DEEP LEARNING IMPROVES PARAMETER ESTIMATION IN REINFORCEMENT LEARNING MODELS

Hua-Dong Xiong *
School of Psychology
Georgia Institute of Technology

Li Ji-An *
Neurosciences Graduate Program
University of California San Diego

Marcelo G. Mattar †
Department of Psychology
New York University

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

ABSTRACT

Reinforcement learning models are widely used in psychology and neuroscience to study cognitive processes underlying decision-making. However, accurately and reliably estimating model parameters is often challenging due to factors ubiquitous in cognitive modeling, such as limited data, measurement noise, and model complexity, hindering the interpretation of these parameters in behavioral and neural data. Here we evaluate whether the deep learning method — combining neural networks with modern optimization techniques — can improve parameter estimation compared to the Nelder–Mead method (`fminsearch`), a de facto baseline in cognitive modeling. We perform a systematic comparison using *ten* distinct value-based decision-making datasets, in which humans and animals perform various bandit tasks. Although both methods achieve comparable predictive performance, they yield different parameter estimates. Notably, parameters obtained through the deep learning method consistently exhibit smaller gaps between training and testing performance (improved generalizability), greater resilience to parameter perturbations (enhanced robustness), superior performance in recovering ground-truth parameters when data availability is constrained (better identifiability), and higher consistency across repeated measurements from the same subject (improved test-retest reliability). Our study highlights the critical importance of systematically assessing generalizability, robustness, and identifiability to ensure meaningful interpretations of cognitive model parameters and advocates for adopting the deep learning method for parameter estimation in cognitive modeling.

***K*eywords** Computational Modeling · Reinforcement Learning · Parameter Estimation · Deep Learning

1 Introduction

In neuroscience and psychology, computational models are widely used to investigate the cognitive processes that drive human and animal behavior. These models offer formalized accounts of how organisms learn from experience, make decisions, and adapt to their surrounding environment. A prominent modeling framework is reinforcement learning (RL), which describes how agents make decisions by learning values based on rewards and task-related feedback ([1]). Previous research has identified neural substrates involved in RL computations ([2]), providing evidence for the neural implementation of RL mechanisms.

These RL models typically employ a small number of free parameters (e.g., learning rates and inverse temperatures) to characterize the hypothesized cognitive strategies of agents. Therefore, accurate and reliable parameter estimation is essential for interpreting these parameters, and linking them to behavioral and neural data. However, achieving reliable parameter estimates is often challenging in practice: tasks may involve complex designs, models may exhibit

*Co-first authors.
†Co-last authors.

non-convex loss landscapes and single-subject data are often limited. Optimization methods also play a critical role by searching for parameter combinations — starting from an initial position in the parameter space — that maximize the model's predictive performance on subjects' behavior.

In practice, researchers commonly use the Nelder–Mead simplex method [3], the default optimization algorithm implemented in MATLAB's `fminsearch` function [4] and SciPy's `fmin` function [5]. Its off-the-shelf availability has led to widespread adoption across virtually every subfield of cognitive modeling (Given its ubiquity, we provide only a representative sample of its applications in cognitive modeling, such as: [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]).

In contrast, although the gradient-based optimization methods (e.g., those based on backpropagation-through-time) are widely used to optimize large machine learning models, they remain underutilized in cognitive modeling (but see [17, 18]).

Moreover, a systematic comparison of these parameter estimation methods across diverse experimental conditions is still lacking.

In this study, we evaluate these optimization methods for estimating parameters in RL models across ten behavioral datasets, where humans and animals perform bandit tasks featuring diverse actions, reward distributions, and time horizons. Specifically, we compare the Nelder–Mead method and a deep learning approach combining neural networks and modern optimization techniques. We find that although both optimization methods achieve similar predictive performance on held-out data, they produce different distributions of estimated parameters across subjects in most tasks. We systematically evaluate both optimization methods along three key dimensions: generalizability, robustness, and identifiability. We show that the deep learning method exhibits a smaller generalization gap, generates more robust parameter estimates, and demonstrates superior parameter recovery performance when data are scarce. Additionally, we show that the deep learning method achieves higher test-retest reliability in a dataset with multiple measurements from a single subject. Our results suggest that different optimization methods can lead RL parameter estimates to converge to different minima, impacting their generalizability, robustness, and identifiability. Furthermore, our findings indicate that neural networks with modern deep-learning pipelines offer 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 (Wilson 2014) [19]; (2) a 3-armed horizon task in humans, an extension of the previous horizon task with varying horizons and information availability (Dubois 2022) [20]; (3) a 2-armed restless bandit task in mice, featuring drifting reward probabilities for binary rewards (Chen 2021) [21]; (4) a 4-armed restless bandit task in humans with continuously drifting rewards (Bahrami 2020) [22]; (5) a 3-armed probabilistic reversal learning task in humans with binary rewards and dynamically changing reward probabilities (Suthaharan 2021) [23, 24]; (6) a 3-armed probabilistic reversal learning task in mice (Groman 2018) [25]; (7, 8) a 2-armed bandit task in humans with continuous rewards sampled from a Gaussian distribution and a horizon of 10 trials (Gershman 2018 and Schurr 2024) [11, 26]; (9) a 2-armed instrumental learning task in humans, where the two arms have different reward probabilities (Dezfouli 2019) [27]; (10) the Iowa Gambling Task in humans (Steingroever 2015) [28]. 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.

