On the difficulty of learning chaotic dynamics with RNNs

Jonas M. Mikhaeil[1,2,*], **Zahra Monfared**[1,4*], and **Daniel Durstewitz**[1,2,3]

j.mikhaeil@columbia.edu, {zahra.monfared, daniel.durstewitz}@zi-mannheim.de

[1]Department of Theoretical Neuroscience, Central Institute of Mental Health, Medical Faculty Mannheim, Heidelberg University, Mannheim, Germany

[2]Faculty of Physics and Astronomy, Heidelberg University, Heidelberg, Germany

[3]Interdisciplinary Center for Scientific Computing, Heidelberg University

[4]Department of Mathematics & Informatics and Cluster of Excellence STRUCTURES, Heidelberg University, Heidelberg, Germany

[*]These authors contributed equally

Abstract

Recurrent neural networks (RNNs) are wide-spread machine learning tools for modeling sequential and time series data. They are notoriously hard to train because their loss gradients backpropagated in time tend to saturate or diverge during training. This is known as the exploding and vanishing gradient problem. Previous solutions to this issue either built on rather complicated, purpose-engineered architectures with gated memory buffers, or - more recently - imposed constraints that ensure convergence to a fixed point or restrict (the eigenspectrum of) the recurrence matrix. Such constraints, however, convey severe limitations on the expressivity of the RNN. Essential intrinsic dynamics such as multistability or chaos are disabled. This is inherently at disaccord with the chaotic nature of many, if not most, time series encountered in nature and society. It is particularly problematic in scientific applications where one aims to reconstruct the underlying dynamical system. Here we offer a comprehensive theoretical treatment of this problem by relating the loss gradients during RNN training to the Lyapunov spectrum of RNN-generated orbits. We mathematically prove that RNNs producing stable equilibrium or cyclic behavior have bounded gradients, whereas the gradients of RNNs with chaotic dynamics always diverge. Based on these analyses and insights we suggest ways of how to optimize the training process on chaotic data according to the system's Lyapunov spectrum, regardless of the employed RNN architecture.

1 Introduction

Recurrent neural networks (RNNs) are widely used across various fields in engineering and science for learning sequential tasks or modeling and predicting time series [55]. Yet, they struggle when long-term temporal dependencies, very slow, or hugely varying time scales are involved [6, 34, 53, 73, 80]. Time series or sequential data with such properties are, however, very common in fields like climate physics [84], neuroscience [22, 75], ecology [87], or language processing [11]. Training RNNs on such data is hard because the loss gradients backpropagated in time easily saturate or diverge in this process. This is commonly referred to as the exploding and vanishing gradient problem (EVGP) [6, 34, 67].

One solution to the EVGP is based on specifically designed RNN architectures with gating mechanisms, such as long short-term memory (LSTM) [35] or gated recurrent units (GRU) [10]. These architectures allow states at earlier time steps to more easily influence activity much later through a

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

kind of protected memory buffer, thus alleviating the EVGP by structural design. In practice, such models need to be backed up by further techniques like gradient clipping to keep the gradients in check [67]. The relatively complex architectural design of these networks impedes their mathematical analysis and requires reverse engineering after training [57, 63, 64, 80]. Partly to forego these complications, a variety of other solutions has been proposed recently, imposing restrictions on the recurrence matrix to bound the gradients [4, 9], or enforcing global stability by design or regularization [19, 49]. Often these procedures dramatically curtail the expressivity of the RNN [45, 66, 80]; in particular, they rule out chaotic dynamics (for reasons discussed further below).

This is at odds with the plethora of chaotic phenomena in nature, engineering, and society. Chaotic dynamics are commonplace, almost default in any complex physical or biological system. This includes scientific areas as diverse as neuroscience [17, 93], physiology [46], geophysics [81], climate systems [88], astrophysics [52], ecology [14], chemical reactions [21], cell [65] or population [59] biology. Chaotic phenomena are also crucial for the understanding of societal and epidemiological processes, such as the spread of diseases [58], or in economics [20]. They are further relevant in purely technical contexts such as electrical engineering [41, 83] or laser optics [43]. They have even been suggested to play an up to now largely neglected, but potentially very significant role in speech recognition [77] and natural language processing [36]. Hence, in almost any practical setting, chaotic phenomena abound. They cannot, in general, be ignored when devising RNN training algorithms.

