Playful and Exploratory Behavior from the Maximum Occupancy Principle

Chiara Mastrogiuseppe
Center for Brain and Cognition, Department of Engineering
Universitat Pompeu Fabra
Barcelona, Spain
chiara.mastrogiuseppe@upf.edu

Rubén Moreno-Bote
Center for Brain and Cognition, Department of Engineering,
Serra Húnter Fellow Programme
Universitat Pompeu Fabra
Barcelona, Spain
ruben.moreno@upf.edu

Abstract

We build on the Maximum Occupancy Principle (MOP) and show complex and playful behavior emerging from intrinsic motivation to occupy action space. Relevantly, the drive to occupy action space as uniformly as possible in the long run, while avoiding terminal states, leads to interesting behaviors, such as non-trivial interaction with external objects. We show that MOP agents in navigation tasks are inherently curious, as they are attracted by the possibility of playing with available objects or using them as tools to visit larger regions of space. This principle is then extended to neural activity (NeuroMOP). We introduce a more complex continuous navigation problem where the motor output of the agent is defined by two units of a recurrent neural network of fixed weights. We show that a MOP controller can drive the network's activity and lead the motor output units to occupy the whole available space. This example highlights the potential of MOP as a principle not only for behavior but also for neural activity. All together, these results indicate MOP as a possible principle underlying various aspects of natural behavior, reconciling multiple perspectives of intrinsic motivation, such as curiosity and exploration.

1 Introduction

Natural behavior is inherently complex and dynamic. When free to act in the world, natural agents display a wide range of behaviors that are far from trivial and complex to characterize [1, 2, 3]. In contrast, in constrained environments (e.g., experimental settings) where a task or a reward function is externally imposed, the inherently variable patterns of behavior collapse into few deterministic patterns that can be studied and understood in term of the neural activity generating them [4, 5, 6]. In these cases, behavior tends to align closely with the specific goals dictated by tasks or rewards, potentially masking the principles driving behavior in more natural settings. Preserving behavioral variability may have been an essential adaptive strategy selected by evolution to navigate the stochastic and unpredictable nature of the world [7, 8]. This raises the question of what drives behavior when task instructions are not available or an extrinsic reward function is not imposed.

If we abandon the idea of extrinsic rewards, what are the intrinsic motivations that drive natural behavior? Different approaches to characterize intrinsic motivation [9] include seeking intrinsic

Accepted at the Intrinsically Motivated Open-ended Learning workshop at NeurIPS 2024.

goals such as novelty and surprise [10, 11, 12] or curiosity-driven exploration [13, 14, 15]. While they clearly have the significant advantage of leading to large exploration – hence potentially to the discovery of new and 'better' states for the agent –, they suffer from the tendency of reducing visitation as soon as the optimal policy is learned. In other words, the agent reduces its curiosity about states that are not 'new', with the consequent collapsing of variability to a (or few) deterministic behavior(s). As well, reward-free RL algorithms as Empowerment (MPOW) [16, 17, 18] or Free Energy Principle (FEP) [19, 20] favor states leading to large and predictable changes or by matching a desired hand-crafted target distribution, respectively. While they do allow for task execution without the need to specify extrinsic rewards, these approaches have been proven to collapse into deterministic and highly stereotyped patterns of behavior at least in deterministic environments [21, 22], thus failing in capturing the observed wide variability of natural behavior.

An alternative approach is to take behavioral variability not uniquely as the consequence of an opportune strategy to maximize reward but as the ultimate goal of agents itself. This idea has been formalized in the Maximum Occupancy Principle (MOP) [21]. MOP agents act in the world with the only goal of maximizing the future cumulative discounted entropy of actions taken and states visited, while extrinsic rewards (such as energy sources) are only the means to keep visiting action-state space. This principle equips agents with a survival instinct and a natural curiosity. Indeed, agents will tend to avoid states that are dangerous for their survival as they do not allow for further entropy production (*survival*) and to prefer states and actions that are less likely. Here, we show that MOP agents navigating in complex environments are naturally attracted by external objects (*curiosity*) and use objects as 'tools' if it increases their possibility to occupy action-state space in the long run. Then, we take a step further into characterizing the neural basis of the observed motor behavior. We explore the extension to neural activity of the Maximum Occupancy Principle (NeuroMOP) [23]. We show that MOP can drive the output of a recurrent neural network (RNN) into complex behaviors in a navigation task, such as the exploration of the whole arena (curiosity) and avoidance of dangerous states (survival).

