University

3 – Psychology Department, Bilkent University

4 – Philosophy Department, Yale University

5 – Brain & Cognitive Sciences Department, MIT

6 – Psychology Department, Harvard University

ABSTRACT

What would it mean for autonomous AI agents to have a ‘self’? One proposal for a minimal notion of self is a representation of one’s body spatio-temporally located in the world, with a tag of that representation as the agent taking actions in the world. This turns self-representation into a constructive inference process of self-orienting, and raises a challenging computational problem that any agent must solve continually. Here we construct a series of novel ‘self-finding’ tasks modeled on simple video games—in which players must identify themselves when there are multiple self-candidates—and show through quantitative behavioral testing that humans are near 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 navigate new settings, and that this is a crucial target for engineers wishing to develop flexible agents.

Keywords: Self; AI; games; reinforcement learning; avatar

What would it mean for artificial intelligence (AI) agents to have a way of representing themselves, to have a 'self'? If we refer to an AI agent as acting autonomously—that is, being self-governed or self-directed—we ought to know what it would even mean for such a system to have a self. There has been much work on 'the self' in philosophy, psychology, and neuroscience (1-13), but none that we know of on how the self can be concretely represented in artificial intelligence algorithms. In a recent theoretical paper (14), we introduced the notion of a 'minimal self' and its role in solving a basic problem that must be solved continually by any intelligent agent—human or artificial—that learns, think and acts for itself. In that paper, we argue for a minimal notion of 'self', a representation that points to a spatio-temporal 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 minimal self-representation in this sense, and explore the case for how such a self can be concretely represented in artificial intelligence algorithms. Paul et al. (14) 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 which humans can solve in a general-purpose way is linking to the correct self-representing entity across situations and environments.

Building on this theoretical work, here we test for a minimal 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. By building on past work on reinforcement learning (RL) (15-19), game playing (20, 21), and cognitive science work on RL and game playing (22-24), we also propose that existing AI probably do not have a minimal self-representation in our sense, and that while some algorithms may be trained to carry out self-orienting in particular environments, they are doing so through particular brittle cues rather than a general-purpose process. The computational challenge which humans can solve in a general-purpose way is linking to the correct self-representing entity across situations and environments.

To appreciate the importance of self-orienting, it is helpful to imagine what it is like to *not* have a correct minimal 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, a point-of-view, and in general center the game around it.

In (14), we argue that finding out which avatar you are in a new game is a particularly useful way to explore our sense of a minimal 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'. This process is so fast it is often not thought of as an official part of the game's goals. There are multiple ways of resolving the process, from futzing with a controller, to simply being told "you are the giant cat". The cues can be linguistic, visual, auditory, haptic, and yet they are all pointing to the same underlying process—identifying the correct entity as one's avatar. The process is crucial, in that most often the game cannot proceed for people until they figure out their avatar. Yes, there is a bag of diamonds to find or a troll to defeat, but all that has to wait until we know who we are, which amounts to setting up and linking to the right avatar representation. Again, games are a useful metaphor here, but the process is meant to apply to everyday situations as well: we cannot reasonably plan in the library until we orient ourselves in the library.

Note that in games and life there are many possible cues that can help us in self-orientation. 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. In a game, a blinking cursor may alert us to our avatar. 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 spatio-temporal whereabouts as the avatar. We do so rapidly and effortlessly, except for salient and infrequent cases in which we are lost, as in a big mall or when playing a five-player game with many agents on the screen. But like many other automatic everyday cognitive processes such as perception or motor-action, the ease with which we perform self-orienting belies its complexity and importance: the ability to accurately self-represent ourselves is what allows us to flexibly navigate the varied settings of life. 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. The ability to efficiently self-orient may be a fundamental aspect of what makes humans flexible learners, and will be needed to create more human-like AI.

In the current work, we extend the proposal of (14) to investigate the extent to which humans and well-known AI algorithms from reinforcement learning (RL) are capable of self-orienting, by creating 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 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 best of contemporary AI and cannot be representative of the latest frontiers in Deep RL. However, they are well-known and well-studied baselines for building agents that operate autonomously in some environment, and they embody the thesis of reinforcement learning that some prominent AI researchers (25-27) have suggested is a scaling route to building fully general AI with human-level intelligence or beyond.

Human Players

All data were collected under approval by the Harvard University-Area Committee on the Use of Human Subjects. We aimed to recruit 20 participants to play each game. Before the 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". No further instruction or feedback of any kind was provided during 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.

The Self Finding Games

All games featured four agents (aka ‘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 (gray). 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. Readers can play the games here: https://prob-self-app-us.herokuapp.com/.

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: (i) *self-orienting*, in which the player figures out which of several possible selves is their real digital self (their avatar), and (ii) *navigation,* in which the player moves the digital self to a rewarding goal.

Each game consisted of 100 levels. The levels of each game obeyed the overall rules of the game, while varying the starting position of the different entities (agents, avatar, walls, goal). All games were created by modifying an existing gridworld gaming environment (28) compatible with the OpenAI Gym Toolkit (29).

To measure general self-orienting, the different games were designed such that agents had to exploit different kinds of cues to successfully self-orient. The Logic Game (Study 1) required players to logically take the most efficient action to eliminate possible candidates for their digital self, since there was only ever one correct action. The Contingency Game (Study 2) required players to find their digital self by exploiting informative contingencies between their keypresses and visible changes in possible candidates for the digital self (henceforth ‘possible selves’). The Switching Mappings Game (Study 3) was a variant of the Contingency Game in which the keypress-action mappings were randomly switched on every level of the game, and the

Switching Embodiments Game (Study 4) was another variant in which the digital self periodically switched embodiments with another possible self every few actions within a level.

Optimal Self Class

In order to assess the extent to which people optimally self-orient in these games, we compared human performance to that of a 'Self Class' which 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.

AI Players

To assess the abilities of well-known game-playing RL algorithms, each of the four games was also played for 2000 levels by the following RL algorithms: DQN, A2C, TRPO, ACER, PPO2. These algorithms (aka pixel-based RL baselines) use a combination of convolutional and fully connected neural network layers to learn from frame-by-frame pixel images of the game. They received a reward of 1 for completing a level of the game.

