Models That Learn What We Learn: The Role of the Basal Ganglia as a Behavioral Regularizer

Frederick Berl[1,✉], **Daeyeol Lee**[2], **Hyojung Seo**[3], **and John D. Murray**[4]

[1]Department of Applied Physics, Yale School of Engineering and Applied Science
[2]The Zanvyl Krieger Mind/Brain Institute, Department of Neuroscience, Psychological and Brain Sciences. Kavli Neuroscience Discovery Institute, Johns Hopkins University
[3]Department of Psychiatry and Neuroscience, Yale School of Medicine
[4]Department of Psychological and Brain Sciences, Dartmouth University

One of the open questions at the intersection of neuroscience and computer science is developing models that learn how we learn; although artificial neural networks are used ubiquitously to solve reinforcement learning problems and model behavior, in complex settings they find solutions to problems that are alien to humans or animals, even when biological constraints are imposed. One can view the brain as a hierarchy of systems working together to solve problems at various levels of complexity and as efficiently as possible, but ANNs have no such structure constraining the types of strategies they learn. Here we demonstrate that a recurrent neural network architecture with a concomitant reinforcement learning model mimicking the interplay between the prefrontal cortex and the basal ganglia is able to learn strategic deviations from RL that are more similar to those learned by monkeys performing the same task than traditional models. Unlike a mixture of experts model, these models learn and work together in a multi-agent framework in order to discover cognitive strategies.

Reinforcement Learning | Cognitive Strategies | Cognitive Control
Correspondence: ***frederick.berl@yale.edu***

Introduction

When modeling cognitive tasks in neuroscience with artificial neural networks, researchers take one of two potential approaches. The first is to train models that can perform the task in order to draw analogies between the dynamics of the model and either attractors of a dynamical system that solves an equivalent problem or the dynamics of a population of neurons from an animal performing the same task. This approach works well in simpler tasks where we can break task steps down into bifurications and attractors, as well as when you are only probing first order effects of task variables like in the case of evidence accumulation tasks (Mante et al., 2013; Galgali et al., 2023). The second approach is instead to train networks that are capable of predicting an animal's behavior. These models learn from the behavioral data instead of by performing the task and may provide insights on the sorts of strategies employed by these animals (Eckstein et al., 2024; Ji-An et al., 2025). These approaches can even be combined in order to infer behavior and strategies (Sussillo and Barak, 2013; Ji-An et al., 2025).

Unfortunately, these methods are not applicable to every problem, especially as they approach real world situations. Some tasks are very difficult to model using a dynamical systems framework, and when the task requires long dependencies on prior steps, the ANNs are much harder to analyze. This is due to the fact that the ANNs are less constrained than the brain and learned behavior can be different than expected since there may not be a single solution but rather multiple (locally) acceptable solutions to such problems. Similarly, when training on animal behavioral data, these networks may fail to provide insights of what may be happening on a neurological level as behavior is an emergent property of structure and not vice versa. There is also the question of how much of the learned behavior of these models is through memorization as opposed to the actual learning of cognitive strategies. Therefore, learning on animal data may tell you about the strategies employed in a task, but gives you no information on how or why that strategy was found and employed. Additionally, it requires having animal data to learn on which also limits the sorts of tasks you can study and therefore what can be learned. Lastly, although researchers can sometimes draw comparisons between the neural dynamics of the animals and the models performing these tasks, in many cases they are not learning anything novel. There is a distinct dearth of models that will learn behavioral strategies similar to animals, i.e. that are capable of (re)producing behaviors without actually being fit to data that contains those behaviors.

There are numerous potential reasons for this, ranging from details such as artificial neurons actually being more capable at maintaining state than biological neurons, or learning through the more powerful backpropagation instead of a biologically plausible method, or even value function computations being different in the brain than in the models, but our hypothesis was that the biggest barrier was structural. RNNs that use value functions are most similar to the Prefrontal Cortex (PFC), but the brain has a hierarchy of systems that work together simultaneously and exert differing levels of control. Because these RNNs are unconstrained, they can oftentimes find solutions that are different than animals', especially in situations where deviations from RL are required and where there are multiple acceptable solutions. Therefore, it makes more sense to build a larger model out of smaller subsystems that have different objectives or inductive biases and are forced to or learn to collaborate instead of a single

large model that learns to perform the task. We took this last approach in order to build models that could better learn strategies similar to those exhibited by monkeys playing the matching pennies cognitive task. This task is a binary zero-sum game where the animals are required to make one of two choices with the objective of matching their opponents' choice **(Fig. 1A)**. These animals played against two computer opponents, where the first opponent only looked at patterns in the monkeys' choices while the second also analyzed how those patterns were influenced by the rewards the monkeys had received. Against both of these opponents, optimal performance is winning 50% of the time, and against the first opponent, the Nash equilibrium allows for simple reinforcement learning strategies, while for the second opponent, only random guessing is theoretically optimal. An infinitely large family of nearly optimal strategies such as those exhibited by the monkeys does exist however.

Here we have a task with known RL-like behavioral signatures, but in order to perform the task optimally animals need to deviate from RL and play more strategically. Therefore, this is the ideal task to learn how the brain performs and makes decisions in such situations. As one can surmise from the second opponent, this task is deceptively complex, and there are additional reasons we chose this task. First, since we have data for the two opponents, one of which can be played against using the standard reinforcement learning strategies animals tend to come up with during these cognitive tasks, and one which requires stochasticity, which the brain as a deterministic system is bad at, it gives us a good comparison and allows us to quantify how much the strategies change from one task to the other. In other words, we can study how animals deviate from RL behavior and develop a model that is better able to capture these changes. Second, with the knowledge that brains are not stochastic, or at least not fully so, we know that the monkeys are integrating over the choices they have made and rewards they have gotten in order to make decisions and this should be corroborated by other analyses performed on the neural data by other researchers.

Results

Monkey behavior is heterogeneous and can be categorized by behavioral biases and deviations from expected RL behavior. Our first question was what sort of strategies the monkeys came up with when playing against the second opponent, where they needed to perform stochastically. In order to extract interpretable behavioral signatures, we fit logistic regressions to the monkey behavior. The reason we are including this despite previously being investigated by other researchers (Lee et al., 2004; Barraclough et al., 2004; Donahue et al., 2013) is because these regressions typically use a basis of monkey choice, opponent choice or reward. However, because we care more about the strategies, we split it up into the four possible (and mutually exclusive) decisions in a trial: Repeat Win, Change Win, Repeat Lose, Change Lose which is non-standard. This basis is related to the expected Win-Stay Lose-Switch behavior, but further conditions on the preceding choices. More rigorously, we defined a generalized logistic regression detailed in the methods section and this basis was demonstrably the best performing **(Fig. S1)** in the strategic monkeys as well as most consistent with our later observations about the level of cognitive control exerted in our models.

When we did this, we noticed that we had two groupings based on possible shapes of the regression coefficients. The first, which we labeled non-strategic, exhibited simpler RL-like dynamics which manifests in the regression coefficients monotonically decaying towards 0 as we can see in **1D**. The second group we denoted as strategic because they more obviously deviated from the expected RL behavior. We saw that there are biases in some of the regression coefficients that are asymmetric between trials where the animal repeated the previous choice and where they changed to their other choice, and which is related to win and loss in the preceding trial, leading some to increase before they decrease. We quantified this across every session for each monkey for both opponent 1 and 2 in **Fig. 1C** by computing the deviations from expected RL behavior after repeating or changing actions. This was done by taking the difference between each of the regression coefficient curves for repeat ($\beta_{repeat} = \beta_{rw} - \beta_{rl}$) and change ($\beta_{change} = \beta_{cl} - \beta_{cw}$, see **Fig. S2**) and computing the metric $\xi = \frac{\beta_1 - \beta_2}{\max|\beta|}$. This metric has an intuitive interpretation predicated on the fact that ideal RL produces regression coefficients that linearly and monotonically decay towards zero. Therefore, behavior that is RL to first order should have $\xi_i \in [0, 1]$, denoted by the dashed box, where the behavioral timescale $\alpha_i \propto 1/\xi_i$. The non-strategic monkeys and the monkeys playing opponent 1 are all in the dashed box, which means that the first component of each regressor was consistently the largest, while the strategic monkeys deviate from this against opponent two. Of note, Monkey H exhibits small deviations from RL in one of the components, but not to the extent observed in the strategic monkeys. The regressions were fit at the monkey level and error bars were computed through leave-one-out cross-validation at the session level.

Given the differences seen between the behavioral regressions of monkeys, we wondered to what extent these behaviors and deviations were due to learning versus inherent to each monkey. We fit each of the three RL models on the data for each monkey to compare them **(Fig. S3)**, although we settled on the simple RL module because the RNN should be able to induce an asymmetry by creating an internal model of the RL module and counteracting it and also because the simple RL model avoids a fine-tuning problem introduced by the asymmetric and forgetting models where the dynamics are sensitive to the ratios $\alpha_{win}/\alpha_{loss}$ and $\Delta_{win}/\Delta_{loss}$ **(Methods)**. In contrast, the simple model allows us to set a single parameter α which determines the RL model behavioral timescale ($\tau_{RL} = 1/\alpha_{RL}$). We also wondered to what extent the decisions the monkeys were making appeared stochastic and how that evolved throughout the experiment, as well as to what extent the smaller RL models were able to capture behavioral variance as compared to the behavioral re-

Figure 1: Monkey Behavior Can Be Categorized Into Strategic and Non-Strategic Based on Behavioral Asymmetry

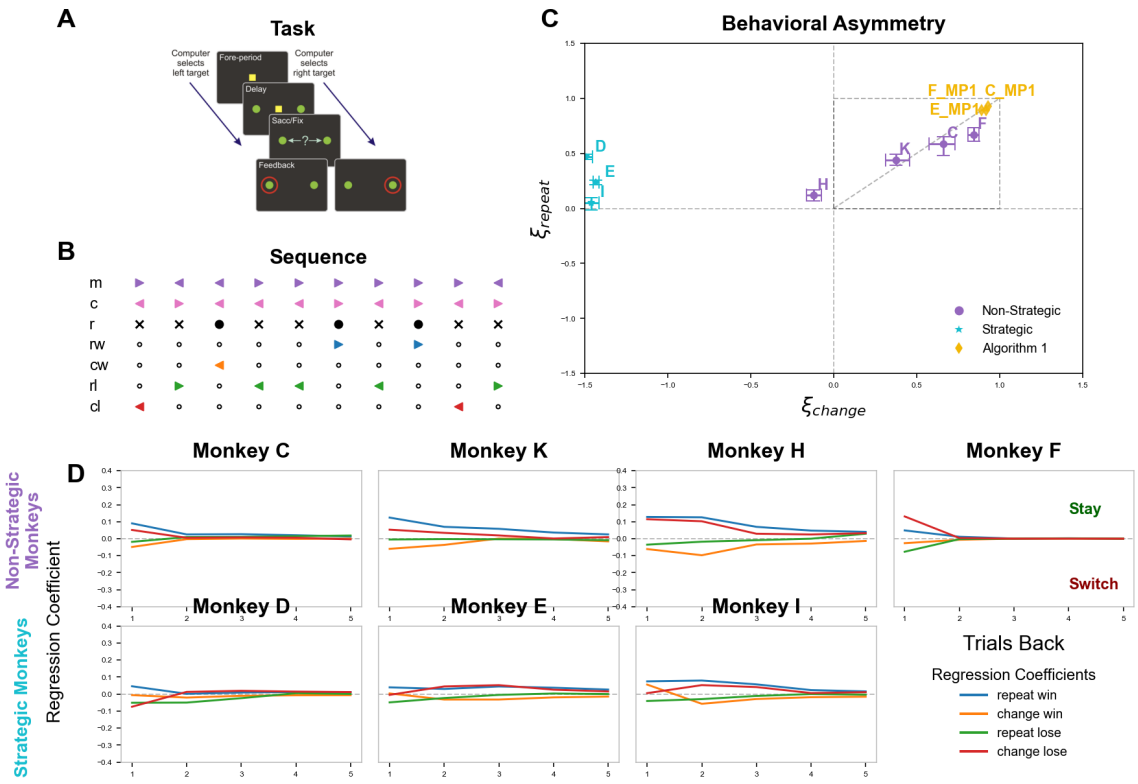


Fig. 1. (A) The matching pennies task is a competitive game where monkeys play against a computer opponent designed to take advantage of statistical biases in their decision making. In order to win, the monkey must predict what the computer opponent will do. There are two versions of this task; opponent 1 only looks at biases in the monkeys' decisions, while opponent 2 also looks at whether these decisions are influenced by rewards. Consequently, standard reinforcement learning strategies such as win-stay and lose-switch are able to perform optimally against opponent 1, but performing well against opponent 2 requires stochasticity. (B) Sequence of 10 trials from a session where monkey E played against the computer. In the behavioral basis used for logistic regressions, we can use the monkey action m and computer action c or reward outcome r in order to predict future choices. Here we introduce a strategic basis where each regressor is composed of combinations of m and r. These regressors are repeat-win rw, change-lose cl, change-win cw, and repeat-lose rl respectively. (C) Measure of deviations from expected RL behavior after wins and losses against opponents 1 and 2. We categorized monkeys into strategic and non-strategic based on their performance versus opponent 2. The dashed region denotes behavior in line with RL strategies, while everything outside of this region deviates from RL. Strategic monkeys show strong behavioral deviations from simple reinforcement learning behavior, while non-strategic monkeys are more consistent with asymmetric RL models such as the forgetting model when playing against opponent 2. (D) Logistic regressions for the entire dataset for each monkey. Strategic monkeys show deviations from reinforcement learning on the loss regressions that push them towards switching again after a change-lose or staying again after a lose-stay. Positive values for the regression coefficients correspond to that action-reward conjunction influencing the animal towards repeating the same choice, while negative values correspond to switching.

gressions. For comparing stochasticity, we used $\alpha \times \beta$ as our metric since this quantity is inversely proportional to $\tau \times T$ and therefore stochasticity.

Here we observe that although the three monkeys seemed to have distinct strategies against opponent 2, their approach to the simpler opponent 1 is similar, with the caveat that monkey E's behavior was more RL-like as it was better predicted by the behavioral regression. In contrast, the simpler RL model fit to monkey E for each of the two opponents learned timescales much longer than what we see for the monkeys' behavior, and with different biases. These models are also only able to perform well on the task when sampling choices probabilistically as opposed to deterministically.

Although the regressions for the three monkeys look similar, we observe that there is a stark difference between monkey E and monkeys C and F. Monkey E's behavior is better predicted by the behavioral regression than the other two and is increasingly deterministic as it plays more and more versus opponent 1. When playing versus the second opponent, all monkeys exhibit a higher degree of stochasticity, and this does not change appreciably as strategies evolve, although the strategic monkeys appear more stochastic than the non-strategic animals.

We can further quantify the deviations from reinforcement learning behavior by comparing how well the RL models and logistic regressions fit each monkey against each opponent. The explainable behavior (to first order) is bounded by the logistic regression, and although the explainable strategic behavior is largely captured by these RL models against opponent 1, they are much worse at modeling the behavioral variation seen against opponent 2. Similarly, when comparing the strategic and non-strategic monkeys against opponent 2 only, the non-strategic monkeys are better fit by the RL models against opponent 2, suggesting that they retain more reinforcement learning behavioral signatures, in line with the behavioral regressions shown in **Fig. 1** and **Fig. 2**. For these violin plots, as well as the logistic regressions shown in Fig. 1, we cut off the first 5000 sessions for monkeys C, E, and F to eliminate any effects caused by acclimation to the second opponent after optimizing their strategies against the first.

