Heterogeneous Effect of Input and Task-optimization on the Dynamics of RNNs

Mohammad Taha Fakharian[1] (fakharian.taha@ut.ac.ir)
School of Electrical and Computer Engineering, University of Tehran,
Tehran, Iran

Alireza Ghalambor[1] (ghalamboral@gmail.com)
Ahvaz Jondishapur University of Medical Sciences,
Ahvaz, Iran

Arman Behrad (arman.behrad@tu-dresden.de)
Computational Neuroscience, Dept. of Child and Adolescent Psychiatry,
Faculty of Medicine, TU Dresden, Germany

Roxana Zeraati[2] (research@roxanazeraati.org)
Dept. of Computational Neuroscience, Max Planck Institute for Biological Cybernetics,
Tübingen, Germany

Shervin Safavi[2] (research@shervinsafavi.org)
Computational Neuroscience, Dept. of Child and Adolescent Psychiatry,
Faculty of Medicine, TU Dresden, Germany
Dept. of Computational Neuroscience, Max Planck Institute for Biological Cybernetics,
Tübingen, Germany

[1] Equal first authors.
[2] Equal senior authors.

Abstract

Reverse-engineering task-optimized recurrent neural networks (RNNs) has become a key framework to uncover mechanisms of brain computation in cognitive tasks. Tasks are often constructed as a set of inputs. Then, RNNs are optimized to achieve a set of computational sub-goals given the inputs. Then, neural dynamics in RNNs can be shaped by two major factors: the effect of input structure (defined by task) and task-based optimization or training. The former better reflects the attributes of the input to the network, while the latter better reflects the connectivity that is shaped by task-based optimization. Although both are major factors shaping the network dynamics in a task-specific fashion, how exactly these factors affect network dynamics remains elusive. Here, we investigate the effect of both factors on discriminating the neural dynamics across tasks. We systematically vary the network architecture and the input conditions, using three distinct recurrent architectures: Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and vanilla RNN (V-RNN), trained on cognitive tasks from the NeuroGym library (Molano-Mazon et al., 2022). While we observed a vast range of heterogeneity across architectures and choices of task on task-specific dynamics, we observed that task structure (rather than task-based optimization of the connectivity) almost dominantly informs about task-specific dynamics.

Keywords: neural dynamics; dynamical similarity analysis; recurrent neural network; dynamics through computation

Dynamics of task-optimized RNNs

Recent advances in task-optimized recurrent neural networks (RNNs) (Driscoll et al., 2024; Yang et al., 2019) have opened a new avenue to directly link neural dynamics to behavior and uncover the underlying mechanisms by reverse engineering the trained RNNs. RNNs are optimized to achieve a set of computational sub-goals given the inputs. Hence, the dynamics of the optimized RNN can be jointly determined by the task's input structure and its computational sub-goals. However, the interplay and exact contribution of each mechanism in shaping RNN dynamics is not fully understood.

Training RNNs on cognitive tasks

To investigate the interplay between the effect of architectural constraints and input structure on neural dynamics, we implemented three distinct recurrent architectures: LSTM, GRU, and Vanilla RNN. All networks utilized ReLU activation functions to ensure consistency in nonlinear properties across architectures. Each architecture was initialized using 5 different seeds and independently trained on four cognitive tasks from the NeuroGym library (Molano-Mazon et al., 2022) using identical learning protocols. The selected cognitive tasks (Figure 1) consist of context-dependent decision-making (CDM), go/no-go (GNG), delayed comparison (DC), and probabilistic reasoning (PR). To allow comparison across tasks, they all shared identical temporal structures across their four processing phases (fixation, stimulus presentation, delay, and decision) with similar timing alignment across all comparable phases for both training and testing.

