INFERRING TIME-VARYING INTERNAL MODELS OF AGENTS THROUGH DYNAMIC STRUCTURE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) models usually assume a stationary internal model structure of agents, which consists of fixed learning rules and environment representations. However, this assumption does not account for real problem-solving by individuals who can exhibit irrational behaviors or hold inaccurate beliefs about their environment. In this work, we present a novel framework called Dynamic Structure Learning (DSL), which allows agents to adapt their learning rules and internal representations dynamically. This structural flexibility enables a deeper understanding of how individuals learn and adapt in real-world scenarios. The DSL framework reconstructs the most likely sequence of agent structures, sourced from a pool of learning rules and environment models, based on observed behaviors. The method provides insights into how an agent's internal structure model evolves as it transitions between different structures throughout the learning process. We applied our framework to study the behavior of rats in a maze task. Our results show that rats progressively refine their mental map of the maze, evolving from a suboptimal representation associated with repetitive errors to an optimal one that guides efficient navigation. Concurrently, their learning rules transition from heuristic-based to more rational approaches. These findings underscore the importance of both credit assignment and representation learning in complex behaviors. Going beyond simple reward-based associations, our research offers valuable insight into the cognitive mechanisms underlying decision-making in natural intelligence. DSL framework allows better understanding and modeling how individuals in real-world scenarios exhibit a level of adaptability that current AI systems have yet to achieve.

1 INTRODUCTION

Behavioral research traditionally explores how individuals address the *credit assignment problem* (CAP), the challenge of attributing 'values' to actions based on their effectiveness in achieving rewards (Doya, 1999; Daw et al., 2005; Niv, 2007; Otto et al., 2013; Dolan & Dayan, 2013; Dezfouli & Balleine, 2013; Cushman & Morris, 2015). Typically, these studies assume a stationary agent structure, where an agent adheres to a consistent learning rule and employs a fixed internal representation of its environment. However, this model does not reflect the complexities of real-world behavior, where an individual's internal environment representation and learning rule can evolve, resulting in more adaptive behavior.

We introduce a Dynamic Structure Learning (DSL) framework designed to capture how agents transition between different internal model structures. In dynamic structure models (Muzy & Zeigler, 2014a; Uhrmacher, 2001; Barros, 1997), changes in structure consist of the addition, deletion, or alteration of model components. We extend this approach here to learning systems. Specifically, we define an Agent Structure (AS) as a combination of an internal environment representation (decision graph) and a learning rule, which is a Reinforcement Learning (RL) algorithm responsible for credit assignment (Figure 1A). By constructing all possible AS combinations from a set of reinforcement learning rules and environment representations, we can infer the most likely sequence of ASs for an individual, based on its behavioral observations.

We apply the Dynamic Structure Learning (DSL) framework to a T-maze task involving rats to investigate their learning behavior during the experiment. Our **first objective** is to investigate whether rats begin with the suboptimal rats' Internal Maze Representation (IMR) and later transition to the optimal one. Specifically, we consider two types of environment representations: a suboptimal representation (IMR_{subOpt}) that could lead to loop errors in the maze, and an optimal representation (IMR_{opt}). **Secondly**, we explore whether rats rely on heuristic learning strategies, such as memorizing past choices, when their observations conflict with their environmental expectations (e.g., when using IMR_{subOpt}). Over time, we assess whether they shift to a more optimal learning rule as they acquire the correct environmental representation. For this, we use a heuristic learning rule called Cognitive Activity-based Credit Assignment (CoACA) (James et al., 2023), inspired by Activity-based Credit Assignment (ACA) (Muzy, 2019). In CoACA, actions with longer durations are considered more memorable and receive higher credits in rewarded episodes. This suboptimal approach is compared with a more optimal learning rule: Discontinuous Reward Reinforcement Learning (DRL), a continuous-time variant of Q-learning (Watkins & Dayan, 1992; Bradtke & Duff, 1994) that aims to maximize expected returns. DRL is based on Temporal Difference (TD) learning, which models dopamine activity in the brain's reward system (Schultz et al., 1997). The combination of two learning rules and two environment representations results in four potential agent structures (ASs):

- *suboptimal AS*: the combination of the suboptimal learning rule (LR_{subOpt} or CoACA) and the suboptimal internal maze representation (without feeder boxes) (IMR_{subOpt}),
- *LR suboptimal AS*: the combination of the suboptimal learning rule (LR_{subOpt} or CoACA) and the optimal internal maze representation (with feeder boxes) (IMR_{opt}),
- *IMR suboptimal AS*: the combination of the optimal learning rule (LR_{opt} or Q-learning) and the suboptimal internal maze representation (without feeder boxes) (IMR_{subOpt}), and
- *optimal AS*: the combination of the optimal learning rule (LR_{opt} or Q-learning) and the optimal internal maze representation (with feeder boxes) (IMR_{opt}).

We use the DSL framework to infer the most likely sequence of agent structures (ASs) employed by the rats based on their observed behavior. We define rats' strategies as their ASs, characterized by the combination of learning rules and internal environment representations they utilize.

Inverse Reinforcement Learning (IRL) (Ziebart et al., 2008; Babes et al., 2011; Michini & How, 2012) and latent dynamics models (Reddy et al., 2018; Herman et al., 2016) are used to capture an agent's behavior - IRL by inferring the agent's reward function, and latent dynamics models by capturing the agent's belief about environmental dynamics. In contrast, our framework allows for the evolution of an individual's learning rule and internal environment representation over time, offering a more realistic approach to real-world learning problems, which cannot be fully captured by the static reward and transition functions inferred by IRL and latent dynamics models. Conceptually, our framework aligns with Bayesian Theory of Mind (BToM) methods (Baker et al., 2009; 2017; Rabinowitz et al., 2018), which infer an agent's mental states, beliefs, desires, intentions, or goals based on observed actions. However, BToM operates in a Partially Observable MDP (POMDP) setting, where the observer is uncertain about an agent's internal states and reward expectations. However, in our framework, the observer is uncertain about the agent's internal environment model and learning rule.

Applying DSL to the rats' dataset shows that: (i) rats that show slower learning progress appear to rely on the *suboptimal AS* during the early stages of the experiment before switching to the *optimal AS*, whereas rats that learn quickly adopt the *optimal AS* from the beginning of the experiment, (ii) rats' switches from the *suboptimal AS* to the *optimal AS* indicate a progressive refinement in their perception of the task structure (environment model). The gradual refinement of the IMR requires the rats to "imagine" and construct novel maze representations consistent with their experience, ultimately defining learning as the ability to forge an accurate mental model of the task.

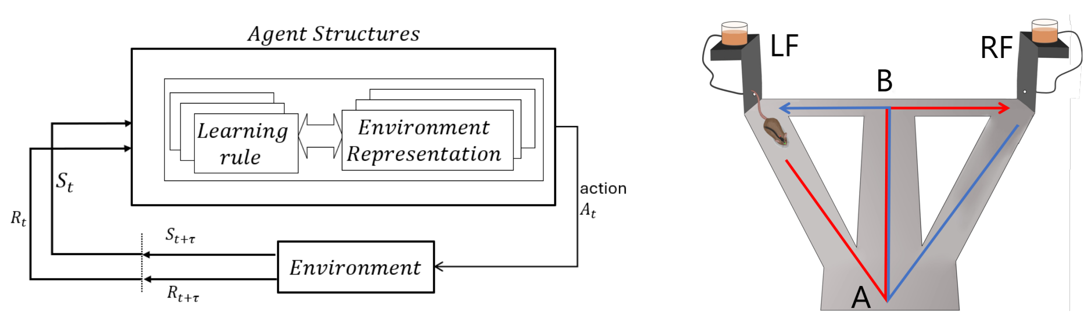
The DSL method introduces a novel approach to understanding learning processes by examining the interaction between the evolving learning rules and internal representations of individuals. By conceptualizing learning as a dual process of environmental modeling and learning rule adaptation, DSL reveals how agents transform their understanding of the environmental and adapt their decision-making rules over time. While our current work focuses on model-free RL methods, the framework can be extended to incorporate model-based RL approaches. This would enable the analysis of complex behaviors, such as the transition between goal-directed (model-based RL) and habitual (model-free RL) behaviors, in learning individuals (Daw et al., 2005; Dolan & Dayan, 2013; Otto et al., 2013). Additionally, the DSL framework could be adapted to accommodate more complex world model representations beyond the standard Markov Decision Process (MDP) that we utilize in this paper. For instance, successor representations (Stachenfeld et al., 2017; Momennejad et al., 2017; Gershman, 2018) and hierarchical models (Botvinick, 2008; Botvinick et al., 2009), which have been explored in human and animal studies, could be integrated into the framework. The dual process perspective of DSL has applications across diverse domains such as psychology (Lee et al., 2012; Dayan & Daw, 2008; Niv, 2009; Doya, 2008), neuroeconomics (Daw & O'Doherty, 2014; Daw & Tobler, 2014; Bossaerts & Murawski, 2015), and neuroscience (Gupta et al., 2010; Stachenfeld et al., 2017; Dupret et al., 2013), where understanding the dynamics of human and animal decision-making is crucial. In conclusion, DSL offers a valuable framework for understanding adaptive intelligence across a wide range of systems.

