DEEP LEARNING IMPROVES PARAMETER ESTIMATION IN REINFORCEMENT LEARNING MODELS

Hua-Dong Xiong[*]
School of Psychology
Georgia Institute of Technology
hdx@gatech.edu

Li Ji-An[*]
Neurosciences Graduate Program
University of California San Diego
jil095@ucsd.edu

Marcelo G. Mattar[†]
Department of Psychology
New York University
marcelo.mattar@nyu.edu

Robert C. Wilson[†]
School of Psychology
Georgia Institute of Technology
rwilson337@gatech.edu

ABSTRACT

Cognitive modeling provides a formal method to articulate and test hypotheses about cognitive processes. However, accurately and reliably estimating model parameters remains challenging due to common issues in behavioral science, such as limited data, measurement noise, experimental constraints, and model complexity. These challenges hinder the interpretability of parameters in both behavioral and neural data.

In this study, we systematically examine how different optimization methods influence the generalizability, identifiability, robustness, and reliability of cognitive parameters in reinforcement learning models, which are widely used in psychology and neuroscience to study the cognitive mechanisms underlying value-based decision-making. Specifically, we compare the Nelder-Mead method (`fminsearch`), the de facto optimization approach in cognitive modeling, with a deep learning pipeline that integrates neural networks and modern optimization techniques, evaluating both across *ten* diverse value-based decision-making datasets involving human and animal participants performing various bandit tasks.

Surprisingly, we find that although both approaches achieve comparable predictive accuracy, they produce divergent parameter estimates. Parameters estimated via the deep learning pipeline consistently demonstrate smaller gaps between training and test performance (better generalizability), increased resistance to parameter perturbations (enhanced robustness), improved recovery of ground-truth parameters in low-data regimes (stronger identifiability), and greater consistency across repeated measurements from the same individuals (better test-retest reliability).

Our findings underscore the importance of systematically evaluating cognitive parameters from multiple perspectives to ensure their meaningful interpretation. Furthermore, the results highlight a critical yet underappreciated factor in cognitive modeling: the choice of optimization algorithm can systematically influence parameter estimates, even when predictive accuracy remains constant. These findings advocate for adopting deep learning-based optimization methods in cognitive modeling to improve parameter estimation.

Keywords Computational Modeling · Parameter Estimation · Reinforcement Learning · Deep Learning

[*]Co-first authors.
[†]Co-last authors.

1 Introduction

In neuroscience and psychology, cognitive models are widely used to investigate the processes underlying human and animal behavior. These models formalize how organisms learn from experience, make decisions, and adapt to their environment. One of the most successful models is reinforcement learning (RL), which describes how agents make decisions by learning value functions based on rewards and task-related feedback (Rescorla and Wagner, 1972; Sutton and Barto, 1998). Previous research has identified neural substrates underlying RL computations (Lee et al., 2012; Niv, 2009; Schultz et al., 1997), providing evidence for the biological implementation of reinforcement learning mechanisms.

Reinforcement learning models typically employ a small number of free parameters (e.g., learning rates, inverse temperatures, and choice perseveration) to characterize hypothesized cognitive processes. Accurately and reliably estimating these parameters is essential for interpreting the model and linking its outputs to behavioral and neural data. In practice, however, obtaining reliable estimates is often challenging due to constraints in human and animal experiments. These challenges include limited data, the inherent stochasticity of behavior, and the low inferential precision of task designs (i.e., the difficulty of precisely isolating specific cognitive mechanisms from observed choices). As a result, cognitive models often exhibit non-convex loss landscapes. Consequently, optimization methods play a critical role in identifying parameter combinations that best predict subjects' behavior.

Despite the importance of optimization, most research has focused on model fit, complexity, and parameter recovery as primary benchmarks for evaluating cognitive models. While these metrics are essential for model comparison, they often overlook the influence of the optimization procedure on parameter estimation outcomes. Many modeling studies implicitly assume that standard optimization routines produce accurate and robust estimates, provided the model fits the data well. However, this assumption often breaks down in practice due to the inherent challenges of parameter estimation in cognitive modeling. A systematic comparison of optimization procedures is therefore needed to evaluate their impact on estimated cognitive parameters, but such investigations remain rare.

The most commonly used parameter estimation technique in cognitive modeling is the Nelder–Mead simplex method (Nelder and Mead, 1965), which is the default optimization algorithm in MATLAB's `fminsearch` (The MathWorks, Inc., 2024) and SciPy's `fmin` (Virtanen et al., 2020) functions. Its off-the-shelf availability has led to widespread adoption across nearly every subfield of cognitive modeling (for representative applications, see Akam et al. (2015); Busemeyer and Myung (1987); Busemeyer and Diederich (2010); Dowman et al. (2016); Forstmann et al. (2011); Gershman (2018); Li et al. (2023); Pouget et al. (2011); Shinn et al. (2020); Wilson et al. (2019); Yechiam and Busemeyer (2005)). In contrast, gradient-based optimization methods, though widely used in modern deep learning, remain underutilized in cognitive modeling (cf. Binz et al. (2024); Giallanza et al. (2024)).

In this study, we evaluate these two optimization methods for estimating parameters in reinforcement learning models across ten behavioral datasets in which humans and animals perform bandit tasks with different action space, reward structures, and time horizons. Specifically, we compare the Nelder–Mead method with a deep learning pipeline that integrates neural networks and modern optimization techniques. We find, surprisingly, that although both optimization methods achieve similar predictive performance on held-out data, they yield different distributions of estimated parameters across subjects in most tasks. We systematically evaluate both optimization methods along three key dimensions: generalizability, robustness, and identifiability. The deep learning pipeline yields a smaller generalization gap, more robust parameter estimates, and superior parameter recovery when data are limited. Additionally, it achieves higher test-retest reliability in a dataset with repeated measurements from the same subject. Our results demonstrate that different optimization methods can cause reinforcement learning parameter estimates to converge on different minima, thereby affecting their generalizability, robustness, and identifiability. Furthermore, our findings indicate that the modern deep learning pipeline provide a promising approach for parameter estimation in cognitive modeling.

2 Results

2.1 Datasets and models

To systematically compare both optimization methods in parameter estimation, we evaluate their fitting performance across ten diverse decision-making datasets: (1) a 2-armed horizon task in humans designed to study exploration behaviors under varying horizons and uncertainty (`Wilson2014`) (Wilson et al., 2014); (2) a 3-armed horizon task in humans, an extension of the previous horizon task with varying horizons and information availability (`Dubois2022`) (Dubois and Hauser, 2022); (3) a 2-armed restless bandit task in mice, featuring drifting reward probabilities for binary rewards (`Chen2021`) (Chen et al., 2021); (4) a 4-armed restless bandit task in humans with continuously drifting rewards (`Bahrami2020`) (Bahrami and Navajas, 2020); (5) a 3-armed probabilistic reversal learning task in humans with binary rewards and dynamically changing reward probabilities (`Suthaharan2021`) (Suthaharan et al., 2021; Reed

et al., 2020); (6) a 3-armed probabilistic reversal learning task in mice (Groman 2018) (Groman et al., 2018); (7, 8) a 2-armed bandit task in humans with continuous rewards sampled from a Gaussian distribution and a horizon of 10 trials (`Gershman2018` and `Schurr2024`) (Gershman, 2018; Schurr et al., 2024); (9) a 2-armed instrumental learning task in humans, where the two arms have different reward probabilities (`Dezfouli2019`) (Dezfouli et al., 2019); (10) the Iowa Gambling Task in humans (`Steingroever2015`) (Steingroever et al., 2015). These tasks encompass diverse experimental designs to investigate biological value-based decision-making across varying reward distributions, environmental volatility, and time horizons, providing a comprehensive benchmark for comparison (for details, see Methods "Datasets").

We adopt the simplest and most widely used formulation of the RL model, in which the action value Q_t for the chosen action A_t at trial t is updated as follows:

$$Q_t(A_t) = Q_{t-1}(A_t) + \alpha \left(r_t - Q_{t-1}(A_t)\right), \tag{1}$$

where r_t is the reward at trial t, and α denotes the learning rate specifying how quickly the values are updated.

For action selection, we use a Softmax policy:

$$\Pr(A_t) = \frac{e^{\beta Q_t(A_t) + \kappa \delta(A_t, A_{t-1})}}{\sum_{A'} e^{\beta Q_t(A') + \kappa \delta(A', A_{t-1})}}, \tag{2}$$

where β is the inverse temperature specifying the action determinism, κ is the choice perseverance parameter reflecting an inertia tendency, and $\delta(A_t, A_{t-1})$ equals 1 if $A_t = A_{t-1}$ and 0 otherwise.