The RL algorithms were drawn from a public repository called 'stable baselines' (30), a set of improved implementations of RL algorithms based on the original OpenAI Baselines repository (31). 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. As a final control for both human and AI players, we also ran the games through a random policy that took random actions.

Study 1: The Logic Game

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. Even so, if no agent moved this was informative in eliminating possible selves as candidates for the digital self, and so an optimal player should learn from these 'non events' too (see Figure 1 for details). 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.

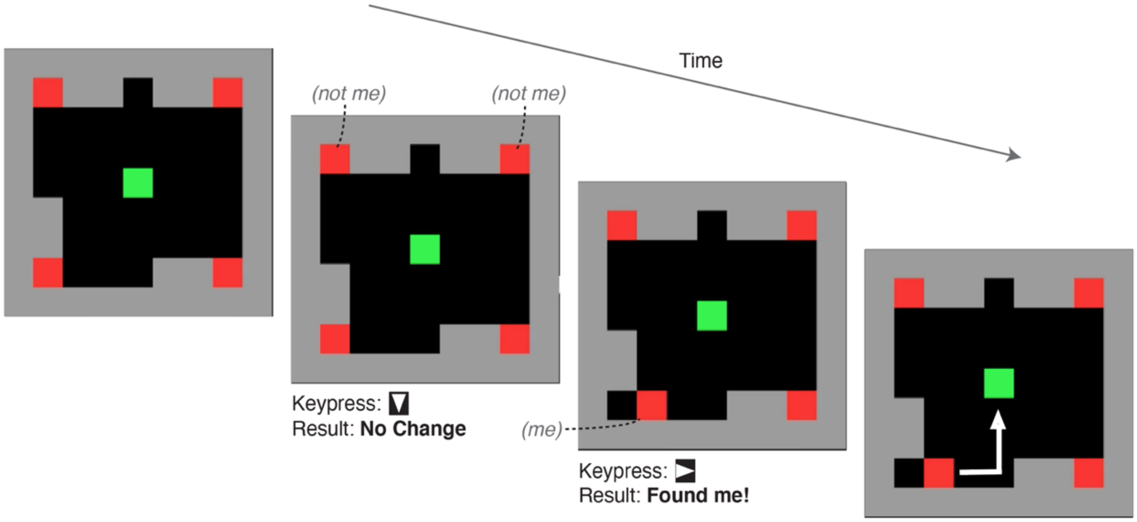

Figure 1. The Logic Game. There are 4 agents (red blocks), one of which is your avatar. The level ends when the avatar reaches the goal (green block). In this example, moving DOWN disambiguates the most possible selves (red)—the top two. If moving down produces no visible change, then you must be one of the bottom two agents. In order 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).

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 to 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. To complement these analyses, we also calculated two-sample Bayes Factor *t* tests between human players and the self-class[1]. We looked for evidence in favor of the null hypothesis of no difference in performance between human players and the optimal self-class (i.e., BF_{01}), which would be evinced by a $BF_{01} > 1$. The results of these tests are depicted with horizontal lines in

[1] Bayes factors (32) can be used to quantify evidence for the null hypothesis. For example, $BF_{01} = 5.0$ 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.

Figure 2B. We see that humans begin to perform indistinguishably from the self-class after just a single level.

Contrasting with human players, the AI agents played for several hundred levels before their performance plateaued. Notably, even after 2000 levels, AI players did not reach human-level performance (Table S1 and Figure 2C). In short, although the artificial agents improved in their performance, humans learned far more quickly, and played more optimally.