Figure 2: Behavioral Deviations from Reinforcement Learning

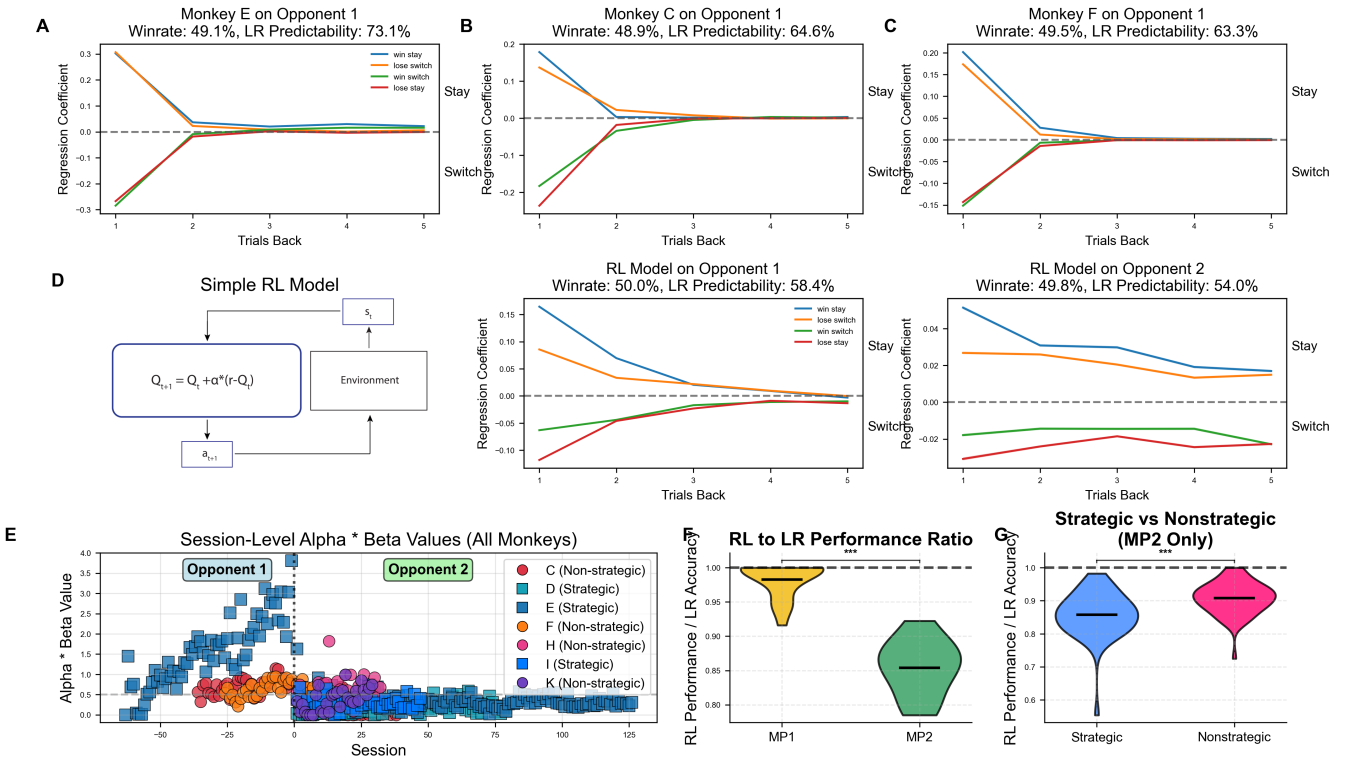


Fig. 2. Behavioral deviations from reinforcement learning. (A) Monkey E, the archetypal strategic monkey, playing against opponent 1. Solutions to opponent 1 include simple win-stay lose-switch strategies. (B,C) Monkeys C and F playing versus opponent 1. They exhibit similar behavioral dynamics against this opponent as monkey E. (D) Symmetric reinforcement learning model, previously used to analyze monkey behavior on this task. These models were fit against the last 10 sessions of monkey E against each opponent. Actions for these models were sampled non-deterministically, as only the forgetting model was capable of accurately playing against opponent 1, and no RL model could perform deterministically against opponent 2. (E) Product of α times inverse temperature β as a measure of stochasticity. Higher values are less stochastic. (F) Comparison of how well the RL model predicts monkey behavior compared to the logistic regressions. Monkeys C, E, and F deviate less from reinforcement learning behavior versus matching pennies opponent 1 than versus 2. (G) Strategic monkeys (D,E,I) show larger deviations from reinforcement learning than non-strategic monkeys (C,F,H,K).

Because these simpler models were unable to reproduce the complex behavior observed during this task, we trained 500 RNN models to play against opponent 2 and characterized them in relation to the monkeys and the RL models. These RNNs were based on the A2C actor-critic architecture used in the landmark paper Wang et al. (2018) to better mimic learning in the PFC than previous models, although we used the Q-value limit of the advantage function ($\lambda = 0$) and added additional constraints to the neurons in order to make them more realistic.

Every comparison was performed by computing the cosine similarity between each corresponding set of regression coefficients, e.g. comparing the repeat-win regression of the model's behavior to a monkey, then repeating the same for the other variables, and finally averaging across the four similarities for each pair considered. The representative RNN was selected by computing the behavioral regressions of 500 RNNs trained to play against opponent 2 and finding the model that maximized the average cosine similarity to the others. The monkey data was aggregated and fit in blocks of roughly 5000 trials, rounded to the nearest session, in order to generate similarity distributions. Statistical significance for the comparisons was calculated by taking the difference between samples in each distribution and bootstrapping a confidence interval. To better quantify the differences between the groups of monkeys and RNNs, we computed Cliff's Delta (δ) in addition to statistical significance. Visually, we can observe that the behavioral differences between groups is significant and that these effects are moderate.

When comparing the RNNs to the RL models, we observed that they produced very different behavior than either group of monkeys despite also learning to evaluate the best decisions using Q values. To our knowledge, no comparison like this been performed for a complex adversarial task. Other than in Wang et al. (2018), the majority of research has approached studying behavior by learning on experimental results to infer strategies from data as opposed to building models that behave and learn like we do, such as Dezfouli et al. (2019), Song et al. (2021), or Ji-An et al. (2025). The reason why these RNNs exhibit behavior that is very different from what we see in animal experiments is because they learn unconstrained solutions and have different inductive biases than the brain. For example, we can visibly observe that the representative model has large regression coefficients past the first two or three trials back, and these coefficients are not monotonically decreasing. This suggests that the model was able

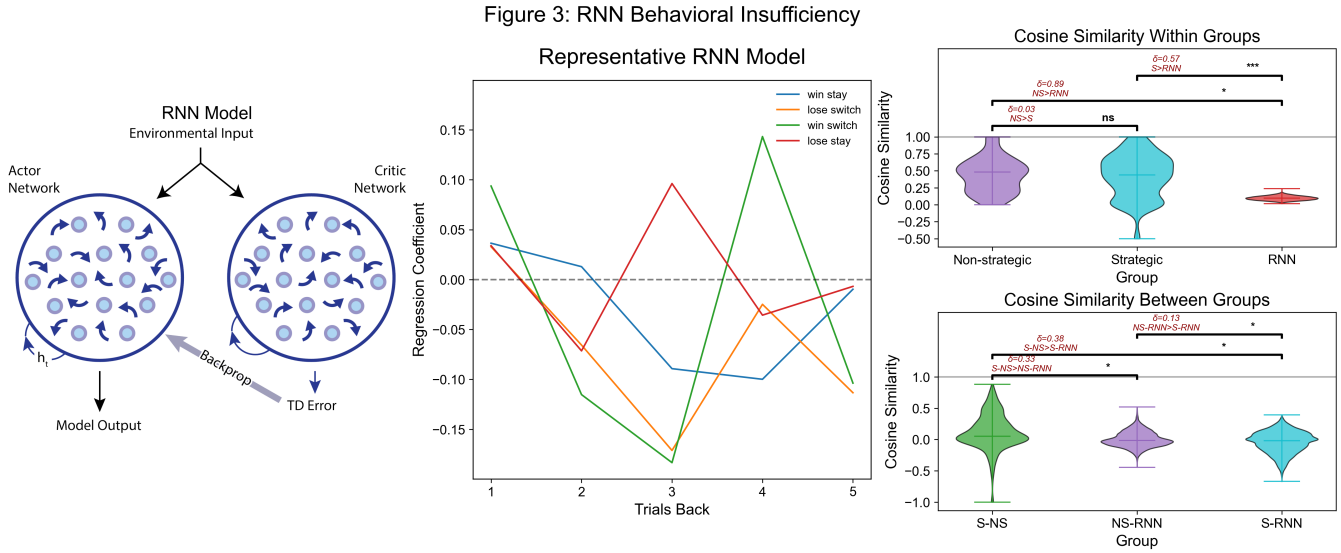


Fig. 3. RNN behavioral insufficiency. (A) Actor-critic RNN architecture. (B) Behavioral regression of the representative RNN. (C) Comparison of within and between group similarities. Differences between monkeys and RNNs are significant, with monkey behavior exhibiting more within-group similarity and a higher quantity of between group similarity. Cliff's δ is a nonparametric measure of the degree of non-overlap between two distributions. $\delta = \frac{\Sigma_{i,j}[x_i > x_j] - [x_i < x_j]}{mn}$.

to learn that the matching pennies opponent 2 only uses sequences of length 4 in its calculus, and so the model should use sequences longer than 4 when computing its next choice. This type of strategy, although a valid solution to the problem, is one that a brain would not be capable of discovering while utilizing evolved cognitive strategies and would likely require orders of magnitude more trials than each animal experienced in order to ascertain. The strategies a brain would come up with are much more sample efficient and observably appear to rely on composing simpler RL strategies instead.

Now that we had established how stark a difference there was between the RL models, the RNN models, and the monkeys, we attempted to create a model that had the flexibility of the RNN while exhibiting the behavioral tendencies of (and deviations from) reinforcement learning. The inspiration for this model came from the interplay between the PFC and the Basal Ganglia (BG) in the brain when making decisions. However, it was unclear how to combine these systems so that our PFC model (RNN) and our BG model (RL) were able to work together without completely overpowering each other. Our first attempts involved averaging the policy probabilities with different weights. Unfortunately, this ran into an issue that was common in multi-agent environments: when computing probabilities or actions in a coordinated multi-agent environment, each agent must have some knowledge, or way of inferring and learning that knowledge, of what the other agents are likely to do (Bansal et al., 2017; Al-Shedivat et al., 2017). This is due in part to the fact that learning under current methods becomes nearly impossible since we now not only have nonstationarity induced by the matching pennies environment, but also by the competing modules, meaning that the relationships between the systems must be robust.

To remedy these problems, there were two changes that needed to be made. The first was that the policies of the two modules needed to be combined in a way that mixed the gradients, which ended up being a linear combination of Q values instead of the polices. The second was to add an entropy regularizer to the critic to improve exploration as well as to allow the RNN module to better infer the behavior of the RL module (Lowe et al., 2017). The updated policy objective function is now

$$\mathcal{L}'_\pi(\phi) = \mathcal{L}_\pi(\phi) - \lambda H(\pi(\phi)) \quad \textbf{(1)}$$

where H is the entropy of the policy probability distribution. When approaching the decision layer shown in **Fig. 4**, we noticed that there was a stark asymmetry in the behavioral logistic regression coefficients across win and loss for a subset of the monkeys. Our first attempts to create these asymmetries were done by combining the Q values simply: $Q = (1-a)\cdot Q_{RL} + a \cdot Q_{RNN}$, equivalent to a weighed geometric mean of the probabilities: $p = p_{RL}^{1-a} \cdot p_{RNN}^{a}$. Although this did induce a bias similar to that seen in the change-lose regression coefficients for the strategic monkeys, it also modified the win regressors in the same way. To remedy this, we made the coefficients a function of reward, so the Q values were combined as

$$Q_{RLRNN} = (1-a(r))\cdot Q_{RL} + a(r)\cdot Q_{RNN} \quad \textbf{(2)}$$

, baking in an asymmetry between win and loss. In our testing, the bias induced by the RNN module was larger (and closer to the monkeys) than that by the RL module, so we opted to keep the simplest RL module possible for ease of characterization. This is similar to the idea behind a hidden markov model, where there are two states each with a model that get swapped between, except that we are manually setting the condition instead of fitting it. This type of hidden model has been successfully employed using the simpler RL models to model rat behavior, but not yet for RNNs (Venditto et al., 2024) as learning the latent state for a simultaneously learning RNN is not a well-posed problem. That being said, it should be very straightforward to make this layer a

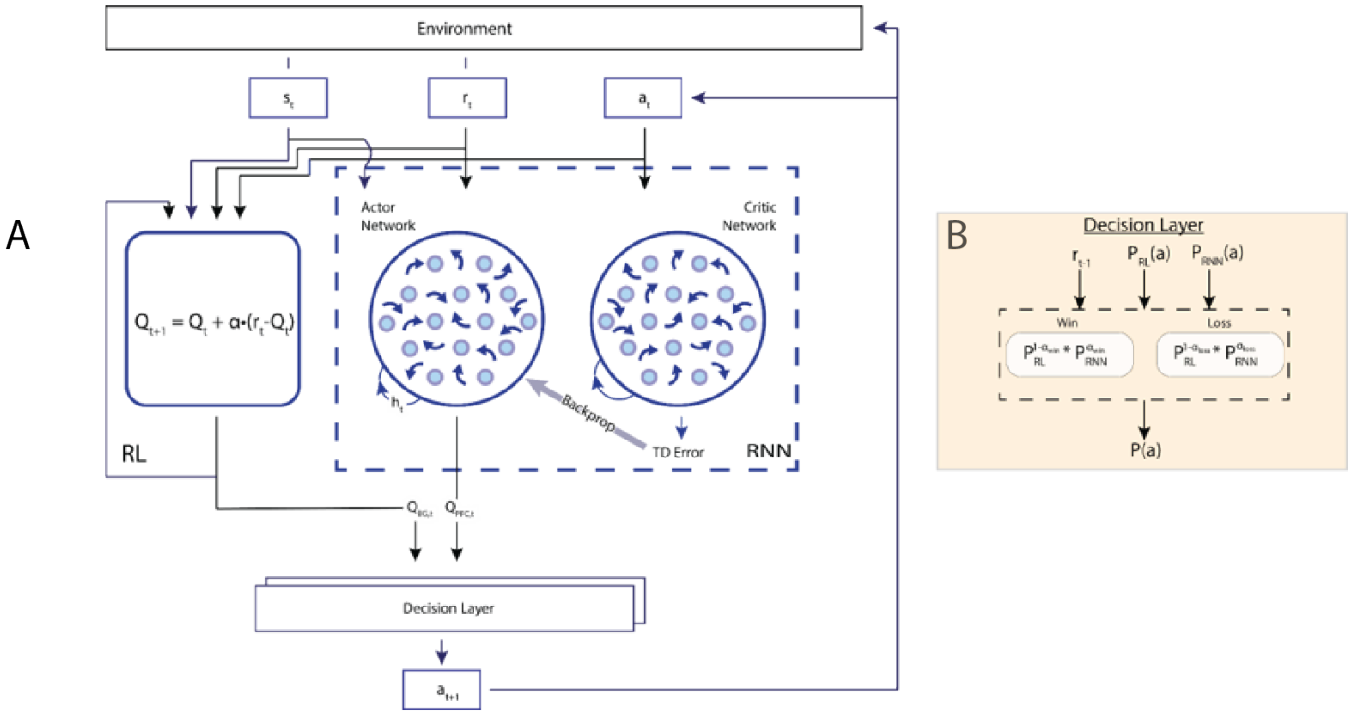


Fig. 4. Hybrid RLRNN model. (A) A simple symmetric RL model is placed in parallel to an RNN. These receive the same sets of inputs and work together to produce an output through the decision layer. (B) Breakdown of the decision layer, which takes the previous reward as well as the probabilities (or equivalently under a log transformation, the value functions). One set of weightings is used after a win, while another set is used after a loss.

constrained recurrent or feedforward neural network with a slower learning rate, approximating this.