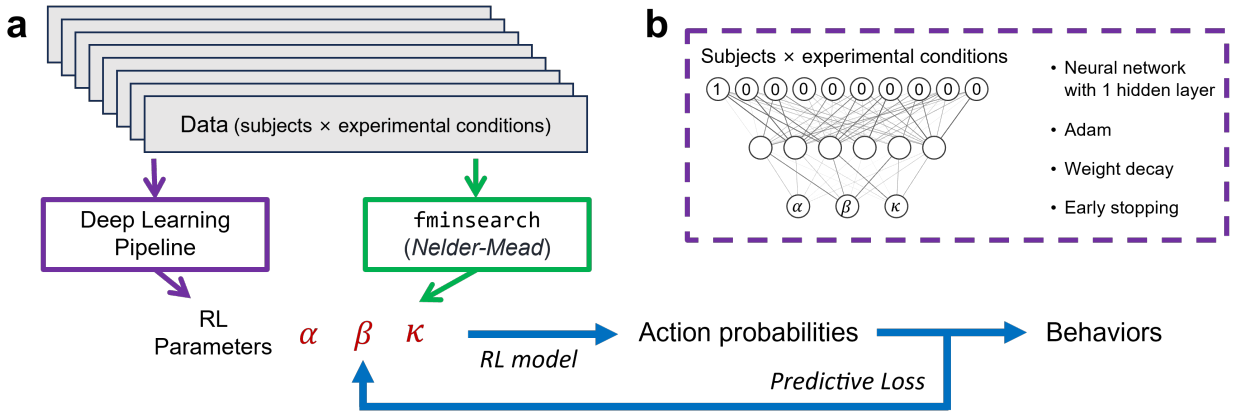

Figure 1: **Comparison of the Nelder-Mead method and the deep learning pipeline. a**, Both approaches optimize parameters in reinforcement learning (RL) models for each subject and each experimental condition to predict their choice behavior. **b**, The deep learning pipeline incorporates a neural network with one hidden layer, the AdamW optimizer (Adam with decoupled weight decay), and early stopping to estimate RL parameters.

2.2 Deep learning pipeline shows better generalizability

We apply the Nelder–Mead method and deep learning pipeline to fit RL models to behavioral data by adjusting a set of RL parameters (α, β, κ) for each subject and experimental condition, aiming to minimize the cross-entropy between the models' predictions and subjects' choices (Fig. 1a). The Nelder–Mead method directly optimizes the RL parameters. In contrast, the deep learning pipeline (Fig. 1b) uses a neural network with one hidden layer, which takes one-hot embeddings of each subject and experimental condition as input and outputs the corresponding RL parameters (see Methods "Neural network model" for details). The network parameters are optimized by the gradient-based optimizer AdamW (Kingma and Ba, 2017) to minimize the cross-entropy between the models' predictions and subjects' choices. To improve generalization, we further incorporate weight-decay regularization (Krogh and Hertz, 1991; Loshchilov and Hutter, 2019) and early stopping (Amari et al., 1997; Finnoff et al., 1993). The fitting performance of both optimization methods is evaluated using cross-validation: the parameters are optimized on training data, and the performance is evaluated on held-out test data (see Methods "Model fitting" for details).

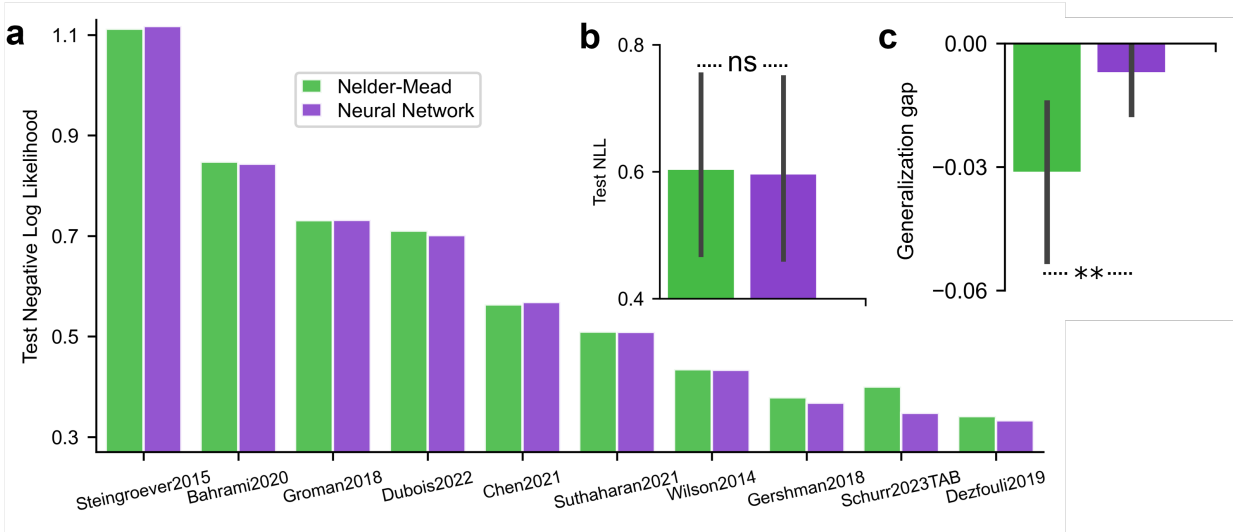

Figure 2: **Comparison of the Nelder–Mead and the deep learning pipeline across ten decision-making datasets. a**, Predictive performance of RL models optimized using both methods. The deep learning pipeline achieves a negative log-likelihood (lower is better) comparable to that of the Nelder–Mead method on held-out test data. **b**, Predictive performance averaged across all datasets. Both methods achieve comparable negative log-likelihoods on held-out test data. **c**, The generalization gap, defined as the difference in negative log-likelihood between training and testing, averaged across all datasets. The Nelder–Mead method exhibits a larger generalization gap, indicating potential overfitting on the training data. Error bars in **b,c** represent 95% confidence interval across ten datasets. `ns`: not significant. `**`: $p < 0.01$.

We found that the Nelder–Mead and deep learning pipeline performed similarly on held-out test data across ten datasets (Fig. 2a). No significant difference in average performance was observed across ten datasets (Fig. 2b, Wilcoxon signed-rank test: $W = 12.0, p = .131$, Cohen's $d = 0.031$), indicating that both methods performed similarly across diverse datasets.

Next, we assess potential overfitting by comparing the generalization gap (Keskar et al., 2017) between the two methods, defined as the difference in negative log-likelihood between the training and test sets. A negative generalization gap indicates potential overfitting on the training data. The deep learning pipeline exhibits a smaller generalization gap than the Nelder–Mead method (Fig. 2c, Wilcoxon signed-rank test: $W = 0.0, p = .002$, Cohen's $d = -0.956$), suggesting that it may produce better parameter estimates with reduced overfitting, thereby improving generalization.

2.3 Deep learning pipeline yields different parameter estimates

To determine whether the two methods yield different parameter estimates, we analyze the marginal distributions of estimated parameters across subjects and experimental conditions in ten datasets. Surprisingly, the two methods produce distinct parameter distributions (Fig. 3) despite achieving similar predictive performance on test data. While the distributions of κ largely overlap between the two methods, those of α and β differ significantly in some datasets. For instance, the deep learning pipeline tends to produce less extreme estimates (i.e., those on the boundary; e.g., see α in `Steingroever2015` and β in `Dubois2022`). This finding suggests that the two methods may find parameter estimates in different regions of the parameter space.

2.4 Deep learning pipeline shows enhanced robustness

Given that the two methods produce different parameter estimates, we examine how these estimates differ in their properties. Specifically, we use expected sharpness (Neyshabur et al., 2017) to evaluate the robustness of each model to parameter perturbations under the two optimization methods. This metric quantifies the model's sensitivity to parameter perturbations, characterizing the flatness of the loss landscape around the identified minima. Such "wide flat minima" are associated with improved generalization (Hochreiter and Schmidhuber, 1997; Neyshabur et al., 2017; Xie et al., 2021).