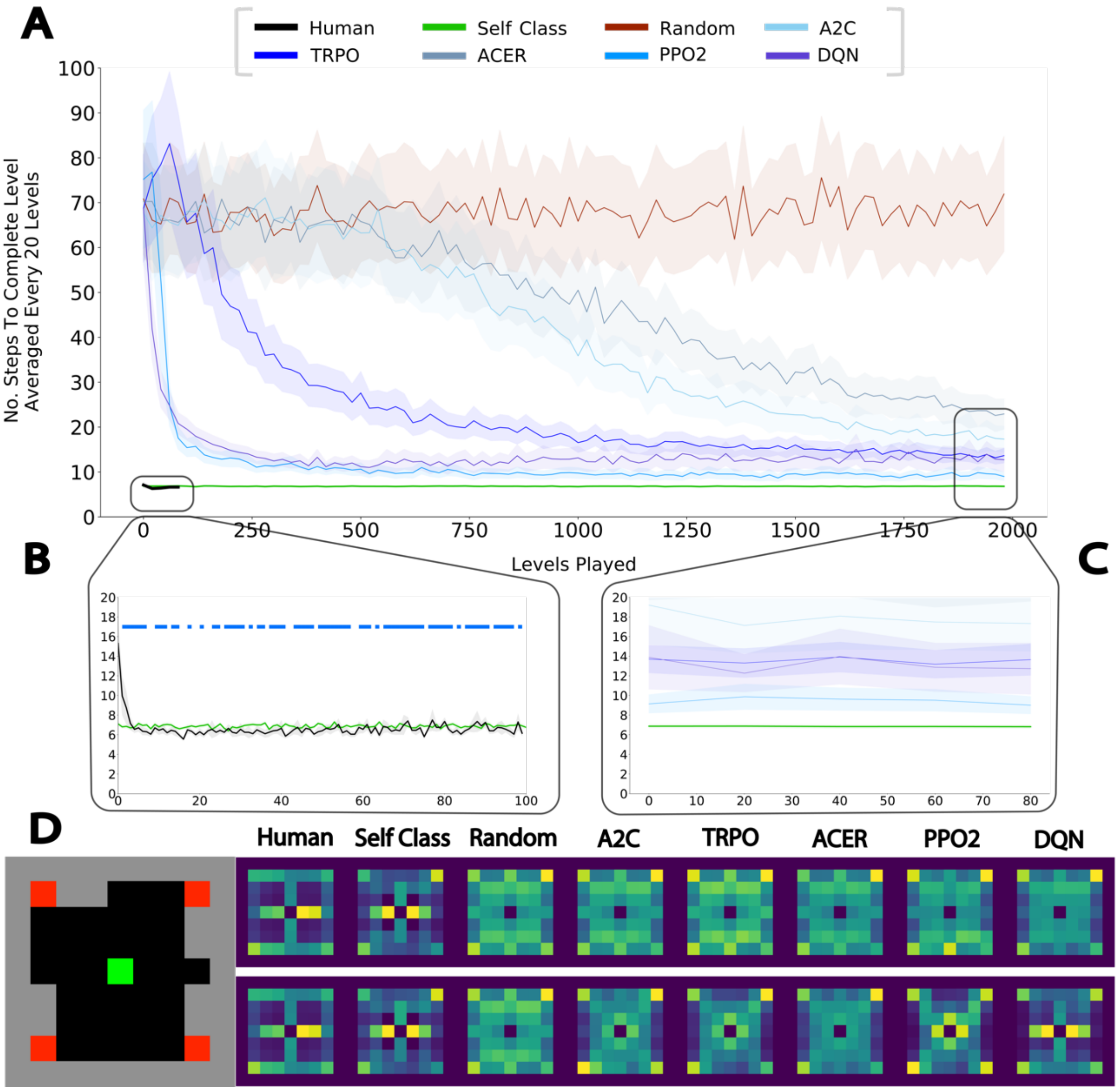

Figure 2. Results of Study 1 (Logic Game). (A) Number of steps taken by all agents, averaged every 20 levels. Error shading reflects standard error of the mean. (B) Zooming in on level-by-level human players and the self-class for the first hundred levels. Horizontal lines above the plot

indicate levels where the human performance is indistinguishable from optimal (i.e., Bayes Factor above 1.0). (C) Zooming in on artificial players for the last hundred levels, averaged every 20 levels. (D) Heatmaps of 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.

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 key press 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 (Figure S1). We find that none of the algorithms show a noticeable improvement in how quickly they self-orient, in other words, although algorithms learned how to navigate to the reward (Figure 1A), they do not learn how to optimally self-orient (Figure S1), whereas human players rapidly reach optimal levels of self-orienting (Figure S1).

A final way to compare human and artificial players is to examine their behavioral patterns within the gaming environment over time. The heatmaps in Figure 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. To quantitatively compare the heatmaps between human and AI players, we measure the mean squared error (MSE) between the heatmaps. We see that the mean MSE score is 6255 for the first hundred levels, and 4689 for the last hundred levels, showing that the behavioral patterns of the artificial agents become more human-like after training. We also performed t-tests of whether humans and each of the RL algorithms differ in how close their MSE scores are relative to the Self Class (Table S5-8). We found that all artificial agents except DQN had a significantly larger MSE from the self-class during their last hundred levels than did humans during their first hundred levels (ps < .001), indicating that most artificial agents were not similar to humans even after 2000 levels of training. In the Supplemental Information, we also show heatmaps for individual participants, and compare human players and artificial agents on other metrics, such as how often the digital self was made to inefficiently interact with other possible selves or walls (Figure S10-14).

In sum, the behavior 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 (i) any description of how the game works, (ii) any explicit instruction to navigate to the goal in as few moves as possible, and even when (iii) 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 optimality 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 move, even though only one agent was truly controlled by the player. In order to orient on their digital self, players needed to eliminate from consideration those avatars that moved in unexpected directions after a given keypress (Figure 3).

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

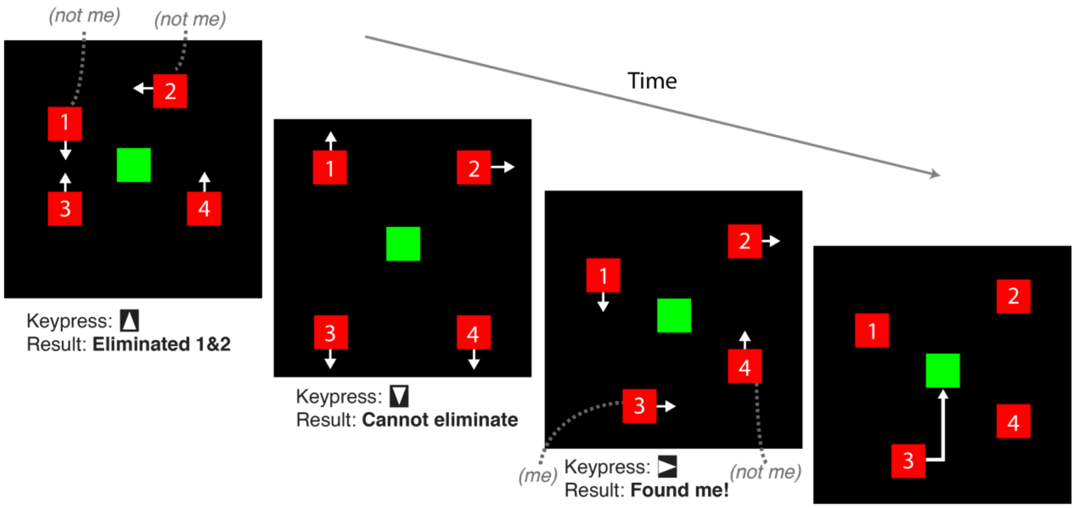

Figure 3. The Contingency Game. In this example, moving UP eliminates the top two candidate selves (#1 and 2), which do not move in the direction of the keypress. In frame 2, moving DOWN does not help you find your digital self, since by chance both the remaining possible selves (i.e., #3 and 4) move 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.

Figure 4A-B shows that human players quickly plateaued, reaching optimal play on the first level, and then fluctuating in and out of optimal play thereafter, perhaps because participants occasionally tried the optimal strategy but then returned to a lazy one, i.e., hitting the same key several times until one avatar was clearly displaced from the others. On average, humans took significantly more steps than the self-class to solve each level over the first 100 levels (M_{human} = 13.3 vs. $M_{\text{self-class}}$ = 9.5, $t(19.5) = 5.1$, $p < .001$, $d = 1.62$).

