Reverse-engineering Recurrent Neural Network solutions to a hierarchical inference task for mice

Rylan Schaeffer[1, 4], Mikail Khona[2], Leenoy Meshulam[3, 4], International Brain Laboratory[4], and Ila Rani Fiete[3,4]

[1]Institute for Applied Computational Science, Harvard University
[2]Department of Physics, Massachusetts Institute of Technology
[3]Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology
[4]International Brain Laboratory

June 8, 2020

Abstract

We study how recurrent neural networks (RNNs) solve a hierarchical inference task involving two latent variables and disparate timescales separated by 1-2 orders of magnitude. The task is of interest to the International Brain Laboratory, a global collaboration of experimental and theoretical neuroscientists studying how the mammalian brain generates behavior. We make four discoveries. First, RNNs learn behavior that is quantitatively similar to ideal Bayesian baselines. Second, RNNs perform inference by learning a two-dimensional subspace defining beliefs about the latent variables. Third, the geometry of RNN dynamics reflects an induced coupling between the two separate inference processes necessary to solve the task. Fourth, we perform model compression through a novel form of knowledge distillation on hidden representations – Representations and Dynamics Distillation (RADD)– to reduce the RNN dynamics to a low-dimensional, highly interpretable model. This technique promises a useful tool for interpretability of high dimensional nonlinear dynamical systems. Altogether, this work yields predictions to guide exploration and analysis of mouse neural data and circuity.

1 Introduction

Decision making involves weighing mutually-exclusive options and choosing the best among them. Selecting the optimal action requires integrating data over time and combining it with prior information in a Bayesian sense. Here we seek to understand how RNNs perform hierarchical inference. For concreteness and for the later goal of comparing against the mammalian brain, we consider a perceptual decision-making and change-point detection task used by the International Brain Laboratory (IBL) [1], a collaboration of twenty-two experimental and theoretical neuroscience laboratories. Optimally solving the IBL task requires using sensory data to infer two latent variables, one cued and one uncued, over two timescales separated by 1-2 orders of magnitude.

We address two questions. First, how do RNNs compare against normative Bayesian baselines on this task, and second, what are the representations, dynamics and mechanisms RNNs employ to perform inference in this task? These questions are of interest to both the neuroscience and the machine learning communities. To neuroscience, RNNs are neurally-plausible mechanistic models that can serve as a good comparison with animal behavior and neural data, as well as a source of scientific hypotheses [15, 16, 31, 5, 27, 10, 8]. To machine learning, we build on prior work reverse engineering how RNNs solve tasks [30, 28, 16, 15, 3, 20, 14, 19], by studying a complicated task that nevertheless has exact Bayesian baselines for comparison, and by contributing task-agnostic analysis techniques.

The IBL task is described in prior work [29], so we include only a brief summary here. On each *trial*, the mouse is shown a (low or medium contrast) stimulus in its left or right visual fields and must indicate on which

side it perceived the stimulus. Upon choosing the correct side, it receives a small reward. Over a number of consecutive trials (a *block*), the stimulus has a higher probability of appearing on one side (left stimulus probability p_s, right stimulus probability $1 - p_s$). In the next block, the stimulus side probabilities switch. The change-points between blocks are not signaled to the mouse. This task involves multiple computations, elements of which have been studied under various names including change-point detection [2, 22].

2 Methods

2.1 IBL task implementation

Each session consists of a variable number trials, indexed n. Each trial is part of a block, with blocks defining the prior probability that a stimulus presented on the trail is shown on the left versus the right. The block side on trial n, denoted $b_n \in \{-1, 1\}$ (-1: left, 1: right), is determined by a 2-state semi-Markov chain with a symmetric transition matrix. The probability of remaining on the same block side as in the last trial is $1 - p_b$; the probability of switching block sides is p_b. The process is semi-Markov because p_b varies as a function of the current block length (l_n) to ensures a minimum block length of 20 and maximum block length of 100, with otherwise geometrically distributed block lengths.

$$\begin{bmatrix} P(b_n = 1) \\ P(b_n = -1) \end{bmatrix} = \begin{bmatrix} 1 - p_b & p_b \\ p_b & 1 - p_b \end{bmatrix} \begin{bmatrix} P(b_{n-1} = 1) \\ P(b_{n-1} = -1) \end{bmatrix} \qquad p_b = \begin{cases} 0 & l_n < 20 \\ p_{b0} & 20 \leq l_n \leq 100 \\ 1 & 100 < l_n \end{cases}$$

The stimulus ($s_n \in \{-1, 1\}$) presented on trial n is either a left or right stimulus, determined by a Bernoulli process with a single fixed parameter p_s, which gives the probability that the stimulus is on the same side as the current block (termed a *concordant* trial). The probability of a *discordant* trial (stimulus on opposite side of block) is $1 - p_s$. In the IBL task, $p_{b0} = 0.02$ and $p_s = 0.8$.

Neural time-constants (10-100 ms) are much shorter than the timescale of trials (~ 1 s), so we model a trial as itself consisting of multiple timesteps indexed by t. A trial terminates one timestep after the RNN takes an action (explained in the next subsection) or after timing out at T_{max} steps, whichever comes first. At the start of the trial, the stimulus side s_n and a stimulus contrast strength μ_n are sampled (Fig. 1). Within a trial, on each step, the RNN receives three scalar inputs. On the first step, all three are 0. For each subsequent step, the RNN receives two noisy observations $o^L_{n,t}, o^R_{n,t}$, sampled i.i.d. from two univariate Gaussians with mean μ_n for the stimulus side and mean 0 for the other. The third input is a reinforcement signal $r_{n,t}$, which takes one of three possible values: a small waiting penalty (-0.05) in every timestep, a reward (+1) if the correct action was taken on the previous step, or a punishment (-1) if the incorrect action was taken on the previous step or the model timed out.

