Exploring a Basis Set of Intrinsic Functions Underlying Neural Computation by Symbolically Programming Recurrent Neural Networks

CCN

Yale University

COGNITIVE & NEURAL COMPUTATION LABORATORY

Daniel Calbick[1], Jason Z. Kim[2], Ilker Yildirim[1]

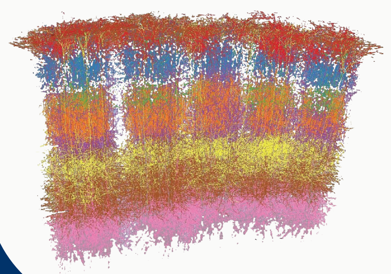
Email: daniel.calbick@yale.edu

[1] Department of Psychology, Yale University; [2] Department of Physics, Cornell University

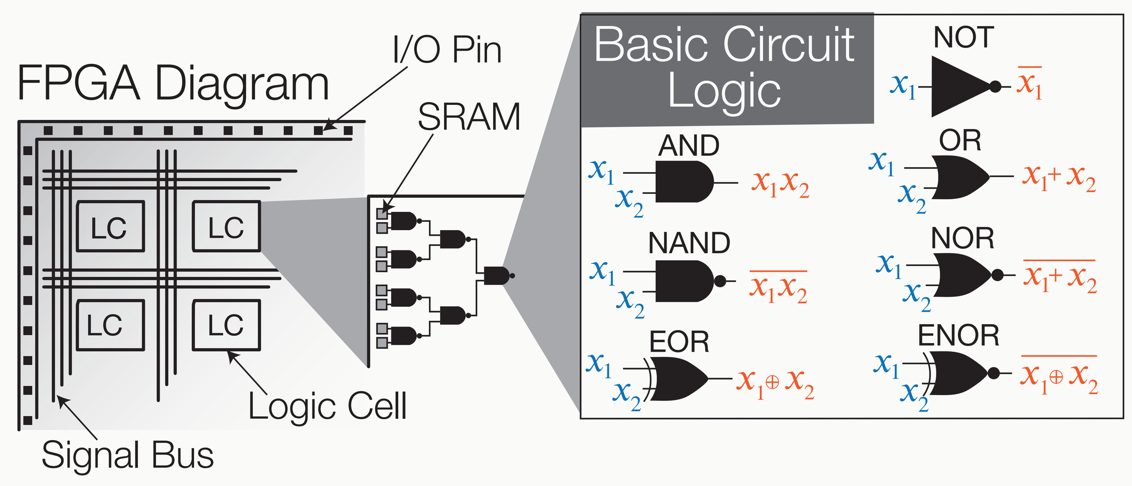
Functional Primitives in the Brain

Functional building blocks allow for circuit-level models of cognition.

Hypothesis: Unit anatomical/functional structures in the brain implement primitive functions that together combine to yield complex cognitive processes.



Example: Hyper Columns have been related to integrated circuits such as Field Programmable Gate Arrays (FPGA)



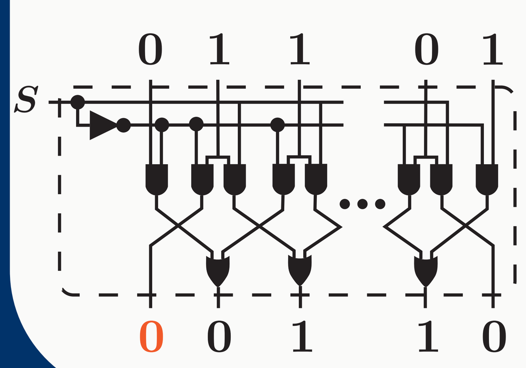
Key Issue: Existing frameworks do not allow for hypothesis-driven exploration of structured functions in neurobiology.

Existing work using **task optimized DNNs** yield black-box hypothesis about functional modules as a function of architectures, training sets, and objective functions.

