frontiers

Gated recurrent units viewed through the lens of continuous time dynamical systems

Ian D. Jordan [1,3], **Piotr Aleksander Sokół** [2] **and Il Memming Park** [1,2,3,*]

[1] *Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY, USA*
[2] *Department of Neurobiology and Behavior, Stony Brook University, Stony Brook, NY, USA*
[3] *Institute of Advanced Computational Science, Stony Brook University, Stony Brook, NY, USA*

Correspondence*:
Il Memming Park : Manuscript contains 7286 words and 12 figures
memming.park@stonybrook.edu

ABSTRACT

Gated recurrent units (GRUs) are specialized memory elements for building recurrent neural networks. Despite their incredible success on various tasks, including extracting dynamics underlying neural data, little is understood about the specific dynamics representable in a GRU network. As a result, it is both difficult to know a priori how successful a GRU network will perform on a given task, and also their capacity to mimic the underlying behavior of their biological counterparts. Using a continuous time analysis, we gain intuition on the inner workings of GRU networks. We restrict our presentation to low dimensions, allowing for a comprehensive visualization. We found a surprisingly rich repertoire of dynamical features that includes stable limit cycles (nonlinear oscillations), multi-stable dynamics with various topologies, and homoclinic bifurcations. At the same time we were unable to train GRU networks to produce continuous attractors, which are hypothesized to exist in biological neural networks. We contextualize the usefulness of different kinds of observed dynamics and support our claims experimentally.

Keywords: Recurrent Neural Networks, Dynamical Systems, Continuous Time, Bifurcations,Time-Series

1 INTRODUCTION

Recurrent neural networks (RNNs) can capture and utilize sequential structure in natural and artificial languages, speech, video, and various other forms of time series. The recurrent information flow within an RNN implies that the data seen in the past has influence on the current state of the RNN, forming a mechanism for having memory through (nonlinear) temporal traces that encode both *what* and *when*. Past works have used RNNs to study neural population dynamics (Costa et al., 2017), and have demonstrated qualitatively similar dynamics between biological neural networks and artificial networks trained under analogous conditions (Mante et al., 2013; Sussillo et al., 2015; Cueva et al., 2020). In turn, this brings into question the efficacy of using such networks as a means to study brain function. With this in mind, training standard vanilla RNNs to capture long-range dependences within a sequence is challenging due to the vanishing gradient problem (Hochreiter, 1991; Bengio et al., 1994). Several special RNN architectures have been proposed to mitigate this issue, notably the long short-term memory (LSTM) units (Hochreiter

and Schmidhuber, 1997) which explicitly guard against unwanted corruption of the information stored in the hidden state until necessary. Recently, a simplification of the LSTM called the *gated recurrent unit* (GRU) (Cho et al., 2014) has become popular in the computational neuroscience and machine learning communities thanks to its performance in speech (Prabhavalkar et al., 2017), music (Choi et al., 2017), video (Dwibedi et al., 2018), and extracting nonlinear dynamics underlying neural data (Pandarinath et al., 2018). However, certain mechanistic tasks, specifically unbounded counting, come easy to LSTM networks but not to GRU networks (Weiss et al., 2018).

Despite these empirical findings, we lack systematic understanding of the internal time evolution of GRU's memory structure and its capability to represent nonlinear temporal dynamics. Such an understanding will make clear what specific tasks (natural and artificial) can or cannot be performed (Bengio et al., 1994), how computation is implemented (Sussillo and Barak, 2012; Beer, 2006), and help to predict qualitative behavior (Zhao and Park, 2016; Beer, 1995). In addition, a great deal of the literature discusses the local dynamics (equilibrium points) of RNNs (Bengio et al., 1994; Sussillo and Barak, 2012), but a complete theory requires an understanding of the global properties as well (Beer, 1995). Furthermore, a deterministic understanding of a GRU network's topological structure will provide fundamental insight as to a trained network's generalization ability, and therefore help in understanding how to seed RNNs for specific tasks (Doya, 1993; Sokół et al., 2019).

In general, the hidden state dynamics of an RNN can be written as $\mathbf{h}_{t+1} = f(\mathbf{h}_t, \mathbf{x}_t)$ where $\mathbf{x}_t$ is the current input in a sequence indexed by t, f is a nonlinear function, and $\mathbf{h}_t$ represents the hidden memory state that carries all information responsible for future output. In the absence of input, $\mathbf{h}_t$ evolves over time on its own:

$$\mathbf{h}_{t+1} = f(\mathbf{h}_t) \tag{1}$$

where $f(\cdot) \coloneqq f(\cdot, \mathbf{0})$ for notational simplicity. In other words, we can consider the temporal evolution of memory stored within an RNN as a trajectory of an autonomous dynamical system defined by (1), and use dynamical systems theory to further investigate and classify the temporal features obtainable in an RNN. In this paper, we intend on providing a deep intuition of the inner workings of the GRU through a continuous time analysis. While RNNs are traditionally implemented in discrete time, we show in the next section that this form of the GRU can be interpreted as a numerical approximation of an underlying system of ordinary differential equations. Historically, discrete time systems are often more challenging to analyze when compared with their continuous time counterparts, primarily due to their more *jumpy* nature, allowing for more complex dynamics in low-dimensions (Pasemann, 1997; Laurent and von Brecht, 2017). Due to the relatively continuous nature of many abstract and physical systems, it may be of great use to analyze the underlying continuous time system of a trained RNN directly in some contexts, while interpreting the added dynamical complexity from the discretization as anomalies from numerical analysis (He et al., 2016; Heath, 2018; LeVeque and Leveque, 1992; Thomas, 1995). Furthermore, the recent development of *Neural Ordinary Differential Equations* have catalyzed the computational neuroscience and machine learning communities to turn much of their attention to continuous-time implementations of neural networks (Chen et al., 2018; Morrill et al., 2021).

We discuss a vast array of observed local and global dynamical structures, and validate the theory by training GRUs to predict time series with prescribed dynamics. As to not compromise the presentation, we restrict our analysis to low dimensions for easy visualization (Zhao and Park, 2016; Beer, 1995). However, given a trained GRU of any finite dimension, the findings here still apply, and can be applied with further

analysis on a case by case basis (more information on this in the discussion). Furthermore, to ensure our work is accessible we will assume a pedagogical approach in our delivery. We recommend Meiss (Meiss, 2007) for more background on the subject.

2 UNDERLYING CONTINUOUS TIME SYSTEM OF GATED RECURRENT UNITS

The GRU uses two internal gating variables: the *update gate* $\mathbf{z}_t$ which protects the d-dimensional hidden state $\mathbf{h}_t \in \mathbb{R}^d$ and the *reset gate* $\mathbf{r}_t$ which allows overwriting of the hidden state and controls the interaction with the input $\mathbf{x}_t \in \mathbb{R}^p$.

$$\mathbf{z}_t = \sigma(\mathbf{W}_z\mathbf{x}_t + \mathbf{U}_z\mathbf{h}_{t-1} + \mathbf{b}_z) \tag{2}$$

$$\mathbf{r}_t = \sigma(\mathbf{W}_r\mathbf{x}_t + \mathbf{U}_r\mathbf{h}_{t-1} + \mathbf{b}_r) \tag{3}$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \tanh(\mathbf{W}_h\mathbf{x}_t + \mathbf{U}_h(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h) + \mathbf{z}_t \odot \mathbf{h}_{t-1} \tag{4}$$

where $\mathbf{W}_z, \mathbf{W}_r, \mathbf{W}_h \in \mathbb{R}^{d\times p}$ and $\mathbf{U}_z, \mathbf{U}_r, \mathbf{U}_h \in \mathbb{R}^{d\times d}$ are the parameter matrices, $\mathbf{b}_z, \mathbf{b}_r, \mathbf{b}_h \in \mathbb{R}^d$ are bias vectors, $\odot$ represents element-wise multiplication, and $\sigma(\mathbf{z}) = 1/(1 + e^{-\mathbf{z}})$ is the element-wise logistic sigmoid function. Note that the hidden state is asymptotically contained within $[-1, 1]^d$ due to the saturating nonlinearities, implying that if the state is initialized outside of this trapping region, it must eventually enter it in finite time and remain in it for all later time.

Note that the update gate $\mathbf{z}_t$ controls how fast each dimension of the hidden state decays, providing an adaptive time constant for memory. Specifically, as $\lim_{\mathbf{z}_t\to 1} \mathbf{h}_t = \mathbf{h}_{t-1}$, GRUs can implement perfect memory of the past and ignore $\mathbf{x}_t$. Hence, a d-dimensional GRU is capable of keeping a near constant memory through the update gate—near constant since $0 < [\mathbf{z}_t]_j < 1$, where $[\cdot]_j$ denotes j-th component of a vector. Moreover, the autoregressive weights (mainly $\mathbf{U}_h$ and $\mathbf{U}_r$) can support time evolving memory ((Laurent and von Brecht, 2017) considered this a hindrance and proposed removing all complex dynamical behavior in a simplified GRU).