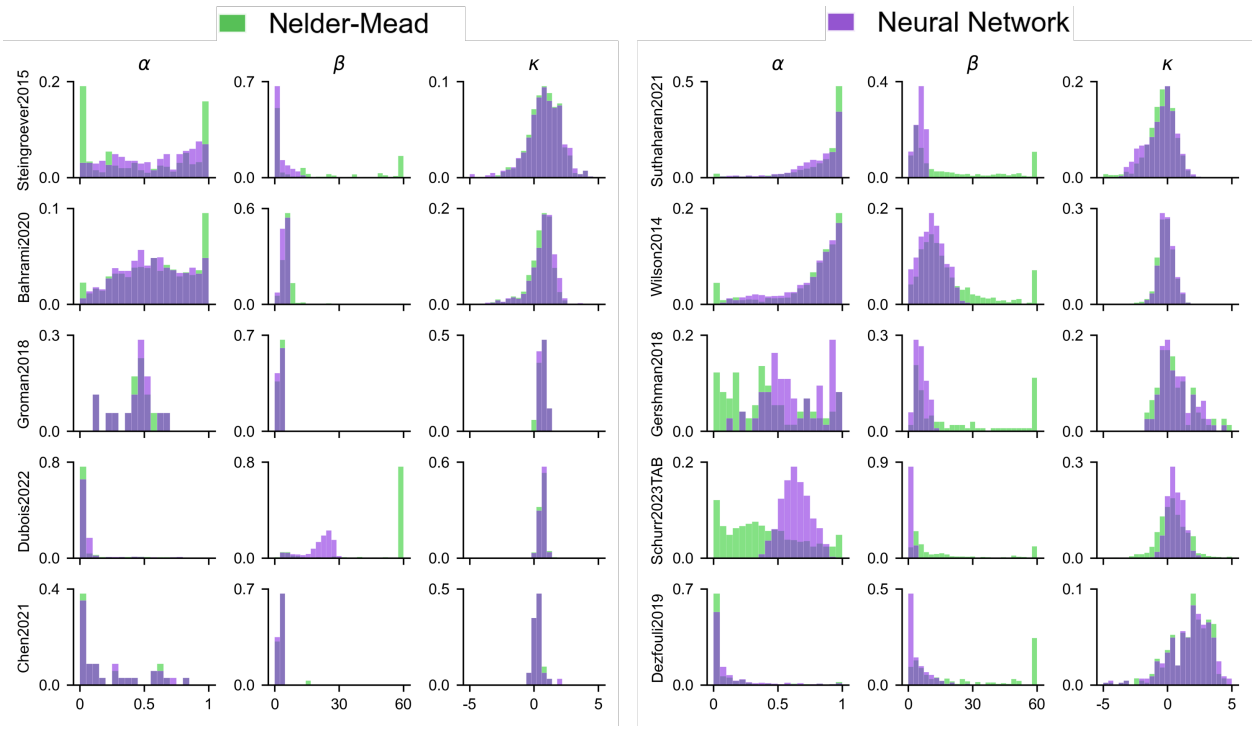

Figure 3: **Marginal distributions of RL parameters estimated by the Nelder–Mead and deep learning pipeline.** Each panel depicts the distribution of a single parameter (α, β, κ) across all subjects and experimental conditions within a dataset. The two methods produce distinct parameter distributions across ten datasets.

We perturbed the RL parameters estimated by both methods on each trial by adding the same independently and identically distributed (i.i.d.) zero-mean white Gaussian noise with varying levels of variance. We then computed the loss (negative log-likelihood) between subjects' behavior and the predictions of RL models using the perturbed parameters. The resulting loss increases were averaged across subjects and experimental conditions for each dataset (Fig. 4a), and then averaged across datasets (Fig. 4b). Across all datasets, we observed that loss increased (indicating decreased performance) with greater noise magnitude. Notably, the loss increased more slowly for the deep learning pipeline than for the Nelder–Mead method (Wilcoxon signed-rank test: $W = 49.0, p < .001$), indicating greater overall robustness. These findings suggest that parameters estimated by the deep learning pipeline are more robust, less sensitive to noise, and correspond to flatter minima.

2.5 Deep learning pipeline shows better identifiability

Having shown that the deep learning pipeline yields RL parameters with greater generalizability and robustness, we now examine another crucial criterion for meaningful interpretation: identifiability, defined as the ability to reliably recover ground-truth parameters from simulated data (Wilson and Collins, 2019). To this end, we generated simulated datasets from the model using fixed, known parameter values, and then re-estimated those parameters by fitting the same model structure to the data, treating those RL parameters as free variables in both optimization methods. If the estimated parameters match the ground-truth values used to generate the data, the model is considered highly identifiable. Conversely, if the ground-truth parameters cannot be reliably recovered, the estimates derived from empirical data may lack interpretability with respect to underlying cognitive processes. Although identifiability is largely shaped by the task, data, and model, the optimization method also plays a critical role. Some methods may locate the global minimum of the loss landscape more efficiently, thereby enhancing identifiability. To compare the identifiability of the two methods, we selected four tasks representing a range of task structures (Fig. 5).

We simulated varying amounts of data from RL models using 125 parameter combinations (five values spanning a plausible range for each parameter) and assessed recovery performance using the correlation and mean absolute error between estimated and ground-truth parameters (see Methods "Parameter recovery"). Recovery performance improved for both methods as the number of trials per subject increased, reaching similar levels with sufficient data (3200 trials).

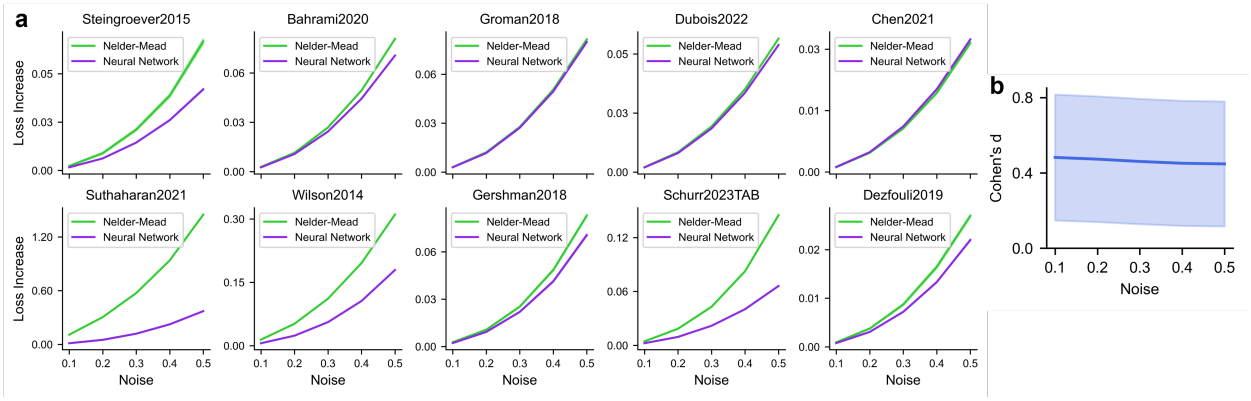

Figure 4: **Model robustness under parameter perturbations at varying noise levels. a,** Parameters estimated by the deep learning pipeline exhibit smaller increases in loss in response to injected white Gaussian noise, compared to the Nelder–Mead method, meaning lower expected sharpness and greater robustness to perturbation. The shaded area (mostly invisible) represents the 95% confidence interval across 10,000 simulations. **b,** Effect size of the difference in loss increase between the Nelder–Mead method and the neural network, in which the Nelder–Mead method shows a greater loss increase across ten datasets. The shaded area denotes the 95% confidence interval across ten datasets.

However, residual errors between estimated and ground-truth parameters remained in some tasks even with sufficient trials (e.g., Fig. 5b, 2-armed instrumental learning, third row), suggesting that identifiability may be fundamentally constrained by the task and model structure.

Notably, the deep learning pipeline outperformed the Nelder–Mead method in most tasks when trial counts per subject were limited (a range comparable to typical sample sizes in psychological experiments, as indicated by red and blue bars in Fig. 5). This result is particularly surprising given the common belief that deep learning methods require large datasets to be effective. These results suggest that the deep learning pipeline is well suited for typical dataset sizes, as it more accurately recovers ground-truth parameters.

2.6 Deep learning pipeline improves test-retest reliability

In previous sections, we showed that the deep learning pipeline demonstrates better generalizability, robustness, and identifiability in estimating RL parameters. But how do these properties translate into a more direct and intuitive connection to psychology and cognitive science? One such metric is test-retest reliability, which captures the temporal consistency of individual differences (a foundational concern in behavioral science), yet it is rarely applied to cognitive parameters.

We analyzed a unique dataset, `Schurr2023TAB` (Schurr et al., 2024), in which subjects performed a two-armed bandit task over 12 consecutive weeks, allowing us to measure correlations across repeated assessments for the same individuals. Following the exclusion criteria and methodology from the original study (Schurr et al., 2024), we computed the intra-class correlation (ICC) of the RL model parameters, a widely used measure of test-retest reliability.

