PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Harnessing the flexibility of neural networks to predict dynamic theoretical parameters underlying human choice behavior

Yoav Ger[1]*, **Eliya Nachmani**[2,3], **Lior Wolf**[4], **Nitzan Shahar**[1,5]

1 Sagol School of Neuroscience, Tel-Aviv University, Tel-Aviv, Israel, **2** School of Electrical Engineering, Tel-Aviv University, Tel-Aviv, Israel, **3** Meta AI Research, Tel-Aviv, Israel, **4** Blavatnik School of Computer Science, Tel-Aviv University, Tel-Aviv, Israel, **5** School of Psychological Sciences, Tel-Aviv University, Tel-Aviv, Israel

* yoavger@mail.tau.ac.il

Citation: Ger Y, Nachmani E, Wolf L, Shahar N (2024) Harnessing the flexibility of neural networks to predict dynamic theoretical parameters underlying human choice behavior. PLoS Comput Biol 20(1): e1011678. https://doi.org/10.1371/journal.pcbi.1011678

Editor: Lusha Zhu, Peking University, CHINA

Received: April 24, 2023

Accepted: November 12, 2023

Published: January 4, 2024

Data Availability Statement: All code for reproducing the results in this paper is available at https://github.com/yoavger/harnessing_the_flexibility_of_nn_to_predict_dynamical.

Funding: This work is supported by the Israeli Science Foundation (Grant 2536/20 awarded to NS), the Tel Aviv University Center for AI and Data Science (TAD, awarded to LW) and the Israeli Science Foundation (grant No. 2923/20, awarded to LW) within the Israel Precision Medicine Partnership program. The funders had no role in

Abstract

Reinforcement learning (RL) models are used extensively to study human behavior. These rely on normative models of behavior and stress interpretability over predictive capabilities. More recently, neural network models have emerged as a descriptive modeling paradigm that is capable of high predictive power yet with limited interpretability. Here, we seek to augment the expressiveness of theoretical RL models with the high flexibility and predictive power of neural networks. We introduce a novel framework, which we term theoretical-RNN (t-RNN), whereby a recurrent neural network is trained to predict trial-by-trial behavior and to infer theoretical RL parameters using artificial data of RL agents performing a two-armed bandit task. In three studies, we then examined the use of our approach to dynamically predict unseen behavior along with time-varying theoretical RL parameters. We first validate our approach using synthetic data with known RL parameters. Next, as a proof-of-concept, we applied our framework to two independent datasets of humans performing the same task. In the first dataset, we describe differences in theoretical RL parameters dynamic among clinical psychiatric vs. healthy controls. In the second dataset, we show that the exploration strategies of humans varied dynamically in response to task phase and difficulty. For all analyses, we found better performance in the prediction of actions for t-RNN compared to the stationary maximum-likelihood RL method. We discuss the use of neural networks to facilitate the estimation of latent RL parameters underlying choice behavior.

Author summary

Currently, neural network models fitted directly to behavioral human data are thought to dramatically outperform theoretical computational models in terms of predictive accuracy. However, these networks do not provide a clear theoretical interpretation of the mechanisms underlying the observed behavior. Generating plausible theoretical explanations for observed human data is a major goal in computational neuroscience. Here, we provide a proof-of-concept for a novel method where a recurrent neural network (RNN)

study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

is trained on artificial data generated from a known theoretical model to predict both trial-by-trial actions and theoretical parameters. We then freeze the RNN weights and use it to predict both actions and theoretical parameters of empirical data. We first validate our approach using synthetic data where the theoretical parameters are known. We then show, using two empirical datasets, that our approach allows dynamic estimation of latent parameters while providing better action predictions compared to theoretical models fitted with a maximum-likelihood approach. This proof-of-concept suggests that neural networks can be trained to predict meaningful time-varying theoretical parameters.

Introduction

A fundamental goal in neuroscience is to describe the underlying computations that animals and human agents might deploy when deliberating between actions. In pursuit of this goal, researchers have used reinforcement learning (RL) models which seek to describe behavior through a concise set of interpretable parameters that are believed to correspond to specific underlying processes (i.e., learning rate, exploration rate, etc.) [1–3]. While these models have generated important findings [4–6], they usually fit behavioral data poorly since they typically adopt a normative approach, focusing on how behavior should be rather than learning a generative model from the data itself [7, 8]. A more recent descriptive approach, in which neural networks are fitted directly to behavioral data, has been shown to exceed RL models in action prediction [9]. Neural networks are extremely flexible and can be used across a wide range of data variants and experimental environments [10, 11]. However, despite neural networks' high predictive performance, they do not provide a clear theoretical interpretation, which is a major goal of brain and behavior research [12, 13]. Alternatively, neural networks could be employed to enhance and refine the parameter estimation for existing theoretical models, where the interpretation of parameters is more readily understandable.

Reinforcement learning models have proven effective in understanding and predicting decision-making behavior in both animals and humans [4, 14–16]. This method involves breaking down behavior into a parameterized computational model that can be fitted to individual choices [1, 2]. The individuals' parameters are then viewed as low-dimensional representation of behavioral data, which provides insight into the latent cognitive configuration that might have generated the observed data. However, fitting RL models to behavioral observations can be challenging as it involves prolonged data collection in order to fit complicated and non-convex functions using iterative maximum-likelihood approximations [17]. To manage this complexity, constraints such as assuming stationary behavior [1] (i.e., fixed parameters) or rationality [8] (i.e., optimality) are imposed, eventually leading to lower predictive accuracy.

Recently, an alternative approach to studying decision-making has gained much attention, involving fitting neural network models directly to behavioral data [9, 18–20]. Unlike theoretical RL models that rely on assumptions about behavior, neural networks require minimal assumptions and can learn complex features from the data without human intervention. This data-driven approach has yielded high predictive accuracy on various behavioral tasks [9, 18]. Although some researchers have used neural network modeling beyond prediction to enhance our theoretical understanding of behavior [19, 21], the challenge of how to utilize the flexibility of neural networks to better interpret behavior remains open.

Here, we present a novel framework we term theoretical-RNN (henceforth, t-RNN), which aims to combine the strengths of RL models with the flexible and predictive power of neural

networks. This is done by first simulating many artificial RL agents performing a two-armed bandit task, and then training an RNN using the trial-by-trial artificial data to predict both the agents' theoretical RL parameters and future actions from observed action-reward history (see Fig 1C). Moreover, the t-RNN recurrent structure allows it to estimate time-varying RL parameters, enabling it to track changes in behavior over time (see Fig 1E).

We start by validating our method using synthetic behavior (for which we know the ground truth) and show that at test time, without further optimization, the network successfully recovers unseen agents' RL parameters. We then present a proof-of-concept using two existing empirical datasets [9, 22], where human individuals performed a two-armed bandit task. For both datasets, we used t-RNN that was initially trained using synthetic data. Then, the network's weights were fixed, and we applied it to predict individuals' actions and to infer trial-by-trial theoretical RL parameters. In the first dataset, we utilized t-RNN to describe higher volatility in theoretical parameters for psychiatric individuals compared with healthy individuals, a group known for unstable internal mental states [23]. In the second dataset, we illustrated that t-RNN was able to capture dynamical changes in individuals' directed and random exploration strategies [24]. Remarkably, across both datasets, we showcased that t-RNN produces