$$\begin{aligned} b_n &\sim P(b|b_{n-1}, p_b(l_n)) \\ s_n = b_n|b_n &\sim Bern(p_s) \\ \mu_n &\sim \mathcal{U}([0, 0.5, 1.0, 1.5, 2.0, 2.5]) \\ o^S_{n,t}|\mu_n &\sim \mathcal{N}(\mu_n, 1) \\ o^{\sim S}_{n,t}|\mu_n &\sim \mathcal{N}(0, 1) \end{aligned}$$

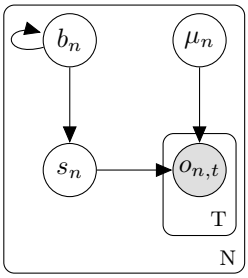

Figure 1: Generative model of the IBL task. Block side b_n is determined by a 2-state semi-Markov chain. Stimulus side s_n is either b_n with probability p_s or $-b_n$ with probability $1 - p_s$. Trial stimulus contrast μ_n determines the observations on each timestep t within a trial: $o^S_{n,t}$ for the stimulus side, $o^{\sim S}_{n,t}$ for the non-stimulus side.

2.2 Recurrent network architecture and training

On each step, the observation $o_{n,t} = \begin{bmatrix} o^L_{n,t} & o^R_{n,t} & r_{n,t} \end{bmatrix}^T$ is input to the RNN. Letting $h_{n,t}$ denote the RNN state on the nth trial and the tth step within the trial, the state is defined by the typical dynamics

$$h_{n,t} = \tanh(W^{rec} h_{n,t-1} + W^{in} o_{n,t} + b^{rec})$$
$$a_{n,t} = \text{softmax}(W^{out} h_{n,t} + b^{out})$$

where $a_{n,t}$ is a probability distribution over the two possible actions (left or right). An action is defined by when the probability mass on either action exceeds a fixed threshold (0.9). We present RNNs with 50 hidden units, but the results are similar for other numbers of units (e.g. 100, 250). We train the RNN under cross entropy loss using stochastic gradient descent with initial learning rate 0.001 and initial momentum = 0.1. RNN parameters were initialized using PyTorch defaults. PyTorch and NumPy random seeds were both set to 1. Our code will be publicly available at `https://github.com/int-brain-lab/ann-rnns`.

2.3 Normative Bayesian baselines

The IBL task involves inference of two latent variables, the stimulus side and the block side. Exact inference can be decomposed into two inference subproblems that occur over different timescales, which we term *stimulus side inference* and *block side inference*:

$$\underbrace{P(s_n | s_{<n}, o_{\leq n, \leq T})}_{\text{Current stimulus posterior}} = \underbrace{\frac{P(o_{n,\leq T} | s_n)}{P(o_{n,\leq T})}}_{\text{Stimulus side inference}} \underbrace{P(s_n | s_{\leq n-1}, o_{\leq n-1, \leq T})}_{\text{Block side inference}}$$

where $\cdot_{\leq m}$ denotes all indices from 1 to m, inclusive. We consider two Bayesian baselines. The *Bayesian actor* performs the task independently from the RNN, but using the same action rule (i.e. an action taken when its stimulus posterior passes the action threshold). The *Bayesian observer* receives the same observations as the RNN, but cannot decide when to act; the RNN therefore determines how long a trial lasts. The Bayesian actor tells us what ceiling performance is, while the Bayesian observer tells us how well the RNN could do given when the RNN chooses to act.

Other than this difference, the Bayesian actor and the Bayesian observer are identical. Both assume perfect knowledge of the task structure and task parameters, and both are comprised of two separate submodels performing inference. The first submodel performs stimulus side inference given the block side, while the other submodel infers block changepoints given the history of true stimuli sides. True stimuli sides can be determined after receiving feedback because the selected action and the ensuing feedback signal (correct or wrong) together fully specify the true stimulus side.

Stimulus side inference occurs at the timescale of a single trial. Since observations within a trial are sampled i.i.d., the observations are conditionally independent given the trial stimulus strength μ_n. The likelihood is therefore:

$$P(o_{n,\leq T} | s_n) = \sum_{\mu_n} P(o_{n,\leq T} | \mu_n) P(\mu_n | s_n) = \sum_{\mu_n} \Big(\prod_{t=1}^{T} P(o_{n,t} | \mu_n) \Big) P(\mu_n | s_n)$$

Block side inference occurs at the timescale of blocks, based on knowledge of the history of true stimuli sides. Our Bayesian baselines assume that the block transitions are Markov (instead of semi-Markov). Both baselines perform Bayesian filtering [25] to compute the block side posterior by alternating between a joint and a conditional, and normalizing after each trial:

$$P(b_n, s_n | s_{\leq n-1}) = \sum_{b_{n-1}} P(s_n | b_n) P(b_n | b_{n-1}) P(b_{n-1} | s_{\leq n-1})$$
$$P(b_n | s_{\leq n}) = \frac{P(b_n, s_n | s_{\leq n-1})}{\sum_{b_n} P(b_n, s_n | s_{\leq n-1})}$$