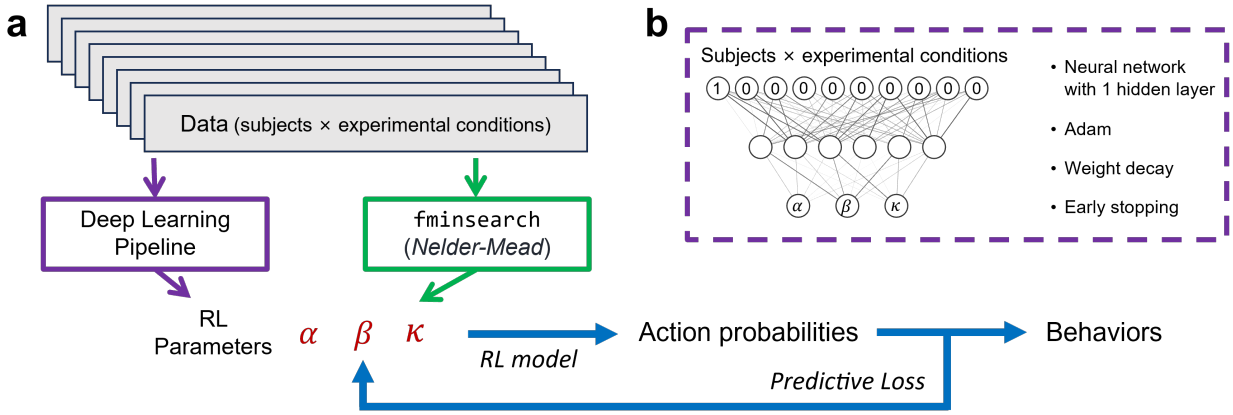

Figure 1: **Comparison of the Nelder-Mead method and the deep learning method. a**, Both approaches optimize parameters in reinforcement learning (RL) models for each subject and each experimental condition to predict their choice behavior. **b**, The deep learning method incorporates a neural network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

2.2 Deep learning method shows better generalizability

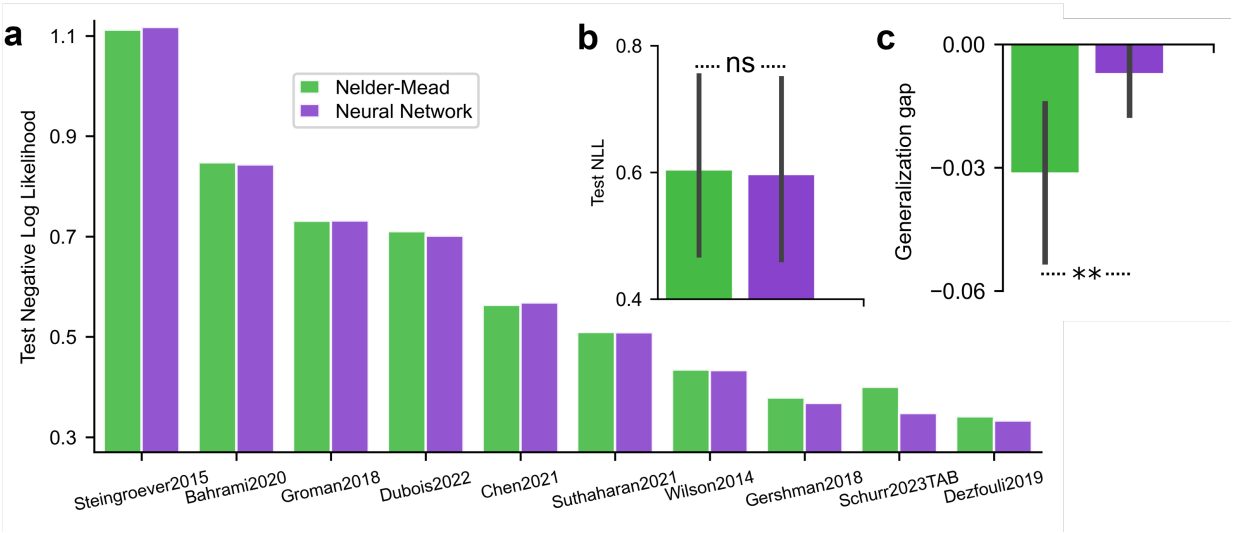

Figure 2: **Comparison of the Nelder–Mead and the deep learning methods across ten decision-making datasets. a**, Predictive performance of RL models optimized using both methods. The deep learning method achieves a negative log-likelihood (lower is better) comparable to that of the Nelder–Mead method 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. `**`: $p < 0.01$.

To compare the Nelder–Mead and deep learning methods for parameter estimation, we apply both methods 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 method (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 in the deep learning method" for details). The network parameters are optimized by the gradient-based optimizer AdamW [29] to minimize the cross-entropy between the models' predictions and subjects' choices. To improve generalization, we further incorporate weight-decay regularization [30, 31] and early stopping [32, 33]. 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).