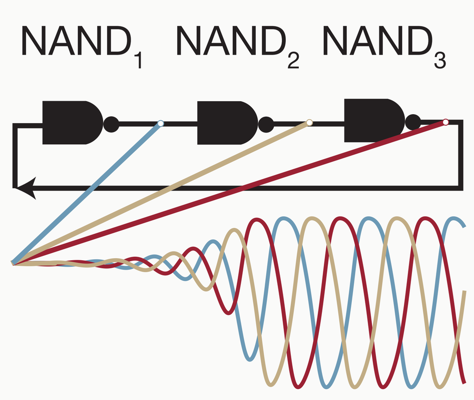
Hypothesis Driven Exploration of Functional Primitives

Candidate Functional Primitives ⟷ **Programmable Neural Networks** ⟶ **Evaluation**

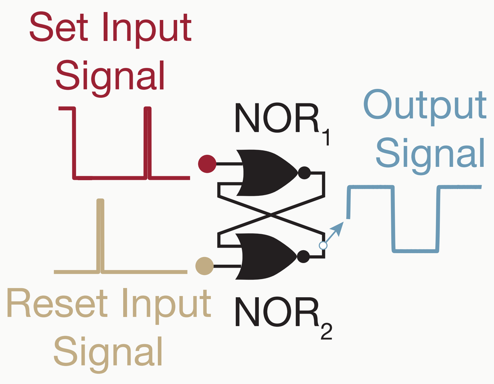
Shifter Cicuit

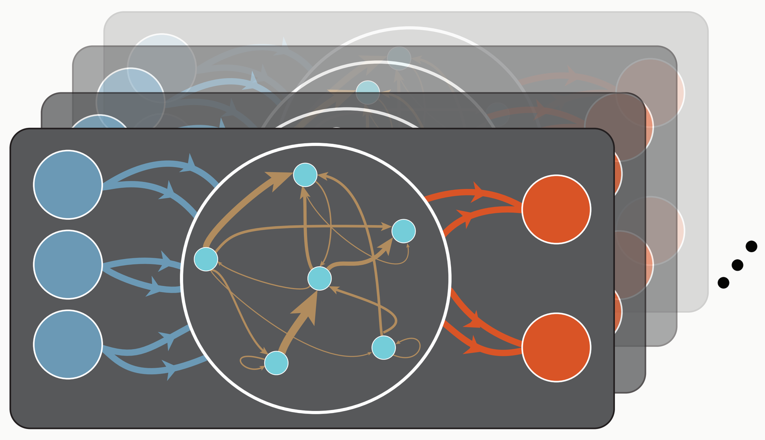


Circuit Oscillator



S-R Latch

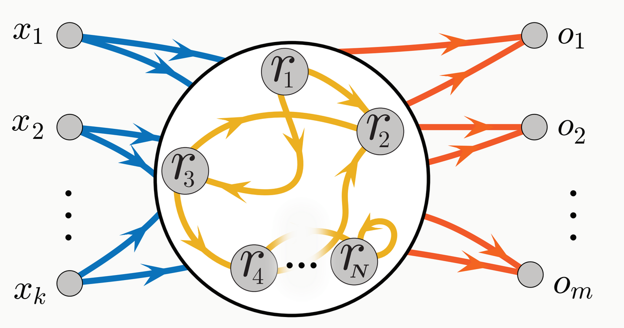




- Performance on multi-task learning
- Computational efficiency
- Robustness to noise