2 Methods

2.1 Maze experiment

Five male Long-Evans rats were used in the experiment. To motivate the rats to collect food rewards from the maze, they were subjected to a food deprivation program by keeping them at 90% of their body weight during the experiment. Each rat has multiple sessions in the maze, where each session lasts 20 minutes. During sessions, rats can freely move around the maze uninterrupted. The T-maze with return arms (Figure 1B) has two feeding places, left feeder (LF) and right feeder (RF), where rats could receive a food reward. The maze consists of a central stem ($100cm$ long), two choice arms (of $50cm$ each) at one end of the central stem, and two lateral arms connecting the other end of the central stem to the choice arms. Before the experiment, the rats were trained in the maze for two days, with one 20-minute session per day during which they were free to explore the maze and collect the sugar pellets that were randomly scattered throughout the maze. The experiment began on the third day with two 20-minute sessions.

Task description In the experiment, rats are rewarded for taking the Good.LF path from the LF feeder box and the Good.RF path from the RF feeder box (Figure 1B). The other 10 paths (Figure 2) do not yield any reward. Therefore, the task for the rats is to learn to associate the Good.LF and Good.RF paths with reward.



(A) RL Agent Structure.

(B) 3D representation of the T-maze experiment

Figure 1: Agent Structure and 3D representation of the T-maze experiment: (**A**) Semi-Markov Decision Problem (SMDP) formulation of a Reinforcement Learning (RL) problem where agent structure is defined as a combination of learning rule and an internal environment representation, with action of the agent having a random duration τ. (**B**) 3D representation of the T-maze experiment: A, B, LF and RF are four choice points. LF and RF represent the Left and Right Feeders, respectively. Reward path from LF, Good.LF, is shown in red, while reward path from RF, Good.RF, is shown in blue.

Figure 3 highlight the differences between slow-learning rats (rat1, rat2, and rat3) and fast-learning rats (rat4 and rat5). The fast-learning rats learn to get rewards from both LF and RF consistently, whereas the slow learning rats seem to get fewer rewards during the early sessions.

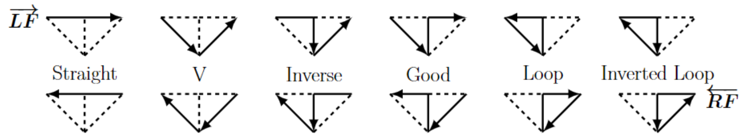


Figure 2: Valid paths in the maze. The rats rarely backtrack due to the narrow maze arm widths, so backward movement is not considered a valid path in our analysis. Top column shows paths starting in Left Feeder (LF) and bottom column shows paths starting in Right Feeder (RF). Rats are rewarded if they take the Good paths from LF and RF.

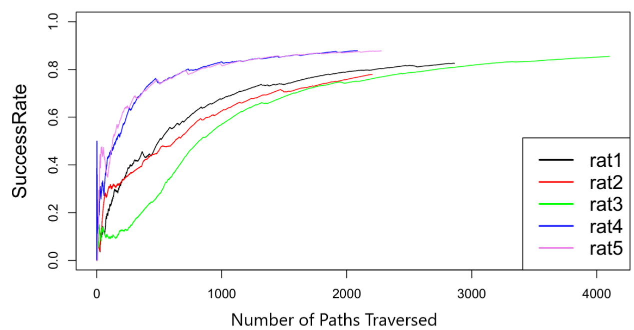


Figure 3: Success rate as proportion of rewarded paths: Success rate computed as the proportion of rewarded paths to the total number of paths traversed. The rats can be categorized as slower learning (rat1, rat2, rat3) or faster learning (rat4, rat5) based on the proportion of rewarded paths.

2.2 Semi-Markov Decision Process

The maze learning task is defined as a Semi-Markov Decision Process (SMDP), which is a generalization of a Markov Decision Process where actions have a random duration. An SMDP can be defined by a tuple (S, A, R, T, F), where S is the set of states, A is the set of actions, R is the reward function that gives the reward associated with each (S, A) in the environment, T is the transition function that gives the transition probabilities $Pr(s'|(s, a))$, $F : F(t|s, a)$, with $t \in \mathbb{R}^+$, gives the probability that the next state s' is reached within time t after action a is chosen in state s.

An *episode* is defined as a minimal segment of the rat's trajectory where the rat starts from one feeder box, visits the other feeder box, and returns to the starting box. Two examples of episode are given below, where τ_{p,n,t_1}, τ_{p,n,t_2} and τ_{p,n,t_3} represents the durations of actions taken at times t_1, t_2 and t_3 in episode n of session p:

$$LF \xrightarrow[\tau_{p,n,t_1}]{(s_{p,n,t_1}, a_{p,n,t_1})} RF \xrightarrow[\tau_{p,n,t_2}]{(s_{p,n,t_2}, a_{p,n,t_2})} LF$$

$$LF \xrightarrow[\tau_{p,n,t_1}]{(s_{p,n,t_1}, a_{p,n,t_1})} LF \xrightarrow[\tau_{p,n,t_2}]{(s_{p,n,t_2}, a_{p,n,t_2})} RF \xrightarrow[\tau_{p,n,t_3}]{(s_{p,n,t_3}, a_{p,n,t_3})} LF$$

2.3 Learning Rules

We employ two learning rules to study the behavior of rats, which are described below.

Cognitive Activity-based Credit Assignment (CoACA) Cognitive Activity-based Credit Assignment (CoACA) uses the concept of activity from Activity-based Credit Assignment (Muzy & Zeigler, 2014b; Muzy, 2019).By prioritizing choices with higher activity (longer duration), CoACA becomes a heuristic decision-making approach – favoring choices that are more memorable due to the effort invested, but not necessarily the most rewarding (James et al., 2023). The CoACA learning rule is further detailed in Section B.1.

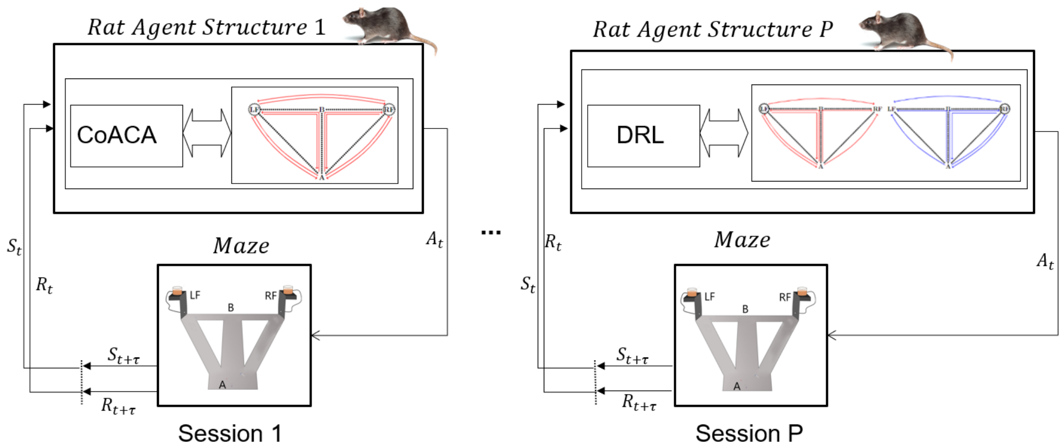
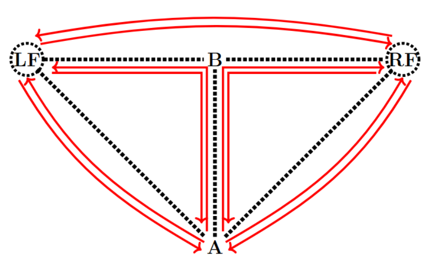


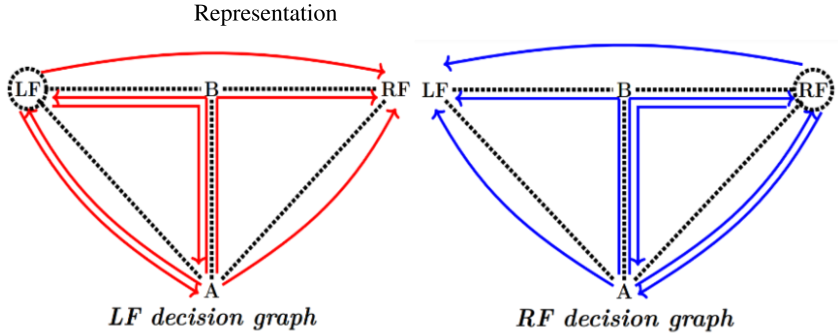
Figure 4: An example of ASs inferred by DSL framework over multiple sessions of rat experiment.

