# Mice adaptively generate choice variability in a deterministic task

**Authors:** Marwen Belkaid [1], Elise Bousseyrol [2], Romain Durand-de Cuttoli [2], Malou Dongelmans [2], Etienne K. Duranté [2], Tarek Ahmed Yahia [2], Steve Didienne [2], Bernadette Hanesse [2], Maxime Come [2], Alex Mourot [2], Jérémie Naudé [2], Olivier Sigaud [1,*] and Philippe Faure [2,* †]

Affiliations :

1 Sorbonne Université, UPMC Univ Paris 06, CNRS, Institut des Systèmes Intelligents et de Robotique (ISIR), 75005 Paris, France.

2 Sorbonne Université, UPMC Univ Paris 06, INSERM, CNRS, Neuroscience Paris Seine - Institut de Biologie Paris Seine (NPS - IBPS), 75005 Paris, France.

* Equally contributing senior authors

[†] correspondence should be addressed P.F. (phfaure@gmail.com)

## Abstract

Can our choices just be driven by chance? To investigate this question, we designed a deterministic setting in which mice reinforce non-repetitive choice sequences, and modeled it using reinforcement learning. Mice progressively increased their choice variability using a memory-free, pseudo-random selection, rather than by learning complex sequences. Our results demonstrate that a decision-making process can self-generate variability and randomness even when the rules governing reward delivery are not stochastic.

## Main

Principles governing random behaviors are still poorly understood, despite well-known ecological examples ranging from preys escaping predators [1] to humans playing competitive games [2]. Dominant theories of behavior and notably reinforcement learning (RL), rely on exploitation, namely the act of repeating actions that have been previously rewarded [3,4]. In this context, choice variability is associated with exploration of environmental contingencies. “Directed” exploration aims at gathering information about environmental contingencies [5,6], whereas random exploration introduces variability regardless of the contingencies [7,8]. Studies have shown that animals are able to increase their choice variability [9], especially when rules change [10,11] or when required to elude predictions about their decisions [12,13]. However, because of the systematic use of probabilistic contingencies, it has remained difficult to experimentally isolate variability generation from environmental conditions. To test the hypothesis that animals can adaptively adjust the randomness of their behavior, we implemented a task where rewards delivery is neither stochastic nor volatile, but where purely random choices constitute a viable strategy.

Mice were trained to perform a sequence of binary choices in an open-field where three target locations were explicitly associated with intra-cranial self-stimulation (ICSS) rewards. Importantly, mice could not receive two consecutive ICSS at the same location. Thus, they had to perform a sequence of choices [14] and at each location to choose the next target amongst the two remaining alternatives (**Fig 1A**). In the training phase, all targets had a 100% probability of reward. We observed that after learning, mice alternated between rewarding locations following a stereotypical circular scheme (**Fig 1B**). Once learning was stabilized, we switched to the complexity condition, in which reward delivery was sequence variability-dependent. More precisely, we estimated the Lempel-Ziv (LZ) complexity [15] of choice subsequences of size 10 (9 past choices + next choice) at each trial. Animals were rewarded when they chose the target associated with the highest complexity. Despite its difficulty, this task is fully deterministic, differentiating it from other approaches [12,13]. Indeed, choice variability cannot be imputed to the inherent stochasticity of the outcomes. We determined that for 25% of all the possible sequences of length 10, the last choice was not rewarded (**Fig 1A**). Therefore, theoretically, while a correct estimation of the complexity of the sequence leads to a success rate of 100%, a pure random selection leads to 75% of success, and a repetitive sequence (e.g. A,B,C,A,B,C...) theoretically grants no reward. We found that mice progressively increased the variability of their choice sequences (**Fig 1B**) and thus their success rate along sessions (**Fig 1C**). This increased variability in the generated sequences was demonstrated by an increase in the normalized LZ-complexity measure (hereafter *NLZcomp*) of the session sequences, a decrease in an entropy measure based on recurrence plot quantification and an increase in the the percentage of U-turns (**Fig 1D**). Furthermore, in the last session 65.5% of the sequences were not significantly different from surrogate sequences generated randomly (**Supp Fig 1A**). Moreover, the success rate was correlated with the *NLZcomp* of the entire session of choice sequences (**Fig 1E**) suggesting that mice increased their reward through an increase variability in their choice. The increase in success rate was associated with an increase of the percentage of U-turns (**Fig 1D** left), yet mice performed a suboptimal U-turn rate of 30%, below the 50% U-turn rate ensuring 100% of rewards. (**Supp Fig 1B**).

From a behavioral point of view, mice thus managed to increase their success rate in a highly demanding task. They did not achieve 100% success but reached performances that indicate a significant level of variability. Given that the task is fully deterministic, the most efficient strategy would be to learn and repeat one (or some) of the 10-choice long sequences that are always rewarded. This strategy ensures the highest success rate but incurs a tremendous memory cost. On the other hand, a purely random selection is another appealing strategy since it is less costly and leads to about 75% of reward. To differentiate between the two strategies and better understand the computational principles underlying variability generation, we examined the ability of a classical RL algorithm to account for the mouse decision-making process under these conditions.

State-action values were learned using the Rescorla-Wagner rule [16] and action selection was based on a softmax policy [3] (**Fig 2A;** see `Methods’). By defining states as vectors including the history of previous locations instead of the current location alone, we were able to vary the memory size of simulated mice and to obtain different solutions from the model accordingly. We found that with no memory (i.e. state = current location), the model learned equal values for both targets in almost all states (**Fig 2B**). In contrast, and in agreement with classical RL, with the history of the nine last choices stored in memory, the model favored the rewarded target in half of the situations by learning higher values (approximately 90 vs 10%) associated with rewarded sequences of choices (**Fig 2B**). In addition, the tendency to perform random choices was dependent not only on the values associated with current choices, but also on the softmax temperature (see `Methods’). Interestingly, this hyperparameter had opposite effects on the model behavior with small and large memory sizes. While increasing the temperature always increased the complexity of choice sequences, it increased the success rate as well for small memory sizes but decreased it for larger memories (**Fig 2C**). This indicates that classical RL can find the optimal solution of the task if using a large memory, while variability is achieved with high choice randomness. A boundary between the two regimes was found between memory sizes of 3 and 4.

Upon optimization of the model to fit mouse behavior, we found that their performance improvement over sessions was best accounted for by an increase of choice randomness using a small memory (**Fig 2D**). This model captured mouse learning better than when using fixed parameters throughout sessions (Bayes factor = 3.46; see `Methods', and **Supp Fig 2D and E**). The model with a memory of size 3 best reproduced mouse behavior (**Fig 2D**), but only slightly better than versions with smaller memories (**Supp Fig 2C**). From a computational perspective, one possible explanation for the fact that although theoretically sufficient, a memory of size 1 fits less than size 3, is that state representation is overly simplified in the model. Thus, with only three states perfectly and unambiguously representing each of the targets, the algorithm was unable to account for the mouse behavioral noise, errors and/or biases. Accordingly, altering the model's state representation to make it more realistic should reduce the size of the memory needed to reproduce mouse performances. To test this hypothesis, we used a variant of the model in which we manipulated state representation ambiguity: each of the locations *{A, B, C}* could be represented by $n \geq 1$ states, with $n = 1$ corresponding to unambiguous states (see `Methods', and **Fig 2E**). As expected, the model fitted better with a smaller memory as representation ambiguity was increased (**Fig 2E**). We also found that the most fit learning rate was higher with ambiguous representations while the randomness factor remained unchanged regardless of ambiguity level (**Fig 2E**). This corroborates that the use of additional memory capacity by the model is due to the model's own limitations rather than an actual need to memorize previous choices. Hence, this computational analysis overall suggests that mice adapted the randomness parameter of their decision-making system to achieve more variability over sessions rather than remembered rewarded choice sequences. This conclusion was further reinforced by a series of behavioral arguments supporting the lack of memorization of choice history in their strategy.