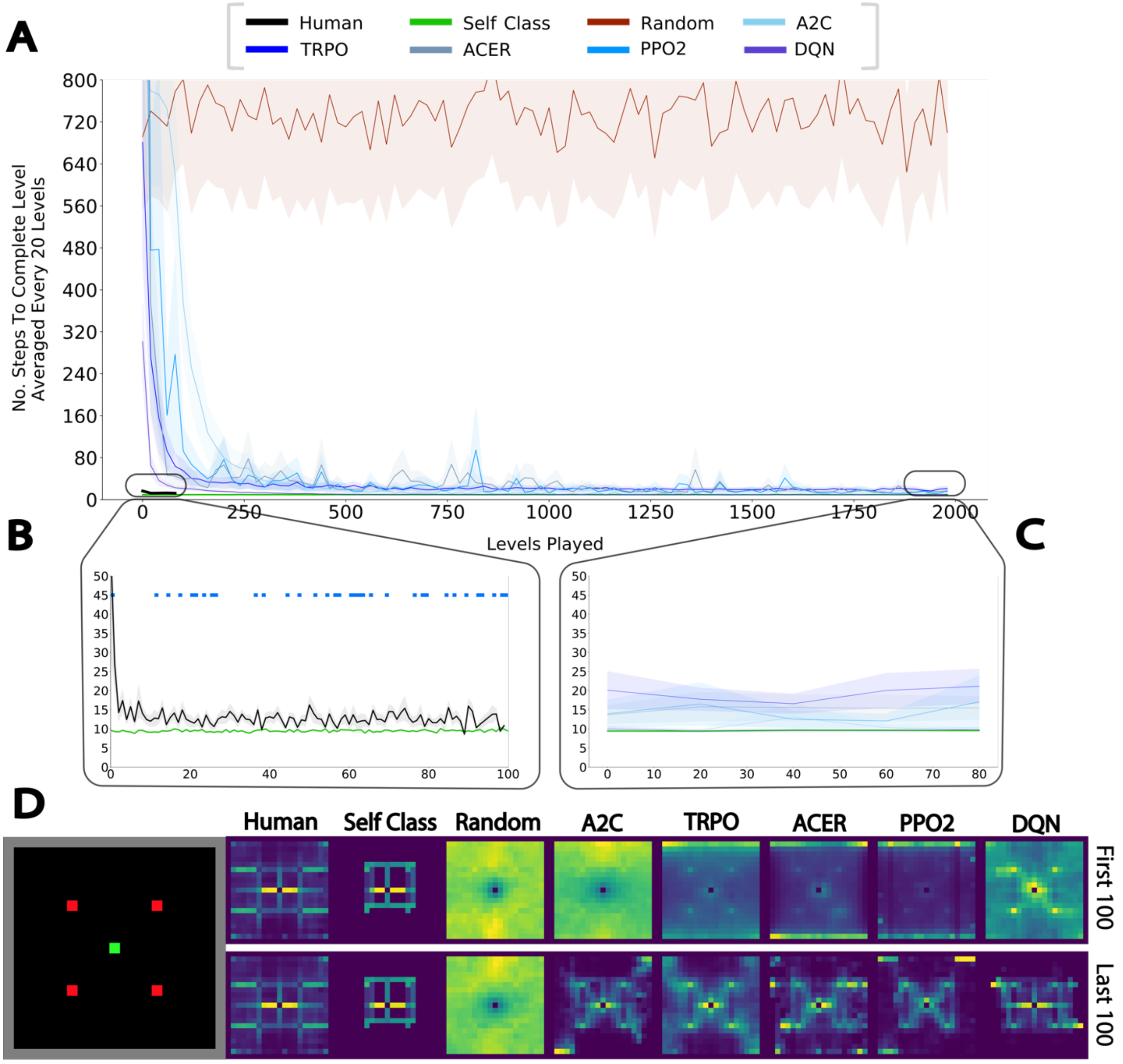

Figure 4. Results of Study 2 (Contingency Game). Number of steps taken by all agents, averaged every 20 levels. Error shading reflects standard error of the mean. (B) Zooming in on level-by-level human players and the self-class for the first hundred levels. Horizontal lines above the plot indicate levels where human performance was indistinguishable from optimal play (i.e., Bayes Factor above 1.0). (C) Zooming in on artificial players for the last hundred levels, averaged every 20 levels. (D) Heatmaps of 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.

Why did human players play slightly suboptimally? As explained above, one likely reason is that they took the ‘lazy’ strategy of initially finding the digital self by repeatedly hitting the same key in one direction, e.g., pressing RIGHT until one agent was clearly more displaced than the others. This is suggested by the human heatmap (Figure 4D), where we see horizontal lines emanating outwards from the starting location (indicated in blueish-green). This behavior 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 than to attend to which of four avatars is consistently responding contingently to changes in your key press. 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 (33).

In contrast to human players, the artificial players required several hundred levels before their performance plateaued (Figure 4D). Unlike in Study 1, some artificial players (DQN, A2C) did achieve optimal performance by the end of training (Figure 4C and Table S2). This is likely because seeing an observable consequence for each action enabled the algorithms to learn. 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 behavioral patterns? The heatmaps in Figure 4D show that, for the first hundred levels, artificial players exhibited more dispersed behavioral patterns than humans and some 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: DQN most resembled human players and other RL algorithms showed an ‘X’ pattern of exploration that was unlike human behavior, whereas the self-class showed the optimal path by not dispersing to the edges. Quantitatively comparing the heatmaps, we see that the mean MSE score is 11028 for the first hundred levels and 6132 for the last hundred levels, showing that the behavioral patterns of the artificial agents became increasingly human-like with training (which is not to say that they have anything like human-like self representations). Table S6 shows that all artificial agents except PPO2 had a significantly different MSE from the self-class during their last hundred levels as compared to humans during their first hundred levels (ps < .001), indicating that no artificial agent was similar to humans even after 2000 levels of training. Interestingly DQN, ACER, and A2C had a *smaller* MSE from the self-class compared to humans, but they still significantly differed from humans, indicating that that although their heatmaps were closer to self-class they had a different self-orientation strategy from humans.