3 Results

3.1 RNN Behavior

3.1.1 RNN behavior matches ideal Bayesian observer behavior

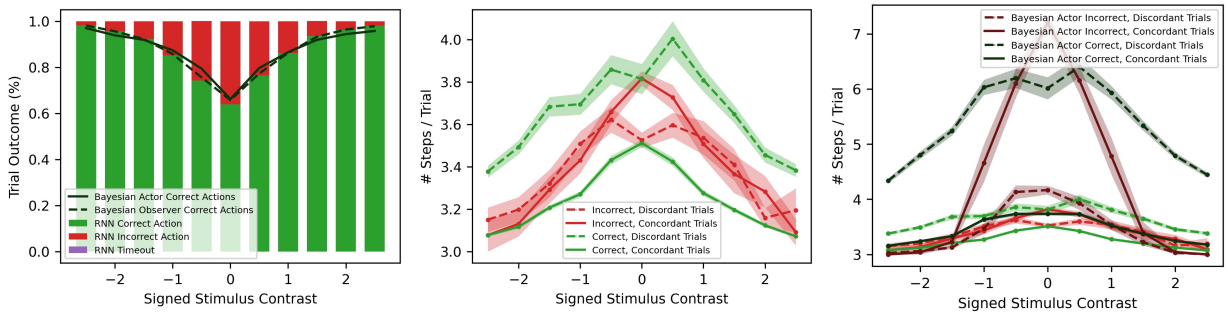

Figure 2: (a) RNN fraction of correct action almost matches Bayesian observer and Bayesian actor. (b) RNN chronometric curves show longer integration time on low contrast trials. (c) Bayesian actor chronometric curves show the actor responds significantly more slowly on low contrast trials than the RNN. RNN curves from (b) have been added for comparison.

We start by quantifying the performance of the RNN. Strikingly, the RNN achieves performance nearly matching the Bayesian observer (Fig. 2a); all three agents display similar accuracy as a function of trial stimulus strength μ_n: the fraction of correct actions is highest for strong stimulus contrast and lowest for weak stimulus contrast. Furthermore, the performance of all three agents is well above chance for zero-contrast trials, meaning all three exploit the block structure of the task.

Chronometric curves, which quantify how quickly the agents select an action as a function of trial stimulus strength, show that both the RNN and the Bayesian actor respond faster on concordant trials (when the stimulus side matches the block side, a higher probability event) than discordant trials (Figs. 2ab). While both the RNN and the Bayesian agents act more slowly for low trial stimulus contrast, the RNN acts significantly faster than the Bayesian actor on trials with low trial stimulus strength (Figs. 2ab. This suboptimal integration of within-trial evidence by the RNN partly explains its slightly worse performance than the Bayesian actor.

3.1.2 RNN leverages block prior when selecting actions

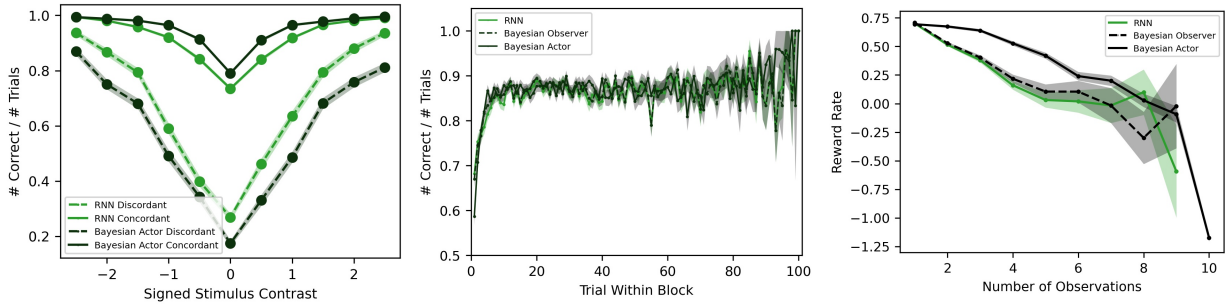

Figure 3: (a) Psychometric curves for RNN and Bayesian actor closely match. Low stimulus contrast values and discordant trial curves indicate the RNN disproportionally weights the likelihood. (b) RNN fraction of correct trials rapidly increases following a block change-point, closely matching the Bayesian observer. (c) The reward rate of the RNN nearly matches the reward rate of the Bayesian observer, but falls short of the Bayesian actor.

We next explored to what extent the RNN leverages the block prior to select actions. The RNN and the Bayesian actor both perform much better on concordant trials than discordant trials (Fig. 3a), but the discrepancy shrinks for large stimulus contrast values. Additionally, in both agents, the gap in stimulus side inference for concordant versus discordant trials is greater for low-contrast trials than for high-contrast trials (Fig. 3a), meaning that when the contrast strength is low, the prior dominates the likelihood, while at high contrasts, the likelihood dominates the prior. The RNN's behavior approaches that of the Bayesian actor, suggesting that the RNN weighs the stimulus side likelihood with the block prior near-optimally.

There are, however, two small quantitative difference indicating the RNN underweighs the prior. First, the concordant-discordant gap is smaller in the RNN than the Bayesian actor. Second, for zero-contrast stimuli, the Bayesian actor's accuracy directly reflects the block prior (0.2/0.8), while the RNN's accuracy is slightly contracted towards chance performance (.5/.5). This is likely not due to deficiencies with inferring the block prior, as the RNN's fraction of correct answers rapidly climbs following a block change-point (Fig. 3b), closely matching the Bayesian observer and the Bayesian actor, and therefore indicating that change-point detection in the RNN is near-optimal.

3.2 RNN Representations

3.2.1 RNN learns 2D dynamics to encode stimulus side and block side

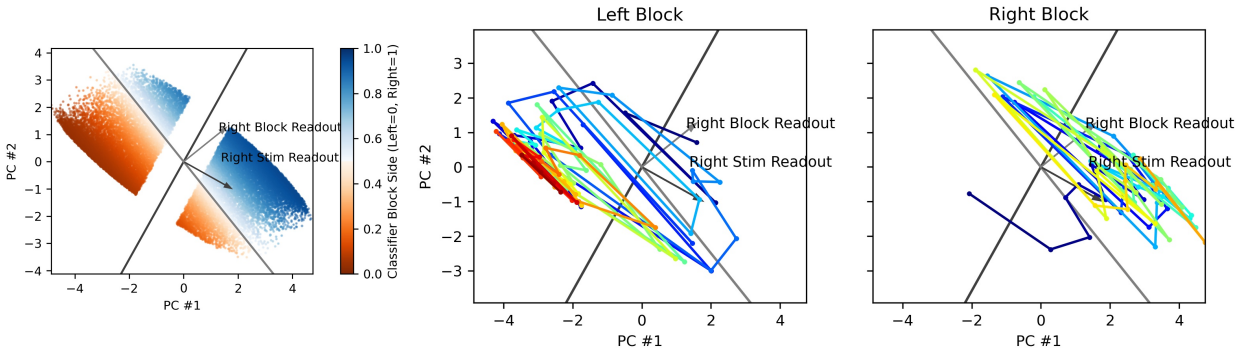

Figure 4: (a) Logistic regression classifies block side from RNN activity with 82.6% accuracy on 33% test data. The block readout and stimulus readout directions are non-orthogonal. (b-c) Example RNN state-space trajectories in a left block and a right block. Color: trial number within block (blue=early, red=late). The RNN activity moves quickly over the stimulus decision boundary and moves slowly along the block readout direction.