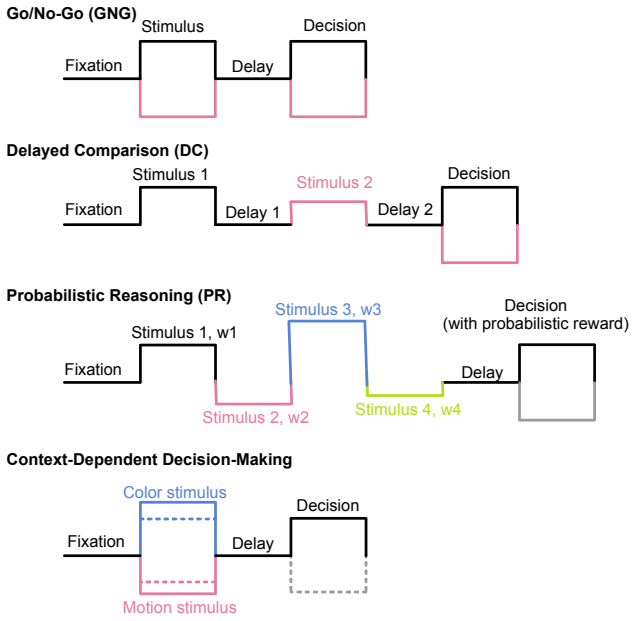

Figure 1: Structure of four cognitive tasks, showing shared processing phases with task- and input-specific requirements.

Assessing factors shaping the dynamics

To compare neural dynamics across networks, we chose Dynamical Similarity Analysis (DSA, Ostrow et al., 2023), rather than alternatives like Centered Kernel Alignment (CKA, Kornblith et al., 2019) or Procrustes Transformation. Although CKA and Procrustes are useful for comparing static representations, they treat temporal dynamics as fixed trajectories, either by finding optimal geometric alignments (Procrustes) or comparing kernel matrices of feature spaces (CKA). DSA was specifically designed for analyzing neural dynamics, characterizing neural activity as evolving trajectories in high-dimensional spaces that more accurately reflect the temporal nature of recurrent network computations (Guilhot et al., 2024).

We assessed the similarity between the dynamics of different networks by applying DSA to RNN unit activity during each task period (e.g., fixation, delay, etc) separately. After collecting the activity patterns, we computed dissimilarity matrices between all network pairs and reduced these matrices to two dimensions using multidimensional scaling (MDS, Kruskal, 1964). The separability of different conditions within this 2D space was then quantified using the clustering accuracy of logistic regression.

To systematically investigate the interplay between the effect of task-optimized connectivity and input structure Figure 2, we developed a comprehensive analytical pipeline cen-

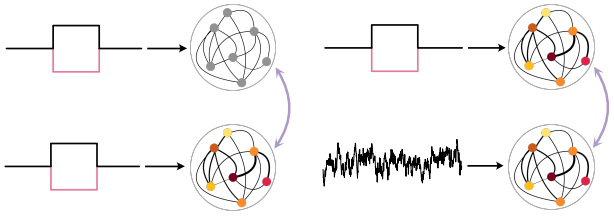

Figure 2: Schematic of DSA comparisons between trained (colored) and untrained (gray) RNNs receiving the same task input (left) and trained RNNs receiving task or random inputs (right). As the random input we used i.i.d noise sampled from a Gaussian distribution with same dimensionality as the original task inputs.

tered on the distinct epochs common among all tasks. In other words, we used the shared structure of cognitive tasks – fixation, stimulus, delay, and decision phases – to directly compare equivalent processing stages across different tasks and architectures.

We first examined whether dynamics during corresponding phases (e.g., delay period that represents a working memory computational sub-goal) across different tasks were better distinguished by computational sub-goals/motifs or by task-specific inputs (Figure 2, left). Using DSA, we projected the dynamics of both trained RNNs and their untrained counterparts into a common dissimilarity space (we used untrained networks as they are not optimized for any of the computational sub-goals/motifs). We then quantified separation accuracy using logistic regression based on two classification tasks: task separation (distinguishing between different tasks regardless of training status) and training separation (distinguishing between trained and untrained networks regardless of task identity). This first analysis focused exclusively on networks receiving structured task inputs to isolate the effects of optimization from input effects.

In our second analysis, we investigated how input structure influences neural dynamics by comparing the same trained networks under two input conditions: networks receiving structured task-relevant inputs (original input) versus random unstructured inputs (Figure 2, right). Using the same DSA methodology, we quantified task separation (distinguishing between different tasks despite input variation) and input separation (distinguishing between structured and random inputs regardless of task). This allows us to asses to what degree the optimization for a specific sub-goal (e.g., working memory that is present during delay period) leads to input-independent dynamics.