Next, we characterized these hybrid models and compared them to RL models, RNNs, and the monkeys. Preliminary testing demonstrated that having strong RNN influence after a loss and weak RNN influence after a win most closely replicated the strategies observed in monkeys, but how to set the weighing parameters remained unclear. We performed a parameter sweep for all $(x, y) \in [0, 1]$ weighing pairs with a granularity of 0.1 by training 30 models of each parameter set, followed by 500 models trained in the neighborhood of the optimal parameters found. Shown in **Fig. 5** is the RL-RNN that was most similar ($s = 0.529$) on average for the strategic monkeys, with a weighing parameter (.9,.1) which corresponds to mostly RNN weighing after a loss and mostly RL after a win. For the non-strategic monkeys, we found that the best parameter set was (.9,0) with an average similarity of 0.581, but the models with $x \in [.1, 1]$ produced models with very similar behavior and corresponded to stronger RL influence. There is also some heterogeneity at the monkey level, but optimal parameters are mostly consistent by group (see **Fig. S7**). These parameter ranges imply that the differences between the groups of monkeys lie in how strongly the animals respond to wins and losses, exerting differing levels of cognitive control. The similarities were calculated by comparing the logistic regressions for each model with behavioral regressions at the monkey level. The performance comparison is the average likelihood under a greedy policy, which assuming that the monkey's choices and opponent's actions are the ground truths, this measures the probability that the model makes the same choice the monkeys do when presented with the same information, as well as whether the models have a learned strategy or are sampling randomness. Similarly, the mutual information comparison is the mutual information contained in action sequences of length 8 (4 model/monkey actions and 4 computer opponent actions interleaved) is shown in **Fig. S8**. One question we had was whether the unconstrained behaviors observed in the RNN models meant that these models were not learning strategies, but rather harnessing stochasticity through instability or noise. Through these performance and mutual information comparisons, not only can we confirm that these RNNs did successfully learn their own strategies, but also that these strategies are completely different than what we would expect from a RL framework.

Although the models shown do not decay as cleanly as the strategic or nonstrategic monkeys, they are able to reproduce the behavioral regressions and change-repeat asymmetry much better than prior cognitive models. The concomitant RNN module is able to frustrate the RL model, inducing a symmetry between both win and both loss actions in the RL module even at intermediate weightings, which is then broken by the decision layer in order to reproduce the strategic deviations similar to those observed during the experiment. Similarly, we can compare how well each type of model can predict the next decision in sequence for each group of monkeys to characterize how much it "thinks" like the animals. We found that the standard RNNs performed worse than chance, suggesting that they discover strategies that are very distinct than those found by the monkeys. Meanwhile, the hybrid models outperform RL models, and regressions fit to most of the data and predict out of sample using 10-

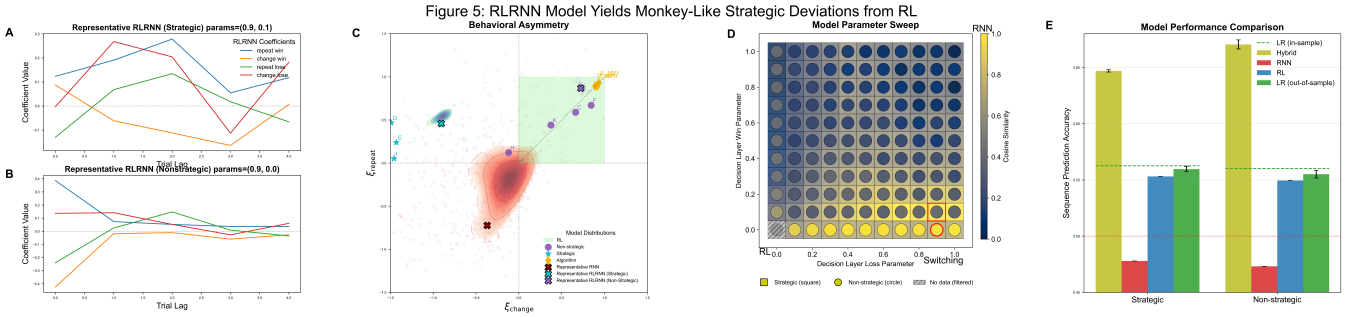


Fig. 5. (A) Logistic regression for the RLRNN with weighting parameters (.9,.1) that was most similar to the strategic monkeys. (B) Logistic regression of the RLRNN with weighting parameters (.8,0) that was most similar to the non-strategic monkeys. (C) Updated behavioral asymmetry plot, now includes the RL, RNN, as well as the (.8,0) and (.9,.1) hybrid RLRNN models. Here we show the distributions of the metrics for models with the aforementioned parameters. The cyan 'x' is the RLRNN model shown to the left for the strategic monkeys, while the purple 'x' is the RLRNN shown for the nonstrategic monkeys. The red 'x' is the representative RNN from Fig. 3A, which maximized similarity to all other RNNs. (D) Parameter grid sweep. Best parameter set was chosen by maximizing average cosine similarity between each model and the monkeys. For candidate best parameter sets that were statistically indistinguishable, we used the model performance metric as the tiebreaker. Monkey H was excluded from the non-strategic group in the sweep as it is somewhere between the rest of the non-strategic monkeys and the strategic monkeys. The cosine similarity was normalized for visualization. (E) Performance comparison between all the models with optimal parameters for each group, the RNNs, and the RL models at predicting monkey behavior for each group. RLRNN model outpredicts all other models, including logistic regressions fit to and predicting in-sample.

fold CV despite not having any prior exposure to animal data (and only requiring the setting of two hyperparameters). This strongly implies that the strategies inherently learned by these models are much closer to those learned by animals, and therefore these models are more appropriate for modeling and studying cognitive behavior than the current paradigm, especially since the pure RNN models perform worse than chance.

Now that we had a sense of what strategies the models learn, we decided to compare models with the same parameters playing against each of the opponents. We trained (.9,.1) RLRNNs on opponent 1 and compared them to the (.9,.1) models trained on opponent 2 for the RLRNN, since they produced behavior that was more similar to the strategic monkeys than most of the other models, and also cleanly separated the contributions of the two internal modules.

Notably, the RLRNN models with a simple Q-learning RL module were able to learn strategies that appeared nearly identical to the monkeys against opponent 1, while the RL model alone was insufficient to reproduce that behavior, and RNNs trained against the same task learned solutions that appeared more like the representative RNN from **Fig. 3**. The influence of the RL model, mediated by the weighing layer, seemed to be sufficient to constrain the solution space for the RNN to produce more naturalistic solutions.

As expected, the timescales for the models trained against opponent 1 were significantly lower than the timescales observed in models that played against opponent 2. Most notably, the reward timescale is larger in the first model even though the strategic timescales are longer in the model trained against opponent 2, which has led us to hypothesize that the neurons with longer timescales are recruited from reward encoding to behavior when going to the harder task. Similarly, the behavioral timescales for the opponent 2 models are significantly longer across the board than for the opponent 1 models, in line with expectations of the longer and more complex cognitive demands of the task.

Discussion

In this study, we addressed a fundamental limitation of the current tools used by researchers to model how brains make decisions. Although current models are widely used and prior claims have been made for the equivalency between ANN behavioral models and the behavior exhibited by animals, we demonstrated that models very similar to the ones used in Wang et al. (2018), albeit using weaker RNN units instead of LSTM cells, were unable to replicate naturalistic behavior for a complex problem despite success at simpler ones. This is due to the fact that RNN models do not have the same inductive biases baked into them as brains do, and so for complex problems they can learn strategies that are outside of our solution space. Another reason why may be because previous papers allow the convenience of model actions being sampled from the policy, but as brains are not random, our RNNs and hybrid RLRNN models were required to make decisions deterministically. This is related to the demonstrable fact that even that simpler models are able to solve many of the problems posed in the literature if decisions are sampled non-deterministically. As **Fig. S3** shows, even the simple RL model is able to perform well against opponent 2 when behaving non-deterministically, but in reality this is because it ends up sampling left and right randomly. We believe this is often overlooked when applying and studying models in cognitive tasks, as it can significantly change what the behavior looks like at the trial level even if it may look the same at the episode level. This may be fine for a bandit or reversal learning task, but in the real world or even on simpler non-stationary problems such as matching pennies, the dynamics are significantly affected. A third reason for these differences could be because the problems tested in prior literature are often less complex in their cognitive demands and therefore do not require significant deviations from simple reinforcement learning-like behavior, but they are difficult to structure using more traditional methods. These models are also comprised of a single module, which means they are less able to inform predictions that can be verified experimentally about the roles of the distinct systems in the brain.

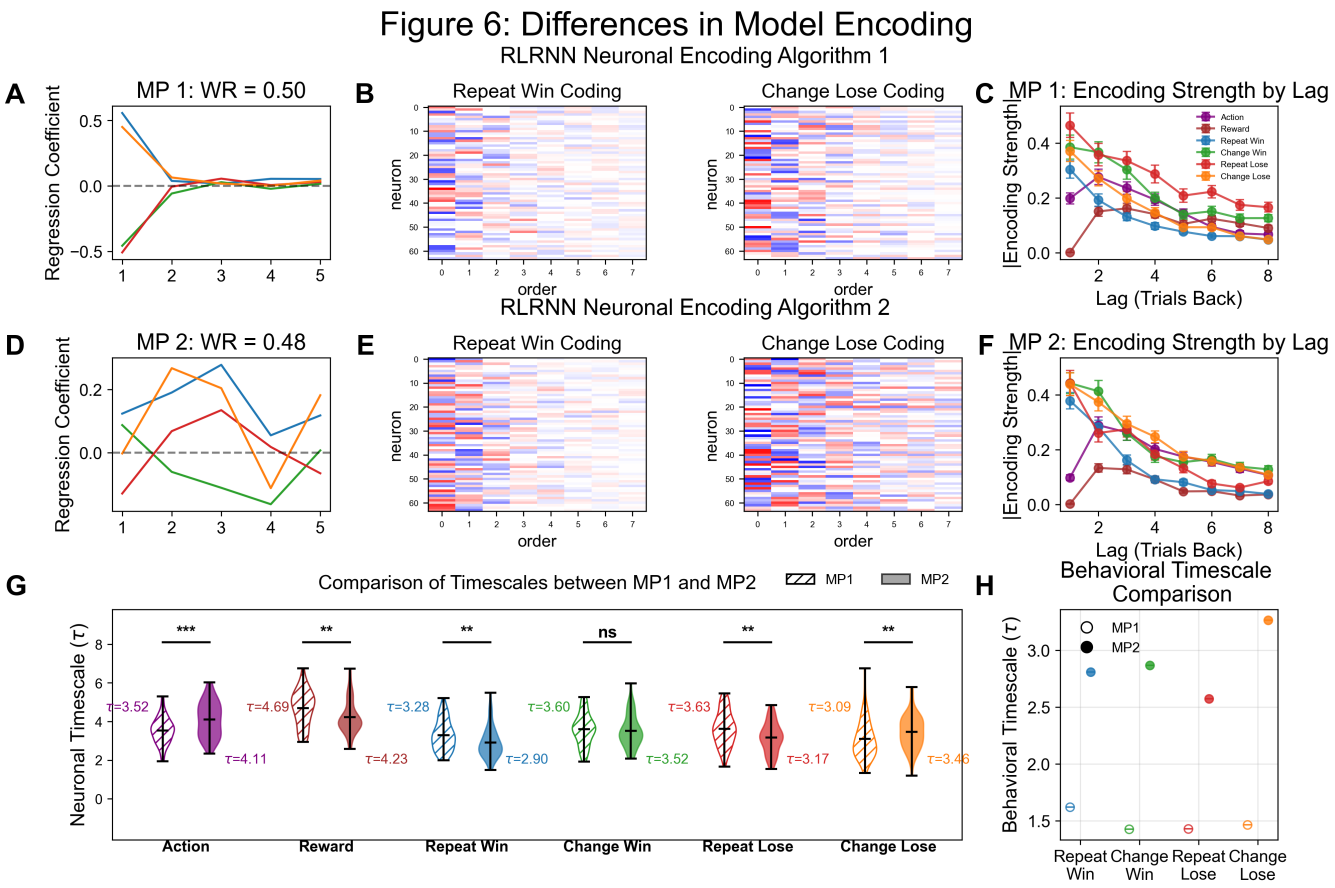


Fig. 6. Comparison between encoding and behavior of two models with the same weighting hyperparameters trained on each opponent. (A) Behavioral regression for an RLRNN trained against opponent 1 with weighting parameters (1,0). This model exhibits clear RL behavior, similar to that demonstrated by the monkeys. (B) Coding strength of prior win stay and lose switch for each neuron in the network. (C) Decay of the encoding strength for each relevant task variable. (D) RLRNN trained against opponent 2. (E) Coding strength of prior win stay and lose switch for each neuron. (F) Decay of the encoding strength for each relevant task variable in the second network. (G) Comparison of neuronal timescales for the two networks. (H) Comparison of the behavioral timescales for the networks shown.

In contrast, our RLRNN is able to find a deterministic solution to a complex problem and exhibit the same types of strategies and deviations from RL behavior seen instead of exploiting a flaw in the opponent, e.g. by learning to have a longer memory than it. Similarly, we can observe that in our model, more cognitive control is required after a loss than a win, which was shown to be the case in Seo and Lee (2009). Moreover, we observed that the models that produced strategic deviations had higher RNN weightings than the models more similar to the nonstrategic animals. Lastly, the fact that the model can predict the behavior of both groups of monkeys better than models fit specifically to the monkeys reinforces the idea that real strategies are also nonstationary and depend more on prior task conditions than expected.

Lastly, based on our analyses [1], we predict that in order to increase the timescales for higher order or more complex effects, such as strategy, the brain has to repurpose longer timescale reward-coding neurons. This can be seen in **Fig. 7**, where models with longer strategy timescales tended to have significantly shorter reward timescales. In studies such as Rigotti et al. (2013), it has been observed that single neuron activity in the PFC is tuned to mixtures of task variables, which is corroborated by our models as well. Therefore, if this effect is present in biological networks, we would expect successively more complex combinations of task variable coding to induce larger suppression of the timescales, negatively affecting working memory. This may potentially be related to cognitive overload as well, where the timescales become too short for the brain to make decisions due to the complexity of the problem presented.

Although we are partitioning animals into two groups and making broader claims about their behavior, cognitive behavior is very complex, especially so during this task, and it is probable that there are more granular categorizations of the strategies employed by the animals as implied by monkey H being an outlier for the behavioral asymmetry plots. However, we were limited both in number of monkeys, as well as how many of them were able to play opponent 1 prior to opponent 2. Therefore we had to make assumptions about some of the conditions which may constitute strategic versus non-strategic behavior or learning. A related limitation is how much data we had per monkey. Each monkey contained a different number of trials as well as different session sizes, which meant that the hypothesis of strategic behavior being related to how well the animals learn could not readily

[1] A forthcoming figure will show these analyses, but reward timescale seems negatively correlated with other timescales across all candidate models

be tested.