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 to the Self Class. Figure S5 plots this distance for the first level and last level for human (level 100) and artificial players (level 2000). 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. Self-orienting in artificial players starts at random on level 1, but by the end of training some of the algorithms (ACER and TRPO) are optimal. This suggests that some algorithms might be learning to behave in a way that is similar to self-orienting, while humans learn to optimally self-orient.

If the artificial players truly learn how to efficiently self-orient, then we should also expect them to be robust to environmental changes that affect the self-orienting task. To explore this, after the artificial agents learned for 2000 levels, we added an additional 'mock possible self' to the game, which was colored 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 a further ~700 levels to recover their pre-perturbation performance levels; although some algorithms, like DQN, never do (Figure S2). This pattern suggests that the algorithms did not learn a robust self-orienting strategy.

In short, Study 2 shows another context in which people efficiently find their digital selves under ambiguous conditions, by presenting 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 the reason they were able to learn to behave in a way that is similar to self-orienting in the Contingency Game but not the Logic Game was that in the contingency game actions always led to observable consequences. However, even though artificial agents learned to behave optimally in the Contingency Game, they failed on the robustness test, suggesting that artificial agents did not actually learn to self-orient and they did not learn and behave in the same way that people do on these tasks.

Studies 3-4 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 game one has never played before. 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 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 ($M_{\text{human}} = 25.7$ vs. $M_{\text{self-class}} = 10.7$, $t(19.0) = 6.7$, $p < .001$, $d = 2.13$). 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, e.g., 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 (Figure 5).

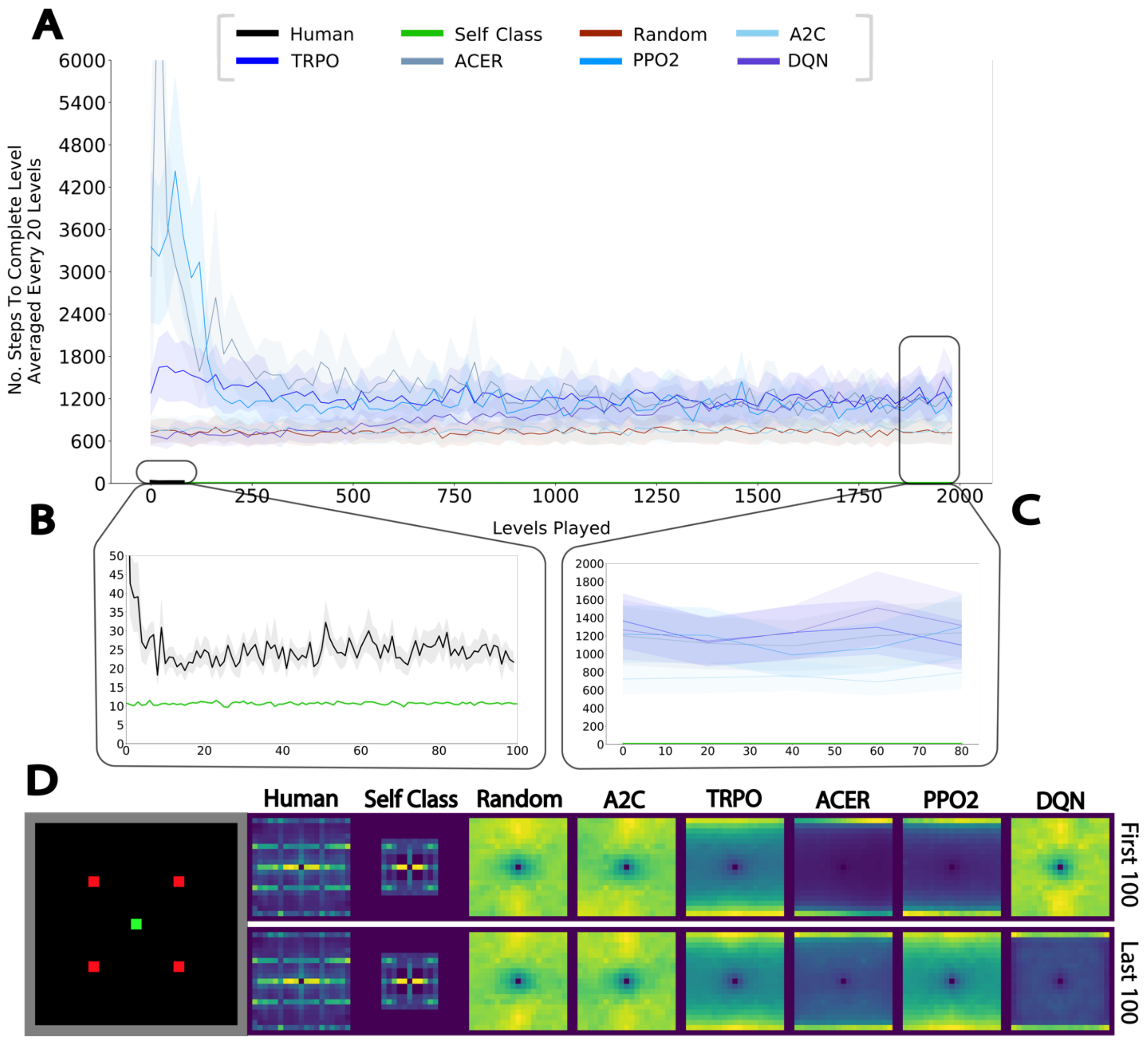

Figure 5. Results of Study 3 (Switching Mappings Game). Number of steps taken by all agents, averaged every 20 levels. Error shading reflects standard error of the mean. (B) Zooming in on level-by-level human players and the self-class for the first hundred levels. Horizontal lines above the plot indicate levels where human performance was indistinguishable from optimal play (i.e., Bayes Factor above 1.0). (C) Zooming in on artificial players for the last hundred levels, averaged every 20 levels. (D) Heatmaps of 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.

