Recurrent networks recognize patterns with low-dimensional oscillations

Keith T. Murray
Department of Brain and Cognitive Sciences
Massachusetts Institute of Technology
Cambridge, USA
ktmurray@mit.edu

***Abstract*—This study proposes a novel dynamical mechanism for pattern recognition discovered by interpreting a recurrent neural network (RNN) trained on a simple task inspired by the SET card game. We interpreted the trained RNN as recognizing patterns via phase shifts in a low-dimensional limit cycle in a manner analogous to transitions in a finite state automaton (FSA). We further validated this interpretation by handcrafting a simple oscillatory model that reproduces the dynamics of the trained RNN. Our findings not only suggest of a potential dynamical mechanism capable of pattern recognition, but also suggest of a potential neural implementation of FSA. Above all, this work contributes to the growing discourse on deep learning model interpretability.**

***Index Terms*—Pattern recognition, Recurrent neural networks, Neural dynamics, Cognitive modeling, Deep learning interpretability**

I. Introduction

Pattern recognition is fundamental to human cognition, influencing perception, language, and reasoning [1]. Despite substantial advances in machine learning algorithms and models [2], [3], understanding the computational and neurological foundations of pattern recognition in human cognition remains a challenge. Recent work applying recurrent neural networks (RNNs) to neural and cognitive modeling have led to the development of a variety of theories concerning the potential dynamical mechanisms responsible for various cognitive abilities [4]–[8]. This study builds on this research direction and these findings by proposing a simple pattern recognition task, inspired by the SET card game, that was solved by an RNN through phase shifts in a low-dimensional limit cycle. We interpreted this learned dynamical mechanism as recognizing patterns in a manner analogous to a finite state automaton (FSA), suggesting a potential neural implementation of FSA. To support this analogy, we handcrafted a simple oscillatory model based on our interpretation that emulates the trained RNN's low-dimensional dynamics. Our findings not only add to the current research direction of theorizing about potential dynamical mechanisms underlying cognition, but also provide insights into deep learning model interpretability.

II. Background

A. Dynamical motifs in RNNs

Previous work by [6] showed that RNNs trained on many cognitive tasks learned a series of modular, low-dimensional, dynamical phenomena (e.g. attractors, limit cycles, bifurcations [11]), termed dynamical motifs, that were shared among all cognitive tasks. In addition, simple tasks would learn simple dynamical motifs and complex tasks would combine multiple of these motifs in a manner apt for performing in the complex task. This work suggests that all cognitive abilities may have associated dynamical phenomena that are easy to interpret and implemented at the neural level. It is possible that the brain only implements a handful of simple dynamical motifs that can be combined and reused to perform all cognitive tasks.

B. Dynamical mechanisms for transitive inference

Previous work by [7] showed that RNNs trained to perform a transitive inference task learned a simple dynamical mechanism characterized by a single oscillation and a collinear encoding of stimulus inputs. To perform transitive inference, the RNN would encode stimuli as a linear input into the dynamical mechanism and the oscillatory activity of the mechanism would subtract the magnitude of subsequent encoded stimuli. Ref. [7] hypothesized that cognitive abilities requiring transitive inference might incorporate this particular dynamical mechanism. Building on these insights and motivations, we sought to uncover a potential dynamical mechanism underlying pattern recognition through training RNNs and interpreting their learned dynamics.

C. Why do interpretable dynamical mechanisms emerge?

The simplicity of the dynamical mechanisms learned in [6] and [7] is initially surprising; however, both studies incorporated substantial regularizations during the training of RNNs that biased the learned dynamics to be low-dimensional and interpretable. These regularizations were an L2 regularization imposed on the trainable weights and on the recurrent activations of the RNN. These constraints are well known to produce interpretable dynamics in RNNs and have a potential biophysical interpretations [9], [10]. Regularizations on the trainable weights reflect a synaptic resource constraint, and regularizations on the recurrent activations reflect a cellular metabolic constraint. In this work, we used these regularizations in order to bias our model to learn an interpretable dynamical mechanism.

III. Methodology

Our methodology is divided into three parts: defining our novel pattern recognition task (sec. III-A), defining the model and its parameters (sec. III-B), and describing our analysis methods (sec. III-C).

