

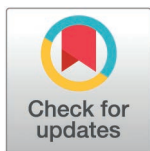
RESEARCH ARTICLE

Decoding behavior with minimal and interpretable agent models

Giorgio Nicoletti¹, Antonio Celani^{1,2*}

1 Quantitative Life Sciences section, The Abdus Salam International Center for Theoretical Physics (ICTP), Trieste, Italy, **2** Department of Oncology, Università degli Studi di Torino, Italy

* acelani@ictp.it



Abstract

Understanding how living organisms process sensory information from their surroundings and translate it into decisions is a fundamental problem across biological scales – from biochemical signalling in single-cells to neural computations in animal brains. In this work, we address this challenge by introducing a method to reconstruct general decision processes directly from behavioral observations alone. Our approach is applicable to any biological agent and does not require prior knowledge of its internal mechanisms or its environment. Our agent model is defined by a recurrent dynamics over a discrete set of internal states which encode and process sensory information, and dictate which actions to execute. We validate our method on synthetic agents and demonstrate that we can exactly recover the agent's behavior for non-trivial tasks. Then, we infer agent models from experimental data of rats performing evidence accumulation and of mice making decisions under uncertainty and in changing environments. In both cases, very few internal states suffice to reproduce the observed behavior with high accuracy. Crucially, the immediate interpretability of the inferred dynamics allows to understand the computational process underlying decision-making.

OPEN ACCESS

Citation: Nicoletti G, Celani A (2026) Decoding behavior with minimal and interpretable agent models. PLoS Comput Biol 22(8): e1014585. <https://doi.org/10.1371/journal.pcbi.1014585>

Editor: Barbara Webb, The University of Edinburgh, UNITED KINGDOM OF GREAT BRITAIN AND NORTHERN IRELAND

Received: February 17, 2026

Accepted: July 16, 2026

Published: August 7, 2026

Copyright: © 2026 Nicoletti, Celani. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data availability statement: The code to infer finite state controllers from behavioral trajectories is available at <https://github.com/giorgionicoletti/FSC-inference-MAPSO>.

Funding: The author(s) received no specific funding for this work.

Author summary

Living organisms constantly process information to make decisions. Which are the computational processes that translate sequences of sensory cues into courses of actions? Since the internal states of an organism typically cannot be observed, it is difficult to answer this question. Even when measurements are possible – such as in large-scale recordings of neural activity – extracting computational principles from a high-dimensional dynamics still poses significant challenges. Here, we introduce a method to infer interpretable models of the decision-making process directly from behavioral trajectories, without making assumptions about the environment. Despite their simplicity, these agent models are expressive enough to accurately describe experimentally observed behaviors and reveal the structure of the underlying neural computational processes.

Competing interests: The authors have declared that no competing interests exist.

Introduction

Decisions are everywhere. Viruses [1], bacteria [2,3], single-cell eukaryotes [4–6], rodents, and mammals [7,8] must continuously make decisions to survive and thrive. Yet, understanding how biological and living systems collect, process, and translate information into actions, whether biochemically or through neural processes, remains a fundamental challenge across fields. A wide range of modelling frameworks to capture biological behavior has been proposed, from Bayesian models to artificial neural networks. However, these approaches typically make strong assumptions about the decision-making process at hand or sacrifice interpretability for predictive accuracy. In this work, we develop a general framework that allows us to infer interpretable decision-making strategies directly from behavioral observations across diverse agents and tasks.

The theory of decision-making processes provides a general mathematical language to describe behavior as the repeated interaction between an *agent* and the *environment* that surrounds it. The boundary between agent and environment is the sensorimotor interface, where information is received and decisions are executed (see Fig 1). Now, suppose we have at our disposal a dataset of sequences of interactions between an agent and the environment. How can we infer from the data a “good” model of the decision process of the agent? In principle, an agent with a perfect memory could make decisions based upon the entire history of past interactions with the environment. In practice, living systems are constrained by finite computational resources. Thus, biological agents must compress such histories in their *internal states*, whose role is to encapsulate all the information that is available to the agent and that is relevant for the task at hand. The biological realization of the internal state can in general be a complex and very high-dimensional object, like the biochemical state of a cell [9], the activity patterns of neurons in a brain [10–12] or, in a more abstract setting, the state of an artificial recurrent neural network [13,14].

Here, we employ *Finite State Controllers* (FSCs) as minimal agent models defined by a discrete set of internal states [15–19]. Formally, FSCs represent a conceptual extension of input-output Hidden Markov Models and variants alike [20–23] with the key difference that, in an FSC, actions are naturally fed back into internal transitions, allowing internal states to maintain a memory of past decisions and use it to inform future ones. The main advantage of FSCs is twofold. First, the discrete structure of internal states can be interpreted in terms of behavioral modes, governed by internal processes whose dynamics are essentially low-dimensional. This is akin to the fact that the computations of RNNs [24–27] and networks of biological neurons [28–33] can be understood in terms of transitions between discrete metastable attractors, or even belief-like representations [34–37]. Describing an agent by an FSC bypasses the need for high-dimensional fast neural dynamics by directly expressing the decision-making process through a highly coarse-grained representation of the slower processes. Furthermore, recent efforts have shown that small RNNs are able to represent a variety of cognitive tasks, suggesting that compact models of behavioral strategies can achieve the performance of more complex models [38]. Second, if the focus is on *decoding* the decision process – i.e., understanding which

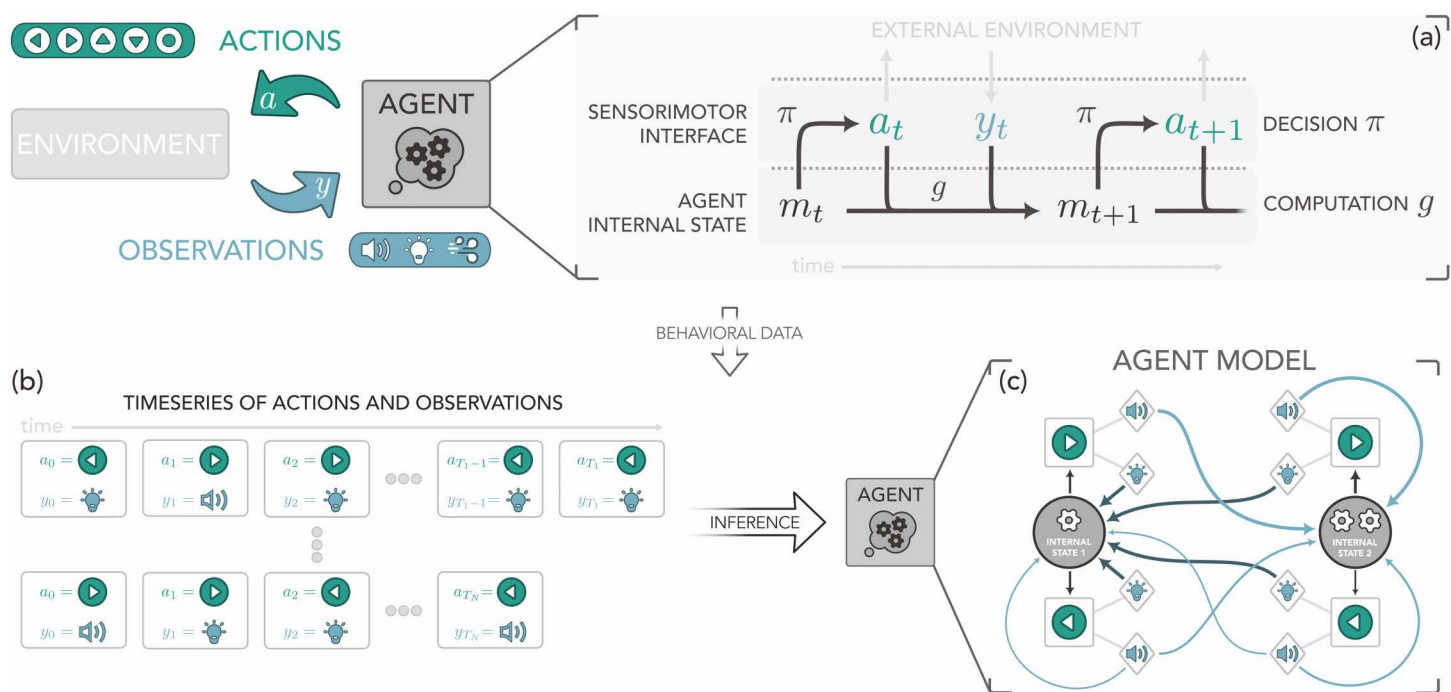


Fig 1. Inference of decision-making processes from behavioral trajectories. (a) We consider a generic agent that interacts with an environment via a sensorimotor interface, by taking elementary actions a (e.g., moving in space or selecting a choice) and receives observations y in response (e.g., sensory cues and rewards). The agent's behavior is driven by an agent state m , which determines its actions through a decision process described by a probability distribution π over the action space. At each step of the decision process, the internal state is updated to account for new observations through an internal computation process modeled by a probability distribution g over the space of internal states. (b) We assume that we only have access to a series of behavioral timeseries of the agent, corresponding to sequences of action and observation pairs (a_t, y_t) . (c) Our goal is to infer the internal model with which the agent makes decisions as a finite state controller (FSC), where the agent's state is replaced by a finite set of discrete internal states. Its decision process is represented by stochastic policy $\pi(a_t | m_t)$ determining the action a_t given the current internal state m_t , and an internal computation model $g(m_{t+1} | m_t, a_t, y_t)$ with which the agent process information by updating its internal state depending on the previous state, the action, and the observation. In the diagram, circular nodes represent internal states, square nodes actions, and diamond nodes observations. The thickness of the arrows between nodes is proportional to the corresponding transition probability, both for the agent policy $\pi(\cdot|m)$ (gray arrows) and the computation $g(\cdot|m, a, y)$ (colored arrows).

<https://doi.org/10.1371/journal.pcbi.1014585.g001>

computational processes are made by an agent – FSCs yield interpretable and tractable descriptions of decision strategies, making the computational processes at play explicit. These considerations lead us to ask: could an FSC with a relatively small number of internal states be a good model for the behavior of an agent engaged in a real task?

To answer this question, we develop a method to infer FSC models from sequences of interactions between the agent and the environment alone. No prior information about the environment is required, nor does the inference ever attempt to model how observations are produced – the focus is entirely on the agent. We first test our method against synthetic agents performing a variety of tasks, from arithmetic computations to spatial navigation, and show that we can efficiently discover FSCs that perfectly reproduce behavior even with a limited number of behavioral trajectories. Then, we exploit our method to infer FSCs from experimental data in rats performing an evidence accumulation task and in mice during a reversal task, where they must make decisions in uncertain and changing environments.

We find that inferred FSCs with few internal states not only reproduce observed behaviors with high accuracy, but also uncover the underlying computational processes driving behavior. Remarkably, in some cases, the inferred FSCs can be explicitly mapped into known decision architectures, such as exponentially smoothed counting or leaky competing

accumulator models. This demonstrates that our approach provides interpretable descriptions that directly connect behavioral observations to potential neural implementations and generate testable hypotheses.

Results

FSCs as minimal agent models

Let us describe the agent's decisions as *actions* $a \in \mathcal{A}$ – where a could be a movement in a given direction, an interaction with an object, or a measurement of some quantity – and *observations* $y \in \mathcal{Y}$ – for instance, sensory cues of different types. In general, the action a_t taken at time-step t by the agent may depend on the full history of past interactions with the environment, $(a_{0:t-1}, y_{0:t-1}) = (a_0, y_0, \dots, a_{t-1}, y_{t-1})$, through some probability distribution $P^{\text{agent}}(a_t | a_{0:t-1}, y_{0:t-1})$.