Heterogeneous effect, but dominated by input

We conducted analyses mentioned above separately for each task epoch or sub-goal (fixation, stimulus, delay, decision) and across all possible task pairs (CDM-PR, CDM-DC, CDM-GNG, DC-PR, DC-GNG, PR-GNG), with all comparisons including three distinct architectures (V-RNN, LSTM, GRU). This comprehensive set of comparisons allowed us to investigate

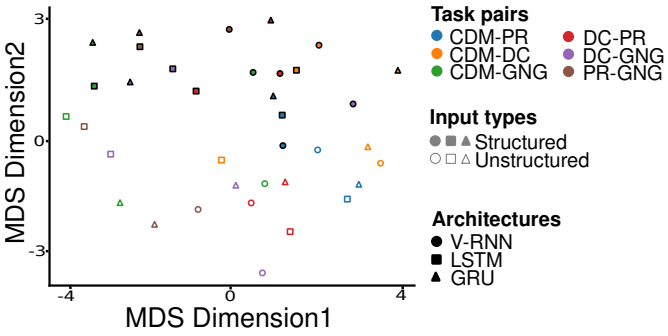

Figure 3: Clustering of network representations across architectures and task pairs. The visualization reveals input structure as the primary organizing principle in neural dynamics.

the effect of three factors on neural dynamics: how the choice of architecture (e.g., GRU), task (e.g., DM), sub-goal/motif (e.g., delay, when the working memory is used). Furthermore, it allows us to investigate how they change throughout task execution and under varying input conditions.

Although it might appear that both task input and task-optimized connectivity should equally contribute to network dynamics and thus allow us to separate the dynamics of the network based on the task equally well, we observed a very heterogeneous contribution. The separation accuracy varied considerably depending on both the specific task pair being examined and the network architecture.

To find possible organizing principles underlying this intricate set of factors affecting task-specific network dynamics, we performed a meta-clustering analysis using MDS on the combined results from both types of analysis mentioned above. Our meta-level analysis suggests that input structure is the primary factor shaping the task-specific network dynamics, irrespective of architecture and choice of the task (Figure 3).

Overall, our results suggest that task-specific network dynamics are heterogeneously shaped by multiple factors (task-input, neural architectures, computational motif, and task-optimized connectivity), with the input structure as the most dominant factor. Future work should expand this analysis to larger datasets and more diverse task structures to better characterize the effect of input structure based on the complexity of the tasks (Huang et al., 2025).

Acknowledgments

This work was supported by the TU Dresden, the German Research Foundation (DFG) grant 550411021, and the Max Planck Society. RZ and SS acknowledge the add-on fellowship from the Joachim Herz Foundation.

References

Driscoll, L. N., Shenoy, K., & Sussillo, D. (2024). Flexible multitask computation in recurrent networks uti-

lizes shared dynamical motifs. *Nature Neuroscience*, *27*(7), 1349–1363.

Guilhot, Q., Wójcik, M., Achterberg, J., & Costa, R. P. (2024). Dynamical similarity analysis can identify compositional dynamics developing in rnns.

Huang, A., Singh, S. H., & Rajan, K. (2025). Measuring and controlling solution degeneracy across task-trained recurrent neural networks.

Kornblith, S., Norouzi, M., Lee, H., & Hinton, G. (2019, September). Similarity of neural network representations revisited. In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of the 36th international conference on machine learning* (pp. 3519–3529, Vol. 97). PMLR.

Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, *29*(1), 1–27.

Molano-Mazon, M., Barbosa, J., Pastor-Ciurana, J., Fradera, M., Zhang, R.-Y., Forest, J., del Pozo Lerida, J., Ji-An, L., Cueva, C. J., de la Rocha, J., et al. (2022). Neurogym: An open resource for developing and sharing neuroscience tasks. *PsyArXiv*.

Ostrow, M., Eisen, A., Kozachkov, L., & Fiete, I. (2023). Beyond geometry: Comparing the temporal structure of computation in neural circuits with dynamical similarity analysis. *Advances in Neural Information Processing Systems*, *36*.

Yang, G. R., Joglekar, M. R., Song, H. F., Newsome, W. T., & Wang, X.-J. (2019). Task representations in neural networks trained to perform many cognitive tasks. *Nature Neuroscience*, *22*(2), 297–306.