A. Task

While there exists a host of pattern recognition and classification tasks in the machine learning community [2], [3], their associated datasets may be too complex and/or noisy to elicit interpretable dynamical mechanisms. We instead took inspiration for the design of our own task from the card game SET [12]. SET is a card game where players race to identify sets of three cards out of a field of 12 cards. Each card has four attributes: shape, color, number, and shading. Each of these attributes has three possible values. A valid set is found when for each attribute, all the values pertaining to that attribute across all three cards are the *same* or *different*. The game is regarded as being cognitively demanding despite the simplicity of the associated patterns [12].

While the complete game of SET involves searching through 12 cards, each with four attributes, our task was abbreviated to only concern classifying three cards, each with only one attribute, as either a valid set or an invalid set. We refer to this attribute as color and the three possible values as *green*, *purple*, and *red*. The task involved sequentially presenting three colors to an agent and tasking the agent with classifying if the colors presented constituted a valid set (i.e. the agent decides if all the colors presented were the *same* or *different*).

All trials of the task were $500ms$. There was a delay period of $30ms$ at the beginning and $100ms$ at the end of the trial where no colors were presented. A color was presented for $20ms$, and there was at least a $30ms$ gap between the presentation of colors (Fig. 1). Given that our agent was an RNN, each color was encoded as a 100-dimensional vector[1] with values drawn from a Gaussian distribution, $\mathcal{N}(0, 1)$. The same encoding vectors were used across all training and testing trials. The RNN was tasked with producing an output of $+1$ to indicate a valid set or -1 to indicate an invalid set at the end of a trial.

The model was trained on a training dataset of 540 trials. The proportion of accepting to rejecting trials in the training dataset was 50%. 27 trails were generated for the testing dataset, where each trial consisted of a distinct combination of presented colors. Trials were mini-batched during training into batches of 108 trials.

B. Model

We describe our model according to the framework proposed in [13].

[1]Our use of a 100-dimensional vector was inspired by [7]. This dimensionality ensured that stimulus representations were uncorrelated in the activity space of the model.

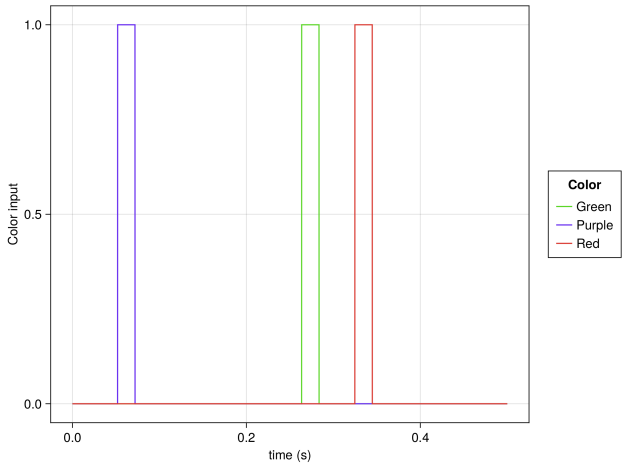

Fig. 1. A sample trial showing the colors *purple*, *green*, and *red* being presented. This trial constitutes a valid set.

Architecture: The model was a continuous-time RNN identical to the networks used in [4], [5], [7]–[10]:

$$\tau \dot{x}_i(t) = -x_i(t) + \sum_{k=1}^{N} J_{ik} r_k(t) + \sum_{k=1}^{N^{in}} B_{ik} u_k(t) + b_i + \eta_i(t) \quad (1)$$

$$r_i(t) = \tanh(x_i(t)) \quad (2)$$

$$z(t) = \sum_{k=1}^{N} W_k r_k(t) + b_{out} \quad (3)$$

τ is interpreted as the time constant for the RNN and was set to $10ms$. $x_i(t)$ is interpreted as the average voltage of the ith subpopulation of neurons at time t. N is the total number of subpopulations of neurons and was set at 100. $u_k(t)$ is interpreted as the kth input stimulus conveying encoded color. N^{in} is the number of input dimensions and was, as stated in sec. III-A, set at 100. $\eta_i(t)$ is a random value drawn from a Gaussian distribution, $\mathcal{N}(0, 0.10)$. $r_i(t)$ is interpreted as the average firing rate for the ith subpopulation of neurons. The activation function that converted average voltage to average firing rate is the hyperbolic tangent (tanh) function. $z(t)$ is interpreted as the average firing rate of the population of output neurons at time t.