When deciding on the proper parametrization to describe each group, we performed a sweep where the weighting was held due to computational constraints. With more resources the parameters should be learnable, though it is a question of what the appropriate learning rate for the weighting layer should be compared to the rest of the network. Additionally, having a weighting layer that is memory dependent, i.e. an RNN, would be the most biologically and physically plausible version of this system. In such a scenario the model would dynamically decide how much to weight each submodule as a function of previous history. Similarly, the RL module can be replaced by a less powerful RNN with a rudimentary or shorter-term cost function. The analysis presented should be considered a starting point, although we believe it is a strong one.

Methods and Materials

A. Generalized logistic regression. To set a baseline for the performance of neural networks on the behavioral data, we began by fitting a standard logistic regression since in previous publications, it was the model that was most accurately able to predict future outcomes (Seo and Lee, 2009; Dezfouli et al., 2019; Song et al., 2021). The standard logistic regression model predicts a binary choice a_t for trial t and is given below. It contains a bias term μ_0, a term dependent on previous choices $a_{t-i} \in \{-1, 1\}$, and term dependent on previous reward $r_{t-i} \in \{0, 1\}$. In the case of a binary choice task such as matching pennies, opponent action can be substituted for either player action or reward as the basis is overcomplete.

$$\log\left(\frac{\pi_t(a_t = R)}{\pi_t(a_t = L)}\right) = \mu_0 + \sum_{i=1}^{n} \mu_i a_{t-i} \tag{3}$$

Although such a model is able to capture first order behavioral effects, it does not account for the conjunctions of choice and reward seen in deviations from RL-like behavior. To account for this, we first begin by defining a generalized logistic regression for D binary variables of order N_d and then use that to split the action-reward space into conjunctions of action $\times$ reward that are better able to capture these deviations.

We can recursively define a composite binary regressor X_n of order n by successively applying Kronecker products to the basic regressor x_τ at time τ in the following way:

$$X_n := X_{n-1} \otimes x_{t-n}, X_1 := x_{t-1}$$

This is the same as defining regressors as one hots and composing them using a direct sum:

$$X_n := (X_{n-1} = x_{t-n}) \oplus (X_{n-1} \neq x_{t-n})$$

or equivalently rewritten using indicator functions:

$$X_n := (\mathbb{1}_{x_{t-n}=x_{t-n+1}} + \mathbb{1}_{x_{t-n}\neq x_{t-n+1}})x_{t-n}$$

allowing us to build up the generalized logistic regression

$$\log\left(\frac{\pi_t(a_t = R)}{\pi_t(a_t = L)}\right) = \mu_0 + \sum_{i=1}^{M}\left(\bigotimes_{d=1}^{D} X_{N_d,i}\right)\vec{\mu_i} \tag{4}$$

with coefficient vector

$$\vec{\mu_i} := (\mu_{d1,1,i}, \ldots, \mu_{d1,N_1,i}, \mu_{d2,1,i}, \ldots)^T.$$

When we use action and rewards as our basis, this simplifies to the following:

$$\log\left(\frac{\pi_t(a_t = R)}{\pi_t(a_t = L)}\right) = \mu_0 + \sum_{i=1}^{M}(R_{N_r} \otimes A_{N_a})\vec{\mu_i} \tag{5}$$

yielding a total of $1 + 2^{N_a}2^{N_r}M$ regression coefficients. The standard regressor basis where only previous actions are used to predict future actions is then $(N_a, N_r) =$ (1,0), or equivalently (0,1), given by equation (2).

Going one step further to the (1,1) case, we have:

$$\log\left(\frac{\pi_t(a_t = R)}{\pi_t(a_t = L)}\right) = \mu_0 + \sum_{i=1}^{n}\left[\mu_{r,i}r_{t-i} + \mu_{\neg r,i}(1 - r_{t-i})\right]a_{t-i} \tag{6}$$

This is interpretable through the normal reinforcement learning framework. For example, if μ_r is positive and $\mu_{\neg r}$ is negative, this corresponds to fitting repeat-win and change-lose behavior. If we instead shift the reward term back by a timestep, we end up with win-stay and lose-switch

For higher order regressions it makes more sense to represent actions relative to prior actions. Although it should be apparent mathematically from the definition of the binary regressor, below is a visualization of how the state space is partitioned depending on how actions are represented:

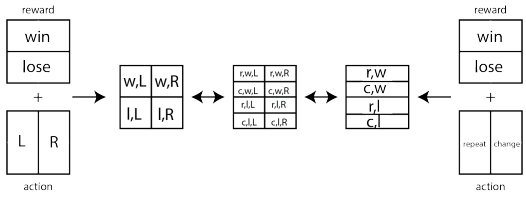


Fig. 7. Representation of how action $\times$ reward and strategy bases partition the state space equivalently.

Going yet another step further to the (2,2) case:

$$\begin{aligned}\log\left(\frac{\pi_t(a_t = R)}{\pi_t(a_t = L)}\right) = \mu_0 + \\ \sum_{i=1}^{n}\Big[&(\mathbb{1}_{ww}\mathbb{1}_r\mu_{ww,r,i} + \mathbb{1}_{wl}\mathbb{1}_r\mu_{wl,r,i} + \\ &\mathbb{1}_{ww}\mathbb{1}_c\mu_{ww,c,i} + \mathbb{1}_{wl}\mathbb{1}_c\mu_{wl,c,i}) \times r_{t-i} \\ &+ (\mathbb{1}_{ll}\mathbb{1}_c\mu_{ll,c,i} + \mathbb{1}_{lw}\mathbb{1}_c\mu_{lw,c,i} \\ &+ \mathbb{1}_{ll}\mathbb{1}_c\mu_{ll,r,i} + \mathbb{1}_{lw}\mathbb{1}_c\mu_{lw,r,i}) \\ &\times (1 - r_{t-i})\Big]a_{t-i}\end{aligned} \tag{7}$$

where ww $\rightarrow$ win-win, lw $\rightarrow$ lose-win, etc., and likewise r $\rightarrow$ repeat and c $\rightarrow$ change. For our analyses, we used the (2,1) model shown in **(Fig. S1)**. We believe this is the appropriate model for a few reasons. First, in the matching pennies cognitive task there is a symmetry between the two actions, so it makes more sense to view actions relative to each other instead of in isolation, termed the strategic basis. Second, a deterministic reinforcement learning model has a tendency to stay on a win and switch on a loss, and because choices made by the opponent are not independent the odds of winning twice in a row or losing twice in a row are less than 25% each, so conditioning only strategy ($N_a = 2, N_r \leq 1$) still allows these regressions to capture deviations from RL while keeping complexity to a minimum. Third, under the hypothesis that wins and losses require different levels of cognitive control, this model is the lowest requisite order since $N_r = 1$. Although the (2,2) should isolate the effects of our RLRNN modules, the additional model complexity makes comparisons for the RL, RNN, and Monkey more difficult to interpret as well as introduces overfitting concerns as the RL model does not need this additional complexity while real behavior potentially has longer dependencies than those captured by (2,2). We also know that since the matching pennies opponent 2 analyzes sequences of length four, it should be completely described by a regression of order (4,4), but because monkey performance is suboptimal, $(N_{m,a}, N_{m,r}) < (4, 4)$ and there are not enough trials within a session and behavior is nonstationary enough that we cannot fit such a complex regression.

$$\begin{aligned}\log\left(\frac{\pi_t(a_t = R)}{\pi_t(a_t = L)}\right) = \mu_0 + \\ \sum_{i=1}^{n}\Big[&(\mathbb{1}_{r_{t-i}}\mathbb{1}_{a_{t-i-1}=a_{t-i}}\mu_{r,repeat,i}) + \\ &(\mathbb{1}_{\neg r_{t-i}}\mathbb{1}_{a_{t-i-1}=a_{t-i}}\mu_{r,change,i}) + \\ &(\mathbb{1}_{r_{t-i}}\mathbb{1}_{a_{t-i-1}\neq a_{t-i}}\mu_{r,change,i}) + \\ &(\mathbb{1}_{\neg r_{t-i}}\mathbb{1}_{a_{t-i-1}\neq a_{t-i}}\mu_{r,repeat,i})\Big]a_{t-i}\end{aligned} \tag{8}$$

One complication when fitting the generalized logistic regression at $N_d > 1$ is caused by the combination of sparsity (due mutual exclusivity between regressors) and indirect colinearity (where if you know the values of the first $N - 1$ regressors, you can compute

the N^{th}). To account for these effects, we fit each time lag sequentially, i.e. first we fit the regressors at lag 1, then held the regressions coefficient constant and fit lag 2, etc.
Although we explicitly defined this for the case of binary variables, it is straightforward to extend it to cases where there are more than two choices or outcomes.
Lastly, when applied to reinforcement learning behavior, we expect logistic regression coefficients to decay linearly (and monotonically) towards zero, as that corresponds to the exponential decay baked into the reinforcement learning update equations. As we have demonstrated, real behavior can deviate from reinforcement learning when the cognitive demands of a task are sufficiently complex.

B. Maximum Likelihood Estimation and Reinforcement Learning Model Fitting. When fitting reinforcement learning models to behavioral data, the best way to do so is usually through maximum likelihood estimation (MLE), unless you have a strong prior in which case you can use maximum a posteriori (MAP) methods. MLE can be seen as a generalized version of least-squares and is more appropriate for this setting as it finds the model that is most likely to have generated the data we are observing. For RL models, we have probabilities for each choice given by the softmax of the Q values: $p(a|s) = \frac{e^{\beta Q(s,a)}}{\sum_{a'} e^{\beta Q(s,a')}}$, which, given a sequence of decisions, corresponds to optimizing the likelihood function $\mathcal{L}(\theta) = \prod_{t=1}^{T} p(a_t|s_t, r_{1,\ldots,t}, \theta)$ for model parameters θ. In practice, it is significantly easier to work with the negative log likelihood for stability.
For our analyses, we tried three RL models: The simple Q-learning model (**?**), an asymmetric Q-learning model where reward vs. no reward lead to different updates, and a forgetting Q-learning model (Barraclough et al., 2004) which is also asymmetric but treats decay separately. Although the forgetting model appeared to best replicate monkey behavior versus opponent 1, it fell short versus opponent 2. In our hybrid models, we used the simple Q-learning model as it is the simplest and also avoids a fine-tuning problem, where we introduce bias through the ratio of update coefficients Δ_+/Δ_- or α_+/α_-.
The update equations are as follows. For the simple and asymmetric models, we have:

$$Q_a \leftarrow (1-\alpha_r)\cdot Q_a + \alpha_r \cdot (r - \gamma \max Q_a) \tag{9}$$

where $\alpha_r = \alpha_{\neg r}$ in the simple case, and $\gamma = 0$ for all analyses. For the forgetting model, the update equation is

$$Q_a \leftarrow \alpha \cdot Q_a + \Delta_a(r) \tag{10}$$

where

$$\Delta_a(r) = \begin{cases} \Delta_+ & \text{if } a \text{ selected and rewarded} \\ \Delta_- & \text{if } a \text{ selected and not rewarded} \\ 0 & \text{if } a \text{ not selected} \end{cases}$$

C. Architecture and learning algorithm. For our work, we used the simplest possible RNN architecture in order to simplify characterization and training in addition to preventing the models from being too powerful. To handicap it further, we constrained the neurons within this RNN to be leaky and therefore require distributed computational expenditure in order to maintain memory, like a standard RNN. This is done by constraining the model to have no self connections (no autapses) which would be used to maintain the state in that unit as well as introducing an additional decay factor for the hidden state. For this decay, we specifically set the neuronal timescale to be equal to one trial.
Beginning with our general equation for an RNN cell, which is slightly different than the standard definition:

$$\tau \frac{dh}{dt} = W_{hh} f(h(t)) + W_{ih} x(t) \tag{11}$$

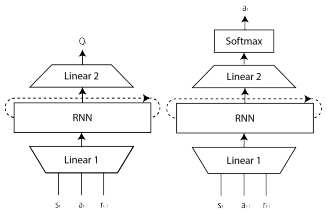
We can then discretize the equation and apply the 'no autapses' constraint by removing the diagonal:

$$\tau(h_t - h_{t-1}) = (W_{hh} - Diag(W_{hh}))f(h_{t-1}) + W_{ih}x(t) \tag{12}$$

which becomes our master equation for the dynamics of our network:

$$h_t = (1 - \frac{1}{\tau})h_{t-1} + (W_{hh} - Diag(W_{hh}))f(h_{t-1}) + W_{ih}x_t \tag{13}$$

and for which we set τ = 1. These units are then put together in an actor-critic architecture with the following structure:



Actor-Critic architecture for our RNNs. Contains a single recurrent layer as well as a layer projecting into and out of the recurrent layer.

Where we deterministically select the action for the RNN, i.e. $a_t = \max(\pi_t)$. Although the softmax layer is unnecessary for our deterministic action selection, we included it so that we could perform future analyses such as how confident the networks are in their actions.

D. Layer initialization. Network initialization is an important preliminary step. The idea is to start with a network initialization that is as close as possible to the timescale requirements of the problem of interest while also having at least one real component of weight matrix eigenvalues be close to (but usually bounded from below by) zero. Much like how the brain performs computations at multiple timescales, since we are trying to model brain-like behavior as realistically as possible, we want a wide range of initial timescales in order to improve learning.
For our input layer and output layers we used the standard He initialization, while for our recurrent layer we developed an algorithm to generate random matrices that satisfy the spectral constraints as well as those of our master equation while learning quicker than $W_h = \frac{1}{\sqrt{n}}\mathcal{N}(0,1)$ and orthogonal initializations.

Algorithm 1 Recurrent Layer Initialization : Random Matrix Initialization with Constraints

Require: Hidden size n, parameter $\tau > 0$, gain $g > 0$, optional ϵ
Ensure: Matrix $W \in \mathbb{R}^{n\times n}$ with controlled spectral radius
$enough\_pos \leftarrow$ False
$not\_too\_large \leftarrow$ False
while not $enough\_pos$ **and not** not_too_large **do**
  $W \leftarrow$ Normal$(0, g)$ of size $n \times n$
  if ϵ is not provided **then**
    Compute eigenvalues $\lambda = \text{eigvals}(W)$
    $l \leftarrow \max|\lambda|$
    $\epsilon \leftarrow \frac{\sqrt{l}}{\sqrt{n}} \div 2$
  end if
  $W \leftarrow \frac{W}{\sqrt{n}} \cdot \left(1 + \frac{1}{\tau} - \epsilon\right)$
  $W \leftarrow W - \text{diag}(W) - \frac{I}{\tau}$
  Compute eigenvalues $\lambda = \text{eigvals}(W)$
  $reals \leftarrow \text{Re}(\lambda)$
  $enough\_pos \leftarrow \sum(reals \geq 0)$
  $not\_too\_large \leftarrow \max(reals) < 1$
end while
return Parameter(W)

This initialization was inspired by He initialization, but also bakes in our constraints, namely the leaky RNN and no autapses, while maintaining a complete eigenvalue spectrum.
The other set of weight constraints are for the decision layer of our hybrid network, which will be detailed in the following section. The normalization used clamps the weights to $\vec{x} \rightarrow \frac{\vec{x}}{||\vec{x}||_p}$, which for our purposes p=1.