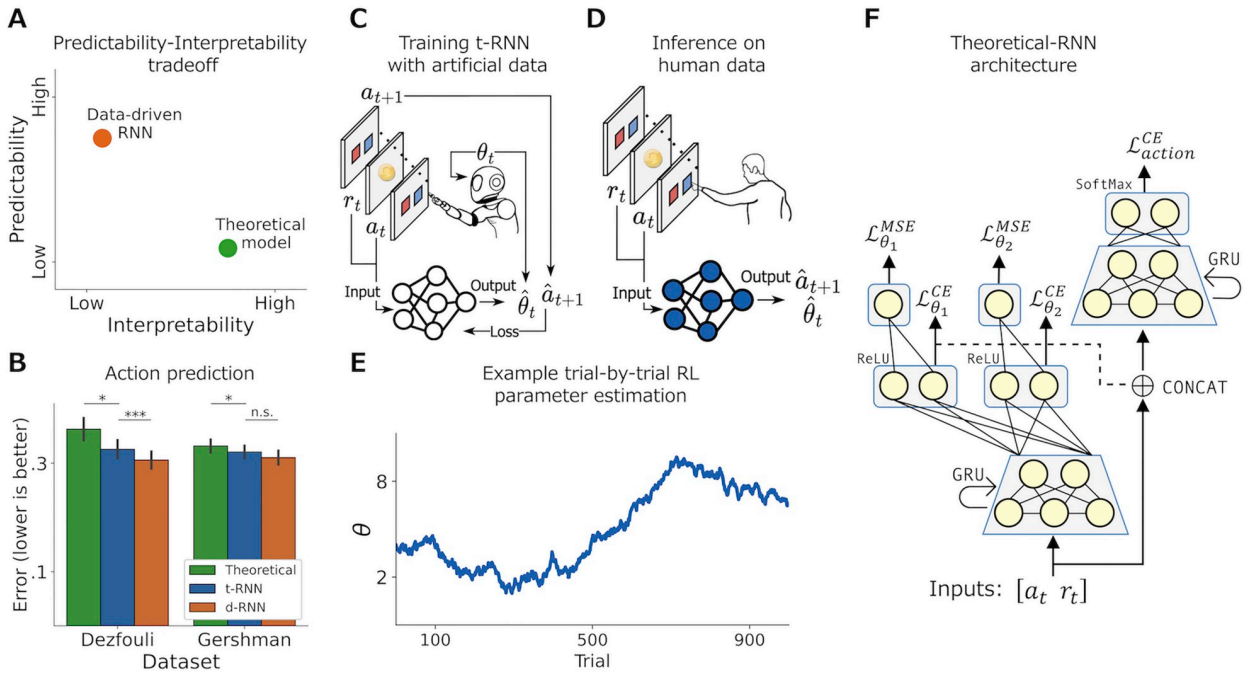

Fig 1. Summary schematic for theoretical-RNN. (A) Illustration of the predictability-interpretability trade-off plane. Theoretical models (green) are thought to be highly interpretable (x-axis), yet their predictive capability (y-axis) is often limited compared to data-driven RNN (d-RNN; orange). We suggest a new way to train a network (t-RNN) that allows dynamic estimation of theoretical RL parameters. **(B)** Empirical results for the analysis of two publicly available datasets where human individuals performed a two-armed bandit task [9, 22]. We present a comparison of action prediction made by d-RNN (orange), t-RNN (blue), and stationary theoretical modeling (green). Overall, we found that t-RNN was able to produce better predictions compared to stationary theoretical modeling fitted using a maximum-likelihood approach (y-axis illustrates error measured with binary cross-entropy; lower is better; black lines indicate s.e.m). **(C)** Illustration of t-RNN training procedure. We first simulated many artificial RL agents performing a two-armed bandit task with known RL parameters. We then familiarized t-RNN with the mapping between the action and reward (a_t, r_t) in trial t, to the action of the next trial a_{t+1} and the theoretical RL parameters in the current trial θ_t. **(D)** Estimating humans RL parameters. Post-training, we froze the network weights and predicted the future action $\hat{a}_{t+1}$, as well as estimated the dynamical RL parameters $\hat{\theta}_t$, of humans performing a similar two-armed bandit task. **(E)** Illustration of t-RNN's ability to estimate time-varying RL parameters, enabling it to track changes in behavior over time. **(F)** t-RNN architecture consists of several fully connected (FC) layers and gated recurrent units (GRU), such that the action prediction and RL parameters estimation are carried out with two different heads. Significant effect at * $p < .05$, *** $p < .001$; n.s., a nonsignificant effect (Wilcoxon signed rank test).

https://doi.org/10.1371/journal.pcbi.1011678.g001

parameter estimations that are meaningful while maintaining a level of predictive accuracy that is higher than theoretical models using a maximum-likelihood estimation approach [1]. Overall, we contend that our method can be used as a general framework, applicable to various theoretical models of behavior and datasets, and we discuss its advantages in enhancing the expressiveness of theoretical models to capture complex non-stationary behavior.

Results

In the current study, we present a novel approach to simultaneously predict human behavioral choices and generate trial-by-trial estimations of theoretical RL parameters using neural networks. To demonstrate this approach, we use the well-known two-armed bandit task widely used in human neuroscience to investigate cognitive and neural processes. Initially, we train a t-RNN on artificial data obtained from theoretical RL agents performing the task and then keep the weights fixed to predict unseen behavior. We validate the effectiveness of our approach using synthetic data, followed by two use cases where we demonstrate t-RNN's ability to predict actions and infer time-varying theoretical RL parameters in empirical datasets where human individuals performed the same task.

Theoretical recurrent neural network (t-RNN) framework

Here we present a proposed method we termed t-RNN and describe how the network is trained to infer underlying theoretical RL parameters from behavioral observation of RL agents engaged in a two-armed bandit.

Two-armed bandit task. We present our model in the context of a generic two-armed bandit task widely used decision-making task to study behavior [1, 15]. At each trial of time t, the agent is forced to choose between two possible actions, a (denoted L and R). Next, a stochastic reward, r, is delivered. The reward probability of each action is fixed during a block of B trials and selected randomly at the beginning of each block from a pre-existing setting (see Materials and methods for more details).

t-RNN architecture. Our model as depicted in Fig 1F, is based on a recurrent neural network (RNN) consisting of fully connected (FC) layers and gated recurrent units (GRU; [25]). At each step, the inputs to the t-RNN are the two observed time-dependent variables, the action a_t and the reward r_t, and it produces two outputs: currents estimate of the agent RL parameters $\hat{\Theta}_t$ (Θ denotes a vector of RL parameters; subscripts t denotes time-varying RL parameters), and prediction of the future action $\hat{a}_{t+1}$.

t-RNN training. The two outputs that t-RNN produces, namely the hidden RL parameters and future action, are processed through different loss terms. The RL parameters are predicted in two stages. First as a multiclass classification task, after employing quantization of each RL parameter to a discretized space (i.e., $\mathbb{R} \rightarrow \mathbb{N}$; see Materials and methods), and then as a regression problem. It is important to note that quantization was used only to stabilize the network training, while the network's final output remains continuous. This procedure is based on previous work [26, 27] which has shown that MSE loss is much harder to optimize and often converges to a fixed output that reflects the mean. To ensure that quantization did not affect our performance in any meaningful way, we ran multiple experiments in which we varied the number of the RL parameters quantization and found similar results (see S1 Table).

In the first stage, a_t (coded using one-hot transformation) and r_t is passed through a single layer of 32 GRU cells and the output is projected to $m = |\Theta|$ FC layers with ReLU activation to obtain the network intermediate prediction of the categorical class of each RL parameter. We

then compute a cross-entropy loss (denoted as $\mathcal{L}_{\theta_i}^{CE}$) with regard to the true class for each RL parameter separately.

Next, to allow a continuous recovery of each RL parameter, in the second stage, the network categorical class prediction (i.e., the logits) is passed through m additional regression heads (each FC layer of size 1; each FC layer is responsible for a different RL parameter). We compute the mean-square error (denoted as $\mathcal{L}_{\theta_i}^{MSE}$) with regard to the true continuous values of each RL parameter separately.

In addition to estimating the parameters, our network also predicts the future action, a_{t+1}. For this aim, we concatenate the network categorical class predictions (i.e., the logits) together with a_t and r_t and passed this vector through a second single layer of 32 GRU cells. We project the output to an FC layer with SoftMax activation to obtain a probability distribution over possible future actions and computed a cross-entropy loss (denoted as $\mathcal{L}_{action}^{CE}$) with regards to the true future action a_{t+1}.

In S6 Fig, we provide an additional experiment to examine the necessity of concatenation and its impact on the results. To investigate this, we employed ablation techniques and trained a version of the t-RNN model without concatenation.

We now define the combined loss function used to train the network,

$$\mathcal{L}_{total} = \mathcal{L}_{action}^{CE} + \sum_{i\in\{\Theta\}} \lambda_{\theta_i}^{CE}\mathcal{L}_{\theta_i}^{CE} + \sum_{i\in\{\Theta\}} \lambda_{\theta_i}^{MSE}\mathcal{L}_{\theta_i}^{MSE}, \tag{1}$$

where λ^{CE} and λ^{MSE} are hyperparameters, set early during the development process such that each loss term is approximately of equal magnitude (see Materials and methods for further details).

Validating t-RNN using synthetic behavior

We first demonstrate our method on simulated data of RL agents for which the ground truth RL parameters are known. Specifically, we trained t-RNN to predict the actions and parameters of a theoretical RL model. We then generated test agents on which the network was not trained with and compared the performance of t-RNN, to both stationary theoretical RL model (using maximum-likelihood [1]) and Bayesian particle filtering [28].

Theoretical RL model. To generate a training set of t-RNN, we simulated Q-learning [29, 30] agents performing a two-armed bandit task. Specifically, for the Q-learning agents, at each experience of time t, the agent updates the Q-values (initialized to zero at the beginning of each block) of the action selected a_t using the obtained reward r_t by,

$$Q_{t+1}(a_t) \leftarrow Q_t(a_t) + \alpha\overbrace{(r_t - Q_t(a_t))}^{\delta_{TD}}, \tag{2}$$

where $0 \leq \alpha \leq 1$ is a learning-rate free-parameter, which determines the extent to which newly acquired information overrides old information, and δ_{TD} is the temporal-difference (TD) error that drives learning.

These Q-values are in turn transformed to a stochastic choice-rule policy in accordance with the Boltzmann distribution,

$$P(a) = \frac{\exp \beta Q(a)}{\sum_{\tilde{a}\in A} \exp \beta Q(\tilde{a})}, \tag{3}$$

where $\beta \geq 0$ is the inverse-temperature free-parameter, which determines the randomness of action selection.

Training t-RNN was done with a synthetic training set of 2,000 simulated Q-learner agents operating with Eqs 2 and 3. Each agent performed a two-armed bandit task for 10 blocks of 100 trials each, and the agents' RL parameters were sampled from a uniform distribution. Importantly, to stabilize the training procedure, we applied two techniques. First, to allow for continuous recovery of each RL parameter, we performed a quantization preprocessing step, where we discretized each RL parameter into evenly spaced bins, which acted as an intermediate prediction target during training. Second, to facilitate t-RNN's flexibility to recover time-varying RL parameters, we simulated agents with non-stationary parameters. For each agent, there was a small probability per trial that the agent's parameters would be re-sampled from the same uniform distribution (see Materials and methods for more details).

For comparison purposes, we included two additional baseline methods for estimating hidden RL parameters. The first baseline method was a stationary Q-learning model (denoted Q-stationary), in which stationary RL parameters were estimated for each agent individually, using a maximum-likelihood estimation. The second baseline method was Bayesian particle filtering (denoted Bayesian) adopted from [28]. Here, underlying RL parameters are assumed to drift slowly according to a Gaussian random walk and are estimated for each agent on a trial-by-trial basis using Bayesian particle filtering (see Materials and methods for more details).