We found that the Nelder–Mead and deep learning methods 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 [34] 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 method 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 method yields different parameter estimates

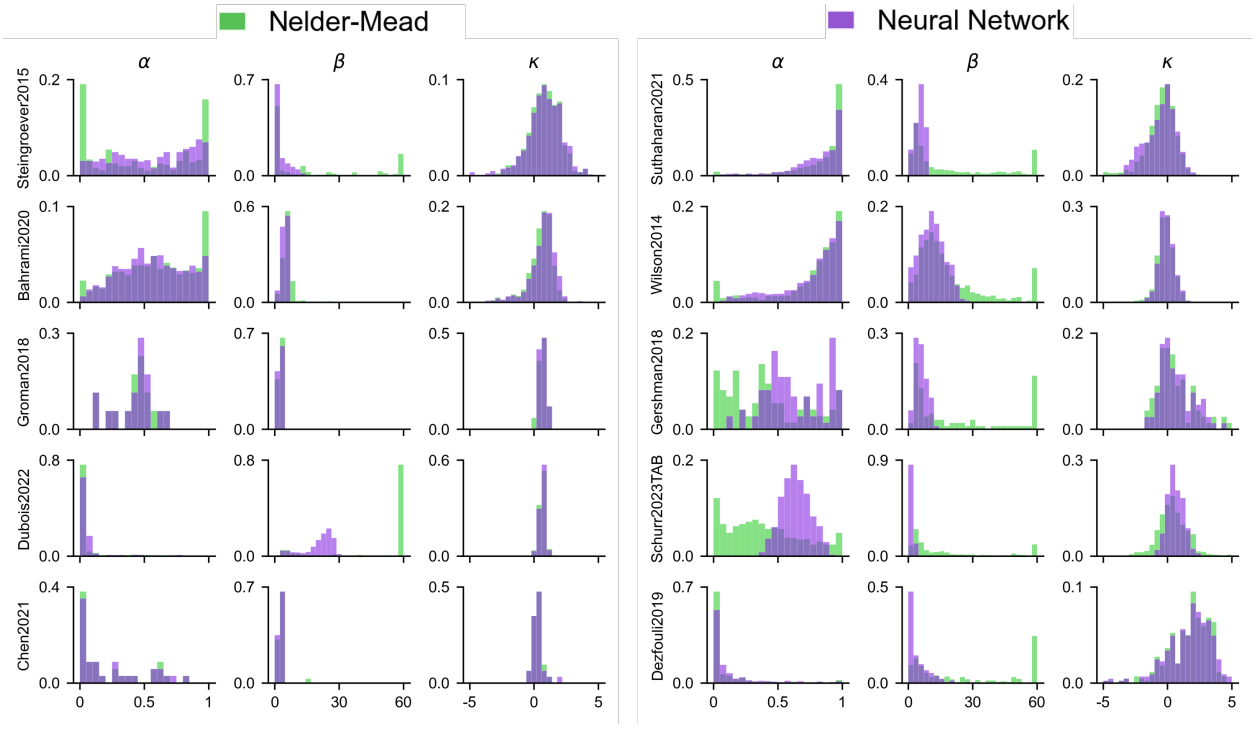

Figure 3: **Marginal distributions of RL parameters estimated by the Nelder–Mead and deep learning methods.** Each panel depicts the distribution of a single parameter (α, β, κ) across all subjects and experimental conditions within a dataset. The two methods produce distinct parameter distributions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

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 method tends to produce less extreme estimates (i.e., those on the boundary; e.g., see α

in Steingroever 2015 and β in Dubois 2022). This finding suggests that the two methods may find parameter estimates in different regions of the parameter space.

2.4 Deep learning method shows enhanced robustness

Given that the two methods produce different parameter estimates, we investigate whether and how these estimates differ in their properties. Specifically, we use the expected sharpness [35] to examine the models' robustness to parameter perturbations in two optimization methods. This sharpness metric quantifies the model's sensitivity relative to parameter perturbations, offering a characterization of the flatness of the loss landscape around the identified minima. Such "wide flat minima" are associated with improved generalization [36, 35, 37].

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

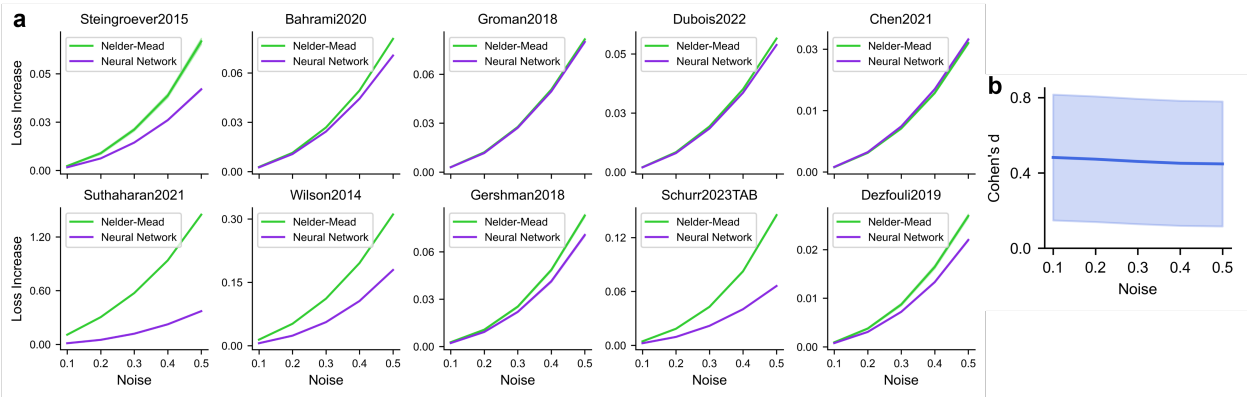

Figure 4: **Model robustness under parameter perturbations at varying noise levels. a,** Parameters estimated by the deep learning method exhibit smaller increases in loss in response to injected white Gaussian noise, compared to the Nelder–Mead method, meaning lower expected sharpness and greater robustness 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.