Rather than keeping a record of past experiences, a realistic agent encodes the history in its internal state $m_t \in \mathcal{M}$. The agent's behavior can then be described in terms of the probability $\pi(a_t | m_t)$ of taking an action a_t given its current internal state m_t , and of the recurrent computation process $g(m_{t+1} | m_t, a_t, y_t)$, which updates the internal state by processing the information that has just been acquired. Both the *policy* π and the *internal state update* g are conditional probability distributions, so that in general the process is stochastic. The description of the decision process is completed by a choice of the initial distribution of internal states $\rho(m_0)$. A *model of the agent* is thus defined as the tuple $(\mathcal{A}, \mathcal{M}, \mathcal{Y}, \rho, \pi, g)$, as we summarize in Fig 1A.

An FSC is an agent characterized by a discrete set of internal states $|\mathcal{M}| = M$. For the tasks considered in this work, $\mathcal{A}$ and $\mathcal{Y}$ are also discrete sets of cardinality $|\mathcal{A}| = A$ and $|\mathcal{Y}| = Y$. We stress here that an FSC implicitly encodes both past observations and actions in m_t through the recurrence of the internal computation g . In this sense, it generalizes the notion of belief about the state of the external environment that characterizes Bayesian agents [39]. Further, there is an exact mapping between FSCs and linearized gated recurrent units, a gating mechanism used in recurrent neural networks (RNNs) [40,41] (see Section D in S1 Appendix). This correspondence provides a bridge between the abstract representation of an FSC and the more biologically interpretable variables of an RNN, and generalizes input-output Hidden Markov Models [20–23] to allow internal states to retain information about past actions and use it to guide future decisions. Finally, we note that FSCs can always be made expressive enough to reproduce the behavior of an arbitrary agent over a time span T , albeit at the punishing price of an exponentially large number of internal states $M \sim (A \times Y)^T$. In this work, however, we focus on FSCs with relatively small M , following the idea that internal states should represent how the agent compresses available information, rather than performing a complex act of mimicry.

Inference of FSCs from behavioral trajectories

We assume that we have $k = 1, \dots, N$ behavioral trajectories, i.e., histories of interactions between agent and environment, each described by a sequence of actions and observations $(a_0^{(k)}, y_0^{(k)}, \dots, a_{T_k}^{(k)}, y_{T_k}^{(k)})$ (see Fig 1B). Our goal is to infer a finite state controller that, at a given observation sequence, will reproduce the actions as closely as possible (Fig 1C). To this end, we seek FSC parameters that maximize the log-likelihood of the FSC actions given the observations, or equivalently solve the optimization problem

$$\underset{\text{FSC}}{\operatorname{argmin}} \left(-\frac{1}{N} \sum_{k=1}^N \log P^{\text{FSC}}_{a_{0:T_k}^{(k)} | y_{0:T_k}^{(k)}} \right) \quad (1)$$

where

$$P^{\text{FSC}}_{a_{0:T} | y_{0:T}} = \sum_{m_0} \rho(m_0) \prod_{t=0}^T \pi(a_t | m_t) g(m_{t+1} | m_t, a_t, y_t). \quad (2)$$

Above, the policy π , the internal computation g , and the initial distribution of the internal states ρ , which fully characterize the agent, are parametrized by softmax functions (see Materials and Methods). The optimization problem in Eq. (1) does not require a model of how observations are generated by the environment. This is a consequence of observations appearing only as conditional dependencies in Eq. (2). Thus, the environment may be arbitrarily complex, and observations may depend on the current action as well as on the entire history of past interactions.

The maximization of the likelihood in (1) is, in general, a non-convex optimization problem. Thus, in order to infer the parameters that define the FSC, we developed a global optimization method, MAPSO, whose core idea is that multiple FSCs are initialized with random parameters and then evolved according to an interacting swarm dynamics. MAPSO is based on Adaptive Particle Swarm Optimization [42], where particles are FSCs, but the topology of the interactions changes with the metric distance during the optimization. After a fixed number of iterations, the FSC that achieved the largest log-likelihood at any point during training is retained (see Materials and Methods and Section E in S1 Appendix for more details). We remark here that we do not add any regularization terms to the likelihood, such as sparsity constraints, although this is possible in principle. In this work, we infer FSCs with different numbers of internal states M and analyze them separately, and find that, in all cases considered here, there exists a value of M above which FSCs start to accurately reproduce relevant behavioral observables. We note that the number of parameters of an FSC is dominated by the internal computation, growing as M^2AY (see Methods). Therefore, in finite datasets, this effectively limits the number of states that can be reliably inferred, especially in scenarios where A and Y are large.

Finally, we note that optimization can also be performed by a local search algorithm, e.g., gradient methods or an adapted Baum-Welch expectation maximization algorithm, augmented with random restarts. While we make no claim of computational supremacy, in our experience, MAPSO turns out to be more robust and efficient in locating global optima, at least for the cases discussed below. In Section F of the S1 Appendix, we show an explicit comparison with expectation maximization and stochastic gradient descent, showing that in some cases MAPSO outperforms them both.

Exact inference of an arithmetic agent

We first test our inference procedure on a purely synthetic agent performing an arithmetic task. The agent can take two actions, “A” or “B”. If the agent takes A, the environment returns a random digit between 0 and 4 with equal probability. Conversely, if it takes the B action, the observation is a digit between 6 and 9. The agent has a counter that is initially set to zero. At the first step, the agent takes a random action with equal probability, observes a digit, and adds it to the counter. At all subsequent times, the agent performs a parity check on the counter and takes action A if it is even and B if it is odd. Then, it receives a new digit and adds it to the counter (Fig 2A). We want to know if our inference algorithm can retrieve the decision-making process just from a dataset of sequences of actions and observations without any information about the inner workings of the environment and of the agent.

It would be natural to identify the internal states of this agent with the counter’s value, but their number would be very large if sequences are long. Are there more compact descriptions of the agent’s decision-making process? In fact, our inference algorithm discovers an exact representation of the agent’s behavior with just $M=3$ internal states, valid for any sequence length.

In Fig 2B, we plot the negative log-likelihood during training with MAPSO from $N=50$ behavioral trajectories of the agent of length $T=100$. We find that the inferred FSC reaches the negative log-likelihood expected from the agent, that is the conditional entropy of the empirical distribution of actions conditioned to observations, which in this case reduces to the entropy of the initial action distribution (see Section A in S1 Appendix). In Fig 2C, we draw the inferred FSC topology. The first state, , is the initial one where the action is taken at random. The other two internal states –  and  – are dedicated to taking the A and B action, respectively, and switch between them if an odd digit is observed. It is easy to check that this FSC exactly implements the agent’s decision-making rules without the need for a counter.

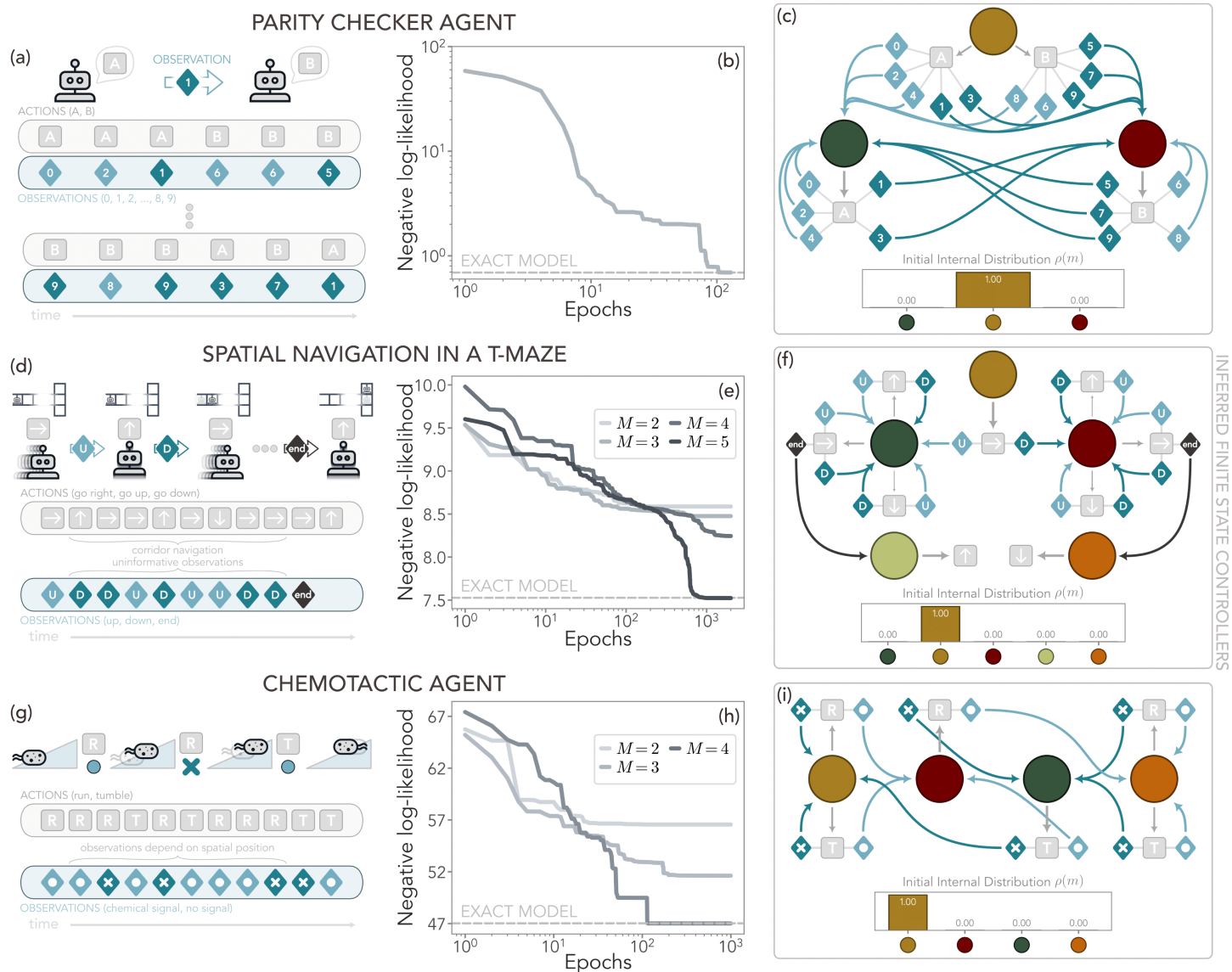


Fig 2. Exact inference of synthetic agents. (a) An agent can take two generic actions, “A” and “B”, and observe a digit. The agent adds the received digits to an internal counter that determines the following action according to a parity check (action A if it is even, B if it is odd). The first action is random. (b) Negative log-likelihood of an FSC with $M=3$ internal states during training with $N=50$ trajectories of $T=100$ steps each. The inferred FSC perfectly reconstructs the agent’s behavior, reaching the expected negative log-likelihood (dashed gray line). (c) Inferred FSC. The $\bullet$ state is the initial one, where the action is taken at random ($\pi(A|\bullet) \approx 0.48$, reflecting the empirical distribution). The $\bullet$ state is dedicated to the action A, whereas the $\bullet$ one always takes B. The transitions between the states are always deterministic and solely depend on the parity of the observation, as expected. (d) An agent performing a spatial navigation task in a T-maze 7 tiles in length. The agent enters the maze with a $\rightarrow$ action and receives an “up” (U) or “down” (D) observation with equal probability. It then either keeps moving to the right (60% probability) or steps up ($\uparrow$) or down ($\downarrow$) with a slight bias that depends on the first observation (30% and 10% probability, respectively). Once it reaches the T-junction, the agent chooses $\uparrow$ or $\downarrow$ solely based on the first observation, ignoring all the ones it received in the corridor. (e) Negative log-likelihood of FSCs with different M during training with $N=1000$ trajectories. With $M=5$, the inferred FSC reaches the expected negative log-likelihood of the exact model (dashed gray line). (f) The $\bullet$ state represents the initial condition, which transitions to one of two “corridor states”, either $\bullet$ (first up observation) or $\bullet$ (down observation). Once it reaches the end of the corridor, the transition $\bullet \rightarrow \bullet$ allows the FSC to correctly take $\uparrow$, while $\bullet \rightarrow \bullet$ results in $\downarrow$. (g) A chemotactic agent in a one-dimensional concentration gradient. At each time, the agent may detect the presence of a chemical (observation $\bullet$) or not detect it ($\times$), with a probability that depends on its concentration, which in turn depends on the position in space. If in the last two steps it received the same observation, the action is chosen at random between tumble (T), which reorients the agent without any displacement, and run (R), which moves the agent in its current direction. If the last observation was $\bullet$ (a detection) and the previous one was $\times$ (no detection), then the agent runs. In the reversed case, it tumbles. (h) Negative log-likelihood of FSCs with