Here we offer a comprehensive theoretical treatment of the relation between RNN dynamics and the behavior of the loss gradients during training. We find a close connection between an RNN's loss gradients and the largest Lyapunov exponent of its freely generated orbits. We mathematically prove that RNNs producing stable fixed point or cyclic behavior have bounded gradients. Crucially, however, the loss gradients of RNNs producing chaotic dynamics always diverge. Hence, the chaotic nature of many time series data induces a *principle* problem, and, despite significant efforts in the past to solve the EVGP, training RNNs on such data remains an open issue. We illustrate the implications of our theory for RNN training on several simulated and empirical chaotic time series, and adapt the old idea of *sparsely forced Back-Propagation Through Time (BPTT)* as a simple yet effective remedy that enables to learn the underlying dynamics despite exploding gradients.

2 Related works

Exploding and vanishing gradients. While 'classical' remedies of the EVGP [6, 34, 67] rest on purpose-tailored architectures with gating mechanisms, which safeguard information flow across longer temporal distances [10, 35], the focus has recently shifted to simpler RNNs that address the EVGP by restricting the recurrence matrix to be orthogonal [31, 32, 37], unitary [4], or antisymmetric [9], or by ensuring globally stable fixed point solutions [38, 40], for example through co-trained Lyapunov functions [49]. However, all these approaches impose strong limitations on the dynamical repertoire of the RNN, enforcing global convergence to fixed points or simple cycles.[1] In doing so, they drastically reduce the expressiveness of these models [45, 66]. To address this problem, Erichson et al. [19] somewhat relaxed the constraints on the recurrence matrix by introducing a skew-symmetric decomposition combined with a Lipschitz condition on the activation function. Another recent approach discretizes oscillator ordinary differential equations (ODEs) to arrive at a stable system of coupled [73] or independent [74] oscillators which increase the RNN's expressiveness while bounding its gradients. By design (and as acknowledged by the authors), neither of these architectures is capable of producing chaotic dynamics, however, as the underlying ODEs do not allow for exponential divergence of close-by trajectories (a prerequisite for chaos). Given these often principle limitations of parametrically or dynamically strongly constrained models, a fruitful direction may be to modify the training process itself, e.g. through modified or auxiliary loss functions [80, 85], or special procedures for parameter updating [39] or loss truncation [61, 95]. Our empirical evaluation will follow up on such ideas, but also highlight that simple loss truncation, windowing, or architectural solutions like LSTMs are not sufficient.

Learning dynamical systems. Surprisingly disconnected from the work on the EVGP and learning long-term dependencies, a huge and long-standing literature deals with training RNNs on nonlinear dynamical systems (DS) [70, 86, 89], including chaotic systems like the famous Lorenz equations [56] or chaotic turbulence in fluid dynamics [54, 69]. Of those, methods based on reservoir computing [69] are special in that they start with a large complex dynamical repertoire to begin with for which

[1] We make this point more formal in Appx. A.1.6.

an optimal mapping onto the observations is learned, together with a feedback to the reservoir (see also [8] for issues associated with this strategy in the context of DS reconstruction). Teacher forcing (TF; [13, 70, 95], see also [28]) is one of the earliest techniques introduced to keep RNN trajectories on track while training. The idea behind TF is to simply replace RNN states by observations when available, thereby also effectively cutting off the gradients. TF essentially derives from ideas in dynamical control theory, and adaptive schemes that increasingly hand over control to the RNN throughout training have been devised [1, 2, 5]. A related technique from the control theory literature is "multiple shooting" [92]: Here the whole observed time series is chopped into chunks, and for each chunk of trajectory a new initial condition is estimated. Explicit constraints ensure continuity between the separate trajectory bits during optimization. State space models and the Expectation-Maximization algorithm became popular particularly in the 90es for uncovering the latent dynamics underlying a set of time series observations [23], and remain an important tool until today [16, 50]. Most recently, approaches based on variational inference and the reparameterization trick [48], like sequential variational autoencoders (SVAE), gained in popularity for DS approximation [33, 51]. "Deterministic" RNNs (i.e., with latent states not treated as random variables), like conventional LSTMs [90], remain top choices for DS reconstruction, however.

