Achieving Stable And Brain-Like Dynamics in Recurrent Models via Contraction Analysis

by

Leo Kozachkov

Submitted to the Department of Brain and Cognitive Sciences
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Feb 2023

Author .....
Department of Brain and Cognitive Sciences
Jan 13, 2023

Certified by .....
Earl K. Miller
Picower Professor of Neuroscience
Thesis Supervisor

Certified by .....
Jean-Jacques Slotine
Professor
Thesis Supervisor

Accepted by .....
Mark Harnett
Graduate Officer, Department of Brain and Cognitive Sciences

Achieving Stable And Brain-Like Dynamics in Recurrent Models via Contraction Analysis

by

Leo Kozachkov

Submitted to the Department of Brain and Cognitive Sciences
on Jan 13, 2023, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

The brain consists of many interconnected networks, each with time-varying and complex dynamics. Despite this, neural activity tends to converge to reproducible sequences of states. How the brain achieves these complex-yet-stable dynamics is unknown. We address this problem using contraction analysis: a set of tools from the dynamical systems and control theory literature. Loosely, a contracting dynamical system is one whose trajectories converge towards a common—potentially time-varying and complicated—trajectory. We apply contraction analysis to recurrent neural networks (RNNs), as well as RNNs of RNNs, to derive constraints on synaptic plasticity rules and weight matrices such that the resulting models are provably contracting. By parametrizing these constraints for optimization in standard deep learning libraries, we show that contraction-constrained networks achieve high performance on sequential processing benchmark tasks (e.g., sequential CIFAR-10), as well as high similarity with frontal lobe neural activity recorded from a behaving non-human primate. Our work—both theoretical and experimental—suggests that stability-constrained recurrent architectures yield promising models of neural activity as the scope of experimental neuroscience expands into studying multiple dynamic, interacting brain areas.

Thesis Supervisor: Earl K. Miller
Title: Picower Professor of Neuroscience

Thesis Supervisor: Jean-Jacques Slotine
Title: Professor

Acknowledgments

This thesis has two dedications. The first is to the memory of my late grandfather, Professor Lev Semyonovich Kozachkov (1927-1987). Your cybernetic interests live on through your computational neuroscientist grandson. The second dedication is "to all the teachers who told me I'd never amount to nothing" [Wallace, 1994].

To Earl Miller and Jean-Jacques Slotine: who could ask for better advisors? You gave me freedom when I wanted it, and guidance when I needed it. You invested time and effort in teaching me how to think. And then when the thinking was done, how to write scientific articles. I benefited enormously from it. I'm proud to be part of your scientific *mishpachah*. Thank you, thank you! To the rest of my thesis committee, Ila Fiete and Cengiz Pehlevan, thank you for your guidance and support.

Next up I would like to thank my wife, Allison Regna, for her unwavering support and love throughout the years. I was just a pimply sixteen year old when we started dating. What you saw in me, I'll never know. All signs pointed very much south. Then I became your boyfriend and in the blink of an eye I'm twenty eight, married, and some kind of doctor! How did you make that all happen?

Marvin The Dog: you wonderful, perfect, spoiled little brat. How did you manage to get your paws under my skin so quickly? Remember when you helped me prove Theorem 2 in Kozachkov et al. [2021], while we were listening to Any Colour You Like?

To Alex and Inna Kozachkov: thank you for the nature, and also the nurture. I am blessed to have such smart, creative, and caring parents. If I am fortunate enough to give you grandkids, I will continue the line of love and creativity and support. I hope I have made you proud.

To Henry Kozachkov: I'm reminded of a quote from Cormac McCarthy's Blood Meridian: "Wolves cull themselves, man. What other creature could?". Henry the knife collector, the knife himself. Being sharp as hell and a decade my senior, you could have cut me down entirely. But instead you shaved the fat from my brain, and sharpened my mind. Not on purpose, I think. But just interacting with you was

enough. I would not be writing this if not for you, that much is obvious. Thank you for blazing the trail, and setting the example. I hope you find peace soon enough, old man. And if you can't find peace, I hope it finds you.

To the friends I've made here at MIT: John Tauber, Michael Happ, Martin Schrimpf. The damned winds are blowing us apart! We have to stay vigilant and make sure we visit each other at least semi-regularly. Thank you for the many laughs and coffees and scientific arguments and political arguments. They made the whole thing worthwhile.

To members of the Miller Lab, new and old: Mikael Lundqvist, Adam Eisen, Scott Brincat, Jefferson Roy, Meredith Manhke–thank you! To Mikael in particular, I hope I didn't beat you up too badly, you retired viking!

To Bood 6: When I think of "me", as in who I am, as in who I *really* am, I think of where I am in Bood 6. Love you all.

To Running on Empty: My brothers in *punk rock*. How should I thank you for something that will keep on going forever? I guess I have to thank you for the songs we wrote, and the songs you helped Volvo Physics write, between 2017 and 2022. Years that will always be weird and special to me.

To Cheer Camp: Thank you guys for the love and the music and hangouts. What luck that graduate school would take me right to you!

To Kevin Feigelis: Ah, my *other* crazed brother on the West Coast! When will you see the grave error in your ways and return to us on the East Coast?! Thank you for the kvetches and the songs and, sometimes, the science.

To my old advisor at Rutgers, Professor Konstantinos Michmizos. Thank you for believing in me, and teaching me the fundamentals of doing research. I would not be here if not for you. I hope one day we can continue our uncovering of the strange and beautiful world of astrocytes.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Computation Through Dynamics

This thesis adopts the view that brains perform their computations through neural dynamics. The framework of computation-through-dynamics (CTD) has been successful in explaining a variety of disparate neural phenomenon [Sussillo, 2014]. In this thesis we focus on one particular aspect of neural dynamics: *reproducibility*. Reproducibility refers to the ability of a dynamical system to produce the same output, regardless of initial conditions and disturbances. This phenomenon has been observed experimentally throughout the years, but lacks a solid theoretical understanding [Steveninck et al., 1997, Churchland et al., 2010, Lundqvist et al., 2022].

1.2 Contraction Analysis

The main theoretical apparatus used in this thesis is *contraction analysis* [Lohmiller and Slotine, 1998]. Contraction analysis is a set of analysis tools developed in the nonlinear control theory literature for understanding the stability of input-driven, nonlinear systems. The essence of a contracting dynamical systems is that it forgets its initial conditions exponentially quickly. In the language of neuroscience: if the brain is a contracting dynamical system, it will eventually do the same thing on two condition-matched trials. A contracting system has reproducible dynamics. The

reason contraction analysis is so powerful is that it applies to nonlinear systems driven by arbitrary inputs. Contracting systems can also be combined with one another in various ways, to stably build up more and more complicated systems.

1.3 Organization of Thesis

All the work discussed in this thesis is based on published or soon-to-be-published material. We direct the reader to the published documents, for any future improvements or errata. In chapter 1, we apply contraction analysis to a simple and widespread model of brain circuits: the recurrent neural network (RNN). Of particular note, we use contraction analysis to derive synaptic plasticity rules which ensure reproducible neural dynamics. This chapter is derived from Kozachkov et al. [2020], which was published in the journal Public Library Open Science (PLOS) Computational Biology. In chapter 2, we put contraction analysis to the test. Specifically, we design a distracted working memory task which tests how the Non-Human Primate (NHP) brain responds to perturbations. We then train thousands of models from different architecture classes on this task, and comparing their representations to those recorded from the NHP brain, we find that contracting RNNs most closely match the neural data. This chapter is derived from Kozachkov et al. [2022a], which is currently hosted on the preprint server bioRxiv. In chapter 3, we show that by using contraction analysis to combine contracting RNNs with one another, we can build "RNNs of RNNs" which is both provably contracting as well as expressive. In particular, we are able to achieve state of the art (SOTA) on several challenging machine learning benchmark tasks. This chapter is derived from Kozachkov et al. [2021], which will appear in the proceedings of Advances in Neural Information Processing Systems 35 (NeurIPS 22).

Chapter 2

Achieving Stable Dynamics in Individual Neural Circuits

2.1 Introduction

Behavior emerges from complex neural dynamics unfolding over time in multi-area brain networks. Even in tightly controlled experimental settings, these neural dynamics often vary between identical trials [Lundqvist et al., 2016, Churchland et al., 2010]. This can be due to a variety of factors including variability in membrane potentials, inputs, plastic changes due to recent experience and so on. Yet, in spite of these fluctuations, brain networks must achieve computational stability: despite being "knocked around" by plasticity and noise, the behavioral output of the brain on two experimentally identical trials needs to be similar. How is this stability achieved?

Stability has played a central role in computational neuroscience since the 1980's, with the advent of models of associative memory that stored neural activation patterns as stable point attractors [Hopfield, 1982, Hirsch, 1989, Cohen and Grossberg, 1983, Lundqvist et al., 2011, Lansner and Ekeberg, 1985], although researchers were thinking about the brain's stability since as early as the 1950's [Ashby, 1952a]. The vast majority of this work is concerned with the stability of activity around points, lines, or planes in neural state space [Dayan and Abbott, 2005, Zhang et al., 2014]. However,

recent neurophysiological studies have revealed that in many cases, single-trial neural activity is highly dynamic, and therefore potentially inconsistent with a static attractor viewpoint [Spaak et al., 2017]. Consequently, there has been a number of recent studies—both computational and experimental—which focus more broadly on the stability of neural *trajectories* [Laje and Buonomano, 2013, Chaisangmongkon et al., 2017] which may be complex and time-varying.

While these studies provide important empirical results and intuitions, they do not offer analytical insight into mechanisms for achieving stable trajectories in recurrent neural networks. Nor do they offer insights into achieving such stability in plastic (or multi-modal) networks. Here we focus on finding conditions that guarantee stable trajectories in recurrent neural networks and thus shed light onto how stable trajectories might be achieved *in vivo*.

To do so, we used contraction analysis, a concept developed in control theory [Lohmiller and Slotine, 1998]. Unlike a chaotic system where perturbations and distortions can be amplified over time, the population activity of a contracting network will converge towards the same trajectory, thus achieving stable dynamics (Figure 1). One way to understand contraction is to represent the state of a network at a given time as a point in the network's 'state-space', for instance the space spanned by the possible firing rates of all the networks' neurons. This state-space has the same number of dimensions as the number of units n in the network. A particular pattern of neural firing rates corresponds to a point in this state-space. This point moves in the n dimensions as the firing rates change and traces out a trajectory over time.

In a contracting network, all such trajectories converge. These contracting dynamics have previously been used in several applications, including neural networks with winner take all dynamics [Rutishauser et al., 2011, 2015] , in a model of action-selection in the basal ganglia [Girard et al., 2008], and to explain how neural synchronization can protect from noise [Tabareau and Slotine, 2006]. Here, we instead explore how contraction can be achieved generally in more complex recurrent neural networks (RNNs) including those with plastic weights. We used RNNs that received arbitrary time-varying inputs and had synapses that changed on biologically relevant timescales

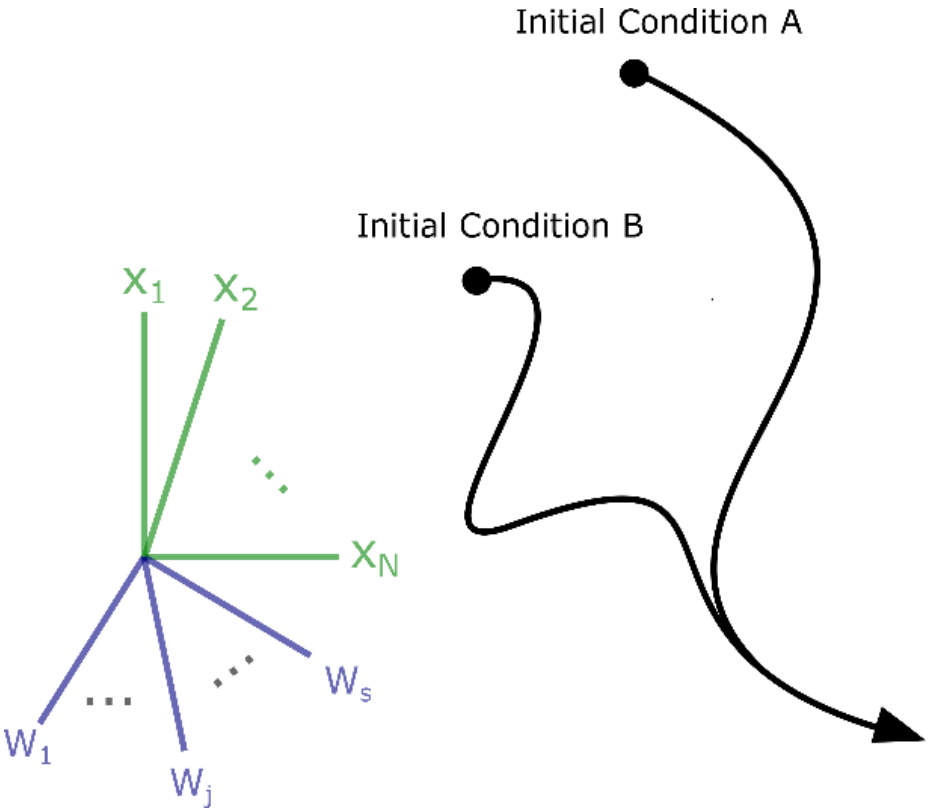

Figure 2-1: Cartoon demonstrating the contraction property. In a network with N neurons and S dynamic synaptic weights, the network activity can be described a trajectory over time in an $N + S$dimensional space. In a contracting system all such trajectories will converge exponentially *in some metric* towards each other over time, regardless of initial conditions. In other words, the distance between any two trajectories shrinks to zero— potentially after transient divergence (as shown).

[Orhan and Ma, 2019]. Our analysis reveals several novel classes of mechanisms that produced contraction including inhibitory Hebbian plasticity, excitatory anti-Hebbian plasticity, excitatory-inhibitory balance, and sparse connectivity. For the first two parts of the Results section, we focus on contraction of *both* neural activity *and* components of the weight matrix (Fig 1). For the remaining parts of the Results section, we hold the weights fixed (i.e they become parameters, not variables) and focus on contraction of neural activity alone.

2.2 Results

The main tool we used to characterize contraction was the logarithmic norm (also known as a matrix measure). The formal definition of the logarithmic norm is as follows: let $\mathbf{A}$ be a matrix in $C^{n\times n}$ and $||\cdot||_i$ be an induced matrix norm on $C^{n\times n}$. Then the corresponding logarithmic norm is the function $\mu(\cdot) : C^{n\times n} \to \mathbb{R}$ defined by

$$\mu(\mathbf{A}) = \lim_{\epsilon \to 0^+} \frac{||I + \epsilon \mathbf{A}||_i - 1}{\epsilon}$$

In the same way that different vector norms induce different matrix norms, different vector norms also induce different logarithmic norms. Two important logarithmic norms which we use throughout the paper are those induced by the vector 1-norm and the vector 2-norm:

$$\mu_1(\mathbf{A}) = \max_j \left[a_{jj} + \sum_{i \neq j}^{n} |a_{ij}| \right] \qquad \mu_2(\mathbf{A}) = \lambda_{max}(\frac{\mathbf{A}^* + \mathbf{A}}{2})$$

Where λ_{max} denotes the largest eigenvalue. To study the contraction properties of RNNs, we applied the logarithmic norm to the RNN's Jacobians. The Jacobian of a dynamical system is a matrix essentially describing the local 'traffic laws' of nearby trajectories of the system in its state space. More formally, it is the matrix of partial derivatives describing how a change in any system variable impacts the rate of change of every other variable in the system. It was shown in Lohmiller and Slotine [1998] that if the logarithmic norm of the Jacobian is negative then all nearby trajectories are funneled towards one another (see S1A Text Section A.1.2 for technical review). This, in turn, implies that all trajectories are funneled towards one another at rate called the contraction rate. The contraction rate and the logarithmic norm are related as follows: the maximum value attained by the absolute value of logarithmic norm of the Jacobian along the network's trajectory is the contraction rate. In other words, if the logarithmic norm of the Jacobian is upper bounded by some negative number $\breve{}c$, where $c > 0$, then the contraction rate is simply c.

Importantly, the above description can be generalized to different metrics. A metric

is a symmetric, positive definite matrix which generalizes the notion of Euclidean distance. Every invertible coordinate transformation $\mathbf{y} = \theta\mathbf{x}$ yields a metric $\mathbf{M} = \theta^T\theta$. To see this, consider the squared norm of

$$||\mathbf{y}||^2 = \mathbf{y}^T\mathbf{y} = \mathbf{x}^T\mathbf{\Theta}^T\mathbf{\Theta}\mathbf{x} = \mathbf{x}^T\mathbf{M}\mathbf{x}$$

Thus, the norm of $\mathbf{y}$ is related to the norm of $\mathbf{x}$ through the metric $\mathbf{M}$. If one can find metric in which the network is contracting—in the sense that its Jacobian has negative logarithmic norm – this implies contraction for all coordinate systems. This makes contraction analysis useful for analyzing systems where exponential convergence of trajectories is preceded by transient divergence (Figure 1) as in recent models of motor cortex [Hennequin et al., 2014]. In this case, it is usually possible to find a coordinate system in which the convergence of trajectories is 'pure'. For example, linear stable systems were recently used in the motor control literature to find initial conditions which produce the most energetic neural response [Hennequin et al., 2014]. They are 'purely' contracting in a metric defined by the eigenvectors of the weight matrix (see Example 5.1 in [Lohmiller and Slotine, 1998]) but transiently diverging in the identity metric (i.e $\mathbf{M} = \mathbf{I}$). Note that the identity metric corresponds to $\mathbf{\Theta} = \mathbf{I}$, which is simply the original, untransformed coordinate system.

2.3 Inhibitory Hebbian Plasticity & Excitatory Anti-Hebbian Plasticity Produce Contraction

It is known that certain forms of synaptic plasticity can quickly lead to extreme instabilities if left unchecked [Dayan and Abbott, 2005]. Thus, the same feature that can aid learning can also yield chaotic neural dynamics if not regulated. It is not known how the brain resolves this dilemma. A growing body of evidence—both experimental and computational—suggests that inhibitory plasticity (that is, the strengthening of inhibitory synapses) can stabilize neural dynamics while simultaneously allowing for learning/training in neural circuits [Vogelsy et al., 2013]. By using the Jacobian

analysis outlined above, we found that inhibitory Hebbian synaptic plasticity (as well as excitatory anti-Hebbian plasticity) indeed leads to stable dynamics in neural circuits. Specifically, we considered neural networks of the following common form:

$$\dot{x}_i = h\left(x_i\right) + \sum_{j=1}^{N} W_{ij} x_j + u_i\left(t\right)$$

where the term x_i denotes the 'activation' of neuron i as a function of time. Here we follow other authors [Hennequin et al., 2014] and interpret x_i as the deviation from the baseline firing rate of neuron i. Note that this interpretation assumes that the baseline firing rates are positive– thus allowing for x to be negative—and large enough so that baseline $+ \ x \ > \ 0$. The term W_{ij} denotes the weight between neurons i and j the term $h\left(x_i\right)$ captures the dynamics neuron i would have in the absence of synaptic input, including self-feedback terms arising from the diagonal elements of the weight matrix—in other words, the dynamics neuron i would have if for all i and j, $W_{ij} = 0$. The term being summed represents the weighted contribution of all the neurons in the network on the activity of neuron i. Finally, the term $u_i(t)$ represents external input into neuron i.

We did not constrain the inputs into the RNN (except that they were not infinite) and we did not specify the particular form of $h\left(x_i\right)$ except that it should be a leak term (i.e has a negative derivative for all x, see S1A Text Section A.2.2.4, e.g $h(x_i) \ = \ \breve{} xi$). Furthermore, we made no assumptions regarding the relative timescales of synaptic and neural activity. Synaptic dynamics were treated on an equal footing as neural dynamics. We considered synaptic plasticity of the following correlational form [Gerstner and Kistler, 2002]:

$$W_{ij} = -k_{ij} x_i x_j - \gamma(t) W_{ij}$$

where the term $k_{ij} > 0$ is the learning rate for each synapse and $\gamma\left(t\right) > 0$ is a decay factor for each synapse. For technical reasons outlined in the appendix (S1A Text Section A.3), we restricted $\mathbf{K}$, the matrix containing the learning rates k_{ij}, to be positive semi-definite, symmetric, and have positive entries. A particular example of $\mathbf{K}$

satisfying these constraints is to have the learning rates of all synapses to be equal (i.e. $k_{ij} = k > 0$). Before we show that (2) leads to overall synaptic and neural contraction, it's useful to spend some time interpreting this plasticity. SinceW_{ij} can be positive or negative (corresponding to excitatory and inhibitory synapses, respectively), and $x_i x_j$ can be positive or negative (corresponding to correlated and anticorrelated neurons, respectively), there are four cases to consider. We summarize these cases in Table 1 and discuss them in details below. By Hebbian plasticity we refer to the increase of synaptic efficiency between correlated neurons [Gerstner and Kistler, 2002]. In the context of simple neural networks with scalar weights, as we consider here, efficiency refers to the absolute value $|w|$ of a weight. Thus, for excitatory synapses,(2) in fact describes anti-Hebbian plasticity, because the positive synaptic weight becomes less positive (and thus less efficient) between correlated neurons and more positive (thus more efficient) for anticorrelated neurons. For inhibitory synapses, (2) describes Hebbian plasticity because the direction of synaptic weight change is negative between correlated neurons, and thus the synapse becomes more efficient [Hosoya et al., 2005], while for anticorrelated neurons the direction of synaptic weight change is positive, and thus the synapse becomes less efficient. Plasticity of this form produced contracting neural and synaptic dynamics regardless of the initial values of the weights and neural activity (Figure 2 and Figure 3). The black trace of Figure 3A shows that this is not simply due to the weights decaying to 0. Thus, this plasticity is not only contraction preserving, it is contracting ensuring. Furthermore, we showed that the network is contracting in a non-identity metric (which we derive from the system parameters in K), opening up the possibility of transient divergent dynamics in the identity metric, as seen in the modelling of motor dynamics [Hennequin et al., 2014].

To explain how inhibitory Hebbian plasticity and excitatory anti-Hebbian plasticity work to produce contraction across a whole network, we needed to deal with the network in a holistic fashion, not by analyzing the dynamics of single neurons. To do so, we conceptualized RNNs with dynamic synapses as a single system formed by combining two subsystems, a neural subsystem and a synaptic subsystem. We showed that the above plasticity rule led the neural and synaptic subsystems to be

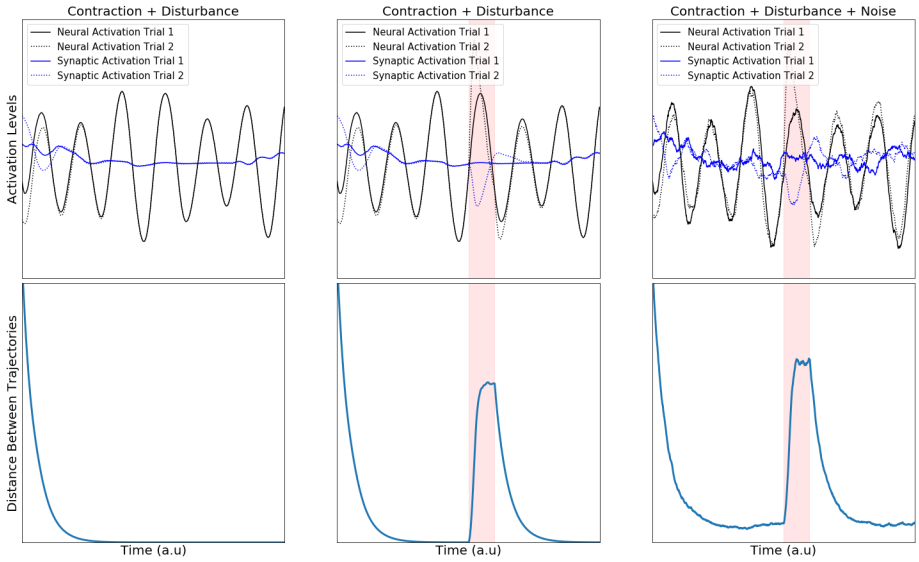

Figure 2-2: Contracting dynamics of neural and synaptic activity. Euclidean distances between synaptic and neural trajectories demonstrate exponential shrinkage over time. The top row of panels shows the activation of a randomly selected neural unit (black) and synapse (blue) across two simulations (dotted and solid line). The bottom row shows the average Euclidean distance in state space for the whole population across simulations with distinct, randomized starting conditions. Leftmost Panel: Simulations of a contracting system where only starting conditions differ over simulations. Center Panel: the same as in Leftmost but with an additional random pulse perturbation in one of the two simulations indicated by a red background shading. Rightmost Panel: same as in Center Panel but with additional sustained noise, unique to each simulation.

independently contracting. Thus contraction analysis of the overall system then boiled down to examining the interactions between these subsystems [Slotine, 2002]. We found that this plasticity works like an interface between these systems. It produces two distinct effects that push networks toward contraction. First, it makes the synaptic weight matrix symmetric (Figure 3A, red trace). This means that the weight between neuron i to j is the same as j to i. We showed this by using the fact that every matrix can be written as the sum of a purely symmetric matrix and a purely anti-symmetric matrix. An anti-symmetric matrix is one where the ij element is the negative of the ji element (i.e. $W_{ij} = -W_{ji}$) and all the diagonal elements are zero. We then showed

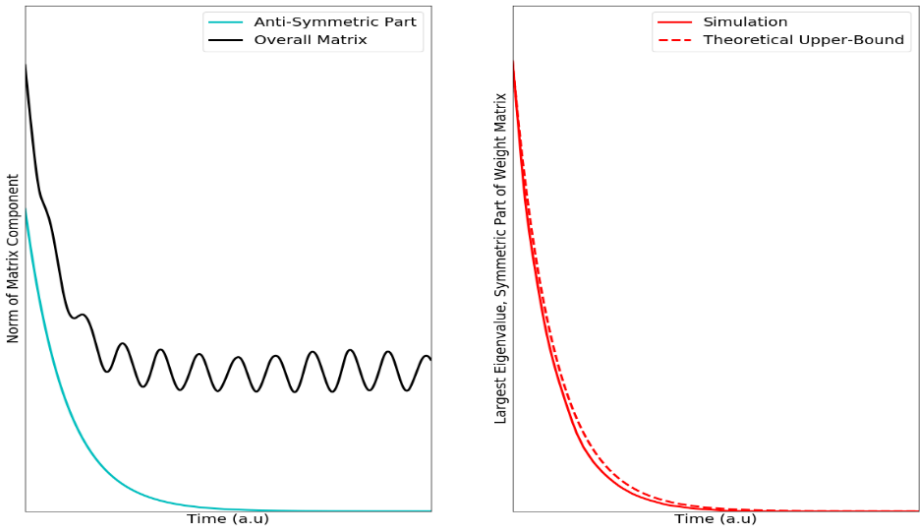

Figure 2-3: (Left) The anti-Hebbian plasticity pushes the weight matrix towards symmetry. Left) Plotted are the spectral norms (largest singular value) of the overall weight matrix as well as the anti-symmetric part of that matrix. Since every square matrix can be uniquely decomposed as the sum of a symmetric and anti-symmetric component—0.5*(W+W') and 0.5*(W-W‘), respectively–the teal curve decaying to zero implies that the matrix becomes symmetric. The black trace shows the spectral norm of the overall weight matrix. If this quantity does not decay to zero, it implies that not all the weights have decayed to zero. On the right, we plot the largest eigenvalue of the symmetric part of W. A prerequisite for overall contraction of the network is that this quantity be less than or equal to the 'leak-rate' of the individual neurons. The dotted line shows our theoretical upper bound for this quantity, and the solid line shows the actual value of taken from a simulation (see methods).

that anti-Hebbian plasticity shrinks the anti-symmetric part of the weight matrix to zero, implying that the weight matrix becomes symmetric. The symmetry of the weight matrix 'cancels out' off-diagonals in the Jacobian matrix (see S1A Text Section A.3) of the overall neural-synaptic system. Loosely speaking, off-diagonal terms in the Jacobian represent potentially destabilizing cross-talk between the two subsystems. Furthermore, anti-Hebbian plasticity makes the weight matrix negative semi-definite. This means that all its eigenvalues are less than or equal to zero (Figure 3).

2.4 Sparse Connectivity Pushes Networks toward Contraction

Synaptic connectivity in the brain is extraordinarily sparse. The adult human brain contains at least $10^1 1$ neurons yet each neuron forms and receives on average only $10^3 - 10^4$ synaptic connections [Kandel et al., 2000]. If the brain's neurons were all-to-all connected this number would be on the order of 10^{11} synaptic connections per neuron ($\frac{10^{11} \cdot 10^{11}}{10^{11}} \frac{\text{synaptic connections}}{\text{neurons}}$. Even in local patches of cortex, such as we model here, connectivity is far from all-to-all; cortical circuits are sparse [Song et al., 2005]. Our analyses revealed that sparse connectivity helps produce global network contraction for many types of synaptic plasticity. To account for the possibility that some synapses may have much slower plasticity than others (and can thus be treated as synapses with fixed amplitude), we made a distinction between the total number of synapses and the total number of plastic synapses. These plastic synapses then changed on a similar time-scale as the neural firing rates. By neural dynamics, we mean the change in neural activity as a function of time. We analyzed RNNs with the structure:

$$\dot{x}_i = h_i\left(x_i\right) + \sum_{j=1}^{N} W_{ij}\left(t\right)\ r(x_j) + u_i\left(t\right)$$

Where $h_i(x_i)$ is a nonlinear leak term (see S1A Text Section A.2.2.4), and $r(x_j)$ is a nonlinear activation function. The RNNs analyzed in this section are identical to those analyzed in the previous section, with the exception of the r terms, which we constrained to be linear. Under the assumption that the plastic synapses have a 'forgetting term', we show in the appendix (S1A Text SectionSection 4) that if the following equation is satisfied for every neuron, then the overall network is contracting:

$$p_i\left(g_{max} w_{max} + \alpha_i r_{max}\right) <\ \beta_i$$

where p_i denotes the total number of afferent synapses into neuron i and d_i denotes the number of afferent plastic synapses into neuron i. Note that since the number of

dynamic synapses cannot be greater than the total number of synapses, d_i has to be a fraction of p_i, This means we can write it as $d_i = \alpha_i p_i$, where α_i is a number between 0 and 1. The term w_{max} refers to the maximum possible absolute efficiency of any single synapse.

That is, $w_{max} = \max_{i,j} |W_{ij}|$. Similarly, the term r_{max} refers to the maximum possible absolute value of r. That is

$$r_{max} = \max_{i,t} \ |r_i(t)|$$

The term β_i denotes the contraction rate of the i^{th} isolated neuron. That is,

$$\beta_i = -\max_{i,t} \frac{\partial h_i}{\partial x_i}(t)$$

Recall from the introduction that the contraction rate measures how quickly the trajectories of a contracting system reconvene after perturbation. Finally, g_{max} refers to the maximum gain of any neuron in the network. That is:

$$g_{max} = \max_{i,t} \left| \frac{\partial r_i}{\partial x_i} \right| (t)$$

Note that because β_i is a positive number by assumption, it is always possible to decrease p_i to the point where (4) is satisfied. Of course, it is possible that the only value of p_i that satisfies (4) is the trivial solution $p_i = 0$, which corresponds to removing all interconnections between neurons. Since these neurons are assumed to be contracting in isolation, the network is trivially contracting. However, if the term inside the parentheses of (4) is small enough, or β_i is large enough, intermediate value of p_i can be found which satisfy the inequality. Because increasing the sparsity of a network corresponds to decreasing p_i, we may conclude that increasing the sparsity of connections pushes the system in the direction of contraction. Note that (4) also implies that the faster the individual neurons are contracting (i.e. the larger β_i is), the denser you can connect them with other neurons while still preserving overall contraction. Up to now we have focused our analysis on the case where synaptic weights vary on a timescale comparable to neurons, and must therefore be factored

into the stability analysis. For the next two sections, we'll apply contraction analysis to neural network in the case where the weights may be regarded as fixed relative to the neural dynamics (i.e there is a separation of timescales).

2.5 E-I Balance Leads to Contraction in Static RNNs

Apart from making connections sparse, one way to ensure contraction is to make synaptic weights small. This can be seen for the case with static synapses by setting $\alpha_i = 0$ in the section above, where w_{max} now has to be small to ensure contraction. Intuitively, this is because very small weights mean that neurons cannot exert much influence on one another. If the neurons are stable before interconnection, they will remain so. Since strong synaptic weights are commonly observed in the brain, we were more interested in studying when contraction can arise irrespective of weight amplitude. Negative and positive synaptic currents are approximately balanced in biology [**?**Wehr and Zador, 2003]. We reasoned that such balance might allow much larger weight amplitudes while still preserving contraction since most of the impact of such synapses cancel and the net effect small. This was indeed the case. To show this, we studied the same RNN as in the section above, while assuming additionally that the weights are static. In particular, we show in the appendix (S1A Text SectionSection 5) that contraction can be assessed by studying the eigenvalues of the symmetric part of W (i.e. $\frac{\mathbf{W}+\mathbf{W}^\mathbf{T}}{\mathbf{2}}$).

Before we discuss the above result in detail, it is useful here to quickly review some facts about the stability of nonlinear systems as compared to the stability of linear systems. In particular, the fact that the eigenvalues of $\mathbf{W}$ are only informative for assessing contraction in regions where the dynamics may be regarded as linear. This is because in linear time-variant (LTI) systems (i.e $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$) stability is completely characterized in terms of the eigenvalues of $\mathbf{A}$. However, this is not true for nonlinear systems, even those of the linear time-varying form $\dot{\mathbf{x}} = \mathbf{A}(t)\,\mathbf{x}$. To see this, consider the following counter-example (from [Slotine and Li, 1991], section 4.2.2):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -1 & e^{2t} \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

The eigenvalues of $\mathbf{A}(t)$ are $(-1, -1)$ for all time, however one can verify by direct evaluation that the solution of this system satisfies $y = y(0)\, e^{-t}, \dot{x} = -x + y(0)\, e^{t}$ which is unstable along x. However, it can be shown straightforwardly that if the eigenvalues of the symmetric part of $\mathbf{A}(t)$ are all negative, then the system is stable [Slotine and Li, 1991]. This fact underlies our analysis, and highlights the reason why the eigenvalues of the symmetric part of $\mathbf{W}$ are important for stability. Returning to our results, we show that if excitatory to inhibitory connections are of equal amplitude (and opposite sign) as inhibitory to excitatory connections, they will not interfere negatively with stability—regardless of amplitude (see S1A Text Section A.5). This is because connections between inhibitory and excitatory units will be in the off-diagonal of the overall weight matrix and get cancelled out when computing the symmetric part. As an intuitive example, consider a two-neuron circuit made of one excitatory neuron and one inhibitory neuron connected recurrently (as in [Murphy and Miller, 2009], Fig 1A). Assume that the overall weight matrix has the following structure:

$$\mathbf{W} = \begin{pmatrix} w & -w \\ w & -w \end{pmatrix}$$

When taking the symmetric part of this matrix, the off-diagonal elements cancel out, leaving only the diagonal elements to consider. Since the eigenvalues of a diagonal matrix are simply its diagonal elements, we can conclude that if the excitatory and inhibitory subpopulations are independently contracting (w is less than the contraction rate of an isolated neuron), then overall contraction is guaranteed. It is straightforward to generalize this simple two-neuron example to circuits achieving E-I balance through interacting populations (see S1A Text Section A.5). It is also straightforward to generalize to the case where E-I and I-E connections do not cancel out exactly neuron by neuron, but rather they cancel out in a statistical sense where the mean amplitudes are matched. Another way to view this is E-I balance is in the framework of

combinations of contracting systems (Fig 4). It is known that combining independently contracting systems in negative feedback preserves contraction [Lohmiller and Slotine, 1998]. We show that E-I balance actually translates to this negative feedback and thus can preserve contraction.

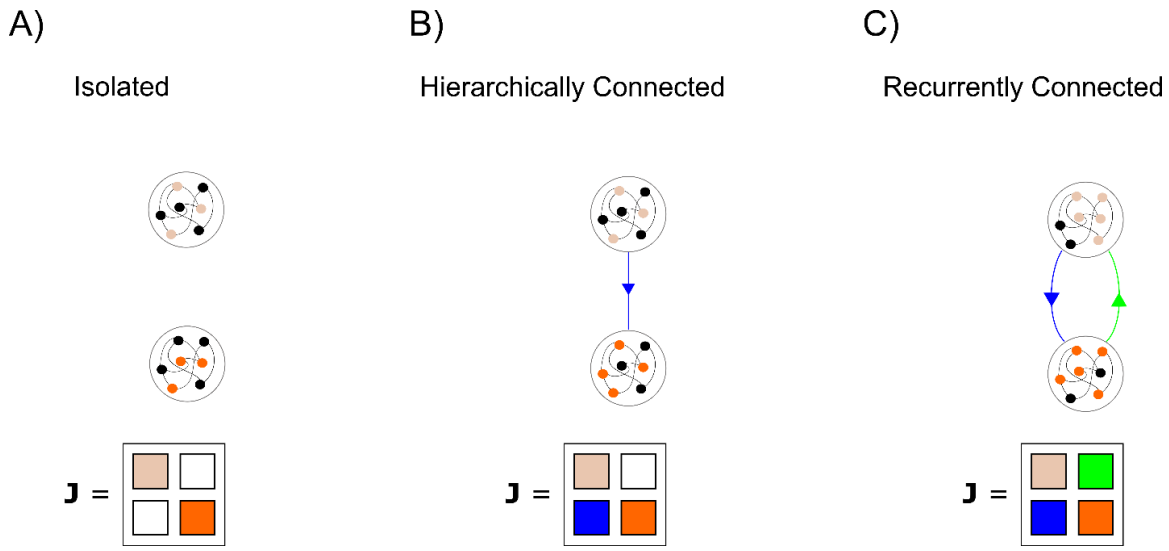

Figure 2-4: Cartoon illustrating the combination properties of contracting systems. A) Two isolated, contracting systems. The Jacobian of the overall system is block diagonal, with all zeros on the off-diagonal—corresponding to the fact that the systems are not connected. B) If one of the systems is connected to the other in a feedforward manner, the overall Jacobian is changed by the presence of non-zero terms on the bottom left block—corresponding to the connections going from the 'top' system to the 'bottom' system. This Jacobian may not be negative definite. However, it is known that a coordinate change exists which will make it negative definite. Thus, hierarchically connected contracting systems are contracting. C) If the systems are reciprocally connected, the system may lose its contracting properties (for example in the case of positive feedback). However, it is known that if the feedforward connections (blue) are 'equal and opposite' to the feedback connections (green) then the overall system is contracting. We use this property in the main text to prove that inhibitory Hebbian plasticity and excitatory anti-Hebbian plasticity lead to contracting neural circuits.

