Two-Area RNN: Representations for Context-Dependent Decisions

Jizhou Guo, Liting Pang, Zhaoyu Zhu & Ziyi Xu
Zhiyuan College
Shanghai Jiao Tong University
{sjtu18640985163, pangliting, zzy12345, xuziyi-0302}@sjtu.edu.cn

Abstract

This paper presents the Two-Area Recurrent Neural Network (2aRNN) model, which extends the understanding of context-dependent decision-making processes by simulating the neural dynamics observed in rhesus monkeys' prefrontal cortex during visual discrimination tasks. The 2aRNN model compartmentalizes the processing of color and motion stimuli and context information into separate neural network regions, allowing for a detailed examination of how the brain integrates multimodal sensory information under varying task demands. Through the replication of dynamic responses and decision behaviors, the model not only explains the flexible selection and integration of sensory inputs based on context but also offers insights into the neural computational framework underlying complex decision-making. The study further explores the impact of temporal scale separation by adjusting the time constants of the two network areas, revealing the model's capacity to handle complex temporal dependencies and its implications for cognitive neuroscience. Our code is available at https://github.com/Aster2024/2aRNN.

1 Introduction

In the field of cognitive neuroscience, understanding how the brain processes and integrates complex sensory information to guide decision-making is a central issue. Particularly in primates, this process involves the coordinated effort of multiple brain regions, including the critical role of the prefrontal cortex (PFC) in decision-making. In recent years, a series of innovative experiments conducted on rhesus monkeys have allowed scientists to delve into the neural mechanisms underlying these advanced cognitive functions (Mante et al., 2013).

The starting point for this study is a series of experiments recording neural activity in rhesus monkeys during visual discrimination tasks. These experiments involved two adult male rhesus monkeys (designated as A and F), which were trained to excel in a two-alternative, forced-choice visual discrimination task. In these tasks, the monkeys were required to make rapid judgments on the direction of motion or color of random dots and report their decisions through saccadic eye movements. By implanting electrodes in the monkeys' heads, researchers were able to monitor neural activity associated with eye movements, as well as electrophysiological signals related to the task.

These experiments have revealed the dynamic changes in the prefrontal cortex when processing motion and color information, and how this information is selectively integrated to produce behavioral outputs. Particularly, how monkeys adjust their processing strategies for sensory information based on the current task demands is evident in the patterns of neural activity. These findings not only provide crucial clues for our understanding of how the brain handles multimodal sensory information but also offer an experimental foundation for developing computational models that can simulate these complex neural dynamics.

Figure 1 demonstrates the dynamic responses of a Recurrent Neural Network (RNN) model that was trained to simulate how the prefrontal cortex (PFC) processes evidence related to choice and integrates this evidence into decision-making. The model successfully replicated the population response trajectories observed in the PFC of macaque monkeys, showing that the same sensory inputs (color and motion) lead to different movements along fixed axes under varying contexts. This

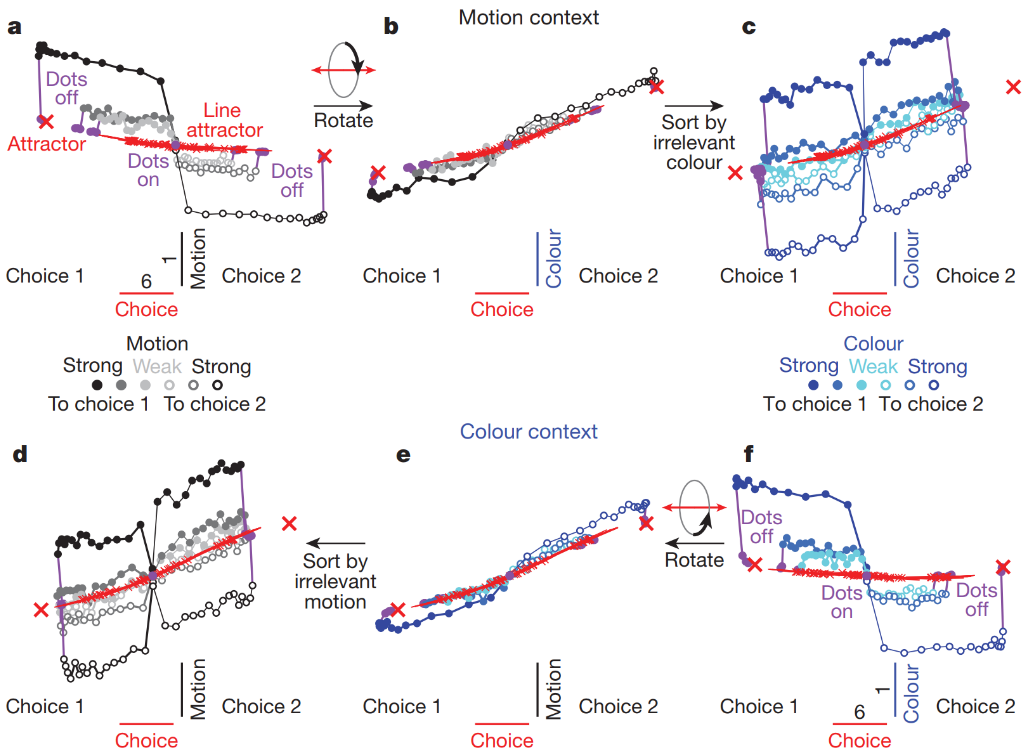


Figure 1: Model dynamics and fixed points analysis of 1aRNN.

reveals that selection and integration are two aspects of a single dynamic process unfolding within the same prefrontal circuits. In this way, the model not only explains how the PFC flexibly selects and integrates sensory inputs based on context but also provides a potential neural computational framework for understanding context-dependency in complex decision-making processes.

Building on the conclusions of Figure 1, this study aims to construct a 2aRNN (Two-Area Recurrent Neural Network) model to simulate the processing of stimuli (color and motion) and context information in separate compartments. Our goal is to validate the effectiveness of the 2aRNN model in simulating complex decision-making tasks by replicating the dynamic responses and decision behaviors observed in Figure 1. Through this compartmentalized simulation, we expect to reveal how the brain processes and integrates multimodal sensory information and makes adaptive decisions in changing environments, achieving conclusions similar to those presented in Figure 1. Our code is available at https://github.com/Aster2024/2aRNN.