We next sought to characterize how the RNN's dynamics subserve inference. The first two principal components (PCs) of RNN activity explain 88.74% of the variance, suggesting it has learnt a low-dimensional solution. The RNN readout matrix W^{out} converts the hidden states $h_{n,t}$ into actions $a_{n,t}$, explicitly giving us the direction along which the RNN encodes its stimulus side belief; we term this the *stimulus readout* vector. 93.39% of the stimulus readout's length lies in the 2D PCA plane.

To identify how block side is encoded in RNN activity, we trained a logistic classifier to predict block side. This classifier had 82.6% accuracy on 33% heldout test data. A separate classifier for block side trained from only the 2-dimensional PCA plane of RNN activity reached 82.5% accuracy (Fig. 4a). In short, the RNN's PCA plane encompasses the two latent variables being inferred: these two dimensions are sufficient to decipher how the network solves the task. Importantly, the (right-side) block and (right-side) stimulus readouts are non-orthogonal (subtending an angle of 73° to each other in the high-dimensional RNN space, and 68° in the PCA plane). This deviation from orthogonality is modest but critical to how the network performs hierarchical inference (as we will explain below).

State-space trajectories (Fig. 4bc) in the PCA plane (trajectories indicate how the RNN state evolves across trials in a block starting at a block change, showing a time-point per trial) show that the state jumps across the stimulus decision boundary on the timescale of trials whereas state evolves slowly and relatively

steadily along the block readout, moving from the wrong block side at the start of a block (encoding the block just before the change-point) to the correct block side.

3.2.2 Observations are integrated to infer stimulus side and block side

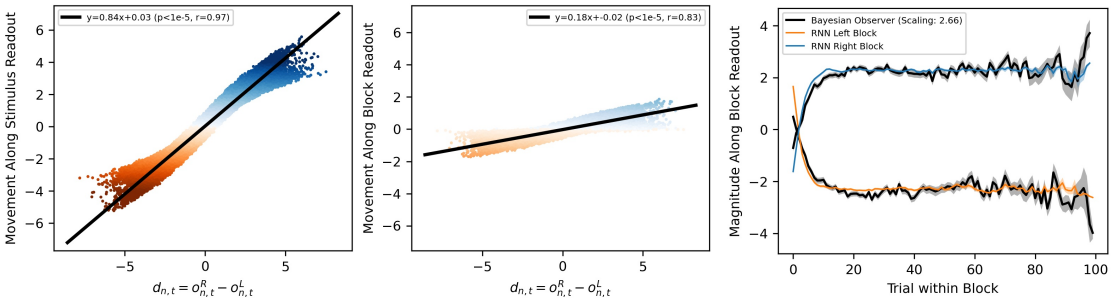

Figure 5: (a-b) Observations push the RNN state along the right trial readout and right block readout directions with two different amplitudes. (c) Instantaneous effects of observations are integrated along the block readout direction to encode the block side.

Based on state-space trajectories, we hypothesized that the RNN infers both the stimulus side and block side by integrating observations at different rates (faster for stimulus inference, slower for block inference). We confirmed this by plotting the change of RNN state along the (right-side) stimulus and (right-side) block directions, as a function of the difference in the right and left observation values $d_{n,t} = o^R_{n,t} - o^L_{n,t}$ (Fig. 5ab). Both had positive slopes (0.84 for stimulus, 0.18 for block) with $p < 1e-5$, confirming that evidence moves the state appropriately along the stimulus readout and block readout vectors. The respective magnitude of these two slopes (the stimulus slope is ≈ 5 times the block slope) match our expectation that stimulus side inference changes more rapidly with observations than the block side.

These instantaneous effects are integrated to infer the block side. The component of RNN activity along the right block readout vector closely matches the average block side posterior estimate of the Bayesian observer (Fig. 5c), up to an arbitrary scaling parameter that we determined through an ordinary least squares fit between the magnitude of the RNN state along the block readout vector and the Bayesian observer's block posterior (the actor is identical to the observer in tracking the block side). This result reveals how the RNN performs efficient change-point detection of the block side.

However, when we compared the RNN's block side belief with the Bayesian observer's block posterior on a trial-by-trial basis, we observed a difference: The RNN block side belief, though matching the observer when averaged across trials, fluctuates more on a trial-by-trial basis (Fig. 6a). These fluctuations are driven by within-trial evidence: single right- (left-) sided trials move the RNN's block belief to the right (left) more strongly than they move the Bayesian observer's block posterior (Fig. 6b). This discrepancy is due to an induced dynamical coupling in the RNN between stimulus and block inference. Specifically, the RNN must update its block and stimulus beliefs simultaneously at each step and therefore cannot decouple the two inference problems, whereas both baselines decouple the two inference problems by controlling *when* information is communicated.