2.6 Relation to Other Models with Fading Memory

As can be seen in Figure 2, contracting systems have 'fading memories'. This means that past events will affect the current state, but that the impact of a transient

perturbation gradually decays over time. Consider the transient input in Figure 2 (red panel) presented on only one of the two trials to the network. Because the input is only present on one trial and not the other we call it a perturbation. When this perturbation occurs, the trajectories of the two trials become separated. However, after the disturbance is removed, the distance between the network's trajectories starts shrinking back to zero again. Thus, the network does not hold onto the memory of the perturbation indefinitely—the memory fades away. A similar property has been used in Echo State Networks (ESNs) and liquid state machines (LSMs) to perform useful brain-inspired computations [Jaeger, 2001]. These networks are an alternative to classical attractor models in which neural computations are performed by entering stable states rather than by 'fading memories' of external inputs [Buonomano and Maass, 2009]. While there are several distinctions between the networks described above and ESNs (e.g ESNs are typically discrete time dynamical systems, rather than continuous), we show in the appendix (S1A Text Section 5.1) that they are a special case of the networks considered here. We show this for ESNs as opposed to LSMs because LSMs are typically implemented on integrate and fire neurons which, because of the spike reset, have a sharp discontinuity in their dynamics—making them unamenable to contraction analysis. By highlighting the link between contraction and ESNs, we demonstrate that the contracting neural networks considered here are in principle capable of performing useful and interesting neural computations. In other words, the strong stability properties of contracting neural networks do not automatically prohibit them from doing interesting computations. By working within the framework of contraction analysis we were able to study networks both with dynamic synapses and non-identity metrics—a much broader model space than allowed by the standard ESN framework.

2.7 Discussion

We studied a fundamental question in neuroscience: How do neural circuits maintain stable dynamics in the presence of disturbance, noisy inputs and plastic change? We

approached this problem from the perspective of dynamical systems theory, in light of the recent successes of understand neural circuits as dynamical systems [?]. We focused on contracting dynamical systems, which are yet largely unexplored in neuroscience, as a solution to the problem outlined above. We did so for three reasons: 1) Contracting systems can be input-driven. This is important because neural circuits are typically bombarded with time-varying inputs either from the environment or from other brain areas. Previous stability analyses have focused primarily on the stability of RNNs without time-varying input. These analyses are most insightful in situations where the input into a circuit can be approximated as either absent or constant. However, naturalistic stimuli tend to be highly time-varying and complex. 2) Contracting systems are robust to noise and disturbances. Perturbations to a contracting system are forgotten at the rate of the contraction and noise therefore does not stack up over time. Importantly, the rate of forgetting (i.e the contraction rate) does not change with the size of the perturbation. Thus dynamic stability can co-exist with high trial-to-trial variability in contracting neural networks, as observed in biology. 3) Contracting systems can be combined with one another in ways that preserve contraction (Fig 4). This is not true of most dynamical systems which can easily 'blow up' when connected in feedback with one another [Ashby, 1952a]. This combination property is important as it is increasingly clear that cognitive functions such as working memory or attention are distributed in multiple cortical and sub-cortical regions [Chatham and Badre, 2015, Halassa and Kastner, 2017]. In particular, prefrontal cortex has been suggested as a hub that can reconfigure the cortical effective network based on task demands [Miller and Cohen, 2001]. Brain networks must therefore be able to effectively reconfigure themselves on a fast time-scale without loss of stability. Most attempts in modelling cognition, for instance working memory, tend to utilize single and often autonomous networks. Contracting networks display a combination of input-driven and autonomous dynamics, and thus have key features necessary for combining modules into flexible and distributed networks. To understand what mechanisms lead to contraction in neural circuits, we applied contraction analysis to RNNs. For RNNs with static weights, we found that the well- known Echo State Networks are a

special case of a contracting network. Since realistic synapses are complex dynamical systems in their own right, we went one step further and asked when neural circuits with dynamic synapses would be contracting. We found that inhibitory Hebbian plasticity as well as excitatory anti-Hebbian plasticity and synaptic sparsity all lead to contraction in a broad class of RNNs. Inhibitory plasticity has recently been the focus of many experimental and computational studies due to its stabilizing nature as well as its capacity for facilitating nontrivial computations in neural circuits [Hennequin et al., 2017]. It is known to give rise to excitatory-inhibitory balance and has been implicated as the mechanism behind many experimental findings such as sparse firing rates in cortex [Vogels et al., 2011]. Similarly, anti-Hebbian plasticity exists across many brain areas and species, such as salamander and rabbit retina [Hosoya et al., 2005], rat hippocampus [Lisman, 1989], electric fish electrosensory lobe [Enikolopov et al., 2018] and mouse prefrontal cortex [Ruan et al., 2014]. Anti-hebbian dynamics can give rise to sparse neural codes which decrease correlations between neural activity and increase overall stimulus representation in the network [**?**]. Because of this on-line decorrelation property, anti-Hebbian plasticity has also been implicated in predictive coding [Enikolopov et al., 2018]. Our findings suggest that it also increase the stability of networks. For more general forms of synaptic dynamics, we showed that synaptic sparsity pushes RNNs towards being contracting. This aligns well with the experimental observation that synaptic connectivity is typically extremely sparse in the brain. Our results suggest that sparsity may be one factor pushing the brain towards dynamical stability. It is therefore interesting that synapses are regulated by homeostatic processes where synapses neighboring an upregulated synapse are immediately downregulated [El-Boustani et al., 2018]. On the same note, we also observed that balancing the connections between excitatory and inhibitory populations leads to contraction. Balance between excitatory and inhibitory synaptic inputs are often observed in biology [**?**], and could thus serve contractive stability purposes. Related computational work on spiking networks has suggested that balanced synaptic currents leads to fast response properties, efficient coding, increased robustness of function and can support complex dynamics related to movements. A main advantage

to our approach is that it provides provable certificates of global contractive stability for nonlinear, time-varying RNNs with synaptic plasticity. This distinguishes it from previous works where—while very interesting and useful—stability is experimentally observed, but not proven [Laje and Buonomano, 2013]. In some cases [Hennequin et al., 2014], linear stability around the origin is proven (which implies that there is a contraction region around the origin) but the size of this region is neither established nor sought after. Indeed, one future direction we are pursuing is the question of: given an RNN, can one provide a certificate of contractive stability in a region? An answer to this question would shed light on the stability properties of known RNN models in the literature (e.g trained RNNs, biologically-detailed spiking models, etc). Experimental neuroscience is moving in the direction of studying many interacting neural circuits simultaneously. This is fueled by the expanding capabilities of recording multiple areas simultaneously in vivo and study their interactions. This increases the need for multi-modal cognitive models. We therefore anticipate that the presented work can provide a useful foundation for how cognition in noisy and distributed computational networks can be understood.

2.8 Materials and Methods

In the interested of space and cohesion, we have placed all the detailed proofs of main results into the appendix. The appendix was written to be self-contained, and thus also contains additional definitions of mathematical objects used throughout the text. Simulations (Figures 2 and 3) were performed in Python. Code to reproduce the figures is available at this link. Numerical integrating was performed using sdeint, an open-source collection of numerical algorithms for integrating stochastic ordinary differential equations.

2.8.1 Figure 2 details

: All parameters and time constants in equations (1) and (2) were set to one. The integration step-size, dt, was set to 1e-2. Initial conditions for both neural and synaptic

activation were drawn uniformly between -1 and 1. Inputs into the network were generated by drawing N frequencies uniformly between dt and 100dt, phases between 0 and 2π, amplitudes between 0 and 20 and generating an N x Time vector of sinusoids with the above parameters. The perturbations of the network was achieved by adding a vector of all 10s (i.e an additive vector input into the network, with each network of the element equal to 10) to the above input on one of the trials for 100 time steps in the middle of the simulation. The noise was generated by driving each neural unit with an independent Weiner process (sigma = .2).

2.8.2 Figure 3 details

The weight matrix used was the same as in Figure 2, leftmost panel (without perturbation, without noise).

2.8.3 Acknowledgment

We thank Pawel Herman for comments on an earlier version of this manuscript. We thank Michael Happ and all members of the Miller Lab for helpful discussions and suggestions. This work was supported by NIMH R37MH087027, The MIT Picower Institute Innovation Fund, ONR MURI N00014-16-1-2832, and Swedish Research Council Starting Grant 2018-04197.

Appendix

2.A Preliminaries

2.A.1 Mathematical Preliminaries

Matrix Notions

We will denote vectors and matrices with bold font. For example: $\mathbf{x} \in \mathbb{R}^N$ is an N-dimensional column vector and $\mathbf{A} \in \mathbb{R}^{N \times M}$ is an $N \times M$ matrix. We use the

notation $\mathbf{P} \succ 0$ to denote a positive definite matrix, which is a square matrix such that, for all $\mathbf{x} \in \mathbb{R}^N \neq \mathbf{0}$, the quadratic form $\mathbf{x}^T\mathbf{P}\mathbf{x}$ is strictly positive:

$$\mathbf{P} \succ 0 \equiv \mathbf{x}^T\mathbf{P}\mathbf{x} > 0 \tag{2.1}$$

When the quadratic form is non-negative (i.e $\mathbf{x}^T\mathbf{P}\mathbf{x} \geq 0$) then we say that that $\mathbf{P}$ is positive semi-definite and write $\mathbf{P} \succcurlyeq 0$. Negative definiteness and semi-definiteness are defined the same way, with the sign of the above inequalities reversed.

Every real, square matrix $\mathbf{Q}$ can be decomposed into a symmetric part and a skew symmetric part. We denote the symmetric part of $\mathbf{Q}$ as $\mathbf{Q}_s$ and define it as follows

$$\mathbf{Q}_s \equiv \frac{\mathbf{Q} + \mathbf{Q}^T}{2} \tag{2.2}$$

and the antisymmetric (skewsymmeric) part as

$$\mathbf{Q}_a \equiv \frac{\mathbf{Q} - \mathbf{Q}^T}{2} \tag{2.3}$$

Note that $\mathbf{Q} = \mathbf{Q}_s + \mathbf{Q}_a$. Also note that without ambiguity, when we say that a non-symmetric $\mathbf{Q}$ is positive definite, we mean that $\mathbf{Q}_s \succ 0$. This is because:

$$\mathbf{x}^T\mathbf{P}\mathbf{x} = \mathbf{x}^T\mathbf{P}_s\mathbf{x}$$

Substituting this equality back into (2.1), we see that a matrix is positive definite if and only if its symmetric part is positive definite. For more information on positive definite matrices, see section 7.1 of Horn et al. [1990] or section 3.5.1 of Slotine and Li [1991], for example.

Matrix Measures

We will also need the fact that a given vector norm $|\cdot|$ will induce a matrix norm $||\cdot||$

$$||\mathbf{Q}|| \equiv \sup_{|x|=1} |\mathbf{Q}\mathbf{x}|$$

as well as a matrix measure $\mu(\cdot)$

$$\mu(\mathbf{Q}) \equiv \lim_{h \to 0} \frac{||\mathbf{I} - h\mathbf{Q}|| - 1}{h}$$

where $\mathbf{I}$ denotes the identity matrix. Different norms induce different matrix measures, but all matrix measures share certain common properties. In particular, for any matrix measure μ we have subadditivity:

$$\mu(\mathbf{X} + \mathbf{Y}) \leq \mu(\mathbf{X}) + \mu(\mathbf{Y}) \tag{2.4}$$

positive homogeneity:

$$\forall c \geq 0, \;\; \mu(c\mathbf{X}) = c\mu(\mathbf{X}) \tag{2.5}$$

as well as

$$\mu(\mathbf{I}) = 1 \text{ and } \mu(-\mathbf{I}) = -1 \tag{2.6}$$

For more information about matrix measures the reader is referred to section 2.2.2 of Vidyasagar [2002] or section 2 of Desoer and Haneda [1972].In this paper we will rely primarily on the matrix measures induced by the infinity-norm $|\cdot|_\infty$ and the 2-norm $|\cdot|_2$. These norms induce the matrix measures

$$\mu_\infty(\mathbf{Q}) = \sup_i (Q_{ii} + \sum_{j \neq i} |Q_{ij}|) \tag{2.7}$$

$$\mu_2(\mathbf{Q}) = \lambda_{\max}(\mathbf{Q}_s) \tag{2.8}$$

Finally, consider the (potentially time-varying), non-singular matrix $\mathbf{\Theta}$ and the associated weighted 2-norm and weighted infinity-norm : $|\mathbf{x}|_{\Theta,2} = |\mathbf{\Theta x}|_2$. The matrix measures induced by these norms are:

$$\mu_{2,\Theta}(\mathbf{Q}) = \mu_2(\dot{\mathbf{\Theta}}\mathbf{\Theta}^{-1} + \mathbf{\Theta Q \Theta}^{-1}) \tag{2.9}$$

$$\mu_{\infty,\Theta}(\mathbf{Q}) = \mu_{\infty}(\dot{\mathbf{\Theta}}\mathbf{\Theta}^{-1} + \mathbf{\Theta}\mathbf{Q}\mathbf{\Theta}^{-1}) \tag{2.10}$$

Matrix Vecorization

Finally, consider an arbitrary $N \times N$ matrix $\mathbf{A}$. The vectorization of $\mathbf{A}$, which we denote by $\mathbf{v}_A$, is the $N^2 \times 1$ column vector obtained by stacking the columns of $\mathbf{A}$ on top of one another, starting from the leftmost column and ending at the rightmost column. Consider the example of:

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

then

$$\mathbf{v}_A = \begin{bmatrix} a \\ c \\ b \\ d \end{bmatrix}$$

We also define the $N^2 \times N^2$ matrix $\mathbf{D}_A$ as the diagonal matrix with the elements of $\mathbf{A}$ along its diagonal, starting with the leftmost column of $\mathbf{A}$ and ending with the rightmost column. For the $\mathbf{A}$ matrix defined above, we have:

$$\mathbf{D}_A = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & c & 0 & 0 \\ 0 & 0 & b & 0 \\ 0 & 0 & 0 & d \end{bmatrix}$$

These two operators have a special relationship to the Hadamard product (element-wise matrix multiplication). Consider a matrix $\mathbf{Z}$ defined as:

$$\mathbf{Z} = \mathbf{X} \odot \mathbf{Y}$$

where $\mathbf{X}$ and $\mathbf{Y}$ are of the same size. Then we have the following relationship:

$$\mathbf{v}_Z = \mathbf{D}_X \mathbf{v}_Y \tag{2.11}$$

which can be verified through direct inspection.

2.A.2 Basic Contraction Definition

In this section we define what a contracting system is, and state several properties of these systems that will be useful for the remainder of the supplementary. The reader is referred to Lohmiller and Slotine [1998] and Sontag [2010] for more details. Consider a deterministic system of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) \tag{2.12}$$

where $\mathbf{f}$ is an $N\times 1$ nonlinear vector function and $\mathbf{x}$ is the $N\times 1$ state vector. We assume that all quantities are real and smooth, by which we mean that any required derivative or partial derivatives exist and are continuous. Denote the $N \times N$ time-varying Jacobian of this system as $\mathbf{J}(\mathbf{x}, t) = \mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}$.

If there exists a matrix measure μ such that

$$\forall \mathbf{x}, t \quad \mu(\mathbf{J}) \leq -\lambda \text{ with } \lambda > 0 \tag{2.13}$$

then all trajectories of (2.12) converge exponentially towards one another with rate λ and the system is said to be *contracting*. In particular, let $R(t) = |\mathbf{x}(t) - \mathbf{x}'(t)|$ denote the distance between any two trajectories of system (2.12) and assume the system is contracting. Then there exists a constant $k \geq R(0)$ such that

$$0 \leq R(t) \leq ke^{-\lambda t}$$

To avoid ambiguity in language, we now define some terms associated with the measures μ_1,μ_2 and μ_∞ . As in Lohmiller and Slotine [1998], consider a metric $\mathbf{M} = \mathbf{\Theta}^T\mathbf{\Theta} \succ 0$ and the following associated quantity, called the *generalized Jacobian*:

$$\mathbf{F} = \dot{\mathbf{\Theta}}\mathbf{\Theta}^{-1} + \mathbf{\Theta}\mathbf{J}\mathbf{\Theta}^{-1} \tag{2.14}$$

If the matrix measures induced by the 1,2, or infinity norm are negative definite when applied to $\mathbf{F}$, i.e.

$$\mu_i(\mathbf{F}) < -\lambda \text{ with } i = 1, 2, \text{ or } \infty \tag{2.15}$$

then we say that the system is contracting in metric $\mathbf{\Theta}$ with rate λ. Calling $\mathbf{\Theta}$ a metric is a slight abuse of language, because the real metric is $\mathbf{M} = \mathbf{\Theta}^T\mathbf{\Theta}$, but since we will always use the symbol $\mathbf{\Theta}$ in the generalized Jacobian, this won't cause any confusion.

2.B Contracting Dynamics of A Leaky Neuron

In this section we apply contraction analysis to the dynamics of a single, leaky, input-driven neuron. Analyzing the case of one neuron will turn out to be helpful when thinking about the case of many neurons.

2.B.1 A Motivating Example

Consider the activation of a pure integrator neuron that is receiving constant input, I:

$$\dot{x} = I \tag{2.16}$$

This differential equation can be solved analytically, and has the solution:

$$x(t) = It + x(0) \tag{2.17}$$

Since the solution does not forget x(0), its initial condition, this system is not contracting. However, if we were to add a 'leak' term, for example $-x$ to (2.16), we would have

$$\dot{x} = -x + I \tag{2.18}$$

which yields the solution

$$x(t) = I + \chi e^{-t} \tag{2.19}$$

with $\chi = x(0) - I$. Since this system forgets $x(0)$ exponentially quickly, we may conclude that it is contracting (of course we also could have computed the Jacobian, which in the case of (2.18) is -1). In the case of nonlinear dynamics and time-varying input, we will not be able to solve for the dynamics of x explicitly to determine contracting properties–instead, we will have to analyze the Jacobian.

2.B.2 Leaky Neuron with No Autapse

Let x denote the activation of a single, isolated, leaky-integrator neuron. Assume that this neuron has no self-connection (i.e no autapse) and obeys the equation:

$$\tau\dot{x} = -\alpha x + I(t) \tag{2.20}$$

where $\tau > 0$ is the time-constant of integration, $\alpha > 0$ is the time-constant of the 'leak', and $I(t)$ is an external input into the neuron. If $\alpha = 0$ this would be a pure integrator neuron (see section 2.B.1).Dividing through by τ, we may rewrite (2.20) as

$$\dot{x} = -\frac{\alpha}{\tau}x + \frac{1}{\tau}I(t)$$

the Jacobian of this scalar system is:

$$J = \frac{\partial \dot{x}}{\partial x} = -\frac{\alpha}{\tau} \tag{2.21}$$

since α and τ are positive, $\frac{\alpha}{\tau}$ is also positive, which means that $-\frac{\alpha}{\tau}$ is negative. Since the Jacobian is negative, we can conclude that the isolated leaky neuron is contracting for all inputs. In the paper we will consider several classes of leaky neurons,

including neurons that obey the equation

$$\tau \dot{x} = h(x) + I(t) \tag{2.22}$$

where $h(x)$ is a nonlinear function that satisfies

$$\exists \beta > 0, \quad \forall\, x, \quad \forall\, t \geq 0, \quad \frac{\partial h}{\partial x} \leq -\beta < 0 \tag{2.23}$$

By the same arguments as above, leaky neurons of this form are also contracting in isolation. Of course, the standard leak term of $-x$ is a special case of $h(x)$. For $h(x) = -x$ we find that $\beta = 1$.

2.C Synaptic Dynamics that Favor or Preserve Contraction

The assumption that synaptic dynamics are local introduces strong structure into the Jacobian of the overall network. In particular, it implies that the 'synaptic' diagonal block of the Jacobian will be a diagonal matrix. We'll now establish some simple facts about matrices with this structure that will be useful later on.

Theorem 1. *Consider the following matrix*

$$\mathbf{F} = \begin{bmatrix} \mathbf{X} & \mathbf{GA} \\ -\mathbf{BG}^T & \mathbf{D} \end{bmatrix}$$

where $\mathbf{X}$ *is an arbitrary negative definite matrix,* $\mathbf{D}$ *is diagonal and negative definite,* $\mathbf{A}$ *and* $\mathbf{B}$ *are diagonal and positive definite and* $\mathbf{G}$ *is arbitrary. Then there exists a diagonal metric* $\mathbf{\Theta}$ *such that*

$$\mathbf{\Theta F \Theta}^{-1} \prec 0 \tag{2.24}$$

Proof. Consider the candidate metric

$$\mathbf{\Theta} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{L} \end{bmatrix}$$

where $\mathbf{L}$ is some nonsingular matrix, to be determined. Applying this to the matrix $\mathbf{F}$, we find that

$$\mathbf{\Theta F \Theta}^{-1} = \begin{bmatrix} \mathbf{X} & \mathbf{GAL}^{-1} \\ -\mathbf{LBG}^T & \mathbf{D} \end{bmatrix}$$

We would like the off-diagonal terms of this matrix to cancel out when taking the symmetric part. To make this happen, we need

$$\mathbf{BL} = \mathbf{AL}^{-1}$$

Which can be easily manipulated into

$$\mathbf{L}^2 = \mathbf{AB}^{-1}$$

and thus

$$\mathbf{L} = \sqrt{\mathbf{AB}^{-1}}$$

The square root is guaranteed to exist because $\mathbf{A}$ and $\mathbf{B}$ are both positive definite and diagonal by assumption. Taking the symmetric part of (2.24), we find that:

$$\mathbf{\Theta F \Theta}^{-1} + (\mathbf{\Theta F \Theta}^{-1})^T = \begin{bmatrix} \mathbf{X} + \mathbf{X}^T & 0 \\ 0 & 2\mathbf{D} \end{bmatrix}$$

which is negative definite since the diagonal blocks are negative definite by assumption. This proves the result.

□

We will use this result in the next section to show that anti-hebbian plasticity gives rise to contracting dynamics.

2.C.1 Anti-Hebbian Dynamics Produce Contraction

Consider RNNs of the following form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{W}, t) = \mathbf{h}(\mathbf{x}) + \mathbf{W}\mathbf{x} + \mathbf{u}(t)$$

Where $h_i(x_i)$ is a leak-term (see (2.23) for definition), $\mathbf{W}$ is the weight matrix of the network, and $\mathbf{u}(t)$ is a time-varying input. And consider the plasticity described in the main text,given by

$$\dot{W}_{ij} = -k_{ij}x_i x_j - \gamma(t)W_{ij}$$

where the matrix of k_{ij} terms, $\mathbf{K}$ must be positive semi-definite for reasons to be discussed. To prove that this plasticity leads to overall contracting dynamics, we need to show that the overall Jacobian of the system is contracting in some metric. We will do this in several steps. To give an overview of the strategy, first write the synaptic dynamics as an $N^2 \times 1$ vector:

$$\dot{\mathbf{v}}_W = \mathbf{g}(\mathbf{v}_W, \mathbf{x})$$

where $\mathbf{v}_W$ denotes the vectorization of $\mathbf{W}$. The Jacobian of the overall neural-synaptic system is then:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial \mathbf{x}} & \frac{\partial \mathbf{f}}{\partial \mathbf{v_W}} \\ \frac{\partial \mathbf{g}}{\partial \mathbf{x}} & \frac{\partial \mathbf{g}}{\partial \mathbf{v_W}} \end{bmatrix} \tag{2.25}$$

The goal will be to show that the off-diagonal blocks of this overall Jacobian cancel out in some metric, while the diagonal blocks are negative definite.

Off-Diagonal Blocks of the Jacobian

To begin, first consider the following observation. For weight dynamcics given by:

$$\forall i \neq j \;\; \dot{w}_{ij} = -x_i x_j - \gamma(t)w_{ij} \;\text{ and }\; \dot{w}_{ii} = -\frac{1}{2}(x_i)^2 - \gamma(t)w_{ii} \tag{2.26}$$

with $\gamma(t) > 0$, we have that the off-diagonal blocks of the overall Jacobian cancel out.

First we will show that w_{ij} converges to w_{ji} after exponential transients for (2.26). Consider a simple distance function between w_{ij} and w_{ji}:

$$V = (w_{ij} - w_{ji})^2$$

evaluating the time-derivative of V and substituting in (2.26), we see that

$$\dot{V} = 2(w_{ij} - w_{ji})(-x_i x_j - \gamma(t) w_{ij} + x_i x_j + \gamma(t) w_{ji}) = -2\gamma(t) V$$

which implies that the distance between w_{ij} and w_{ji} shrinks exponentially with at least rate $\min_t(\gamma(t))$, independently of initial condition or the particular trajectory of $\mathbf{x}$. For the rest of the analysis we will therefore assume that $w_{ij} \approx w_{ji}$. Using the symmetry of $\mathbf{W}$, the following four relationship can be shown:

$$\forall i \neq j \;\; \frac{\partial \dot{w}_{ij}}{\partial x_i} = -x_j \;\text{ and that }\; \frac{\partial \dot{x}_i}{\partial w_{ij}} = x_j$$

as well as

$$\forall i \neq j \;\; \frac{\partial \dot{w}_{ji}}{\partial x_i} = -x_j \;\text{ and that }\; \frac{\partial \dot{x}_i}{\partial w_{ji}} = x_j$$

The first three of these relationships can be verified by direct differentiation of (2.26). The last relationship can be seen by using the symmetry of $\mathbf{W}$. In particular, that:

$$\frac{\partial \dot{x}_i}{\partial w_{ji}} \approx \frac{\partial \dot{x}_i}{\partial w_{ij}} = x_j$$

What we have shown now is that under the synaptic dynamics defined above, we have the following relationship:

$$\frac{\partial \mathbf{g}}{\partial \mathbf{x}} = -\frac{\partial \mathbf{f}}{\partial \mathbf{v}_W}^T \tag{2.27}$$

Now , define a 'Hebbian' term as $\mathbf{H} \equiv \mathbf{x}\mathbf{x}^T$. The synaptic dynamics defined by (2.26) can in matrix notation can be written as

$$\dot{\mathbf{W}} = -\mathbf{P} \odot \mathbf{H} - \gamma(t)\mathbf{W} \tag{2.28}$$

where

$$P_{ij} = \begin{cases} 1 & \text{if } i \neq j \\ \frac{1}{2} & \text{if } i = j \end{cases}$$

vectorizing (2.28) and using the relation (2.11), we can see that

$$\dot{\mathbf{v}}_W = -\mathbf{D}_P \mathbf{v}_H - \gamma(t)\mathbf{v}_W$$

and differentiating this expression with respect to $\mathbf{x}$, we have:

$$\frac{\partial \dot{\mathbf{v}}_W}{\partial \mathbf{x}} = -\mathbf{D}_P \frac{\partial \mathbf{v}_H}{\partial \mathbf{x}} = -\frac{\partial \mathbf{f}}{\partial \mathbf{v}_W}$$

and finally, left multiplying this expression by the inverse of $\mathbf{D}_P$ we have

$$\frac{\partial \mathbf{v}_H}{\partial \mathbf{x}} = \mathbf{D}_P^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{w}}^T \tag{2.29}$$

Note that this expression is only true when $\mathbf{W}$ is symmetric. Now consider the synaptic dynamics in the main text:

$$\dot{\mathbf{W}} = -\mathbf{K} \odot \mathbf{H} - \gamma(t)\mathbf{W}$$

where $\mathbf{K}$ is symmetric , positive semi-definite and has positive entries. Vectorizing in the same manner as above and substituting in (2.29), we find that

$$\frac{\partial \mathbf{v}_{\dot{W}}}{\partial \mathbf{x}} = -\mathbf{D}_K \frac{\partial \mathbf{v}_H}{\partial \mathbf{x}} = -\mathbf{D}_K \mathbf{D}_P^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{v}_W}$$

If $\mathbf{D}_K$ is positive (which will happen when $\mathbf{K}$ has all positive entries), then $\mathbf{D}_K \mathbf{D}_P^{-1}$ is diagonal and positive definite, and thus Theorem 1 applies–this is a feedback

combination that automatically preserves contraction.In other words, a metric can be found in terms of $\mathbf{D}_K\mathbf{D}_P^{-1}$ which leaves the diagonal blocks of the Jacobian untouched and forces the off-diagonal blocks to cancel out when computing the symmetric part of the Jacobian.

Diagonal Blocks of the Jacobian

The upper left block of the Jacobian is

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} + \mathbf{W}(t)$$

thus, for this matrix to be negative definite we need

$$\mu_2(\mathbf{W}(t)) < \beta \tag{2.30}$$

To see that this will be the case, consider that the synaptic dynamics defined in the main text can be written in matrix notation as

$$\dot{\mathbf{W}} = -\mathbf{K} \odot \mathbf{x}\mathbf{x}^T - \gamma(t)\mathbf{W}$$

Now discretize the differential equations for $\mathbf{W}$:

$$\dot{\mathbf{W}} \approx \frac{\mathbf{W}_{t+1} - \mathbf{W}_t}{\Delta t} = -\mathbf{K} \odot \mathbf{x}_t\mathbf{x}_t^T - \mathbf{W}_t$$

rearranging this to collect all the t terms on the right hand side, we have

$$\mathbf{W}_{t+1} = (1 - \Delta t)\mathbf{W}_t - \Delta t\mathbf{K} \odot \mathbf{x}_t\mathbf{x}_t^T$$

To make derivations less tedious, let's define $\beta \equiv \Delta t > 0$ and $\alpha \equiv (1 - \beta)$. Note that since that, ideally, Δt is small, we can say that $0 < \alpha < 1$. Now we have the discrete dynamical system for $\mathbf{W}$:

$$\mathbf{W}_{t+1} = \alpha\mathbf{W}_t - \beta\mathbf{K} \odot \mathbf{x}_t\mathbf{x}_t^T$$

We know need to notice that $-\mathbf{K} \odot \mathbf{x}_t \mathbf{x}_t^T$ is symmetric (since $\mathbf{K}$ and $\mathbf{x}_t \mathbf{x}_t^T$ are symmetric). It is also negative semi-definite, by the Schur Product Theorem. Let's denote this expression by $\mathbf{N}_t = \mathbf{N}_t^T \prec 0$ so that

$$\mathbf{W}_{t+1} = \alpha \mathbf{W}_t + \beta \mathbf{N}_t$$

Taking the symmetric part of both sides we find that:

$$(\mathbf{W}_{t+1})_s = \alpha (\mathbf{W}_t)_s + \beta \mathbf{N}_t$$

We can now use Weyl's inequality for the sum of Hermitian matrices (see premilinaries) to conclude that

$$\mu_2(\mathbf{W}_{t+1}) \leq \alpha \mu_2(\mathbf{W}_t)$$

and therefore that

$$\mu_2(\mathbf{W}_t)) \leq \alpha^t \mu_2(\mathbf{W}_0)$$

which, since $0 < \alpha < 1$, means that the symmetric part of $\mathbf{W}$ decays exponentially to zero. Recall that for contraction we needed $\mu_2(\mathbf{W}) < \beta$. Assuming that $\mu_2(\mathbf{W}_0) > \beta$, the upper-bound for the number of steps this will take is $O(\ln(\frac{\mu_2(\mathbf{W}_0)}{\beta}))$.

To summarize, we have shown that the synaptic dynamics described by

$$\dot{\mathbf{W}} = -\mathbf{K} \odot \mathbf{x}\mathbf{x}^T - \gamma(t)\mathbf{W} \tag{2.31}$$

where $\mathbf{K}$ is a symmetric, positive semi-definite matrix with positive entries, lead to contracting neural and synaptic dynamics in a wide class of RNNs. We did this by showing that the overall Jacobian (2.25) has diagonal blocks that 'cancel' out and diagonal blocks that are negative definite–implying negativedefinitness of the whole Jacobian and thus contraction of the combined neural and synaptic system. Crucially, to show this we had to use a metric other than the identity metric.

2.D Sparsity Ensures Contraction

In this section we show that it is possible to ensure contraction–without any knowledge of the specific synaptic dynamics–by keeping the network sufficiently sparse. This result is a straightfoward application of the μ_∞ matrix measure to the Jacobian of the overall neural and synaptic system. Consider neural networks of the form

$$\dot{x}_i = h(x_i) + \sum_{j \neq i}^{N} W_{ij} r_j + u_i(t) \tag{2.32}$$

where h_i is a leak-term and r_i is bounded activation function that satisfies the following two properties:

$$|r_i| \leq r_{max} \text{ and } 0 < g_{min} \leq \frac{\partial r_i}{\partial x_i} \leq g_{max} \tag{2.33}$$

We consider synaptic dynamics of the form

$$\dot{W}_{ij} = g(W_{ij}, x_i, x_j) \tag{2.34}$$

That is to say, we only consider synaptic dynamics that are *local*, in the sense that the change in weight is only a function of the weight itself, and the pre and post synaptic neurons.

It will be useful to define, for neuron i, the total number of afferent synapses and the total number of afferent *dynamic* synapses which we denote p_i and d_i, respectively. Note that d_i is a fraction of p_i and can therefore be written as $d_i = \alpha_i p_i$ for some $0 \leq \alpha_i \leq 1$. It will also be useful to give names to one more global system parameter: the maximum amplitude (absolute value) of all the weights in the network: $w_* \equiv \max_{ij}(|w_{ij}|)$. For the neural rows of the Jacobian, we see that the μ_∞ measure can be upper bounded as

$$sup_i(\frac{\partial h_i}{\partial x_i} + \sum_{j \neq i}^{N} |\frac{\partial \dot{x}_i}{\partial x_i}| + \sum_{q=1}^{d_i} |\frac{\partial \dot{x}_i}{\partial W_{iq}}|) < 0 \tag{2.35}$$

By using the following two relations:

$$\frac{\partial \dot{x}_i}{\partial x_j} = W_{ij}\frac{\partial r_j}{\partial x_j} \text{ and } \frac{\partial \dot{x}_i}{\partial W_{ij}} = r_j$$

we can see that (2.35) can be assured if

$$\forall i, t \;\; p_i g_{max} w_* + d_i r_{max} < \beta$$

Substituting in the relationship $d_i = \alpha_i p_i$ into the above, we get:

$$\forall i, t \;\; p_i[g_{max} w_* + \alpha_i r_{max}] < \beta \tag{2.36}$$

We still have the 'synaptic' rows of the Jacobian to worry about, but these are much simpler. The locality assumption implies that there are *at most* three elements per row. These elements are the derivatives of g_{ij} with respect to x_i and x_j and the derivative of g_{ij} with respect to w_{ij}. Thus, if

$$\forall i, j, t \;\; |\frac{\partial g_{ij}}{\partial x_i}| + |\frac{\partial g_{ij}}{\partial x_j}| + \frac{\partial g_{ij}}{\partial w_{ij}} < 0 \tag{2.37}$$

the bottom two Jacobian blocks satisfy the matrix measure condition. Intuitively, this inequality tells us that the 'self-stabilization' of the synapses must be stronger than the change in plasticity due to neural activity. For example, if one imagines that g_{ij} contains a constant leak term of the form $-\gamma w_{ij}$ where $\gamma > 0$, then the inequality above is a sufficient condition on how large γ should be.

In summary, we have shown that if the synapses are contracting (i.e (2.37) is satisfied) then (2.36) tells us that it is possible to assure contraction of the overall neural and synaptic network by making the network sufficiently sparse.

2.E Contraction in RNNs with Static Weights

In this section we'll analyze RNNs of the general form

$$\dot{x}_i = h_i(x_i) + \sum_{i=1}^{n} W_{ij}\phi(g_i x_i) + u_i(t) \tag{2.38}$$

Where h_i is a leak-term (see (2.23) for definition), ϕ is a nonlinear function with a linear 'regime' (such as tanh(x)), g_i is a neuron-specific gain-term that satisfies:

$$\forall i,\ \ 0 < g_{min} \leq g_i \leq g_{max}$$

and $u_i(t)$ is a time-varying input into neuron i. For simplicity, we focus here on the regime where

$$\phi(g_i x_i) \approx g_i x_i$$

In this case we may rewrite (2.38) in matrix form as:

$$\dot{\mathbf{x}} = \mathbf{h}(\mathbf{x}) + \mathbf{WGx} + \mathbf{u}(t) \tag{2.39}$$

where $\mathbf{G}$ is a constant diagonal, positive definite matrix such that $G_{ii} = g_i$. The Jacobian of (2.38) is a state-dependent matrix:

$$\mathbf{J} = \mathbf{L} + \mathbf{WG} \tag{2.40}$$

where $\mathbf{L}$ is a diagonal matrix such that $L_{ii} = \frac{\partial h_i}{\partial x_i} \leq -\beta$. Consider the following metric transformation to yield a generalized Jacobian, $\mathbf{F}$:

$$\mathbf{\Theta} = \mathbf{G}^{1/2}\mathbf{D} \tag{2.41}$$

Where $\mathbf{D}$ is an arbitrary, non-singular diagonal matrix. Applying this metric to (2.40) we find that

$$\begin{aligned} \mathbf{F} &= \mathbf{\Theta J \Theta}^{-1} \\ &= \mathbf{\Theta L \Theta}^{-1} + \mathbf{\Theta WG \Theta}^{-1} \\ &= \mathbf{L} + \mathbf{G}^{1/2}[\mathbf{DWD}^{-1}]\mathbf{G}^{1/2} \\ &= \mathbf{G}^{1/2}[\mathbf{G}^{-1}\mathbf{L} + \mathbf{DWD}^{-1}]\mathbf{G}^{1/2} \end{aligned} \tag{2.42}$$

Which implies that a sufficient condition for contraction in the RNN described by (2.39) is that

$$\mu_{D,2}(\mathbf{W}) + \frac{\beta}{g_{max}} \leq -c, \quad c > 0 \tag{2.43}$$

If (2.43) is true, then the network described by (2.39) is contracting with rate of $\lambda_F = c\, g_{min}$.