2 Related Works [1]

2.1 Context-Dependent Decision-Making Models

Context-dependent decision-making is a fundamental aspect of human cognition that has garnered significant attention in cognitive science and neuroscience research (Lloyd & Leslie, 2013; Waskom et al., 2017). Notably, (Mante et al., 2013) proposed a context-dependent decision-making model to simulate the intricate behavior of the prefrontal cortex, advancing our understanding of neural mechanisms underlying adaptive decision processes.

[1]This corresponds to Problem 1.

2.2 Task-Oriented RNN Training

Recurrent Neural Networks (RNNs) (Medsker et al., 2001) have emerged as powerful tools for modeling sequential data, demonstrating remarkable success across various domains, including complex decision-making processes (Zhang et al., 2021). Task-oriented RNN training, a specialized approach to optimizing these networks, involves tailoring the learning process to specific objectives or tasks. This methodology is particularly relevant when modeling the sophisticated cognitive processes described in (Mante et al., 2013), as it allows for the precise calibration of neural networks to mimic observed neurobiological phenomena.

2.3 One-Area RNN

As shown in Figure 2, the article modeled PFC responses with an RNN defined by the following equations:

$$\tau \frac{\mathrm{d}\mathbf{h}}{\mathrm{d}t} = -\mathbf{h} + \mathbf{W}^{\mathrm{rec}}\phi(\mathbf{h}) + \mathbf{W}^{\mathrm{stim}}\mathbf{x}^{\mathrm{stim}} + \mathbf{W}^{\mathrm{ctx}}\mathbf{x}^{\mathrm{ctx}} + \xi$$

$$\phi(x) = \tanh(x), \quad \xi(t) \sim \mathcal{N}(0,1) \quad \text{i.i.d.}$$

$$\mathbf{y} = \mathbf{W}^{\mathrm{out}}\phi(\mathbf{h})$$

The 1aRNN model provides a framework for investigating how neural networks selectively integrate sensory inputs from different contexts by modeling neural activity within a single region. As shown in Fig 2, the inputs to the model are the sensory stimulus input signals x^{stim} (color and motion) and the task context input signal x^{ctx}, and the output was either $+1$ or -1, and corresponded to the correct choice given the inputs and the context.

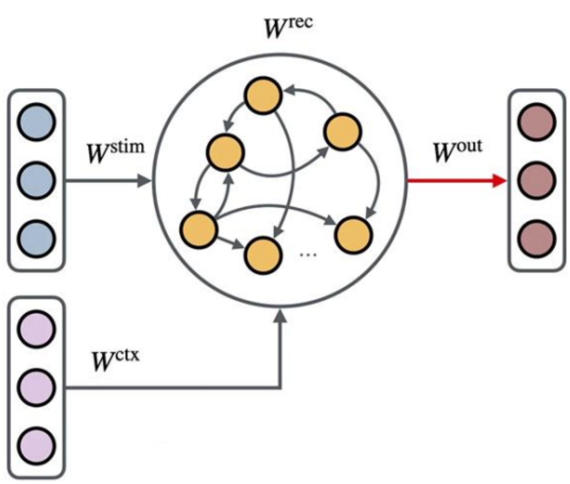


Figure 2: 1aRNN

The optimizer used to train the model is Adam (Kingma, 2014), which is a first-order gradient-based optimization method that uses an adaptive learning rate to update the model's weights.The Adam optimizer combines the strengths of both RMSprop and Momentum optimization algorithms, and is a widely used deep learning optimization algorithm.The loss function for training takes the mean square error loss function

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2,$$

where y_i is the true value, $\hat{y}_i$ is the predicted value, and n is the number of samples. Accuracy is calculated by comparing the selection of model output with metadata['action'] with the following formula

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}.$$

When using the model to simulate a monkey making a decision, we divide the task into four phases corresponding to times $[300, 1000, 900, 500]$ (milliseconds). The specific task content of each phase is as follows:

- Fixation (300): this is the initial phase of the task, which usually indicates that the target or the participant's eyes are focused on a fixation point before the stimulus is presented. In neuroscience tasks, this phase is usually used to wait or prepare to receive the stimulus. This is when the context information x^{cxt} is started to be provided to the model and is provided at each stage.
- Stimulus (1000): in this stage the stimuli x^{stm} (motion, color) will be provided and the model receives processing to deal with these stimuli.
- Delay (900): This stage is usually a "delay" stage, during which the model may need to maintain memory of previous stimuli (e.g., a short-term memory task) without receiving new stimuli.
- Response (500): The final phase is the response phase, during which the model needs to make a decision based on the previous stimulus and/or context, and give the appropriate choice (1 or -1).

3 Approach - Two-Area RNN

The One-Area RNN, while capable of modeling basic decision-making processes, has limitations in capturing the complexity of context-dependent decisions. Its single-area architecture may struggle to effectively separate context processing from decision formation, potentially leading to suboptimal performance in tasks requiring flexible adaptation to changing contexts. To address these limitations, we propose the Two-Area RNN model. This modular structure, inspired by the distinct functional areas observed in the prefrontal cortex (Mante et al., 2013), allows for a more nuanced representation of context-dependent information processing. By segregating the network into two interacting areas, we hypothesize that the Two-Area RNN can more effectively integrate contextual information with sensory inputs, leading to more robust and flexible decision-making in context-dependent tasks.