Equation (1) was approximated with Euler's method using a step size (Δt) of $10ms$ for all trials. The entirety of our model was coded in Julia [14] using the Lux framework [15] and SciML ecosystem [16], [17].

Learning algorithm: We used the AdamW learning algorithm [18] and reverse mode automatic differentiation to update trainable parameters. The trainable parameters were the initial state $\mathbf{x}(t = 0)$, recurrent matrix $\mathbf{J}$, input matrix $\mathbf{B}$, recurrent bias $\mathbf{b}$, output matrix $\mathbf{W}$, and output bias b_{out}. The initial learning rate was set to 10^{-4}. As referenced in sec. II-C, an L2 regularization was imposed on the trainable parameters. We implemented this regularization through the AdamW learning algorithm with the weight decay parameter set to 10^{-4}.

Objective function: We used a mean squared error (MSE) objective function to measure the difference between the observed $z(t)$ and the expected $\hat{z}(t)$. The last $50ms$, corresponding to the last 5 time steps, of $z(t)$ of each trial were measured against $\hat{z}(t)$. As referenced in sec. II-C, an L2 regularization was imposed on the recurrent activations of the RNN. We implemented this regularization as an added term in the objective function. The full objective function was the following:

$$\mathcal{L}(\hat{z}, z, \mathbf{r}) = \frac{1}{5} \sum_{t=T-5}^{T} (\hat{z}(t) - z(t))^2 + \frac{\lambda}{TN} \sum_{t,i=1}^{T,N} r_i^2(t) \Delta t \quad (4)$$

T is the total number of time steps across all trials and, via Euler's method, was set at 50. λ is the L2 regularization parameter and was set to 10^{-4}.

C. Analysis

In order to interpret the learned dynamics of our model, we used principal component analysis (PCA). As a prevalent dimensionality reduction technique, PCA has been used in previous work to visualize and interpret the low-dimensional dynamics of RNNs [4]–[8]. In our study, we applied PCA to the firing rates of the RNN, $\mathbf{r}(t)$, under conditions of no recurrent noise, $\eta_i(t) = 0$. This approach enabled us to identify potential dynamical mechanisms.

However, it is crucial to recognize that PCA primarily uncovers the strongest correlations among the firing rates of the RNN rather than elucidating causal, mechanistic properties of the entire model [19]. To address this inherent limitation of PCA, we supplemented our analysis by developing a simplified model. This model, guided by insights from PCA, emulated the dynamics of the RNN and served to verify the mechanisms suggested by our PCA-based interpretation. This combination of methods strengthens the validity of our findings.

IV. Hypothesis

Our initial hypothesis posited that PCA would reveal that our trained model learned a low-dimensional network comprised of attractive fixed points. We hypothesized that the model's dynamics would originate from a central attractor within this network and the presentation of encoded colors would then transition these dynamics towards surrounding attractors. At the end of the trial, the final attractor that the dynamics settled into would have an associated classification label of either a valid or an invalid set (Fig. 2).

This hypothesis was predicated on two fundamental assumptions:

1) The model's dynamics would exhibit attraction.
2) Presented colors would be linearly encoded into the model's dynamics.

The first assumption drew inspiration from the Hopfield network [20], where memories are stored as attractive fixed points. Unlike the Hopfield network, which only receives stimuli at the onset of its dynamics, our hypothesis takes into account a sequential presentation of colors as required by our task. Despite this difference, the Hopfield network's central insight—that memories in a neural circuit can be encoded with attractive dynamics—has found validation in experimental work [21], [22]. Therefore, we integrated this concept of attractive dynamics into our hypothesized dynamical mechanism.

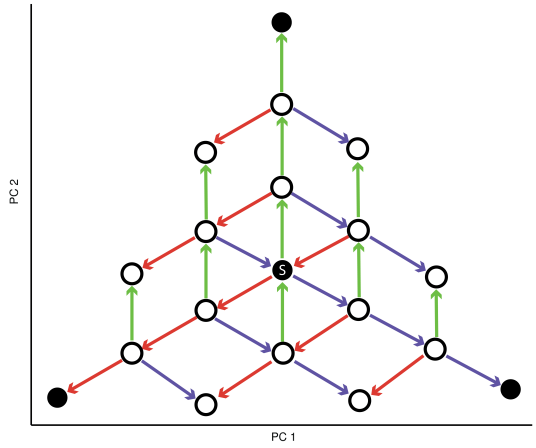

Fig. 2. Our hypothesized dynamical mechanism. Each circle is an attractive fixed point. Filled circles denote attractors classified as a valid set. Empty circles denote attractors classified as an invalid set. The model's dynamics begin in the central attractor. Our first and second assumption resulted in a hexagonal tiling of the networks fixed points.