Discounted Reward Reinforcement Learning (DRL) A continuous-time version of Q-learning called SMDP Q-learning, which uses temporal difference (TD) errors to iteratively update Q-values, defines the rational behavior of agents based on an exponential discounting of future rewards (Bradtke & Duff, 1994). The DRL learning rule is further detailed in Section B.2.

2.4 Internal Maze Representations



(A) IMR_{subOpt}: Suboptimal Internal Maze Representation



(B) IMR_{opt}: Optimal Internal Maze Representation

Figure 5: Suboptimal and optimal maze representations: (**A**) IMR_{subOpt}: Suboptimal Internal Maze Representation captures the state and action spaces of the rats in the maze when they do not not account for starting feeder box. (**B**) IMR_{opt}: Optimal Internal Maze Representation captures the state and action spaces of the rats in the maze with two different decision graphs based on the starting feeder (indicated by dotted circles): *LF decision graph* and the *RF decision graph*.

Given the initial high loop error rate, which decreases with learning (see Section A), we propose two distinct MDP representations to model a potential shift in the rats' internal maze representation over time.

Suboptimal maze representation: IMR_{subOpt} (Figure 5A) illustrates a suboptimal decision graph of the maze, representing the state and action spaces of rats in a simplified, but suboptimal manner. The rat's state is defined solely by its current position within the maze, without distinguishing between trajectories based on the starting feeder box. This suboptimal representation can lead to loop errors in the maze (described in detail in Section A.1) as the reward path from LF to RF ($LF \rightarrow A \rightarrow B \rightarrow RF$) shares the trajectory $A \rightarrow B \rightarrow RF$ with the loop path from RF ($RF \rightarrow A \rightarrow B \rightarrow RF$).

Optimal maze representation: IMR_{opt} (Figure 5B) illustrates an optimal decision graph of the maze, representing the state and action spaces of rats in a more complex, but optimal manner. The reward path from LF to RF ($LF \rightarrow A \rightarrow B \rightarrow RF$) belongs to the *LF decision graph*, while the reward path from RF to LF ($RF \rightarrow A \rightarrow B \rightarrow LF$) and the loop path from RF back to itself ($RF \rightarrow A \rightarrow B \rightarrow RF$) belong to the *RF decision graph*. This separation prevents credit sharing between the LF reward path and RF loop path. In IMR_{opt}, the rat's state is represented as a tuple consisting of the starting feeder box and the current position in the maze.

2.4.1 RATS' AGENT STRUCTURES

To model the rats' behavior in the maze, we propose four distinct agent structures, summarized in Table 1. These structures result from combinations of two learning rules, LR_{subOpt} and LR_{opt}, and two environment representations, IMR_{subOpt} and IMR_{opt}.

Agent Structure (AS)	**Learning Rule**	**Internal Maze Representation (IMR)**	**Description**
suboptimal AS	LR_{subOpt} (CoACA)	IMR_{subOpt}	Suboptimal learning rule and suboptimal maze representation.
LR suboptimal AS	LR_{subOpt} (CoACA)	IMR_{opt}	Suboptimal learning, but optimal maze representation.
IMR suboptimal AS	LR_{opt} (DRL)	IMR_{subOpt}	Optimal learning rule, but suboptimal maze representation.
optimal AS	LR_{opt} (DRL)	IMR_{opt}	Optimal learning rule with optimal maze representation.

Table 1: Four different agent structures based on combinations of learning rules and internal maze representations.

2.5 INFERRING RATS' SWITCHING AGENT STRUCTURES

Our objective is to infer the agent structure (AS) used by the rats in each session based on their experimental trajectories. The AS in session p is represented by $x_p \in \{$*suboptimal AS*, *LR suboptimal AS*, *IMR suboptimal AS*, *optimal AS*$\}$. The complete log-likelihood, consisting of the joint distribution of the unknown ASs $x_{1:P}$ and the observed trajectories for each session $y_{1:P}$, where P is the final session, can be expressed as:

$$\log Pr_\theta(x_{1:P}, y_{1:P}) = \log \mu(x_1) + \sum_{p=1}^{P} \log g_\theta(y_p|x_p) + \sum_{p=1}^{P-1} \log f_\theta(x_{p+1}|x_{1:p}) \tag{1}$$

where the initial probabilities $\mu(x_1)$ are uniformly initialized to 0.25, $g_\theta(y_p|x_p)$ gives the likelihood of observations y_p in the p^{th} session and $f_\theta(x_{p+1}|x_{1:p})$ gives the transition probabilities of AS given all past ASs and θ represents the parameters estimated from the experimental data of rats.

From an observer's perspective, we assume that rats do not adopt new ASs once they acquire IMR_{opt}, as they begin to maximize rewards immediately upon learning IMR_{opt}. Since their behavior stabilizes and does not change further once they learn IMR_{opt}, we assume that no additional AS changes occur. Thus, in theory, rats can learn the optimal policy using both CoACA and DRL alongside IMR_{opt}. Therefore, we restrict the rats from exploring new ASs after acquiring IMR_{opt}. As a result, we focus on six specific ASs, categorized into two groups:

- **Switching from suboptimal to optimal representation:** The rat might start with IMR_{subOpt}, but can still switch to the optimal one later.
- **Sticking with the optimal representation:** Once a rat chooses an AS with IMR_{opt}, it stays with that choice throughout the experiment.

By focusing on below six possible AS combinations, we create a more realistic model that captures the decision-making switch process of the rats:

- *suboptimal AS* $\rightarrow$ *LR suboptimal AS*
- *suboptimal AS* $\rightarrow$ *optimal AS*
- *IMR suboptimal AS* $\rightarrow$ *LR suboptimal AS*
- *IMR suboptimal AS* $\rightarrow$ *optimal AS*
- *LR suboptimal AS*
- *optimal AS*

We use a time-varying transition function based on the Chinese Restaurant Process (CRP) (Aldous et al., 2006) to capture the evolution of ASs according to the six possibilities above. This function defines the probability of employing an AS based on its popularity (the number of times it has been chosen previously). The transition function $f_\theta(x_p|x_{1:p-1})$ is defined below.

For $k = 1, 2, 3, 4$ representing the four ASs, the occurrences of each of the four ASs in the previous sessions $p-1$ is given by:

$$n_k = \sum_{i=1}^{p-1} \mathbb{1}_{(x_i=k)}$$

The number of ASs that been chosen at least once until session p is given by:

$$chosenASCount = \sum_{k=1}^{4} \mathbb{1}_{(n_k>0)}$$

The transition function $f_\theta(x_p|x_{1:p-1})$ is defined for two scenarios: Case 1, where the AS with IMR_{opt} has not yet been selected, allowing the rat to explore new ASs, and Case 2, where the AS with IMR_{opt} has already been chosen, limiting the rat to switching between previously selected ASs without trying any new ones.

Case 1: If *optimal AS* or *LR suboptimal AS* has not been selected until session p, the probability of selecting AS in session p is given by:

$$f_\theta(x_p = k|x_{1:p-1}) = \begin{cases} \dfrac{n_k}{p-1+\alpha_{crp}}, & \text{if } n_k > 0 \\ \\ \dfrac{\frac{\alpha_{crp}}{4-chosenASCount}}{p-1+\alpha_{crp}}, & \text{otherwise} \end{cases} \tag{2}$$

where n_k is the number of times AS k has been selected during sessions $1 : p-1$, α_{crp} is the concentration parameter of CRP. If the rat has not yet selected an optimal AS or the suboptimal AS with IMR_{opt}, the probability of selecting an already-used AS is proportional to how often it was selected previously (n_k), while the probability of choosing a new AS depends on the concentration parameter (α_{crp}) and the number of ASs not yet explored.