To further control for differences in fitting performance between the two methods, we selected a subset of subjects (41 out of 90) for whom the negative log-likelihood (NLL) differed by less than 0.02 (Fig. 6a; NLL quantile: Q1 = -0.006, Q3 = 0.012). These results remain consistent when applying different threshold values.

Despite controlling for fitting performance, the two methods produced distinct parameter distributions (Fig. 6b). The deep learning pipeline also demonstrated greater robustness to parameter perturbations (Fig. 6c). Crucially, the deep learning pipeline yielded significantly higher ICC values for the estimated RL parameters (Fig. 6d): β: $t(40) = 3.25, p = .002,$ Cohen's $d = 0.52$; κ: $t(40) = 5.10, p < .001,$ Cohen's $d = 0.97$. The exception was α, which showed no significant difference ($t(40) = 0.71, p = .481,$ Cohen's $d = 0.06$), indicating stronger test-retest reliability overall. These findings suggest that the deep learning pipeline extracts more consistent parameter estimates across repeated measurements from the same individuals.

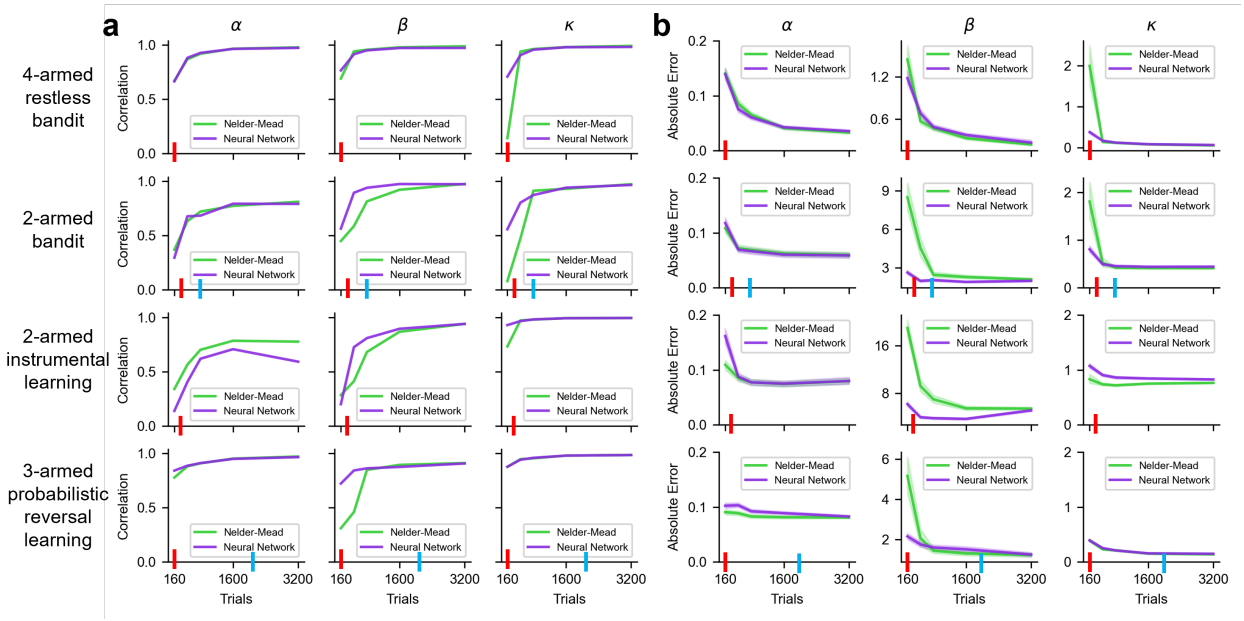

Figure 5: **Parameter recovery using the Nelder–Mead and deep learning pipeline across varying amounts of simulated data from an RL model.** The deep learning pipeline achieves higher or comparable **(a)** correlations with ground-truth parameters and **(b)** lower or comparable absolute errors across most tasks. The shaded area (barely visible) denotes the 95% confidence interval across all parameter combinations. Red and blue ticks on the x-axis indicate the average number of trials per subject in our datasets: 4-armed restless bandit (red - `Bahrami2020`); 2-armed bandit (red - `Gershman2018`, Schurr 2023TAB; blue - Wilson 2014); 2-armed instrumental learning (red - `Dezfouli2019`); and 3-armed probabilistic reversal learning (red - `Suthaharan2021`; blue - Groman 2018).

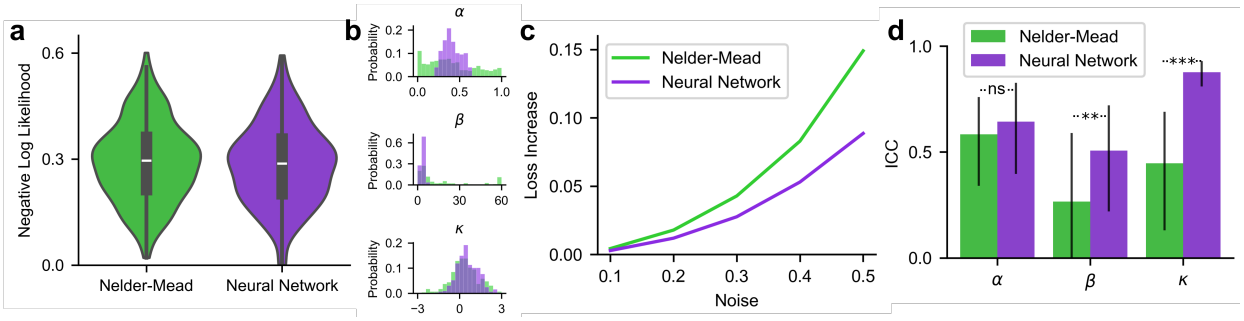

Figure 6: **Comparison of the Nelder–Mead method and the deep learning pipeline on the `Schurr2023TAB` dataset with multiple measurements from the same subjects.** A subset of subjects was selected to control for fitting performance (see main text). **a**, Both methods show similar distributions of fitting performance. **b**, The two methods yield distinct parameter distributions. **c**, The deep learning pipeline exhibits greater robustness to parameter perturbations. The shaded area (mostly not visible) represents the 95% confidence interval across 10,000 simulations. **d**, The deep learning pipeline achieves higher ICC values for the estimated parameters across eight weekly measurements from the same subjects, compared to the Nelder–Mead method. Error bars represent 95% confidence intervals across subjects. Significance levels are denoted as follows: ns = not significant, * $p < .05$, ** $p < .01$, *** $p < .001$.

3 Discussion

Accurate and reliable parameter estimation in cognitive models is essential for interpreting underlying cognitive strategies and linking them to neural data. In this work, we systematically examined two classes of optimization methods for parameter estimation in RL models across ten value-based decision-making datasets, focusing on four key dimensions: fitting performance, generalizability, robustness, identifiability and test-retest reliability. While both the de facto Nelder–Mead method (`fminsearch`) and the deep learning pipeline achieved comparable fitting performance on held-out test data, they surprisingly produced distinct parameter distributions. Furthermore, the deep learning pipeline exhibited a smaller generalization gap, lower sensitivity to parameter noise, and superior parameter recovery in low-data settings. Most notably, it substantially improved test-retest reliability—a key determinant of replicability in cognitive research. Our contribution lies not only in demonstrating that deep learning can outperform traditional methods, but also in showing that the choice of optimization algorithm can significantly influence estimated cognitive parameters across diverse datasets, even when predictive accuracy remains constant. To the best of our knowledge, this study provides the first systematic evaluation of how different optimization methods affect parameter estimation across multiple, diverse datasets in cognitive modeling.

Generalizability refers to a model's ability to perform well on unseen, held-out data—and ideally, under novel conditions. Better generalization suggests that a model has captured deeper underlying mechanisms that more effectively summarize behavioral patterns. In our study, both methods achieved similar test performance across the ten datasets, suggesting no significant advantage in out-of-sample accuracy for the deep learning pipeline. This may be attributable to the simplicity of the RL model formulation used here. However, we observed that the generalization gap—the difference in performance between training and test data—was smaller for the deep learning pipeline, indicating better generalizability in a weaker sense. We anticipate that the deep learning pipeline will demonstrate greater advantages in predictive performance when applied to more complex cognitive models and higher-dimensional tasks, given the well-established success of deep neural networks in learning from high-dimensional data.