To investigate the memory structure further, let us consider the dynamics of the hidden state in the absence of input, i.e. $\mathbf{x}_t = 0, \forall t$, which is of the form (1). From a dynamical system's point of view, all inputs to the system can be understood as perturbations to the autonomous system, and therefore have no effect on the set of achievable dynamics. To utilize the rich descriptive language of continuous time dynamical systems theory, we recognize the autonomous GRU-RNN as a weighted forward Euler discretization to the following continuous time dynamical system:

$$\mathbf{z}(t) = \sigma(\mathbf{U}_z\mathbf{h}(t) + \mathbf{b}_z) \tag{5}$$

$$\mathbf{r}(t) = \sigma(\mathbf{U}_r\mathbf{h}(t) + \mathbf{b}_r) \tag{6}$$

$$\dot{\mathbf{h}} = (1 - \mathbf{z}(t)) \odot (\tanh(\mathbf{U}_h(\mathbf{r}(t) \odot \mathbf{h}(t)) + \mathbf{b}_h) - \mathbf{h}(t)) \tag{7}$$

where $\dot{\mathbf{h}} \equiv \frac{\mathrm{d}\mathbf{h}(t)}{\mathrm{d}t}$. Since both $\sigma(\cdot)$ and $\tanh(\cdot)$ are smooth, this continuous limit is justified and serves as a basis for further analysis, as all GRU networks are attempting to approximate this continuous limit. In the following, GRU will refer to the continuous time version (7). Note that the update gate $\mathbf{z}(t)$ again plays the role of a state-dependent time constant for memory decay. We note, however, that $\mathbf{z}(t)$ adjusts flow speed point-wise, resulting in non-constant nonlinear slowing of all trajectories, as $\mathbf{z}(t) \in (0, 1)$. Since $1 - \mathbf{z}(t) > 0$, and thus cannot change sign, it acts as a homeomorphism between (7) and the same system with this leading multiplicative term removed. Therefore, it does not change the topological structure of

the dynamics (Kuznetsov, 1998), and we can safely ignore the effects of $\mathbf{z}(t)$ in the following theoretical analysis sections (3 & 4). In these sections we set $\mathbf{U}_z = 0$ and $\mathbf{b}_z = 0$. A derivation of the continuous time GRU can be found in section 1 of the supplementary material. Further detail on the effects of $\mathbf{z}(t)$ are discussed in the final section of this paper.

3 STABILITY ANALYSIS OF A ONE DIMENSIONAL GRU

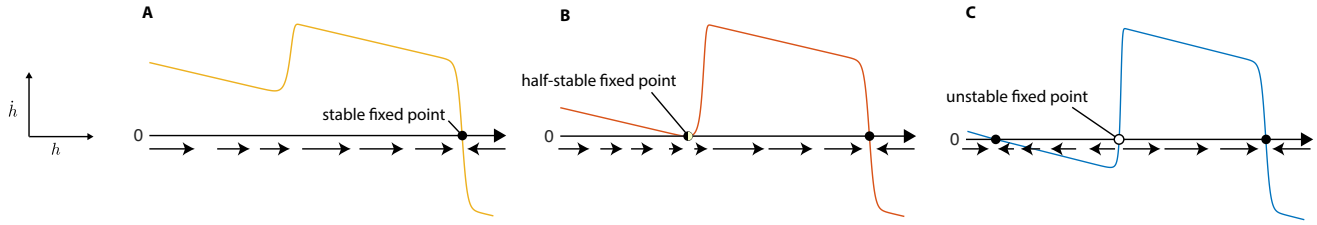

Figure 1. Three possible types of one dimensional flow for a 1D GRU. When $\dot{h} > 0$, $h(t)$ increases. This flow is indicated by a rightward arrow. Nodes ($\{h \mid \dot{h}(h) = 0\}$) are represented as circles and classified by their stability (Meiss, 2007).

For a 1D GRU* ($d = 1$), (7) reduces to a one dimensional dynamical system where every variable is a scalar. The expressive power of a 1D GRU is quite limited, as only three stability structures (topologies) exist (see section 2 in the supplementary material): (**A**) a single stable node, (**B**) a stable node and a half-stable node, and (**C**) two stable nodes separated by an unstable node (see Fig. 1). The corresponding time evolution of the hidden state are (A) decay to a fixed value, (B) decay to a fixed value, but from one direction halt at an intermediate value until perturbed, or (C) decay to one of two fixed values (bistability). The bistability can be used to model a switch, such as in the context of simple decision making, where inputs can perturb the system back and forth between states.

The topology the GRU takes is determined by its parameters. If the GRU begins in a region of the parameter space corresponding to (A), we can smoothly vary the parameters to transverse (B) in the parameter space, and end up at (C). This is commonly known as a saddle-node bifurcation. Speaking generally, a bifurcation is the change in topology of a dynamical system, resulting from a smooth change in parameters. The point in parameter space at which the bifurcation occurs is called the bifurcation point (e.g. Fig. 1B), and we will refer to the fixed point that changes its stability at the bifurcation point as the *bifurcation fixed point* (e.g. the half-stable fixed point in Fig. 1B). The codimension of a bifurcation is the number of parameters which must vary in order to remain on the bifurcation manifold. In the case of our example, a saddle-node bifurcation is codimension-1 (Kuznetsov, 1998). Right before transitioning to (B), from (A), the flow near where the half-stable node would appear can exhibit arbitrarily slow flow. We will refer to these as *slow points* (Sussillo and Barak, 2012). In this context, slow points allow for metastable states, where a trajectory will flow towards the slow point, remain there for a period of time, before moving to the stable fixed point.

4 ANALYSIS OF A TWO DIMENSIONAL GRU

We will see that the addition of a second GRU opens up a substantial variety of possible topological structures. For notational simplicity, we denote the two dimensions of $\mathbf{h}$ as x and y. We visualize the flow fields defined by (7) in 2-dimensions as *phase portraits* which reveal the topological structures of interest (Meiss, 2007). For starters, the phase portrait of two independent bistable GRUs can be visualized

*The number/dimension of GRUs references to the dimension of the hidden state dynamics.

as Fig. 2A. It clearly shows 4 stable states as expected, with a total of 9 fixed points. This could be thought of as a continuous-time continuous-space implementation of a finite state machine with 4 states (Fig. 2B). The 3 types of observed fixed points—stable (sinks), unstable (sources), and saddle points—exhibit locally linear dynamics, however, the global geometry is nonlinear and their topological structures can vary depending on their arrangement.

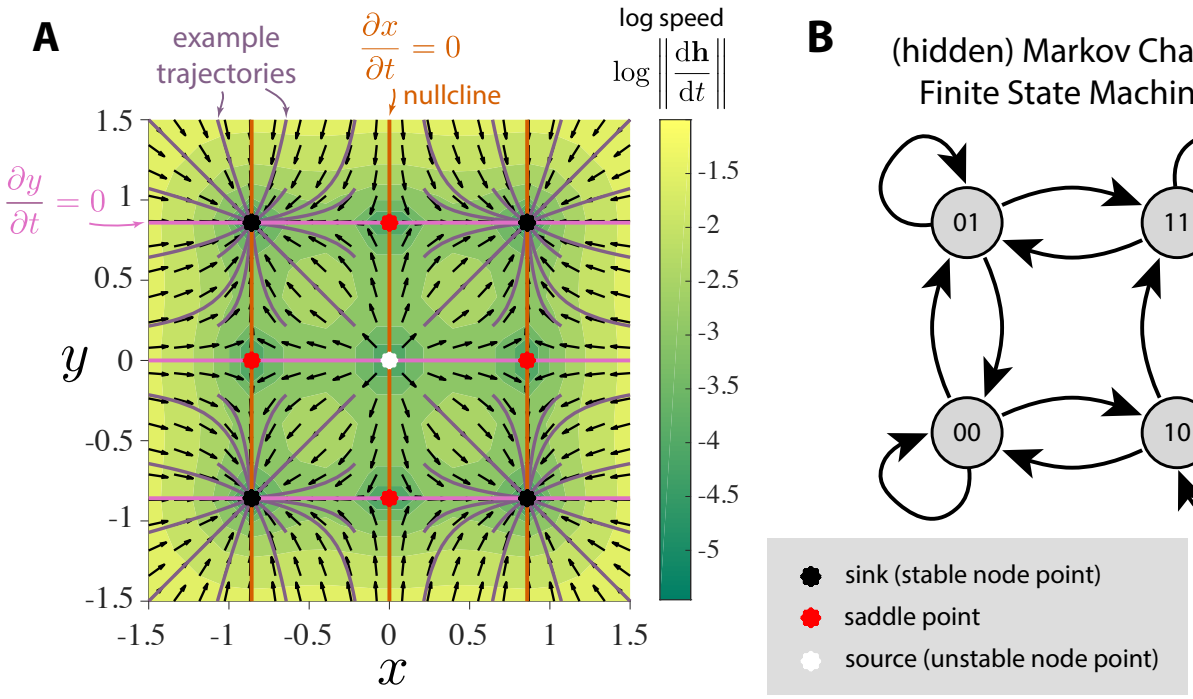

Figure 2. Illustrative example of two independent bistable GRUs. (**A**) Phase portrait. The flow field $\dot{\mathbf{h}} = [\dot{x}, \dot{y}]^\top$ is decomposed into direction (black arrows) and speed (color). Purple lines represent trajectories of the hidden state which converge to one of the four stable fixed points. Note the four quadrants coincide with the basin of attraction for each of the stable nodes. The fixed points appear when the x- and y-nullclines intersect. (**B**) The four stable nodes of this system can be interpreted as a continuous analogue of 4-discrete states with input-driven transitions.

We explored stability structures attainable by 2D GRUs. Due to the relatively large number of observed topologies, this section's main focus will be on demonstrating all observed local and global dynamical features obtainable by 2D GRUs. A catalog of all known topologies can be found in section 3 of the supplementary material, along with the parameters of every phase portrait depicted in this paper. We cannot say whether or not this catalog is exhaustive, but the sheer number of structures found is a testament to the expressive power of the GRU network, even in low dimensions.

Before proceeding, let us take this time to describe all the local dynamical features observed. In addition to the previously mentioned three types of fixed points, 2D GRUs can exhibit a variety of bifurcation fixed points, resulting from regions of parameter space that separate all topologies restricted to simple fixed points (i.e stable, unstable, and saddle points). Behaviorally speaking, these fixed points act as hybrids between the previous three, resulting in a much richer set of obtainable dynamics. In Fig. 3, we show all observed types of fixed points.[†] While no codimension-2 bifurcation fixed points were observed in the 2D

[†]2D GRUs feature both codimension-1 and pseudo-codimension-2 bifurcation fixed points. In codimension-1, we have the saddle-node bifurcation fixed point, as expected from its existence in the 1D GRU case. These can be thought of as both the fusion of a stable fixed point and a saddle point, and the fusion of an unstable fixed point and a saddle point. We will refer to these fixed points as saddle-node bifurcation fixed points of the first kind and second kind respectively.