2 The Maximum Occupancy Principle

We build on MOP [21, 22] and model a discrete-time sequential Markov decision process $(\mathcal{S}, \mathcal{A}, p, R)$ where p is the state transition probability function and R is the function defining the intrinsic reward. At every time step t the agent is in a state $s_t \in \mathcal{S}$ and interacts with the environment through actions $a_t \in \mathcal{A}$ sampled from a policy $\pi(a_t|s_t)$. The agent experiences a sequence of states and actions $\tau = (s_0, a_0, s_1, a_1, \ldots, s_t, a_t, \ldots)$ and receives an intrinsic reward given by

$$R(\tau) = -\sum_t \gamma^t \ln \left(\pi^\alpha(a_t|s_t) p^\beta(s_{t+1}|s_t, a_t)\right) , \tag{1}$$

where $\alpha > 0$ and $\beta \geq 0$ are the parameters rescaling the policy and the transition probability term and $0 \leq \gamma < 1$ is the discount factor. This return defines an agent favoring actions and states of lower probability whenever possible. The value function is defined as the expected intrinsic return and can be rewritten as

$$\begin{aligned} V(s) &= \mathbb{E}_{a_t \sim \pi(\cdot|s_t), s_{t+1} \sim p(\cdot|s_t, a_t)} \left[R(\tau)|s_0 = s\right] \\ &= \mathbb{E}_{a_t \sim \pi(\cdot|s_t), s_{t+1} \sim p(\cdot|s_t, a_t)} \left[\sum_t \gamma^t (\alpha \mathcal{H}(\mathcal{A}|s_t) + \beta \mathcal{H}(\mathcal{S}'|s_t, a_t))|s_0 = s\right] , \end{aligned} \tag{2}$$

where s is taken as the initial condition of the expected return, $\mathcal{S}'$ is the state space in the next time step and we recognize an action-entropy term $\mathcal{H}(\mathcal{A}|s) = -\sum_a \pi(a) \ln \pi(a|s)$ and a state-entropy term $\mathcal{H}(\mathcal{S}|s, a) = -\sum_{s'} p(s'|s, a) \ln p(s'|s, a)$. We consider state-dependent action sets $\mathcal{A}(s)$. In particular, we introduce terminal states $s^\dagger$ as states where doing `nothing` is the only available action and that cannot be escaped. Therefore in a terminal state future entropy is zero, that is, its corresponding value function is zero, $V(s^\dagger) = 0$. An agent maximizing the return in Eq. 1 will naturally avoid terminal states, as any non-terminal state will be preferred given its value greater than zero.

The optimal policy for the MOP agent is

$$\pi^*(a_t|s_t) = Z^{-1}(s_t) \exp \left(\alpha^{-1} \beta \mathcal{H}(\mathcal{S}'|s_t, a_t) + \alpha^{-1} \gamma \sum_{s_{t+1}} p(s_{t+1}|s_t, a_t) V^*(s_{t+1})\right) , \tag{3}$$