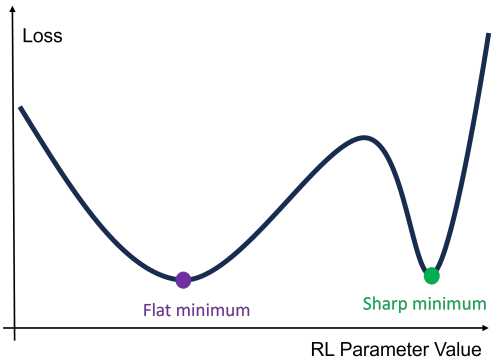

Figure 7: **Illustration of parameter flatness and sharpness.** In a non-convex loss landscape, different parameter estimates may produce similar losses. flatter minima, where loss remains stabler under small parameter perturbations, are generally considered better solutions and are often associated with stronger generalizability.

In our study, robustness is measured by expected sharpness—the sensitivity of the loss function to parameter perturbations. Despite different parameter estimates yielding similar losses, we found that the solutions identified by the deep learning pipeline exhibit lower sharpness (i.e., greater flatness). Although such flatness is rarely evaluated in cognitive models (but see Frank (2014)), the connection between "wide flat minima" and improved generalization is well established in the machine learning literature (Fig. 7) (Hochreiter and Schmidhuber, 1997; Dinh et al., 2017; Hoffer et al., 2017; Neyshabur, 2017; Xie et al., 2021). Intuitively, when the loss function is highly sensitive to parameter perturbations at sharp minima, stochastic gradient-based optimizers (such as AdamW, which we used) are more likely to "escape" these regions due to noisy updates. Such optimizers act as entropy maximizers, favoring solutions with broader support in parameter space (Chaudhari et al., 2019). They have been theoretically shown to prefer flat over sharp minima, leading to more generalizable solutions in deep neural networks (Xie et al., 2021). In addition, other components of the deep learning pipeline, such as overparameterization and regularization techniques (e.g., weight decay and early stopping), may introduce implicit biases that further promote generalization. Our expected sharpness metric is closely related to the trace of the Hessian (or the Fisher information; see below) and connects to the PAC-Bayesian framework for bounding generalization error in neural networks (Bjorck et al., 2021; Dziugaite and Roy, 2017; Fukumizu et al., 2019; Gubri et al., 2024; Xie et al., 2021). In summary, the deep learning pipeline we employed enhances the robustness and reliability of parameter estimation in cognitive models.

The parameter recovery test provides a measure of model identifiability by examining whether ground-truth parameters can be accurately recovered from simulated data generated using those parameters. High identifiability indicates that the estimated values faithfully represent the underlying cognitive mechanisms. We demonstrate that the deep learning pipeline achieves superior parameter recovery across four widely used decision-making tasks, particularly in the low-data regime typical of human behavioral experiments. Parameter identifiability is closely related to the Fisher information, which underlies the theoretical consistency of maximum likelihood estimation. Taken together, the findings on flatness and identifiability suggest that the deep learning pipeline more reliably identifies flat minima that both generalize better and align more closely with ground-truth parameters compared to the Nelder–Mead method.

Finally, as a concrete application, we show that the deep learning pipeline significantly improves test-retest reliability, which measures the consistency of parameter estimates across repeated measurements. Prior research has reported that the test-retest reliability of RL parameters is generally poor to modest [cite]. Owing to the improved generalizability, robustness, and identifiability of the deep learning pipeline, the observed increase in test-retest reliability suggests that the estimated parameters are more stable over time, providing more accurate characterizations of the underlying cognitive processes. This improvement enhances longitudinal cognitive research, including cognitive phenotyping in studies of individual differences in personality, development, and psychiatry [cite].

At first glance, identifiability and robustness may appear to be contradictory: the former requires high sensitivity of the likelihood function to changes in parameters (i.e., higher Fisher information), while the latter favors low sensitivity (i.e., a smaller negative Hessian) in the parameter space. However, as shown in the Methods ("Relations between identifiability, robustness, Hessian, and Fisher information"), identifiability and robustness are in fact complementary properties. A well-designed computational model can balance both, where higher Fisher information ensures identifiability, while a bounded Hessian ensures robustness.

In conclusion, our study systematically highlights an important and underappreciated factor in cognitive modeling: the choice of optimization algorithm can significantly shape parameter estimates, even when predictive accuracy is equivalent. Compared to the commonly used Nelder–Mead method, the deep learning pipeline—leveraging modern optimization techniques—produces more generalizable, robust, identifiable, and reliable parameter estimates, positioning it as a promising and scalable tool for cognitive modeling.

4 Method

4.1 Datasets

The ten datasets include several types of decision-making tasks performed by both humans and animals, featuring different levels of uncertainty, various reward structures, and horizon lengths.

The two-armed horizon task (Wilson et al., 2014) involves two options (arms), where the value difference between the two arms is sampled from 4, 8, 12, 20, and 30, and rewards are sampled from a normal distribution. The task includes a forced-choice phase in the first four trials, which facilitates the control of the uncertainty level regarding the value of each option. In the free-choice phase, the horizon is set to either 1 or 6, allowing researchers to study how decision-making changes with the amount of available information in the future. This dataset includes data from multiple labs and experiments (Wilson et al., 2014; Feng et al., 2021; Somerville et al., 2017; Waltz et al., 2020; Sadeghiyeh et al., 2020; Zajkowski et al., 2017) with n_{subjects} = 609 and n_{trials} = 754,890.

A three-armed version of this horizon task is presented in (Dubois and Hauser, 2022). This version increases task complexity by introducing a third arm, requiring more strategic considerations when choosing the best option. This dataset includes n_{subjects} = 1,012 and n_{trials} = 173,120.

The two-armed restless bandit task in mice (Chen et al., 2021) uses binary rewards. In this task, the reward probability for each arm changes gradually over time, requiring the mice to continuously adapt their choices based on these shifting probabilities. This dataset includes n_{subjects} = 32 and n_{trials} = 70,778.

The four-armed restless bandit task in humans (Bahrami and Navajas, 2020) features continuously evolving arm values ranging from 0 to 100, which follow an Ornstein–Uhlenbeck process toward 50. The initial values are sampled from a normal distribution, and rewards are drawn based on the arm's current value. Thus, the task incorporates both reward stochasticity and environmental volatility due to the fluctuating arm values. This dataset includes n_{subjects} = 965 and n_{trials} = 139,816.

The three-armed probabilistic reversal learning task in humans (Suthaharan et al., 2021) requires subjects to choose between three arms, each associated with different reward probabilities. In the first two blocks, probabilities are set to [0.9, 0.5, 0.1], and in later blocks, they shift to [0.8, 0.5, 0.2]. After ten consecutive selections of the best option, the reward probabilities are reversed, allowing researchers to examine how individuals respond to changing reward contingencies. This dataset includes n_{subjects} = 1,012 and n_{trials} = 173,120.

The three-armed probabilistic reversal learning task was also conducted in mice (Groman et al., 2018), with reward probabilities of [0.8, 0.5, 0.2], enabling cross-species comparisons of decision-making strategies. This dataset includes n_{subjects} = 17 and n_{trials} = 38,475.

The two-armed bandit task in humans (Gershman, 2018) has a horizon length of 10, with mean reward values sampled from a normal distribution. In experimental condition 1, the mean reward for one arm is fixed at 0, while the other arm's mean reward varies. In experimental condition 2, both arms have varying mean reward values, allowing researchers to examine how subjects balance exploration and exploitation. This dataset includes n_{subjects} = 89 and n_{trials} = 17,800.

Schurr et al. (2024) also tested Experimental Condition 2 of the two-armed bandit task to assess test-retest reliability. In this study, subjects engaged in a two-armed bandit task over 12 consecutive weeks. The dataset comprises 90 subjects and 78,820 trials. Participant exclusion criteria were consistent with those reported in the original paper.

The two-armed instrumental learning task in humans (Dezfouli et al., 2019) involves binary rewards, where the probability of obtaining a reward from one arm is higher than from the other. For the arm with the higher reward probability, the reward probabilities are [0.25, 0.125, 0.08]. For the arm with the lower reward probability, the reward probability is 0.05. The duration of each experimental condition is fixed, allowing the analysis of learning behavior over a defined period, particularly how subjects adapt to consistent reward probabilities. This dataset includes n_{subjects} = 101 and n_{trials} = 132,251.

Finally, the Iowa Gambling Task (Steingroever et al., 2015) presents subjects with four decks of cards, each associated with different reward and punishment structures. This task assesses decision-making under risk, as subjects must learn which decks are more beneficial in the long term based on experienced rewards and penalties. This dataset includes data from multiple labs and experiments, with n_{subjects} = 577 and n_{trials} = 62,525.