Strikingly, the artificial agents were not able to learn the game at all (Figure 5A-C and Table S3). Most likely, this is because their action policies depended on observing consistent action-state mappings, which in this game came undone every time a key-action mapping switched between levels. To test this possibility, we also re-ran ten random seeds of all artificial agents on variants of the game in which the key-action mappings switched less frequently: every hundred or two hundred levels. In these cases, we saw that the algorithms did learn, although they had to relearn from scratch every time the key-action mappings were shuffled (Figure S6). Hence, 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 behavioral patterns? When we shuffled the key-action mappings after each level, the behaviors of the artificial agents were dispersed and random-looking (Figure 5D), appearing different from human play. By plotting the average distance of the digital self from the reward on the first and last levels of play (Figure S8), we again see that the artificial agents did not improve (although behavioral patterns became more distinguished if we decreased how often keys are shuffled; Figure S7). In contrast, humans did improve, although even after a hundred levels they were still suboptimal. As expected, Table S7 shows that all artificial agents were significantly further away from the self-class during their last hundred levels of play than were humans during their first hundred levels of play (*p*s < .001), indicating that artificial agents were not similar to humans even after 2000 levels of gameplay.

It is important to emphasize the qualitative difference between human and RL agent behavior 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. Rather, the RL agents are stuck in a local microcosm each time, and try to generalize local key-mappings to the next level, which constantly fails them. Such failures may seem obvious given the way that RL agents learn and behave, but that is exactly our point: well-known algorithms do not seem to learn and behave in the same way that people do on these tasks.

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 over specialized 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. 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 behavior 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 artificial algorithms would be less efficient than human players.

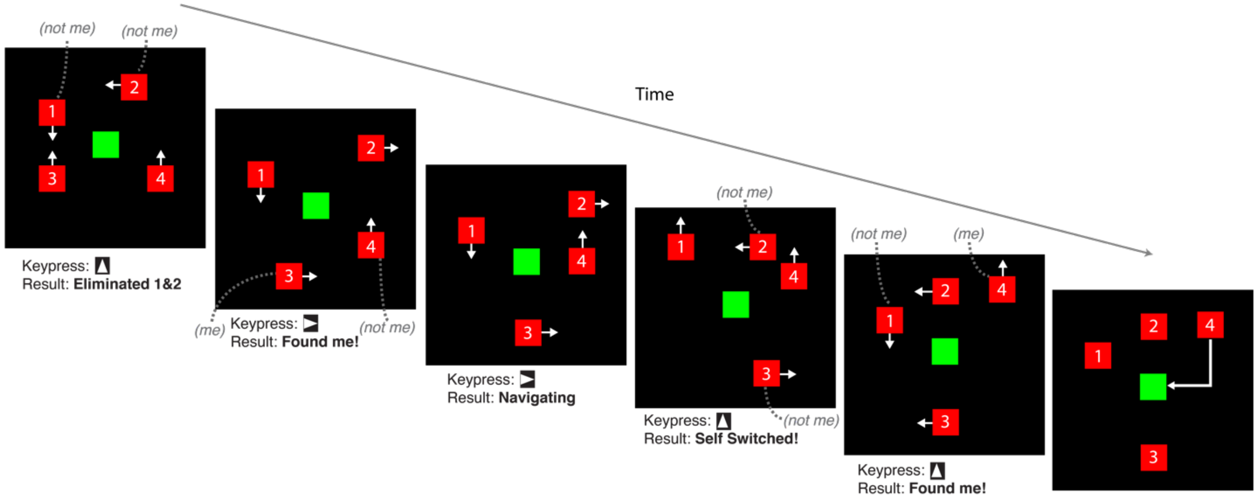

Figure 6. The Switching Embodiments Game. In this example, moving UP on the first frame eliminates two candidate selves (#1, 2), because they do not move in the direction of the keypress. Moving RIGHT on the second frame eliminates the last possible self (#4), revealing the digital self (#3). On frame 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.

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 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.