different M during training with $N=250$ trajectories of $T=100$ steps each. With $M=4$, the FSC reaches the expected likelihood. (i) The internal states of the FSC encode for the four combinations of past observations: $\bullet \sim (y_{t-2}, y_{t-1}) = (\times, \times)$; $\bullet \sim (\times, \bullet)$; $\bullet \sim (\bullet, \times)$; $\bullet \sim (\bullet, \bullet)$ and works as a shift-register memory.

<https://doi.org/10.1371/journal.pcbi.1014585.g002>

Inference of a spatial navigation task

We now consider the case of a synthetic agent performing a spatial navigation task (see, e.g., [43]) that requires the exact memorization of some observations, selectively ignoring other ones, and making associations between long-past and just-received observations (Fig 2D).

The agent is stepping on a tiled T-maze and can take three actions: if it takes action $\rightarrow$, it makes a step to the right, whereas actions $\uparrow, \downarrow$ leave it at the same location (it bounces on the corridor walls). At the beginning of the task, the agent enters the maze from the left, always taking the $\rightarrow$ action, and receives an observation that can be either “up” or “down” with equal probability. At subsequent steps, the agent either chooses action $\rightarrow$ most of the time or takes $\uparrow$ or $\downarrow$ with a slight bias towards the action that matches the observation received at the entrance step. As it travels along the corridor, the agent keeps receiving observations “up” or “down” with equal probability until it reaches the T-junction and receives the observation “end”. At this time, it deterministically takes $\uparrow$ if the initial observation was “up” and $\downarrow$ if it was “down”. All other observations made along the corridor do not matter – they are just noise.

In Fig 2E, we show the negative log-likelihood of the inferred FSCs with a different number M of internal states. With $M=5$, the FSC perfectly reconstructs the agent’s behavior (Fig 2f), whereas with fewer internal states the controller fails to memorize and/or make the required associations. The node $\bullet$ represents the “blank slate” internal state of the agent as it enters the maze. Then, the FSC branches to one of two “corridor states” depending on the initial observation, and there it remains until the junction is reached. $\bullet$ remembers the first “up” observation, so that the corresponding policy has a slight bias in the up direction. After the “end” observation, the FSC switches from $\bullet$ to $\bullet$, where the up action is taken deterministically. The other states, $\bullet$ and $\bullet$, represent instead the first “down” observation, allowing the FSC to perfectly match the agent’s behavior.

Inference of a chemotactic agent

Finally, we consider a synthetic chemotactic agent that moves in a one-dimensional concentration gradient. At each time, the agent may detect the presence of a chemical ($y_t = \bullet$) or not ($y_t = \times$) with a probability that depends on its position, and can either decide to run – keep on moving in its current direction, $a_t = R$ – or tumble – randomly reorient without moving, $a_t = T$. The agent chooses the next action based on the last two observations according to the following policy. Assigning value $y=1$ to the detection ($\bullet$) and $y=0$ to no detection ($\times$), the agent computes the difference between the last two observations $y_{t-1} - y_{t-2}$. If the difference is positive, then the agent chooses to run; if it is negative it tumbles; and finally, if it is zero, it chooses at random with equal probability (see Fig 2G). At the beginning of the task, we assume that the agent has not detected a chemical in the last two time-steps. We remark that for this agent the histories of actions and observations are fully entangled: the probability of receiving an observation depends on the position, which in turn depends on the entire past history of actions, each of which depends on previous observations, and so on and so forth.

In Fig 2H, we show that the negative log-likelihood of the inferred FSCs for $M=4$ reaches the value expected for the chosen agent. The corresponding FSC is shown in Fig 2G. Inspecting the structure of the transitions between internal states, one realizes that the nodes exactly encode the ordered pair of last observations and the FSC works as a shift-register memory: new observations enter to the right of the register while older observations are pushed to the left and eventually out of it.

We note in passing that if the internal computation matrices are low-rank, then the likelihood is invariant under some linear mixing of the internal states (which is not just a trivial permutation), leading to multiple FSCs with exactly the same

behavior. This lack of identifiability indeed takes place for the chemotactic agent and is discussed in detail in Section D of the [S1 Appendix](#). We stress that this is not a limitation of FSCs, but a general property of any behavioral model, as it signals a genuine symmetry of the task. Any equivalent FSC provides an equally valid description of the agent's strategy.

Evidence accumulation in rats

We now consider an experimental paradigm where rats are trained to perform evidence accumulation during an auditory discrimination task (data from Ref. [44]). During each trial, rats initially insert their nose into a central point for a fixed amount of time (1.5s, signalled by a light-emitting diode). During this time, two trains of randomly timed auditory clicks are played simultaneously, both to the left and to the right of the rat, with different click rates and total durations. At the end of the fixation period, the rat decides to poke either to the left or to the right, and receives a water reward if it turns towards the side where more clicks were played ([Fig 3A](#)). The sum of the left and right click rates is fixed at 40Hz, with subsequent trials varying in difficulty depending on the ratio of the click rates.

From the behavioral trajectories, we can discretize time (we set $\Delta t = 1/40$ s) and define four different observations and three actions. At each timestep, as long as the light is on, the agent may only take the action “listen” and receive a “left” ($y = -1$) or “right” ($y = 1$) observation if it hears a click on the corresponding side, or a “silence” observation ($y = 0$) if it hears neither. When the light-emitting diode turns off, the agent receives an “end” observation and can take one of the two actions “go left” or “go right”.

Among the data in [44], we extracted the behavioral data of three rats, each across two different days. To average out noise in the rats' behavior and accurately sample different task difficulties, we retained approximately 4500 trials, with 50% used for inferring the FSC parameters and 50% to validate their performance. In [Fig 3B](#), we show the negative log-likelihood of the best-performing FSCs for different numbers of internal states M . Remarkably, we find a sharp decrease at $M=4$, after which the performance quickly plateaus. Indeed, the FSCs with $M \geq 4$ closely reproduce the observed rats' performance ([Fig 3C](#)) with slight deviations visible only for $M=4$. This signals that an FSC with a relatively small number of internal states is expressive enough to reproduce the behavioral trajectories.

A small FSC recapitulates the essential features of decision-making

With $M=2$ and $M=3$, the inferred FSCs are trivial, as their limited internal structure can only reproduce the fixation period during which rats listen until the “end” cue (see [Fig 3d](#) and Section D in [S1 Appendix](#)). For example, with $M=2$, no computation can be done in the first internal state because “left” and “right” observations always circle back to this state. When “end” is observed, the FSC transitions to the other state and can only choose “go right” or “go left” at random, regardless of the past history. This model can only reproduce the average final decision across all data, and therefore cannot accumulate past evidence. Inferred FSCs with $M=3$ nodes similarly fall short.

The structure of the interactions for this task hints at a layered organization of the internal states: a computation layer where the agent can only take the “listen” action, and a decision layer responsible for taking the “go” actions and accessible only upon receiving the “end” signal. For computation to be possible, at least two nodes in the upper layer are required, whereas two are both necessary and sufficient for the final decision. We therefore focused on FSC architectures with $M-2$ computation layer nodes and 2 decision layer nodes, imposing this structure a priori for inference efficiency. A posteriori, we then check that these structures are indeed those that minimize the negative log-likelihood. In Section C of the [S1 Appendix](#), we also show that the same FSCs can be recovered using a curriculum learning strategy, which allows the inference procedure to discover the two-layer structure directly from data.

Let us first focus on the simplest case, $M=4$, shown in [Fig 3E](#). We reiterate that, with the layered structure induced by the random interruption, only the two internal states of the top layer are used for internal computation. Despite the small number of states, this FSC is able to achieve a high accuracy in reconstructing behavior, with the  state encoding for left clicks, and  for right ones. The simplicity of the model allows to compute analytically the predicted probability