GRU system, a sort of *pseudo-codimension-2* bifurcation fixed point was seen by placing a sink, source, and two saddle points sufficiently close together, such that, when implemented, all four points remain below machine precision, thereby acting as a single fixed point. Fig. 4 further demonstrates this concept, and Fig. 3B depicts and example. We will discuss later that this sort of pseudo-bifurcation point allows the system to exhibit *homoclinic-like* behavior on a two dimensional compact set. In Fig. 3A, we see 11 fixed points, the maximum number of fixed points observed in a 2D GRU system. A closer look at this system reveals one interpretation as a continuous analogue of 5-discrete states with input-driven transitions, similar to that depicted in Fig. 2. This imposes a possible upper bound on the network's capacity to encode a finite set of states in this manner.

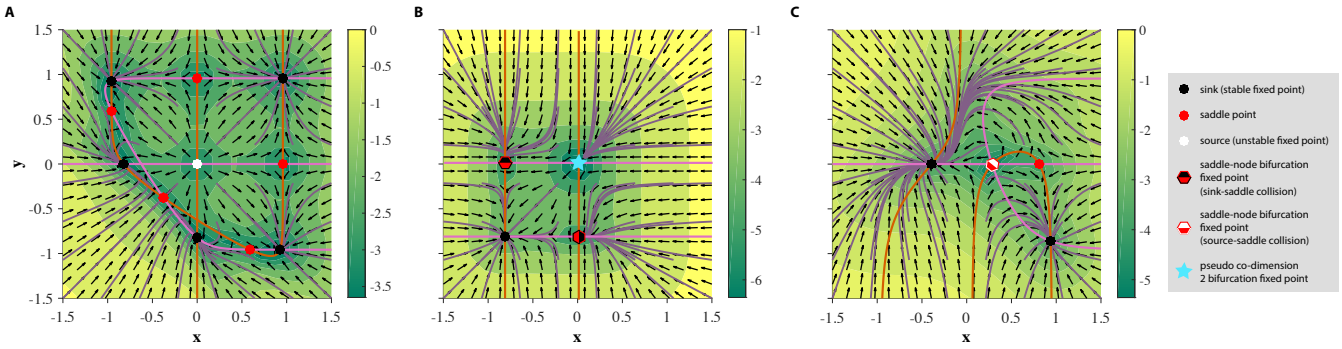

Figure 3. Existence of all observed simple fixed points and bifurcation fixed points with 2D GRUs, depicted in phase space. Orange and pink lines represent the x and y nullclines respectively. Purple lines indicate various trajectories of the hidden state. Direction of the flow is determined by the black arrows, where the colormap underlaying the figure depicts the magnitude of the velocity of the flow in log scale.

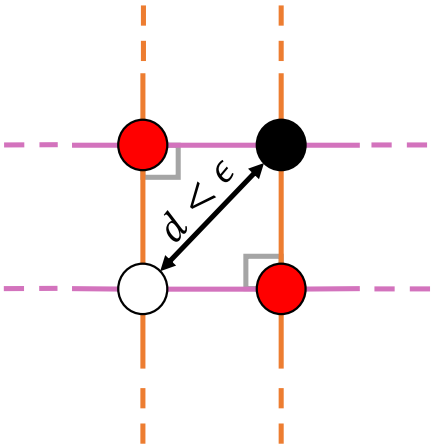

Figure 4. A cartoon representation of the observed *pseudo-codimension-2* bifurcation fixed point. This structure occurs in implementation when placing a sink (top right), a source (bottom left), and two saddle points (top left and bottom right) close enough together, such that the distance between the two points furthest away from one another d is below machine precision ϵ. Under such conditions, the local dynamics behave as a hybridization of all four points. Since at least two parameters need to be adjusted in order to achieve this behavior, we give it the label of *pseudo-codimension-2*; *pseudo* because d can never equal 0 in this system.

The addition of bifurcation fixed points opens the door to dynamically realize more sophisticated models. Take for example the four state system depicted in Fig. 3B. If the hidden state is set to initialize in the first quadrant of phase space (i.e $(0, \infty)^2$), the trajectory will flow towards the pseudo-codimension-2 bifurcation fixed point at the origin. Introducing noise through the input will stochastically cause the trajectory to approach the stable fixed point at $(-1, -1)$ either directly, or by first flowing into one of the two saddle-node bifurcation fixed points of the first kind. Models of this sort can be used in a variety of applications, such as perceptual decision making (Wong and Wang, 2006; Churchland and Cunningham, 2014).

We will begin our investigation into the non-local dynamics observed with 2D GRUs by showing the existence of an Andronov-Hopf bifurcation, where a stable fixed point bifurcates into an unstable fixed point surrounded by a limit cycle. A limit cycle is an attracting set with a well defined basin of attraction. However, unlike a stable fixed point, where trajectories initialized in the basin of attraction flow towards a single point, a limit cycle pulls trajectories into a stable periodic orbit. If the periodic orbit surrounds an unstable fixed point the attractor is *self-exciting*, otherwise it is a *hidden attractor* (Meiss, 2007). While hidden attractors have been observed in various 2D systems, they have not been found in the 2D GRU system, and we conjecture that they do not exist. If all parameters are set to zero except for the hidden state weights, which are parameterized as a rotation matrix with an associated gain, we can introduce rotation into the vector field as a function of gain and rotation angle. Properly tuning these parameters will give rise to a limit cycle; a result of the saturating nonlinearity impeding the rotating flow velocity sufficiently distant from the origin, thereby pulling trajectories towards a closed orbit.

For $\alpha, \beta \in \mathbb{R}^+$ and $s \in \mathbb{R}$,

$$\mathbf{U}_z, \mathbf{U}_r, \mathbf{b}_z, \mathbf{b}_h = 0,\ \mathbf{U}_h = \beta \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}, \mathbf{b}_r = s \tag{8}$$

Let $\beta = 3$ and $s = 0$. If $\alpha = \frac{\pi}{3}$, the system has a single stable fixed point (stable spiral), as depicted in Fig. 5A. If we continuously decrease α, the system undergoes an Andronov-Hopf bifurcation at approximately $\alpha = \frac{\pi}{3.8}$. As α continuously decreases, the orbital period increases, and as the nullclines can be made arbitrarily close together, the length of this orbital period can be set arbitrarily. Fig. 5B shows an example of a relatively short orbital period, and Fig. 5C depicts the behavior seen for slower orbits. If we continue allowing α to decrease, the system will undergo four simultaneous saddle-node bifurcations, and end up in a state topologically equivalent to that depicted in Fig. 2A. Fig. 6-Left depicts regions of the parameter space of (7) parameterized by (8), where the Andronov-Hopf bifurcation manifolds can be clearly seen. Fig.6-Right demonstrates one effect the reset gate can have on the frequency of the oscillations. If we alter the bias vector b_r, the expected oscillation period changes for regions of the $\alpha - \beta$ parameter space which exhibit a limit cycle. Computationally speaking, limit cycles are a common dynamical structure for modeling neuron bursting (Izhikevich, 2007), taking place in many foundational works including the Hodgkin-Huxley model (Hodgkin and Huxley, 1952) and the FitzHugh-Nagumo Model (FitzHugh, 1961). Such dynamics also arise in various population level dynamics in artificial tasks, such as sine wave generation (Sussillo and Barak, 2012). Furthermore, initializing the hidden state matrix U_h of an even dimensional continuous-time RNN (tanh or GRU) with 2×2 blocks along the diagonal and zeros everywhere else is theoretically shown to aid in learning long-term dependencies, when all the blocks act as decoupled oscillators (Sokół et al., 2019).

Regarding the second non-local dynamical feature, it can be shown that a 2D GRU can undergo a homoclinic bifurcation, where a periodic orbit (in this case a limit cycle) expands and collides with a saddle at the bifurcation point. At this bifurcation point the system exhibits a homoclinic orbit, where trajectories initialized on the orbit fall into the same fixed point in both forward and backward time. In order to demonstration this behavior, let the parameters of the network be defined as follows:

For $\gamma \in \mathbb{R}$,

$$\mathbf{U}_z, \mathbf{U}_r, \mathbf{b}_z, \mathbf{b}_r = 0,\ \mathbf{U}_h = 3 \begin{bmatrix} \cos\frac{\pi}{20} & \sin\frac{\pi}{20} \\ -\sin\frac{\pi}{20} & \cos\frac{\pi}{20} \end{bmatrix}, \mathbf{b}_h = \begin{bmatrix} 0.32 \\ \gamma \end{bmatrix} \tag{9}$$