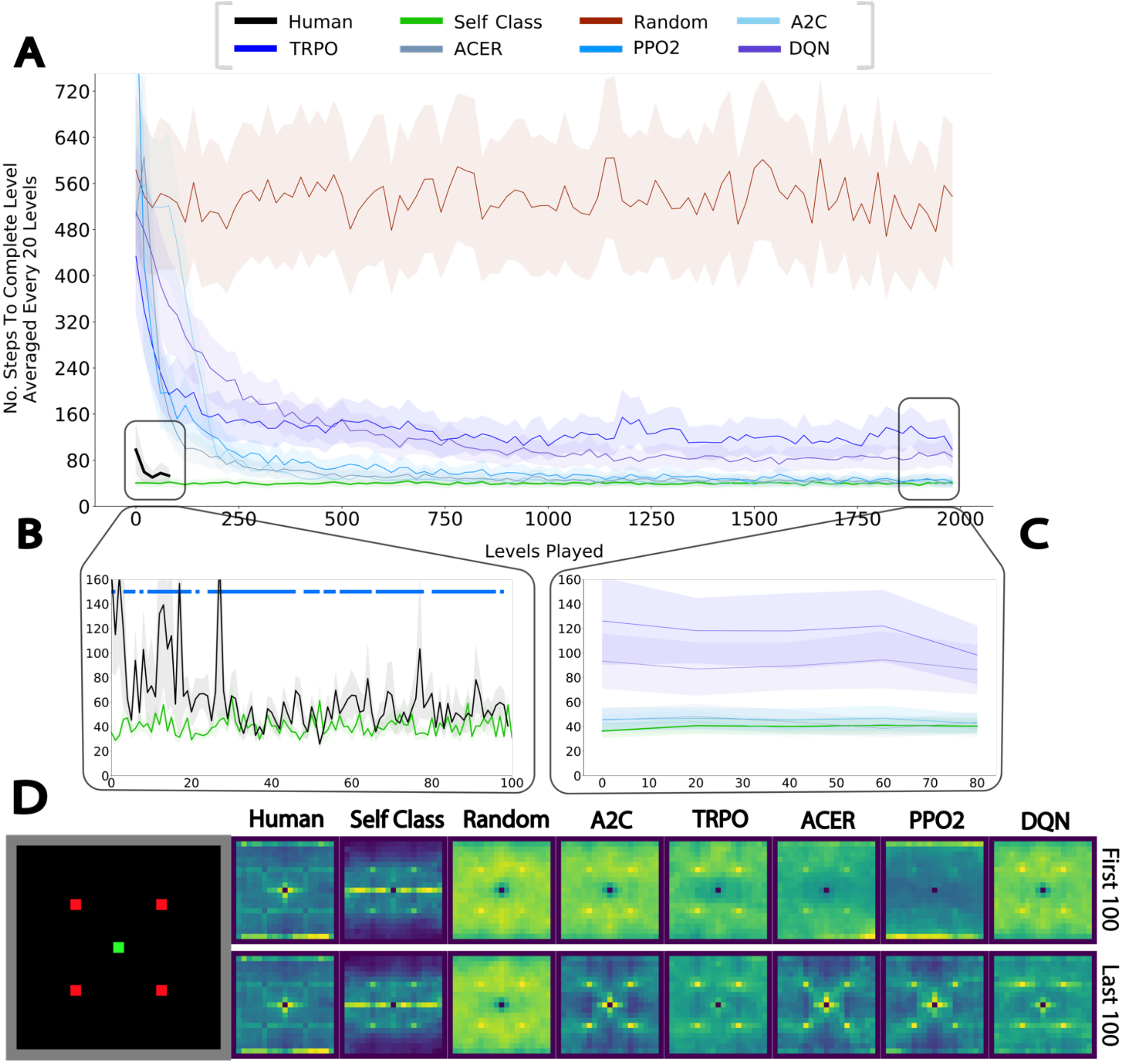

Figure 7. Results for Study 4 (Switching Embodiments Game). (A) Number of steps taken by all agents, averaged every 20 levels. Error shading reflects standard error of the mean. (B) Zooming

in on level-by-level human players and the self-class for the first hundred levels. Horizontal lines above the plot indicate levels where human performance is indistinguishable from optimal play (i.e., Bayes Factor above 1.0). (C) Zooming in on artificial players for the last hundred levels, averaged every 20 levels. (D) Heatmaps of 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.

All artificial agents were able to learn the game, with some (A2C, ACER, and PPO2) even reaching optimal play after ~ 600 levels (Figure 7A-C and Table S4). 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 maneuver 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.

Did humans and RL agents follow similar behavioral patterns? The exploration heatmaps 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 (Figure 7D). Otherwise, the exploration patterns look similar. After learning, some artificial agents (PPO2, A2C, ACER) began to resemble the self-class. Quantitatively comparing the heatmaps, we see that the MSE score is 8567 for the first hundred levels and 4649 for the last hundred, showing that patterns of algorithms become more human-like after training. However, Table S8 shows that all artificial agents except TRPO were significantly *closer* to the self-class during their last hundred levels of play than were humans during their first hundred levels of play (ps < .001), indicating that most artificial agents were not similar to humans after 2000 levels of gameplay.

By plotting the average distance of the digital self from the reward on the first and last levels of play (Figure S9), we see that both human and artificial agents improved.

As in Studies 1-2, we tested whether the artificial players learned a general strategy of self-orienting by adding a 'mock possible self', which was colored red and moved like the other possible selves, and which could never be the digital self that agents controlled. Different from Studies 1-3, after this mock agent was added, most AI players (i.e., PPO2, ACER, A2C) kept performing close to optimal (Figure S3), perhaps because (as we noted above) these agents learned a useful strategy of keeping all agents closer to the goal. Based on this hypothesis, we created a stronger stress test that should pose more of a challenge to these algorithms, if they are indeed using such a strategy: after every embodiment switch, the mock self started navigating toward the goal, so as to 'fool' the algorithm into trying to navigate it to the reward; when the mock self was just one step away from the goal, we prevented it from reaching the goal by moving randomly again, until the next embodiment switch. Most artificial agents failed this test (Figure S4), never recovering their pre-perturbation performance levels, suggesting that they did not learn a robust self-orienting strategy. Rather, given that the RL agents were disrupted by this second perturbation, this suggests they were just attempting to move whichever agent was closest to the reward.

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, they learned to play at optimal levels, but they failed on the robustness test, suggesting that artificial agents do not actually learn to self-orient.

General Discussion

What would it mean for a machine to have a minimal notion of 'self'? Paul et al. (14) propose a representation that points to a spatio-temporal entity in the world and tags it as the agent that is doing the representing and taking actions in the world. The paper develops a new account of how and why such a self should be concretely represented in artificial intelligence algorithms, in order solve a basic problem that must be solved continually by any intelligent agent—human or artificial—that learns, thinks and acts for itself. Extending this proposal, we tasked current game-playing algorithms with learning such information from scratch, and found them to be structurally unsuited for ever learning the notion that the game contains an avatar entity, and self-orienting towards an avatar. Even when RL algorithms did learn to play the games we studied, our stress tests showed that these algorithms did not actually learn a robust self-orienting strategy that generalizes. These results are consistent with the interpretation that the RL agents tested here did not ever learn to self-orient, even though they made progress in solving specific instantiations of games using less flexible strategies. Although the algorithms that we used are not the best of contemporary AI, they are well-known baselines for building agents that operate autonomously in some environments and they embody the thesis of reinforcement learning that some prominent AI researchers (25-27) have suggested is a scaling route to building fully general AI with human-level intelligence or beyond.

Our contributions include a test bed that can be expanded to examine whether algorithms have a minimal notion of self. Specifically, we created a series of litmus tests of whether agents have a minimal self-representation, and metrics of optimality and AI comparisons by which to assess the extent to which humans are effective at self-orienting.

Finally, although we have studied an ability that happens very quickly in humans, we also see how consequential this ability 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 concreteness to an ancient topic of "self representation" that previously escaped computational rigor. While the very latest AI can even play multiple games (34, 35), our results predict that they will learn these games in a non-human like way.

Methods

Study 1: Logic Game

*Participants***.** We recruited 20 participants (40% female, M_{age} = 42), paying them $1 each. All participants passed attention checks, and at least one of the two comprehension checks. Participants completed the game in 7.2 minutes on average.

Game and Agents. The Logic Game consisted of a 9x9 grid space. Each possible self was neighbored by three walls. On each level we varied two factors: the positions of the walls neighboring each possible self, and the starting location of the digital self (Figure S15). The game is visualized in Figure 1, and parameters for each of the game-playing RL agents are provided in the Supplemental Information (Tables S9-13). In order to optimally self-orient and navigate the digital self to the reward, the hard-coded self-class employed the logic in Figure S16.