Results. To validate our method, after training t-RNN, we simulated 30 additional test agents (Materials and methods) on which the network was not trained and recorded the predictions made by t-RNN as well as the two baseline methods. First, we calculated the binary cross-entropy (BCE) between the true actions and predicted actions of each of the three models (stationary Q-learning, Bayesian, and t-RNN). We found that t-RNN had the lowest error compared to the two other models ($p < .01$ when comparing Q-stationary and t-RNN using Wilcoxon signed rank test, and $p < .05$ when comparing Bayesian and t-RNN; see BCE_{action} in Fig 2A;). Second, we calculated the mean-squared error (MSE) between the true and estimated

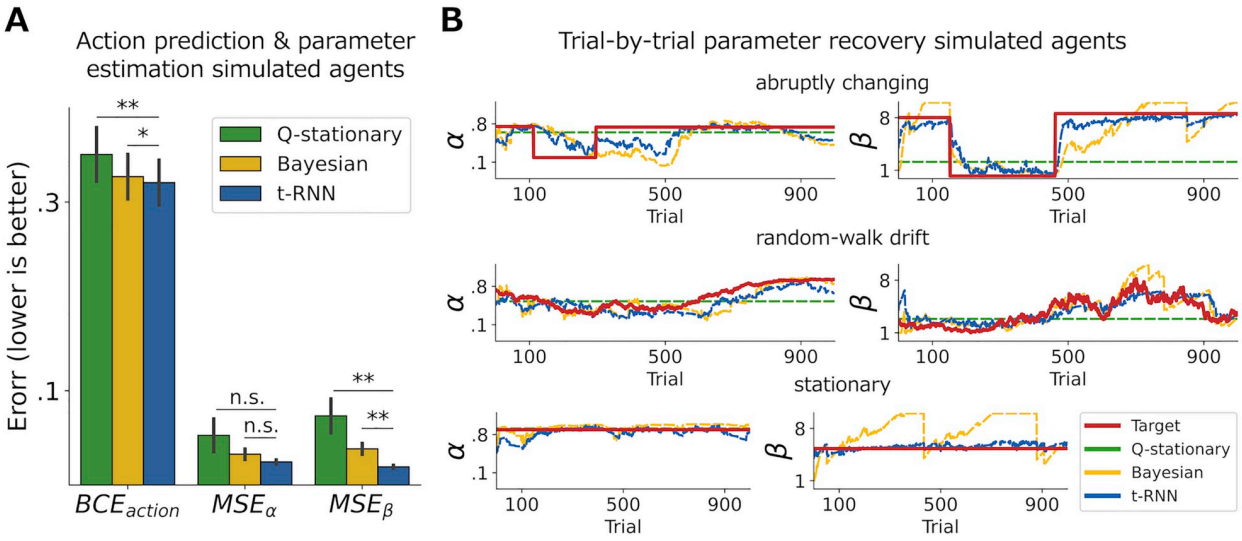

Fig 2. Validating the ability of t-RNN to predict both agents' actions and to infer RL parameters of simulated data. (A) Action prediction (measured with binary cross entropy; BCE) and RL parameter estimation error (measured with mean-square error; MSE) averaged across 30 artificial test agents (black lines indicate s.e.m). We found that our method exceeded alternative methods (Q-stationary [Maximum-likelihood], green; Bayesian [particle filtering], yellow) in both action prediction (left) and parameter estimation (middle and right). **(B)** Recovery of theoretical RL α learning-rate (left) and β inverse-temperature (right) parameters of three example simulated agents with different RL parameter trajectories. These trajectories included abruptly changing (top), random-walk drift (middle), and stationary (bottom) RL parameters. Overall, we found that our network (blue dotted line) was able to effectively recover the true dynamic RL parameters (red solid line). This demonstrates the versatile ability of t-RNN to accurately capture a wide range of time-varying RL parameter dynamics. Significant effect at $^{*}\ p < .05$, $^{**}\ p < .01$; n.s., a nonsignificant effect (Wilcoxon signed rank test).

https://doi.org/10.1371/journal.pcbi.1011678.g002

RL parameters and found that t-RNN outperformed the two alternatives methods in terms of MSE_β ($p < .01$ when comparing Q-stationary and Bayesian to t-RNN using Wilcoxon signed rank test; see Fig 2A). Note that although t-RNN exhibited the lowest MSEα compared to both baseline methods, it was not significant. This finding confirmed that our network can accurately recover hidden RL parameters of test agents operating with various underlying RL parameter dynamics (see S2 Table for raw results).

To further examine and illustrate the t-RNN capabilities, we plotted (see Fig 2B) the true and recovered RL parameters of three types of test agents: abruptly changing RL parameters (top), gradually changing RL parameters (middle), and agents with stationary RL parameters (bottom). Visual inspection of Fig 2B indicates that t-RNN generated predictions that closely matched the true RL parameter trajectories. Overall, these results indicate t-RNN flexible ability to recover underlying RL parameters while keeping high predictive accuracy. Additionally, we examine the recovery of agents simulated in a volatile environment, where we observed that t-RNN maintains superior performance compared to both baseline methods (see S3 Table). We also examine the ability of t-RNN to estimate stable parameters compared to the stationary Q-learning method as a function of the number of observations (see S1 Text). Our findings demonstrate that t-RNN performs well for stable parameters and is comparable to the stationary method when more than one parameter is included in the theoretical model. Moreover, for a low number of observations (below 50 trials), t-RNN outperforms the stationary estimation.

Using t-RNN to interpret behavior of psychiatric individuals

After confirming the ability of our method to recover hidden RL parameters with a variety of test agents, we next present an application of t-RNN to an empirical behavioral dataset collected by Dezfouli et al., 2019 [9] where 33 individuals diagnosed with bipolar disorder, 34 with depression and 34 individuals who were matched healthy controls (total of $N = 101$) performed two-armed bandit task (see Materials and methods for more details).

Theoretical RL model. To model subjects' behavior (and to train t-RNN), we consider the same RL Q-learning model as mentioned above (see Eqs 2 and 3) with an additional action preservation free-parameter added to the SoftMax choice-rule policy,

$$P(a) = \frac{\exp \beta(Q(a) + \kappa_t(a))}{\sum_{\tilde{a} \in A} \exp \beta(Q(\tilde{a}) + \kappa_t(\tilde{a}))}, \qquad \kappa_t(a) = \begin{cases} \kappa & \text{if } a = a_{t-1} \\ 0 & \text{otherwise} \end{cases}, \tag{4}$$

where $-0.5 \leq \kappa \leq 0.5$ is an action preservation free-parameter determining the agent's tendency to stick with (or switch) a previous action regardless of the reward [31].

Results. Following training of the t-RNN with a synthetic training set of Q-learner agents using Eqs 2 and 4, we fixed the network's weights and recorded the predictions made by the t-RNN on the behavioral dataset of humans who performed a two-armed bandit task [9]. Additionally, we fitted a stationary Q-learning with preservation model (denoted QP-stationary) to each subject, which estimated stationary RL parameters via maximum-likelihood estimation, as well as a Bayesian model fitted with particle filtering. Furthermore, we fitted a data-driven RNN (d-RNN) to subject behavior, as described in [9], which solely produced action predictions (see Materials and methods for more details).

As expected, we found that t-RNN was able to maintain a higher action prediction of human behavior compared to the QP-stationary model but not compared to d-RNN (see Fig 1B; $p < .05$ when comparing QP-stationary and t-RNN using Wilcoxon signed rank test, and $p < .0005$ when comparing t-RNN and d-RNN). To further explore this result, we divided the

subjects according to their diagnostic groups and examined the action predictions within each group. We found that t-RNN superiority over QP-stationary and Bayesian models was mainly due to the improved action prediction of bipolar and depressed individuals ($p < .01$ comparing QP-stationary and Bayesian to t-RNN in the bipolar group; $p < .01$ comparing QP-stationary to t-RNN and $p < .001$ comparing Bayesian to t-RNN in the depressed group; Wilcoxon signed rank test), while there was no significant difference in its prediction accuracy for healthy controls ($p = 0.36$ comparing QP-stationary to t-RNN; $p = 0.07$ comparing Bayesian to t-RNN; see Fig 3A and S4 Table for the raw results).

We next turn to explore the dynamic estimation of the theoretical parameters. First, we assessed the volatility in κ preservation estimates estimated trial-by-trial using t-RNN. Our results revealed that the bipolar group exhibited lower values and greater volatility in these estimates compared to the healthy group ($M = 0.15$, $SD = 0.30$ for the bipolar group, $M = 0.35$, $SD = 0.14$ for the healthy group; see Fig 3B). This finding might correspond with the clinical nature of the disorder where volatility is a main feature [23]. However, for the aim of the current study, this is mainly a proof-of-concept to the t-RNN's ability to inform us regarding