E. Objective Functions. The purpose of structuring RNNs in an actor-critic network is to have at least one (critic) network computing the value (Q) function of the actions possible from current state, while an actor infers the Q values and chooses an action. In practice, this type of model is used to improve learning through stabilization by computing a baseline, which you can then use to offload accurate estimation and then learn from it. There is a neurological basis for using this type of model as well, which has been explored in models of the basal ganglia and the prefrontal cortex (Barto, 1995; Joel et al., 2002; Silvetti et al., 2014; Liakoni et al., 2022). Similarly to Mnih et al. (2016); Wang et al. (2018) we used Advantage Actor-Critic value functions but with additional regularization terms meant to both distribute activity among the neurons and promote separation of task variables through sparsity.

$$\begin{aligned}
\mathcal{L} &= \mathcal{L}_\pi + \mathcal{L}_Q + \mathcal{L}_H + \mathcal{L}_{reg} \\
\nabla\mathcal{L}_\pi &= \mathbb{E}[\nabla_\phi \log\pi(a_t|s_t)\cdot(Q_\theta(s_t,a_t,\theta) - V(s_t)] \\
\nabla\mathcal{L}_Q &= \frac{1}{2}\nabla_\theta\mathbb{E}[(y - Q_\theta(s_t,a_t,\theta))^2] \\
\nabla\mathcal{L}_H &= -\beta_H\nabla_\phi H(\pi(a_t|s_t)) \\
\mathcal{L}_{reg} &= \lambda_1||W_{hh}||_1 + \lambda_2||(1-\frac{1}{\tau})h + W_{hh}h + W_{ih}x||_2 \\
y_t(r,s',d) &= r + \gamma(1-d)V_{\phi,t+1}(s,a), \quad \tilde{a}' \sim \phi_\theta(\cdot|s')
\end{aligned} \tag{14}$$

where a_t, s_t, and y_t are the actions, states, and n-step bootstrapped targets for the Q functions respectively.

F. Training and testing Models were trained by playing against the . No parameter optimization besides the sweeps was performed,

We utilized early stopping contingent on the average reward in the task. We fit . Models that had a sub 45% winrate (50% being optimal but not realistic) were excluded from all analyses.
All RLRNN models were trained with the following fixed parameters:
Where we chose the hyperparameters as follows:

- $hidden\_dim \leftarrow 64$ because 32 hidden units did not seem sufficient in early testing.
- $max\_steps \leftarrow 200$ was a compromise between noise and time, since longer episode lengths learned more slowly because the matching pennies opponent had more choice sequences to compute as well as the value functions being harder to compute.
- $Qalpha \leftarrow 0.2$ because the matching pennies opponents have a memory of four trials back, and this α corresponds to a timescale of 5 trials. A comparison of the three models fit on monkey E data for both opponents and then playing against that opponent is shown in the first supplemental figure.
- $q\_init \leftarrow 0.5$ sets the initial Q values for both choices to be 0.5.
- $gamma \leftarrow 0.75$ is the reward discounting for the model, set to this value since the timescale is effectively $\frac{1}{1-\gamma} \rightarrow 4$

G. Model performance metrics. To compare behavior between model and animal behavior, we used cosine similarity between the regression coefficients for the monkeys and models.

$$\vec{C}_{\text{model}} = [\vec{c}_{\text{model, RW}}, \vec{c}_{\text{model, CW}}, \vec{c}_{\text{model, RL}}, \vec{c}_{\text{model, CL}}]$$

$$\vec{C}_{\text{monkey}} = [\vec{c}_{\text{monkey, RW}}, \vec{c}_{\text{monkey, CW}}, \vec{c}_{\text{monkey, RL}}, \vec{c}_{\text{monkey, CL}}]$$

$$S_k = \frac{\vec{c}_{\text{model, k}} \cdot \vec{c}_{\text{monkey, k}}}{\|\vec{c}_{\text{model, k}}\| \|\vec{c}_{\text{monkey, k}}\|}$$

$$S = \frac{1}{4}\left(S_{\text{RW}} + S_{\text{CW}} + S_{\text{RL}} + S_{\text{CL}}\right) \tag{15}$$

In order to confirm that our models were in fact learning strategies instead of sampling randomly, we used mutual information. With a correction for finite sample bias, the average mutual information for a long sequence of decisions is:

$$I = -\sum_{i=1}^{N^L}\sum_{j=1}^{N} \hat{p}_{ij} \log_2 \frac{\hat{p}_{ij}}{\hat{p}_i \hat{p}_j} - \frac{(N^L-1)(N-1)}{\ln(N+1)\cdot L} \tag{16}$$

where p_i is the probability for the i-th outcome in the input even sequence of length N^L, p_j is the probability of the j-th outcome in the output event, and p_{ij} is the joitnt probability for the i-th input event and j-th output event, where there are N possible outcomes at each time and for sequence length L.
Similarly, our performance metric, which is how well a regression or model is able to predict the next action in the sequence of monkey decisions, is equivalent to the likelihood under a greedy policy (since we are assuming determinism) divided by the number of trials predicted.

$$\bar{\mathcal{L}} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{a_i}(\max(\pi(\theta, a_{i-1}, s_{i-1}))) \tag{17}$$

H. Matching pennies task and opponents. There are two existing databases containing matching pennies task data, which we combined for our analyses. The first dataset contains behavioral data for monkeys C, E, and F against the two variants of the matching pennies opponents, which are both detailed at the end of this section. The second dataset contains behavioral data as well as neuronal spike measurements for monkeys C, D, E, H, I, and K. For the former, what was stored was a session number, monkey choice, opponent choice, and whether the monkey was rewarded. In this dataset, the monkeys were unlabeled, but fortunately there was some overlap between the two datasets that allowed us to identify monkeys C and E through monkey choice sequence matching comparisons, and therefore we knew that the third monkey must be was F. However, the disjoint nature of the monkey C and E data meant that we needed to stitch both datasets together somehow. As the sessions were numbered differently between the two datasets, we again performed choice sequence matching, this time in order to piece together a timeline for all the opponent 2 sessions. For the most part these matched nicely; sessions not present in one slotted between two sessions in another, but in a few cases we had to use our best judgment on how stitch together the datasets temporally. The temporal ordering of the data matters because the monkey behavior is inherently non-stationary, as monkeys play the game successively their strategies seem to shift due to learning.
We have two variants of the matching pennies opponents. The first opponent only looks at patterns in the sequences of choices made by the player, while the second opponent looks at both patterns in the sequences of choices as well as how those sequences are affected by rewards received. Only monkeys C, E, and F played against opponent 1, but all monkeys (C, D, E, F, H, I, K) played versus opponent 2.

Opponent 1 looks at the sequences of actions previously chosen by the player and uses them in order to predict the next most likely choice. Because this only looks at biases in the actions, a simple repeat-win change-lose strategy is sufficient for optimal play (assuming no side bias exists).
Opponent 2 is very similar to Opponent 1 except it also looks at sequences of choices conditioned on reward. Unlike Opponent 1, against which RL strategies can perform well, Opponent 2 will, for example, realize that a stay likely follows a win and punish the player. If this opponent cannot discern a pattern with statistical significance, it will select a choice randomly. This happens in roughly 40% of the trials across the seven monkeys.

Algorithm 2 MP Opponents

```
for depth d = 1, ..., D do
    Compute bias on recent trials where choice history was the same
    for t = d+1, ..., T do
        Search for trials with the same sequence of decisions of depth d
        Compute most likely next choice based on previous sequences
        Run binomial tests to determine whether choice biases are significant
        if Bias detected is significant and larger than previously stored bias then
            Save probabilities
        end if
    end for
    if Opponent 2 then
        for depth d = 1, ..., D do
            for t = d+1, ..., T do
                Search for trials with the same sequence of decisions
                and sequence of rewards
                Compute most likely choice based on previous choice
                and reward sequences
                Test whether choice × reward biases are significant
                if Bias detected is significant and larger than previously stored bias
then
                    Save probabilities
                end if
            end for
        end for

        Use saved biases to modify probabilities and sample next choice
```

I. Neuronal and behavioral timescales. To compute the neural and behavioral timescales, we calculated the center of mass of the encoding for each task variable, taking advantage of the relationship.

$$\tau_{neuronal} = \frac{\int_0^\infty x e^{x/\tau}\, dx}{\int_0^\infty e^{x/\tau}\, dx} \tag{18}$$

For the behavior, since a linear decay in the behavioral regressions corresponds to an exponential decay for RL model, we used

$$\tau_{behavior} \approx \frac{\int_0^T \lambda(x)\, x\, dx}{\int_0^T \lambda(x)\, dx} \tag{19}$$

instead. This is approximate, as the behavioral regressions do not decay as cleanly as the neuronal regressions.

J. Code and data availability Code has been made available in a github repo: https://github.com/fberl/Preprint_Repository.
The data used is not presently publicly available. Contact Daeyeol Lee if interested.

References

1. Valerio Mante, David Sussillo, Krishna V. Shenoy, and William T. Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503(7474): 78–84, 2013. ISSN 0028-0836. doi: 10.1038/nature12742.
2. Aniruddh R. Galgali, Maneesh Sahani, and Valerio Mante. Residual dynamics resolves recurrent contributions to neural computation. *Nature Neuroscience*, 26(2):326–338, 2023. ISSN 1097-6256. doi: 10.1038/s41593-022-01230-2.
3. Maria Eckstein, Christopher Summerfield, Nathaniel Daw, and Kevin J Miller. Hybrid neural-cognitive models reveal how memory shapes human reward learning. 2024. doi: 10.31234/osf.io/u9ks4.
4. Li Ji-An, Marcus K. Benna, and Marcelo G. Mattar. Discovering cognitive strategies with tiny recurrent neural networks. *Nature*, pages 1–9, 2025. ISSN 0028-0836. doi: 10.1038/s41586-025-09142-4.
5. David Sussillo and Omri Barak. Opening the black box: Low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural Computation*, 25(3):626–649, 2013. ISSN 0899-7667. doi: 10.1162/neco_a_00409.

Table 1. RLRNN Model Configuration Parameters

"hidden_dim": 64	"l1": 0.01
"max_steps": 200	"l2": 0.05
"batch_size": 8	"RL": true
"leaky_tau": 1	"leaky_q": true
"leaky_policy": true	"Qalpha": 0.2
"gamma": 0.75	"lambd": 0
"q_init": 0.5	"use_linear2": false
"scaletype": "adaptive"	"norm_p": 1