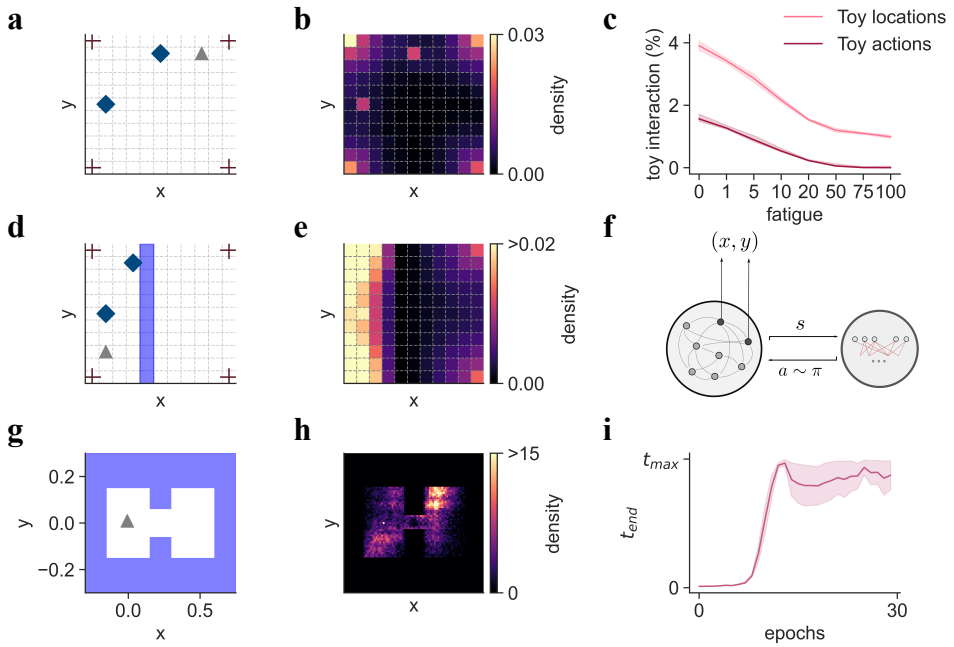

Figure 1: **a-c** Playful interactions with external objects. **a** Scheme of the toy arena. An agent (grey triangle) navigates a gridworld where four extrinsic food rewards are located in the four corner of the arena (red plus) and two toys are in fixed locations (blue diamond). **b** Arena occupancy for `fatigue` $= 0$. **c** As the fatigue of interacting with the toy increases, the agent decreases the visitation of the grid positions where the toys are located (toy locations) and reduces to zero the 'playing' with the toys (toy actions). **d-e** Tendency to interact with objects can lead to larger exploration. **d** Scheme of the river arena: extrinsic food rewards in the four corner of the arena (red plus), a river (blue stripe) and two toys (blue diamond) that can now be moved around by the agent (grey triangle). The agent can cross the river and reach the other side of the arena only by picking up the toy and moving with it. **e** Arena occupancy for `fatigue` $= 0$. **f-i** NeuroMOP in a navigating task. **f** Position in the arena is defined by two motor output neurons of an RNN driven by the currents injected from an external controller following MOP. **g** Scheme of the arena. The agent needs to control the motor output (grey triangle). Terminal absorbing states are encountered when falling in the sea surrounding the 'safe' area. **h** The agent drives the RNN into exploring both sides of the arena, as uniformly as possible and avoiding falling into the sea. **i** Learning of the value function with lifetime increasing over epochs ($t_{max} = 1000$ steps during training). In (a-e) we simulate $N_{traj} = 10$ trajectories and in (f-i) $N_{ag} = 5$ agents are trained. Error bars are standard errors of the mean (SEM).

where $Z(s) = \sum_a \exp\left(\alpha^{-1}\beta\mathcal{H}(\mathcal{S}'|s,a) + \alpha^{-1}\gamma \sum_{s'} p(s'|s,a)V^*(s')\right)$ is the partition function.

The optimal value function coming from following the optimal policy obeys the consistency equation

$$V^*(s_t) = \alpha \ln Z(s_t) = \alpha \ln \sum_{a_t} \exp\left(\alpha^{-1}\beta\mathcal{H}(\mathcal{S}'|s_t,a_t) + \alpha^{-1}\gamma \sum_{s_{t+1}} p(s_{t+1}|s_t,a_t)V^*(s_{t+1})\right). \tag{4}$$

In continuous state spaces, the value function is approximated via a one-hidden layer feedforward network trained to minimize the mean squared error between the approximated value and the one obeying the consistency equation in Eq. 4. Details of the training are provided in Appendix A.3.

3 Results

We first consider a discrete gridworld. The environment consists of a 11×11 grid with food sources in the four corners of the grid. In the environment, two toys are placed in fixed locations (Fig. 1a). The agent can move of one step in all directions in space and, when in a position of the grid where a

toy is located (toy location), it can also 'play' with it by either lifting it up, rotating it, throwing it in the air or bouncing it (toy actions). The agent has an internal energy u that depletes at every action of one unit ($\Delta u = -1$) and that can be restored by reaching the food sources ($\Delta u = +10$, up to a total maximum internal energy of $U_{max} = 100$). The agent enters a terminal state if it depletes all the internal energy. Note that, while reaching food is necessary to restore internal energy, there is no incentive to the interaction with the external toys outside an increased action set. We show that the agent has a natural tendency to spend time at the toy locations, uniformly between the two, and to interact with them by largely picking toy actions (Fig. 1b). Interacting with the toy can be costly in terms of *fatigue*, therefore we model an action-dependent energy consumption with an additional energy depletion ($\Delta u = -1 - \texttt{fatigue}$) for the toy actions. As the `fatigue` increases, the agent progressively reduces the interaction with the toys (i.e., percentage of toy actions selected) until it stops playing with them altogether and treating toy-related states (as measured by the percentage of time spent in toy locations) as any other state (Fig. 1c).