We first looked for evidence of repeated choice patterns in mouse sequences using a Markov chain analysis (see `Methods'). We found that the behavior at the end of the complexity condition was Markovian (**Fig 3A**). In other words, the information about the immediately preceding transition (i.e. to the left or to the right) was necessary to determine the following one (e.g. $p(L) \neq P(L|L)$) but looking two steps back was not informative on future decisions (e.g. $p(L|LL) \approx P(L|L)$). Moreover, we tested whether mice changed directions more often when confronted to the absence of rewards. Indeed, when first facing this complex task after a learning phase in which all targets were systematically rewarding, mice could use a win-stay-lose-switch strategy [12] as a heuristic. Yet, we found that being rewarded (or not) had no effect on the next transition, neither at the beginning nor the end of the complexity condition (**Fig 3B**). To further support the notion that mice did not actually memorize rewarded sequences to solve the task, we finally performed a new experiments in which the complexity condition was followed by a probabilistic condition (**Fig 3C**) in which all targets were rewarded with a 75% probability (the same frequency reached at the end of the previous condition). We hypothesized that if mice were memorizing and repeating sequences in the complexity setting, they would detect this change and modify their behavior. In contrast, we observed that their behavior remained unchanged under the probabilistic condition (**Fig 3C**). Overall, we found no evidence of sequence memorization nor any behavioral pattern that might have been used by mice as a heuristic to solve the complex task.

Exploration and choice variability are generally studied by introducing stochasticity and/or volatility in environmental outcomes [12-14,17] . In this work, we took a step further toward understanding the processes underlying the generation of variability per se, independently from environmental conditions. Confronted with a deterministic task which yet favors complex choice sequences, mice avoided repetitions by approaching random selection. Whether and how random patterns could be generated by the brain has always been puzzling. One hypothesis holds that in human, the process leverages memory [18], to ensure the equality of response usage for example [19], whereas a second hypothesis suggests that the lack of memory may help eliminate counterproductive biases [20] [21]. Our results argue in favor of the latter view: mice did not use their memory but rather adaptively tuned their decision-making parameters to maximize choice randomness.

## Methods

**Animals**: Male C57BL/6J (WT) mice obtained from Charles Rivers Laboratories France (L'Arbresle Cedex, France) were used. Mice arrived to the animal facility at 8 weeks of age, and were housed individually for at least 2 weeks before the electrode implantation. Behavioral tasks started 1 week after implantation to insure full recovery. Since intracranial self-stimulation (ICSS) does not require food deprivation, all mice had ad libitum access to food and water except during behavioral sessions. The temperature (20-22 °C) and humidity was automatically controlled and a circadian light cycle of 12/12 h light-dark cycle (lights on at 8:30 am) was maintained in the animal facility. All experiments were performed during the light cycle, between 09:00 a.m. and 5:00 p.m. Experiments were conducted at Sorbonne University, Paris, France, in accordance with the local regulations for animal experiments as well as the recommendations for animal experiments issued by the European Council (directives 219/1990 and 220/1990).

**ICSS:** Mice were introduced into a stereotaxic frame and implanted unilaterally with bipolar stimulating electrodes for ICSS in the medial forebrain bundle (MFB, anteroposterior = 1.4 mm, mediolateral = ±1.2 mm, from the bregma, and dorsoventral = 4.8 mm from the dura). After recovery from surgery (1 week), the efficacy of electrical stimulation was verified in an open field with an explicit square target (side = 1 cm) at its center. Each time a mouse was detected in the area (D = 3 cm) of the target, a 200-ms train of twenty 0.5-ms biphasic square waves pulsed at 100 Hz was generated by a stimulator. Mice self-stimulating at least 50 times in a 5 minutes session were kept for the behavioral sessions. In the training condition, ICSS intensity was adjusted so that mice self-stimulated between 50 and 150 times per session at the end of the training (ninth and tenth session), then the current intensity was kept the same throughout the different settings.

**Complexity task:** In the complexity condition, reward delivery was determined by an algorithm that estimated the grammatical complexity of animals' choice sequences. More specifically, at a trial in which the animal was at the target location A and had to choose between B and C, we compared the LZ-complexity [15] of the subsequences comprised of the 9 past choices and B or C. Both choices were rewarded if those subsequences were of equal complexity. Otherwise, only the option making the subsequence of highest complexity was rewarded.

**Measures of choice variability:** Two measures of complexity were used to analyze mouse behavior. First, the normalized LZ-complexity (referred to as *NLZcomp* or simply *complexity* throughout the paper) which corresponds to the LZ-complexity divided by the average LZ-complexity of 1000 sequences of the same length generated randomly (a surrogate) with the constraint that two consecutive characters could not be equal like in the experimental setup. *NLZcomp* is small for highly repetitive sequence and is close to 1 for uncorrelated, random signals. Second, the entropy of the frequency distribution of the diagonal length (noted *ENT*), taken from recurrence quantification analysis (RQA). RQA is a series of methods in which the dynamics of complex systems are studied using recurrence plots (RP) [22,23] where diagonal lines illustrate recurrent patterns. Thus, the entropy of diagonal lines reflects the deterministic structure of the system and is smaller for uncorrelated, random signals. RQA was measured using the Recurrence-Plot Python module of the "pyunicorn.timeseries*"* package.

**Computational models:** The task was represented as a Markov Decision Process (MDP) with three states *s* ∈ {A, B, C} and three actions *a* ∈ {GoToA, GoToB, GoToC}, respectively corresponding to the rewarded locations and the transitions between them. State-action values *Q(s,a)* were learned using the Rescorla-Wagner rule [16]:

$$\Delta Q(\mathbf{s_t}, a_t) = \alpha(\mathcal{U}_{t+1} - Q(\mathbf{s_t}, a_t)) \qquad (1)$$

where $\boldsymbol{s_t}=[s_t, s_{t-1}, ..., s_{t-m}]$ is the current state which may include the memory of up to the m^th^ past location, $a_t$ the current action, α the learning rate and *U* the utility function defined as follows:

$$\mathcal{U}_{t+1} = \begin{cases} (1-\kappa).r_{t+1} & \text{if } \ s_{t+1} = s_{t-1} \\ r_{t+1} & \text{otherwise} \end{cases} \qquad (2)$$

where *r* is the reward function and $\kappa$ the U-turn cost parameter modeling the motor cost or any bias against the action leading the animal back to its previous location. The U-turn cost was necessary to reproduce mouse stereotypical trajectories at the end of the training phase (see **Supp Fig 2**).

Action selection was performed using a softmax policy, meaning that in state $s_t$ the action $a_t$ is selected with the probability:

$$P(a_t|\mathbf{s_t}) = \frac{e^{Q(\mathbf{s_t},a_t)/\tau}}{\sum_a e^{Q(\mathbf{s_t},a)/\tau}} \qquad (3)$$

where $\boldsymbol{\tau}$ is the temperature parameter. This parameter reduces the sensitivity to the difference in actions values thus increasing the amount of noise or randomness in decision-making. The U-turn cost $\kappa$ has the opposite effect since it represents a behavioral bias and constrains choice randomness. We refer to the hyperparameter defined as $\varrho = \boldsymbol{\tau} / \kappa$ as the randomness parameter.

In the version referred to as *BasicRL* (see **Supp Fig 2**), we did not include any memory of previous locations nor any U-turn cost. In other words, *m=0* (i.e. $\boldsymbol{s_t} =[ s_t ]$) and $\kappa$ = 0.

To manipulate state representation ambiguity (see **Fig 2**), each of the locations {A, B, C} could be represented by *n≥1* states. For simplicity, we used *n*=1, 2 and 3 for all locations for what we referred to as `null', `low, and `med' levels of ambiguity. This allowed us to present a proof of concept regarding the potential impact of using a perfect state representation in our model.