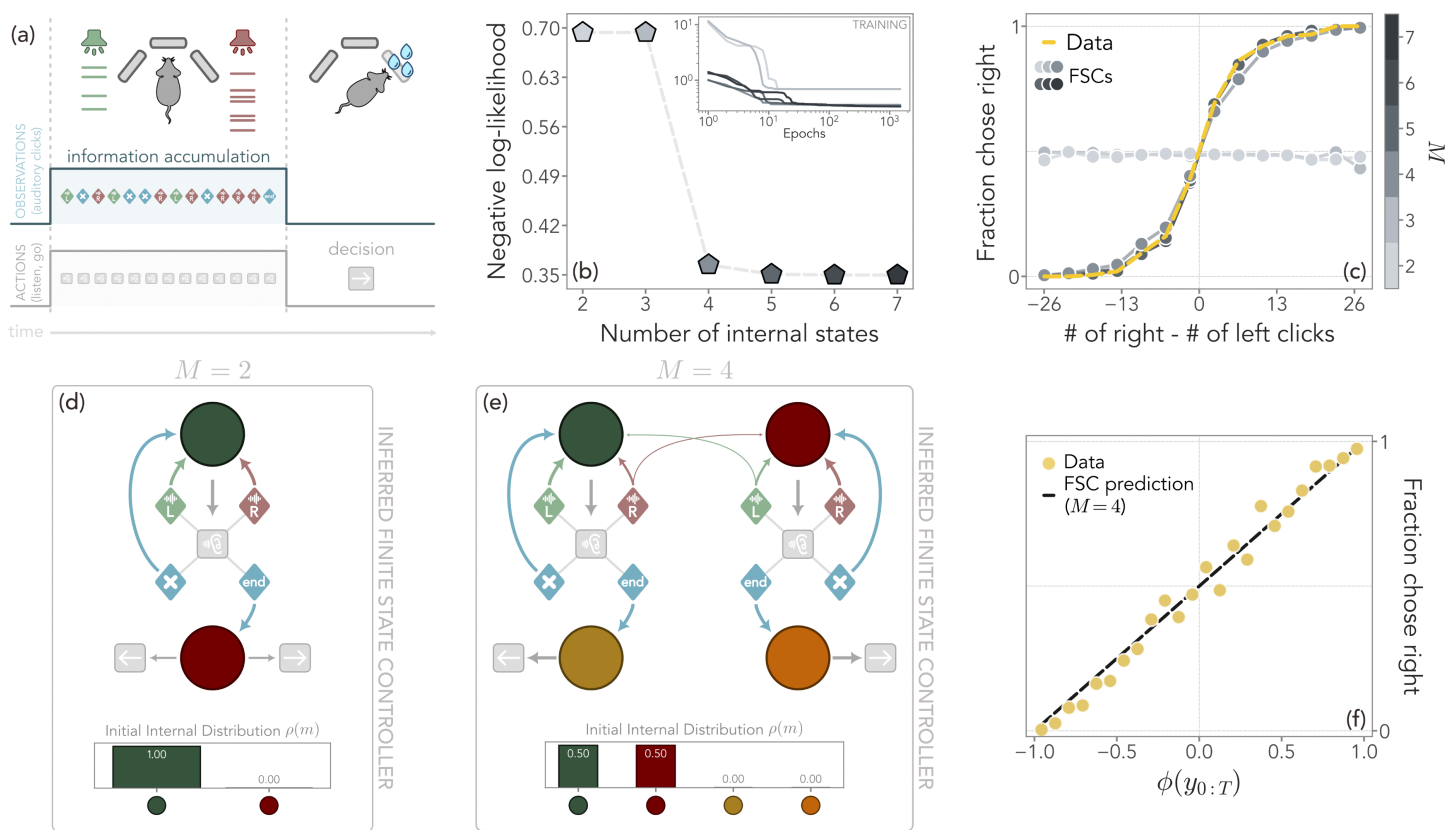


Fig 3. Evidence accumulation in rats during perceptual decision making. (a) Sketch of the experiment. During each trial, rats fixate their nose into a central port for 1.5s. During this time, corresponding to the action "listen", trains of randomly timed auditory clicks are played both to the left and to the right, with different frequencies and durations (between 0.2s and 1s) which represent three observations: left, right, or silence ("x" in the figure). After nose fixation ends, the rat receives an "end" observation, and pokes in one of the side ports (actions $\leftarrow$ or $\rightarrow$). If it turns to the side where more clicks have been played, it receives a water reward. Data of approximately 4500 trials (half for training, half for validation) from [44]. (b) Negative log-likelihood over the validation set of the inferred FSCs with different numbers of internal states. For $M \geq 4$, a two-layer structure, which reproduces the structure of the experiments, was imposed a priori to speed up convergence. Inset: negative log-likelihood over the training set during inference. (c) At $M \geq 4$, FSCs (grayscale dots) accurately reproduce rats' performance (dashed sand line, averaged across trials and individuals) in choosing right as a function of the difference between the number of clicks. A further improvement can be seen for $M=5$ onward. (d) Inferred FSC with $M=2$ internal states. The FSC learns to take a $\leftarrow$ or $\rightarrow$ action only after the "end" observation, but cannot perform any computation due to the lack of internal states, and thus must choose at random. (e) Inferred FSC with $M=4$ internal states and a fixed two-layer structure. The two internal states on the top (●, ●) are responsible for taking the "listen" action, acting as a computation layer where the agent accumulates evidence. The two at the bottom (●, ●) decide whether to go left or right. The state ● remains unchanged if a click is heard on the left or no click is heard, whereas it switches to ● with a probability $\epsilon \approx 0.2$. If "end" is heard, the trial ends and ● switches with probability one to ●, where the agent chooses to go to the left. The situation is reversed for the states ● and ●, the latter of which goes to the right. (f) It can be shown that this FSC is computing an exponential smoothing of the difference between right and left observations ϕ with a smoothing factor ϵ (Eq. (3)). Indeed, the fraction of times that the rats choose to go right is well-predicted by its value at the end of the trial. Since ϕ is captured by the internal state ●, the FSC prediction simply corresponds to the bisector of the plane.

<https://doi.org/10.1371/journal.pcbi.1014585.g003>

of choosing an action at the end of the trial, $\phi(T+1) = p(A_{T+1} = \rightarrow | y_{0:T}) - p(A_{T+1} = \leftarrow | y_{0:T})$, which obeys the recursion relation

$$\phi(t+1) = \begin{cases} \phi(t) & \text{if } y_t = 0 \\ (1-\epsilon)\phi(t) + \epsilon y_t & \text{if } y_t = \pm 1 \end{cases} \quad (3)$$

with $\phi(0) = 0$. Note that $\phi(t)$ is an exponential smoothing of the difference between right and left observations, with smoothing factor ϵ . Here, $\epsilon \approx 0.21$ is the probability of switching between the two upper memories (see Section D in S1 Appendix). In Fig 3F, we show that indeed the fraction of trials in which the rats choose to go right is well predicted by the value of ϕ at the end of the trial. Thus, the simplest nontrivial FSC with $M=4$ interprets the decision-making process as the following algorithmic procedure: compute the exponentially recency-weighted difference of right and left clicks and, at the end of the trial, bias the choice proportionally to this quantity.

Additional internal states allow longer memory of past events

We now seek to understand how the FSC with $M=5$ improves upon the previous one, allowing it to reproduce the rats' behavior with greater accuracy and to saturate the likelihood (see Figs 3B and 3C).

In Fig 4A, we draw the inferred FSC with five internal states. Similarly to the previous case, we find two nodes in the computation layer (● and ●) that encode the prevalence of left and right clicks, respectively. Once the trial ends, they deterministically transition to the corresponding state in the decision layer (● and ●) where the agent goes to the left and to the right, respectively. However, the transitions between ● and ● are now mediated by an intermediate node ●. If the "end" observation is received while the FSC is in this state, it transitions to the decision layer with a slight left bias, reflecting an overall observed preference by the rats. Importantly, the ● state is also the one where the FSC is initialized.

In Fig 4B, we plot the probability of occupying one of the computation states (●, ●, ●) right before the trial ends as a function of the difference between the number of clicks heard on the right and on the left. When most observations are from the right, the FSC activates the ● state, and vice versa for the ● state. Crucially, the occupation of the middle ● state peaks when the trial ends in a region of uncertainty, where the difference between clicks is close to zero.

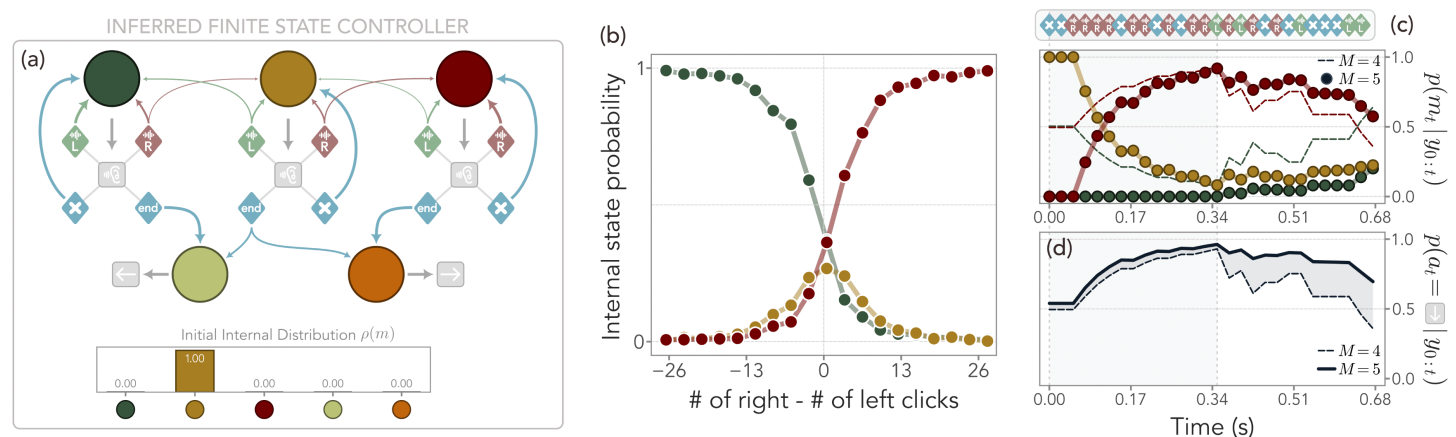


Fig 4. Inferred FSC with a two-layer structure and $M=5$ internal states for evidence accumulation. (a) Similarly to the case of $M=4$, the leftmost (●) and rightmost (●) internal states in the computation layer are responsible for encoding the prevalence of left and right clicks, respectively, and go in the corresponding direction once the trial ends (respectively ● and ●). Importantly, the switch between them is regulated by the ● internal state, which transitions to the left or the right when the corresponding click is heard with a probability $\eta \approx 0.29$. The FSC is initialized at this internal state. If the trial ends in the ● state, it transitions to either ● or ● approximately at random (0.56 and 0.44, respectively). (b) Probability of occupying one of the upper internal states (●, ●, ●) right before the trial ends, averaged over the observation sequences in the data, as a function of the difference between right and left clicks. The central state is mostly active in an uncertainty region where the difference in the number of clicks is close to zero. (c-d) For a fixed observation sequence $y_{0:t}$ the probability of occupying a given internal state $p(m_t|y_{0:t})$ may be different between the FSC with $M=5$ (colored dots and solid lines) and with $M=4$ (dashed lines). In particular, when a set of congruent observations is followed by incongruent ones (e.g., a series of "right" observations first and then a series of "left" ones), the central state acts as a buffer and maintains the occupancy of the right state at higher levels. This reflects in a slower and delayed decay of the probability of taking the action "go right" if the trial were to end at time t (bottom panel).

<https://doi.org/10.1371/journal.pcbi.1014585.g004>

This intermediate node plays an important role. In Fig 4C, we consider a fixed observation sequence $y_{0:T}$ from the data, characterized first by a series of “right” observations followed by noisy ones where the rat mostly hears “left”. We compute the probability of occupying a given internal state at time $t \in [0, T]$, $\hat{p}(m_t|y_{0:t})$, and compare it with the $M=4$ case. We find that, after an initial transient, both FSCs reach a high level of occupancy of , since only right clicks were heard. However, when the incongruent observations start to appear, the  state acts as a buffer and delays the transition to , which is instead quickly occupied when $M=4$.

We can compute the predicted probability of choosing an action at the end of the trial for this FSC as well (see Section D in S1 Appendix). Remarkably, the calculations reveal that the rats’ decision is captured by a leaky competing accumulator model (LCA) [46] with lateral inhibition. Namely, the preference for choosing right is proportional to the difference of two accumulators $\phi^\pm(t)$ – favoring state  and , respectively – at the final time. These obey the equations

$$\phi^+(t+1) = \begin{cases} (1-\xi)\phi^+(t) - \eta\phi^-(t) + \xi^2 & \text{if } y_t = 1 \\ (1-\eta)\phi^+(t) & \text{if } y_t = -1 \end{cases} \quad (4)$$

and

$$\phi^-(t+1) = \begin{cases} (1-\eta)\phi^-(t) & \text{if } y_t = 1 \\ (1-\xi)\phi^-(t) - \eta\phi^+(t) + \xi^2 & \text{if } y_t = -1 \end{cases} \quad (5)$$

with $\eta \approx 0.27$ the probability of exiting the central state , and $\xi \approx 0.13$ the probability of exiting one of the extremal states ( and ). Thus, the passive accumulator – the one that contrasts with the current evidence y_t – inhibits the active one. Due to this lateral inhibition, the probability of taking the right action if the trial were stopped at time t decays more slowly, allowing the FSC with $M=5$ to reproduce the rats’ behavior more accurately and helping identify the side on which more clicks were played, as shown in Fig 4D. Notice that the fact that $M=5$ provides a better model for the rats’ decision-making process is also compatible with previous results showing that individual rats exhibit diverse temporal weighting, represented by the ϵ and η parameters in the inferred FSC.