We further consider a modification of this environment by introducing a river that divides the gridworld into two sections the agent cannot cross alone or it enters in a terminal state. Terminal states can also occur if the agent's internal energy reaches zero. In the left side of the environment, the agent finds two toys it can now pick up, carry around with it, and put down at its own will. Only when endowed with it, the agent can cross the river and reach the other side (Fig. 1d) (the toy is, e.g., a floating ball). Here, we model a forgetful agent not keeping track of changes in the toy positions, hence the agent may encounter one of the toys and cross the river only by keeping its policy as stochastic as possible. We see the agent reaching the other side and occupying both sides of the river (Fig. 1e). As before, we introduce a fatigue parameter by modelling a state-dependent energy consumption, with $\Delta u = -1 - \texttt{fatigue}$, when the agent is in possess of a toy. As the fatigue increases, the agent reduces the time it carries the toy around the arena and reduces to zero the time spent on the right side of the arena (not shown).

Finally, how could behavioral variability be generated by the brain? Here we ask what the neural basis of the behavioral variability is [7], and hypothesize a new link with neural variability by extending the principle to neural activity (NeuroMOP) [23]. Specifically, we model a MOP agent (e.g., the central nervous system) interacting with an environment modelled via an RNN of fixed weights (e.g., the peripheral nervous system). Location in a continuous world is defined by two fixed motor output neuron (Fig. 1f). The arena consists of two squared islands connected by a narrow bridge and surrounded by the sea. Terminal states are encountered whenever the agent (i.e., its (x, y) location) falls into the sea (Fig. 1g). The agent follows MOP, hence it aims at maximizing the entropy of the series of actions taken, which are currents injected into the RNN. We see the agent learns to control the complex dynamics of the RNN as to explore all the available safe states of the world and to avoid terminal states (Fig. 1h). Given the continuous state space, the value function is approximated via a feedforward network of one hidden layer. By interacting with the environment, the agent learns to implement MOP by avoiding terminal states and surviving the whole length of the simulation (Fig. 1i).

4 Conclusions

We explored three different navigation tasks to highlight key aspects and potential directions for the MOP framework. First, by means of a state dependent action set, we were able to qualitatively reproduce the observed human tendency of being object-oriented [24], reflecting curiosity towards external objects. Our results have shown that MOP agents are naturally attracted by the possibility of interacting with external elements (e.g., toys) located in the arena, while maintaining a tendency to move around. This feature is a defining property of MOP framework, and it underscores a critical point: MOP agents act stochastically when they can and *because* they can, without requiring any (extrinsic) motivations. In other words, actions that elicit the possibility of more actions and experiencing more states will be sought by MOP agents, arguably a defining characteristic of the intuitive notion of curiosity. In a second example, we highlighted MOP's robustness in navigating partially observable environments. The agent's inherent tendency to interact with external objects enabled it to visit novel states and maintain large occupancy, even when objects location changed and was kept unknown. Finally, we investigated whether the ability of MOP agents to navigate in non trivial environments and its variable motor behavior could be originated from variable neural activity. We explored NeuroMOP [23], which posits that maximum occupancy of the series of neural activity

states visited can be taken as a goal of the nervous system. We have shown that a controller following MOP can drive the motor output of a high-dimensional RNN to navigate in a complex environment, suggesting that MOP may serve as a principle underlying natural agents beyond behavior. These findings provide insights on the observed neural variability [25, 26, 27] and its potential causal link to curiosity and behavioral variability as a whole [7].

In all these tasks, large state visitation arises from the stochasticity of actions, as we consider deterministic environments: it is the optimization of action-state path entropy that leads to large behavioral diversity. In contrast, optimizing solely for future state occupancy would not lead to the same agent behavior: a stereotyped strategy, such as uniformly covering the space in a zig-zag pattern, could achieve this goal yet result in highly deterministic behavior. The action entropy term in the intrinsic reward of MOP agents ensures that they strive not only to reach as many states as possible but to do so using a wide range of actions. This ensures the largest possible behavioral repertoire, balancing action and state variability.