Contraction in Circuits with Perfect E-I Balance

One practical implication of (2.43) is that we can now exploit feedback structure present in $\mathbf{W}$ in a way that would have been missed by using the spectral norm. As an example, consider a network with weight matrix

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{EE} & \mathbf{W}_{EI} \\ \mathbf{W}_{IE} & \mathbf{W}_{II} \end{bmatrix} \tag{2.44}$$

Assume that $\mathbf{W}_{IE}$ is a matrix containing all positive entries and $\mathbf{W}_{EI}$ is a matrix containing all negative entries. Further assume that:

$$-k\mathbf{W}_{IE}^T = \mathbf{W}_{EI} \tag{2.45}$$

where $k > 0$. In other words, (2.45) asserts that this circuit has perfect E-I balance, up to a constant scalar factor. In this case, the weight matrix becomes:

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{EE} & -k\mathbf{W}_{IE}^T \\ \mathbf{W}_{IE} & \mathbf{W}_{II} \end{bmatrix}$$

This corresponds to two interacting neural populations, one inhibitory and one excitatory. The excitatory population excites the inhibitory one through the $\mathbf{W}_{IE}$ submatrix and the inhibitory population inhibits the excitatory population through the $-k\mathbf{W}_{IE}^T$ submatrix. Denote the (unweighted) matrix measures of the EE and II subnetworks μ_{EE} and μ_{II}, respectively:

$$\mu_2(\mathbf{W}_{EE}) = \mu_{EE} \text{ and } \mu_2(\mathbf{W}_{II}) = \mu_{II} \tag{2.46}$$

It's straightforward to show that by picking $\mathbf{D}$ as:

$$\mathbf{D} = \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \sqrt{k}\mathbf{I} \end{bmatrix}$$

we get:

$$\begin{aligned} \mu_{D,2}(\mathbf{W}) &= \frac{1}{2}\lambda_{\max}([\mathbf{D}\mathbf{W}\mathbf{D}^{-1} + \mathbf{D}^{-1}\mathbf{W}^T\mathbf{D}]) \\ &= \frac{1}{2}\lambda_{\max}(\begin{bmatrix} \mathbf{W}_{EE} + \mathbf{W}_{EE}^T & 0 \\ 0 & \mathbf{W}_{II} + \mathbf{W}_{II}^T \end{bmatrix}) \\ &= \max[\mu_2(\mathbf{W}_{EE}), \mu_2(\mathbf{W}_{II})] \end{aligned} \tag{2.47}$$

which implies that

$$\mu_{D,2}(\mathbf{W}) = \max[\mu_{EE}, \mu_{II}]$$

This shows that by appropriately scaling μ_{EE} and μ_{II} to satisfy (2.43), the RNN can be made contracting *independent* of the strength of individual elements in the diagonal blocks.

Contraction in Circuits With Statistical E-I Balance

In a biological system, perfect E-I balance (i.e (2.45)) cannot reasonably be expected. However, some basic results from matrix algebra and random matrix theory can tell us that contraction can be assured even in the case where E-I balance occurs in an average sense.

In particular, we'll need the following result from matrix algebra. Consider a symmetric block matrix of the form:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix} \tag{2.48}$$

where $\mathbf{A}$ and $\mathbf{C}$ are negative definite and symmetric, with $\lambda_{\max}(\mathbf{A}) = -\lambda_A < 0$ and $\lambda_{\max}(\mathbf{C}) = -\lambda_C < 0$. Then a standard result from matrix algebra Horn et al. [1990] says that a sufficient condition for negative-definiteness of (2.48) is that:

$$||\mathbf{B}||_2^2 < \lambda_A \lambda_C \tag{2.49}$$

We'll also need the following result from random matrix theory. Consider an $N \times N$ symmetric matrix, $\mathbf{R}$, whose elements above and including the diagonal are distributed according to a Gaussian distribution:

$$R_{ij} = \mathcal{N}(0, \sigma^2) \text{ for } j \geq i$$

Then for large N, we have from Füredi and Komlós [1981] that, in expectation:

$$||\mathbf{R}||_2 \approx 2\sigma\sqrt{N} \tag{2.50}$$

We can use (2.49) together with (2.50) for how much deviation from perfect E-I balance a circuit of size N can tolerate while preserving contraction. In particular, consider again the matrix defined in (2.44) and assume that (2.43) is satisfied for each diagonal block. That is, assume:

$$\mu_{EE} + \beta g_{max}^{-1} \leq -c \text{ and } \mu_{EE} + \beta g_{max}^{-1} \leq -c \text{ with } c > 0$$

and assume $N_E = N_I = \frac{N}{2}$ for simplicity. We define $\mathbf{R}$ as follows:

$$\mathbf{R} \equiv \frac{\mathbf{W}_{EI} + \mathbf{W}_{IE}^T}{2}$$

from (2.49), we can assure contraction in regions where:

$$||\mathbf{R}||_2 < c$$

To find this region, we will upper bound the spectral norm of $\mathbf{R}$ as follows: instead of the case where E-I and I-E connections cancel out exactly, consider the case when instead they are each drawn from a distribution whose *means* cancel out exactly. Namely:

$$W_{IE}^{ij} \sim \mathcal{N}(m, \sigma_{IE}^2) \text{ and } W_{EI}^{ij} \sim \mathcal{N}(-m, \sigma_{EI}^2)$$

This implies that:

$$R_{ij} = \frac{W_{IE}^{ij} + W_{EI}^{ji}}{2} \sim \mathcal{N}(0, \frac{\sigma_{IE}^2 + \sigma_{EI}^2}{4})$$

Thus, the matrix $\mathbf{R}$ is a symmetric, Gaussian matrix with zero mean and variance $\frac{\sigma_{IE}^2+\sigma_{EI}^2}{4}$. By plugging this expression into (2.50) we can conclude contraction when the following relationship holds between $\mathbf{R}$ and the eigenvalues of the E-E and I-I connection matrices:

$$\sigma_{IE}^2 + \sigma_{EI}^2 < \frac{c^2}{N}$$

Of course, in the limit as the spreads go to zero, we arrive back at the case of perfect E-I balance with $k = 1$, which always yields contraction.

Another way to get a sense of the tolerance' for imperfect E-I balance is to use the Frobenius norm of $\mathbf{R}$, which upper-bounds the spectral norm. Assume that for each connection between excitatory neuron i and inhibitory neuron j, we have that

$$|W_{EI}^{ij} + W_{IE}^{ji}|^2 \leq \kappa^2 \text{ with } \kappa > 0$$

This straightforwardly implies that the Frobenius norm of $\mathbf{R}$ is upper bounded by:

$$||\mathbf{R}||_F \leq \kappa N$$

Which means that it is possible to ensure contraction by scaling κ with N, since:

$$||\mathbf{R}||_2 \leq ||\mathbf{R}||_F \leq \kappa N < c$$

So by scaling κ, which measures how imperfectly each E-I synapse is balanced by the corresponding I-E synapse, we can guarantee contraction.

2.E.1 Relationship to Echo State Networks

Echo state networks (ESNs) are contracting, nonlinear, discrete-time RNNs with static synaptic weights. While attempts have been made to apply the Echo-State framework to continuous time systems, these attempts have relied on discretizations of the underlying continuous dynamics. What determines if a network is an ESN is the 'Echo-State Property' (ESP), which is typically in the form of a bound on the spectral norm of the weight matrix. Unlike our result above (see 2.43) this does not allow one to factor in stabilizing influences within the weight matrix, such as E-I balance. Nevertheless, contraction analysis can be applied to discrete time systems. For completeness, we will now do this to and derive a slight generalization of a known sufficient condition for the ESP in networks where the activation function is more flexible than $\tanh(x)$.

We will need the following result. Consider a discrete-time dynamical system:

$$\mathbf{x}_{n+1} = \mathbf{T}(\mathbf{x}_n, \mathbf{u}_n)$$

It was shown in Lohmiller and Slotine [1998] that this system will be contracting if there exists some non-singular matrix $\mathbf{\Theta}_n$ such that

$$\sigma_{\max}(\mathbf{\Theta}_{n+1}\frac{\partial \mathbf{T}}{\partial \mathbf{x}}\mathbf{\Theta}_n^{-1}) < 1$$

Theorem 2. *Consider the discrete time RNN*

$$\mathbf{x}_{n+1} = \mathbf{W}\mathbf{r}(\mathbf{x}_n + \mathbf{u}_n) \tag{2.51}$$

where $r_i = r_j = r$ is a nonlinearity (e.g. tanh(x)) that satisfies

$$0 \leq \frac{\partial r}{\partial x}(x) \leq g \tag{2.52}$$

If the matrix $\tilde{\mathbf{W}} \equiv \sqrt{g}\mathbf{W}$ is Schur diagonally stable, then the RNN is contracting, and therefore has the ESP. That is, if there exists a postive definite, diagonal matrix $\mathbf{P}$ such that:

$$\tilde{\mathbf{W}}^T\mathbf{P}\tilde{\mathbf{W}} - \mathbf{P} < 0 \ \textit{with} \ \tilde{\mathbf{W}} \equiv g\mathbf{W}$$

the RNN is contracting.

Proof. For the RNN defined in the theorem, we have

$$\sigma_{\max}(\mathbf{\Theta}_{n+1}\mathbf{W}\mathbf{\Phi}\mathbf{\Theta}_n^{-1}) < 1$$

Let $\Theta_n = \mathbf{P}^{1/2}$, where $\mathbf{P}$ is diagonal and positive definite. Using the sub-multiplicitivity of the spectral norm, we find that a sufficient condition for contraction is:

$$\sigma_{\max}(\mathbf{P}^{1/2}\mathbf{W}\mathbf{P}^{-1/2}) < g^{-1}$$

we can easily rewrite this condition as

$$\mathbf{W}^T\mathbf{P}\mathbf{W} - \mathbf{P}g^{-2} < 0$$

Multiplying both sides by g^2, we find that the RNN is contracting if

$$g^2\mathbf{W}^T\mathbf{P}\mathbf{W} - \mathbf{P} < 0 \iff \tilde{\mathbf{W}}^T\mathbf{P}\tilde{\mathbf{W}} - \mathbf{P} < 0$$

Which can be readily recognized as the statement that the matrix $\tilde{\mathbf{W}} = g\mathbf{W}$ is Schur diagonally stable. □

Remark 1. *The above stability criterion immediatly applies to RNNs of the form:*

$$\mathbf{x}_{n+1} = \mathbf{r}(\mathbf{W}\mathbf{x}_n + \mathbf{u}_n)$$

for nonsingular $\mathbf{W}$*. This can be seen using the following coordinate transform and applying it to* (2.51)*:*

$$\mathbf{y} = \mathbf{W}^{-1}\mathbf{x}$$

$$\mathbf{y}_{n+1} = \mathbf{W}^{-1}\mathbf{x}_{n+1} = \mathbf{W}^{-1}[\mathbf{W}\mathbf{r}(\mathbf{x}_n + \mathbf{u}_n)] = \mathbf{r}(\mathbf{W}\mathbf{y}_n + \mathbf{u}_n)$$

Since $\delta\mathbf{y} = \mathbf{W}^{-1}\delta\mathbf{x}$*, this means that* $\delta\mathbf{x} \to 0 \iff \delta\mathbf{y} \to 0$*,*

Remark 2. *For the specific case of* $r(x) = \tanh(x)$*,and therefore* $g = 1$*, this result already appears in Yildiz et al. [2012].*

Remark 3. *A common RNN uses the discontinuous activation function:*

$$\mathbf{r}(x_i) = \max(0, x_i)$$

This does not pose a problem for the above stability condition, for the following reason. Define a diagonal, 'switching' matrix $\mathbf{S}$ *in the following way:*

$$S_{ii} = \begin{cases} 1 & \text{if } x_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

And note that the dynamics of the ReLu network can be written:

$$\mathbf{x}_{n+1} = \mathbf{W}\mathbf{S}\mathbf{x}_n + \mathbf{u}_n$$

The Jacobian of this system is:

$$\mathbf{J} = \mathbf{W}\mathbf{S}$$

Since $\mathbf{S}$ *is diagonal and* $||\mathbf{S}|| \leq 1$*, we can use the same diagonal metric transfor-*

mation as above to find that a sufficient condition for a ReLu RNN to be contracting is that $\mathbf{W}$ *is Schur diagonally stable. In other words, if there exists some positive and diagonal* $\mathbf{P}$ *such that*

$$\mathbf{W}^T\mathbf{P}\mathbf{W} - \mathbf{P} < 0$$

the ReLu network is contracting.

Chapter 3

Robust and Brain-Like Working Memory Through Short-Term Synaptic Plasticity

3.1 Introduction

Working memory (WM), the holding of information "online" and available for processing, is central to higher cognitive functions. A well-established neural correlate of WM is spiking over a memory delay [Funahashi et al., 1989, Goldman-Rakic, 1995, Fuster and Alexander, 1971]. For many years, this was thought to be the sole mechanism underlying WM maintenance. The idea is that sensory inputs elicit unique patterns of spiking that are sustained via recurrent connections [Amit and Brunel, 1997], creating attractor states—stable patterns of activity that retain the WM [Hopfield, 1982]. It seems evident that these attractor dynamics play an important role in WM. Recent observations, however, have suggested that there may be more going on. A few neurons seem to show spiking that looks persistent enough to be an attractor state, but the bulk of neurons show memory delay spiking that is sparse. This is especially true when spiking is examined in real time (i.e., on single trials) because averaging across trials can create the appearance of persistence even when the underlying activity is

quite sparse.

This all begs the question of how WMs are maintained over these gaps in time with little-to-no spiking [Lundqvist et al., 2018]. One possibility was suggested by observations of short-term synaptic plasticity (STSP), transient (< 1 second) changes in synaptic weights induced by spiking, in circuits in the prefrontal cortex [Wong and Wang, 2006]. Several groups have suggested updating the attractor dynamics model with this feature [Lundqvist et al., 2011]. The idea is that STSP helps the spiking. Spikes induce a transient "impression" in the synaptic weights that can maintain the network state between spikes [Knoblauch et al., 2010]. Evidence for STSP comes from techniques like patch-clamp recording that are difficult to implement in the working brain, especially in NHPs. Thus, we tested the role of STSP in WM by using computational modeling in conjunction with "ground-truthing" via analysis of spiking recorded from the PFC of a NHP performing a WM task. We found that PFC spiking carried little-to-no stimulus-specific WM information across the delay. We aimed to determine if network models with STSP can solve the working memory task, whether they have properties similar to those seen in the actual brain, and whether STSP endows functional advantages. The answer to these questions was "yes".

We trained Recurrent Neural Networks (RNNs) with and without STSP to test how it affects network performance and function. We focused on the key property of robustness [Stokes, 2015]. Working memories must be maintained in the face of distractions. Networks need to deal with noise and show graceful degradation (i.e., continue to function when portions of the network are damaged). Our analysis showed RNNs with and without STSP were robust against distractors. However, only the RNNs with STSP were "brain-like"—their activity more closely resembled activity recorded from the prefrontal cortex of a NHP performing a WM task. RNNs with STSP were also more robust against synaptic ablation. Thus, STSP offers functional advantages and explains how WM can be maintained between stimulus presentations.

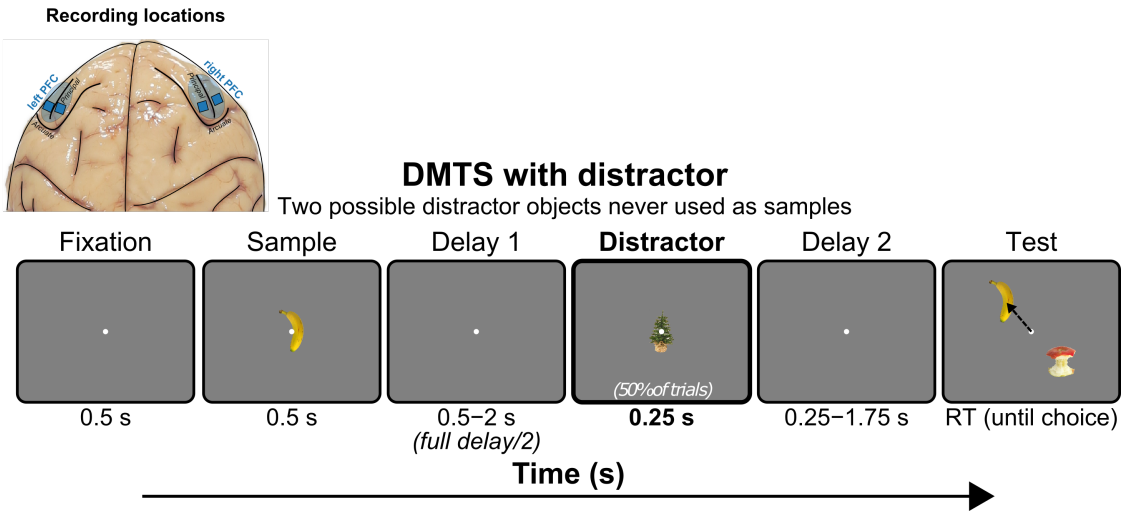

Figure 3.1.1: Electrode location and task structure. Utah arrays were implanted bilaterally in dorsolateral PFC (dlPFC) and ventrolateral PFC (vlPFC). Animal performed a distracted delayed match-to-sample task. Each trial began with visual fixation on the middle of the screen for 0.5s. Fixation was maintained throughout the trial until the behavioral response. The delay length was parametrically varied from 1 – 4 s in five logarithmic steps, randomly chosen each trial. At mid-delay a neutral distractor (1 of 2 possible objects never used as samples) was presented randomly on 50% of trials. During the multi-choice test the NHP was allowed to freely saccade between all objects on the screen. The final choice was indicated by fixating on it for at least one second.

3.2 Results

A NHP was trained to perform an object delayed-match-to-sample task (Figure 1). The NHP was shown a sample object and had to choose its match after a variable-length memory delay. At mid-delay a distractor object (1 of 2 possible objects never used as samples) was presented (for 0.25s) on 50% randomly chosen trials. We recorded multi-unit activity (MUA) bilaterally in dorsolateral PFC (dlPFC) and ventrolateral PFC (vlPFC) using four 64-electrode Utah arrays for a total of 256 electrodes. The animal learned to do the task consistently at 99% accuracy for both distractor and non-distractor trials.

3.2.1 Sample Information in Population Neural Activity Was Weak Over Longer Delays

First, we examined MUA recorded from the lateral PFC. To quantify the amount of sample object information carried by spiking, we used a linear classifier (see Methods for details)1. This showed that spiking carried sample object information for about one second after the sample disappeared. From the start of the delay period the decoder accuracy decreased steadily towards chance (Figure 2a, Figure S 1). We corroborated this by measuring the distance between neural population activity for all pairs of sample objects. That gave an average distance between experimental conditions at every timepoint (see Methods). This showed that the distance between population MUA activity for different samples returned to pre-stimulus levels (Figure 2b, Figure S 2). Interestingly, we found that this was not simply due to spiking returning to pre-sample values. We determined this by training a classifier to discriminate between pre and post sample spiking activity. We found that this classifier was consistently able to discriminate between pre and post sample spiking activity over the delay (Figure S 3). Population Neural Activity Was Robust to Distractors We found that when the mid-delay distractor was presented, neural trajectories diverged from that of non-distractor trials (Figure 2c, Figure S 4). Once the distractor disappeared, the trajectories quickly reconvened, indicating neural stability. This was true for all five delay lengths used (1 – 4 s in five logarithmic steps, Figure S 4). The time course of trajectory reconvening was roughly exponential with a time constant of 200 milliseconds (Figure 2c, Figure S 4). We determined this by fitting an exponential function to the recovery curves and measuring the inverse of the fitted decay constants. This is consistent with prior observations that time constants in cortex peak at this value in the PFC 28.

This all raises a couple of questions. 1. How can PFC networks support WM task performance when, across the population, sample information in spiking is relatively weak? 2. How do PFC networks achieve the stability to recover from distractors? To answer these questions, we used RNN modeling and neural network theory. As we

will show, the two questions share a common answer: STSP.

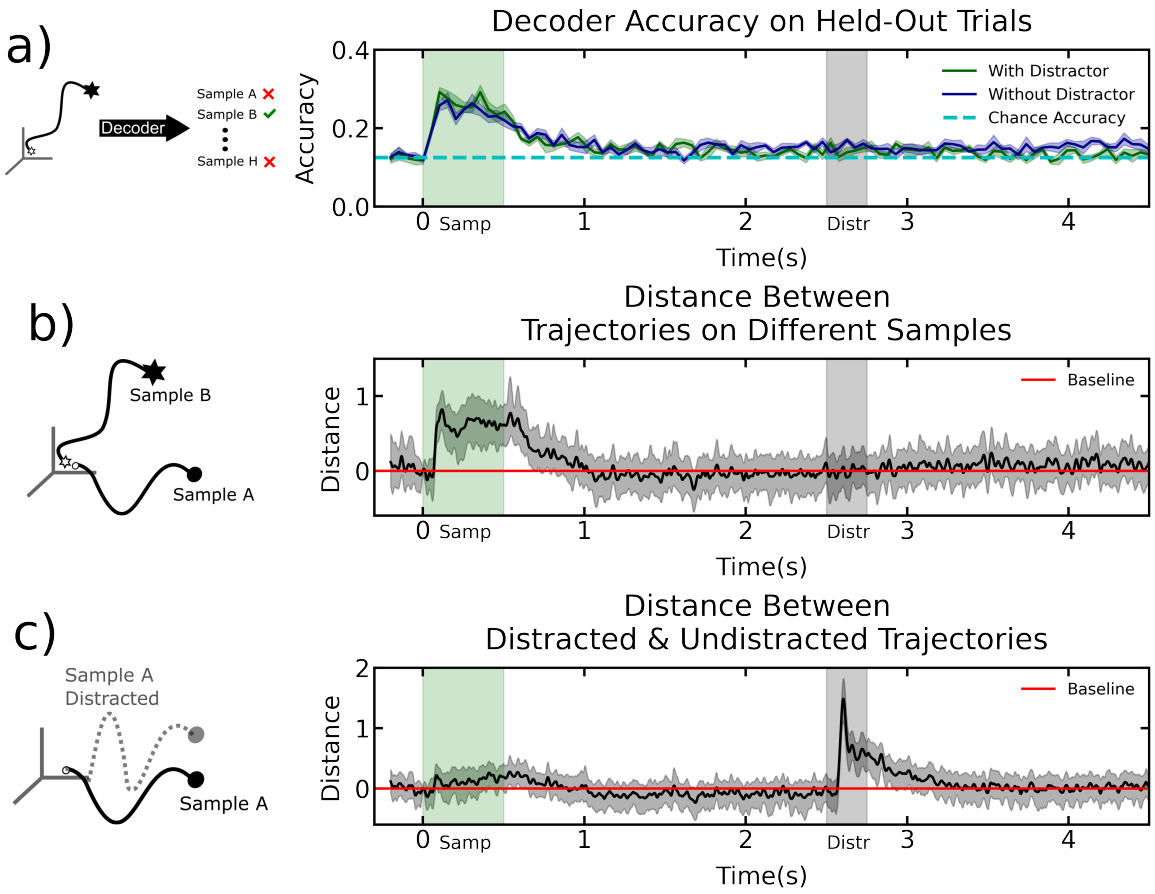

Figure 3.2.1: a) Left: training a decoder to predict sample identity given a neural trajectory. Right: The decoder accuracy on held-out trials for distracted vs. undistracted trials b) Left: comparing trial-averaged trajectories corresponding to different samples. Right: The average pairwise distance in state-space between trajectories elicited by all possible sample images. Normalized by the average pre-stim distance. c) Left: comparing trial-averaged distracted vs. non-distracted trajectories through neural state space. Right: The distance in state space between distracted vs. non-distracted trajectories throughout the trial. Shown are trials with a delay of four seconds.

3.2.2 RNNs With STSP are More Brain-Like

RNNs with and without STSP were able to successfully perform the object delayed match to sample task. However, only the RNNs with STSP did so with activity that was similar to that seen in the actual PFC. Our main hypothesis space consisted of four different kinds of RNNs: two with fixed synaptic weights (fixed after training)

and two with STSP. The two fixed-weight networks were 'vanilla' RNNs. They differed only in their activation functions. One used a hyperbolic tan (tanh), the other used a rectified linear (ReLU). We will refer to these fixed synapse networks as FS-tanh and FS-relu respectively. We choose these two activation functions because they represent two sides of a spectrum. Tanh units can become unresponsive for very large inputs (i.e saturate) while ReLU units cannot. Both activations are commonly used throughout computational neuroscience and machine learning [Dayan and Abbott, 2005, Maheswaranathan et al., 2019]. They lead RNNs to prefer one strategy over another for performing a task. ReLU units are ideal for forming line attractors, while tanh units are ideal for forming point attractors [Maheswaranathan et al., 2019]. Additionally, we also trained Long-Short-Term-Memory networks31 (LSTMs) and Gated Recurrent Unit networks [Cho et al., 2014] (GRUs). These two networks are popular in machine learning and are well-known for their excellent performance capabilities. However, they are not biologically-plausible models of the brain[O'Reilly and Frank, 2006], and thus serve as a useful tool for dissociating our measures of brain-similarity and robustness.

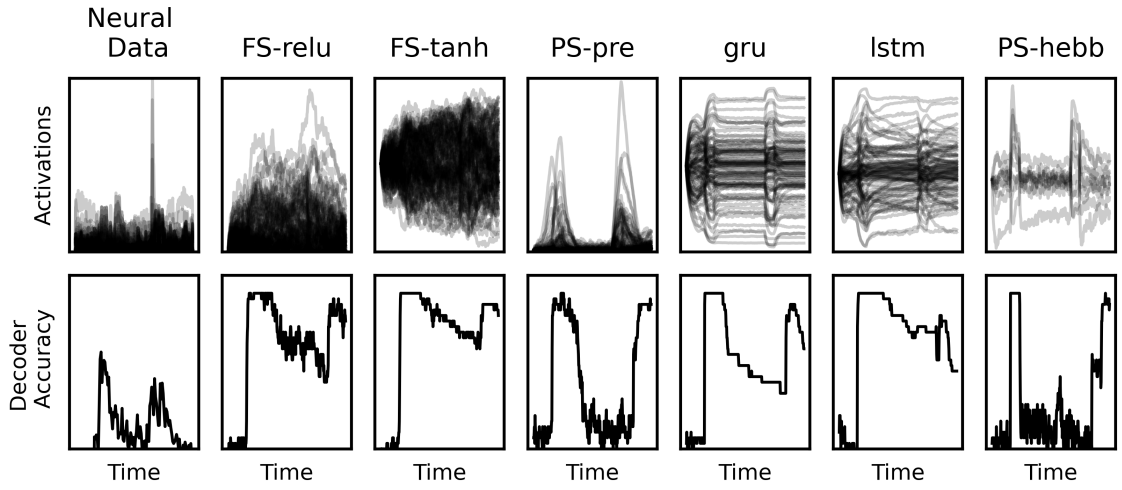

Figure 3.2.2: Example neural activations and their corresponding decoder curves. Top row: neural activity corresponding to a single trial condition. The leftmost panel is the actual neural data. The other panels are artificial neural networks, whose details are described in the main text. Bottom row: decoder accuracy curves corresponding to the neural activations in the top row.

The synaptic weights for the fixed-synapse networks are only adjusted during training. After training, they are untouched. The other two models use STSP to adjust synaptic weights during each trial. This represents the distinction between long-term and short-term memory. Long-term memory is acquired over the course of task-optimization and remains fixed throughout the trial. Short-term memory changes during the trial and reflects the contents of working memory. We reset the state of the plastic synapses at the beginning of each trial.

One RNN with STSP was based on a model introduced by Mongillo et al19. The model uses a set of simplified equations for synaptic calcium dynamics to endow the RNN with STSP. The Mongillo model adjusts synaptic weights based on presynaptic neural activity. For this reason we will call it PS-pre. The other RNN with STSP was introduced by Kozachkov et al. [2020] and adjusts synaptic weights in a way that depends on both the pre and post synaptic neural activity. For this reason, we will call it PS-hebb. It used excitatory anti-Hebbian and inhibitory Hebbian mechanisms to stabilize the RNN34.

All models were trained with backpropagation-through-time using a standard deep learning library35. While the FS models were trained without any explicit constraints, the PS models needed to be parameterized to ensure they satisfy certain properties throughout training. In particular, the PS-pre model had separate excitatory and inhibitory neurons. The weight matrix therefore needed to be parameterized so that these populations always remained separate (see Methods). Likewise, for PS-hebb the weights were parameterized so the network remained stable throughout training. To ensure that our results did not depend on the particular choice of training hyperparameters, we trained the six models under a wide range of hyperparameter settings. These hyperparameters were: the degree of activity regularization, degree of parameter regularization during training, and the size of the network (called hidden size, referring to the number of hidden units). Roughly 2000 models were trained in total. We refer the reader to the Methods section for more details on the models and the training process. As in the analysis of the experimental data, we used a decoder to read out sample object information over time. We did this for both the

fixed-synapse and plastic-synapse models. For the fixed-synapse RNNs we trained the decoder on trajectories from the spike rates. For the STSP RNNs, we trained two separate decoders: one on the spike-rates and one on the synaptic weights. In the LSTM and GRU models, there are no variables which can clearly be denoted as "STSP weights" (since they are not biological models), so we exclude them from this analysis.

In all models, sample presentation elicited a large increase in decoding accuracy (Figure 3 and Figure 4). Spike rate decoding of sample information in the fixed-synapse models remained high throughout the memory delay. This is in sharp contrast to the actual PFC MUA data. The PFC MUA showed a large drop in sample object decoding accuracy once the sample disappeared and especially for delays longer than one second. This was mirrored in the STSP models (Figure 3 and Figure 4). However, sample decoding accuracy on the synaptic weights for the STSP models remained high over the entire delay, including longer delays (Figure 4).

This raises the question: do STSP models better capture the dynamics of the real PFC? We addressed this question by comparing the similarity of each of the 2000 trained models to the actual neural data. To quantify the similarity between model and brain, we used the Pearson correlation between the decoder accuracy curves for the real neural data and the RNNs (Figure 3, bottom row). We did this for each delay length and averaged the correlation scores to get a final brain-RNN correlation value for each model. We did this analysis with and without the LSTM and GRU networks, to dissociate the potential performance improvements of LSTM/GRU with their brain-similarity. The results without these networks are shown in Figure 5, the results with are shown in Figure S8.

This analysis revealed that the PS-hebb model was the most similar to the brain across a wide range of hyperparameters (Figure 5, left column). Interestingly, for a particular choice of parameter regularization, this analysis also revealed that increasing the amount of activity regularization led to a corresponding increase in brain-similarity for the PS-pre model (Figure 5, top left panel). Within the fixed-synapse model class, the models that were most brain-like across all hyperparameter values showed a

marked dip in neural decoder accuracy during the delay period—as expected. However, this dip was far less pronounced than in the models with plastic synapses. The result was that models without synaptic plasticity had a lower similarity to the actual neural data.

An important difference between measuring information in RNNs and brains is that we subsample neurons in the brain. It is possible that the decoding accuracy curves are affected by this subsampling. To control for this, we repeated the above analysis while subsampling at four different values. We randomly picked 0.01%,1.0%,10.0% and 100.0% of the neurons in each RNN to decode from. The result was that STSP networks were more brain-like and this did not change because of subsampling (Figure S 5).

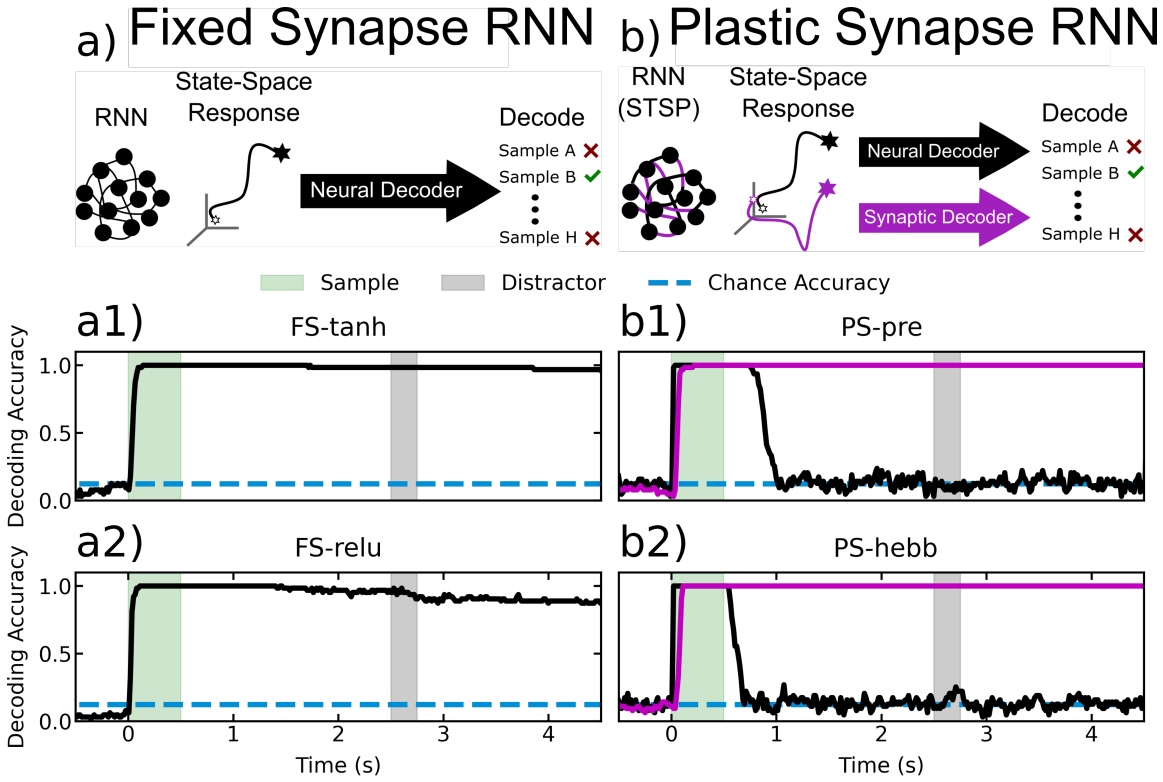

Figure 3.2.3: Decoding accuracy for RNNs. a) A decoder is trained to predict the sample label from RNN neural trajectories, for the fixed-synapse RNNs. a1) The accuracy of the trained decoder as a function of time from sample onset for FS-tanh. a2) The same plot as (a1), for FS-relu. b) Two decoders are trained on the neural and synaptic trajectories separately. Black lines indicate neural decoding, purple indicate synaptic decoding. b1) Neural and synaptic decoding accuracy as a function of time for PS-pre. b2) Same plot as in (b1) but for PS-hebb.

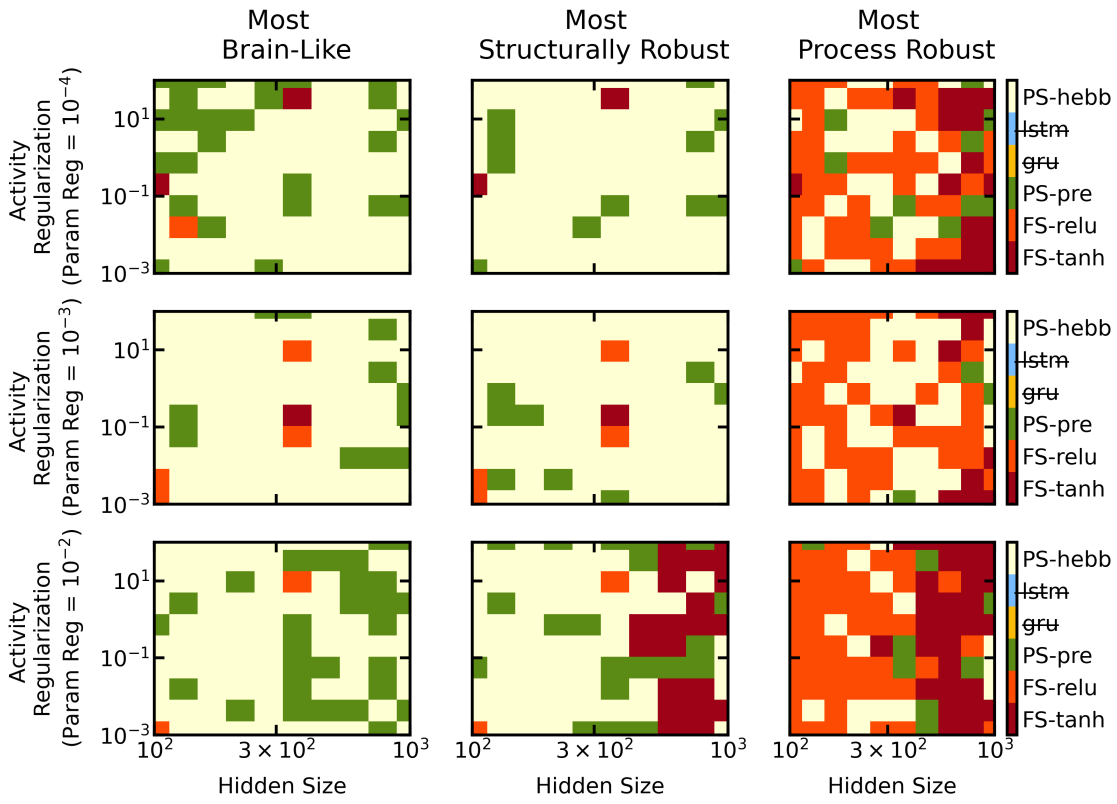

Figure 3.2.4: Results of the hyperparameter sweep across number of hidden neurons, parameter regularization strength, and activity regularization strength. Each row corresponds to a different parameter regularization (1e-4,1e-3,1e-2). Each column corresponds to a different observed quantity (brain-likeness, structural robustness, process robustness). Each subplot is a 10x10 grid, corresponding to 10 possible hidden size / activity regularization configurations. Shown in each square of that grid is the most brain-like/robust network corresponding to that particular hyperparameter configuration. LSTM and GRU networks were excluding (see Fig S8 for corresponding figure when they are included). Each color corresponds to a different network. For the PS-hebb networks, the number of neurons was reduced by a factor of 10 for each point along the x-axis of the above subplots, for reasons explained in the Methods.

