

Deciphering value learning rules in fruit flies using a model-driven approach

A Master's Thesis

Rishika Mohanta

Integrated BS-MS

Indian Institute of Science Education and Research Pune

HHMI Janelia Research Campus

$$V_{i+1}(\text{odor}) = \begin{cases} (1 - \alpha_v)V_i + \alpha_v(R_i + \gamma \bar{V}) & \text{if chosen} \\ (1 - \kappa_v)H_i & \text{if not chosen} \end{cases}$$

$$H_{i+1}(\text{odor}) = \begin{cases} (1 - \kappa_h)H_i + \alpha_v & \text{if chosen} \\ H_i & \text{if not chosen} \end{cases}$$

Deciphering value learning rules in fruit flies using a model-driven approach

$$V_i = \sigma(w_V \text{MAD}(V_{1..i}) + w_H \text{MAD}(H_{1..i}) + w_b)$$

Rishika Mohanta

Indian Institute of Science Education and Research, Pune

Work conducted under the guidance of

Dr. Glenn Turner

HHMI Janelia Research Campus, Ashburn



Certificate

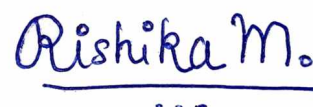
This is to certify that this dissertation titled “Deciphering value learning rules in fruit flies using a model-driven approach” towards the partial fulfillment of the BSMS dual degree program at the Indian Institute of Science Education and Research, Pune represents work carried out by Rishika Mohanta at the Janelia Research Campus under the supervision of Dr. Glenn Turner, Group Leader, HHMI Janelia Research Campus during the academic year 2021-2021.



Dr. Glenn Turner

Group Leader

HHMI Janelia Research Campus



Rishika Mohanta

BS-MS Student

Batch of 2017

Declaration

I hereby declare that the matter embodied in the report entitled “Deciphering value learning rules in fruit flies using a model-driven approach” is the result of work carried out by me at the Janelia Research Campus of the Howard Hughes Medical Institute in Ashburn, VA, USA under the supervision of Dr. Glenn Turner and the same has not been submitted elsewhere for any other degree.



Dr. Glenn Turner

Group Leader

HHMI Janelia Research Campus



Rishika Mohanta

BS-MS Student

Batch of 2017

Table of Contents

List of Figures	6
List of Tables	9
Abstract	11
Acknowledgments	12
Introduction	13
Methods	22
Fly Strains and Rearing	22
Odor Preparation	23
Behavior	24
Rajagopalan (2022) "Fixed Block" Dataset	25
High-Throughput Behavioral Rig	25
Part-by-Part Rig Breakdown	26
Experiment Structure	30
Closed-Loop Image Processing	31
Post-hoc Data Processing	35
Calibrating the Rig	37
Learning Experiments	38
Sample Experiment	38
Mohanta (2022) "Variable Block" Behavioral Dataset	39
Analysis	41
Behavioral Data analysis and modelling	41
Choice Engineering using Q-Learning Models	58
High-Throughput Y-Maze Experiments	61
Statistics	63
Code and Data Availability	63
Results	64

Analysis of Rajagopalan (2022) "Fixed Block" dataset	64
Value learning rules for fruit fly behavior	64
De-novo value learning rule estimation using artificial neural networks	76
Choice engineering for Fruit flies	92
High-Throughput Y-Maze Experiments	98
Optimizing the 16Y experimental setup	98
Mohanta (2022) "Variable Block" dataset	112
Discussion	129
Statistical Tables	138
References	155

List of Figures

Figure 1. Mushroom Body of the Fruit fly (<i>Drosophila melanogaster</i>).	14
Figure 2. Foraging as a 2AFC Task and the limitations of the Matching Law.	17
Figure 3. Reinforcement Learning in the Fly Brain through Value Learning.	19
Figure 4. Choice Engineering Paradigm.	21
Figure 5. High-level schematic of the 16Y high-throughput Y-maze behavioral rig.	28
Figure 6. Schematic of closed-loop control for running parallel experiments.	32
Figure 7. Post-hoc processed variables for high-throughput Y-maze data.	35
Figure 8. LED and Camera Calibration of the 4 Y-arenas used for experiments	37
Figure 9. Design of “Variable Block” 2 Alternative Forced Choice (2AFC) experiments.	40
Figure 10. Map of Cognitive Feature to Model Identity. For a description of models and cognitive features take a look at Table 6 and Table 7.	42
Figure 11. Derivation of the Accept-Reject Policy.	47
Figure 12. Schematic of q-Network Output Symmetrization (qNOS)	55
Figure 13. Open-loop choice engineering using stochastic optimization techniques	59
Figure 14. Q-Learning Models of Rajagopalan (2022) "Fixed Block" dataset reveals that including learning-independent forgetting, perseverance, and temporal discounting in the value update improves the model's explanatory power.	66
Figure 15. Predicted choice probabilities for different models show diminishing differences with more complex models.	67
Figure 16. Q-Learning Models preserve the matching behavior observed in behavior.	72
Figure 17. The dynamics of value underlying different models reveal	74

differences in local variance.

Figure 18. Neural networks can flexibly estimate the value learning rules via imitation learning.	77
Figure 19. Small neural networks can explain fly behavior by estimating the dynamics of changing value of odors.	80
Figure 20. Understanding FFqNs as a conditional first-order discrete dynamical system.	83
Figure 21. Dynamical systems analysis of an asymmetric FFqN reveals a system of unreliable attractors with weak perseverance.	84
Figure 22. Dynamical systems analysis of a symmetric FFqN reveals a system of reliable attractors with stronger perseverance.	85
Figure 23. Dissecting the symmetric RqN reveals a possible separation of timescales that improves the performance of the RqN.	87
Figure 24. Kernel regression analysis to predict the principle components (PCs) of the hidden dynamics reveals the role of nonlinearity in the non-dominant PCs and suggests perseverance behavior.	90
Figure 25. Optimization of choice engineering reward schedules.	94
Figure 26. Choice Engineering provides candidate reward schedules for testing learning rules.	95
Figure 27. Optimal schedules predicted by DF-LT-OS-QL models only show a weak increase in bias than those predicted by F-RL-QL models.	97
Figure 28. Strong Learning and asymmetric preference are observed for 24 hr starved flies in a high-throughput behavioral rig.	99
Figure 29. Slower choices on reward learning are explained by slower movement and residence in the last rewarded arm.	101
Figure 30. Change in odor preference as a function of reward history is a consequence of multiple kinematic factors.	104
Figure 31. PA vs. EL choices show asymmetric, non-specific learning, especially at low reward probabilities, and a naive preference toward EL in 24 hr-starved flies.	107
Figure 32. MHO vs. HAL choices show symmetric learning across starvation states with starvation-sensitive naive preference.	110

Figure 33. Mohanta (2022) “Variable Block” dataset shows probability matching across a broad sample of the space of dynamic baited-reward 2-alternative forced choice tasks.	113
Figure 34. Constrained matching law models can predict future behavior with small integration windows.	115
Figure 35. Logistic kernel regression models perform only as well as the best matching models	117
Figure 36. Results from fitting Q-learning models on Mohanta (2022) “Variable Block” dataset roughly reproduces the results from Rajagopalan (2022) "Fixed Block" dataset.	120
Figure 37. Difference between the parameter estimates from the Mohanta (2022) and Rajagopalan (2022) "Fixed Block" datasets.	125
Figure 38. Results from fitting neural networks to the Mohanta (2022) "Variable Block" dataset also roughly reproduces the observations from Rajagopalan (2022) "Fixed Block" dataset.	127

List of Tables

Table 1. Fly Genotypes used for the experiments.	22
Table 2. Baiting Probabilities for the Rajagopalan (2022) "Fixed Block" dataset.	25
Table 3. Variables used to define each trial for any experiment.	30
Table 4. All saved variables from each fly experiment on the high-throughput rig.	34
Table 5. All processed variables from each fly experiment on the 16Y rig.	36
Table 6. Q-Learning Model Variants used for fitting to data.	44
Table 7. Summary of the significance of cognitive features	45
Table 8. Q-Learning Model Fit Parameters for Rajagopalan (2022) "Fixed Block" dataset.	70
Table 9. Q-Learning Model Fit Parameters Rajagopalan (2022) "Fixed Block" dataset (contd).	71
Table 10. Parameters for constrained matching law models	114
Table 11. Q-Learning Model Fit Parameters for Mohanta (2022) "Variable Block" dataset.	123
Table 12. Q-Learning Model Fit Parameters for Mohanta (2022) "Variable Block" dataset (contd).	124
Table 13. Q-Learning model fit statistics on the Rajagopalan (2022) "Fixed Block" dataset.	139
Table 14. ANOVA summary (Cognitive variables vs. model parameters) for the Rajagopalan (2022) "Fixed Block" dataset.	140
Table 15. Matching law statistics for the Rajagopalan (2022) "Fixed Block" dataset models.	141
Table 16. Local value variance statistics for the Rajagopalan (2022) "Fixed Block" dataset	142
Table 17. Statistics for the comparison of neural networks trained on the Rajagopalan (2022) "Fixed Block" dataset .	143
Table 18. Statistics for the Fly Kinematics.	144

Table 19. Statistics for the Choice Index.	144
Table 20. Statistics for the Learning Index	145
Table 21. ANOVA for effect on learning rate for OCT vs. MCH choices	145
Table 22. ANOVA for effect on learning rate for PA vs. EL choices	145
Table 23. ANOVA for effect on learning rate for MHO vs. HAL choice.	145
Table 24. Statistics for the constrained matching law model fits for the Mohanta (2022) "Variable Block" dataset	146
Table 25. Statistics for Linear Kernel Regression Models for the Mohanta (2022) "Variable Block" dataset	146
Table 26. Parameters and Statistics for C + R.C (30) regression model for the Mohanta (2022) "Variable Block" dataset	147
Table 27. Parameters and Statistics for R + C (30) regression model for the Mohanta (2022) "Variable Block" dataset	148
Table 28. Parameters and Statistics for R + R.C (30) regression model for the Mohanta (2022) "Variable Block" dataset	149
Table 29. Parameters and Statistics for C + R + R.C (30) regression model for the Mohanta (2022) "Variable Block" dataset	151
Table 30. Q-Learning model fit statistics on the Mohanta (2022) "Variable Block" dataset.	152
Table 31. ANOVA summary (Cognitive variables vs. model parameters) for the Mohanta (2022) "Variable Block" dataset.	153
Table 32. Statistics for the comparison of neural networks for the Mohanta (2022) "Variable Block" dataset.	154

Abstract

Navigating the world requires an animal to make choices in a dynamic and uncertain world. Therefore, animals can benefit by adapting their behavior to past experiences, but the exact nature of the computations performed and their neural implementations are currently unclear. Extensive prior knowledge about fruit flies (*D. melanogaster*) provides a unique opportunity to explore the mechanistic basis of cognitive factors underlying decision-making. However, to disentangle between different mechanisms, we require a large number of choice trajectories from single flies. We, therefore, scale-up a Y-maze olfactory choice assay to run 16 flies in parallel to allow us to build and test better models using behavioral perturbation methods such as choice engineering. We take two complementary approaches to explore various learning rules that the fly may use - a model-fitting approach and a novel de-novo learning rule synthesis approach. Firstly, we fit increasingly complex reinforcement learning rules to explain choice. We find that approximating perseverance/habits explains and predicts individual choice outcomes. Next, we develop a flexible framework using small neural networks to infer learning rules and predict choices. We find that small neural networks with less than < 5 neurons trained to estimate odor values can accurately predict decisions across flies better than the best reinforcement learning models. We analyze the functioning of these networks to reveal underlying dynamics that reiterate the presence of perseverance behavior. We successfully reproduce most of our observations across different behavioral setups. Our results suggest that habit-forming tendencies beyond naive reward-seeking may influence flies' choices.

Acknowledgments

First and foremost, I thank my supervisor, Dr. Glenn Turner, for his constant support, invaluable advice, and helpful discussions. His excitement about my work and his willingness to try out new creative directions of questioning have encouraged and inspired me throughout the dissertation. I would also like to thank Adithya Rajagopalan, Ph.D. Student at the Turner Lab, for the enlightening discussions, collection of “fixed block” experimental data, and his technical help with the rig. I also thank Jeff Talbot, Steven Sawtelle, Peter Polidoro, and Tobias Goulet in Janelia Experimental Technology. They patiently helped me build, debug and rebuild the rig multiple times. I must also thank the entire team at FlyCore, without whom none of our experiments would have been possible.

I also thank Dr. Mehrab Modi, Dr. Yichun Shuai, Dr. Karen Hibbard, and all the Turner and Aso Labs members at HHMI Janelia Research Campus. Their kind help and support have made my life at Janelia a wonderful and enriching experience, both personally and academically. I am also grateful to Dr. Kevin Miller, Dr. Maria Eckstien, and Dr. Gregory Wayne from the Google Deepmind team; Dr. Jan Funke, Dr. James Fitzgerald, Dr. Srini Turaga, Dr. Tzushuan Ma, Maanasa Natarajan and Yash Mehta from Computation & Theory at Janelia, and Dr. Collins Assisi from IISER Pune for their mentorship and support.

Finally, I would like to express my gratitude to my close friends in the Apartment B&C, Ethologists Assemble, Füd, and D&D groups, my loving partner Francesca O’Hop, and my amazingly supportive family, especially my late grandfather, who always believed in me even when I did not. Without their tremendous support and encouragement over the past year, it would be impossible for me to complete this study. Finally, I would like to thank the Howard Hughes Medical Institute (HHMI) and Kishore Vaigyanik Protsahan Yojna (KVPY) Fellowship [SB-1712051] for funding this incredible opportunity and the Indian Institute of Science Education and Research (IISER) Pune for the studentship that allowed me to conduct this thesis.

Introduction

Navigating the world often requires an animal to choose between different available actions or stimuli, such as foraging between different patches of food with varying levels of reward (Kamil, 1985; Shettleworth, 1985). This choice is further complicated because rewards are not always certain, and the environment also changes over time (Anselme & Güntürkün, 2019; Kilpatrick et al., 2021). In such situations, animals must also learn to adapt to changed conditions by accumulating information from their past experiences to guide their behavior (Dickinson, 2012; Dukas, 2008; Krebs & Inman, 2015; Mery, 2008). The nature of the computation that animals perform based on their reward and choice history is a field of active study across humans, non-human primates, and rodents (Gadziola et al., 2020; Lak et al., 2020; Rushworth & Behrens, 2008; Schultz, 2016; Sul et al., 2010). However, due to the scale and inherent complexity of vertebrate brains, most studies can only suggest functional similarities between cognitive computations and broad neuronal populations, thus creating a significant divide between the study of neuroscience at the systems level and the study of cognitive principles (Premack, 2007). In contrast, fruit flies (*Drosophila melanogaster*) have an extensive array of genetic tools for monitoring and manipulating single-neuron and population activity (Hales et al., 2015; Oswald et al., 2015; Simpson & Looger, 2018). Further, despite having relatively smaller brains, they have been shown to exhibit a variety of complex behaviors, such as multisensory learning, decision-making, and navigation (Guo & Guo, 2005; Haberkern & Jayaraman, 2016; Yagi et al., 2016). This, combined with the detailed connectomics knowledge (Li et al., 2020), allows us to map cognitive function to neuronal circuitry mechanistically.

The fruitfly mushroom body (MB) (Figure 1. A) has been well-characterized to encode odor valence dynamically via dopamine-modulated synaptic plasticity (Aso & Rubin, 2016; Heisenberg, 2003; Hige et al., 2015). Odor is represented as a sparse combinatorial code in the ~2000 Kenyon cells (KCs) and activates the downstream mushroom-body output neurons (MBONs) that have been shown to trigger upwind walking or turning at odor boundaries (Aso et al., 2014) (Figure 1. B). The strength of the synapses between KCs and MBONs is mediated by dopamine released by dopaminergic neurons (DANs) that receive input from reward (e.g., sugar) or punishment (e.g., bitter/shock) sensory neurons. Dopamine released by the DANs

either depresses or potentiates the KC to MBON synapses depending on the timing of the release relative to KC activity (Handler et al., 2019) therefore modifying the tendency to walk upwind (appetitive behavior) or turn (aversive behavior) when the KCs are activated by odor stimulus. Reward-sensitive DANs are paired with aversive behavior-inducing MBONs and vice-versa. When KC activation (i.e., odor encounter) precedes dopamine release (i.e., odor outcome), KC-MBON synapses are depressed, and when the order of KC activation and dopamine release is reversed KC-MBON synapses are potentiated. Therefore, reward exposure reduces aversive behavior, and punishment decreases appetitive behavior.

Beyond the above highly simplified description, however, MBONs provide direct feedback to other MBONs and connect to DANs directly and indirectly via downstream interneurons. These feedback connections can potentially be used to perform complex computations that can regulate the amount of dopamine released in an atypical non-linear fashion and update the learned behavioral response to odor encounters. Therefore, the fruit fly mushroom body is a potential site for implementing complex behavioral foraging strategies.

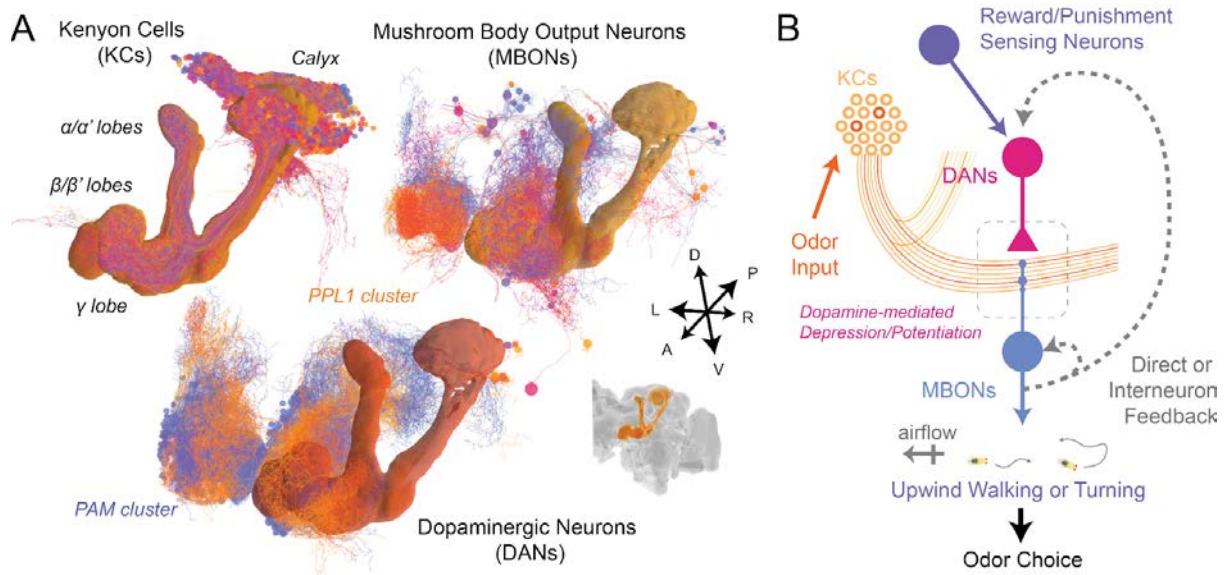


Figure 1. Mushroom Body of the Fruit fly (*Drosophila melanogaster*).

The three main neuronal cell types of the fruit fly mushroom body: i. Kenyon cells receive input from upstream odor circuits at the calyx; ii. Mushroom body output neurons that receive input from the Kenyon Cells; iii. Dopaminergic neurons (typically reward-sensitive PAM cluster and punishment-sensitive PPL1 cluster) that modulate KC-MBON synapses.

(A) 3D neuron reconstruction of mushroom body neurons from Hemibrain v1.2.1 electron microscopy dataset rendered using navis 1.3.1 and plotly.

(B) A simplified circuit schematic for the fruit fly mushroom body. Yellow: Inactive KCs; Orange: Odor-activated KCs; Purple: Reward/Punishment sensing neurons; Pink: Reward/Punishment sensitive dopaminergic neurons; Blue: Aversive/Appetitive MBONs; Grey: Direct/Indirect feedback connections. Note that each “neuron” in the schematic represents a population of neurons shown in subfigure A.

In contrast to the detailed knowledge about the MB anatomical structure, few studies (Rajagopalan et al., 2022; Seidenbecher et al., 2020) have investigated the scope of cognitive processes that underlie the olfactory choice behavior of fruit flies in dynamic probabilistic contexts. Rajagopalan et al., 2022 established a general foraging assay in flies using a two-alternative forced choice (2AFC) task, which has ethological significance across animal taxa (Figure 2. A–C). The authors show that fruit flies, when faced with probabilistic rewards that change with time, show operant matching behavior much like pigeons, monkeys, and honeybees (Greggers & Menzel, 1993; Herrnstein, 1961; Lau & Glimcher, 2005; Sugrue et al., 2004) (Figure 2. C). This observation suggests that the fundamental strategies for foraging are likely to be broadly conserved across the animal kingdom. Therefore, insight into how the fly behaves can give us a valuable understanding of fundamental computations underlying foraging behavior. Since the neural anatomy and mechanism of synaptic plasticity are relatively well understood and experimentally tractable in flies, such insight will allow us to bridge the current gap between what we know about memory formation through plasticity and how the learned associations are utilized during behavior, i.e., the ‘learning rules’.

While Rajagopalan et al., 2022 shows that a reward expectation-based learning rule is necessary to produce any operant matching behavior in fruit flies, many other different factors can contribute to the learning rules that a fly utilizes. These include but are not limited to an increase in valence on reward association (Rescorla & Holland, 1982), forgetting older experiences (Davis & Zhong, 2017; Gonzalez et al., 1967), and either persistence or increased exploration following reward omission (Beckmann & Chow, 2015; Beron et al., 2022; Costa et al., 2016; Hermoso-Mendizabal et al., 2020). Furthermore, operant matching as a ‘strategy’ is a very low-dimensional statistic to describe the behavior that throws away information about short-term variations in choice. Figure 2.D illustrates a toy example where two fundamentally different strategies show the same operant matching outcome (Figure 2.D; rows). Fly 1 & 2 both continue to choose an odor as long as it is rewarded; however, Fly 2 switches the preferred odor every time the expected reward is not delivered (omission-averse strategy). Fly 1 continues to choose the odor after the first reward omission and switches at the second omission (persevering strategy) (Figure 2.D; red arrows). While both flies produce different

choice sequences (Figure 2.D; column 2, rows 2 and 3) for the same reward sequence (Figure 2.D; column 2, row 1), the choice ratio and reward ratios are approximately equal for both flies. Thus, the matching law fails to capture the intricacies of the behavior, warranting the need for a more nuanced theoretical framework to understand the foraging behavior in flies.

Reinforcement Learning (RL) (Sutton & Barto, 2018) presents one such flexible theoretical framework to understand foraging behavior. The RL framework divides the world into two elements: the Agent and the Environment, which interact using actions, observable states, and rewards (Figure 3. A). Under the RL framework, the agent's goal is to use the observed states and past experiences to find the best actions to maximize rewards. Animals can be seen as analogous to RL agents, where in foraging settings, maximizing rewards provides a survival advantage. Over the last decade, there have been many attempts to use various different Reinforcement Learning (RL) algorithms that incorporate different cognitive factors to explain choice behavior across animal taxa. Such model comparison approaches have helped identify possible underlying processes (Niv, 2009; Shteingart & Loewenstein, 2014; Zhang et al., 2019). Algorithms that learn by iteratively updating the state-action values (Q) (which we refer to as 'Value Learning') (Sutton & Barto, 2018, Chapter 6) provide a particularly exciting approach that can directly be mapped to the circuitry of the fruit fly mushroom body (Figure 3. B–D; see caption). We explore different variations of value learning rules (Figure 3. E) to explore how fruit flies behave in the foraging task described in Rajagopalan et al., 2022 using a model-comparison approach. We also develop a novel method that exploits the universal function approximation property of neural networks to estimate the Q -update rule directly from the behavior.

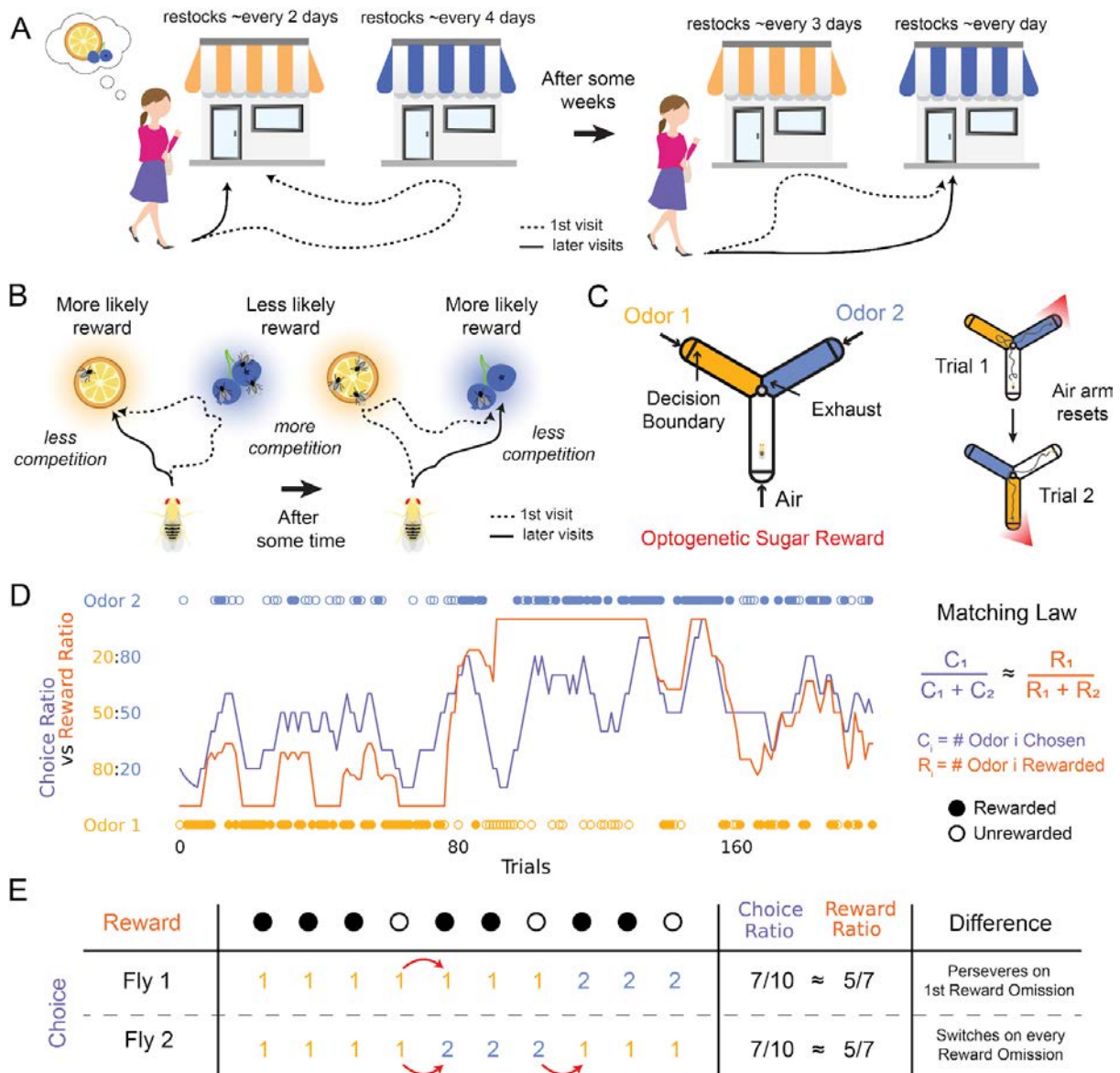


Figure 2. Foraging as a 2AFC Task and the limitations of the Matching Law.

(A) Humans face foraging challenges in daily life. Consider someone looking for fresh fruits but have two comparable options for grocery stores that they can visit, but the two stores restock supplies at a different (unknown) frequency. Therefore, the probability of finding fresh fruits will differ between the two stores and can be estimated by the person after a few visits allowing them to make better decisions about which store to visit. However, the restocking frequency might change after a few weeks, and the person has to update their estimates to make the best choices.

(B) Flies, too, can face dynamically changing reward probabilities during foraging as they might have to compete with other individuals for limited resources. For example,

consider a fly with two possible food sources: lemons and blueberries. A naive fly (dotted line) visits the lemons to find many competitors and receives a reward with low probability. Then, on finding the blueberries learns that the blueberries have fewer competitors and more probability of reward (solid line). As more flies do the same, the distribution of competitors changes, and the fly must learn to switch to the lemons for more reward.

(C) The decision-making process underlying foraging can be replicated in an artificial Y-maze with two odorized and one clean-air arm. Each trial is completed when the decision boundary on an odorized arm is crossed. The relative orientations of the odor arms are randomized to ensure flies do not learn directional associations. A probabilistic reward is delivered through optogenetic activation of sugar-sensing neurons.

(D) Flies show operant matching behavior. Operant matching law is an optimal strategy for foraging where the choices closely follow the same ratio as the rewards received for the different choices (right). Figure reproduced with data from Rajagopalan et al., 2022, with permission. Orange and Blue dots in the reward schedule represent choosing Odor 1 and 2, respectively. Filled and empty dots represent the rewarded choice and unrewarded choices, respectively. The lines represent the reward and choice ratios calculated for 10 trials till the current trial (including the current trial).

(E) A toy example of the limitation of the matching law. See main text. Column 2 provides the reward and choice sequence between odor 1 (orange) and odor 2 (blue). Column 3 shows the estimate of choice and reward ratios. Red arrows highlight transitions in chosen odors.

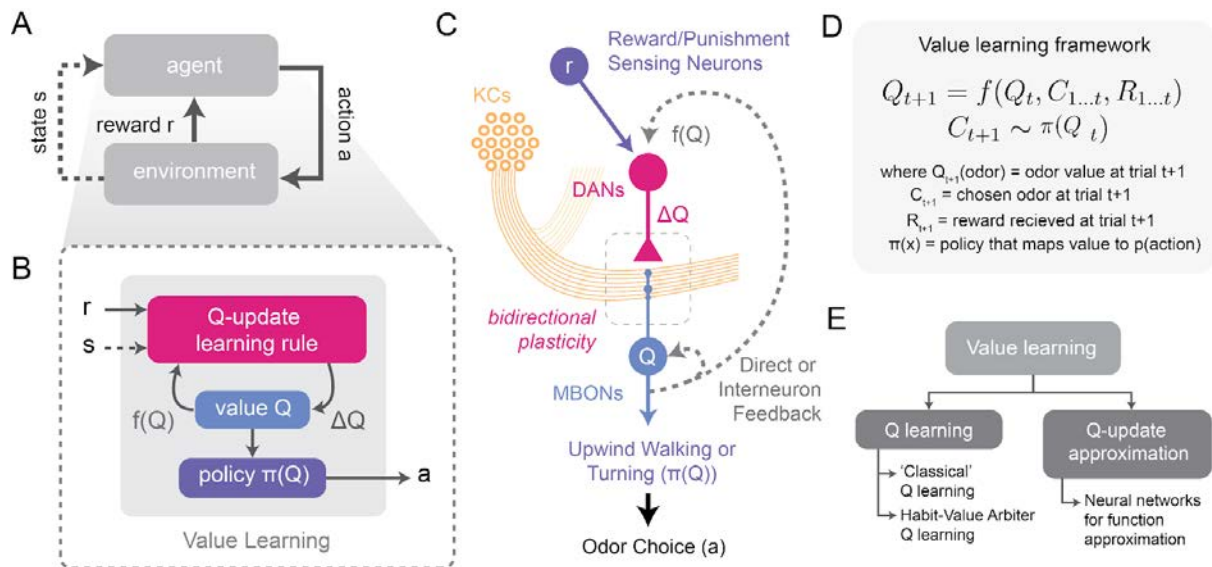


Figure 3. Reinforcement Learning in the Fly Brain through Value Learning.

(A) The Reinforcement Learning (RL) Framework. The agent receives information from the environment in the form of the outcomes for past actions (reward r ; can be +ve or -ve) and the world's current condition (state s ; can be a high dimensional input). Using this information, the agent chooses the best action (a) to perform in order to maximize its reward. In turn, the environment receives the action, updates the state, and gives the appropriate reward to the agent.

(B) Value learning is a type of RL framework that involves three major elements: i. Value (Q) - a measure of how much reward an animal expects given the state and action; ii. Policy ($\pi(Q)$) - a function that transforms the value to a probability of taking any action and determines the action taken by the animal; iii. Q-update Reinforcement Learning Algorithm that updates the value of the state and action, given the information from the environment.

(C) Mapping action value learning to the fruit fly MB. MBON activity during odor exposure encodes stimulus valence and, therefore, can represent the action value of choosing an odor (Q). The $KC \rightarrow MBON$ synaptic weights are updated bidirectionally by DANs, allowing value updation (δQ). DANs receive reward/punishment signals (R) from sugar/bitter/shock-sensing neurons. MB-intrinsic and MB-extrinsic interneuronal circuits can provide complex feedback ($F(Q)$) from the MBONs to the DANs. Thus, DANs can integrate reward signals and feedback to implement complex learning rules. The downstream circuitry then transforms the value code to

behavioral patterns such as upwind walking and turning that result in the choice outcome, i.e., the policy ($\pi(Q)$).

(D) Functional form of the value learning framework for odor preference. The first equation represents how past choices and reward history is integrated with the past value to get the new updated value. The second equation represents how the policy transforms the value into choice distribution.

(E) Variations of value learning. Q-Learning is the most common form of value learning. Over the last two decades, many variations of Q-learning have been developed to explain animal behavior. We divide them into two categories: i. classical Q-learning; ii. habit-value arbiter Q-learning. We also develop a novel modeling framework to infer learning rules from behavior which we call Q-update approximation.

However, while models can be used to explain the variance in the flies' observed decisions, these models may not reflect the actual computations underlying the fly behavior. Therefore, a robust experimental paradigm is required for testing models that predict the dynamics of choice. An often-used method for model identification, especially in systems biology, is to find ways to perturb or 'break' the normal functioning of a system in a predictable fashion using the understanding from a model and observe what happens when the same perturbation is replicated in the natural system (Markowitz, 2010). Taking inspiration, we can translate this principle to the study of cognitive processes by designing experiments capable of predictably perturbing a given model and compare this to whether the same experiment also perturbs the behavior in actual flies. For this purpose, we utilize "choice engineering" (Figure 4.). In this computational framework, the goal is to test how to maximally bias preference between two alternatives in a 2AFC task while keeping the total amount of reward the same for the two choices (Dan & Loewenstein, 2019; Dezfouli et al., 2020).

Next, to improve the quality of our analysis and enable the testing of more complex cognitive principles, we design and develop a high throughput experimental Y-Maze assay that allows the testing of 16 fruit flies simultaneously. We design the rig to be

highly flexible, allowing for easy exploration of the space of choice tasks and providing a deeper understanding of the cognitive principles underlying foraging decision-making in fruit flies. We explore the behavioral basis of the observed choices using the kinematics of the observed fruit fly behavior and find the optimal experimental conditions (such as level of starvation, odor combinations, etc.) for testing dynamic choice behavior in fruit flies. With this assay, we then collect a richer dataset of fruit fly behavior in a randomized sample of the space of possible tasks. We manage to reproduce the results of the previous experiments, thus providing us with solid evidence of habitual behavior in fruit flies.

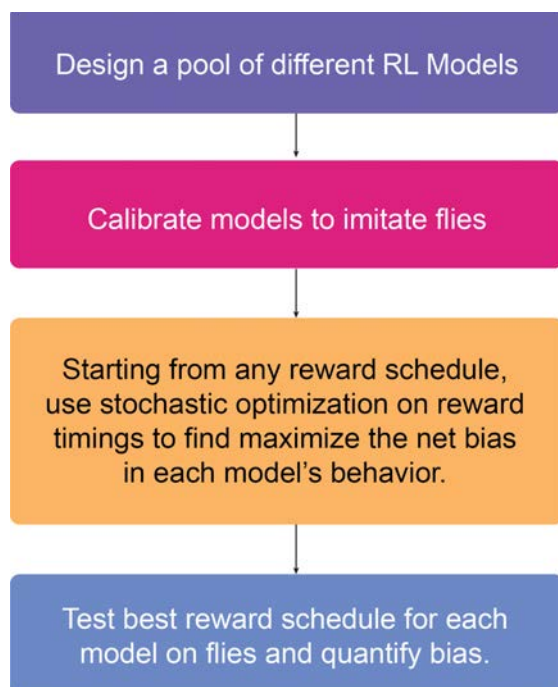


Figure 4. Choice Engineering Paradigm.

A *reward schedule* is a series of choice-reward outcomes for both odors that the fly can choose. The 'optimal' reward schedule for choice engineering is the series of odor-reward associations that maximizes the number of choices made towards a preferred side, providing a predictable behavioral perturbation that can be tested on fruit flies. We sample the space of the reward structures using stochastic optimization techniques to find the optimal reward schedule.

Methods

Fly Strains and Rearing

All *Drosophila melanogaster* used for the experiments (Table 1) were reared at Janelia in plastic vials containing standard cornmeal media supplemented with 0.2 mM (1:500) all-trans-retinal. The vials were stored in incubators and kept in complete darkness at 21°C and 60% relative humidity. The following crosses were set up for the experiment:

Genotype	Expression Target	Reference
w; Gr64f-Gal4/CyO; Gr64f-Gal4/TM3 x 20XUAS-CsChrimson-mVenus attp18	Channelrhodopsin CsChrimson expressed in sucrose-sensitive Gr64f neurons for optogenetic reward delivery	(Haber Kern et al., 2019)

Table 1. Fly Genotypes used for the experiments.

Fly crosses were flipped every 2-3 days for three weeks to prevent vials from becoming overcrowded. Within 24 hrs of eclosion, the progeny were transferred to fresh vials with cornmeal media supplemented with 0.4 mM (1:250) all-trans-retinal for 2-4 days. The progeny were then sorted to identify the correct phenotypes by cold anesthesia of around 1-3°C on a cold plate, and females of the appropriate genotype were transferred to starvation vials. Starvation vials contained nutrient free 1% agarose to prevent desiccation. Flies were starved between 4-13 hrs / 13-28 hrs / 28-37 hrs / 51-64 hrs before being aspirated into the Y-arena for experiments. Cornmeal food (10l) was prepared by boiling a solution of 59.66 g agar (fly agar, Tic Gums Inc, Belcamp, MD, USA) in 7.23 l water. A cornmeal and yeast mixture was then prepared using 664.84 g cornmeal (Quaker Yellow Corn Meal, Quaker Oats Company, Chicago, IL, USA) and 160.68 g yeast (inactive dry yeast, Genesee Scientific, San Diego, CA, USA) added to 1.59 l of water, which was then added to the boiling agar. 0.4 l of molasses was added and allowed to simmer. After cooling, 42 ml of Propionic acid, 79.55 ml Tegosept antifungal agent (Genesee Scientific, San Diego, CA, USA), and diluted amounts of 100mM all-trans-retinal (in ethanol) stock were added to the required concentration for different retinal concentrations in the food.

Odor Preparation

The following odors were prepared and used for the single-fly Y-Maze experiments:

1. 4-Methyl-cyclo-hexanol (MCH) [Sigma-Aldrich 153095] diluted in paraffin oil [Sigma-Aldrich 18512] at a 1:500 (v/v) concentration and then air-diluted to a fourth of this concentration.
2. 3-Octanol (OCT) [Sigma-Aldrich 218405] diluted in paraffin oil [Sigma-Aldrich 18512] at a 1:500 concentration and then air-diluted to a fourth of this concentration.

Both odors were replaced with freshly prepared odor solutions every ten days.

The following odors were prepared and used for the high-throughput Y-Maze experiments:

1. 4-Methyl-cyclo-hexanol (MCH) [Sigma-Aldrich 153095] diluted in paraffin oil [Sigma-Aldrich 18512] at a 1:500 (v/v) and 1:250 (v/v) concentration. (Both concentrations are replaced once a week.)
 2. 3-Octanol (OCT) [Sigma-Aldrich 218405] diluted in paraffin oil [Sigma-Aldrich 18512] at a 1:500 (v/v) and 1:1000 (v/v) concentration. (Both concentrations are replaced once a week)
 3. Pentyl Acetate (PA) [Sigma Aldrich 109584] diluted in paraffin oil [Sigma-Aldrich 18512] at a 1:10000 (v/v) concentration. (Replaced every day)
 4. (–)-Ethyl Lactate (EL) [Sigma Aldrich E34102] diluted in paraffin oil [Sigma-Aldrich 18512] at a 1:10000 (v/v) concentration. (Replaced every day)
 5. Hexanal (HAL) [Sigma Aldrich 115606] diluted in paraffin oil [Sigma-Aldrich 18512] at a 1:1000 (v/v) concentration. (Replaced every two days)
- 6-Methyl-5-hepten-2-one (MHO) [Sigma Aldrich M48805] diluted in paraffin oil [Sigma-Aldrich 18512] at a 1:1000 (v/v) concentration. (Replaced every two days)

Behavior

All data used in this thesis were collected using two different behavioral rigs. Data from a single fly Y-maze rig (Rajagopalan et al., 2022) was used for the model fitting and reward schedule testing. A high-throughput 16-fly Y-maze rig was developed and calibrated during the period of this thesis. It was then used for the rest of the experiments. Both rigs were designed and fabricated by the Janelia Experimental Team (JET, HHMI Janelia Research Campus) with the guidance of Adithya Rajagopalan (Ph.D. Candidate, Turner Lab, HHMI Janelia Research Campus) and Rishika Mohanta (Research Technician, Turner Lab, HHMI Janelia Research Campus). All experiments are done with no light source present to avoid the influence of visual place learning.

Baiting/"Reward-Hold" as a model of naturalistic foraging

In all our behavioral experiments, the reward delivery utilizes 'baiting' or 'reward-hold', i.e., if a reward is randomly drawn for one odor on a particular trial, the reward persists for that odor until it is chosen. Intuitively, baiting provides a model of foraging where the probability of reward depends on the animal's choices. Baiting ensures that the effective reward probability on the unchosen arm increases over time. If the animal continues to exploit a single choice, it forgoes a higher chance of reward on the other odor, even if the other odor is not as frequently rewarding. Thus, baiting promotes exploration as a better strategy (Bari et al., 2019; Sugrue et al., 2004).

Suppose the animal has a reliable estimate of the probabilities of reward delivery for all options without baiting. In that case, the optimal solution is probability maximization (also known as overmatching), where the animal continues to exploit the best option. However, probability maximization is rarely observed in animals including in supposedly "rational" humans. Under a baiting paradigm, the optimal strategy is known to be more widely observed probability matching. In nature, such a scenario may be observed by a fruit fly where the rewards become ready at different rates, such as fruits (beneficial as food/oviposition) falling from trees at different rates that remain available for the fly until exploited.

Rajagopalan (2022) "Fixed Block" Dataset

The dataset consists of the reward and choice history of 21 flies for ~240 trials divided into three blocks. The number trials in each block was kept fixed at 80 trials for all experiments. Across each block, the more rewarded odor is switched. The reward baiting probabilities are varied such that the total probability (reward gain) across the odors is always equal to 1 (10 flies) or 0.5 (8 flies) (Table 2). The dataset was then divided into a training dataset (18 flies) and a test dataset (3 flies). The training dataset was truncated to 200 trials to ensure an equal number of observed trials for all flies.

High-Throughput Behavioral Rig

To expand on the Y-Maze foraging experiments described in Rajagopalan et al., 2022, we developed a high throughput experimental rig that scales up, parallelizes, and expands on the range of possible experiments for the single-fly Y-maze assay to 16 simultaneous fly experiments. The setup of the experimental rig is described in Figure 5. A breakdown of the different labeled parts is provided below.

FlyID	Block 1	Block 2	Block 3	FlyID	Block 1	Block 2	Block 3
0	0.89/0.11	0.11/0.89	0.67/0.33	11	0.06/0.45	0.45/0.06	0.25/0.25
1	0.80/0.20	0.11/0.89	0.89/0.11	12	0.45/0.06	0.06/0.45	0.33/0.17
2	0.20/0.80	0.80/0.20	0.33/0.67	13	0.40/0.10	0.06/0.45	0.45/0.06
3	0.50/0.50	0.89/0.11	0.20/0.80	14	0.33/0.17	0.17/0.33	0.45/0.06
4	0.20/0.80	0.80/0.20	0.89/0.11	15	0.45/0.06	0.10/0.40	0.25/0.25
5	0.67/0.33	0.33/0.67	0.89/0.11	16	0.25/0.25	0.45/0.06	0.10/0.40
6	0.50/0.50	0.67/0.33	0.20/0.80	17	0.40/0.10	0.17/0.33	0.33/0.17
7	0.33/0.67	0.50/0.50	0.67/0.33	18	0.2/0.8	0.8/0.2	0.33/0.67
8	0.20/0.80	0.67/0.33	0.80/0.20	19	0.33/0.67	0.5/0.5	0.67/0.33
9	0.67/0.33	0.20/0.80	0.50/0.50	20	0.4/0.1	0.06/0.45	0.46/0.06
10	0.50/0.50	0.89/0.11	0.33/0.67				

Table 2. Baiting Probabilities for the Rajagopalan (2022) "Fixed Block" dataset.

The two numbers represent the baiting probabilities for OCT and MCH choices. The probabilities in bold are for the holdout test data, and the rest are used for training.

Part-by-Part Rig Breakdown

(1) MFC Assembly: A single 5 liters per minute (LPM) clean airstream is equally split into 16 parallel streams and injected into 16 Whisper-series MCW Mass Flow Controllers (MFC) (Alicat Scientific, Arizona, USA) with a 0.3 liters per minute (LPM) outflow. Using a serial connection, the MFCs were connected to the rig workstation using two daisy-chained Alicat BB9 multi-drop breakout boxes (Alicat Scientific, Arizona, USA).

(2) Humidifier Bottle: Each 0.3 LPM airstream out of the MFCs is injected into a 100 ml PYREX media bottle (Cole-Parmer, Illinois, USA) with an airtight 2-Port GL-45 Cap (CP Lab Safety, California, USA) filled with 50 ml distilled water to humidify the air.

(3) Odor Vial Assembly: The ejected air from the humidifier bottle is split into three equal airstreams using an IDEX 7-port 1/4-28 Low-Pressure Manifold Assembly with alternate outflow blocked using IDEX 1/4-28 Port Plugs (Cole-Parmer, Illinois, USA). The three airstreams were injected into 40ml National Scientific 28X95, 24-400, Amber Vials (Analytics Shop, Georgia, USA) with custom-designed PTFE vial caps (Tru-Plastics, Wisconsin, USA) placed in an in-house custom 3D-printed 12-vial olfactometer assembly.