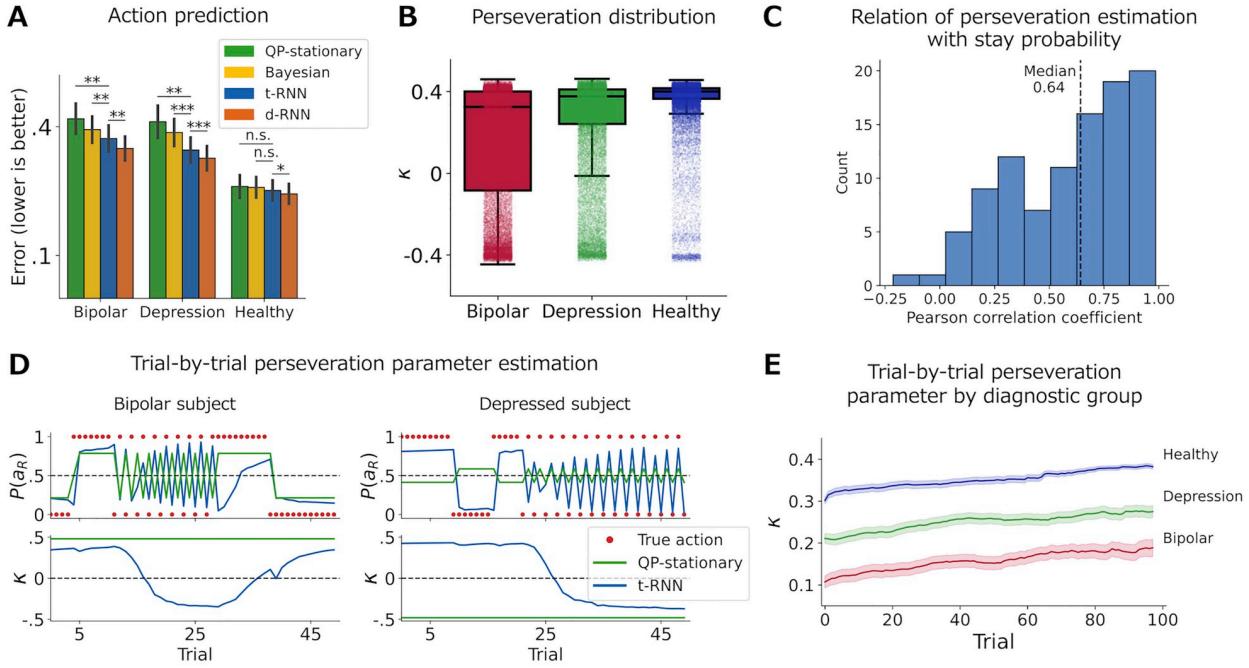

Fig 3. Application of t-RNN to behavioral data of psychiatric and non-psychiatric individuals [9]. (A) Action prediction (measured with BCE; black lines indicate s.e.m) divided by diagnostic labels. Bipolar and depressed subjects at the group level are explained significantly better by t-RNN (blue) compared to the QP-stationary model (green) that assumes that the RL parameters are fixed, as well as to the Bayesian model (yellow). **(B)** Boxplot of the time-varying κ preservation parameter estimates by t-RNN across the three diagnostic groups (middle black solid lines denote the median; light dots indicate a single trial estimate of the κ preservation parameter). Result suggests higher volatility in the clinical groups mainly the bipolar group, compared with the healthy group. **(C)** Distribution of the Pearson correlation between stay probability (calculated using moving average of 10 trials) and time-varying κ preservation parameter produced by t-RNN for each subject individually. Result shows a strong relation between t-RNN time-varying RL κ parameters estimation and moving average of the stay probability. **(D)** Sequence of selected actions of two example subjects, with bipolar on the left and depressed on the right. The top panel shows the action prediction, and the bottom panel shows the RL κ parameter estimation produced by t-RNN (blue) and QP-stationary model (green) for a 50-trial segment (see S4 Fig for all trials). Both subjects seemed to switch between selecting the same action for several trials to repeatedly alternating between both actions every trial (red dots). We found that the t-RNN model was able to detect this visually apparent shift in behavior, estimating a change in the κ action perseveration parameter. In contrast, the QP-stationary model was not able to capture this transition. **(E)** Trial-by-trial κ preservation parameter estimates by t-RNN averaged over the first 100 trials of each block and for each diagnostic group separately (shaded area signifies the s.e.m). In all three groups, subjects show a steady increase in their tendency to perseverate their actions as the block progresses. Significant effect at * $p < .05$, ** $p < .01$, *** $p < .001$; n.s., a nonsignificant effect (Wilcoxon signed rank test).

https://doi.org/10.1371/journal.pcbi.1011678.g003

volatility in underlying parameters, while also providing better action predictions compared with a stationary QP-stationary model. In S5 Table, we show additional distribution of all RL parameters produced by t-RNN.

Second, to gain insight into the association between RL parameter estimation and observed behavior, we selected two clinical participants with bipolar disorder and depression whose behavior displayed clear transitions between high (sticking with the same action) and low (switching the action selection every trial) levels of perseveration. We plotted the sequence of selected actions alongside the action predictions and inferred κ estimates generated by t-RNN, as well as the predictions of the QP-stationary model for a 50-trial segment (see Fig 3D and S4 Fig for all trials). We found that the observed change in action perseveration was closely coupled with a change in the κ parameter, and the t-RNN model accurately detected the visually apparent shift in behavior. In contrast, the stationary model, which assumes fixed RL parameters, was unable to capture this transition, resulting in lower action prediction accuracy. These results highlight the utility of t-RNN in capturing dynamic changes in RL parameters and their association with observed behavior.

Third, to further support that t-RNN parameters estimation indeed corresponds well with changes in observed behavior across all individuals, we turn to a well-established and easy-to-interpret model-agnostic measurement and calculated a moving average of the stay probabilities (probability of repeating the previous action; window size of 10 trials). If indeed the t-RNN κ perseverate dynamic parameter is doing a good job of capturing fluctuations in perseveration, we should expect a high correlation with the moving average model-agnostic estimate. We calculated a Pearson correlation between the moving average and t-RNN κ perseverate trial-by-trial estimation for each individual (see Fig 3C). Overall, we found that the average Pearson across individuals was very high (Median = 0.64; see S1 Fig for further detailed examples). We also included a similar analysis with the β inverse-temperature parameter (see S2 and S3 Figs).

Finally, we plotted the trial-by-trial κ perseveration average for each group, showing that overall, across groups individuals increased their tendency to perseverate with the previous action selection which might reflect an overall reduction motivation as the block progresses (see Fig 3E). Overall, we were able to show that t-RNN is able to produce trial-by-trial estimates of RL parameters, while also keeping higher predictive accuracy compared with QP-stationary model. In the current dataset, we provided a proof-of-concept on how t-RNN might be able to inform of higher volatility in RL parameters in clinical psychiatric groups.

Using t-RNN to interpret exploration behavior of human individuals

To demonstrate the generality of our proposed method, we present a second application of t-RNN to empirical behavioral dataset collected by Gershman, 2018 [22] where $N = 44$ humans performed two-armed bandit task to examine human exploration strategies.

Theoretical RL model. To model subjects' behavior, we consider the same model hybrid exploration model presented in Gershman, 2018 [22]. The learning model included a Kalman filtering algorithm [15, 32] where agents recursively compute the posterior mean and variance for each action by,

$$Q_{t+1}(a_t) \leftarrow Q_t(a_t) + K_t(r_t - Q_t(a_t)), \tag{5}$$

$$\sigma^2_{t+1}(a_t) \leftarrow \sigma^2_t(a_t) - K_t\sigma^2_t(a_t), \tag{6}$$

where $Q_{t+1}(a)$ and $\sigma^2_{t+1}(a)$ is the posterior mean and variance estimates over the value of action

a, respectively and K_t is the Kalman gain given by,

$$K_t = \frac{\sigma_t^2(a_t)}{\sigma_t^2(a_t) + \tau_t^2(a_t)}, \tag{7}$$

where $\tau_t^2(a)$ is the error variance. At the beginning of each block the initial values estimate for both actions were set to the prior mean, namely, $Q_0(a) = 0$, $\sigma_0^2(a) = 100$ and $\tau_0^2(a) = 10$ (see Materials and methods for more details).

Action values estimate were transformed to action policy with a hybrid of random and directed exploration [24] choice-rule,

$$P(a = L) = \Phi\left(\beta \frac{Q_t(L) - Q_t(R)}{\sqrt{\sigma_t^2(L) + \sigma_t^2(R)}} + \gamma[\sigma_t(L) - \sigma_t(R)]\right), \tag{8}$$

where Φ is the cumulative distribution function (CDF) of the standard Gaussian distribution (i.e., probit), $0 \leq \beta \leq 4$ is a free-parameter that controls the contributions of random exploration (similar in spirit to the inverse-temperature parameter of the SoftMax choice-rule of Eq 3) and $0 \leq \gamma \leq 1$ is a free-parameter that controls the contributions of directed exploration.

This model can also be interpreted as a combination of Thompson sampling [33] and UCB policies [34], where the first term in the brackets promotes a random exploration that is proportional to the difference in the value estimates, $Q_t(L) - Q_t(R)$ together with the total uncertainty $\sqrt{\sigma_t^2(L) + \sigma_t^2(R)}$ much like Thompson sampling. The second term in the brackets, $\sigma_t(L) - \sigma_t(R)$, is the so-called relative uncertainty, synonyms with UCB policy which promotes a directed exploration toward the less certain option (i.e., information bonus).

Results. After the training of the t-RNN with a synthetic training set of Q-learner agents using Eqs 5, 6, 7 and 8, we fixed the network's weights and recorded the predictions made by the t-RNN on the behavioral dataset of humans who performed a two-armed bandit task [22]. Additionally, we fitted a stationary hybrid exploration model, a Bayesian model, and a d-RNN to subject behavior (Materials and methods). We found significantly higher action prediction for t-RNN compared to both the stationary hybrid exploration model ($p < .05$; Wilcoxon signed rank test) and the Bayesian model ($p < .01$), but no significant difference in prediction accuracy when comparing t-RNN to d-RNN ($p = 0.12$; see Fig 4A and S6 Table for raw results).

To investigate the trial-by-trial dynamics of the t-RNN's theoretical parameter estimation, we conducted a manual search of the data to identify an instance in which an individual made an ungreedy action. Specifically, we located a trial where an individual chose option L despite having been rewarded for option R for the previous four consecutive trials. We observed a sharp decrease in the t-RNN's β random exploration parameter estimation, indicating its sensitivity to changes in observed behavior (see Fig 4B). To determine whether this pattern generalizes across individuals and trials, we aggregated all trials (a total of 32 trials) in which individuals selected the better rewarding arm for three or more consecutive trials, were then rewarded, and subsequently switched to the alternative, poorer arm. Our analysis revealed an overall decrease in the β random exploration parameter produced by t-RNN in these cases (mean ± s.e.m. before: 2.1 ± 0.079, after: 1.40 ± 0.080; see Fig 4C). In S5 Fig, we provide a similar example for the γ directed exploration parameter.