3.2.3 STSP Increases Structural Robustness

The two models with STSP were almost always more structurally more robust than the models with fixed synaptic weights (Figure 5, middle column). We examined how the performance accuracy of the trained network varied as we randomly ablated a varying fraction of synaptic weights. We found that the STSP network's performance

on the task remained high even when as many as half of the synapses were ablated. By contrast, the fixed-synapse models were highly sensitive to synaptic ablation. Their performance severely degraded even with ablation of only 10-20% of the synapses.

We repeated this analysis over all the models trained with different hyperparameters. From each model we defined two measurements to quantify robustness. The first was robustness to structural noise. The second was the robustness to synaptic noise. Both measurements were calculated in a similar way. We took the weighted average RNN performance over all the noise values tested. We weighed each term in the sum by the corresponding noise value. The reason for this weighting scheme is simple. If the RNN performance is high when the noise is low, this does not tell us much about robustness. However, if the RNN performance is high when the noise is high, this does indicate robustness. Thus, a natural measure of robustness is the product of noise and performance.

This analysis revealed that the PS-hebb and PS-pre models were more robust to structural perturbations than the fixed synapse networks across a wide range of hyperparameters (Figure 5 and Figure S 8). Interestingly, the fixed synapse networks tended to be more robust to process noise (although always less brain-like). We repeated this analysis with the LSTM and GRU networks. This revealed that, out of the six kinds of networks, the LSTM and GRU networks were the most robust to structural noise and process noise across a wide range of hyperparameters (Figure S 8). However, they were almost always less brain-like than the PS-hebb and PS-pre models. This suggests that brain-similarity does not imply robustness, and vice versa. To better understand how the different trained models achieved WM that was robust to distractors we investigated how these networks organized their activity in state-space.

We found that the fixed-synapse RNNs learned to perform the task by using simple attractors. To visualize these attractors, we projected the high-dimensional RNN activity into a lower dimensional space using Linear Discriminant Analysis (Figure 6). We reasoned that this projection would give us the clearest visualization of the underlying state space attractors. For the plastic-synapse RNNs, we did this for both the neural and synaptic state-space. To better quantify the attractor properties of

these networks, we measured the distances between state-space trajectories of the most brain-like models, as we did for the neural data (Figure 7). This revealed different trajectories for different sample objects. Each trajectory settled into a steady state that was unique for each sample, indicating the presence of attractors (Figure 6, Figure 7). Comparison between trials with and without a distractor showed that the distractor temporarily knocked trajectories out of the attractor state. These trajectories quickly returned to the pre-distractor attractor state. This is how the fixed-synapse RNNs achieve robustness to the distractor. Once the neural trajectories were 'captured' by the appropriate attractor, they remained fixed there and can return there after being perturbed. In terms of neural activity this means that the spike rates stayed elevated across the delay period corresponding to the sample identity. In other words, these models achieve robust WM through persistent neural activity as a consequence of attractor dynamics, as observed in many prior models.

The models with STSP did not achieve robust WM through persistent neural spiking, but instead relied on the STSP. We again examined trajectories using spikes rates from the STSP models (Figure 8). The neural distance between trajectories for different samples increased during sample presentation but then dropped back down near zero during the delay, especially longer delays. This mirrors the results from actual MUA activity in the PFC (see above). This stands in contrast to the fixed-synapse models (Figure 7), in which strong persistent spiking observed. However, unlike the fixed-synapse models, in the STSP models we could measure trajectories and distances in synaptic state-space. This is analogous to using spike rates but instead we measure synaptic weights over time (as was done in Masse36). This revealed that trajectories for different samples in synaptic state-space remained elevated across the delay period (Figure 8), indicating sample information was being maintained by STSP even when spiking levels were low.

As we found in the experimentally recorded MUA, in the STSP models the spike rate trajectories between distractor and non-distractor trials increased during distractor presentation but then quickly decreased back to pre-distraction levels (Figure 8). By contrast, the distractor had a longer lasting effect in synaptic weights for the STSP

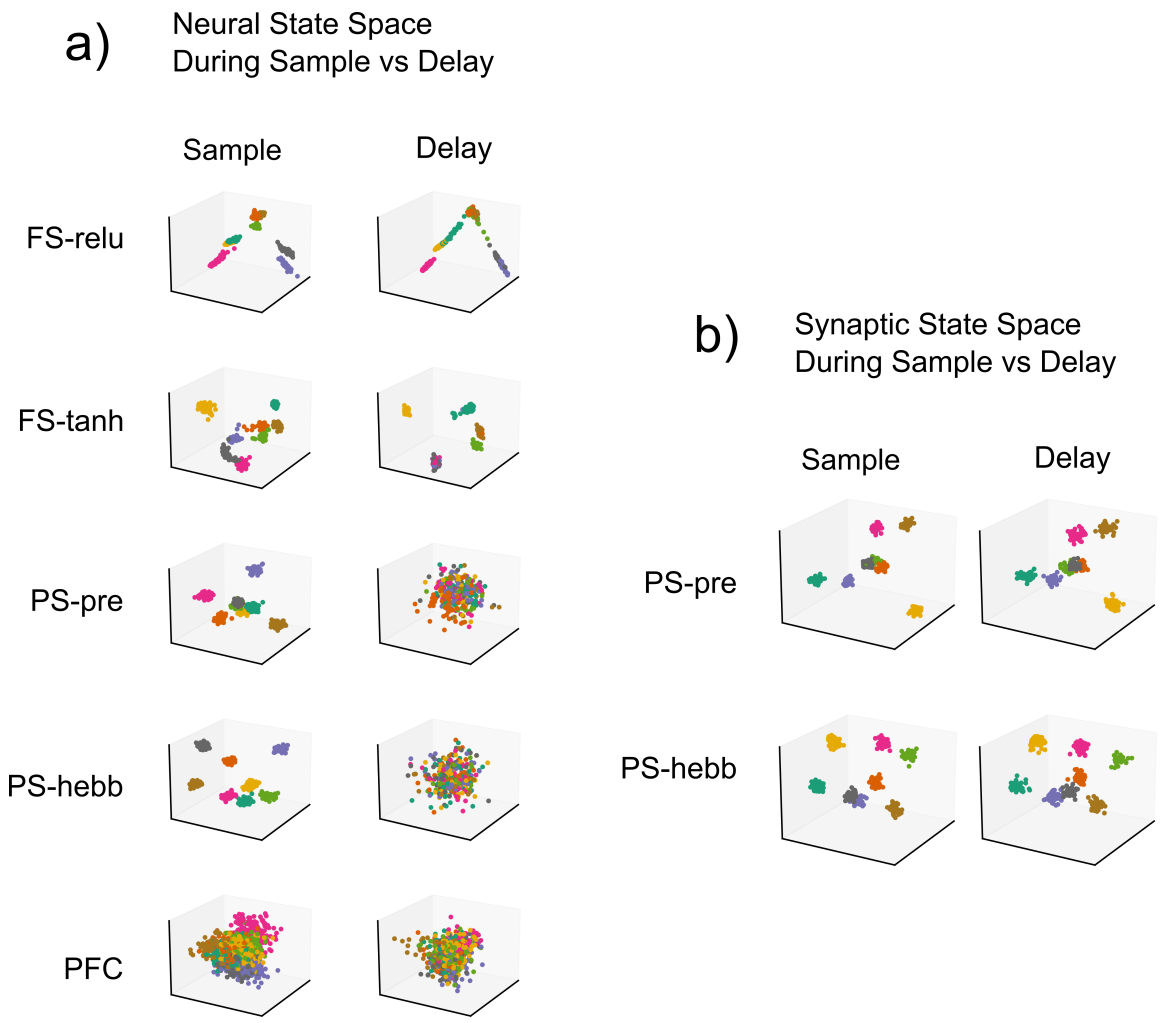

Figure 3.2.5: Dimensionality reduced space plots for the most brain-like models. We time-averaged RNN activity during the sample-period as well as 500ms before the end of the delay period. We used Linear Discriminant Analysis (LDA) to project the data into three-dimensions. To account for differences in training/test splits, we used an Orthogonal Procrustes operation to rotationally align the sample and delay period activity. Colors denote sample IDs. Fixed synapse models have activity organized around simple attractors in state space. There is one attractor for each sample ID. Plastic synapse models exhibit high sample-separability in synaptic state space, and limited separability in the neural state space during the delay period. Similarly, PFC exhibited higher neural discriminability during the sample than during the delay period.

models (Figure 8). These plots show the synaptic distance decreasing on a longer timescale than the neurons. Since the synaptic distance decreased on a timescale several times larger than the intrinsic time-constant of synapses, this indicates that the neurons and synapses interact in a way that increases their effective time constant 29.

This phenomenon is also found in simple linear systems, where a judicious choice of weight matrix (for example a marginally stable weight matrix) can lead to an increased effective time constant 29. This could also potentially increase the susceptibility of the STSP networks to distractions. However, notably, we found that this was not the case. This increased time constant did not affect the ability of the STSP networks to perform the task. The STSP networks performed at a high level (above 90% accuracy) on both distractor and non-distractor trials.

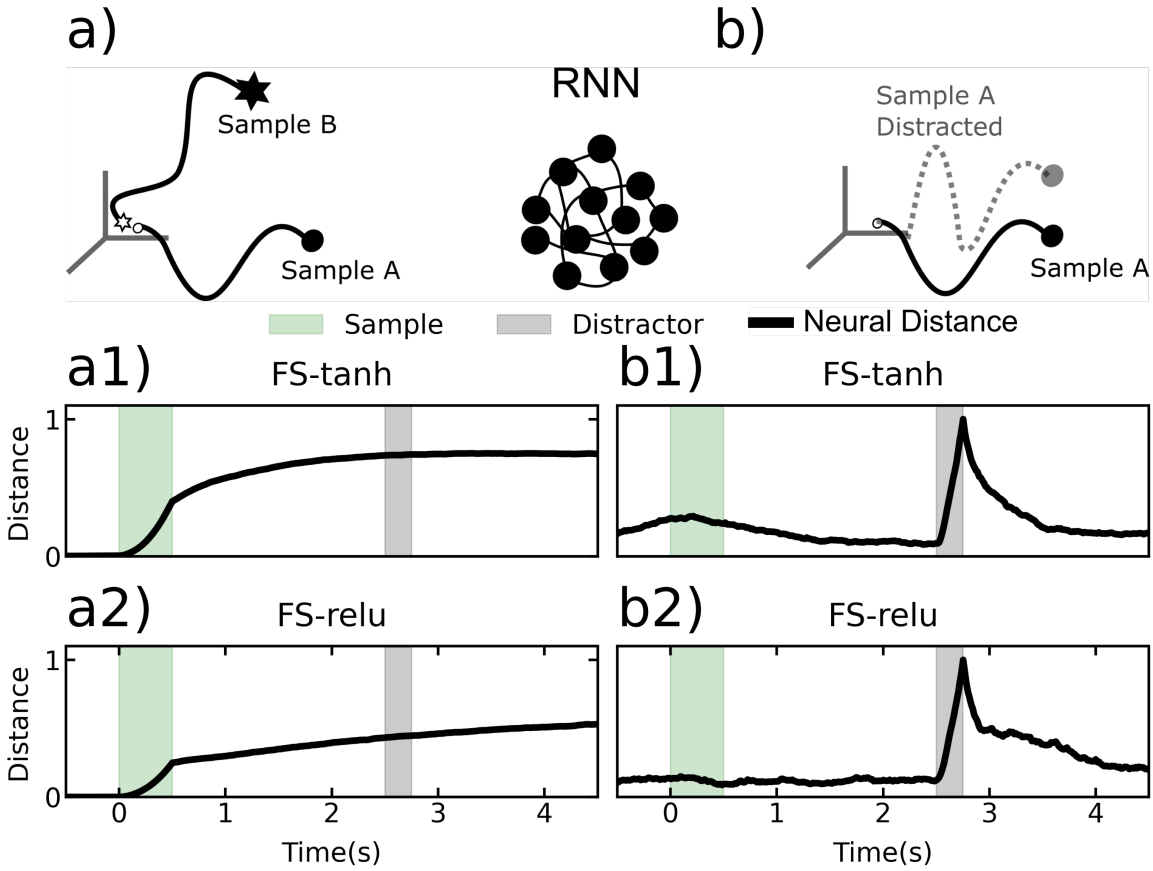

Figure 3.2.6: Distances between neural trajectories within a sample condition and between sample conditions, for fixed synapse RNNs. All models used were the most 'brain-like', as determined by the methodology in section "RNNs With STSP are More Brain-Like". a) Cartoon of trial-averaged RNN trajectories corresponding to two different sample conditions for the fixed-synapse RNNs. a1) Average pairwise distance between trajectories on different sample conditions, for the fixed synapse network with tanh activation (FS-tanh). a2) The same plot as in a1, but for FS-relu. b1) The average distance between distracted and undistracted trajectories. Average taken over all sample conditions. Results shown for FS-tanh. b2) Same results as in b1, but for FS-relu.

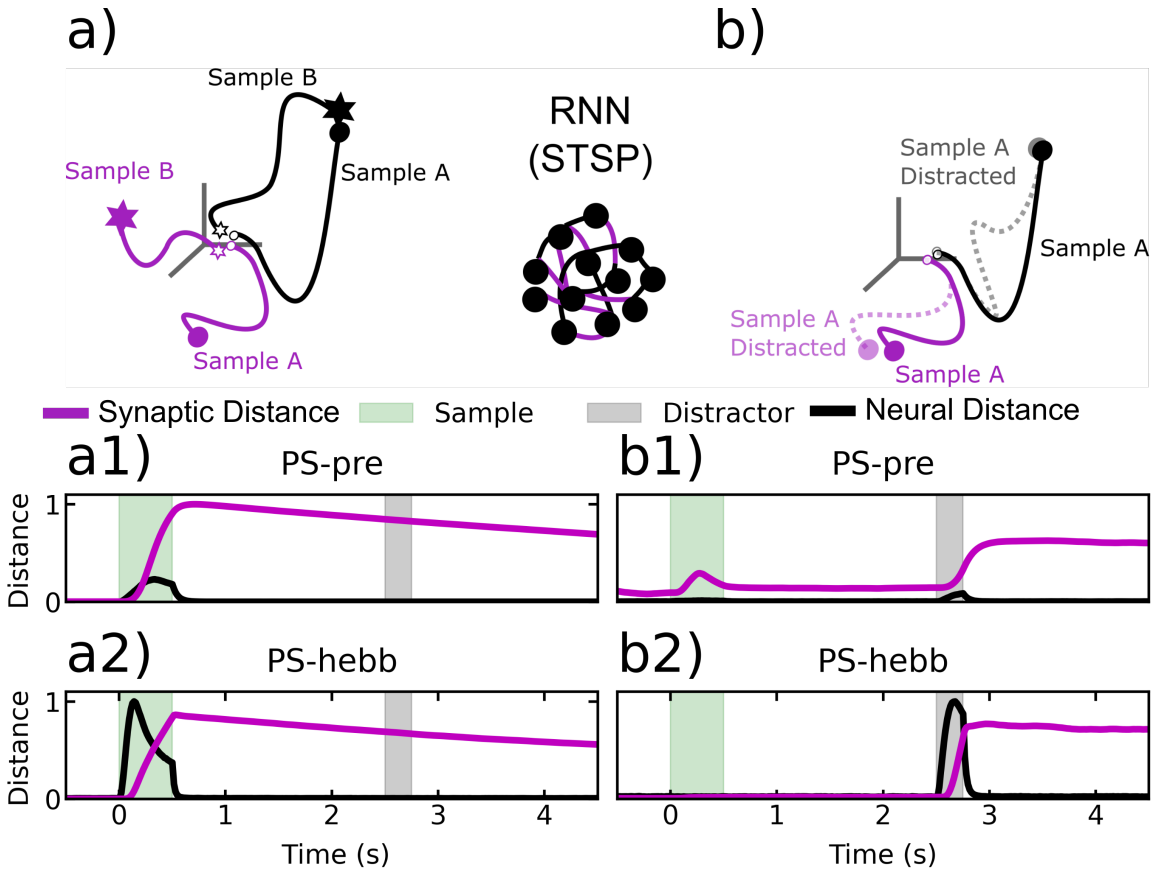

Figure 3.2.7: Distances between neural and synaptic trajectories within a sample condition and between sample conditions, for plastic synapse RNNs. Black lines correspond to neural trajectories, purple lines correspond to synaptic trajectories. a) Cartoon of trial-averaged neural and synaptic RNN trajectories corresponding to two different sample conditions for the plastic-synapse RNNs. a1) Average pairwise distance between neural and synaptic trajectories on different sample conditions, for PS-pre. a2) The same plot as in a1, but for PS-hebb. b1) The average distance between distracted and undistracted trajectories. Average taken over all sample conditions. Results shown for PS-pre. b2) Same results as in b1, but for PS-hebb.

3.3 Discussion

We found that while RNNs with and without STSP showed robustness against distractors, the RNNs with STSP were more brain-like. We also found that RNNs with STSP were more structurally robust than RNNs without STSP. STSP models showed graceful degradation to synaptic loss. In the STSP models, decodability of spike rates decreased during the memory delay (especially longer delays). Importantly, the WMs could be decoded from the synaptic weights. This was in contrast to the RNNs

without STSP, where spike-rate decodability remained high over the entire memory delay. This high spike-rate decodability did not match observations of actual spiking recorded from the PFC. In sum, STSP can not only maintain information over gaps of no spiking, it also adds functional advantages. And, notably, adding STSP to RNNs makes them exhibit brain-like behavior.

Some forms of STSP worked better than others. We found that purely Hebbian STSP models were difficult to train, in agreement with previous studies [Orhan and Ma, 2019]. By contrast, our anti-Hebbian STSP model, PS-hebb, (which reduced synaptic weights when spiking was too correlated) was successful and did not require any external weight clipping. We and others have found that anti-Hebbian STSP plays a crucial role in network stability and function [Tyulmankov et al., 2021]. And like others, we found that the synaptic weights in the PS-pre model had to be limited to a certain range to maintain stability and trainability [Masse et al., 2019]. This assumption is biologically plausible, as biological synapses are limited in the amount of resources that can be recruited by a presynaptic spike [Tsodyks and Markram, 1997].

We found that the LSTM and GRU networks tended to be more robust to structural perturbations as well as process noise than the other networks. However, they were less brain-like than the STSP networks. This suggests that STSP may carry additional functional advantages in the brain beyond robustness, such as energy-efficiency (spikes are metabolically expensive). We also found that the STSP networks were more structurally robustness than the fixed-synapse networks. This suggests that fixed-synapse networks require a "fine-tuning" of their synaptic weights, to appropriately mold their attractor landscape in a way that subserves the demands of the WM task. By contrast, we found that fixed-synapse networks tended to be more robust to process noise than the STSP networks. Given the superior brain-similarity of the STSP networks, this suggests that the brain may optimize for higher structural robustness as opposed to higher robustness to process noise. Indeed, the turnover rate of dendritic synapses in sensory and motor brain areas is as high as 40% every five days.

The STSP models are in line with a variety of studies suggesting that cognition

is more complex than steady attractor states. For example, sustained attention was long-thought to depend on attractor-like steady-state spiking. However, like studies of WM, this may have been an artifact of averaging spiking across trials [Miller et al., 1996]. Examination of sustained attention in real time (on single trials) has shown that, behaviorally, attention waxes and wanes rhythmically at theta. There is a corresponding waxing and waning of spiking synchronized to LFP theta oscillations. Likewise, neural correlates of WM show sparse bursts of spiking linked to oscillatory dynamics when examined in real time [Lundqvist et al., 2016].

Whether WM relies on persistent attractor dynamics (i.e., persistent spiking) alone or sparse spiking combined with STSP is of current interest [Stokes, 2015] . A recent computational study has provided insight. Masse et al trained a WM model with STSP [Masse et al., 2019]. They found that there was sparse spiking when WMs were being simply maintained and more spiking when WMs were being used and manipulated. This is consistent with other STSP-based WM models. For example, Lundqvist et al [Lundqvist et al., 2016], found that spiking increases when WMs are being read out for use. This makes sense because the brain cannot read information from the synapses directly. Spikes are needed to "ping" the network to read information from the synaptic weights. Thus, models with STSP can work for both modes: Sparse spiking when WMs are being maintained and more spiking when WMs are being used.

It is also worth noting that studies that report higher levels of memory-delay spiking tend to be those that used spatial delayed response tasks. Spatial delayed response may involve more motor inhibition than WM per se [Miller et al., 2018]. The animal knows the forthcoming behavioral response and is inhibiting it while waiting for a "go" signal. It is like revving the engine while keeping your foot on the brake. By contrast, in WM tasks like delayed match-to-sample (i.e., the task used in this study), the behavioral response is not known until after the memory delay. They involve holding information for further computation, not inhibiting a behavioral response. Tasks like delayed match-to-sample are well known to produce lower levels of memory delay spiking14,15,36,44 and thus require more than neural attractor states alone. That being said, replication of our findings in other animals and similar tasks is needed

to strengthen our claims. Furthermore, while we limited ourselves to single-circuit WM mechanisms, it will be useful in future work to explore the potential role of other brain areas (such as hippocampus), other forms of synaptic plasticity, other computational mechanisms (such as rotational dynamics [Libby and Buschman, 2021]) and perhaps even different cell types.

The structural robustness and provable stability [Kozachkov et al., 2020] added by PS-hebb, the most brain-like network we trained, is critical not just to WM but to top-down control in general. A control system which is overly sensitive to perturbations (e.g distractors) would quickly propagate these disturbances to the rest of the brain, leading to errant behavior. Moreover, robustness and stability are intimately tied to modularity [Slotine, 2002]. Stable networks can be linked with other stable networks in a way that preserves stability. Without that property, the whole network must be retrained when new modules are added to subserve complex or composite behavior. Our results suggest that STSP has a dual role. It can maintain information while simultaneously ensuring robustness in an energy-efficient manner9. We speculate that stability could be a key component to understanding the way specialized modules in the brain dynamically cooperate to form cohesive perceptions, plans, and actions.

3.4 Methods

3.4.1 Subject and Task

The non-human primate subject used in our experiments was a male rhesus macaque monkey (Macaca mulatta), aged 17. All procedures followed the guidelines of the Massachusetts Institute of Technology Committee on Animal Care and the National Institutes of Health.

As described in the main text, Utah arrays were implanted bilaterally in dorsolateral PFC (dlPFC) and ventrolateral PFC (vlPFC). Animal performed a distracted delayed match-to-sample task. Each trial began with visual fixation on the middle of the screen for 0.5s. Fixation was maintained throughout the trial until the behavioral

response. The delay length was parametrically varied from 1 – 4 s in five logarithmic steps, randomly chosen each trial. At mid-delay a neutral distractor (1 of 2 possible objects never used as samples) was presented randomly on 50

3.4.2 Data Analysis Methods

Distances Between Neural Trajectories

All neural spiking data was first smoothed using a 10ms Gaussian kernel. We refer to the smoothed spiking data as the firing rate. To quantify the difference in population activity between different conditions, we computed the distance between neural trajectories. We define a neural trajectory as a vector of firing rates for N recorded neurons evolving in time. For the distractor vs non-distractor comparison, denote

$$x_t^{s,d,l} \in [0,\infty)^N$$

the neural trajectory at time t on trial $l \in \{1, \ldots, L\}$ for stimulus identity $s \in \{1,\ldots,S\}$ and where $d \in \{0,1\}$ denotes the presence or absence of a distractor in the delay phase. We first computed trial averages as

$$\bar{x}_t^{s,d} = \frac{1}{L}\sum_{l=1}^{L} x_t^{s,d,l}$$

Then for each s we computed the distance between distracted and non-distracted trajectories, using the Euclidean norm, and took the average over stimuli:

$$d_t^{\text{dist}} = \frac{1}{S}\sum_{s=1}^{S} \left|\left|\bar{x}_t^{s,1} - \bar{x}_t^{s,0}\right|\right|^2$$

For the comparison between trajectories of different stimulus identities, we used only non-distractor trials ($d = 0$). We first averaged over trials as above for each stimulus to get

$$\bar{x}_t^{s} = \frac{1}{L}\sum_{l=1}^{L} x_t^{s,0,l}$$

and calculated the mean distance between each pair of stimuli as:

$$d_t^{\text{stim}} = \frac{2}{S(S-1)} \sum_{1 \le i < j \le S} \left|\left| \bar{x}_t^i - \bar{x}_t^j \right|\right|^2$$

Confidence Intervals for Trajectory Distances

As our final measurement we calculated the mean over six experimental sessions for both d_t^{dist} and d_t^{stim}. To quantify uncertainty that captures both the session and trial variability, we used a non-parametric multi-level bootstrap. Briefly, for $b \in 1, \ldots, B$ with $B = 1000$, we first sampled with replacement over sessions (that is, we randomly selected a session with replacement E times, where E is the total number sessions in the dataset - in our case $E = 6$), and then for each resampled session, we resampled trials with replacement in the trial-average steps described above, yielding the bootstrap samples $d_t^{\text{dist},b}$ and $d_t^{\text{stim},b}$. We took the 2.5^{th} and 97.5^{th} percentiles across b as our lower and upper values for the 95% confidence interval.

Sample Decoding

For sample decoding experiments, we first smoothed trials (as above, with 10ms Gaussian kernel), and then took average rates per neuron in 50ms time bins and attempted to decode sample type from the population activity at each time point. For our classifier, we used a linear Support Vector Machine and standard regularization (C=1 in SciKit Learn). For each session we used a 10-fold cross-validation to calculate decoding accuracy. (That is, we held out 10% of trials as a test set, and trained on the remaining 90% of the data. We did this for 10 non-overlapping test sets, and then took the average test accuracy). We performed a decoding analysis on each session separately, and then took the average and standard error across sessions for our result. Change from Baseline Decoding We smoothed trials and took average rates as in the sample decoding above. Then, for a given sample type, we took the population activity from 400-350ms before the stimulus (or distractor) and attempted to decode subsequent population activity against this baseline activity. We performed 10-fold cross-validation as above for calculating test accuracy, and calculated the average

performance over samples for each session. Plots show the average and standard error across sessions.

Sensitivity to Smoothing and Bin Size

Throughout our data analysis we used a 10ms Gaussian kernel to produce firing rates, and in the decoding analysis, we used the mean firing rate in 50ms time bins. To ensure our trajectory distance results did not depend specifically on the choice of Gaussian kernel, we examined trajectory distances using 5, 10, and 15ms Gaussian kernels (Figure S6). For the distance between trajectories with different sample identities, we took the mean distance within the sample presentation period (0, 0.5s) and the 100ms preceding test time (t_{test}- 0.1s, t_{test}), for each delay and each session, and then compared the sample distances and test distances. Similarly, for distances between distracted and non-distracted trajectories, we took the mean distance within the distractor presentation ($t_{dist}, t_{dist} + 0.25s$), and the ($t_{test} - 0.1s, t_{test}$), again for each delay and each session. To ensure our decoding results did not depend on the specific choice of Gaussian kernel and length of time bin, we again used 5, 10, and 15ms Gaussian kernels and additionally used 20ms, 50ms, and 100ms time bins (Figure S7). We then looked at sample decoding accuracy in the non-distracted trials, for each combination of Gaussian kernel and time bin, again taking a mean for the sample presentation (0, 0.5s), and a mean for the pre-test period (t_{test}- 0.1s, t_{test}) for each delay and each session. The Wilcoxon signed-rank test was used for comparisons between time epochs (e.g. Sample vs Pre-Test), with p-value corrections for multiple comparisons made using the Benjamini-Hochberg procedure (with $\alpha = 0.05$).

Chapter 4

RNNs of RNNs: Recursive Construction of Stable Assemblies of Recurrent Neural Networks

The combination and reuse of primitive “modules" has enabled a great deal of progress in computer science, engineering, and biology. Modularity is particularly apparent in the structure of the brain, as different parts are specialized for different functions [Kandel et al., 2000]. Accordingly, most experimental studies throughout the history of neuroscience have focused on a single brain area in association with a single behavior [Abbott and Svoboda, 2020]. Similarly, RNN models of brain function have mostly been limited to a single RNN modeling a single area. However, neuroscience is entering an age where recording from many different brain areas simultaneously during complex behaviors is possible. As experimental neuroscience has shifted towards multi-area recordings, computational techniques for analyzing, modeling, and interpreting these multi-area recordings have blossomed [Mashour et al., 2020, Abbott and Svoboda, 2020, Perich et al., 2021, Semedo et al., 2019, Yang and Molano-Mazón, 2021, Machado et al., 2022]. Despite this, RNN theory has lagged behind.

The theoretical question of RNN stability is crucial for understanding information propagation and manipulation [Vogt et al., 2020, Engelken et al., 2020, Kozachkov et al., 2022a]. The conditions under which single, autonomous RNNs are chaotic or

stable are well-studied, in particular when the RNN weights are randomly chosen and the number of neurons tends to infinity [Sompolinsky et al., 1988, Engelken et al., 2020]. However, there is very little work addressing the theoretical question of stability in 'networks of networks'. Two facts make this question challenging. Firstly, connecting two stable systems does not, in general, lead to a stable overall system. This is true even for linear systems. Secondly, there is a massive amount of feedback between brain areas, so one cannot reasonably assume near-decomposability [Simon, 1962, Abbott and Svoboda, 2020].

Given that the brain seems to dynamically reorganize and adapt interareal connectivity to meet task demands and environmental constraints [Sych et al., 2022], this question of how stability is maintained is of the utmost importance. Here we take a bottom-up approach, more specifically asking "what stability properties of the individual modules lend themselves to rapid reorganization?"

4.0.1 Contraction Analysis

We focus on a special type of stability, known as contractive stability [Lohmiller and Slotine, 1998]. Loosely, a contracting system is a dynamical system that forgets its initial conditions exponentially quickly. Contractive stability is a strong form of dynamical stability which implies many other forms of stability, such as certain types of input-to-state stability [Sontag, 2010]. See Section 4.A.2 for a mathematical primer on contraction analysis.

Contraction analysis has found wide application in nonlinear control theory [Manchester and Slotine, 2017], synchronization [Pham and Slotine, 2007], and robotics [Chung and Slotine, 2009], but has only recently begun to find application in neuroscience and machine learning [Boffi et al., 2020, Wensing and Slotine, 2020, Kozachkov et al., 2020, Revay and Manchester, 2020, Jafarpour et al., 2021, Kozachkov et al., 2022a, Centorrino et al., 2022, Burghi et al., 2022, Kozachkov et al., 2022b]. Contraction analysis is useful for neuroscience because it is directly applicable to systems with external inputs. It also allows for modular stability-preserving *combination* properties to be derived (Figure 4.0.1). The resulting contracting combinations can involve

individual systems with different dynamics, as long as those dynamics are contracting [Slotine and Lohmiller, 2001, Slotine, 2003].

Moreover, modular stability and specifically contractive stability have relevance to evolutionary biology [Simon, 1962, Slotine and Lohmiller, 2001]. In particular, it is thought that the majority of traits that have developed over the last 400+ million years are the result of evolutionary forces acting on regulatory elements that combine core components, rather than mutations in the core components themselves. This mechanism of action makes meaningful variation in population phenotypes much more feasible to achieve, and is appropriately titled "facilitated variation" [Gerhart and Kirschner, 2007]. In addition to the biological evidence for facilitated variation, computational models have demonstrated that this approach produces populations which are better able to generalize to new environments [Parter et al., 2008], an ability that will be critical to further develop in deep learning systems. However, the tractability of these evolutionary processes hinges on some mechanism for ensuring stability of combinations. Because contraction analysis tools allow complicated contracting systems to be built up recursively from simpler elements, this form of stability would be well suited for biological systems [Slotine and Liu, 2012]. Our work with the Sparse Combo Net in Section 4.3 has direct parallels to facilitated variation, in that we train this combination network architecture *only* through training connections between contracting subnetworks.

Ultimately, our contributions are three-fold:

- A novel parameterization for feedback combination of contracting RNNs that enables direct optimization using standard deep learning libraries.
- Novel contraction conditions for continuous-time nonlinear RNNs, to use in conjunction with the combination condition. We also identify flaws in stability proofs from prior literature.
- Experiments demonstrating that our 'network of networks' sets a new state of the art for stability-constrained RNNs on benchmark sequential processing tasks.

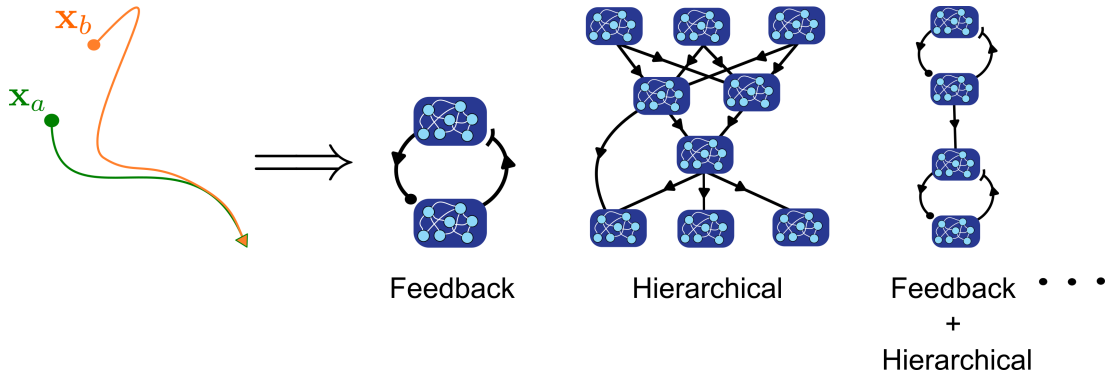

Figure 4.0.1: Contractive stability implies a modularity principle. Because contraction analysis tools allow complicated contracting systems to be built up recursively from simpler elements, this form of stability is well suited for understanding biological systems. Contracting combinations can be made between systems with very different dynamics, as long as those dynamics are contracting.

4.1 Network of Networks Model

In this paper we analyze rate-based neural networks. Unlike spiking neural networks, these models are continuous and smooth. We consider the following RNN introduced by [Wilson and Cowan, 1972], which may be viewed as an approximation to a more biophysically-detailed spiking network:

$$\tau\dot{\mathbf{x}} = -\mathbf{x} + \mathbf{W}\phi(\mathbf{x}) + \mathbf{u}(t) \tag{4.1}$$

Here $\tau > 0$ is the time-constant of the network [Dayan and Abbott, 2005], and the vector $\mathbf{x} \in \mathbb{R}^n$ contains the voltages of all n neurons in the network. The voltages are converted into firing-rates through a static nonlinearity ϕ. We only consider monotonic activation functions with bounded slope: in other words, $0 \leq \phi' \leq g$ (unless otherwise noted). We do not restrain the sign of the nonlinearity. Common example nonlinearities $\phi(x)$ that satisfy these constraints are hyperbolic tangent and ReLU. The matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$ contains the synaptic weights of the RNN. It is this matrix that ultimately determines the stability of (4.1), and will be a main target for our analysis. Finally, $\mathbf{u}(t)$ is the potentially time-varying external input into the network, capturing both explicit input into the RNN from the external world, as well as unmodeled dynamics from other brain areas.

Note that (4.1) is equivalent to another commonly used class of RNNs where the term $\mathbf{Wx} + \mathbf{u}$ appears inside the nonlinearity. See Section 4.A.1 or [Miller and Fumarola, 2012] for details. Our mathematical results apply equally well to both types of RNNs.

In order to extend (4.1) into a model of multiple interacting neural populations, we introduce the index i, which runs from 1 to p, where p is the total number of RNNs in the collection of RNNs. For now we will assume linear interactions between RNNs, because this is the simplest case. The linearity assumption can also be motivated by the fact that RNNs have been found to be well-approximated by linear systems in many neuroscience contexts [Sussillo and Barak, 2013, Langdon and Engel, 2022]. This leads to the following equation for the 'network of networks':

$$\tau \dot{\mathbf{x}}_i = -\mathbf{x}_i + \mathbf{W}_i \phi(\mathbf{x}_i) + \sum_{j=1}^{p} \mathbf{L}_{ij} \mathbf{x}_j + \mathbf{u}_i(t) \quad \forall i = 1 \cdots p \tag{4.2}$$

where the matrix $\mathbf{L}_{ij}$ captures the interaction between RNN i and RNN j. If RNN i has N_i neurons and RNN j has N_j neurons, then $\mathbf{L}_{ij}$ is an $N_i \times N_j$ matrix.

We can now formalize the question posed in the introduction. Namely, "what types of connections between stable RNNs automatically preserve stability?" becomes "what restrictions on $\mathbf{L}_{ij}$ must be met in order to ensure overall stability of the network of networks?". We will now derive two combination 'primitives', negative feedback and hierarchical, which allow for recursive connection of contracting modules.