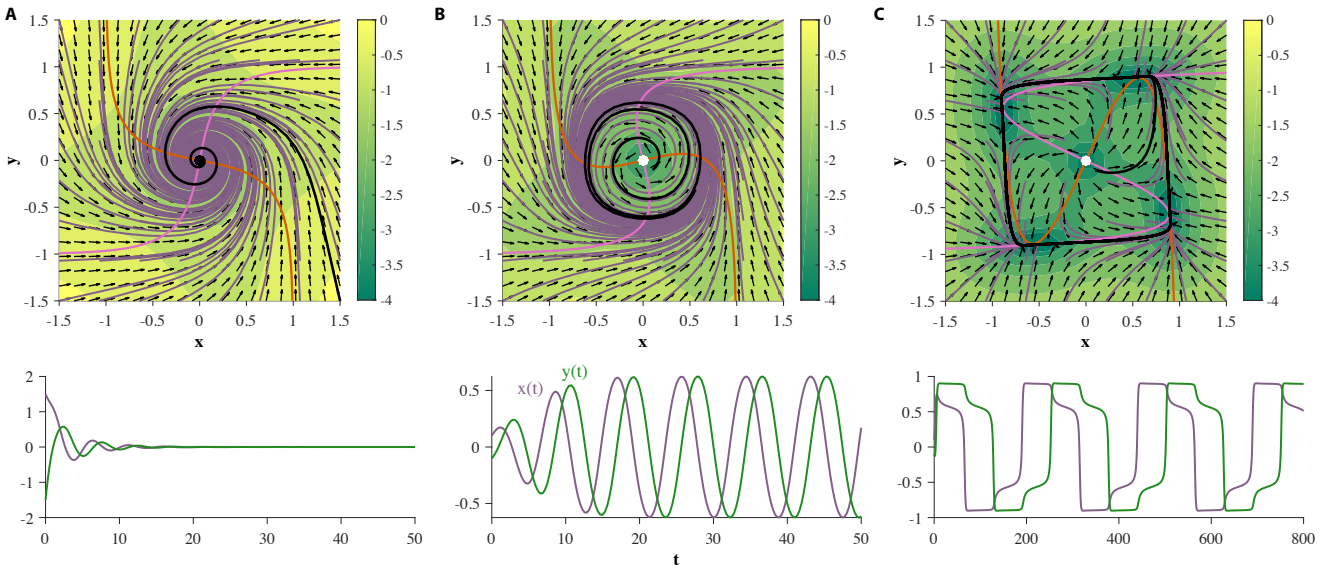

Figure 5. Two GRUs exhibit an Andronov-Hopf bifurcation, where the parameters are defined by (8). When $\alpha = \frac{\pi}{3}$ the system exhibits a single stable fixed point at the origin (Fig. 5A). If α decreases continuously, a limit cycle emerges around the fixed point, and the fixed point changes stability (Fig. 5B). Allowing α to decrease further increases the size and orbital period of the limit cycle (Fig. 5C). The bottom row represents the hidden state as a function of time, for a single trajectory (denoted by black trajectories in each corresponding phase portrait).

Under this parameterization the 2D GRU system exhibits a homoclinic orbit when $\gamma = 0.054085$ (Fig 7). In order to showcase this bifurcation as well as the previous Andronov-Hopf bifurcation sequentially in action we turn to Fig 8, where the parameters are defined by (9) and γ is initialized at 0.051 in Fig 8A.

In addition to proper homoclinic orbits, we observe that 2D GRUs can exhibit one or two bounded planar regions of homoclinic-like orbits for a given set of parameters, as shown in Fig. 9A and 9B respectively. Any trajectory initialized in one of these regions will flow into the pseudo-codimension-2 bifurcation fixed point at the origin, regardless of which direction time flows in. Since the the pseudo-codimension-2 bifurcation fixed point is technically a cluster of four fixed points, including one source and one sink, as demonstrated in Fig. 4, there is actually no homoclinic loop. However, due to the close proximity of these fixed points, trajectories repelled away from the source, but within the basin of attraction of the sink, will appear homoclinic due to the use of finite precision. This featured behavior enables the accurate depiction of various models, including neuron spiking (Izhikevich, 2007).

With finite-fixed point topologies and global structures out of the way, the next logical question to ask is *can 2D GRUs exhibit an infinite number of fixed points?* Such behavior is often desirable in models that require stationary attraction to non-point structures, such as line attractors and ring attractors. Computationally, movement along a line attractor may be interpreted as integration (Mante et al., 2013), and has been shown as a crucial population level mechanism in various tasks, including sentiment analysis (Maheswaranathan et al., 2019a) and decision making (Mante et al., 2013). In a similar light, movement around a ring attractor my computationally represent either modular integration or arithmetic. One known application of ring attractor dynamics in neuroscience is a representation of heading direction (Kim et al., 2017). While such behavior in the continuous GRU system has yet to be seen, an approximation of a line attractor can be made, as depicted in Fig. 10. We will refer to this phenomenon as a *pseudo-line attractor*,

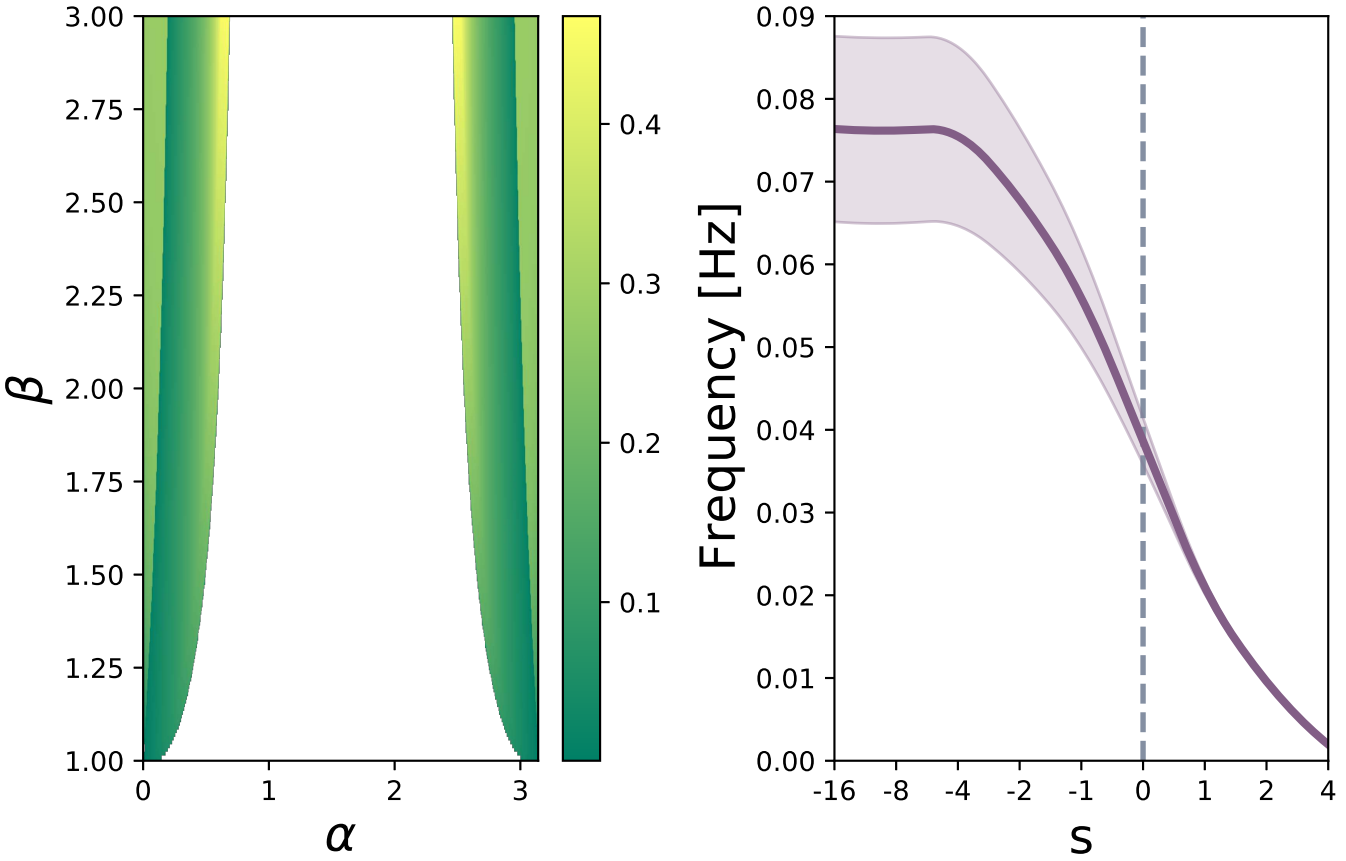

Figure 6. (Fig. 6-Left) parameter sweep of (8) over $\alpha \in (0, \pi)$ (rotation matrix angle) and $\beta \in (1, 3)$ (gain term), for $s = 0$. Color map indicates oscillation frequency in Hertz, where white space shows parameter combinations where no limit cycle exists. (Fig. 6-Right) average oscillation frequency across regions of the displayed $\alpha - \beta$ parameter space where a limit cycle exists. The purple shaded region depicts variance of oscillation frequency. Increasing s slows down the average frequency of the limit cycles, while simultaneously reducing variance.

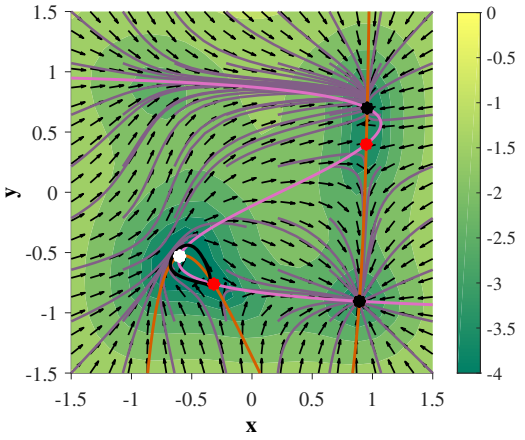

Figure 7. A 2D GRU parameterized by (9) expresses a homoclinic orbit when $\gamma = 0.054085$ (denoted by a black trajectory). Trajectories initialized on the homoclinic orbit will approach the same fixed point in both forward and backward time.

where the nullclines remain sufficiently close on a small finite interval, thereby allowing for arbitrarily slow flow, by means of slow points.