2.5 Deep learning method shows better identifiability

We have shown that the RL parameters estimated by the deep learning method exhibit greater generalizability and robustness. To meaningfully interpret these estimated parameters, a crucial criterion is their identifiability — specifically, whether the ground-truth model parameters can be reliably recovered from the simulated data (i.e., parameter recovery) [38]. In this process, simulated data are first generated using the model with a known set of parameters. The same model, with free parameters, is then fitted on the simulated data using the two optimization methods. If the estimated parameters match the ground-truth parameters that generate the data, the model is considered to have high identifiability. Conversely, if the ground-truth parameters cannot be reliably recovered from simulated data, it suggests that the parameters estimated from empirical data may not be interpretable in terms of the underlying cognitive processes. While identifiability is primarily determined by the task, data, model, and choice of ground-truth parameters, the optimization methods also play a critical role — some may locate the global minimum of the loss landscape more efficiently, thereby enhancing identifiability. To compare the identifiability of the two optimization methods, we selected four tasks that reflect a range of task structures (Fig. 5).

We first simulated varying amounts of data generated by RL models using 125 parameter combinations (5 values for each of the three parameters) and then assessed the parameter recovery performance using the correlation and mean absolute error between estimated and ground-truth parameters (see Methods "Parameter recovery"). We found that the parameter recovery performance of both methods improves as the number of trials per subject increases, reaching similar levels with sufficient data (3200 trials). However, the residual errors between estimated and ground-truth parameters remain significant in some tasks even with sufficient trials (e.g., Fig. 5b, 2-armed instructional learning in the third row), suggesting that the identifiability may be fundamentally limited by the task and model structure. Notably, the deep learning method outperformed the Nelder–Mead method in most tasks when the number of trials per subject is limited — a range comparable to typical sample sizes in psychological experiments (indicated by red and blue bars). This finding suggests that the deep learning method is preferable for datasets of typical sizes, as it more accurately recovers ground-truth parameters.

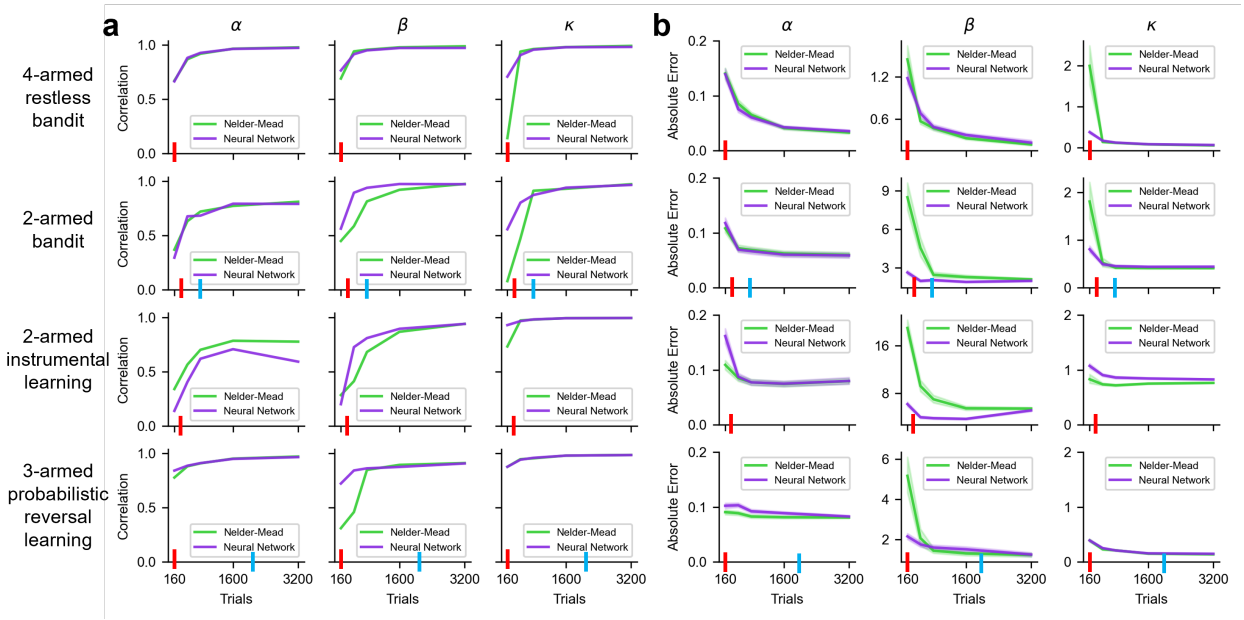

Figure 5: **Parameter recovery using the Nelder–Mead and deep learning methods across varying amounts of simulated data from an RL model.** The deep learning method achieves higher or comparable **(a)** correlations with ground-truth parameters and **(b)** lower or comparable absolute errors across most 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 - Bahrami 2020); 2-armed bandit (red - Gershman 2018, Schurr 2023TAB; blue - Wilson 2014); 2-armed instrumental learning (red - Dezfouli 2019); and 3-armed probabilistic reversal learning (red - Suthaharan 2021; blue - Groman 2018).