4.1.1 Generalized Negative Feedback Between RNNs Preserves Stability

We set aside for a moment the problem of determining when (4.1) is contracting. For now, assume that we have a collection of p contracting RNNs interacting through equation (4.2). Recall that contraction is defined with respect to a *metric*, a way of measuring distances between trajectories in state space. Thus, the ith RNN is contracting with respect to some metric $\mathbf{M}_i$. We assume for simplicity that this metric

is constant, which means that $\mathbf{M}_i$ is simply a symmetric, positive definite matrix. In the case where every RNN receives feedback from every other, we can preserve stability by ensuring these connections are negative feedback. In the simplest case where all RNN modules are contracting in the identity metric, the negative feedback may be written as:

$$\mathbf{L}_{ij} = -\mathbf{L}_{ji}^T$$

This is a well known result from the contraction analysis literature [Slotine, 2003]. Our first novel contribution is a generalization and parameterization of this negative feedback which allows for direct optimization using gradient-based techniques. In particular, if each $\mathbf{L}_{ij}$ is parameterized as follows:

$$\mathbf{L}_{ij} = \mathbf{B}_{ij} - \mathbf{M}_i^{-1}\mathbf{B}_{ji}^T\mathbf{M}_j \quad \forall i, j \tag{4.3}$$

for arbitrary matrix $\mathbf{B}_{ij}$, then the overall network of networks retains the assumed contraction properties of the RNN subnetworks. We provide a detailed proof in Section 4.C.1, but the basic idea relies on ensuring that $\mathbf{L}_{ij} = -\mathbf{L}_{ji}^T$ *in the appropriate metric.* This can be achieved via the constraint:

$$\mathbf{M}_i\mathbf{L}_{ij} = -\mathbf{L}_{ji}^T\mathbf{M}_j$$

Plugging (4.3) into the above expression verifies that it is indeed satisfied. Because contraction analysis relies on analyzing the symmetric part of Jacobian matrices, the skew-symmetry of $\mathbf{L}_{ij}$ 'cancels out' when computing the symmetric part, and leaves the stability of the subnetwork RNNs untouched.

Recursive Properties of Contracting Combinations The feedback combination (4.3), taken together with hierarchical combinations, may be used as combination primitives for recursively constructing complicated networks of networks while automatically maintaining stability. The recursion comes from the fact that once a modular system is shown to be contracting it may be treated as a single contracting

system, which may in turn be combined with other contracting systems, *ad infinitum*. Note that while the feedback (4.3) requires linear interareal connections, hierarchical interareal connections may be nonlinear [Lohmiller and Slotine, 1998].

4.2 Many Different Ways to Achieve Local Contraction

In this section we return to the question of achieving contraction in the subnetwork RNNs. Recall that we wish to find restrictions on $\mathbf{W}_i$ such that the ith subnetwork RNN, described by (4.1), is contracting. Here we derive five such novel conditions (see Section 4.C for detailed proofs). As we will discuss, not all contraction conditions are equally useful - for example some conditions are easier to optimize or have higher model capacity than others. In this section we also point out some flaws in existing stability proofs in the RNN literature, and suggest some pathways towards correcting them.

Theorem 3 (Absolute Value Restricted Weights)**.** *Let* $|\mathbf{W}|$ *denote the matrix formed by taking the element-wise absolute value of* $\mathbf{W}$*. If there exists a positive, diagonal* $\mathbf{P}$ *such that:*

$$\mathbf{P}(g|\mathbf{W}| - \mathbf{I}) + (g|\mathbf{W}| - \mathbf{I})^T\mathbf{P} \prec 0$$

then (4.1) *is contracting in metric* $\mathbf{P}$*. If* $W_{ii} \leq 0$*, then* $|W|_{ii}$ *may be set to zero to reduce conservatism.*

It is easy to find matrices that satisfy Theorem 3, and given a matrix the condition is as easy to check as linear stability is. Moreover, the condition guarantees we can obtain a metric that the system is known to contract in (see Section 4.3.1 for details). It is less straightforward to enforce this condition during training, however we found that subnetworks constrained by Theorem 3 can achieve high performance in practice by simply fixing $\mathbf{W}$ and only optimizing the connections *between* subnetworks (Section 4.3.2). As there are fewer parameters to optimize, this training technique is faster.

Theorem 4 (Symmetric Weights)**.** *If* $\mathbf{W} = \mathbf{W}^T$ *and* $g\mathbf{W} \prec \mathbf{I}$*, and* $\phi' > 0$*, then (4.1) is contracting.*

It has been known since the early 1990s that if (4.1) is autonomous (i.e the input $\mathbf{u}$ is constant) and has symmetric weights with eigenvalues less than $1/g$, then there exists a unique fixed point that the network converges to from any initial condition [Matsuoka, 1992]. Theorem 4 generalizes this statement to say that if (4.1) has symmetric weights with eigenvalues less than $1/g$, it is contracting. This includes previous results as a special case, because an *autonomous* contracting system has a unique fixed point which the network converges to from any initial condition.

Theorem 5 (Product of Diagonal and Orthogonal Weights)**.** *If there exists positive diagonal matrices* $\mathbf{P}_1$ *and* $\mathbf{P}_2$*, as well as* $\mathbf{Q} = \mathbf{Q}^T \succ 0$ *such that*

$$\mathbf{W} = -\mathbf{P}_1\mathbf{Q}\mathbf{P}_2$$

then (4.1) is contracting in metric $\mathbf{M} = (\mathbf{P}_1\mathbf{Q}\mathbf{P}_1)^{-1}$*.*

In contrast to the first two contraction conditions, Theorem 5 is very easy to optimize. To meet the constraint that the $\mathbf{P}$ matrices are positive, one can parameterize their diagonal elements as $P_{ii} = d_i^2 + \epsilon$, for some small positive constant ϵ, and optimize d_i directly. To meet the positive definiteness constraint on $\mathbf{Q}$, one may parameterize it as $\mathbf{Q} = \mathbf{E}^T\mathbf{E} + \epsilon\mathbf{I}$ and optimize $\mathbf{E}$ directly.

Theorem 6 (Triangular Weights)**.** *If* $g\mathbf{W} - \mathbf{I}$ *is triangular and Hurwitz, then (4.1) is contracting in a diagonal metric.*

Theorem 6 follows from the fact that a hierarchy of contracting systems is also contracting.

Theorem 7 (Singular Value Restricted Weights)**.** *If there exists a positive diagonal matrix* $\mathbf{P}$ *such that:*

$$g^2\mathbf{W}^T\mathbf{P}\mathbf{W} - \mathbf{P} \prec 0$$

then (4.1) is contracting in metric $\mathbf{P}$*.*

In the case of discrete-time RNNs, this contraction condition has been proved by many different authors in many different settings. When $\mathbf{P} = \mathbf{I}$, it is known as the echo-state condition for discrete-time RNNs [Jaeger, 2001]. This was then generalized to diagonal $\mathbf{P}$ by Buehner and Young [2006]. More recently, the original echo-state condition was rediscovered by Miller and Hardt [2018] in the machine learning literature. Following this rediscovery, the condition was generalized to $\mathbf{P} \neq \mathbf{I}$ by Revay and Manchester [2020]. Here we show that it applies to continuous-time RNNs as well.

4.2.1 What do the Jacobian Eigenvalues Tell Us?

Several recent papers in ML, e.g [Haber and Ruthotto, 2017, Chang et al., 2019], claim that a sufficient condition for stability of the nonlinear system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$$

is that the associated Jacobian matrix $\mathbf{J}(\mathbf{x}, t) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ has eigenvalues whose real parts are strictly negative, i.e:

$$\max_i \mathrm{Re}(\lambda_i(\mathbf{J}(\mathbf{x}, t)) \leq -\alpha$$

with $\alpha > 0$. However, this claim is generally false - see Section 4.4.2 in [Slotine and Li, 1991].

In the *specific* case of the RNN (4.1), it appears that the eigenvalues of the symmetric part of $\mathbf{W}$ do provide information on global stability in a number of applications. For example, in [Matsuoka, 1992] it was shown that if $\mathbf{W}_s = \frac{1}{2}(\mathbf{W} + \mathbf{W}^T)$ has all its eigenvalues less than unity, and $\mathbf{u}$ is constant, then (4.1) has a unique, globally asymptotically stable fixed point. This condition also implies that the real parts of the eigenvalues of the Jacobian are uniformly negative. Moreover, in [Chang et al., 2019] it was shown that setting the symmetric part of $\mathbf{W}_s = \frac{1}{2}(\mathbf{W} + \mathbf{W}^T)$ almost equal to zero (yet slightly negative) led to rotational, yet stable dynamics in practice. This leads us to the following theorem, which shows that if the slopes of the

activation functions change sufficiently slowly as a function of time, then the condition in [Matsuoka, 1992] in fact implies global contraction of (4.1).

Theorem 8. *Let* $\mathbf{D}$ *be a positive, diagonal matrix with* $D_{ii} = \frac{d\phi_i}{dx_i}$*, and let* $\mathbf{P}$ *be an arbitrary, positive diagonal matrix. If:*

$$(g\mathbf{W} - \mathbf{I})\mathbf{P} + \mathbf{P}(g\mathbf{W}^T - \mathbf{I}) \preceq -c\mathbf{P} \quad \textit{and} \quad \dot{\mathbf{D}} - cg^{-1}\mathbf{D} \preceq -\beta\mathbf{D}$$

for $c, \beta > 0$*, then (4.1) is contracting in metric* $\mathbf{D}$ *with rate* β*.*

We stress however, that it is an open question whether or not diagonal stability of $\mathbf{W}$ implies that (4.1) is contracting. It has been conjectured that diagonal stability of $g\mathbf{W} - \mathbf{I}$ is a sufficient condition for global contraction of (4.1) [Revay et al., 2020], however this has been difficult to prove. To better characterize this conjecture, we present Theorem 9, which shows by way of counterexample that diagonal stability of $g\mathbf{W} - \mathbf{I}$ does not imply global contraction in a *constant* metric for (4.1).

Theorem 9. *Satisfaction of the condition* $g\mathbf{W}_{sym} - \mathbf{I} \prec 0$ *is* ***not*** *sufficient to show global contraction of the general nonlinear RNN (4.1) in any* ***constant*** *metric. High levels of antisymmetry in* $\mathbf{W}$ *can make it impossible to find such a metric, which we demonstrate via a* 2×2 *counterexample of the following form, with* $c \geq 2$ *when* $g = 1$*:*
$\mathbf{W} = \begin{bmatrix} 0 & -c \\ c & 0 \end{bmatrix}$

4.3 Stability-Constrained Network of Networks Perform Well on Benchmarks

A natural concern is that stability of an RNN may come at the cost of its expressivity, which is particularly relevant for integrating information over long timescales. To investigate whether this might be an issue for our model, we trained a stability-constrained network-of-networks on three benchmark sequential image classification tasks: sequential MNIST, permuted seqMNIST, and sequential CIFAR10. These tasks

are often used to measure information processing ability over long sequences [Le et al., 2015]. Images are presented pixel-by-pixel, and the network makes a prediction at the end of the sequence. In permuted seqMNIST, pixels are input in a fixed but random order.

All of our experiments were done on networks governed by (4.2). The nonlinear subnetwork RNNs were connected to each other via linear all-to-all negative feedback, given by (4.3). For all subnetworks we use the ReLU activation function. To enforce contraction of each individual subnetwork, we focused on two stability constraints from our theoretical results: Theorems 3 and 7. In the case of Theorem 3, we did not train the individual subnetworks' weight matrices, but only trained the connections *between* subnetworks (Figure 4.3.1B). For Theorem 7, we trained all parameters of the model (Figure 4.3.1C).

We refer to networks with subnetworks constrained by Theorem 3 as 'Sparse Combo Nets' and to networks with subnetworks constrained by Theorem 7 as 'SVD Combo Nets'. Throughout the experimental results we use the notation '$p \times n$ network' - such a network consists of p distinct subnetwork RNNs, with each such subnetwork RNN containing n units.

4.3.1 Network Initialization and Training

For the Sparse Combo Net we were not able to find a parameterization to continuously update the internal RNN weights during training in a way that preserved contraction. However, it is easy to randomly generate matrices with a particular likelihood of meeting the Theorem 3 condition by selecting an appropriate sparsity level and limit on entry magnitude. Sparsity in particular is of interest due to its relevance in neurobiology and machine learning, so it is convenient that the condition makes it easy to verify stability of many different sparse RNNs. As $g = 1$ for ReLU activation, we check potential subnetwork matrices $\mathbf{W}$ by simply verifying linear stability of $|\mathbf{W}| - \mathbf{I}$.

Because every RNN meeting the condition has a corresponding well-defined stable LTI system contracting in the same metric, it is also easy to find a metric to use in our training algorithm: solving for $\mathbf{M}$ in $-\mathbf{I} = \mathbf{M}\mathbf{A} + \mathbf{A}^T\mathbf{M}$ will produce a valid metric

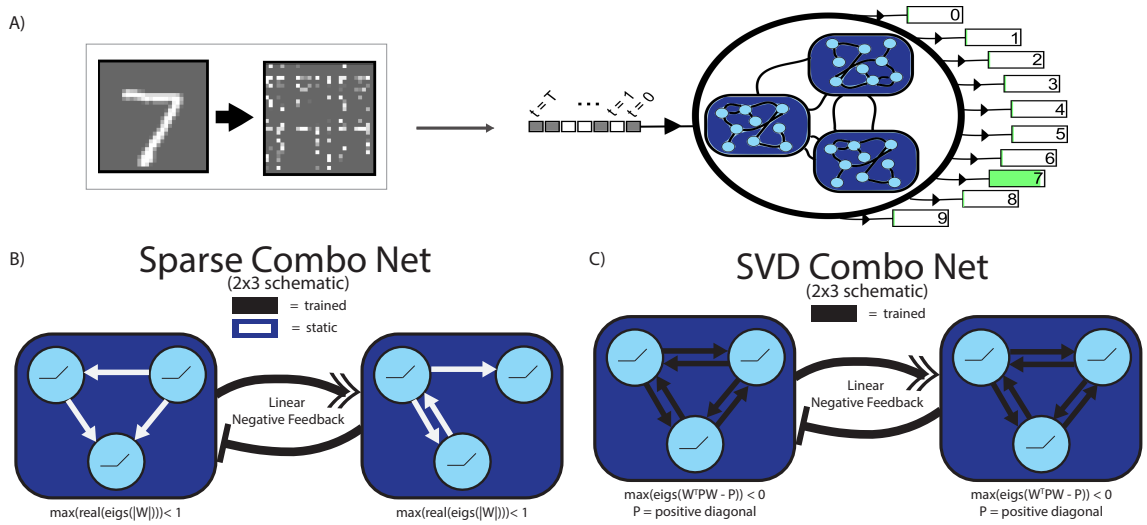

Figure 4.3.1: Summary of task structure and network architectures. Images from MNIST (or CIFAR10) were flattened into an array of pixels and fed sequentially into the modular 'network of networks', with classification based on the output at the last time-step. For MNIST, each image was also permuted in a fixed manner (A). The subnetwork 'modules' of our architecture were constrained to meet either Theorem 3 via sparse initialization (B) or Theorem 7 via direct parameterization (C). Linear negative feedback connections were trained between the subnetworks according to (4.3).

for any stable LTI system $\mathbf{A}$ [Slotine and Li, 1991]. We utilize the fact that Hurwitz Metzler matrices are diagonally stable to improve efficiency of computing $\mathbf{M}$ (as well as in our proof of Theorem 3).

We therefore randomly generated fixed subnetworks satisfying Theorem 3 and trained only the linear connections between them (Figure 4.3.2), as well as the linear input and output layers. More information on network initialization, hyperparameter tuning, and training algorithm is provided in Section 4.D.

For the SVD Combo Net on the other hand, we ensured contraction of each subnetwork RNN by direct parameterization (described in Section 4.E), thus allowing all weights to be trained.

4.3.2 Results

The Sparse Combo Net architecture achieved the highest overall performance on both permuted seqMNIST and seqCIFAR10, with 96.94% and 65.72% best test accuracies

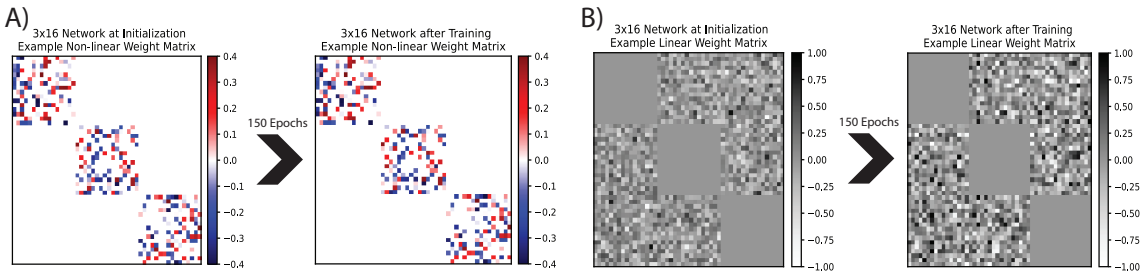

Figure 4.3.2: Example 3x16 Sparse Combo Net. Nonlinear intra-subnetwork weights are initialized using a set sparsity, and do not change in training (A). Linear inter-subnetwork connections are constrained to be antisymmetric with respect to the overall network metric, and are updated in training (B).

respectively - thereby setting a new SOTA for stable RNNs (Table 4.3.1). Furthermore, we were able to reproduce SOTA scores over several repetitions, including 10 trials of seqCIFAR10. Along with repeatability of results, we also show that the contraction constraint on the connections between subnetworks ($\mathbf{L}$ in (4.3)) is important for performance, particularly in the Sparse Combo Net (Section 4.3.2).

Additionally, we profile how various architecture settings impact performance of our networks. In both networks, we found that increasing the total number of neurons improved task performance, but with diminishing returns (Section 4.3.2). We also found that the sparsity of the hidden-to-hidden weights in Sparse Combo Net had a large impact on the final network performance (Section 4.3.2).

Experiments with Network Size

Understanding the effect of size on network performance is important to practical application of these architectures. For both Sparse Combo Net and SVD Combo Net, increasing the number of subnetworks while holding other settings constant (including fixing the size of each subnetwork at 32 units) was able to increase network test accuracy on permuted seqMNIST to a point (Figure 4.3.3).

The greatest performance jump happened when increasing from one module (37.1% Sparse Combo Net, 61.8% SVD Combo Net) to two modules (89.1% Sparse Combo Net, 92.9% SVD Combo Net). After that the performance increased steadily with number of modules until saturating at $\sim 97\%$ for Sparse Combo Net and $\sim 95\%$ for

Name	Stable RNN?	Params	sMNIST Repeats Mean (n) [Min]	psMNIST Repeats Mean (n) [Min]	sCIFAR10 Repeats Mean (n) [Min]	Seq MNIST Best	PerSeq MNIST Best	Seq CIFAR Best
LSTM [Chang et al., 2019]		68K	—	—	—	97.3%	92.7%	59.7%
Transformer [Trinh et al., 2018]		0.5M	—	—	—	98.9%	97.9%	62.2%
Antisymmetric [Chang et al., 2019]	?	36K	—	—	—	98%	95.8%	58.7%
Sparse Combo Net	✓	130K	—	**96.85%** (4) [**96.65%**]	64.72% (10) [63.73%]	99.04%	**96.94%**	**65.72%**
Lipschitz [Erichson et al., 2021]	✓	134K	99.2% (10) [99.0%]	95.9% (10) [95.6%]	—	**99.4%**	96.3%	64.2%
CKConv [Romero et al., 2021]		1M	—	—	—	99.32%	98.54%	63.74%
S4 [Gu et al., 2022]		7.9M	—	—	—	**99.63%**	**98.7%**	**91.13%**
Trellis [Bai et al., 2019]		8M	—	—	—	99.2%	98.13%	73.42%

Table 4.3.1: Published benchmarks for sequential MNIST, permuted MNIST, and sequential CIFAR10 best test accuracy. Architectures are grouped into three categories: baselines, best performing RNNs with claimed stability guarantee*, and networks achieving overall SOTA. Within each grouping, networks are ordered by number of trainable parameters (for CIFAR10 if it differed across tasks). Our network is highlighted. Where possible, we include information on repeatability.
*For more on stability guarantees in machine learning, see Section 4.2.1

SVD Combo Net.

As the internal subnetwork weights are not trained in Sparse Combo Net, it

is unsurprising that its performance was substantially worse at the smallest sizes. However Sparse Combo Net surpassed SVD Combo Net by the 12×32 network size, which contains a modest 384 total units. Due to the better performance of the Sparse Combo Net, we focused additional analyses there. Note also that the SVD Combo Net never reached 55% test accuracy for CIFAR10 in our early experiments.

We then evaluated task performance as the *modularity* of a Sparse Combo Net (fixed to have 352 total units) was varied. We observed an inverse U shape (Figure 4.D.1B), with poor performance of a 1×352 net and an 88×4 net, and best performance from a 44×8 net. However, this experiment compared similar sparsity levels, while in practice we can achieve better performance with larger subnetworks by leveraging sparsity in a way not possible for smaller ones.

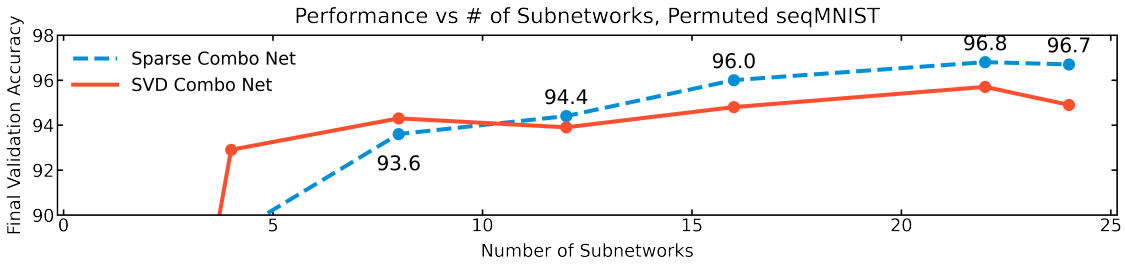

Figure 4.3.3: Permuted seqMNIST performance plotted against the number of subnetworks. Each subnetwork has 32 neurons. Results are shown for both Sparse Combo Net and SVD Combo Net.

Experiments with Sparsity

Because of the link between sparsity and stability as well as the biological relevance of sparsity, we explored in detail how subnetwork sparsity affects the performance of Sparse Combo Net. We ran a number of experiments on the permuted seqMNIST task, varying sparsity level while holding network size and other hyperparameters constant. Here we use "n% sparsity level" to refer to a network with subnetworks that have just n% of their weights non-zero.

We observed a large (> 5 percentage point) performance boost when switching from a 26.5% sparsity level to a 10% sparsity level in the 11×32 Sparse Combo Net

(Figure 4.3.4), and subsequently decided to test significantly sparser subnetworks in a 16×32 Sparse Combo Net. We trained networks with sparsity levels of 5%, 3.3%, and 1%, as well as 10% for baseline comparison (Figure 4.D.2A). A 3.3% sparsity level produced the best results, leading to our SOTA performance for stable networks on both permuted seqMNIST and seqCIFAR10. With a component RNN size of just 32 units, this sparsity level is quite small, containing only one or two directional connections per neuron on average (Figure 4.D.7).

As sparsity had such a positive effect on task performance, we did additional analyses to better understand why. We found that decreasing the magnitude of non-zero elements while holding sparsity level constant decreased task performance (Figure 4.D.2B), suggesting that the effect is driven in part by the fact that sparsity enables higher magnitude non-zero elements while still maintaining stability.

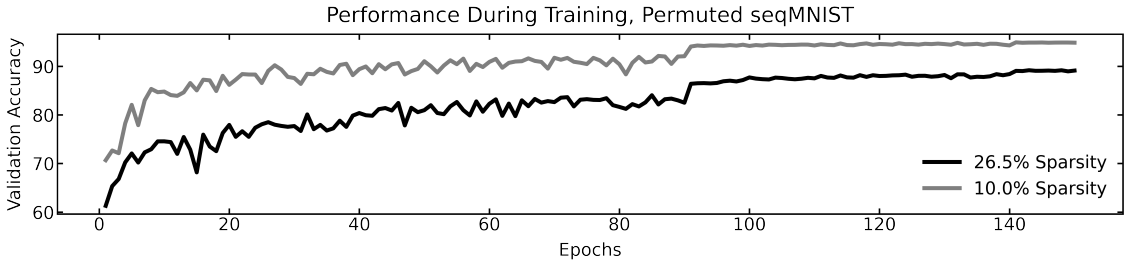

Figure 4.3.4: Permuted seqMNIST performance over the course of training for two 11×32 Sparse Combo Nets with different sparsity levels.

The use of sparsity in subnetworks to improve performance suggests another interesting direction that could enable better scalability of total network size - enforcing sparsity in the linear feedback weight matrix ($\mathbf{L}$). We performed pilot testing of this idea, which showed promise in mitigating the saturation effect seen in Figure 4.3.3. Those results are detailed in Section 4.D.2 (Table 4.D.1).

Repeatability and Controls

Sparse Combo Net does not have the connections within its subnetworks trained, so network performance could be particularly susceptible to random initialization. Thus we ran repeatability studies on permuted sequential MNIST and sequential CIFAR10

using our best network settings (16×32 with subnetwork sparsity level of 3.3%) and an extended training period. Mean performance over 4 trials of permuted seqMNIST was 96.85% with 0.019 variance, while mean performance over 10 trials of seqCIFAR10 was 64.72% with 0.406 variance. Note we also ran a number of additional experiments on size and sparsity settings, described in Section 4.D.2.

Across the permuted seqMNIST trials, best test accuracy always fell between 96.65% and 96.94%, a range much smaller than the differences seen with changing sparsity settings and network size. Three of the four trials showed best test accuracy $\geq$ 96.88%, despite some variability in early training performance (Figure 4.D.3). Similarly, eight of the ten seqCIFAR10 trials had test accuracy $>$ 64.3%, with all results falling between 63.73% and 65.72% (Figure 4.D.4). This robustly establishes a new SOTA for stable RNNs, comfortably beating the previously reported (single run) 64.2% test accuracy achieved by Lipschitz RNN [Erichson et al., 2021].

As a control study, we also tested how sensitive the Sparse Combo Net was to the stabilization condition on the interconnection matrix ($\mathbf{L}$ in (4.3)). To do so, we initialized the individual RNN modules in a 24×32 network as before, but set $\mathbf{L} = \mathbf{B}$ and did not constrain $\mathbf{B}$ at all during training, thus no longer ensuring contraction of the overall system. This resulted in 47.0% test accuracy on the permuted seqMNIST task, a stark decrease from the original 96.7% test accuracy - thereby demonstrating the utility of the contraction condition.

4.4 Discussion

Biologists have long noted that modularity provides organisms with stability and robustness [Kitano, 2004]. The other direction – that stability and robustness provide modularity – is well known to engineers [Khalil, 2002, Slotine and Li, 1991, Slotine, 2003], but has been less appreciated in biology. We use this principle to build and train provably stable assemblies of recurrent neural networks. Like real brains, the components of our "RNN of RNNs" can communicate with one another through a mix of hierarchical and feedback connections. In particular, we theoretically characterized

conditions under which an RNN of RNNs will be stable, given that each individual RNN is stable. We also provided several novel stability conditions for single RNNs that are compatible with these stability-preserving interareal connections. Our results contribute towards understanding how the brain maintains stable and accurate function in the presence of massive interareal feedback, as well as external inputs.

The question of neural stability is one of the oldest questions in computational neuroscience. Indeed, cyberneticists were concerned with this question before the term 'computational neuroscience' existed [Wiener, 1948, Ashby, 1952b]. Stability is a central component in several influential neuroscience theories [Hopfield, 1982, Seung, 1996, Murphy and Miller, 2009], perhaps the most well-known being that memories are stored as stable point attractors [Hopfield, 1982]. Our work shows that stability continues to be a useful concept for computational neuroscience as the field transitions from focusing on single brain areas to many interacting brain areas.

While primarily motivated by neuroscience, our approach is also relevant for machine learning. Deep learning models can be as inscrutable as they are powerful. This opacity limits conceptual progress and may be dangerous in safety-critical applications like autonomous driving or human-centered robotics. Given that stability is a fundamental property of dynamical systems – and is intimately linked to concepts of control, generalization, efficiency, and robustness – the ability to guarantee stability of a recurrent model will be important for ensuring deep networks behave as we expect them to [Richards et al., 2018, Choromanski et al., 2020, Revay et al., 2021, Rodriguez et al., 2022].

In the case of RNNs, one difficulty is that providing a certificate of *stability* is often impossible or computationally impractical. However, the stability conditions we derive here allow for recursive construction of complicated RNNs while automatically preserving stability. By parameterizing our conditions for easy optimization using gradient-based techniques, we successfully trained our architecture on challenging sequential processing benchmarks. The high test accuracy our networks achieved with a small number of trainable parameters demonstrates that stability does not necessarily come at the cost of expressivity. Thus, our results likewise contribute

towards understanding stability certification of RNNs.

In future work, we will explore how our contraction-constrained RNNs of RNNs perform on a variety of neuroscience tasks, in particular tasks with multimodal structure [Yang et al., 2019]. One desiderata for those future models is that they learn representations which are formally similar to those observed in the brain [Yamins et al., 2014, Schrimpf et al., 2020, Williams et al., 2021], in complement with the structural similarities already shared. Moreover, a "network of networks" approach will be especially relevant to challenging multimodal machine learning problems, such as the simultaneous processing of audio and video. Therefore the advancement of neuroscience theory and machine learning remain hand-in-hand for our next lines of questioning. Indeed, combinations of trained networks have already seen groundbreaking success in DeepMind's AlphaGo [Silver et al., 2016].

As well as the many potential experimental applications, there are numerous theoretical future directions suggested by our work. Networks with more biologically-plausible weight update rules, such as models discussed in [Kozachkov et al., 2020], would be a fruitful neuroscience context in which to explore our conditions. One promising avenue of study there is to examine input-dependent stability of the learning process. In the context of machine learning, our stability conditions could be applied to the end-to-end training of multidimensional recurrent neural networks [Graves et al., 2007], which have clear structural parallels to our RNNs of RNNs but lack known stability guarantees.

In sum, recursively building network combinations in an effective and stable fashion while also allowing for continual refinement of the individual networks, as nature does for biological networks, will require new analysis tools. Here we have taken a concrete step towards the development of such tools, not only through our theoretical results, but also through their application to create stable combination network architectures that perform well in practice on benchmark tasks.

Appendix

4.A Extended Background

4.A.1 Two Different RNNs

Note that in neuroscience, the variable $\mathbf{x}$ in equation (4.1) is typically thought of as a vector of neural membrane potentials. It was shown in [Miller and Fumarola, 2012] that the RNN (4.1) is equivalent via an affine transformation to another commonly used RNN model,

$$\tau\dot{\mathbf{y}} = -\mathbf{y} + \phi(\mathbf{W}\mathbf{y} + \mathbf{b}(t)) \tag{4.4}$$

where the variable $\mathbf{y}$ is interpreted as a vector of firing rates, rather than membrane potentials. The two models are related by the transformation $\mathbf{x} = \mathbf{W}\mathbf{y} + \mathbf{b}$, which yields

$$\tau\dot{\mathbf{x}} = \mathbf{W}(-\mathbf{y} + \phi(\mathbf{W}\mathbf{y} + \mathbf{b})) + \tau\dot{\mathbf{b}} = -\mathbf{x} + \mathbf{W}\phi(\mathbf{x}) + \mathbf{v}$$

where $\mathbf{v} \equiv \mathbf{b} + \tau\dot{\mathbf{b}}$. Thus $\mathbf{b}$ is a low-pass filtered version of $\mathbf{v}$ (or conversely, $\mathbf{v}$ may be viewed as a first order prediction of $\mathbf{b}$) and the contraction properties of the system are unaffected by the affine transformation. Note that the above equivalence holds even in the case where $\mathbf{W}$ is not invertible. In this case, the two models are proven to be equivalent, provided that $\mathbf{b}(0)$ and $\mathbf{y}(0)$ satisfy certain conditions–which are always possible to satisfy [Miller and Fumarola, 2012]. Therefore, any contraction condition derived for the x (or y) system automatically implies contraction of the other system. We exploit this freedom freely throughout the paper.

4.A.2 Contraction Math

It can be shown that the non-autonomous system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$$

is contracting if there exists a metric $\mathbf{M}(\mathbf{x},t) = \mathbf{\Theta}(\mathbf{x},t)^T\mathbf{\Theta}(\mathbf{x},t) \succ 0$ such that uniformly

$$\dot{\mathbf{M}} + \mathbf{M}\mathbf{J} + \mathbf{J}^T\mathbf{M} \preceq -\beta\mathbf{M}$$

where $\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ and $\beta > 0$. For more details see the main reference [Lohmiller and Slotine, 1998]. Similarly, a non-autonomous discrete-time system

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, t)$$

is contracting if

$$\mathbf{J}^T\mathbf{M}_{t+1}\mathbf{J} - \mathbf{M}_t \preceq -\beta\mathbf{M}_t$$

Feedback and Hierarchical Combinations

Consider two systems, independently contracting in constant metrics $\mathbf{M}_1$ and $\mathbf{M}_2$, which are combined in feedback:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x},t) + \mathbf{B}\mathbf{y} \\ \dot{\mathbf{y}} &= \mathbf{g}(\mathbf{y},t) + \mathbf{G}\mathbf{x} \end{aligned} \qquad \text{(Feedback Combination)}$$

If the following relationship between $\mathbf{B}, \mathbf{G}, \mathbf{M}_1$, and $\mathbf{M}_2$ is satisfied:

$$\mathbf{B} = -\mathbf{M}_1^{-1}\mathbf{G}^T\mathbf{M}_2$$

then the combined system is contracting as well. This may be seen as a special case of the feedback combination derived in [Tabareau and Slotine, 2006]. The situation is even simpler for hierarchical combinations. Consider again two systems, independently contracting in some metrics, which are combined in hierarchy:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$$
$$\dot{\mathbf{y}} = \mathbf{g}(\mathbf{y}, t) + \mathbf{h}(\mathbf{x}, t) \qquad \text{(Hierarchical Combination)}$$

where $\mathbf{h}(\mathbf{x}, t)$ is a function with *bounded* Jacobian. Then this combined system is contracting in a diagonal metric, as shown in [Lohmiller and Slotine, 1998]. By recursion, this extends to hierarchies of arbitrary depth.

4.B Extended Discussion

Given the paucity of existing theory on modular networks, our novel stability conditions and proof of concept combination architectures are a significant step in an important new direction. The "network of networks" approach is evident in the biological brain, and has seen early practical success in applications such as AlphaGo. There is much evidence this line of questioning will be critical in the future, and our work is the first on stable *modular* networks.

Furthermore, we develop an architecture based on such combinations of "vanilla" RNNs that is both stable and achieves high performance on benchmark sequence classification tasks using few trainable parameters (small particularly for sequential CIFAR10). When considering just the facts about the network, it really has no business performing anywhere near as well as it does. Note also that without the stability condition in place, the network performance indeed drops substantially.

In order to facilitate the extension of this important line of thinking, we provide additional context on the limitations of our current approach as well as promising ideas for future directions in this section.

4.B.1 Limitations

One drawback of our approach is that we parameterize each weight matrix in a special way to guarantee stability. In all cases, this requires us to parameterize matrices as

the product of several other matrices. Thus, we gain stability at the cost of increasing the number of parameters, which can slow down training.

Another current drawback is that we only consider constant metrics. In theory, contraction metrics can be state-dependent as well as time-varying. Thus it is possible that we have overly restricted the space of models we consider. Similarly, negative feedback is not the only way to preserve contraction when combining two contracting systems. There are known small-gain theorems in the contraction analysis literature which accomplish the same task [Slotine, 2003]. However, parameterizing these conditions is less straightforward than parameterizing the negative-feedback condition.

A third limitation of the present work is that it does not give a recipe on how to incorporate anatomical knowledge into the building of 'RNNs of RNNs'. Our current approach is 'bottom up', in the sense that we describe complicated networks which can be built from simpler ones while ensuring stability at every level of construction. However, for building biological models of the brain it is important to incorporate known anatomical detail (i.e V4 projects to PFC, PFC projects back to V4, etc). How to do this in a way that preserves stability is an open and interesting question.

Lastly, we only tested our networks on sequential image classification benchmarks. Future work will include other benchmarks such as character or word-level language modeling. Additionally, while we conducted preliminary experiments exploring the role of scale (i.e number of subnetwork RNNs), we did not pursue this at the sizes reached by many modern deep learning applications. Thus it is currently unclear if the performance of these stability-constrained models will scale well enough with the number of subnetworks (or the number of neurons per subnetwork). Testing this correctly will require extensive experimentation with the various initialization and training settings.

4.B.2 Future Directions

There are numerous future directions enabled by this work. For example, Theorem 8 suggests that a less restrictive contraction condition on W in terms of the eigenvalues of the symmetric part is possible and desirable. Meanwhile, Theorem 9 provides

important groundwork in finding such a condition, as it shows the need for a time-varying metric. Investigation of input-dependent metrics could be a fruitful next line of research, and would have far-reaching implications in disciplines such as curriculum learning.

Furthermore, the beneficial impact of sparsity on training these stable models suggests a potential avenue for additional experimental work – in particular adding a regularizing sparsity term during training. This could allow Sparse Combo Net to have its internal subnetwork weights trained without losing the stability guarantee, and conversely it could allow SVD Combo Net to reap some of the performance benefits of Sparse Combo Net without giving up the flexibility allowed by training said internal weights.

As described in the limitations above, a major experimental next step will be to test our architectures at greater scale and on more difficult tasks. Since 'network of network' approaches are becoming increasingly popular, our methodology is relevant to a variety of task types, including reinforcement learning applications. Given the biological inspiration, multi-modal learning tasks may also be of particular relevance.

4.C Proofs for Main Results

4.C.1 Proof of Feedback Combination Property

Here we apply existing contraction analysis results to derive equation (4.3). Because (4.3) is a parameterization of a known contraction conditions [Slotine, 2003], we provide the following statement in the form of a corollary.