3.3 RNN Mechanism

3.3.1 RNN dynamics and connectivity are consistent with bistable/line-attractor dynamics

Given our hypotheses for how the dynamics of the RNN perform inference, we turned our attention to identifying the circuit mechanism(s). Ordering the hidden units using hierarchical clustering with Pearson correlation as the similarity metric revealed two clear subpopulations (Fig. 7a). Units in one subpopulation are strongly correlated with other units in the same subpopulation and strongly anticorrelated with units in the other subpopulation. Applying the same ordering to the recurrent weight matrix revealed self-excitation within each subpopulation and mutual inhibition between subpopulations (Fig. 7b).

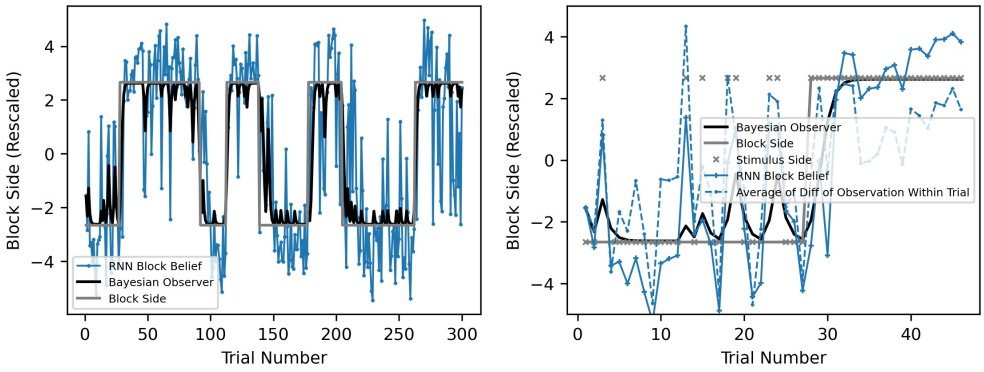

Figure 6: (a) RNN activity magnitude along the block readout closely matches Bayesian observer's block posterior and the true block side. (b) Significant jumps in RNN activity magnitude along the block readout correspond to trials with large jumps in evidence, given by $o^R_{n,t} - o^L_{n,t}$.

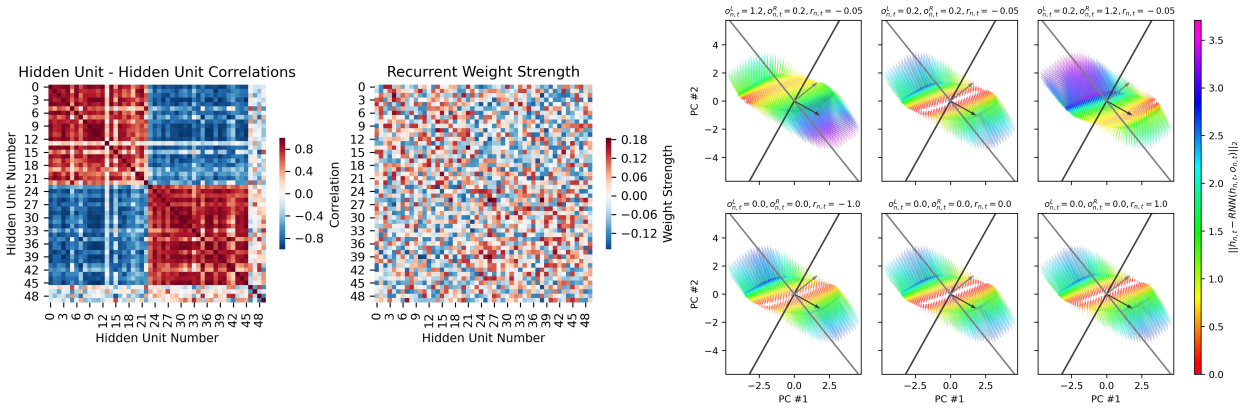

Figure 7: (a) Ordering RNN units based on Pearson correlation reveals two anticorrelated subpopulations. (b) Applying the same correlation-based ordering reveals self-excitatory, mutually-inhibitory connections between subpopulations. (c) RNN vector fields (evolving over one timestep) under six input conditions.

This is strongly reminiscent of circuits in the brain capable of producing 1-dimensional line-attractor dynamics or bistable attractor dynamics, depending on the strength of the excitatory and inhibitory recurrent connections [26, 18, 32, 17, 7, 23, 21]. Circuits with these dynamics have been studied in tasks involving a single variable, but not in tasks involving two (interacting) variables. We now explain how the same circuit can perform hierarchical inference on two latent variables.

Visualizing the RNN vector fields (Fig. 7c) better reveals the behavior of the system. When the stimulus is strong, the network exhibits one of two attractors, in the right-block right-stimulus quadrant or in the left-block left-stimulus quadrant. When the stimulus is absent and there is no feedback, the network exhibits a 1-dimensional line attractor. The line attractor is mainly aligned with the block readout, which allows the RNN to preserve its block side belief across trials. The persistent representation of the block side must be continuous even though the block side itself takes one of two discrete values, because the block belief is continuous-valued. The line attractor has a small projection along the stimulus readout, which translates the block belief into a stimulus prior for the next trial by biasing the RNN to select the concordant stimulus side in its decision.

Surprisingly, feedback about whether the selected action was correct has little effect (Fig. 7c). We speculate this is because the RNN's actions are typically correct, rendering feedback less useful, and because combining feedback with the chosen action to determine the correct action requires more complex computation that is harder to learn.