The second assumption was influenced by a variety of computational studies that demonstrated the crucial role linear encoding of stimuli can play in shaping an RNN's learned dynamical mechanism [4]–[7]. Furthermore, given that the input matrix $\mathbf{B}$ in equation (1) is not a function of time, voltage, or firing rate, we assumed that the presentation of a color at any point in the dynamics should result in a linear perturbation of the same magnitude and direction as a presentation of the same color at any other point in the dynamics. Thus, our hypothesis adopted this linear encoding of colors as a fundamental assumption.

We further noticed that this hypothesis of transitions between attractors in a network of fixed points closely resembles the transitions between states in an FSA. Previous work has drawn the analogy between RNNs and FSA before [23], and in this study, we sought to further build on this analogy in the context of low-dimensional dynamical mechanisms.

V. Results

A. Training and testing accuracy

The fully-trained RNN model demonstrated robust performance, achieving an accuracy of 96.30% on training data and 100.00% on testing data. In the absence of recurrent noise, the model achieved perfect accuracy, reaching an accuracy of 100.00% on both training and testing data. This perfect accuracy was expected due to the algorithmic nature of the task [24], [25].

Throughout a trial, the model's output, $z(t)$, exhibited a large oscillation with *perturbations* corresponding to the pre-

sentation of encoded colors (Fig. 3). To recognize valid sets, the output's oscillation ended the trial in the positive phase, yielding a value of $+1$. Conversely, to recognize invalid sets, the output's oscillation ended the trial in the negative phase, yielding a value of -1. In the absence of encoded color input, the model completes two full oscillations. The *perturbations* corresponding to color presentation indicated some influence on the oscillation, yet the computational significance remained unclear without further analysis.

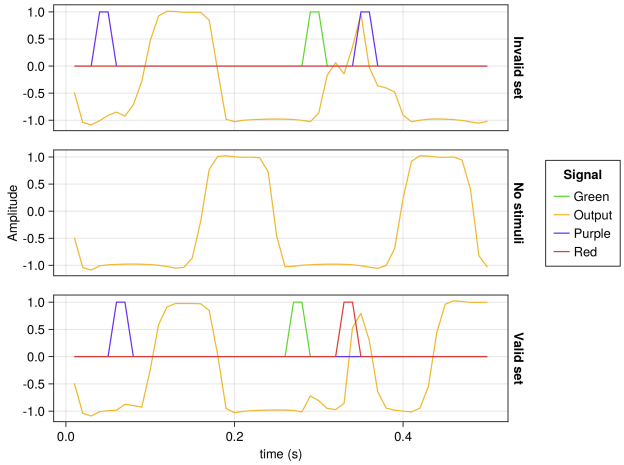

Fig. 3. Three examples of trials and the resulting model output. The top row displays a trial with an invalid set and the resulting model output of -1. The middle row displays a trial with no presented colors and demonstrates the model output completing two full oscillations. The bottom row displays a trial with a valid set and the resulting model output of -1. Note the *perturbations* caused by the presentation of colors in the top and bottom rows.

B. PCA insights

PCA revealed that the model learned a low-dimensional limit cycle[2] which completed two full rotations during a trial (Fig. 4). At the end of the trial, PCA revealed that the dynamics of the model would lie in one of three distinct distributions along the limit cycle. Two distributions corresponded to invalid sets and one distribution corresponded to valid sets. These findings indicated that the presentation of colors would shift the phase of the limit cycle. We inferred this phase shift to be the computational affect of the *perturbations* seen in Fig. 3.

In order to confirm that the presentation of colors led to phase shifts in the model's limit cycle, we rotated PCA trajectories and examined the resulting oscillations. This rotation was conducted due to the observation that projecting the model's dynamics entirely onto principal component (PC) 1 would lead to a mixing of distributions corresponding to valid and invalid sets. Through a 60-degree rotation, the distributions corresponding to valid and invalid sets became maximally separable when projected onto rotated principal component (rPC) 1.

Fig. 5 illustrates the model's dynamics projected onto rPC 1 over time for various trials. By examining the influence

[2]Refer to [11] for an overview of limit cycles in nonlinear dynamics.