See: <https://github.com/neurorishika/TurnerLab-4Y-ODA-2022>

(4) Olfactometer: A custom Arduino-based olfactometer is used to redirect airflow from the three odor vials to any combination of downstream arms of a given Y-arena using a nine electromagnetic valve assembly for each set of three odor vials. Sixteen olfactometers (one for each Y-arena) interface with a Raspberry Pi 4 Model B that, in turn, communicates with the rig workstation via a local ethernet connection.

See: https://janelia-kicad.github.io/y_arena_odor_controller/

(5) Y-Arena Chambers and LED Panel: (a) A custom-built airtight Y-arena chamber made of opaque white plastic is used to hold the flies during the experiment. (b) A transparent airtight, depressurized 3D printed chamber. (c) A custom-built 4-quadrant Arduino-based RGB-IR LED Panel is placed below it to provide a

constant outflow of air and a continuous IR backlight to observe the flies in 4 arenas simultaneously. The same LED panel is used to provide red (625nm typical wavelength), blue (465nm typical wavelength) or green (525nm typical wavelength), or a mixture of optogenetic stimulation in any predefined temporal pattern. The LED panel is separated from the Y-arena by a diffuser material. Four LED panels for a total of sixteen arenas connect to the rig workstation using a serial connection.

See: <https://janelia-experimental-technology.github.io/y-arena/>

(6) Flowmeter with Valve: Four depressurized chambers from four Y-arenas are combined and connected to a hard vacuum outlet through an OMEGA FL-2012 Variable Area Flowmeter with the flow rate adjusted to 1.5 liters per minute (which is 1.25x times the total inflow to ensure a drop in pressure in the depressurized chamber despite any leaks)

(7) IR Camera Assembly: A single FLIR BFS-U3-13Y3M-C USB 3.1 Blackfly S Monochrome Camera with a Navitar NMV-8M1 8mm F/1.4 lens and a 55mm GF-X IR72055 720 nm IR long-pass filter (Edmund Optics, New Jersey, USA) were used to image all 16Y arenas simultaneously with a 100 Hz clock. The camera was connected to the rig workstation via USB 4.0, and frames were acquired to Python using FLIR SpinView API.

(8) Workstation: A Dell Precision 5820 Tower with 16-core 3.7 GHz Intel Xeon W-1245 CPU, 64 GB RAM, and 48 GB Nvidia Quadro A6000 GPU was used to interface with the high-throughput rig.

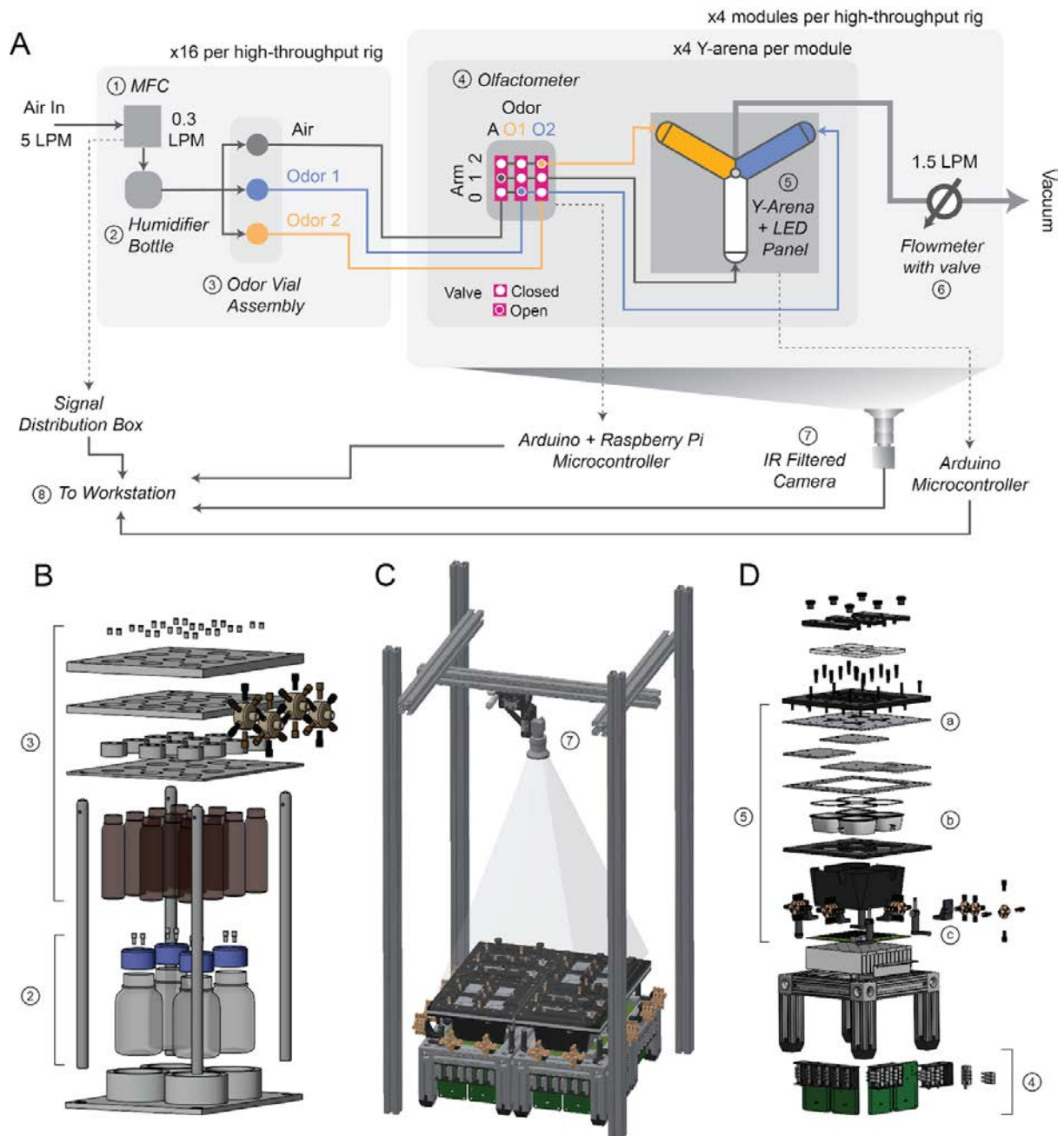


Figure 5. High-level schematic of the 16Y high-throughput Y-maze behavioral rig.

(A) Overall schematic of the 16Y assembly. Four Humidifier bottles (1 per arena) and 12 (3 per arena) were combined as a single ‘Odor Distribution Assembly’. Four Olfactometers and a single LED Panel are combined with four Y-arena chambers to form a single ‘4Y Module’. Four identical 4Y-Modules and Odor Distribution Assemblies combine to form the entire 16Y experimental rig.

(B) Exploded View of an Odor Distribution Assembly. (2) represents the four Humidifier bottles and Holders that supply humidified air for four arenas; (3) represents the Odor Vial Assembly that splits the humidified airstream to create three parallel air streams that can be odorized. Rishika Mohanta (Turner Lab, HHMI Janelia Research Campus, Virginia, USA) designed all parts of the module parts.

(C) Overall Arrangement of the 4Y Module and Camera for the 16Y Setup. The camera (7) is placed on a 3D micromanipulator using support beams over the 4Y Modules. The placement is made such that all 16 Arenas are in a common field of view.

(D) Exploded View of the 4Y Module. (4) represents the four olfactometers for the four Y-arenas; (5) represents the four combined Y-Arena chambers (a,b) and LED Panels (c). All module parts were designed at jET (Janelia Experimental Team, Virginia, USA).

Versilon SE 200 1/8" OD x 1/16" ID and 1/4" OD x 1/8" ID (McMaster Carr, Illinois, USA) tubing were used to direct airflow at all points. The entire rig is placed inside a dark, thermally insulated box in a temperature-controlled room. All experiments are done at 24-26°C.

Experiment Structure

Each experiment is defined using a trial structure during which the odors in different arms are kept fixed. Each trial is defined using 17 variables described in Table 3. The trial variables are provided to the closed-loop controller (see section below) as a predefined CSV file or as a Python function that takes in the trial, choice, and reward history and returns the variables for the subsequent trial.

Typically for two-alternative forced choice (2AFC) experiments, the arm in which the fly started the trial is filled with clean air, and the other two arms have two different odors.

Trial Variable	Description
Trial#	Trial number being described
P(R Air), P(R O1), P(R O2)	(baited/unbaited) reward probability on choosing the air arm, odor 1 arm or odor 2 arm respectively
Odor(Start), Odor(Left), Odor(Right)	odor provided in the trial start arm, left arm and right arm respectively
CStim(Air), CStim(O1), CStim(O2)	reward stimulus file (*.stim [JSON format]) for air reward, odor 1 reward and odor 2 reward respectively (see documentation for <i>sixteeny</i> package)
StayTime(Air), StayTime(O1), StayTime(O2)	how long the fly has to stay in the air arm, odor 1 arm or odor 2 arm to qualify as a reward (in seconds)
Baited	whether or not the trial is baited (1 = baited) (see section on baiting below)
Timed	whether the trial ends after a fixed time (0 = not timed, >0 = trial duration in seconds)
OdorDelay	delay between trial start and flipping odor valve (in seconds)
UStim	reward stimulus file (*.stim [JSON Format]) if the trial is timed (see documentation for <i>sixteeny</i> package)

Table 3. Variables used to define each trial for any experiment.

Closed-Loop Image Processing

A custom GPU-accelerated hardware-software interface package, *sixteeny* was written in Python 3.9 for running any arbitrary experiment on the high-throughput 16Y rig. Each experiment is divided into a trial structure. Trials change after fixed time intervals or at the end of each choice. A *choice* is defined as walking into an arm for more than a predefined fraction ($f = 0.8$ for experiments) of the arm length, therefore, entering the 'reward zone'. The main operational flow for running any experiment is described in Figure 6. A.

On starting an experiment using the 16Y-Experimenter GUI (not shown), the Python script `main.py` is executed. All hardware interfaces (MFC, Odor Valves, LED Panel, and Camera) are initialized, and the airflow rate is verified to ensure within range, after which clean air is allowed to enter all the arms. The sudden airflow start typically startles the flies, causing them to explore the arena, allowing us to effectively calculate a background image by averaging frames over 30 seconds. We do this after the flies are introduced into the arenas to ensure that the tracking is insensitive to small variations in lid placement before and after loading. The saved masks generated using 16Y Mask Designer GUI (not shown) are loaded (Figure 6. B) and are overlaid on the captured background for the experimenter to verify before starting the experiment. All utility objects (*Experimenter* and *ArenaTracker*) are initialized for every fly experiment. *Experimenter* objects are used to determine the subsequent trials based on CSV files or Python script-based experiments. *ArenaTracker* objects are used to track current and past experimental states. The code then starts multiple queues on parallel threads to send email notifications and save frames to disk asynchronously. The main thread operates using three different memory devices. On the camera memory, frames are continuously acquired at 100 FPS. For every processing tick, the latest image is fetched to the GPU memory and processed using NVIDIA GPU-accelerated CuCIM scikit-image package to find each detected blip of activity, which is then passed onto the CPU memory. Using the masks, the most significant blip in each arena is identified and located in different arms and reward regions (Figure 6. B). The experimental states are updated using the fly's location, corresponding rewards are accumulated and executed for all arenas at once, and trials are started by changing valve configurations. As experiments are completed, an email notification is sent, and after the last

experiment is over, all experiment variables are saved (see Table 4); then, the control flow waits till all queues are complete and finally exits.

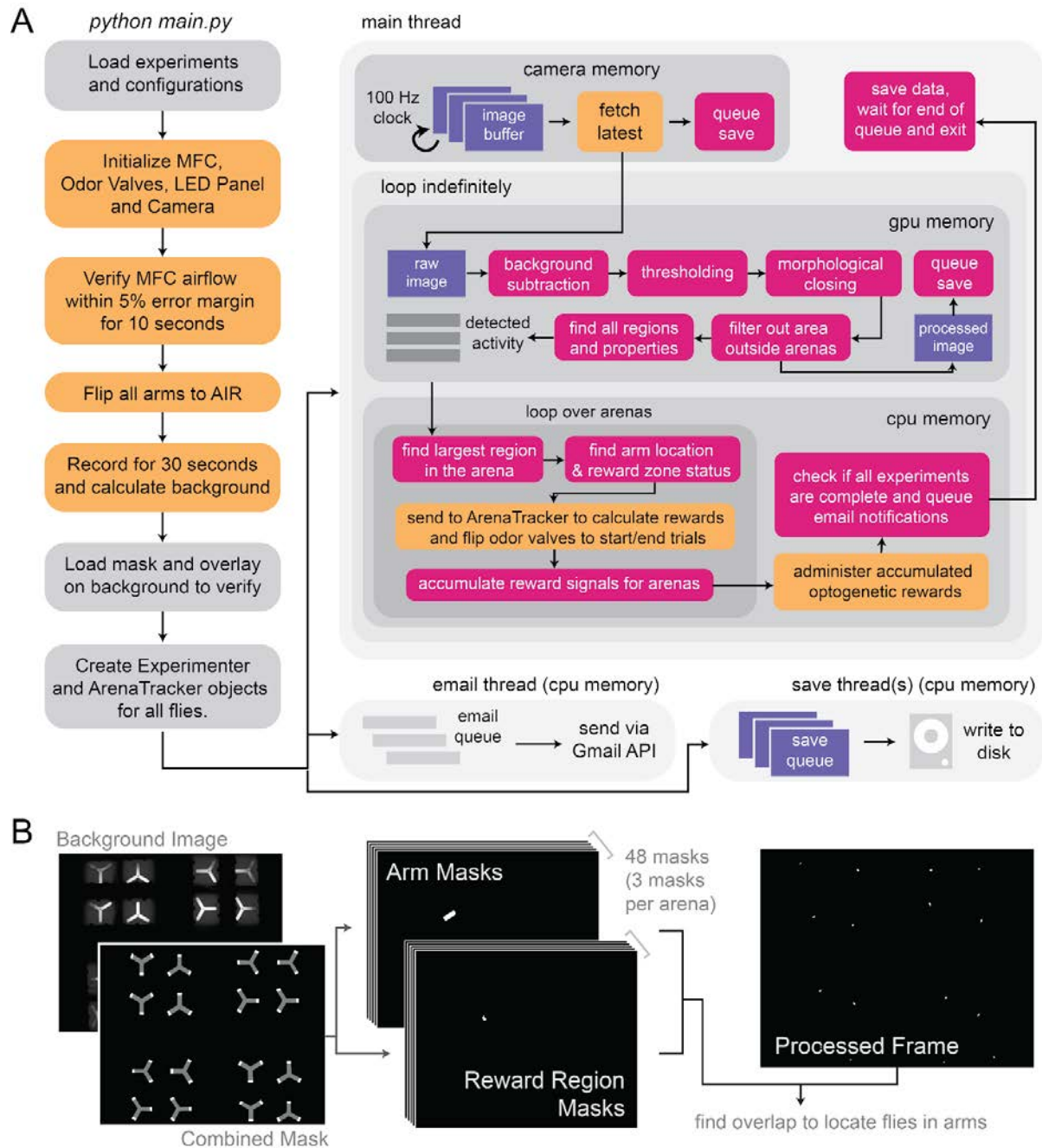


Figure 6. Schematic of closed-loop control for running parallel experiments.

(A) Operational loop for running an experiment using the main.py Python script in the sixteeny Python package. Gray represents preparatory steps, Orange represents hardware interfacing steps, Purple represents image data in the memory, and Pink represents closed-loop control steps. For a summary of the control flow, see the main text.

(B) Schematic of Mask-based Filtering and Localization. To quickly find the current position of every fly in different arms and reward zones, we use a system of 96 masks (48 for reward regions with one per reward zone on each of three arms on 16 arenas & 48 for each of three arms on sixteen arenas). A combined mask (left) can be used to filter activity on the processed frame (right). Looking for overlap between each detected blip and the Arm and Reward Region masks (center) allows us to efficiently identify the location and reward zone status (whether the fly is in a reward zone).

Code Available at: https://github.com/neurorishika/TurnerLab_Opto2AFC_16Y

Data Variable	Description
fly_positions	centroid of the fly (w.r.t. the frame coordinates) in every frame
frame_times	ISO timestamp for the frame
current_arms	arm in which the fly is at every frame
current_trial	trial number for the fly at every frame
current_reward_zone_status	whether or not the fly is in a reward zone in every frame
chosen_arms	relative arm chosen in each trial (1 = left, 2 = right)
chosen_odor	odor chosen in each trial (1 = odor 1, 2 = odor 2)
reward_delivered	whether or not a reward was delivered in each trial (1 = yes)
trial_start_times	ISO timestamp for trial start
trial_end_times	ISO timestamp for trial end
trial_odor_start_delay	how much delay between trial start and flipping odor valve
time_spent_in_reward_zone	how much time was spent in reward zone before reward delivery
length_of_trials	time between start and end of trial
odor_vectors	valve configuration sent to rig for each trial
trial_baited	whether or not the trial was 'baited'
reward_states	whether or not a reward is available after each trial
start_arms	which absolute arm number the trial started in
n_trials	total number of trials to be completed
max_frame_count	maximum frames allowed to be captured
trial_count	total number of trials actually completed
experiment_states	all trial definition variables for the experiment

Table 4. All saved variables from each fly experiment on the high-throughput rig.

Post-hoc Data Processing

The saved data is then processed using a custom 16Y-Data-Processor GUI (not shown) to generate many additional variables to facilitate the analysis of run experiments. The processed variables are described in Table 5 and Figure 7.

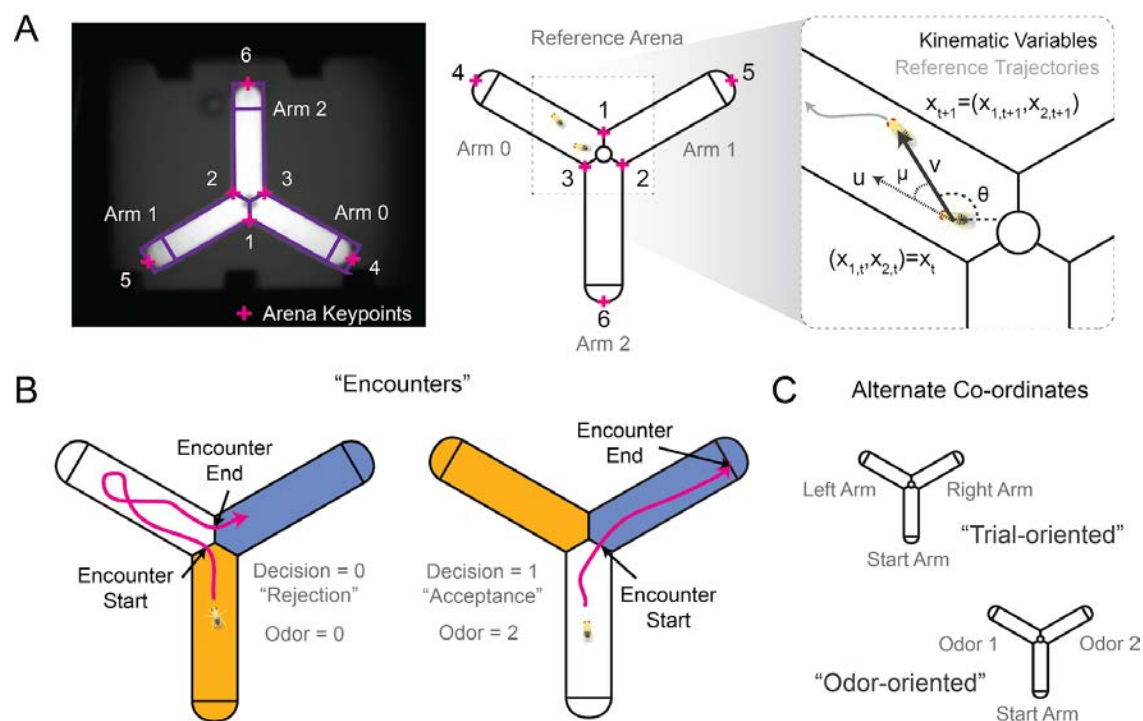


Figure 7. Post-hoc processed variables for high-throughput Y-maze data.

(A) Calculation of a reference coordinate system. Each arena is characterized using six key points: 3 points at the intersection of the arms and the three ends of the arms (left). These key points can be used to generate an affine transform to a reference coordinate system (middle). In the reference coordinate system, different kinematic variables can be calculated from the reference trajectories (right), such as speed (v), the direction of motion (θ), upwind speed (u), upwind motion direction (μ) by using the information about the change in position (x_t). See Table 5 for more details.

(B) *Encounters* boundaries are defined as every time a fly experiences a different odor condition (including air). Boundaries can happen at the end of a trial (right; Encounter "Acceptance") or if a fly enters and leaves with the trial not being completed (left; Encounter "Rejection").

(C) Reference coordinate systems can be reoriented to align them with respect to the start arms and odor positions for easier comparison between trials.

Data Variable	Description
reference_fly_positions	Position of the fly affine transformed into a reference arena layout in every frame (see Figure 7. A). Any missing values where the fly was not visible are linearly interpolated between the last known locations (in inches)
instantaneous_speed	Magnitude of the instantaneous velocity of each fly between frames (in inch/sec) (see 'v' in Figure 7. A). Calculated as $v = \frac{\ x_{t+1} - x_t\ }{\Delta t}$ where $\ \cdot\ $ is the euclidean norm,
instantaneous_motion_direction	Absolute direction of motion in the reference coordinate system between every frame (in radians) (see 'θ' in Figure 7. A) Calculated as $\theta = \arctan\left(\frac{x_{2,t+1} - x_{2,t}}{x_{1,t+1} - x_{1,t}}\right)$
instantaneous_upwind_motion_direction	Direction of motion with respect to the upwind direction between every frame (in radians) (see 'μ' in Figure 7. A) Calculated as $\mu = K - \theta$ where $K = 5\pi/6$ if fly is in Arm 0, $K = \pi/6$ if fly is in Arm 1, $K = 3\pi/2$ if fly is in arm 2. (+ve values = right of wind; -ve values = left of wind)
instantaneous_upwind_speed	Speed of motion in the upwind direction between every frame (in inch/sec) (see 'u' in Figure 7.A) Calculated as $u = v \sin(\mu)$
current_odor	Current odor that the fly is in every frame (0,1 or 2 depending on the odor configuration)
encounter_trial_number	Running count of the encounter that the fly is in (see Figure 7. B for the definition of an encounter)
encounter_odor	Odor that the fly experienced in an encounter (0,1 or 2 depending on the odor configuration)
encounter_durations	Duration of an encounter (in seconds)
encounter_decisions	Whether or not the fly accepted or rejected an odor (see Figure 7. B) (1=acceptance, 0=rejection)
encounter_rewards	Whether or not the fly was rewarded when accepting an odor (1=rewarded, 0=unrewarded)
encounter_start_time	ISO timestamp of encounter start
trial_odor_residence_times	Length of time the fly spend in each odor during a trial (in seconds)
rewarded_frames	Whether or not a frame was rewarded (1=rewarded, 0=unrewarded)
trial_oriented_position	Fly coordinates repositioned such that the arm in which the fly starts is always at the bottom (see Figure 7. C)
odor_oriented_position	Fly coordinates repositioned such that the arm in which the fly starts is always at the bottom and odor 1 and odor 2 are to the left and right respectively (see Figure 7. C)

Table 5. All processed variables from each fly experiment on the 16Y rig.

Calibrating the Rig

In order to ensure the high throughput rig is functional and comparable to the single Y-maze assay, we checked the red (625nm) LED stimulation intensity at different power levels (% maximum input to the LED panel) using a PM100D Compact Power and Energy Meter Console (Thorlabs Inc., NJ, USA) (Figure 8.). We also checked the airflow into every arm when the valve was left open using a handheld FLDA3225C Direct Read Flowmeter (Omega Engineering, CT, USA) to make sure that the inflow was roughly equal to what was expected (data not shown). Closed-loop tracking, valve switching and LED activation was initially checked by manually introducing obstructions over the reward regions in the arenas. Further, flies were introduced and observed during the experiment to see if the LEDs were appropriately triggered when the flies reach the reward zone. Finally, learning experiments were run on the rig to verify the functionality of the Rig.

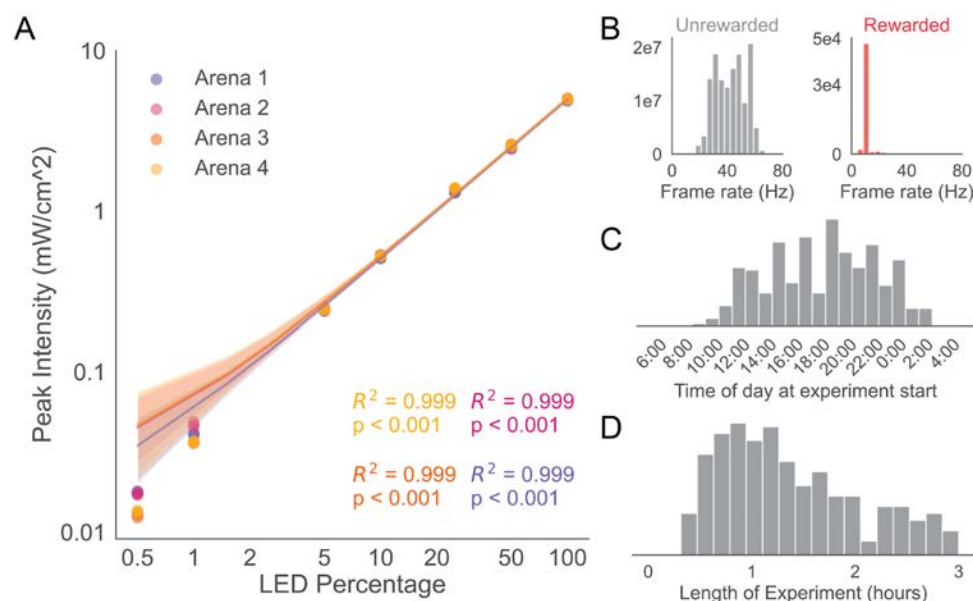


Figure 8. LED and Camera Calibration of the 4 Y-arenas used for experiments

(A) LED Power-Intensity log-log calibration curve comparison across 4 Y-arenas fitted with a linear fit ($p=4.6e-32$, $2.6e-31$, $1.29e-30$, $5.35e-30$).

(B) Frame rate statistics for the experiments for rewarded and unrewarded frames.

(C) Histogram of the time of the day when experiments were run.

(D) Histogram of the duration of each experiment.

Learning Experiments

For the OCT vs. MCH experiments, initially, OCT and MCH were prepared at 1:500 (v/v), but the choices were found to have a strong bias towards MCH in naive trials (data not shown). Since odors are anecdotally known to be aversive at higher concentrations, we tried to balance innate preference by using OCT at 1:1000 (v/v) and 1:250 (v/v) concentrations. All these experiments were performed with 24 hr starved flies, all sixteen Y-arenas active. However, some arenas were found to have leaks in the air supply, and the data was subsequently discarded. Rewards were administered with a 500ms red LED pulse at 25% power (1.35 mW/cm²).

All further experiments were run with a different pair of odors on 4 Y-arena that were verified to be air-sealed, linear with respect to LED power, and fully functional (Figure 8). For PA vs. EL experiments, the odors were at 1:1000 (v/v) concentration. Rewards were administered with a weaker and shorter 250ms red LED pulse at 5% power (0.242 mW/cm²) to reduce the strength of learned association in order to promote dynamic behavior. For MHO vs. HAL choices, we used the same concentration (both at 1:1000 (v/v) concentration) and the same reward configuration as the PA vs. EL experiments but at three different levels of starvation: 4-13 hours, 28-37 hours, and 51-64 hours.

Sample Experiment

A processed video of the sample experiment run on the 16Y High-Throughput Rig can be found here: https://www.youtube.com/watch?v=KypBN_mJscM

Video was generated in Python 3.9.2 using Pillow 9.2.0 and edited by Francesca O'Hop (Freelance Video Editor, VA, USA).

Mohanta (2022) “Variable Block” Behavioral Dataset

While simple associative learning and reversal experiments can inform the underlying basis of behavior, they constrain behavior to remain somewhat stable within each block and showing flexibility in preference only at the sparse number of boundaries between two blocks. As a result, any modeling analysis that might be done to understand this behavior will be biased toward these relatively simple dynamics. Therefore, to effectively sample the space of reward environments that flies can navigate, we need to provide them with a choice environment where reward associations are uncertain and changes happen at range of frequencies, much like naturalistic environments. For this purpose, we design a series of Variable Block experiments (Figure 9.) to sample this space systematically.

We initially ran 1:250 (v/v) OCT vs. 1:1000 (v/v) MCH and 1:1000 (v/v) PA vs. 1:1000 (v/v) EL “Variable Block” experiments at different reward conditions and starvation states. However, we were faced with strong bias effects toward one of the odors that prevented dynamic learning, i.e., MCH and EL, respectively (data not shown). We subsequently ran a set of “Variable Block” 1:1000 (v/v) MHO vs. 1:1000 (v/v) HAL experiments along with their reciprocals (experiments with the odor-reward associations flipped between the two odors) with multiple replicates for a single experiment. Rewards were delivered with a 250ms red LED pulse at 5% power (0.242 mW/cm²).

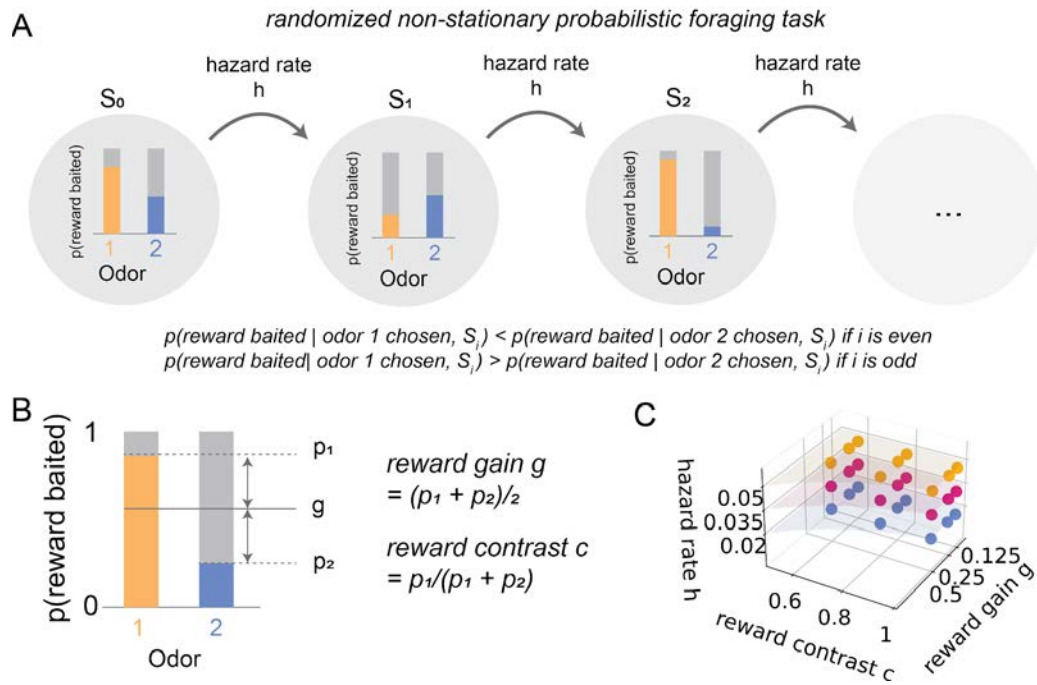


Figure 9. Design of “Variable Block” 2 Alternative Forced Choice (2AFC) experiments.

(A) State transitions in the Variable Block experiments. The state defines the reward-baiting probability for both odors. Markov-state updates happen at the end of each trial. Transitions to the next state (S_i to S_{i+1}) happen with a probability of h (referred to as the hazard rate). Alternatively, the experiment remains in the same state with a probability of $1-h$. A *block* is defined as the trials where the state is conserved. Therefore, the length of a block is a geometric distribution. The odor associated with a greater reward baiting probability is always switched between the two states. Further, we rounded off the sampled block lengths to the nearest 5th trial. Within each state, the rewards are baited (see the section on Baiting above).

(B) Each state is characterized by two values: reward gain (g) and reward contrast (c) which scale the average reward rate and separation of value, respectively. The quantities together define the baiting probabilities for both odors.

(C) All experiments are sampled from the space of reward gain, reward contrast, and hazard rate. The hazard rate is kept constant for a session, but the reward gain and contrast are sampled independently for each block of trials. Reward gain is chosen to be either 0.5, 0.25 or 0.125; reward contrast is chosen to either 1.0, 0.8, and 0.6; Hazard rate is chosen to be 0.02, 0.035, 0.05.

Analysis

Behavioral Data analysis and modelling

Cognitive Q-Learning Models for predicting future choices

The 2AFC task is modeled as a 1-State Markov Decision Process (MDP) using OpenAI Gym 0.22.0 (Brockman et al., 2016) and Python 3.9.7. We create an extensive library of value-learning RL models that combine a diverse set of cognitive principles into the Q-learning framework. The first class of Q-learning rules we consider are the classical Q-learning equations that have the following general functional form:

$$Q_{t+1}(\text{odor}) = \begin{cases} (1 - \alpha')Q_t + \alpha(R_t + \gamma\tilde{Q}) & \text{if chosen and rewarded} \\ (1 - \tau)Q_t + \alpha(\Theta + \gamma\tilde{Q}) & \text{if chosen and unrewarded} \\ (1 - \kappa)Q_t & \text{if not chosen} \end{cases}$$

where Q_t is the “value” of the odor i.e., the expected amount of reward, after the trial t , and R_t is the reward received in trial t . The update terms include (1) an associative increase in value on reward pairing (learning rate: α); (2) reward prediction error strength (error strength: α'); (3) sensitivity to future expectation of reward (discount rate: γ) (Hayden, 2016; Sutton & Barto, 2018); (4) forgetting of learned odor-value over time that may or may not be independent from the learning rate (forgetting rate: κ) (Ito & Doya, 2009); (5) aversion/perseverance on reward-omission (omission sensitivity: Θ) (Ito & Doya, 2009; López-Yépez et al., 2021; Miller et al., 2021) (6) independent time-scales for response extinction (extinction rate: τ) (Goodman & Packard, 2019).

We also test variants of habit-value arbiter models (Miller et al., 2019) that combine estimates of repeated action (habits) by integrating an “Action Prediction Error” (Greenstreet et al., 2022) and estimates of associated reward (value) by integrating a “Reward Prediction Error” to form a combined Q-estimate with the following function form:

$$V_{t+1}(odor) = \begin{cases} (1 - \alpha_v)V_t + \alpha_v(R_t + \gamma\tilde{V}) & \text{if chosen} \\ (1 - \kappa_v)H_t & \text{if not chosen} \end{cases}$$

$$H_{t+1}(odor) = \begin{cases} (1 - \kappa_h)H_t + \alpha_v & \text{if chosen} \\ (1 - \kappa_h)H_t & \text{if not chosen} \end{cases}$$

$$Q_t(odor) = w_t\theta_V V_t + (1 - w_t)\theta_H H_t$$

$$w_t = \sigma(w_V MAD(V_{1...t}) + w_H MAD(H_{1...t}) + w_b)$$

The update equations for this class of models have different learning terms (Table 6 α_h and α_v) and forgetting terms (κ_h and κ_v) for updating stimulus association (V) and habits (H) which are then weighted based on a linear combination of the estimates of variation (Mean Absolute Deviation MAD(x); chosen for numerical stability in Theano) combined with an absolute strength for each variable (θ_h and θ_v) to determine the final action value (Q). This class of models therefore allow not only reward driven computations but dynamically shifting between goal-directed and habit-driven behavior using complex cognitive variables such as reward and action variability.

We finally describe 24 Q-learning variants that combine the different terms in the Q-learning equations that are summarized in Table 6 which together span a space of 8 cognitive features summarized in Table 7 (Figure 9).

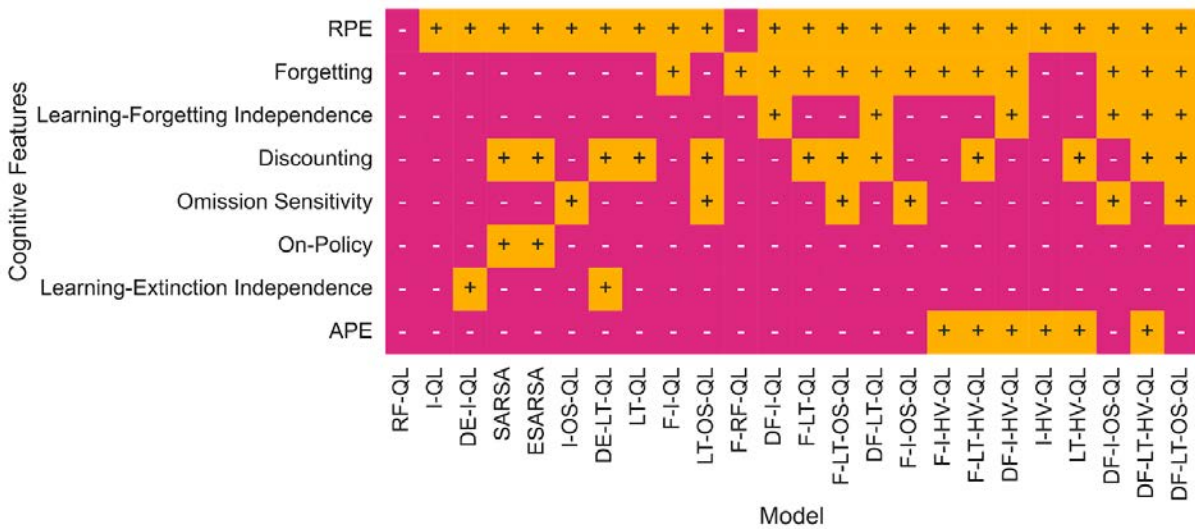


Figure 10. Map of Cognitive Feature to Model Identity. For a description of models and cognitive features take a look at Table 6 and Table 7.

Model	# Params	Abbreviation	Model Constraints
Differential Forgetting Long-Term Habit-Value Arbitrator Q-Learning	10	DF-LT-HV-QL	$\kappa_h = \alpha_h, \tilde{V} = \max(V_t)$
Forgetting Long-Term Habit-Value Arbitrator Q-Learning	10	F-LT-HV-QL	$\kappa_v = \alpha_v, \kappa_h = \alpha_h, \tilde{V} = \max(V_t)$
Long-Term Habit-Value Arbitrator Q-Learning	10	LT-HV-QL	$\kappa_v = 0, \kappa_h = \alpha_h, \tilde{V} = \max(V_t)$
Differential Forgetting Immediate Habit-Value Arbitrator Q-Learning	10	DF-I-HV-QL	$\kappa_h = \alpha_h, \gamma = 0, \tilde{V} = \max(V_t)$
Forgetting Immediate Habit-Value Arbitrator Q-Learning	9	F-I-HV-QL	$\kappa_v = \alpha_v, \kappa_h = \alpha_h, \gamma = 0, \tilde{V} = \max(V_t)$
Immediate Habit-Value Arbitrator Q-Learning	9	I-HV-QL	$\kappa_v = 0, \kappa_h = \alpha_h, \gamma = 0, \tilde{V} = \max(V_t)$
Differential Forgetting Long-Term Omission-Sensitive Q-Learning	6	DF-LT-OS-QL	$\tau = \alpha = \alpha', \tilde{Q} = \max(Q_t)$
Forgetting Long-Term Omission-Sensitive Q-Learning	5	F-LT-OS-QL	$\kappa = \tau = \alpha = \alpha', \tilde{Q} = \max(Q_t)$
Long-Term Omission-Sensitive Q-Learning	5	LT-OS-QL	$\kappa = 0, \tau = \alpha = \alpha', \tilde{Q} = \max(Q_t)$
Differential Extinction Long-Term Q-Learning	5	DE-LT-QL	$\kappa = 0, \alpha = \alpha', \Theta = 0, \tilde{Q} = \max(Q_t)$
Differential Forgetting Long-Term Q-Learning	5	DF-LT-QL	$\tau = \alpha = \alpha', \Theta = 0, \tilde{Q} = \max(Q_t)$
Forgetting Long-Term Q-Learning	4	F-LT-QL	$\kappa = \tau = \alpha = \alpha', \Theta = 0, \tilde{Q} = \max(Q_t)$
Long-Term Q-Learning	4	LT-QL	$\kappa = 0, \tau = \alpha = \alpha', \Theta = 0, \tilde{Q} = \max(Q_t)$
Expected SARSA	4	ESARSA	$\kappa = 0, \tau = \alpha = \alpha', \Theta = 0, \tilde{Q} = \mathbb{E}[Q_t]$
SARSA	4	SARSA	$\kappa = 0, \tau = \alpha = \alpha', \Theta = 0, \tilde{Q} = Q_t[C_{t+1}]$
Differential Forgetting Immediate Omission-Sensitive Q-Learning	5	DF-I-OS-QL	$\tau = \alpha = \alpha', \gamma = 0$
Forgetting Immediate Omission-Sensitive Q-Learning	4	F-I-OS-QL	$\kappa = \alpha = \alpha', \tau = \alpha, \gamma = 0$
Immediate Omission-Sensitive Q-Learning	4	I-OS-QL	$\kappa = 0, \tau = \alpha = \alpha', \gamma = 0$
Differential Extinction Immediate Q-Learning	4	DE-I-QL	$\kappa = 0, \alpha = \alpha', \Theta = 0, \gamma = 0$
Differential Forgetting Immediate Q-Learning	4	DF-I-QL	$\tau = \alpha, \Theta = 0, \gamma = 0$
Forgetting Immediate Q-Learning	3	F-I-QL	$\kappa = \tau = \alpha = \alpha', \Theta = 0, \gamma = 0$

Model	# Params	Abbreviation	Model Constraints
Immediate Q-Learning	3	I-QL	$\kappa = 0, \tau = \alpha = \alpha', \Theta = 0, \gamma = 0$
Forgetting RPE-free Q-Learning	3	F-RF-QL	$\kappa = \tau = \alpha, \alpha' = 0, \Theta = 0, \gamma = 0$
RPE-free Q-Learning	3	RF-QL	$\kappa = 0, \tau = \alpha, \alpha' = 0, \Theta = 0, \gamma = 0$

Table 6. Q-Learning Model Variants used for fitting to data.

Cognitive Feature	Significance
Reward Prediction Error (RPE)	When RPEs are used for updating value, there is a recursive relationship of the value with its update, i.e., it is a linear function of the difference between the received reward and the value itself. It allows for a bidirectional update of the value of the chosen option depending on whether it is more or less than the expected value allowing for more dynamic control of behavior.
Forgetting	Typically in Q-Learning, the value update only happens for the chosen option, while the value of alternatives remains the same. In the presence of forgetting, the value of unchosen options can decrease over time.
Learning-Forgetting Independence	Learning and forgetting can happen at different rates through different pathways; this cognitive feature allows that flexibility.
Discounting	Under (temporal) discounting in Q-Learning, the value is updated not based on immediate expected reward but the total future expected reward based on a decreasing weight further into the future (estimated using the value as an approximation). This feature is handy for n-state Markov decision processes (MDP) where $n > 1$, unlike our two-choice task, but in a 1-state framework, discounting with high discount rates (close to 1) can allow a greater dynamic range of preference as the resulting strategy tends towards probability maximization in the two choice case (Kelly, 1981)

Cognitive Feature	Significance
Omission Sensitivity	This cognitive feature modulated the directionality and strength of the extinction response when a reward is not given. Positive omission sensitivity leads to a preservative attractor, i.e., if a choice is made and reward is omitted, there is an increased preference towards that odor if the omission sensitivity strength is greater than the extinction strength. Alternatively, a negative omission sensitivity pushes the preference toward the other options by reducing the value beyond extinction, thereby promoting switching between options.
On-policy	On-policy learning allows the estimation of the future reward to be dependent on the policy itself rather than a policy-free maximization-based estimate of the "best" possible reward discounted over time.
Learning-Extinction Independence	Q-learning models usually weigh the experienced reward/future rewards equal to the past estimate of the reward. However, it need not be the same. This cognitive feature allows for a difference from calculating the expectation. It allows us to scale the weight of the expectation in the difference allowing for an extinction rate different from the learning rate.
Action Prediction Error (APE)	Action prediction errors allow a continuous leaky estimate of how many times an option has been chosen (habits) mixed with the reward frequency (value) estimate for each option weighted by a function of their variability. APEs promote habitual behavior when the value estimates are stable and reliable.

Table 7. Summary of the significance of cognitive features

There are also multiple ways to model the policy that converts the resultant odor values into odor choices. In our Y-Maze arena, flies can only experience one odor at a time. Our current understanding of the drosophila mushroom body suggests, that while the memories are stored in the KC-MBON synapses consistently, they can only be retrieved by odor exposure triggering KC activation. This implies that during behavior memories are retrieved in a sequential fashion with each memory depending on the current odor. Therefore, their choice behavior is better described as a series of accept-reject decisions rather than a single binary choice (Hall-McMaster & Luyckx, 2019; Hayden, 2016). As a result, some randomness in the final choice outcome emerges purely based on which odor the fly encountered first.

However, this makes modeling exploration behavior using traditional policy definitions such as epsilon-greedy, softmax (Abbott & Dayan, 2001; Sutton & Barto, 2018), or epsilon-softmax (Shteingart et al., 2013) prone to inflated estimates of exploration. This is because these models often assume a linear relationship between value and final probability of choosing an odor. As a result, when the policy parameters are inferred from behavior that might be resulting from a non-linear response function resulting from sequential behavior in a task where the values are constantly changing, a linear approximation gets constrained to the average behavior. As a result, the parameters might not capture strong changes in in preferences, and thus provide inflated estimates of exploration. Thus, to go from odor value to odor choice, we designed a novel policy function which we call the Accept-Reject Policy. The policy function predicts the probability of 'accepting' an odor, i.e., the probability that the fly will walk to the decision boundary after encountering an odor. It uses a linearly scaled sigmoid transform of the odor value estimated using the Q-learning equations and then transforms it to a binary decision outcome.