Case 2: If either *optimal AS* or *LR suboptimal AS* is selected once:

$$f_\theta(x_p = k|x_{1:p-1}) = \begin{cases} \dfrac{n_k + \frac{\alpha_{crp}}{chosenASCount}}{p-1+\alpha_{crp}}, & \text{if } n_k > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Once the rat selects either an optimal AS or the suboptimal AS with IMR_{opt}, it transitions to a restricted phase where only previously chosen ASs can be selected. The probability of selecting an AS depends on how often it was chosen previously, adjusted by a fraction of α_{crp} for all used ASs, while new ASs are no longer considered.

In our study, observations y_p are the trajectories of the rat in a particular session p and $g(y_p|x_p)$ gives the probability of trajectory y_p in session p:

$$g_\theta(y_p|x_p) = \prod_{n=1}^{N_p} \prod_{t=1}^{T_{n,p}} Pr(a_{p,n,t}|s_{p,n,t})$$

where N_p represents the total number of episodes in session p and depending on the value of x_p, $Pr(a_{p,n,t}|s_{p,n,t})$ can be given either by Equation (7) or Equation (10).

To infer ASs of rats from their behavioral observations, we employ the Dynamic Structure Learning (DSL) method in Algorithm 1 that computes the smoothing distribution of ASs given by $Pr_\theta(x_p|y_{1:P})$ and takes the Maximum A Posteriori estimate to determine the AS in each session. We use Conditional Particle Filter With Ancestor Sampling (CPF-AS) to generate samples from the joint smoothing distribution $Pr_\theta(x_{1:P}|y_{1:P})$ (Lindsten et al., 2014).

In the first step of DSL, we estimate the model parameters θ by computing the maximum likelihood estimate using the approach from Lindsten et al. (2013); Lindholm & Lindsten (2018), which combines Stochastic Approximation Expectation-Maximization (SAEM) with CPF-AS, as outlined in Algorithm A1. In the second step, these estimated parameters are used to compute the joint smoothing distribution $Pr_\theta(x_{1:P}|y_{1:P})$ using Algorithm A2. Finally, the sequence of ASs in each session is determined as the Maximum A Posteriori (MAP) estimate of the smoothing distribution $\Pr_\theta(x_p|y_{1:P})$, where p represents the current session and P is the final session. These steps are detailed in Algorithm 1, while CPF-AS is described in Algorithm A3.

Algorithm 1 Dynamic Structure Learning

Input: Rat behavioral data $y_{1:P}$
Output: Inferred ASs $x_{1:P}$
1. Estimate Parameters:
Run PSAEM (Algorithm A1) to estimate:
$\theta = (\alpha^1_{CoACA}, \gamma^1_{CoACA}, \alpha^2_{CoACA}, \gamma^2_{CoACA}, \alpha^3_{DRL}, \lambda^3_{DRL}, \alpha^4_{DRL}, \lambda^4_{DRL}, \alpha_{crp})$
2. Compute Joint Smoothing Distribution:
Use θ and Algorithm A2 to find $Pr_\theta(x_{1:P}|y_{1:P})$
3. Infer ASs:
For each session p, compute $x_p = \text{argmax}_x Pr_\theta(x_p|y_{1:P})$

Here the model parameters θ include the following: *suboptimal AS*: $\alpha^1_{CoACA}, \gamma^1_{CoACA}$, *LR suboptimal AS*: $\alpha^2_{CoACA}, \gamma^2_{CoACA}$, *IMR suboptimal AS*: $\alpha^3_{DRL}, \lambda^3_{DRL}$, *optimal AS*: $\alpha^4_{DRL}, \lambda^4_{DRL}$, CRP concentration parameter: α_{crp}.

3 Results

3.1 Inference on rat data

To infer how rats switch between agent structures (ASs), we used the Dynamic Structure Learning (DSL) method (see Algorithm 1). This involved first performing model fitting on the experimental data by combining the Conditional Particle Filter with Ancestor Sampling (CPF-AS) (Lindsten et al., 2014) (see Algorithm A3) with Stochastic Approximation Expectation-Maximization (SAEM), following Algorithm A1 (Lindsten, 2013; Lindholm & Lindsten, 2018). The model parameters estimated through Algorithm A1 are presented in Table A.2. The agent structures (ASs) for each session were identified by calculating the Maximum A Posteriori (MAP) estimate of the smoothing distribution $Pr(x_p|y_{1:P})$ determined using Algorithm A2.

Inference results in Figure 6 show that the slow learning rats - rat1, rat2 and rat3, utilize the *suboptimal AS* during the initial few sessions before switching the *optimal AS*. In addition, rat1 seems to switch between *suboptimal AS* and *optimal AS*, before settling on *optimal AS*. In contrast, fast learning rats (rat4 and rat5) seem to learn the optimal maze representation early in the experiment and their behaviour is captured by *optimal AS* throughout the experiment. The behaviour of the slow learning rats - rat1, rat2 and rat3 - where they use the *suboptimal AS* in the first sessions leads to a high frequency of loop errors (Table A.1) without learning the good path from LF and RF. The fast learning rats, on the other hand, are quicker to use the *optimal AS*, even if they also make loop errors in the beginning (Table A.1).

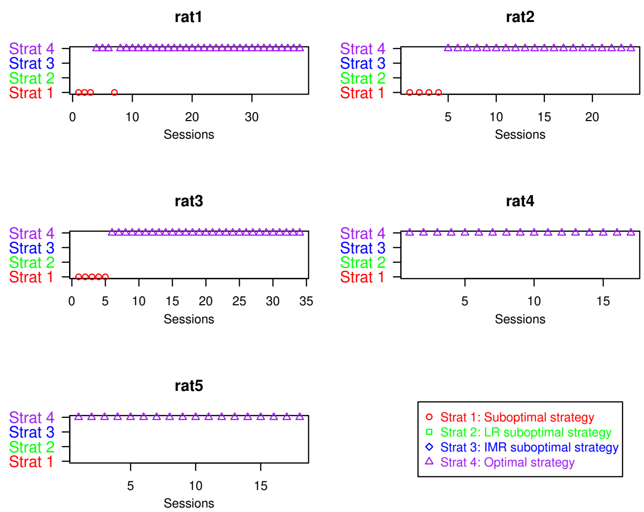


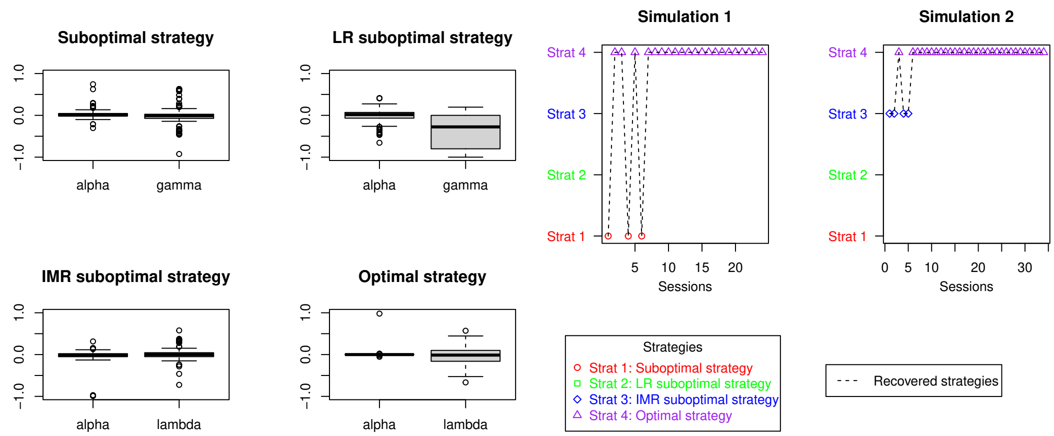
Figure 6: Agent Structures (ASs) of rats inferred using DSL method (see Algorithm 1). ASs result in “strategies” followed by the rats to obtain rewards.