Mice decision-making in a reversal task

We now consider a different experiment, where mice are trained to perform a reversal task and make decisions in a changing and uncertain environment (data from [45]). At each step, a mouse can choose from two side ports. One of the two ports has a high probability, e.g., 0.8, of delivering a water reward (the high port), whereas the other has a probability 0.2 of rewarding the mice (the low port). After the completion of each trial, the high and the low ports have a probability 0.02 of switching. This results in blocks of consecutive trials, of variable length, where the environment is predictable, even if uncertain. When the environment switches, however, the mouse has to swiftly adapt its strategy to the change (Fig 5A).

From the behavioral trajectories, we can immediately identify two possible actions – “choose left” or “choose right” – and two observations – “water” and “no water”. Importantly, and differently from the previous case, the observations received by the agent are now explicitly dependent on the action it has taken. Furthermore, since any observation can follow from an action, we do not impose any a priori structure on the FSC. We consider the 159 trials reported in [45], with 80% of them used for inference and 20% for validation. In Fig 5B, we plot the negative log-likelihood of the inferred FSC for different numbers of internal states. Once more, we find that the likelihood plateaus already at relatively small values of M . However, at variance with the previous section, we find pronounced improvements when moving from $M=2$ to $M=4$. In Section B of the S1 Appendix, we explicitly compare the inferred FSCs with different types of hidden Markov models. We find that, while some HMMs can infer the immediate reward-to-action mapping, they cannot reproduce the mice’s commitment due to their inability to encode past actions in their internal states.

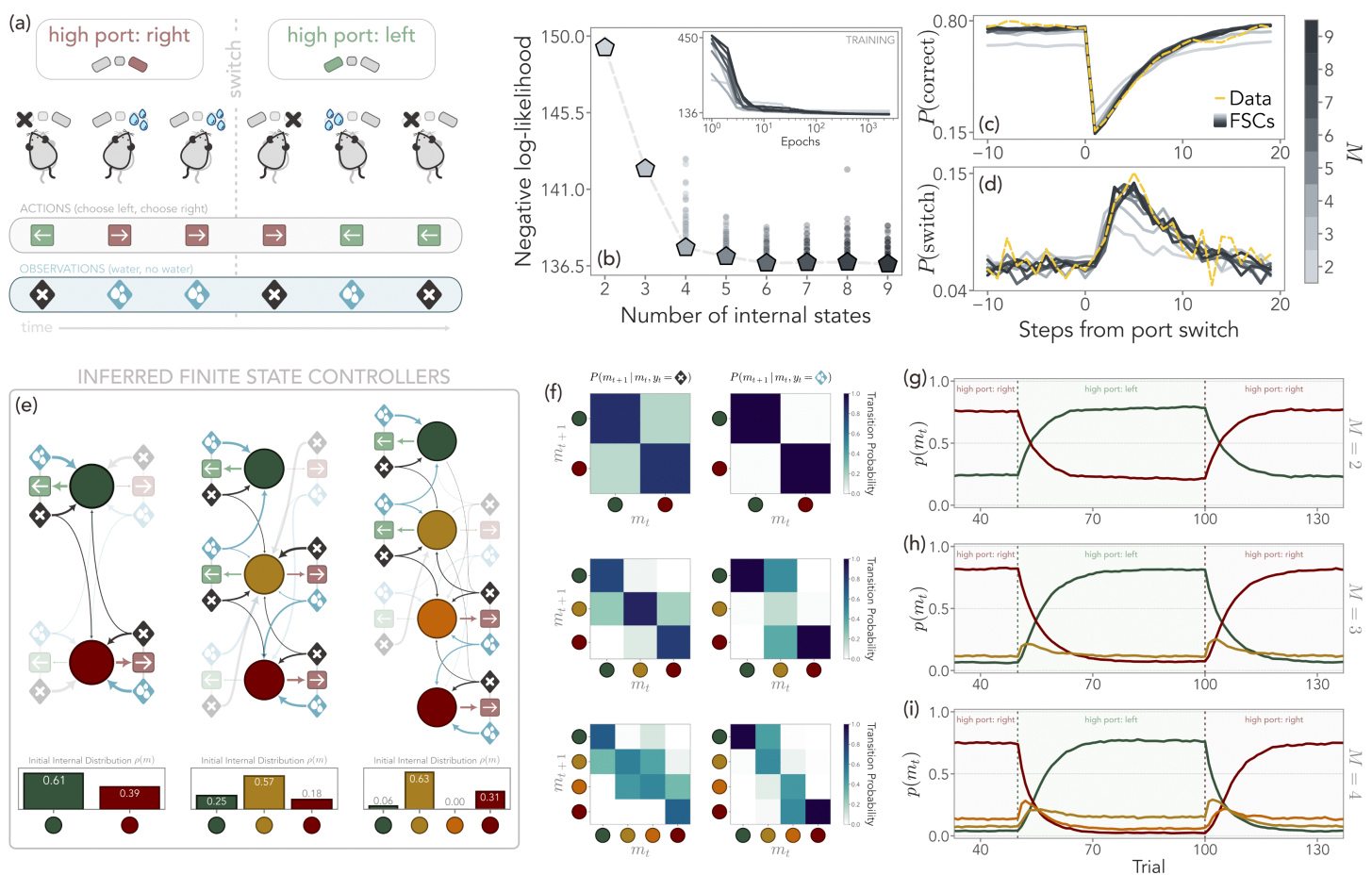


Fig 5. Probabilistic decision-making strategies in mice. (a) Sketch of the experiment. Mice are trained to select one of two side ports, representing two possible actions. Each port has a different probability of giving a water reward, either 80% (the high port) or 20% (the low port), corresponding to the two possible observations. At each time, the high port may randomly switch with a 2% probability, so that mice have to adjust their choices accordingly. Data of 159 behavioral trajectories (126 for training, and 33 for validation) from [45]. (b) Negative log-likelihood over the validation set of the inferred FSCs with different numbers of internal states. Pentagons represent the FSCs with minimum loss across restarts of the inference procedure, and selected restarts are represented by dots. Inset: evolution of the negative log-likelihood over the training set during inference. (c-d) The FSCs reproduce both the probability of selecting the higher reward port ($P(\text{correct})$) and the probability of switching port selection when the high port changes ($P(\text{switch})$), with an accuracy that increases with the number of internal states M and saturates from $M=4$. (e) Inferred FSC with $M=2$, $M=3$, and $M=4$. From an internal state, green arrows represent the probability of taking the action left, and red arrows the probability of choosing right. Transitions between the states depend on the observation, either when a reward is received (blue) or not (black). For ease of visualization, actions with low probability and the corresponding state transitions have a higher transparency. In all cases, we can identify two extremal states (● and ●) primarily encoding the left or right action, respectively. For $M>2$, the central states (● and ●) buffer the transitions between the extremal ones, similarly to Fig 4. (f) Marginalizing over the actions emphasizes the linear structure of the FSC. For $M>2$, the FSCs have a high probability of transitioning to the central states in the absence of a reward, whereas they drift towards the extremal states when a reward is obtained. (g-i) Probability of occupying the internal states as a function of time and for a specific sequence of high ports (right, left, right), averaged over 10^4 trajectories. With $M=3$ and $M=4$, the central states quickly activate when the port changes, playing the role of switch detectors. This allows the FSC to mimic the mice's strategy and switch to the high-reward action.

<https://doi.org/10.1371/journal.pcbi.1014585.g005>

Simple FSCs successfully capture the switching behavior

To quantitatively assess the ability of the FSC in capturing the decision process of the mice, we compute the probability of selecting the high reward port $P(\text{correct})$ and the probability of switching port selection when the high port changes $P(\text{switch})$, both as a function of the time from a switch [45]. We find that the inferred FSCs with $M=4$ or above are able to

reproduce the data very accurately. Remarkably, even for $M=2$ and $M=3$, the FSCs are still able to capture the timescales at which mice switch their behavior on average.

In Fig 5E, we draw the structure of the inferred FSCs with $M=2, 3, 4$. In all cases, we find two extremal states (● and ●) where the probability of choosing left and right, respectively, is significantly higher. This suggests that these states encode the agent's confidence that the corresponding side is the one with a higher chance of delivering the reward. For $M=3$, a central state ● emerges, from which the action is taken almost at random. Similarly to the previous example, this state buffers the transition between the extremal ones, delaying the commitment to a particular action. For $M=4$, this effect is ascribed to two separate central states (●, ●) where the policy still allows taking both actions, but is skewed towards the left and right ones, respectively. By marginalizing over the actions, the transitions between the internal states at fixed observation emphasize that all FSCs have a linear structure (Fig 5F). That is, the inferred internal computation g drives the agent towards the extremal states when a reward is obtained, whereas it transitions to the central ones in the absence of a reward. This linear structure is maintained even for larger M , suggesting that it can consistently model the internal decision process of the mice. The diminishing return in terms of increased accuracy from $M=4$ onward is further highlighted by the fact that at large M some inferred internal states become disconnected from the rest, signaling that they are not needed to reproduce the observed behavior (see Section C in S1 Appendix).

Emergence of internal switch detectors accurately reproduces the decision-making process

We now focus on the mechanisms that enable FSCs with larger M to better represent mice behavior. To this end, we generate behavioral trajectories from the FSCs for a fixed sequence of high ports (right, left, right) and track the probability of occupying a given internal state (Figs 5G, 5H and 5I). For $M=2$, the FSC is able to correctly follow the sequence of environments, but the probability of occupying the state corresponding to the worst action remains high (e.g., ● when the high port is the right one). This results in a lower probability of taking the best action, as also shown in Fig 5C.

For $M=3$, accuracy substantially increases thanks to the middle state ●. In Fig 5H we show that the probability $p(m_t = ●)$ peaks right after the high port changes. Hence, ● acts as a switch detector, quickly activating when the number of rewards decreases more than expected in the previous state. The FSC with $M=4$ is able to further improve its model of behavior thanks to the emergence of two dedicated switch detectors. As we show in Fig 5I, the ● activates when the high port changes from the right to the left, whereas ● is responsible for detecting the left-to-right switch. The policies $\pi(a|●)$ and $\pi(a|●)$ are still dominated by the right and left action, respectively, but are stochastic, allowing these states to probe the opposite action. The presence of these two switch detectors allows the accurate reconstruction of the mice behavior.

Discussion

The inference of decision-making processes hinges upon the availability of data about the interaction between agent and environment in the form of histories of actions and observations. In the cases discussed above, the identification of actions and observations is straightforward, but more complex situations command more attention. Several technical tools are now available to preprocess raw behavioral data and isolate suitable candidates for actions and observations [47–50]. In general, the key step resides in the choice of the level of description that the agent model wants to address. For instance, in a human navigation task, there is a coexistence of low-level actions and observations – like taking a step in some direction and noticing the presence of obstacles – and high-level actions and observations – like moving from one room to another and reading direction signs. Further, here we focused on the setting where observations y_t are received by the agent after actions a_t , representing agents that actively interact with the environment. In this case, actions can serve a dual purpose: achieving a goal – such as moving towards a target or obtaining food – and gathering information – for instance, revealing occluded objects or exploring the environment. Several, if not most, sensory processes can be described at a scale where they belong to this category [51,52]. Nevertheless, other choices are possible (see Section B in S1 Appendix), and our framework can be readily applied to them as well.