In similar navigation tasks, other reward-free intrinsic motivation algorithms as MPOW [16, 17, 18] and FEP [19, 20] have been proven to collapse into trivial and deterministic behavior [21, 22]. We expect analogous collapse in the tasks described here. For instance, in the first example, there is no incentive to interact with the toy (e.g., leading to changes in the state space), hence the results are likely consistent with previous findings [22]. Different behavior may be expected in the river example, where MPOW is likely to show a tool-usage bias by virtue of maximizing the mutual information between actions and states, with a bias for the initial location of the toy. Yet, tendency to interact with the external tool can come at the expenses of the exploration of the arena. As a result, in cases where location of the object is later changed and unknown, MPOW may potentially fail in finding and using the object again.

In the framework of intrinsic motivation, a large behavioral repertoire can emerge by allowing agents to dynamically choose their goals and generate behaviors to achieve them. This approach is observed in intrinsically motivated, goal-conditioned agents known as autotelic agents [28]. Autotelic agents display a large behavioral repertoire, making them a fair comparison to MOP agents. However, as currently modeled, autotelic agents operate under a crucial distinction: each goal is tied to a distinct reward function, and policies must be defined or learned to pursue the selected goals. Consequently, their behavioral repertoire, while diverse, remains goal-conditioned and constrained to those behaviors necessary for goal achievement. In contrast, MOP agents have a single intrinsic objective – maximizing occupancy, and thus implicitly ensuring survival – and spontaneously generate a wide array of temporary subgoals to support this aim. This allows for the greatest possible behavioral diversity, even within each subgoal, enabling MOP agents to maintain the broadest possible behavioral repertoire.

Overall, our results highlight interesting features of MOP: (1) the intrinsic balance between playful interactions and state visitation and (2) its potential to provide an explanation for neural variability and its correlation with variable behavior.

Acknowledgments and Disclosure of Funding

This work is supported by the Howard Hughes Medical Institute (HHMI, ref 55008742), Ministry of Science and Innovation, State Research Agency, European Union (Project PID2023-146524NB-I00 financed by MCIN/AEI/10.13039/501100011033/ ERDF, EU) and ICREA Academia to R.M.-B. and AGAUR-FI ajuts from Generalitat de Catalunya/ESF (2024 FI-B3 00020) to C.M.

References

[1] M. H. Dickinson, C. T. Farley, R. J. Full, M. A. Koehl, R. Kram, and S. Lehman. How animals move: an integrative view. *Science*, 288(5463):100–106, 2000.

[2] S. R. Datta, D. J. Anderson, K. Branson, P. Perona, and A. Leifer. Computational neuroethology: A call to action. *Neuron*, 810(1):11–24, 2019.

[3] D. J. Anderson and P. Perona. Toward a science of computational ethology. *Neuron*, 84(1):18–31, 2014.

[4] M. Jazayeri and A. Afraz. Navigating the Neural Space in Search of the Neural Code. *Neuron*, 93(5):1003–1014, 2017.

[5] J. W. Krakauer, A. A. Ghazanfar, A. Gomez-Marin, M. A. MacIver, and D. Poeppel. Neuroscience needs behavior: Correcting a reductionist bias. *Neuron*, 93(3):480–490, 2017.

[6] A. Gomez-Marin, J. J. Paton, A. R. Kampff, R. M. Costa, and Z. F. Mainen. Big behavioral data: psychology, ethology and the foundations of neuroscience. *Nature Neuroscience*, 17(11):1455–1462, 2014.

[7] A. Renart and C. K. Machens. Variability in neural activity and behavior. *Current Opinion in Neurobiology*, 25:211–220, 2014.

[8] R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*. MIT press Cambridge, 1998.

[9] P. Y. Oudeyer and F. Kaplan. What is intrinsic motivation? a typology of computational approaches. *Frontiers in neurorobotics*, 1(6), 2007.

[10] J. Lehman and K. O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223, 2011.

[11] J. Achiam and S. Sastry. Surprise-based intrinsic motivation for deep reinforcement learning. 2017. ArXiv: 1703.01732.

[12] A. Modirshanechi, S. Becker, J. Brea, and W. Gerstner. Surprise and novelty in the brain. *Current Opinion in Neurobiology*, 82:102758, 2023.

[13] Y. Burda, H. Edwards, D. Pathak, A. Storkey, T. Darrell, and A. A. Efros. Large-Scale Study of Curiosity-Driven Learning. In *International Conference on Learning Representations*, 2019.

[14] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 2778–2787. PMLR, 2017.