Symbolically Programming (not training) RNNs to Implement Hypothesis

1. Coarse Network Architecture & Equations



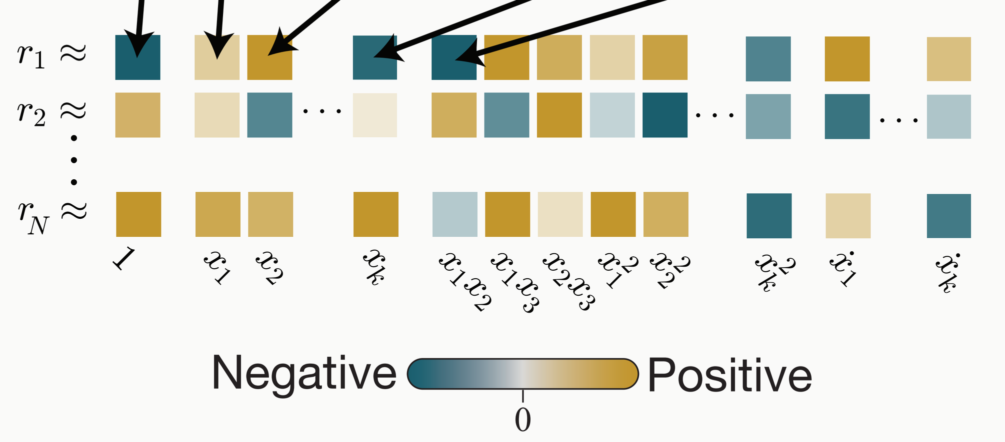
$$\frac{1}{\gamma}\dot{\boldsymbol{r}} = -\boldsymbol{r} + \boldsymbol{g}\,(A\boldsymbol{r} + B\boldsymbol{x} + \boldsymbol{d})$$

$$\boldsymbol{o} = W\boldsymbol{r}\ ;\ W = \underset{W}{\operatorname{argmin}} \|W\mathrm{r} - o\|$$

2. Decomposed Network State (r)

Taylor Series Decomposition

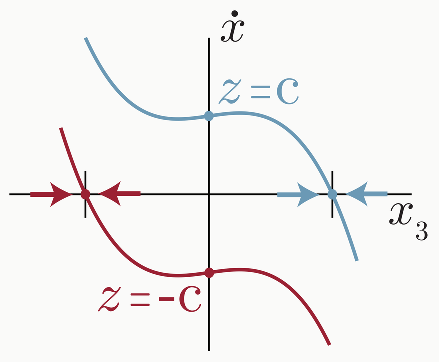
$$r_1 \approx h_1^* + \frac{\partial h_1^*}{\partial x_1}x_1 + \frac{\partial h_1^*}{\partial x_2}x_2 + \cdots + \frac{\partial h_1^*}{\partial x_k}x_k + \frac{\partial^2 h_1^*}{\partial x_1 \partial x_2}x_1x_2 + \cdots$$



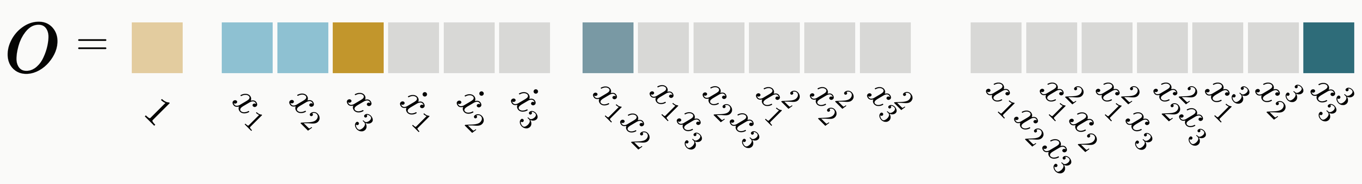
3. Pitchfork Bifurcation Supports Binary Logic

$$\dot{x} = ax_3 + bx_3^3 + z \qquad z = \mathrm{c} + \frac{(x_1 + \mathrm{c})\ (\text{-}x_2 - \mathrm{c})}{2(\mathrm{c})}$$

z implements logic table, influencing the phase diagram to evolve to either c or -c depending on inputs x_1 & x_2