Although connections between DS ideas and loss gradients have been drawn early on [6], so far only particular scenarios (like fixed point attractors) have been considered. Closest to our work is recent work by Schmidt et al. [80], where non-divergence of loss gradients is established when RNNs converge to fixed points or cycles. However, this was done only for the particular class of piecewise-linear RNNs (PLRNNs), more restrictive conditions for cycles were imposed than assumed here, and - most importantly - the chaotic case on which we focus here was not considered. Recent studies [18, 91] also point out the general connections between Lyapunov exponents and loss gradients that we develop in sect. 3.1, but do not provide any in-depth theoretical treatment, proofs, or empirical evaluation of methods to alleviate exploding gradients in training, as we do here. Thus, a systematic theoretical framework that relates RNN dynamics more generally, and across a range of different RNN architectures, to the behavior of its training gradients, is still lacking so far.

3 Theoretical analysis: Relation between RNN dynamics and loss gradients

In our analysis, we will cover all major types of asymptotic dynamics (fixed points, cycles, chaos, and quasi-periodicity), and mathematically investigate their implications for the loss gradients. We will do this for all major classes of RNNs, including standard RNNs with largely arbitrary activation function, LSTMs, GRUs, and PLRNNs. The next section will first develop and illustrate the basic intuition behind the relations between RNN dynamics and loss gradients.

3.1 Preliminaries: RNN dynamics and loss gradients

Formally, all popular RNN architectures, including LSTMs, GRUs, or PLRNNs, are discrete time DS, defined by a (first-order-Markovian) recursive prescription for the temporal evolution of the latent states $\boldsymbol{z}_t \in \mathbb{R}^M$ of the general form

$$\boldsymbol{z}_t = F_{\boldsymbol{\theta}}(\boldsymbol{z}_{t-1}, \boldsymbol{s}_t), \tag{1}$$

where $\boldsymbol{s}_t \in \mathbb{R}^N$ is the input at time t and $\boldsymbol{\theta}$ are RNN parameters. Map $F_{\boldsymbol{\theta}}$ may be instantiated by any of the common RNN architectures: For instance, for standard RNNs we have $F_{\boldsymbol{\theta}}(\boldsymbol{z}_{t-1}, \boldsymbol{s}_t) = f\big(\boldsymbol{W}\boldsymbol{z}_{t-1} + \boldsymbol{B}\boldsymbol{s}_t + \boldsymbol{h}\big)$, where f is an element-wise activation function like $\tanh$ or a rectified linear unit (ReLU), $\boldsymbol{W}$ a connection matrix, matrix $\boldsymbol{B}$ weighs the inputs, and $\boldsymbol{h}$ is the usual bias term (see sect. A.1.3–A.1.6 for the definition of other RNN models explored here).

Assuming we start at some initial value $\boldsymbol{z}_1 \in \mathbb{R}^M$, and given a sequence of external inputs $\boldsymbol{S} = \{\boldsymbol{s}_t\}$, we can recursively rewrite eq. (1) as

$$\boldsymbol{z}_T = F_{\boldsymbol{\theta}}(F_{\boldsymbol{\theta}}(F_{\boldsymbol{\theta}}(...F_{\boldsymbol{\theta}}(\boldsymbol{z}_1, \boldsymbol{s}_2)...))) \;=:\; F_{\boldsymbol{\theta}}^{T-1}(\boldsymbol{z}_1, \{\boldsymbol{s}_t\}). \tag{2}$$

In DS theory, we characterize the long-term behavior of such sequences by its spectrum of Lyapunov exponents. The Lyapunov exponents estimate the exponential growth rates in different local directions of the system's state space, and the largest Lyapunov exponent gives the dominant exponential behavior. Let us denote the system's Jacobian at time t by

$$\boldsymbol{J}_t \;:=\; \frac{\partial F_{\boldsymbol{\theta}}(\boldsymbol{z}_{t-1}, \boldsymbol{s}_t)}{\partial \boldsymbol{z}_{t-1}} \;=\; \frac{\partial \boldsymbol{z}_t}{\partial \boldsymbol{z}_{t-1}}. \tag{3}$$