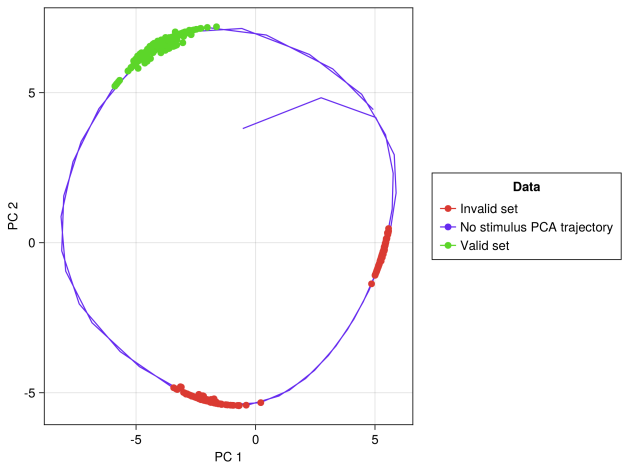

Fig. 4. The model's dynamics projected onto PC 1 and PC 2. The purple trajectory is the projected dynamics of the model during a trial with no stimulus. The green points are the endpoints of the model's dynamics during trials with valid sets. The red points are the endpoints of the model's dynamics during trials with invalid sets.

of presented colors on the dynamics across different trials, we developed hypotheses about the mechanistic effects of presented colors on the underlying limit cycle. *Green* appeared to slightly perturb the model's dynamics, *purple* appeared to rush the model's dynamics, and *red* appeared to reset the model's dynamics.

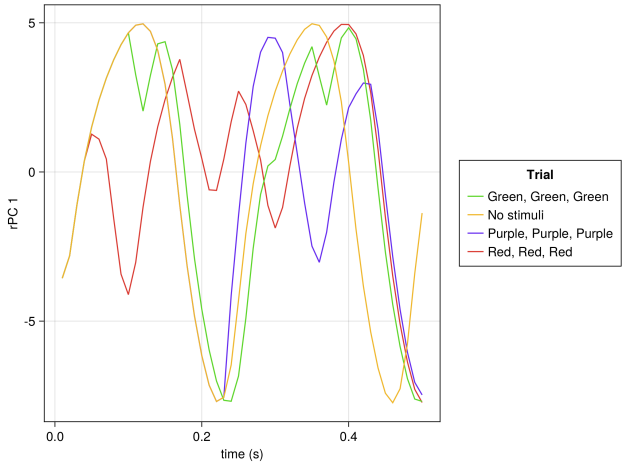

Fig. 5. The model's dynamics projected onto rPC 1 for various trials. In the trial with no stimuli, the dynamics performed two complete oscillations. In the all-*green* trial, the dynamics were minimally perturbed. In the all-*purple* trial, the presentation of the color seemed to shift the phase of the limit cycle forward. In the all-*red* trial, the color presentation appeared to shift the phase backward.

Contrary to our initial hypothesis—which suggested that the model would learn a dynamical mechanism characterized by attractive fixed points and a linear encoding of colors—the oscillatory dynamics of both the output (Fig. 3) and the PCA trajectories (Fig. 4 and Fig. 5) led us to reconsider our fundamental assumptions. Instead, we discovered that the

model learned a dynamical mechanism characterized by phase shifts in a limit cycle. Remarkably, the analogy to an FSA persisted with this new dynamical mechanism. Rather than the FSA residing in the state space of the dynamics, it now exists in the phase angle state space of the limit cycle with phase shifts analogous to state transitions (Fig. 6).

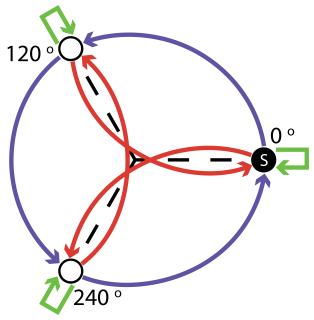

Fig. 6. The learned FSA exists in the phase angle state space of the limit cycle. The automaton starts at a phase angle of 0 degrees and the presentation of colors causes transitions to other states, each representing a distinct phase angle in the limit cycle.