Next, we investigated the overall dynamics of the two exploration parameters (random and directed) within blocks. Specifically, we examined blocks where the true expected value difference between the two arms was above or equal to 19 (referred to as "easy" blocks; total of 91 blocks) and blocks where the true expected value difference between the two arms was below or equal to 1 (referred to as "hard" blocks; total of 110 blocks). It is worth noting that

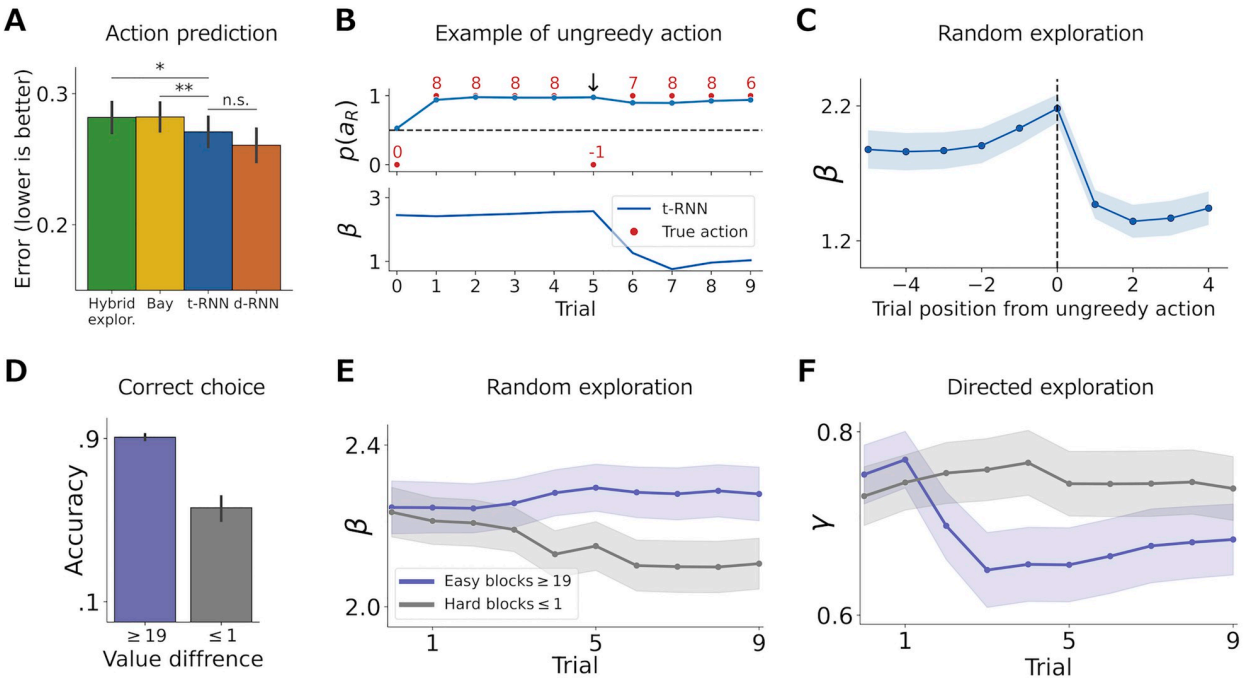

Fig 4. Application of t-RNN to investigate changes in random and directed exploration [22]. (A) Action prediction (measured with binary cross entropy; BCE; black lines indicate s.e.m). We found a significant improvement in the action prediction of the t-RNN model (blue) compared to the hybrid exploration model (green; * $p < .05$) and the Bayesian (yellow; ** $p < .01$). **(B)** Example of a single instantiation of ungreedy action. The top panel depicts the sequence of chosen actions, obtain rewards, and t-RNN action predictions as red dots, red numbers, and blue solid lines, respectively. The subject consecutively chooses the higher value action and suddenly switches and chooses the lower value action (denoted with a downward arrow). The bottom panel depicts t-RNN β random exploration parameter estimation, where it estimates a step decrease in the parameter after the ungreedy action selection. **(C)** Dynamics of the random exploration parameter averaged across all instantiations of ungreedy action. t-RNN estimates a sharp and sudden decrease in the random exploration parameter after an ungreedy action is taken. **(D)** Correct choice rate of individuals' actions divided by blocks where the true value difference is high ($\geq$ 19; easy blocks; purple) and blocks where the true value difference is low ($\leq$ 1; hard blocks; grey). Subjects are more accurate when it is easy to distinguish the correct action compared to when it is hard. **(E)** Dynamics of the β random exploration parameter estimation produced by t-RNN average overall easy (purple) and hard (grey) blocks. Random exploration increases in hard blocks (grey) where it is difficult to tell apart which action is best, whereas in the easy blocks (purple) random exploration decreases as subjects become more deterministic in their action selection. **(F)** Dynamics of the γ directed exploration parameter average overall easy (purple) and hard (grey) blocks. Directed exploration is high at the beginning of the block where uncertainty is equal in both conditions, however only in the easy blocks (purple) where it is easy to differentiate between the better/worst arm, directed exploration decreases as the block evolves. In panel C, E and F solid line/dots signifies the mean estimation and the shaded area correspond to s.e.m.

https://doi.org/10.1371/journal.pcbi.1011678.g004

participants were not informed whether a block was easy or hard and were required to start with a similar strategy in the first trial where the difficulty of the block was unknown. We hypothesized that participants would change their exploration strategy across time, and predicted that in easy blocks, individuals would reduce random exploration since the best action is easy to detect, whereas, in hard blocks, individuals would show a higher tendency for random exploration (indicating either noisy behavior due to block difficulty or a desire to test the arms in a more heterogeneous fashion to gain more information, where the information gain is relatively small for the participant).

We found that participants had a higher correct choice rate in easy blocks compared to hard blocks (mean ± s.e.m. easy: 0.90 ± 0.007, hard: 0.56 ± 0.003; see Fig 4D). Consistent with our hypothesis, t-RNN estimated an increase in random exploration (i.e., a decrease in the β parameter) across hard blocks and a decrease across easy blocks (see Fig 4E). We also observed the same pattern for the γ directed exploration, where overall directed exploration was high at the beginning of the block, possibly reflecting the individual's lack of knowledge about the

environment. However, there was a selective decrease in directed exploration only for easy blocks, where the best action was easy to locate (see Fig 4F).

Our results indicate that individuals modify their exploration strategies based on both the block difficulty and phase, as evidenced by changes in both random and directed exploration parameters. Importantly, t-RNN was successful in capturing these nuanced changes in exploration strategies, thus providing a second proof of the utility of our method.

Discussion

In this work, we introduce the notion of a "theoretical-RNN" (i.e., t-RNN), where neural networks are utilized to augment the expressiveness of theoretical models of behavior. Specifically, we train an RNN on artificial data simulated from a theoretical model, freeze the network weights, and use the network to predict trial-by-trial theoretical RL parameters and future actions. The immediate benefit of this approach is its ability to infer non-stationary trial-by-trial theoretical parameters, thus allowing us to examine dynamics in latent parameters thought to underlie human behavior. We first validated our framework with simulated data, showing our ability to recover both stationary and non-stationary theoretical parameters. We then applied this approach to two independent empirical datasets. In the first dataset, we demonstrated that t-RNN can capture the volatile behavior patterns of psychiatric patients, which differ significantly from healthy controls. In the second dataset, we show that t-RNN can identify the changes in human exploration strategies that are modulated by task difficulty and block phase.

Several studies have explored the benefits of using neural networks fitted directly to behavioral data to examine various behavioral phenomena, including response time in task switching paradigms [35], the identification of novel theories in behavioral economics under risky choice [19], and particularly relevant to the current research, reinforcement learning (RL) tasks [9, 18, 21, 36]. For example, Dezfouli et al., 2019 [9] trained a recurrent neural network (RNN) to predict future human actions in a two-armed bandit task which was found to outperform a standard RL model in pure action prediction. This result was also generalized to other, more complicated decision-making tasks, such as the four-armed bandit [36] and build-your-own-stimulus tasks [18]. Additionally, in [37, 38] the authors employed a feedforward neural network fed with inputs such as mouse stress, affective phenotype, and prior task performance to examine how neurological measures control RL parameters. More recently, in order to study individuals' differences, [21] developed an autoencoder model, where an RNN encoder was used to map each agent's behavior into a low-dimensional latent space. The low-dimensional latent space produces a disentangled representation that can be used to study variations in participants' decision-making strategies. Overall, previous work shows the high ability of neural networks to learn to some extent the latent data generating model of human behavior across different paradigms. However, the main challenge remains the low interpretability of these methods.

Here, we consider our research as an expansion of the growing trend in using neural network models to study behavior and identify four distinct contributions of using a t-RNN framework. First, our method enhances the utilization of existing theoretical RL models, leveraging parameters with clear and straightforward interpretations. This allows us to dynamically estimate latent RL parameters based on empirical choice behavior. Additionally, we demonstrate that this approach exceeds stationary theoretical models fitted with conventional approaches (e.g., maximum-likelihood) by providing better predictive accuracy. Second, a known difficulty when training neural networks with empirical data is the need to obtain a large training set. Here, we demonstrate the validity of training RNN using synthetic data

which can be arbitrarily large. Post training, and without further optimization, the trained network is then used to infer empirical behavioral data. Third, our method also holds the benefit that it can be applied directly and online to newly acquired data, thus providing solutions for latent parameter estimation in real time. This allows for online monitoring of behavior during cognitive tasks, facilitating personalized interventions or manipulations based on each individual's current state. Fourth, the generality of our framework may be extended beyond choice behavior by utilizing theoretical models of other modalities. For example, given reaction times observation, we might be able to employ t-RNN to predict the dynamic theoretical parameters of evidence accumulation models. Another possibility is to utilize theoretical models of neural data (e.g., attractor networks) to train t-RNN in predicting latent theoretical parameters from the neural recordings.

An important benefit of t-RNN is its ability to predict theoretical parameters in a dynamic manner. While some recent studies have recognized the importance of relaxing the assumption of stationarity parameters in behavioral data [39–42], it remains common to treat behavior as a stationary process. Specifically, most of the computational modeling work in behavioral RL studies is performed by fitting a single set of parameters to each participant. Here, we compare our t-RNN framework to the Bayesian particle filtering method used in [14, 28, 43] which estimates hidden RL parameters assumed to slowly drift according to a Gaussian random walk. Our approach requires less prior specification (i.e., manual specification of prior and hyper-prior distributions) and provides a more flexible way to recover a wider range of RL theoretical parameter trajectories.

Another important aspect in the current work is the fact that a specific generative theoretical model, which serves as the supervised training signal for t-RNN needs to be specified. If human behavior is governed by a substantially different model, our method may have difficulties in accurately predicting behavior. Therefore, t-RNN is limited in its ability to predict parameters and actions, to the extent that the RL theoretical model it was introduced to during training accurately describes the participant's behavior. However, the high flexibility of neural networks opens the way for using t-RNN in much more sophisticated ways in the future. For example, future studies can train t-RNN on simulated data that were obtained from different agents acting according to different models. Moreover, these agents might switch from one model to the other arbitrarily. This might allow the dynamic trial-by-trial estimation of both parameters and models that might generate the observed behavior, and allow to estimate moments where the subject is switching from one to another model. Furthermore, since t-RNN is trained on synthetic data that can be infinitely large, it might be possible to introduce during training a vast model space and then predict the most suitable model, parameter, and action dynamically for a single empirical dataset. Thus, by incorporating all three predictions into the training process (models, parameters, and actions), future studies might be able to better address the issue of model misspecification, as a single t-RNN model may have the ability to predict various theoretical models.