Corollary 1 (Network of Networks)**.** *Consider a collection of p subnetwork RNNs governed by* (4.1)*. Assume that these RNNs each have hidden-to-hidden weight matrices $\{\mathbf{W}_1, \dots, \mathbf{W}_p\}$ and are independently contracting in metrics $\{\mathbf{M}_1, \dots, \mathbf{M}_p\}$. Define the block matrices $\tilde{\mathbf{W}} \equiv$ BlockDiag$(\mathbf{W}_1, \dots, \mathbf{W}_p)$, $\tilde{\mathbf{M}} \equiv$ BlockDiag$(\mathbf{M}_1, \dots, \mathbf{M}_p)$,the overall state vector $\tilde{\mathbf{x}}^T \equiv (\mathbf{x}_1^T \cdots \mathbf{x}_p^T)$, and finally $\tilde{\mathbf{u}}^T \equiv (\mathbf{u}_1^T \cdots \mathbf{u}_p^T)$ Then the following*

'network of networks' is globally contracting in metric $\tilde{\mathbf{M}}$*:*

$$\begin{aligned} \tau\dot{\tilde{\mathbf{x}}} &= -\tilde{\mathbf{x}} + \tilde{\mathbf{W}}\phi(\tilde{\mathbf{x}}) + \tilde{\mathbf{u}} + \mathbf{L}\tilde{\mathbf{x}} \\ \mathbf{L} &\equiv \mathbf{B} - \tilde{\mathbf{M}}^{-1}\mathbf{B}^T\tilde{\mathbf{M}} \end{aligned} \tag{4.5}$$

Where $\mathbf{B}$ *is an arbitrary square matrix. The matrix* $\mathbf{L}$ *is made up of many individual sub-matrices (the* $\mathbf{L}_{ij}$ *terms in* (4.2)*) which define the interareal connectivity of the overall network.*

Proof. Consider the differential Lyapunov function:

$$V = \frac{1}{2}\delta\mathbf{x}^T\tilde{\mathbf{M}}\delta\mathbf{x}$$

The time-derivative of this function is:

$$\dot{V} = \delta\mathbf{x}^T\tilde{\mathbf{M}}\delta\dot{\mathbf{x}} = \delta\mathbf{x}^T(\underbrace{\tilde{\mathbf{M}}\tilde{\mathbf{J}} + \tilde{\mathbf{J}}^T\tilde{\mathbf{M}}}_{\text{Jacobian of RNNs before interconnection}} + \underbrace{\tilde{\mathbf{M}}\mathbf{L} + \mathbf{L}^T\tilde{\mathbf{M}}}_{\text{Interconnection Jacobian}})\delta\mathbf{x}$$

Since we assume that the RNNs are contracting in isolation, the first term in this sum is less that the slowest contracting rate of the individual RNNs, which we call $\lambda > 0$. The second term in the sum is the zero matrix, by construction. Thus the time-derivative of V is upper-bounded by:

$$\dot{V} \leq -2\lambda V$$

This implies that the network of networks is contracting with rate λ. □

4.C.2 Proof of Theorem 3

Our first theorem is motivated by the observation that if the y-system (described in Section 4.A.1) is to be interpreted as a vector of firing rates, it must stay positive for all time. For a linear, time-invariant system with positive states, diagonal stability is equivalent to stability. Therefore a natural question is if diagonal stability of a

linearized y-system implies anything about stability of the nonlinear system. More formally, given an excitatory neural network (i.e $\forall ij, W_{ij} \geq 0$), if the linear system

$$\dot{\mathbf{x}} = -\mathbf{x} + g\mathbf{W}\mathbf{x}$$

is stable, then there exists a positive diagonal matrix P such that:

$$\mathbf{P}(g\mathbf{W} - \mathbf{I}) + (g\mathbf{W} - \mathbf{I})^T\mathbf{P} \prec 0$$

The following theorem shows that the nonlinear system (4.1) is indeed contracting in metric $\mathbf{P}$, and extends this result to a more general $\mathbf{W}$ by considering only the magnitudes of the weights.

Theorem 1. *Let $|\mathbf{W}|$ denote the matrix formed by taking the element-wise absolute value of $\mathbf{W}$. If there exists a positive, diagonal $\mathbf{P}$ such that:*

$$\mathbf{P}(g|\mathbf{W}| - \mathbf{I}) + (g|\mathbf{W}| - \mathbf{I})^T\mathbf{P} \prec 0$$

then (4.1) *is contracting in metric $\mathbf{P}$. Moreover, if $W_{ii} \leq 0$, then $|W|_{ii}$ may be set to zero to reduce conservatism.*

This condition is particularly straightforward in the common special case where the network does not have any self weights, with the leak term driving stability. While it can be applied to a more general $\mathbf{W}$, the condition will of course not be met if the network was relying on highly negative values on the diagonal of $\mathbf{W}$ for linear stability. As demonstrated by counterexample in the proof of Theorem 3, it can be impossible to use the same metric $\mathbf{P}$ for the nonlinear RNN in such cases.

Theorem 3 allows many weight matrices with low magnitudes or a generally sparse structure to be verified as contracting in the nonlinear system (4.1), by simply checking a linear stability condition (as linear stability is equivalent to diagonal stability for Metzler matrices too [Narendra and Shorten, 2010]).

Beyond verifying contraction, Theorem 3 actually provides a metric, with little need for additional computation. Not only is it of inherent interest that the same

metric can be shared across systems in this case, it is also of use in machine learning applications, where stability certificates are becoming increasingly necessary. Critically, it is feasible to enforce the condition during training via L2 regularization on $\mathbf{W}$. More generally, there are a variety of systems of interest that meet this condition but do not meet the well-known maximum singular value condition, including those with a hierarchical structure.

Proof. Consider the differential, quadratic Lyapunov function:

$$V = \delta\mathbf{x}^T\mathbf{P}\delta\mathbf{x}$$

where $\mathbf{P} \succ 0$ is diagonal. The time derivative of V is:

$$\dot{V} = 2\delta\mathbf{x}^T\mathbf{P}\dot{\delta\mathbf{x}} = 2\delta\mathbf{x}^T\mathbf{P}\mathbf{J}\delta\mathbf{x} = -2\delta\mathbf{x}^T\mathbf{P}\delta\mathbf{x} + 2\delta\mathbf{x}^T\mathbf{P}\mathbf{W}\mathbf{D}\delta\mathbf{x}$$

where $\mathbf{D}$ is a diagonal matrix such that $\mathbf{D}_{ii} = \frac{d\phi_i}{dx} \geq 0$. We can upper bound the quadratic form on the right as follows:

$$\delta\mathbf{x}^T\mathbf{P}\mathbf{W}\mathbf{D}\delta\mathbf{x} = \sum_{ij} P_i W_{ij} D_j \delta x_i \delta x_j \leq$$

$$\sum_i P_i W_{ii} D_i |\delta x_i|^2 + \sum_{ij, i\neq j} P_i |W_{ij}| D_j |\delta x_i||\delta x_j| \leq g|\delta\mathbf{x}|^T\mathbf{P}|\mathbf{W}||\delta\mathbf{x}|$$

If $W_{ii} \leq 0$, the term $P_i W_{ii} D_i |\delta x_i|^2$ contributes non-positively to the overall sum, and can therefore be set to zero without disrupting the inequality. Now using the fact that $\mathbf{P}$ is positive and diagonal, and therefore $\delta\mathbf{x}^T\mathbf{P}\delta\mathbf{x} = |\delta\mathbf{x}|^T\mathbf{P}|\delta\mathbf{x}|$, we can upper bound $\dot{V}$ as:

$$\dot{V} \leq |\delta\mathbf{x}|^T(-2\mathbf{P} + \mathbf{P}|\mathbf{W}| + |\mathbf{W}|\mathbf{P})|\delta\mathbf{x}| = |\delta\mathbf{x}|^T[(\mathbf{P}(|\mathbf{W}| - \mathbf{I}) + (|\mathbf{W}|^T - \mathbf{I})\mathbf{P})]|\delta\mathbf{x}|$$

where $|W|_{ij} = |W_{ij}|$, and $|W|_{ii} = 0$ if $W_{ii} \leq 0$ and $|W|_{ii} = |W_{ii}|$ if $W_{ii} > 0$. This completes the proof.

Note that $\mathbf{W} - \mathbf{I}$ is Metzler, and therefore will be Hurwitz stable if and only if $\mathbf{P}$ exists [Narendra and Shorten, 2010].

It is also worth noting that highly negative diagonal values in $\mathbf{W}$ will prevent the same metric $\mathbf{P}$ from being used for the nonlinear system. Therefore the method used in this proof cannot feasibly be adapted to further relax the treatment of the diagonal part of $\mathbf{W}$.

The intuitive reason behind this is that in the symmetric part of the Jacobian, $\frac{\mathbf{PWD}+\mathbf{DW}^T\mathbf{P}}{2} - \mathbf{P}$, the diagonal self weights will also be scaled down by small $\mathbf{D}$, while the leak portion $-\mathbf{P}$ remains untouched by $\mathbf{D}$.

Now we actually demonstrate a counterexample, presenting a 2×2 symmetric Metzler matrix $\mathbf{W}$ that is contracting in the identity in the linear system, but cannot be contracting *in the identity* in the nonlinear system (4.1):

$$\mathbf{W} = \begin{bmatrix} -9 & 2.5 \\ 2.5 & 0 \end{bmatrix}$$

To see that it is not possible for the more general nonlinear system with these weights to be contracting in the identity, take $\mathbf{D} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$. Now

$$(\mathbf{WD})_{sym} - \mathbf{I} = \begin{bmatrix} -1 & 1.25 \\ 1.25 & -1 \end{bmatrix}$$

which has a positive eigenvalue of $\frac{1}{4}$.

□

4.C.3 Proof of Theorem 4

While regularization may push networks towards satisfying Theorem 3, strictly enforcing the condition during optimization is not straightforward. This motivates the

rest of our theorems, which derive contraction results for specially structured weight matrices. Unlike Theorem 3, these results have direct parameterizations which can easily be plugged into modern optimization libraries.

Theorem 2. *If* $\mathbf{W} = \mathbf{W}^T$ *and* $g\mathbf{W} \prec \mathbf{I}$*, and and* $\phi' > 0$ *then (4.1) is contracting.*

When $\mathbf{W}$ is symmetric, (4.1) may be seen as a continuous-time Hopfield network. Continuous-time Hopfield networks with symmetric weights were recently shown to be closely related to Transformer architectures [Krotov and Hopfield, 2020, Ramsauer et al., 2020]. Specifically, the dot-product attention rule may be seen as a discretization of the continuous-time Hopfield network with softmax activation function [Krotov and Hopfield, 2020]. Our results here provide a simple sufficient (and nearly necessary, see above remark) condition for global exponential stability of a given *trajectory* for the Hopfield network. In the case where the input into the network is constant, this trajectory is a fixed point. Moreover, each trajectory associated with a unique input is guaranteed to be unique. Finally, we note that our results are flexible with respect to activation functions so long as they satisfy the slope-restriction condition. This flexibility may be useful when, for example, considering recent work showing that standard activation functions may be advantageously replaced by attention mechanisms [Dai et al., 2020].

Proof. We begin by writing $\mathbf{W} = \mathbf{R} - \mathbf{P}$ for some unknown $\mathbf{R} = \mathbf{R}^T$ and $\mathbf{P} = \mathbf{P}^T \succ 0$. The approach of this proof is to show by construction that the condition $g\mathbf{W} \prec \mathbf{I}$ implies the existence of an $\mathbf{R}$ and $\mathbf{P}$ such that the system is contracting in metric $\mathbf{P}$. We consider the y version of the RNN, which as discussed above is equivalent to the x version via an affine transformation.

The differential Lyapunov condition associated to the RNN is:

$$\delta\mathbf{x}^T[-2\mathbf{M} + \mathbf{MDW} + \mathbf{WDM} + \beta\mathbf{M}]\delta\mathbf{x} \leq 0 \tag{4.6}$$

Where $\mathbf{M}, \mathbf{W} \in \mathbb{R}^{n\times n}$. Let us now make the substitution $\mathbf{M} = \mathbf{P}$ and $\mathbf{W} = \mathbf{R} - \mathbf{P}$:

$$\delta\mathbf{x}^T[-2\mathbf{P} + \mathbf{PD}(\mathbf{R}-\mathbf{P}) + (\mathbf{R}-\mathbf{P})\mathbf{DP} + \beta\mathbf{P}]\delta\mathbf{x} \leq 0 \tag{4.7}$$

Collecting terms, we get:

$$\delta\mathbf{x}^T[-2\mathbf{P} + \mathbf{PDR} + \mathbf{RDP} - 2\mathbf{PDP} + \beta\mathbf{P}]\delta\mathbf{x} \leq 0 \tag{4.8}$$

We can rewrite (4.8) as a quadratic form over a block matrix, as follows:

$$\begin{bmatrix}\delta\mathbf{x}^T & \delta\mathbf{x}^T\end{bmatrix}\begin{bmatrix}(\beta-2)\mathbf{P} & \mathbf{RDP} \\ \mathbf{PDR} & -2\mathbf{PDP}\end{bmatrix}\begin{bmatrix}\delta\mathbf{x} \\ \delta\mathbf{x}\end{bmatrix} \leq 0 \tag{4.9}$$

Now the question becomes, when is (4.9) satisfied? One way to ensure that (4.9) is satisfied is to ensure that the associated block matrix is always (i.e for all $\mathbf{D}$) negative semi-definite. In that case the inequality will hold over *all* possible vectors, not just $\begin{bmatrix}\delta\mathbf{x} & \delta\mathbf{x}\end{bmatrix}^T$. In other words, the question is now what constraints on the sub-matrices $\mathbf{P}, \mathbf{D}$ and $\mathbf{R}$ ensure that:

$$\forall\mathbf{y} \in \mathbb{R}^{2n}, \quad \mathbf{y}^T\begin{bmatrix}(2-\beta)\mathbf{P} & -\mathbf{RDP} \\ -\mathbf{PDR} & 2\mathbf{PDP}\end{bmatrix}\mathbf{y} \geq 0 \tag{4.10}$$

Note that we have multiplied both sides of the inequality by a minus sign. But this is nothing but the definition of a positive semi-definite matrix. Using the Schur complement [Gallier et al., 2020](Proposition 2.1), we know that the block matrix is positive semi-definite iff $\mathbf{PDP} \succ 0$ and:

$$(2-\beta)\mathbf{P} - \mathbf{RDP}(2\mathbf{PDP})^{-1}\mathbf{PDR} =$$
$$(2-\beta)\mathbf{P} - \frac{1}{2}(\mathbf{RDR}) \succeq (2-\beta)\mathbf{P} - \frac{g}{2}(\mathbf{RR}) \succeq 0$$

We continue by setting $\mathbf{P} = \gamma^2\mathbf{RR}$ with $\gamma^2 = \frac{g}{2(2-\beta)}$, so that the above inequality is satisfied. At this point, we have shown that if $\mathbf{W}$ can be written as:

$$\mathbf{W} = \mathbf{R} - \gamma^2 \mathbf{R}\mathbf{R}$$

then (4.1) is contracting in metric $\mathbf{M} = \gamma^2 \mathbf{R}\mathbf{R}$. What remains to be shown is that if the condition:

$$g\mathbf{W} - \mathbf{I} \prec 0$$

Is satisfied, then this implies the existence of an $\mathbf{R}$ such that the above is true. To show that this is indeed the case, assume that:

$$\frac{1}{4\gamma^2}\mathbf{I} - \mathbf{W} \succeq 0$$

Substituting in the definition of γ, this is just the statement that:

$$\frac{2(2-\beta)}{4g}\mathbf{I} - \mathbf{W} \succeq 0$$

Setting $\beta = 2\lambda > 0$, this yields:

$$(1-\lambda)\mathbf{I} \succeq g\mathbf{W}$$

Since $\mathbf{W}$ is symmetric by assumption, we have the eigendecomposition:

$$\frac{1}{4\gamma^2}\mathbf{I} - \mathbf{W} = \mathbf{V}(\frac{1}{4\gamma^2}\mathbf{I} - \mathbf{\Lambda})\mathbf{V}^T$$

where $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ and $\mathbf{\Lambda}$ is a diagonal matrix containing the eigenvalues of $\mathbf{W}$. Denote the symmetric square-root of this expression as $\mathbf{S}$:

$$\mathbf{S} = \mathbf{V}\sqrt{(\frac{1}{4\gamma^2}\mathbf{I} - \mathbf{\Lambda})}\mathbf{V}^T = \mathbf{S}^T$$

Which implies that:

$$\frac{1}{4\gamma^2}\mathbf{I} - \mathbf{W} = \mathbf{S}^T\mathbf{S}$$

We now define $\mathbf{R}$ in terms of $\mathbf{S}$ as follows:

$$\mathbf{R} = \frac{1}{\gamma}\mathbf{S} + \frac{1}{2\gamma^2}\mathbf{I}$$

Which means that:

$$\frac{1}{4\gamma^2}\mathbf{I} - \mathbf{W} = (\gamma\mathbf{R} - \frac{1}{2\gamma}\mathbf{I})(\gamma\mathbf{R} - \frac{1}{2\gamma}\mathbf{I})$$

Expanding out the right side, we get:

$$\frac{1}{4\gamma^2}\mathbf{I} - \mathbf{W} = \gamma^2\mathbf{R}\mathbf{R} - \mathbf{R} + \frac{1}{4\gamma^2}\mathbf{I}$$

Subtracting $\frac{1}{4\gamma^2}\mathbf{I}$ from both sides yields:

$$\mathbf{W} = \mathbf{R} - \gamma^2\mathbf{R}\mathbf{R}$$

As desired.

□

4.C.4 Proof of Theorem 5

Theorem 3. *If there exists positive diagonal matrices* $\mathbf{P}_1$ *and* $\mathbf{P}_2$*, as well as* $\mathbf{Q} = \mathbf{Q}^T \succ 0$ *such that*

$$\mathbf{W} = -\mathbf{P}_1\mathbf{Q}\mathbf{P}_2$$

then (4.1) is contracting in metric $\mathbf{M} = (\mathbf{P}_1\mathbf{Q}\mathbf{P}_1)^{-1}$*.*

Proof. Consider again a differential Lyapunov function:

$$V = \delta\mathbf{x}^T\mathbf{M}\delta\mathbf{x}$$

the time derivative is equal to:

$$\dot{V} = -2V + \delta\mathbf{x}^T\mathbf{M}\mathbf{W}\mathbf{D}\delta\mathbf{x}$$

Substituting in the definitions of $\mathbf{W}$ and $\mathbf{M}$, we get:

$$\dot{V} = -2V - \delta\mathbf{x}^T\mathbf{P}_1^{-1}\mathbf{P}_2\mathbf{D}\delta\mathbf{x} \leq -2V$$

Therefore V converges exponentially to zero.

□

4.C.5 Proof of Theorem 6

Theorem 4. *If $g\mathbf{W} - \mathbf{I}$ is triangular and Hurwitz, then (4.1) is contracting in a diagonal metric.*

Note that in the case of a triangular weight matrix, the system (4.1) may be seen as a feedforward (i.e hierarchical) network. Therefore, this result follows from the combination properties of contracting systems. However, our proof provides a means of explicitly finding a metric for this system.

Proof. Without loss of generality, assume that $\mathbf{W}$ is lower triangular. This implies that $W_{ij} = 0$ if $i \leq j$. Now consider the generalized Jacobian:

$$\mathbf{F} = -\mathbf{I} + \mathbf{\Gamma}\mathbf{W}\mathbf{D}\mathbf{\Gamma}^{-1}$$

with $\mathbf{\Gamma}$ diagonal and $\Gamma_i = \epsilon^i$ where $\epsilon > 0$. Because $\mathbf{\Gamma}$ is diagonal, the generalized Jacobian is equal to:

$$\mathbf{F} = -\mathbf{I} + \mathbf{\Gamma}\mathbf{W}\mathbf{\Gamma}^{-1}\mathbf{D}$$

Now note that:

$$(\mathbf{\Gamma}\mathbf{W}\mathbf{\Gamma}^{-1})_{ij} = \epsilon^i W_{ij}\epsilon^{-j} = W_{ij}\epsilon^{i-j}$$

Where $i \leq j$, we have $W_{ij} = 0$ by assumption. Therefore, the only nonzero entries are where $i \geq j$. This means that by making ϵ arbitrarily small, we can make $\mathbf{\Gamma}\mathbf{W}\mathbf{\Gamma}^{-1}$

approach a diagonal matrix with W_{ii} along the diagonal. Therefore, if:

$$\max_i gW_{ii} - 1 < 0$$

the nonlinear system is contracting. Since $\mathbf{W}$ is triangular, W_{ii} are the eigenvalues of $\mathbf{W}$, meaning that this condition is equivalent to $g\mathbf{W} - \mathbf{I}$ being Hurwitz.

□

4.C.6 Proof of Theorem 7

Theorem 5. *If there exists a positive diagonal matrix* $\mathbf{P}$ *such that:*

$$g^2\mathbf{W}^T\mathbf{P}\mathbf{W} - \mathbf{P} \prec 0$$

then (4.1) is contracting in metric $\mathbf{P}$*.*

Note that this is equivalent to the discrete-time diagonal stability condition developed in [Revay and Manchester, 2020], for a constant metric. Note also that when $\mathbf{M} = \mathbf{I}$, Theorem 7 is identical to checking the maximum singular value of $\mathbf{W}$, a previously established condition for stability of (4.1). However a much larger set of weight matrices are found via the condition when $\mathbf{M} = \mathbf{P}$ instead.

Proof. Consider the generalized Jacobian:

$$\mathbf{F} = \mathbf{P}^{1/2}\mathbf{J}\mathbf{P}^{-1/2} = -\mathbf{I} + \mathbf{P}^{1/2}\mathbf{W}\mathbf{P}^{-1/2}\mathbf{D}$$

where $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}_{ii} = \frac{d\phi_i}{dx_i} \geq 0$. Using the subadditivity of the matrix measure μ_2 of the generalized Jacobian we get:

$$\mu_2(\mathbf{F}) \leq -1 + \mu_2(\mathbf{P}^{1/2}\mathbf{W}\mathbf{P}^{-1/2}\mathbf{D})$$

Now using the fact that $\mu_2(\cdot) \leq ||\cdot||_2$ we have:

$$\mu_2(\mathbf{F}) \leq -1 + ||\mathbf{P}^{1/2}\mathbf{W}\mathbf{P}^{-1/2}\mathbf{D})||_2 \leq -1 + g||\mathbf{P}^{1/2}\mathbf{W}\mathbf{P}^{-1/2}||_2$$

Using the definition of the 2-norm, imposing the condition $\mu_2(\mathbf{F}) \leq 0$ may be written:

$$g^2\mathbf{W}^T\mathbf{P}\mathbf{W} - \mathbf{P} \prec 0$$

which completes the proof.

□

4.C.7 Proof of Theorem 8

Theorem 6. *Let* $\mathbf{D}$ *be a positive, diagonal matrix with* $D_{ii} = \frac{d\phi_i}{dx_i}$*, and let* $\mathbf{P}$ *be an arbitrary, positive diagonal matrix. If:*

$$(g\mathbf{W} - \mathbf{I})\mathbf{P} + \mathbf{P}(g\mathbf{W}^T - \mathbf{I}) \preceq -c\mathbf{P}$$

and

$$\dot{\mathbf{D}} - cg^{-1}\mathbf{D} \preceq -\beta\mathbf{D}$$

for $c, \beta > 0$*, then (4.1) is contracting in metric* $\mathbf{D}$ *with rate* β*.*

Proof. Consider the differential, quadratic Lyapunov function:

$$V = \delta\mathbf{x}^T\mathbf{P}\mathbf{D}\delta\mathbf{x}$$

where $\mathbf{D} \succ 0$ is as defined above. The time derivative of V is:

$$\dot{V} = \delta\mathbf{x}^T\mathbf{P}\dot{\mathbf{D}}\delta\mathbf{x} + \delta\mathbf{x}^T(-2\mathbf{P}\mathbf{D} + \mathbf{P}\mathbf{D}\mathbf{W}\mathbf{D} + \mathbf{D}\mathbf{W}^T\mathbf{D}\mathbf{P})\delta\mathbf{x}$$

The second term on the right can be factored as:

$$\delta\mathbf{x}^T(-2\mathbf{PD}+\mathbf{PDWD}+\mathbf{DW}^T\mathbf{DP})\delta\mathbf{x} =$$
$$\delta\mathbf{x}^T\mathbf{D}(-2\mathbf{PD}^{-1}+\mathbf{PW}+\mathbf{W}^T\mathbf{P})\mathbf{D}\delta\mathbf{x} \leq$$
$$\delta\mathbf{x}^T\mathbf{D}(-2\mathbf{P}g^{-1}+\mathbf{PW}+\mathbf{W}^T\mathbf{P})\mathbf{D}\delta\mathbf{x} =$$
$$\delta\mathbf{x}^T\mathbf{D}[\mathbf{P}(\mathbf{W}-g^{-1}\mathbf{I})+(\mathbf{W}^T-g^{-1}\mathbf{I})\mathbf{P}]\mathbf{D}\delta\mathbf{x} \leq$$
$$-cg^{-1}\delta\mathbf{x}^T\mathbf{PD}^2\delta\mathbf{x}$$

where the last inequality was obtained by substituting in the first assumption above. Combining this with the expression for $\dot{V}$, we have:

$$\dot{V} \leq \delta\mathbf{x}^T\mathbf{P}\dot{\mathbf{D}}\delta\mathbf{x} - cg^{-1}\delta\mathbf{x}^T\mathbf{PD}^2\delta\mathbf{x}$$

Substituting in the second assumption, we have:

$$\dot{V} \leq \delta\mathbf{x}^T\mathbf{P}(\dot{\mathbf{D}}-cg^{-1}\mathbf{D}^2)\delta\mathbf{x} \leq -\beta\delta\mathbf{x}^T\mathbf{PD}\delta\mathbf{x} = -\beta V$$

and thus V converges exponentially to 0 with rate β. □

4.C.8 Proof of Theorem 9

Theorem 7. *Satisfaction of the condition*

$$g\mathbf{W}_{sym}-\mathbf{I} \prec 0$$

*is **NOT** sufficient to show global contraction of the general nonlinear RNN (4.1) in any constant metric. High levels of antisymmetry in* $\mathbf{W}$ *can make it impossible to find such a metric, which we demonstrate via a* 2×2 *counterexample of the form*

$$\mathbf{W} = \begin{bmatrix} 0 & -c \\ c & 0 \end{bmatrix}$$

with $c \geq 2$.

Note that $g\mathbf{W}_{sym} - \mathbf{I} = g\frac{\mathbf{W}+\mathbf{W}^T}{2} - \mathbf{I} \prec 0$ is equivalent to the condition for contraction of the system with *linear* activation in the identity metric.

The main intuition behind this counterexample is that high levels of antisymmetry can prevent a constant metric from being found in the nonlinear system. This is because $\mathbf{D}$ is a diagonal matrix with values between 0 and 1, so the primary functionality it can have in the symmetric part of the Jacobian is to downweight the outputs of certain neurons selectively. In the extreme case of all 0 or 1 values, we can think of this as selecting a subnetwork of the original network, and taking each of the remaining neurons to be single unit systems receiving input from the subnetwork. For a given static configuration of $\mathbf{D}$ (think linear gains), this is a hierarchical system that will be stable if the subnetwork is stable. But as $\mathbf{D}$ can evolve over time when a nonlinearity is introduced, we would need to find a constant metric that can serve completely distinct hierarchical structures simultaneously - which is not always possible.

Put in terms of matrix algebra, D can zero out columns of $\mathbf{W}$, but not their corresponding rows. So for a given weight pair w_{ij}, w_{ji}, which has entry in $\mathbf{W}_{sym} = \frac{w_{ij}+w_{ji}}{2}$, if $D_i = 0$ and $D_j = 1$, the i, j entry in $(\mathbf{WD})_{sym}$ will be guaranteed to have lower magnitude if the signs of w_{ij} and w_{ji} are the same, but guaranteed to have higher magnitude if the signs are different. Thus if the linear system would be stable based on magnitudes alone $\mathbf{D}$ poses no real threat, but if the linear system requires antisymmetry to be stable, $\mathbf{D}$ can make proving contraction quite complicated (if possible at all).

Proof. The nonlinear system is globally contracting in a *constant* metric if there exists a symmetric, positive definite $\mathbf{M}$ such that the symmetric part of the Jacobian for the system, $(\mathbf{MWD})_{sym} - \mathbf{M}$ is negative definite uniformly. Therefore $(\mathbf{MWD})_{sym} - \mathbf{M} \prec 0$ must hold for all possible $\mathbf{D}$ if $\mathbf{M}$ is a constant metric the system *globally* contracts in with any allowed activation function, as some combination of settings to obtain a particular $\mathbf{D}$ can always be found.

Thus to prove the main claim, we present here a simple 2-neuron system that is

contracting in the identity metric with linear activation function, but can be shown to have no $\mathbf{M}$ that simultaneously satisfies the $(\mathbf{MWD})_{sym} - \mathbf{M} \prec 0$ condition for two different possible $\mathbf{D}$ matrices.

To begin, take

$$\mathbf{W} = \begin{bmatrix} 0 & -2 \\ 2 & 0 \end{bmatrix}$$

Note that any off-diagonal magnitude ≥ 2 would work, as this is the point at which $\frac{1}{2}$ of one of the weights (found in $\mathbf{W}_{sym}$ when the other is zeroed) will have magnitude too large for $(\mathbf{WD})_{sym} - \mathbf{I}$ to be stable.

Looking at the linear system, we can see it is contracting in the identity because

$$\mathbf{W}_{sym} - \mathbf{I} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \prec 0$$

Now consider $(\mathbf{MWD})_{sym} - \mathbf{M}$ with $\mathbf{D}$ taking two possible values of

$$\mathbf{D}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad and \quad \mathbf{D}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

We want to find some symmetric, positive definite $\mathbf{M} = \begin{bmatrix} a & m \\ m & b \end{bmatrix}$ such that $(\mathbf{MWD}_1)_{sym} - \mathbf{M}$ and $(\mathbf{MWD}_2)_{sym} - \mathbf{M}$ are both negative definite.

Working out the matrix multiplication, we get

$$(\mathbf{MWD}_1)_{sym} - \mathbf{M} = \begin{bmatrix} 2m - a & b - m \\ b - m & -b \end{bmatrix}$$

and

$$(\mathbf{MWD}_2)_{sym} - \mathbf{M} = \begin{bmatrix} -a & -(a+m) \\ -(a+m) & -2m-b \end{bmatrix}$$

We can now check necessary conditions for negative definiteness on these two matrices, as well as for positive definiteness on $\mathbf{M}$, to try to find an $\mathbf{M}$ that will satisfy all these conditions simultaneously. In this process we will reach a contradiction, showing that no such $\mathbf{M}$ can exist.

A necessary condition for positive definiteness in a real, symmetric $n \times n$ matrix $\mathbf{X}$ is $x_{ii} > 0$, and for negative definiteness $x_{ii} < 0$. Another well known necessary condition for definiteness of a real symmetric matrix is $|x_{ii} + x_{jj}| > |x_{ij} + x_{ji}| = 2|x_{ij}| \ \forall i \neq j$. See [Weisstein] for more info on these conditions.

Thus we will require a and b to be positive, and can identify the following conditions as necessary for our 3 matrices to all meet the requisite definiteness conditions:

$$2m < a \tag{4.11}$$

$$-2m < b \tag{4.12}$$

$$|2m - (a+b)| > 2|b-m| \tag{4.13}$$

$$|-2m - (a+b)| > 2|a+m| \tag{4.14}$$

Note that the necessary condition for $\mathbf{M}$ to be PD, $a + b > 2|m|$, is not listed, as it is automatically satisfied if (4.11) and (4.12) are.

It is easy to see that if $m = 0$, conditions (4.13) and (4.14) will result in the contradictory conditions $a > b$ and $b > a$ respectively, so we will require a metric with off-diagonal elements. To make the absolute values easier to deal with, we will check $m > 0$ and $m < 0$ cases independently.

First we take $m > 0$. By condition (4.11) we must have $a > 2m$, so between that and knowing the signs of all unknowns are positive, we can reduce many of the absolute values. Condition (4.13) becomes $a + b - 2m > |2b - 2m|$, and condition (4.14) becomes $a + b + 2m > 2a + 2m$, which is equivalent to $b > a$. If $b > a$ we must also have $b > m$, so condition (4.13) further reduces to $a + b - 2m > 2b - 2m$, which is equivalent to $a > b$. Therefore we have again reached contradictory conditions.

A very similar approach can be applied when $m < 0$. Using condition (4.12) and the known signs we reduce condition (4.13) to $2|m| + a + b > 2b + 2|m|$, i.e. $a > b$. Meanwhile condition (4.14) works out to $a + b - 2|m| > 2a - 2|m|$, i.e. $b > a$.

Therefore it is impossible for a single constant $\mathbf{M}$ to accommodate both $\mathbf{D}_1$ and $\mathbf{D}_2$, so that no constant metric can exist for $\mathbf{W}$ to be contracting in when a nonlinearity is introduced that can possibly have derivative reaching both of these configurations. One real world example of such a nonlinearity is ReLU. Given a sufficiently high negative input to one of the units and a sufficiently high positive input to the other, $\mathbf{D}$ can reach one of these configurations. The targeted inputs could then flip at any time to reach the other configuration.

An additional condition we could impose on the activation function is to require it to be a strictly increasing function, so that the activation function derivative can never actually reach 0. We will now show that a very similar counterexample applies in this case, by taking

$$\mathbf{D}_{1*} = \begin{bmatrix} 1 & 0 \\ 0 & \epsilon \end{bmatrix} \quad \textit{and} \quad \mathbf{D}_{2*} = \begin{bmatrix} \epsilon & 0 \\ 0 & 1 \end{bmatrix}$$

Note here that the $\mathbf{W}$ used above produced a $(\mathbf{WD})_{sym} - \mathbf{I}$ that just barely avoided being negative definite with the original $\mathbf{D}_1$ and $\mathbf{D}_2$, so we will have to increase the values on the off-diagonals a bit for this next example. In fact anything with magnitude larger than 2 will have some $\epsilon > 0$ that will cause a constant metric to be impossible, but for simplicity we will now take

$$\mathbf{W}_* = \begin{bmatrix} 0 & -4 \\ 4 & 0 \end{bmatrix}$$

Note that with $\mathbf{W}_*$, even just halving one of the off-diagonals while keeping the other intact will produce a $(\mathbf{WD})_{sym} - \mathbf{I}$ that is not negative definite. Anything less than halving however will keep the identity metric valid. Therefore, we expect that taking ϵ in $\mathbf{D}_{1*}$ and $\mathbf{D}_{2*}$ to be in the range $0.5 \geq \epsilon > 0$ will also cause issues when trying to obtain a constant metric.

We will now actually show via a similar proof to the above that $\mathbf{M}$ is impossible to find for $\mathbf{W}_*$ when $\epsilon \leq 0.5$. This result is compelling because it not only shows that ϵ does not need to be a particularly small value, but it also drives home the point about antisymmetry - the larger in magnitude the antisymmetric weights are, the larger the ϵ where we will begin to encounter problems.

Working out the matrix multiplication again, we now get

$$(\mathbf{MW}_*\mathbf{D}_{1*})_{sym} - \mathbf{M} = \begin{bmatrix} 4m - a & 2b - m - 2a\epsilon \\ b - m - 2a\epsilon & -4m\epsilon - b \end{bmatrix}$$

and

$$(\mathbf{MW}_*\mathbf{D}_{2*})_{sym} - \mathbf{M} = \begin{bmatrix} 4m\epsilon - a & -(2a + m - 2b\epsilon) \\ -(2a + m - 2b\epsilon) & -4m - b \end{bmatrix}$$

Resulting in two new main necessary conditions:

$$|4m - a - b - 4m\epsilon| > 2|2b - m - 2a\epsilon| \tag{4.15}$$

$$|4m\epsilon - a - b - 4m| > 2|2a + m - 2b\epsilon| \tag{4.16}$$

As well as new conditions on the diagonal elements:

$$4m - a < 0 \tag{4.17}$$

$$-4m - b < 0 \tag{4.18}$$

We will now proceed with trying to find a, b, m that can simultaneously meet all conditions, setting $\epsilon = 0.5$ for simplicity.

Looking at $m = 0$, we can see again that $\mathbf{M}$ will require off-diagonal elements, as condition (4.15) is now equivalent to the condition $a + b > |4b - 2a|$ and condition (4.16) is similarly now equivalent to $a + b > |4a - 2b|$.

Evaluating these conditions in more detail, if we assume $4b > 2a$ and $4a > 2b$, we can remove the absolute value and the conditions work out to the contradicting $3a > 3b$ and $3b > 3a$ respectively. As an aside, if $\epsilon > 0.5$, this would no longer be the case, whereas with $\epsilon < 0.5$, the conditions would be pushed even further in opposite directions.

If we instead assume $2a > 4b$, this means $4a > 2b$, so the latter condition would still lead to $b > a$, contradicting the original assumption of $2a > 4b$. $2b > 4a$ causes a contradiction analogously. Trying $4b = 2a$ will lead to the other condition becoming $b > 2a$, once again a contradiction. Thus a diagonal $\mathbf{M}$ is impossible

So now we again break down the conditions into $m > 0$ and $m < 0$ cases, first looking at $m > 0$. Using condition (4.17) and knowing all unknowns have positive sign,

condition (4.15) reduces to $a+b-2m > |4b-2(a+m)|$ and condition (4.16) reduces to $a+b+2m > |4a-2(b-m)|$. This looks remarkably similar to the $m=0$ case, except now condition (4.15) has $-2m$ added to both sides (inside the absolute value), and condition (4.16) has $2m$ added to both sides in the same manner. If $4b > 2(a+m)$ the $-2m$ term on each side will simply cancel, and similarly if $4a > 2(b-m)$ the $+2m$ terms will cancel, leaving us with the same contradictory conditions as before.

Therefore we check $2(a+m) > 4b$. This rearranges to $2a > 2(2b-m) > 2(b-m)$, so that from condition (4.16) we get $b > a$. Subbing condition (4.17) in to $2(a+m) > 4b$ gives $8b < 4a+4m < 5a$ i.e. $b < \frac{5}{8}a$, a contradiction. The analogous issue arises if trying $2(b-m) > 4a$. Trying $2(a+m) = 4b$ gives $m = 2b-a$, which in condition (4.16) results in $5b-a > |6a-6b|$, while in condition (4.17) leads to $5a > 8b$, so (4.16) can further reduce to $5b-a > 6a-6b$ i.e. $11b > 7a$. But $b > \frac{7}{11}a$ and $b < \frac{5}{8}a$ is a contradiction. Thus there is no way for $m > 0$ to work.