6. Daeyeol Lee, Michelle L. Conroy, Benjamin P. McGreevy, and Dominic J. Barraclough. Reinforcement learning and decision making in monkeys during a competitive game. *Cognitive Brain Research*, 22(1):45–58, 2004. ISSN 0926-6410. doi: 10.1016/j.cogbrainres.2004.07.007.
7. Dominic J. Barraclough, Michelle L. Conroy, and Daeyeol Lee. Prefrontal cortex and decision making in a mixed-strategy game. *Nature Neuroscience*, 7(4):404–410, 2004. ISSN 1097-6256. doi: 10.1038/nn1209.
8. Christopher H. Donahue, Hyojung Seo, and Daeyeol Lee. Cortical signals for rewarded actions and strategic exploration. *Neuron*, 80(1):223–234, 2013. ISSN 0896-6273. doi: 10.1016/j.neuron.2013.07.040.
9. Jane X. Wang, Zeb Kurth-Nelson, Dharshan Kumaran, Dhruva Tirumala, Hubert Soyer, Joel Z. Leibo, Demis Hassabis, and Matthew Botvinick. Prefrontal cortex as a meta-reinforcement learning system. *Nature Neuroscience*, 21(6):860–868, 2018. ISSN 1097-6256. doi: 10.1038/s41593-018-0147-8.
10. Amir Dezfouli, Kristi Griffiths, Fabio Ramos, Peter Dayan, and Bernard W. Balleine. Models that learn how humans learn: The case of decision-making and its disorders. *PLOS Computational Biology*, 15(6):e1006903, 2019. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1006903.
11. Mingyu Song, Yael Niv, and Mingbo Cai. Using recurrent neural networks to understand human reward learning. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 43(43), 2021.
12. Trapit Bansal, Jakub Pachocki, Szymon Sidor, Ilya Sutskever, and Igor Mordatch. Emergent complexity via multi-agent competition. *arXiv*, 2017. doi: 10.48550/arxiv.1710.03748.
13. Maruan Al-Shedivat, Trapit Bansal, Yuri Burda, Ilya Sutskever, Igor Mordatch, and Pieter Abbeel. Continuous adaptation via meta-learning in nonstationary and competitive environments. *arXiv*, 2017. doi: 10.48550/arxiv.1710.03641.
14. Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *arXiv*, 2017. doi: 10.48550/arxiv.1706.02275.
15. S. Venditto, K. Miller, C. Brody, and N. Daw. Dynamic reinforcement learning reveals time-dependent shifts in strategy during reward learning. *eLife Preprint*, 2024.
16. Hyojung Seo and Daeyeol Lee. Behavioral and neural changes after gains and losses of conditioned reinforcers. *The Journal of Neuroscience*, 29(11):3627–3641, 2009. ISSN 0270-6474. doi: 10.1523/jneurosci.4726-08.2009.
17. Mattia Rigotti, Omri Barak, Melissa R. Warden, Xiao-Jing Wang, Nathaniel D. Daw, Earl K. Miller, and Stefano Fusi. The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497(7451):585–590, 2013. ISSN 0028-0836. doi: 10.1038/nature12160.
18. Barto. Adaptive critics and the basal ganglia. 1995.
19. Daphna Joel, Yael Niv, and Eytan Ruppin. Actor–critic models of the basal ganglia: new anatomical and computational perspectives. *Neural Networks*, 15(4-6):535–547, 2002. ISSN 0893-6080. doi: 10.1016/s0893-6080(02)00047-3.
20. Massimo Silvetti, William Alexander, Tom Verguts, and Joshua W. Brown. From conflict management to reward-based decision making: Actors and critics in primate medial frontal cortex. *Neuroscience & Biobehavioral Reviews*, 46:44–57, 2014. ISSN 0149-7634. doi: 10.1016/j.neubiorev.2013.11.003.
21. Vasiliki Liakoni, Marco P. Lehmann, Alireza Modirshanechi, Johanni Brea, Antoine Lutti, Wulfram Gerstner, and Kerstin Preuschoff. Brain signals of a surprise-actor-critic model: Evidence for multiple learning modules in human decision making. *NeuroImage*, 246:118780, 2022. ISSN 1053-8119. doi: 10.1016/j.neuroimage.2021.118780.
22. Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. *arXiv*, 2016. doi: 10.48550/arxiv.1602.01783.
23. Guangyu Robert Yang, Madhura R. Joglekar, H. Francis Song, William T. Newsome, and Xiao-Jing Wang. Task representations in neural networks trained to perform many cognitive tasks. *Nature Neuroscience*, 22(2):297–306, 2019. ISSN 1097-6256. doi: 10.1038/s41593-018-0310-2.
24. Natasha Sigala, Makoto Kusunoki, Ian Nimmo-Smith, David Gaffan, and John Duncan. Hierarchical coding for sequential task events in the monkey prefrontal cortex. *Proceedings of the National Academy of Sciences*, 105(33):11969–11974, 2008. ISSN 0027-8424. doi: 10.1073/pnas.0802569105.
25. Jake P. Stroud, Mason A. Porter, Guillaume Hennequin, and Tim P. Vogels. Motor primitives in space and time via targeted gain modulation in cortical networks. *Nature Neuroscience*, 21(12):1774–1783, 2018. ISSN 1097-6256. doi: 10.1038/s41593-018-0276-0.
26. Laurence T. Hunt, W. M. Nishantha Malalasekera, Archy O. de Berker, Bruno Miranda, Simon F. Farmer, Timothy E. J. Behrens, and Steven W. Kennerley. Triple dissociation of attention and decision computations across prefrontal cortex. *Nature Neuroscience*, 21(10):1471–1481, 2018. ISSN 1097-6256. doi: 10.1038/s41593-018-0239-5.
27. Guangyu Robert Yang and Xiao-Jing Wang. Artificial neural networks for neuroscientists: A primer. *Neuron*, 107(6):1048–1070, 2020. ISSN 0896-6273. doi: 10.1016/j.neuron.2020.09.005.
28. Mark G. Stokes, Makoto Kusunoki, Natasha Sigala, Hamed Nili, David Gaffan, and John Duncan. Dynamic coding for cognitive control in prefrontal cortex. *Neuron*, 78(2):364–375, 2013. ISSN 0896-6273. doi: 10.1016/j.neuron.2013.01.039.
29. Makoto Ito and Kenji Doya. Validation of decision-making models and analysis of decision variables in the rat basal ganglia. *The Journal of Neuroscience*, 29(31):9861–9874, 2009. ISSN 0270-6474. doi: 10.1523/jneurosci.6157-08.2009.
30. Soyoun Kim, Jaewon Hwang, Hyojung Seo, and Daeyeol Lee. Valuation of uncertain and delayed rewards in primate prefrontal cortex. *Neural Networks*, 22(3):294–304, 2009. ISSN 0893-6080. doi: 10.1016/j.neunet.2009.03.010.
31. Alireza Soltani, John D Murray, Hyojung Seo, and Daeyeol Lee. Timescales of cognition in the brain. *Current Opinion in Behavioral Sciences*, 41:30–37, 2021. ISSN 2352-1546. doi: 10.1016/j.cobeha.2021.03.003.
32. Mehran Spitmaan, Hyojung Seo, Daeyeol Lee, and Alireza Soltani. Multiple timescales of neural dynamics and integration of task-relevant signals across cortex. *Proceedings of the National Academy of Sciences*, 117(36):22522, 2020. doi: 10.1073/pnas.2005993117.
33. A. Singer. From graph to manifold laplacian: The convergence rate. *Applied and Computational Harmonic Analysis*, 21(1):128–134, 2006. ISSN 10635203. doi: 10.1016/j.acha.2006.03.004.
34. Michael L. Platt and Paul W. Glimcher. Neural correlates of decision variables in parietal cortex. *Nature*, 400(6741):233–238, 1999. ISSN 1476-4687. doi: 10.1038/22268.
35. Hyojung Seo, Dominic Barraclough, and Daeyeol Lee. Lateral intraparietal cortex and reinforcement learning during a mixed-strategy game undefined journal of neuroscience, 2009.
36. Hyojung Seo, Dominic J Barraclough, and Daeyeol Lee. Dynamic signals related to choices and outcomes in the dorsolateral prefrontal cortex undefined cerebral cortex undefined oxford academic, 2007.
37. Daeyeol Lee and Hyojung Seo. Neural basis of strategic decision making. *Trends in Neurosciences*, 39(1):40–48, 2016. ISSN 1878-108X. doi: 10.1016/j.tins.2015.11.002.
38. Michael E. Rule and Timothy O'Leary. Self-healing codes: How stable neural populations can track continually reconfiguring neural representations. *Proceedings of the National Academy of Sciences*, 119(7), 2022. ISSN 0027-8424. doi: 10.1073/pnas.2106692119.
39. T. M. Bartol, B. R. Land, E. E. Salpeter, and M. M. Salpeter. Monte carlo simulation of miniature endplate current generation in the vertebrate neuromuscular junction. *Biophysical Journal*, 59(6):1290–1307, 1991. ISSN 0006-3495. doi: 10.1016/s0006-3495(91)82344-x.
40. Wei Ji Ma, Jeffrey M Beck, Peter E Latham, and Alexandre Pouget. Bayesian inference with probabilistic population codes. *Nature Neuroscience*, 9(11):1432–1438, 2006. ISSN 1097-6256. doi: 10.1038/nn1790.
41. Tim Vogels, Robert Froemke, Nicolas Doyon, Matthieu Gilson, Julie Haas, Robert Liu, Arianna Maffei, Paul Miller, Corette Wierenga, Melanie Woodin, Friedemann Zenke, and Henning Sprekeler. Inhibitory synaptic plasticity: spike timing-dependence and putative network function. *Frontiers in Neural Circuits*, 7, 2013. ISSN 1662-5110.
42. Peter Dayan and L. F. Abbott. *Theoretical neuroscience: computational and mathematical modeling of neural systems*. Computational neuroscience. Massachusetts Institute of Technology Press, Cambridge, Mass, 2001. ISBN 978-0-262-04199-7.
43. William Lotter, Gabriel Kreiman, and David Cox. A neural network trained for prediction mimics diverse features of biological neurons and perception. *Nature Machine Intelligence*, 2(4):210–219, 2020. ISSN 2522-5839. doi: 10.1038/s42256-020-0170-9.
44. Ruben S van Bergen, Wei Ji Ma, Michael S Pratte, and Janneke F M Jehee. Sensory uncertainty decoded from visual cortex predicts behavior. *Nature Neuroscience*, 18(12):1728–1730, 2015. ISSN 1097-6256. doi: 10.1038/nn.4150.
45. Mehrdad Jazayeri and Srdjan Ostojic. Interpreting neural computations by examining intrinsic and embedding dimensionality of neural activity. *Current Opinion in Neurobiology*, 70:113–120, 2021. ISSN 0959-4388. doi: 10.1016/j.conb.2021.08.002.
46. Alireza Soltani, Daeyeol Lee, and Xiao-Jing Wang. Neural mechanism for stochastic behavior during a competitive game. *Neural networks : the official journal of the International Neural Network Society*, 19(8):1075–1090, 2006. ISSN 0893-6080. doi: 10.1016/j.neunet.2006.05.044.
47. A. Destexhe, Z. F. Mainen, and T. J. Sejnowski. An efficient method for computing synaptic conductances based on a kinetic model of receptor binding. *Neural Computation*, 6(1):14–18, 1994. ISSN 0899-7667. doi: 10.1162/neco.1994.6.1.14.
48. Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3-4):279–292, 1992. ISSN 0885-6125. doi: 10.1007/bf00992698.
49. Matthew T Kaufman, Mark M Churchland, Stephen I Ryu, and Krishna V Shenoy. Cortical activity in the null space: permitting preparation without movement. *Nature Neuroscience*, 17(3):440–448, 2014. ISSN 1097-6256. doi: 10.1038/nn.3643.
50. David Sussillo, Mark M Churchland, Matthew T Kaufman, and Krishna V Shenoy. A neural network that finds a naturalistic solution for the production of muscle activity. *Nature Neuroscience*, 18(7):1025–1033, 2015. ISSN 1097-6256. doi: 10.1038/nn.4042.
51. Mark M. Churchland, John P. Cunningham, Matthew T. Kaufman, Justin D. Foster, Paul

Nuyujukian, Stephen I. Ryu, and Krishna V. Shenoy. Neural population dynamics during reaching. *Nature*, 487(7405):51–56, 2012. ISSN 0028-0836. doi: 10.1038/nature11129.
52. Xiaohan Zhang, Mohamad Altrabulsi, Wenqi Xu, Ralf Wimmer, Michael M Halassa, and Zhe S Chen. Multiplicative couplings facilitate rapid learning and information gating in recurrent neural networks. 2025. doi: 10.1101/2025.07.11.663676.
53. Dmitry Kobak, Wieland Brendel, Christos Constantinidis, Claudia E Feierstein, Adam Kepecs, Zachary F Mainen, Xue-Lian Qi, Ranulfo Romo, Naoshige Uchida, and Christian K Machens. Demixed principal component analysis of neural population data. *eLife*, 5:e10989, 2016. doi: 10.7554/elife.10989.
54. Anton Maximilian Schäfer and Hans Georg Zimmermann. Artificial neural networks – ICANN 2006, 16th international conference, athens, greece, september 10-14, 2006. proceedings, part i. *Lecture Notes in Computer Science*, pages 632–640, 2006. ISSN 0302-9743. doi: 10.1007/11840817_66.
55. Peiran Gao, Eric Trautmann, Byron Yu, Gopal Santhanam, Stephen Ryu, Krishna Shenoy, and Surya Ganguli. A theory of multineuronal dimensionality, dynamics and measurement. *bioRxiv*, page 214262, 2017. doi: 10.1101/214262.
56. Kentaro Ohno and Atsutoshi Kumagai. Recurrent neural networks for learning long-term temporal dependencies with reanalysis of time scale representation. *arXiv*, 2021. doi: 10.48550/arxiv.2111.03282.
57. Johannes Alt, Laszlo Erdos, and Torben Krüger. Spectral radius of random matrices with independent entries. *arXiv*, 2019. doi: 10.48550/arxiv.1907.13631.
58. Benjamin Drukarch, Micha M. M. Wilhelmus, and Shamit Shrivastava. The thermodynamic theory of action potential propagation: a sound basis for unification of the physics of nerve impulses. *Reviews in the Neurosciences*, 33(3):285–302, 2022. ISSN 0334-1763. doi: 10.1515/revneuro-2021-0094.
59. Zoubin Ghahramani. Introduction to hidden markov models and bayesian networks.
60. Andrew J. Quinn, Diego Vidaurre, Romesh Abeysuriya, Robert Becker, Anna C. Nobre, and Mark W. Woolrich. Task-evoked dynamic network analysis through hidden markov modeling. *Frontiers in Neuroscience*, 12:603, 2018. ISSN 1662-4548. doi: 10.3389/fnins.2018.00603.
61. Aditi Jha, Victor Geadah, and Jonathan W. Pillow. Modeling complex animal behavior with latent state inverse reinforcement learning. *bioRxiv*, page 2024.11.13.623515, 2024. doi: 10.1101/2024.11.13.623515.
62. Silvan C. Quax, Michele D'Asaro, and Marcel A. J. van Gerven. Adaptive time scales in recurrent neural networks. *Scientific Reports*, 10(1):11360, 2020. doi: 10.1038/s41598-020-68169-x.
63. Laura N. Driscoll, Krishna Shenoy, and David Sussillo. Flexible multitask computation in recurrent networks utilizes shared dynamical motifs. *Nature Neuroscience*, 27(7):1349–1363, 2024. ISSN 1097-6256. doi: 10.1038/s41593-024-01668-6.
64. Simon Ciranka, Juan Linde-Domingo, Ivan Padezhki, Clara Wicharz, Charley M. Wu, and Bernhard Spitzer. Asymmetric reinforcement learning facilitates human inference of transitive relations. *Nature Human Behaviour*, 6(4):555–564, 2022. doi: 10.1038/s41562-021-01263-w.
65. John D Murray, Alberto Bernacchia, David J Freedman, Ranulfo Romo, Jonathan D Wallis, Xinying Cai, Camillo Padoa-Schioppa, Tatiana Pasternak, Hyojung Seo, Daeyeol Lee, and Xiao-Jing Wang. A hierarchy of intrinsic timescales across primate cortex. *Nature Neuroscience*, 17(12):1661–1663, 2014. ISSN 1097-6256. doi: 10.1038/nn.3862.
66. Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. ISSN 0028-0836. doi: 10.1038/nature14236.
67. H Francis Song, Guangyu R Yang, and Xiao-Jing Wang. Reward-based training of recurrent neural networks for cognitive and value-based tasks. *eLife*, 6:e21492, 2017. doi: 10.7554/elife.21492.
68. Robert C Wilson and Anne GE Collins. Ten simple rules for the computational modeling of behavioral data. *eLife*, 8:e49547, 2019. doi: 10.7554/elife.49547.
69. Milos Stankovic. Multi-agent reinforcement learning. *2016 13th Symposium on Neural Networks and Applications (NEUREL)*, pages 1–1, 2016. doi: 10.1109/neurel.2016.7800108.
70. Alberto Bernacchia, Hyojung Seo, Daeyeol Lee, and Xiao-Jing Wang. A reservoir of time constants for memory traces in cortical neurons. *Nature Neuroscience*, 14(3):366–372, 2011. ISSN 1097-6256. doi: 10.1038/nn.2752.
71. Stefano Recanatesi, Serena Bradde, Vijay Balasubramanian, Nicholas A. Steinmetz, and Eric Shea-Brown. A scale-dependent measure of system dimensionality. *bioRxiv*, page 2020.12.19.423618, 2020. doi: 10.1101/2020.12.19.423618.
72. Guillaume Hennequin, Tim P. Vogels, and Wulfram Gerstner. Optimal control of transient dynamics in balanced networks supports generation of complex movements. *Neuron*, 82(6):1394–1406, 2014. ISSN 0896-6273. doi: 10.1016/j.neuron.2014.04.045.
73. John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv*, 2015. doi: 10.48550/arxiv.1506.02438.
74. Zhizhou Ren, Guangxiang Zhu, Hao Hu, Beining Han, Jianglun Chen, and Chongjie Zhang. On the estimation bias in double q-learning. *arXiv*, 2021. doi: 10.48550/arxiv.2109.14419.
75. Dong-Ki Kim, Miao Liu, Matthew Riemer, Chuangchuang Sun, Marwa Abdulhai, Golnaz Habibi, Sebastian Lopez-Cot, Gerald Tesauro, and Jonathan P. How. A POLICY GRADIENT ALGORITHM FOR LEARNING TO LEARN IN MULTIAGENT REINFORCEMENT LEARNING. .
76. Yoav Ger, Moni Shahar, and Nitzan Shahar. Using recurrent neural network to estimate irreducible stochasticity in human choice-behavior.
77. H. Francis Song, Guangyu R. Yang, and Xiao-Jing Wang. Training excitatory-inhibitory recurrent neural networks for cognitive tasks: A simple and flexible framework. *PLoS Computational Biology*, 12(2):e1004792, 2016. ISSN 1553-734X. doi: 10.1371/journal.pcbi.1004792.
78. Grace W Lindsay. Testing the tools of systems neuroscience on artificial neural networks. *arXiv*, 2022. doi: 10.48550/arxiv.2202.07035.
79. Kayson Fakhar and Claus C. Hilgetag. Systematic perturbation of an artificial neural network: A step towards quantifying causal contributions in the brain. *bioRxiv*, page 2021.11.04.467251, 2021. doi: 10.1101/2021.11.04.467251.
80. Qianyi Li and Haim Sompolinsky. Statistical mechanics of deep linear neural networks: The backpropagating kernel renormalization.
81. Mikail Khona, Sarthak Chandra, Joy J Ma, and Ila Fiete. Winning the lottery with neural connectivity constraints: faster learning across cognitive tasks with spatially constrained sparse RNNs. *arXiv*, 2022. doi: 10.48550/arxiv.2207.03523.
82. Barbara Feulner, Matthew G. Perich, Raeed H. Chowdhury, Lee E. Miller, Juan Álvaro Gallego, and Claudia Clopath. Small, correlated changes in synaptic connectivity may facilitate rapid motor learning. *bioRxiv*, page 2021.10.01.462728, 2021. doi: 10.1101/2021.10.01.462728.
83. T. L. Veuthey, K. Derosier, S. Kondapavulur, and K. Ganguly. Single-trial cross-area neural population dynamics during long-term skill learning. *Nature Communications*, 11(1):4057, 2020. doi: 10.1038/s41467-020-17902-1.
84. Shih-Yi Tseng, Selmaan N. Chettih, Charlotte Arlt, Roberto Barroso-Luque, and Christopher D. Harvey. Shared and specialized coding across posterior cortical areas for dynamic navigation decisions. *Neuron*, 110(15):2484–2502.e16, 2022. ISSN 0896-6273. doi: 10.1016/j.neuron.2022.05.012.
85. Nathaniel D. Daw. Are we of two minds? *Nature Neuroscience*, 21(11):1497–1499, 2018. ISSN 1097-6256. doi: 10.1038/s41593-018-0258-2.
86. Mitchell Ostrow, Robert Guangyu Yang, and Hyojung Seo. Representational geometry of social inference and generalization in a competitive game.
87. Michael Kleinman, Chandramouli Chandrasekaran, and Jonathan C. Kao. Recurrent neural network models of multi-area computation underlying decision-making. *bioRxiv*, page 798553, 2020. doi: 10.1101/798553.
88. Zach Cohen, Brian DePasquale, Mikio C. Aoi, and Jonathan W. Pillow. Recurrent dynamics of prefrontal cortex during context-dependent decision-making. *bioRxiv*, page 2020.11.27.401539, 2020. doi: 10.1101/2020.11.27.401539.
89. Daniel Durstewitz, Georgia Koppe, and Max Ingo Thurm. Reconstructing computational dynamics from neural measurements with recurrent neural networks. *bioRxiv*, page 2022.10.31.514408, 2022. doi: 10.1101/2022.10.31.514408.
90. Sean E. Cavanagh, John P. Towers, Joni D. Wallis, Laurence T. Hunt, and Steven W. Kennerley. Reconciling persistent and dynamic hypotheses of working memory coding in prefrontal cortex. *Nature Communications*, 9(1):3498, 2018. doi: 10.1038/s41467-018-05873-3.
91. Adrian Valente, Srdjan Ostojic, and Jonathan Pillow. Probing the relationship between linear dynamical systems and low-rank recurrent neural network models. *arXiv*, 2021. doi: 10.48550/arxiv.2110.09804.
92. Jonathan W. Pillow and Maneesh Sahani. Editorial overview: Machine learning, big data, and neuroscience. *Current Opinion in Neurobiology*, 2019.
93. Manuel Molano-Mazon, Yuxiu Shao, Daniel Duque, Guangyu Robert Yang, Srdjan Ostojic, and Jaime de la Rocha. Ecologically pre-trained RNNs explain suboptimal animal decisions. *bioRxiv*, page 2021.05.15.444287, 2022. doi: 10.1101/2021.05.15.444287.
94. Daeyeol Lee, Hyojung Seo, and Min Whan Jung. Neural basis of reinforcement learning and decision making. *Annual Review of Neuroscience*, 35(1):287–308, 2012. ISSN 0147-006x. doi: 10.1146/annurev-neuro-062111-150512.
95. Yael Niv. Reinforcement learning in the brain.
96. Valeria Fascianelli, Fabio Stefanini, Satoshi Tsujimoto, Aldo Genovesio, and Stefano Fusi. Neural representational geometry correlates with behavioral differences between monkeys. 2022. doi: 10.1101/2022.10.05.511024.
97. Brian DePasquale, Carlos D. Brody, and Jonathan W. Pillow. Neural population dynamics underlying evidence accumulation in multiple rat brain regions. *bioRxiv*, page 2021.10.28.465122, 2024. doi: 10.1101/2021.10.28.465122.
98. Kyle Aitken and Stefan Mihalas. Neural population dynamics of computing with synaptic modulations. *bioRxiv*, page 2022.06.27.497776, 2022. doi: 10.1101/2022.06.27.497776.
99. Konstantin Volzhenin, Jean-Pierre Changeux, and Guillaume Dumas. Multilevel development of cognitive abilities in an artificial neural network. *bioRxiv*, page 2022.01.24.477526, 2022. doi: 10.1101/2022.01.24.477526.
100. Qianli Yang, Zhongqiao Lin, Wenyi Zhang, Jianshu Li, Xiyuan Chen, Jiaqi Zhang, and Tianming Yang. Monkey plays pac-man with compositional strategies and hierarchical decision-making. *bioRxiv*, page 2021.10.02.462713, 2021. doi: 10.1101/2021.10.02.462713.
101. T Anderson Keller, Qinghe Gao, and Max Welling. Modeling category-selective cortical regions with topographic variational autoencoders. *arXiv*, 2021. doi: 10.48550/arxiv.2110.13911.
102. Minyoung Huh, Hossein Mobahi, Richard Zhang, Brian Cheung, Pulkit Agrawal, and Phillip Isola. The low-rank simplicity bias in deep networks. *arXiv*, 2021. doi: 10.48550/arxiv.2103.10427.
103. Bobak T Kiani, Randall Balestriero, Yubei Chen, Seth Lloyd, and Yann LeCun. Joint embedding self-supervised learning in the kernel regime. *arXiv*, 2022. doi: 10.48550/arxiv.2209.14884.
104. Timothy Doyeon Kim, Thomas Zhihao Luo, Jonathan W. Pillow, and Carlos D. Brody. Inferring latent dynamics underlying neural population activity via neural differential equations. .
105. Kamesh Krishnamurthy, Tankut Can, and David J. Schwab. Theory of gating in recurrent neural networks. *Physical Review X*, 2022.
106. Nicholas A. Roy, Ji Hyun Bak, The International Brain Laboratory, Athena Akrami, Carlos D. Brody, and Jonathan W. Pillow. Extracting the dynamics of behavior in sensory decision-making experiments. *Neuron*, 109(4):597–610.e6, 2021. ISSN 0896-6273. doi: 10.1016/j.neuron.2020.12.004.
107. Paul I. Jaffe, Russell A. Poldrack, Robert J. Schafer, and Patrick G. Bissett. Discovering dynamical models of human behavior. *bioRxiv*, page 2022.03.20.484666, 2022. doi: 10.1101/2022.03.20.484666.
108. Takuya Ito, Guangyu Robert Yang, Patryk Laurent, Douglas H. Schultz, and Michael W. Cole. Constructing neural network models from brain data reveals representational trans-