4.2 Neural network model

The neural network we used in the deep learning pipeline consists of a single hidden layer with no activation functions and is defined as follows:

$$[\alpha, \beta, \kappa] = \mathbf{W}_2(\mathbf{W}_1\mathbf{e}_{\text{sub}} + \mathbf{b}_2) + \mathbf{b}_1 \tag{3}$$

where $\mathbf{e}_{\text{sub}} \in \mathbb{R}^{n\times n}$ is a one-hot embedding for each subject per experimental condition, with n representing the total number of subjects multiplied by the number of experimental conditions in a given task. The weight matrix $\mathbf{W}_1 \in \mathbb{R}^{h\times n}$ maps the one-hot embedding to a hidden representation, while $\mathbf{b}_1 \in \mathbb{R}^h$ is the corresponding bias term. The second weight matrix $\mathbf{W}_2 \in \mathbb{R}^{3\times h}$ maps the hidden representation to the RL parameters, with bias term $\mathbf{b}_2 \in \mathbb{R}^3$. While this two-layer linear network is equivalent to a single-layer network in its representational power, they have substantially different training dynamics (Saxe et al., 2013). The output RL parameters α, β, κ are then fed in the RL models.

In PyTorch implementation, the entire RL recurrent computation and neural network feedforward computation are both embedded within the computational graph, enabling gradients to be computed via backpropagation through time.

4.3 Model fitting

The optimization objective is the negative log-likelihood of the observed choice data:

$$\mathcal{L} = -\sum_{t=1}^{T}\sum_{i=1}^{n}\Big[c_t^{(i)}\log\big(p_t^{(i)}\big) + \big(1 - c_t^{(i)}\big)\log\Big(1 - p_t^{(i)}\Big)\Big]. \tag{4}$$

Here, $c_t^{(i)}$ denotes the actual choice of subjects i on trial t and $p_t^{(i)}$ denotes the model's prediction probability for the subjects' choices (Eq. 2).

To mitigate overfitting and ensure a fair comparison between both optimization methods, we implemented a rigorous cross-validation protocol. For the Nelder–Mead method, we employed five-fold cross-validation, allocating 80% of the data for training and 20% for testing. For the deep learning pipeline, we adopted a nested cross-validation approach (Ji-An et al., 2023). Specifically, 70% of the data was designated for training, 12.5% for validation (used for early stopping and hyperparameter tuning), and 17.5% for testing. The outer loop comprised eight folds, while the inner loop included five folds.

Given the variability in datasets, experimental designs, and trial counts, we tailored the cross-validation data splits accordingly. For datasets comprising multiple short blocks with identical trial counts (Gershman, 2018; Schurr et al., 2024), we partition data into different blocks. For datasets with multiple short blocks of varying trial counts (each block containing fewer than 40 trials) (Wilson et al., 2014; Dubois and Hauser, 2022), we divided the data into non-overlapping 40-trial segments. When necessary, we added padding to maintain structural uniformity; padded trials were excluded from loss computation. For datasets with relatively few blocks and fewer than 200 trials per block (Bahrami and Navajas, 2020; Suthaharan et al., 2021; Reed et al., 2020; Steingroever et al., 2015), we split the data into non-overlapping trial sets for training, validation, and testing, applying masking as needed (Hans et al., 2024). For datasets with few blocks but more than 200 trials (Chen et al., 2021; Groman et al., 2018), we divided each session into non-overlapping blocks of approximately 100 trials.

For the Nelder–Mead optimization method, parameter fitting for each subject within a given experimental condition underwent 64 independent initializations. For reinforcement learning (RL) parameter estimation, we uniformly sampled four initial points: $\alpha \in [0, 1]$, $\beta \in [0, 60]$, and $\kappa \in [-5, 5]$.

To enhance the robustness of the deep learning pipeline, we augmented the training data by adding Gaussian noise $N(0, 0.01)$ to continuous rewards. Early stopping was applied when validation performance did not improve after 50 training epochs, ensuring that only models with superior validation loss were retained. This strategy has effectively prevented overfitting. Hyperparameters were optimized over the following search space: learning rate from {0.01, 0.02, 0.03}, weight decay from {0.01, 0.1, 0.2}, dropout from {0, 0.1}, and hidden layer sizes h from {50, 100}. The AdamW optimizer with default hyperparameters was used for training. Constraints were enforced on RL parameter estimation, ensuring $\alpha \in [0, 1]$ and $\beta \geq 0$.

The best model was selected based on the lowest negative log-likelihood on the test data, averaged across all folds. For the RL parameters, we averaged them over cross-validation folds. In the Nelder–Mead method, we used all five folds. In the deep learning pipeline, we use five selected folds (outer j and inner fold 0, where $j = 0, 1, 2, 3, 4$). Our results remained consistent for other fold selection protocols.

4.4 Estimating parameter robustness

We assessed parameter robustness by perturbing model parameters and measuring the resulting change in the loss function. Specifically, we added white Gaussian noise at each trial to the fitted parameters $\hat{\theta} \in \{\hat{\alpha}, \hat{\beta}_t, \hat{\kappa}_t\}$ estimated by each optimization method. The noise was independently sampled from a zero-mean Gaussian distribution, with variance scaled by the average parameter:

$$\tilde{\theta}_t = \hat{\theta} + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2 \mu_{\hat{\theta}}^2). \tag{5}$$

μ_θ is the parameter averaged over all subjects and both optimization methods, which ensures a fair comparison. The noise variance coefficient σ^2 was selected from $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. For each predefined σ and optimization method, we generated 10,000 perturbed parameter sets. Each set was used to simulate model predictions, and the corresponding loss was computed as the negative log-likelihood between perturbed model output probabilities and observed behavioral data. We defined robustness (expected sharpness) as the average increase in loss, quantified by the difference between the perturbed and original losses.

4.5 Parameter recovery

For each task, we selected simulated parameters within the most plausible range to perform the parameter recovery test (Wilson and Collins, 2019). This range was approximately defined using the interquartile range (second-to-third quartile) of parameter distributions obtained from our two methods. For tasks where subjects exhibited highly concentrated parameter distributions, we slightly expanded the range. This resulted in the following parameter ranges for each task: $\alpha \in [0.3, 0.9]$, $\beta \in [4, 10]$, and $\kappa \in [0, 1.5]$ for the four-armed restless bandit task (Bahrami and Navajas, 2020); $\alpha \in [0.3, 0.8]$, $\beta \in [4, 15]$, and $\kappa \in [-0.5, 1.5]$ for the two-armed bandit task (Gershman, 2018; Schurr et al., 2024); $\alpha \in [0.05, 0.35]$, $\beta \in [1, 20]$, and $\kappa \in [0, 3]$ for the two-armed instrumental learning task (Dezfouli et al., 2019); and $\alpha \in [0.4, 0.9]$, $\beta \in [4, 12]$, and $\kappa \in [-1.5, 0.5]$ for the three-armed probabilistic reversal learning task (Suthaharan et al., 2021; Reed et al., 2020; Groman et al., 2018).

Within these ranges, we uniformly sampled five values for each parameter. This sampling effectively decouples correlations between the ground-truth parameters. Using these 125 parameter combinations, we simulated data for 1, 3, 5, 10, and 20 sequence, with each sequence consisting of 160 trials. This resulted in datasets containing 160, 480, 800, 1,600, and 3,200 trials per parameter set.

For the four-armed restless bandit task and the three-armed probabilistic reversal learning task, each sequence contained a single block of 160 trials. In the three-armed probabilistic reversal learning task, we used reward probabilities of [0.8, 0.5, 0.2]. In the two-armed bandit task, each sequence consisted of eight blocks of 20 trials, and we used experimental condition 2 from (Gershman, 2018). In the two-armed instrumental learning task, each sequence consisted of two blocks of 80 trials, and we used reward probabilities of [0.125, 0.05], corresponding to the condition in which one arm had a higher reward probability. These choices were made to align with the original task designs.

We then estimated the parameters from the simulated data using both optimization methods across different data sizes, following the same model fitting and selection procedure described in Methods Model fitting. Due to the computational cost of model recovery and the fact that we had 125 distinct parameter combinations, we performed the recovery procedure once for each combination. While this introduces some stochasticity at the level of individual simulations, the large number of combinations provides a robust basis for assessing overall trends in recovery performance.

4.6 Parameter test-retest reliability

To assess the test-retest reliability of RL parameters over multiple measurements, we computed non-parametric bootstrapped intraclass correlations, ICC(2,1), using 1,000 samples, following methods from (Enkavi et al., 2019; Vallat, 2018). For each RL parameter, we used the mean of the posterior distribution as a point estimate and calculated the ICC based on the data of 90 participants with 12 sessions each. However, ICC cannot be computed when data is missing. To address this, we included all participants but restricted the analysis to the maximal number of sessions with complete data, using eight weekly measurements from the same subjects.