[15] A. Modirshanechi, K. Kondrakiewicz, W. Gerstner, and S. Haesler. Curiosity-driven exploration: foundations in neuroscience and computational modeling. *Trends in Neurosciences*, 46(12), 2023.

[16] A.S. Klyubin, D. Polani, and C. L. Nehaniv. Empowerment: a universal agent-centric measure of control. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, pages 128–135, 2005.

[17] T. Jung, D. Polani, and P. Stone. Empowerment for continuous agent—environment systems. *Adaptive Behavior*, 19(1):16–39, 2011.

[18] S. Mohamed and D. Jimenez Rezende. Variational Information Maximisation for Intrinsically Motivated Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

[19] K. Friston, J. Kilner, and L. Harrison. A free energy principle for the brain. *Journal of Physiology-Paris*, 100(1-3):70–87, 2006.

[20] L. Da Costa, N. Sajid, T. Parr, K. Friston, and R. Smith. Reward Maximization Through Discrete Active Inference. *Neural Computation*, 35(5):807–852, 2023.

[21] J. Ramírez-Ruiz, D. Grytskyy, C. Mastrogiuseppe, Y. Habib, and R. Moreno-Bote. Complex behavior from intrinsic motivation to occupy future action-state path space. *Nature Communications*, 15(1):6368, 2024.

[22] R. Moreno-Bote and J. Ramirez-Ruiz. Empowerment, free energy principle and maximum occupancy principle compared. In *NeurIPS 2023 workshop: Information-Theoretic Principles in Cognitive Systems*, 2023.

[23] C. Mastrogiuseppe and R. Moreno-Bote. Controlled maximal variability along with reliable performance in recurrent neural networks. In *Proceedings of the 38th International Conference on Neural Information Processing Systems (NeurIPS 2024)*, 2024.

[24] M. S. Tomov, P. A. Tsividis, T. Pouncy, J. B. Tenenbaum, and S. J. Gershman. The neural architecture of theory-based reinforcement learning. *Neuron*, 111(8):1331–1344, 2023.

[25] W. R. Softky and C. Koch. The highly irregular firing of cortical cells is inconsistent with temporal integration of random EPSPs. *J. Neurosci.*, 13(1):334–350, 1993.

[26] D. J. Tolhurst, J. A. Movshon, and A. F. Dean. The statistical reliability of signals in single neurons in cat and monkey visual cortex. *Vision Res*, 23(8):775–785, 1983.

[27] M. N. Shadlen and W. T. Newsome. The Variable Discharge of Cortical Neurons: Implications for Connectivity, Computation, and Information Coding. *J. Neurosci.*, 18(10):3870–3896, 1998.

[28] C. Colas, T. Karch, O. Sigaud, and P. Y. Oudeyer. Autotelic agents with intrinsically motivated goal-conditioned reinforcement learning: a short survey, 2022. arXiv : 2012.09830.

[29] H. Sompolinsky, A. Crisanti, and H. J. Sommers. Chaos in Random Neural Networks. *Phys. Rev. Lett.*, 61(3):259–262, 1988.

A Details of the simulation

We provide here the details of the examples in the main text. In this paper, we implement MOP agents maximizing action path entropy in deterministic environments, that is, MOP with $\alpha = 1$ and $\beta = 0$. In all examples, the discount factor is set at $\gamma = 0.9$.

A.1 Toy example

Environment A 11×11 arena with four food sources in the four corners. Two toys are located in the arena at fixed toy locations.

State space The Cartesian product between the spatial location (x, y) and internal energy u, i.e., a scalar value in the range $[0, 100]$. When at food source, internal energy is restored of $\Delta u = +r_{food} = +10$ units. When reaching maximum energy capacity, the internal energy state is not changed even when at a food source. Internal energy is decreased at every action of one unit, $\Delta u = -1$. An additional energy decrease is associated with the toy actions via a parameter of $fatigue$ we vary in the simulations, for a total decrease of $\Delta u = -1 - fatigue$. Terminal states $s^\dagger$ are defined as absorbing states in the degenerate space where $(x, y, u = 0)$, independently of the location (x, y).

Action space Action set is state dependent. When not in a toy location or in a terminal state, the agent has 9 actions: `up`, `down`, `left`, `right`, `up left`, `up right`, `down left`, `down right`, and `nothing`. When close to a wall, the number of available actions remains the same but position is unchanged when the agent chooses to go into walls. When in a state s where the location corresponds to one of the toys ($(x, y) = (x_{toy}, y_{toy})$), the agent has 4 additional toy actions: `picking it up`, `throwing it in the air`, `bouncing it`, `rotating it`. Finally, whenever the agent is in an absorbing state $s^\dagger$, only `nothing` is available.