Finally, trying $m < 0$, we now use condition (4.18) and the signs of the unknowns to reduce condition (4.15) to $a+b+2|m| > |4b-2(a-|m|)|$ and condition (4.16) to $a+b-2|m| > |4a-2(b+|m|)|$. These two conditions are clearly directly analogous to in the $m > 0$ case, where b now acts as a with condition (4.18) being $b > 4|m|$. Therefore the proof is complete. □

4.D Sparse Combo Net Details

Here we provide comprehensive information on the methodology and results for Sparse Combo Net, including some supplementary experimental results. See the Appendix table of contents for a guide to this section.

4.D.1 Extended Methods

Initialization and Training

As described in the main text, the nonlinear RNN weights for Sparse Combo Net were randomly generated based on given sparsity and entry magnitude settings, and then confirmed to meet the Theorem 3 condition (or discarded if not). For a sparsity level of x and a magnitude limit of y, each subnetwork $\mathbf{W}$ was generated by drawing uniformly from between $-y$ and y with $x\%$ density using scipy.sparse.random, and then zeroing out the diagonal entries. For various potential x and y settings, we quantified both the likelihood that a generated $\mathbf{W}$ would satisfy Theorem 3, and the resulting network performance. Of course this is also dependent on subnetwork size, as larger subnetworks enable greater sparsity. The information we have obtained so far is documented in Section 4.D.2, in particular 4.D.2.

In training the linear connections between the described nonlinear RNN subnetworks, we constrained the matrix $\mathbf{B}$ in (4.3) to reflect underlying modularity assumptions. In particular, we only train the off-diagonal blocks of $\mathbf{B}$ and mask the diagonal blocks. We do this to maintain the interpretation of $\mathbf{L}$ as the matrix containing the connection weights *between* different modules, as diagonal blocks would correspond to self-connections. Furthermore, we only train the lower-triangular blocks of $\mathbf{B}$ while masking the others, to increase training speed.

To obtain the subnetwork RNN metrics necessary for training these linear connections, scipy.integrate.quad was used with default settings to solve for $\mathbf{M}$ in the equation $-\mathbf{I} = \mathbf{M}\mathbf{W} + \mathbf{W}^T\mathbf{M}$, as described in the main text. This was done by integrating $e^{\mathbf{W}^T t}\mathbf{Q}e^{\mathbf{W}t}dt$ from 0 to ∞. For efficiency reasons, and due to the guaranteed existence of a diagonal metric in the case of Theorem 3, integration was only performed to solve for the diagonal elements of $\mathbf{M}$. Therefore a check was added prior to training to confirm that the initialized network indeed satisfied Theorem 3 with metric $\mathbf{M}$. However, it was never triggered by our initialization method.

Initial training hyperparameter tuning was done primarily with 10×16 combination networks on the permuted seqMNIST task, starting with settings based on existing

literature on this task, and verifying promising settings using a 15×16 network (Table 4.D.2). Initialization settings were held the same throughout, as was later done for the size comparison trials (described in Section 4.D.2).

Once hyperparameters were decided upon, the trials reported on in the main text began. Most of these experiments were also done on permuted seqMNIST, where we characterized performance of networks with different sizes and sparsity levels/entry magnitudes. When we moved to the sequential CIFAR10 task, we began by simply training with the same best settings that were found from these experiments. The results of all attempted trials are reported in Section 4.D.4.

Unless specified otherwise, all networks reported on in the main text were trained for 150 epochs, using an Adam optimizer with initial learning rate 1e-3 and weight decay 1e-5. The learning rate was cut to 1e-4 after 90 epochs and to 1e-5 after 140. After identifying the most promising settings, we ran repetitions trials on the best networks for 200 epochs with learning rate cuts after epochs 140 and 190.

Code and Datasets

All Sparse Combo Nets described in the main text were trained using a single GPU on Google Colab. Code to replicate all experiments can be found here: https://colab.research.google.com/drive/1JCT5OMgaMVK_Xh8BDFNRrEsyF0Ojvg10?usp=sharing

Runtime for the best performing architecture settings on the sequential CIFAR10 task was $\sim$ 24 hours. A Colab Pro+ account was used to limit timeouts and prioritize GPU access.

The datasets we used for our tasks were MNIST and CIFAR10, downloaded via PyTorch. MNIST is a handwritten digits classification task, consisting of 60,000 training images and 10,000 testing images (each 28x28 and grayscale). It is made available under the terms of the Creative Commons Attribution-Share Alike 3.0 license. CIFAR10 is a dataset of 32x32 color images, split evenly among the following 10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. It also contains 60,000/10,000 training/test images, and is distributed under the MIT License.

As mentioned in the main text, we presented these images to our networks pixel by pixel. In the case of MNIST, we used an additional modification to the dataset by permuting the pixels in a randomly determined (but fixed across the dataset) way. The use of these datasets is included in the above link to our code.

Extended results information begins on the next page.

4.D.2 Extended Results

In this subsection, we describe additional results we did not get to in the main text, related to scalability, modularity, sparsity, and repeatability. We also add some further discussion of these results, along with more detailed information on the respective experimental set ups.

Network Size and Modularity Comparison

For the Sparse Combo Net specifically we had additional experiments on architecture size and unit distribution besides what was depicted in the main text. The results of these supplemental experiments both replicated the observed effect in Figure 4.3.3 of network size instead using subnetworks with 16 units each as well as higher density (Figure 4.D.1A), and evaluated how task performance varies with modularity of a network fixed to have 352 total units (Figure 4.D.1B). In the modularity experiment we observed an inverse U shape, with poor performance of a 1×352 net and an 88×4 net, and best performance from a 44×8 net. Note that this experiment compared similar sparsity levels across the different subnetwork sizes. In practice we can achieve better performance with larger subnetworks by leveraging sparsity in a way not possible in smaller subnetworks. These additional experiments are now described in more detail below.

For the initial round of size comparison trials using subnetworks of 16 units each (Figure 4.D.1A), the nonlinear RNN weights were set by drawing uniformly from between -0.4 and 0.4 with 40% density using scipy.sparse.random, and then zeroing out the diagonal entries. These settings were chosen because they resulted in $\sim 1\%$ of 16 by 16 weight matrices meeting the Theorem 3 condition. During initialization only the matrices meeting this condition were kept, finishing when the desired number of component RNNs had been set - producing a block diagonal $\mathbf{W}$ like pictured in Figure 4.3.2A. This same initialization process was used throughout our experiments. In later experiments we vary the density and magnitude settings.

For the size experiments, we held static the number of units and initialization

settings for each component RNN, and tested the effect of changing the number of components in the combination network. 1, 3, 5, 10, 15, 20, 22, 25, and 30 components were tested in this experiment (Figure 4.D.1A). Increasing the number of components initially lead to great improvements in test accuracy, but had diminishing returns - test accuracy consistently hit $\sim 93\%$ with a large enough number of subnetworks, but neither loss nor accuracy showed meaningful improvement past the 22×16 network. Interestingly, early training loss and accuracy became substantially worse once the number of components increased past a certain point, falling from 70% to 43% epoch 1 test accuracy between the 22×16 and 30×16 networks. The complete set of results can be found in Table 4.D.3.

Note that the size experiment described in the main text (Figure 4.3.3A) was a repetition of this original experiment, but now using 32 unit subnetworks with the best performing sparsity settings. The results for the repetition can be found in Table 4.D.5.

To better understand how the modularity of the combination networks affects performance, the next experiment held the number of total units constant at 352, selected due to the prior success of the 22×16 network, and tested different allocations of these units amongst component RNNs. Thus 1×352, 11×32, 44×8, and 88×4 networks were trained to compare against the 22×16 (Figure 4.D.1B). Increasing the modularity improved performance to a point, with the 44×8 network resulting in final test accuracy of 94.44%, while conversely the 11×32 resulted in decreased test accuracy. However, the 88×4 network was unable to learn, and a 352×1 network would theoretically just be a scaled linear anti-symmetric network.

Because larger networks require different sparsity settings to meet the Theorem 3 condition, these were not held constant between trials in the modularity comparison experiment (Figure 4.D.1B), but rather selected in the same way between trials - looking for settings that keep density and scalar balanced and result in $\sim 1\%$ of the matrices meeting the condition. The scalar was applied after sampling non-zero entries from a uniform distribution between -1 and 1. The resulting settings were 7.5% density and 0.077 scalar for 352 unit component RNN, 26.5% density and 0.27

scalar for 32 unit component RNN, 60% density and 0.7 scalar for 8 unit component RNN, and 100% density and 1.0 scalar for 4 unit component RNN. The complete set of results for the modularity experiment can be found in Table 4.D.4.

Sparsity Settings Comparison

Density and scalar settings for the component nonlinear RNNs were initially chosen for each network size using the percentage of random networks that met the Theorem 3 condition. For scalar s, a component network would have non-zero entries sampled uniformly between $-s$ and s.

When we began experimenting with sparsity in the initialization (after seeing the large performance difference in Figure 4.3.4), we split the previously described scalar setting into two different scalars - one applied before a random matrix was checked against the Theorem 3 condition, and one applied after a matrix was selected. Of course the latter must be ≤ 1 to guarantee stability is preserved. The scalar was separated out after we noticed that at 5% density, random 32 by 32 weight matrices met the condition roughly 1% of the time whether the scalar was 10 or 100000 - $\sim$ 85% of sampled matrices using scalar 10 would continue to meet the condition even if multiplied by a factor of 10000. Therefore we wanted a mechanism that could bias selection towards matrices that are stable due to their sparsity and not due to magnitude constraints, while still keeping the elements to a reasonable size for training purposes.

Ultimately, both sparsity and magnitude had a clear effect on performance (Figure 4.D.2). Increases in both had a positive correlation with accuracy and loss through most parameters tested. Best test accuracy overall was 96.79%, which was obtained by both a 16×32 network with 5% density and entries between -5 and 5, and a 16×32 network with 3.3% density and entries between -6 and 6. The latter also achieved the best epoch 1 test accuracy observed of 86.79%. Thus we chose to go with these settings for our extended training repetitions on permuted seqMNIST.

It is also worth noting that upon investigation of the subnetwork weight matrices across these trials, the sparser networks had substantially lower maximum eigenvalue

of $|\mathbf{W}|$, suggesting that stronger stability can actually correlate with *improved* performance on sequential tasks. This could be due to a mechanism such as that described in [Radhakrishnana et al., 2020].

Results from the initial trials of different sparsity levels across different network sizes can be found in Table 4.D.6. Results from the more thorough testing of different sparsity levels and magnitudes in a 16×32 network can be found in Table 4.D.7.

Repeatability Tests

Here we give additional details on the repeatability tests described in the main text, as well as introducing some additional results using other hyperparameters (done after the initial SOTA-setting experiments) on sequential CIFAR10.

To further improve performance once network settings were explored on permuted seqMNIST, an extended training run was tested on the best performing option. Settings were kept the same as above using a 3.3% density 16×32 network, except training now ran for over 200 epochs, with just a single learning rate cut occurring after epoch 200 (exact number of epochs varied based on runtime limit). This experiment was repeated four times and resulted in 96.94% best test accuracy, as described in the main text (Figure 4.D.3). Table 4.D.8 reports additional details on these trials.

As mentioned, we also characterized Sparse Combo Net performance on the more challenging CIFAR10 sequential image classification task, across a greater number of trials. Ten trials were run for 200 epochs with learning rate scaled after epochs 140 and 190. All other settings were the same as for the permuted seqMNIST repeatability trials. As reported in the main text, the mean test accuracy observed was 64.72%, with variance 0.406, and range 63.73%-65.72%. Training loss and test accuracy over the course of learning are plotted for all ten trials in Figure 4.D.4, and exact numbers for each trial are provided in Table 4.D.10.

As we encountered difficulties with Colab GPU assignment when we started working on these repetitions, we also trained nine networks over a smaller number of epochs (Figure 4.D.5). These networks were trained using the 150 epoch paradigm previously described, although only four of the nine completed training within the 24 hour

runtime limit. Complete results for these trials can be found in Table 4.D.11. Mean performance among the shorter training trials was 62.82% test accuracy with variance 0.95.

Prior to beginning the repeatability experiments on seqCIFAR10, we explored alternative hyperparameters on this task (Table 4.D.9). While we ultimately ended up using the same hyperparameters as for permuted seqMNIST, these results further support the robustness of our architecture.

To complete our benchmarking table, we also ran a single 150 epoch trial of our best network settings on the sequential MNIST task. Test accuracy over the course of training for this trial is depicted in Figure 4.D.6.

Scalability Pilot: Feedback Sparsity Tests

Not only did the 16×32 Sparse Combo Net with 3.3% density achieve test accuracy sequential CIFAR10 that is the highest to date for a provably stable RNN, it was higher than the 1 million parameter CKConv network, which set a recent SOTA for permuted seqMNIST accuracy (Table 4.3.1). Our network has $130,000$ trainable parameters by comparison. Thus we achieve very impressive results given the characteristics of our architecture.

However, the results of the network size experiments do raise some concern about the scalability of the approach. Here we provide pilot results suggesting that this issue can likely be addressed via the introduction of sparsity in the linear inter-subnetwork connectivity matrix $\mathbf{L}$. While all results in the main text use all-to-all negative feedback, we have early results suggesting that in larger networks, fewer negative feedback connections may perform better, thus preventing performance saturation with scaling.

Below are the results of testing of this idea in a 24×32 Sparse Combo Net on the sequential CIFAR10 task. We varied the number of feedback connections that were fixed at 0 while all other settings remained static (Table 4.D.1). The resulting performance took an inverse U shape, where the network with only 50% of possible feedback connections non-zero had the best test accuracy of 65.14%, achieved in just

124 epochs of training.

Size	Feedback Density	Epochs	Best Overall Test Acc.	Best Test Acc. Through 85 Epochs
24 × 32	100%	86	52.7%	52.7%
24 × 32	75%	88	56.49%	56.48%
24 × 32	66.6%	89	58.84%	58.84%
24 × 32	50%	124	65.14%	58.01%
24 × 32	33.3%	129	61.86%	56.05%
24 × 32	25%	92	54.26%	50.54%
24 × 32	0%	130	39.8%	38.38%
16 × 32	100%	150	64.63%	55.82%
16 × 32	75%	150	64.12%	57.23%
16 × 32	50%	127	59.87%	54.26%

Table 4.D.1: Results from pilot testing on the sparsity of negative feedback connections in a 24×32 Sparse Combo Net and a 16×32 Sparse Combo Net. Feedback Density refers to the percentage of possible subnetwork pairings that were trained in negative feedback, while the remaining inter-network connections were held at 0. All networks were trained with the same 150 epoch training paradigm as mentioned in the main text, but were stopped after hitting a 24 hour runtime limit. Decreasing Feedback Density is a promising path towards further improving performance as the size of Sparse Combo Nets is scaled. The ideal amount of feedback density will likely vary with the size of the combination network.

4.D.3 Architecture Interpretability

In this subsection, we give additional information on trainable parameters and properties of the network weights, to assist in interpreting the Sparse Combo Net architecture

and its training process.

Number of Parameters

To report on the number of trainable parameters, we used the following formula:

$\frac{n^2 - M*C^2}{2} + i*n + n*o + n + o$

Where n is the total number of units in the $M \times C$ combination network, o is the total number of output nodes for the task, and i is the total number of input nodes for the task. Thus for the 16×32 networks highlighted here, we have 129034 trainable parameters for the MNIST tasks, and 130058 trainable parameters for sequential CIFAR10.

Note that the naive estimate for the number of trainable parameters would be $n^2 + i*n + n*o + n + o$, corresponding to the number of weights in $\mathbf{L}$, the number of weights in the feedforward linear input layer, the number of weights in the feedforward linear output layer, and the bias terms for the input and output layers, respectively. However, because of the combination property constraints on $\mathbf{L}$, only the lower triangular portion of a block off-diagonal matrix is actually trained, and $\mathbf{L}$ is then defined in terms of this matrix and the metric $\mathbf{M}$. Thus we subtract $M*C^2$ to remove the block diagonal portions corresponding to nonlinear RNN components, and then divide by 2 to obtain only the lower half.

Inspecting Trained Network Weights

After training was completed, we inspected the state of all networks described in the main text, pulling both the nonlinear ($\mathbf{W}$) and linear ($\mathbf{L}$) weight matrices from both initialization time and the final model. For $\mathbf{W}$, we confirmed it did not change over training, and inspected the max real part of the eigenvalues of $|\mathbf{W}|$ in accordance with Theorem 1. The densest tested matrices tended to have $\lambda_{max}(|\mathbf{W}|) > 0.9$, while the sparsest ones tended to have $\lambda_{max}(|\mathbf{W}|) < 0.1$. For $\mathbf{L}$, we checked the maximum element and the maximum singular value before and after training. In general, both went up over the course of training, but by a modest amount.

4.D.4 Tables of Results by Trial (Supplementary Documentation)

Table 4.D.2 shows all trials run on permuted sequential MNIST before beginning the more systematic experiments reported on in the main text. Notably, our networks did not require an extensive hyperparameter tuning process.

Tables 4.D.3 and 4.D.4 report additional details on the initial size and modularity experiments (Figure 4.D.1). Table 4.D.5 reports the results of the repeated experiment on Sparse Combo Net size, this time using 32 unit component subnetworks with best sparsity settings (Figure 4.3.3A).

Tables 4.D.6 and 4.D.7 report results from all trials related to our sparsity experiments (Figures 4.3.4 and 4.D.2).

Table 4.D.8 provides further information on the four trials in the permuted seqMNIST repeatability experiments (Figure 4.D.3).

Table 4.D.9 reports the results of all trials of different hyperparameters on the sequential CIFAR10 task, in chronological order. Ultimately the same settings as those used for permuted seqMNIST were chosen.

Table 4.D.10 shows the results of all ten trials in the seqCIFAR10 repeatability experiments (Figure 4.D.4). Table 4.D.11 shows results from nine additional seqCIFAR10 trials of shorter training duration (Figure 4.D.5), run to increase sample size while unable to access the higher quality Colab GPUs.

Finally, Table 4.D.1 reports the results of our pilot trial on introducing sparsity into the linear feedback connection matrix - as this table was presented in Section 4.D.2 however, we do not reproduce it here.

Permuted seqMNIST Trials

Size	Epochs	Adam WD	Initial LR	LR Schedule	Final Test Acc.
10 × 16	150	5e-5	5e-3	0.1 after 91	84%
10 × 16	150	1e-5	1e-2	0.1 after 50,100	85%
15 × 16	150	2e-4	5e-3	0.1 after 50,100	84%
10 × 16	150	2e-4	1e-2	0.5 every 10	81%
10 × 16	200	2e-4	1e-2	0.5 after 10 then every 30	81%
10 × 16	171*	5e-5	1e-2	0.75 after 10,20,60,100 then every 15	84%
15 × 16	179*	1e-5	1e-3	0.1 after 100,150	90%

Table 4.D.2: Training hyperparameter tuning trials, presented in chronological order. * indicates that training was cut short by the 24 hour Colab runtime limit. LR Schedule describes the scalar the learning rate was multiplied by, and at what epochs. The best performing network is highlighted, and represents the training settings we used throughout most of the main text.

Size	Final Test Acc.	Epoch 1 Test Acc.	Final Train Loss
1×16	38.69%	24.61%	1.7005
3×16	70.56%	40.47%	0.9033
5×16	77.86%	47.99%	0.7104
10×16	85.82%	61.38%	0.4736
15×16	90.28%	69.09%	0.3156
20×16	92.26%	71.72%	0.2392
22×16	93.01%	70.11%	0.2073
25×16	92.99%	61.81%	0.2017
30×16	93.16%	43.21%	0.1991

Table 4.D.3: Results for combination networks containing different numbers of component 16-unit RNNs. Training hyperparameters and network initialization settings were kept the same across all trials, and all trials completed the full 150 epochs.

Size	Final Test Acc.	Epoch 1 Test Acc.	Final Train Loss
1×352	40.17%	26.97%	1.662
11×32	89.12%	61.29%	0.3781
22×16	93.01%	70.11%	0.2073
44×8	94.44%	25.78%	0.1500
88×4	10.99%	10.99%	2E+35

Table 4.D.4: Results for different distributions of 352 total units across a combination network. This number was chosen based on prior 22×16 network performance. For each component RNN size tested, the same procedure was used to select appropriate density and scalar settings. All networks otherwise used the same settings, as in the size experiments.

Name	Size	Final Test Acc.
Sparse Combo Net	1×32	37.1%
Sparse Combo Net	4×32	89.1%
Sparse Combo Net	8×32	93.6%
Sparse Combo Net	12×32	94.4%
Sparse Combo Net	16×32	96%
Sparse Combo Net	22×32	96.8%
Sparse Combo Net	24×32	96.7%
Control	24×32	47%

Table 4.D.5: Results for Sparse Combo Nets containing different numbers of component 32-unit RNNs with best found initialization settings, using the standard 150 epoch training paradigm. This experiment was run to demonstrate repeatability of the size results seen in Table 4.D.3. All trials were run to completion. A control trial was also run with the largest tested network size - the connections between subnetworks were no longer constrained, and thus this control combination network is not certifiably stable.

Size	Density	Scalar	Final Test Acc.	Epoch 1 Test Acc.	Final Train Loss
11 × 32	26.5%	0.27	89.12%	61.29%	0.3781
11 × 32	10%	1.0	94.86%	70.67%	0.1278
22 × 16	40%	0.4	93.01%	70.11%	0.2073
22 × 16	20%	1.0	95.27%	76.58%	0.0924
22 × 16	10%	1.0	94.26%	71.53%	0.1425
44×8	60%	0.7	94.44%	25.78%	0.1500
44×8	50%	1.0	95.05%	30.52%	0.1180

Table 4.D.6: Results for different initialization settings - varying sparsity and magnitude of the component RNNs for different network sizes. All other settings remained constant across trials, using our selected 150 epoch training paradigm.

Density	Pre-select Scalar	Post-select Scalar	Final Test Acc.	Epoch 1 Test Acc.	Final Train Loss
10%	1.0	1.0	95.87%	73.67%	0.074
5%	10.0	0.1	95.11%	73.10%	0.1311
5%	10.0	0.2	96.15%	82.50%	0.0051
5%	10.0	0.5	96.69%	75.76%	0.0001
5%	6.0	1.0	96.41%	21.55%	3.3E-5
5%	7.5	1.0	16.75%	11.39%	3068967
3.3%	30.0	0.1	96.24%	83.89%	0.0005
3.3%	30.0	0.2	96.54%	86.79%	4E-5
1%	10.0	1.0	96.04%	81.2%	0.0001

Table 4.D.7: Further optimizing the sparsity settings for high performance using a 16×32 network. The final scalar is the product of the pre-selection and post-selection scalars. Note that the 5% density and 7.5 scalar network was killed after 18 epochs due to exploding gradient. All other trials ran for a full 150 epochs.

Epochs	Best Test Acc.	Epoch 1 Test Acc.	Best Train Loss
208	96.65%	61.15%	0.00224
255	96.94%	84.19%	5e-5
250	96.88%	81.08%	5e-5
210	96.93%	67.19%	0.00069

Table 4.D.8: Repeatability of the best network settings on permuted seqMNIST. Four trials of 16×32 networks with 3.3% density and entries between -6 and 6, trained for a 24 hour period with a single learning rate cut (0.1 scalar) after epoch 200. All other training settings remained the same as our selected hyperparameters. Trials are presented in chronological order. The mean test accuracy achieved was 96.85% with Variance 0.019.

seqCIFAR10 Trials

Density	Pre-select Scalar	Post-select Scalar	Epochs	Adam WD	Initial LR	LR Schedule	Best Test Acc.
3.3%	30	0.2	150	1e-5	1e-3	0.1 after 90,140	64.63%
3.3%	30	0.2	34*	1e-5	5e-3	0.1 after 90,140	35.42%
5%	6	1	150	1e-5	1e-3	0.1 after 90,140	60.9%
5%	10	0.5	150	1e-5	1e-4	0.1 after 90,140	54.86%
3.3%	30	0.2	150	1e-5	5e-4	0.1 after 90,140	61.83%
3.3%	30	0.2	200	1e-6	2e-3	0.1 after 140,190	62.31%
3.3%	30	0.2	186*	1e-5	1e-3	0.1 after 140,190	64.75%
3.3%	30	0.2	132*	1e-6	1e-3	0.1 after 140,190	62.31%
5%	10	0.5	195*	1e-5	1e-3	0.1 after 140,190	64.68%

Table 4.D.9: Additional hyperparameter tuning for the seqCIFAR10 task, presented in chronological order. * indicates that training was cut short by the 24 hour Colab runtime limit, or in the case of high learning rate was killed intentionally due to exploding gradient. LR Schedule describes the scalar the learning rate was multiplied by, and at what epochs. The best performing network is highlighted. Ultimately we decided on the same network settings and training hyperparameters for further testing, just extending the training period to 200 epochs with the learning rate cuts occurring after epochs 90 and 140.

Epochs	Best Test Acc.	Epoch 1 Test Acc.	Best Train Loss
186	64.75%	10.53%	0.83
200	64.04%	26.35%	0.787
200	64.32%	26.76%	0.778
170	64.88%	20.65%	0.857
200	65.72%	27.28%	0.813
200	63.73%	10.47%	0.826
200	65.03%	18.75%	0.83
200	64.71%	32.36%	0.799
200	64.4%	10.09%	0.83
200	65.63%	30.25%	0.792

Table 4.D.10: Repeatability of the best network settings on seqCIFAR10. Ten trials of 16×32 networks with 3.3% density and entries between -6 and 6, trained for 200 epochs with learning rate scaled by 0.1 after epochs 140 and 190. All other training settings remained the same as before. Trials are presented in chronological order. The mean test accuracy achieved was 64.72% with Variance 0.406. Most trials completed all 200 epochs, but two were cut short due to runtime limits.

Epochs	Best Test Acc.	Epoch 1 Test Acc.	Best Train Loss
150	64.63%	28.91%	0.837
150	63.51%	9.7%	0.9
148	62.35%	17.51%	0.89
126	63.33%	23.61%	0.903
129	61.45%	28.93%	0.937
150	62.21%	25.82%	0.9
150	63.42%	23.51%	0.86
147	62.05%	27.67%	0.882
131	62.41%	30.33%	0.912

Table 4.D.11: An additional nine trials investigating the repeatability of our results on the seqCIFAR10 task. For these trials we used the same 150 epoch training paradigm as previously, although only four of the networks were able to fully complete training. These trials were done to expand our sample size while access was limited to only slower GPUs. The mean observed test accuracy among the shorter trials was 62.82%, with variance of 0.95.

4.E SVD Combo Net Details

4.E.1 Parameterization Information

For the SVD Combo Net, we ensured contraction by directly parameterizing each of the $\mathbf{W}_i$ $(i = 1, 2 \dots p)$ as:

$$\mathbf{W}_i = \mathbf{\Phi}_i^{-1} \mathbf{U}_i \mathbf{\Sigma}_i \mathbf{V}_i^T \mathbf{\Phi}_i \tag{4.19}$$

where $\mathbf{\Phi}_i$ is diagonal and nonsingular, $\mathbf{U}_i$ and $\mathbf{V}_i$ are orthogonal, and $\mathbf{\Sigma}_i$ is diagonal with $\Sigma_{ii} \in [0, g^{-1})$. We ensure orthogonality of $\mathbf{U}_i$ and $\mathbf{V}_i$ during training by exploiting the fact that the matrix exponential of a skew-symmetric matrix is orthogonal, as was done in [Lezcano-Casado and Martınez-Rubio, 2019]. The network constructed from these subnetworks using (4.2) is contracting in metric $\tilde{\mathbf{M}} = \text{BlockDiag}(\mathbf{\Phi}_1^2, \dots, \mathbf{\Phi}_p^2)$.

4.E.2 Control Experiment

To do the SVD Combo Network stability control experiment, we still constrained the weights of the subnetwork modules using the above parameterization, but we removed any constraint on the connections between modules. Thus each individual module was still guaranteed to be contracting, but the overall system had no such guarantee in this control trial. The experiment was run using a 24×32 combination network and the permuted sequential MNIST task.

In contrast to the primary control experiment run on Sparse Combo Net, for the SVD Combo Net we saw only a very slight performance decrease when the between-module connections were no longer constrained for the overall stability certificate. Test accuracy in the control run was 94.56%, as compared to the 94.9% observed with the standard SVD Combo Net.

This disparity in performance decrease makes sense when considering that the hidden-to-hidden weights of SVD Combo Network are trainable, while those of the Sparse Combo Network are not. Whatever instabilities are introduced by the lack of

constraint on the inter-subnetwork connections likely cannot be adequately compensated for in the Sparse Combo Network.

4.E.3 Model Code

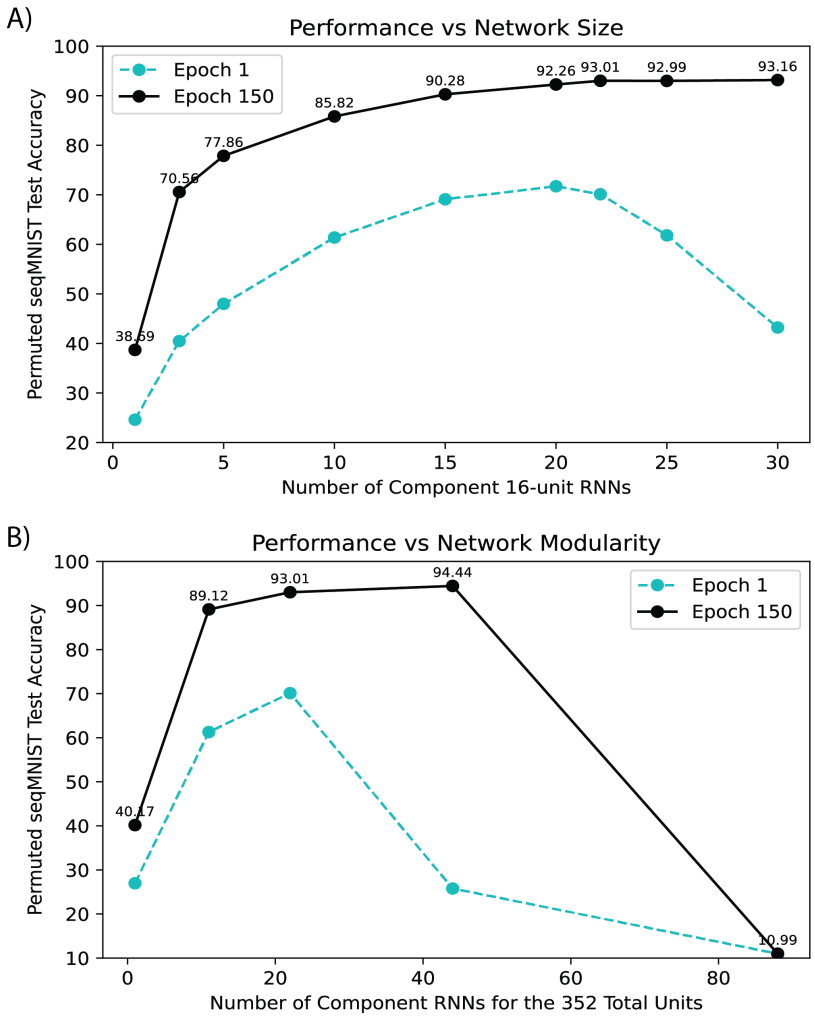

Figure 4.D.1: Performance of Sparse Combo Nets on the Permuted seqMNIST task by combination network size. We test the effects on final and first epoch test accuracy of both total network size and network modularity. The former is assessed by varying the number of subnetworks while each subnetwork is fixed at 16 units (A), and the latter by varying the distribution of units across different numbers of subnetworks with the total sum of units in the network fixed at 352 (B). Note that these experiments were run prior to optimizing the sparsity initialization settings. Experiments on total network size were later repeated with the final sparsity settings (Figure 4.3.3A). The results of both the size experiments are consistent.

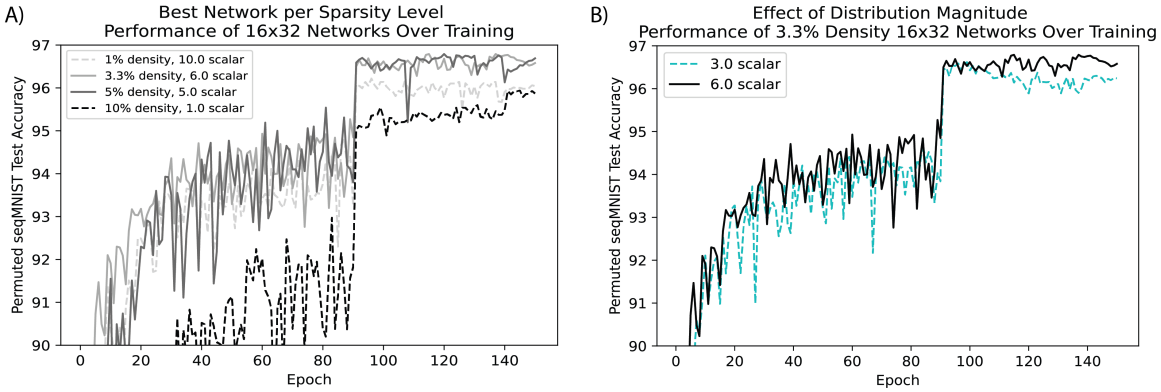

Figure 4.D.2: Permuted seqMNIST performance by component RNN initialization settings. Test accuracy is plotted over the course of training for four 16×32 networks with different density levels and entry magnitudes (A), highlighting the role of sparsity in network performance. Test accuracy is then plotted over the course of training for two 3.3% density 16×32 networks with different entry magnitudes (B), to demonstrate the role of the scalar. When the magnitude becomes too high however, performance is out of view of the current axis limits.

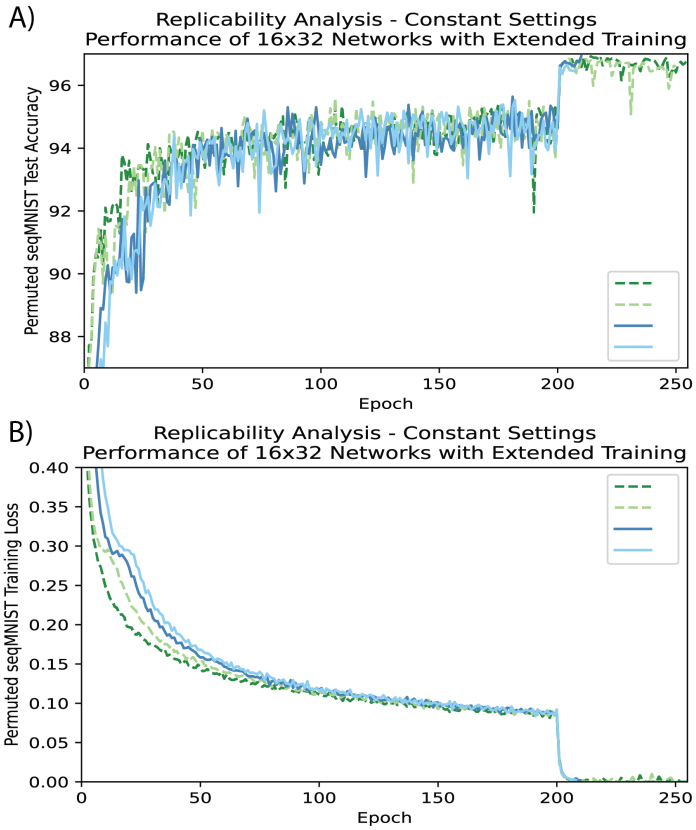

Figure 4.D.3: Permuted seqMNIST performance on repeated trials. Four different 16×32 networks with 3.3% density and entries between -6 and 6 were trained for 24 hours, with a single learning rate cut after epoch 200. (A) depicts test accuracy for each of the networks over the course of training. (B) depicts the training loss for the same networks. Exact numbers are reported in Table 4.D.8.

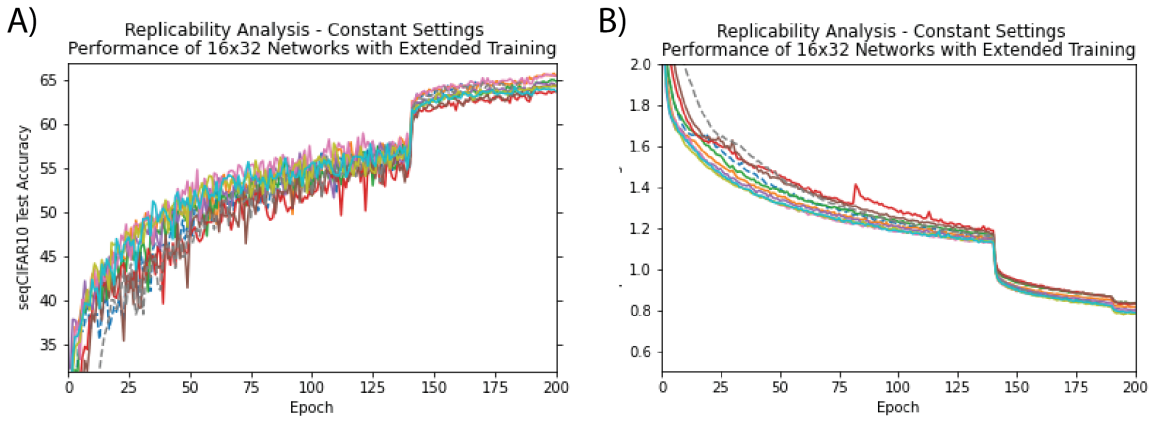

Figure 4.D.4: seqCIFAR10 performance on repeated trials. Ten different 16×32 networks with 3.3% density and entries between -6 and 6 were trained for 200 epochs, with learning rate divided by 10 after epochs 140 and 190. (A) depicts test accuracy for each of the networks over the course of training. (B) depicts the training loss for the same networks. Exact numbers are reported in Table 4.D.10.