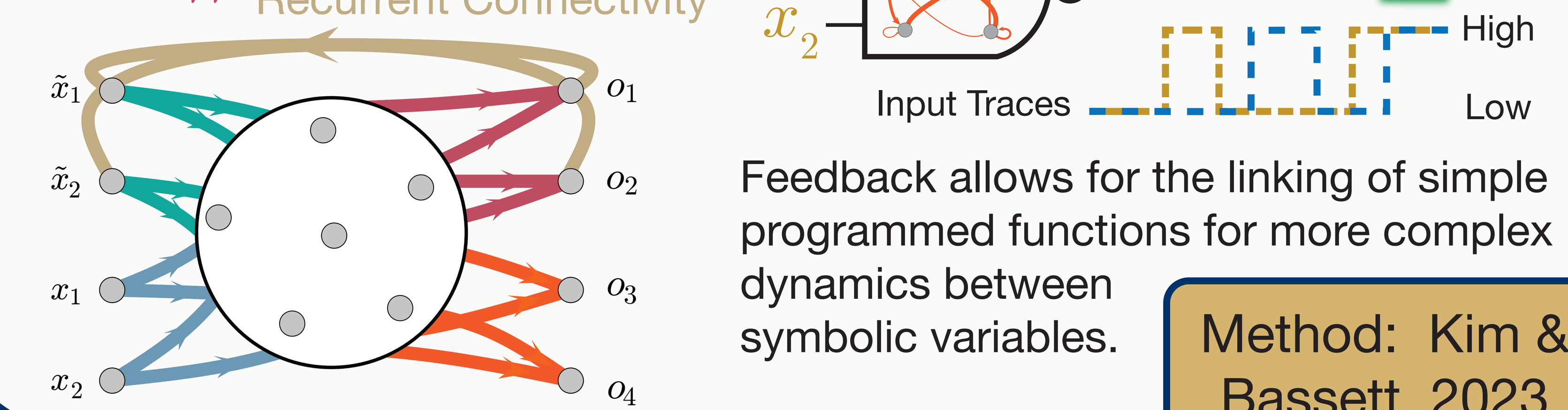


4. Programmed Output Matrix for NAND Gate



5. Scripting & Feedback

$A = \tilde{B}\tilde{W}$ Programmed Recurrent Connectivity



Feedback allows for the linking of simple programmed functions for more complex dynamics between symbolic variables.

Method: Kim & Bassett, 2023

Conclusions

- Programming RNNs to implement hypotheses starting withprimative intrinsic functions, is able to provide insights into complex brain computations and can allow us to explore cognitive functions such as learning and memory.
- Future work will expand the scope of intrinsic functions, their implementations, complexity of their interconnections, amount/type of tasks, and compare their activity to empirical data.

We thank the Yale Center for Research Computing for their support in managing the Milgram computing cluster.

Evaluation Setting: Multi-task Learning

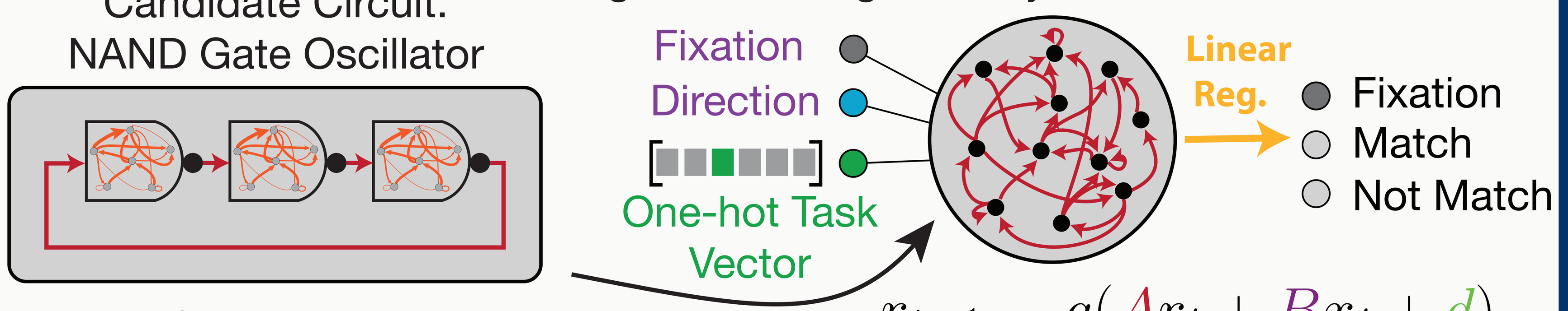
Task	Sample (e.g.)	Test (e.g.)	Response
DMS			Yes if directions match
Anti-DMS			Yes if directions do notmatch
DMC			Yes if directions are same category
Anti-DMC			Yes if directions are different category
OIC			Report direction category
Anti-OIC			Report opposite category