Value function Optimal value is found via value iteration using the iterative map corresponding to Eq. 2. The iterative map has been proven to converge to a unique solution regardless of the initial condition for value functions in the first orthant [21]. Iteration is stopped at epoch l^* where $\max_s |V^{(l^*)}(s) - V^{(l^*-1)}(s)| < 0.01$.

A.2 River example

Environment A 11×11 arena with four food sources in the four corners. Initially, two toys are located in the left part of the arena. The agent can modify the toy locations at its own will by moving the objects around. A river divides vertically the arena at $(x_{river}, y_{river}) = (5, \cdot)$, and the agent can uniquely cross it by using the external toy (e.g., a floating ball).

State space The Cartesian product between the spatial location (x, y), the internal state u, i.e., a scalar value in the range $[0, 100]$, and a binary toy variable toy flagging whether the agent has ($toy = 1$) or has not ($toy = 0$) one of the toys with it. Note that we only provide the agent with a flag and not with the full updated locations of the toys. When at food source, internal energy is restored of $\Delta u = +r_{food} = +10$ units. When reaching maximum energy capacity, the internal energy state is not changed even when at a food source. Internal energy is decreased at every action of one unit as $\Delta u = -1$. An additional energy decrease is associated with the toy actions via a parameter of $fatigue$ we vary in the simulations for a total decrease of $\Delta u = -1 - fatigue$. Terminal states $s^\dagger$ can be encountered either by depletion of internal energy (i.e., $(x, y, u = 0, toy)$), independently of the location and the toy, or by crossing the river without an external toy (i.e., $(x_{river}, y_{river}, u, toy = 0)$), independently of the internal energy.

Action space When the agent is not in possess of the toy ($toy = 0$) and it is not in a toy location (i.e., $(x, y) \neq (x_{toy}, y_{toy})$), it has 9 possible actions: `up`, `down`, `left`, `right`, `up left`, `up right`, `down left`, `down right`, and `nothing`. The action space is not restricted by the presence of the river and the agent can select actions that lead into it. Contrarily, when close to a wall, position is unchanged when the agent chooses to go into walls. According to the toy flag, the agent may have additional actions. When the agent does not hold the toy ($toy = 0$) and in a toy location (i.e., with $(x, y) = (x_{toy}, y_{toy})$), it has an additional action that is to `pick it up`, changing its internal toy flag

to $toy = 1$. When the agent holds the toy ($toy = 1$), in all locations it has an additional action that is to `put it down` there, changing its internal toy flag to $toy = 0$.

Value function Optimal value is found via value iteration using the same iterative map described above and defined in [21]. Again, iteration is stopped at epoch l^* where $\max_s |V^{(l^*)}(s) - V^{(l^*-1)}(s)| < 0.01$. Here, changes in the toy locations do not allow to compute the value function exactly. We train the iterative map for the initial fixed locations of the toy (e.g., locations shown in Fig. 1d) and employ the same value representation throughout the whole simulated trajectory. In other words, navigation in this gridworld relies on the value approximation corresponding to the optimal value computed initially and before changes in the toy locations. After the initial toy locations are changed, the agent has partial information as it remains unaware of the states where it will have an increased action set (e.g., states where it can pick up the toy) until it experiences those states directly. We use this approximation as we find the agent to be still able to survive and navigate the world, albeit with less exploration (Fig. 1e).

A.3 Continuous motor example

Environment A $[-1, 1] \times [-1, 1]$ continuous arena. The arena consists of two small islands connected by a narrow bridge ('safe area') and surrounded by water. The position in the arena is defined by the activities of the two motor output neurons ($(x, y) = (x_1, x_2)$, with x_1 and x_2 two activities of the RNN).