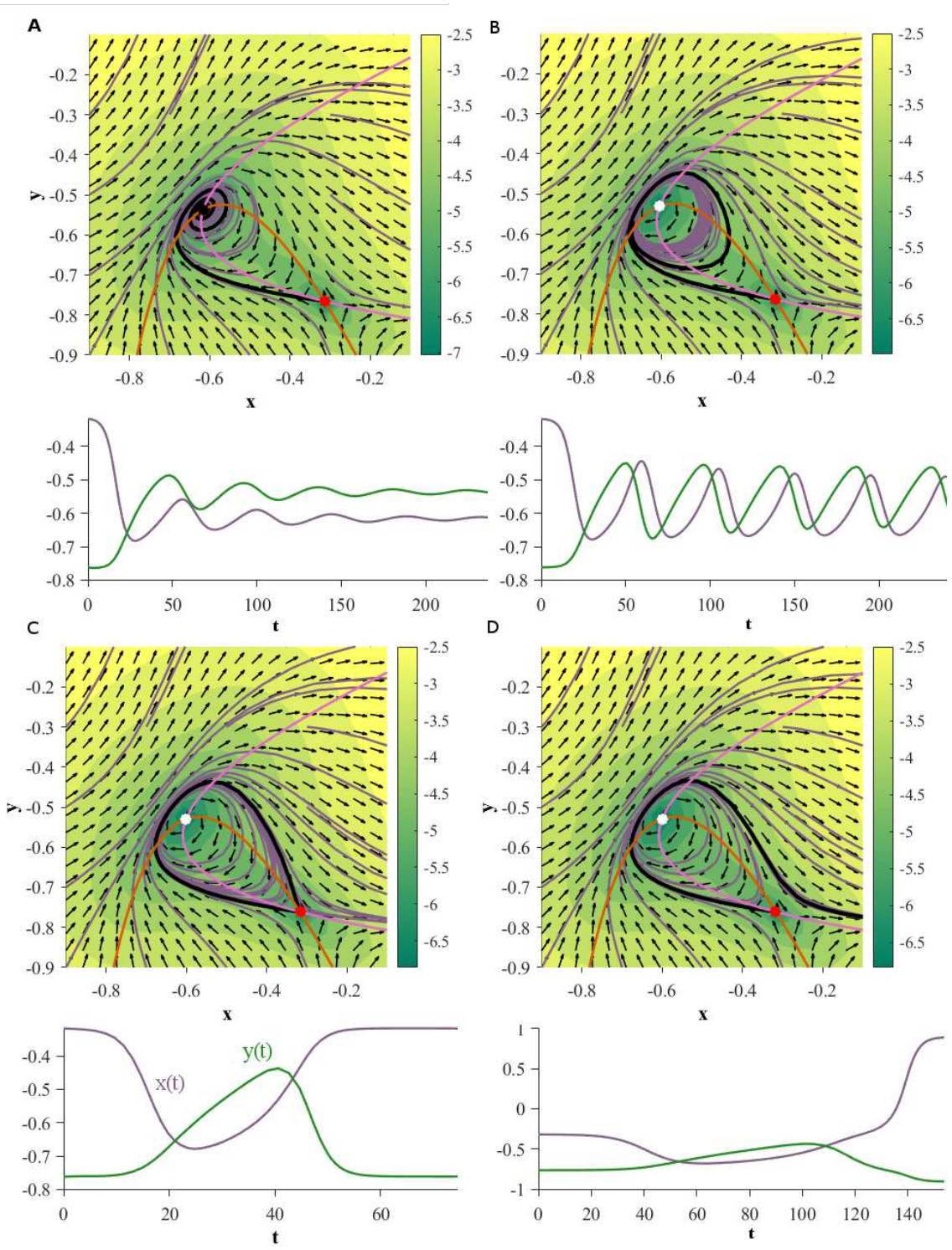

Figure 8. Two GRUs exhibit an Andronov-Hopf bifurcation followed by a homoclinic bifurcation under the same parameterization. The plots directly under each phase portrait depict the time evolution of the black trajectory for the corresponding system. 8A ($\gamma = 0.051$): the system exhibits a stable fixed point. 8B ($\gamma = 0.0535$): the system has undergone an Andronov-Hopf bifurcation and exhibits a stable limit cycle. 8C ($\gamma = 0.054085$): the limit cycle collides with the saddle point, creating a homoclinic orbit. 8D ($\gamma = 0.0542$): the system has undergone a homoclinic bifurcation exhibits neither a homoclinic orbit nor a limit cycle.

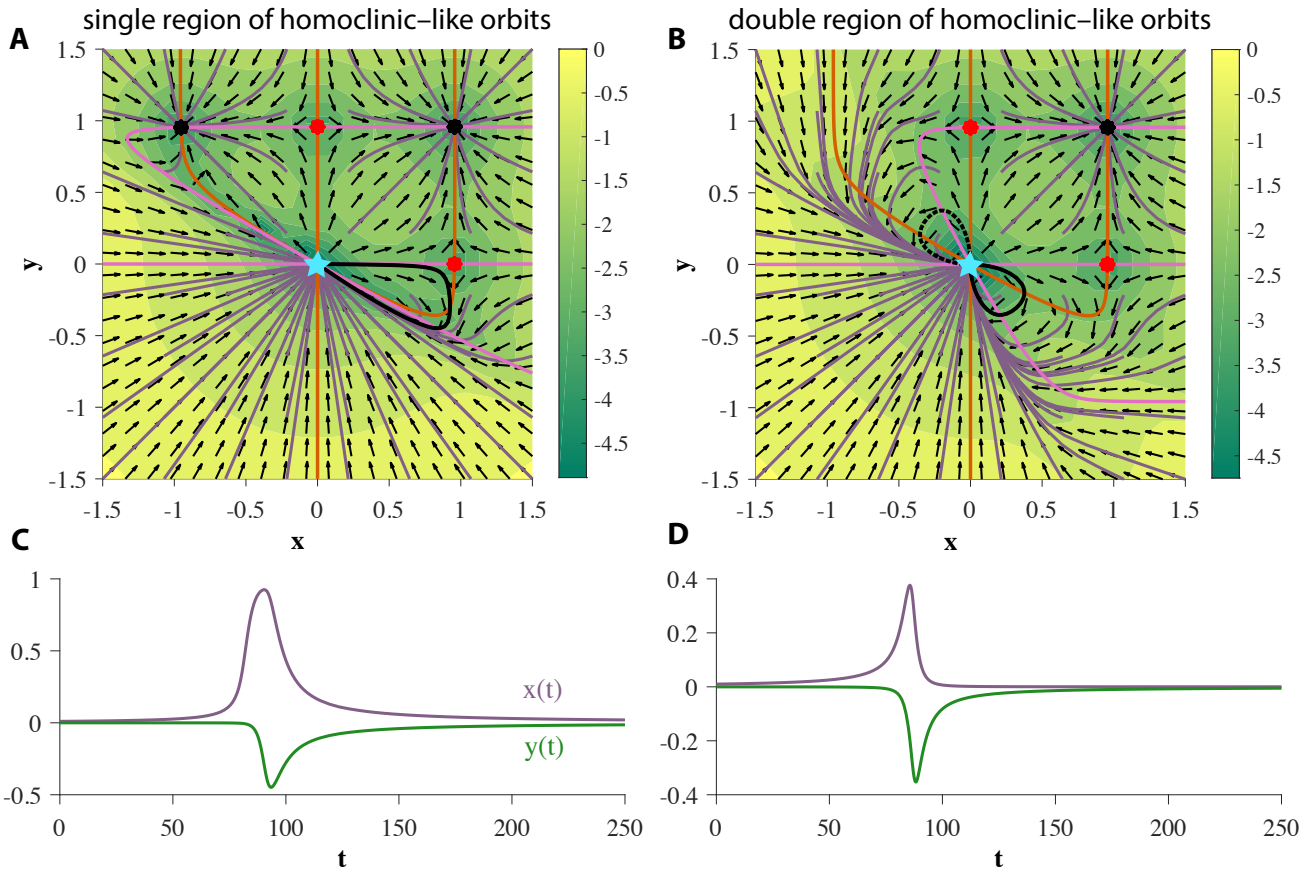

Figure 9. Two GRUs exhibit 2D bounded regions of homoclinic-like behavior. 9C and 9D represent the hidden state as a function of time for a single initial condition within the homoclinic-like region(s) of the single and double homoclinic-like region cases respectively (denoted by solid black trajectories in each corresponding phase portrait).

5 EXPERIMENTS: TIME-SERIES PREDICTION

As a means to put our theory to practice, in this section we explore several examples of time series prediction of continuous time planar dynamical systems using 2D GRUs. Results from the previous section indicate what dynamical features can be learned by this RNN, and suggest cases by which training will fail. All of the following computer experiments consist of an RNN, by which the hidden layer is made up of a 2D GRU, followed by a linear output layer. The network is trained to make a 29-step prediction from a given initial observation, and no further input through prediction. As such, to produce accurate predictions, the RNN must rely solely on the hidden layer dynamics.

We train the network to minimize the following multi-step loss function:

$$\mathcal{L}(\theta) = \frac{1}{T}\sum_{i=1}^{N_{\text{traj}}}\sum_{k=1}^{T} \|\hat{\mathbf{w}}_i(k;\mathbf{w}_i(0)) - \mathbf{w}_i(k)\|_2^2 \tag{10}$$

where θ are the parameters of the GRU and linear readout, $T = 29$ is the prediction horizon, $\mathbf{w}_i(t)$ is the i-th time series generated by the true system, and $\hat{\mathbf{w}}(k;\mathbf{w}_0)$ is the k-step prediction given $\mathbf{w}_0$.