We stress that we did not make any attempt at modeling the environment, nor did we require any prior information about it. However, in the event that some knowledge about the nature of the task is available beforehand, it can be used to appropriately design the agent model. In all these cases, one can always check a posteriori that the negative log-likelihood of the constrained FSCs is lower than that inferred without any structural assumption. For example, in the evidence accumulation task above, the presence of an external signal that ends the acquisition of information naturally suggests adopting a two-layer topology for the FSC. In the case of a free-response experiment in which the agent freely decides when to go right or left (see, e.g., [53]), a simple linear topology may turn out to be sufficient.

When an environment model is known, such as in the trial-based experimental settings considered here, FSCs can be used as generative models when a model of the environment is known. In some cases – such as the evidence accumulation tasks for rats – observations are independent of actions, and thus the inferred FSCs can generate actions in response to arbitrarily chosen sequences of observations. In general, however, the generation of behavioral trajectories requires knowledge about how observations depend on actions. In any event, using the inferred FSC in generative mode gives the possibility of predicting the agent's response to a crafted environment and suggesting ways to steer the agent's behavior in a desired direction.

While our approach is well-suited for those cases where elementary actions and observations are inherently discrete, real tasks are often described by continuous, possibly high-dimensional behavioral variables. In some cases, it may still be possible to segment behavioral data and obtain once more trajectories of discrete behavioral modes – actions – and observations. Different methods may allow one to analyze behavior at different scales, and an interesting question would then be whether the inferred FSCs remain consistent or whether they reflect the specific behavioral scale captured by the segmentation. In the latter case, comparing FSCs may be a helpful tool for understanding the hierarchical structure of behavior: internal states that persist across segmentations would correspond to robust computational motifs, while states that appear only at finer scales would reveal transient or low-level strategies that are averaged out at coarser descriptions. In practice, however, inference becomes increasingly complex as the number of actions and observations increases, which entails an increase in the number of parameters, effectively bounding the number of internal states that can be resolved from finite behavioral datasets. To deal with these scenarios, a possible extension would be to let the agent explicitly filter irrelevant inputs according to its internal state. This would allow the FSC to extract task-relevant observations from raw sensory inputs, reducing the effective number of observations that each internal state must account for. A complementary approach would be to take into account the continuous nature of actions and observations by resorting to function approximation in order to represent the agent with a manageable number of parameters. For instance, one could replace the parametrization in the Methods with a more general softmax with appropriately selected feature vectors, or endow each internal state with a learnable dynamical model, similarly to other state space models [54]. While a discussion of these approaches goes beyond the scope of this paper, we can report some success in some selected tasks. Further work still remains to be done in this direction.

In general, we stress that the minimalist design of agent models like FSCs is especially useful if a high-level description of behavior is sought. Indeed, an agent model based on low-level actions and observations alone typically requires a larger number of internal states to account for all the corresponding low-level computations, whereas fewer internal states often suffice at the higher level of description. Sometimes, however, it is impossible to avoid the granularity of the lowest levels of description, and this may lead to an unsustainable growth of complexity of the agent model. In these cases, a good compromise is achieved by a hierarchical organization in which different levels are stacked one upon another [55]. For instance, behavior trees [56] are a specific example of hierarchical decision structures widely used in robotics and in game development that could provide inspiration for building parsimonious agent models. In such scenarios, one could imagine inferring directly hierarchical FSCs or behavioral trees to take the granularity of behavior directly into account. Furthermore, it will also be interesting to quantitatively understand in which cases few discrete states can provide good descriptions of behavioral strategies, and when instead other approaches are needed – e.g., behavior can only be described with genuinely continuous internal states.

Finally, we can connect FSC and neural computations by their formal correspondence with linear fully gated recurrent units. For the evidence accumulation tasks, this is even more explicit by the finding that the inferred FSC with $M=5$ nodes can be interpreted as a leaky competing accumulator model. More generally, the internal states of FSC could be viewed as attractors of a recurrent neural network dynamics. This interpretation is consistent with the fact that recurrent neural networks trained on some task display a fast dynamics of convergence to attractors and slow transitions between them, which can be triggered by external inputs or occur spontaneously. This slow dynamics can be effectively represented by discrete states and transitions [24,28,57–59], and it has been suggested that it underlies flexible decisions and computations through dynamics [27,60]. In this context, linear arrangements of internal states, like the ones that emerge in the analysis of rats' and mice's behavior, might then be interpreted as proxies for line attractors [32,61]. The fact that FSC inference appears to be robust even with small numbers of trajectories also opens the avenue to fit agent models directly to single animals and quantify the variability across individuals. In future works, one could pair such individual-based inference with single-trial neural recordings from the same individuals to test whether the inferred internal states correspond to identifiable neural signatures.

In this work, we discussed applications to experimental data of rodent behavior, but our approach does not make assumptions about the nature of the agent. Any underlying computational process, be it neural, biochemical, or artificial, as long as it supports a coarse-grained description in terms of discrete states, could be a suitable candidate to be described by an FSC. It would be interesting to explore the applicability of our method to a broader range of biological agents and tasks, including human decision-making.

Methods

FSC parametrization

A finite state controller with M internal states is defined by three probability distributions – the policy, the internal computation, and the initial state distribution – which we parametrize using softmax functions. The policy $\pi(a|m)$ models the probability of taking an action $a \in \mathcal{A}$ from a specific internal state $m \in \mathcal{M}$, with $|\mathcal{A}| = A$ and $|\mathcal{M}| = M$. Hence,

$$\pi(a|m) \propto \exp\left(\theta_{a,m}^{(\pi)}\right)$$

where $\theta^{(\pi)} \in \mathbb{R}^{A \times M}$ are the parameters we seek to infer. The internal computation $g(m'|m, a, y)$ models the probability of transitioning to a state m' from a state m , conditioned on the action a taken by the agent and the observation $y \in \mathcal{Y}$ it received, with $|\mathcal{Y}| = Y$. As before,

$$g(m'|m, a, y) \propto \exp\left(\theta_{m',m,a,y}^{(g)}\right)$$

where $\theta^{(g)} \in \mathbb{R}^{M \times M \times A \times Y}$. Finally, the initial internal state of the FSC is specified by

$$\rho(m) \propto \exp\left(\theta_m^{(\rho)}\right)$$

with $\theta^{(\rho)} \in \mathbb{R}^M$. Therefore, the total number of parameters is $M(1 + A + MAY)$.

Comparison with Hidden Markov Models

Finite state controllers may be viewed as a generalized type of state space models (SSMs) [62] with recurrent transitions. One of the crucial features of FSCs is that transitions between internal states explicitly depend on both actions and observations. This recurrence allows them to encode previous actions and observations in their internal states. In Fig 6 we explicitly compare FSCs (Fig 6A) with different types of hidden Markov models (HMMs).

In a standard HMM (Fig 6B) transitions between internal states are autonomous and depend only on the previous state, while the policy – also known as the emission probability – depends on the current internal state, as in FSCs. Thus, observations do not influence transitions. A possible generalization is HMMs where emissions are parametrized by a generalized linear model (GLM-HMMs), so that the policy explicitly depends on observations [22] (Fig 6C). More generally, in input-output hidden Markov models (IO-HMMs), the dependence on observations enters both the emission probability and the internal computation [20] (Fig 6D). In Section B of the S1 Appendix, we show that these types of models – as well as extended versions of GLM-HMMs whose policy depends on the previous actions as well – fail to capture the mice behavior in the reversal task due to the fact that the dynamics of the internal states is disentangled from that of the emissions, which is instead a key dependency in FSCs.

FSC inference

The goal is to identify the FSC which is as “close” as possible to the true unknown agent, i.e., solve the minimization problem

$$\operatorname{argmin}_{\text{FSC}} \mathbb{E} \left(D \left(P_{A_{0:T} | Y_{0:T}}^{\text{agent}} \parallel P_{A_{0:T} | Y_{0:T}}^{\text{FSC}} \right) \right)$$

where FSC is a shorthand for the set of parameters θ . Above, $\mathbb{E}$ is the expectation over the distribution of trajectories of actions and observations $(A_{0:T}, Y_{0:T})$ generated by the interaction of the agent and the environment, D is the Kullback-Leibler divergence, and

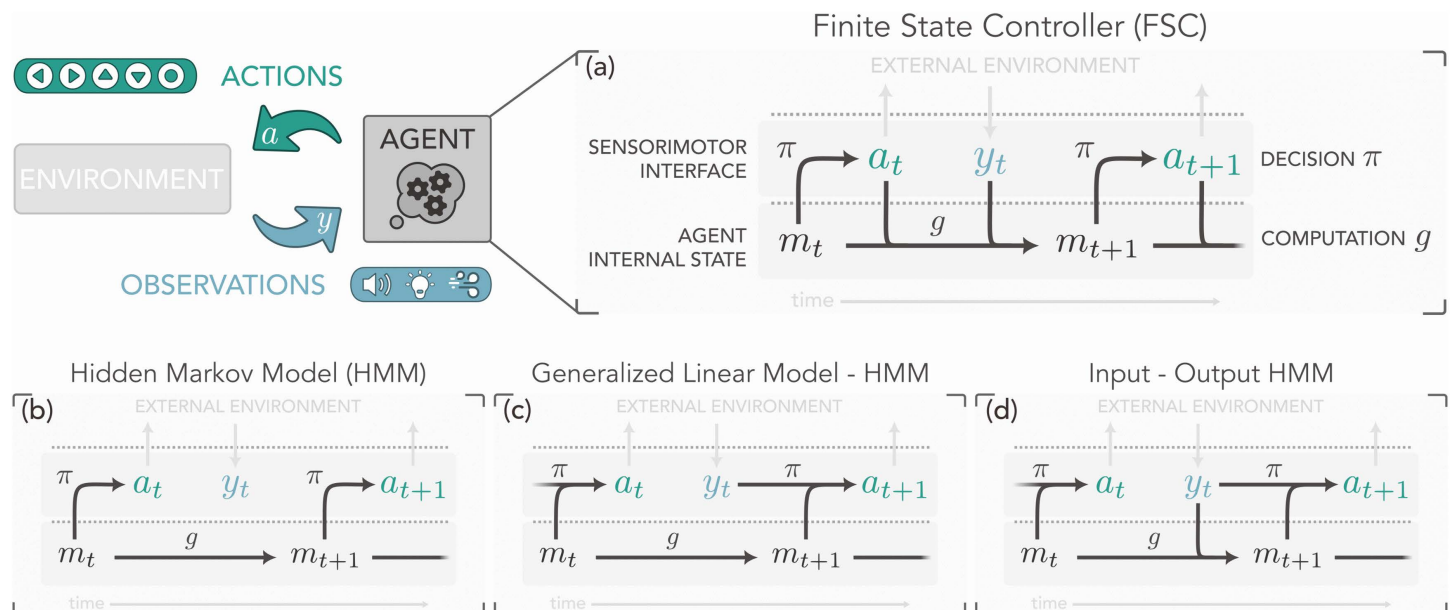


Fig 6. Comparison between finite state controllers and other state space models. (a) The FSC model of the agent, where transitions between internal states are fully recurrent - i.e., they depend on the previous observation, action, and state. (b) In a hidden Markov model (HMM), observations are ignored, and transitions between internal states are autonomous. The policy is known as the emission probability, and solely depends on the current internal state (as in FSCs). (c) In a generalized linear model HMM (GLM-HMM), the transitions between internal states remain autonomous, but the policy depends explicitly on the observation via a generalized linear model [22]. (d) In a general input-output HMM [20], both internal transitions and the policy depend on the observation, as in FSCs.