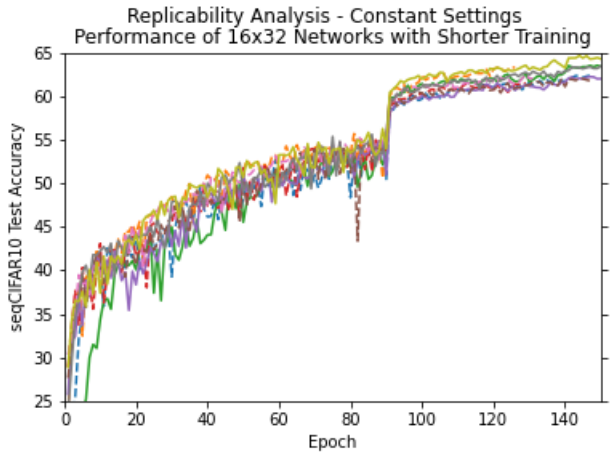

Figure 4.D.5: seqCIFAR10 performance on repeated trials with shorter training (done to complete more trials). Nine different 16×32 networks with 3.3% density and entries between -6 and 6 were set up to train for 150 epochs, with learning rate divided by 10 after epochs 90 and 140. Most of these networks hit runtime limit before completing, however they all got through at least 100 epochs and all had test accuracy exceed 61%. This figure depicts test accuracy for each of the networks over the course of training. Networks that completed training are plotted as solid lines, while those that were cut short are dashed.

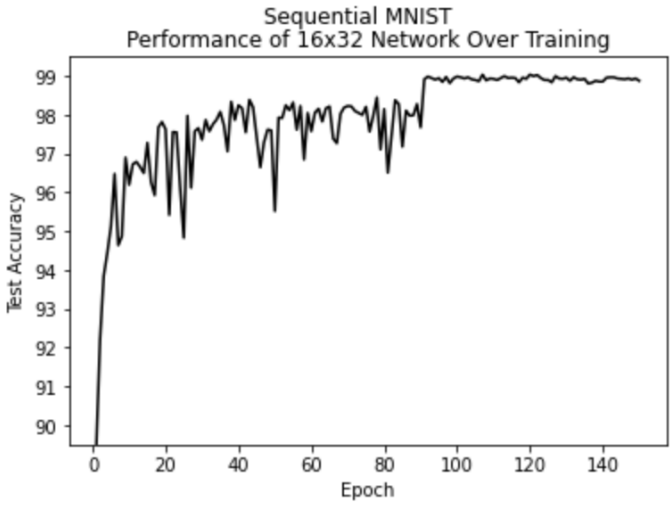

Figure 4.D.6: Performance over training on the seqMNIST task for a 16×32 network with best settings (using 150 epoch training protocol). Final test accuracy exceeded 99%.

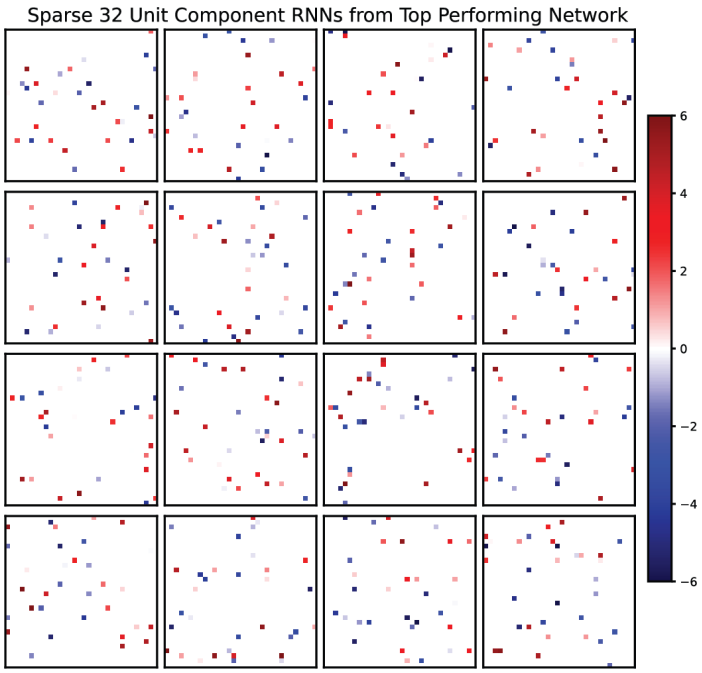

Figure 4.D.7: Weight matrices for each of the 32 unit nonlinear component RNNs that were used in the best performing 16×32 network on permuted sequential MNIST.

```python
#define network
class rnnAssembly_SV(LightningModule):
    '''
    Pytorch module for training the following system:
        tau*dx/dt = -x + W*phi(x) + L*x+ u(t)
    where tau > 0, phi is a nonlinearity, W is block diagonal, L is some 'contracting' combination matrix and u is some input.

    '''

    def __init__(self, input_size, hidden_sizes, output_size,alpha,A):
        super(rnnAssembly_SV, self).__init__()
        self.input_size = input_size
        self.hidden_sizes = hidden_sizes
        self.hidden_size = int(np.sum(hidden_sizes))
        ns = hidden_sizes
        self.output_size = output_size
        self.alpha = alpha

        #input to hidden weights
        self.weight_ih = nn.Parameter(torch.normal(0,1/np.sqrt(self.hidden_size),(self.hidden_size, self.input_size)))
        #hidden to output weights
        self.weight_ho = nn.Parameter(torch.normal(0,1/np.sqrt(self.hidden_size),(self.output_size, self.hidden_size)))

        #mask to modularize overall network
        self.register_buffer("V_mask", create_mask_given_A(torch.eye(len(ns)),ns).bool())

        #parameterization of hidden-to-hidden networks
        self.U = nn.Parameter(torch.eye(self.hidden_size))
        self.V = nn.Parameter(torch.eye(self.hidden_size))
        self.Sigma = nn.Parameter(torch.ones((self.hidden_size,)))
        self.Phi = nn.Parameter(torch.normal(1,1/np.sqrt(self.hidden_size),(self.hidden_size,)))
        self.sing_val_eps = 1e-3
        self.register_buffer("L_mask", create_mask_given_A(A,ns).bool())
        self.register_buffer("S_offset",(self.sing_val_eps)*torch.ones(self.Sigma.shape[0]))

        #trainable inter-subnetwork weights
        self.L_train = nn.Parameter(self.L_mask*torch.normal(0,1/np.sqrt(np.mean(ns)),(np.sum(ns), np.sum(ns))))

        #biases
        self.bias_oh = nn.Parameter(torch.normal(0,1/np.sqrt(self.hidden_size),(1,self.output_size)))
        self.bias_hh = nn.Parameter(torch.normal(0,1/np.sqrt(self.hidden_size),(1,self.hidden_size)))

    def forward(self, input):

        #mask the left and right orth 'stem' matrices to produce diagonal-block structure
        U_masked = self.U*self.V_mask
        V_masked = self.V*self.V_mask

        #take skew-symmetric part of these matrices
        #exponentiate skew-symmetric part to produce left and right orthogonal matrices
        O1 = torch.matrix_exp(U_masked - U_masked.T)
        O2 = torch.matrix_exp(V_masked - V_masked.T)

        #get diagonal of singular value matrix, with elements between [0,1)
        S = torch.exp(-self.Sigma**2 -self.S_offset)

        #construct the weight matrix to be contracting in metric determined by Phi
        W = torch.diag_embed(self.Phi) @ (O1 @ torch.diag_embed(S) @ O2.T) @ torch.diag_embed(1/self.Phi)

        #get metric and metric inverse
        M = torch.diag_embed(1/(self.Phi**2))
        M_inv = torch.diag_embed(self.Phi**2)

        #get mask for network-to-network coupling
        L_masked = self.L_train*self.L_mask

        #initialize hidden state of the network
        inputs = input.unbind(1)
        state = torch.zeros((input.shape[0],self.hidden_size),device = self.device)
        state = state.type_as(state)

        #propagate input through the dynamics and store outputs
        for i in range(len(inputs)):
            fx = -state + F.relu(state @ W.T + inputs[i] @ (self.weight_ih.T) + self.bias_hh) + state @ (L_masked.T - (M @ L_masked) @ M_inv)
            state = state + self.alpha*fx
        hy = state @ (self.weight_ho.T)
        return hy, state
```

Figure 4.E.1: Pytorch Lightning code for SVD Combo Net cell.

Chapter 5

Discussion

5.1 Main Contributions

This thesis uses contraction analysis to build better models of recurrent cortical processing, as well as state-of-the-art machine learning systems. More specifically, the main contributions of this thesis are as follows:

- Use contraction analysis to derive biologically-plausible conditions that ensure reproducibility in individual recurrent neural networks.
- Thorough hyperparameter search reveals that contracting recurrent neural networks are more brain-like and structurally robust than other recurrent neural networks.
- Use to contraction analysis to build recurrent neural networks *of* recurrent neural networks (RNN of RNNs), and demonstrate that these networks achieve state-of-the-art in several challenging machine learning benchmarks.

5.2 Limitations and Future Directions

One limitation of the work in this thesis is that the contraction conditions it uses are *global*. This condition implies that all trajectories of the system converge, regardless of

initial condition. Global contractivity is probably too strong a condition to reasonably expect in all brain regions. It is more likely that local contractivity, whereby all trajectories within a particular region of state space, can and will be achieved in neural circuits. Indeed, this is the idea behind the famous content addressable memory of Hopfield [Hopfield, 1982]. A main challenge in deriving local contraction conditions is that one needs to know where the contraction regions are a priori, and then one needs to show that the system never leaves that region (i.e the region is a forward-invariant set). There is currently no way to analytically or algorthimically describe these regions.

In future work, we would like to train the RNN of RNNs model on challenging neuroscience tasks (e.g, Imagenet [Deng et al., 2009]), and compare it's representations to neural data explicitly. More generally, we would like to expand the scope and flexibility of the RNN of RNNs framework to accommodate tasks from many different modalities, as well as comparisons against neural data from many different regions [Yang and Molano-Mazon, 2021]. Finally, there is a need for modelling complex animal behaviors in ecologically relevant situations [Zador et al., 2022]. It is an open question to what extend theory-based approaches, such as the one explored in this thesis, are useful to neurethological, normative modelling approaches [Merel et al., 2019].

Bibliography

Larry Abbott and Karel Svoboda. Brain-wide interactions between neural circuits. *Current Opinion in Neurobiology*, 65:iii–v, Dec 2020. ISSN 09594388. doi: 10.1016/j.conb.2020.12.012.

D. J. Amit and N. Brunel. Model of global spontaneous activity and local structured activity during delay periods in the cerebral cortex. *Cerebral Cortex (New York, N.Y.: 1991)*, 7(3):237–252, 5 1997. ISSN 1047-3211. doi: 10.1093/cercor/7.3.237. PMID: 9143444.

William Ashby. *Design for a brain: The origin of adaptive behaviour*. Chapman & Hall Ltd, 1952a. ISBN 9401513201.

William Ashby. *Design for a brain: The origin of adaptive behaviour*. Springer Science & Business Media, 1952b.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Trellis networks for sequence modeling. *ICLR*, 2019.

Nicholas M Boffi, Stephen Tu, Nikolai Matni, Jean-Jacques E Slotine, and Vikas Sindhwani. Learning stability certificates from data. *CoRL*, 2020.

Michael Buehner and Peter Young. A tighter bound for the echo state property. *IEEE Transactions on Neural Networks*, 17(3):820–824, 2006.

V Dean Buonomano and Wolfgang Maass. State-dependent computations: spatiotemporal processing in cortical networks. 2009. doi: 10.1038/nrn2558.

Thiago B Burghi, Timothy O'Leary, and Rodolphe Sepulchre. Distributed online estimation of biophysical neural networks. *arXiv preprint arXiv:2204.01472*, 2022.

Veronica Centorrino, Francesco Bullo, and Giovanni Russo. Contraction analysis of hopfield neural networks with hebbian learning. *arXiv preprint arXiv:2204.05382*, 2022.

Warasinee Chaisangmongkon, Sruthi K. Swaminathan, David J. Freedman, and Xiao Jing Wang. Computing by robust transience: How the fronto-parietal network performs sequential, category-based decisions. *Neuron*, 93(6):1504–1517.e4, 3 2017. ISSN 10974199. doi: 10.1016/j.neuron.2017.03.002.

Bo Chang, Minmin Chen, Eldad Haber, and Ed H Chi. Antisymmetricrnn: A dynamical system view on recurrent neural networks. *arXiv preprint arXiv:1902.09689*, 2019.

Christopher H Chatham and David Badre. Multiple gates on working memory. *Current opinion in behavioral sciences*, 1:23–31, 2 2015. ISSN 2352-1546. doi: 10.1016/j.cobeha.2014.08.001.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. 10 2014. doi: 10.48550/arXiv.1409.1259. URL http://arxiv.org/abs/1409.1259. arXiv:1409.1259 [cs, stat].

Krzysztof M Choromanski, Jared Quincy Davis, Valerii Likhosherstov, Xingyou Song, Jean-Jacques Slotine, Jacob Varley, Honglak Lee, Adrian Weller, and Vikas Sindhwani. Ode to an ode. *Advances in Neural Information Processing Systems*, 33: 3338–3350, 2020.

Soon-Jo Chung and Jean-Jacques E Slotine. Cooperative robot control and concurrent synchronization of lagrangian systems. *IEEE transactions on Robotics*, 25(3): 686–700, 2009.

Mark M. Churchland, Byron M. Yu, John P. Cunningham, Leo P. Sugrue, Marlene R. Cohen, Greg S. Corrado, William T. Newsome, Andrew M. Clark, Paymon Hosseini, Benjamin B. Scott, David C. Bradley, Matthew A. Smith, Adam Kohn, J. Anthony Movshon, Katherine M. Armstrong, Tirin Moore, Steve W. Chang, Lawrence H. Snyder, Stephen G. Lisberger, Nicholas J. Priebe, Ian M. Finn, David Ferster, Stephen I. Ryu, Gopal Santhanam, Maneesh Sahani, and V. Krishna Shenoy. Stimulus onset quenches neural variability: A widespread cortical phenomenon. *Nature Neuroscience*, 13(3):369–378, 2010. ISSN 10976256. doi: 10.1038/nn.2501.

Michael A. Cohen and Stephen Grossberg. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-13(5):815–826, 1983. ISSN 21682909. doi: 10.1109/TSMC.1983.6313075.

Yimian Dai, Stefan Oehmcke, Fabian Gieseke, Yiquan Wu, and Kobus Barnard. Attention as activation. *arXiv preprint arXiv:2007.07729v2*, 2020.

Peter Dayan and Laurence F. Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. MIT Press, 8 2005. ISBN 978-0-262-54185-5. Google-Books-ID: fLT4DwAAQBAJ.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Charles Desoer and Hiromasa Haneda. The measure of a matrix as a tool to analyze computer algorithms for circuit analysis. *IEEE Transactions on Circuit Theory*, 19 (5):480–486, 1972.

Sami El-Boustani, Jacque P. K. Ip, Vincent Breton-Provencher, Graham W. Knott, Hiroyuki Okuno, Haruhiko Bito, and Mriganka Sur. Locally coordinated synaptic plasticity of visual cortex neurons in vivo. *Science*, 360(6395):1349–1354, 6 2018. ISSN 0036-8075. doi: 10.1126/science.aao0862.

Rainer Engelken, Fred Wolf, and Larry F Abbott. Lyapunov spectra of chaotic recurrent neural networks. *arXiv preprint arXiv:2006.02427*, 2020.

Armen G Enikolopov, LF Abbott, and Nathaniel B Sawtell. Internally generated predictions enhance neural and behavioral detection of sensory stimuli in an electric fish. 2018. doi: 10.1016/j.neuron.2018.06.006.

N. Benjamin Erichson, Omri Azencot, Alejandro Queiruga, Liam Hodgkinson, and Michael W. Mahoney. Lipschitz recurrent neural networks. *ICLR*, 2021.

S. Funahashi, C. J. Bruce, and P. S. Goldman-Rakic. Mnemonic coding of visual space in the monkey's dorsolateral prefrontal cortex. *Journal of Neurophysiology*, 61(2): 331–349, 2 1989. ISSN 0022-3077. doi: 10.1152/jn.1989.61.2.331. PMID: 2918358.

Zoltán Füredi and János Komlós. The eigenvalues of random symmetric matrices. *Combinatorica*, 1(3):233–241, 1981.

J. M. Fuster and G. E. Alexander. Neuron activity related to short-term memory. *Science (New York, N.Y.)*, 173(3997):652–654, 8 1971. ISSN 0036-8075. doi: 10.1126/science.173.3997.652. PMID: 4998337.

Jean Gallier et al. The schur complement and symmetric positive semidefinite (and definite) matrices (2019). *URL https://www. cis. upenn. edu/jean/schur-comp. pdf*, 2020.

John Gerhart and Marc Kirschner. The theory of facilitated variation. *Proceedings of the National Academy of Sciences*, 104(1):8582–8589, 2007. doi: 10.1073/pnas.0701035104.

Wulfram Gerstner and Werner M. Kistler. Mathematical formulations of hebbian learning. *Biological Cybernetics*, 87(5-6):404–415, 2002. ISSN 03401200. doi: 10.1007/s00422-002-0353-y.

B Girard, N Tabareau, Q C Pham, A Berthoz, and J.-J Slotine. Where neuroscience and dynamic system theory meet autonomous robotics: A contracting basal ganglia model for action selection. *Neural Networks*, 21:628–641, 2008. doi: 10.1016/j.neunet.2008.03.009.

P. S Goldman-Rakic. Cellular basis of working memory. *Neuron*, 14(3):477–485, 3 1995. ISSN 0896-6273. doi: 10.1016/0896-6273(95)90304-6.

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Multi-dimensional recurrent neural networks. In *International conference on artificial neural networks*, pages 549–558. Springer, 2007.

Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. *ICLR*, 2022.

Eldad Haber and Lars Ruthotto. Stable architectures for deep neural networks. *Inverse Problems*, 34(1):014004, 2017.

Michael M Halassa and Sabine Kastner. Thalamic functions in distributed cognitive control. *Nature neuroscience*, 20(12):1669–1679, 2017. ISSN 1097-6256.

Guillaume Hennequin, Tim P. Vogels, and Wulfram Gerstner. Optimal control of transient dynamics in balanced networks supports generation of complex movements. *Neuron*, 82(6):1394–1406, 6 2014. ISSN 10974199. doi: 10.1016/j.neuron.2014.04.045.

Guillaume Hennequin, Everton J. Agnes, and Tim P. Vogels. Inhibitory plasticity: Balance, control, and codependence. *Annual Review of Neuroscience*, 40(1):557–579, 2017. ISSN 0147-006X. doi: 10.1146/annurev-neuro-072116-031005.

Morris W Hirsch. Convergent activation dynamics in continuous time networks. *Neural Networks*, 2(5):331–349, 1989. ISSN 08936080. doi: 10.1016/0893-6080(89)90018-X.

J J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558, 4 1982. ISSN 0027-8424. PMID: 6953413 PMCID: PMC346238.

Roger A Horn, Roger A Horn, and Charles R Johnson. *Matrix analysis*. Cambridge university press, 1990.

Toshihiko Hosoya, Stephen A Baccus, and Markus Meister. Dynamic predictive coding by the retina. *Nature*, 436:71, 7 2005.

Herbert Jaeger. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. 2001.

Saber Jafarpour, Alexander Davydov, Anton Proskurnikov, and Francesco Bullo. Robust implicit networks via non-euclidean contractions. *Advances in Neural Information Processing Systems*, 34:9857–9868, 2021.

Eric R Kandel, James H Schwartz, Thomas M Jessell, Steven Siegelbaum, A James Hudspeth, and Sarah Mack. *Principles of neural science*, volume 4. McGraw-hill New York, 2000.

Hassan K Khalil. *Nonlinear systems*, volume 3. 2002.

Hiroaki Kitano. Biological robustness. *Nature Reviews Genetics*, 5(1111):826–837, Nov 2004. ISSN 1471-0064. doi: 10.1038/nrg1471.

Andreas Knoblauch, Günther Palm, and Friedrich T. Sommer. Memory capacities for synaptic and structural plasticity. *Neural Computation*, 22(2):289–341, 2 2010. ISSN 0899-7667. doi: 10.1162/neco.2009.08-07-588.

Leo Kozachkov, Mikael Lundqvist, Jean-Jacques Slotine, and Earl K Miller. Achieving stable dynamics in neural circuits. *PLoS computational biology*, 16(8):e1007659, 2020.

Leo Kozachkov, Michaela Ennis, and Jean-Jacques Slotine. Rnns of rnns: Recursive construction of stable assemblies of recurrent neural networks. *arXiv preprint arXiv:2106.08928*, 2021.

Leo Kozachkov, John Tauber, Mikael Lundqvist, Scott L Brincat, Jean-Jacques Slotine, and Earl K Miller. Robust working memory through short-term synaptic plasticity. *bioRxiv*, 2022a.

Leo Kozachkov, Patrick M Wensing, and Jean-Jacques Slotine. Generalization in supervised learning through riemannian contraction. *arXiv preprint arXiv:2201.06656*, 2022b.

Dmitry Krotov and John Hopfield. Large associative memory problem in neurobiology and machine learning. *arXiv preprint arXiv:2008.06996*, 2020.

Rodrigo Laje and V Dean Buonomano. Robust timing and motor patterns by taming chaos in recurrent neural networks. *Nature Neuroscience*, 16(7):925–933, 2013. ISSN 10976256. doi: 10.1038/nn.3405.

Christopher Langdon and Tatiana A Engel. Latent circuit inference from heterogeneous neural responses during cognitive tasks. *bioRxiv*, 2022.

Anders Lansner and Orjan Ekeberg. Reliability and speed of recall in an associative network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7 (4):490–498, 1 1985. ISSN 0162-8828. doi: 10.1109/tpami.1985.4767688.

Quoc V Le, Navdeep Jaitly, and Geoffrey E Hinton. A simple way to initialize recurrent networks of rectified linear units. *arXiv preprint arXiv:1504.00941*, 2015.

Mario Lezcano-Casado and David Martınez-Rubio. Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group. In *International Conference on Machine Learning*, pages 3794–3803. PMLR, 2019.

Alexandra Libby and Timothy J. Buschman. Rotational dynamics reduce interference between sensory and memory representations. *Nature Neuroscience*, 24(5):715–726, 5 2021. ISSN 1097-6256, 1546-1726. doi: 10.1038/s41593-021-00821-9.

J Lisman. A mechanism for the hebb and the anti-hebb processes underlying learning and memory. *Proceedings of the National Academy of Sciences*, 86(23):9574–9578, 12 1989. ISSN 0027-8424. doi: 10.1073/PNAS.86.23.9574.

Winfried Lohmiller and Jean-Jacques E Slotine. On contraction analysis for non-linear systems. *Automatica*, 34(6):683–696, 1998.

Mikael Lundqvist, Pawel Herman, and Anders Lansner. Theta and gamma power increases and alpha/beta power decreases with memory load in an attractor network model. *Journal of Cognitive Neuroscience*, 23(10):3008–3020, 10 2011. ISSN 0898-929X. doi: 10.1162/jocn_a_00029.

Mikael Lundqvist, Jonas Rose, Pawel Herman, Scott L. Brincat, Timothy J. Buschman, and Earl K. Miller. Gamma and beta bursts underlie working memory. *Neuron*, 90 (1):152–164, 4 2016. ISSN 08966273. doi: 10.1016/j.neuron.2016.02.028.

Mikael Lundqvist, Pawel Herman, and Earl K. Miller. Working memory: Delay activity, yes! persistent activity? maybe not. *The Journal of Neuroscience*, 38 (32):7013–7019, 8 2018. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.2485-17.2018. PMID: 30089640 PMCID: PMC6083456.

Mikael Lundqvist, Jonas Rose, Scott L Brincat, Melissa R Warden, Timothy J Buschman, Pawel Herman, and Earl K Miller. Reduced variability of bursting activity during working memory. *Scientific reports*, 12(1):1–10, 2022.

Timothy A. Machado, Isaac V. Kauvar, and Karl Deisseroth. Multiregion neuronal activity: the forest and the trees. *Nature Reviews Neuroscience*, page 1–22, Oct 2022. ISSN 1471-0048. doi: 10.1038/s41583-022-00634-0.

Niru Maheswaranathan, Alex Williams, Matthew Golub, Surya Ganguli, and David Sussillo. Universality and individuality in neural dynamics across large populations of recurrent networks. volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/5f5d472067f77b5c88f69f1bcfda1e08-Abstract.html. [Online; accessed 2022-05-02].

Ian R Manchester and Jean-Jacques E Slotine. Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design. *IEEE Transactions on Automatic Control*, 62(6):3046–3053, 2017.

George A Mashour, Pieter Roelfsema, Jean-Pierre Changeux, and Stanislas Dehaene. Conscious processing and the global neuronal workspace hypothesis. *Neuron*, 105 (5):776–798, 2020.

Nicolas Y Masse, Guangyu R Yang, H Francis Song, Xiao-Jing Wang, and David J Freedman. Circuit mechanisms for the maintenance and manipulation of information in working memory. *Nature neuroscience*, 22(7):1159–1167, 2019.

Kiyotoshi Matsuoka. Stability conditions for nonlinear continuous neural networks with asymmetric connection weights. *Neural networks*, 5(3):495–500, 1992.

Josh Merel, Diego Aldarondo, Jesse Marshall, Yuval Tassa, Greg Wayne, and Bence Ölveczky. Deep neuroethology of a virtual rodent. *arXiv preprint arXiv:1911.09451*, 2019.

E. K. Miller and J. D. Cohen. An integrative theory of prefrontal cortex function. *Annual Review of Neuroscience*, 24:167–202, 2001. ISSN 0147-006X. doi: 10.1146/annurev.neuro.24.1.167. PMID: 11283309.

Earl K. Miller, Cynthia A. Erickson, and Robert Desimone. Neural mechanisms of visual working memory in prefrontal cortex of the macaque. *The Journal of Neuroscience*, 16(16):5154–5167, 8 1996. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.16-16-05154.1996.

Earl K. Miller, Mikael Lundqvist, and André M. Bastos. Working memory 2.0. *Neuron*, 100(2):463–475, 10 2018. ISSN 0896-6273. doi: 10.1016/j.neuron.2018.09.023.

John Miller and Moritz Hardt. Stable recurrent models. *arXiv preprint arXiv:1805.10369*, 2018.

Kenneth D Miller and Francesco Fumarola. Mathematical equivalence of two common forms of firing rate models of neural networks. *Neural computation*, 24(1):25–31, 2012.

Brendan K Murphy and Kenneth D Miller. Balanced amplification: a new mechanism of selective amplification of neural activity patterns. *Neuron*, 61(4):635–648, 2009.

Kumpati S. Narendra and Robert Shorten. Hurwitz stability of metzler matrices. *IEEE Transactions On Automatic Control*, 55(6):1484–1487, 2010.

Randall C. O'Reilly and Michael J. Frank. Making working memory work: A computational model of learning in the prefrontal cortex and basal ganglia. *Neural Computation*, 18(2):283–328, 2 2006. ISSN 0899-7667, 1530-888X. doi: 10.1162/089976606775093909.

A. Emin Orhan and Wei Ji Ma. A diverse range of factors affect the nature of neural representations underlying short-term memory. *Nature Neuroscience*, 22(2):275–283, 2 2019. ISSN 1097-6256, 1546-1726. doi: 10.1038/s41593-018-0314-y.

Merav Parter, Nadav Kashtan, and Uri Alon. Facilitated variation: How evolution learns from past environments to generalize to new environments. *PLOS Computational Biology*, 4(11), 2008.

Matthew G. Perich, Charlotte Arlt, Sofia Soares, Megan E. Young, Clayton P. Mosher, Juri Minxha, Eugene Carter, Ueli Rutishauser, Peter H. Rudebeck, Christopher D. Harvey, and Kanaka Rajan. Inferring brain-wide interactions using data-constrained recurrent neural network models. 2021. doi: 10.1101/2020.12.18.423348. URL https://www.biorxiv.org/content/early/2021/03/11/2020.12.18.423348.

Quang-Cuong Pham and Jean-Jacques Slotine. Stable concurrent synchronization in dynamic system networks. *Neural networks*, 20(1):62–77, 2007.

Adityanarayanan Radhakrishnana, Mikhail Belkin, and Caroline Uhler. Overparameterized neural networks implement associative memory. *PNAS*, 117:27162–27170, 2020.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020.

Max Revay and Ian Manchester. Contracting implicit recurrent neural networks: Stable models with improved trainability. In *Learning for Dynamics and Control*, pages 393–403. PMLR, 2020.

Max Revay, Ruigang Wang, and Ian R Manchester. Lipschitz bounded equilibrium networks. *arXiv e-prints*, pages arXiv–2010, 2020.

Max Revay, Ruigang Wang, and Ian R Manchester. Recurrent equilibrium networks: Unconstrained learning of stable and robust dynamical models. *arXiv preprint arXiv:2104.05942*, 2021.

Spencer M Richards, Felix Berkenkamp, and Andreas Krause. The lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems. In *Conference on Robot Learning*, pages 466–476. PMLR, 2018.

Ivan Dario Jimenez Rodriguez, Aaron Ames, and Yisong Yue. Lyanet: A lyapunov framework for training neural odes. In *International Conference on Machine Learning*, pages 18687–18703. PMLR, 2022.

David W. Romero, Anna Kuzina, Erik J. Bekkers, Jakub M. Tomczak, and Mark Hoogendoorn. Ckconv: Continuous kernel convolution for sequential data. *arXiv preprint arXiv:2102.02611v1*, 2021.

Hongyu Ruan, Taixiang Saur, and Wei-Dong Yao. Dopamine-enabled anti-hebbian timing-dependent plasticity in prefrontal circuitry. *Frontiers in neural circuits*, 8:38, 2014. ISSN 1662-5110.

Ueli Rutishauser, Rodney J Douglas, and Jean-Jacques Slotine. Collective stability of networks of winner-take-all circuits. *Neural computation*, 23(3):735–773, 2011.

Ueli Rutishauser, Jean-Jacques Slotine, and Rodney Douglas. Computation in dynamically bounded asymmetric systems. *PLoS Comput Biol*, 11(1):e1004039, 2015.

Martin Schrimpf, Jonas Kubilius, Ha Hong, Najib J Majaj, Rishi Rajalingham, Elias B Issa, Kohitij Kar, Pouya Bashivan, Jonathan Prescott-Roy, Franziska Geiger, et al. Brain-score: Which artificial neural network for object recognition is most brain-like? *BioRxiv*, page 407007, 2020.

João D Semedo, Amin Zandvakili, Christian K Machens, M Yu Byron, and Adam Kohn. Cortical areas interact through a communication subspace. *Neuron*, 102(1): 249–259, 2019.

H Sebastian Seung. How the brain keeps the eyes still. *Proceedings of the National Academy of Sciences*, 93(23):13339–13344, 1996.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

Herbert A. Simon. The architecture of complexity. *Proceedings of the American Philosophical Society*, 106(6):467–482, 1962. ISSN 0003-049X.

Jean-Jacques Slotine. Modular stability tools for distributed computation and control. *Int. J. Adaptive Control and Signal Processing*, 17(6), 2003.

Jean-Jacques Slotine and Yang-Yu Liu. The missing link. *Nature Physics*, 8(7): 512–513, 2012.

Jean-Jacques Slotine and Winfried Lohmiller. Modularity, evolution, and the binding problem: a view from stability theory. *Neural Networks*, 14(2):137–145, 2001.

Jean-jacques E. Slotine. Modular stability tools for distributed computation and control, 2002.

Jean-Jacques E Slotine and Weiping Li. *Applied Nonlinear Control*. Prentice-Hall, 1991.

Haim Sompolinsky, Andrea Crisanti, and Hans-Jurgen Sommers. Chaos in random neural networks. *Physical review letters*, 61(3):259, 1988.

Sen Song, Per Jesper Sjöström, Markus Reigl, Sacha Nelson, and Dmitri B. Chklovskii. Highly nonrandom features of synaptic connectivity in local cortical circuits. *PLoS Biology*, 3(3):0507–0519, 2005. ISSN 15457885. doi: 10.1371/journal.pbio.0030068.

Eduardo D Sontag. Contractive systems with inputs. In *Perspectives in mathematical system theory, control, and signal processing*, pages 217–228. Springer, 2010.

Eelke Spaak, Kei Watanabe, Shintaro Funahashi, and Mark G Stokes. Stable and dynamic coding for working memory in primate prefrontal cortex. *Journal of Neuroscience*, 37(27):6503–6516, 2017. ISSN 15292401. doi: 10.1523/JNEUROSCI.3364-16.2017.

Van Rob R De Ruyter Steveninck, Geoffrey D Lewen, Steven P Strong, Roland Koberle, and William Bialek. Reproducibility and variability in neural spike trains. 275(March), 1997.

Mark G. Stokes. 'activity-silent' working memory in prefrontal cortex: a dynamic coding framework. *Trends in Cognitive Sciences*, 19(7):394–405, 7 2015. ISSN 13646613. doi: 10.1016/j.tics.2015.05.004.

David Sussillo. Neural circuits as computational dynamical systems. *Current opinion in neurobiology*, 25:156–163, 2014. ISSN 0959-4388.

David Sussillo and Omri Barak. Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural computation*, 25(3):626–649, 2013.

Yaroslav Sych, Aleksejs Fomins, Leonardo Novelli, and Fritjof Helmchen. Dynamic reorganization of the cortico-basal ganglia-thalamo-cortical network during task learning. *Cell Reports*, 40(12):111394, 2022.

Nicolas Tabareau and Jean-Jacques Slotine. Notes on contraction theory. *arXiv preprint nlin/0601011*, 2006.

Trieu H. Trinh, Andrew M. Dai, Minh-Thang Luong, and Quoc V. Le. Learning longer-term dependencies in rnns with auxiliary losses. *arXiv preprint arXiv:1803.00144v3*, 2018.

Misha V Tsodyks and Henry Markram. The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability. *Proceedings of the national academy of sciences*, 94(2):719–723, 1997. publisher: National Acad Sciences.

Danil Tyulmankov, Guangyu Robert Yang, and L. F. Abbott. Meta-learning synaptic plasticity and memory addressing for continual familiarity detection. *Neuron*, 12 2021. ISSN 0896-6273. doi: 10.1016/j.neuron.2021.11.009. URL https://www.sciencedirect.com/science/article/pii/S0896627321009478. [Online; accessed 2021-12-24].

Mathukumalli Vidyasagar. *Nonlinear systems analysis*, volume 42. Siam, 2002.

T. P. Vogels, H. Sprekeler, F. Zenke, C. Clopath, and W. Gerstner. Inhibitory plasticity balances excitation and inhibition in sensory pathways and memory networks. *Science*, 334(6062):1569–1573, 12 2011. ISSN 0036-8075. doi: 10.1126/science.1211095.

T. P. Vogelsy, R. C. Froemkey, N. Doyon, M. Gilson, J. S. Haas, R. Liu, A. Maffei, P. Miller, C. J. Wierenga, M. Woodin, F. Zenke, and H. Sprekelery. Inhibitory synaptic plasticity: Spike timing-dependence and putative network function. *Frontiers in Neural Circuits*, (JUNE), 6 2013. ISSN 16625110. doi: 10.3389/fncir.2013.00119.

Ryan Vogt, Maximilian Puelma Touzel, Eli Shlizerman, and Guillaume Lajoie. On lyapunov exponents for rnns: Understanding information propagation using dynamical systems tools. *arXiv preprint arXiv:2006.14123*, 2020.

Christopher (Notorious B.I.G) George Latore Wallace. Juicy. *Ready to Die*, 1994.

Michael Wehr and Anthony M. Zador. Balanced inhibition underlies tuning and sharpens spike timing in auditory cortex. *Nature*, 426(6965):442–446, 11 2003. ISSN 0028-0836. doi: 10.1038/nature02116.

Eric W. Weisstein. Positive definite matrix. URL https://mathworld.wolfram.com/PositiveDefiniteMatrix.html.

Patrick M Wensing and Jean-Jacques Slotine. Beyond convexity—contraction and global convergence of gradient descent. *Plos one*, 15(8):e0236661, 2020.

Norbert Wiener. *Cybernetics or Control and Communication in the Animal and the Machine*. MIT press, 1948.

Alex H Williams, Erin Kunz, Simon Kornblith, and Scott Linderman. Generalized shape metrics on neural representations. *Advances in Neural Information Processing Systems*, 34:4738–4750, 2021.

Hugh R. Wilson and Jack D. Cowan. Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysical Journal*, 12(1):1–24, 1972. ISSN 0006-3495. doi: https://doi.org/10.1016/S0006-3495(72)86068-5.

Kong-Fatt Wong and Xiao-Jing Wang. A recurrent network mechanism of time integration in perceptual decisions. *Journal of Neuroscience*, 26(4):1314–1328, 1 2006. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.3733-05.2006. publisher: Society for Neuroscience section: Articles PMID: 16436619.

Daniel LK Yamins, Ha Hong, Charles F Cadieu, Ethan A Solomon, Darren Seibert, and James J DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the national academy of sciences*, 111(23):8619–8624, 2014.

Guangyu R Yang and Manuel Molano-Mazon. Next-generation of recurrent neural network models for cognition, Apr 2021. URL psyarxiv.com/w34n2.

Guangyu Robert Yang and Manuel Molano-Mazón. Towards the next generation of recurrent network models for cognitive neuroscience. *Current opinion in neurobiology*, 70:182–192, 2021.

Guangyu Robert Yang, Madhura R Joglekar, H Francis Song, William T Newsome, and Xiao-Jing Wang. Task representations in neural networks trained to perform many cognitive tasks. *Nature neuroscience*, 22(2):297–306, 2019.

Izzet B Yildiz, Herbert Jaeger, and Stefan J Kiebel. Re-visiting the echo state property. *Neural networks*, 35:1–9, 2012.

Anthony Zador, Blake Richards, Bence Ölveczky, Sean Escola, Yoshua Bengio, Kwabena Boahen, Matthew Botvinick, Dmitri Chklovskii, Anne Churchland, Claudia Clopath, et al. Toward next-generation artificial intelligence: Catalyzing the neuroai revolution. *arXiv preprint arXiv:2210.08340*, 2022.

Huaguang Zhang, Zhanshan Wang, and Derong Liu. A comprehensive review of stability analysis of continuous-time recurrent neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 25(7):1229–1262, 2014. ISSN 21622388. doi: 10.1109/TNNLS.2014.2317880.