Reinforcement-Based Control of Information Processing in Recurrent Neural Networks Produces Optimal Speed-Accuracy Tradeoff

Ivan Grahek* (ivan_grahek@brown.edu)
Alekh Karkada Ashok* (alekh_karkada_ashok@brown.edu)
Atsushi Kikumoto (atsushi_kikumoto@brown.edu)
Thomas Serre (thomas_serre@brown.edu)
Michael J. Frank (michael_frank@brown.edu)

Department of Cognitive, Linguistic, and Psychological Sciences
Carney Institute for Brain Science
Brown University
Providence, RI, USA

*These authors contributed equally to this work

Abstract

Optimal decision-making entails not only arriving at the best choice but doing so in the most efficient way possible. Critically, humans and other animals adjust the speed and accuracy of their decisions to the demands of the current task. Recurrent neural networks (RNNs) can process noisy sequential bits of evidence and are used as models of decision-making. However, they are typically trained on input sequences of fixed length, and thus have no notion of decision time. Here, we develop an RNN with a separate controller network that adjusts the number of RNN steps taken in a decision-making task. Using reinforcement learning in the controller, this architecture optimally trades off decision time and accuracy. In this way, it aligns with normative models of human decision-making, and produces a natural notion of decision time.

Keywords: decision-making; recurrent neural networks; decision time; reinforcement learning

Humans routinely adjust how they process information and make decisions based on current task demands (Botvinick et al., 2001; Bogacz et al., 2006). This allows rapid decisions when the task is easy or the stakes are low, and increased accuracy as demands increase (Manohar et al., 2015; Leng et al., 2021). Recurrent neural networks (RNNs) have emerged as a model of human and animal decision-making, allowing for mechanistic interpretations of the neural dynamics observed in the prefrontal cortex (Mante et al., 2013). However, RNNs are commonly trained on a fixed number of sequential inputs (e.g., evidence in favor of one choice). Thus, they have no notion of decision time, a measure critical for understanding natural human decision-making.

Here we develop a neural network architecture combining an RNN (representing the cortex; Mante et al., 2013) trained to solve a perceptual decision-making task with a controller network (representing the basal ganglia; Ratcliff & Frank, 2012; Herz et al., 2016) trained to decide when to stop accumulating evidence and commit a response. Balancing performance-based rewards against the costs of processing time, this model displays a speed-accuracy tradeoff and makes faster decisions as processing costs increase. This architecture allows for a comparative investigation of neural representations that emerge in artificial and natural decision-makers.

RNN Controller model (RNNC)

Task Network (RNN)

We trained a network with 2 recurrently connected nonlinear neurons to perform an analogue of the Random Dot Motion Task (RDM; Mante et al., 2013) that required continuous evidence accumulation (Fig. 1A). While small network size allowed for full tractability, it can easily be increased in future research. At each timestep, the network received noisy evidence in favor of a left or right response (Lo & Wang, 2006; Mante et al., 2013). Coherence was fixed (1.0) and could be either positive or negative, determining the correct label (y_t; left vs. right response), and noise was randomly sampled from the fixed distribution at each timestep:

$$x_t = coherence + Normal(0, \sigma = 0.5) \quad (1)$$

We trained the network to predict correct labels (left vs. right response) using backpropagation through time and cross-entropy loss. Activity of the hidden layer at each step depended on task inputs, previous hidden layer activity, and network weights and biases:

$$h_{t+1} = F(x_t, h_t, w_F) \quad (2)$$

Hidden layer activity at each time point was read out through a linear layer determining network's response ($\hat{y}_t$). We trained 5 Task RNNs to take a fixed number of steps on 8000 trials. As expected, networks that took more steps (i.e., longer evidence integration), achieved higher accuracy on 2000 held-out test trials (Fig. 1B). Networks were trained on a fixed coherence level (-1 or 1), but their performance on test trials was systematic across a wide range of coherence levels not experienced during training. This indicated that they have learned to accumulate evidence, rather than memorize responses for a coherence level experienced during training. Furthermore, networks trained with more steps performed better across coherence levels as evidenced by steeper psychometric functions (Fig. 1C). Hidden layer activity was randomly initialized on each trial, and within-trial activity of the Task RNN trained on 40 time steps (Fig. 1D) was used to train the Controller network via reinforcement learning.

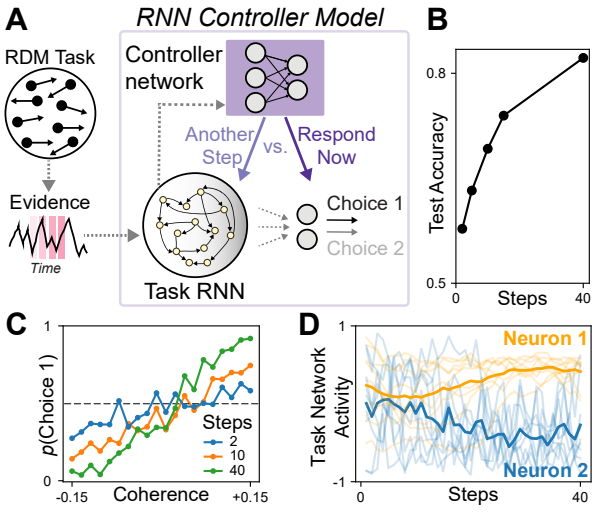

Figure 1: **A.** Random Dot Motion task and network architecture. **B.** Task RNNs trained to take more steps have higher accuracy on the RDM task. **C.** Networks that take more steps perform better across coherence levels. **D.** Single-trial and average activity of the Task RNN Neurons across time steps.