3.3.2 Model compression by distillation of hidden unit representations

We would like to extract a low-dimensional, interpretable model of the RNN to reveal the RNN's effective circuit. To do so, we propose a variation of knowledge distillation [6, 4, 12] in which we train a small RNN with output states $\hat{z}_t$ to reproduce the *hidden states* of the original RNN. This differs from conventional distillation in which the small network is trained on the *output* probabilities or logits of the original model (but a similar technique was used in BERT Transformer networks for NLP [13, 11]). We call our approach *Representation and Dynamics Distillation* (RADD). Specifically, we train the parameters A', B' of a small RNN to recapitulate a low-dimensional projection of the original RNN's hidden state dynamics ($\{z_t \equiv Ph_t\}_{t=1}^{T}$, starting from initial condition $\hat{z}_1 = Ph_1$, where P is the $M \times N$-dimensional dimension-reducing projection matrix[1]. After selecting a projection P, the distilled RNN is trained on the following L_2 loss using conventional methods:

$$\arg\min_{A',B'} \sum_{t=1}^{T} ||z_t - f(A'\hat{z}_{t-1} + B'o_t)||^2.$$

3.3.3 Reduced model preserves RRN geometry and recovers meaningful parameters

The dynamics of the original RNN are well-captured by its first two principal components, suggesting that a mere 2-unit distilled RNN might suffice to capture its dynamics. Indeed, a 2-unit distilled RNN (with rows in P set to the block and trial side readout vectors; similar results are obtained with P set to the first two principal components) emulates the original RNN well: The Δ-timestep decoherence in state is the same across three systems, $||h_{t+\Delta} - h_t||$, $||z_{t+\Delta} - z_t||$ (Fig. 8a). States in the distilled RNN evolve in a qualitatively similar way across the trial and block boundaries over multiple trials as the original RNN (Fig. 8b). Moreover, depending on the magnitude of the distilled system's readout vector (a free parameter), the distilled system can slightly outperform the full RNN (distilled 86.87%, full 85.50%) on the task. The distilled 2-unit RNN recognizes blocks in the same way as the original RNN, whereas a 2-unit RNN trained directly on the task itself fails to recognize blocks (Fig. 8c) despite being trained for four times as many gradient steps.

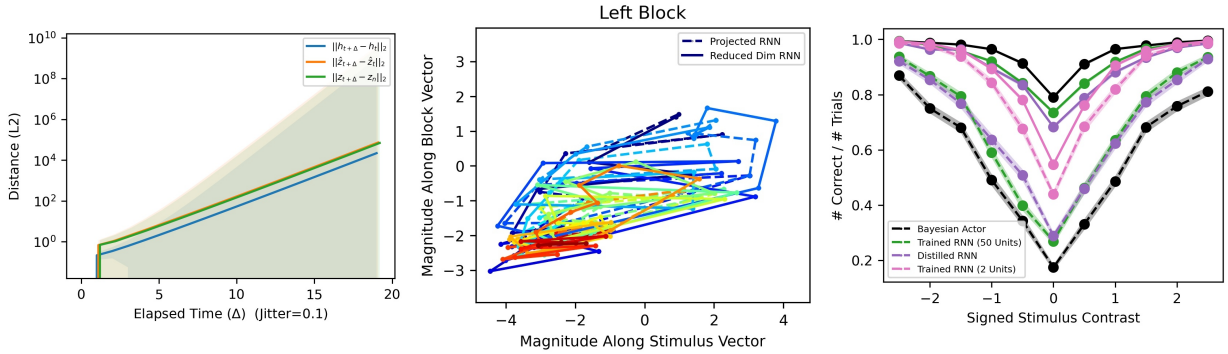

Figure 8: (a) Distance decoheres at the same rate in the three state spaces: RNN, projected RNN, distilled RNN. Projected RNN and distilled RNN have nearly identical values (horizontal displacement added to make both visible). (b) Distilled RNN state space trajectories closely match projected RNN trajectories. (c) Comparative performance of 50-unit original RNN, 2-unit distilled RNN, and 2-unit task-trained RNN.

The distilled RNN, whose units correspond to stimulus and block side beliefs, has sensible parameters:

$$\hat{z}_{n,t} = \begin{bmatrix} \text{Stimulus Belief}_{n,t} \\ \text{Block Belief}_{n,t} \end{bmatrix} = \tanh\left(\begin{bmatrix} 0.54 & 0.31 \\ 0.19 & 0.84 \end{bmatrix} \hat{z}_{n,t-1} + \begin{bmatrix} -0.19 & 0.20 & 0.005 \\ -0.04 & 0.04 & 0.021 \end{bmatrix} \begin{bmatrix} o^L_{n,t} \\ o^R_{n,t} \\ r_{n,t} \end{bmatrix} \right).$$

The recurrent weights show that the stimulus belief and block belief reinforce one another, and both decay to 0 without input observations. The input weights show that observations drive the stimulus and block side

[1]We assume that the N-dimensional states of the original RNN lie in a D-dimensional linear subspace $\mathbb{R}^D \subset \mathbb{R}^N$. The projection P can be selected such that $D \leq M \leq N$ and such that its nullspace does not intersect this D-dimensional subspace. A good choice for P are the top principal vectors of the original RNN states.

beliefs in a common direction, but that the movement caused by a single observation is 5 times greater along the stimulus direction than the block direction. The state space trajectories (Fig. 8b) visually agree with this intuition: each left-to-right (stimulus side) movement corresponds to a small up-right (block side) movement. Further, the feedback input receives negligible weighting, consistent with our earlier observation.

4 Discussion

In conclusion, RNNs attain near-optimal performance on a hierarchical inference task, as measured against Bayesian observers and actors that have full knowledge of the task. We have characterized the representations, dynamics, and mechanisms underlying inference in the RNN. In future work, we will leverage these models, together with work being developed by others, to better understand mouse behavior and neural representations. We expect it will be fruitful to explore RADD in the context of reinforcement learning [24, 9].

5 Broader Impact

We appreciate NeurIPS asking researchers to evaluate the ethical dimensions of their work. We believe that the bulk of this work, focused on understanding mechanisms of how neural networks solve basic inference problems, does not have any direct or detrimental social ramifications. It is possible that RADD, as a technique for more interpretable ANNs, could be used in other scenarios to better understand biases learned in RNNs.