The slow learners showed the cognitive flexibility over time to recognise the need to incorporate “start feeder box” into their internal maze representation and to transition to an optimal behaviour AS. The transition from a suboptimal to an optimal AS over successive sessions highlights two key aspects of the learning process:

- Ability of rats to “imagine” and adopt a new, more complex internal maze representation that matches their empirical observations.
- Nature of learning as an ongoing process of refining and improving the internal maze representation.

Simulation Validation We used simulations to analyze how well DSL recovers the true ASs used to generate the simulated data. In Section Section 2.5, we define learning as the point at which rats infer IMR_{opt}. Once rats infer IMR_{opt}, it is assumed to have learned the task, and its AS evolutions are restricted to six possible combinations where an AS with IMR_{subOpt} can change to an AS with IMR_{opt}. Simulated trajectories of rats were generated based on the six possible combinations defined in Section 2.5. Parameter recovery on simulated data using Algorithm A1 is plotted as boxplot of the recovery error between the true parameter value and the value recovered from the simulated data is shown in Figure 7A. Overall, the parameter recovery error is small, except in the case of *LR suboptimal AS*. The parameter γ^2_{CoACA} exhibits high variance during recovery, likely because it represents a forgetfulness factor in Equation (6) that decays to zero with the square root of the session number p, allowing for a broader range of parameter estimates.

AS recovery is tested by using DSL method (see Algorithm 1) to recover ASs from simulated data. Figure 7B shows two examples where the true ASs were perfectly recovered. The recovery rate of agent structures (ASs) across sessions, based on 300 simulations with 60 instances of each of the six possible AS combinations for 5 rats, is shown in Table A.3.



(A) Boxplots of errors between parameter estimates from the DSL method and true values on simulated data

(B) Successful recovery examples using DSL method

Figure 7: Simulation Validation: (**A**) Boxplots showing the errors between the parameter values estimated by the DSL method and the true values on simulated data. The data is based on 300 simulations, with 60 simulations per each of the six possible ASs. (**B**) Recovery examples with successful recovery using DSL method on simulated data: Simulation 1 (using rat1 parameters) where AS switches from *suboptimal AS* $\rightarrow$ *optimal AS*; Simulation 2 (using rat3 parameters) where AS switches from *LR suboptimal AS* $\rightarrow$ *optimal AS*.

4 CONCLUSION

We developed a Dynamic Structure Learning (DSL) framework to infer an agents' internal models based on their evolving cognitive processes. DSL models an agent's internal dynamics as the interaction between its learning rule and its internal environment representation, reconstructing the most likely sequence of agent structures (ASs) from observed behavior.

We applied DSL to test whether the rats' strategies evolved during learning, defining four agent structures (ASs) by combining two maze representations (suboptimal and optimal) with two learning rules (heuristic and optimal). The *optimal AS*, which paired the optimal maze representation with the optimal learning rule, maximized rewards, while suboptimal ASs resulted in more errors. Inference showed that slow learners initially relied on the *suboptimal AS* and gradually transitioning to the *optimal AS* over time, in contrast to fast learners, who adopted the *optimal AS* early on. Slow-learning rats use a heuristic credit assignment scheme (CoACA) that prioritizes previously rewarded choices with longer durations. This behavior may arise when their internal environment model (IMR_{subOpt}) conflicts with their observations - such as the absence of rewards from the loop path. In such cases, the rats rely on the heuristic learning rule rather than optimizing based on their internal model (Mousavi & Gigerenzer, 2017).

Our model captures rats' switching between internal model structures but assumes fixed internal models within sessions. While it is plausible that rats transition gradually from a suboptimal to an optimal internal representation ($IMR_{subOpt} \rightarrow IMR_{opt}$), it's challenging to accurately infer such subtle changes from observational data. Therefore, we focused on identifying the two most significant representations that explain most of the rats' behavioral changes in our experiment. By modeling the transition from suboptimal to optimal maze representations, we demonstrate how learning involves "imagining" new world models. This capacity for generating novel ideas from past experiences is key to natural intelligence (Buzsáki & Tingley, 2018; Comrie et al., 2022; Kurth-Nelson et al., 2023), enabling adaptability across environments—a capability that current AI models lack. Understanding the computational mechanisms behind this imaginative process could bridge the gap between natural and artificial intelligence, helping build more flexible and robust AI systems (Lake et al., 2017; Botvinick et al., 2017; Siemens et al., 2022).

REFERENCES

David J Aldous, Illdar A Ibragimov, and Jean Jacod. *Summer School of Probability of Saint-Flour XIII, 1983*, volume 1117. Springer, 2006.

Monica Babes, Vukosi Marivate, Kaushik Subramanian, and Michael L Littman. Apprenticeship learning about multiple intentions. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 897–904, 2011.

Chris L Baker, Rebecca Saxe, and Joshua B Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009.

Chris L Baker, Julian Jara-Ettinger, Rebecca Saxe, and Joshua B Tenenbaum. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour*, 1(4): 0064, 2017.

Fernando J Barros. Modeling formalisms for dynamic structure systems. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 7(4):501–515, 1997.

Francesco Biscani and Dario Izzo. A parallel global multiobjective framework for optimization: pagmo. *Journal of Open Source Software*, 5(53):2338, 2020. doi: 10.21105/joss.02338. URL https://doi.org/10.21105/joss.02338.

Peter Bossaerts and Carsten Murawski. From behavioural economics to neuroeconomics to decision neuroscience: the ascent of biology in research on human decision making. *Current Opinion in Behavioral Sciences*, 5:37–42, 2015.

Matthew Botvinick, David GT Barrett, Peter Battaglia, Nando de Freitas, Darshan Kumaran, Joel Z Leibo, Timothy Lillicrap, Joseph Modayil, Mohamed Shakir, Neil C Rabinowitz, et al. Building machines that learn and think for themselves. *Behavioral and Brain Sciences*, 40, 2017.

Matthew M Botvinick. Hierarchical models of behavior and prefrontal function. *Trends in cognitive sciences*, 12(5):201–208, 2008.

Matthew M Botvinick, Yael Niv, and Andew G Barto. Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective. *cognition*, 113(3):262–280, 2009.

Steven Bradtke and Michael Duff. Reinforcement learning methods for continuous-time markov decision problems. *Advances in neural information processing systems*, 7, 1994.

György Buzsáki and David Tingley. Space and time: the hippocampus as a sequence generator. *Trends in cognitive sciences*, 22(10):853–869, 2018.

Nicolas Chopin, Omiros Papaspiliopoulos, et al. *An introduction to sequential Monte Carlo*, volume 4. Springer, 2020.

Alison E Comrie, Loren M Frank, and Kenneth Kay. Imagination as a fundamental function of the hippocampus. *Philosophical Transactions of the Royal Society B*, 377(1866):20210336, 2022.

Fiery Cushman and Adam Morris. Habitual control of goal selection in humans. *Proceedings of the National Academy of Sciences*, 112(45):13817–13822, 2015.

Nathaniel D Daw and John P O'Doherty. Multiple systems for value learning. In *Neuroeconomics*, pp. 393–410. Elsevier, 2014.

Nathaniel D Daw and Philippe N Tobler. Value learning through reinforcement: the basics of dopamine and reinforcement learning. In *Neuroeconomics*, pp. 283–298. Elsevier, 2014.

Nathaniel D Daw, Yael Niv, and Peter Dayan. Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. *Nature neuroscience*, 8(12):1704–1711, 2005.

Peter Dayan and Nathaniel D Daw. Decision theory, reinforcement learning, and the brain. *Cognitive, Affective, & Behavioral Neuroscience*, 8(4):429–453, 2008.

Amir Dezfouli and Bernard W Balleine. Actions, action sequences and habits: evidence that goal-directed and habitual action control are hierarchically organized. *PLoS computational biology*, 9 (12):e1003364, 2013.

Ray J Dolan and Peter Dayan. Goals and habits in the brain. *Neuron*, 80(2):312–325, 2013.

Kenji Doya. What are the computations of the cerebellum, the basal ganglia and the cerebral cortex? *Neural networks*, 12(7-8):961–974, 1999.

Kenji Doya. Modulators of decision making. *Nature neuroscience*, 11(4):410–416, 2008.

David Dupret, Joseph O'Neill, and Jozsef Csicsvari. Dynamic reconfiguration of hippocampal interneuron circuits during spatial learning. *Neuron*, 78(1):166–180, 2013.