Then, the maximum Lyapunov exponent along an RNN trajectory $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_T, \cdots\}$ is defined as

$$\lambda_{max} := \lim_{T\to\infty} \frac{1}{T} \log \left\| \prod_{r=0}^{T-2} \boldsymbol{J}_{T-r} \right\|, \tag{4}$$

where $\|\cdot\|$ denotes the spectral norm (or any subordinate norm) of a matrix. If $\lambda_{max} < 0$ nearby trajectories will ultimately converge to a fixed point or cycle, while for $\lambda_{max} > 0$ (a necessary condition for chaos) initially nearby trajectories will exponentially separate, i.e. we will have divergence along one (or more) directions in state space. This accounts for the sensitive dependence on initial conditions in chaotic systems.

Now let $\mathcal{L}(\boldsymbol{W}, \boldsymbol{B}, \boldsymbol{h})$ be some loss function employed for RNN training that decomposes in time as $\mathcal{L} = \sum_{t=1}^{T} \mathcal{L}_t$. Suppose we fancy BPTT as our training algorithm (similar derivations could be performed for Real Time Recurrent Learning [RTRL]), we recursively develop the loss gradients w.r.t. some RNN parameter θ in time (i.e., across layers of the RNN unrolled in time) as

$$\frac{\partial \mathcal{L}}{\partial \theta} = \sum_{t=1}^{T} \frac{\partial \mathcal{L}_t}{\partial \theta} \quad \text{with} \quad \frac{\partial \mathcal{L}_t}{\partial \theta} = \sum_{r=1}^{t} \frac{\partial \mathcal{L}_t}{\partial \boldsymbol{z}_t} \frac{\partial \boldsymbol{z}_t}{\partial \boldsymbol{z}_r} \frac{\partial^+ \boldsymbol{z}_r}{\partial \theta}, \tag{5}$$

and

$$\begin{aligned} \frac{\partial \boldsymbol{z}_t}{\partial \boldsymbol{z}_r} &= \frac{\partial \boldsymbol{z}_t}{\partial \boldsymbol{z}_{t-1}} \frac{\partial \boldsymbol{z}_{t-1}}{\partial \boldsymbol{z}_{t-2}} \cdots \frac{\partial \boldsymbol{z}_{r+1}}{\partial \boldsymbol{z}_r} \\ &= \prod_{k=0}^{t-r-1} \frac{\partial \boldsymbol{z}_{t-k}}{\partial \boldsymbol{z}_{t-k-1}} = \prod_{k=0}^{t-r-1} \boldsymbol{J}_{t-k}, \end{aligned} \tag{6}$$

where ∂^+ denotes the immediate derivative. Now observe that the behavior of the loss gradients crucially depends on the *product series* of Jacobians in eqn. (6): If the maximum absolute eigenvalues of the Jacobians $\boldsymbol{J}_t$ will, in the geometric mean, be larger than 1 (i.e., $\left\| \prod_{r=0}^{T-2} \boldsymbol{J}_{T-r} \right\|^{1/T} > 1$), gradients will explode as $T \to \infty$, while they will saturate if $\left\| \prod_{r=0}^{T-2} \boldsymbol{J}_{T-r} \right\|^{1/T} < 1$. Thus, the key point to note is that the same terms that occur in the definition of the Lyapunov spectrum, eqn. (4), resurface in the loss gradients, eqn. (5) & (6). This accounts for the tight links between system dynamics and gradients.