The hidden states are initialized at zero for each trajectory. The RNN is then trained for 4000 epochs, using ADAM (Kingma and Ba, 2014) in whole batch mode to minimize the loss function, i.e., the mean

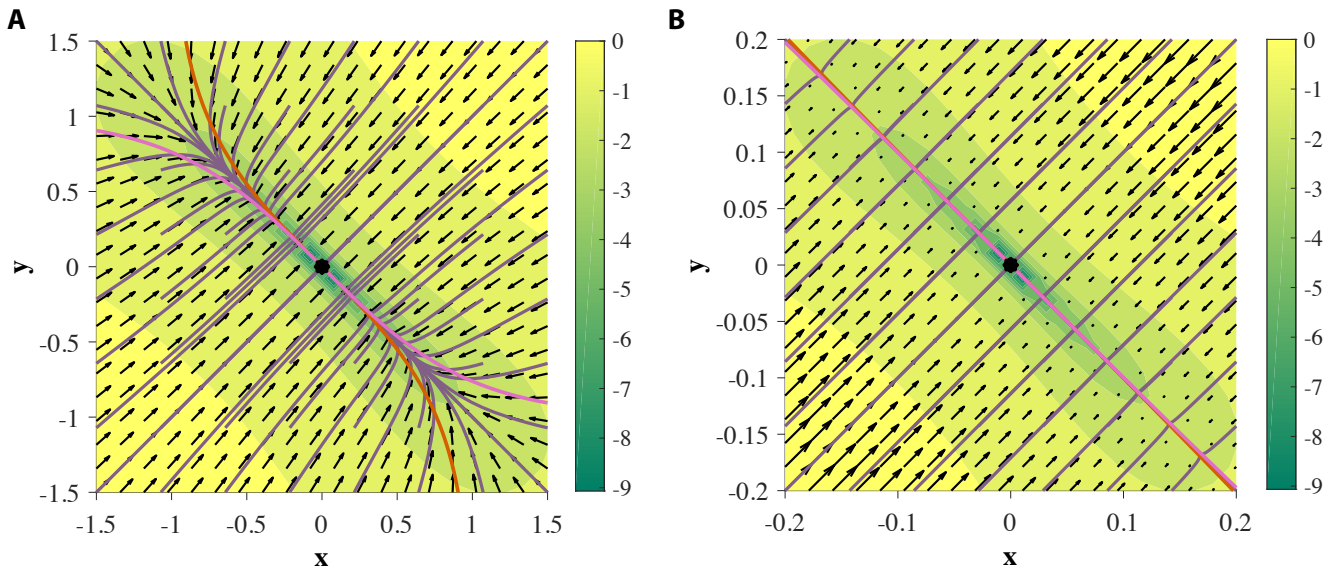

Figure 10. Two GRUs exhibit a pseudo-line attractor. Nullclines intersect at one point, but are close enough on a finite region to mimic an analytic line attractor in practice. 10A and 10B depict the same phase portrait on $[-1.5, 1.5]^2$ and $[-0.2, 0.2]^2$ respectively.

square error between the predicted trajectory and the data. $N_{\text{traj}} = 667$ time series were used for training. Fig. 11 depicts the experimental results of the RNN's attempt at learning each dynamical system we describe below.

5.1 Limit Cycle

To test if 2D GRUs can learn a limit cycle, we use a simple nonlinear oscillator called the FitzHugh-Nagumo Model (FitzHugh, 1961). The FitzHugh-Nagumo model is defined by: $\dot{x} = x - \frac{x^3}{3} - y + I_{\text{ext}},\ \tau\dot{y} = x + a - by$, where in this experiment we will chose $\tau = 12.5$, $a = 0.7$, $b = 0.8$, and $I_{\text{ext}} = \mathcal{N}(0.7, 0.04)$. Under this choice of model parameters, the system will exhibit an unstable fixed point (unstable spiral) surrounded by a limit cycle (Fig. 11). As shown in section 4, 2D GRUs are capable of representing this topology. The results of this experiment verify this claim (Fig. 11), as 2D GRUs can capture topologically equivalent dynamics.

5.2 Line Attractor

As discussed in section 4, 2D GRUs can exhibit a pseudo-line attractor, by which the system mimics an analytic line attractor on a small finite domain. We will use the simplest representation of a planar line attractor: $\dot{x} = -x, \quad \dot{y} = 0$. This system will exhibit a line attractor along the y-axis, at $x = 0$ (Fig. 11). Trajectories will flow directly perpendicular towards the attractor. white Gaussian noise $\mathcal{N}(0, 0.1I)$ in the training data. While the hidden state dynamics of the trained network do not perfectly match that of an analytic line attractor, there exists a small subinterval near each of the fixed points acting as a pseudo-line attractor (Fig. 11). As such, the added affine transformation (linear readout) can scale and reorient this subinterval on a finite domain. Since all attractors in a d-dimensional GRU are bound to $[-1, 1]^d$, no line attractor can extend infinitely in any given direction, which matches well with the GRUs inability to perform unbounded counting, as the continuous analog of such a task would require a trajectory to move along such an attractor.

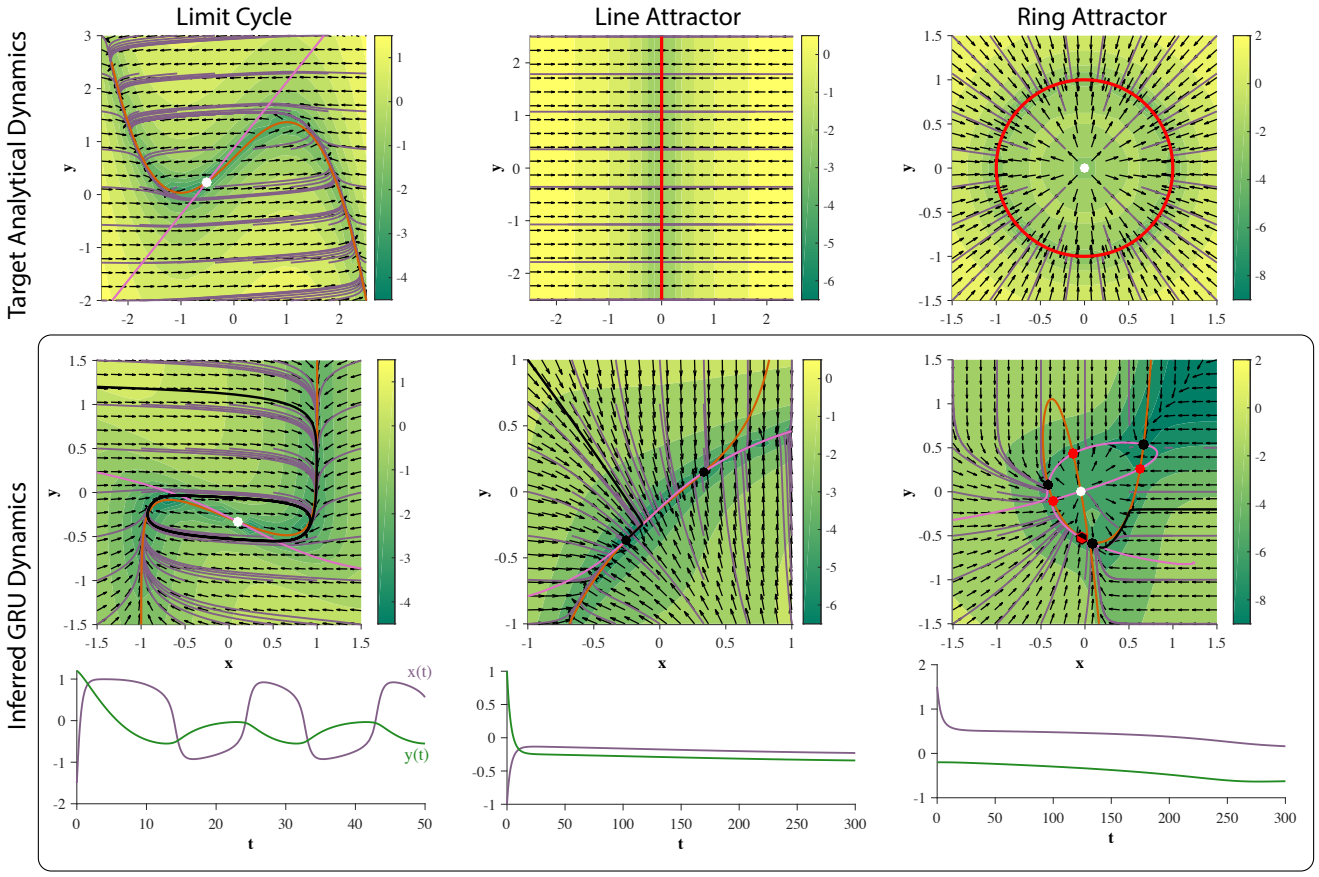

Figure 11. Training 2D GRUs. (top row) Phase portraits of target dynamical systems. Red solid lines represent 1-dimensional attractors. See main text for each system. (middle row) GRU dynamics learned from corresponding 29-step forecasting tasks. The prediction is an affine transformation of the hidden state. (bottom row) An example time series generated through closed-loop prediction of the trained GRU (denoted by a black trajectory). GRU fails to learn the ring attractor.

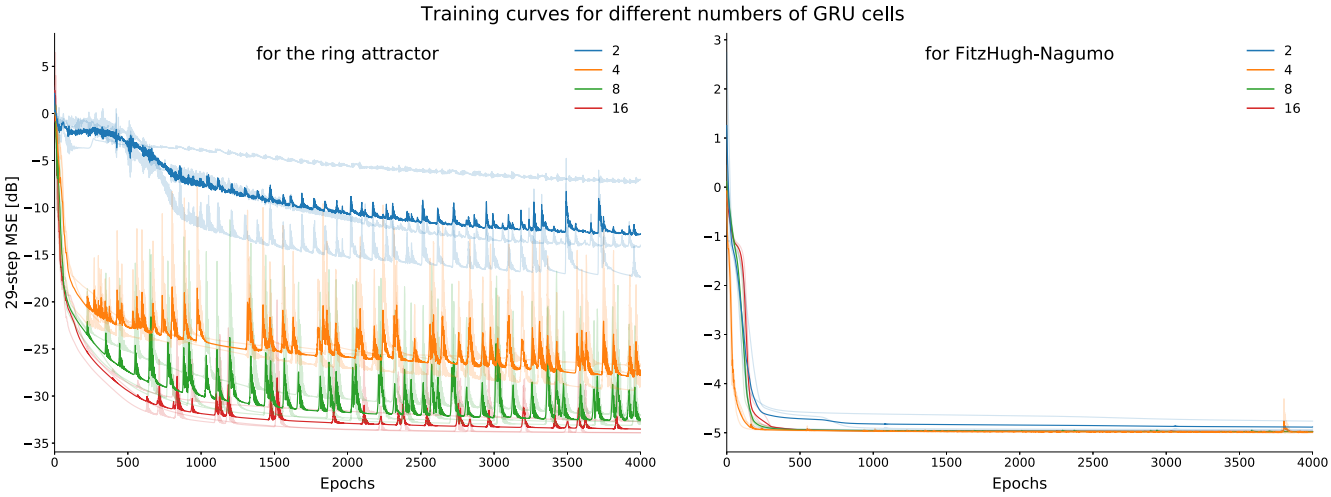

Figure 12. Average learning curves (training loss) for ring attractor (left) and the FitzHugh-Nagumo (right) dynamics. Note that the performance of the ring attractor improves as the dimensionality of the GRU increases unlike the FHN dynamics. Four network sizes (2, 4, 8, 16 dimensional GRU) were trained 3 times with different initializations, depicted by the more lightly colored curves.