State space The vector defined by the activities of the RNN, i.e., $s = (x_1, x_2, \ldots, x_N)$ with $N = 100$. The activities evolve through the dynamics

$$x_i(t+1) = x_i(t) + \delta t \left(-x_i(t) + \tanh \left(\sum_j J_{ij} x_j(t) + I_i(t) \right) \right) , \tag{5}$$

where $J_{ij} \sim \mathcal{N}(0, g^2/N)$ are the fixed internal weights of the RNN, $\delta t = 0.05$ is the time step and $I_i(t)$ are the currents injected by the external controller (agent). The injected currents are defined by the actions taken as

$$I_i(t) = \rho \sum_{k=1}^{M} K_{ik} a_k(t) , \tag{6}$$

where $K_{ik} \sim U(0, 1)$ are positive input weights sampled from a uniform distribution and $\rho = 5$. is a parameter that scales the strength of the currents. We employ a saturating transfer function ($\tanh(\cdot)$), which leads to chaotic dynamics when the internal recurrent connections are strong enough [29], making it even harder for the MOP agent to control the RNN's activities. Terminal states are encountered whenever the position defined by the motor output neurons is outside the 'safe area', i.e., when the agent falls into any of the degenerate locations defining the sea ($(x_1, x_2, \cdot) = (x_{sea}, y_{sea})$). Terminal states are hence dependent on the non-output neurons x_i for $i = 3, \ldots, N$ only via the dynamics defined in Eq. 5.

Action space When the agent is in a 'safe area', it can act on the RNN via an $M = 8$-dimensional vector of binary components $a = (a_1, \ldots, a_M)$ where $a_k = \{-1, 1\}$ for $k = 1, \ldots, M$. Whenever it hits a terminal state, action space irreversibly collapses into a space where the only available action is `nothing`, i.e., $a_k = 0$ for $k = 1, \ldots, M$.

The Value approximator Given the high-dimensional continuous state space ($N = 100$), we use a feed-forward network (FFN) to approximate the optimal value function $V^*(s)$ with $V(s, w)$. The FFN of parameters w consists of one hidden layer of N_{hid} neurons and one single output neuron, where $V(s, w)$ is the activity of the output neuron. The approximator receives as an input the activities x of the RNN. It can be proven that the optimal value function satisfies the consistency equation defined in Eq. 4 [21]. Analogously, we define the expected evolution of the approximated value function satisfying the Bellman consistency equation $V_{\mathcal{B}}(s, w)$ as

$$V_{\mathcal{B}}(s_t, w) = \alpha \ln \sum_{a_t \in \mathcal{A}(s_t)} \exp \left(\alpha^{-1} \beta \mathcal{H}(\mathcal{S}'|s_t, a_t) + \alpha^{-1} \gamma \sum_{s_{t+1}} p(s_{t+1}|s_t, a_t) V(s_{t+1}, w) \right) \tag{7}$$

Table 1: Hyperparameters for the continuous world, including the parameters of the value approximator (FFN).

Parameter	Value
RNN neurons N	100
action components	$\{-1, 1\}$
action dimensionality M	8
FFN hidden layers	1
FFN hidden units per layer	256
FFN input units	100
FFN nonlinearity	ReLU
training epochs	30
number of agents	5
batch size	10
optimizer	Adam
learning rate η	0.01

and note that, if the approximated value function $V(s, w)$ were equal to the optimal value $V^*(s)$, its expected evolution would coincide with the value function itself, i.e., $V_{\mathcal{B}}(\cdot) = V^*(\cdot)$ (see Eq. 4).

Therefore, in each epoch l ($l = 1, .., N_{ep} = 30$), parameters of the FFN are optimized by minimizing the summed squared error along trajectories between the approximated value and its evolution satisfying the Bellman consistency equation as

$$\mathcal{L}_l(w) = \frac{1}{N_{traj}} \sum_{\tau=1}^{N_{traj}} \frac{1}{t_{end}^{(\tau)}} \sum_{t=1}^{t_{end}^{(\tau)}} \Big(V(s_t = x^{(\tau)}(t), w) - V_{\mathcal{B}}(s_t = x^{(\tau)}(t), w_l) \Big)^2 \ , \tag{8}$$

where the squared error is accumulated over a batch of N_{traj} paths, each path τ characterized by its own visited state trajectory $x^{(\tau)} = \big(x^{(\tau)}(0), x^{(\tau)}(1), \ldots, x^{(\tau)}(t), \ldots\big)$ and its own lifetime $t_{end}^{(\tau)}$, defined as the minimum between the time when the network reaches a terminal state and the maximum episode duration t_{max}. Each path τ is generated by sampling actions from the policy in Eq. (3) with the same initial condition $x(0) = x$, that is, a random initialization of the RNN's activities. The policy depends on the specific values of the FFN's weights at epoch l. The parameters of the network are updated at each epoch l using Adam as optimizer.