References

[1] Larry F. Abbott, Dora E. Angelaki, Matteo Carandini, Anne K. Churchland, Yang Dan, Peter Dayan, Sophie Deneve, Ila Fiete, Surya Ganguli, Kenneth D. Harris, Michael Häusser, Sonja Hofer, Peter E. Latham, Zachary F. Mainen, Thomas Mrsic-Flogel, Liam Paninski, Jonathan W. Pillow, Alexandre Pouget, Karel Svoboda, Ilana B. Witten, and Anthony M. Zador. "An International Laboratory for Systems and Computational Neuroscience". en. In: *Neuron* 96.6 (Dec. 2017), pp. 1213–1218. ISSN: 08966273. DOI: 10.1016/j.neuron.2017.12.013. URL: https://linkinghub.elsevier.com/retrieve/pii/S0896627317311364 (visited on 05/23/2020).

[2] Ryan Prescott Adams and David J. C. MacKay. "Bayesian Online Changepoint Detection". en. In: *arXiv:0710.3742 [stat]* (Oct. 2007). arXiv: 0710.3742. URL: http://arxiv.org/abs/0710.3742 (visited on 05/22/2020).

[3] Leila Arras, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. "Explaining Recurrent Neural Network Predictions in Sentiment Analysis". en. In: *arXiv:1706.07206 [cs, stat]* (Aug. 2017). arXiv: 1706.07206. URL: http://arxiv.org/abs/1706.07206 (visited on 06/03/2020).

[4] Jimmy Ba and Rich Caruana. "Do Deep Nets Really Need to be Deep?" In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger. Curran Associates, Inc., 2014, pp. 2654–2662. URL: http://papers.nips.cc/paper/5484-do-deep-nets-really-need-to-be-deep.pdf.

[5] Andrea Banino, Caswell Barry, Benigno Uria, Charles Blundell, Timothy Lillicrap, Piotr Mirowski, Alexander Pritzel, Martin J. Chadwick, Thomas Degris, Joseph Modayil, Greg Wayne, Hubert Soyer, Fabio Viola, Brian Zhang, Ross Goroshin, Neil Rabinowitz, Razvan Pascanu, Charlie Beattie, Stig Petersen, Amir Sadik, Stephen Gaffney, Helen King, Koray Kavukcuoglu, Demis Hassabis, Raia Hadsell, and Dharshan Kumaran. "Vector-based navigation using grid-like representations in artificial agents". en. In: *Nature* 557.7705 (May 2018), pp. 429–433. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-018-0102-6. URL: http://www.nature.com/articles/s41586-018-0102-6 (visited on 06/03/2020).

[6] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. "Model compression". In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2006, pp. 535–541.

[7] Rishidev Chaudhuri and Ila Fiete. "Computational principles of memory". en. In: *Nature Neuroscience* 19.3 (Mar. 2016), pp. 394–403. ISSN: 1097-6256, 1546-1726. DOI: 10.1038/nn.4237. URL: http://www.nature.com/articles/nn.4237 (visited on 06/02/2020).

[8] Christopher J Cueva and Xue-Xin Wei. "Emergence of grid-like representations by training recurrent neural networks to perform spatial localization." en. In: *International Conference on Learning Representations* (2018), p. 19.

[9] Wojciech Marian Czarnecki, Razvan Pascanu, Simon Osindero, Siddhant M. Jayakumar, Grzegorz Swirszcz, and Max Jaderberg. "Distilling Policy Distillation". en. In: *arXiv:1902.02186 [cs, stat]* (Feb. 2019). arXiv: 1902.02186. URL: http://arxiv.org/abs/1902.02186 (visited on 06/04/2020).

[10] Will Dabney, Zeb Kurth-Nelson, Naoshige Uchida, Clara Kwon Starkweather, Demis Hassabis, Rémi Munos, and Matthew Botvinick. "A distributional code for value in dopamine-based reinforcement learning". en. In: *Nature* 577.7792 (Jan. 2020), pp. 671–675. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-019-1924-6. URL: http://www.nature.com/articles/s41586-019-1924-6 (visited on 04/01/2020).

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". en. In: *arXiv:1810.04805 [cs]* (May 2019). arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805 (visited on 06/04/2020).

[12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. "Distilling the Knowledge in a Neural Network". en. In: *arXiv:1503.02531 [cs, stat]* (Mar. 2015). arXiv: 1503.02531. URL: http://arxiv.org/abs/1503.02531 (visited on 06/04/2020).

[13] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. "TinyBERT: Distilling BERT for Natural Language Understanding". en. In: *arXiv:1909.10351 [cs]* (Dec. 2019). arXiv: 1909.10351. URL: http://arxiv.org/abs/1909.10351 (visited on 06/04/2020).

[14] Ian D. Jordan, Piotr Aleksander Sokol, and Il Memming Park. "Gated recurrent units viewed through the lens of continuous time dynamical systems". en. In: *arXiv:1906.01005 [cs, stat]* (June 2019). arXiv: 1906.01005. URL: http://arxiv.org/abs/1906.01005 (visited on 06/03/2020).

[15] Ingmar Kanitscheider and Ila Fiete. *Emergence of dynamically reconfigurable hippocampal responses by learning to perform probabilistic spatial reasoning*. en. preprint. Neuroscience, Dec. 2017. DOI: 10.1101/231159. URL: http://biorxiv.org/lookup/doi/10.1101/231159 (visited on 06/03/2020).

[16] Ingmar Kanitscheider and Ila Fiete. "Training recurrent networks to generate hypotheses about how the brain solves hard navigation problems". en. In: (), p. 10.

[17] Charles D. Kopec, Jeffrey C. Erlich, Bingni W. Brunton, Karl Deisseroth, and Carlos D. Brody. "Cortical and Subcortical Contributions to Short-Term Memory for Orienting Movements". en. In: *Neuron* 88.2 (Oct. 2015), pp. 367–377. ISSN: 08966273. DOI: 10.1016/j.neuron.2015.08.033. URL: https://linkinghub.elsevier.com/retrieve/pii/S0896627315007278 (visited on 05/20/2020).