**Model Fitting:** The main model-fitting results presented in this paper were obtained by fitting the behavior of the mice in the training and complexity conditions session by session independently. This process aimed to determine which values of the two hyperparameters m and $\varrho = \boldsymbol{\tau} / \kappa$ make the model behave as mice in terms of success rate (i.e. percentage of rewarded actions) and complexity (i.e. variability of decisions). Our main goal was to decide between the two listed strategies allowing to solve the task: repeat rewarded sequences or choose randomly. Therefore, we momentarily put aside the question of learning speed and only considered the model behavior after convergence. α was set to 0.1 in these simulations.

Hyperparameters were selected through random search [24] (see ranges listed in **Supplementary Table 1**). The model was run for $2.10^6$ iterations for each parameter set. The fitness score with respect to mice average data at each session was calculated as follows:

$$fitness = 1 - D_{session}$$
$$\text{with} \quad D_{session} = \frac{1}{2}(|\widehat{S} - \overline{S}| + |\widehat{C} - \overline{C}|) \qquad (4,5)$$

where $\overline{S}$ and $\overline{C}$ are the average success rate and complexity in mice respectively and $\widehat{S}$ and $\widehat{C}$ the model success rate and complexity -- all the four ∊ [0,1]. Simulations were long enough for the learning to converge. Thus, instead of multiple runs for each parameter set, which would have been computationally costly, $\widehat{S}$ and $\widehat{C}$ were averaged over the last 10 simulated sessions. We considered that 1 simulated session = 200 iterations, which is an upper bound for the number of trials performed by mice in one actual session.

Since mice were systematically rewarded during training, their success rate in this condition was not meaningful. Thus, to assess the ability of the model to reproduce stereotypically circular trajectories in the last training session, we replaced $\widehat{S}$ and $\overline{S}$ in Equation (5) by $\widehat{U}$ and $\overline{U}$ representing the average U-turn rates for mice and for the model respectively.

Additional simulations were conducted with two goals: 1) test whether one single parameter set could fit mice behavior without the need to change parameter values over sessions, 2) test the influence of state representation ambiguity on memory use in the computational model. Therefore, each simulation attempted to reproduce mouse behavior from training to the complexity condition. Hence, the learning rate α was optimized in addition to the previously mentioned m and $\varrho = \boldsymbol{\tau} / \kappa$ hyperparameters (see ranges listed in **Supplementary Table 1**). Each parameter set was tested over 20 different runs. Each run is a simulation of 4000 iterations, which amounts to 10 training sessions and 10 complexity sessions since simulated sessions consist in 200 iterations. The fitness score was computed as the average score over the last training session and the 10 complexity sessions using Equations (4) and (5). Using a grid search ensured comparable values for different levels of ambiguity (`null', `low, and `med'; see previous section). Given the additional computational cost induced by higher ambiguity levels, we gradually decreased the upper bound of the memory size range in order to avoid long and useless computations in uninteresting regions of the search space.

**Markov chain analysis:** Markov chain analysis allows to mathematically describe the dynamic behavior of the system, i.e. transitions from one state to another, in probabilistic terms. A process is a first-order Markov chain (or more simply *Markovian*) if the transition probability from state A to a state B depends only on the current state A and not on the previous ones. Put differently, the current state contains all the information that could influence the realization of the next state. A classical way to demonstrate that a process is Markovian is to show that the sequence cannot be described by a zeroth-order process, i.e. that $P(B|A) \neq P(B)$, and that the second-order probability is not required to describe the state transitions, i.e. that $P(B|A) = P(B|AC)$.

In this paper, we analyzed the 0th, 1st and 2nd order probabilities in sequences performed by each mouse in the last session of the complex condition (c10). Using the targets A, B, and C as the Markov chain states would have provided a limited amount of data. Instead, we described states as movements to the left (L) or to the right (R) thereby obtaining larger pools of data (e.g. R = A → B, B →C, C → A) and a more compact description (e.g. two 0th order groups instead of three). To assess the influence of rewards on mouse decisions when switching to the complexity condition (i.e. win-stay-lose-switch strategy), we also compared the probability of going forward P(F) or backward P(U) with the conditional probabilities given the presence or the absence of reward (e.g. $P(F_{rw})$ or $P(U_{unrw})$). In this case, F = R→R, L→L and U = R→L, R→L.

**Bayesian model comparison:** Bayesian model comparison aims to quantify the support for a model over another based on their respective likelihoods $P(D|M)$, i.e. the probability that data D are produced under the assumption of the model $M$. In our case, it is useful to compare the fitness of the model $M_{ind}$ fitted session-by-session independently to that of the model $M_{con}$ fitted to all sessions in a continuous way. Since these models do not produce explicit likelihood measures, we used approximate Bayesian computation: considering the 15 best fits (i.e. the 15 parameter sets that granted the highest fitness score), we estimated the models likelihood as the fraction of $(\widehat{S}, \widehat{C})$ couples that were within the confidence intervals of mouse data. Then, the Bayes factor was calculated as the ratio between the two competing likelihoods:

(6)

B > 3 was considered to be a substantial evidence in favor of $M_{ind}$ over $M_{con}$ [25].

**Statistical analysis**: No statistical methods were used to predetermine sample sizes. Our sample sizes are comparable to many studies using similar techniques and animal models. The total number of observations (N) in each group as well as details about the statistical tests were reported in figure captions. Error bars indicate 95% confidence intervals. Parametric statistical tests (*t*-test when comparing two groups or ANOVA when more) were used when data followed a normal distribution (Shapiro test with $p > 0.05$) and non-parametric tests (Mann-Whitney when samples are independent and Wilcoxon when they are paired) when they did not. Homogeneity of variances was checked preliminarily (Bartlett's test with $p > 0.05$) and the unpaired t-tests were Welch-corrected if needed. All statistical tests were applied using the scipy.stats Python module. They were all two-sided except Mann-Whitney. $p > 0.05$ was considered to be statistically non-significant.

Table 1: Hyperparameter ranges used in random search in independent, session-by-session model fitting ($N_{samples} = 6000$) and in grid search in continuous, all-sessions model fitting.

**Hyperparameter random search ranges**

| **Label** | **Range** | **Step** | **Description** |
|---|---|---|---|
| $m$ | $[0, 9]$ | 1 | Memory size |
| $\tau$ | $[1, 20]$ | *continuous* | Softmax temperature |
| $\kappa$ | $[0, 1]$ | *continuous* | U-turn cost |

**Hyperparameter grid search ranges**

| **Label** | **Range** | **Step** | **Description** |
|---|---|---|---|
| $m$ | $[0, 9]$ | 1 | Memory size (no ambiguity) |
|  | $[0, 7]$ | 1 | Memory size (low ambiguity) |
|  | $[0, 5]$ | 1 | Memory size (medium ambiguity) |
| $\alpha$ | $2^{-i}, i \in [0, 10]$ | 1 | Learning rate |
| $\tau$ | $2^{i}, i \in [-4, 4]$ | 1 | Softmax temperature |
| $\kappa$ | $[0.5, 0.95]$ | 0.05 | U-turn cost |

## References

1. Driver, P. M. & Humphries, D. A. Protean behaviour. (Oxford University Press, USA, 1988).

2. Rapoport, A. & Budescu, D. V. Generation of random series in two-person strictly competitive games. J Exp Psychol Gen 121, 352–363 (1992).

3. Sutton, R. S. & Barto, A. G. Reinforcement Learning. (MIT Press, 1998).

4. Schultz, W. Getting formal with dopamine and reward. Neuron 36, 241–263 (2002).

5. Cohen, J. D., McClure, S. M. & Yu, A. J. Should I stay or should I go? How the human brain manages the trade-off between exploitation and exploration. Philos Trans R Soc Lond, B, Biol Sci 362, 933–942 (2007).

6. Rao, R. P. N. Decision making under uncertainty: a neural model based on partially observable markov decision processes. Front Comput Neurosci 4, 146 (2010).