Some limitations of our work. First, while our study demonstrated the ability of t-RNN to capture the high temporal dynamics of theoretical parameters, we did not provide a detailed explanation for the underlying nature of such changes. Although our current study did not include biological measurements, we believe that our approach might be able to facilitate future research where changes in latent parameters can be linked to meaningful biological indicators. Integrating the t-RNN methodology with these indicators may offer a deeper understanding of the neural correlates of behavior and the interplay between latent parameters and physiological processes. Second, we acknowledge that the training settings can influence t-RNN performance, our study aims to present a proof-of-concept for the primary framework of using RNNs to predict theoretical parameters, with further work required to test the impact

of different training settings on t-RNN performance. Third, our approach provided point estimates of latent parameters. To allow detailed conclusions for empirical data there is also a need to provide some confidence interval around these point estimates. There are several options that might facilitate such estimation. One option is to train several networks, estimate latent parameters for each subject and trial using all train networks, and then provide some summary statistics per data point. As a preliminary examination for this approach we did an ensemble analysis and found that it can indeed provide plausible variance estimates that are somewhat similar to the Bayesian uncertainty estimation of the particle filtering approach (see S7 Fig). However, other approaches could also be used, such as network dropout [44], or directly training the network to provide both point estimates and certainty estimates. Further studies should examine these alternatives in detail to provide further understanding for how variance estimates should be obtained. Fourth, t-RNN was compared in the current study with a particle filtering approach that requires the setting of a drift-rate parameter. We were able to show that using different settings of drift-rate did not change the overall conclusion of our study. Yet, another option might be to tailor the drift-rate parameter per individual. For example, if a training/validation/test set is available, one can search (e.g., using grid search) for the optimal drift-rate per subject and then use that to estimate parameters and behavior in the test set. However, this will require a large amount of data points per individual (which is typically hard to collect in human studies due to fatigue and other practical considerations). Furthermore, this approach will be different in essence from t-RNN, since the t-RNN approach provides behavior predictions and parameters estimation without any training with empirical data. As such it can be applied more efficiently (after training with artificial data and freezing the network's weights). Lastly, it is interesting to speculate how the complexity of the theoretical model might affect our methods. Previous studies have demonstrated that increasing model complexity may be necessary to effectively capture a broader range of animal and human behaviors [45]. We hypothesize that for more complex theoretical models we will find an advantage for t-RNN, as neural networks are known to be able to learn complex non-linear mappings [46]. Additionally, it is plausible that under certain conditions, t-RNN may estimate parameters with fewer observations compared to more traditional approaches (see S1 Text). However, further studies are required to ensure that our approach generalizes across a wide range of model complexities.

In conclusion, we argue that the advantages of our method outweigh its limitations and that we present a novel and versatile framework for enhancing the expressiveness of theoretical models to capture complex non-stationary behavior using neural networks. We believe that t-RNN serves as a bridge between traditional RL models and modern neural network models, allowing for a unified approach that can retain the strengths of both modeling paradigms.

Materials and methods

t-RNN Implementation

We implemented our network in PyTorch library [47], training with a single NVIDIA GeForce RTX 3080. We splitted the training set to 80/20, where 80% was used for training and the remaining 20% for validation. We use Adam optimizer [48] and a constant learning rate of 0.001. The number of hidden cells of the GRU layer was determined by a grid search selecting the one with the best validation loss which turn out to be 32 cells. We use early stopping together with dropout ($p = 0.2$ in both GRU layers) to prevent overfitting. We found that the best performance is obtained after a short training of fewer than 50 epochs. We also tested our model with different batch sizes (100,500,1000) and found the same results; therefore, we chose a batch sizes of 1000 to speed up the training time.

Application of t-RNN to simulated data

Task structure. Agents performed a two-armed bandit task for 10 blocks of 100 trials each. At each trial the agent received a stochastic binary reward of $r = 1$ for rewarded trials, or $r = 0$ otherwise. The reward probability of each action is fixed during a block of 100 trials and is selected randomly at the beginning of each block from three possible Bernoulli distribution: $\{P_{r_L}, P_{r_R}\} = \{0.1, 0.9\}, \{0.5, 0.5\}, \{0.9, 0.1\}$.

t-RNN training. Training t-RNN was done with a synthetic training set of 2,000 simulated Q-learners agents operating with Eqs 2 and 3. Each agent performed a two-armed bandit task for 10 blocks of 100 trials each. Each agents RL parameters were sampled from a uniform distribution of $\alpha \sim U(0, 1)$ and $\beta \sim U(0, 10)$. In the quantization preprocessing step, we discretized the α parameter into five evenly spaced bins (of size 0.2), and the β parameter into five evenly spaced bins (of size 2).

Importantly, to facilitate t-RNN flexibility to recover time-varying RL parameters, we simulated agents with non-stationary parameters. For each agent, there was a small probability of ($p = 0.005$) per trial that the agent's parameters will be re-sampled from the same uniform distribution. To keep the set from changing too rapidly, the overall number of trials in which the parameters were re-sampled was limited to four, and after each trial of parameters re-sampling the probability of another change was fixed to zero for 100 trials.

The combined loss function (Eq 1) included the following λ hyperparameters values: $\lambda_{\alpha}^{CE} = 0.2, \lambda_{\beta}^{CE} = 0.2, \lambda_{\alpha}^{MSE} = 6, \lambda_{\beta}^{MSE} = 0.1$, so that each loss term is approximately of equal magnitude.

Stationary Q-learning. Here, RL parameters were estimated for each agent individually, using a maximum-likelihood estimation approach (Python SciPy [49] minimized function with L-BFGS-B optimization, and five different starting search locations to avoid local maxima). This approach allowed for an estimation of a single and fixed set of RL parameters for each agent [1].

Bayesian particle filtering. Here, underlying RL parameters are assumed to drift slowly according to a Gaussian random walk,

$$\theta_{t+1} \leftarrow \theta_t + \epsilon_t \qquad \epsilon \sim \mathcal{N}(0, \sigma) \tag{9}$$

and are estimated for each agent on a trial-by-trial basis using Bayesian particle filtering. We implemented the same model as described by Samejima et al., 2003 [28]. To recover the simulated data, we set 1,000 particles to approximate the distribution of two hidden RL parameters $\{\alpha, \beta\}$. Initial distribution of the particles were set to be Gaussians, with means of {0, 0} and variances of {3, 1}, respectively. We set the hyper-prior for the dynamics of α and β as $\sigma_\alpha = 0.1$ and $\sigma_\beta = 0.05$, respectively (we also tested using hyper-prior as low as $\sigma_\alpha/\sigma_\beta = 0.005/0.0005$ and us high as $\sigma_\alpha/\sigma_\beta = 0.05/0.005$ and $\sigma_\alpha/\sigma_\beta = 0.1/0.5$, these setting resulted in poorer results). We also clipped the β to range between (0,10) to match the estimation range of other models.

Test agent. In order to validate and compare the performance of t-RNN against baseline methods, we simulated a test set of 30 artificial Q-learning agents (Eqs 2 and 3). Each agent was simulated on the same task structure, consisting of 10 blocks of 100 trials, and his RL parameters were sampled from a uniform distribution of $\alpha \sim U(0, 1)$ and $\beta \sim U(0, 10)$. The test set included three groups of 10 agents each. The first group had stationary RL parameters, while the second group had RL parameters that abruptly changed with a small probability for each trial ($p = 0.005$). The third group had RL parameters that drifted slowly according to a Gaussian random walk (Eq 9) with sigma values of $\sigma_\alpha = 0.1$ and $\sigma_\beta = 0.05$.

Application of t-RNN to psychiatric human individuals

Behavioral dataset and task structure [9]. The behavioral dataset includes $N = 101$ individuals that performed the bandit task for 12 blocks, with a free-choice rate of 40 second per block. At each trial subjects received a stochastic binary reward of $r = 1$ for rewarded trials, or $r = 0$ otherwise. The reward probability of each action was fixed during a block and selected at the beginning of each block from six possible settings:$\{P_{r_L}, P_{r_R}\} = \{0.25, 0.05\}, \{0.125, 0.05\}, \{0.08, 0.05\}, \{0.05, 0.25\}, \{0.05, 0.125\}, \{0.05, 0.08\}$. Each possible reward setting was repeated twice, for a total of 12 blocks. The data included task trials of 33 individuals diagnosed with bipolar disorder, 34 with depression and 34 individuals who were matched healthy controls (see Dezfouli et al., 2019 [9] for further details).

t-RNN training. Training t-RNN was done with a synthetic training set of 2,000 simulated Q-learners agents operating with Eqs 2 and 4. Each agent performed a two-armed bandit task for 12 blocks of 100 trials each. Each agents RL parameters were sampled from a uniform distribution of $\alpha \sim U(0, 0.2)$, $\beta \sim U(0, 10)$ and $\kappa \sim U(-0.5, 0.5)$. In the quantization preprocessing step, we discretized the α parameter into three evenly spaced bins (of size 0.066), the β parameter into five evenly spaced bins (of size 2) and the κ parameter into five evenly spaced bins (of size 0.2). Again, to facilitate t-RNN flexibility to recover time-varying RL parameters, we simulated agents with non-stationary parameters, with the same setting as described above.

The combined loss function (Eq 1) included the following λ hyperparameters values: $\lambda_{\alpha}^{CE} = 0.3$, $\lambda_{\beta}^{CE} = 0.25$, $\lambda_{\kappa}^{CE} = 0.3$, $\lambda_{\alpha}^{MSE} = 10$, $\lambda_{\beta}^{MSE} = 0.05$, $\lambda_{\kappa}^{MSE} = 10$, so that each loss term is approximately of equal magnitude.

d-RNN model. Data driven RNN was implemented following the same RNN model as described by Dezfouli et al., 2019 [9] (term here d-RNN). d-RNN is composed of a single GRU layer of 10 hidden units. The inputs to the d-RNN are the current action a_t and the reward r_t received after taking that action. The outputs are probabilities over the action of the next step a_{t+1}. We train the d-RNN with and Adam optimizer, learning rate of 0.001 and batch size of 1000. We assess performance with the same leave-one-out cross-validation schema depicted in [9]. At each CV round one of the subjects was withheld and d-RNN was train using data from the remaining subjects. Importantly, this procedure is done separately for each diagnostic group. For example, in order to predict the actions of a withheld bipolar subject, we use data from only the remaining bipolar subjects to train the d-RNN model.