While we previously observed that the presentation of the color *green* did impact the model's dynamics (Fig. 5), we interpreted the learned dynamical mechanism under the assumption that *green* did not impact the dynamics in order to simplify the FSA analogy (Fig. 6). We interpreted *purple* as adding 120^o to the phase angle and *red* as subtracting $120°$. Hence, it becomes clear how presenting all the *same* or *different* colors would cause the limit cycle's phase shifts to sum to $0°$, $360°$, or $-360°$ and cause the model's output to end the trial in the valid set phase.

To test the accuracy of our interpreted dynamical mechanism, we made predictions about the ending phase of the model's dynamics when projected onto rPC 1 in trials with invalid sets. For example, if the sequence of presented colors is *purple*, *green*, *purple*, the model should end $120°$ behind the valid set phase. Conversely, if the sequence of presented colors is *red*, *green*, *red*, the model should end $120°$ ahead of the valid set phase. Fig. 7 supports these predictions and our interpretation of the learned dynamical mechanism.

C. Handcrafting an oscillatory model

To further validate our interpretation of the learned dynamical mechanism and its analogy with FSA, we designed a simple oscillatory model to emulate the dynamics of our trained model. Specifically, we handcrafted the following equation to reproduce the projection of our model's dynamics onto rPC 1:

$$f(t) = 7\sin(\frac{2\pi}{0.29}t + \int_0^t \phi(\tau)\,d\tau) - 1.5 \qquad (5)$$

$\phi(\tau)$ corresponds to the presentation of an encoded color: if *purple* is presented at time τ, then $\phi(\tau) = \frac{2\pi}{3}$; if *red* is presented, then $\phi(\tau) = -\frac{2\pi}{3}$; and if *green* or no color is presented, then $\phi(\tau) = 0$.

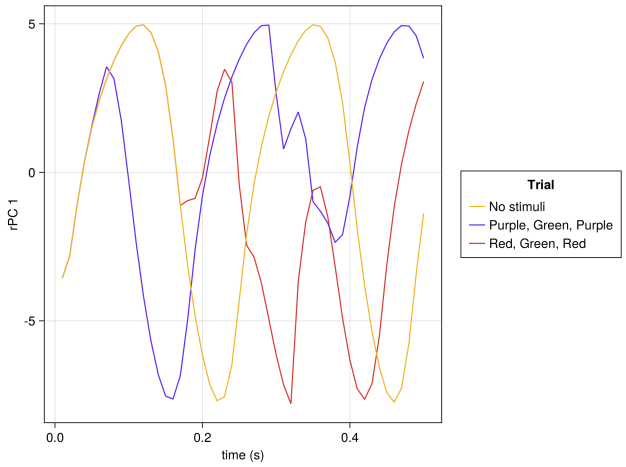

Fig. 7. The ending phase of rPC 1 for trials of invalid sets supports our interpretation of the learned dynamical mechanism. A *purple*, *green*, *purple* trial results in rPC 1 ending $120°$ behind the valid set phase. A *red*, *green*, *red* trial results in rPC 1 ending $120°$ ahead of the valid set phase.

It is worth noting that equation (5) simplifies many of the dynamical properties observed in rPC 1. For instance, the frequency of the dynamics was nonuniform (Fig. 4) and color presentation caused complex perturbations in the model's dynamics (Fig. 5). However, equation (5) models the frequency as constant at $\frac{2\pi}{0.29}$ and models color presentation as causing simple perturbations by integration. Despite these simplifications, equation (5) qualitatively captures the model's dynamics projected onto rPC 1 (Fig. 8). The similarity between the dynamics of our trained model and our handcrafted model further supports our interpretation of the learned dynamical mechanism.

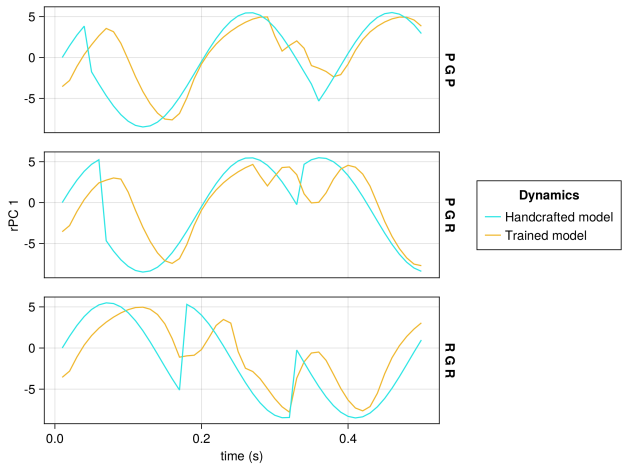

Fig. 8. Our handcrafted oscillatory model qualitatively captures our trained model's dynamics projected onto rPC 1. The top row shows model dynamics in a *purple*, *green*, *purple* trial. The middle row shows model dynamics in a *purple*, *green*, *red* trial. The bottom row shows model dynamics in a *red*, *green*, *red* trial.