7. Wilson, R. C., Geana, A., White, J. M., Ludvig, E. A. & Cohen, J. D. Humans use directed and random exploration to solve the explore-exploit dilemma. J Exp Psychol Gen 143, 2074–2081 (2014).

8. Mansouri, F. A., Koechlin, E., Rosa, M. G. P. & Buckley, M. J. Managing competing goals - a key role for the frontopolar cortex. Nat Rev Neurosci 18, 645–657 (2017).

9. Grunow, A. & Neuringer, A. Learning to vary and varying to learn. Psychonomic bulletin & review 9, 250–258 (2002).

10. Daw, N. D., O'Doherty, J. P., Dayan, P., Seymour, B. & Dolan, R. J. Cortical substrates for exploratory decisions in humans. Nature 441, 876–879 (2006).

11. Karlsson, M. P., Tervo, D. G. R. & Karpova, A. Y. Network Resets in Medial Prefrontal Cortex Mark the Onset of Behavioral Uncertainty. Science 338, 135–139 (2012).

12. Lee, D., Conroy, M. L., McGreevy, B. P. & Barraclough, D. J. Reinforcement learning and decision making in monkeys during a competitive game. Cognitive brain research 22, 45–58 (2004).

13. Tervo, D. G. R. et al. Behavioral variability through stochastic choice and its gating by anterior cingulate cortex. Cell 159, 21–32 (2014).

14. Naudé, J. et al. Nicotinic receptors in the ventral tegmental area promote uncertainty-seeking. Nat Neurosci 19, 471–478 (2016).

15. Lempel, A. & Ziv, J. On the Complexity of Finite Sequences. IEEE Trans. Information Theory 22, 75–81 (1976).

16. Rescorla, R. A. & Wagner, A. in 64–99 (pdfs.semanticscholar.org, 1972).

17. Cinotti, F. et al. Dopamine regulates the exploration-exploitation trade-off in rats. 1–36 (2019). doi:10.1101/482802

18. Towse, J. N. & Cheshire, A. Random number generation and working memory. European Journal of Cognitive Psychology 19, 374–394 (2007).

19. Oomens, W., Maes, J. H. R., Hasselman, F. & Egger, J. I. M. A Time Series Approach to Random Number Generation: Using Recurrence Quantification Analysis to Capture Executive Behavior. Front. Hum. Neurosci. 9, 319 (2015).

20. Wagenaar, W. Generation of random sequences by human subjects: A critical survey of literature. Psychological Bulletin 77, 65–72 (1972).

21. Maes, J. H. R., Eling, P. A. T. M., Reelick, M. F. & Kessels, R. P. C. Assessing executive functioning: on the validity, reliability, and sensitivity of a click/point random number generation task in healthy adults and patients with cognitive decline. J Clin Exp Neuropsychol 33, 366–378 (2011).

22. Marwan, N., Romano, M. C., Thiel, M. & Kurths, J. Recurrence plots for the analysis of complex systems. Physics Reports 438, 237–329 (2007).

23. Faure, P. & Lesne, A. Recurrence plots for symbolic sequences. Int. J. Bifur. Chaos 20, 1731–1749 (2010).

24. Bergstra, J. & Bengio, Y. Random Search for Hyper-Parameter Optimization. Journal of Machine Learning Research 13, 281–305 (2012).

25. Kass, R. E. & Raftery, A. E. Bayes Factors. Journal of the American Statistical Association 90, 773–795 (1995).

## Acknowledgements

This work was supported by the Centre National de la Recherche Scientifique CNRS UMR 8246, the Foundation for Medical Research (FRM, Equipe FRM DEQ2013326488 to P.F), the French National Cancer Institute Grant TABAC-16-022 (to P.F.) and within the Labex SMART (ANR-11-LABX-65) supported by French state funds managed by the ANR within the Investissements d'Avenir programme under reference ANR-11-IDEX-0004-02. P.F. and J.N. laboratories are part of the École des Neurosciences de Paris Ile-de-France RTRA network. P.F. and J.N. are members of LabEx Bio-Psy.

## Author Contributions

PF designed the behavioral experiment. PF, EB, RDC, MD, ED, TAY, BH and MC performed the behavioral experiments, PF and MB analyzed the behavioral data. MB developed the computational model, SD developed some acquisition tools. JN and OS contributed to modeling studies and data analysis. MB, PF, JN and OS wrote the manuscript with inputs from AM.

## Competing interests

The authors declare no competing interests.