2.6 Deep learning method improves test-retest reliability

We have shown that the deep learning method demonstrates superior generalizability, robustness, and identifiability. But how do these properties translate into more direct metrics of interest in psychology and cognitive science? To address this, we focus on test-retest reliability, a core concept in psychology, which assesses the consistency of scores for the same subjects tested under similar conditions at different time points. We analyzed a unique dataset, Schurr2023TAB ([26]), in which subjects performed a two-armed bandit task over 12 consecutive weeks, enabling us to measure the correlation between multiple assessments across different weeks for the same subjects. Following the exclusion criteria and methodology from the original study ([26]), 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) of the two methods differed by less than 0.02 (Fig. 6a; NLL quantile: Q1 = -0.006, Q3 = 0.012). The following results remain consistent when using different threshold values.

Despite controlling for fitting performance, the two methods produced distinct parameter distributions (Fig. 6b). The deep learning method also demonstrates greater robustness to parameter perturbations (Fig. 6c). Crucially, the deep learning method yielded significantly higher ICC values (Fig. 6d) for the estimated RL parameters (β: $t(40) = 3.25, p = .002$, Cohen's $d = 0.52$; κ: $t(40) = 5.10, p < .001$, Cohen's $d = 0.97$); with the exception of α $t(40) = 0.71, p = .481$, Cohen's $d = 0.06$), indicating stronger test-retest reliability. These findings suggest that the deep learning method can extract more consistent parameter estimates from different measurements of the same subjects.

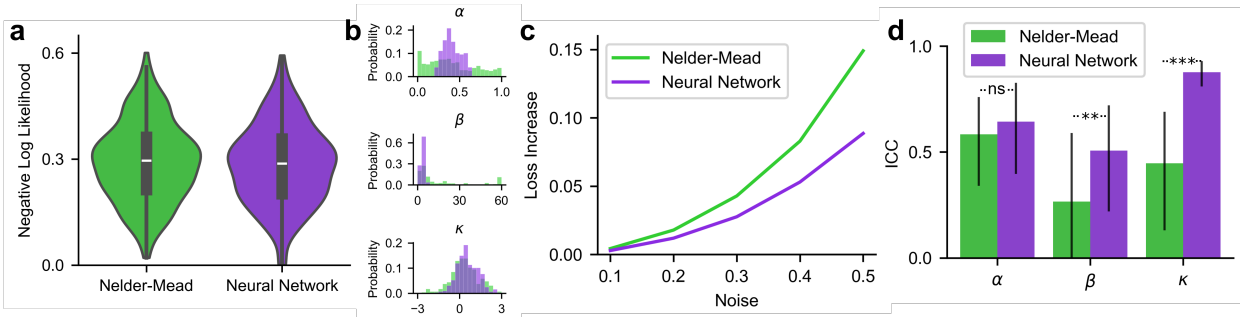

Figure 6: **Comparing the Nelder–Mead and the deep learning method on the Schurr 2023TAB dataset with multiple measurements from the same subjects.** A subset of subjects is selected to control for fitting performance (see the main text). **a**, Both methods exhibit similar distributions of fitting performance. **b**, Two methods yield different parameter distributions. **c**, The deep learning method shows greater robustness to parameter perturbations. The shaded area (mostly invisible) represents the 95% confidence interval across 10,000 simulations. **d**, The deep learning method demonstrates 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. Our work systematically examined two classes of optimization methods for parameter estimation in RL models across ten value-based decision-making datasets, focusing on three key dimensions: generalizability, robustness, and identifiability. While the de facto Nelder–Mead (`fminsearch`) and deep learning methods achieve comparable fitting performance on held-out test data, they surprisingly produce distinct parameter distributions. Furthermore, the deep learning method exhibited a smaller generalization gap, lower sensitivity to parameter noise, and superior parameter recovery in low-data settings. Crucially, the deep learning method substantially improved test-retest reliability, a key determinant of replicability in cognitive research. A key contribution of this work is demonstrating that optimization methods can significantly influence estimated cognitive parameters in diverse datasets. To the best of our knowledge, this study provides the first systematic evaluation of how different optimization methods affect parameter estimation across diverse datasets in cognitive modeling.

Generalizability refers to a model's ability to perform well on unseen held-out data, and ideally, new conditions. Better generalization indicates a model has captured deeper underlying mechanisms that more effectively summarize the data. Our study found that both methods achieved similar test performance across the ten datasets examined, suggesting a lack of significant advantage in test performance for the deep learning method, possibly due to the simplicity of the RL model formulation used in our study. However, we found that the generalization gap – the difference in performance between training and test data, is smaller for the deep learning method, suggesting a better generalizability in a weaker sense. We expect that the deep learning method will demonstrate greater advantages in test performance when applied to more complex cognitive models in higher dimensional tasks, given the fundamental success of deep neural networks in learning from high-dimensional data.