<https://doi.org/10.1371/journal.pcbi.1014585.g006>

$$P_{A_{0:T}|Y_{0:T}}^{\text{FSC}}(a_{0:T}|y_{0:T}) = \sum_{m_{0:T+1}} \rho(m_0) \prod_{t=0}^T \pi(a_t|m_t) \times \\ \times g(m_{t+1}|m_t, a_t, y_t)$$

where the policy, the internal computation, and the initial distribution are parametrized as described in the previous section. Noting that

$$\underset{\text{FSC}}{\operatorname{argmin}} \mathbb{E} \left(D \left(P_{A_{0:T}|Y_{0:T}}^{\text{agent}} \parallel P_{A_{0:T}|Y_{0:T}}^{\text{FSC}} \right) \right) \\ = \underset{\text{FSC}}{\operatorname{argmin}} \mathbb{E} \left(-\log P_{A_{0:T}|Y_{0:T}}^{\text{FSC}} \right)$$

and replacing the expectation value with the empirical distribution over a set of N behavioral trajectories $(a_0^{(k)}, y_0^{(k)}, \dots, a_{T_k}^{(k)}, y_{T_k}^{(k)})$ for $k = 1, \dots, N$, one arrives at the maximum-likelihood problem (1)

$$\underset{\text{FSC}}{\operatorname{argmin}} \left(-\frac{1}{N} \sum_{k=1}^N \log P_{a_{0:T_k}^{(k)}|y_{0:T_k}^{(k)}}^{\text{FSC}} \right).$$

Metric-adaptive particle swarm optimization

The main idea of MAPSO is sketched in Fig 7. First, N_p FSCs are initialized with random parameters. We can think of each FSC as a particle θ_i in the θ space. The parameters of each particle are updated according to a swarming dynamics, where the velocity of each particle is controlled by three different terms. First, an inertia term decreases the velocity with the number of epochs. The second term is a so-called cognitive coefficient, where the velocity vector of the p -th particle evolves to point towards the best set of parameters ever seen by the particle ($\theta_p^{(\text{cog})}$). Finally, the last term is a social coefficient, where the velocity tends to point towards the best set of parameters ever seen by one of the neighbors of the particles ($\theta_p^{(\text{soc})}$), where the neighbors are specified by a network of interaction that changes as the algorithm progresses. At the beginning, during a “global phase” all particles interact with one another. Then, a “local metric phase” takes place, where at each step of the algorithm, each particle interacts only with a finite number of neighbors defined by the Euclidean distance between them. The number of neighbors decreases until the particles interact only with their nearest neighbors, before increasing back. The algorithm ends with a global phase to guarantee that all particles share the same set of best parameters.

All the parameters governing the relative strength of these terms are adaptively set according to the fuzzy rules introduced in [42]. Furthermore, at each step, the swarming dynamics is hybridized with a genetic algorithm, where each $\theta_p^{(\text{soc})}$ is mutated along a random direction and retained if it reduces the corresponding negative log-likelihood. MAPSO runs for a fixed number of epochs, where each epoch corresponds to a velocity update, a particle update, a parameter strength update, a computation of the metric distances between the particles and the resulting topology, and a mutation step. See Section F of the S1 Appendix for more details and an analysis of MAPSO robustness against hyperparameter choices.

MAPSO training schedule

To select the best inferred FSC, we consider a cross-validation scheme. We randomly split the data into a training set (50% of the available trajectories in the rats’ experiments, and 80% of the available trajectories in the mice’s experiments) and a validation set (the remaining trajectories). Due to the random split, we note that this implies that the training and validation sets will have trials that, in general, are interdispersed in time. In scenarios where long-timescale effects such as

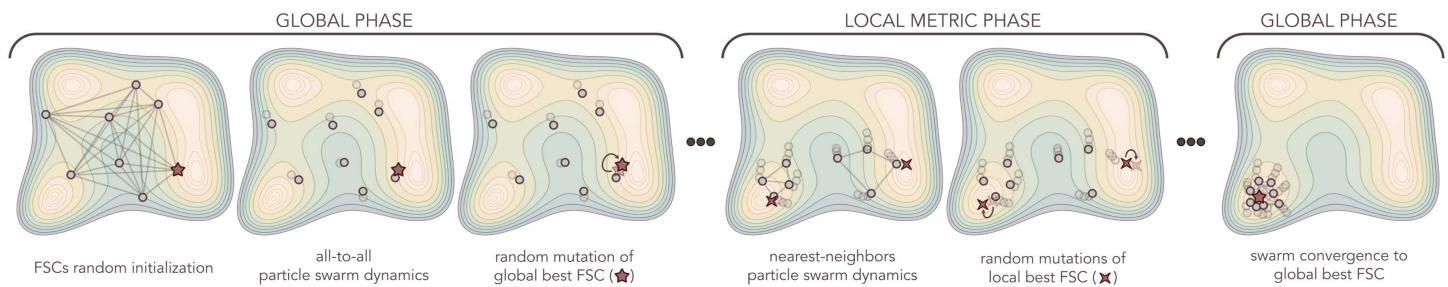


Fig 7. Sketch of the basic structure of metric-adaptive particle swarm optimization (MAPSO). A given number of FSCs are initialized randomly, representing a swarm of particles in the high-dimensional parameter space of the policy, computation, and initial distribution. During an initial global phase, their parameters are updated according to an all-to-all swarming dynamics, where each FSC tends to move towards the best set of parameters it has seen so far, as well as towards the best set of parameters ever found by the swarm. After each step of the swarm evolution, this global best is mutated along a random direction and is kept if the mutation improves the likelihood, and the swarm dynamics is updated according to pre-defined adaptive rules [42]. The global phase is followed by a local metric phase, where the swarm dynamics only involve a finite number of neighbors defined by the Euclidean distance between the particles. During this phase, the FSCs evolve towards the local best set of parameters, i.e., the best set of parameters ever found by the neighbors. The number of neighbors decreases before increasing back, and the algorithm ends with a second global phase.

<https://doi.org/10.1371/journal.pcbi.1014585.g007>

satiation may be important, more careful splits may be chosen. After running MAPSO for a fixed number of epochs over the training set, the best set of parameters ever found by a particle is returned. These parameters correspond to $\theta_p^{(\text{soc})}$ due to the final global phase where all particles interact with one another. Since the optimization problem is non-convex, we run the training procedure multiple times for a given number of internal states M , each with a random set of initial conditions for MAPSO. Then, we compute the negative log-likelihood over the validation set of each of the inferred FSCs, and retain the one with the smallest.

In this work, the initial parameters were extracted from a multivariate Gaussian distribution centered around zero and with diagonal covariance, but other choices are possible to bias the initial search in specific regions of the parameter space. We also note that MAPSO allows for other schedules, such as sequential restarts, where the particles' initial position is not independent but is centered around the best parameters previously found with varying variance. See Section F of the [S1 Appendix](#) for an analysis of MAPSO's results across restarts.

Supporting information

S1 Appendix. Supporting information. Additional descriptions of finite state controllers, the inference procedure, comparison with other models and methods, theoretical derivations, and figures.

(PDF)

Acknowledgments

We acknowledge useful discussions with E. Panizon and K. V. B.F Verano in the early stages of this work.

Author contributions

Conceptualization: Giorgio Nicoletti, Antonio Celani.

Formal analysis: Giorgio Nicoletti, Antonio Celani.

Investigation: Giorgio Nicoletti.

Methodology: Giorgio Nicoletti, Antonio Celani.

Software: Giorgio Nicoletti.

Supervision: Antonio Celani.

Validation: Giorgio Nicoletti.

Visualization: Giorgio Nicoletti.

Writing – original draft: Giorgio Nicoletti, Antonio Celani.

Writing – review & editing: Giorgio Nicoletti, Antonio Celani.

References

1. Golding I. Decision making in living cells: lessons from a simple system. *Annu Rev Biophys.* 2011;40:63–80. <https://doi.org/10.1146/annurev-biophys-042910-155227> PMID: [21545284](#)
2. Balázsi G, Oudenaarden A, Collins JJ. Cellular decision making and biological noise: from microbes to mammals. *Cell.* 2011; 144(6): 910–25. <https://doi.org/10.1016/j.cell.2011.01.030>
3. Berg HC. *E. coli in motion*. Springer; 2004.
4. Colman-Lerner A, Gordon A, Serra E, Chin T, Resnekov O, Endy D, et al. Regulated cell-to-cell variation in a cell-fate decision system. *Nature.* 2005;437(7059):699–706. <https://doi.org/10.1038/nature03998> PMID: [16170311](#)
5. Montague PR, Dayan P, Person C, Sejnowski TJ. Bee foraging in uncertain environments using predictive hebbian learning. *Nature.* 1995;377(6551):725–8. <https://doi.org/10.1038/377725a0> PMID: [7477260](#)
6. Chittka L, Skorupski P, Raine NE. Speed-accuracy tradeoffs in animal decision making. *Trends Ecol Evol.* 2009;24(7):400–7. <https://doi.org/10.1016/j.tree.2009.02.010> PMID: [19409649](#)
7. Carandini M, Churchland AK. Probing perceptual decisions in rodents. *Nat Neurosci.* 2013;16(7):824–31. <https://doi.org/10.1038/nn.3410> PMID: [23799475](#)
8. Hanks TD, Summerfield C. Perceptual decision making in rodents, monkeys, and humans. *Neuron.* 2017;93(1):15–31. <https://doi.org/10.1016/j.neuron.2016.12.003> PMID: [28056343](#)
9. Perkins TJ, Swain PS. Strategies for cellular decision-making. *Mol Syst Biol.* 2009;5:326. <https://doi.org/10.1038/msb.2009.83> PMID: [19920811](#)
10. Gold JI, Shadlen MN. The neural basis of decision making. *Annu Rev Neurosci.* 2007;30:535–74. <https://doi.org/10.1146/annurev.neuro.29.051605.113038> PMID: [17600525](#)
11. Lee D, Seo H, Jung MW. Neural basis of reinforcement learning and decision making. *Annu Rev Neurosci.* 2012;35:287–308. <https://doi.org/10.1146/annurev-neuro-062111-150512> PMID: [22462543](#)
12. Wohrer A, Humphries MD, Machens CK. Population-wide distributions of neural activity during perceptual decision-making. *Prog Neurobiol.* 2013;103:156–93. <https://doi.org/10.1016/j.pneurobio.2012.09.004> PMID: [23123501](#)
13. Wang X-J. Decision making in recurrent neuronal circuits. *Neuron.* 2008;60(2):215–34. <https://doi.org/10.1016/j.neuron.2008.09.034> PMID: [18957215](#)
14. Heess N, Hunt JJ, Lillicrap TP, Silver D. Memory-based control with recurrent neural networks. 2015:1512.04455.
15. Meuleau N, Peshkin L, Kim K, Kaelbling LP. Learning finite-state controllers for partially observable environments. In: *Proceedings of the fifteenth conference on uncertainty in artificial intelligence*. 1999; 427–36.
16. Kaelbling LP, Littman ML, Cassandra AR. Planning and acting in partially observable stochastic domains. *Artificial Intelligence.* 1998;101(1–2):99–134. [https://doi.org/10.1016/s0004-3702\(98\)00023-x](https://doi.org/10.1016/s0004-3702(98)00023-x)
17. Aberdeen D, Baxter J. Scaling internal-state policy-gradient methods for POMDPs. In: *Proc. 19th Int. Conf. Machine Learning*. 2002. 3–10.
18. Amato C, Bonet B, Zilberstein S. Finite-state controllers based on mealy machines for centralized and decentralized POMDPs. *AAAI.* 2010;24(1):1052–8. <https://doi.org/10.1609/aaai.v24i1.7748>
19. Verano KVB, Panizon E, Celani A. Olfactory search with finite-state controllers. *Proc Natl Acad Sci U S A.* 2023;120(34):e2304230120. <https://doi.org/10.1073/pnas.2304230120> PMID: [37579168](#)
20. Bengio Y, Frasconi P. Input-output HMMs for sequence processing. *IEEE Trans Neural Netw.* 1996;7(5):1231–49. <https://doi.org/10.1109/72.536317> PMID: [18263517](#)
21. Calhoun AJ, Pillow JW, Murthy M. Unsupervised identification of the internal states that shape natural behavior. *Nat Neurosci.* 2019;22(12):2040–9. <https://doi.org/10.1038/s41593-019-0533-x> PMID: [31768056](#)
22. Ashwood ZC, Roy NA, Stone IR, International Brain Laboratory, Urai AE, Churchland AK, et al. Mice alternate between discrete strategies during perceptual decision-making. *Nat Neurosci.* 2022;25(2):201–12. <https://doi.org/10.1038/s41593-021-01007-z> PMID: [35132235](#)
23. Bolkan SS, Stone IR, Pinto L, Ashwood ZC, Iravedra Garcia JM, Herman AL, et al. Opponent control of behavior by dorsomedial striatal pathways depends on task demands and internal state. *Nat Neurosci.* 2022;25(3):345–57. <https://doi.org/10.1038/s41593-022-01021-9> PMID: [35260863](#)
24. Sussillo D, Barak O. Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural Comput.* 2013;25(3):626–49. https://doi.org/10.1162/NECO_a_00409 PMID: [23272922](#)