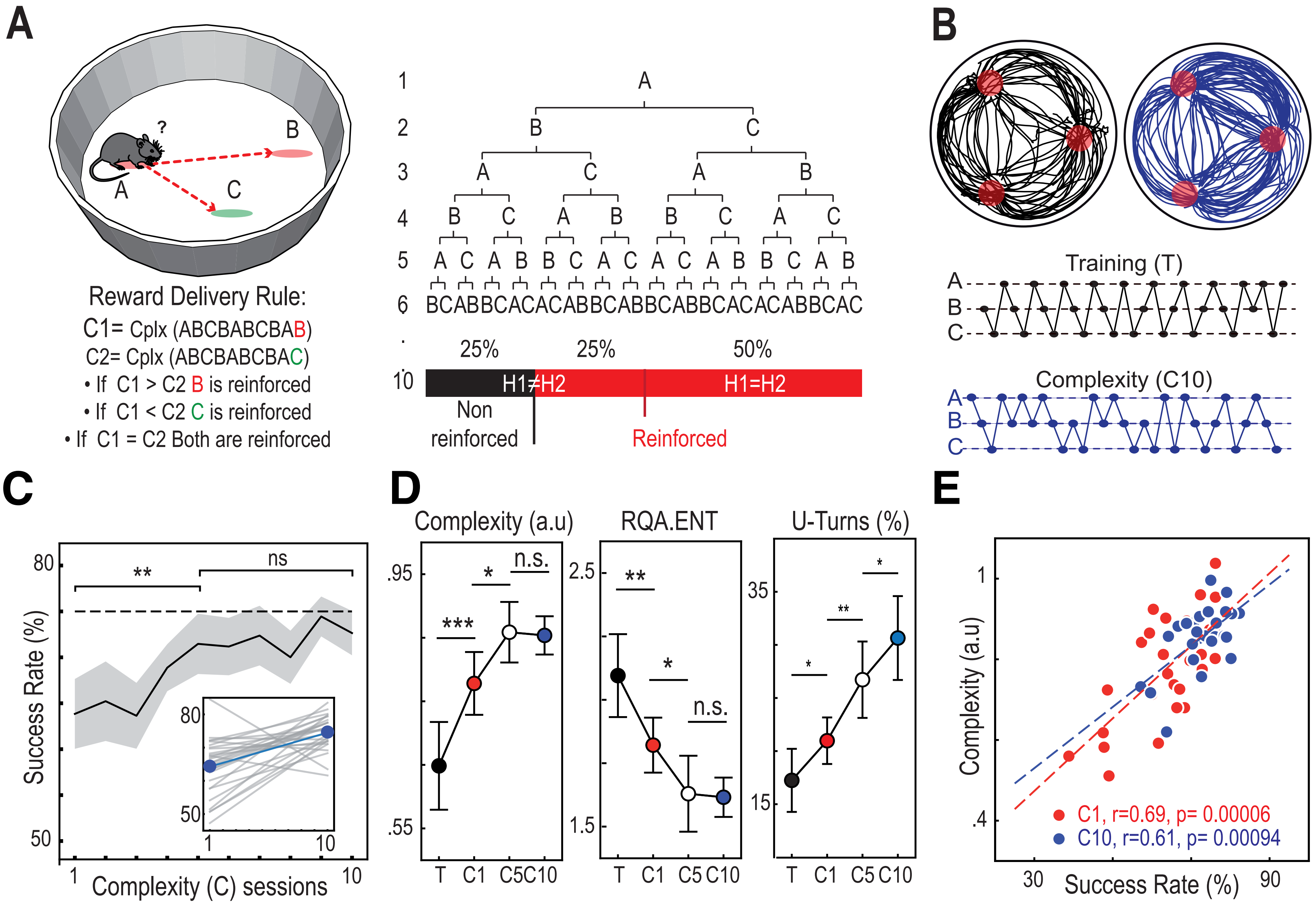
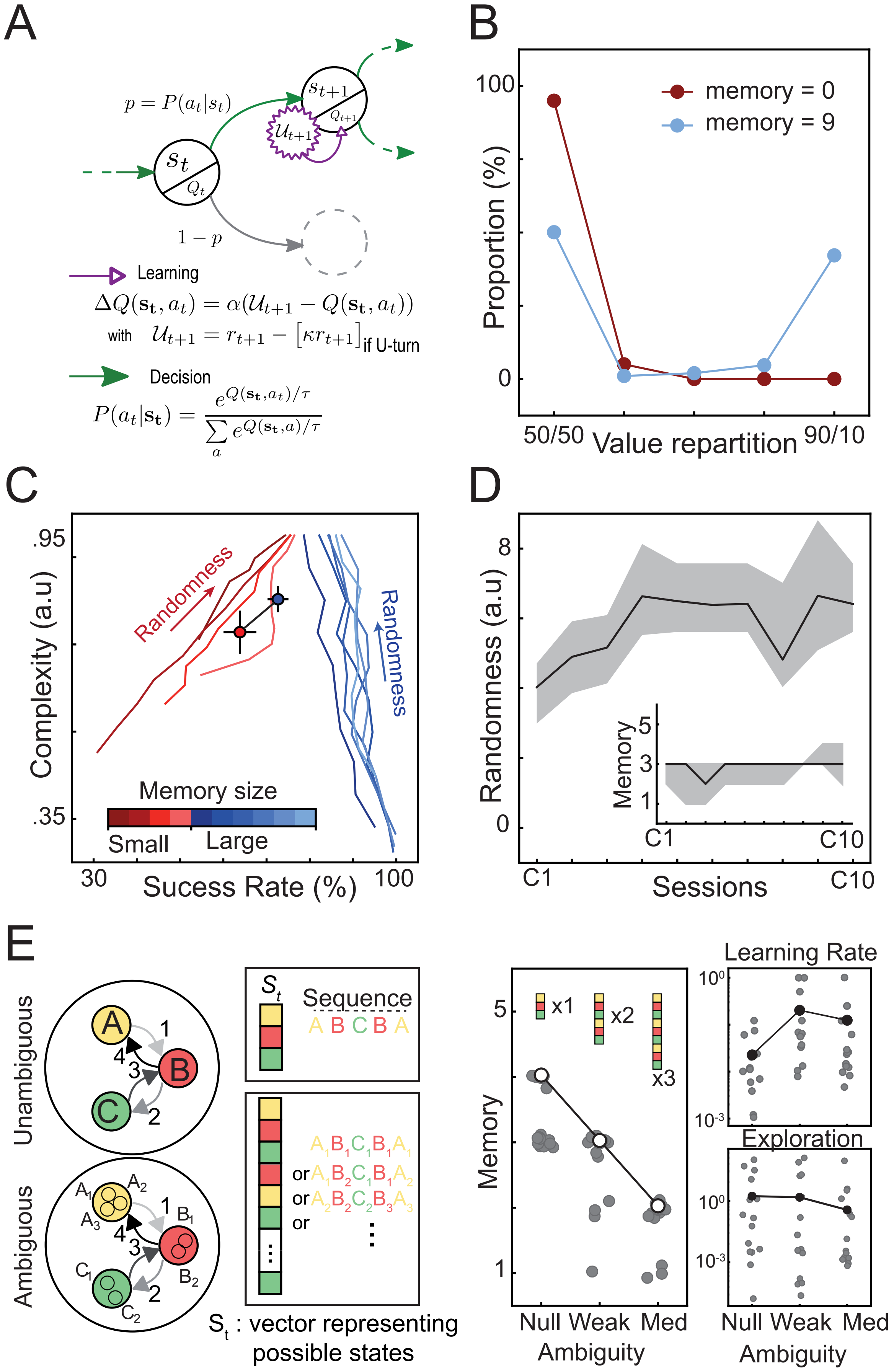

A
?
A
B
C
Reward Delivery Rule:
C1= Cplx (ABCBABCBAB)
C2= Cplx (ABCBABCBAC)
• If C1 > C2 B is reinforced
• If C1 < C2 C is reinforced
• If C1 = C2 Both are reinforced
1 2 3 4 5 6 . 10
A
B C
A C A B
B C A B B C A C
A C A B B C A C A C A B B C A B
BCABBCACACABBCABBCABBCACACABBCAC
25% 25% 50%
H1≠H2
H1=H2
Non reinforced
Reinforced
B
Training (T)
A
B
C
Complexity (C10)
A
B
C
C
Success Rate (%)
80
50
**
ns
80
50
1
10
1
Complexity (C) sessions
10
D
Complexity (a.u)
.95
.55
***
*
n.s.
T C1 C5 C10
RQA.ENT
2.5
1.5
**
*
n.s.
T C1 C5 C10
U-Turns (%)
35
15
*
**
*
T C1 C5 C10
E
Complexity (a.u)
1
.4
C1, r=0.69, p= 0.00006
C10, r=0.61, p= 0.00094
30
Success Rate (%)
90

Fig 1

**Figure 1: Mice generate unpredictable decisions. A)** *Left*, Task setting. Mice were trained to perform a sequence of binary choices between the three target locations (A, B and C) associated with ICSS rewards. In the complexity condition, animals were rewarded when they chose the target which increased the grammatical complexity of the sequence (last 9 choices considered). *Right*, Analysis of all possible combinations of choices in subsequences of size 10. Considering these subsequences by pairs sharing the same first 9 elements, 50% of them are pairs with equal complexity. Only 25% of the subsequences are not rewarded in the complexity condition. **B)** Typical trajectories before and after switching to the complexity condition. In the training phase (before complexity), mice alternated between the equally rewarded targets following a stereotypical circular scheme (e.g. BCABCABC…). At the end of the complexity condition, choice sequence became more variable. **C)** Increase of the success rate over sessions in the complexity setting. Mice improved their performance in the first sessions (c01 versus c05, T = 223.5, p = 0.015, Wilcoxon test) then reached a plateau (c05 versus c10, t(25) = −0.43, p = 0.670, paired t-test) close to the theoretical 75% success rate of random selection (c10, t(25) = −1.87, p = 0.073, single sample t-test). The shaded area represents a 95% confidence interval. *Inset*, grey lines represent linear regressions of individual mice performance progressions for individual mice and the blue line represents the average progression. **D)** Increase of the behavior complexity over sessions. *Left*: the *NLZcomp* measure of complexity increased in the beginning (training versus c01, T = 52, p = 0.0009, Wilcoxon test, c01 versus c05, t(26) = −2.67, p = 0.012, paired t-test) before reaching a plateau (c05 versus c10, T = 171, p = 0.909, Wilcoxon test). The average complexity reached by animal is lower than 1 (c10, t(25) = −9.34, p = 10 −9 , single sample t-test), which corresponds to the complexity of random sequences. *Middle:* the RQA *ENT* entropy-based measure of complexity decreased over sessions (training versus c01, t(26) = 2.81, p = 0.009, paired t-test, c01 versus c05, T = 92, p = 0.019, Wilcoxon test, c05 versus c10, T = 116, p = 0.13, Wilcoxon test). *Right*: The rate of U-turns increased over sessions (training versus c01, t(26) = -2.21, p = 0.036, c01versus c05, t(26) = -3.07, p = 0.004, paired t -test, c05 versus c10, T = 75, p = 0.010, Wilcoxon test). Error bars represent 95% confidence intervals. **E)** Correlation between individual success rate and complexity of mice sequences. Also noteworthy is the decrease in data dispersion in session c10 compared to c1. N = 27 in all sessions except c10 where N = 26. * p < 0.05, ** p < 0.01, *** p < 0.001. n.s., not significant at p > 0.05.