To mimic the partitioned thinking of the brain, we construct a 2aRNN (two-region recurrent neural network) model that simulates the processing of stimuli (color and motion) and contextual information in different regions. As shown in Figure 3, the article modeled PFC responses with a

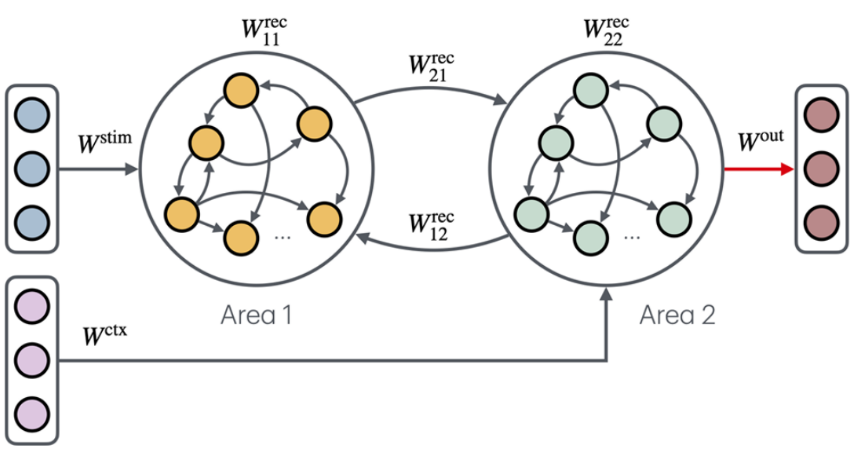


Figure 3: 2aRNN

Two-Area-RNN defined by the following equations:

$$\tau_1 \frac{\mathrm{d}\mathbf{h}_1}{\mathrm{d}t} = -\mathbf{h}_1 + \mathbf{W}_{11}^{\mathrm{rec}}\phi(\mathbf{h}_1) + \mathbf{W}_{12}^{\mathrm{rec}}\phi(\mathbf{h}_2) + \mathbf{W}^{\mathrm{stim}}\mathbf{x}^{\mathrm{stim}} + \xi_1$$

$$\tau_2 \frac{\mathrm{d}\mathbf{h}_2}{\mathrm{d}t} = -\mathbf{h}_2 + \mathbf{W}_{21}^{\mathrm{rec}}\phi(\mathbf{h}_1) + \mathbf{W}_{22}^{\mathrm{rec}}\phi(\mathbf{h}_2) + \mathbf{W}^{\mathrm{ctx}}\mathbf{x}^{\mathrm{ctx}} + \xi_2$$

$$\phi(x) = \tanh(x), \quad \xi(t) \sim \mathcal{N}(0,1) \quad \text{i.i.d.}$$
$$\mathbf{y} = \mathbf{W}^{\text{out}}\phi(\mathbf{h})$$

The 2aRNN model consists of two RNNs corresponding to the network parameter matrices $\mathbf{W}_{11}^{\text{rec}}$ and $\mathbf{W}_{22}^{\text{rec}}$, which are passed through the $\mathbf{W}_{12}^{\text{rec}}$ and $\mathbf{W}_{21}^{\text{rec}}$ to communicate. The inputs to the model are sensory stimulus input signals x^{stim} (color and action) and task context input signals x^{ctx} to the first network (Area 1) and the second network (Area 2), respectively, and the outputs are either $+1$ or -1, corresponding to the correct choices under the stimulus and the context.

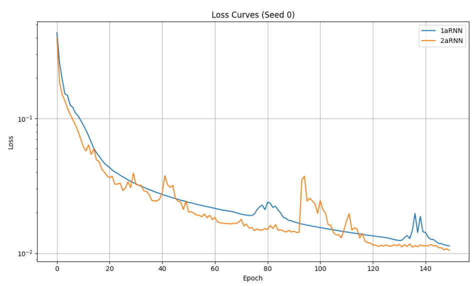
The training and simulation of the model are consistent with the training and simulation process of the 1aRNN.

4 Model Training[2]

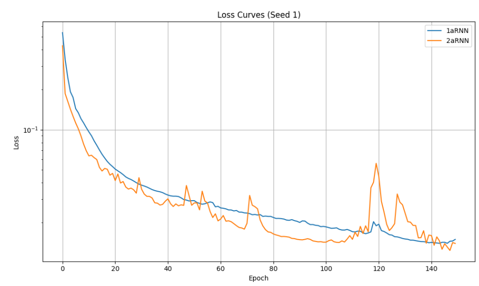
4.1 Method

We train under different seeds, with each training session consisting of 100 epochs. In each epoch, 100 trials are randomly generated. We use a batch size of 20 and employ the Adam optimizer (Kingma, 2014) with a learning rate of 1×10^{-3}.

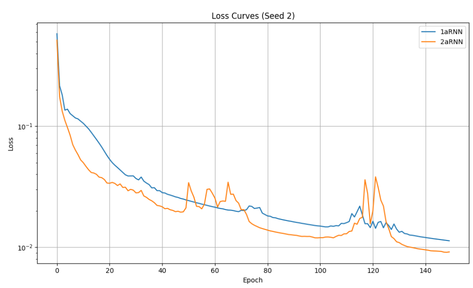
4.2 Result



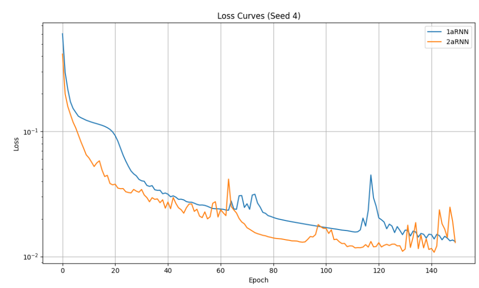
(a) seed1



(b) seed2



(c) seed3



(d) seed4

Figure 4: Training Loss of 1aRNN and 2aRNN

1. We can easily see the loss of 2aRNN model drop down much faster than 1aRNN model at the beginning phase of training.
2. The loss curve of 2aRNN model shows significant oscillation. The loss curve of 1aRNN is relatively smooth.
3. The two loss curve are both unstable at final training stage, they both have big oscillations at final training stage.

[2]This corresponds to Problem 2 & 3.