5.3 Ring Attractor

For this experiment, a dynamical system representing a standard ring attractor of radius one is used: $\dot{x} = -(x^2 + y^2 - 1)x;\ \dot{y} = -(x^2 + y^2 - 1)y$. This system exhibits an attracting ring, centered around an unstable fixed point. We added Gaussian noise $\mathcal{N}(0, 0.1I)$ in the training data.

In our analysis we did not observe two GRUs exhibit this set of dynamics, and the results of this experiment, demonstrated in Fig. 11, suggest they cannot. Rather, the hidden state dynamics fall into an observed finite fixed point topology (see case xxix in section 3 of the supplementary material). In addition, we robustly see this over multiple initializations, and the quality of approximation improves as the dimensionality of GRU increases (Fig. 12), suggesting that many GRUs are required to obtain a sufficient approximation of this set of dynamics for a practical task (Funahashi and Nakamura, 1993).

6 DISCUSSION

Through example and experiment we indicated classes of dynamics which are crucial in expressing various known neural computations and obtainable with the 2D GRU network. We demonstrated the system's inability to learn continuous attractors, seemingly in any finite dimension, a structure hypothesized to exist in various neural representations. While the GRU network was not originally made as a neuroscientific model, there has been considerable work done showing high qualitative similarity between the underlying dynamics of neural recordings and artificial RNNs on the population level (Mante et al., 2013; Sussillo et al., 2015). Furthermore, recent research has modified such artificial models to simulate various neurobiological phenomenon (Heeger and Mackey, 2019). One recent study demonstrated that trained RNNs of different architectures and nonlinearities express very similar fixed point topologies to one another when successfully trained on the same tasks (Maheswaranathan et al., 2019b), suggesting a possible connection in the dynamics of artificial networks and neural population dynamics. As such, an understanding of the obtainable dynamical features in a GRU network allow one to comment on the efficacy of using such an architecture as an analog of brain dynamics at the population level.

Although this manuscript simplified the problem by considering the 2D GRU, a lot of research has resulted in interpreting cortical dynamics as low dimensional continuous time dynamical systems (Zhao and Park, 2020; Harvey et al., 2012; MacDowell and Buschman, 2020; Flesch et al., 2021; Mante et al., 2013; Cueva et al., 2020). This is not to say that most standard neuroscience inspired tasks can be solved with such a low dimensional network. However, demonstrating that common dynamical features in neuroscience can arise in low dimensions can aid in one's ability to comment on attributes of large networks. These attributes include features such as sparsity of synaptic connections. For example, spiking models exhibiting sparse connectivity have been shown to perform comparatively with fully connected RNNs (Bellec et al., 2018). Additionally, pruning (i.e. removing) substantial percentages of synaptic connections in a trained RNN is known to often result in little to no drop in the network's performance on the task it was trained on (Frankle and Carbin, 2018). This suggests two more examinable properties of large networks. The first is redundancy or multiple realizations of the dynamical mechanisms needed to enact a computation existing within the same network. For example, if only one limit cycle is sufficient to accurately perform a desired task, a trained network may exhibit multiple limit cycles, each qualitatively acting identically towards the overall computation. The second is the robustness of each topological structure to synaptic perturbation/pruning. For example, if we have some dynamical structure, say a limit cycle, how much can we move around in parameter space while still maintaining the existence of that structure?

In a related light, the GRU architecture has been used within more complex machine learning setups to interpret the real-time dynamics of neural recordings (Willett et al., 2021; Pandarinath et al., 2018).

These tools allow researchers to better understand and study the differences between neural responses, trial to trial. Knowledge of the inner workings and expressive power of GRU networks can only further our understanding of the limitations and optimization of such setups by the same line of reasoning previously stated, thereby helping to advance this class of technologies, aiding the field of neuroscience as a whole.

The most compared RNN architecture to the GRU is LSTM, as GRU was designed as both a model and computational simplification of this preexisting design in discrete time implementation. LSTM, for a significant period of time, was arguably the most popular discrete time RNN architecture, outperforming other models of the time on many benchmark tasks. However, there is one caveat when comparing the continuous time implementations of LSTM and GRU. A one dimensional LSTM (i.e. a single LSTM unit) is a two dimensional dynamical system, as information is stored in both the system's hidden state and cell state (Hochreiter and Schmidhuber, 1997). With the choice of analysis we use to dissect the GRU in this paper, LSTM is a vastly different class of system. We would expect to see a different and more limited array of dynamics for an LSTM unit when compared with the 2D GRU. However, we wouldn't consider this a fair comparison.

One attribute of the GRU architecture we chose to disregard in this manuscript was the influence of the update gate $\mathbf{z}(t)$. As stated in section 2, every element of this gate is bound to $(0, 1)^d$. Since (7) only has one term containing the update gate, $(1 - \mathbf{z}(t))$, which can be factored out, the fixed point topology does not depend on $\mathbf{z}(t)$, as this term is always strictly positive. The role this gate plays is to adjust the point-wise speed of flow, and therefore can bring rise to slow manifolds. Because each element of $\mathbf{z}(t)$ can become arbitrarily close to the value of one, regions of phase space associated with an element of the update-gate sufficiently close to one will experience seemingly no motion in the directions associated with those elements. For example, in the 2D GRU system, if the first element of $\mathbf{z}(t)$ is sufficiently close to one, the trajectory will maintain a near fixed value in x. These slow points are not actual fixed points. Therefore, in the autonomous system, trajectories traversing them will eventually overcome this stoppage given sufficient time. However, this may add one complicating factor for analyzing implemented continuous time GRUs in practice. The use of finite precision allows for the flow speed to dip below machine precision, essentially creating *pseudo-attractors* in these regions. The areas of phase space containing these points will qualitatively behave as attracting sets, but not by traditional dynamical systems terms, making them more difficult to analyze. If needed, we recommend looking at $\mathbf{z}(t)$ separately, because this term acts independently from the remaining terms in the continuous time system. Therefore, any slow points found can be superimposed with the traditional fixed points in phase space. In order to avoid the effects of finite precision all together, the system can be realized through a hardware implementation (Jordan and Park, 2020). However, proper care needs to be given in order to mitigate analog imperfections.

Unlike the update gate, we demonstrated that the reset gate $\mathbf{r}(t)$ affects the network's fixed point topology, allowing for more complicated classes of dynamics, including homoclinic-like orbits. These effects are best described through the shape of the nullclines. We will keep things qualitative here as to help build intuition. In 2D, if every element of the reset gate weight matrix $\mathbf{U}_r$ and bias $\mathbf{b}_r$ is zero, nullclines can form two shapes. First is a *sigmoid-like* shape (Fig. 5A, 10, and 11 (inferred limit cycle and line attractor)), allowing them to intersect a line (or hyperplane in higher dimensions) orthogonal to their associated dimension a single time. The second is an *s-like* shape (Fig. 5B, 5C, 7, and 11 (limit cycle)), allowing them to intersect a line orthogonal to their associated dimension up to three times. The peak and trough of the s-like shape can be stretched infinitely as well (Fig. 2A). In this case, two fo the three resultant seemingly disconnected nullclines associated with a given dimension can be placed arbitrarily close together (Fig. 3B). Varying $\mathbf{r}(t)$ allows the geometry of the nullclines to take on several additional shapes. The first of these additional

structures is a *pitchfork-like* shape (Fig. 3A, 3C, 9). By disconnecting two of the *prongs* from the pitchfork we get our second structure, simultaneously exhibiting a sigmoid-like shape and a *U-like* shape (Fig. 3C). Bending the ends of the "U" at infinity down into $\mathbb{R}^2$ connects them, forming our third structure, an *O-like* shape (Fig. 11 (inferred ring attractor – orange nullcline)). This O-like shape can then also intersect the additional segment of the nullcline, creating one continuous curve (Fig. 11 (inferred ring attractor – pink nullcline). One consequence of the reset-gate is the additional capacity to encode information in the form of stable fixed points. If we neglect $\mathbf{r}(t)$, we can obtain up to four sinks (Fig. 2A), as we are limited to the intersections of the nullclines; two sets of three parallel lines. Incorporating $\mathbf{r}(t)$ increases the number of fixed points obtainable (Fig. 3A). Refer to section 3 of the supplementary material to see how these nullcline structures lead to a vast array of different fixed point topologies.

Several interesting extensions to this work immediately come to mind. For one, the extension to a 3D continuous time GRU network opens up the door for the possibility of more complex dynamical features. Three spatial dimensions are the minimum required to experience chaotic dynamics in nonlinear systems (Meiss, 2007), and due to the vast size of the GRU parameter space, even in low dimensions, such behavior is probable. Similarly, additional types of bifurcations may be present, including bifurcations of limit cycles, allowing for more complex oscillatory behavior (Kuznetsov, 1998). Furthermore, higher dimensional GRUs may bring rise to complex center manifolds, requiring center manifold reduction to better analyze and interpret the phase space dynamics (Carr, 1981). While we considered the underlying GRU topology separate from training, considering how the attractor structure influences learning can bring insight into successfully implementing RNN models (Sokół et al., 2019). As of yet, this topic of research is mostly uncharted. We believe such findings, along with the work presented in this manuscript, will unlock new avenues of research on the trainability of recurrent neural networks and help to further understand their mathematical parallels with biological neural networks.