A
$p = P(a_t|s_t)$
$s_{t+1}$
$Q_{t+1}$
$\mathcal{U}_{t+1}$
$s_t$
$Q_t$
$1-p$
Learning
$\Delta Q(\mathbf{s_t}, a_t) = \alpha(\mathcal{U}_{t+1} - Q(\mathbf{s_t}, a_t))$
with $\mathcal{U}_{t+1} = r_{t+1} - [\kappa r_{t+1}]_{\text{if U-turn}}$
Decision
$P(a_t|\mathbf{s_t}) = \frac{e^{Q(\mathbf{s_t},a_t)/\tau}}{\sum_a e^{Q(\mathbf{s_t},a)/\tau}}$
B
memory = 0
memory = 9
Proportion (%)
100
0
50/50
Value repartition
90/10
C
Randomness
Randomness
Complexity (a.u)
.95
.35
Memory size
Small
Large
30
Sucess Rate (%)
100
D
Randomness (a.u)
8
0
Memory
5
3
1
C1
C10
C1
Sessions
C10
E
Unambiguous
A
B
C
1
2
3
4
Ambiguous
$A_1$
$A_2$
$A_3$
$B_1$
$B_2$
$C_1$
$C_2$
1
2
3
4
$S_t$
Sequence
A B C B A
$A_1B_1C_1B_1A_1$
or $A_1B_2C_1B_1A_2$
or $A_2B_2C_2B_3A_3$
or
⋮
$S_t$ : vector representing possible states
Memory
5
1
x1
x2
x3
Null Weak Med
Ambiguity
Learning Rate
$10^0$
$10^{-3}$
Exploration
$10^0$
$10^{-3}$
Null Weak Med
Ambiguity

# Figure 2

**Figure 2: Computational modeling suggests a memory-free pseudo-random selection behind the mice generation of variability. A)** Schematic illustration of the computational model fitted to mouse behavior. As in classical reinforcement learning implementations, state-action values were learned using the Rescorla-Wagner rule and action selection was based on a softmax policy. Two adaptations were applied: i) rewards were discounted by a U-turn cost $\varkappa$ in the utility function in order to reproduce mouse circular trajectories in the training phase (see `Methods'); ii) states were represented as vectors in order to simulate mouse memory of previous choices. **B)** Repartition of the values learned by the model with memory size equal to 0 or 9. With a large memory, the model is able to favor the rewarded target (approx. 90/10 repartition) in the situations where the two options are not equally rewarded (see Fig 1A). With no memory, both actions have about the same value (approx. 50/50 repartition) in almost 100% of the states. **C)** Influence of increased randomness on success rate and complexity for various memory sizes. The randomness hyperparameter is defined as $\boldsymbol{\tau} / \kappa$ . Red and blue dots represent experimental data of mice in the last training and complexity sessions respectively. Error bars represent 95% confidence intervals. **D)** Model fitting results. With an increase of exploration and a small memory, the model fits the improvement in mice performance. The shaded areas represent values of the 15 best parameter sets. Dark lines represent the average randomness value (continuous values) and the best fitting memory size (discrete values) respectively. **E)** Schematic of the difference between ambiguous and unambiguous state representations and simulation results. Top-Left: The main simulations rely on an unambiguous representation of states in which each choice sequence is represented by one perfectly recognized code. Bottom-Left: With more ambiguous states, the same sequence can be encoded by a variety of representations. Right: With higher representation ambiguity, the model better fits mouse performance with a smaller memory (null, weak and medium ambiguity, H = 27.21, p = 10 −6 , Kruskal-Wallis test, null versus weak, U = 136, p = 0.006, weak versus med, U = 139, p = 0.002, Mann-Whitney test) and with a higher learning rate (null, weak and medium ambiguity, H = 7.61, p = 0.022, Kruskal-Wallis test, null versus weak, U = 45.5, p = 0.016, null versus med, U = 54, p = 0.026, weak versus med, U = 101, p = 0.63, Mann-Whitney test) but a similar exploration rate (null, weak and medium ambiguity, H = 3.64, p = 0.267, Kruskal-Wallis test). Grey dots represent the 15 best fitting parameter sets. White dots represent the best fit in case of a discrete variable (memory) while black dots represent the average in case of continuous variables (temperature and learning rate). N = 15.

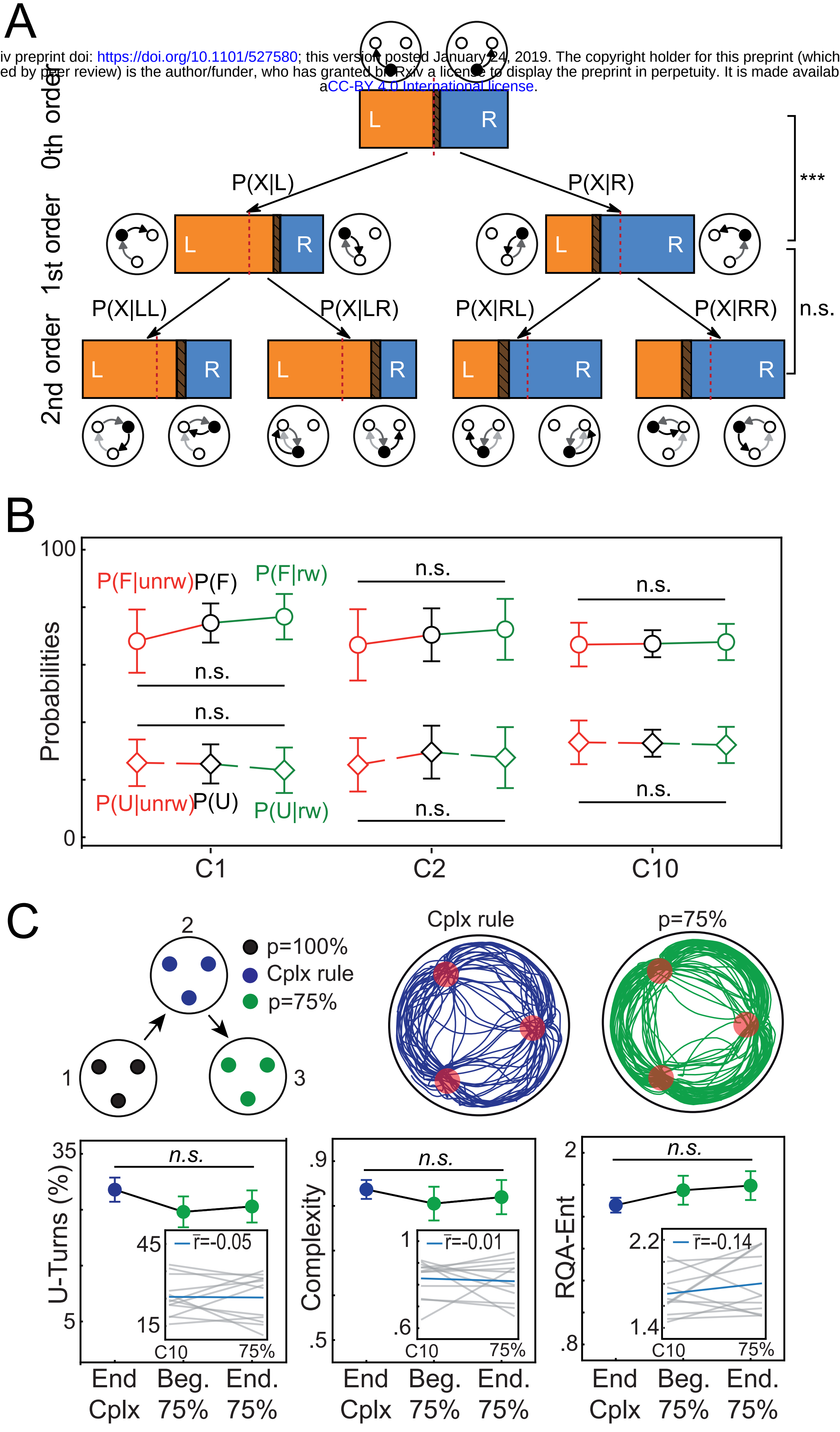

A
0th order
1st order
2nd order
L
R
P(X|L)
P(X|R)
P(X|LL)
P(X|LR)
P(X|RL)
P(X|RR)
***
n.s.
B
Probabilities
100
0
P(F|unrw)
P(F)
P(F|rw)
P(U|unrw)
P(U)
P(U|rw)
n.s.
C1
C2
C10
C
1
2
3
p=100%
Cplx rule
p=75%
Cplx rule
p=75%
U-Turns (%)
35
5
45
15
r̄=-0.05
Complexity
.9
.5
1
.6
r̄=-0.01
RQA-Ent
2
.8
2.2
1.4
r̄=-0.14
C10
75%
n.s.
End Cplx
Beg. 75%
End. 75%