5 Reproduction of Fig.5 in (Mante et al., 2013)[3]

5.1 Method

To display trajectories in state space, we projected the population responses onto the axes of a subspace using linear regression.

$$h = \beta_1 \cdot choice.$$

$$h = \beta_2 \cdot motion.$$

$$h = \beta_3 \cdot colour.$$

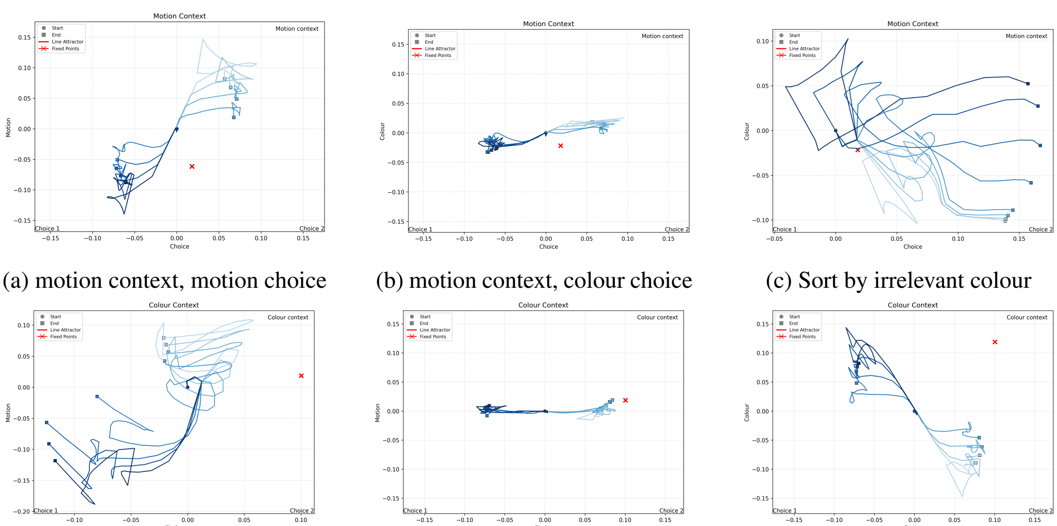
Here, h is the hidden layer, $\beta_1, \beta_2, \beta_3$ are viewed as the direction in state space.
Then we orthogonalized these directions with QR-decomposition.

$$B := [\beta_1, \beta_2, \beta_3] = QR.$$

Q is an orthogonal matrix, and R is an upper triangular matrix. We interpreted the first column of Q as the choice axis, the second column of Q as the motion axis and the third column of Q as the colour axis.
We projected the trajectories onto choice-motion and choice-colour plane respectively.

5.2 Result

5.2.1 1aRNN: reproduction of Fig 1



(a) motion context, motion choice

(b) motion context, colour choice

(c) Sort by irrelevant colour

(d) Sort by irrelevant motion

(e) colour choice, motion choice

(f) colour choice, colour choice

Figure 5: Model dynamics and fixed points analysis

We tried to reproduce fig 1, but it didn't quite work, and here's where we got the same conclusion as in the original article:

1. We found that in the motion (color) context, motion (color) -choice plane (Fig. 5a, 5f), given different motion (color) stimuli, the model gives the correct choice.
2. The trajectory shows the process of the model receiving the stimulus, processing the information, and making a decision. As shown in Fig. 5a, 5f, the trajectory starts at the coordinate origin, and after accepting the stimulus, it first moves rapidly along the corresponding direction of the motion (color) axis, then processes the information, moves rapidly along the corresponding direction of the choice axis, and finally gets the correct choice.

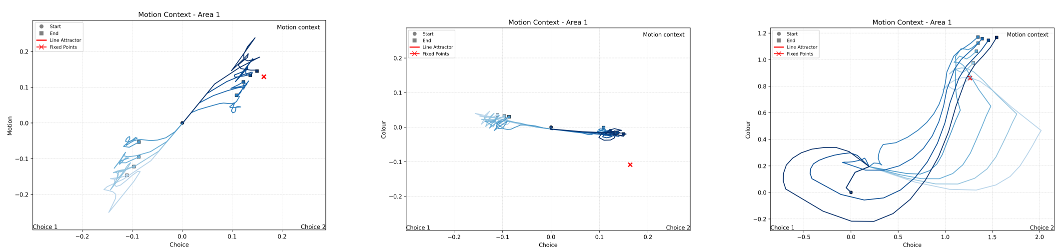
[3]This corresponds to Problem 4 & 5.

3. For different sizes of stimulus, the trajectory is also reflected. In Fig 5a, 5f, as the absolute value of the motion (color) stimulus increases, the projection of the trajectory on the motion (colour) axis is larger.
4. The projections of motion stimuli on the color axis (Fig. 1.b) and color stimuli on the motion axis (Fig. 1.e) are very close to each other and nearly converge to zero. This observation supports the conclusion that changes in one stimulus dimension have minimal impact on the network's representation of another stimulus dimension.
5. We found that in the motion(colour) context, color(motion)-choice plane (Fig. 5c, 5d), given different color (motion) stimuli, the model made consistent choices in the face of different stimuli. It indicates that the model will only focus on the information related to the context in the given context situation, and irrelevant stimuli in a given context have minimal impact on the network's response.

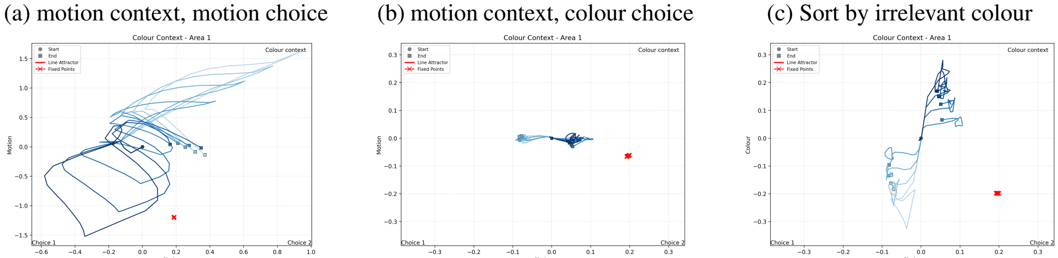
Here's what we failed to accomplish:

1. We were not able to find the same fixed points and linear attractor as in Fig. 1. we could only find one immobile point, which is indeed close to the end point on one side in the motion (color) context, motion (color)-choice plan, but in the other plots, it is in a very strange position. Therefore, the following is only a summary of the analysis of the immovable point in the original paper.