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 method exhibit lower sharpness (i.e., greater flatness). Although such flatness is rarely evaluated in cognitive models (but see [39]), the connection between "wide flat minima" and improved generalization is well established in the machine learning literature (Fig. 7) [36, 40, 41, 42, 37]. Intuitively, when the loss function is highly sensitive

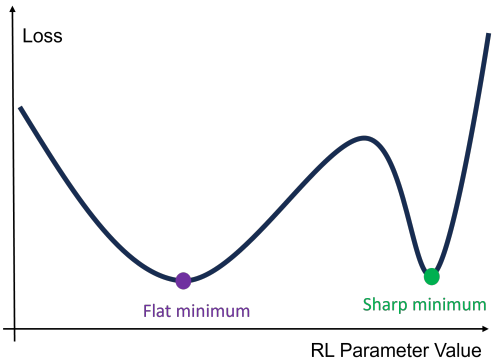

Figure 7: **Illustration of parameter flatness and sharpness.** In a non-convex loss landscape, different parameter estimates may produce similar losses. flatter minima, where loss remains stabler under small parameter perturbations, are generally considered better solutions and are often associated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

to parameter perturbation 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 stochastic optimizers act as entropy maximizers, favoring solutions with broader support in parameter space [43]. They have been theoretically shown to favor flat over sharp minima, resulting in more generalizable solutions in deep neural networks ([37]). In addition, other components of the deep learning method, such as overparameterization and regularization techniques (e.g., weight decay and early stopping), may also introduce implicit biases that facilitate 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 neural networks' generalization error ([44, 45, 46, 47, 37]. In summary, the deep learning method we used improves the reliability of parameter estimation in cognitive models.

The parameter recovery test provides a measure of model identifiability, by examining whether the ground-truth parameters can be accurately recovered from simulated data generated using these parameters. High identifiability indicates that the estimated parameter values serve as faithful representations of the underlying mechanisms. We demonstrate that the deep learning method achieves superior parameter recovery across four widely used decision-making tasks, particularly in the low-data regime typical of human behavioral experiments. Such 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 method 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 demonstrate that the deep learning method significantly improves test-retest reliability, which measures the consistency of parameter estimates across multiple measurements. Owing to its improved generalizability, robustness, and identifiability, the higher test-retest reliability suggests that the parameters are both reliable and stable over time, thereby providing more accurate characterizations of the underlying mechanisms that generate the observed behavioral data — a critical requirement for longitudinal cognitive research.

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

In conclusion, our study systematically compares two optimization methods. The deep learning method, leveraging modern optimization techniques, provides more generalizable, robust, and identifiable 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 ([19]) 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 ([19, 48, 49, 50, 51, 52]) with n_{subjects} = 609 and n_{trials} = 754,890.

A three-armed version of this horizon task is presented in [20]. 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 ([21]) 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 ([22]) 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 ([23]) 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 ([25]), 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 ([11]) 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. ([26]) 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 ([27]) 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 ([28]) 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 in the deep learning method

The neural network we used 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 [53]. 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 method, we adopted a nested cross-validation approach ([54]). 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 [11, 26], we partition data into different blocks. For datasets with multiple short blocks of varying trial counts (each block containing fewer than 40 trials) [19, 20], 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 [22, 23, 24, 28], we split the data into non-overlapping trial sets for training, validation, and testing, applying masking as needed [55]. For datasets with few blocks but more than 200 trials [21, 25], 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 method, 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 method, 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:

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

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 ([38]). 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 ([22]); $\alpha \in [0.3, 0.8]$, $\beta \in [4, 15]$, and $\kappa \in [-0.5, 1.5]$ for the two-armed bandit task ([11, 26]); $\alpha \in [0.05, 0.35]$, $\beta \in [1, 20]$, and $\kappa \in [0, 3]$ for the two-armed instrumental learning task ([27]); 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 ([23, 24, 25]).

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 [11]. 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 [56, 57]. 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

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

[2] Wolfram Schultz, Peter Dayan, and P. Read Montague. A Neural Substrate of Prediction and Reward. *Science*, 275(5306):1593–1599, March 1997. Publisher: American Association for the Advancement of Science.

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

[4] The MathWorks, Inc. MATLAB (R2024a), 2024.

[5] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, March 2020. Publisher: Nature Publishing Group.

[6] Thomas Akam, Rui Costa, and Peter Dayan. Simple Plans or Sophisticated Habits? State, Transition and Learning Interactions in the Two-Step Task. *PLOS Computational Biology*, 11(12):e1004648, December 2015.

[7] Jerome R. Busemeyer and In Jae Myung. Resource allocation decision making in an uncertain environment. *Acta Psychologica*, 66(1):1–19, October 1987.

[8] Jerome R. Busemeyer and Adele Diederich. *Cognitive Modeling*. SAGE, 2010. Google-Books-ID: R7KDF35g5LQC.

[9] Robert Dowman, Benjamin Ritz, and Kathleen Fowler. A connectionist modeling study of the neural mechanisms underlying pain's ability to reorient attention. *Cognitive, Affective, & Behavioral Neuroscience*, 16(4):689–708, August 2016.

[10] Birte U. Forstmann, Marc Tittgemeyer, Eric-Jan Wagenmakers, Jan Derrfuss, Davide Imperati, and Scott Brown. The Speed-Accuracy Tradeoff in the Elderly Brain: A Structural Model-Based Approach. *Journal of Neuroscience*, 31(47):17242–17249, November 2011. Publisher: Society for Neuroscience Section: Articles.