Figure 3

**Figure 3: Behavioral evidence of the absence of memorization in mouse choices. A)** Tree representation of the Markovian structure of mouse behavior in session c10 (N = 26). In the expression of probabilities, P(X) refers to P(L) or P(R), whose repartition is illustrated in the horizontal bars (respectively in orange and blue). Dashed area inside the bars represent overlapping 95% confidence intervals. The probability of a transition (i.e. to the left or to the right) is different from the probability of the same transition given the previous one (p(L) versus P(L|L), t(25) = −7.86, p = 3.10 −8, p(L) versus P(L|R), t(25) = 7.57, p = 6.10 −8, p(R) versus P(R|R), t(25) = −7.57, p = 6.10 −8, p(R) versus P(R|L), t(25) = 7.86, p = 3.10 −8, paired t-test). However, the probability given two previous transitions is not different from the latter (p(L|L) versus P(L|LL), t(25) = 1.36, p = 0.183, p(L|L) versus P(L|LR), t(25) = −1.66, p = 0.108, p(L|R) versus P(L|RL), t(25) = −0.05, p = 0.960, p(L|R) versus P(L|RR), t(25) = −0.17, p = 0.860, p(R|R) versus P(R|RR), t(25) = 0.17, p = 0.860, p(R|R) versus P(R|RL), t(25) = 0.05, p = 0.960, p(R|L) versus P(R|LR), t(25) = 1.66, p = 0.108, p(R|L) versus P(R|LL), t(25) = −1.36, p = 0.183, paired t-test). **B)** Absence of influence of rewards on mouse decisions. P(F) and P(U) respectively refer to the probabilities of going forward (e.g. A→B→C) and making a U-turn (e.g. A→B→A). These probabilities were not different from the conditional probabilities given that the previous choice was rewarded or not (c01, P(F), P(F|rw) and P(F|unrw), H = 2.93, p = 0.230, P(U), P(U|rw) and P(U|unrw), H = 1.09, p = 0.579, c02, P(F), P(F|rw) and P(F|unrw), H = 1.08, p = 0.581, P(U), P(U|rw) and P(U|unrw), H = 0.82, p = 0.661, c10, P(F), P(F|rw) and P(F|unrw), H = 0.50, p = 0.778, P(U), P(U|rw) and P(U|unrw), H = 0.50, p = 0.778, Kruskal-Wallis test). This means that the change in mice behavior in the complexity condition was not stereotypically driven by the outcome of their choices (e.g. ‘u-turn if not rewarded’). Error bars in B represent 95% confidence intervals. N = 27 in c01 and c02 and N = 26 in c10. **C)** *Top-Left:* Schematic illustration of the sequence of conditions in new experiments aiming to assess mice sequence memorization in mice. In the probabilistic condition, choices are rewarded with the same frequency reached at the end of the competitor condition but in a stochastic way. *Top-Right*: Typical trajectories in the complexity condition and in the probabilistic condition. *Bottom*: Comparison of mouse behavior between conditions. The U-turn rate, *NLZcomp* complexity measure and RQA *ENT* measure remain unchanged (pooled ‘end vc’, ‘beg. p75’ and ‘end p75’, U-turn rate, H = 4.22, p = 0.120, Complexity, H = 0.90, p = 0.637, RQA ENT, H = 4.57, p = 0.101, Kruskal-Wallis test). Error bars represent 95% confidence intervals. * p < 0.05, ** p < 0.01, *** p < 0.001. n.s., not significant at p > 0.05.

## Supplementary Figures:

**Supplementary Figure 1: A) Individual comparison with random sequence of choices:** Cumulative distribution of the K parameters calculated for each sequence of choices in sessions Training (T), C1 and C10. For each experimental sequence of choices, the term K=($\mu_o$-<$\mu_s$>)/ $\sigma_s$ was calculated, where $\mu_o$ is the complexity of the original data, <$\mu_s$>) and $\sigma_s$ are the mean and standard deviation of the complexity of the surrogate series(i.e. for each experimental series, 1000 random sequences of the same length in which two consecutive elements could not be equal). We then tested the hypothesis that each original set was different from surrogates. Assuming Gaussian statistics, a limit of K=-2.5 and -1.96 indicates respectively a confidence of 99.4% and 95% that $\mu_o \geq \mu_s$ . We found that 65.4% of mouse sequences in session c10 were not different from surrogates with a confidence of 99.4%. N = 27 in all sessions except c10 where N = 26. **B) Theoretical number of rewards and U-turns in the last 10 choices of a sequence of length 19:** *Left*: Cumulative distribution of the number of rewards obtained in the last 10 choices of the total number of sequences of length 19. Middle: Histogram of the number of U-turns in the last 10 choices. *Right*: Histogram of the number of U-turns in all possible sequences with 100% of reward (top) or 40% of reward (bottom). The optimal U-turn rate to maximize rewards is 50%

**Supplementary Figure 2: Model fitting results for different variants of the model. A)** Comparison between the U-turn rates achieved by a basic RL algorithm, by our model and by mice in the training phase. A basic RL algorithm is unable to reproduce the stereotypical circular trajectories observed in mice during and after training (t(35) = −16.21, p = 10 −17 , Welch t-test; see Figure 1B for mouse trajectories). Indeed, with equal probabilities of reward at all targets, this algorithm learns equal state-action values and randomly chooses between them. Discounting the reward function by a U-turn cost representing previous locations in the state vector (see 'Methods') allows the model to make the same percentage of U-turns as mice in the free condition (t(35) = 0.99, p = 0.32, Welch t-test). **B)**, **C)** and **D)** Comparison of fitness scores with independent and continuous model fitting procedures. 'Independent' refers to the session-by-session hyperparameter optimization, 'continuous' to the optimization of model for all sessions in a row (see 'Methods'). **B)** Best fitness scores obtained with each of the tested memory sizes session by session. Smaller memories fit better. **C)** Comparison of the fitness score obtained by the N = 15 best fits in the independent and continuous fitting procedure. In continuous fitting, three levels of state representation ambiguity are shown (see 'Methods'). Error bars represent 95% confidence intervals. **D)** Success rates and complexity levels obtained in simulations in comparison with those obtained by mice. The shaded area represents the 95% confidence interval for mice. Only best fits are represented for model simulations (average of 20 runs). The model fitted for each session independently follows the evolution of behavioral data thanks to the adaptation of the exploration hyperparameter. In contrast, fitting all sessions with one single parameter set fails to reproduce the same evolution. In particular, the model exhibits a radical increase in success rate and complexity between sessions c01 and c02 as opposed to a more gradual progress in mice.