5.2.2 2aRNN: reproduction of Fig 1



(a) motion context, motion choice (b) motion context, colour choice (c) Sort by irrelevant colour

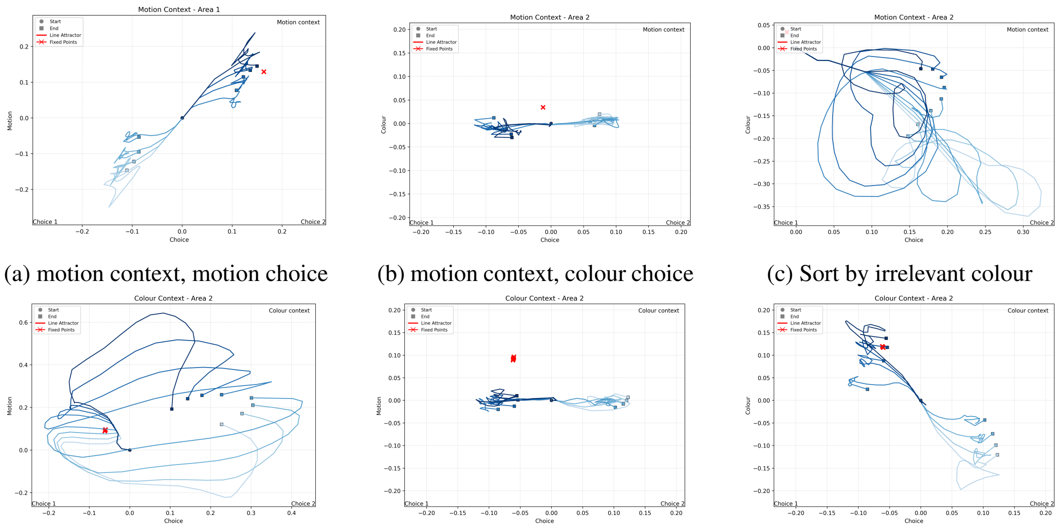


(d) Sort by irrelevant motion (e) colour choice, motion choice (f) colour choice, colour choice

Figure 6: Model dynamics and fixed points analysis area1

In 2aRNN we can find similar conclusions as in 1aRNN:

1. Conclusions 1–5 in the 1aRNN can all be seen in each region of the 2aRNN and will not be repeated.
2. Comparing 2aRNN with 1aRNN, it was found that the projections of 2aRNN under the motion (color) context for different sized color (motion) stimuli under the colour (motion) axis were much more indistinguishable, and were almost merged into a line or intertwined.(Fig. 6b,6e,7b,7e)
3. Comparing the two areas in the 2aRNN, it is found that for the whole trajectory, Area1 trajectory moves faster along the motion (color) axis when receiving stimulus at the very beginning, and it moves faster and farther along the choice axis when processing the information to make a choice in the middle. This is because Area1 is responsible for receiving



(a) motion context, motion choice (b) motion context, colour choice (c) Sort by irrelevant colour

(d) Sort by irrelevant motion (e) colour choice, motion choice (f) colour choice, colour choice

Figure 7: Model dynamics and fixed points analysis area2

the stimulus information, while Area2 is responsible for receiving the context information and outputting the decision at the end, so the trajectory graph reflects the division of labor between the two Areas.

5.3 FIXED POINT ANALYSIS

Since we were not able to find the correct fixed points and slow points, we can only summarize the fixed point analysis in the original article.

To discover the dynamical structure of the trained 1aRNN, the original paper found a large sample of the RNN's fixed points and slow points by minimizing the function:

$$\mathbf{F}(\mathbf{h}) = \frac{1}{2}\| - \mathbf{h} + \mathbf{W}^{\text{rec}}\phi(\mathbf{h}) + \mathbf{W}^{\text{ctx}}\mathbf{x}^{\text{ctx}}\|^2$$

corresponding to the 1aRNN update equation with input stimulus 0 and no noise.Since the network actually implements two dynamical systems, one for each context, we studied the dynamical systems for the motion context and the color context separately.

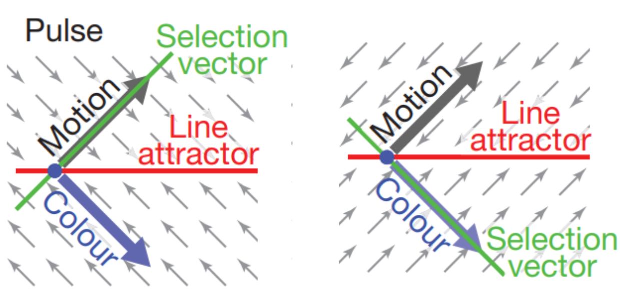


Figure 8: Fixed point analysis

They performed a linear stability analysis around each identified slow points $h*$. We used a first-order Taylor series approximation of the network update equation to create a linear dynamical system,

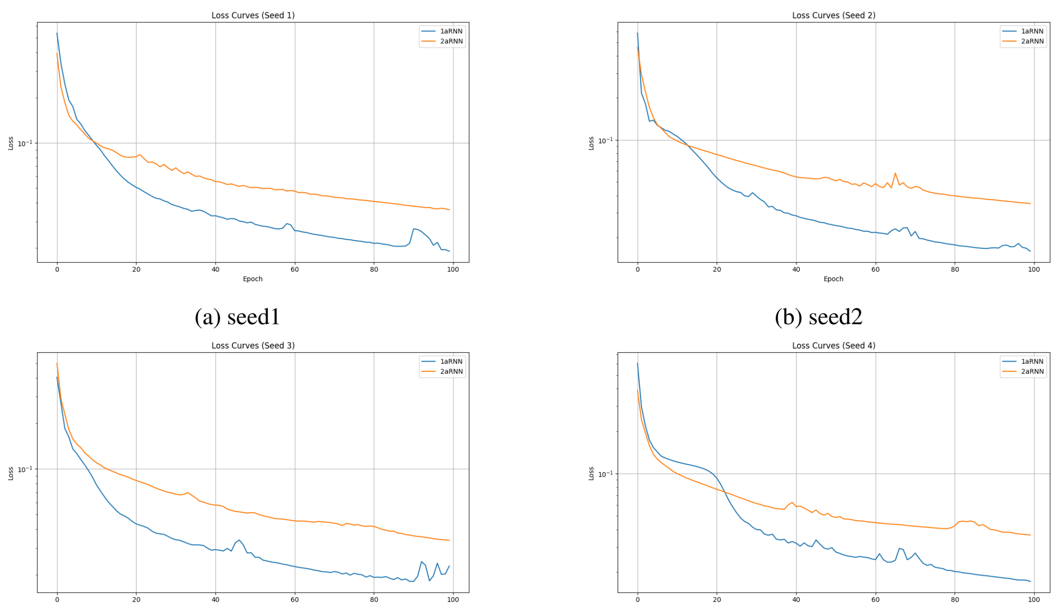
$$\dot{\delta\mathbf{h}} = \mathbf{F}'(\mathbf{h}^*)\delta\mathbf{h},$$

and then perform an eigenvector decomposition on the matrix $\mathbf{F}'(\mathbf{h}*)$ to obtain a set of right and left eigenvectors. In a linear system, one eigenvalue is approximately zero, while all other eigenvalues have large negative real parts.As shown in fig 1 and fig 8, the right and left eigenvectors associated with the zero eigenvalue correspond to the selection vectors (motion axis or color axis) and line attractor, respectively.