VI. Discussion

In this study, we demonstrated how an RNN, trained on a simple pattern recognition task inspired by the SET card game, could recognize patterns via phase shifts in a limit cycle. We employed PCA to identify three distinct distributions of endpoints in our model's dynamics, each corresponding to a distinct phase angle in the limit cycle (Fig. 4), and showed how the presentation of colors caused transitions between these phase angles (Fig. 5). Additionally, we drew an analogy between phase shifts in the limit cycle and state transitions in an FSA (Fig. 6). The validity of our interpretation of the learned dynamical mechanism, as well as the FSA analogy, was strengthened by the prediction of the ending phase of our model's dynamics (Fig. 7) and the construction of a handcrafted oscillatory model that mimicked these dynamics (Fig. 8). In essence, we have developed a robust narrative detailing the learned dynamical mechanism of an RNN trained on a simple pattern recognition task.

Interestingly, the dynamical mechanism learned in our study challenged two fundamental assumptions pertaining to our original hypothesis, specifically that the model would learn attractive dynamics and linearly encode colors. Although there are theoretical [26] and computational [7], [8] precedents for oscillatory dynamics in RNNs, it remains unclear why our model opted for a mechanism characterized by oscillatory dynamics rather than attractive dynamics. Given the universal function approximation capabilities of RNNs [27], it is theoretically possible to for an RNN to learn a dynamical mechanism similar to our initial hypothesis. Perhaps by fine-tuning various model and task-related hyperparameters, we could bias our model towards learning attractive dynamics [28].

Furthermore, the cognitive plausibility of FSA deserves further exploration. While FSA are simple computational models with deep roots in cognitive science [29], the simplicity of our task and stimuli likely contributed to the emergence of an FSA-like dynamical mechanism. Introducing more complex stimuli and tasks might disrupt this FSA analogy, but it's also possible that the underlying algorithm our model might implement is represented by an FSA employing the dynamical mechanism elucidated in this study.

Looking ahead, it would be valuable to investigate how the mechanism uncovered here could integrate with other dynamical mechanisms to enable more complex computations. In particular, we are interested in how our dynamical mechanism might be integrated into the dynamics of an RNN trained on a full game of SET complete with visual search across all 12 cards [12]. Ultimately, this line of research could lead toward constructing a dynamical computer capable of programming in a manner similar to conventional digital computers [30].

Lastly, our findings and methodologies may have meaningful implications for mechanistic interpretability in the field of deep learning. This domain is focused on constructing mechanistic models of the learned computations of deep learning models with the aim to elucidate phenomena related to generalization and adversarial attacks [24], [25]. Our work aligns with this research direction and the approaches used may prove valuable for future interpretability studies.

Acknowledgments

We would like to thank Sabrina Drammis, Joshua Liu, Nancy Lynch, Nikasha Patel, and Brabeeba Wang for helpful discussions pertaining to the conceptualization of this work. We also acknowledge the assistance of OpenAI's ChatGPT in editing portions of this manuscript.

References

[1] B. Inhelder and J. Piaget, *The Early Growth of Logic in the Child: Classification and Seriation*. New York: Harper and Row, 1964.

[2] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2013.

[4] D. Sussillo and O. Barak, "Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks," *Neural Computation*, vol. 25, no. 3, pp. 626–649, 2013.

[5] V. Mante, D. Sussillo, K. V. Shenoy, and W. T. Newsome, "Context-dependent computation by recurrent dynamics in prefrontal cortex," *Nature*, vol. 503, no. 7474, pp. 78–84, 2013.

[6] L. Driscoll, K. Shenoy, and D. Sussillo, "Flexible multitask computation in recurrent networks utilizes shared dynamical motifs," bioRxiv, 2022.

[7] K. Kay, X. Wei, R. Khajeh, M. Beiran, C. J. Cueva, G. Jensen, V. P. Ferrera, and L. F. Abbott, "Neural dynamics and geometry for transitive inference," bioRxiv, 2022.