Samuel J Gershman. The successor representation: its computational logic and neural substrates. *Journal of Neuroscience*, 38(33):7193–7200, 2018.

Anoopum S Gupta, Matthijs AA Van Der Meer, David S Touretzky, and A David Redish. Hippocampal replay is not a simple function of experience. *Neuron*, 65(5):695–705, 2010.

Michael Herman, Tobias Gindele, Jörg Wagner, Felix Schmitt, and Wolfram Burgard. Inverse reinforcement learning with simultaneous estimation of rewards and dynamics. In *Artificial intelligence and statistics*, pp. 102–110. PMLR, 2016.

Ashwin James, Patricia Reynaud-Bouret, Giulia Mezzadri, Francesca Sargolini, Ingrid Bethus, and Alexandre Muzy. Strategy inference during learning via cognitive activity-based credit assignment models. *Scientific Reports*, 13(1):9408, Jun 2023. ISSN 2045-2322. doi: 10.1038/s41598-023-33604-2. URL https://doi.org/10.1038/s41598-023-33604-2.

Zeb Kurth-Nelson, Timothy Behrens, Greg Wayne, Kevin Miller, Lennart Luettgau, Ray Dolan, Yunzhe Liu, and Philipp Schwartenbeck. Replay and compositional computation. *Neuron*, 111 (4):454–469, 2023.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40:e253, 2017.

Daeyeol Lee, Hyojung Seo, and Min Whan Jung. Neural basis of reinforcement learning and decision making. *Annual review of neuroscience*, 35(1):287–308, 2012.

Andreas Lindholm and Fredrik Lindsten. Learning dynamical systems with particle stochastic approximation em. *arXiv preprint arXiv:1806.09548*, 2018.

Fredrik Lindsten. An efficient stochastic approximation em algorithm using conditional particle filters. In *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6274–6278. IEEE, 2013.

Fredrik Lindsten, Thomas B Schön, et al. Backward simulation methods for monte carlo statistical inference. *Foundations and Trends® in Machine Learning*, 6(1):1–143, 2013.

Fredrik Lindsten, Michael I Jordan, and Thomas B Schon. Particle gibbs with ancestor sampling. *Journal of Machine Learning Research*, 15:2145–2184, 2014.

Kevin Lloyd, Nadine Becker, Matthew W Jones, and Rafal Bogacz. Learning to use working memory: a reinforcement learning gating model of rule acquisition in rats. *Frontiers in computational neuroscience*, 6:87, 2012.

Bernard Michini and Jonathan P How. Bayesian nonparametric inverse reinforcement learning. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II 23*, pp. 148–163. Springer, 2012.

Ida Momennejad, Evan M Russek, Jin H Cheong, Matthew M Botvinick, Nathaniel Douglass Daw, and Samuel J Gershman. The successor representation in human reinforcement learning. *Nature human behaviour*, 1(9):680–692, 2017.

Shabnam Mousavi and Gerd Gigerenzer. Heuristics are tools for uncertainty. *Homo Oeconomicus*, 34:361–379, 2017.

Alexandre Muzy. Exploiting activity for the modeling and simulation of dynamics and learning processes in hierarchical (neurocognitive) systems. *Computing in Science & Engineering*, 21(1): 84–93, 2019.

Alexandre Muzy and Bernard P Zeigler. Specification of dynamic structure discrete event systems using single point encapsulated control functions. *International Journal of Modeling, Simulation, and Scientific Computing*, 5(03):1450012, 2014a.

Alexandre Muzy and Bernard P Zeigler. Activity-based credit assignment heuristic for simulation-based stochastic search in a hierarchical model base of systems. *IEEE Systems Journal*, 11(4): 1916–1927, 2014b.

Yael Niv. Cost, benefit, tonic, phasic: what do response rates tell us about dopamine and motivation? *Annals of the New York Academy of Sciences*, 1104(1):357–376, 2007.

Yael Niv. Reinforcement learning in the brain. *Journal of Mathematical Psychology*, 53(3):139–154, 2009.

A Ross Otto, Samuel J Gershman, Arthur B Markman, and Nathaniel D Daw. The curse of planning: dissecting multiple reinforcement-learning systems by taxing the central executive. *Psychological science*, 24(5):751–761, 2013.

Neil Rabinowitz, Frank Perbet, Francis Song, Chiyuan Zhang, SM Ali Eslami, and Matthew Botvinick. Machine theory of mind. In *International conference on machine learning*, pp. 4218–4227. PMLR, 2018.

Sid Reddy, Anca Dragan, and Sergey Levine. Where do you think you're going?: Inferring beliefs about dynamics from behavior. *Advances in Neural Information Processing Systems*, 31, 2018.

Wolfram Schultz, Peter Dayan, and P Read Montague. A neural substrate of prediction and reward. *Science*, 275(5306):1593–1599, 1997.

George Siemens, Fernando Marmolejo-Ramos, Florence Gabriel, Kelsey Medeiros, Rebecca Marrone, Srecko Joksimovic, and Maarten de Laat. Human and artificial cognition. *Computers and Education: Artificial Intelligence*, 3:100107, 2022.

Kimberly L Stachenfeld, Matthew M Botvinick, and Samuel J Gershman. The hippocampus as a predictive map. *Nature neuroscience*, 20(11):1643–1653, 2017.

Adelinde M Uhrmacher. Dynamic structures in modeling and simulation: a reflective approach. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 11(2):206–232, 2001.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8:279–292, 1992.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

Eric A Zilli and Michael E Hasselmo. Modeling the role of working memory and episodic memory in behavioral tasks. *Hippocampus*, 18(2):193–209, 2008.

APPENDIX

A BEHAVIORAL ANALYSIS

A.1 COGNITIVE INSIGHTS INTO RATS' SUBOPTIMAL BEHAVIORS

During the learning stage, the rats make significantly more loop path errors compared to other errors. Table A.1 shows the number of errors of each type (loop, backward loop, reverse and v) (Figure 2) made by the rats during the learning stage (first 400 paths). *Chi-square test* shows that the looping errors occur above chance levels and cannot be explained as simply random chance events during the learning phase of the rats.

Table A.1: Error path comparison

	V	**Inverse**	**Loop**	**Inverted Loop**	**Are all wrong paths equally likely?**
rat1	2	1	43	1	No ($pval < 2.2 \cdot 10^{-16}$)
rat2	2	0	19	2	No ($pval = 6 \cdot 10^{-9}$)
rat3	5	4	72	4	No ($pval < 2.2 \cdot 10^{-16}$)
rat4	8	3	13	5	No ($pval = 4.9 \cdot 10^{-2}$)
rat5	6	2	17	4	No ($pval = 3.3 \cdot 10^{-4}$)

A possible explanation for the high number of loop errors is that the rats might be misinterpreting the reward association. The final segment of their successful path from the Left Feeder (LF) (red dotted line in Figure A.1) could be mistakenly linked to the reward itself (located at the Right Feeder, RF). Since both the successful "Good.LF" path and the looping "Loop.RF" path share the segment A $\rightarrow$ B $\rightarrow$ RF, rats might attempt to replicate this sequence even when starting from RF, hoping to receive another reward (depicted by the blue dotted line in Figure A.1).

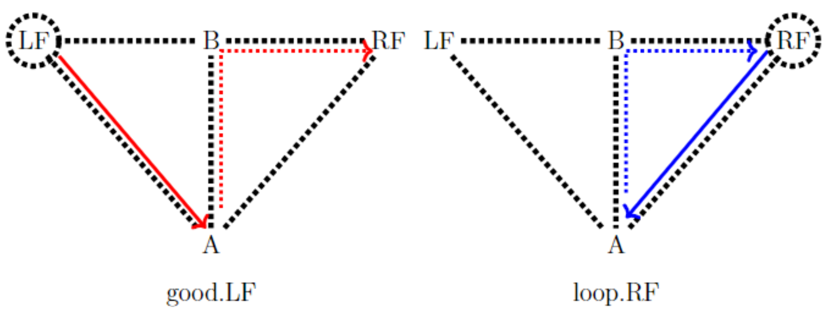


Figure A.1: Loop error: rats mistakenly associate the trajectory $A \rightarrow B \rightarrow RF$ with reward. In suboptimal representation, $A \rightarrow B \rightarrow RF$ while starting in LF (in Good.LF) is same as $A \rightarrow B \rightarrow RF$ while coming from RF (in loop.RF). Dotted circle indicates the starting feeder box.