6 Testing another time ratio [4]

Our methodological approach closely mirrors the aforementioned procedure. However, in this iteration, we introduce a key modification to the 2aRNN model: we establish the relationship $\tau_2 = 10\tau_1$. This adjustment aims to explore potential novel insights or divergent outcomes. By setting $\tau_2 = 10\tau_1$ instead of $\tau_2 = \tau_1$, we expect to observe significant changes in the model's dynamics. This modification likely introduces a temporal scale separation between Area 1 and Area 2, potentially enhancing the network's ability to process information at different timescales simultaneously. Area 1 may become more responsive to rapid input changes, while Area 2 could integrate information over longer periods, possibly improving the model's capacity to handle complex temporal dependencies. The findings from this modified experimental setup are presented below.

6.1 Training Result



(a) seed1

(b) seed2

(c) seed3

(d) seed4

Figure 9: Training Loss of 1aRNN and 2aRNN with $\tau_2 = 10\tau_1$

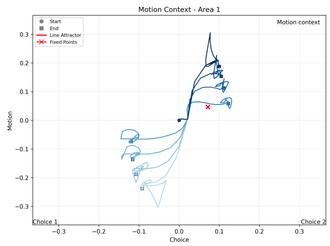
1. The loss curve of the 2aRNN model decreases more slowly when $\tau_2 = 10\tau_1$.
2. The loss profile of the 2aRNN model at $\tau_2 = 10\tau_1$ is smoother than that of the 2aRNN model at $\tau_2 = \tau_1$, which shows significant oscillations.

6.2 2aRNN: reproduction of Fig 1 with $\tau_2 = 10\tau_1$

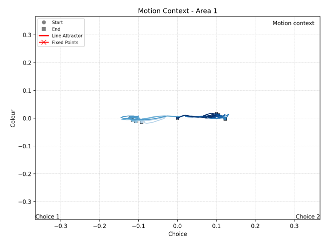
When comparing Fig 10 and Fig 11 ($\tau_2 = 10\tau_1$) to Fig 6 (equal time constants), we observe several notable differences in the model dynamics of area 1:

- **Trajectory Complexity:** Fig 10 and Fig 11 exhibit more complex and intricate trajectories across all contexts. This suggests that the longer time constant in area 2 leads to more elaborate information processing in area 1.

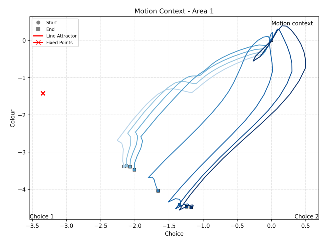
[4]This corresponds to Problem 6 (Bonus exercise).



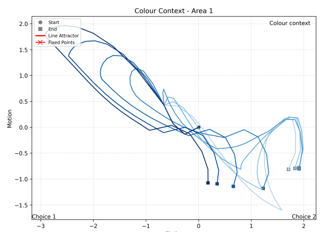
(a) motion context, motion choice



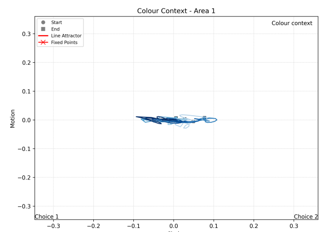
(b) motion context, colour choice



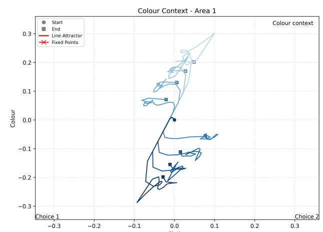
(c) Sort by irrelevant colour



(d) Sort by irrelevant motion

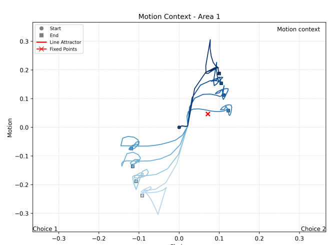


(e) colour choice, motion choice

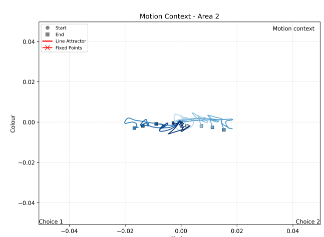


(f) colour choice, colour choice

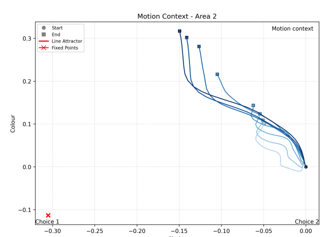
Figure 10: Model dynamics and fixed points analysis area1 with $\tau_2 = 10\tau_1$



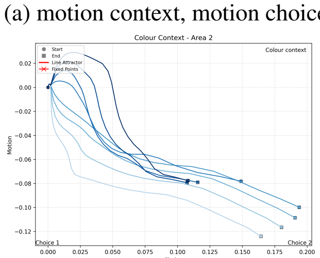
(a) motion context, motion choice



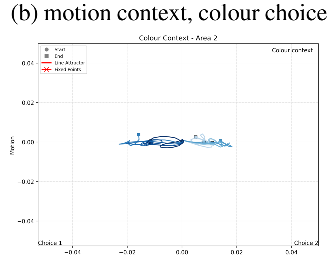
(b) motion context, colour choice



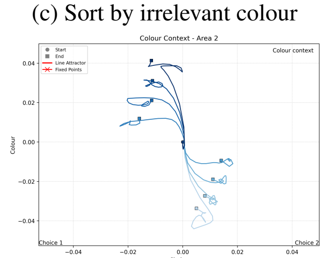
(c) Sort by irrelevant colour



(d) Sort by irrelevant motion



(e) colour choice, motion choice



(f) colour choice, colour choice

Figure 11: Model dynamics and fixed points analysis area2 with $\tau_2 = 10\tau_1$

- **Trajectory Spread:** The trajectories in Fig 10 and Fig 11 generally occupy a larger portion of the phase space, suggesting that the network explores a wider range of states before settling.