If $p_{1|a}$, $p_{1|1}$ and $p_{1|2}$ are the probabilities of the fly accepting Odor 1 given it is in the air arm, Odor 1 arm or Odor 2 arm respectively, we can write the following equations from the possible state transitions (Figure 11. A, B):

$$p_{1|1} = q_1 + 0.5(1 - q_1)p_{1|a} + 0.5(1 - q_1)p_{1|2}$$

$$p_{1|2} = 0.5(1 - q_2)p_{1|a} + 0.5(1 - q_2)p_{1|1}$$

$$p_{1|a} = 0.5p_{1|1} + 0.5p_{1|2}$$

Solving these equations for $[\pi(Q)]_i = p_{i|a}$ (since all trials start in air), we can arrive at the Accept-Reject policy function defined as follows:

$$q_i = \sigma(mQ_i + c), \quad i = 1, 2$$

$$[\pi(Q)]_i = \frac{q_i(3 - q_{i'})}{3q_i + 3q_{i'} - 2q_iq_{i'}}$$

$$i, i' = 1, 2 \text{ or } 2, 1$$

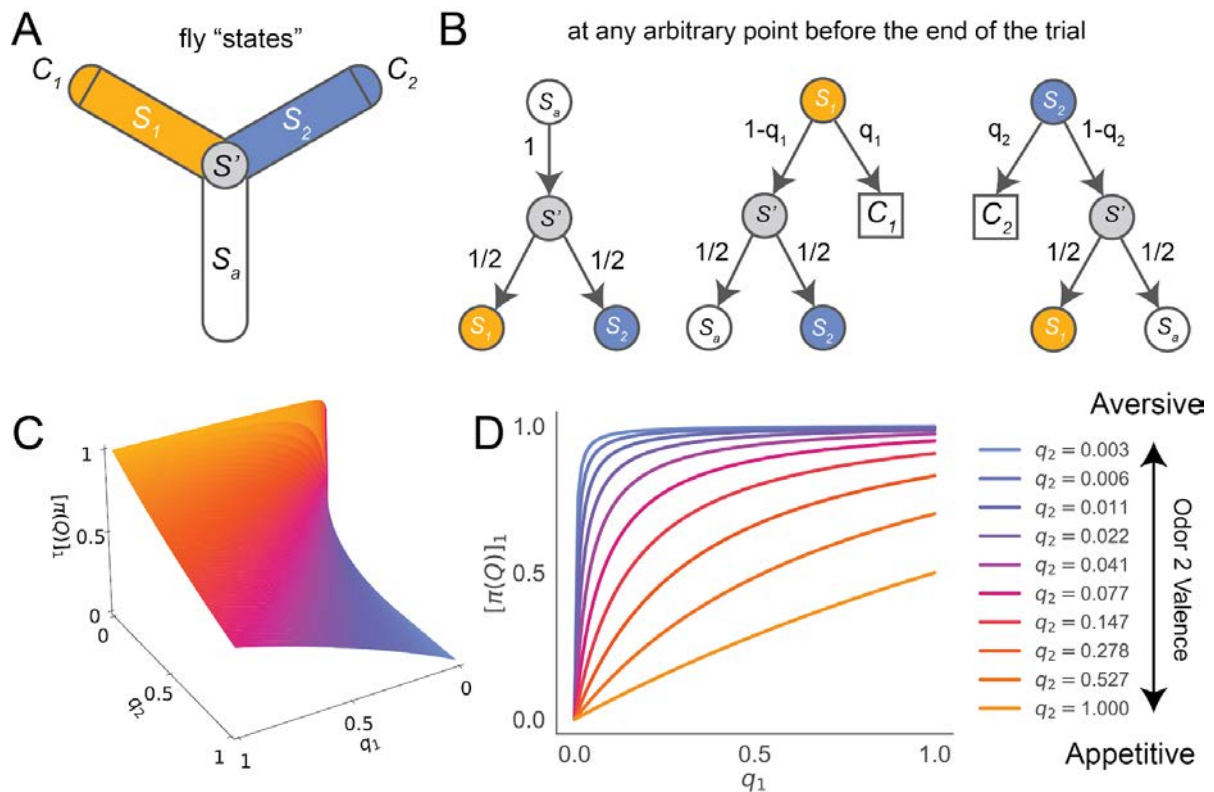


Figure 11. Derivation and Characterization of the Accept-Reject Policy.

(A) At any point in time, the fly can be in one of 4 areas of the Y-Maze, which we define as fly "states," i.e., the air arm (S_a), the odor 1 arm (S_1), the odor 2 arm (S_2) or at the decision boundary (S'). Further, a trial only terminates with a choice state where odor 1 is chosen (C_1) or odor 2 is chosen (C_2).

(B) Let q_i be the probability that odor $i = 1$ or 2 is accepted. Assuming that once the fly reaches the decision boundary, it will necessarily enter one of the two other arms with equal probability, we can define all possible transitions starting from any of the arms. Circles represent the arm in which the fly is. Square represents a choice that leads to the termination of the trial.

(C) A 3D plot of the odor 1 choice probability ($[\pi(Q)]_i; i = 1$) in terms of the acceptance probability of the two odors (q_1 and q_2). Note the non-linear response of the function that allows for both exploratory behavior (choice probability close to 0.5) and greedy behavior (choice probability close to 1) with small changes in acceptance probability.

(D) A cross-section of the policy function at different odor 2 valences. ($q < 0.5$ = Aversive, $q > 0.5$ = Appetitive)

Bayesian Inference of model parameters

We then fitted these models to the experimental data. Bayesian model-fitting was done using a Hamiltonian Monte-Carlo (HMC) based Bayesian sampling NUTS Sampling with non-informative ($Beta(1,1)$) or weakly-informative priors ($HalfNormal(0,10)$) on PyMC3 3.11.4 (Wiecki et al., 2022). We sampled all parameters for 20000 samples (5000 samples across four parallel chains) with 5000 burn-in iterations for each chain. Convergence across four parallel MCMC chains and effective sample sizes were evaluated for all parameters to ensure most values are fully converged (>1.1) and have a high effective sample size (>3000 samples). We then assessed the model quality of fit using the deviance-scaled Watanabe-Akaike Information Criteria (WAIC), which accounts for the effective number of parameters in the model. WAIC typically converges to Leave-One-Out Cross Validation (LOOCV) score, making it very useful for bayesian model comparison (McElreath, 2016; Vehtari et al., 2017).

Estimation of Smoothed Choice Probabilities for visualization

We often need to visualize models' predictions and compare them to the observed behavior. However, it is not intuitive to guess the fit quality by looking at actual choices and comparing them to probabilities. Therefore to help visualize it, we pad the choice probabilities (for odor 2) and the choice sequences (0 = odor 1 and 1 = odor 2) at the start with 0.5 and take a 10-trial sliding window average such that each average value gives us the number of times odor 2 was chosen/expected to be chosen in the past ten trials (including the current trial). This running mean allows us to visualize the average choice probability at every trial. Note that this assumes that changes in probabilities are smooth and not rapid. At the same time, this visualization can help make the results more intuitive; therefore, it should not be used for any calculations as it forces smooth local variability.

We tested the goodness of fit on training data and predictive power of the models on held-out test flies using Normalized Likelihood (Miller et al., 2021), which is defined as:

$$\bar{L} = \exp\left[\frac{1}{T} \sum_i^T x_i \log(\hat{p}) + (1 - x_i) \log(1 - \hat{p})\right]$$

where T is the number of trials, x_i is the observed choice (1 = odor 1, 0 = odor 2), and \hat{p} is the predicted probability of choosing odor 1.

We also compared other experimental observations, such as operant matching, by comparing blockwise log choice odds and log reward odds (Sugrue et al., 2004; Todorov et al., 1983) to estimate matching strength s and bias b based on the following formula:

$$\log\left(\frac{C_1}{C_2}\right) = b + s \log\left(\frac{R_1}{R_2}\right)$$

where C_1 and C_2 are the number of times in a single block (trials with the same probability) where odors 1 and 2 are chosen, respectively; further, R_1 and R_2 are the numbers of rewards associated with each odor within the same block. We simulated 50 replicates of the experiments described in Table 1 using the fitted models. We then estimated the blockwise log choice and reward odds from the simulated data and fitted them using a bootstrapped linear model ($n=1000$ bootstraps) to estimate the confidence intervals for matching strength (s) and the bias (b) using SciPy 1.7.1.

In order to explore the dynamic computations underlying the different models, we observed the choice probabilities and underlying acceptance probabilities for the two odors (estimated by applying the fitted logistic transform on the value predictions). We simulate 1000 replicates of a single random “Variable Block” experiment (see the section on “Variable Block” experiments). To systematically look at the local variance in the acceptance probabilities across models, we simulated 1000 different “Variable Block” experiments for every model, quantified a running standard deviation for a ten-trial window, and took the average for the entire session using NumPy 1.22.3.

Constrained Matching Law models for predicting future choices

To understand how well matching law can predict the behavior, we use a model with four parameters inspired by the generalized matching law: history size (H), matching strength (s) and matching bias (b) and maximum certainty (l_{max}). In this model, we try to predict the the future choice using the estimated reward odds over a past time window of size H (i.e., $\log(R_{1,t-H:t}/R_{2,t-H:t})$), however since the log reward odds can easily explode to -ve or +ve infinity if only one side is rewarded in the past, which would lead to certain predictions for the choice and also lead to numerical instability. Further it would fail to model any inherent randomness in the behavior at extreme log reward odds, so we limit the predicted log odds to $[-l_{max}, l_{max}]$. And therefore the choice probabilities are given by:

$$\log\left(\frac{p(C_{t+1} = \text{Odor 1})}{p(C_{t+1} = \text{Odor 2})}\right) = b + s \min(\max(\log\left(\frac{R_{1,t-H:t}}{R_{2,t-H:t}}\right), -l_{max}), l_{max})$$

We fit the models to the “Variable Block” training dataset by setting $H \in \{5, 10, 15, 30, 60\}$ trials and optimizing the other variables by minimizing negative log likelihood on the data using Neadler Mead optimization in SciPy 1.7.1 and use Normalized Likelihood (see earlier section) for model evaluation and comparison.

Logistic Kernel Regression models for predicting future choices

In the logistic kernel regression model for choice, we try to predict the future choice as a logistic regression on the choice and reward history of the animal. For this, we create a design matrix with a sliding window of size H (history size) over the choice (C; can be -1s or 1s) and reward (R; can be 1s or 0s) sequence and also their product (interaction term R·C; can be -1s, 0s or 1s). We pad the sequences with zeros at the beginning such that each window only has the history before the trial given by the window number. The three windows are joined next to each other in different combinations (C + R + R·C or R + C or R + R·C or C+ R·C). Therefore, the values along the windows become the independent variables used to regress the next choice at trial t (given by the window number). The different models can be formalized as follows:

R + C + R·C model

$$p(C_{t+1} = 1) = \sigma\left(b + \sum_{i=t-H}^t K_{R,i} \cdot R_i + \sum_{i=t-H}^t K_{C,i} \cdot C_i + \sum_{i=t-H}^t K_{RC,i} \cdot R_i \cdot C_i\right)$$

R + R·C model

$$p(C_{t+1} = 1) = \sigma\left(b + \sum_{i=t-H}^t K_{R,i} \cdot R_i + \sum_{i=t-H}^t K_{RC,i} \cdot R_i \cdot C_i\right)$$

C + R·C model

$$p(C_{t+1} = 1) = \sigma\left(b + \sum_{i=t-H}^t K_{C,i} \cdot C_i + \sum_{i=t-H}^t K_{RC,i} \cdot R_i \cdot C_i\right)$$

R + C model

$$p(C_{t+1} = 1) = \sigma\left(b + \sum_{i=t-H}^t K_{C,i} \cdot C_i + \sum_{i=t-H}^t K_{R,i} \cdot R_i\right)$$

where H is the history size, b is the bias, and $K_{X,t}$ are the regression coefficients for the term X at time t, and σ is the sigmoid function. The models are fit using logistic regression with L2 regularization (optimized using cross-validation) in scikit-learn 1.0.2. The models fits are evaluated using Normalized Likelihood (see above section).

Q-Approximation using artificial neural networks

We train artificial neural networks to mimic fly behavioral data. We train two classes of neural networks. Firstly, we design a simple recurrent network which we call the Recurrent q-Network (RqN), that takes in a 2-dimensional sequence of past rewards (0 for unrewarded or 1 for rewarded) and choices (-1 for odor 1, +1 for odor 2). The neural network then tries to predict the acceptance probabilities for two odors (see above section on Accept-Reject policy) in the subsequent trial, which is representative of an estimate of odor value. This value is then transformed into choice probabilities using a differentiable version of the Accept-Reject policy described earlier. Our RqN is composed of a reservoir $R(.)$ of NR recurrently connected neurons that receive the sequence of t trials of past choices ($C_{1:t}$) and ($R_{1:t}$). A single-layer decoder $d(.)$ takes as input the hidden dynamics of the reservoir $R(.)$. It is then transformed to the acceptance probabilities q after passing it through a special “hard-soft” sigmoid nonlinearity σ_{hs} , which are transformed into action probabilities using the Accept-Reject policy $\pi(Q)$.

$$p(C_{t+1}) \sim \pi([q_{t+1}^1, q_{t+1}^2])$$
$$[q_{1:t+1}^1, q_{1:t+1}^2] = \sigma_{hs}(d(R([C_{1:t}, R_{1:t}])))$$
$$\sigma_{hs}(x) = 0.79 \text{ hardsigmoid}(x) + 0.21 \text{ sigmoid}(x)$$

Sigmoid nonlinearities have a diminishing gradient closer to the limits of its range (0,1); it becomes difficult to learn very strong or weak predictions of acceptance probabilities. On the other hand, piecewise linear nonlinearities are either unbounded ($ReLU$) or have the “dying nonlinearity” problem where the gradient becomes zero ($ReLU$ or $hardsigmoid$). Therefore, we design and use a mixed nonlinearity dominated by a linear function ($hardsigmoid$) between the range [0.01,0.99] and dominated by a saturating $sigmoid$ beyond those values, weakening the effect of both challenges and facilitating the learning of strong probabilities. We try out different sizes $R \in \{2, 3, 5, 10, 100, 200\}$ hidden neurons for the reservoir, including networks with a minimal number of neurons, to look for the most parsimonious behavioral models.

However, this method of using an RNN to learn the value dynamics has a significant limitation in interpretability as we do not know how the change in value is integrated

over history. All the past choices and rewards can potentially influence the change in value at every next trial. In order to deal with this limitation, we develop an alternate formulation of a recurrent computation by developing a Feedforward q-Network (FFqN) that takes in only four inputs at every trial. Using the choice and reward in the prior trial (C_t and R_t , respectively) and the predicted acceptance probabilities in the last trial (q_t^1 and q_t^2), the FFqN predicts the updated acceptance probabilities for the next trial. The network is composed of a series of feedforward hidden layers $h_1, h_2 \dots h_d$, where d is the depth of the network with intermediate ReLU nonlinearities, and each layer has n neurons. The choice probability at any arbitrary trial can be found by recursively passing the choice and reward history to the FFqN while feeding the output of the last iteration as the input to the next. We try different depths and widths for the hidden layers of the feedforward network with $(h,n) = \{(1,2), (2,2), (1,5), (2,5), (1,10), (2,10), (3,10), (2,100), (3,100)\}$.

$$p(C_{t+1}) \sim \pi([q_{t+1}^1, q_{t+1}^2])$$

$$[q_{t+1}^1, q_{t+1}^2] = F([q_t^1, q_t^2, C_t, R_t])$$

$$F(x) = \sigma_{hs}(h_d(\text{ReLU}(h_{d-1}(\text{ReLU}(\dots h_1(x)\dots))))))$$

$$[q_0^1, q_0^2] = [0.5, 0.5]$$

While this architecture has the limitation that it has only two dimensions of memory, i.e., q^1 and q^2 , and each step of the update is only dependent on the choice and outcome of a single trial, it provides us an easily interpretable framework to understand the behavior by systematically dissecting the learned function $F(x)$ which updates two variables $[q^1, q^2]$ using only 2 binary variables $[C_t, R_t]$ and therefore have only four possible combinations: (a) Odor 1 chosen and rewarded (C-R+), (b) Odor 2 chosen (C+R+) and rewarded, (c) Odor 1 chosen and not rewarded (C-R-) (d) Odor 2 chosen and not rewarded (C+R-). The update in the acceptance probabilities can therefore be fully characterized as a vector field over $[0,1] \times [0,1]$ under the four different conditions and studied as a conditional first-order discrete dynamical system with typical methods used in studies of non-linear dynamics and dynamical systems theory such as attractor analysis.

However, there is one major drawback with both the proposed models. As a result of their flexibility, these network architectures are susceptible to even slight differences in the preference or learning rate between odors that might present in the behavior. Since there is a large variability between different individuals, there is potentially a chance that the networks get stuck in minima with strong asymmetric behavior. The observed dynamics are thus linked to the odor identity making them less generalizable and informative of fly behavior. While one solution would be to perform data augmentation to symmetrize the dataset to make it more balanced, this would increase the biological variability that, combined with the small dataset size, might drastically reduce the chances of discovering the general underlying learning rule. Therefore we introduce a new symmetrization technique that we refer to as q-Network Output Symmetrization (qNOS) (Figure 12.).

We build and train the networks in PyTorch 1.11.0 by minimizing the Cross-Entropy loss between the predicted choice probabilities and the actual binary choice behavior observed in the flies. The predicted choice probabilities are calculated using the Accept-Reject policy function on the network output. We minimize the loss using Stochastic Gradient Descent with an adaptive moment estimation (Adam) optimizer. Weight decay (L2 regularization) was set to $1e-5$ and the learning rate to $5e-4$. We train the 25 replicates of each network architecture to produce an ensemble of trained networks to help find the generalized learning rule used by the flies rather than being dependent on the inferred values for one well-fit model. The training dataset was randomly split in an 80:20 ratio to training and validation sets for each ensemble fit. The validation set was used to implement early stopping. For the overall training, patience was set to 2500 epochs, and training was done for a maximum of 10000 epochs.

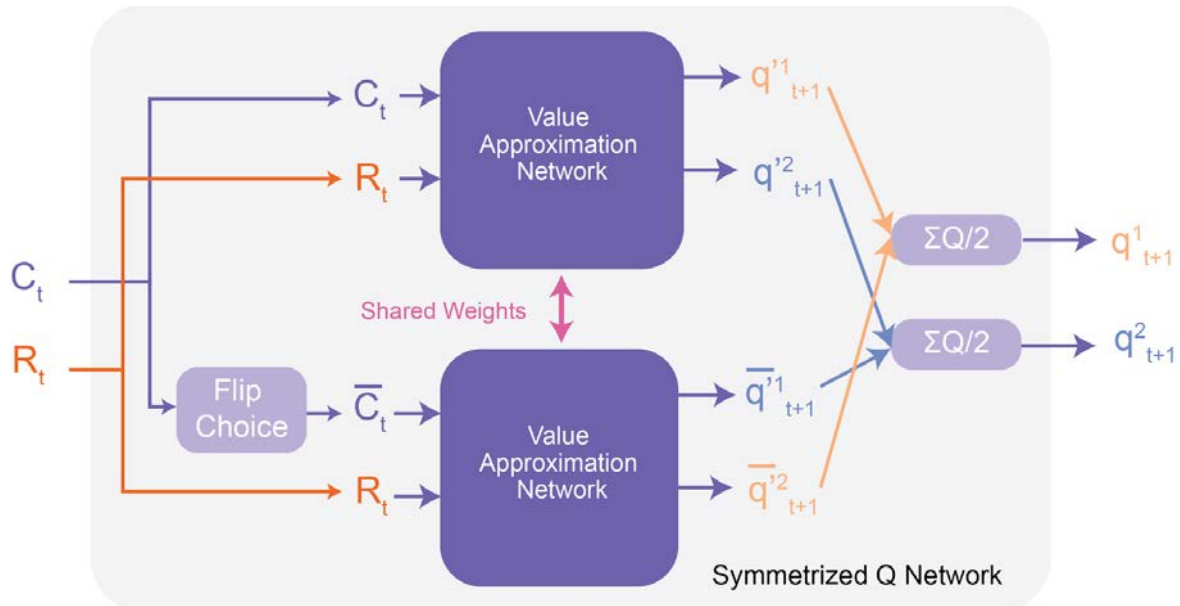


Figure 12. Schematic of q-Network Output Symmetrization (qNOS)

qNOS modifies the architecture of the network in order to ensure that the network's output is always symmetric, i.e., if the identities of the odors were flipped, the predicted acceptance probabilities would also be exactly flipped. We do this by creating a copy of the choice input, flipping the odor identities, and passing it into a copy of the network. The outputs of the copies of the network ($q^{1'}, q^{2'}$, and $\bar{q}^{1'}, \bar{q}^{2'}$ respectively) are cross-averaged between the two copies of the network, i.e., $q^{1'}$ is mixed with $\bar{q}^{2'}$ to get the final output q^1 and vice versa. While this effectively increases the number of independent neurons in the network, we retain the same number of parameters by coupling networks' activities through shared weights. This symmetrization ensures that learning both possible directions of odor choice-reward association happens simultaneously.

Quality Control and Analysis of trained q-Networks

To validate that the models are performing realistically, we simulate two sets of experiments: i) Reward paired with odor 1 for 100 trials (exp 1) and ii) Reward paired with odor 2 for 100 trials (exp 2). We then calculated two scores as follows:

$$\text{Learning Score} = E[\text{Odor 2 is chosen in exp 2}] + E[\text{Odor 1 is chosen in exp 1}] - 1$$

$$\text{Asymmetry Score} = | E[\text{Odor 1 is chosen in exp 1}] - E[\text{Odor 2 is chosen in exp 2}] |$$

The learning score is always between -1 and 1, where -1 corresponds to perfectly learning a negative association and 1 is perfectly learning a perfect positive association. The asymmetry score lies between 0 and 1, where zero means it learns equal associations on both odors irrespective of magnitude and 1 when the associations are biased in the same direction despite opposite reward pairing. We filter out trained networks with low learning scores <0.75 and high asymmetry scores >0.25 .

In order to analyse the trained FFqN, we evaluated the output of the network at 100×100 linearly spaced grid points on acceptance probabilities $q^1 \times q^2 = [0,1] \times [0,1]$ for the four conditions $C \times R = \{-1,1\} \times \{1,0\}$ (alternatively referred to as C-R+, C-R-, C+R+ and C+R-). We subtract the original acceptance probabilities from the final output to find the update vectors, which we map onto the space. We visualize the generated vector field as a stream plot. In order to find the attractors in the network, we randomly choose a starting point and simulate the update under the same condition repeatedly for a maximum of 1000 trials and check if the update has converged to a stable value. We do this across multiple initial conditions to find multiple fixed point attractors if present. We quantify the predicted choice probability at each point in the space by applying the Accept-Reject policy (see the previous section) to the acceptance probabilities.

To characterize the nature of the fixed point attractors, we look at the choice probability at the attractor and transform it to an asymptotic choice index as follows:

$$\text{Asymptotic Choice Index} = 2 P(\text{Choosing Odor 2}) - 1$$

For analyzing the trained symmetric RqNs, we pass all the training data (we use the training data since we only have a small test dataset) through the RqN to get the

dynamics of the preference. However, we extract the dynamics of the hidden reservoir neurons and apply PCA across the neuron identity axis on the hidden dynamics of all the training data. Since PCA is unique up to the sign of the PC axis, we need to make the different trained models comparable. We infer the axis sign by flipping the sign on the PCs and comparing it to the set of predicted PCs for the first ensemble as a reference using the sign of Pearson's correlation between PCs as a criterion. The sign information was then used to visualize PCs and later to look at the trained kernels. We reconstructed the hidden dynamics from the PCs using different numbers of principle components and compared the prediction by passing the input through the decoder $d(\cdot)$ (see section earlier) using Normalized Likelihood (see section earlier). We calculate the autocorrelation on each PC and find the lag at which the autocorrelation is at half its initial maximum to compute the half-life.

For Kernel Regression Analysis for the PCs, we create a design matrix with a sliding window of 80 trials over the choice (C; can be -1s or 1s) and reward (R; can be 1s or 0s) sequence and also their product (interaction term R·C; can be -1s, 0s or 1s). The values are padded with zeros at the beginning such that each window has the history before the trial given by the window number. The three windows are joined next to each other, and the values along the windows become the independent variables used to regress the future PCs. The quality of the fit was estimated with an R² score, and the predictions were compared using Normalized Likelihood.

Choice Engineering using Q-Learning Models

As defined earlier, a reward schedule is a series of deterministic choice-reward outcomes for both odors that the fly can choose in every trial. We develop two stochastic optimization methods for finding reward schedules that maximize the bias for different models: (a) Genetic Optimization (Yang, 2014) and (b) Thermal Annealing (Dan & Loewenstein, 2019) (Figure 13.). To find the maximally-biasing schedules, we start from either a random reward schedule or a naive best guess, i.e., a 'primacy' schedule where all the rewards for the target odor are localized at the start and the end of the session for the distractor (non-target) odor. The idea here is that if there is a strong primacy effect, the first experienced reward association will have the best retained preference.

The optimization is repeated ten times for each algorithm/initialization pair, and the 'best' (maximally-biasing schedules) are chosen after a maximum of 200 generations of optimization. We also implement an early stopping paradigm for optimization where the optimization is terminated if no better schedules are found over ten generations. We optimized the reward schedules for five representative models across the spectrum of model fits (RF-QL, LT-QL, F-RF-QL, DF-LT-QL, and DF-LT-OS-QL). All algorithms were implemented in Python 3.9.7 using custom code.

To find the best schedules for experimental testing, all the top reward schedules from each initialization/algorithm pair were re-evaluated for bias using 1000 independent agents of the model being tested. The top 10 schedules were identified. The bias distribution of the top-ranking schedule was compared with the other nine schedules, and only ones that were not significantly lower ($p > 0.05$; Mann-Witney U Test) were kept. Eight schedules were randomly sampled (with replacement) from this set. Two reward sequences were generated, each with one of the two odors as the target odor, to ensure that the results were not biased by choice of the target odor and run on the single fly Y-maze described in Rajagopalan et al., 2022. OCT (1:500 v/v with 1:10 air dilution) and MCH (1:500 v/v with 1:10 air dilution) were used as the target and distractor odors.

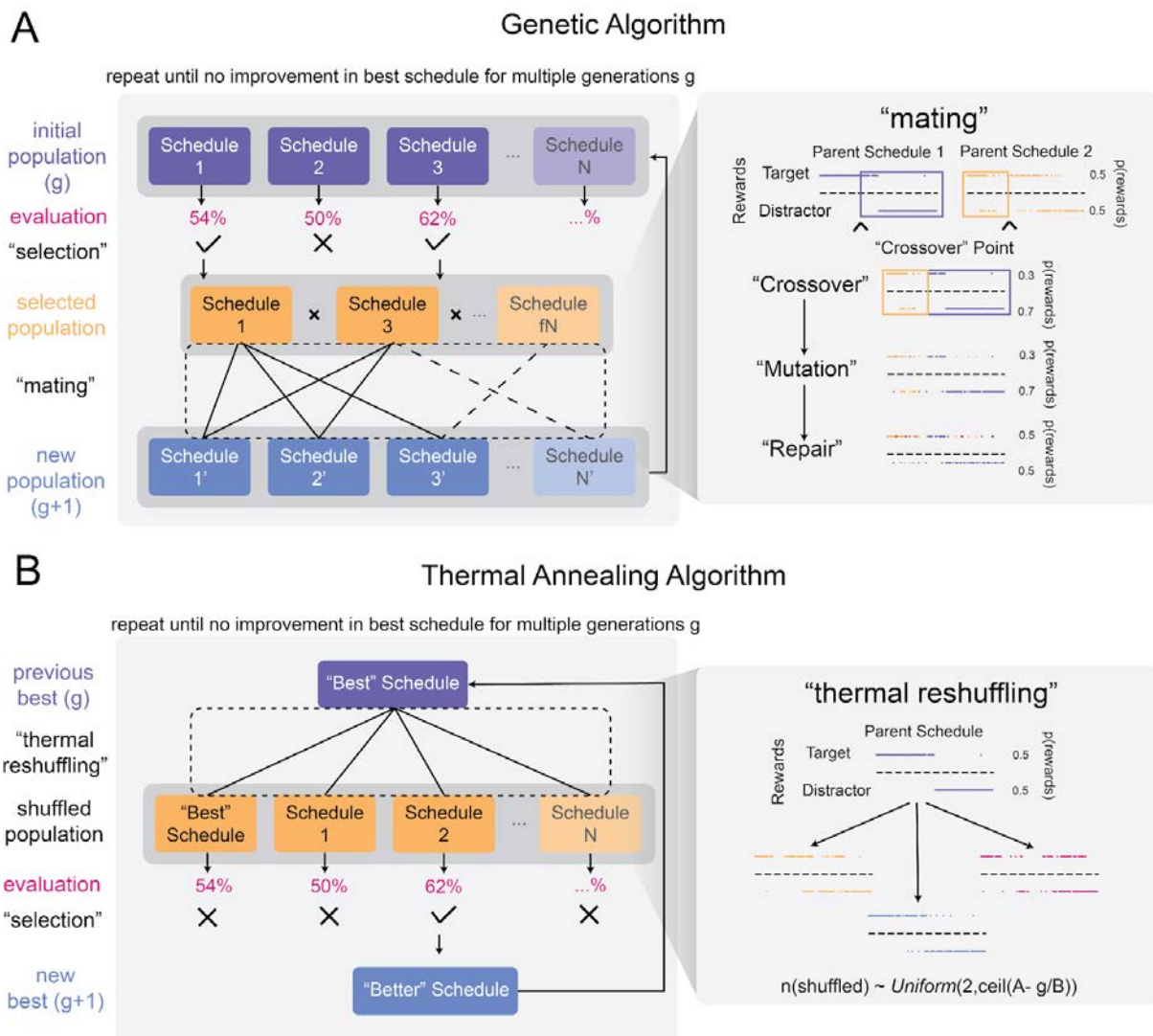


Figure 13. Open-loop choice engineering using stochastic optimization techniques

(A) Genetic Algorithm approach: A population of $N = 100$ reward schedules is initialized. In every generation, there are three steps: i) Evaluation: 1000 agents of the RL model being tested are simulated for each schedule; ii) Selection: The top $f = 20\%$ schedules with the maximum average bias (% choices where the target odor was chosen) are kept, and the rest are discarded. iii) Mating (see inset): From the surviving population, pairs of parents are randomly selected (with replacement) to generate the new population. New children are created from the parents by swapping blocks of rewards between parents defined by $n+1$ "crossover" points along the session where $n \sim \text{Poisson}(0.25)$. New mutations are added by randomly shuffling 5% of the trials for each odor independently. Further, a repair process

randomly removes excess rewards or compensates for reward deficits to ensure the number of rewards remains constant.

(B) Thermal Annealing approach: A single schedule is taken, and its rewards are randomly shuffled to generate a population of 100 new schedules, keeping a copy of the original schedule in the population. The number of shuffles is randomly chosen uniformly between 2 and T where the temperature $T = \lceil A - g/B \rceil$ where $A = 100$, $B = 2$, and g is the generation number. We simulate 1000 agents of the model being tested on each schedule, and the 'best' option is kept, and the entire process is repeated.

High-Throughput Y-Maze Experiments

Quantifying Preference and Learning

To evaluate the strength of preference in an experimental phase, we used the choice index defined below:

$$\text{Choice Index} = (2 \times fC_2) - 1$$

Where fC_2 is the fraction of trials where odor 2 is chosen during an experimental phase.

Since the choice index is always bounded between -1 and 1, any measure of a difference between choice indices is subject to edge effects. Since the choice index measures the probability of choosing an odor, we use a logit transform on the fraction of times odor 2 was chosen to recover the underlying strength of preference. However, the logit is not defined if the fraction of trials where odor 2 was chosen is 1 or 0. Therefore, the value of the fraction was bounded between $[1/N, 1-1/N]$, where N is the number of trials, as any probability higher than $1-1/N$ or lower than $1/N$ will be indistinguishable. The change in this score between experimental phases determines the strength of learning. We call this the learning index:

$$\text{Learning index}_{B-A} = \text{logit}(\max(\min(fC_{2B}, 1-1/N), 1/N)) - \text{logit}(\max(\min(fC_{2A}, 1-1/N), 1/N))$$

where fC_{2X} is the fraction of trials where odor 2 is chosen during an experimental phase X .

Quantifying fly kinematics

All variables used for the analysis are defined in Table 4 and Table 5.

We calculate the average instantaneous speed for a trial by filtering the instantaneous speeds by the current trial number and averaging all frames. Residence densities are calculated by summing up and normalizing all the odor-oriented positions for all trials in the subdivision of the experiments. The difference in odor residence times is compared by directly subtracting the trial odor residence times from each other. The difference in odor rejections is counted by looking at the number of encounters in a trial where the encounter decision outcomes are rejections further filtered by the identity of the odor and then

subtracted from each other. Instantaneous speeds were filtered by current odor and averaged per trial, and the difference was taken to calculate the speed difference in odor. Preference was calculated by looking at the identity of the encounter odors, looking at the fraction of encounters per odor, and taking the difference.

Statistics

All statistics were performed in Python 3.9.7 using the SciPy 1.7.1 and statsmodels 0.12.2. All statistical tables are summarized in the *Statistical Results* section.

For any statistics on simulated/bootstrapped estimates, to prevent inflated estimates of statistical significance, we use an m-out-of-n bootstrap-based sample-size correction for all statistical tests with m = number of flies the data was collected on and n = number of simulated/bootstrap samples or ensembles. Under this paradigm, we perform the statistical test between randomly subsampled sets of m data points out of n data points (either paired or unpaired across simulations/ensembles). We then estimate the 95% percentile p-value or 5% effect size to make claims about statistical significance/effect with 95% certainty based on this value, provided the underlying distribution is well-behaved for bootstrapping.

Symbol conventions are maintained throughout the thesis.

Statistical Significance is symbolized using stars.

ns : not significant, * : <0.05 , ** <0.01 , *** : <0.001 , **** : <0.0001 ;

Effect Sizes are represented using carets.

For Cohen's d , neg : <0.2 = negligible, ^ : <0.5 = small, ^^ : <0.8 = medium, ^^ : >0.474 = large; For Cliff's δ , neg : <0.142 = negligible, ^ : <0.33 = small, ^^ : <0.474 = medium, ^^ : >0.474 = large; For Matched-pair Rank-Biserial Correlation : <0.1 : negligible, <0.3 : small, <0.4 : medium, >0.4 : large.

Code and Data Availability

All Q-Learning models and neural network models developed as a part of this project are available for public use as a Reinforcement Learning package for Python at: <https://github.com/neurorishika/FIYMazeRL>. Analysis code for all experiments are available at https://github.com/neurorishika/FIYMazeRL_Analysis and https://github.com/neurorishika/FIYMazeRL_ChoiceEngg. Experimental datasets from the 16Y assay will be made public in the future.

Results

This section divided into two sub-sections for logical consistency: (1) Analysis of Rajagopalan (2022) “Fixed Block” Dataset, and (2) High-Throughput Y-Maze Experiments.

Analysis of Rajagopalan (2022) "Fixed Block" dataset

Value learning rules for fruit fly behavior

Cognitive Q-Learning models that include forgetting and perseverance are needed to better explain fly behavior

We developed a Python-based computational framework (FIYMazeRL) for the simulation and bayesian model-fitting of various Reinforcement Learning models performing a 2AFC task. We fit 24 Q-Learning Models incorporating different cognitive features (Figure 14.; see Table 6 and Table 7 for details on cognitive features) on data collected from 21 flies in a dynamic reward learning task (Rajagopalan et al., 2022; see methods for details on the dataset). We find that most models reliably converge (close to 1) and are well sampled (Effective sample size > 3000); however, some parameters in the more complex models need further sampling for more reliable estimates (Table 8 and Table 9). We also find that some of the cognitive variables in the models have a substantial impact on the values of the parameters (Table 14; ANOVA Test).

We find that the differences in estimates of predictive accuracy using Normalized Likelihood [Test] between models are limited by the number of test data points ($n = 3$ flies). As a result, the other models are not significantly different from the best predictive model (DF-LT-QL) and have a small effect size. The quality of fit estimate (deviance-scaled WAIC) shows that models that include learning-independent forgetting, temporal discounting, and perseverance (in the form of omission-sensitivity or action prediction errors, i.e., DF-LT-OS-QL and DF-LT-HV-QL) perform significantly better compared to all other models (Table 13; z-test and Cohen's d effect size).

We visualize the predictions of the models from the test set flies using smoothed choice probabilities from both the data and the model fits (Figure 15. A–E). We find

that all models successfully change their preferences in one direction. However, RF-QL models fail to change their preference in the opposing direction. All other models can dynamically track and predict the changing preferences bidirectionally. Simple models (such as LT-QL) can only produce small perturbations in the probability and do not predict substantial preference changes. The better models show similar but sharper predictions (i.e., closer to 0 or 1). It is essential to note that the predicted probabilities are closer to 0.5 (alternatively, 'softer') than the smoothed choice probabilities across trials estimated from the data.

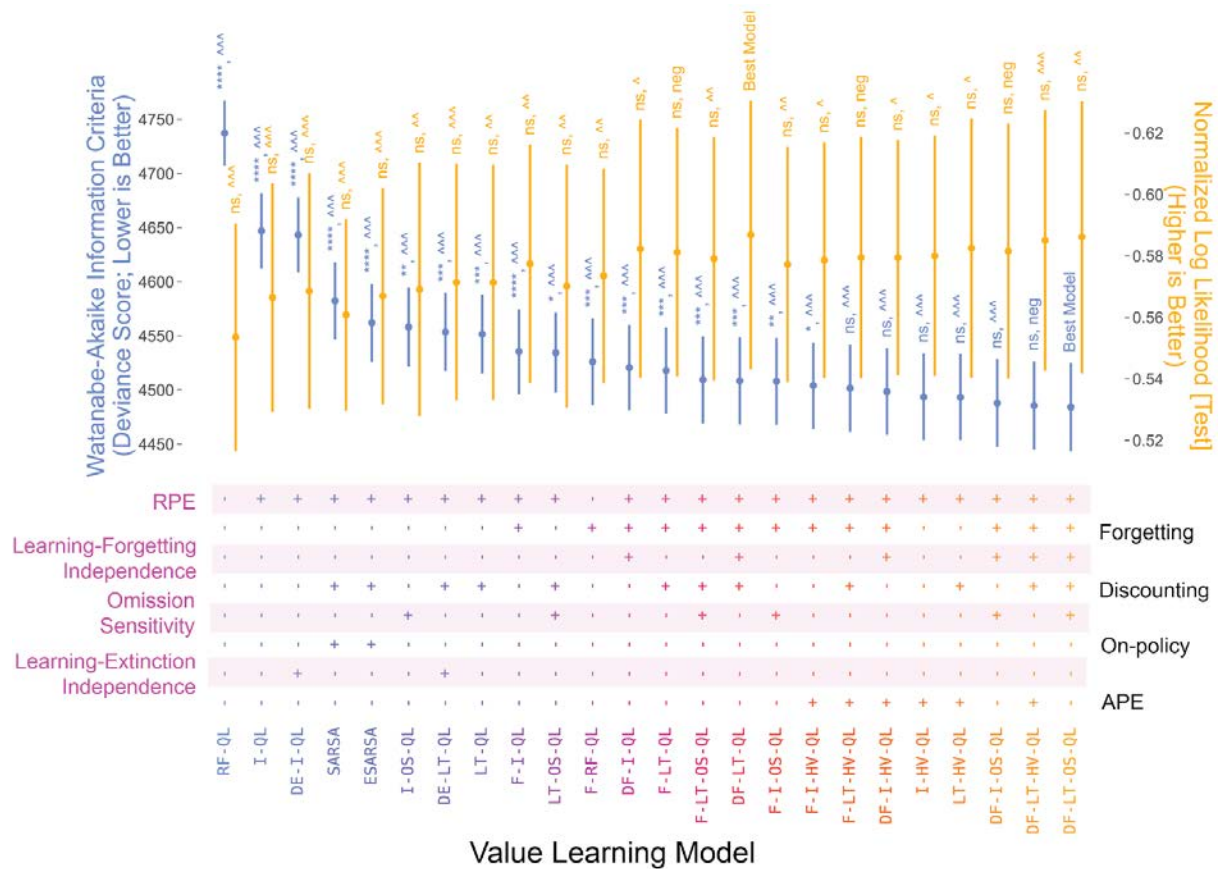


Figure 14. Q-Learning Models of Rajagopalan (2022) "Fixed Block" dataset reveals that including learning-independent forgetting, perseverance, and temporal discounting in the value update improves the model’s explanatory power.

The goodness of fit is estimated using the deviance-scaled Watanabe-Akaike Information Criterion (WAIC; blue), which is a bayesian posterior estimate of parameter count adjusted deviance. The difference of each model’s WAIC relative to the best model is compared using a two-sided z-test (stars for statistical significance; see methods) and Cohen’s d (carets for effect size). Predictive accuracy estimated using Normalized Likelihood [Test] (yellow) is compared relative to the best model using a bootstrap-corrected two-sided paired samples t-test (m=3 flies, n=1000 bootstraps; see methods) (stars for statistical significance) and paired Cohen’s d (carets for effect size). The ‘+’ and ‘-’ symbols at the bottom signify which cognitive features (see Figure 10. and Table 7) are included in the model. Error bars show Standard Error for WAIC and Normalized Likelihood [Test]. See Table 13 for statistics, p-values, and effect sizes.

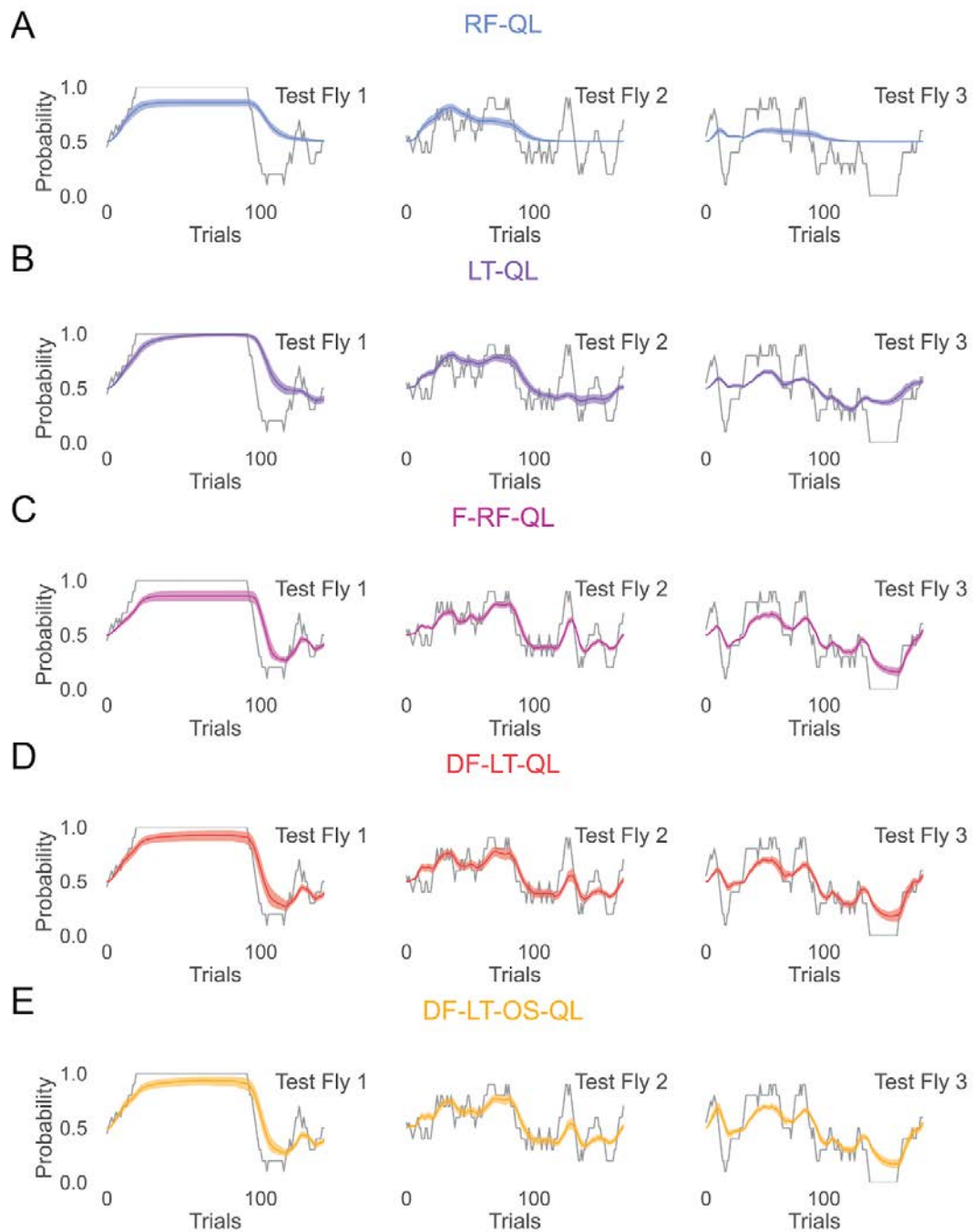


Figure 15. Predicted choice probabilities for different models show diminishing differences with more complex models.

(A–E) Smoothed predicted choice probabilities with 95% confidence interval estimated from 1000 simulations of 5 representative models across the spectrum of model fits for three flies that were not trained on the data overlaid on smoothed choice probabilities estimated from the data with a ten-trial window (see methods).

Cognitive Q-Learning Models cannot fully explain the degree of matching observed in the experiments

We simulate multiple replicates of the experiments from which the data was collected (Table 2) to see if the models can capture the same phenomenological behavior of operant matching observed in Rajagopalan et al., 2022. We find that most models other than the RF-QL model show operant matching behavior with undermatching (i.e., options are chosen at a lower frequency than the reward probability) (Figure 16. A). Intriguingly, the strength of observed matching is even weaker than the observed data for all models (Figure 16. B; Table 15). The DE-LT-QL model is the only exception that shows more substantial matching than the observed data. Further, we observe that the model behavior is unbiased ($b \approx 0$) while the observed data is not (Figure 16. C; Table 15).