Stationary Q-learning with preservation (QP-stationary). Here, stationary RL parameters are estimated for each subject individually, with maximum-likelihood estimation (Python SciPy [49] minimized function with L-BFGS-B optimization, and five different starting search locations to avoid local maxima).

Bayesian particle filtering. For each individual, we set 1000 particles to approximate the distribution of three hidden RL parameters {α, β, κ}, Gaussian with means of {−2, 1, 0} and variances of {1.5, 1, 1.5}, respectively. The hyper-prior for the dynamics of $\sigma_{\alpha} = 0.05$, $\sigma_{\beta} = 0.005$, and $\sigma_{\kappa} = 0.05$. We clipped the beta to range between (0,10) to match the estimation range of other models.

Application of t-RNN to interpret exploration behavior of human individuals

Behavioral dataset and task structure [22]. The behavioral dataset includes $N = 44$ individuals that performed the bandit task for 20 blocks of 10 trials each. At each trial subjects received a stochastic reward drawn from a Gaussian distribution with means of μ_L,μ_R (for

actions L and R respectively) and a variance (i.e., error variance) of $\tau_L^2 = \tau_R^2 = 10$. The mean reward probability of each action was fixed during a block and was drawn at the beginning of each block from a Gaussian distribution, with mean 0 and variance 100 (see Gershman, 2018 [22] for further details).

t-RNN training. Training t-RNN was done with a synthetic training set of 10,000 simulated Q-learners agents operating with Eqs 5, 6, 7 and 8. Each agent performed a two-armed bandit task for 20 blocks of 10 trials each. Each agents RL parameters were sampled from a uniform distribution of $\beta \sim U(0, 4)$ and $\gamma \sim U(0, 1)$. In the quantization preprocessing step, we discretized the β parameter into five evenly spaced bins (of size 0.8), the γ parameter into five evenly spaced bins (of size 0.2). Again, to facilitate t-RNN flexibility to recover time-varying RL parameters, we simulated agents with non-stationary parameters, with the same setting as described above. For each agent, there was a small probability of ($p = 0.02$) per trial that the agent's parameters will be re-sampled from the same uniform distribution. The overall number of trials in which the parameters were re-sampled was limited to four, and after each trial of parameters re-sampling the probability of another change was fixed to zero for 10 trials.

The combined loss function (Eq 1) included the following λ hyperparameters values: $\lambda_\beta^{CE} = 0.25$, $\lambda_\gamma^{CE} = 0.25$, $\lambda_\beta^{MSE} = 1$, $\lambda_\gamma^{MSE} = 10$, so that each loss term is approximately of equal magnitude.

d-RNN model. Similar to the one described above, with the two differences being are that the batch size was 200 and the LOOCV schema was performed over all 44 individuals.

Stationary hybrid exploration model. We adopted a similar model as described in [22], in which we estimated two stationary (i.e., fixed) parameters for each individual separately. Namely, a β random exploration parameter and a γ directed exploration parameter using maximum-likelihood estimation.

Bayesian particle filtering. For each individual, we set 1,000 particles to approximate the distribution of two hidden RL parameters $\{\beta, \gamma\}$. Initial distribution of the particles were set to be Gaussians, with means of $\{0, -2\}$ and variances of $\{1, 1\}$, respectively. The hyper-prior for the dynamics of β was $\sigma_\beta = 0.1$, and for γ was $\sigma_\gamma = 0.1$. We clipped the β to range between (0,4) to match the estimation range of other models.

Supporting information

S1 Text. Sensitivity of t-RNN as a function of the number of observations and free parameters in the theoretical RL model.
(PDF)

S1 Table. Sensitivity of t-RNN to parameters quantization. To ensure that quantization did not affect our performance, we ran two experiments where we quantized the RL parameters to as low as 3 bins (α and β parameters to 3 evenly spaced bins each) and up to 10 bins (α and β parameters to 10 evenly spaced bins each). We found similar performance across all three quantization settings.
(PDF)

S2 Table. Action prediction and RL parameter recovery of simulated test agents. Summary table of the raw results presented in the validation study of the main text (see Fig 2A).
(PDF)

S3 Table. Volatile environment. To assess the performance of t-RNN model in a volatile environment, we conducted an additional analysis similar to the one described in the main text (see Validating t-RNN using synthetic behavior) with agents simulated in a two-armed bandit task. However, here instead of a fixed reward probability for each arm, the reward expected value

schedule was governed by a random walk with a drift rate of 0.025, and upper and lower bounds of 0.15 and 0.85, respectively. Our findings align with the conclusions mentioned in the main text, demonstrating that the t-RNN outperformed the alternatives in terms of both action prediction and parameter estimation. Therefore, we conclude that our conclusions regarding tRNN performance generalized to two-armed bandit tasks in a volatile environment.
(PDF)

S4 Table. Action prediction for Dezfouli et al., 2019 [9] dataset. Summary table of the raw results presented in the main text (see Fig 3A).
(PDF)

S5 Table. Summary statistics of RL parameters estimations by diagnostic group. Summary statistics of the trial-by-trial RL parameter estimation produced by t-RNN for each diagnostic group.
(PDF)

S6 Table. Action prediction for Gershman 2018 [22] dataset. Summary table of the raw results presented in the main text (see Fig 4A).
(PDF)

S1 Fig. Relation of perseveration estimation with stay probability. The left panels show the trial-by-trial RL κ perseveration parameter estimation using t-RNN (blue) and QP-stationarity (green) methods, along with the moving average calculation of the stay probabilities (red; window size of 10 trials) for three example subjects (one from each diagnostic group). The right panels show the corresponding Pearson correlation between the moving average stay probabilities and κ parameter estimation of t-RNN and QP-stationarity model (red dashed line denotes the identity). The results indicate a strong correlation between the stay probabilities and κ parameter estimation of t-RNN ($r^2 > 0.9$), but not of the QP-stationarity ($r \approx 0$), which fails to detect changes in subject behavior throughout the task.
(PDF)

S2 Fig. Additional analysis inverse-temperature estimation. (A) Boxplot of the time-varying β estimates for each diagnostic group (middle black solid lines denote the median; light dots indicate a single trial estimate of the κ preservation parameter). Results indicate that all groups exhibit similar estimates, with a slight increase in the healthy group compared to the clinical groups. This suggests that the healthy group's action selection was slightly less random than the clinical group's. **(B)** Trial-by-trial β inverse-temperature parameter estimates by t-RNN averaged over the first 100 trials of each block and for each diagnostic group separately (shaded area signifies the s.e.m). In all three groups, subjects show an increase in their β estimates (behavior became less random as the block progressed), but the baseline differs, with clinical groups having a lower starting point. **(C)** Distribution of the Pearson correlation between the choice probabilities produced by t-RNN (calculated using a moving average of 10 trials) and the time-varying β inverse-temperature produced by t-RNN for each subject individually. The results indicate a strong relationship between t-RNN time-varying RL β parameter estimation and the moving average of the choice probabilities.
(PDF)

S3 Fig. Relation of inverse-temperature estimation with random choice. The left panels show the trial-by-trial RL β parameter estimation of t-RNN (blue; divided by 20 to fit the scale) and a moving average calculation of the absolute value difference between the choice probabilities produced by t-RNN and random choice probability (red; window size of 10 trials) for three example subjects (one from each diagnostic group). The right panels show the

corresponding Pearson correlation between the moving average choice probabilities and β parameter estimation of t-RNN (red dashed line denotes the identity). The results indicate a strong correlation between the two estimations.
(PDF)

S4 Fig. Supplement Fig 3D. In Fig 3D we depicted specific trials to illustrate the estimation of parameters across the two methods (QP-stationary, t-RNN). For completeness, we present the full range of trials for the same subjects (black boxes represent the trials that are depicted in Fig 3D). **(A)** Example bipolar subject. **(B)** Example depression subject. Action prediction (top) and theoretical RL κ parameters estimation (bottom).
(PDF)

S5 Fig. Directed exploration dynamics of repeated action blocks. (A) Example of a single instantiation of a single action block. The top panel depicts the sequence of chosen actions, obtain rewards, and t-RNN action predictions as red dots, red numbers, and blue solid lines, respectively. The subject consecutively chooses the same action. The bottom panel depicts t-RNN γ directed exploration parameter estimation, where it estimates a step decrease in the parameter throughout the block. **(B)** Dynamics of the directed exploration parameter averaged across all instantiations of a single action block. t-RNN estimates a sharp decrease in the γ directed exploration parameter in these blocks.
(PDF)

S6 Fig. t-RNN concatenation ablation. Training t-RNN without concatenation (green bar) led to a degradation in the performance in the action prediction compared to t-RNN trained with concatenation (blue bar) for both empirical datasets (error measured in BCE; black lines indicate s.e.m; $^{***}\ p < .001$; $^{**}\ p < .01$).
(PDF)

S7 Fig. Using ensemble method to gain variance estimation for t-RNN. (a) alpha parameter estimation, **(b)** beta parameter estimation. The top panel represents recovery of stationary RL parameters, the middle depicts recovery of abruptly changing RL parameters, and the bottom shows recovery of gradually changing RL parameters. Ground truth (red), stationary Q-learning (green), Bayesian (yellow), our t-RNN with static training set (pink), and t-RNN (blue). Error bars, Bayesian model (in yellow; calculated using the variance over the particles). Error bars, ours (in blue; calculated using 10 different runs of our model).
(PDF)

Acknowledgments

We thank Yuval Ger for her help with the figure design.

Author Contributions

Conceptualization: Yoav Ger.

Data curation: Yoav Ger.

Formal analysis: Yoav Ger, Eliya Nachmani, Lior Wolf, Nitzan Shahar.

Funding acquisition: Nitzan Shahar.

Investigation: Yoav Ger.