[11] Samuel J. Gershman. Deconstructing the human algorithms for exploration. *Cognition*, 173:34–42, April 2018.

[12] Xiaodong Li, Ruixin Su, Yilin Chen, and Tianming Yang. Optimal policy for uncertainty estimation concurrent with decision making. *Cell Reports*, 42(3):112232, March 2023.

[13] Pierre Pouget, Gordon D. Logan, Thomas J. Palmeri, Leanne Boucher, Martin Paré, and Jeffrey D. Schall. Neural Basis of Adaptive Response Time Adjustment during Saccade Countermanding. *Journal of Neuroscience*, 31(35):12604–12612, August 2011. Publisher: Society for Neuroscience Section: Articles.

[14] Maxwell Shinn, Norman H Lam, and John D Murray. A flexible framework for simulating and fitting generalized drift-diffusion models. *eLife*, 9:e56938, August 2020. Publisher: eLife Sciences Publications, Ltd.

[15] Robert C. Wilson, Amitai Shenhav, Mark Straccia, and Jonathan D. Cohen. The Eighty Five Percent Rule for optimal learning. *Nature Communications*, 10(1):4646, December 2019.

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

[17] Marcel Binz, Elif Akata, Matthias Bethge, Franziska Brändle, Fred Callaway, Julian Coda-Forno, Peter Dayan, Can Demircan, Maria K. Eckstein, Noémi Éltető, Thomas L. Griffiths, Susanne Haridi, Akshay K. Jagadish, Li Ji-An, Alexander Kipnis, Sreejan Kumar, Tobias Ludwig, Marvin Mathony, Marcelo Mattar, Alireza Modirshanechi, Surabhi S. Nath, Joshua C. Peterson, Milena Rmus, Evan M. Russek, Tankred Saanum, Natalia Scharfenberg, Johannes A. Schubert, Luca M. Schulze Buschoff, Nishad Singhi, Xin Sui, Mirko Thalmann, Fabian Theis, Vuong Truong, Vishaal Udandarao, Konstantinos Voudouris, Robert Wilson, Kristin Witte, Shuchen Wu, Dirk Wulff, Huadong Xiong, and Eric Schulz. Centaur: a foundation model of human cognition, November 2024. arXiv:2410.20268 [cs].

[18] Tyler Giallanza, Declan Campbell, Jonathan D. Cohen, and Timothy T. Rogers. An integrated model of semantics and control. *Psychological Review*, July 2024.

[19] Robert C. Wilson, Andra Geana, John M. White, Elliot A. Ludvig, and Jonathan D. Cohen. Humans use directed and random exploration to solve the explore–exploit dilemma. *Journal of Experimental Psychology: General*, 143(6):2074–2081, 2014.

[20] Magda Dubois and Tobias U. Hauser. Value-free random exploration is linked to impulsivity. *Nature Communications*, 13(1):4542, August 2022. Publisher: Nature Publishing Group.

[21] Cathy S Chen, Evan Knep, Autumn Han, R Becket Ebitz, and Nicola M Grissom. Sex differences in learning from exploration. *eLife*, 10:e69748, November 2021. Publisher: eLife Sciences Publications, Ltd.

[22] Bahador Bahrami and Joaquin Navajas. 4 Arm Bandit Task Dataset, August 2020. Publisher: OSF.

[23] Praveen Suthaharan, Erin J. Reed, Pantelis Leptourgos, Joshua G. Kenney, Stefan Uddenberg, Christoph D. Mathys, Leib Litman, Jonathan Robinson, Aaron J. Moss, Jane R. Taylor, Stephanie M. Groman, and Philip R. Corlett. Paranoia and belief updating during the COVID-19 crisis. *Nature Human Behaviour*, 5(9):1190–1202, September 2021. Publisher: Nature Publishing Group.

[24] Erin J Reed, Stefan Uddenberg, Praveen Suthaharan, Christoph D Mathys, Jane R Taylor, Stephanie Mary Groman, and Philip R Corlett. Paranoia as a deficit in non-social belief updating. *eLife*, 9:e56345, May 2020. Publisher: eLife Sciences Publications, Ltd.

[25] Stephanie M. Groman, Katherine M. Rich, Nathaniel J. Smith, Daeyeol Lee, and Jane R. Taylor. Chronic Exposure to Methamphetamine Disrupts Reinforcement-Based Decision Making in Rats. *Neuropsychopharmacology*, 43(4):770–780, March 2018. Publisher: Nature Publishing Group.

[26] Roey Schurr, Daniel Reznik, Hanna Hillman, Rahul Bhui, and Samuel J. Gershman. Dynamic computational phenotyping of human cognition. *Nature Human Behaviour*, pages 1–15, February 2024. Publisher: Nature Publishing Group.

[27] 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, June 2019.

[28] Helen Steingroever, Daniel J. Fridberg, Annette Horstmann, Kimberly L. Kjome, Veena Kumari, Scott D. Lane, Tiago V. Maia, James L. McClelland, Thorsten Pachur, Preethi Premkumar, Julie C. Stout, Ruud Wetzels, Stacey Wood, Darrell A. Worthy, and Eric-Jan Wagenmakers. Data from 617 Healthy Participants Performing the Iowa Gambling Task: A "Many Labs" Collaboration. *Journal of Open Psychology Data*, 3(1), June 2015.