Fixation	Sample	Delay	Test / Response
200ms	300ms	1000ms	300ms

Multi-task Learning with Programmed RNNs

Hypothesis: Programming oscillatory dynamics into RNNs will support multi-task learning and working memory

Candidate Circuit: NAND Gate Oscillator



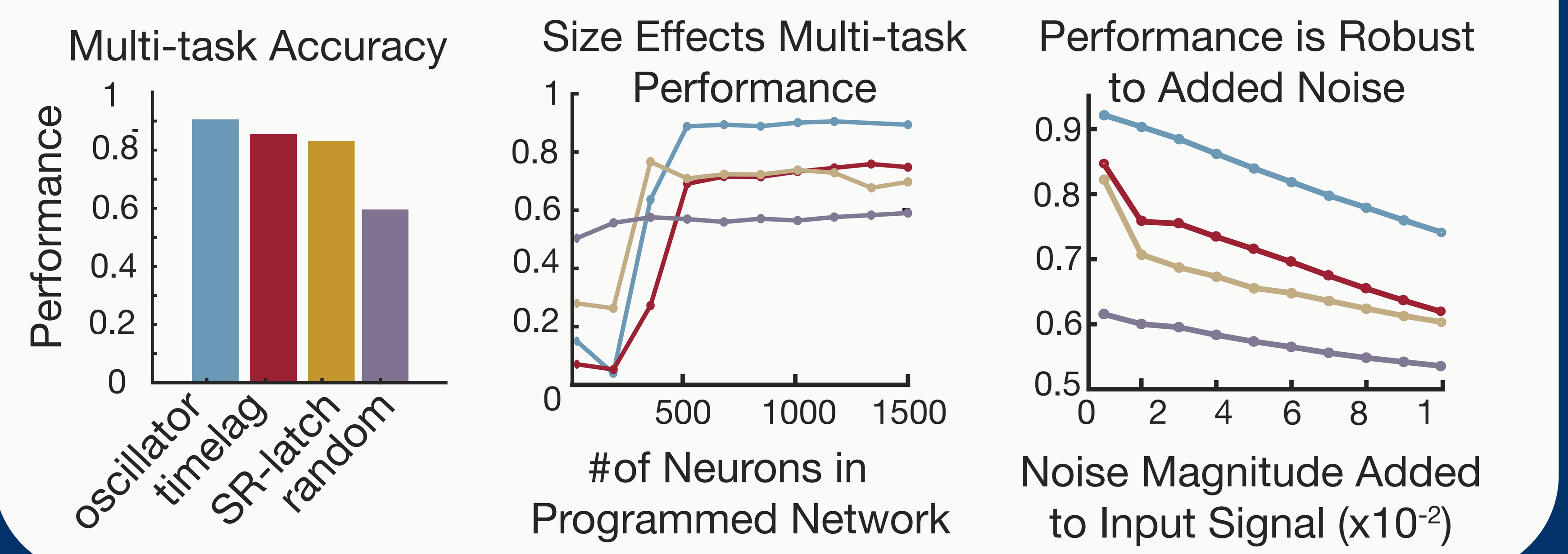
Using feedback to chain neural logic gates to form a more complex circuits

$$r_{t+1} = g(Ar_t + Bx_t + d)$$

A = Programmed Circuit Dynamics

Results

Programmed RNNs perform better than randomly weighted networks, exhibit improved performance with model size, and show robustness to noise. Type of functional primative hypotheses has variable effects on performance.



References & Acknowledgments

Kim, J. Z., & Bassett, D. S. (2023). A neural machine code and programming language for the reservoir computer. Nature Machine Intelligence.

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.

Masse, N. Y., Rosen, M. C., Tsao, D. Y., & Freedman, D. J. (2022). Flexible cognition in context-modulated reservoir networks. bioRxiv, 2022–05.

Yale Center for Research Computing