Methodology: Yoav Ger, Eliya Nachmani, Lior Wolf, Nitzan Shahar.

Software: Yoav Ger.

Supervision: Lior Wolf, Nitzan Shahar.

Visualization: Yoav Ger.

Writing – original draft: Yoav Ger, Eliya Nachmani, Lior Wolf, Nitzan Shahar.

Writing – review & editing: Yoav Ger, Lior Wolf, Nitzan Shahar.

References

1. Daw ND. Trial-by-trial data analysis using computational models. Decision making, affect, and learning: Attention and performance XXIII. 2011; 23(1). https://doi.org/10.1093/acprof:oso/9780199600434.003.0001
2. Wilson RC, Collins AG. Ten simple rules for the computational modeling of behavioral data. Elife. 2019; 8:e49547. https://doi.org/10.7554/eLife.49547 PMID: 31769410
3. Eckstein MK, Master SL, Xia L, Dahl RE, Wilbrecht L, Collins AG. The interpretation of computational model parameters depends on the context. Elife. 2022; 11:e75474. https://doi.org/10.7554/eLife.75474 PMID: 36331872
4. Schultz W, Dayan P, Montague PR. A neural substrate of prediction and reward. Science. 1997; 275 (5306):1593–1599. https://doi.org/10.1126/science.275.5306.1593 PMID: 9054347
5. Daw ND, Gershman SJ, Seymour B, Dayan P, Dolan RJ. Model-based influences on humans' choices and striatal prediction errors. Neuron. 2011; 69(6):1204–1215. https://doi.org/10.1016/j.neuron.2011.02.027 PMID: 21435563
6. Montague PR, Dolan RJ, Friston KJ, Dayan P. Computational psychiatry. Trends in cognitive sciences. 2012; 16(1):72–80. https://doi.org/10.1016/j.tics.2011.11.018 PMID: 22177032
7. Dayan P, Daw ND. Decision theory, reinforcement learning, and the brain. Cognitive, Affective, & Behavioral Neuroscience. 2008; 8(4):429–453. https://doi.org/10.3758/CABN.8.4.429 PMID: 19033240
8. Niv Y. Reinforcement learning in the brain. Journal of Mathematical Psychology. 2009; 53(3):139–154. https://doi.org/10.1016/j.jmp.2008.12.005
9. Dezfouli A, Griffiths K, Ramos F, Dayan P, Balleine BW. Models that learn how humans learn: the case of decision-making and its disorders. PLoS computational biology. 2019; 15(6):e1006903. https://doi.org/10.1371/journal.pcbi.1006903 PMID: 31185008
10. Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural networks. 1989; 2(5):359–366. https://doi.org/10.1016/0893-6080(89)90020-8
11. Siegelmann HT, Sontag ED. On the computational power of neural nets. In: Proceedings of the fifth annual workshop on Computational learning theory; 1992. p. 440–449.
12. Yarkoni T, Westfall J. Choosing prediction over explanation in psychology: Lessons from machine learning. Perspectives on Psychological Science. 2017; 12(6):1100–1122. https://doi.org/10.1177/1745691617693393 PMID: 28841086
13. Hasson U, Nastase SA, Goldstein A. Direct fit to nature: an evolutionary perspective on biological and artificial neural networks. Neuron. 2020; 105(3):416–434. https://doi.org/10.1016/j.neuron.2019.12.002 PMID: 32027833
14. Samejima K, Ueda Y, Doya K, Kimura M. Representation of action-specific reward values in the striatum. Science. 2005; 310(5752):1337–1340. https://doi.org/10.1126/science.1115270 PMID: 16311337
15. Daw ND, O'doherty JP, Dayan P, Seymour B, Dolan RJ. Cortical substrates for exploratory decisions in humans. Nature. 2006; 441(7095):876–879. https://doi.org/10.1038/nature04766 PMID: 16778890
16. Behrens TE, Woolrich MW, Walton ME, Rushworth MF. Learning the value of information in an uncertain world. Nature neuroscience. 2007; 10(9):1214–1221. https://doi.org/10.1038/nn1954 PMID: 17676057
17. Acerbi L, Ma WJ. Practical Bayesian optimization for model fitting with Bayesian adaptive direct search. Advances in neural information processing systems. 2017; 30.
18. Song M, Niv Y, Cai M. Using Recurrent Neural Networks to Understand Human Reward Learning. In: Proceedings of the Annual Meeting of the Cognitive Science Society. vol. 43; 2021. p. 1388–1394.

19. Peterson JC, Bourgin DD, Agrawal M, Reichman D, Griffiths TL. Using large-scale experiments and machine learning to discover theories of human decision-making. Science. 2021; 372(6547):1209–1214. https://doi.org/10.1126/science.abe2629 PMID: 34112693
20. Schaeffer R., Khona M., Meshulam L., International Brain Laboratory, Fiete I. R. Reverse-engineering recurrent neural network solutions to a hierarchical inference task for mice. Advances in Neural Information Processing Systems. 2020; 33:4584–4596.
21. Dezfouli A, Ashtiani H, Ghattas O, Nock R, Dayan P, Ong CS. Disentangled behavioural representations. Advances in neural information processing systems. 2019; 32.
22. Gershman SJ. Deconstructing the human algorithms for exploration. Cognition. 2018; 173:34–42. https://doi.org/10.1016/j.cognition.2017.12.014 PMID: 29289795
23. Widiger T. A., Frances A. J., Pincus H. A. E., Ross R. E. DSM-IV sourcebook, Vol. 3. American Psychiatric Publishing, Inc.; 1997.
24. Wilson RC, Geana A, White JM, Ludvig EA, Cohen JD. Humans use directed and random exploration to solve the explore–exploit dilemma. Journal of Experimental Psychology: General. 2014; 143 (6):2074. https://doi.org/10.1037/a0038199 PMID: 25347535
25. Cho K, Van Merriënboer B, Bahdanau D, Bengio Y. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:14091259. 2014;.
26. Ranjan R, Castillo CD, Chellappa R. L2-constrained softmax loss for discriminative face verification. arXiv preprint arXiv:170309507. 2017;.
27. Fei-Fei L, Karpathy A. Stanford's cs231n class notes; 2015.
28. Samejima K, Doya K, Ueda Y, Kimura M. Estimating internal variables and paramters of a learning agent by a particle filter. Advances in neural information processing systems. 2003; 16.
29. Watkins CJ, Dayan P. Q-learning. Machine learning. 1992; 8(3):279–292. https://doi.org/10.1023/A:1022676722315
30. Sutton RS, Barto AG. Reinforcement learning: An introduction. MIT press; 2018.
31. Seymour B, Daw ND, Roiser JP, Dayan P, Dolan R. Serotonin selectively modulates reward value in human decision-making. Journal of Neuroscience. 2012; 32(17):5833–5842. https://doi.org/10.1523/JNEUROSCI.0053-12.2012 PMID: 22539845
32. Bishop CM, Nasrabadi NM. Pattern recognition and machine learning. vol. 4. Springer; 2006.
33. Thompson WR. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. Biometrika. 1933; 25(3-4):285–294. https://doi.org/10.1093/biomet/25.3-4.285
34. Auer P, Cesa-Bianchi N, Fischer P. Finite-time analysis of the multiarmed bandit problem. Machine learning. 2002; 47:235–256. https://doi.org/10.1023/A:1013689704352
35. Jaffe PI, Poldrack RA, Schafer RJ, Bissett PG. Modelling human behaviour in cognitive tasks with latent dynamical systems. Nature Human Behaviour. 2023; p. 1–15. PMID: 36658212
36. Fintz M, Osadchy M, Hertz U. Using deep learning to predict human decisions and using cognitive models to explain deep learning models. Scientific reports. 2022; 12(1):4736. https://doi.org/10.1038/s41598-022-08863-0 PMID: 35304572
37. Sandi C, Gerstner W, Lukšys G. Stress, noradrenaline, and realistic prediction of mouse behaviour using reinforcement learning. Advances in Neural Information Processing Systems. 2008; 21.
38. Luksys G, Gerstner W, Sandi C. Stress, genotype and norepinephrine in the prediction of mouse behavior using reinforcement learning. Nature neuroscience. 2009; 12(9):1180–1186. https://doi.org/10.1038/nn.2374 PMID: 19684590
39. Roy NA, Bak JH, Akrami A, Brody C, Pillow JW. Efficient inference for time-varying behavior during learning. Advances in neural information processing systems. 2018; 31. PMID: 31244514
40. Ashwood Z, Roy NA, Bak JH, Pillow JW. Inferring learning rules from animal decision-making. Advances in Neural Information Processing Systems. 2020; 33:3442–3453. PMID: 36177341
41. Ashwood ZC, Roy NA, Stone IR, Urai AE, Churchland AK, Pouget A, et al. Mice alternate between discrete strategies during perceptual decision-making. Nature Neuroscience. 2022; 25(2):201–212. https://doi.org/10.1038/s41593-021-01007-z PMID: 35132235
42. Ashwood Z, Jha A, Pillow JW. Dynamic Inverse Reinforcement Learning for Characterizing Animal Behavior. Advances in Neural Information Processing Systems. 2022; 35:29663–29676.
43. Ito M, Doya K. Validation of decision-making models and analysis of decision variables in the rat basal ganglia. Journal of Neuroscience. 2009; 29(31):9861–9874. https://doi.org/10.1523/JNEUROSCI.6157-08.2009 PMID: 19657038
44. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research. 2014; 15(1):1929–1958.

45. Miller K, Botvinick M, Brody C. From predictive models to cognitive models: Separable behavioral processes underlying reward learning in the rat. bioRxiv p. 461129. publisher; 2021.

46. Gonçalves PJ, Lueckmann JM, Deistler M, Nonnenmacher M, Öcal K, Bassetto G, et al. Training deep neural density estimators to identify mechanistic models of neural dynamics. eLife. 2020; 9:e56261. https://doi.org/10.7554/eLife.56261 PMID: 32940606

47. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems. 2019; 32.

48. Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980. 2014;.

49. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature methods. 2020; 17(3):261–272. https://doi.org/10.1038/s41592-019-0686-2 PMID: 32015543