Controller Network

The Controller Network received hidden states activity of the Task RNN at each time step and learned to decide whether

to take another step or commit a response at the current step (Fig. 1A). It was trained to maximize rewards determined as:

$$R = \begin{cases} -stepcost_t, & \text{if } a_t = \text{another step} \\ +1, & \text{if } a_t = \text{respond now and } y_t \neq y_t \\ -5, & \text{if } a_t = \text{respond now and } y_t = y_t \end{cases} \quad (3)$$

The cost of processing time at each step was implemented as a function of a fixed step cost and the current time step:

$$stepcost_t = \sum_{i=1}^{t} cost \times t \quad (4)$$

We implemented the controller network (Fig. 1A) as a deep RL agent (instantiated as actor and critic feedforward multilayer perceptrons) trained using Proximal Policy Optimization (PPO; Schulman et al., 2017). The Controller Network learned a policy mapping states (Task RNN unit activities) to actions (take another step or respond at the current step).

Assuming a linear summation of evidence, the value-maximizing number of processing steps depends on the cost per step and the expected reward (Fig. 2A-white panel). In the RDM task with the reward structure we deployed, a value-optimal agent should reduce the number of steps it takes as the cost per step increases (Fig. 2A-blue dots).

Results

We trained 10 networks with different step costs (0.0004-0.004) on 10000 trials. The weights of the pre-trained Task RNN were frozen and connected to the Controller Network. When the Controller Network decided to stop, the response was read from the Task RNN readout layer. These networks formed the RNN Controller (RNNC) architecture (Fig. 1A). The architecture was implemented in PyTorch (Paszke et al., 2019) and trained using a custom Gymnasium environment (Towers et al., 2024). Controller Network's parameters were optimized using stochastic gradient descent.

The RNNC model decreased the number of steps it took (i.e., decision time) as the step cost increased. Average number of steps across 10000 trials of training (Fig. 2A-orange dots) closely matched the value-maximizing number of steps for the trials that the model was trained on (blue dots). Different step costs produced cost functions that varied in how flat they are around the maximum (Fig. 2A heatmap). Thus, for some costs the area around the maximum had similar expected values (e.g., left-most cost), while others had more peaked value maxima (e.g., right-most cost). The RNNC model displayed more variability in steps taken (error bars represent standard deviations) when trained with step costs producing flatter cost functions.

As RNNC models with different step costs took different number of steps during task performance, their decision accuracy changed. Increasing the number of steps (i.e., decision time) led to an increase in average accuracy across trials (Fig. 2B). Thus, the model displayed a speed-accuracy tradeoff, a core characteristic of human decision-making (Heitz, 2014).

Conclusions

Here we introduce the RNNC architecture which leverages reinforcement learning on RNN activity to decide when to commit to a decision. We connect previous work using feedforward and hand-tuned control architectures (Botvinick et al., 2001; Simen et al., 2006) with recent machine learning approaches to estimate reaction times (Goetschalckx et al., 2023). In so doing, we show that combining RNNs solving decision-making tasks with a controller architecture optimized via RL produces a behavior resembling human decision-making.

In this preliminary work we focused on one control signal (decision threshold; Bogacz et al., 2006), which is dynamically regulated by the basal ganglia (Ratcliff & Frank, 2012; Herz et al., 2016; Doi et al., 2020; Pagnier et al., 2024), and is subject to RL, and one type of decision cost (time; (Bogacz et al., 2006; Simen et al., 2006; Kurzban et al., 2013). However, the RNNC model can easily be extended to study other control signals (e.g., feature-based attention) and representations that modulate control signals (e.g., task difficulty, conflict, reward). Thus, the RNNC model will allow for a systematic study of representations that emerge in neural networks bounded by human-like costs. This will allow for a more in-depth comparison of neural dynamics and representations that emerge in human and artificial cognition. The immediate next step will be to jointly train the Task and Controller Networks at different coherence levels in the RDM task. This will allow us to investigate whether task difficulty representations emerge in the model, and how they guide decision times and accuracy. Such investigation will allow for a direct comparison with behavioral and neural data from human and animal decision-making tasks.