4.7 Relations between identifiability, robustness, Hessian, and Fisher information

In this section, we examine the mathematical relationship between robustness (expected sharpness) and identifiability measured in this paper and well-known properties like Fisher information and the Hessian.

Below we consider a statistical model with one parameter θ (the generalization to more parameters is direct). The data X is generated by the model where the parameter takes $\theta = \theta^*$ ($X \sim \theta^*$). The log-likelihood function (equivalent to the negative loss) is defined as

$$\ell(\theta; X) = \log p(X \mid \theta). \tag{6}$$

When θ is a local minimum, ℓ can be locally expanded to the second-order approximation:

$$\ell(\theta + \Delta\theta; X) \approx \ell(\theta; X) + \nabla_\theta \ell(\theta; X)\Delta\theta + \frac{1}{2}\nabla_\theta^2 \ell(\theta; X)\Delta\theta^2 = \ell(\theta; X) + \frac{1}{2}\nabla_\theta^2 \ell(\theta; X)\Delta\theta^2, \tag{7}$$

where the term $\nabla_\theta \ell(\theta; X)$ vanishes. The expected sharpness is then given by:

$$\mathbb{E}_{\Delta\theta}\, \mathbb{E}_{X\sim\theta^*}[\ell(\theta + \Delta\theta; X) - \ell(\theta; X)] \approx \frac{1}{2}\, \mathbb{E}_{X\sim\theta^*}[\nabla_\theta^2 \ell(\theta; X)]\, \mathbb{E}_{\Delta\theta}[\Delta\theta^2] = \frac{1}{2}\, \mathbb{E}_{X\sim\theta^*}\, \mathbf{H}(\theta; X)\, \mathbb{E}_{\Delta\theta}[\Delta\theta^2], \tag{8}$$

where $\mathbf{H}$ is the Hessian matrix which captures the local curvature. When $\Delta\theta \to 0$, the equation between expected sharpness and expected Hessian becomes exact.

When $\theta = \theta^*$, the Fisher information matrix, defined as

$$\mathbf{I}(\theta^*) = \mathbb{E}_{X\sim\theta^*}\left[\nabla \ell(\theta^*; X)\right]^2. \tag{9}$$

is equal to the negative expected Hessian (under standard regularity conditions):

$$\mathbf{I}(\theta^*) = -\mathbb{E}_{X\sim\theta^*}\left[\nabla^2 \ell(\theta^*; X)\right] = -\,\mathbb{E}_{X\sim\theta^*}\, \mathbf{H}(\theta^*; X). \tag{10}$$

The Fisher information matrix measures the amount of information that the data X carries about the unknown ground-truth parameter θ^*. Thus, it directly reflects the identifiability of the ground-truth parameters, as shown in the fact that the maximum likelihood estimator converges in distribution to a normal distribution:

$$\sqrt{N}(\hat{\theta}_{\text{MLE}} - \theta^*) \xrightarrow{d} \mathcal{N}(0, \mathbf{I}(\theta^*)^{-1}). \tag{11}$$

In practice, both the Fisher information matrix and the Hessian are estimated empirically:

$$\hat{F}(\theta^*) = \frac{1}{N}\sum_{i=1}^{N}(\nabla_{\theta^*}\ell(\theta^*; x_i))^2. \tag{12}$$

$$\hat{H}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\nabla_\theta^2 \ell(\theta; x_i). \tag{13}$$

Specifically, these concepts measure distinct aspects of model parameters. The Fisher information for identifiability is always calculated at the ground truth parameters: if $\hat{\mathbf{I}}(\theta_1^*; \{x_1, ..., x_n\} \sim \theta_1^*) > \hat{\mathbf{I}}(\theta_2^*; \{y_1, ..., y_n\} \sim \theta_2^*)$, then θ_1^* has higher identifiability. On the other hand, if $-\hat{\mathbf{H}}(\theta_1^*; \{x_1, ..., x_n\} \sim \theta_1^*) < -\hat{\mathbf{H}}(\theta_2; \{x_1, ..., x_n\} \sim \theta_1^*)$, then then θ_1^* has lower expected sharpness (higher robustness) than θ_2. Notice that θ_2 is referring to a local minimum of the data generated by the ground-truth parameter θ_1^*. Therefore, θ_1^* can have both high identifiability and high robustness.

Acknowledgments

We are grateful to all the researchers who generously shared their datasets, enabling the comparative analysis presented in this study. This work was supported by start-up funding awarded to RCW. We also acknowledge the use of the Advanced Computing Environment (PACE) at the Georgia Institute of Technology, which provided essential computational resources for this research.

Author contributions

Hua-Dong Xiong, Conceptualization, Formal analysis, Data curation, Investigation, Methodology, Project administration, Software, Visualization, Writing – original draft, Writing – review and editing, Conceived the work; Li Ji-An, Conceptualization, Methodology, Project administration, Writing – original draft, Writing – review and editing, Conceived the work; Marcelo G. Mattar, Supervision, Writing – review and editing; Robert C. Wilson, Resources, Supervision, Funding acquisition, Writing – review and editing.

References

Akam, T., Costa, R., and Dayan, P. (2015). Simple Plans or Sophisticated Habits? State, Transition and Learning Interactions in the Two-Step Task. *PLOS Computational Biology*, 11(12):e1004648.

Amari, S., Murata, N., Muller, K.-R., Finke, M., and Yang, H. (1997). Asymptotic statistical theory of overtraining and cross-validation. *IEEE Transactions on Neural Networks*, 8(5):985–996.

Bahrami, B. and Navajas, J. (2020). 4 Arm Bandit Task Dataset. Publisher: OSF.

Binz, M., Akata, E., Bethge, M., Brändle, F., Callaway, F., Coda-Forno, J., Dayan, P., Demircan, C., Eckstein, M. K., Éltető, N., Griffiths, T. L., Haridi, S., Jagadish, A. K., Ji-An, L., Kipnis, A., Kumar, S., Ludwig, T., Mathony, M., Mattar, M., Modirshanechi, A., Nath, S. S., Peterson, J. C., Rmus, M., Russek, E. M., Saanum, T., Scharfenberg, N., Schubert, J. A., Buschoff, L. M. S., Singhi, N., Sui, X., Thalmann, M., Theis, F., Truong, V., Udandarao, V., Voudouris, K., Wilson, R., Witte, K., Wu, S., Wulff, D., Xiong, H., and Schulz, E. (2024). Centaur: a foundation model of human cognition. arXiv:2410.20268 [cs].

Bjorck, J., Weinberger, K. Q., and Gomes, C. (2021). Understanding Decoupled and Early Weight Decay. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(8):6777–6785. Number: 8.

Busemeyer, J. R. and Diederich, A. (2010). *Cognitive Modeling*. SAGE. Google-Books-ID: R7KDF35g5LQC.

Busemeyer, J. R. and Myung, I. J. (1987). Resource allocation decision making in an uncertain environment. *Acta Psychologica*, 66(1):1–19.

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J., Sagun, L., and Zecchina, R. (2019). Entropy-SGD: biasing gradient descent into wide valleys*. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018. Publisher: IOP Publishing and SISSA.

Chen, C. S., Knep, E., Han, A., Ebitz, R. B., and Grissom, N. M. (2021). Sex differences in learning from exploration. *eLife*, 10:e69748. Publisher: eLife Sciences Publications, Ltd.

Dezfouli, A., Griffiths, K., Ramos, F., Dayan, P., and Balleine, B. W. (2019). Models that learn how humans learn: The case of decision-making and its disorders. *PLOS Computational Biology*, 15(6):e1006903.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. (2017). Sharp Minima Can Generalize For Deep Nets. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1019–1028. PMLR. ISSN: 2640-3498.

Dowman, R., Ritz, B., and Fowler, K. (2016). A connectionist modeling study of the neural mechanisms underlying pain's ability to reorient attention. *Cognitive, Affective, & Behavioral Neuroscience*, 16(4):689–708.

Dubois, M. and Hauser, T. U. (2022). Value-free random exploration is linked to impulsivity. *Nature Communications*, 13(1):4542. Publisher: Nature Publishing Group.

Dziugaite, G. K. and Roy, D. M. (2017). Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data. arXiv:1703.11008 [cs].