[29] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017. arXiv:1412.6980 [cs].

[30] Anders Krogh and John Hertz. A Simple Weight Decay Can Improve Generalization. In *Advances in Neural Information Processing Systems*, volume 4. Morgan-Kaufmann, 1991.

[31] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization, January 2019. arXiv:1711.05101 [cs].

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

[33] William Finnoff, Ferdinand Hergert, and Hans Georg Zimmermann. Improving model selection by nonconvergent methods. *Neural Networks*, 6(6):771–783, January 1993.

[34] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima, February 2017. arXiv:1609.04836 [cs].

[35] Behnam Neyshabur, Srinadh Bhojanapalli, David Mcallester, and Nati Srebro. Exploring Generalization in Deep Learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[36] Sepp Hochreiter and Jürgen Schmidhuber. Flat Minima. *Neural Computation*, 9(1):1–42, January 1997.

[37] Zeke Xie, Issei Sato, and Masashi Sugiyama. A Diffusion Theory For Deep Learning Dynamics: Stochastic Gradient Descent Exponentially Favors Flat Minima, January 2021. arXiv:2002.03495 [cs].

[38] Robert C Wilson and Anne GE Collins. Ten simple rules for the computational modeling of behavioral data. *eLife*, 8:e49547, November 2019.

[39] Michael Frank. Assessing cognitive models: visualization strategies, April 2014.

[40] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp Minima Can Generalize For Deep Nets. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1019–1028. PMLR, July 2017. ISSN: 2640-3498.

[41] Elad Hoffer, Itay Hubara, and Daniel Soudry. 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., 2017.

[42] Behnam Neyshabur. Implicit Regularization in Deep Learning, September 2017. arXiv:1709.01953 [cs].

[43] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-SGD: biasing gradient descent into wide valleys*. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018, December 2019. Publisher: IOP Publishing and SISSA.

[44] Johan Bjorck, Kilian Q. Weinberger, and Carla Gomes. Understanding Decoupled and Early Weight Decay. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(8):6777–6785, May 2021. Number: 8.

[45] Gintare Karolina Dziugaite and Daniel M. Roy. Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data, October 2017. arXiv:1703.11008 [cs].

[46] Kenji Fukumizu, Shoichiro Yamaguchi, Yoh-ichi Mototake, and Mirai Tanaka. Semi-flat minima and saddle points by embedding neural networks to overparameterization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[47] Martin Gubri, Maxime Cordy, and Yves Le Traon. Going Further: Flatness at the Rescue of Early Stopping for Adversarial Example Transferability, February 2024. arXiv:2304.02688 [cs].

[48] Samuel F. Feng, Siyu Wang, Sylvia Zarnescu, and Robert C. Wilson. The dynamics of explore–exploit decisions reveal a signal-to-noise mechanism for random exploration. *Scientific Reports*, 11(1):3077, February 2021.

[49] Leah H. Somerville, Stephanie F. Sasse, Megan C. Garrad, Andrew T. Drysdale, Nadine Abi Akar, Catherine Insel, and Robert C. Wilson. Charting the expansion of strategic exploratory behavior during adolescence. *Journal of Experimental Psychology: General*, 146(2):155–164, February 2017.

[50] James A. Waltz, Robert C. Wilson, Matthew A. Albrecht, Michael J. Frank, and James M. Gold. Differential Effects of Psychotic Illness on Directed and Random Exploration. *Computational Psychiatry*, 4(0), August 2020.

[51] Hashem Sadeghiyeh, Siyu Wang, Maxwell R. Alberhasky, Hannah M. Kyllo, Amitai Shenhav, and Robert C. Wilson. Temporal discounting correlates with directed exploration but not with random exploration. *Scientific Reports*, 10(1):4020, March 2020. Publisher: Nature Publishing Group.

[52] Wojciech K Zajkowski, Malgorzata Kossut, and Robert C Wilson. A causal role for right frontopolar cortex in directed, but not random, exploration. *eLife*, 6:e27430, September 2017. Publisher: eLife Sciences Publications, Ltd.

[53] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[54] Li Ji-An, Marcus K. Benna, and Marcelo G. Mattar. Automatic Discovery of Cognitive Strategies with Tiny Recurrent Neural Networks, May 2023. Pages: 2023.04.12.536629 Section: New Results.

[55] Abhimanyu Hans, Yuxin Wen, Neel Jain, John Kirchenbauer, Hamid Kazemi, Prajwal Singhania, Siddharth Singh, Gowthami Somepalli, Jonas Geiping, Abhinav Bhatele, and Tom Goldstein. Be like a Goldfish, Don't Memorize! Mitigating Memorization in Generative LLMs, June 2024. arXiv:2406.10209 [cs].

[56] A. Zeynep Enkavi, Ian W. Eisenberg, Patrick G. Bissett, Gina L. Mazza, David P. MacKinnon, Lisa A. Marsch, and Russell A. Poldrack. Large-scale analysis of test–retest reliabilities of self-regulation measures. *Proceedings of the National Academy of Sciences*, 116(12):5472–5477, March 2019. Publisher: Proceedings of the National Academy of Sciences.

[57] Raphael Vallat. Pingouin: statistics in Python. *Journal of Open Source Software*, 3(31):1026, November 2018.