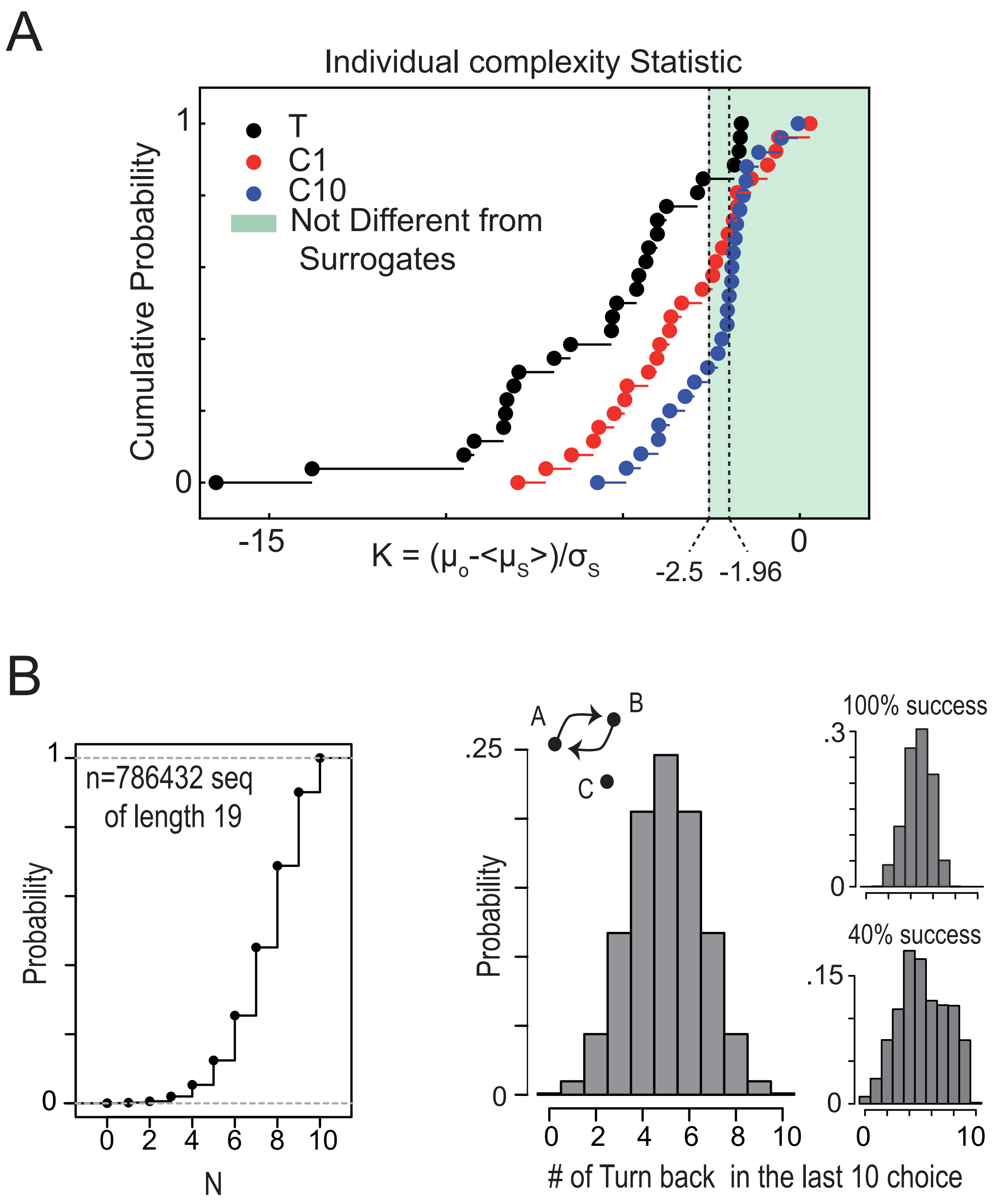
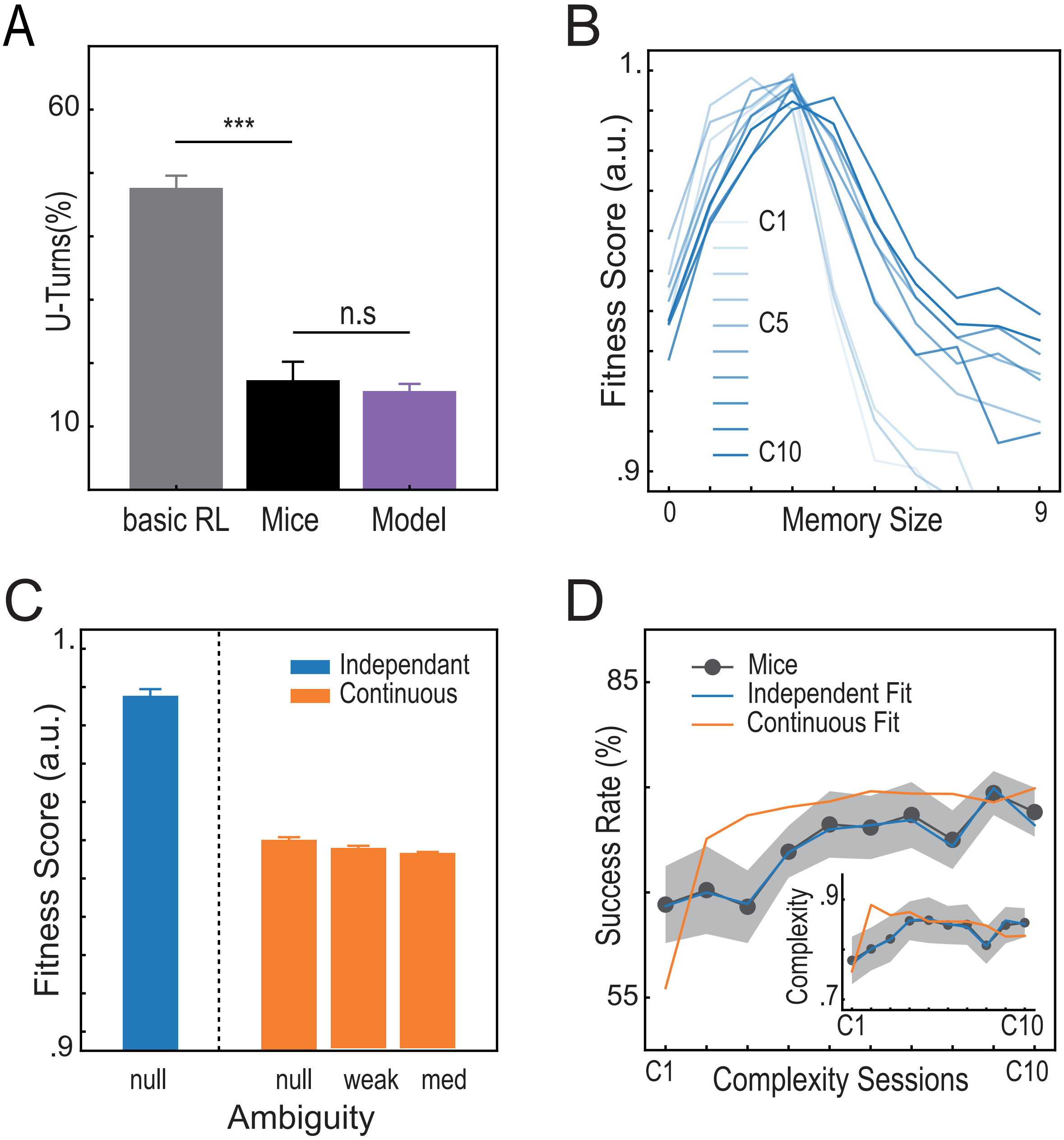


Figure Supp 1


A
60
10
U-Turns(%)
***
n.s
basic RL
Mice
Model
B
1.
.9
Fitness Score (a.u.)
C1
C5
C10
0
Memory Size
9
C
1.
.9
Fitness Score (a.u.)
Independant
Continuous
null
null
weak
med
Ambiguity
D
85
55
Success Rate (%)
Mice
Independent Fit
Continuous Fit
Complexity
.9
.7
C1
C10
C1
Complexity Sessions
C10

Supp Fig 2