Enkavi, A. Z., Eisenberg, I. W., Bissett, P. G., Mazza, G. L., MacKinnon, D. P., Marsch, L. A., and Poldrack, R. A. (2019). Large-scale analysis of test–retest reliabilities of self-regulation measures. *Proceedings of the National Academy of Sciences*, 116(12):5472–5477. Publisher: Proceedings of the National Academy of Sciences.

Feng, S. F., Wang, S., Zarnescu, S., and Wilson, R. C. (2021). The dynamics of explore–exploit decisions reveal a signal-to-noise mechanism for random exploration. *Scientific Reports*, 11(1):3077.

Finnoff, W., Hergert, F., and Zimmermann, H. G. (1993). Improving model selection by nonconvergent methods. *Neural Networks*, 6(6):771–783.

Forstmann, B. U., Tittgemeyer, M., Wagenmakers, E.-J., Derrfuss, J., Imperati, D., and Brown, S. (2011). The Speed-Accuracy Tradeoff in the Elderly Brain: A Structural Model-Based Approach. *Journal of Neuroscience*, 31(47):17242–17249. Publisher: Society for Neuroscience Section: Articles.

Frank, M. (2014). Assessing cognitive models: visualization strategies.

Fukumizu, K., Yamaguchi, S., Mototake, Y.-i., and Tanaka, M. (2019). Semi-flat minima and saddle points by embedding neural networks to overparameterization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Gershman, S. J. (2018). Deconstructing the human algorithms for exploration. *Cognition*, 173:34–42.

Giallanza, T., Campbell, D., Cohen, J. D., and Rogers, T. T. (2024). An integrated model of semantics and control. *Psychological Review*.

Groman, S. M., Rich, K. M., Smith, N. J., Lee, D., and Taylor, J. R. (2018). Chronic Exposure to Methamphetamine Disrupts Reinforcement-Based Decision Making in Rats. *Neuropsychopharmacology*, 43(4):770–780. Publisher: Nature Publishing Group.

Gubri, M., Cordy, M., and Traon, Y. L. (2024). Going Further: Flatness at the Rescue of Early Stopping for Adversarial Example Transferability. arXiv:2304.02688 [cs].

Hans, A., Wen, Y., Jain, N., Kirchenbauer, J., Kazemi, H., Singhania, P., Singh, S., Somepalli, G., Geiping, J., Bhatele, A., and Goldstein, T. (2024). Be like a Goldfish, Don't Memorize! Mitigating Memorization in Generative LLMs. arXiv:2406.10209 [cs].

Hochreiter, S. and Schmidhuber, J. (1997). Flat Minima. *Neural Computation*, 9(1):1–42.

Hoffer, E., Hubara, I., and Soudry, D. (2017). Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ji-An, L., Benna, M. K., and Mattar, M. G. (2023). Automatic Discovery of Cognitive Strategies with Tiny Recurrent Neural Networks. Pages: 2023.04.12.536629 Section: New Results.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. (2017). On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. arXiv:1609.04836 [cs].

Kingma, D. P. and Ba, J. (2017). Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs].

Krogh, A. and Hertz, J. (1991). A Simple Weight Decay Can Improve Generalization. In *Advances in Neural Information Processing Systems*, volume 4. Morgan-Kaufmann.

Lee, D., Seo, H., and Jung, M. W. (2012). Neural Basis of Reinforcement Learning and Decision Making. *Annual review of neuroscience*, 35:287–308.

Li, X., Su, R., Chen, Y., and Yang, T. (2023). Optimal policy for uncertainty estimation concurrent with decision making. *Cell Reports*, 42(3):112232.

Loshchilov, I. and Hutter, F. (2019). Decoupled Weight Decay Regularization. arXiv:1711.05101 [cs].

Nelder, J. A. and Mead, R. (1965). A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313.

Neyshabur, B. (2017). Implicit Regularization in Deep Learning. arXiv:1709.01953 [cs].

Neyshabur, B., Bhojanapalli, S., Mcallester, D., and Srebro, N. (2017). Exploring Generalization in Deep Learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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

Pouget, P., Logan, G. D., Palmeri, T. J., Boucher, L., Paré, M., and Schall, J. D. (2011). Neural Basis of Adaptive Response Time Adjustment during Saccade Countermanding. *Journal of Neuroscience*, 31(35):12604–12612. Publisher: Society for Neuroscience Section: Articles.

Reed, E. J., Uddenberg, S., Suthaharan, P., Mathys, C. D., Taylor, J. R., Groman, S. M., and Corlett, P. R. (2020). Paranoia as a deficit in non-social belief updating. *eLife*, 9:e56345. Publisher: eLife Sciences Publications, Ltd.

Rescorla, R. A. and Wagner, A. R. (1972). A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and non-reinforcement. *Classical conditioning, Current research and theory*, 2:64–69. Publisher: Appleton-Century-Crofts.

Sadeghiyeh, H., Wang, S., Alberhasky, M. R., Kyllo, H. M., Shenhav, A., and Wilson, R. C. (2020). Temporal discounting correlates with directed exploration but not with random exploration. *Scientific Reports*, 10(1):4020. Publisher: Nature Publishing Group.

Saxe, A. M., McClelland, J. L., and Ganguli, S. (2013). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*.

Schultz, W., Dayan, P., and Montague, P. R. (1997). A Neural Substrate of Prediction and Reward. *Science*, 275(5306):1593–1599. Publisher: American Association for the Advancement of Science.

Schurr, R., Reznik, D., Hillman, H., Bhui, R., and Gershman, S. J. (2024). Dynamic computational phenotyping of human cognition. *Nature Human Behaviour*, pages 1–15. Publisher: Nature Publishing Group.

Shinn, M., Lam, N. H., and Murray, J. D. (2020). A flexible framework for simulating and fitting generalized drift-diffusion models. *eLife*, 9:e56938. Publisher: eLife Sciences Publications, Ltd.

Somerville, L. H., Sasse, S. F., Garrad, M. C., Drysdale, A. T., Abi Akar, N., Insel, C., and Wilson, R. C. (2017). Charting the expansion of strategic exploratory behavior during adolescence. *Journal of Experimental Psychology: General*, 146(2):155–164.

Steingroever, H., Fridberg, D. J., Horstmann, A., Kjome, K. L., Kumari, V., Lane, S. D., Maia, T. V., McClelland, J. L., Pachur, T., Premkumar, P., Stout, J. C., Wetzels, R., Wood, S., Worthy, D. A., and Wagenmakers, E.-J. (2015). Data from 617 Healthy Participants Performing the Iowa Gambling Task: A "Many Labs" Collaboration. *Journal of Open Psychology Data*, 3(1).

Suthaharan, P., Reed, E. J., Leptourgos, P., Kenney, J. G., Uddenberg, S., Mathys, C. D., Litman, L., Robinson, J., Moss, A. J., Taylor, J. R., Groman, S. M., and Corlett, P. R. (2021). Paranoia and belief updating during the COVID-19 crisis. *Nature Human Behaviour*, 5(9):1190–1202. Publisher: Nature Publishing Group.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

The MathWorks, Inc. (2024). MATLAB (R2024a).

Vallat, R. (2018). Pingouin: statistics in Python. *Journal of Open Source Software*, 3(31):1026.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, , Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., and van Mulbregt, P. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272. Publisher: Nature Publishing Group.

Waltz, J. A., Wilson, R. C., Albrecht, M. A., Frank, M. J., and Gold, J. M. (2020). Differential Effects of Psychotic Illness on Directed and Random Exploration. *Computational Psychiatry*, 4(0).

Wilson, R. C. and Collins, A. G. (2019). Ten simple rules for the computational modeling of behavioral data. *eLife*, 8:e49547.

Wilson, R. C., Geana, A., White, J. M., Ludvig, E. A., and Cohen, J. D. (2014). Humans use directed and random exploration to solve the explore–exploit dilemma. *Journal of Experimental Psychology: General*, 143(6):2074–2081.

Wilson, R. C., Shenhav, A., Straccia, M., and Cohen, J. D. (2019). The Eighty Five Percent Rule for optimal learning. *Nature Communications*, 10(1):4646.

Xie, Z., Sato, I., and Sugiyama, M. (2021). A Diffusion Theory For Deep Learning Dynamics: Stochastic Gradient Descent Exponentially Favors Flat Minima. arXiv:2002.03495 [cs].

Yechiam, E. and Busemeyer, J. R. (2005). Comparison of basic assumptions embedded in learning models for experience-based decision making. *Psychonomic Bulletin & Review*, 12(3):387–402.

Zajkowski, W. K., Kossut, M., and Wilson, R. C. (2017). A causal role for right frontopolar cortex in directed, but not random, exploration. *eLife*, 6:e27430. Publisher: eLife Sciences Publications, Ltd.