Model	α	κ	τ	γ	θ	m	c
Differential Forgetting Long-Term Habit-Value Arbitrator Q-Learning	-	-	-	0.7 (0.52-0.86) [N=351; R=1.0]	-	2.19 (1.38-3.08) [N=625; R=1.0]	-2.64 (-3.59--1.75) [N=10943; R=1.0]
Forgetting Long-Term Habit-Value Arbitrator Q-Learning	-	-	-	0.4 (0.1-0.7) [N=45; R=1.0]	-	2.67 (1.74-3.63) [N=52; R=1.0]	-2.03 (-2.98--1.2) [N=137; R=1.0]
Long-Term Habit-Value Arbitrator Q-Learning	-	-	-	0.64 (0.27-0.86) [N=3964; R=1.0]	-	2.7 (1.85-3.58) [N=5920; R=1.0]	-3.31 (-4.54--2.16) [N=4464; R=1.0]
Differential Forgetting Immediate Habit-Value Arbitrator Q-Learning	-	-	-	-	-	3.02 (2.21-3.88) [N=11119; R=1.0]	-2.4 (-3.34--1.54) [N=15967; R=1.0]
Forgetting Immediate Habit-Value Arbitrator Q-Learning	-	-	-	-	-	3.14 (2.33-4.03) [N=6071; R=1.0]	-2.29 (-3.22--1.41) [N=7908; R=1.0]
Differential Forgetting Immediate Q-Learning	0.09 (0.06-0.13) [N=12238; R=1.0]	0.26 (0.17-0.36) [N=14113; R=1.0]	-	-	-	3.07 (2.36-3.83) [N=11678; R=1.0]	-2.34 (-3.29--1.4) [N=13837; R=1.0]
Forgetting Immediate Q-Learning	0.13 (0.1-0.17) [N=8987; R=1.0]	-	-	-	-	2.48 (2.02-3.05) [N=7022; R=1.0]	-2.1 (-3.08--1.21) [N=7255; R=1.0]
Immediate Q-Learning	0.06 (0.05-0.08) [N=11123; R=1.0]	-	-	-	-	2.54 (2.1-3.03) [N=8632; R=1.0]	-2.57 (-3.44--1.79) [N=8996; R=1.0]
Forgetting RPE-free Q-Learning	0.42 (0.32-0.51) [N=9371; R=1.0]	-	-	-	-	0.66 (0.49-0.85) [N=6868; R=1.0]	-1.23 (-1.7--0.78) [N=7491; R=1.0]
RPE-free Q-Learning	0.49 (0.12-0.94) [N=5593; R=1.0]	-	-	-	-	0.65 (0.19-1.47) [N=5689; R=1.0]	-1.22 (-1.49--0.94) [N=6935; R=1.0]
Immediate Habit-Value Arbitrator Q-Learning	-	-	-	-	-	3.02 (2.24-3.87) [N=14481; R=1.0]	-2.66 (-3.6--1.78) [N=19181; R=1.0]
Differential Extinction Long-Term Q-Learning	0.16 (0.12-0.2) [N=5178; R=1.0]	-	0.16 (0.12-0.2) [N=5023; R=1.0]	0.82 (0.73-0.9) [N=6152; R=1.0]	-	1.24 (0.94-1.56) [N=6165; R=1.0]	-5.24 (-6.29--4.13) [N=6424; R=1.0]

Differential Forgetting Long-Term Q Learning	0.25 (0.18–0.33) [N=8224; R=1.0]	0.08 (0.04–0.13) [N=7971; R=1.0]	-	0.72 (0.6–0.85) [N=7390; R=1.0]	-	1.44 (0.92–1.97) [N=6808; R=1.0]	-2.18 (-2.97–-1.4) [N=7115; R=1.0]
Forgetting Long-Term Q Learning	0.17 (0.13–0.22) [N=11326; R=1.0]	-	-	0.6 (0.39–0.8) [N=11202; R=1.0]	-	1.79 (1.16–2.43) [N=7733; R=1.0]	-1.71 (-2.54–-1.05) [N=9182; R=1.0]
Long-Term Q Learning	0.16 (0.12–0.2) [N=5908; R=1.0]	-	-	0.82 (0.75–0.89) [N=5218; R=1.0]	-	1.23 (0.95–1.5) [N=5532; R=1.0]	-5.29 (-6.4–-4.21) [N=5998; R=1.0]
Differential Extinction Immediate Q Learning	0.09 (0.06–0.12) [N=12335; R=1.0]	-	0.04 (0.03–0.06) [N=12656; R=1.0]	-	-	2.39 (2.02–2.79) [N=11112; R=1.0]	-2.91 (-3.86–-2.07) [N=11158; R=1.0]
Expected SARSA	0.15 (0.11–0.18) [N=6293; R=1.0]	-	-	0.84 (0.77–0.91) [N=5528; R=1.0]	-	1.38 (1.12–1.66) [N=5764; R=1.0]	-4.98 (-6.0–-4.0) [N=5856; R=1.0]
SARSA	0.16 (0.12–0.19) [N=5984; R=1.0]	-	-	0.83 (0.75–0.9) [N=5278; R=1.0]	-	1.2 (0.95–1.45) [N=6387; R=1.0]	-4.7 (-5.67–-3.81) [N=6386; R=1.0]
Differential Forgetting Long-Term Omission Sensitive Q Learning	0.26 (0.19–0.33) [N=12234; R=1.0]	0.06 (0.03–0.08) [N=10202; R=1.0]	-	0.46 (0.21–0.69) [N=7890; R=1.0]	0.42 (0.26–0.6) [N=9864; R=1.0]	1.99 (1.32–2.73) [N=8476; R=1.0]	-2.87 (-3.83–-1.92) [N=9615; R=1.0]
Forgetting Long-Term Omission Sensitive Q Learning	0.16 (0.12–0.2) [N=8634; R=1.0]	-	-	0.33 (0.01–0.59) [N=7140; R=1.0]	0.25 (0.09–0.41) [N=9587; R=1.0]	2.07 (1.35–2.84) [N=6509; R=1.0]	-1.95 (-2.86–-1.13) [N=10014; R=1.0]
Long-Term Omission Sensitive Q Learning	0.14 (0.1–0.17) [N=5680; R=1.0]	-	-	0.67 (0.51–0.82) [N=4675; R=1.0]	0.34 (0.2–0.47) [N=6890; R=1.0]	1.82 (1.27–2.37) [N=4899; R=1.0]	-5.26 (-6.28–-4.25) [N=6834; R=1.0]
Differential Forgetting Immediate Omission Sensitive Q Learning	0.23 (0.16–0.29) [N=13150; R=1.0]	0.06 (0.04–0.09) [N=9167; R=1.0]	-	-	0.57 (0.45–0.68) [N=11639; R=1.0]	2.87 (2.24–3.51) [N=12879; R=1.0]	-2.65 (-3.59–-1.8) [N=10245; R=1.0]
Forgetting Immediate Omission Sensitive Q Learning	0.14 (0.11–0.18) [N=13370; R=1.0]	-	-	-	0.34 (0.22–0.47) [N=14190; R=1.0]	2.54 (1.96–3.17) [N=9004; R=1.0]	-2.07 (-3.02–-1.21) [N=9743; R=1.0]
Immediate Omission Sensitive Q Learning	0.09 (0.07–0.1) [N=13807; R=1.0]	-	-	-	0.53 (0.45–0.62) [N=12566; R=1.0]	3.58 (3.12–4.06) [N=15164; R=1.0]	-4.1 (-4.94–-3.28) [N=13996; R=1.0]

Table 8. Q-Learning Model Fit Parameters for Rajagopalan (2022) "Fixed Block" dataset.

Mean (95% Credible Interval), Effective Sample Size (N), and Convergence (R) are provided. Parameters not meeting our quality standards (see methods) are marked in bold. ANOVA results in Table 14.

Model	α_v	α_h	θ_v	θ_h	κ_v	w_v	w_h	w_b
Differential Forgetting Long-Term Habit-Value Arbitrator Q-Learning	0.22 (0.12–0.4) [N=159; R=1.0]	0.76 (0.05–1.0) [N=176; R=1.0]	0.83 (0.6–1.0) [N=14140; R=1.0]	0.69 (0.34–1.0) [N=2596; R=1.0]	0.04 (0.01–0.1) [N=217; R=1.0]	0.29 (-1.33–1.87) [N=22862; R=1.0]	0.8 (-0.62–2.26) [N=11886; R=1.0]	-0.34 (-1.66–0.96) [N=837; R=1.0]
Forgetting Long-Term Habit-Value Arbitrator Q-Learning	0.17 (0.08–0.31) [N=9; R=1.4]	0.58 (0.02–0.98) [N=9; R=1.4]	0.87 (0.68–1.0) [N=15533; R=1.0]	0.7 (0.34–1.0) [N=311; R=1.0]	-	-0.04 (-1.83–1.65) [N=22251; R=1.0]	1.17 (-0.33–2.61) [N=11215; R=1.0]	-0.28 (-1.63–0.93) [N=29; R=1.1]
Long-Term Habit-Value Arbitrator Q-Learning	0.25 (0.1–0.47) [N=3864; R=1.0]	0.18 (0.08–0.31) [N=3550; R=1.0]	0.72 (0.39–1.0) [N=6473; R=1.0]	0.87 (0.65–1.0) [N=8243; R=1.0]	-	0.08 (-1.83–2.0) [N=5968; R=1.0]	-1.05 (-2.43–0.4) [N=5231; R=1.0]	-0.29 (-1.26–0.75) [N=8641; R=1.0]
Differential Forgetting Immediate Habit-Value Arbitrator Q-Learning	0.4 (0.17–0.7) [N=1236; R=1.0]	0.1 (0.04–0.14) [N=1175; R=1.0]	0.78 (0.43–1.0) [N=7028; R=1.0]	0.86 (0.63–1.0) [N=6365; R=1.0]	0.16 (0.0–0.32) [N=9398; R=1.0]	0.13 (-1.56–1.83) [N=15459; R=1.0]	-0.18 (-1.98–1.8) [N=3675; R=1.0]	0.1 (-1.02–1.16) [N=6310; R=1.0]
Forgetting Immediate Habit-Value Arbitrator Q-Learning	0.24 (0.06–0.42) [N=18; R=1.2]	0.23 (0.02–0.9) [N=19; R=1.2]	0.88 (0.65–1.0) [N=6887; R=1.0]	0.78 (0.47–1.0) [N=234; R=1.0]	-	0.03 (-1.69–1.79) [N=15275; R=1.0]	0.73 (-1.34–2.45) [N=281; R=1.0]	-0.05 (-1.24–1.05) [N=65; R=1.0]
Immediate Habit-Value Arbitrator Q-Learning	0.37 (0.18–0.57) [N=19919; R=1.0]	0.12 (0.09–0.16) [N=20928; R=1.0]	0.55 (0.25–0.94) [N=9702; R=1.0]	0.91 (0.77–1.0) [N=15795; R=1.0]	-	-0.79 (-2.39–0.8) [N=20996; R=1.0]	-1.32 (-2.48–0.18) [N=13601; R=1.0]	-0.12 (-1.18–0.94) [N=10741; R=1.0]

Table 9. Q-Learning Model Fit Parameters for Rajagopalan (2022) "Fixed Block" dataset (contd). Mean (95% Credible Interval), Effective Sample Size (N), and Convergence (R) are provided. Parameters not meeting our quality standards (see methods) are marked in bold.

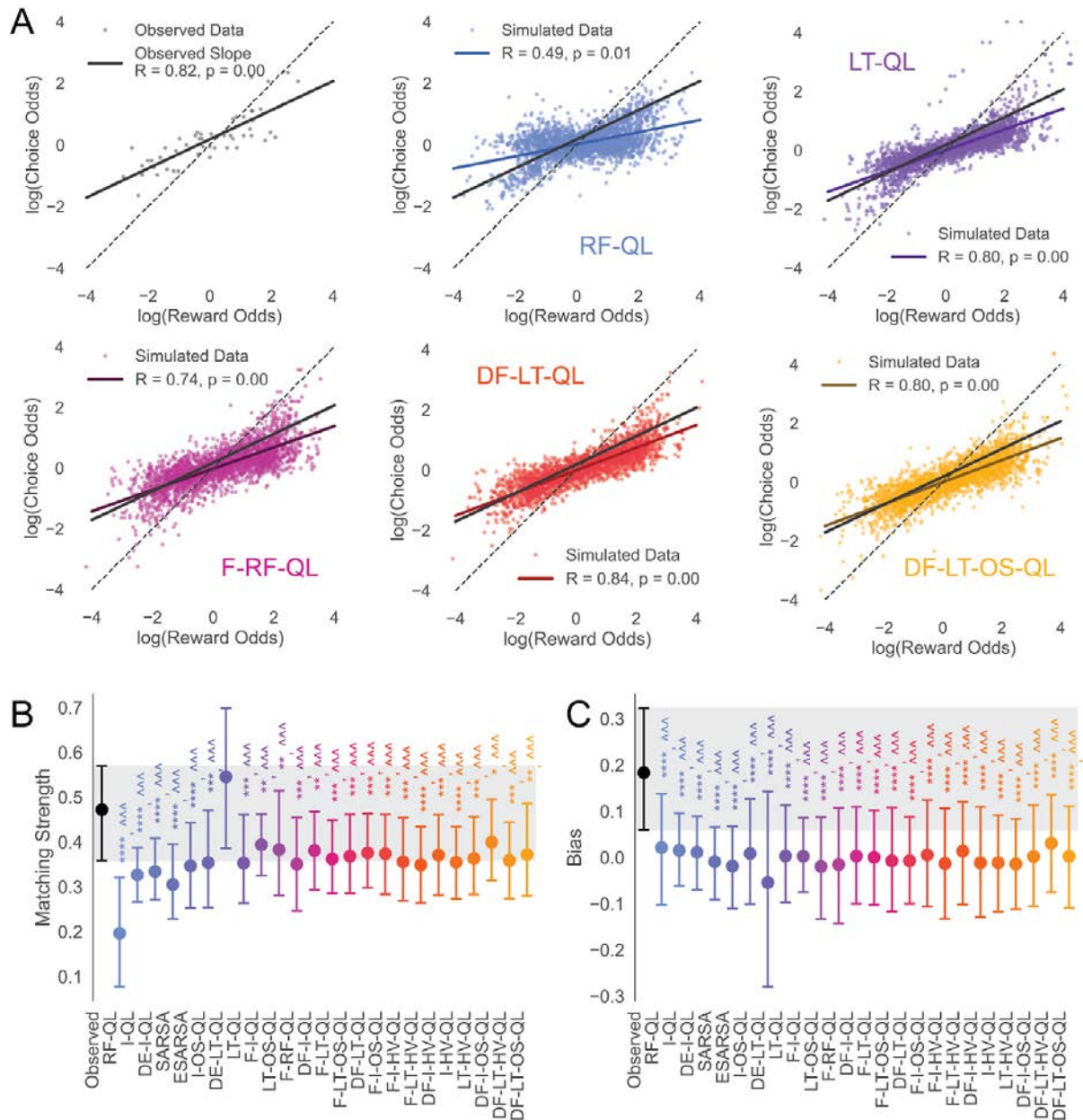


Figure 16. Q-Learning Models preserve the matching behavior observed in behavior.

(A) Generalized matching law observed as a linear function between log(choice odds) vs. log(reward odds) within each block of trials with static baiting probabilities for the experimental data and simulations of 50 repeats of the 18 experiments for the different models. Five representative models along the spectrum of the model fits are visualized. Linear fit, correlation coefficient R , and associated p -value are plotted and reported.

(B–C) Matching strength and bias (see methods) for the data and the model simulations with bootstrapped 95% CI. The grey band represents the 95% confidence interval for the observed data. Model behavior is compared to the experimental data using bootstrap-corrected Mann-Whitney test ($m=18$ flies, $n=1000$ simulations, 1000 random bootstraps; see methods) (stars for statistical significance) and Cliff's delta effect size (caret for effect size). See Table 15 for statistics, p-values, and effect sizes.

Cognitive Q-Learning models differ in the dynamics of the value despite similar average behavior

Since the behavior for the different models that performed better than the RF-QL models had very similar predictions and behavior, we sought to understand the differences in the underlying computations between different models. For this, we looked at the dynamics of the estimated value. We do so by looking at how the acceptance probabilities for the two odors predicted by each model change over time. We simulate a random “Variable Block” experiment (Figure 17. A; see methods) in response to which a simulated fly can replicate a matching behavior (Figure 17. B) and then look at the underlying value code of the models. It reveals that the RF-QL fails to dynamically change its preference as it first learns to accept the rewarded odor and then learns to accept the other odor but cannot “forget” this learned association. Therefore, it learns to prefer an odor and then balance its preference with another odor but fails to choose beyond them further (Figure 17. C).

Beyond the RF-QL model, all models start to have very similar average dynamics in terms of the observed preference and acceptance probabilities; however, looking at the trajectories of a single session does reveal apparent differences in the dynamics (Figure 17. D–G). In the better models, we observe much faster dynamics that allow rapid changes in acceptance probabilities and preference. This local variance is quantified and is found to be present in all models with forgetting (Figure 17. H; Table 16)

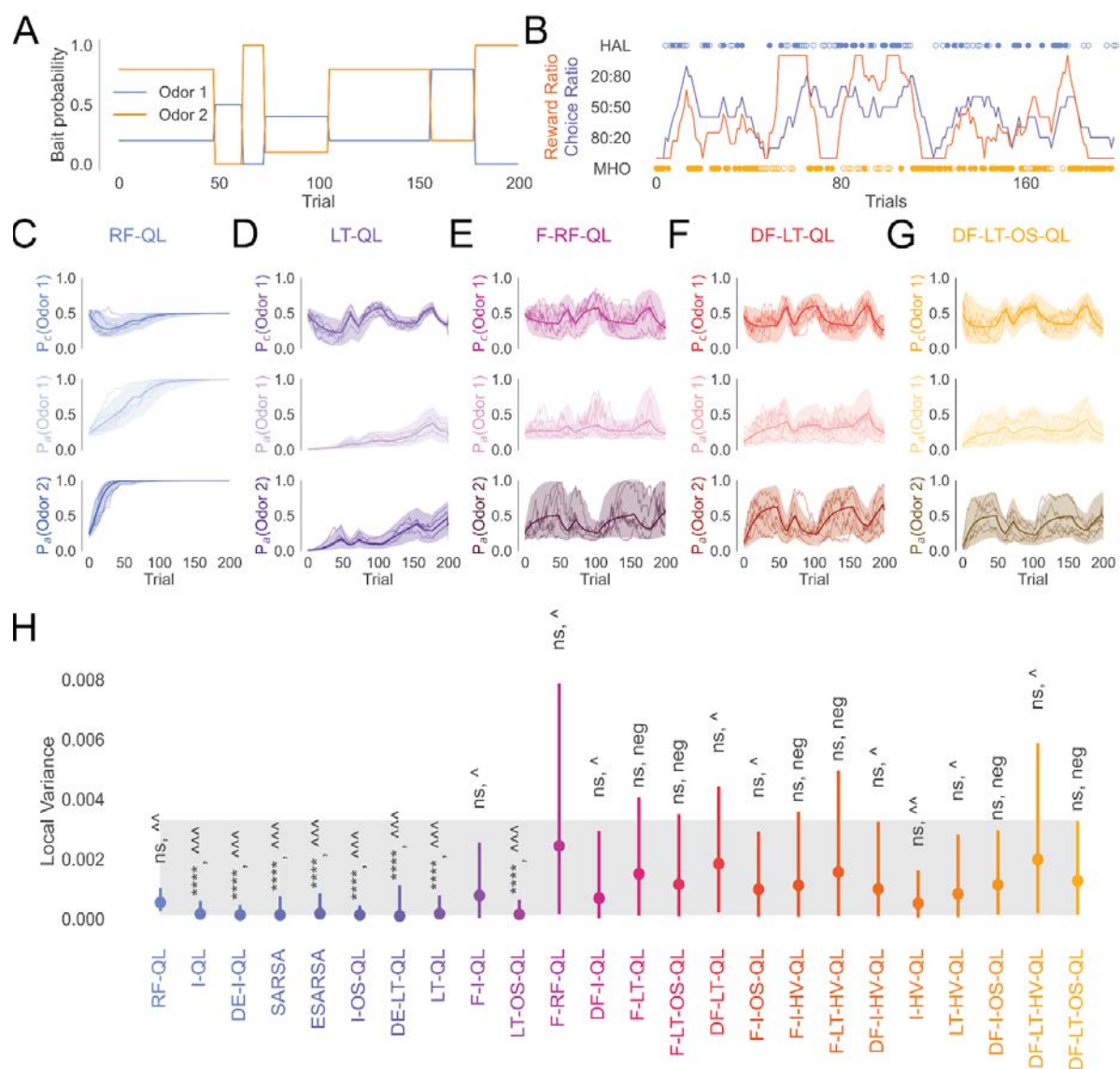


Figure 17. The dynamics of value underlying different models reveal differences in local variance.

(A) An example of a single random “Variable Block” experiment generated by simulating a simple Markov chain (see methods).

(B) Behavioral trajectory of a simulated fly using the best model (DF-LT-OS-QL) on the example Variable Block experiment that shows a matching between running reward ratio and choice ratio.

(C–G) Underlying preference dynamics for five representative models across the spectrum of model fits visualized using i) choice probability $P_c(\text{Odor 2})$ = the

probability of choosing odor 2; ii) acceptance probability $P_a(\text{Odor 1 or Odor 2})$ = the probability of choosing odor 1 or odor 2 (representative of its value) along with its 95% confidence interval (shaded area) calculated with 1000 independent trajectories. Five sample trajectories from the simulated data are shown overlaid on the data.

(H) Quantification of the local variance across a single session for different models. The shaded area represents the 95% confidence interval of the best model. Differences from the best model are quantified using bootstrap-corrected Mann-Whitney U test ($m=18$ flies, $n=1000$ simulations; unpaired data was sampled using 1000 bootstraps; see methods) (stars for statistical significance) and Cliff's delta effect size (carets for effect size). See Table 16 for p-values and effect sizes.

De-novo value learning rule estimation using artificial neural networks

Our Q-learning approach is faced with the limitation that if there are features that a fly utilizes in its learning process that we do not include in our analysis, we will be unable to discover the importance of these features. We might be missing out on a large class of algorithms because we are only sampling a small fraction of the space of all value learning rules, a fraction of which are likely to be utilized by the fly. (Figure 18. A) Therefore, there is a need for an unbiased framework to understand and model the learning rule used by a fly. For this purpose, we look to artificial neural networks, which are known to be “universal function approximators” (Schäfer & Zimmermann, 2006; Sonoda & Murata, 2017) to try and create a “universal value approximator” by incorporating a neural network into the reinforcement learning framework (Figure 18. B). Instead of optimizing the neural network to perform the same task as the flies optimally, we use an imitation learning framework to infer the trajectory of value updates (Figure 18. B).

In order to infer the dynamics of value underlying the behavior we observe in flies, we train an ensemble of two major classes of neural networks: Feedforward q-Networks (FFqN) and Recurrent q-Networks (RqN) (Figure 18. C). Both types of networks are trained to predict the value (defined as the probability of accepting each odor) given the sequence of past choices and rewards. We try two variants for each class of neural network: asymmetric and symmetric. Symmetric variants are constrained to produce the exact flipped preferences if the identity of the input odors is flipped. In contrast, asymmetric variants are allowed to respond to two odors with different learning rules.

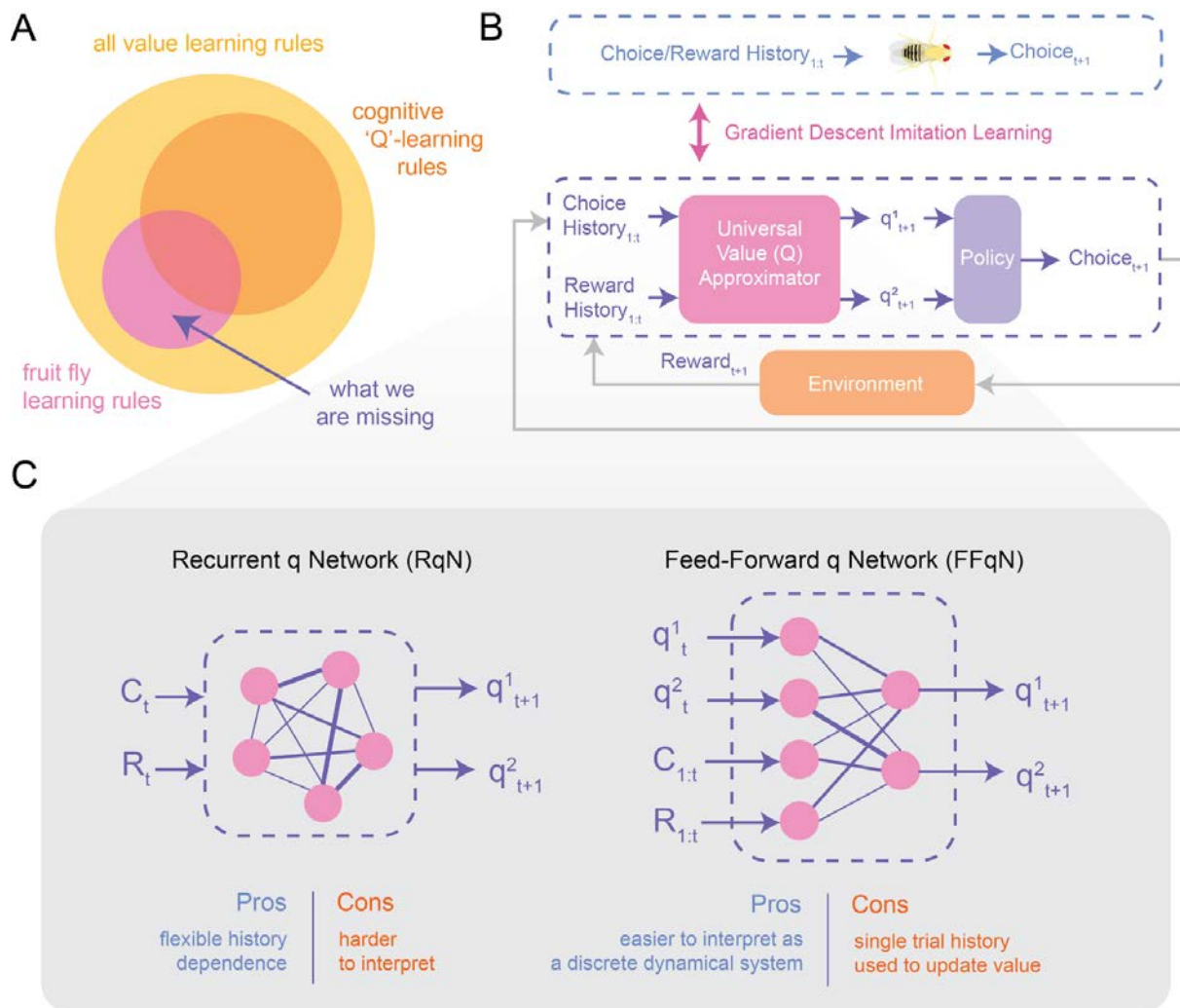


Figure 18. Neural networks can flexibly estimate the value learning rules via imitation learning.

(A) Venn Diagram of how we think the space of value learning rules are organized. The space of value learning is much bigger than the space we sample using our cognitive feature Q-learning models and is constrained by many assumptions. The actual space of learning rules that the fly uses may only partially overlap with our models; therefore, we need a way to sample the space with minimal assumptions.

(B) Our framework of value learning essentially needs a black box Universal Value (Q) Approximator (pink) that is capable of taking all of the histories and using it to predict the acceptance probabilities (q^1 and q^2 ; representative of odor value). These probabilities are then transformed by the Accept-Reject policy (see methods) and sampled to give the choices. The choices are then associated with rewards from the environment. In order to find what this black box does, we can use a Neural Network

to try and imitate the behavior observed constrained by the same value learning framework.

(C) We can use many different architectures for the neural network to approximate the behavior. However, the two leading types of artificial neural networks (ANNs) used by Machine-Learning (ML) researchers are Recurrent Neural Networks (RNNs) and Feedforward Neural Networks (FFNNs). We use these ANNs to create two different classes of value estimation networks. Recurrent q-Networks (RqNs) take in the entire sequence of past histories to predict the acceptance probabilities in the subsequent trial. Feedforward q-Networks take only the acceptance probabilities of the last trial and update them using the choice and reward from the current trial to update the acceptance probabilities for the subsequent trial.

Small Neural Networks can approximate Q-update functions, often better than Q-learning models

We look at how well different architectures (ranging from small to large networks) of neural networks fit and predict behavior (Figure 19. A–B). We find that even small RqNs with reservoirs of less than five neurons are good at both fitting and predicting the behavior, with the best overall model being the symmetric RqN (i.e., symRqN(3) with a reservoir with six effective, hidden neurons) performing even better than the best Q-learning model (Figure 19. A; right). The relatively simpler FFqNs also manage to capture and predict the model but fail to perform as well as the best Q-learning models (Figure 19. A; left). We visualize the quality of the predictions by looking at the smoothed choice probabilities predicted by the data and the neural networks and find that all of the best models from network class-variant pairs capture the dynamics of the preference quite well. However, the differences between model quality are minimal and are hard to observe after smoothening (Figure 19. B).

and effect sizes, including a comparison of training Normalized Likelihood using the same statistical measures.

(B) Smoothed predicted choice probabilities for 3 test flies with a 95% confidence interval estimated from 25 ensemble models for the best network architectures from each network class/variant overlaid on smoothed choice probabilities estimated from the data with a ten trial window (see methods).

Feedforward q-Networks reveal perseverance behavior

While both classes of neural networks manage to capture aspects of the preference behavior, we next try to dissect the behavior of the neural networks to understand the underlying learning rule. For this purpose, the Feedforward q-Network is much more tractable as it can be analyzed as a first-order discrete dynamical system. Four vector fields uniquely describe the entire estimated value approximation function. We can characterize the learning rule by looking at the fixed point attractors of the vector fields across ensembles of trained neural networks (Figure 20.).

Doing this for the asymmetric FFqNs reveals a set of vector fields that are highly variable between the trained neural networks in the ensemble. However, they shared a common feature of a single unique fixed point attractor for every condition (Figure 21. A). We noticed that some of the trained neural networks failed to produce a value update that allows any form of associative learning. These fits failed to associate rewards with any odors (traces not shown). Therefore we simulated two simple 100-trial learning experiments 100 times, with each odor always being paired with odor, and filtered out trained networks with asymmetric learning or weak learning (Figure 21. B). We notice that the four fixed point attractors for the four conditions always correspond with a set of acceptance probabilities that are associated with an increased preference towards the choice, irrespective of the odor (Figure 21. C). We quantified the predicted choice probabilities at the fixed point attractors for the four conditions; we found that except for the C+R- condition (Odor 2 chosen but not rewarded), future choices were asymptotically biased toward continuing to choose the same odor irrespective of reward. That is, there is a tendency to persevere toward the last action (Figure 21.. D).

We replicate this analysis for the symmetric FFqN and find that not only do the vector flows look more reliable, but also the learning score and the position of the fixed point attractors across trained networks from the ensemble appear to be more reliable (Figure 22. A–C). When we quantify the predicted asymptotic choice index, we find that the networks show reliable and robust perseverance (Figure 22. D).

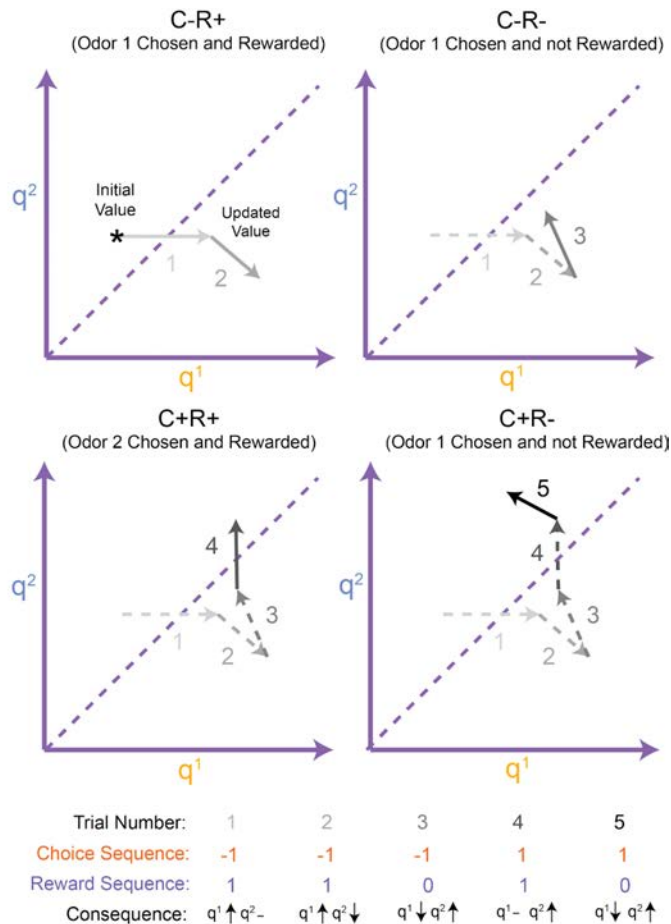


Figure 20. Understanding FFqNs as a conditional first-order discrete dynamical system.

Consider an example sequence of choices and rewards starting at trial 1. The value of the two odors initially can be anywhere on the space of q^1 and q^2 ; say it is at the point marked by the asterisk. In the first trial, where odor 1 is chosen and rewarded, the acceptance probability of odor 1 increases, and odor 2 remains the same (See arrow 1 in the C-R+ space). Since the change in probability only depends on the initial position and the (C, R) condition, the vector update is always uniquely defined for every point in the space. Similarly, in the subsequent trial, the update continues on the same condition C-R+, but this time the acceptance probability of odor 2 might reduce at this new point in the space. In the subsequent trial, the condition changes to C-R- where an independent vector field is defined, which leads to a decrease in the acceptance probability of odor 1 and an increase in odor 2. This vector continues over different (C, R) conditions over successive trials resulting in the overall behavior. However, any trajectory is fully defined by the four vector fields for the four conditions, the sequence of (C, R) conditions, and the initial conditions.

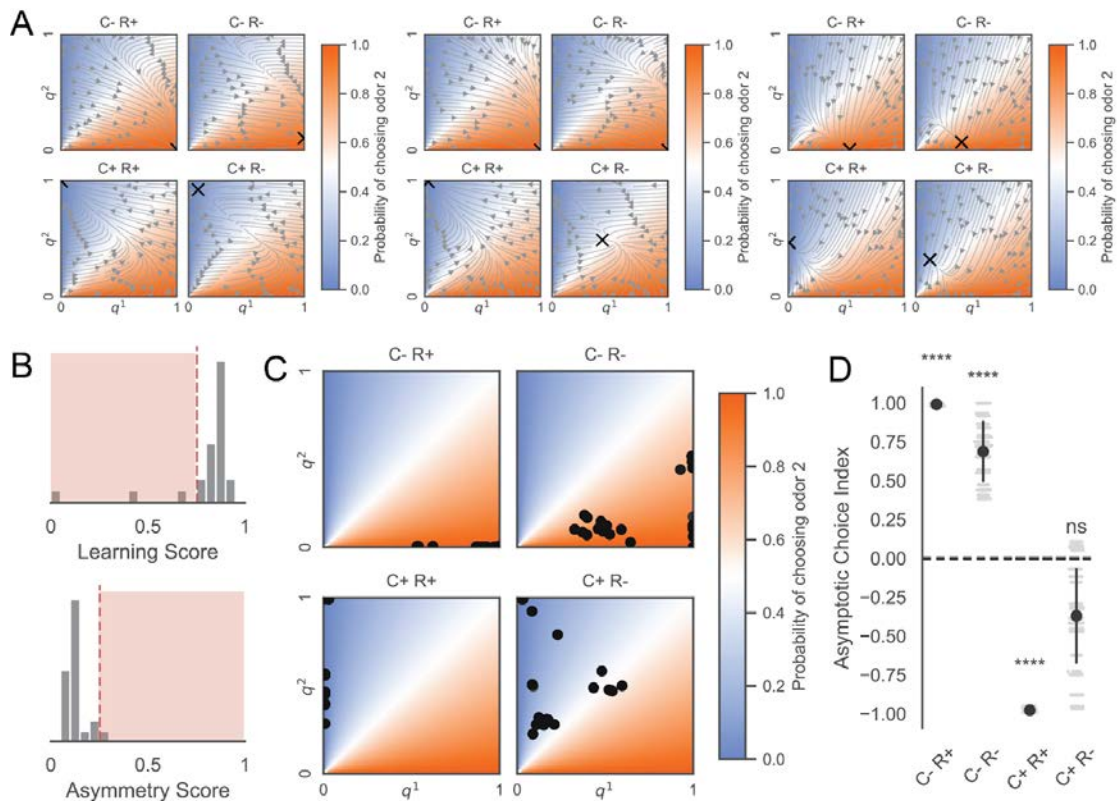


Figure 21. Dynamical systems analysis of an asymmetric FFqN reveals a system of unreliable attractors with weak perseverance.

(A) Vector fields for the acceptance probability update under the four Choice-Reward conditions represented as flows with the final choice probability represented as a heatmap with the simulated fixed points (from 100 independent initializations) marked with a cross. The estimated vector fields for three independent trained networks from the ensemble are shown.

(B) Histograms of learning and asymmetry scores for all the trained networks from the ensemble (for an explanation of scores, see methods).

(C) Position of all the fixed point attractors across the trained and filtered ensemble of asymmetric FFqNs marked on the space of acceptance probabilities with a black dot.

(D) Predicted preference of odors at the fixed point attractors of the different choice-reward conditions for all trained and filtered asymmetric FFqNs of the ensemble compared from zero using a two-sided bootstrap test (stars for significance; $p=0.000$ for all values other than C+R- where $p=0.176$).

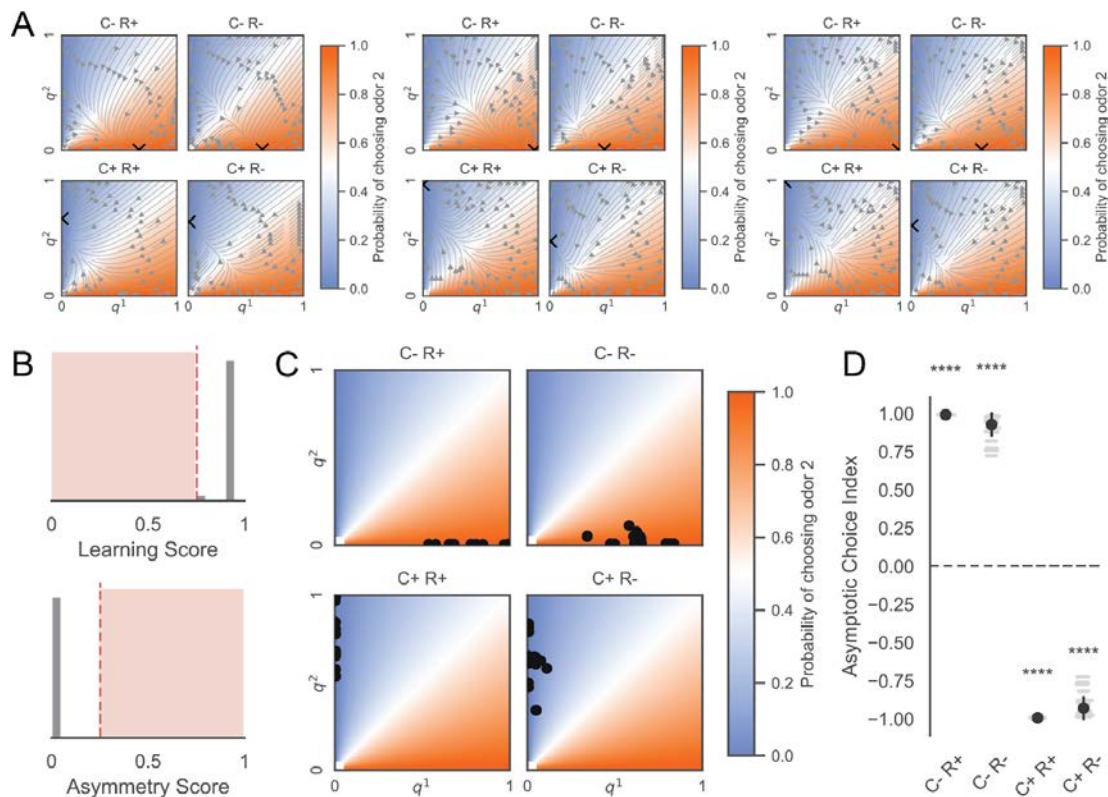


Figure 22. Dynamical systems analysis of a symmetric FFqN reveals a system of reliable attractors with stronger perseverance.

(A) Vector fields for the acceptance probability update under the four Choice-Reward conditions represented as flows with the final choice probability represented as a heatmap with the simulated fixed points attractors (from 100 independent initializations) marked with a cross. The estimated vector fields for three independent trained networks from the ensembles are shown.

(B) Histograms of quantified learning and asymmetry scores for all the trained networks from the ensemble. Same as Figure 21..

(C) Position of all the fixed point attractors across the trained and filtered ensemble of symmetric FFqNs marked on the space of acceptance probabilities with a black dot.

(D) Predicted preference of odors at the fixed point attractors of the different choice-reward conditions for all trained and filtered symmetric FFqNs of the ensemble compared from zero using a two-sided bootstrap test (stars for significance; $p=0.000$ for all values).

Recurrent q-Networks likely have a separation of timescales that drives stronger changes in preference.

Due to its architecture, we can easily interpret the dynamics of the FFqN. However, it fails to capture choice behavior even as well as the best Q-learning models. Therefore, there is a need to understand how and why the RqN manages to explain the behavior better. For this purpose, we look at the hidden dynamics of the RqN underlying the choice behavior of the flies. For simplicity, we limit ourselves to the symmetric RQNs.

In order to be able to make a comparison between different trained networks from the ensemble, we needed a projection of the dynamics to a common space. Therefore we looked at the principal components (PCs) of the hidden reservoir dynamics for the best symmetric RqN during the choice-reward trajectories observed in the original training dataset. We find that the first PC seems to strongly capture the trend observed in preference dynamics (Figure 23. A). The first PC seems only to capture 60% of the variability (Figure 23. B). To test whether the additional variability is essential to explain the behavior, we successively removed the later PCs one after the other. We then used the reconstructed hidden dynamics to predict the final choice and compared the Normalized Likelihood. We find that removing the last two PCs affects the prediction quality considerably; but, removing the second PC has a much stronger effect (Figure 23. C).

To understand the changes in the predictions, we look at the examples where the difference is the largest. While the last two PCs seem to modulate the strength of change in response to the outcomes of choices, the second PC seems to introduce faster timescale perturbations that modify the choice preference (Figure 23. D). To quantify this, we look at the autocorrelation of the PCs (Figure 23. E). While the differences are not significant after correcting for sample size, there is a large decrease in the half-life in the later PCs suggesting faster timescale dynamics that influence the choice dynamics.

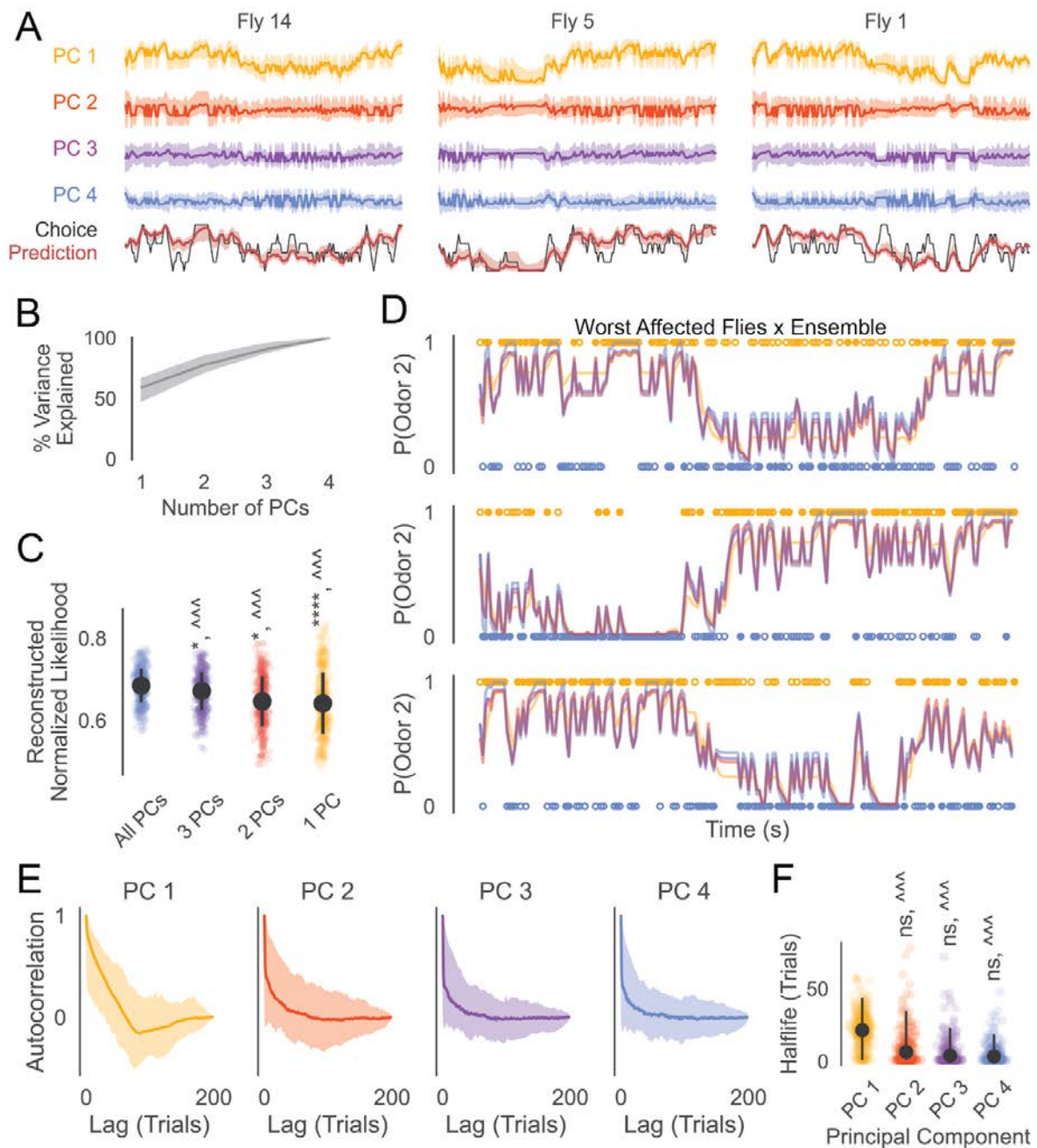


Figure 23. Dissecting the symmetric RqN reveals a possible separation of timescales that improves the performance of the RqN.

(A) Four Principal Components (PCs) of the hidden dynamics of the best symmetric RqN (symRqN(2) with four effective, hidden neurons) recovered using Principal Component Analysis (PCA) over the time axis. A 95% confidence interval (shaded area) was estimated from 25 trained networks from the ensemble shown alongside the smoothed choice probabilities from the data (black) and the prediction (red) (bottom; see methods). PCs were aligned by maximizing the correlation between

different trained networks to account for sign degeneracy in PCA methods. Data is shown for three flies from the training data (see subfigure D).

(B) Cumulative variance explained by the principal components with 95% confidence estimated over 25 trained networks from the ensemble.

(C) Contribution of each principle component was explored using a reconstruction Normalized Likelihood calculated by sequentially removing the PCs with the least contribution and then reconstructing the hidden dynamics fed to the decoder and policy to generate predictions for choice probabilities. Reconstructed normalized likelihood compared to log likelihood where all PCs are preserved using bootstrap-corrected two-sided Mann-Whitney-Wilcoxon test (stars for statistical significance; $p=0.0397$, 0.0131 , $7.62e-6$ respectively) and bootstrap-corrected matched-pairs rank biserial correlation effect size (carets for effect size; $r = 0.837$, 0.805 , 0.665 respectively) ($m=18$ flies, $n=25$ ensembles for bootstrap correction; see methods)

(D) Effect of removing principal components on the smoothed predicted choice probabilities for the three most affected flies in the ensemble most affected by removing the last three principal components. Color of the lines represent the different removed components described in subfigure C

(E) Autocorrelation plot of the different PCs with a 95% confidence interval estimated with 18 flies across 25 ensembles.