CONFLICT OF INTEREST STATEMENT

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

AUTHOR CONTRIBUTIONS

I.D.J. performed the analysis, I.D.J, P.A.S, and I.M.P wrote the manuscript. P.A.S. performed the numerical experiments. I.M.P. conceived the idea, advised, and edited the manuscript. All authors read and approved the final manuscript.

FUNDING

This work was supported by NIH EB-026946, and NSF IIS-1845836. I.D.J. was supported partially by the Institute of Advanced Computational Science Jr. Researcher Fellowship, at Stony Brook University.

ACKNOWLEDGMENTS

The authors thank Josue Nassar, Brian O'Donnell, David Sussillo, Aminur Rahman, Denis Blackmore, Braden Brinkman, Yuan Zhao, and D.S for helpful feedback and conversations regarding the analysis and writing of this manuscript.

REFERENCES

Costa R, Assael IA, Shillingford B, de Freitas N, Vogels T. Cortical microcircuits as gated-recurrent neural networks. Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, et al., editors,

Advances in Neural Information Processing Systems 30 (Curran Associates, Inc.) (2017), 272–283.
Mante V, Sussillo D, Shenoy K, Newsome W. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature* **503** (2013) 78–84.
Sussillo D, Churchland M, Kaufman MT, Shenoy K. A neural network that finds a naturalistic solution for the production of muscle activity. *Nature Neuroscience* **18** (2015) 1025–1033.
Cueva CJ, Saez A, Marcos E, Genovesio A, Jazayeri M, Romo R, et al. Low-dimensional dynamics for working memory and time encoding. *Proceedings of the National Academy of Sciences* **117** (2020) 23021 – 23032.
Hochreiter S. *Untersuchungen zu dynamischen neuronalen Netzen*. Ph.D. thesis, TU Munich (1991). Advisor J. Schmidhuber.
Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks* **5** (1994) 157–166. doi:10.1109/72.279181.
Hochreiter S, Schmidhuber J. Long Short-Term Memory. *Neural Computation* **9** (1997) 1735–1780. doi:10.1162/neco.1997.9.8.1735.
Cho K, van Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv:1406.1078 [cs, stat]* (2014). ArXiv: 1406.1078.
Prabhavalkar R, Rao K, Sainath TN, Li B, Johnson L, Jaitly N. A Comparison of Sequence-to-Sequence Models for Speech Recognition. *Interspeech 2017* (ISCA) (2017), 939–943. doi:10.21437/Interspeech.2017-233.
Choi K, Fazekas G, Sandler M, Cho K. Convolutional recurrent neural networks for music classification. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2017), 2392–2396. doi:10.1109/ICASSP.2017.7952585.
Dwibedi D, Sermanet P, Tompson J. Temporal Reasoning in Videos Using Convolutional Gated Recurrent Units (2018), 1111–1116.
Pandarinath C, O'Shea DJ, Collins J, Jozefowicz R, Stavisky SD, Kao JC, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature Methods* **15** (2018) 805–815. doi:10.1038/s41592-018-0109-9.
Weiss G, Goldberg Y, Yahav E. On the Practical Computational Power of Finite Precision RNNs for Language Recognition. *arXiv:1805.04908 [cs, stat]* (2018). ArXiv: 1805.04908.
Sussillo D, Barak O. Opening the black box: Low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural Computation* **25** (2012) 626–649. doi:10.1162/NECO_a_00409.
Beer RD. Parameter space structure of continuous-time recurrent neural networks. *Neural Comput.* **18** (2006) 3009–3051.
Zhao Y, Park IM. Interpretable nonlinear dynamic modeling of neural trajectories. *Advances in Neural Information Processing Systems (NIPS)* (2016).
Beer RD. On the dynamics of small Continuous-Time recurrent neural networks. *Adapt. Behav.* **3** (1995) 469–509.
Doya K. Bifurcations of recurrent neural networks in gradient descent learning. *IEEE Trans. Neural Netw.* (1993).
Sokół PA, Jordan I, Kadile E, Park IM. Adjoint dynamics of stable limit cycle neural networks. *2019 53rd Asilomar Conference on Signals, Systems, and Computers* (2019), 884–887. doi:10.1109/IEEECONF44664.2019.9049080.
Pasemann F. A simple chaotic neuron. *Physica D: Nonlinear Phenomena* **104** (1997) 205–211. doi:10.1016/S0167-2789(96)00239-4.

Laurent T, von Brecht J. A recurrent neural network without chaos. *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings* (OpenReview.net) (2017).

He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 770–778. doi:10.1109/CVPR.2016.90.

Heath MT. *Scientific Computing: An Introductory Survey, Revised Second Edition* (Philadelphia: SIAM - Society for Industrial and Applied Mathematics), second edition edn. (2018).

LeVeque RJ, Leveque R. *Numerical Methods for Conservation Laws* (Basel ; Boston: Birkhäuser), 2nd edition edn. (1992).

Thomas JW. *Numerical Partial Differential Equations: Finite Difference Methods* (New York: Springer), 1st ed. 1995. corr. 2nd printing 1998 edition edn. (1995).

Chen RTQ, Rubanova Y, Bettencourt J, Duvenaud DK. Neural ordinary differential equations. Bengio S, Wallach H, Larochelle H, Grauman K, Cesa-Bianchi N, Garnett R, editors, *Advances in Neural Information Processing Systems* (Curran Associates, Inc.) (2018), vol. 31.

Morrill J, Salvi C, Kidger P, Foster J, Lyons T. Neural Rough Differential Equations for Long Time Series. *arXiv:2009.08295 [cs, math, stat]* (2021). ArXiv: 2009.08295.

Meiss J. *Differential Dynamical Systems*. Mathematical Modeling and Computation (Society for Industrial and Applied Mathematics) (2007). doi:10.1137/1.9780898718232.

Kuznetsov YA. *Elements of Applied Bifurcation Theory (2nd Ed.)* (Berlin, Heidelberg: Springer-Verlag) (1998).

Wong KF, Wang XJ. A recurrent network mechanism of time integration in perceptual decisions. *The Journal of Neuroscience: The Official Journal of the Society for Neuroscience* **26** (2006) 1314–1328. doi:10.1523/JNEUROSCI.3733-05.2006.

Churchland MM, Cunningham JP. A Dynamical Basis Set for Generating Reaches. *Cold Spring Harbor Symposia on Quantitative Biology* **79** (2014) 67–80. doi:10.1101/sqb.2014.79.024703.

Izhikevich EM. *Dynamical systems in neuroscience* (MIT press) (2007).

Hodgkin AL, Huxley AF. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology* **117** (1952) 500–544. doi:10.1113/jphysiol.1952.sp004764.

FitzHugh R. Impulses and Physiological States in Theoretical Models of Nerve Membrane. *Biophysical Journal* **1** (1961) 445–466. doi:10.1016/S0006-3495(61)86902-6.

Maheswaranathan N, Williams A, Golub MD, Ganguli S, Sussillo D. Reverse engineering recurrent networks for sentiment classification reveals line attractor dynamics. *arXiv:1906.10720 [cs, stat]* (2019a). ArXiv: 1906.10720.

Kim SS, Rouault H, Druckmann S, Jayaraman V. Ring attractor dynamics in the Drosophila central brain. *Science* **356** (2017) 849–853. doi:10.1126/science.aal4835. Publisher: American Association for the Advancement of Science Section: Reports.

Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]* (2014). ArXiv: 1412.6980.

Funahashi Ki, Nakamura Y. Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Networks* **6** (1993) 801–806. doi:10.1016/S0893-6080(05)80125-X.

Heeger DJ, Mackey WE. Oscillatory recurrent gated neural integrator circuits (ORGaNICs), a unifying theoretical framework for neural dynamics. *Proceedings of the National Academy of Sciences* **116** (2019) 22783–22794. doi:10.1073/pnas.1911633116. Publisher: National Academy of Sciences Section: Biological Sciences.

Maheswaranathan N, Williams A, Golub M, Ganguli S, Sussillo D. Universality and individuality in neural dynamics across large populations of recurrent networks. *Advances in Neural Information Processing Systems* (Curran Associates, Inc.) (2019b), vol. 32.

Zhao Y, Park IM. Variational Online Learning of Neural Dynamics. *Frontiers in Computational Neuroscience* **14** (2020). doi:10.3389/fncom.2020.00071. Publisher: Frontiers.

Harvey CD, Coen P, Tank DW. Choice-specific sequences in parietal cortex during a virtual-navigation decision task. *Nature* **484** (2012) 62–68. doi:10.1038/nature10918. Number: 7392 Publisher: Nature Publishing Group.

MacDowell CJ, Buschman TJ. Low-Dimensional Spatiotemporal Dynamics Underlie Cortex-wide Neural Activity. *Current Biology* **30** (2020) 2665–2680.e8. doi:10.1016/j.cub.2020.04.090.

Flesch T, Juechems K, Dumbalska T, Saxe A, Summerfield C. Rich and lazy learning of task representations in brains and neural networks. *bioRxiv* (2021) 2021.04.23.441128. doi:10.1101/2021.04.23.441128. Publisher: Cold Spring Harbor Laboratory Section: New Results.

Bellec G, Salaj D, Subramoney A, Legenstein R, Maass W. Long short-term memory and learning-to-learn in networks of spiking neurons. *arXiv:1803.09574 [cs, q-bio]* (2018). ArXiv: 1803.09574.

Frankle J, Carbin M. The Lottery Ticket Hypothesis: Training Pruned Neural Networks (2018).

Willett FR, Avansino DT, Hochberg LR, Henderson JM, Shenoy KV. High-performance brain-to-text communication via handwriting. *Nature* **593** (2021) 249–254. doi:10.1038/s41586-021-03506-2. Number: 7858 Publisher: Nature Publishing Group.

Jordan ID, Park IM. Birhythmic analog circuit maze: A nonlinear neurostimulation testbed. *Entropy* **22** (2020). doi:10.3390/e22050537.

Carr J. *Applications of Centre Manifold Theory* (New York, Heidelberg, Berlin: Springer), 1982nd edition edn. (1981).