These observations suggest that increasing the time constant of area 2 relative to area 1 leads to more complex dynamics and potentially richer representational capacity in the two-area RNN model for context-dependent decision-making.

7 Comparative Analysis of 1aRNN and 2aRNN with Reduced Hidden Size [5]

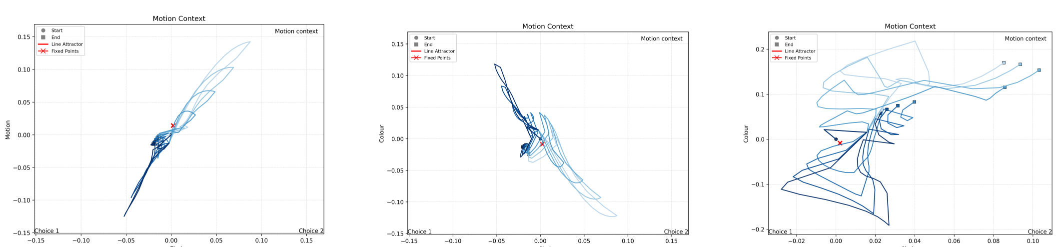
To further differentiate the performance of the 1aRNN and 2aRNN models, we conducted an additional experiment by reducing the hidden size of both networks to 20 units. This modification yielded two significant findings:

7.1 Accuracy Disparity

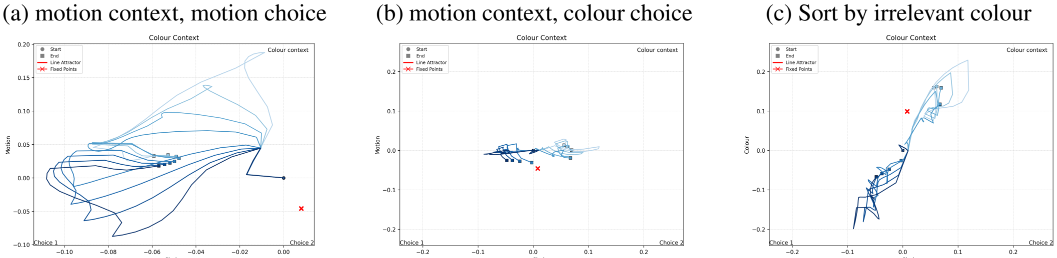
The reduction in hidden size led to a notable divergence in accuracy between the two models. Using 5 seeds, the 1aRNN model achieved an accuracy of 85.80% $\pm$ 18.43%, while the 2aRNN model maintained a high accuracy of 96.00% $\pm$ 3.74%. This substantial difference in performance underscores the 2aRNN model's superior capability in handling context-dependent decision-making tasks, even with limited computational resources. The 2aRNN's ability to maintain near-perfect accuracy despite the reduced hidden size demonstrates its enhanced efficiency in processing and integrating multimodal sensory information and context cues.

7.2 Enhanced Phase Portrait Differentiation

The second key observation was a more pronounced differentiation in the phase portraits, particularly in the regions corresponding to task-irrelevant variables. This increased contrast in the phase space representations provides clearer visual evidence of the fundamental differences in how 1aRNN and 2aRNN process and represent information. The 2aRNN's distinct handling of task-relevant and task-irrelevant information becomes more apparent, highlighting its sophisticated approach to context-dependent information processing.



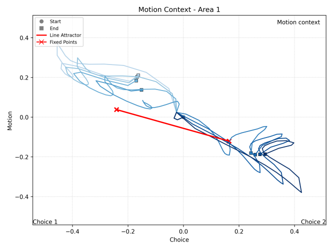
(a) motion context, motion choice (b) motion context, colour choice (c) Sort by irrelevant colour



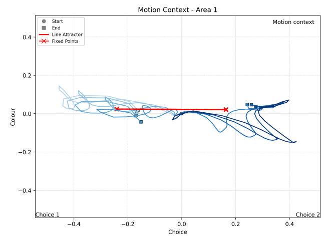
(d) Sort by irrelevant motion (e) colour choice, motion choice (f) colour choice, colour choice

Figure 12: Model dynamics and fixed points analysis for 1aRNN with reduced hidden size

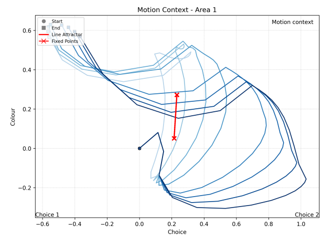
[5] This section was added based on the instructor's feedback and suggestions for improvement.



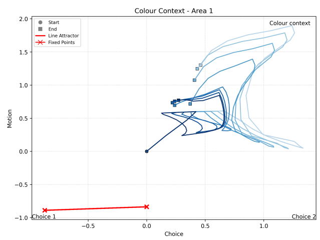
(a) motion context, motion choice



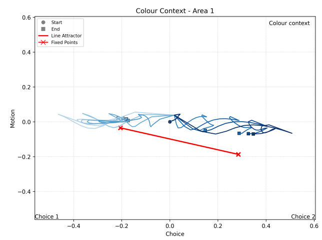
(b) motion context, colour choice



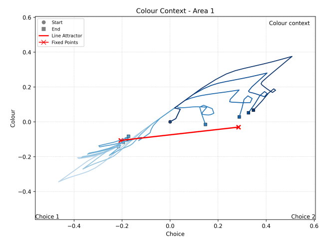
(c) Sort by irrelevant colour



(d) Sort by irrelevant motion

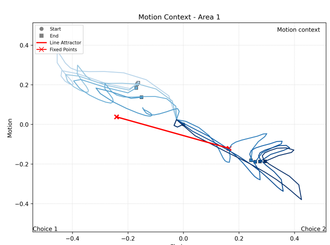


(e) colour choice, motion choice

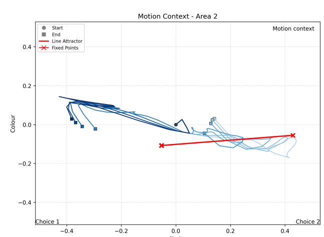


(f) colour choice, colour choice

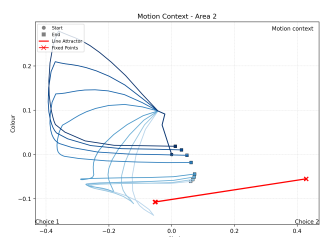
Figure 13: Model dynamics and fixed points analysis for area1 of 2aRNN with reduced hidden size