(F) Halflife of the autocorrelation lag quantified for the different PCs. Black bars represent a 95% confidence interval calculated using 18 flies across 25 ensembles. Lag for the first PC is compared to the rest using two-sided Mann-Whitney-Wilcoxon test (stars for statistical significance; $p=0.5011$, 0.4044 , 0.1682 respectively) and bootstrap-corrected matched-pairs rank biserial correlation effect size (carets for effect size; $r = 0.861$, 0.933 , 0.932 respectively) ($m=18$ flies, $n=25$ ensembles for bootstrap correction; see methods)

Kernel regression analysis on the PCs of RqNs suggest perseverance but fail to capture the sharp transitions

In order to understand what the PCs of the hidden dynamics were capturing, we tried to fit a simple linear kernel regressor from the past choices, rewards, and interaction term to the PCs of the hidden dynamics (Figure 24. A). We found by looking at the learned kernels that the first PC seems strongly influenced by the interaction term. There is a weak influence of the most recent choices, yet again suggesting perseverance behavior (Figure 24. B). However, the linear models fail to capture much of the variation in the latter PCs (Figure 24. B–D). As a result, we see that when we use the hidden dynamics from the PCs (Figure 24. E) predicted by the linear model, the quality of prediction sharply drops (Figure 24. F)

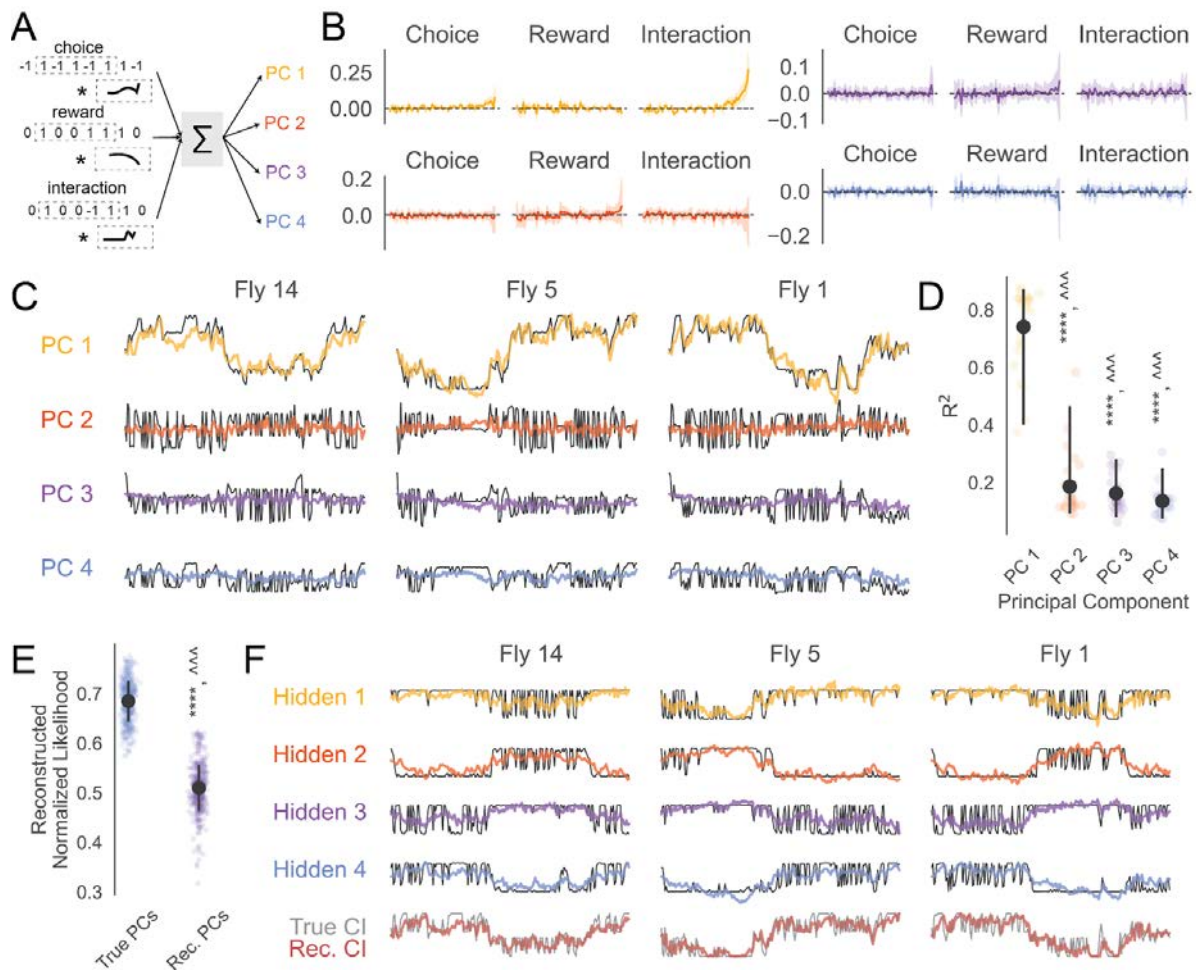


Figure 24. Kernel regression analysis to predict the principle components (PCs) of the hidden dynamics reveals the role of nonlinearity in the non-dominant PCs and suggests perseverance behavior.

(A) Kernel regression analysis applies convolutions with learned kernels on past time windows of choice, reward, and the interaction term (choice \times reward). It sums them together to predict the future value of a PC of the hidden dynamics for the best symmetric RqN (symRqN(2)).

(B) Learnt kernels for the choice, reward, and interaction term to predict the values of different PCs with a 95% confidence interval estimated from 25 trained networks from the ensemble.

(C) Predicted (colored) and actual values (black) of the principle components predicted by a linear model for the same flies as Figure 23.

(D) Coefficient of determination (R^2) for the linear fits for the four different PCs with the 95% confidence interval. The first PC is compared to the rest using two-sided Mann-Whitney-Wilcoxon test (stars for statistical significance; $p=1.78e-7$, $5.96e-8$, $5.960e-8$ respectively) and bootstrap-corrected matched-pairs rank biserial correlation effect size (carets for effect size; $r = 0.987$, 1.0 , 1.0 respectively) ($m=18$ flies, $n=25$ ensembles for bootstrap correction; see methods)

(E) Predicted (colored) and actual (black) hidden dynamics for the four hidden neurons shown alongside the true (black) and predicted (red) trial-wise choice probabilities. Rec. stands for reconstruction.

(F) Reconstruction Normalized Likelihoods compared between a prediction with the actual PCs and the linear regression reconstructed PCs compared using a two-sided Mann-Whitney-Wilcoxon test (stars for statistical significance; $p=7.629e-6$ respectively) and bootstrap-corrected matched-pairs rank biserial correlation effect size (carets for effect size; $r = 1.0$ respectively) ($m=18$ flies, $n=25$ ensembles for bootstrap correction; see methods)

Choice engineering for Fruit flies

Open-Loop choice engineering is prone to degeneracy

In order to experimentally test how well the Cognitive Q-learning models explain the behavior, we wanted to design experiments that would help us distinguish between models. For this purpose, we attempted to utilize a “choice engineering” paradigm developed by Dan & Loewenstein, 2019. The goal of choice engineering is to find a sequence of rewards such that the animal maximally chooses a target odor (and not a distractor odor) even when the total number of rewards associated with the different options is a constant (Dan & Loewenstein, 2019; Dezfouli et al., 2020). We use an open-loop paradigm where we search for a fixed sequence of 100 trials of reward-odor choice associations (where the reward delivered does not depend on the history of choices made by the animals; see methods).

Since the number of ways of organizing an equal number of rewards for two options (referred to as “reward schedules”) is astronomical (approximately 1058 for 50% reward probabilities across just 100 trials), it becomes impossible to test every possibility on behaving animals. However, the same space can be sampled computationally using stochastic optimization methods on models fitted to mimic flies with different value-learning rules, as described earlier. We do this for 5 of the fitted models using different stochastic optimization from random or structured initial conditions. We see that the population of optimized reward schedules has an increased bias that saturates over multiple generations (Figure 25. A). Different models saturate at different levels that often depend on the specific model being optimized and the schedule initialization with ‘Primed’ initialization (see methods), typically showing higher bias (Figure 25. B–F). We see that for some of the ‘Primed’ initialization optimization trajectories, the schedules sometimes get worse. This is because the fitness is estimated using a limited number of simulations and therefore is susceptible to stochastic variability in fitness even for the same schedule. Therefore, often there can be a small reduction in fitness by chance and the best schedule is only unique upto a certain amount of variability constrained by the number of simulations.

We look at the top ranking schedules for each of the models optimized along with the distribution of biases for 1000 agents of the tested model (Figure 26.). We find that

simpler models (RF-QL and LT-QL) are more strongly biased than the better models (F-RF-QL, DF-LT-QL, and DF-LT-OS-QL), which have a smaller maximum bias. A simple 'primacy'-like schedule is the maximally biasing schedule for the simpler models (Figure 26. A, C). For better models, it is harder to recognize a clear structure in the best-ranking schedules. The F-RF-QL model appears to show a stronger bias for 'primacy'-like schedules where the rewards are concentrated in two blocks with the target odor rewarded first (Figure 26. E). On the other hand, DF-LT-OS-QL has a stronger bias for more 'smeared' schedules where the rewards are more sparsely distributed over time rather than concentrated into blocks (Figure 26. I). The intermediate DF-LT-QL model appears to have a strong bias for schedules where the rewards are distributed in smeared schedules which roughly appear to be organized in alternating blocks (Figure 26. G).

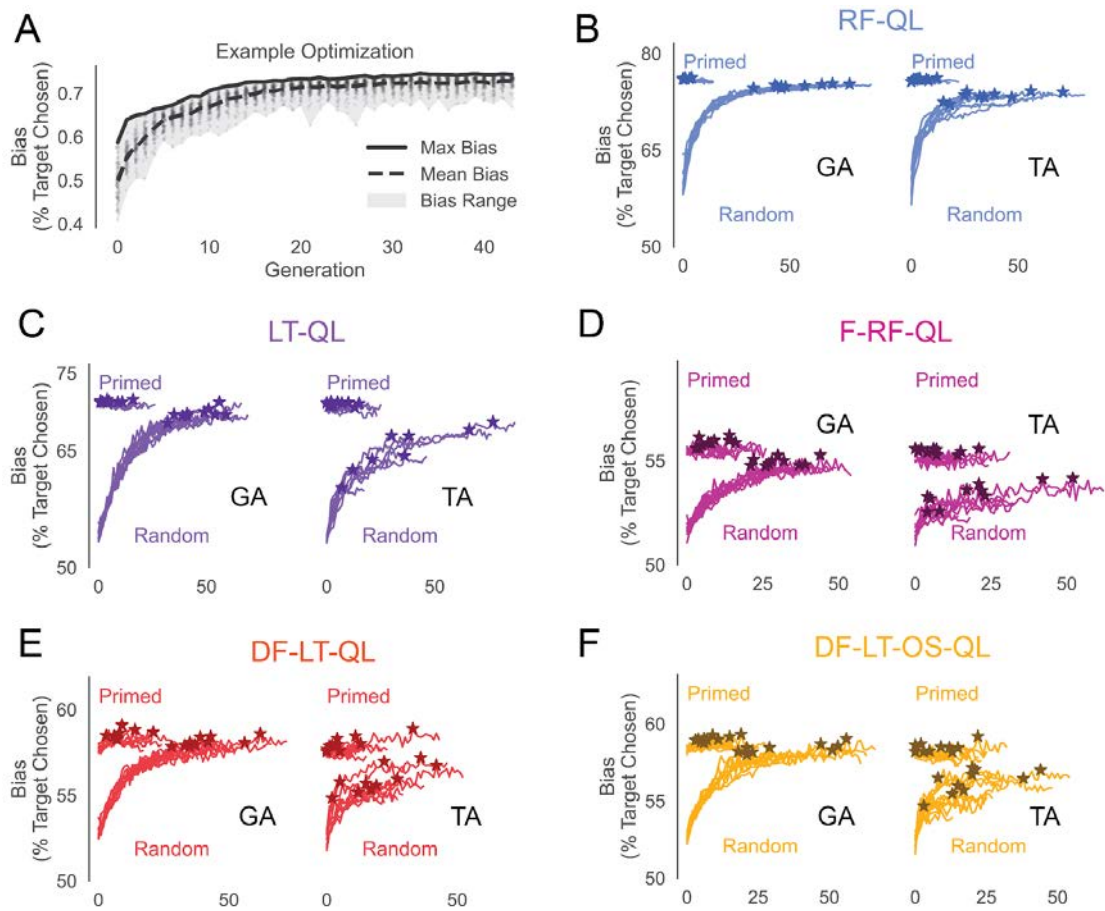


Figure 25. Optimization of choice engineering reward schedules.

(A) Example of a single stochastic optimization process for the DF-LT-OS-QL model using a genetic algorithm with the range of biases (shaded area), average bias (dotted line), and best bias (solid line) for the population of reward schedules.

(B–F) Traces of the best bias across multiple generations of stochastic optimization for five representative models. Replicates of different initializations (primed and random; see methods) and different optimization techniques are visualized. GA represents the Genetic Algorithm (left); TA represents Thermal Annealing (right). Stars mark the final “best” discovered schedule for each initialization.

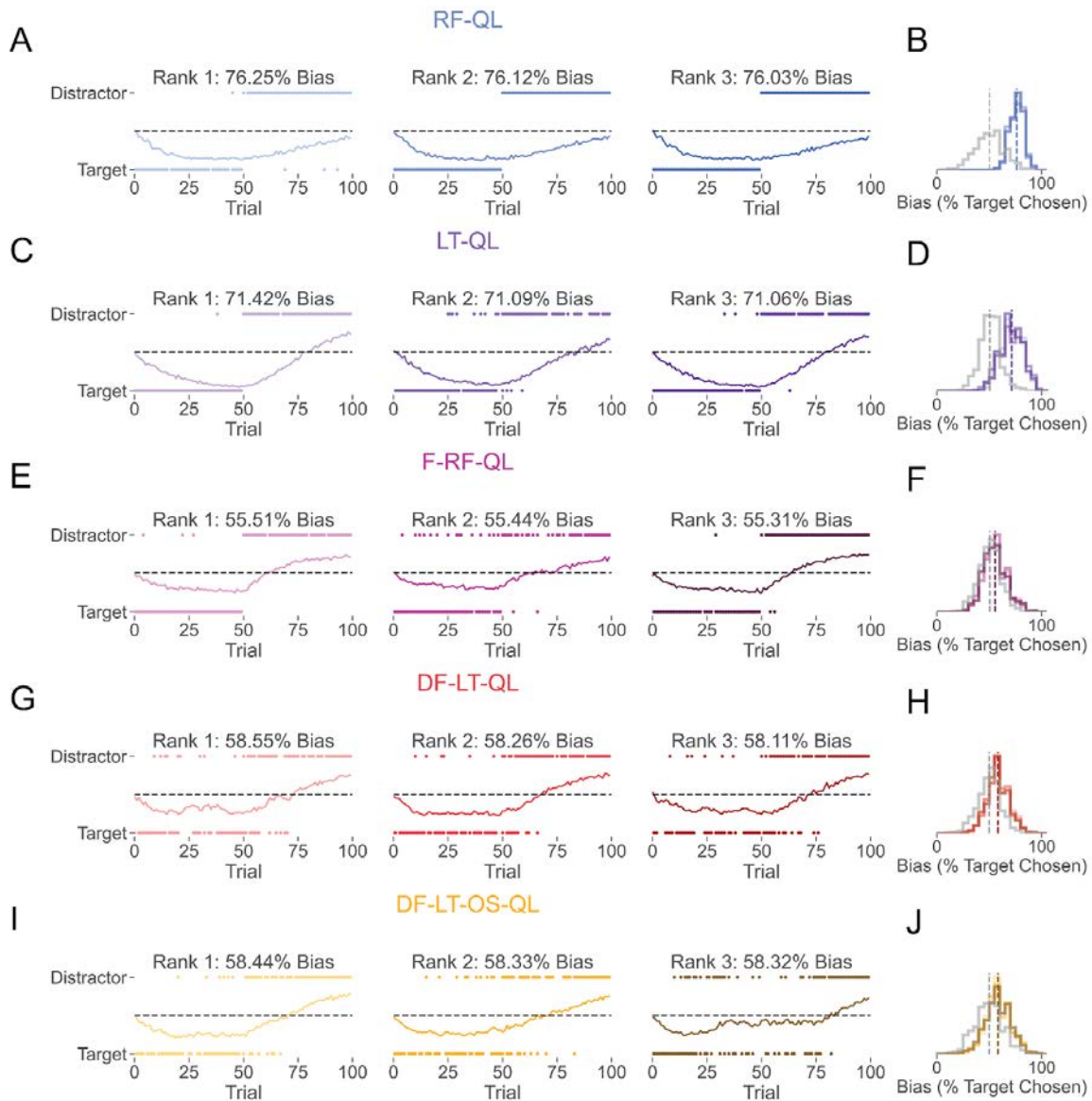


Figure 26. Choice Engineering provides candidate reward schedules for testing learning rules.

(A,C,E,G,I) Comparison of the top 3 maximally biasing reward schedules for five representative models. The top three schedules for each representative model are plotted. Dots represent the reward schedule i.e., rewards for the distractor and target odors for each trial. Absence of a dot represents the omission of reward on choice. Dotted lines represent no preference, and colored lines represent trial-wise bias for 1000 simulated agents.

(B,D,F,H,J) Distribution of overall biases over a 100 trial session for 1000 simulated agents for the top 3 maximally biasing schedules for five representative models compared to a schedule when equally spaced rewards are given identically on both odors (in gray).

In order to see if the predicted reward schedules are capable of actually biasing fly behavior, we run experiments on the single fly Y-maze described in Rajagopalan et al., 2022. We use the optimized reward schedules for two models: F-RF-QL and DF-LT-OS-QL, which have very similar average dynamics but have a fundamental difference in that the former does not implement a reward prediction error (RPE).

We observe that the schedules for F-RF-QL show robust trial-averaged learning with time, with the preference shifting strongly toward the target odor and then reversing after the reward associations transition to the distractor odor (Figure 27. A). On the other hand, the schedules for the DF-LT-OS-QL show a weaker trial averaged learning, but the preference persists for longer (Figure 27. B). However, when we look at the distribution of the bias for each fly and compare them between the two models, we find a small effect towards a higher bias in the optimally biasing schedules predicted by the DF-LT-OS-QL model, but at the small sample size, the result is subject to the large behavioral variability and therefore is not statistically significant (Figure 27. C). Therefore, we see a need to scale up our ability to run single fly choice experiments. Surprisingly, we note that the bias observed was stronger than what was predicted by the model optimizations (Figure 26. F, J)

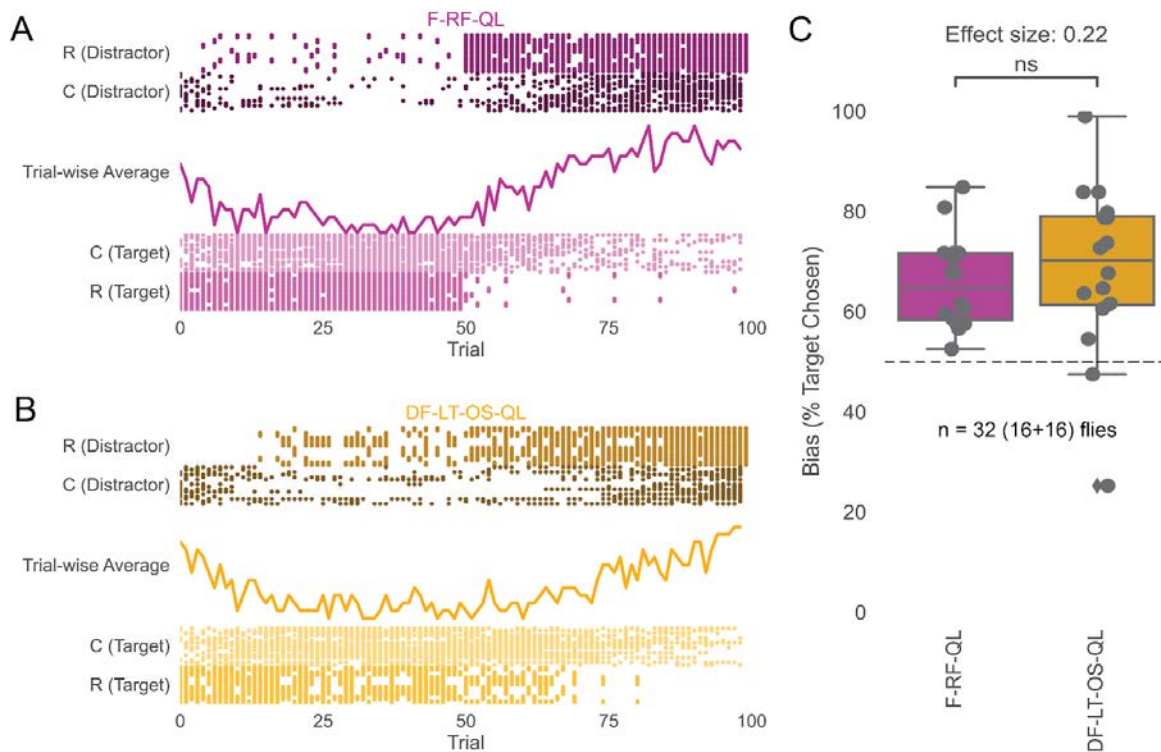


Figure 27. Optimal schedules predicted by DF-LT-OS-QL models only show a weak increase in bias than those predicted by F-RL-QL models.

(A–B) Reward (dark) and choice (light) sequences for 16 flies tested in a single fly Y-maze for reward schedules predicted by both F-RL-QL and DF-LT-OS-QL models. For each set of schedules, eight flies were run with MCH at the target and eight with OCT as the target. The trial-wise average preference is visualized in the middle.

(C) Bias of each fly (% target chosen over a 100 trial session) for the two sets of schedules are found to be statistically non-significant but show a slight increase in bias ($p = 0.2994$; Mann Whitney U Test; $\delta = 0.2188$; Cliff's delta effect size) for 16 flies for each set of schedules.

High-Throughput Y-Maze Experiments

In order to expand our ability to collect experimental data from flies performing dynamics choice experiments described in Rajagopalan et al., 2022, we designed a high-throughput behavioral rig capable of running 16 simultaneous experiments (see methods for details).

Optimizing the 16Y experimental setup

In order to test whether the flies can learn in the 16 Y-Arena, we ran a simple set of learning and reversal experiments with Octanol (OCT) and Methylcyclohexanol (MCH), which are considered the standard for fruit fly olfaction experiments. The flies were allowed to choose between OCT and MCH, present in two arms. No rewards were administered for the first 40 trials (Naive Phase), after which either OCT or MCH was rewarded for the subsequent 60 trials (Training Phase). After 60 trials of OCT/MCH rewards, the rewarded odor was switched for the subsequent 60 trials to evaluate whether the fly could forget a previous association and switch preference over time (Reversal Phase).

Strong learning, slow reversal and asymmetric preference is observed in typical OCT vs. MCH choice experiments

In our experiments, we observe that, as expected, starved flies can learn associations between the odor (OCT or MCH) and fictive sugar reward and then subsequently unlearn the association to reverse the preference toward a different odor in both possible directions (Figure 28. A–B). However, we find that for the OCT vs. MCH choice, there is a strong naive preference for the high-throughput behavioral rig (Figure 28. C; right), and the strong initial bias leads to a strong preference for MCH (Figure 28. C; left). After adjusting for the inherent non-linearity of the measure, we find that the training phase has a more substantial effect than the reversal phase across the two odors (Figure 28. D). The interaction of odor and order has the most evident effect on learning (ANOVA test; see Table 21). These results suggest that the first-odor reward pairing block has a much stronger effect on learning and is not easily reversed.

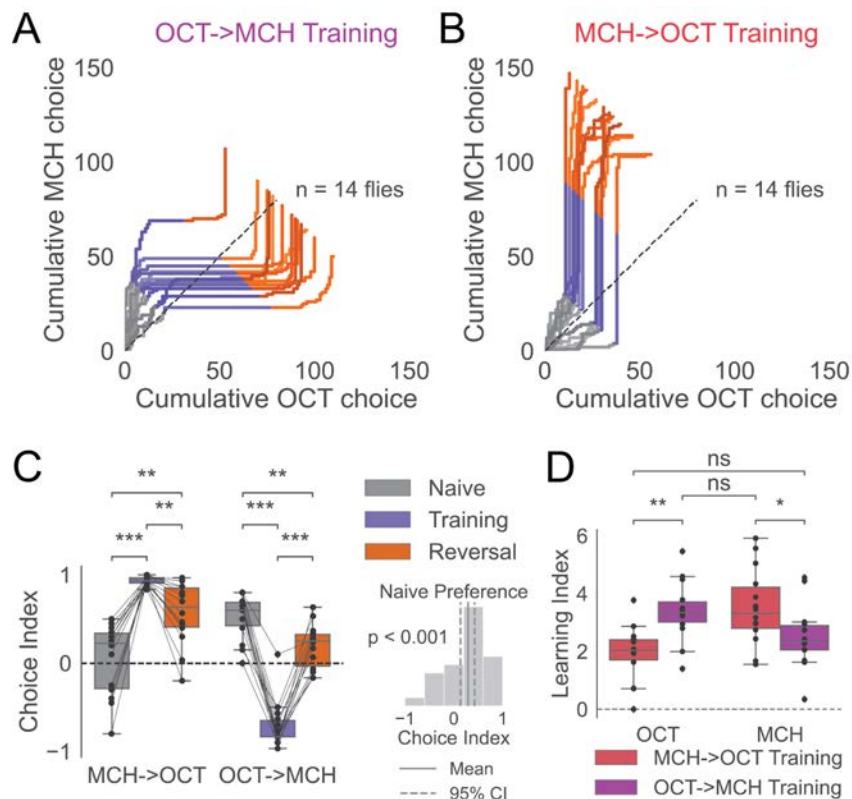


Figure 28. Strong Learning and asymmetric preference are observed for 24 hr starved flies in a high-throughput behavioral rig.

(A–B) Cumulative choices of OCT and MCH over time in experiments with 40 unrewarded trials (Naive) followed by 60 trials of pairing OCT(A)/MCH(B) with reward (Training), followed by 60 trials of pairing the opposite odor, i.e., MCH(A)/OCT(B) with certain reward (Reversal). The slope of the curve gives instantaneous preference.

(C) Choice index (+ve is MCH preference, -ve is OCT preference; see methods) quantified across the three phases (left). Values are compared using two-sided paired samples Mann-Whitney-Wilcoxon test (stars for statistical significance; see Table 19). Overall naive preference with a 95% confidence interval (right) compared to zero with a two-sided one-sample t-test (stars for statistical significance; $p = 9.99e-04$).

(D) Learning index (+ve is reward association for the paired odor; see methods) quantified for the two odors under the training and reversal condition compared using two-sided Mann-Whitney U test (stars for statistical significance; see Table 20)

Unexpected dynamics of choice times for OCT vs. MCH can be explained using fly kinematics

While we found that the flies were capable of learning odor associations that biased their behavior, we found, contrary to the results from Rajagopalan et al., 2022, that the duration of the trials increased after training, followed by a decrease at the start of the reversal phase (Figure 29. A).

To understand the basis of this unexpected deviation, we looked at the processed kinematic variables extracted from the behavior (see methods). We find that the increase in choice time is accompanied by a significant decrease in the average instantaneous speed of the fly throughout the trial at the start of the training phase and an increase in the reversal phase (Figure 29. B). Similarly, this is accompanied by a significant increase in the time spent in the air arm at the start of the training phase. Note that this is where the animal was previously rewarded. A substantial decrease in the same at the start of the reversal is also observed (Figure 29. C). We find that both variables, $\log(\text{speed})$ and air residence time, are negatively correlated with the $\log(\text{length of the trials})$ and the length of the trials (Figure 29. D) together, explaining the choice times.

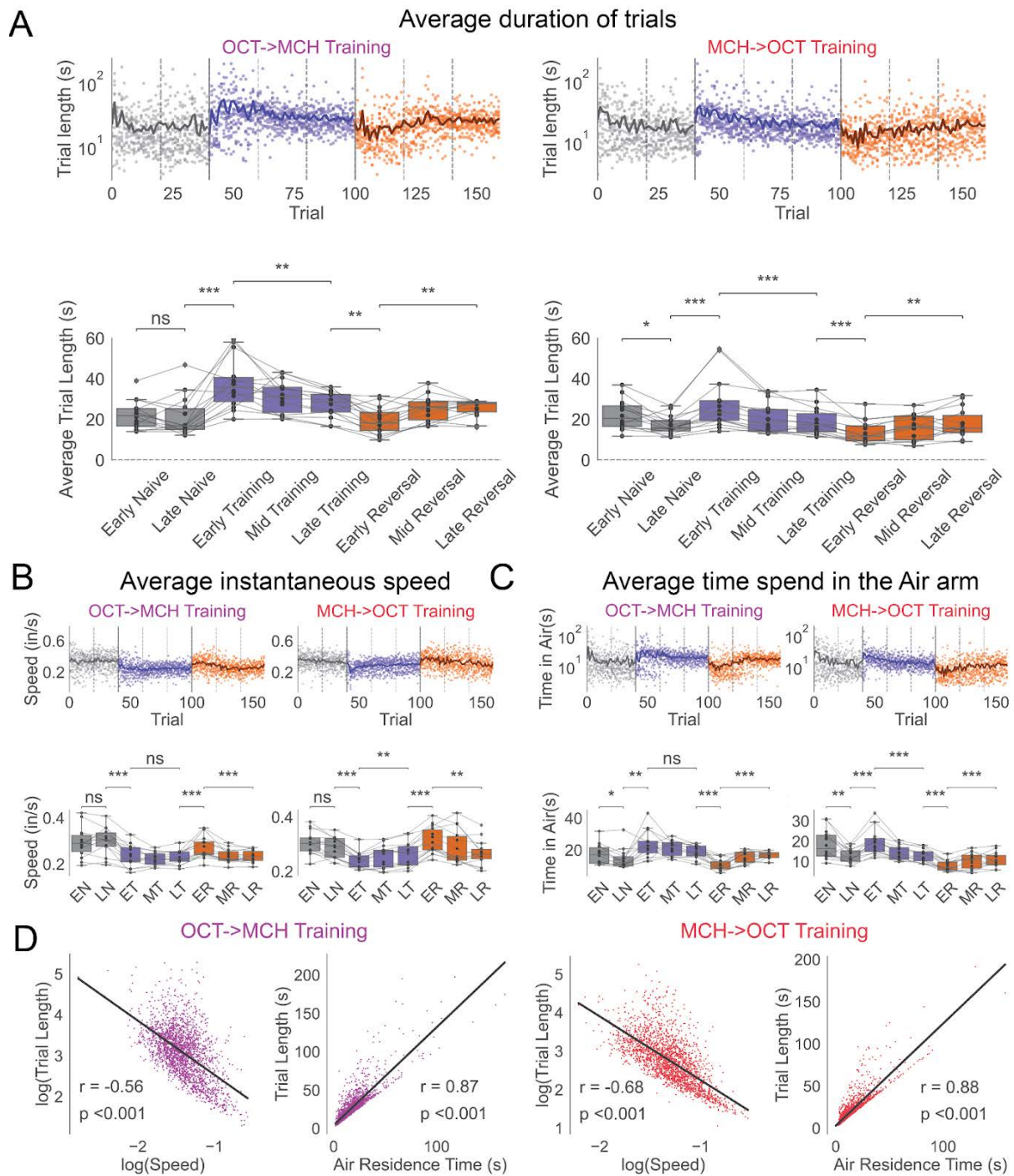


Figure 29. Slower choices on reward learning are explained by slower movement and residence in the last rewarded arm.

(A) Duration of each trial across all 14 experimental flies for the two learning experiments. Plotted along with the mean for each experiment (black) (top). Binned average trial times for flies across 20-trial subdivisions of the experimental phases (bottom) compared using two-sided paired samples Mann-Whitney-Wilcoxon test (stars for statistical significance; see Table 18)

(B) Average instantaneous speed in a trial across all 14 experimental flies for each of the two learning experiments. Plotted along with the mean for each experiment (black) (top). Binned average speeds for flies across 20-trial subdivisions of the experimental phases (bottom) compared using two-sided paired samples Mann-Whitney-Wilcoxon test (stars for statistical significance; see Table 18)

(C) Average time spent in the air arm for a trial across all 14 experimental flies for the two learning experiments. Plotted along with the mean for each experiment (black) (top). Binned average time spent in the air arm for flies across 20-trial subdivisions of the experimental phases (bottom) compared using two-sided paired samples Mann-Whitney-Wilcoxon test (stars for statistical significance; see Table 18)

(D) Log of trial duration compared to a log of average instantaneous speed (left) and trial duration compared to time spent in the air arm (right) for every trial across 14 flies for each experiment using Pearson's correlation ($p=1.51e-182$, 0.0, $2.37e-299$, 0.0 from left to right).

EN: Early Naive Phase; LN: Late Naive Phase; ET: Early Training Phase; MT: Mid Training Phase; LT: Late Training Phase; ER: Early Reversal Phase; MR: Mid Reversal Phase; LR: Late Reversal Phase.

Multiple kinematic factors underlie observed choice dynamics

However, next, we wanted to understand what factors influence the actual choices made by the flies, for which we looked at how the residence of the fly in the odorized arms changes over time (Figure 30. A).

We can see that in the initial naive trials, the flies have a slightly higher density at the ends of the MCH arm suggesting the flies move further into the arm with MCH. Still, with training, the density in the arm with the unrewarded odor goes down. However, it is faster when MCH is being trained first. Further, this effect is reversed when the reversal phase starts. The density in the new unrewarded arm reduced much less with the same number of trials for both odors but appeared to be more when MCH is rewarded (Figure 30. A). Quantifying these results we do see that with time, the flies spend more time in the arms that paired with the reward (Figure 30. B). This could be because of multiple possible reasons: (a) the fly could be avoiding the other arms, (b) the fly has a stronger drive to move in the rewarded odor (c) the fly has a stronger preference to enter the arm. When we quantify these factors, we find that all of these are true and influence behavior (Figure 30. C–E). The fly rapidly prefers the arm with the odor paired with the reward and moves faster in the same arm (Figure 30. D–E). Over training, this effect becomes more robust and is accompanied by an increase in the number of times it rejects the odor that is not paired with reward (Figure 30. C). All these trends are then flipped during the reversal phase. The effects are more substantial when the reversal phase pairs MCH with reward.

Subsequently, we tested different pairs of odors to find the right experimental conditions to study the dynamics of choice in the high-throughput rig. Firstly, we try PA (pentyl acetate) and EL (ethyl lactate), which, other than OCT and MCH, are typically also used as standard odors in fruit fly olfaction experiments (Campbell et al., 2013).

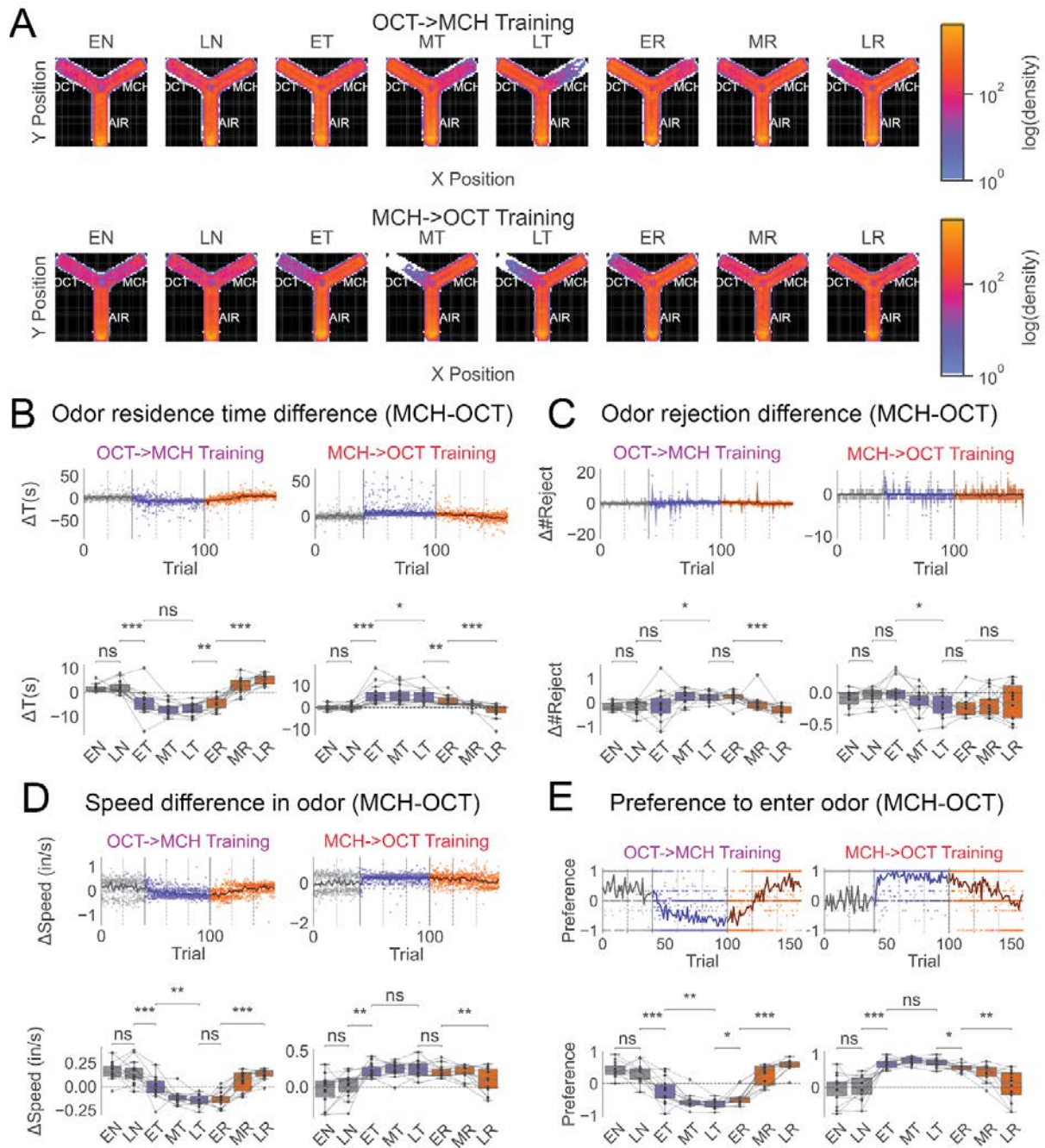


Figure 30. Change in odor preference as a function of reward history is a consequence of multiple kinematic factors.

(A) Residence of flies in the Y-arena (oriented to odor identity; left arm is OCT, the right arm is MCH; bottom is air) across each subdivision of the experimental phases in both experiments.

(B) Difference in the time spent in the MCH arm and the OCT arm in every trial across all 14 experimental flies for each of the two learning experiments. Plotted

along with the mean for each experiment (black) (top). The binned difference for flies across 20 trial subdivisions of the experimental phases (bottom) compared using two-sided paired samples Mann-Whitney-Wilcoxon test (stars for statistical significance; see Table 18)

(C) Difference in the number of times MCH is rejected and OCT is rejected in every trial across all 14 experimental flies for each of the two learning experiments. Plotted along with the mean for each experiment (black) (top). The binned difference for flies across 20 trial subdivisions of the experimental phases (bottom) compared using two-sided paired samples Mann-Whitney-Wilcoxon test (stars for statistical significance; see Table 18)

(D) Difference in the average instantaneous speed in the MCH arm and the OCT arm in every trial across all 14 experimental flies for each of the two learning experiments, along with the mean for each experiment (black) (top). The binned difference for flies across 20 trial subdivisions of the experimental phases (bottom) compared using two-sided paired samples Mann-Whitney-Wilcoxon test (stars for statistical significance; see Table 18)

(E) Difference in the fraction of times the MCH arm is entered, and the OCT arm is entered in every trial across all 14 experimental flies for each of the two learning experiments. Plotted along with the mean for each experiment (black) (top). The binned difference for flies across 20 trial subdivisions of the experimental phases (bottom) compared using two-sided paired samples Mann-Whitney-Wilcoxon test (stars for statistical significance; see Table 18)

EN: Early Naive Phase; LN: Late Naive Phase; ET: Early Training Phase; MT: Mid Training Phase; LT: Late Training Phase; ER: Early Reversal Phase; MR: Mid Reversal Phase; LR: Late Reversal Phase.

Asymmetric naive preference and asymmetric non-specific Learning is observed in PA vs. EL choice experiments across different reward probabilities

PA (pentyl acetate) vs. EL (ethyl lactate) choices were tested under a shorter training/reversal experiment with 10 naive phase trials, 45 training phase trials, and 45 reversal phase trials to speed up the experimental throughput. Different reward probabilities during the training/reversal phase were tested to verify that flies can distinguish between different reward uncertainties. We observe that while flies can learn strong PA vs. EL associations at high reward probabilities, the learning becomes asymmetric. Especially at low reward probabilities, the flies seem to learn stronger associations (Figure 31. A, C, D), and the effect is strengthened when paired with rewards first (Figure 31. D).

There is also a robust naive preference for EL (Figure 31. C). Moreover, we also observe non-specific learning where a fly learns a positive association with the unrewarded odor (-ve learning index), which appears to happen more often when EL is the unrewarded odor (Figure 31. D). We see that all three factors: odor identity, order of training, and reward probability, strongly affect the learning (ANOVA test; see Table 22).

Therefore, we needed a different odor combination with more balanced learning between the two odors. Since Hexanal (HAL) and 6-Methyl-5-hepten-2-one (MHO) are known to show similar levels of Kenyon cell activity on odor exposure (Honegger et al., 2011), we next attempted to pair HAL and MHO with rewards.

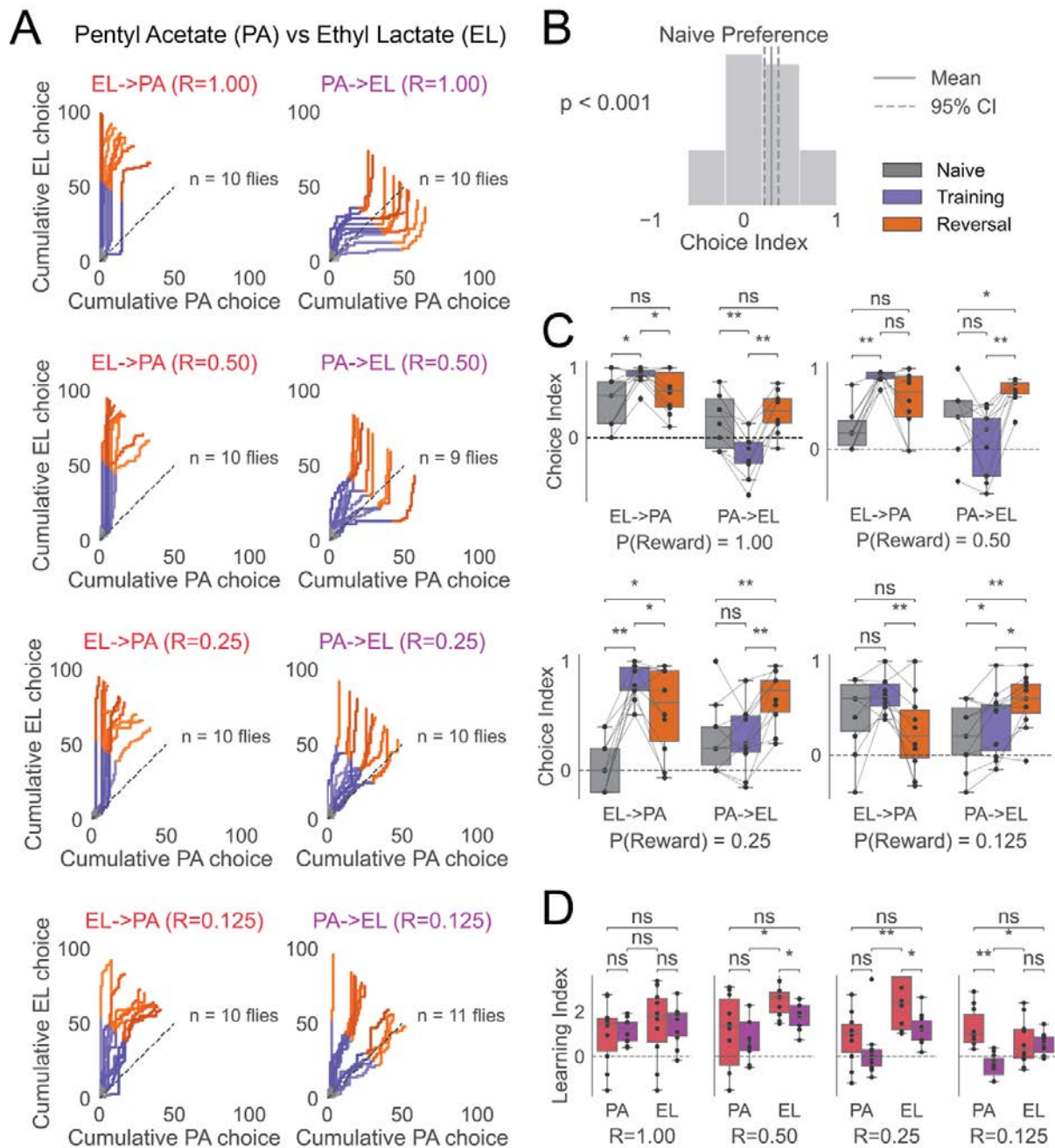


Figure 31. PA vs. EL choices show asymmetric, non-specific learning, especially at low reward probabilities, and a naive preference toward EL in 24 hr-starved flies.

(A) Cumulative choices of PA and EL over time in experiments with 10 unrewarded trials (Naive) followed by 45 trials of pairing EL/PA with reward (Training), followed by 45 trials of pairing the opposite odor, i.e., PA/EL with reward (Reversal). The reward pairing is varied to have different reward $P(R)$ probabilities = 0.125, 0.25, 0.5, and 1. The slope of the curve gives instantaneous preference.

(B) Overall naive preference with 95% confidence interval compared to zero with a two-sided one-sample t-test ($p = 0.00$) quantified using choice index (+ve is EL preference, -ve is PA preference; see methods).

(C) Choice index quantified across the three experimental phases and reward probabilities compared using two-sided paired samples Mann-Whitney-Wilcoxon test (stars for statistical significance; see Table 19).

(D) Learning index (+ve is reward association for the paired odor; see methods) quantified for the two odors under the training and reversal condition for different reward probabilities compared using two-sided Mann-Whitney U test (stars for statistical significance; see Table 20)

Symmetric Learning is observed MHO vs. HAL choice experiments across starvation states

To test the effect of starvation on the learning behavior, we also tested MHO vs. HAL choices in a 10 naive phase, 45 training phase, and 45 reversal phase experiment done at three different starvation conditions: 4-13 hours of starvation, 28-37 hours of starvation and 51-64 hours of starvation.