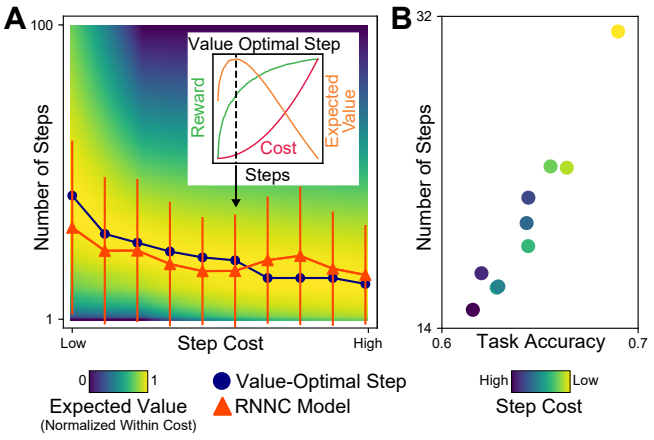

Figure 2: **A.** Optimal number of steps are calculated by considering the expected accuracy (under linear summation of evidence), costs and rewards at each step. The average number of steps taken by the RNNC model (orange; error bars are standard deviations) reduces with increasing costs, closely matching value-optimal steps (blue). **B.** The model trades off decision time (i.e., number of steps) for task accuracy.

Acknowledgments

IG and AK wish to thank the members of the Nonlinear Dynamics Journal Club and the MacKay 2003 and Murphy 2022 book clubs for many helpful discussions. MJF was funded by ONR MURI Award N00014-23-1-2792 and NIMH P50MH106435-06A1. AKA and TS were funded by ONR grant N00014-24-1-2026. Computing hardware supported by NIH Office of the Director grant S10OD025181 through the Center for Computation and Visualization at Brown University.

References

Bogacz, R., Brown, E., Moehlis, J., Holmes, P., & Cohen, J. D. (2006). The physics of optimal decision making: A formal analysis of models of performance in two-alternative forced-choice tasks. *Psychological Review*, *113*(4), 700–765. doi: 10.1037/0033-295X.113.4.700

Botvinick, M. M., Braver, T. S., Barch, D. M., Carter, C. S., & Cohen, J. D. (2001). Conflict monitoring and cognitive control. *Psychological Review*, *108*(3), 624–652. doi: 10.1037/0033-295X.108.3.624

Doi, T., Fan, Y., Gold, J. I., & Ding, L. (2020). The caudate nucleus contributes causally to decisions that balance reward and uncertain visual information. *eLife*, *9*, e56694. doi: 10.7554/eLife.56694

Goetschalckx, L., Govindarajan, L. N., Karkada Ashok, A., Ahuja, A., Sheinberg, D., & Serre, T. (2023). Computing a human-like reaction time metric from stable recurrent vision models. In *Advances in neural information processing systems* (Vol. 36, pp. 14338–14365).

Heitz, R. P. (2014). The speed-accuracy tradeoff: History, physiology, methodology, and behavior. *Frontiers in Neuroscience*, *8*. doi: 10.3389/fnins.2014.00150

Herz, D. M., Zavala, B. A., Bogacz, R., & Brown, P. (2016). Neural correlates of decision thresholds in the human subthalamic nucleus. *Current Biology*, *26*(7), 916–920. doi: 10.1016/j.cub.2016.01.051

Kurzban, R., Duckworth, A., Kable, J. W., & Myers, J. (2013). An opportunity cost model of subjective effort and task performance. *Behavioral and Brain Sciences*, *36*(6), 661–679. doi: 10.1017/S0140525X12003196

Leng, X., Yee, D., Ritz, H., & Shenhav, A. (2021). Dissociable influences of reward and punishment on adaptive cognitive control. *PLoS Computational Biology*, *17*(12), e1009737.

Lo, C.-C., & Wang, X.-J. (2006). Cortico–basal ganglia circuit mechanism for a decision threshold in reaction time tasks. *Nature Neuroscience*, *9*(7), 956–963. doi: 10.1038/nn1722

Manohar, S. G., Chong, T. T.-J., Apps, M. A. J., Batla, A., Stamelou, M., Jarman, P. R., ... Husain, M. (2015). Reward pays the cost of noise reduction in motor and cognitive control. *Current Biology*, *25*(13), 1707–1716. doi: 10.1016/j.cub.2015.05.038

Mante, V., Sussillo, D., Shenoy, K. V., & Newsome, W. T. (2013). Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, *503*(7474), 78–84.

Pagnier, G. J., Asaad, W. F., & Frank, M. J. (2024). Double dissociation of dopamine and subthalamic nucleus stimulation on effortful cost/benefit decision making. *Current Biology*, *34*(3), 655–660.e3. doi: 10.1016/j.cub.2023.12.045

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems* (Vol. 32).

Ratcliff, R., & Frank, M. J. (2012). Reinforcement-based decision making in corticostriatal circuits: Mutual constraints by neurocomputational and diffusion models. *Neural Computation*, *24*(5), 1186–1229. doi: 10.1162/NECO$_{a0}$0270

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). *Proximal policy optimization algorithms* (Tech. Rep.). arXiv. doi: 10.48550/arXiv.1707.06347

Simen, P., Cohen, J. D., & Holmes, P. (2006). Rapid decision threshold modulation by reward rate in a neural network. *Neural Networks*, *19*(8), 1013–1026. doi: 10.1016/j.neunet.2006.05.038

Towers, M., Terry, J. K., Kwiatkowski, A., Balis, J. U., Cola, G., Deleu, T., ... Younis, O. G. (2024). Gymnasium (v1.0.0a1) [Computer software manual]. doi: 10.5281/zenodo.10655021