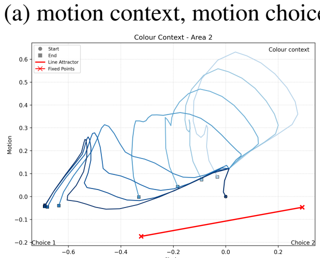
(a) motion context, motion choice



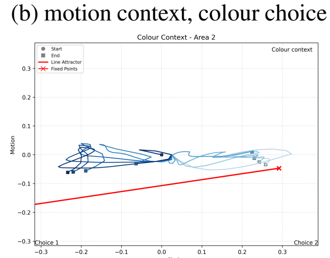
(b) motion context, colour choice



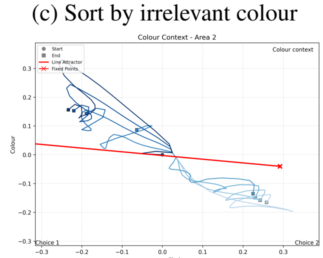
(c) Sort by irrelevant colour



(d) Sort by irrelevant motion



(e) colour choice, motion choice



(f) colour choice, colour choice

Figure 14: Model dynamics and fixed points analysis for area2 of 2aRNN with reduced hidden size

These findings not only reinforce the superiority of the 2aRNN model in context-dependent decision-making tasks but also demonstrate its robustness and efficiency when operating with limited computational resources. The enhanced differentiation in phase portraits provides valuable insights into the underlying mechanisms of information processing in these neural network architectures.

8 Conclusion

In this study, we successfully developed and implemented a Two-Area Recurrent Neural Network (2aRNN) model to simulate context-dependent decision-making processes. Our model effectively compartmentalized the processing of color and motion stimuli, as well as context information, into separate neural network regions. This approach allowed us to replicate the dynamic responses and decision behaviors observed in the prefrontal cortex of rhesus monkeys during visual discrimination tasks.

The 2aRNN model demonstrated its capability to integrate multimodal sensory information under varying task demands, providing insights into how the brain flexibly selects and processes inputs based on contextual cues. By adjusting the time constants of the two network areas, we explored the impact of temporal scale separation, revealing the model's ability to handle complex temporal dependencies in cognitive processes.

While our results are promising and align with previous findings, there are several areas for improvement and future research:

1. **Visualization Enhancement:** The current graphical representations of our model's dynamics could be refined to improve clarity and interpretability. Future work should focus on developing more sophisticated visualization techniques to better illustrate the complex interactions within the 2aRNN.
2. **Advanced Architectures:** Although our RNN-based approach yielded valuable results, exploring more modern architectures such as Transformers (Vaswani, 2017) or other attention-based models could potentially capture even more nuanced aspects of context-dependent decision-making.
3. **Biological Plausibility:** Further research could focus on enhancing the biological plausibility of our model, incorporating more detailed neurophysiological constraints and mechanisms observed in actual prefrontal cortex circuits.
4. **Extended Task Domains:** While our model successfully simulated visual discrimination tasks, future work could extend the 2aRNN to other cognitive domains, testing its generalizability and potential for modeling a broader range of context-dependent behaviors.

In conclusion, our 2aRNN model represents a significant step forward in computational neuroscience's ability to simulate and understand complex cognitive processes. By successfully modeling context-dependent decision-making, we have provided a valuable tool for future research in cognitive neuroscience and artificial intelligence. As we continue to refine and expand this model, we anticipate it will contribute to a deeper understanding of how the brain integrates information and adapts to changing environmental demands.

Author Contributions

Jizhou Guo wrote the majority of the code, explored various methods, and authored sections 2.1, 2.2, 6, 7, and 8. Liting Pang attempted to implement the phase portrait section of the paper (although unsuccessful) and wrote all remaining parts of the paper (except section 4.1 and above) as well as a portion of the course slides. Zhaoyu Zhu also attempted the phase portrait section (although unsuccessful) and later proposed the crucial idea of using features from all time steps for fitting instead of using features from a single time step or averaging certain states. This idea significantly improved the results and was implemented by Jizhou Guo. Zhaoyu Zhu also contributed to writing part of the course slides. Ziyi Xu contributed to all figure parts of course slides and authored section 4.1.

Acknowledgments

We extend our sincere gratitude to Mr. Kai Chen for his patient guidance and assistance, which greatly enhanced our learning experience. We are also deeply appreciative of the instruction and teachings provided by Prof. Douglas Zhou and Mr. Mingzhang Wang during the course, which significantly contributed to our understanding and knowledge in this field.

References

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kevin Lloyd and David S Leslie. Context-dependent decision-making: a simple bayesian model. *Journal of The Royal Society Interface*, 10(82):20130069, 2013.

Valerio Mante, David Sussillo, Krishna V Shenoy, and William T Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *nature*, 503(7474):78–84, 2013.

Larry R Medsker, Lakhmi Jain, et al. Recurrent neural networks. *Design and Applications*, 5(64-67): 2, 2001.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Michael L Waskom, Michael C Frank, and Anthony D Wagner. Adaptive engagement of cognitive control in context-dependent decision making. *Cerebral cortex*, 27(2):1270–1284, 2017.

Xiaohan Zhang, Lu Liu, Guodong Long, Jing Jiang, and Shenquan Liu. Episodic memory governs choices: An rnn-based reinforcement learning model for decision-making task. *Neural Networks*, 134:1–10, 2021.