We find that the flies can form a weak association with both odors at low starvation levels. However, at higher levels of starvation, the learning effect becomes apparent (Figure 32. A–B). We also see no significant difference between the learning of different odors or the order in which they are learned (Figure 32. C). The only significant effect is starvation (ANOVA test; see Table 23). We see a significant naive preference toward HAL at lower starvation levels, but it disappears at very high starvation levels (Figure 32. D). Since we require moderate learning effects, we use HAL and MHO for the rest of our experiments at 13-30 hours of starvation.

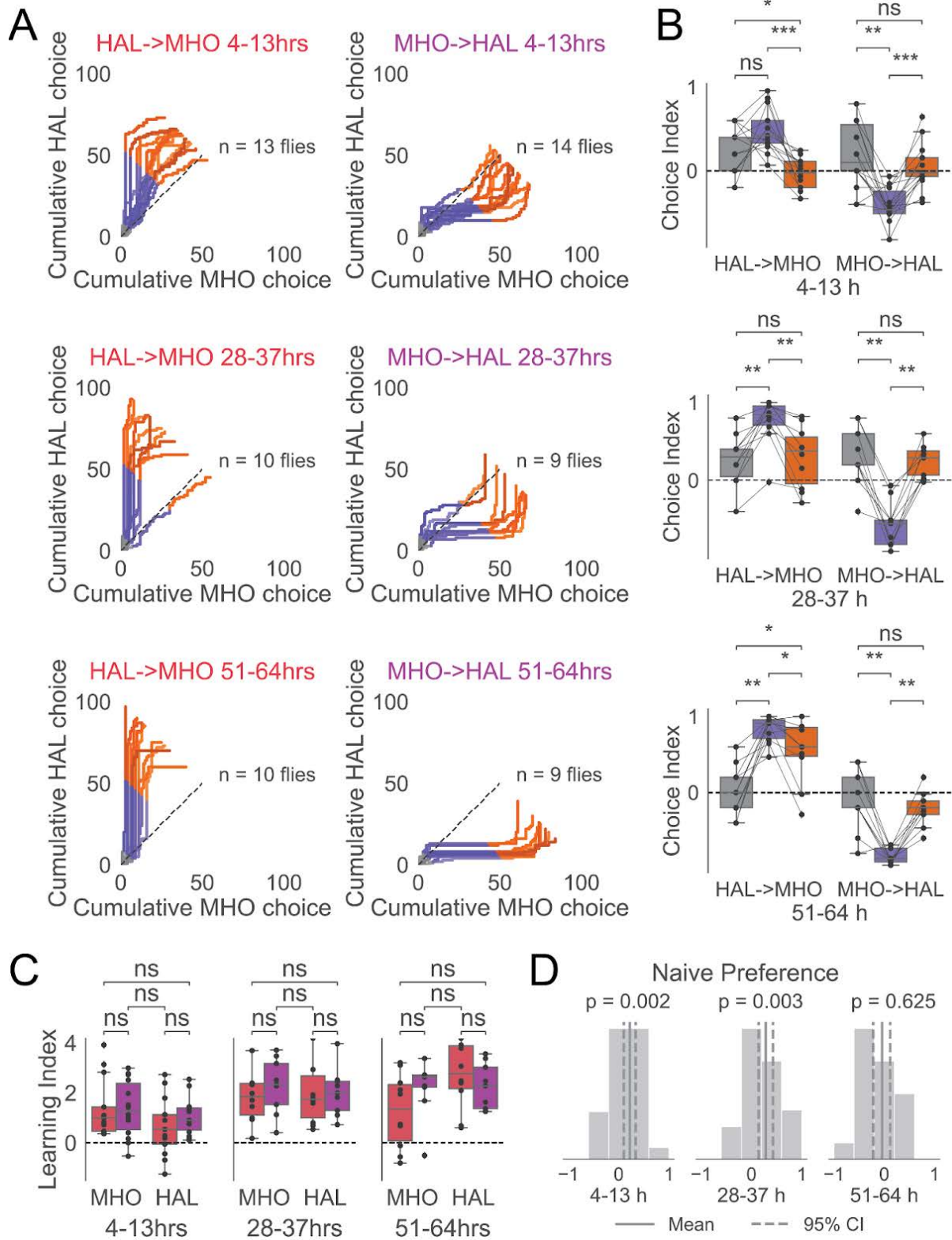


Figure 32. MHO vs. HAL choices show symmetric learning across starvation states with starvation-sensitive naive preference.

(A) Cumulative choices of HAL and MHO over time in experiments with 10 unrewarded trials (Naive) followed by 45 trials of pairing HAL/MHO with certain reward (Training), followed by 45 trials of pairing the opposite odor, i.e., MHO/HAL with reward (Reversal). The experiments were performed at different levels of starvation. The slope of the curve gives instantaneous preference.

(B) Choice index (+ve is HAL preference, -ve is MHO preference; see methods) quantified across the three experimental phases and levels of starvation. Values are compared using two-sided paired samples Mann-Whitney-Wilcoxon test (stars for statistical significance; see Table 19).

(D) Learning index (+ve is reward association for the paired odor; see methods) quantified for the two odors under the training and reversal condition for different probabilities compared using two-sided Mann-Whitney U test (stars for statistical significance; see Table 20)

(B) Overall naive preference with 95% confidence interval across starvation levels compared to zero with a two-sided one-sample t-test using choice index (+ve is EL preference, -ve is PA preference; see methods).

Mohanta (2022) “Variable Block” dataset

In order to build an in-depth model of fly behavior, we needed data from tasks that broadly samples the space of choice behavior that we can observe in a fruitfly. For this purpose, we collect data and analyze a dataset of 132 flies performing 22 different “Variable Block” experiments (see methods) (Figure 33. A) with six flies for each experiment with three flies for each possible pairing of odor with reward (HAL rewarded first, i.e., ‘forward’ experiments/ MHO rewarded first, i.e., ‘reciprocal’ experiments).

We observe that the flies seem to replicate the result of operant matching in the experiments with previously observed undermatching. Further, more robust matching appears to be concentrated in the longer blocks (warm color; Figure 33. B). Further, the dataset also contains examples of strong and fast learning (Fly 74 vs. 41; Figure 33. C), broadly samples the task space (see methods), and somewhat uniformly spans data from different transitions in odor-associated baiting probabilities. We then divide the data into two sets: a) training data: 2 randomly chosen flies from each “forward” and “reciprocal” experiments; b) test data: rest of the flies, i.e., one random fly from each “forward” and “reciprocal experiment”.

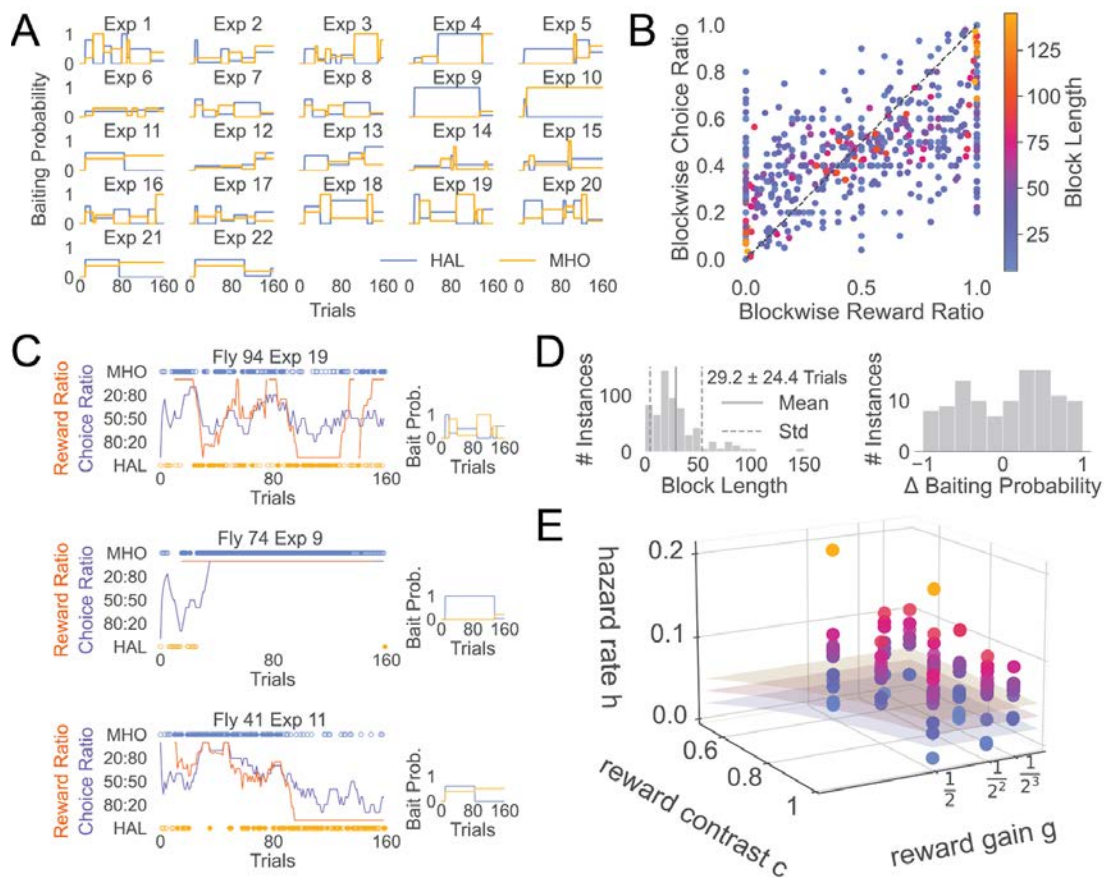


Figure 33. Mohanta (2022) “Variable Block” dataset shows probability matching across a broad sample of the space of dynamic baited-reward 2-alternative forced choice tasks.

(A) Set of baiting probabilities for 22 “forward” experiments that were run on three different flies each, along with three flies on “reciprocal” experiments where the odor identities were flipped.

(B) Blockwise reward ratios and Blockwise choice ratios are compared for the dataset colored by the length of the block in which the ratio is calculated.

(C) Three random example choice trajectories (left) from the data with the associated baiting probabilities (right). Orange and Blue dots in the reward schedule represent choosing Odor 1 and 2, respectively. Filled and empty dots represent the rewarded choice and unrewarded choices, respectively. The red and purple lines represent the reward and choice ratios calculated for 10 trials before the current trial (including the current trial).

(D) Histograms of the length of the blocks along with the 95% confidence interval (left) and the change in baiting probabilities of an odor between two successive blocks.

(E) Points of the task space (see methods) defined by reward gain, reward contrast, and estimated hazard rates sampled in the experiments with hazard rate estimated by looking at the reciprocal of the length of blocks observed in an experiment under each condition.

Constrained matching law models can explain the observed behavior

In order to set a baseline on how well a simple operant matching strategy can explain the behavior, we fit the training data on simple constrained matching law models with different windows of integration (see methods). We find that the model which uses the last five trials to predict the subsequent choice fits and predicts the data the best (Figure 34. A). Most matching models seem to capture aspects of the behavior, but with longer integration windows, the predictions fail to capture the short-term dynamics of choice (Figure 34. B)

We also find that the fit parameters for all the matching law models suggest a matching bias of 0.22 (Table 10), which implies a stronger preference for HAL, consistent with our previous experiments. We also see that the best constrained matching law model predicts a strong tendency to match based on recent reward and choice ratio estimates since the matching strength ($s = 0.92$) is close to 1.

Model	Matching Bias (b; +ve is HAL)	Matching Strength (s)	I_{max}
matching(5)	0.22 (0.12–0.33)	0.92 (0.79–1.04)	1.48 (1.26–1.73)
matching(10)	0.22 (0.12–0.32)	0.70 (0.63–0.79)	1.90 (1.62–2.20)
matching(15)	0.22 (0.12–0.31)	0.64 (0.57–0.71)	2.28 (1.95–2.67)
matching(30)	0.20 (0.12–0.28)	0.55 (0.49–0.62)	3.31 (2.75–3.92)
matching(60)	0.18 (0.10–0.26)	0.66 (0.58–0.75)	2.92 (2.36–3.53)

Table 10. Parameters for constrained matching law models

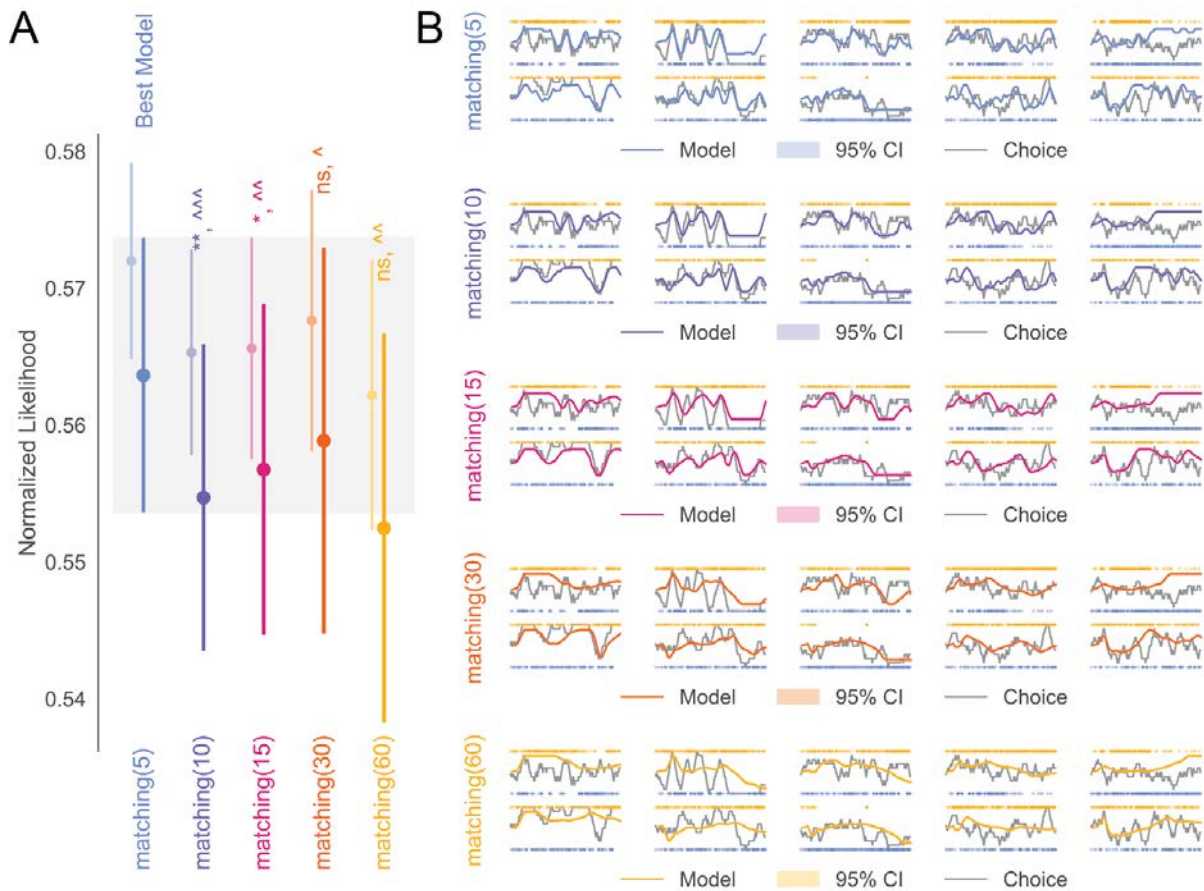


Figure 34. Constrained matching law models can predict future behavior with small integration windows.

(A) Comparison of the goodness of fit and predictive power estimated using Normalized Likelihood on training data and testing data, respectively, for different fits of the constrained matched law models (see methods) with different sizes of integration windows. Light and dark error bars represent the mean and standard error of training and test Normalized Likelihood fitted using 1000 bootstrapped samples on the training dataset. Test Normalized Likelihood of each of the models is compared to the best model (matching(5) - constrained matching law model with an integration time window of 5 trials) using a bootstrap-corrected two-sided paired samples Mann-Whitney-Wilcoxon test (stars for statistical significance) and bootstrap-corrected matched-pairs rank biserial correlation effect size (caret for effect size) ($m=44$ flies, $n=1000$ bootstraps; see methods). See Table 24 for p-values and effect sizes, including a comparison of training Normalized Likelihood using the same statistical measures.

(B) Smoothed predicted choice probabilities for ten random test flies with 95% confidence interval estimated from 1000 bootstrap fits overlaid on smoothed choice probabilities estimated from the data with a 10 trial window (see methods) for the five matching models.

Logistic kernel regression models can capture the dynamics of the behavior through leaky integration

While matching law gives us an intuitive model of integrating over history, it is a heuristic that might explain the data well. It may only correlate with the computation the fly uses to choose between odors. Further, we needed to set a bound on how well we could explain fly behavior using direct linear integration of past information. We, therefore, look at how well an overparameterized logistic kernel regression model explains this behavior, an approach also previously used in Rajagopalan et al., 2022.

We find that while the model that considers the reward, choice, and choice-reward interaction for the last 60 trials fits the data the best, the model that considers the choice and the choice-reward interaction for the last 30 trials overall predicts the behavior in the test data the best. Nevertheless, the fit is not significantly different from the best matching model (Figure 35. A). Further, the models that include the interaction term track and predict the preferences' changes decently well (Figure 35. B).

Looking at the kernel regression coefficients, we see that the influence of the interaction term is essential with greater weights for recent history that decays exponentially into the past across all models (Figure 35. C). Whenever the choice terms are included, there appears to be some positive influence from the choice (peaking at around five trials, i.e., the resolution of defined blocks). However, it does not appear to be a very strong effect (Figure 35. C; Table 26, Table 27, Table 29).

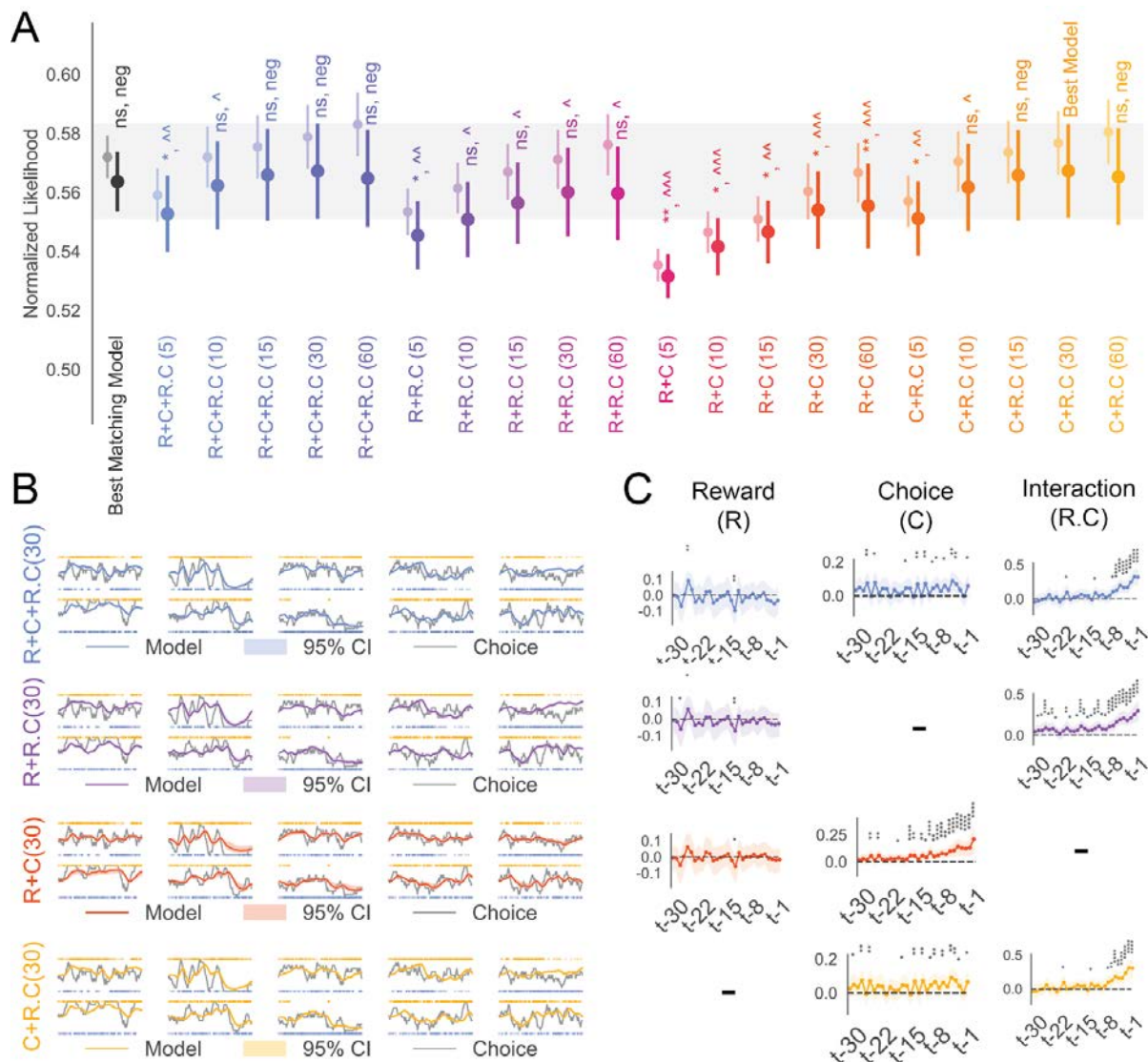


Figure 35. Logistic kernel regression models perform only as well as the best matching models

(A) Comparison of the goodness of fit and predictive power estimated using Normalized Likelihood on training data and testing data, respectively, for different logistic kernel regression models (see methods) with different sizes of integration windows. Light and dark error bars represent the mean and standard error of training and test Normalized Likelihood fitted using 1000 bootstrapped samples on the training dataset. Test Normalized Likelihood of each of the models is compared to the best model (C + R·C (30) Model with a 30 trial integration window) using a bootstrap-corrected two-sided paired samples Mann-Whitney-Wilcoxon test (stars for statistical significance) and bootstrap-corrected matched-pairs rank biserial correlation effect size (carets for effect size) (m=44 flies, n=1000 bootstraps; see

methods). See Table 25 for p-values and effect sizes, including a comparison of training Normalized Likelihood using the same statistical measures.

(B) Smoothed predicted choice probabilities for ten random test flies with a 95% confidence interval estimated from 1000 bootstrap fits overlaid on smoothed choice probabilities estimated from the data with a 10-trial window (see methods) for the four models with a 30-trial integration window.

(C) Kernel Regression Coefficients ($K_{x,t}$; see methods) for different terms estimated for the four models with a 30-trial integration window across 1000 bootstrap fits compared from zero using a two-sided bootstrap test (stars for significance). See Table 25, Table 26, Table 27, Table 28, Table 29 for the values of the coefficients and associated statistics.

Q-Learning models reliably capture the dynamics of choice with a small number of parameters, however the effect of adding cognitive features is non-trivial

Next, we tried replicating the analysis we did with the Rajagopalan (2022) "Fixed Block" dataset to try and model the behavior using Q-learning models with increasing cognitive complexity. We find that similar to the previous results, while there is large variability in the quality of the prediction, models that include temporal discounting and perseverance implemented through action prediction error or omission sensitivity, seem to explain and predict the data better than the other models (Figure 36. A). There are a few differences that we observe. Firstly, the best model differs from the previous fits with a habit-value arbiter q-learning model without forgetting (LT-HV-QL) explaining the data better than other models. However, there is a large degree of variability. The best model is not significantly different from the next best, which includes perseverance and forgetting at an independent timescale from learning (Figure 36. A). We also observe that the model with just learning-independent forgetting and temporal discounting (DF-LT-QL) is the third best model (after adjusting for the number of parameters) and is not statistically different from the best models. Suggesting the explanatory power of perseverance is not very strong.

We also observe that with the increasing cognitive complexity, the models appear to improve somewhat at tracking preference dynamics (Figure 36. B–G) with diminishing returns, as previously seen in the Rajagopalan (2022) "Fixed Block" dataset. It is important to note that the best q-learning models predict data (Normalized Likelihood = 0.5756 ± 0.0169 SE) only slightly better than the best linear model (Normalized Likelihood = 0.5673 ± 0.0159 SE) and do not appear to be very different. Next, we wanted to see where the difference lies between the estimated parameters of the models trained on the Rajagopalan (2022) "Fixed Block" dataset (Table 8, Table 9) and the Mohanta (2022) "Variable Block" dataset (Table 11, Table 12). We find that most of the significant differences are in the policy parameters (weights and intercepts to transform value/habits to acceptance probabilities), with minor changes to other parameters scattered around the space of models (Figure 37.). Also, we see that cognitive features seem to have a more substantial explanatory power in the variation observed in the parameters than observed in the previous dataset (ANOVA Test; Table 31).

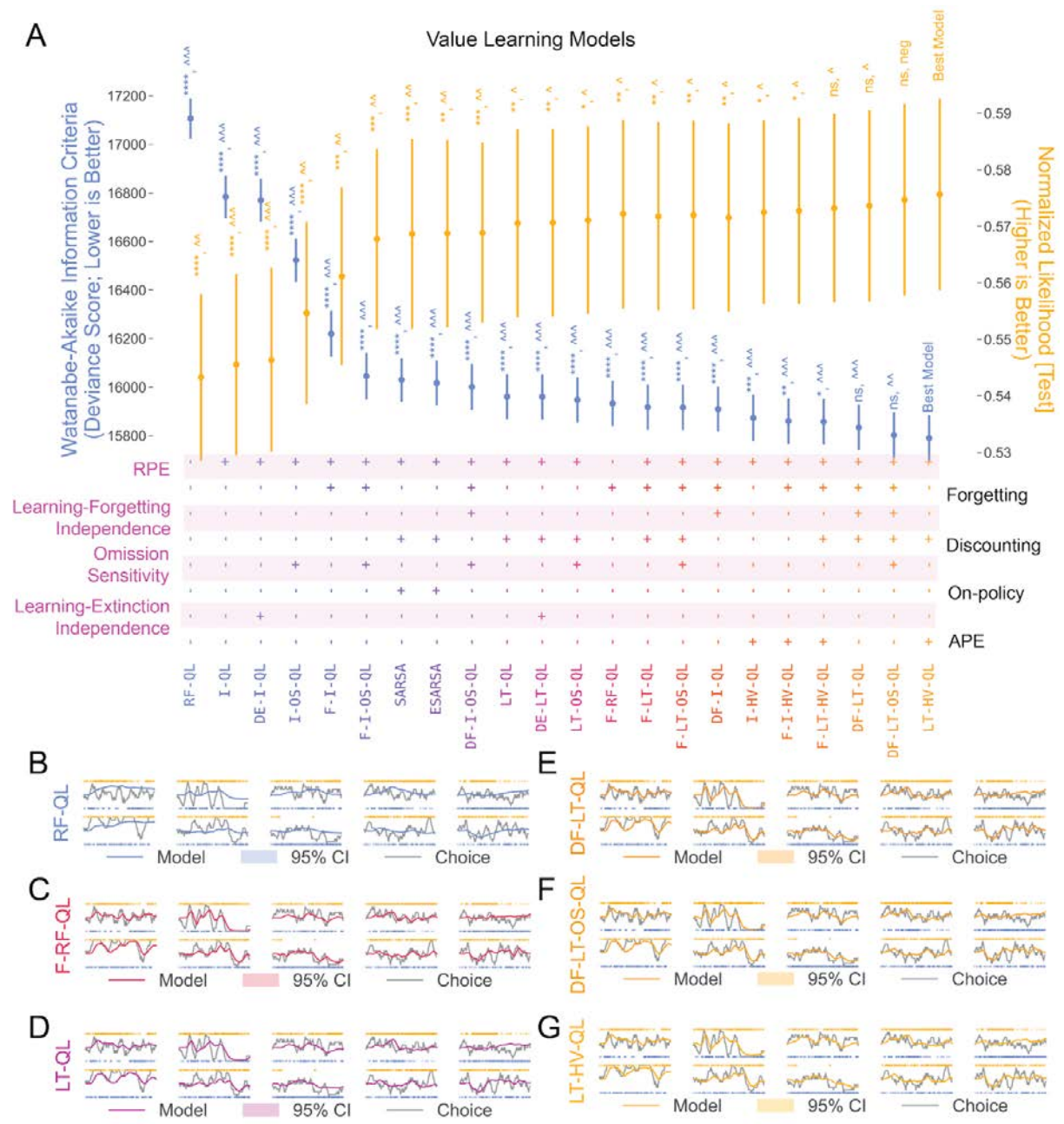


Figure 36. Results from fitting Q-learning models on Mohanta (2022) “Variable Block” dataset roughly reproduces the results from Rajagopalan “Fixed Block” (2022) dataset.

(A) Goodness of fit is estimated using the deviance-scaled Watanabe-Akaike Information Criterion (WAIC; blue). The difference of each model’s WAIC relative to the best model is compared using a two-sided z-test (stars for statistical significance; see methods) and Cohen’s d (carets for effect size). Predictive accuracy estimated using Normalized Likelihood [Test] (yellow) is compared relative to the best model using a bootstrap-corrected two-sided paired samples t-test (m=44 flies, n=100

bootstraps; see methods) (stars for statistical significance and paired Cohen's d (carets for effect size). The '+' and '-' symbols at the bottom signify which cognitive features (see Table 7) are included in the model. Error bars show Standard Error for WAIC and Normalized Likelihood [Test]. See Table 30 for statistics, p-values, and effect sizes.

(B–G) Smoothed predicted choice probabilities for ten random test flies with a 95% confidence interval estimated from 100 bootstrap fits overlaid on smoothed choice probabilities estimated from the data with a 10-trial window (see methods) for the six representative models from the dataset.

Model	α	κ	τ	γ	θ	m	c
Forgetting Long-Term Habit-Value Arbitrator Q-Learning	-	-	-	0.71 (0.64–0.78) [N=11944; R=1.0]	-	2.97 (2.19–3.85) [N=9267; R=1.0]	-3.8 (-4.31–-3.29) [N=19305; R=1.0]
Long-Term Habit-Value Arbitrator Q-Learning	-	-	-	0.88 (0.84–0.91) [N=9068; R=1.0]	-	4.14 (3.27–5.11) [N=8158; R=1.0]	-6.3 (-7.16–-5.47) [N=9067; R=1.0]
Forgetting Immediate Habit-Value Arbitrator Q-Learning	-	-	-	-	-	5.66 (4.81–6.56) [N=14200; R=1.0]	-4.74 (-5.56–-4.0) [N=22256; R=1.0]
Differential Forgetting Immediate Q-Learning	0.04 (0.04–0.05) [N=11950; R=1.0]	0.32 (0.27–0.36) [N=13457; R=1.0]	-	-	-	6.72 (5.99–7.46) [N=11113; R=1.0]	-4.19 (-4.78–-3.62) [N=13741; R=1.0]
Forgetting Immediate Q-Learning	0.07 (0.06–0.08) [N=11051; R=1.0]	-	-	-	-	3.82 (3.57–4.07) [N=8967; R=1.0]	-3.94 (-4.77–-3.2) [N=8747; R=1.0]
Immediate Q-Learning	0.03 (0.03–0.03) [N=10053; R=1.0]	-	-	-	-	5.54 (4.95–6.12) [N=8273; R=1.0]	-2.9 (-3.2–-2.62) [N=9349; R=1.0]
Forgetting RPE-free Q-Learning	0.42 (0.36–0.47) [N=8851; R=1.0]	-	-	-	-	0.58 (0.52–0.64) [N=8163; R=1.0]	-3.41 (-3.78–-3.05) [N=11053; R=1.0]
RPE-free Q-Learning	0.36 (0.05–0.83) [N=1916; R=1.0]	-	-	-	-	0.5 (0.1–1.29) [N=1900; R=1.0]	-2.16 (-2.31–-2.02) [N=5121; R=1.0]
Immediate Habit-Value Arbitrator Q-Learning	-	-	-	-	-	5.62 (4.74–6.5) [N=14355; R=1.0]	-4.74 (-5.54–-3.98) [N=22976; R=1.0]
Differential Extinction Long-Term Q-Learning	0.22 (0.19–0.25) [N=4883; R=1.0]	-	0.22 (0.19–0.25) [N=4862; R=1.0]	0.94 (0.92–0.97) [N=7498; R=1.0]	-	1.1 (0.98–1.23) [N=5685; R=1.0]	-7.94 (-8.67–-7.24) [N=5623; R=1.0]
Differential Forgetting Long-Term Q-Learning	0.29 (0.25–0.33) [N=8223; R=1.0]	0.04 (0.03–0.05) [N=8173; R=1.0]	-	0.88 (0.85–0.9) [N=7298; R=1.0]	-	1.15 (0.98–1.31) [N=6954; R=1.0]	-4.25 (-4.75–-3.76) [N=7423; R=1.0]
Forgetting Long-Term Q-Learning	0.18 (0.16–0.2) [N=9011; R=1.0]	-	-	0.82 (0.77–0.87) [N=8822; R=1.0]	-	1.43 (1.23–1.63) [N=8096; R=1.0]	-3.69 (-4.2–-3.23) [N=11586; R=1.0]

Long-Term Q Learning	0.22 (0.19–0.24) [N=7014; R=1.0]	-	-	0.95 (0.94–0.96) [N=7199; R=1.0]	-	1.09 (0.97–1.22) [N=6439; R=1.0]	-7.83 (-8.52–-7.19) [N=7638; R=1.0]
Differential Extinction Immediate Q Learning	0.05 (0.04–0.06) [N=7170; R=1.0]	-	0.02 (0.02–0.03) [N=10362; R=1.0]	-	-	3.78 (3.38–4.27) [N=6452; R=1.0]	-3.88 (-4.65–-3.15) [N=7658; R=1.0]
Expected SARSA	0.17 (0.15–0.19) [N=7529; R=1.0]	-	-	0.97 (0.96–0.99) [N=8462; R=1.0]	-	1.4 (1.27–1.53) [N=7206; R=1.0]	-6.96 (-7.54–-6.41) [N=7961; R=1.0]
SARSA	0.23 (0.2–0.26) [N=8378; R=1.0]	-	-	0.96 (0.95–0.98) [N=8603; R=1.0]	-	1.08 (0.96–1.21) [N=7975; R=1.0]	-6.7 (-7.24–-6.19) [N=8819; R=1.0]
Differential Forgetting Long-Term Omission Sensitive Q Learning	0.29 (0.26–0.33) [N=10126; R=1.0]	0.03 (0.02–0.04) [N=10841; R=1.0]	-	0.82 (0.78–0.86) [N=8399; R=1.0]	0.2 (0.13–0.28) [N=11174; R=1.0]	1.35 (1.14–1.57) [N=8428; R=1.0]	-4.63 (-5.2–-4.09) [N=9737; R=1.0]
Forgetting Long-Term Omission Sensitive Q Learning	0.17 (0.15–0.2) [N=9070; R=1.0]	-	-	0.79 (0.73–0.85) [N=7979; R=1.0]	0.05 (-0.01–0.11) [N=11314; R=1.0]	1.45 (1.24–1.66) [N=8193; R=1.0]	-3.82 (-4.43–-3.29) [N=12272; R=1.0]
Long-Term Omission Sensitive Q Learning	0.22 (0.19–0.25) [N=7548; R=1.0]	-	-	0.94 (0.92–0.96) [N=7451; R=1.0]	0.1 (0.04–0.16) [N=10661; R=1.0]	1.14 (0.99–1.28) [N=7424; R=1.0]	-8.19 (-8.9–-7.45) [N=8155; R=1.0]
Differential Forgetting Immediate Omission Sensitive Q Learning	0.18 (0.15–0.2) [N=11689; R=1.0]	0.04 (0.03–0.05) [N=13806; R=1.0]	-	-	0.56 (0.51–0.61) [N=11561; R=1.0]	3.73 (3.41–4.08) [N=12592; R=1.0]	-4.08 (-4.84–-3.34) [N=11625; R=1.0]
Forgetting Immediate Omission Sensitive Q Learning	0.09 (0.08–0.1) [N=12736; R=1.0]	-	-	-	0.29 (0.25–0.33) [N=15424; R=1.0]	3.39 (3.18–3.59) [N=11401; R=1.0]	-4.28 (-5.07–-3.46) [N=14054; R=1.0]
Immediate Omission Sensitive Q Learning	0.05 (0.05–0.06) [N=11180; R=1.0]	-	-	-	0.41 (0.37–0.45) [N=9533; R=1.0]	5.03 (4.74–5.35) [N=12484; R=1.0]	-5.13 (-5.8–-4.47) [N=9626; R=1.0]

Table 11. Q-Learning Model Fit Parameters for Mohanta (2022) "Variable Block" dataset. Mean (95% Credible Interval), Effective Sample Size (N) and Convergence (R) are provided. Parameters that do not meet out quality standards (see methods) are marked in bold. ANOVA results in Table 31

Model	α_v	α_h	θ_v	θ_h	w_v	w_h	w_b
Forgetting Long-Term Habit-Value Arbitrator Q-Learning	0.2 (0.18–0.24) [N=15696; R=1.0]	0.02 (0.01–0.03) [N=13875; R=1.0]	0.91 (0.76–1.0) [N=12446; R=1.0]	0.67 (0.39–1.0) [N=8574; R=1.0]	-0.86 (-2.05–0.39) [N=13479; R=1.0]	1.79 (0.88–2.74) [N=13639; R=1.0]	0.47 (-0.15–1.11) [N=9330; R=1.0]
Long-Term Habit-Value Arbitrator Q-Learning	0.37 (0.32–0.43) [N=14572; R=1.0]	0.07 (0.05–0.08) [N=10421; R=1.0]	0.3 (0.2–0.41) [N=8411; R=1.0]	0.95 (0.86–1.0) [N=14716; R=1.0]	-0.26 (-1.36–0.83) [N=14967; R=1.0]	-2.03 (-2.7–-1.38) [N=9021; R=1.0]	-1.5 (-2.16–-0.85) [N=7227; R=1.0]
Forgetting Immediate Habit-Value Arbitrator Q-Learning	0.18 (0.14–0.21) [N=20494; R=1.0]	0.08 (0.06–0.1) [N=16175; R=1.0]	0.34 (0.27–0.42) [N=13882; R=1.0]	0.97 (0.92–1.0) [N=15663; R=1.0]	0.99 (-0.59–2.6) [N=20749; R=1.0]	-3.3 (-3.98–-2.65) [N=13352; R=1.0]	-3.2 (-3.78–-2.64) [N=11518; R=1.0]
Immediate Habit-Value Arbitrator Q-Learning	0.34 (0.28–0.41) [N=19846; R=1.0]	0.1 (0.08–0.11) [N=16412; R=1.0]	0.21 (0.17–0.26) [N=13236; R=1.0]	0.97 (0.92–1.0) [N=19653; R=1.0]	0.51 (-0.44–1.5) [N=16099; R=1.0]	-2.7 (-3.23–-2.22) [N=14365; R=1.0]	-2.36 (-2.82–-1.9) [N=11676; R=1.0]

Table 12. Q-Learning Model Fit Parameters for Mohanta (2022) "Variable Block" dataset (contd). Mean (95% Credible Interval), Effective Sample Size (N) and Convergence (R) are provided. Parameters that do not meet out quality standards (see methods) are marked in bold. ANOVA results in Table 31

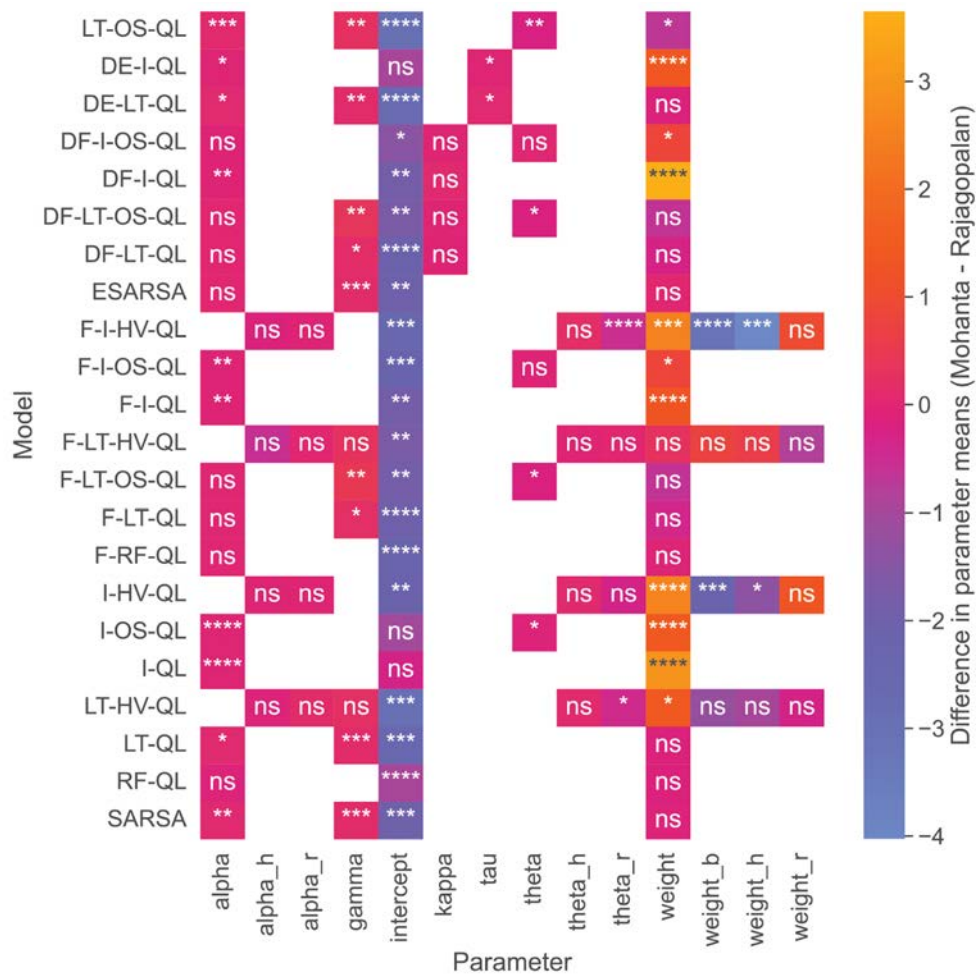


Figure 37. Difference between the parameter estimates from the Mohanta (2022) and Rajagopalan (2022) "Fixed Block" datasets.

Heatmap of the difference in the means of parameter estimates from the two datasets and the difference is tested using a simple z test (stars for statistical significance).

Recurrent q-networks capture the behavior well and Feedforward q-networks show perseverance behavior

We replicate the neural network-based de-novo model synthesis approach on the Mohanta (2022) "Variable Block" dataset. We find that yet again that while small FFqN manages to capture the behavior of the flies, RqNs perform the best in explaining the behavior well, better than any other model (Figure 38. A). However, one difference is that the best model is the asymmetric RqN with 100 reservoir neurons (asymRqN(100)). The performance is not significantly different from the asymmetric model with 2 reservoir neurons (asymRqN(2)) (Figure 38. A).

On symmetrization, the best model is the one with 4 effective neurons (symRqN(2)). While all q-networks manage to track and predict the changing preference, the RqNs do it better than all previous models (Figure 38. B–E). In order to validate our past results of finding perseverance in FFqNs, we find reliable preservative attractors under the no-reward condition for both symmetric and asymmetric FFqNs (Figure 38. B–E) reproducing what we saw with the Rajagopalan (2022) "Fixed Block" dataset.

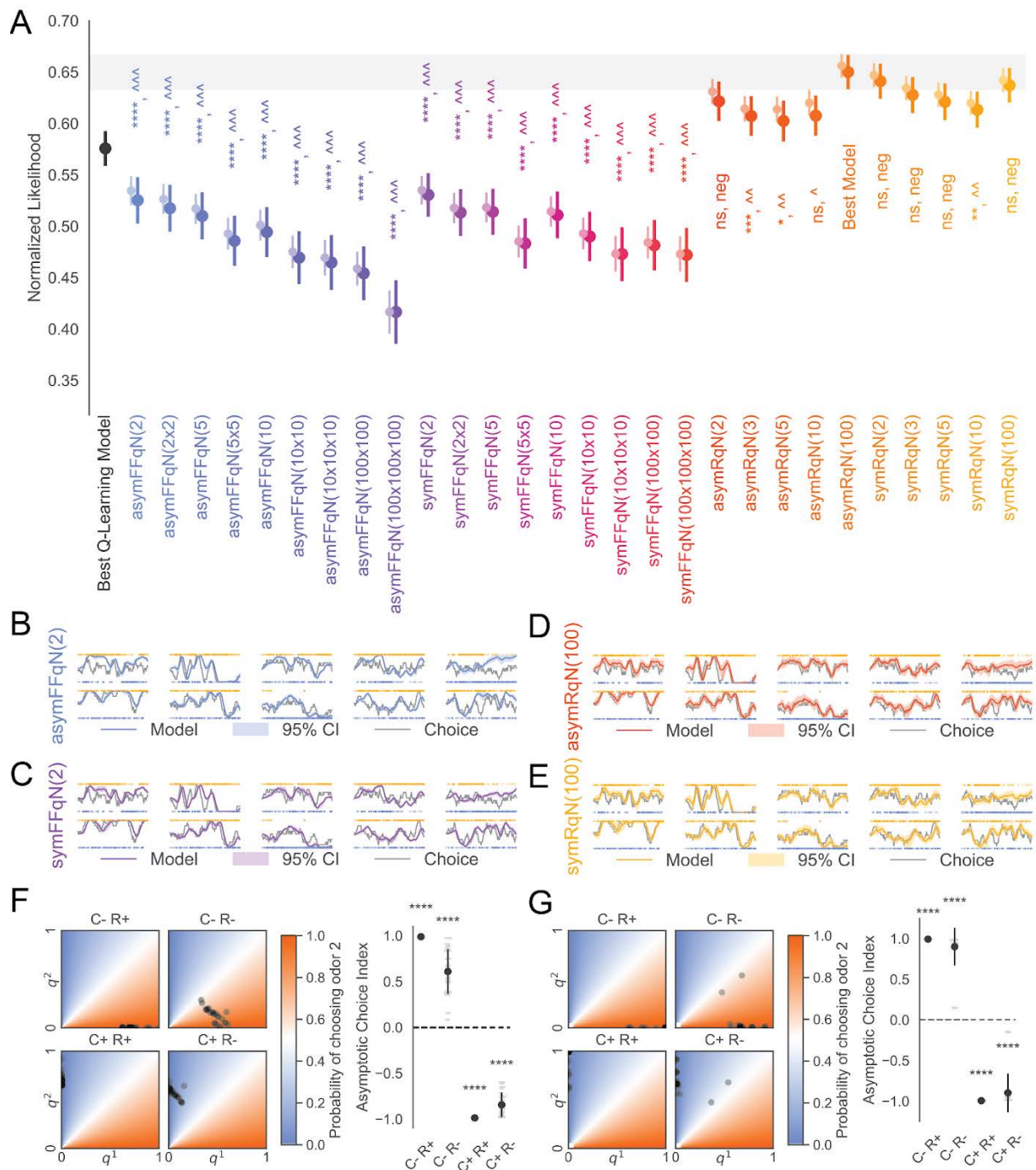


Figure 38. Results from fitting neural networks to the Mohanta (2022) "Variable Block" dataset also roughly reproduces the observations from Rajagopalan (2022) "Fixed Block" dataset.