[18] C. K. Machens. "Flexible Control of Mutual Inhibition: A Neural Model of Two-Interval Discrimination". en. In: *Science* 307.5712 (Feb. 2005), pp. 1121–1124. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1104171. URL: https://www.sciencemag.org/lookup/doi/10.1126/science.1104171 (visited on 04/11/2020).

[19] Niru Maheswaranathan and David Sussillo. "How recurrent networks implement contextual processing in sentiment analysis". en. In: *arXiv:2004.08013 [cs, stat]* (Apr. 2020). arXiv: 2004.08013. URL: http://arxiv.org/abs/2004.08013 (visited on 05/23/2020).

[20] Niru Maheswaranathan, Alex Williams, Matthew Golub, Surya Ganguli, and David Sussillo. "Reverse engineering recurrent networks for sentiment classification reveals line attractor dynamics". en. In: *Neural Information Processing Systems* (), p. 10.

[21] Valerio Mante, David Sussillo, Krishna V. Shenoy, and William T. Newsome. "Context-dependent computation by recurrent dynamics in prefrontal cortex". en. In: *Nature* 503.7474 (Nov. 2013), pp. 78–84. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature12742. URL: http://www.nature.com/articles/nature12742 (visited on 06/03/2020).

[22] Elyse H. Norton, Luigi Acerbi, Wei Ji Ma, and Michael S. Landy. "Human online adaptation to changes in prior probability". en. In: *PLOS Computational Biology* 15.7 (July 2019). Ed. by Ulrik R. Beierholm, e1006681. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1006681. URL: https://dx.plos.org/10.1371/journal.pcbi.1006681 (visited on 05/23/2020).

[23] Alex T. Piet, Jeffrey C. Erlich, Charles D. Kopec, and Carlos D. Brody. "Rat Prefrontal Cortex Inactivations during Decision Making Are Explained by Bistable Attractor Dynamics". en. In: *Neural Computation* 29.11 (Nov. 2017), pp. 2861–2886. ISSN: 0899-7667, 1530-888X. DOI: 10.1162/neco_a_01005. URL: http://www.mitpressjournals.org/doi/abs/10.1162/neco_a_01005 (visited on 04/14/2020).

[24] Andrei A. Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. "Policy Distillation". en. In: *arXiv:1511.06295 [cs]* (Jan. 2016). arXiv: 1511.06295. URL: http://arxiv.org/abs/1511.06295 (visited on 06/04/2020).

[25] Maneesh Sahani. *Latent Variable Models for Time Series*. Lecture. University College London, 2017.

[26] H. S. Seung. "How the brain keeps the eyes still". en. In: *Proceedings of the National Academy of Sciences* 93.23 (Nov. 1996), pp. 13339–13344. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.93.23.13339. URL: http://www.pnas.org/cgi/doi/10.1073/pnas.93.23.13339 (visited on 06/02/2020).

[27] Ben Sorscher, Gabriel Mel, Surya Ganguli, and Samuel Ocko. "A unified theory for the origin of grid cells through the lens of pattern formation". en. In: (), p. 11.

[28] David Sussillo and Omri Barak. "Opening the Black Box: Low-Dimensional Dynamics in High-Dimensional Recurrent Neural Networks". en. In: *Neural Computation* 25.3 (Mar. 2013), pp. 626–649. ISSN: 0899-7667, 1530-888X. DOI: 10.1162/NECO_a_00409. URL: http://www.mitpressjournals.org/doi/10.1162/NECO_a_00409 (visited on 06/03/2020).

[29] The International Brain Laboratory, Valeria Aguillon-Rodriguez, Dora E. Angelaki, Hannah M. Bayer, Niccolò Bonacchi, Matteo Carandini, Fanny Cazettes, Gaelle A. Chapuis, Anne K. Churchland, Yang Dan, Eric E. Dewitt, Mayo Faulkner, Hamish Forrest, Laura M. Haetzel, Michael Hausser, Sonja B. Hofer, Fei Hu, Anup Khanal, Christopher S. Krasniak, Inês Laranjeira, Zachary F. Mainen, Guido T. Meijer, Nathaniel J. Miska, Thomas D. Mrsic-Flogel, Masayoshi Murakami, Jean-Paul Noel, Alejandro Pan-Vazquez, Josh I. Sanders, Karolina Z. Socha, Rebecca Terry, Anne E. Urai, Hernando M. Vergara, Miles J. Wells, Christian J. Wilson, Ilana B. Witten, Lauren E. Wool, and Anthony Zador. *A standardized and reproducible method to measure decision-making in mice*. en. preprint. Neuroscience, Jan. 2020. DOI: 10.1101/2020.01.17.909838. URL: http://biorxiv.org/lookup/doi/10.1101/2020.01.17.909838 (visited on 05/23/2020).

[30] Fu-Sheng Tsung and Garrison W Cottrell. "Phase-Space Learning". en. In: (), p. 8.

[31] Jane X. Wang, Zeb Kurth-Nelson, Dharshan Kumaran, Dhruva Tirumala, Hubert Soyer, Joel Z. Leibo, Demis Hassabis, and Matthew Botvinick. "Prefrontal cortex as a meta-reinforcement learning system". en. In: *Nature Neuroscience* 21.6 (June 2018), pp. 860–868. ISSN: 1097-6256, 1546-1726. DOI: 10.1038/s41593-018-0147-8. URL: http://www.nature.com/articles/s41593-018-0147-8 (visited on 04/11/2020).

[32] K.-F. Wong. "A Recurrent Network Mechanism of Time Integration in Perceptual Decisions". en. In: *Journal of Neuroscience* 26.4 (Jan. 2006), pp. 1314–1328. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.3733-05.2006. URL: http://www.jneurosci.org/cgi/doi/10.1523/JNEUROSCI.3733-05.2006 (visited on 05/20/2020).