[8] M. Pals, J. H. Macke, and O. Barak, "Trained recurrent neural networks develop phase-locked limit cycles in a working memory task," bioRxiv, 2023.

[9] D. Sussillo, M. M. Churchland, M. T. Kaufman, and K. V. Shenoy, "A neural network that finds a naturalistic solution for the production of muscle activity," *Nature Neuroscience*, vol. 18, no. 7, pp. 1025–1033, 2015.

[10] C. J. Cueva and X.-X. Wei, "Emergence of grid-like representations by training recurrent neural networks to perform spatial localization," arXiv preprint arXiv:1803.07770, 2018.

[11] S. H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, And Engineering*. Boulder: Westview Press, 2000.

[12] L. McMahon, G. Gordon, H. Gordon, and R. Gordon, *The Joy of SET: The Many Mathematical Dimensions of a Seemingly Simple Card Game*. Princeton: Princeton University Press, 2017.

[13] B. A. Richards, T. P. Lillicrap, P. Beaudoin, Y. Bengio, R. Bogacz, A. Christensen, C. Clopath, R. P. Costa, A. de Berker, S. Ganguli, et al., "A deep learning framework for neuroscience," *Nature Neuroscience*, vol. 22, no. 11, pp. 1761–1770, 2019.

[14] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM Review*, vol. 59, no. 1, pp. 65–98, 2017.

[15] A. Pal, "Lux: Explicit Parameterization of Deep Neural Networks in Julia," GitHub repository, 2022. [Online]. Available: https://github.com/avik-pal/Lux.jl/. [Accessed Mar. 31, 2023].

[16] C. Rackauckas, Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. Skinner, A. Ramadhan, and A. Edelman, "Universal differential equations for scientific machine learning," arXiv preprint arXiv:2001.04385, 2020.

[17] C. Rackauckas and Q. Nie, "Differentialequations. jl–a performant and feature-rich ecosystem for solving differential equations in julia," *Journal of Open Research Software*, vol. 5, no. 1, 2017.

[18] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," arXiv preprint arXiv:1711.05101, 2017.

[19] C. Langdon and T. A. Engel, "Latent circuit inference from heterogeneous neural responses during cognitive tasks," bioRxiv, 2022.

[20] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[21] W. Chaisangmongkon, S. K. Swaminathan, D. J. Freedman, and X.-J. Wang, "Computing by robust transience: How the fronto-parietal network performs sequential, category-based decisions," *Neuron*, vol. 93, no. 6, pp. 1504–1517, 2017.

[22] M. Khona and I. R. Fiete, "Attractor and integrator networks in the brain," *Nature Neuroscience*, vol. 23, no. 12, pp. 744–766, 2022.

[23] A. Cleeremans, D. Servan-Schreiber, and J. L. McClelland, "Finite state automata and simple recurrent networks," *Neural Computation*, vol. 1, no. 3, pp. 372–381, 1989.

[24] A. Power, Y. Burda, H. Edwards, I. Babuschkin, and V. Misra, "Grokking: Generalization beyond overfitting on small algorithmic datasets," arXiv preprint arXiv:2201.02177, 2022.

[25] N. Nanda, L. Chan, T. Liberum, J. Smith, and J. Steinhardt, "Progress measures for grokking via mechanistic interpretability," arXiv preprint arXiv:2301.05217, 2023.

[26] F. C. Hoppensteadt and E. M. Izhikevich, "Oscillatory neurocomputers with dynamic connectivity," *Physical Review Letters*, vol. 82, no. 14, pp. 2983–2986, 1999.

[27] K. Funahashi and Y. Nakamura, "Approximation of dynamical systems by continuous time recurrent neural networks," *Neural Networks*, vol. 6, no. 6, pp. 801–806, 1993.

[28] R. Schaeffer, M. Khona, and I. R. Fiete, "No free lunch from deep learning in neuroscience: A case study through models of the entorhinal-hippocampal circuit", In Proc. Advances in Neural Information Processing Systems, vol. 35, pp. 16052–16067, 2022.

[29] N. Chomsky, *Syntactic Structures*. Berlin, Boston: De Gruyter Mouton, 1957.

[30] H. Jaeger, "Towards a generalized theory comprising digital, neuromorphic and unconventional computing," *Neuromorphic Computing and Engineering*, vol. 1, no. 1, pp. 012002, 2021.