25. Sussillo D. Neural circuits as computational dynamical systems. *Curr Opin Neurobiol.* 2014;25:156–63. <https://doi.org/10.1016/j.conb.2014.01.008> PMID: [24509098](#)
26. Duncker L, Driscoll L, Shenoy KV, Sahani M, Sussillo D. Organizing recurrent network dynamics by task-computation to enable continual learning. *Advance in neural information processing systems.* 2020;33:14387–97.
27. Driscoll LN, Shenoy K, Sussillo D. Flexible multitask computation in recurrent networks utilizes shared dynamical motifs. *Nat Neurosci.* 2024;27(7):1349–63. <https://doi.org/10.1038/s41593-024-01668-6> PMID: [38982201](#)
28. Wong K-F, Wang X-J. A recurrent network mechanism of time integration in perceptual decisions. *J Neurosci.* 2006;26(4):1314–28. <https://doi.org/10.1523/JNEUROSCI.3733-05.2006> PMID: [16436619](#)
29. Guo ZV, Inagaki HK, Daie K, Druckmann S, Gerfen CR, Svoboda K. Maintenance of persistent activity in a frontal thalamocortical loop. *Nature.* 2017;545(7653):181–6. <https://doi.org/10.1038/nature22324> PMID: [28467817](#)
30. Inagaki HK, Fontolan L, Romani S, Svoboda K. Discrete attractor dynamics underlies persistent activity in the frontal cortex. *Nature.* 2019;566(7743):212–7. <https://doi.org/10.1038/s41586-019-0919-7> PMID: [30728503](#)
31. Mazzucato L, La Camera G, Fontanini A. Expectation-induced modulation of metastable activity underlies faster coding of sensory stimuli. *Nat Neurosci.* 2019;22(5):787–96. <https://doi.org/10.1038/s41593-019-0364-9> PMID: [30936557](#)
32. Khona M, Fiete IR. Attractor and integrator networks in the brain. *Nat Rev Neurosci.* 2022;23(12):744–66. <https://doi.org/10.1038/s41583-022-00642-0> PMID: [36329249](#)
33. Recanatesi S, Pereira-Obilinovic U, Murakami M, Mainen Z, Mazzucato L. Metastable attractors explain the variable timing of stable behavioral action sequences. *Neuron.* 2022;110(1):139–153.e9. <https://doi.org/10.1016/j.neuron.2021.10.011> PMID: [34717794](#)
34. Knutson B, Taylor J, Kaufman M, Peterson R, Glover G. Distributed neural representation of expected value. *J Neurosci.* 2005;25(19):4806–12. <https://doi.org/10.1523/JNEUROSCI.0642-05.2005> PMID: [15888656](#)
35. Schwartenbeck P, FitzGerald THB, Dolan R. Neural signals encoding shifts in beliefs. *Neuroimage.* 2016;125:578–86. <https://doi.org/10.1016/j.neuroimage.2015.10.067> PMID: [26520774](#)
36. Starkweather CK, Gershman SJ, Uchida N. The medial prefrontal cortex shapes dopamine reward prediction errors under state uncertainty. *Neuron.* 2018;98(3):616–629.e6. <https://doi.org/10.1016/j.neuron.2018.03.036> PMID: [29656872](#)
37. Hennig JA, Romero Pinto SA, Yamaguchi T, Linderman SW, Uchida N, Gershman SJ. Emergence of belief-like representations through reinforcement learning. *PLoS Comput Biol.* 2023;19(9):e1011067. <https://doi.org/10.1371/journal.pcbi.1011067> PMID: [37695776](#)
38. Ji-An L, Benna MK, Mattar MG. Discovering cognitive strategies with tiny recurrent neural networks. *Nature.* 2025;644(8078):993–1001. <https://doi.org/10.1038/s41586-025-09142-4> PMID: [40604278](#)
39. Spaan MT. Partially observable Markov decision processes", In: Reinforcement learning: state-of-the-art, Springer; 2012. 387–414.
40. Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling", arXiv preprint arXiv:1412.2014;3555.
41. Ravanelli M, Brakel P, Omologo M, Bengio Y. Light gated recurrent units for speech recognition. *IEEE Trans Emerg Top Comput Intell.* 2018;2(2):92–102. <https://doi.org/10.1109/tetci.2017.2762739>
42. Zhan Z-H, Zhang J, Li Y, Chung HS-H. Adaptive particle swarm optimization. *IEEE Trans Syst Man Cybern B Cybern.* 2009;39(6):1362–81. <https://doi.org/10.1109/TSMCB.2009.2015956> PMID: [19362911](#)
43. Bakker B. Reinforcement learning with long short-term memory. *Advances in neural inform process systems.* 2001.
44. Luo TZ, Kim TD, Gupta D, Bondy AG, Kopec CD, Elliott VA, et al. Transitions in dynamical regime and neural mode during perceptual decisions. *Nature.* 2025;646(8087):1156–66. <https://doi.org/10.1038/s41586-025-09528-4> PMID: [40963029](#)
45. Beron CC, Neufeld SQ, Linderman SW, Sabatini BL. Mice exhibit stochastic and efficient action switching during probabilistic decision making. *Proc Natl Acad Sci U S A.* 2022;119(15):e2113961119. <https://doi.org/10.1073/pnas.2113961119> PMID: [35385355](#)
46. Usher M, McClelland JL. The time course of perceptual choice: the leaky, competing accumulator model. *Psychol Rev.* 2001;108(3):550–92. <https://doi.org/10.1037/0033-295x.108.3.550> PMID: [11488378](#)
47. Berman GJ, Bialek W, Shaevitz JW. Predictability and hierarchy in *Drosophila* behavior. *Proc Natl Acad Sci U S A.* 2016;113(42):11943–8. <https://doi.org/10.1073/pnas.1607601113> PMID: [27702892](#)
48. Mathis A, Mamidanna P, Cury KM, Abe T, Murthy VN, Mathis MW, et al. DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nat Neurosci.* 2018;21(9):1281–9. <https://doi.org/10.1038/s41593-018-0209-y> PMID: [30127430](#)
49. Costa AC, Ahamed T, Jordan D, Stephens GJ. A Markovian dynamics for *Caenorhabditis elegans* behavior across scales. *Proc Natl Acad Sci U S A.* 2024;121(32):e2318805121. <https://doi.org/10.1073/pnas.2318805121> PMID: [39083417](#)
50. Sridhar G, Vergassola M, Marques JC, Orger MB, Costa AC, Wyart C. Uncovering multiscale structure in the variability of larval zebrafish navigation. *Proc Natl Acad Sci U S A.* 2024;121(47):e2410254121. <https://doi.org/10.1073/pnas.2410254121> PMID: [39546569](#)
51. Schroeder CE, Wilson DA, Radman T, Scharfman H, Lakatos P. Dynamics of active sensing and perceptual selection. *Curr Opin Neurobiol.* 2010;20(2):172–6. <https://doi.org/10.1016/j.conb.2010.02.010> PMID: [20307966](#)

52. Yang SC-H, Wolpert DM, Lengyel M. Theoretical perspectives on active sensing. *Curr Opin Behav Sci.* 2018;11:100–8. <https://doi.org/10.1016/j.cobeha.2016.06.009> PMID: [30175197](https://pubmed.ncbi.nlm.nih.gov/30175197/)
53. Chakravarty S, Delgado-Sallent C, Kane GA, Xia H, Do QH, Senne RA, Scott BB. A cross-species framework for investigating perceptual evidence accumulation. 2024.
54. Linderman S, Johnson M, Miller A, Adams R, Blei D, Paninski L. Bayesian learning and inference in recurrent switching linear dynamical systems. In: *Artificial intelligence and statistics.* 2017; 914–22.
55. Botvinick MM. Hierarchical reinforcement learning and decision making. *Curr Opin Neurobiol.* 2012;22(6):956–62. <https://doi.org/10.1016/j.conb.2012.05.008> PMID: [22695048](https://pubmed.ncbi.nlm.nih.gov/22695048/)
56. Ögren P, Sprague CI. Behavior trees in robot control systems. *Annu Rev Control Robot Auton Syst.* 2022;5(1):81–107. <https://doi.org/10.1146/annurev-control-042920-095314>
57. Giles CL, Miller CB, Chen D, Chen HH, Sun GZ, Lee YC. Learning and extracting finite state automata with second-order recurrent neural networks. *Neural Computation.* 1992;4(3):393–405. <https://doi.org/10.1162/neco.1992.4.3.393>
58. Omlin CW, Giles CL. Constructing deterministic finite-state automata in recurrent neural networks. *J ACM.* 1996;43(6):937–72. <https://doi.org/10.1145/235809.235811>
59. Casey M. The dynamics of discrete-time computation, with application to recurrent neural networks and finite state machine extraction. *Neural Comput.* 1996;8(6):1135–78. <https://doi.org/10.1162/neco.1996.8.6.1135> PMID: [8768390](https://pubmed.ncbi.nlm.nih.gov/8768390/)
60. Vyas S, Golub MD, Sussillo D, Shenoy KV. Computation through neural population dynamics. *Annu Rev Neurosci.* 2020;43:249–75. <https://doi.org/10.1146/annurev-neuro-092619-094115> PMID: [32640928](https://pubmed.ncbi.nlm.nih.gov/32640928/)
61. Pagan M, Tang VD, Aoi MC, Pillow JW, Mante V, Sussillo D, et al. Individual variability of neural computations underlying flexible decisions. *Nature.* 2025;639(8054):421–9. <https://doi.org/10.1038/s41586-024-08433-6> PMID: [39608399](https://pubmed.ncbi.nlm.nih.gov/39608399/)
62. Friedland B. *Control system design: an introduction to state-space methods.* Courier Corporation; 2012.