Study 2: Contingency Game

Participants. We recruited 20 participants (42% female, M_{age} = 44), paying them $1 each. All passed attention checks and at least one of the two comprehension checks. They completed the game in 9.8 minutes on average.

Game and Agents. The Contingency Game consisted of a 21x21 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 9x9 space centered at their starting locations. To optimally orient on the digital self and navigate it to the reward, the self-class employed the logic in Figure S17.

Study 3: Switching Mappings Game

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

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 Figure S18.

Study 4: Switching Embodiments Game

Participants. We recruited 18 participants (56% female, M_{age} = 38), paying them $4 each. All passed attention checks and at least one of the two comprehension checks. They completed the game in 32 minutes on average. The study took approximately 20 minutes longer to complete than the Contingency Game (Study 2) and 12 minutes longer to complete than the Switching Mappings Game (Study 3), probably because the embodiment switching disrupted play.

Game and Agents. The environment was the same as the Contingency Game. The digital self switched embodiments every 7 moves—exactly one move before when an optimal player in Studies 2a-b would have finished the game. In order to optimally self-orient and navigate to the reward, the self-class employed the logic in Figure S19.

REFERENCES

1. James W, Burkhardt F, Bowers F, & Skrupskelis IK (1890) *The principles of psychology* (Macmillan London).
2. Belk RW (2013) Extended self in a digital world. *Journal of Consumer Research* 40(3):477-500.
3. Buckner RL & Carroll DC (2007) Self-projection and the brain. *Trends in Cognitive Sciences* 11(2):49-57.
4. Dennett DC (2014) The self as the center of narrative gravity. *Self and consciousness*, (Psychology Press), pp 111-123.
5. Sui J & Humphreys GW (2015) The integrative self: How self-reference integrates perception and memory. *Trends in Cognitive Sciences* 19(12):719-728.
6. Blanke O & Metzinger T (2009) Full-body illusions and minimal phenomenal selfhood. *Trends in Cognitive Sciences* 13(1):7-13.
7. Bem DJ (1967) Self-perception: An alternative interpretation of cognitive dissonance phenomena. *Psychol. Rev.* 74(3):183.
8. McConnell AR (2011) The multiple self-aspects framework: Self-concept representation and its implications. *Personality and Social Psychology Review* 15(1):3-27.

9. Sanchez-Vives MV & Slater M (2005) From presence to consciousness through virtual reality. *Nature Reviews Neuroscience* 6(4):332-339.

10. Strawson G (1996) The sense of the self. *London Review of Books* 18(8).

11. Dennett DC (2016) Where am I? *Science fiction and philosophy: from time travel to superintelligence*, ed Schneider S (John Wiley & Sons).

12. Nozick R (1981) *Philosophical explanations* (Harvard University Press).

13. Perry J (1972) Can the self divide? *The Journal of Philosophy* 69(16):463-488.

14. Paul L, Ullman TE, De Freitas J, & Tenenbaum J (2022) "Reverse-engineering a self.

15. Andrychowicz M*, et al.* (2017) Hindsight experience replay. *Advances in Neural Information Processing Systems* 30.

16. Hausknecht M & Stone P (2015) Deep recurrent q-learning for partially observable mdps. *2015 AAAI Fall Symposium Series*.

17. Schaul T, Quan J, Antonoglou I, & Silver D (2015) Prioritized experience replay. *arXiv preprint arXiv:1511.05952*.

18. Van Hasselt H, Guez A, & Silver D (2016) Deep reinforcement learning with double q-learning. *Proceedings of the AAAI Conference on Artificial Intelligence*.

19. Wang Z*, et al.* (2016) Dueling network architectures for deep reinforcement learning. *International Conference on Machine Learning*, (PMLR), pp 1995-2003.

20. Mnih V*, et al.* (2013) Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

21. Kaiser L*, et al.* (2019) Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*.

22. Dubey R, Agrawal P, Pathak D, Griffiths TL, & Efros AA (2018) Investigating human priors for playing video games. *arXiv preprint arXiv:1802.10217*.

23. Tsividis PA*, et al.* (2021) Human-level reinforcement learning through theory-based modeling, exploration, and planning. *arXiv preprint arXiv:2107.12544*.

24. Tsividis PA, Pouncy T, Xu JL, Tenenbaum JB, & Gershman SJ (2017) Human learning in Atari. *2017 AAAI spring symposium series*.

25. Silver D, Singh S, Precup D, & Sutton RS (2021) Reward is enough. *Artificial Intelligence* 299:103535.

26. Botvinick M*, et al.* (2017) Building machines that learn and think for themselves. *Behavioral and Brain Sciences* 40.

27. Botvinick M*, et al.* (2017) Building machines that learn and think for themselves: Commentary on lake et al., behavioral and brain sciences, 2017. *arXiv preprint arXiv:1711.08378*.

28. Pan X*, et al.* (2019) How You Act Tells a Lot: Privacy-Leakage Attack on Deep Reinforcement Learning. *arXiv preprint arXiv:1904.11082*.

29. Brockman G*, et al.* (2016) Openai gym. *arXiv preprint arXiv:1606.01540*.

30. Hill A*, et al.* (2018) Stable baselines.

31. Dhariwal P*, et al.* (2017) Openai baselines. https://github.com/openai/baselines, 2017.

32. Rouder JN, Speckman PL, Sun D, Morey RD, & Iverson G (2009) Bayesian t tests for accepting and rejecting the null hypothesis. *Psychonomic Bulletin & Review* 16(2):225-237.

33. Vul E, Goodman N, Griffiths TL, & Tenenbaum JB (2014) One and done? Optimal decisions from very few samples. *Cognitive Science* 38(4):599-637.

34. Reed S*, et al.* (2022) A generalist agent. *arXiv preprint arXiv:2205.06175*.

35. Schrittwieser J*, et al.* (2020) Mastering atari, go, chess and shogi by planning with a learned model. *Nature* 588(7839):604-609.