formations linked to adaptive behavior. *Nature Communications*, 13(1):673, 2022. doi: 10.1038/s41467-022-28323-7.

109. Nicolas Y. Masse, Guangyu R. Yang, H. Francis Song, Xiao-Jing Wang, and David J. Freedman. Circuit mechanisms for the maintenance and manipulation of information in working memory. *Nature Neuroscience*, 22(7):1159–1167, 2019. ISSN 1097-6256. doi: 10.1038/s41593-019-0414-3.
110. Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. Beyond neural scaling laws: beating power law scaling via data pruning.
111. Amir Dezfouli, Richard Nock, and Peter Dayan. Adversarial vulnerabilities of human decision-making. *Proceedings of the National Academy of Sciences*, 117(46):29221–29228, 2020. ISSN 0027-8424. doi: 10.1073/pnas.2016921117.
112. Diederick Kingma and Jimmy Lei Ba. ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION.
113. Niloufar Razmi and Matthew R. Nassar. Adaptive learning through temporal dynamics of state representation. *bioRxiv*, page 2020.08.03.231068, 2021. doi: 10.1101/2020.08.03.231068.
114. Cory Shain and William Schuler. A deep learning approach to analyzing continuous-time systems. *arXiv*, 2022. doi: 10.48550/arxiv.2209.12128.
115. Li Ji-An, Marcus K. Benna, and Marcelo G. Mattar. Discovering cognitive strategies with tiny recurrent neural networks. *bioRxiv*, page 2023.04.12.536629, 2024. doi: 10.1101/2023.04.12.536629.
116. ANTON MAXIMILIAN SCHÄFER and HANS-GEORG ZIMMERMANN. RECURRENT NEURAL NETWORKS ARE UNIVERSAL APPROXIMATORS. *International Journal of Neural Systems*, 17(04):253–263, 2007. doi: 10.1142/s0129065707001111. PMID: 17696290.
117. W. A. Wagenaar. Generation of random sequences by human subjects: A critical survey of literature. *Psychological Bulletin*, 77(1):65–72, 1972. ISSN 0033-2909. doi: 10.1037/h0032060.

Supplemental Figures

Fig. S1. $N_a = 2, N_r = 1$ model produces the best in and out of sample predictive performance on the strategic monkeys in our dataset. Above is a comparison of the four lowest order models fit to monkey E, as well as a histogram showing in-sample and out-of-sample performance of each across all monkeys.

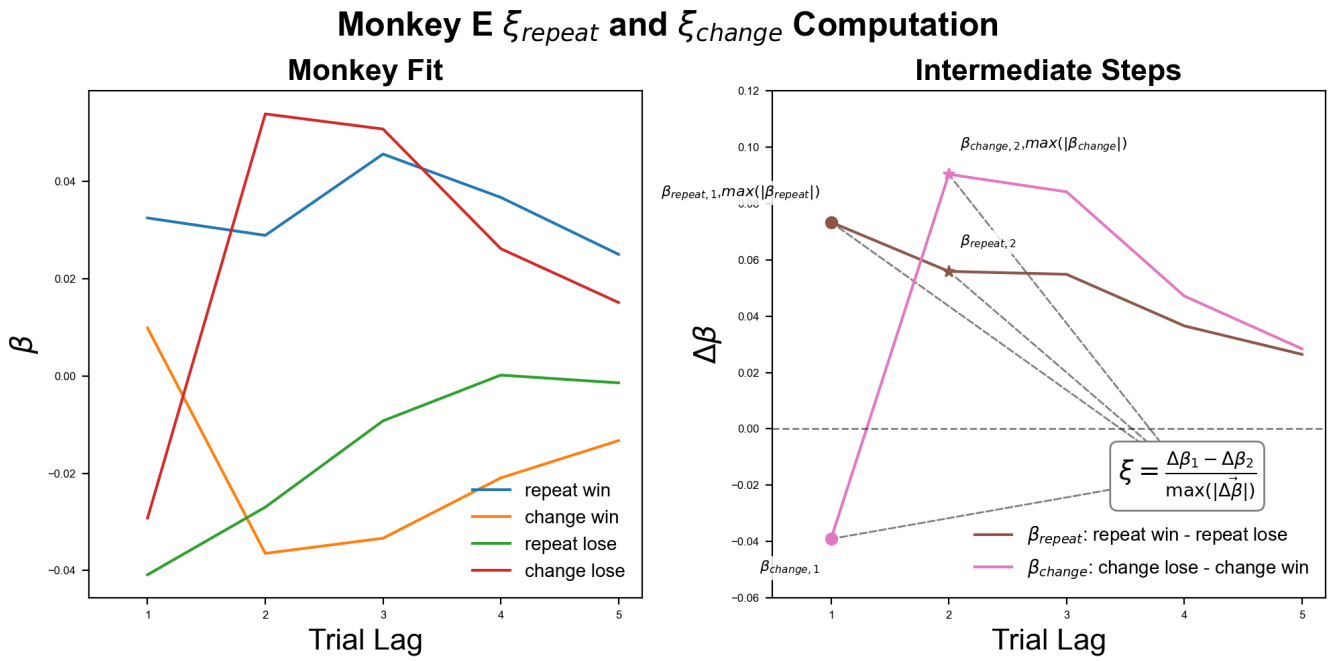


Fig. S2. Schematic of the computation of our RL behavioral deviation metric ξ from the behavioral regressions.

RL Model Comparison: Monkey E Data (MP1 vs MP2)

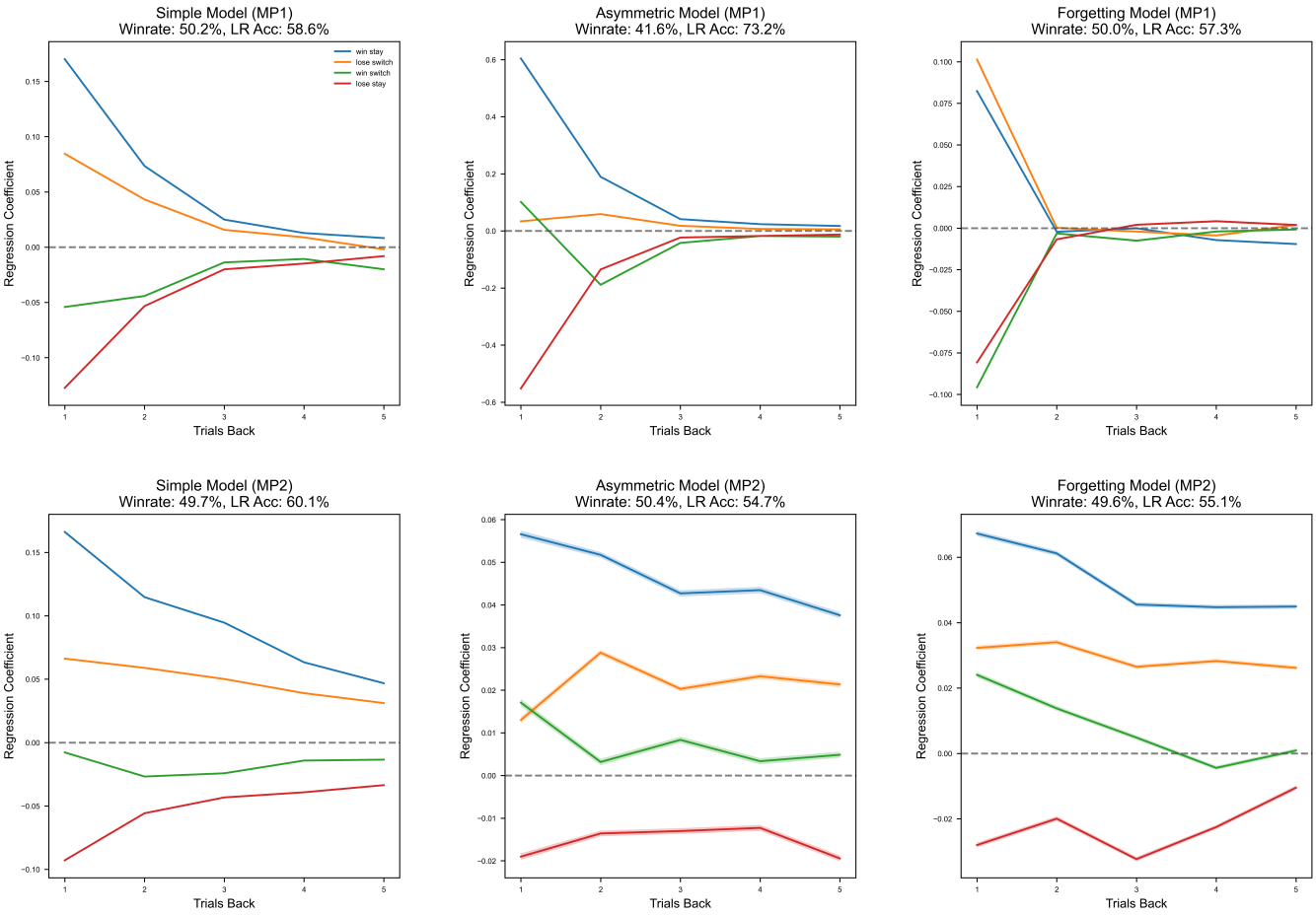


Fig. S3. Comparison of the three RL models mentioned, in order of complexity. The forgetting model recreates the behavioral dynamics observed by the monkeys playing against opponent 1, but fails against opponent 2. These models all sample their choices nondeterministically, as deterministic RL models are unable to perform against the matching pennies opponent.