Alternate explanations for the high number of loop errors are possible, but they are not in agreement with experiment data:

- The loop path could arise because the rats forget which feeder they come from and mistakenly decide to return to the same feeder. If this were the case, then this behavior should consistently persist throughout the experiment, which is not the case. The rats stop making loop errors after they learn the both good paths.
- It is possible that the rats receive a reward and simply want to revisit the same feeder, anticipating more rewards. However, if their sole motivation were to return to the last feeder, a similar preference for both loop and "inverted loop" (returning directly to LF) would be expected. However, in the rats' dataset, we do not observe the same preference for the inverted loop as for the loop path, suggesting a different underlying cause.

Based on the explanation that rats make more loop errors due to mistakenly associating the final segment of the "Good" path with the reward (as shown in Figure A.1), we can hypothesize that these loop errors arise because rats are unaware that "starting feeder box" defines the next reward path. Based on this insight, we will proceed to define both a suboptimal and an optimal decision graph in the subsequent section to further understand and characterize the ASs of rats.

A.2 Behavioral models

A.2.1 Internal Maze Representations (IMR): Suboptimal vs Optimal

Figure 5A represents IMR_{subOpt}, a suboptimal version of the maze decision graph, not accounting for the *starting feeder box*. Here $A \rightarrow B \rightarrow RF$ coming from LF shares the same representation with $A \rightarrow B \rightarrow RF$ coming from RF, thus leading rats to make loop errors (Figure A.1) while searching for rewards.

The optimal maze representation in the maze task, IMR_{opt} (Figure 5B) has a larger state space with a separate decision graph for trajectories starting from LF and trajectories starting from RF. Unlike IMR_{subOpt}, IMR_{opt}, differentiates trajectories $A \rightarrow B \rightarrow RF$ coming from LF and $A \rightarrow B \rightarrow RF$ coming from RF, thus avoids loop errors in the maze.

B Learning Rules

We employ two learning rules to capture the behavior of rats: Cognitive Activity-based Credit Assignment (James et al., 2023) (CoACA), that represents a heuristic learning rule and Discounted Reward Reinforcement Learning (DRL), which implements continuous-time Q-learning (Bradtke & Duff, 1994), representing a more optimal learning rule. Since the task requires rats to remember their starting feeder boxes and, in general, animals are known to employ their working memory (WM) in learning tasks (Lloyd et al., 2012; Zilli & Hasselmo, 2008), we incorporate the memory of one episode into both CoACA and DRL. CoACA implements this by memorizing the actions from last episode, in DRL, eligibility trace implements the working memory of one episode.

B.1 Cognitive Activity-based Credit Assignment (CoACA)

Cognitive Activity-based Credit Assignment (CoACA), which is based on the activity of actions, is used to model heuristic decision-making in rats. Activity is computed as the duration of an action, relative to the duration of an episode:

$$A(s_{p,n,t_i}, a_{p,n,t_i}) = \frac{\tau_{p,n,t_i}}{\sum_{i=1}^{M} \tau_{p,n,t_i}} \tag{4}$$

where t_i represents the the time of the i^{th} action in episode n of session p, where $i \in [0, M]$ with M being the total number of actions in the n^{th} episode of p^{th} session. τ_{p,n,t_i} represents the duration of the action taken at time t_i in episode n of session p.

At the end of an episode n in session p, credits of all (s, a) selected during the episode are updated:

$$K_{p,n+1}(s,a) = K_{p,n}(s,a) + \alpha \times \sum_{i=1}^{M} A(s_{p,n,t_i}, a_{p,n,t_i}) \mathbb{1}_{s_{p,n,t_i}=s} \mathbb{1}_{a_{p,n,t_i}=a} R_{p,n} \ \forall (s,a) \tag{5}$$

Here t_i represents the time at which i^{th} action of episode n in session p was taken, $i \in [1, M]$, $R_{p,n} = \{0, 1, 2\}$ is the total reward obtained in episode n and α is the learning parameter $(0, 1]$. CoACA implicitly employs memory trace of one episode as it requires the agent to maintain a memory of its choices in the last episode. At the end of a session, the credits of all (s, a) pairs in the maze are decayed:

$$K_{p+1,1}(s,a) = (1 - \frac{\gamma}{\sqrt{p}}) \times K_{p,N_p}(s,a) \tag{6}$$

where $\gamma \in [0, 1]$ is forgetfulness parameter, which decays with time, *i.e.*, the rats forget less and less with training and N_p represents the final episode of session p.

The probability of selecting an action a in state $s_{p,n,t}$ is computed using the softmax rule:

$$Pr_{coaca}(a|s_{p,n,t}) = \frac{\exp(K_{p,n}(s_{p,n,t}, a))}{\sum_{a'} \exp(K_{p,n}(s_{p,n,t}, a'))} \tag{7}$$

In contrast to traditional RL which views action duration as a cost to minimize, CoACA interprets duration as the effort invested in a choice. This distinction is captured in CoACA's concept of activity, which acts as a measure of action effort.

B.2 Discounted Reinforcement Learning (DRL)

We employ a continuous-time version of Q-learning (Bradtke & Duff, 1994) to model the optimal learning rule in rats, referred to as Discounted Reinforcement Learning (DRL). The continuous-time Q-learning approach is outlined below. Let s_{p,n,t_1}, a_{p,n,t_1} be part of episode n of session p, leading to new state s_{p,n,t_2} after duration τ_{p,n,t_1} with a reward $r(s_{p,n,t_1}, a_{p,n,t_1}) = \exp(-\beta\tau_{p,n,t_1})R_{t_1+\tau_{p,n,t_1}}$ where $R_{t_1+\tau_{p,n,t_1}} = \{0,1\}$ is the reward obtained in the maze after time τ_{p,n,t_1} for taking action a_{p,n,t_1} at time t_1, and β is the exponential discount factor applied to future rewards. This state transition can be noted as:

$$(s_{p,n,t_1}, a_{p,n,t_1}) \xrightarrow[r(s_{p,n,t_1}, a_{p,n,t_1})]{duration=\tau_{p,n,t_1}} s_{p,n,t_2}$$

Since CoACA implicitly implements a memory trace of an episode, we implement an eligibility trace in DRL, lasting for the duration of a single episode. At time $t_2 = t_1 + \tau_{p,n,t_1}$ after taking action a_{p,n,t_1} at time t_1, eligibility trace e_{p,n,t_2} is updated as below:

$$e_{p,n,t_2}(s,a) = \begin{cases} \lambda \exp(-\beta\tau_{p,n,t_1}) e_{p,n,t_1}(s,a) + 1, & \text{if } (s,a) = \\ & (s_{p,n,t_1}, a_{p,n,t_1}) \\ \lambda \exp(-\beta\tau_{p,n,t_1}) e_{p,n,t_1}(s,a), & \text{otherwise} \end{cases} \tag{8}$$

where $e_{p,n,t_1}(s,a)$ represents the eligibility trace of state-action pair (s,a) at time t_1 in episode n of session p. At the end of an episode, $e(s,a) = 0 \; \forall (s,a)$.

Temporal difference prediction error δ is given by:

$$\delta = r(s_{p,n,t_1}, a_{p,n,t_1}) + \exp(-\beta\tau) \max_{a'} Q(s_{p,n,t_2}, a') - Q(s_{p,n,t_1}, a_{p,n,t_1})$$

TD update is given by: (9)

$$\forall (s,a): \quad Q_{p,n,t}(s,a) \longleftarrow Q_{p,n,t_1}(s,a) + \alpha\delta e_{p,n,t_1}(s,a)$$

The probability of selection of action a in state $s_{p,n,t}$ is

$$Pr_{drl}(a|s_{p,n,t}) = \frac{exp(Q(s_{p,n,t_2}, a))}{\sum_{a'} exp(Q(s_{p,n,t}, a'))} \tag{10}$$

C Dynamic Structure Learning (DSL) Algorithms

In DSL (see Algorithm 1), the first step is to learn the best fitting model parameters using Particle Stochastic Approximation Expectation Maximization (PSAEM), which is described below.