(A) Comparison of the goodness of fit and predictive power estimated using Normalized Likelihood on training data and testing data, respectively, for different neural network architectures trained to estimate value from data and predict the choices. Light and dark error bars represent the mean and standard error of training

and test Normalized Likelihood, respectively. Test Normalized Likelihood of each of the models is compared to the best model (asymRqN(100)) using a bootstrap-corrected two-sided paired samples t-test (stars for statistical significance) and bootstrap-corrected paired cohen's d effect size (carets for effect size) (m=44 flies, n=25 ensembles for bootstrap correction; see methods). See Table 22 for p-values and effect sizes, including a comparison of training Normalized Likelihood using the same statistical measures.

(B–E) Smoothed predicted choice probabilities for ten random test flies with a 95% confidence interval estimated from 25 ensemble models for the best network architectures from each network class/variant overlaid on smoothed choice probabilities estimated from the data with a 10-trial window (see methods).

(F) Position of all the fixed point attractors across the trained and filtered ensemble of asymmetric FFqNs marked on the space of acceptance probabilities with black dots (left). Predicted preference of odors at the fixed point attractors of the different choice-reward conditions for all trained and filtered asymmetric FFqNs of the ensemble compared from zero using a two-sided bootstrap test (stars for significance; $p=0.000$ for all values).

(G) Position of all the fixed point attractors across the trained and filtered ensemble of symmetric FFqNs marked on the space of acceptance probabilities with black dots (left). Predicted preference of odors at the fixed point attractors of the different choice-reward conditions for all trained and filtered symmetric FFqNs of the ensemble compared from zero using a two-sided bootstrap test (stars for significance; $p=0.000$ for all values).

Discussion

Foraging by continually making choices, i.e., investing energetic or cognitive resources towards one of many alternative options available, is a behavior that is fundamental to animal life. In a complex dynamic environment, different strategies to collect rewards (or avoid punishments) can be used to improve the performance of an animal, possibly impacting its survival. In this work, we first use past observations from choice experiments in a dynamic reward environment (Rajagopalan et al., 2022) to understand how fruit fly behavior can be understood and explained from a Reinforcement Learning perspective. Notably, we explore the framework of Value Learning which can be directly mapped to the known circuitry of the fly.

Simple Q-Learning models have been used to understand the behavior of flies before (Rajagopalan et al., 2022; Seidenbecher et al., 2020) and can predict behavior decently. However, the models perform much better when we add additional cognitive considerations about forgetting and long-term discounting (Figure 14). Particularly of note is the inclusion of the idea of “habits” or “perseverance”, defined as a tendency to keep doing the same thing independent of the associated reward. Including “habits” through multiple phenomenological strategies improves how well we can explain the Rajagopalan (2022) “Fixed Block” dataset. But, with the small sample size, it is hard to claim strong predictive power. Distinguishing models becomes particularly hard since all “good” behavioral models perform very similarly (Figure 15. and Figure 17.).

Further, we observe that, strangely, while these models can show us the operant matching behavior observed in the experimental data, the strength of the matching is found to be significantly weaker (Figure 16.). While the zero bias in matching is a consequence of our modeling assuming that the learning is symmetric, the problem of “soft” predictions (close to random change probability i.e., 0.5, for choosing any odor) and undermatching can also be seen in the predicted probabilities of choosing odors (Figure 15.). Multiple possible reasons might explain this. Firstly, this may be a consequence of our model assumptions of symmetry. The models assume symmetry in the model parameters and initial values for both odors and across flies but are fitted on a dataset that is inherently asymmetrical and highly variable. As a result, the inferred strength of learning is an intermediate value between the strength of

independently learning associations with the two odors and across flies, thus resulting in intermediate “soft” predictions. This problem can be mitigated by allowing different initial conditions and parameter values for the two odors and across flies which can be implemented using bayesian hierarchical models (Albert & Hu, 2019, Chapter 10). We implemented hierarchical cognitive Q-learning in our FIYMazeRL framework, which performs better at explaining behavior than the models explored in this work (data not shown); however, it requires more analysis to understand how they do so.

Nonetheless, this will likely not solve the problem of “soft” predictions, as suggested by our results from de-novo value learning rule estimation using neural networks (Figure 19.). In this approach, asymmetric networks are allowed to have different behavior for the two odors, while symmetric networks are not. Nevertheless, we do not observe any apparent differences in how well they explain or predict behavior. We can see that both variants of neural networks are capable of “strong” predictions, but the FFqNs do not generalize well enough to make better predictions than the Q-Learning framework.

This observation brings us to the alternative hypothesis that a fly’s behavior results from a different temporal integration rule or non-linearity than the one proposed by the Q-learning framework. It is also possible that, unlike our model assumptions, the parameters (such as the learning rate, forgetting rate, omission sensitivity, etc.) may change dynamically over time or have multiple overlapping timescales i.e., the underlying dynamics of the learning might have changes that happen over a few trials and changes that happen over a large number of trials simultaneously. As a result, our model inference averages out the timescales giving us “soft” predictions. The RqN results provide evidence for this hypothesis.

RqNs seem to allow a separation of timescales (possibly through activity on orthogonal subspaces within the network) which is likely contributing to the better predictive power of the method (Figure 20.). This might also explain why FFqNs do not explain the data well despite giving “strong” predictions. FFqNs can only perform instantaneous value updates but are not constrained by the smooth updates of the Q-learning framework. Therefore, they might reflect the sharp non-linear changes in dynamics but are forced into a single timescale, limiting how well they perform while

still showing robust changes in preference. However, it is essential to note that the faster timescale components of the activity that modulate the choice preferences are not explained by linear regression over past experiences (Figure 21.). This implies that there is likely some “meta-computation” that the RNNs can perform on its hidden dynamics to modulate the behavior, a direction that has not yet been explored.

Taking a deeper look at the results from fitting neural networks, we observe that our performance reduces rather than improves with more complex models with more parameters. This result is especially evident for the FFqN models (Figure 19.). Most likely, this is because of the vanishing gradient problem that has plagued recursive models throughout their history (Pascanu et al., 2013). While the network we are learning is feedforward, each update on a single trial depends on the network’s output on the previous trial; therefore, the errors need to be propagated to the start of the first trial to learn the shared parameters simultaneously. Further, the vanishing gradient problem is exacerbated since there is a bottleneck with only two variables (q_1 and q_2) between trials through which errors must be propagated. Despite these issues, we can train networks capable of predicting behavior.

With our analysis of simple FFqNs and RNNs with a minimal number of neurons, we again see the presence of a tendency to “persevere” emerges. Using dynamical systems analysis on the behavior of symmetric and asymmetric variants of FFqNs and linear approximations on the behavior of the RqNs, we can discover a positive influence of past choice on the value of future iterations of the same choice, even when the choice is not reinforced. This converging result from the Q-learning approach suggests a role of habits in the fly’s behavior.

This brings us to our attempt to test these models through behavioral perturbations through “choice engineering”. We find the most biasing open-loop reward schedules for five representative cognitive Q-learning models. We see that the differences between the predicted most biasing open-loop rewards schedules are very minute but are possibly due to minor differences in the cognitive basis underlying the behavior of the models (Figure 26.). For example, the models with no forgetting (RF-QL/LT-QL) or no extinction (F-RF-QL/RF-QL) are likely to learn strong associations. If there is a strong pairing of rewards for one odor, these models can only slowly change their preference. As a result, the most biasing schedules show a

stronger 'primacy'-like block structure. Alternatively, suppose the behavior is more influenced by perseverance (DF-LT-OS-QL). In that case, lower reward probabilities can still drive strong preference, allowing the choice engineer to 'sneak in' more reward on the distractor odor. We experimentally test the reward schedules for two models: F-RF-QL (which does not have a reward prediction error) and DF-LT-OS-QL, which combines all the different cognitive features that explain the behavior well. Surprisingly, in the experiments, we observed a stronger bias in the behavior than was predicted by the models. This effect is possibly due to the same reasons as the "soft" predictions made by the models, as the same effect would also result in more intermediate values of bias than the actual flies. However, due to the high biological variability in the experiments, we cannot see more than a small effect of increased bias toward the schedules predicted by the DF-LT-OS-QL. This result was not surprising to us as the models have very similar average behavior (Figure 15. and Figure 17.). It is also possible, that the choice engineered experiments are "out-of-sample" tasks, i.e. the experiments represent a space of tasks that were never included in the training set. Therefore, to test any model via behavioral perturbations, we either needed an alternate method for designing or providing perturbations or a way to improve the throughput of the experimental assay in order to more broadly sample the space of tasks.

A possible line of experiments would be to try and exploit the differences in the dynamics of value for each individual (Figure 17.) by using a closed-loop adversary such as the one used in Dezfouli et al., 2020. Such a method might provide greater contrast between models. Further, one could also optimize the closed-loop adversary to drive the maximal separation between models instead of maximizing bias. This method would allow for pairwise comparison between any two models by looking at the final behavior of a fly and comparing it to the behavior observed in a fly.

Nevertheless, we recognized a need for a high throughput Y-maze assay to account for the relatively small sample size from single fly experiments compared to population assays such as the classic T-maze (Tempel et al., 1983) or Circular Arena assay (Aso et al., 2014). Therefore we developed a high-throughput 16-fly Y-Maze assay that allowed us to expand our experimental throughput massively (Figure 5.). We developed the assay to be as flexible as possible and capable of running virtually any form of closed-loop or open-loop olfactory forced-choice assay or pavlovian

conditioning assay that requires up to 2 odors in parallel. The high-throughput rig allows us to perform massive genetic screens with minimal effort and test various hypotheses about fruit fly behavior, including but not limited to risky choice experiments (Cavagnaro et al., 2013; Pachur et al., 2013) and large scale world-state inference experiments discovered through task enumeration (Ma & Hermundstad, 2022), which would allow us to test the cognitive limits of the behavior with minimal time investment.

Further, the accompanying data analysis pipeline also gives us any kinematic variables that can be extracted from the behavior to help further guide our analysis. Using this information, we can explain the unexpected dynamics of choice times and the underlying kinematic factors that influence the choice (Figure 29. and Figure 30.). The most exciting result from this analysis was that we observed more extended residence in the odor paired with reward, higher movement in the same odor, and more robust rejection of the other odor. These observations suggest a greater motivational drive in the odor with greater value and more frequent rejection in the arm with less value, which is what we would expect based on our model of value learning (Figure 3.).

We also see a strong preference to enter the arm with the odor, which is not entirely consistent with our behavioral policy, which is dependent on the rejection of an odor after entering it as the primary driver for preference. However, we do not see it as a significant issue. It is possible that due to the miniaturization of the original single-fly Y-maze used in Rajagopalan et al., 2022, our odor boundaries are not very sharp. Consequently, the fly can experience an odor before entering the marked boundaries of an arm. Upto an approximation, such behavior can also be effectively seen a rapid rejection of the other odor. Therefore, one way to improve our modeling efforts would be to use all of the information about the trajectories available to create a cognitive model that predicts the actual movement of a fly and use it to build an accurate kinematic model of the decision-making process underlying the behavior of a fly. However, alternatively, the fly could be experiencing both odors at the boundary either as a well-mixed combination or a intermittent sequences of two odors at different frequencies, which is a level of behavioral resolution we are current not capable of capturing or including in our models.

We discover some interesting behaviors through our process of optimizing the combination of odors for the high throughput maze. We see strong biased preferences for OCT vs. MCH choice, which are typically considered a comparable pair of odors in the field of fruit fly olfaction (Figure 28.). We believe this is because of the reduced choice cost in the miniaturized Y-maze, as the fly needs to move for a shorter distance to make a choice. As a result, the rig can accentuate even slight differences in innate preferences or learning.

For PA vs. EL choices, we consistently see learning of EL at low probabilities of reward even when PA is paired with a reward (Figure 31.). The simplest explanation would be that some residual odor leads to cross-contamination of learning effects. However, this effect seems to become much rarer at higher reward probabilities suggesting that this is likely not the case. Further, since experiments were interleaved on the same day, it is unlikely to be due to differences between fly populations. An alternate explanation is that since the time between successive odor exposures is relatively short, there may be some trace conditioning, especially for EL, which is known to have a much broader Kenyon cell activation (Honegger et al., 2011). With more frequent reward pairing, it is possible that the activation of the PA-related populations becomes more predictive of the reward and therefore triggers the appropriate learning behavior. A mechanism for such plasticity is currently unknown. Further, we see a significant primacy effect for OCT vs. MCH and PA vs. EL. choices even at low starvation levels where the fly is barely motivated to seek reward (Figure 28. and Figure 31.).

We finally establish MHO vs. HAL as the best comparable odors without a significant odor-sensitivity or order-sensitivity (primacy) in behavior (Figure 32.). The only effect that significantly influences learning is related to starvation level. We then use these odors to create a large dataset to sample the space of dynamic choice behavior and create a new dataset of 2AFC experiments in flies. This new dataset also shows operant matching behavior observed in Rajagopalan et al., 2022, therefore validating that flies reproducibly show matching behavior across different experimental setups. We establish that a matching model with a short-term integration of 5 trials and a finite maximum choice probability can explain the behavior well and performs as well as the best linear regression model (Figure 33. and Figure 34.). The linear model's behavior is primarily driven by the reward-choice association (interaction term). We

see some spurious positive influence of past choices on the behavior, which seems to suggest a weak influence of habits contrary to the stronger history dependence observed by Rajagopalan et al., 2022. In both these results, the time window with which integration happens seems to be sensitive to 5-trial histories (for both matching and choice integration), which is peculiar as it is the resolution of the block switching in our experiments (see methods). It is possible that the flies are somehow sensitive to the block switch at this resolution.

Further, replicating the model fitting on the Mohanta (2022) "Variable Block" dataset reveals that the significant differences between the models are with the policy parameters, which transform the value to choice. This observation is not surprising because the changes in the size of the experimental setup and odor pair will likely drive changes in these parameters as the energetic costs and innate motivational drives will be different. Further, perseverance seems to improve the model fit and predictions, but the results are not as trivial as expected from the previous analysis. Some models without perseverance also seem to perform as well as ones without it (Figure 36.). However, our analysis of the trained FFqNs reveals robust perseverance (Figure 37.).

Consequently, we can only claim that there is a weak habit-forming tendency under the new experimental conditions. This behavioral change may be due to the differences in the experimental rig and task structure. Firstly, we use odors optimized for maximal symmetry and high degrees of exploration and, as a result, may have a low persistence of preference between blocks and, therefore, within blocks. Further, it is possible that at the concentration of the odors used, the odor representation is only strong enough to drive direct reward learning and not habitual behavior. Alternatively, it is possible that since the new experiments are a lot more dynamic (average block size = ~ 30 trials in Mohanta (2022) "Variable Block" dataset vs. 80 in Rajagopalan (2022) "Fixed Block" dataset), and therefore the effect of habit learning appears to be stronger. Further data analysis might reveal more evident effects of habit learning on a subset of the data with more extended block sizes. Lastly, another reason for not observing habits as strongly would be that habits are sensitive to the evolutionary significance of the odors itself, however this is highly unlikely as there is very little evidence that the identity of the odor is reflected in the activation of the fruit fly MB. That means while one can distinguish between two odors A and B

given the activation of the KCs, it is unlikely that one can identify which was odor A or which was odor B. As a result all computations implemented in the MB and direct feedback are likely independent of the identity. However, it is entirely possible that computations mediated by indirect feedback can receive input from the lateral horn (LH) and as a result might be sensitive to odor identity.

Habits in behavior seem counterintuitive because they would force an animal to keep doing the same thing repeatedly; nevertheless, it must be noted that it is typically not the case. One hypothesis to explain this is that on account of the complex nature of the animal's biology, behavior is rarely fully deterministic but rather stochastic with weak or strong biases towards specific outcomes. Therefore, as long as no pathological influence overrides the inherent randomness of behavior, animals will likely not entirely fall into purely habit-driven attractors. In recent days, multiple lines of evidence seem to suggest the existence of similar habitual behavior during foraging experiments in mice and rats (Beron et al., 2022; Greenstreet et al., 2022; Miller et al., 2019). Therefore, habits appear to be a convergent strategy across the animal kingdom. However, it is not clear how or why habits may have emerged. Many possible theories about habit formation suggest that habits reduce cognitive load on the animal (Beron et al., 2022; Wood et al., 2014) or are more stable during learning (Kim et al., 2015). It is also possible that habits are only optimal when considering the risks associated with exploration (e.g., in the presence of a threat, exploration may have a much greater cost by exposing the animal to an uncertain outcome).

In any case, there is increasing interest in understanding the neural underpinnings of habitual behavior, and the dopaminergic system has been implicated in this system across mammalian systems from mice and rats (Bogacz, 2020; Greenstreet et al., 2022) to humans where deficits in habit formation have a pathological significance in Parkinson's disorder (Bannard et al., 2019; F. Hernandez et al., 2015). Due to the highly conserved nature of this behavior, the fruit fly provides us with an opportunity to explore the mechanistic underpinnings of habitual behavior. Under our value learning framework of the mushroom body, habits are likely implemented through (direct or indirect) feedback connections from MBONs to the DANs. Recently, it was shown that MBONs from one compartment could drive learning across other compartments via interneurons enabling second-order conditioning (Yamada et al.,

2022). Furthermore, there is evidence that some of these interneurons directly influence upwind walking behavior (unpublished data; Mohanta et al., 2019) and that activity in dopaminergic neurons is sensitive to context-dependent motion signals (Zolin et al., 2021). All of these together suggest the existence of a possible pathway by which odor-gated signals can drive plasticity in the mushroom body and therefore strengthen upwind walking behavior through repeated action even in the absence of reward. Alternatively, feedback from multisensory odor and wind-sensitive circuits further downstream of the mushroom body in the fan-shaped body (Matheson et al., 2022) may also provide an action learning signal to the dopaminergic neurons.

On a different note, as mentioned earlier, we do not fully understand why the RNNs perform so much better. The task structure may confound the argument for the separation of timescales since the first PC of the hidden dynamics for RqNs trained on Rajagopalan (2022) "Fixed Block" dataset seem to have an autocorrelation dropoff of ~ 80 trials which is equal to the size of the block. However, in the analysis of RqNs trained on Mohanta (2022), we do not see such a clear separation (data not shown) and therefore needs more analysis. Further dissection of the non-linearity and temporal dynamics underlying these neural network-based models may give further insight into their function.

Multiple possible directions might be helpful to pursue. Firstly, one can attempt to condense the underlying non-linearities of the network using methods for equation discovery typically utilized in physics. One such strategy, known as Sparse Identification Nonlinear Dynamics (SINDy), can be directly applied to the behavior of the neural networks to derive phenomenological equations that describe the non-linearity of value update using a form of lasso regression analysis (Brunton et al., 2016). We attempted to apply this to the trained FFqN vector fields; however, the resulting equations that explained the dynamics were not readily interpretable (data not shown). Nevertheless, further bootstrapped reliability analysis on the discovered terms can potentially reveal valuable insight into the dynamics. Similarly, Symbolic Regression (SR) techniques such as AI Feynman (Udrescu & Tegmark, 2020) might also reveal interesting dynamical rules to understand the behavior of the networks. A potential probabilistic formulation of equation discovery methods can be directly applied to behavioral data.

Alternatively, the improved network assumptions can be directly incorporated into the neural network architectures. Diverse RNN architectures have been explored in the past to be able to sustain dynamics across multiple timescales (Alpay et al., 2016), of which clockwork RNNs (CW-RNNs) (Koutník et al., 2014) provide a simple interpretable way to incorporate multiple timescales of dynamics into independent modules to try and understand the dynamics. Alternatively, more robust architectures such as LSTMs (already implemented in FIYMazeRL) can be explored, or variational implementations of RNNs can be used to recover probabilistic generative rules to understand the behavior. These modifications, combined with other strategies to improve the training of the RNNs (Hafner, 2017), might reveal even more powerful learning rules with much better explanatory and predictive power.

Therefore, our cognitive modeling, experimental design, and observations open up multiple new directions to explore the computational limits of the fruit fly's brain, giving us a unique opportunity to find the neural mechanistic underpinnings of cognitive behavior through future experiments.

Statistical Tables

Abbreviations:

NL : Normalized Likelihood

SE : Standard Error

WAIC : Watanabe Akaike Information Criteria

pWAIC: Bayesian effective number of parameters

CI : Confidence Interval

d: Cohen's d

r: Matched-pair Rank Biserial Correlation

df: degree of freedom

Model	Rank	WAIC	pWAIC	SE	WAIC p-value	WAIC Effect Size	Test NL Mean	Test NL SE	Test NL p-value	Test NL Effect Size
RF-QL	23	4737.381	1.939	30.131	0.0000	7.67	0.554	0.037	0.1796	1.17
I-QL	22	4646.981	2.522	34.625	0.0000	6.09	0.566	0.037	0.2088	1.06
DE-I-QL	21	4643.346	4.065	34.637	0.0000	6.04	0.569	0.038	0.2208	1.02
SARSA	20	4582.409	3.442	35.705	0.0000	4.81	0.561	0.031	0.2981	0.80
ESARSA	19	4562.066	3.705	36.087	0.0001	3.93	0.567	0.035	0.2660	0.88
I-OS-QL	18	4558.224	2.898	36.634	0.0012	3.24	0.569	0.041	0.3036	0.79
DE-LT-QL	17	4553.607	4.356	36.162	0.0004	3.54	0.571	0.038	0.2762	0.86
LT-QL	16	4551.505	3.678	36.239	0.0006	3.44	0.571	0.038	0.3050	0.79
F-I-QL	15	4535.540	2.688	39.254	0.0001	3.99	0.578	0.039	0.3760	0.65
LT-OS-QL	14	4534.488	4.142	37.086	0.0126	2.50	0.570	0.040	0.3048	0.79
F-RF-QL	13	4526.067	3.581	40.086	0.0001	3.87	0.574	0.035	0.4122	0.59
DF-I-QL	12	4520.778	3.456	39.173	0.0004	3.57	0.582	0.042	0.8375	-0.23
F-LT-QL	11	4517.939	4.193	39.880	0.0001	3.86	0.581	0.040	0.7242	-0.16
F-LT-OS-QL	10	4509.503	4.154	40.098	0.0009	3.31	0.579	0.040	0.4393	0.54
DF-LT-QL	9	4508.366	5.522	40.081	0.0003	3.64	0.587	0.044	-	-
F-I-OS-QL	8	4507.970	3.532	40.063	0.0018	3.13	0.577	0.038	0.3855	0.64
F-I-HV-QL	7	4504.069	12.365	39.843	0.0412	2.04	0.579	0.038	0.5369	0.40
F-LT-HV-QL	6	4501.679	9.827	40.215	0.0901	1.69	0.579	0.039	0.5564	0.18
DF-I-HV-QL	5	4498.553	9.153	40.036	0.0686	1.82	0.579	0.038	0.5593	0.37
I-HV-QL	4	4493.564	5.931	40.239	0.1496	1.44	0.580	0.039	0.5716	0.35
LT-HV-QL	3	4493.279	11.564	39.848	0.1021	1.63	0.583	0.042	0.8624	-0.26
DF-I-OS-QL	2	4487.811	4.274	40.404	0.3247	0.98	0.582	0.041	0.8056	-0.06
DF-LT-HV-QL	1	4485.671	9.771	40.652	0.8707	0.16	0.585	0.042	0.8791	-0.88
DF-LT-OS-QL	0	4484.173	5.219	40.677	-	-	0.586	0.044	0.9281	-0.63

Table 13. Q-Learning model fit statistics on the Rajagopalan (2022) "Fixed Block" dataset.

Cognitive Factor	weight	intercept	alpha	gamma	tau	kappa	weight_r	weight_h	weight_b	alpha_r	kappa_r	alpha_h	theta_r	theta_h	theta
RPE	0.27 (p < 0.0001)	0.15 (p < 0.0001)	0.57 (p < 0.0001)	-	-	-	-	-	-	-	-	-	-	-	-
Forgetting	0.03 (p < 0.0001)	0.39 (p < 0.0001)	0.03 (p < 0.0001)	0.35 (p < 0.0001)	-	-	0.05 (p < 0.0001)	0.46 (p < 0.0001)	0.00 (p < 0.0001)	0.03 (p < 0.0001)	-	0.13 (p < 0.0001)	0.25 (p < 0.0001)	0.12 (p < 0.0001)	0.02 (p < 0.0001)
Learning - Forgetting	0.00 (p < 0.0001)	0.01 (p < 0.0001)	0.03 (p < 0.0001)	0.11 (p < 0.0001)	-	-	0.01 (p < 0.0001)	0.04 (p < 0.0001)	0.00 (p < 0.0001)	0.10 (p < 0.0001)	-	0.00 (p < 0.0001)	0.02 (p < 0.0001)	0.01 (p < 0.0001)	0.37 (p < 0.0001)
Discounting	0.32 (p < 0.0001)	0.07 (p < 0.0001)	0.06 (p < 0.0001)	-	0.92 (p < 0.0001)	0.27 (p < 0.0001)	0.03 (p < 0.0001)	0.05 (p < 0.0001)	0.05 (p < 0.0001)	0.21 (p < 0.0001)	0.35 (p < 0.0001)	0.26 (p < 0.0001)	0.04 (p < 0.0001)	0.08 (p < 0.0001)	0.29 (p < 0.0001)
Omission Sensitivity	0.01 (p < 0.0001)	0.01 (p < 0.0001)	0.01 (p < 0.0001)	0.14 (p < 0.0001)	-	0.39 (p < 0.0001)	-	-	-	-	-	-	-	-	-
On-Policy	0.02 (p < 0.0001)	0.02 (p < 0.0001)	0.00 (p=0.0007)	0.01 (p < 0.0001)	-	-	-	-	-	-	-	-	-	-	-
Learning - Extinction Independence	0.03 (p < 0.0001)	0.01 (p < 0.0001)	0.00 (p < 0.0001)	0.00 (p < 0.0001)	-	-	-	-	-	-	-	-	-	-	-
APE	0.08 (p < 0.0001)	0.00 (p < 0.0001)	-	0.05 (p < 0.0001)	-	-	-	-	-	-	-	-	-	-	-
residual	0.24	0.34	0.31	0.34	0.08	0.34	0.92	0.45	0.95	0.66	0.65	0.61	0.7	0.79	0.32

Table 14. ANOVA summary (Cognitive variables vs. model parameters) for the Rajagopalan (2022) "Fixed Block" dataset.

Eta-squared Effect Size and p-value reported.

Model	Matching Strength p-value	Matching Strength Effect Size	Bias p-value	Bias Effect Size
RF-QL	3.8E-07	-1.00	2.8E-05	-0.93
F-RF-QL	2.0E-04	-0.88	3.6E-06	-0.97
I-QL	3.6E-06	-0.98	3.1E-06	-0.97
LT-QL	2.0E-04	-0.87	4.8E-06	-0.97
F-I-QL	3.8E-03	-0.77	2.2E-06	-0.98
F-LT-QL	1.8E-04	-0.89	3.6E-06	-0.97
DF-I-QL	1.8E-03	-0.80	4.2E-06	-0.97
DF-LT-QL	6.0E-04	-0.83	1.9E-06	-0.98
DE-I-QL	6.5E-06	-0.96	2.6E-06	-0.98
DE-LT-QL	1.7E-01	0.53	1.2E-05	-0.95
I-OS-QL	2.6E-04	-0.86	6.5E-06	-0.96
LT-OS-QL	8.2E-03	-0.72	1.6E-06	-0.98
F-I-OS-QL	9.5E-04	-0.83	5.6E-06	-0.96
F-LT-OS-QL	4.2E-04	-0.85	3.1E-06	-0.97
DF-I-OS-QL	2.0E-02	-0.69	4.2E-06	-0.97
DF-LT-OS-QL	1.5E-03	-0.80	3.6E-06	-0.97
SARSA	2.2E-06	-0.98	1.0E-06	-0.99
ESARSA	4.8E-05	-0.91	1.0E-06	-0.99
I-HV-QL	6.3E-05	-0.91	1.9E-06	-0.98
F-I-HV-QL	1.2E-04	-0.89	3.6E-06	-0.97
DF-I-HV-QL	6.7E-04	-0.84	3.1E-06	-0.97
LT-HV-QL	2.9E-04	-0.87	1.6E-06	-0.98
F-LT-HV-QL	4.2E-05	-0.92	8.8E-06	-0.95
DF-LT-HV-QL	1.4E-04	-0.89	1.8E-05	-0.93

Table 15. Matching law statistics for the Rajagopalan (2022) "Fixed Block" dataset models.

Model	Mean	95% CI	p-value	d
RF-QL	0.0005	(0.0003–0.0010)	0.141238	0.577484
I-QL	0.0002	(0.0000–0.0006)	4.17E-05	0.912213
DE-I-QL	0.0001	(0.0000–0.0005)	1.81E-05	0.937707
SARSA	0.0001	(0.0000–0.0008)	2.40E-05	0.923446
ESARSA	0.0002	(0.0000–0.0008)	6.27E-05	0.891955
I-OS-QL	0.0001	(0.0000–0.0005)	1.36E-05	0.939428
DE-LT-QL	0.0001	(0.0000–0.0011)	2.76E-05	0.926866
LT-QL	0.0002	(0.0000–0.0008)	5.49E-05	0.894374
F-I-QL	0.0008	(0.0000–0.0026)	0.558337	0.385367
LT-OS-QL	0.0002	(0.0000–0.0006)	3.64E-05	0.916433
F-RF-QL	0.0024	(0.0002–0.0079)	0.715975	-0.32058
DF-I-QL	0.0007	(0.0000–0.0029)	0.2114	0.486076
F-LT-QL	0.0015	(0.0001–0.0041)	0.962143	-0.08805
F-LT-OS-QL	0.0012	(0.0001–0.0035)	0.962148	0.118275
DF-LT-QL	0.0018	(0.0002–0.0044)	0.763745	-0.30735
F-I-OS-QL	0.001	(0.0001–0.0029)	0.911826	0.215011
F-I-HV-QL	0.0011	(0.0001–0.0036)	0.936956	0.156868
F-LT-HV-QL	0.0016	(0.0001–0.0050)	0.962148	-0.05367
DF-I-HV-QL	0.001	(0.0001–0.0033)	0.861856	0.2284
I-HV-QL	0.0005	(0.0000–0.0016)	0.069126	0.595963
LT-HV-QL	0.0008	(0.0001–0.0028)	0.669291	0.344591
DF-I-OS-QL	0.0011	(0.0001–0.0030)	0.962148	0.083189
DF-LT-HV-QL	0.002	(0.0002–0.0059)	0.812432	-0.29785
DF-LT-OS-QL	0.0013	(0.0001–0.0033)	-	-

Table 16. Local value variance statistics for the Rajagopalan (2022) "Fixed Block" dataset

Model	Test NL	Test SE	Test p-value	Test d	Training NL	Training SE	Training p-value	Training d
asymFFqN(2)	0.5057	0.0558	0.0475	2.5550	0.4628	0.0115	0.0000	4.2982
asymFFqN(2x2)	0.5145	0.0535	0.0483	2.5332	0.4720	0.0110	0.0000	3.9339
asymFFqN(5)	0.4720	0.0692	0.0448	2.6366	0.4233	0.0156	0.0000	4.5587
asymFFqN(5x5)	0.4288	0.0689	0.0340	3.0529	0.3821	0.0196	0.0000	4.4857
asymFFqN(10)	0.4531	0.0628	0.0344	3.0344	0.4170	0.0152	0.0000	4.6629
asymFFqN(10x10)	0.4501	0.0646	0.0345	3.0490	0.4118	0.0153	0.0000	4.4897
asymFFqN(10x10x10)	0.4192	0.0701	0.0328	3.1105	0.3686	0.0182	0.0000	4.6684
asymFFqN(100x100)	0.3927	0.0708	0.0302	3.2496	0.3447	0.0198	0.0000	3.9427
asymFFqN(100x100x100)	0.3347	0.0673	0.0240	3.6697	0.2889	0.0203	0.0000	4.2239
symFFqN(2)	0.5091	0.0616	0.0326	3.1193	0.4623	0.0106	0.0000	4.6030
symFFqN(2x2)	0.5043	0.0609	0.0486	2.5311	0.4627	0.0115	0.0000	4.0740
symFFqN(5)	0.4734	0.0692	0.0424	2.7298	0.4257	0.0160	0.0000	4.4325
symFFqN(5x5)	0.4383	0.0703	0.0313	3.1861	0.3921	0.0155	0.0000	4.3800
symFFqN(10)	0.4657	0.0729	0.0398	2.8066	0.4214	0.0160	0.0000	4.3241
symFFqN(10x10)	0.4424	0.0715	0.0283	3.3622	0.3961	0.0185	0.0000	3.7497
symFFqN(10x10x10)	0.4618	0.0702	0.0354	2.9964	0.4127	0.0153	0.0000	4.2232
symFFqN(100x100)	0.4094	0.0744	0.0467	2.5807	0.3644	0.0189	0.0000	4.4033
symFFqN(100x100x100)	0.4141	0.0768	0.0377	2.8944	0.3647	0.0203	0.0000	3.8594
asymRqN(2)	0.7251	0.0405	0.8673	0.1094	0.6820	0.0100	0.5301	0.1514
asymRqN(3)	0.7290	0.0411	-	-	0.6855	0.0103	-	-
asymRqN(5)	0.7198	0.0451	0.8204	0.1505	0.6791	0.0094	0.4599	0.1782
asymRqN(10)	0.7178	0.0436	0.8597	0.1158	0.6805	0.0108	0.6648	0.1044
asymRqN(100)	0.7131	0.0408	0.7155	0.2456	0.6842	0.0120	0.2083	0.3083
asymRqN(200)	0.6694	0.0570	0.8547	0.1199	0.6615	0.0180	0.6032	0.1259
symRqN(2)	0.7267	0.0469	0.8200	0.1496	0.6846	0.0097	0.6464	0.1101
symRqN(3)	0.7245	0.0453	0.8775	0.1023	0.6806	0.0096	0.3561	0.2237
symRqN(5)	0.7081	0.0487	0.6832	0.2729	0.6685	0.0117	0.6841	0.0976
symRqN(10)	0.7178	0.0470	0.8260	0.1443	0.6770	0.0109	0.3525	0.2255
symRqN(100)	0.6964	0.0659	0.7101	0.2482	0.6519	0.0239	0.4438	0.1921
symRqN(200)	0.7162	0.0403	0.9411	0.0482	0.6790	0.0126	0.7457	0.0787

Table 17. Statistics for the comparison of neural networks trained on the Rajagopalan (2022) "Fixed Block" dataset .

Experiment	Kinematic Variable	EN vs. LN	LN vs. ET	ET vs. LT	LT vs. ER	ER vs. LR
OCT->MCH	Trial Length	0.5016	0.0009	0.0012	0.0031	0.0012
MCH->OCT	Trial Length	0.0134	0.0001	0.0001	0.0001	0.0040
OCT->MCH	Time in Air	0.0245	0.0023	0.0001	0.1040	0.0001
MCH->OCT	Time in Air	0.0017	0.0001	0.0001	0.0001	0.0009
OCT->MCH	Average Speed	0.6257	0.0001	0.0001	0.8077	0.0001
MCH->OCT	Average Speed	0.0580	0.0001	0.0001	0.0052	0.0017
OCT->MCH	$\Delta T(\text{MCH-OCT})$	0.5416	0.0004	0.0031	0.2958	0.0001
MCH->OCT	$\Delta T(\text{MCH-OCT})$	0.9032	0.0001	0.0017	0.0353	0.0002
OCT->MCH	Preference(MCH-OCT)	0.3910	0.0004	0.0134	0.0023	0.0001
MCH->OCT	Preference(MCH-OCT)	0.8077	0.0001	0.0134	0.5416	0.0012
OCT->MCH	$\Delta \text{Speed}(\text{MCH-OCT})$	0.8552	0.0002	0.8552	0.0067	0.0001
MCH->OCT	$\Delta \text{Speed}(\text{MCH-OCT})$	0.0676	0.0012	0.2958	0.5016	0.0040
OCT->MCH	$\Delta \# \text{Reject}(\text{MCH-OCT})$	0.3910	0.8552	0.9165	0.0245	0.0002
MCH->OCT	$\Delta \# \text{Reject}(\text{MCH-OCT})$	0.3046	0.6999	0.9164	0.0121	0.5830

Table 18. Statistics for the Fly Kinematics.

Experiment	Naïve vs. Training	Training vs. Reversal	Naïve vs. Reversal
OCT->MCH Training	0.0001	0.0001	0.0040
MCH->OCT Training	0.0001	0.0015	0.0031
PA->EL P(R)= 0.125	0.0244	0.0108	0.0069
EL->PA P(R)= 0.125	0.1232	0.0059	0.4316
PA->EL P(R)= 0.25	0.9528	0.0020	0.0059
EL->PA P(R)= 0.25	0.0020	0.0438	0.0208
PA->EL P(R)= 0.5	0.0977	0.0039	0.0391
EL->PA P(R)= 0.5	0.0020	0.0840	0.0645
PA->EL P(R)= 1	0.0020	0.0039	0.5566
EL->PA P(R)= 1	0.0137	0.0209	0.1230
MHO->HAL 4-13 hrs	0.0030	0.0001	0.3575
HAL->MHO 4-13 hrs	0.1465	0.0002	0.0134
MHO->HAL 28-37 hrs	0.0039	0.0039	0.5703
HAL->MHO 28-37 hrs	0.0020	0.0020	0.6953
MHO->HAL 51-64 hrs	0.0078	0.0039	0.5281
HAL->MHO 51-64 hrs	0.0020	0.0371	0.0488

Table 19. Statistics for the Choice Index.

Comparison	Within Odor 1	Within Odor 2	Within training odor	Within reversal odor
OCT vs. MCH	0.0016	0.0326	0.9817	0.1478
PA vs. EL R=1.0	0.9097	0.7337	0.2896	0.4274
PA vs. EL R=0.5	0.8382	0.0373	0.0160	0.5401
PA vs. EL R=0.25	0.1508	0.0451	0.0013	0.4727
PA vs. EL R=0.125	0.0028	0.8601	0.0265	0.1051
MHO vs. HAL 4-13 hrs	0.8651	0.3198	0.1262	0.7159
MHO vs. HAL 28-37 hrs	0.4377	0.7133	0.6534	0.7132
MHO vs. HAL 51-64 hrs	0.1113	0.3074	0.5956	0.1530

Table 20. Statistics for the Learning Index

	df	sum_sq	mean_sq	F	PR(>F)	Significance
C(Order)	1	0.741423	0.741423	0.607563	0.43924	ns
C(Odor)	1	1.690672	1.690672	1.385429	0.244538	ns
C(Order):C(Odor)	1	21.39994	21.39994	17.53628	0.000109	***
Residual	52	63.45683	1.220324			

Table 21. ANOVA for effect on learning rate for OCT vs. MCH choices

	df	sum_sq	mean_sq	F	PR(>F)	Significance
C(Order)	1	12.14906	12.14906	9.256622	0.002789	***
C(Reward Probability)	3	30.97385	10.32462	7.866542	6.75E-05	****
C(Odor)	1	25.65799	25.65799	19.54936	1.92E-05	***
C(Order):C(Reward Probability)	3	5.816209	1.938736	1.477164	0.223271	ns
C(Order):C(Odor)	1	0.000113	0.000113	8.63E-05	0.9926	ns
C(Reward Probability):C(Odor)	3	7.864809	2.621603	1.997454	0.117021	ns
C(Order):C(Reward Probability):C(Odor)	3	7.125734	2.375245	1.809749	0.148031	ns
Residual	144	188.996	1.312472			

Table 22. ANOVA for effect on learning rate for PA vs. EL choices

	df	sum_sq	mean_sq	F	PR(>F)	Significance
C(Order)	1	0.778867	0.778867	0.552306	0.458854	ns
C(Starvation)	2	33.94385	16.97192	12.03505	1.75E-05	****
C(Odor)	1	0.000505	0.000505	0.000358	0.984934	ns
C(Order):C(Starvation)	2	0.063699	0.03185	0.022585	0.977672	ns
C(Order):C(Odor)	1	2.101451	2.101451	1.490171	0.224624	ns
C(Starvation):C(Odor)	2	8.488852	4.244426	3.009786	0.053102	ns
C(Order):C(Starvation):C(Odor)	2	6.355494	3.177747	2.253388	0.109545	ns
Residual	118	166.4046	1.410209			

Table 23. ANOVA for effect on learning rate for MHO vs. HAL choice.