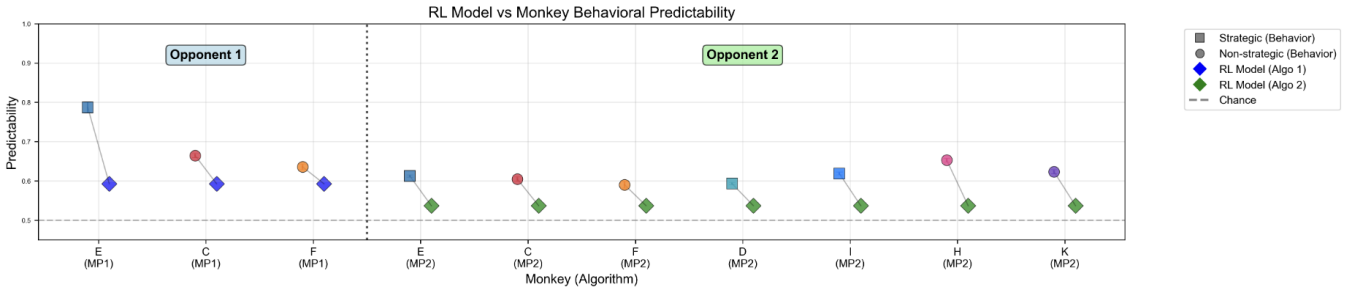


Fig. S4. Comparison of how well the logistic regression and RL model can predict each monkey's behavior for models fit at the monkey level. The logistic regression is significantly more capable than the RL model, but that is to be expected as it fits 21 terms instead of 2.

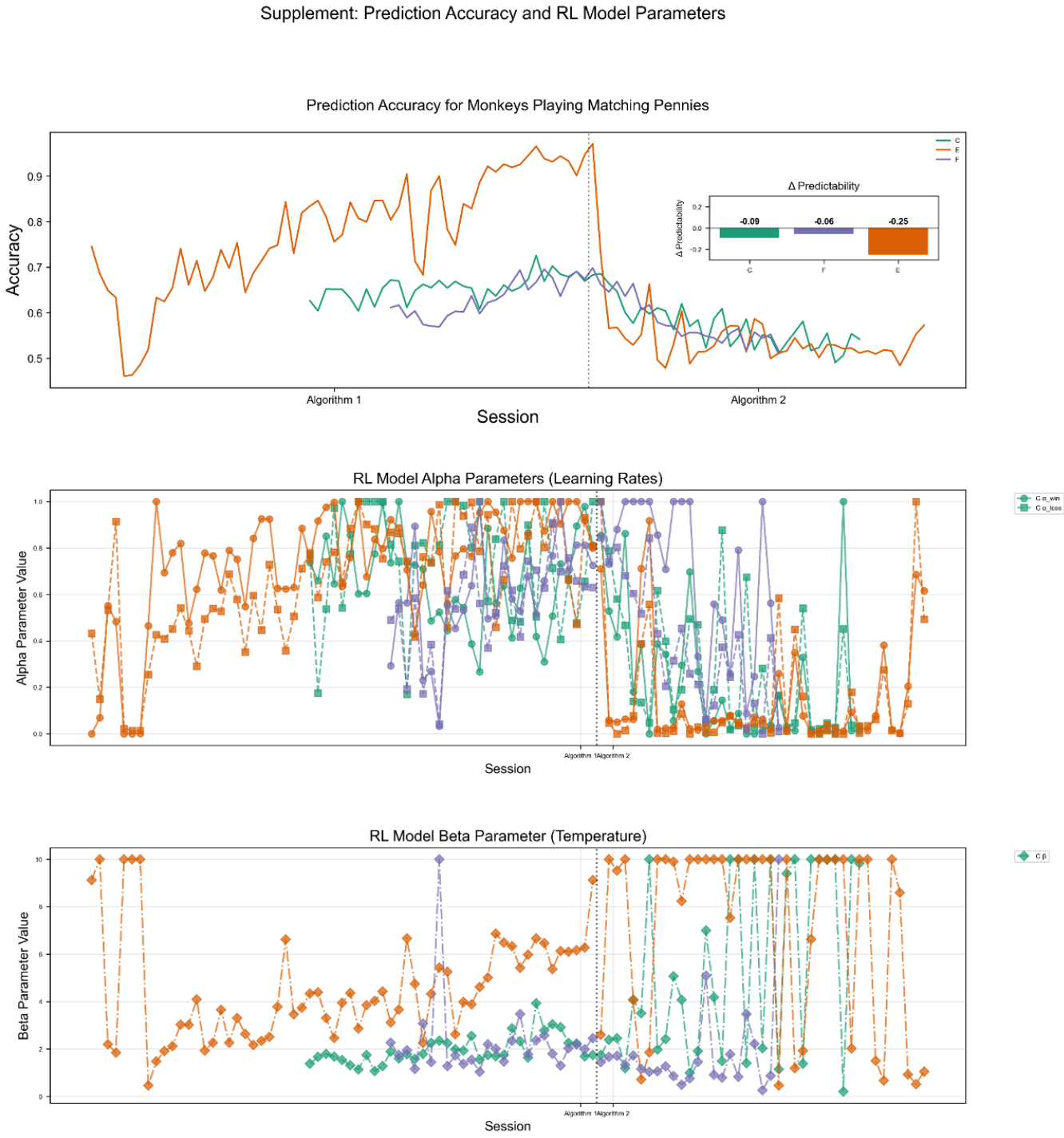


Fig. S5. Top: Logistic regression prediction accuracy for the each monkeys choices, ordered by session. We can observe that against opponent 1, all the monkeys get more predictable as they play more games against the opponent, especially monkey E. Transitioning to opponent 2, all monkeys increasingly become more difficult to predict. Inset is the average change between opponents in behavioral predictability for each monkey. Middle: Session-wise alpha value for the RL model fits. Behavioral timescales when playing against opponent 1 are much shorter than when playing opponent 2. Bottom: Inverse temperature fit to each session.

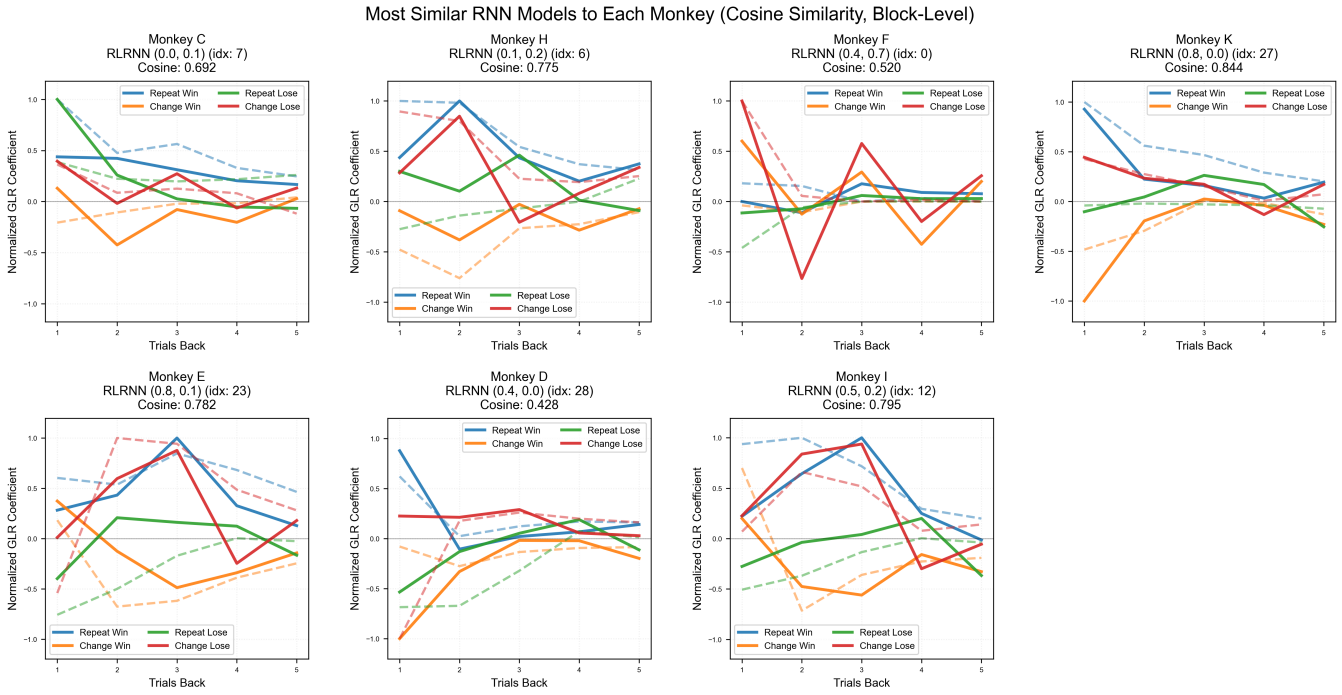


Fig. S6. Trial block-level regressions with the model most similar to it based on cosine similarity. We can observe that the RLRNN models with various parameters can generate very similar behavior to the animals' changing strategies. The solid lines are animal fits, while the dash lines are the models.

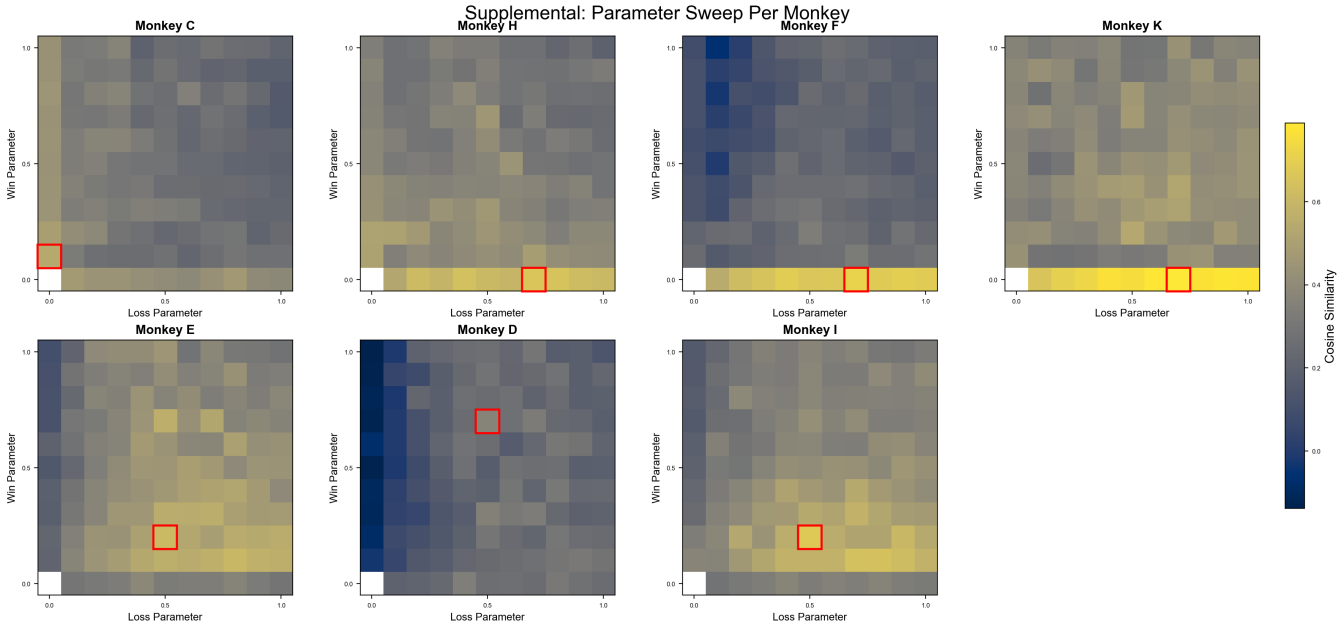


Fig. S7. Un-normalized trial block-level parameter similarities for each monkey. Each regression was fit to blocks of roughly 5000 trials, rounded to the nearest session, unlike Fig. 5 which used the entire dataset per monkey averaged over all monkeys in a group.

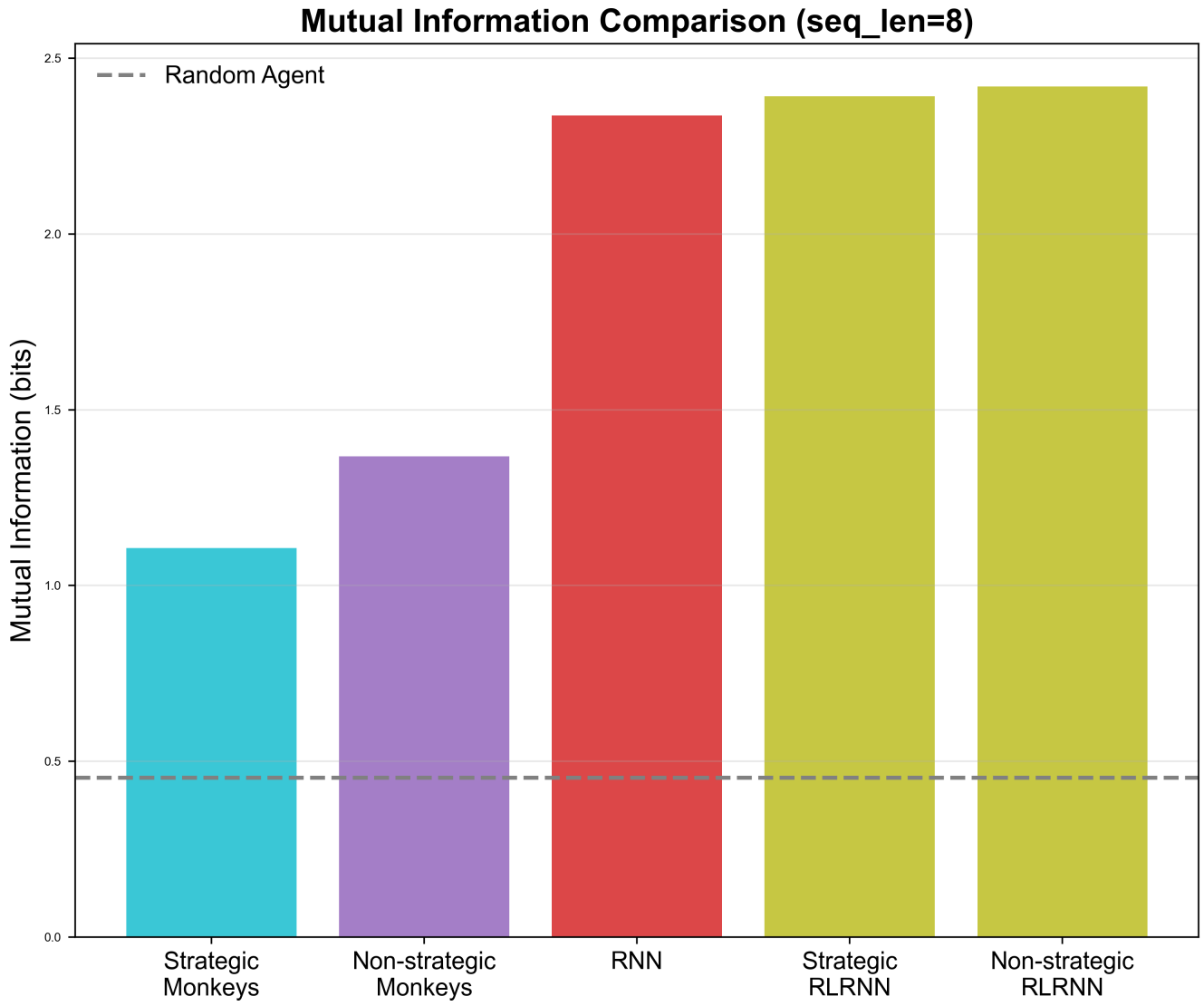


Fig. S8. Mutual information between monkey/model and opponent for the animals as well as RNN and RLRNN. The MI for a random actor is shown to provide a baseline. The mutual information comparison demonstrates that this outperformance is not due to the RNNs learning to harness randomness through instability. The RNNs seem to be learning some sort of deterministic strategy. The strategic RLRNN MI uses (1,0) and (1,.1) models, while the nonstrategic RLRNN MI uses (1,0) and (.9,0).