Particle Stochastic Approximation Expectation Maximization (PSAEM) In Algorithm A1, model parameters θ are estimated by computing the maximum likelihood estimate by combining Stochastic Approximation Expectation-Maximization (SAEM) with CPF-AS (Lindsten et al., 2013; Lindholm & Lindsten, 2018). Line 8 of Algorithm A1 represents the E-step of SAEM, where CPF-AS is used to estimate $\hat{Q}_k(\theta)$ using Equation 1. In the M-step (Algorithm A1, line 9), new parameters θ_k, maximizing the $\hat{Q}_k(\theta)$, are determined using the Self-adaptive Differential Evolution optimiser from the Pagmo cpp package (Biscani & Izzo, 2020).

Discount rate β_{DRL} in *IMR suboptimal AS* and *optimal AS* are set to 10^{-4} so that CoACA and DRL models have two parameters each.

Algorithm A1 Particle Stochastic Approximation Expectation Maximization (PSAEM)

1: **Initialize:**
2: Set $\theta_0 = (\alpha^1_{CoACA}, \gamma^1_{CoACA}, \alpha^2_{CoACA}, \gamma^2_{CoACA}, \alpha^3_{DRL}, \lambda^3_{DRL}, \alpha^4_{DRL}, \lambda^4_{DRL}, \alpha_{crp})$
3: Set $\beta^2_{DRL}, \beta^4_{DRL}$ to 10^{-4}
4: Set $\hat{Q}_0(\theta) = 0$
5: Set reference trajectory $x_{1:P}[0]$ arbitrarily
6: **for** $k \geq 1$ **do**
7: Run CPF-AS (Algorithm A3) with N particles and reference trajectory as $x_{1:P}[k-1]$
8: Compute SAEM update by

$$\hat{Q}_k(\theta) = (1-\gamma_k)\hat{Q}_{k-1}(\theta) + \gamma_k \sum_{i=1}^{N} \frac{w^i_P}{\sum_l w^l_P} \log Pr_\theta(x^i_{1:P}, y_{1:P}) \tag{11}$$

where w^i_P is the importance weight of i^{th} particle after final session P, computed by Algorithm A3
9: Compute $\theta_k = \arg\max_\theta \hat{Q}_k(\theta)$
10: Sample particle j with $Pr(j = i) \propto w^i_P$
11: Set $x_{1:P}[k] = x^j_{1:P}$
12: **end for**

Smoothing Algorithm The second step of DSL involves using the parameters estimated with PSAEM to compute the smoothing distribution of Agent Structures (ASs). This algorithm is described below.

Algorithm A2 Smoothing Algorithm

1: **Input:** $x_{1:P}[0]$
2: **Input:** $\theta = (\alpha^1_{CoACA}, \gamma^1_{CoACA}, \alpha^2_{CoACA}, \gamma^2_{CoACA}, \alpha^3_{DRL}, \lambda^3_{DRL}, \alpha^4_{DRL}, \lambda^4_{DRL}, \alpha_{crp})$
3: **Output:** $x_{1:P}[1], x_{1:P}[2], \ldots, x_{1:P}[K]$
4: **for** $k = 1$ to K **do**
5: Run CPF-AS (Algorithm A3) with N particles and reference trajectory as $x_{1:P}[k-1]$ to generate N new agent structure (AS) sequences and particle weights $\{x^i_{1:P}, w^i_P\}^N_{i=1}$.
6: Sample particle j with $Pr(j = i) \propto w^i_P$
7: Set $x_{1:P}[k] = x^j_{1:P}$
8: **end for**

Conditional Particle Filter with Ancestor Sampling (CPF-AS) CPF-AS is used to compute the smoothing distribution by running with $N = 30$ particles. Each particle i has an ancestral trajectory a^i_p that represents the ASs from sessions $1 : p-1$. The ancestral path of each particle represents a potential sequence of ASs, reflecting the behavior of a rat in the maze. Each particle maintains its own unique set of credits or q-values for each of the four different ASs based on its ancestral trajectory a^i_p. A locally optimal proposal distribution is used to propagate particles to time p given by (Chopin et al., 2020)

$$r(x_p|x_{1:p-1}, y_p) = \frac{f_\theta(x_p|x_{1:p-1})g_\theta(y_p|x_p)}{\sum_{x_p} f_\theta(x_p|x_{1:p-1})g_\theta(y_p|x_p)} \tag{12}$$

In CPF-AS, the N^{th} particle ASs $x^N_{1:P}$ are deterministically set to input reference trajectory. The ancestor of the N^{th} particle is resampled based on the ancestor weights given by Equation (13). Since the ASs evolve in non-Markovian manner in our models, (Lindsten et al., 2014) provides a a non-Markovian adaptation where the product is truncated to L steps, which implies a gradual decay of the non-Markovian influence of the current time step p over the next L steps. In our analysis we set ($L = 5$).

Algorithm A3 Conditional Particle Filter with Ancestor Sampling (CPF-AS)

Input: Reference Trajectory $x'_{1:P}$
Input: Truncation parameter $L = 5$
Input: $\theta = (\alpha^1_{CoACA}, \gamma^1_{CoACA}, \alpha^2_{CoACA}, \gamma^2_{CoACA}, \alpha^3_{DRL}, \lambda^3_{DRL}, \alpha^4_{DRL}, \lambda^4_{DRL}, \alpha_{crp})$
Output: Trajectory $x^\star_{1:P}$
for $i = 1$ to $N-1$ **do**
  Draw $x^i_1 \sim r(x_1|y_1)$
end for
Set $x^N_1 = x_1[k]$
for $i = 1$ to $N-1$ **do**
  Set $\tilde{w}^i_1 = \frac{g_\theta(y_1|x^i_1) Pr(x^i_1)}{r_\theta(x^i_1|y_1)}$
end for
for $p = 2$ to P **do**
  for $i = 1$ to $N-1$ **do**
    Draw a^i_p with $Pr(a^i_p = j) \propto w^j_{p-1}$
  end for
  for $i = 1$ to $N-1$ **do**
    Draw $x^i_p \sim r(x_p|x^{a^i_p}_{1:p-1}, y_p)$
  end for
  Set $x^N_p = x'_p$
  Draw a^N_p with

$$Pr(a^i_p = j) \propto w^j_{p-1} \prod_{s=p}^{p-1+L} g_\theta(y_s|x^j_{1:p-1}, x'_{p:s}) f_\theta(x'_s|x^j_{1:p-1}, x'_{p:s-1}) \tag{13}$$

  for $i = 1$ to N **do**
    Set $x^i_{1:p} = \{x^{a^i_p}_{1:p-1}, x^i_p\}$
  end for
  for $i = 1$ to N **do**
    Set

$$\tilde{w}^i_p = \frac{g_\theta(y_p|x^i_p) f_\theta(x^i_p|x^i_{1:p-1})}{r(x^i_p|x^i_{1:p-1}, y_p)} \tag{14}$$

  end for
end for
Sample particle j with $Pr(j = i) \propto w^i_P$
Set $x^\star_{1:P} = x^j_{1:P}$

D Model Parameters Estimated From Rats' Behavioral Data

The model parameters estimated using Algorithm A1 in Step 2 of DSL are given below.

Table A.2: Parameters estimated using Algorithm 2 on experimental data of rats

Rats	acaSubopt		acaOpt		drlSubopt		drlOpt		α_{crp}
	α	γ	α	γ	α	λ	α	λ	
rat1	0.07	0.37	0.93	0.85	0.14	0.12	0.03	0.90	1.88
rat2	0.32	0.56	0.26	0.92	0.73	0.88	0.07	0.43	4.03
rat3	0.077	0.19	0.71	0.85	0.66	0.31	0.02	0.65	4.18
rat4	0.26	0.46	0.94	0.80	0.14	0.81	0.05	0.72	1.63
rat5	0.78	0.06	0.59	0.96	0.52	0.05	0.05	0.52	4.15

E Simulation Validation

Table A.3: Recovery rate of agent structures (ASs) across sessions for six different AS combinations, determined using the DSL method on simulated data

	Recovered AS				
True AS	suboptimal AS	LR suboptimal AS	IMR suboptimal AS	optimal AS	None
suboptimal AS	0.90	0.09	0.01	0	0
LR suboptimal AS	0.01	0.99	0	0	0
suboptimal AS	0.96	0	0.01	0.03	0
optimal AS	0	0	0	1	0
IMR suboptimal AS	0	0	0.96	0.04	0
LR suboptimal AS	0	0.90	0.01	0.08	0.01
IMR suboptimal AS	0	0	0.96	0.04	0
optimal AS	0	0	0	1	0
LR suboptimal AS	0	1	0	0	0
optimal AS	0	0	0	1	0