Model	Test NL	Test NL SE	Test p-value	Test r	Training NL	Training NL SE	Training p-value	Training r
matching(5)	0.5637	0.0100	0.5720	0.0072	-	-	-	-
matching(10)	0.5547	0.0112	0.5653	0.0075	0.0051	0.4847	0.0008	0.4137
matching(15)	0.5568	0.0121	0.5656	0.0081	0.0378	0.3596	0.0057	0.3396
matching(30)	0.5589	0.0141	0.5676	0.0095	0.2385	0.2040	0.0835	0.2124
matching(60)	0.5525	0.0142	0.5622	0.0098	0.0619	0.3232	0.0074	0.3289

Table 24. Statistics for the constrained matching law model fits for the Mohanta (2022) "Variable Block" dataset

Model	Test NL	Test NL SE	Test p-value	Test r	Training NL	Training NL SE	Training p-value	Training r
Best Matching Model	0.5637	0.0100	0.5674	0.0990	0.5720	0.0072	0.9436	0.0087
R+C+R.C (5)	0.5528	0.0130	0.0229	0.3938	0.5591	0.0090	0.8842	0.0179
R+C+R.C (10)	0.5623	0.0149	0.2628	0.1938	0.5720	0.0104	0.9502	0.0077
R+C+R.C (15)	0.5659	0.0155	0.9628	0.0081	0.5754	0.0107	0.9372	0.0097
R+C+R.C (30)	0.5672	0.0161	0.9535	0.0101	0.5788	0.0108	0.9502	0.0077
R+C+R.C (60)	0.5648	0.0165	0.7620	0.0524	0.5830	0.0108	0.9469	0.0082
R+R.C (5)	0.5455	0.0115	0.0411	0.3535	0.5534	0.0079	0.8191	0.0281
R+R.C (10)	0.5508	0.0128	0.1126	0.2746	0.5614	0.0087	0.9206	0.0122
R+R.C (15)	0.5564	0.0138	0.4915	0.1191	0.5670	0.0094	0.9372	0.0097
R+R.C (30)	0.5601	0.0150	0.4989	0.1171	0.5712	0.0100	0.9535	0.0072
R+R.C (60)	0.5597	0.0159	0.2432	0.2020	0.5762	0.0104	0.9502	0.0077
R+C (5)	0.5316	0.0075	0.0075	0.4626	0.5354	0.0056	0.3162	0.1230
R+C (10)	0.5416	0.0096	0.0138	0.4263	0.5465	0.0071	0.7553	0.0383
R+C (15)	0.5466	0.0106	0.0259	0.3858	0.5509	0.0077	0.8384	0.0250
R+C (30)	0.5540	0.0131	0.0184	0.4081	0.5604	0.0094	0.9107	0.0138
R+C (60)	0.5554	0.0144	0.0068	0.4686	0.5667	0.0101	0.9271	0.0112
C+R.C (5)	0.5512	0.0125	0.0282	0.3798	0.5570	0.0087	0.9008	0.0153
C+R.C (10)	0.5617	0.0147	0.2294	0.2081	0.5705	0.0102	0.9403	0.0092
C+R.C (15)	0.5658	0.0154	0.9446	0.0120	0.5736	0.0106	0.9337	0.0102
C+R.C (30)	0.5673	0.0159	-	-	0.5767	0.0108	-	-
C+R.C (60)	0.5653	0.0163	0.8159	0.0403	0.5805	0.0109	0.9436	0.0087

Table 25. Statistics for Linear Kernel Regression Models for the Mohanta (2022) "Variable Block" dataset

Lag	Mean(95% CI) [C]	p-value [C]	Mean(95% CI) [R.C]	p-value [R.C]
t-30	0.027 (-0.024–0.082)	0.027 (ns)	-0.040 (-0.129–0.041)	-0.040 (ns)
t-29	0.050 (-0.008–0.113)	0.050 (*)	-0.033 (-0.124–0.059)	-0.033 (ns)
t-28	0.033 (-0.016–0.080)	0.033 (ns)	-0.010 (-0.096–0.080)	-0.010 (ns)
t-27	0.074 (0.014–0.137)	0.074 (**)	0.004 (-0.085–0.088)	0.004 (ns)
t-26	-0.006 (-0.069–0.055)	-0.006 (ns)	0.062 (-0.020–0.155)	0.062 (ns)
t-25	0.071 (0.013–0.132)	0.071 (**)	-0.001 (-0.093–0.091)	-0.001 (ns)
t-24	0.014 (-0.052–0.076)	0.014 (ns)	0.015 (-0.079–0.109)	0.015 (ns)
t-23	0.043 (-0.012–0.104)	0.043 (ns)	-0.046 (-0.132–0.035)	-0.046 (ns)
t-22	0.038 (-0.025–0.103)	0.038 (ns)	0.003 (-0.096–0.109)	0.003 (ns)
t-21	-0.009 (-0.064–0.042)	-0.009 (ns)	0.096 (0.006–0.199)	0.096 (*)
t-20	0.033 (-0.027–0.086)	0.033 (ns)	0.001 (-0.091–0.088)	0.001 (ns)
t-19	0.013 (-0.049–0.074)	0.013 (ns)	0.012 (-0.067–0.098)	0.012 (ns)
t-18	0.016 (-0.032–0.065)	0.016 (ns)	0.024 (-0.067–0.113)	0.024 (ns)
t-17	0.047 (-0.005–0.097)	0.047 (*)	0.060 (-0.022–0.142)	0.060 (ns)
t-16	0.043 (-0.015–0.102)	0.043 (ns)	0.040 (-0.042–0.121)	0.040 (ns)
t-15	-0.010 (-0.063–0.038)	-0.010 (ns)	0.046 (-0.033–0.133)	0.046 (ns)
t-14	0.069 (0.020–0.121)	0.069 (**)	-0.019 (-0.108–0.067)	-0.019 (ns)
t-13	0.001 (-0.064–0.057)	0.001 (ns)	0.077 (-0.016–0.169)	0.077 (*)
t-12	0.071 (0.023–0.123)	0.071 (**)	0.040 (-0.045–0.124)	0.040 (ns)
t-11	0.016 (-0.034–0.066)	0.016 (ns)	0.052 (-0.020–0.129)	0.052 (ns)
t-10	0.047 (-0.003–0.096)	0.047 (*)	0.015 (-0.074–0.099)	0.015 (ns)
t-9	0.065 (0.010–0.118)	0.065 (**)	0.044 (-0.038–0.133)	0.044 (ns)
t-8	0.031 (-0.022–0.086)	0.031 (ns)	0.099 (0.007–0.188)	0.099 (*)
t-7	0.059 (0.008–0.110)	0.059 (**)	0.125 (0.051–0.203)	0.125 (**)
t-6	0.038 (-0.022–0.092)	0.038 (ns)	0.186 (0.102–0.277)	0.186 (****)
t-5	0.090 (0.039–0.140)	0.090 (**)	0.159 (0.079–0.240)	0.159 (**)
t-4	0.075 (0.030–0.124)	0.075 (**)	0.159 (0.073–0.247)	0.159 (****)
t-3	0.040 (-0.009–0.086)	0.040 (*)	0.238 (0.130–0.340)	0.238 (****)
t-2	0.004 (-0.052–0.068)	0.004 (ns)	0.314 (0.152–0.421)	0.314 (****)
t-1	0.065 (0.001–0.125)	0.065 (*)	0.308 (0.157–0.412)	0.308 (****)

Table 26. Parameters and Statistics for C + R.C (30) regression model for the Mohanta (2022) "Variable Block" dataset

Lag	Mean(95% CI) [C]	p-value [C]	Mean(95% CI) [R]	p-value [R]
t-30	0.015 (-0.030–0.053)	0.015 (ns)	-0.008 (-0.090–0.066)	-0.008 (ns)
t-29	0.027 (-0.013–0.068)	0.027 (ns)	-0.009 (-0.087–0.060)	-0.009 (ns)
t-28	0.018 (-0.026–0.057)	0.018 (ns)	-0.053 (-0.165–0.005)	-0.053 (ns)
t-27	0.054 (0.020–0.104)	0.054 (**)	0.002 (-0.081–0.084)	0.002 (ns)
t-26	0.015 (-0.031–0.053)	0.015 (ns)	0.061 (-0.002–0.163)	0.061 (ns)
t-25	0.056 (0.023–0.105)	0.056 (**)	0.028 (-0.034–0.118)	0.028 (ns)
t-24	0.020 (-0.019–0.056)	0.020 (ns)	-0.023 (-0.109–0.039)	-0.023 (ns)
t-23	0.018 (-0.022–0.051)	0.018 (ns)	-0.008 (-0.091–0.069)	-0.008 (ns)
t-22	0.029 (-0.016–0.071)	0.029 (ns)	-0.019 (-0.090–0.037)	-0.019 (ns)
t-21	0.024 (-0.020–0.060)	0.024 (ns)	0.028 (-0.035–0.122)	0.028 (ns)
t-20	0.035 (-0.003–0.077)	0.035 (*)	0.020 (-0.046–0.109)	0.020 (ns)
t-19	0.023 (-0.024–0.063)	0.023 (ns)	-0.024 (-0.113–0.049)	-0.024 (ns)
t-18	0.023 (-0.016–0.057)	0.023 (ns)	-0.014 (-0.105–0.064)	-0.014 (ns)
t-17	0.058 (0.032–0.095)	0.058 (****)	-0.006 (-0.077–0.066)	-0.006 (ns)
t-16	0.056 (0.024–0.104)	0.056 (**)	0.003 (-0.060–0.074)	0.003 (ns)
t-15	0.023 (-0.012–0.050)	0.023 (ns)	0.019 (-0.053–0.112)	0.019 (ns)
t-14	0.057 (0.027–0.099)	0.057 (**)	-0.021 (-0.103–0.044)	-0.021 (ns)
t-13	0.039 (-0.004–0.083)	0.039 (*)	-0.060 (-0.159–0.002)	-0.060 (*)
t-12	0.084 (0.040–0.136)	0.084 (****)	0.025 (-0.025–0.105)	0.025 (ns)
t-11	0.049 (0.014–0.085)	0.049 (**)	-0.016 (-0.095–0.050)	-0.016 (ns)
t-10	0.061 (0.031–0.101)	0.061 (****)	-0.000 (-0.076–0.072)	-0.000 (ns)
t-9	0.077 (0.042–0.122)	0.077 (****)	0.000 (-0.063–0.065)	0.000 (ns)
t-8	0.070 (0.038–0.118)	0.070 (**)	0.015 (-0.032–0.087)	0.015 (ns)
t-7	0.099 (0.050–0.150)	0.099 (****)	-0.002 (-0.071–0.069)	-0.002 (ns)
t-6	0.106 (0.051–0.165)	0.106 (****)	0.009 (-0.061–0.082)	0.009 (ns)
t-5	0.142 (0.061–0.209)	0.142 (****)	0.017 (-0.037–0.092)	0.017 (ns)
t-4	0.127 (0.056–0.192)	0.127 (****)	-0.009 (-0.092–0.068)	-0.009 (ns)
t-3	0.123 (0.057–0.181)	0.123 (****)	-0.014 (-0.094–0.045)	-0.014 (ns)
t-2	0.131 (0.057–0.199)	0.131 (****)	-0.020 (-0.098–0.051)	-0.020 (ns)
t-1	0.211 (0.074–0.305)	0.211 (****)	-0.017 (-0.090–0.043)	-0.017 (ns)

Table 27. Parameters and Statistics for R + C (30) regression model for the Mohanta (2022) "Variable Block" dataset

Lag	Mean(95% CI) [R]	p-value [R]	Mean(95% CI) [R.C]	p-value [R.C]
t-30	-0.010 (-0.084–0.070)	-0.010 (ns)	0.045 (-0.025–0.112)	0.045 (ns)
t-29	-0.018 (-0.097–0.054)	-0.018 (ns)	0.057 (-0.007–0.118)	0.057 (*)
t-28	-0.061 (-0.166–0.009)	-0.061 (*)	0.056 (-0.007–0.129)	0.056 (*)
t-27	-0.004 (-0.091–0.087)	-0.004 (ns)	0.090 (0.040–0.162)	0.090 (****)
t-26	0.057 (-0.003–0.146)	0.057 (*)	0.069 (0.009–0.142)	0.069 (*)
t-25	0.011 (-0.059–0.091)	0.011 (ns)	0.083 (0.032–0.153)	0.083 (**)
t-24	-0.038 (-0.124–0.024)	-0.038 (ns)	0.048 (-0.003–0.097)	0.048 (*)
t-23	-0.027 (-0.107–0.038)	-0.027 (ns)	0.021 (-0.051–0.074)	0.021 (ns)
t-22	-0.036 (-0.110–0.023)	-0.036 (ns)	0.052 (-0.014–0.115)	0.052 (ns)
t-21	0.010 (-0.067–0.110)	0.010 (ns)	0.087 (0.026–0.168)	0.087 (**)
t-20	0.009 (-0.062–0.093)	0.009 (ns)	0.051 (-0.004–0.106)	0.051 (*)
t-19	-0.038 (-0.130–0.030)	-0.038 (ns)	0.048 (-0.011–0.102)	0.048 (ns)
t-18	-0.030 (-0.118–0.047)	-0.030 (ns)	0.061 (0.002–0.123)	0.061 (*)
t-17	-0.010 (-0.083–0.063)	-0.010 (ns)	0.110 (0.065–0.172)	0.110 (****)
t-16	-0.005 (-0.068–0.066)	-0.005 (ns)	0.084 (0.026–0.148)	0.084 (**)
t-15	0.013 (-0.055–0.093)	0.013 (ns)	0.053 (-0.006–0.103)	0.053 (*)
t-14	-0.033 (-0.110–0.030)	-0.033 (ns)	0.061 (-0.005–0.121)	0.061 (*)
t-13	-0.073 (-0.164–0.015)	-0.073 (**)	0.087 (0.019–0.152)	0.087 (**)
t-12	0.015 (-0.040–0.076)	0.015 (ns)	0.110 (0.060–0.165)	0.110 (****)
t-11	-0.031 (-0.106–0.033)	-0.031 (ns)	0.076 (0.011–0.136)	0.076 (*)
t-10	-0.018 (-0.096–0.053)	-0.018 (ns)	0.073 (0.004–0.127)	0.073 (*)
t-9	-0.018 (-0.090–0.043)	-0.018 (ns)	0.105 (0.056–0.160)	0.105 (****)
t-8	0.001 (-0.056–0.059)	0.001 (ns)	0.117 (0.051–0.192)	0.117 (****)
t-7	-0.028 (-0.098–0.033)	-0.028 (ns)	0.163 (0.113–0.229)	0.163 (****)
t-6	-0.004 (-0.073–0.072)	-0.004 (ns)	0.193 (0.127–0.277)	0.193 (****)
t-5	0.003 (-0.047–0.069)	0.003 (ns)	0.209 (0.137–0.295)	0.209 (****)
t-4	-0.023 (-0.106–0.052)	-0.023 (ns)	0.180 (0.124–0.257)	0.180 (****)
t-3	-0.031 (-0.112–0.033)	-0.031 (ns)	0.212 (0.144–0.298)	0.212 (****)
t-2	-0.034 (-0.111–0.030)	-0.034 (ns)	0.253 (0.157–0.356)	0.253 (****)
t-1	-0.030 (-0.106–0.028)	-0.030 (ns)	0.297 (0.180–0.410)	0.297 (****)

Table 28. Parameters and Statistics for R + R.C (30) regression model for the Mohanta (2022) "Variable Block" dataset

Lag	Mean(95% CI) [C]	p-value [C]	Mean(95% CI) [R]	p-value [R]	Mean(95% CI) [R.C]	p-value [R.C]
t-30	0.034 (-0.019-0.084)	0.034 (ns)	-0.001 (-0.087-0.088)	-0.001 (ns)	-0.046 (-0.138-0.034)	-0.046 (ns)
t-29	0.049 (-0.013-0.109)	0.049 (ns)	-0.011 (-0.093-0.076)	-0.011 (ns)	-0.031 (-0.122-0.056)	-0.031 (ns)
t-28	0.032 (-0.015-0.078)	0.032 (ns)	-0.069 (-0.165-0.019)	-0.069 (ns)	-0.008 (-0.090-0.082)	-0.008 (ns)
t-27	0.076 (0.016-0.134)	0.076 (**)	0.006 (-0.091-0.096)	0.006 (ns)	0.002 (-0.081-0.086)	0.002 (ns)
t-26	-0.003 (-0.064-0.052)	-0.003 (ns)	0.092 (0.015-0.175)	0.092 (**)	0.058 (-0.026-0.155)	0.058 (ns)
t-25	0.079 (0.018-0.139)	0.079 (*)	0.034 (-0.044-0.119)	0.034 (ns)	-0.007 (-0.097-0.082)	-0.007 (ns)
t-24	0.013 (-0.053-0.075)	0.013 (ns)	-0.044 (-0.135-0.044)	-0.044 (ns)	0.023 (-0.075-0.121)	0.023 (ns)
t-23	0.043 (-0.017-0.105)	0.043 (ns)	-0.018 (-0.109-0.070)	-0.018 (ns)	-0.046 (-0.141-0.034)	-0.046 (ns)
t-22	0.040 (-0.024-0.105)	0.040 (ns)	-0.041 (-0.120-0.039)	-0.041 (ns)	0.004 (-0.095-0.110)	0.004 (ns)
t-21	-0.006 (-0.063-0.047)	-0.006 (ns)	0.021 (-0.070-0.110)	0.021 (ns)	0.094 (0.004-0.197)	0.094 (*)
t-20	0.035 (-0.017-0.093)	0.035 (ns)	0.024 (-0.075-0.121)	0.024 (ns)	-0.005 (-0.095-0.078)	-0.005 (ns)
t-19	0.013 (-0.044-0.069)	0.013 (ns)	-0.048 (-0.148-0.046)	-0.048 (ns)	0.021 (-0.054-0.098)	0.021 (ns)
t-18	0.015 (-0.036-0.070)	0.015 (ns)	-0.030 (-0.131-0.065)	-0.030 (ns)	0.030 (-0.059-0.123)	0.030 (ns)
t-17	0.046 (-0.003-0.095)	0.046 (*)	-0.011 (-0.103-0.073)	-0.011 (ns)	0.064 (-0.011-0.147)	0.064 (ns)
t-16	0.044 (-0.015-0.101)	0.044 (ns)	-0.002 (-0.082-0.078)	-0.002 (ns)	0.044 (-0.040-0.130)	0.044 (ns)
t-15	-0.009 (-0.059-0.042)	-0.009 (ns)	0.021 (-0.073-0.116)	0.021 (ns)	0.043 (-0.043-0.128)	0.043 (ns)
t-14	0.069 (0.022-0.119)	0.069 (**)	-0.034 (-0.115-0.053)	-0.034 (ns)	-0.014 (-0.104-0.069)	-0.014 (ns)
t-13	-0.003 (-0.060-0.055)	-0.003 (ns)	-0.099 (-0.183-0.017)	-0.099 (**)	0.085 (0.002-0.175)	0.085 (*)
t-12	0.072 (0.018-0.123)	0.072 (**)	0.029 (-0.046-0.115)	0.029 (ns)	0.042 (-0.046-0.126)	0.042 (ns)
t-11	0.011 (-0.037-0.059)	0.011 (ns)	-0.043 (-0.129-0.039)	-0.043 (ns)	0.060 (-0.019-0.138)	0.060 (ns)
t-10	0.045 (-0.002-0.090)	0.045 (*)	-0.015 (-0.102-0.067)	-0.015 (ns)	0.016 (-0.072-0.098)	0.016 (ns)
t-9	0.060 (0.005-0.115)	0.060 (*)	-0.016 (-0.089-0.061)	-0.016 (ns)	0.049 (-0.044-0.137)	0.049 (ns)
t-8	0.028 (-0.024-0.080)	0.028 (ns)	0.010 (-0.056-0.082)	0.010 (ns)	0.106 (0.013-0.199)	0.106 (*)
t-7	0.058 (0.009-0.109)	0.058 (*)	-0.021 (-0.102-0.053)	-0.021 (ns)	0.127 (0.049-0.206)	0.127 (****)
t-6	0.034 (-0.021-0.088)	0.034 (ns)	-0.008 (-0.096-0.077)	-0.008 (ns)	0.193 (0.105-0.287)	0.193 (****)
t-5	0.088 (0.035-0.139)	0.088 (**)	0.014 (-0.055-0.091)	0.014 (ns)	0.165 (0.091-0.249)	0.165 (****)

Lag	Mean(95% CI) [C]	p-value [C]	Mean(95% CI) [R]	p-value [R]	Mean(95% CI) [R-C]	p-value [R-C]
t-4	0.069 (0.018–0.118)	0.069 (**)	-0.030 (-0.115–0.064)	-0.030 (ns)	0.169 (0.078–0.265)	0.169 (****)
t-3	0.035 (-0.013–0.085)	0.035 (ns)	-0.038 (-0.126–0.045)	-0.038 (ns)	0.246 (0.151–0.338)	0.246 (****)
t-2	-0.003 (-0.058–0.056)	-0.003 (ns)	-0.054 (-0.142–0.027)	-0.054 (ns)	0.323 (0.216–0.426)	0.323 (****)
t-1	0.058 (-0.006–0.116)	0.058 (*)	-0.034 (-0.113–0.045)	-0.034 (ns)	0.319 (0.201–0.429)	0.319 (****)

Table 29. Parameters and Statistics for C + R + R.C (30) regression model for the Mohanta (2022) "Variable Block" dataset

Model	Rank	WAIC	pWAIC	SE	WAIC p-value	WAIC Effect Size	Test NL Mean	Test NL SE	Test NL p-value	Test NL Effect Size
RF-QL	21	17107.34	2.3609	82.8165	0.0000	19.6561	0.5433	0.0148	0.0000	0.7119
I-QL	20	16784.69	2.9514	88.0865	0.0000	16.4262	0.5455	0.0160	0.0000	0.8631
DE-I-QL	19	16770.19	3.2039	88.6000	0.0000	16.4444	0.5464	0.0162	0.0000	0.8467
I-OS-QL	18	16523.21	3.4472	89.7384	0.0000	14.4346	0.5547	0.0161	0.0000	0.7581
F-I-QL	17	16219.74	2.5555	94.7385	0.0000	10.0856	0.5611	0.0157	0.0002	0.6206
F-I-OS-QL	16	16045.22	3.1696	95.7450	0.0000	8.1446	0.5678	0.0159	0.0008	0.5413
SARSA	15	16030.15	4.6139	90.8139	0.0000	6.5232	0.5687	0.0168	0.0003	0.5888
ESARSA	14	16017.44	4.4857	91.6024	0.0000	6.7082	0.5688	0.0166	0.0015	0.5118
DF-I-OS-QL	13	16001.25	4.5270	95.1876	0.0000	7.3718	0.5689	0.0159	0.0004	0.5731
LT-QL	12	15960.57	4.5395	92.4267	0.0000	5.4212	0.5705	0.0166	0.0053	0.4429
DE-LT-QL	11	15960.12	5.4952	92.6002	0.0000	5.4549	0.5706	0.0166	0.0062	0.4345
LT-OS-QL	10	15947.45	5.6627	92.8901	0.0000	5.3344	0.5711	0.0166	0.0110	0.4007
F-RF-QL	9	15932.56	3.9579	93.4024	0.0000	4.3739	0.5722	0.0167	0.0056	0.4402
F-LT-QL	8	15917.8	4.0875	93.6404	0.0000	4.1463	0.5718	0.0167	0.0037	0.4625
F-LT-OS-QL	7	15916.32	5.1006	93.7372	0.0000	4.2238	0.5720	0.0167	0.0065	0.4310
DF-I-QL	6	15909.13	3.4566	92.1175	0.0001	3.9559	0.5716	0.0166	0.0068	0.4288
I-HV-QL	5	15873.05	6.6716	94.7419	0.0006	3.4541	0.5725	0.0162	0.0154	0.3804
F-I-HV-QL	4	15859.94	5.9649	94.0498	0.0016	3.1494	0.5728	0.0165	0.0192	0.3669
F-LT-HV-QL	3	15857.81	6.6859	94.1239	0.0195	2.3349	0.5732	0.0167	0.1596	0.2158
DF-LT-QL	2	15833.81	5.3402	93.2778	0.0803	1.7492	0.5736	0.0169	0.0831	0.2675
DF-LT-OS-QL	1	15802.58	6.0364	93.8576	0.5245	0.6365	0.5747	0.0170	0.5689	0.0866
LT-HV-QL	0	15790.49	9.3756	93.9523	-	-	0.5757	0.0170	-	-

Table 30. Q-Learning model fit statistics on the Mohanta (2022) "Variable Block" dataset.

Cognitive Factor	weight	intercept	alpha	gamma	tau	kappa	weight_r	weight_h	weight_b	alpha_r	kappa_r	alpha_h	theta_r	theta_h	theta
RPE	0.14 (p < 0.0001)	0.16 (p < 0.0001)	0.35 (p < 0.0001)	-	-	-	-	-	-	-	-	-	-	-	0.14 (p < 0.0001)
Forgetting	0.00 (p < 0.0001)	0.23 (p < 0.0001)	0.01 (p < 0.0001)	0.67 (p < 0.0001)	-	-	0.00 (p < 0.0001)	0.00 (p < 0.0001)	0.16 (p < 0.0001)	0.04 (p < 0.0001)	0.88 (p < 0.0001)	0.28 (p < 0.0001)	0.44 (p < 0.0001)	0.20 (p < 0.0001)	0.00 (p < 0.0001)
Learning - Forgetting	0.00 (p < 0.0001)	0.00 (p=0.70)	0.03 (p < 0.0001)	0.09 (p < 0.0001)	-	-	0.23 (p < 0.0001)	-	-	-	-	-	-	-	0.00 (p < 0.0001)
Discounting	0.59 (p < 0.0001)	0.17 (p < 0.0001)	0.33 (p < 0.0001)	-	0.99 (p < 0.0001)	0.34 (p < 0.0001)	0.72 (p < 0.0001)	0.45 (p < 0.0001)	0.51 (p < 0.0001)	0.66 (p < 0.0001)	0.03 (p < 0.0001)	0.57 (p < 0.0001)	0.34 (p < 0.0001)	0.26 (p < 0.0001)	0.59 (p < 0.0001)
Omission Sensitivity	0.03 (p < 0.0001)	0.01 (p < 0.0001)	0.01 (p < 0.0001)	0.00 (p < 0.0001)	-	0.34 (p < 0.0001)	-	-	-	-	-	-	-	-	0.03 (p < 0.0001)
On-Policy	0.01 (p < 0.0001)	0.01 (p < 0.0001)	0.00 (p < 0.0001)	0.02 (p < 0.0001)	-	-	-	-	-	-	-	-	-	-	0.01 (p < 0.0001)
Learning - Extinction Independence	0.05 (p < 0.0001)	0.00 (p < 0.0001)	0.00 (p < 0.0001)	0.01 (p < 0.0001)	-	-	-	-	-	-	-	-	-	-	0.05 (p < 0.0001)
APE	0.05 (p < 0.0001)	0.00 (p < 0.0001)	-	0.14 (p < 0.0001)	-	-	-	-	-	-	-	-	-	-	0.05 (p < 0.0001)
residual	0.12	0.42	0.28	0.07	0.01	0.32	0.05	0.54	0.33	0.3	0.09	0.15	0.22	0.55	0.12

Table 31. ANOVA summary (Cognitive variables vs. model parameters) for the Mohanta (2022) "Variable Block" dataset.

Eta-squared Effect Size and p-value reported.

Model	Test NL	Test SE	Test p-value	Test d	Training NL	Training SE	Training p-value	Training d
asymFFqN(2)	0.5252	0.0226	0.0000	1.6445	0.5345	0.0142	0.0000	1.9054
asymFFqN(2x2)	0.5175	0.0228	0.0000	1.6768	0.5264	0.0146	0.0000	1.9839
asymFFqN(5)	0.5100	0.0229	0.0000	1.6843	0.5171	0.0145	0.0000	1.9578
asymFFqN(5x5)	0.4857	0.0244	0.0000	1.6415	0.4928	0.0154	0.0000	1.8025
asymFFqN(10)	0.4943	0.0242	0.0000	1.7057	0.5011	0.0150	0.0000	1.8968
asymFFqN(10x10)	0.4694	0.0257	0.0000	1.7818	0.4750	0.0159	0.0000	2.1313
asymFFqN(10x10x10)	0.4647	0.0267	0.0000	1.6706	0.4694	0.0172	0.0000	1.8403
asymFFqN(100x100)	0.4542	0.0263	0.0000	1.7116	0.4587	0.0165	0.0000	2.0419
asymFFqN(100x100x100)	0.4166	0.0309	0.0000	1.6783	0.4166	0.0210	0.0000	1.8260
symFFqN(2)	0.5305	0.0212	0.0000	1.7848	0.5351	0.0139	0.0000	1.9348
symFFqN(2x2)	0.5133	0.0227	0.0000	1.6773	0.5177	0.0148	0.0000	1.7236
symFFqN(5)	0.5140	0.0225	0.0000	1.6475	0.5183	0.0145	0.0000	1.7996
symFFqN(5x5)	0.4832	0.0245	0.0000	1.6755	0.4852	0.0155	0.0000	1.7920
symFFqN(10)	0.5110	0.0228	0.0000	1.6839	0.5144	0.0147	0.0000	1.7893
symFFqN(10x10)	0.4900	0.0241	0.0000	1.6404	0.4929	0.0154	0.0000	1.7439
symFFqN(10x10x10)	0.4729	0.0262	0.0000	1.6736	0.4732	0.0174	0.0000	1.7120
symFFqN(100x100)	0.4815	0.0245	0.0000	1.7485	0.4842	0.0156	0.0000	1.7792
symFFqN(100x100x100)	0.4721	0.0263	0.0000	1.5523	0.4729	0.0170	0.0000	1.7536
asymRqN(2)	0.6216	0.0191	0.3099	0.1555	0.6309	0.0124	0.8907	0.0937
asymRqN(3)	0.6072	0.0194	0.0004	0.5762	0.6142	0.0125	0.2185	0.5718
asymRqN(5)	0.6024	0.0198	0.0457	0.5059	0.6137	0.0127	0.4126	0.3680
asymRqN(10)	0.6075	0.0197	0.1483	0.2315	0.6200	0.0125	0.8554	0.1043
asymRqN(100)	0.6500	0.0166	-	-	0.6561	0.0116	1.0000	-
symRqN(2)	0.6411	0.0171	0.7946	0.0398	0.6470	0.0117	0.9129	0.0656
symRqN(3)	0.6277	0.0175	0.6315	0.0731	0.6343	0.0118	0.7191	0.2028
symRqN(5)	0.6210	0.0178	0.3256	0.1517	0.6280	0.0119	0.8454	0.0975
symRqN(10)	0.6133	0.0176	0.0053	0.5302	0.6198	0.0117	0.2811	0.5232
symRqN(100)	0.6371	0.0169	0.7549	0.0480	0.6421	0.0115	0.8975	0.0675

Table 32. Statistics for the comparison of neural networks for the Mohanta (2022) "Variable Block" dataset.

References

- Abbott, L. F., & Dayan, P. (2001). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. MIT Press.
- Albert, J., & Hu, J. (2019). *Probability and Bayesian Modeling*. CRC Press.
- Alpay, T., Heinrich, S., & Wermter, S. (2016). Learning Multiple Timescales in Recurrent Neural Networks. In A. E. P. Villa, P. Masulli, & A. J. Pons Rivero (Eds.), *Artificial Neural Networks and Machine Learning – ICANN 2016* (pp. 132–139). Springer International Publishing.
https://doi.org/10.1007/978-3-319-44778-0_16
- Anselme, P., & Güntürkün, O. (2019). How foraging works: Uncertainty magnifies food-seeking motivation. *Behavioral and Brain Sciences*, 42.
<https://doi.org/10.1017/S0140525X18000948>
- Aso, Y., & Rubin, G. M. (2016). Dopaminergic neurons write and update memories with cell-type-specific rules. *ELife*, 5, e16135.
<https://doi.org/10.7554/eLife.16135>
- Aso, Y., Sitaraman, D., Ichinose, T., Kaun, K. R., Vogt, K., Belliard-Guérin, G., Plaçais, P.-Y., Robie, A. A., Yamagata, N., Schnaitmann, C., Rowell, W. J., Johnston, R. M., Ngo, T.-T. B., Chen, N., Korff, W., Nitabach, M. N., Heberlein, U., Preat, T., Branson, K. M., ... Rubin, G. M. (2014). Mushroom body output neurons encode valence and guide memory-based action selection in *Drosophila*. *ELife*, 3, e04580. <https://doi.org/10.7554/eLife.04580>
- Bannard, C., Leriche, M., Bandmann, O., Brown, C. H., Ferracane, E., Sánchez-Ferro, Á., Obeso, J., Redgrave, P., & Stafford, T. (2019). Reduced habit-driven errors in Parkinson's Disease. *Scientific Reports*, 9, 3423.
<https://doi.org/10.1038/s41598-019-39294-z>

- Bari, B. A., Grossman, C. D., Lubin, E. E., Rajagopalan, A. E., Cressy, J. I., & Cohen, J. Y. (2019). Stable Representations of Decision Variables for Flexible Behavior. *Neuron*, 103(5), 922-933.e7.
<https://doi.org/10.1016/j.neuron.2019.06.001>
- Beckmann, J. S., & Chow, J. J. (2015). Isolating the incentive salience of reward-associated stimuli: Value, choice, and persistence. *Learning & Memory*, 22(2), 116–127. <https://doi.org/10.1101/lm.037382.114>
- Beron, C. C., Neufeld, S. Q., Linderman, S. W., & Sabatini, B. L. (2022). Mice exhibit stochastic and efficient action switching during probabilistic decision making. *Proceedings of the National Academy of Sciences*, 119(15), e2113961119.
<https://doi.org/10.1073/pnas.2113961119>
- Bogacz, R. (2020). Dopamine role in learning and action inference. *ELife*, 9, e53262.
<https://doi.org/10.7554/eLife.53262>
- Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). OpenAI Gym. ArXiv:1606.01540 [Cs].
<http://arxiv.org/abs/1606.01540>
- Brunton, S. L., Proctor, J. L., & Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15), 3932–3937.
<https://doi.org/10.1073/pnas.1517384113>
- Campbell, R. A. A., Honegger, K. S., Qin, H., Li, W., Demir, E., & Turner, G. C. (2013). Imaging a Population Code for Odor Identity in the *Drosophila* Mushroom Body. *Journal of Neuroscience*, 33(25), 10568–10581.
<https://doi.org/10.1523/JNEUROSCI.0682-12.2013>
- Cavagnaro, D. R., Gonzalez, R., Myung, J. I., & Pitt, M. A. (2013). Optimal Decision

- Stimuli for Risky Choice Experiments: An Adaptive Approach. *Management Science*, 59(2), 358–375. <https://doi.org/10.1287/mnsc.1120.1558>
- Costa, T. M., Hebets, E. A., Melo, D., & Willemart, R. H. (2016). Costly learning: Preference for familiar food persists despite negative impact on survival. *Biology Letters*, 12(7), 20160256. <https://doi.org/10.1098/rsbl.2016.0256>
- Dan, O., & Loewenstein, Y. (2019). From choice architecture to choice engineering. *Nature Communications*, 10(1), Article 1. <https://doi.org/10.1038/s41467-019-10825-6>
- Davis, R. L., & Zhong, Y. (2017). The Biology of Forgetting – A Perspective. *Neuron*, 95(3), 490–503. <https://doi.org/10.1016/j.neuron.2017.05.039>
- Dezfouli, A., Nock, R., & Dayan, P. (2020). Adversarial vulnerabilities of human decision-making. *Proceedings of the National Academy of Sciences*, 117(46), 29221–29228. <https://doi.org/10.1073/pnas.2016921117>
- Dickinson, A. (2012). Associative learning and animal cognition. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367(1603), 2733–2742. <https://doi.org/10.1098/rstb.2012.0220>
- Dukas, R. (2008). Evolutionary Biology of Insect Learning. *Annual Review of Entomology*, 53(1), 145–160. <https://doi.org/10.1146/annurev.ento.53.103106.093343>
- F. Hernandez, L., Redgrave, P., & Obeso, J. (2015). Habitual behavior and dopamine cell vulnerability in Parkinson disease. *Frontiers in Neuroanatomy*, 9. <https://www.frontiersin.org/articles/10.3389/fnana.2015.00099>
- Gadziola, M. A., Stetzik, L. A., Wright, K. N., Milton, A. J., Arakawa, K., del Mar Cortijo, M., & Wesson, D. W. (2020). A Neural System that Represents the Association of Odors with Rewarded Outcomes and Promotes Behavioral

Engagement. *Cell Reports*, 32(3), 107919.

<https://doi.org/10.1016/j.celrep.2020.107919>

Gonzalez, R. C., Behrend, E. R., & Bitterman, M. E. (1967). Reversal Learning and Forgetting in Bird and Fish. *Science*, 158(3800), 519–521.

Goodman, J., & Packard, M. G. (2019). There Is More Than One Kind of Extinction Learning. *Frontiers in Systems Neuroscience*, 13.

<https://www.frontiersin.org/article/10.3389/fnsys.2019.00016>

Greenstreet, F., Vergara, H. M., Pati, S., Schwarz, L., Wisdom, M., Marbach, F., Johansson, Y., Rollik, L., Moskovitz, T., Clopath, C., & Stephenson-Jones, M. (2022). Action prediction error: A value-free dopaminergic teaching signal that drives stable learning (p. 2022.09.12.507572). *bioRxiv*.

<https://doi.org/10.1101/2022.09.12.507572>

Greggers, U., & Menzel, R. (1993). Memory dynamics and foraging strategies of honeybees. *Behavioral Ecology and Sociobiology*, 32(1), 17–29.

<https://doi.org/10.1007/BF00172219>

Guo, J., & Guo, A. (2005). Crossmodal Interactions Between Olfactory and Visual Learning in *Drosophila*. *Science*, 309(5732), 307–310.

<https://doi.org/10.1126/science.1111280>

Haber Kern, H., Basnak, M. A., Ahanonu, B., Schauder, D., Cohen, J. D., Bolstad, M., Bruns, C., & Jayaraman, V. (2019). Visually Guided Behavior and Optogenetically Induced Learning in Head-Fixed Flies Exploring a Virtual Landscape. *Current Biology*, 29(10), 1647-1659.e8.

<https://doi.org/10.1016/j.cub.2019.04.033>

Haber Kern, H., & Jayaraman, V. (2016). Studying small brains to understand the building blocks of cognition. *Current Opinion in Neurobiology*, 37, 59–65.

<https://doi.org/10.1016/j.conb.2016.01.007>

Hafner, D. (2017). Tips for Training Recurrent Neural Networks.

<https://danijar.com/tips-for-training-recurrent-neural-networks/>

Hales, K. G., Korey, C. A., Larracuenta, A. M., & Roberts, D. M. (2015). Genetics on the Fly: A Primer on the *Drosophila* Model System. *Genetics*, 201(3), 815–842. <https://doi.org/10.1534/genetics.115.183392>

Hall-McMaster, S., & Luyckx, F. (2019). Revisiting foraging approaches in neuroscience. *Cognitive, Affective & Behavioral Neuroscience*, 19(2), 225–230. <https://doi.org/10.3758/s13415-018-00682-z>

Handler, A., Graham, T. G. W., Cohn, R., Morantte, I., Siliciano, A. F., Zeng, J., Li, Y., & Ruta, V. (2019). Distinct Dopamine Receptor Pathways Underlie the Temporal Sensitivity of Associative Learning. *Cell*, 178(1), 60-75.e19. <https://doi.org/10.1016/j.cell.2019.05.040>

Hayden, B. Y. (2016). Time discounting and time preference in animals: A critical review. *Psychonomic Bulletin & Review*, 23(1), 39–53. <https://doi.org/10.3758/s13423-015-0879-3>

Heisenberg, M. (2003). Mushroom body memoir: From maps to models. *Nature Reviews Neuroscience*, 4(4), Article 4. <https://doi.org/10.1038/nrn1074>

Hermoso-Mendizabal, A., Hyafil, A., Rueda-Orozco, P. E., Jaramillo, S., Robbe, D., & de la Rocha, J. (2020). Response outcomes gate the impact of expectations on perceptual decisions. *Nature Communications*, 11(1), Article 1. <https://doi.org/10.1038/s41467-020-14824-w>

Herrnstein, R. J. (1961). Relative and absolute strength of response as a function of frequency of reinforcement. *Journal of the Experimental Analysis of Behavior*, 4(3), 267–272. <https://doi.org/10.1901/jeab.1961.4-267>

- Hige, T., Aso, Y., Modi, M. N., Rubin, G. M., & Turner, G. C. (2015). Heterosynaptic plasticity underlies aversive olfactory learning in *Drosophila*. *Neuron*, 88(5), 985–998. <https://doi.org/10.1016/j.neuron.2015.11.003>
- Honegger, K. S., Campbell, R. A. A., & Turner, G. C. (2011). Cellular-Resolution Population Imaging Reveals Robust Sparse Coding in the *Drosophila* Mushroom Body. *Journal of Neuroscience*, 31(33), 11772–11785. <https://doi.org/10.1523/JNEUROSCI.1099-11.2011>
- Ito, M., & Doya, K. (2009). Validation of Decision-Making Models and Analysis of Decision Variables in the Rat Basal Ganglia. *Journal of Neuroscience*, 29(31), 9861–9874. <https://doi.org/10.1523/JNEUROSCI.6157-08.2009>
- Kamil, A. (1985). The Ecology of Foraging Behavior: Implications for Animal Learning and Memory. *Annual Review of Psychology*, 36, 141–169. <https://doi.org/10.1146/annurev.psych.36.1.141>
- Kelly, F. P. (1981). Multi-Armed Bandits with Discount Factor Near One: The Bernoulli Case. *The Annals of Statistics*, 9(5), 987–1001.
- Kilpatrick, Z. P., Davidson, J. D., & El Hady, A. (2021). Uncertainty drives deviations in normative foraging decision strategies. *Journal of The Royal Society Interface*, 18(180), 20210337. <https://doi.org/10.1098/rsif.2021.0337>
- Kim, H. F., Ghazizadeh, A., & Hikosaka, O. (2015). Dopamine Neurons Encoding Long-Term Memory of Object Value for Habitual Behavior. *Cell*, 163(5), 1165–1175. <https://doi.org/10.1016/j.cell.2015.10.063>
- Koutník, J., Greff, K., Gomez, F., & Schmidhuber, J. (2014). A Clockwork RNN (arXiv:1402.3511). arXiv. <https://doi.org/10.48550/arXiv.1402.3511>
- Krebs, J. R., & Inman, A. J. (2015). Learning and Foraging: Individuals, Groups, and Populations. *The American Naturalist*. <https://doi.org/10.1086/285397>

- Lak, A., Okun, M., Moss, M. M., Gurnani, H., Farrell, K., Wells, M. J., Reddy, C. B., Kepecs, A., Harris, K. D., & Carandini, M. (2020). Dopaminergic and Prefrontal Basis of Learning from Sensory Confidence and Reward Value. *Neuron*, 105(4), 700-711.e6. <https://doi.org/10.1016/j.neuron.2019.11.018>
- Lau, B., & Glimcher, P. W. (2005). Dynamic Response-by-Response Models of Matching Behavior in Rhesus Monkeys. *Journal of the Experimental Analysis of Behavior*, 84(3), 555–579. <https://doi.org/10.1901/jeab.2005.110-04>
- Li, F., Lindsey, J. W., Marin, E. C., Otto, N., Dreher, M., Dempsey, G., Stark, I., Bates, A. S., Pleijzier, M. W., Schlegel, P., Nern, A., Takemura, S., Eckstein, N., Yang, T., Francis, A., Braun, A., Parekh, R., Costa, M., Scheffer, L. K., ... Rubin, G. M. (2020). The connectome of the adult *Drosophila* mushroom body provides insights into function. *ELife*, 9, e62576. <https://doi.org/10.7554/eLife.62576>
- López-Yépez, J. S., Martin, J., Hulme, O., & Kvitsiani, D. (2021). Choice history effects in mice and humans improve reward harvesting efficiency. *PLOS Computational Biology*, 17(10), e1009452. <https://doi.org/10.1371/journal.pcbi.1009452>
- Ma, T., & Hermundstad, A. M. (2022). A vast space of compact strategies for highly efficient decisions (p. 2022.08.10.503471). *bioRxiv*. <https://doi.org/10.1101/2022.08.10.503471>
- Markowetz, F. (2010). How to Understand the Cell by Breaking It: Network Analysis of Gene Perturbation Screens. *PLoS Computational Biology*, 6(2), e1000655. <https://doi.org/10.1371/journal.pcbi.1000655>
- Matheson, A. M. M., Lanz, A. J., Medina, A. M., Licata, A. M., Currier, T. A., Syed, M. H., & Nagel, K. I. (2022). A neural circuit for wind-guided olfactory navigation.

Nature Communications, 13(1), Article 1.

<https://doi.org/10.1038/s41467-022-32247-7>

McElreath, R. (2016). *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*. CRC Press.

Mery, F. (2008). Evolutionary biology of learning in insects: The search for food. In *Insect Taste*. Taylor & Francis.

Miller, K. J., Botvinick, M. M., & Brody, C. D. (2021). From predictive models to cognitive models: Separable behavioral processes underlying reward learning in the rat (p. 461129). *bioRxiv*. <https://doi.org/10.1101/461129>

Miller, K. J., Shenhav, A., & Ludvig, E. A. (2019). Habits without values. *Psychological Review*, 126(2), 292–311. <https://doi.org/10.1037/rev0000120>

Mohanta, R., Turner, G. C., & Shuai, Y. (2019). Investigating Odor-based Learning in Closed-Loop Fly-on-Ball VR. <https://doi.org/10.6084/m9.figshare.9751004.v2>

Niv, Y. (2009). Reinforcement learning in the brain. *Journal of Mathematical Psychology*, 53, 139–154. <https://doi.org/10.1016/j.jmp.2008.12.005>

Owald, D., Lin, S., & Waddell, S. (2015). Light, heat, action: Neural control of fruit fly behaviour. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 370(1677), 20140211. <https://doi.org/10.1098/rstb.2014.0211>

Pachur, T., Hertwig, R., Gigerenzer, G., & Brandstätter, E. (2013). Testing process predictions of models of risky choice: A quantitative model comparison approach. *Frontiers in Psychology*, 4, 646. <https://doi.org/10.3389/fpsyg.2013.00646>

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training Recurrent Neural Networks (arXiv:1211.5063). *arXiv*. <https://doi.org/10.48550/arXiv.1211.5063>

- Premack, D. (2007). Human and animal cognition: Continuity and discontinuity. *Proceedings of the National Academy of Sciences*, 104(35), 13861–13867. <https://doi.org/10.1073/pnas.0706147104>
- Rajagopalan, A. E., Darshan, R., Fitzgerald, J. E., & Turner, G. C. (2022). Expectation-based learning rules underlie dynamic foraging in *Drosophila* (p. 2022.05.24.493252). *bioRxiv*. <https://doi.org/10.1101/2022.05.24.493252>
- Rescorla, R. A., & Holland, P. C. (1982). Behavioral Studies of Associative Learning in Animals. *Annual Review of Psychology*, 33(1), 265–308. <https://doi.org/10.1146/annurev.ps.33.020182.001405>
- Rushworth, M. F. S., & Behrens, T. E. J. (2008). Choice, uncertainty and value in prefrontal and cingulate cortex. *Nature Neuroscience*, 11(4), Article 4. <https://doi.org/10.1038/nn2066>
- Schäfer, A. M., & Zimmermann, H. G. (2006). Recurrent Neural Networks Are Universal Approximators. In S. D. Kollias, A. Stafylopatis, W. Duch, & E. Oja (Eds.), *Artificial Neural Networks – ICANN 2006* (pp. 632–640). Springer. https://doi.org/10.1007/11840817_66
- Schultz, W. (2016). Dopamine reward prediction error coding. *Dialogues in Clinical Neuroscience*, 18(1), 23–32.
- Seidenbecher, S. E., Sanders, J. I., Philipsborn, A. C. von, & Kvitsiani, D. (2020). Reward foraging task and model-based analysis reveal how fruit flies learn value of available options. *PLOS ONE*, 15(10), e0239616. <https://doi.org/10.1371/journal.pone.0239616>
- Shettleworth, S. J. (1985). Foraging, memory, and constraints on learning. *Annals of the New York Academy of Sciences*, 443, 216–226. <https://doi.org/10.1111/j.1749-6632.1985.tb27075.x>

- Shteingart, H., & Loewenstein, Y. (2014). Reinforcement learning and human behavior. *Current Opinion in Neurobiology*, 25, 93–98.
<https://doi.org/10.1016/j.conb.2013.12.004>
- Simpson, J. H., & Looger, L. L. (2018). Functional Imaging and Optogenetics in *Drosophila*. *Genetics*, 208(4), 1291–1309.
<https://doi.org/10.1534/genetics.117.300228>
- Sonoda, S., & Murata, N. (2017). Neural network with unbounded activation functions is universal approximator. *Applied and Computational Harmonic Analysis*, 43(2), 233–268. <https://doi.org/10.1016/j.acha.2015.12.005>
- Sugrue, L. P., Corrado, G. S., & Newsome, W. T. (2004). Matching Behavior and the Representation of Value in the Parietal Cortex. *Science*, 304(5678), 1782–1787. <https://doi.org/10.1126/science.1094765>
- Sul, J. H., Kim, H., Huh, N., Lee, D., & Jung, M. W. (2010). Distinct Roles of Rodent Orbitofrontal and Medial Prefrontal Cortex in Decision Making. *Neuron*, 66(3), 449–460. <https://doi.org/10.1016/j.neuron.2010.03.033>
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). A Bradford Book.
- Tempel, B. L., Bonini, N., Dawson, D. R., & Quinn, W. G. (1983). Reward learning in normal and mutant *Drosophila*. *Proceedings of the National Academy of Sciences*, 80(5), 1482–1486. <https://doi.org/10.1073/pnas.80.5.1482>
- Todorov, J. C., de Oliveira Castro, J. M., Hanna, E. S., Bittencourt de Sa, M. C., & Barreto, M. Q. (1983). Choice, experience, and the generalized matching law. *Journal of the Experimental Analysis of Behavior*, 40(2), 99–111.
<https://doi.org/10.1901/jeab.1983.40-99>
- Udrescu, S.-M., & Tegmark, M. (2020). *AI Feynman: A physics-inspired method for*

symbolic regression. *Science Advances*, 6(16), eaay2631.

<https://doi.org/10.1126/sciadv.aay2631>

Vehtari, A., Gelman, A., & Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, 27(5), 1413–1432. <https://doi.org/10.1007/s11222-016-9696-4>

Wiecki, T., Salvatier, J., Patil, A., Kochurov, M., Vieira, R., Engels, B., Lao, J., Colin, Martin, O., Osthege, M., Willard, B. T., Seyboldt, A., Rochford, A., rpgoldman, Paz, L., Meyer, K., Coyle, P., Gorelli, M. E., Kumar, R., ... Domenzain, L. M. (2022). *pymc-devs/pymc: 4.0.0 beta 6*. Zenodo.

<https://doi.org/10.5281/zenodo.6396757>

Wood, W., Labrecque, J. S., Lin, P.-Y., & Runger, D. (2014). Habits in dual-process models. In *Dual-process theories of the social mind* (pp. 371–385). The Guilford Press.

Yagi, R., Mabuchi, Y., Mizunami, M., & Tanaka, N. K. (2016). Convergence of multimodal sensory pathways to the mushroom body calyx in *Drosophila melanogaster*. *Scientific Reports*, 6(1), Article 1.

<https://doi.org/10.1038/srep29481>

Yamada, D., Bushey, D., Feng, L., Hibbard, K., Sammons, M., Funke, J., Litwin-Kumar, A., Hige, T., & Aso, Y. (2022). Hierarchical architecture of dopaminergic circuits enables second-order conditioning in *Drosophila* (p. 2022.03.30.486484). *bioRxiv*. <https://doi.org/10.1101/2022.03.30.486484>

Yang, X.-S. (Ed.). (2014). *Nature-Inspired Optimization Algorithms*. In *Nature-Inspired Optimization Algorithms* (p. i). Elsevier.

<https://doi.org/10.1016/B978-0-12-416743-8.00016-6>

Zhang, Y., Tsang, T. K., Bushong, E. A., Chu, L.-A., Chiang, A.-S., Ellisman, M. H.,

Reingruber, J., & Su, C.-Y. (2019). Asymmetric ephaptic inhibition between compartmentalized olfactory receptor neurons. *Nature Communications*, 10(1), 1560. <https://doi.org/10.1038/s41467-019-09346-z>

Zolin, A., Cohn, R., Pang, R., Siliciano, A. F., Fairhall, A. L., & Ruta, V. (2021). Context-dependent representations of movement in *Drosophila* dopaminergic reinforcement pathways. *Nature Neuroscience*, 24(11), 1555–1566. <https://doi.org/10.1038/s41593-021-00929-y>