3.2 Fixed points and cyclic dynamics

Let us start by considering the simplest types of limit dynamics that can occur in RNNs (or any discrete-time DS): fixed points and cycles. In fact, by far most of the literature on global stability in RNNs and on loss gradients focused on just fixed points [9, 19, 49], with only few authors who recently started to also connect cyclic behavior to loss gradients [73, 80]. Recall that a fixed point of a recursive map $\boldsymbol{z}_t = F(\boldsymbol{z}_{t-1})$ is defined as a point $\boldsymbol{z}^*$ for which we have $\boldsymbol{z}^* = F(\boldsymbol{z}^*)$.[2] Likewise, a k-cycle ($k > 1$) is a set of temporally consecutive periodic points $P_k := \{\boldsymbol{z}_{t_1}, \boldsymbol{z}_{t_2}, \ldots, \boldsymbol{z}_{t_k}\} = \{\boldsymbol{z}_{t_1}, F(\boldsymbol{z}_{t_1}), \ldots, F^{k-1}(\boldsymbol{z}_{t_1})\}$ that we obtain from recursive application of the map such that each of the cyclic points $\boldsymbol{z}_{t_r} \in P_k$ is a fixed point of the k times iterated map F^k (with k being the smallest positive integer for which this holds). To simplify the subsequent treatment, we will collectively refer to fixed points and cycles as k-cycles ($k \geq 1$). Further recall that a fixed point or k-cycle is called *stable* if the maximum absolute eigenvalue of the Jacobian evaluated at that point is smaller than 1, *neutrally stable* if exactly 1, and *unstable* otherwise. Although the results we develop in this and the following sections will hold more widely, we will restrict our attention to recursive maps $F_{\boldsymbol{\theta}}$ from the class of RNNs $\mathcal{R} = \{\texttt{standardRNN}, \texttt{LSTM}, \texttt{GRU}, \texttt{PLRNN}\}$ (see Appx. A.1 for details).

Based on the observations made in the previous sections we can state the following theorem that links RNN dynamics and loss gradients:

Theorem 1. *Consider an RNN $F_{\boldsymbol{\theta}} \in \mathcal{R}$ parameterized by $\boldsymbol{\theta}$, and assume that it converges to a stable fixed point or k-cycle Γ_k ($k \geq 1$) with $\mathcal{B}_{\Gamma_k}$ as its basin of attraction. Then for every $\boldsymbol{z}_1 \in \mathcal{B}_{\Gamma_k}$ (i) the Jacobian $\frac{\partial \boldsymbol{z}_T}{\partial \boldsymbol{z}_1}$ exponentially vanishes as $T \to \infty$; (ii) for Γ_k the tangent vectors $\frac{\partial \boldsymbol{z}_T}{\partial \theta}$ and thus*

[2] From here on we will suppress the explicit dependence on external inputs $\boldsymbol{s}_t$ notation-wise, see Remark 2.

the gradient of the loss function, $\frac{\partial \mathcal{L}_T}{\partial \boldsymbol{\theta}}$, will be bounded from above, i.e. will not diverge for $T \to \infty$; and (iii) for the PLRNN (27) both $\left\| \frac{\partial \boldsymbol{z}_T}{\partial \theta} \right\|$ and $\left\| \frac{\partial \mathcal{L}_T}{\partial \boldsymbol{\theta}} \right\|$ will remain bounded for every $\boldsymbol{z}_1 \in \mathcal{B}_{\Gamma_k}$ as $T \to \infty$.

Proof. (i) Assume that Γ_k is a stable k-cycle $(k \geq 1)$ denoted by

$$\Gamma_k = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_T, \cdots\} = \{\boldsymbol{z}_{t^{*k}}, \boldsymbol{z}_{t^{*k}-1}, \cdots, \boldsymbol{z}_{t^{*k}-(k-1)}, \boldsymbol{z}_{t^{*k}}, \boldsymbol{z}_{t^{*k}-1}, \cdots, \boldsymbol{z}_{t^{*k}-(k-1)}, \cdots\}. \tag{7}$$

Then, the largest Lyapunov exponent of Γ_k is given by

$$\begin{aligned} \lambda_{\Gamma_k} &= \lim_{t\to\infty} \frac{1}{t} \ln \left\| J_t^* \, J_{t-1}^* \, \cdots \, J_2^* \right\| \\ &= \lim_{j\to\infty} \frac{1}{jk} \ln \left\| \left(\prod_{s=0}^{k-1} J_{t^{*k}-s} \right)^j \right\|. \end{aligned} \tag{8}$$

By assumption of stability of Γ_k we have $\lambda_{\Gamma_k} < 0$ and also $\rho\big(\prod_{s=0}^{k-1} J_{t^{*k}-s}\big) < 1$ (the spectral radius), which implies

$$\lim_{t\to\infty} J_t^* \, J_{t-1}^* \, \cdots \, J_2^* = \lim_{j\to\infty} \left(\prod_{s=0}^{k-1} J_{t^{*k}-s} \right)^j = 0. \tag{9}$$

Now suppose that $\mathcal{O}_{\boldsymbol{z}_1}$ is an orbit of the map eqn. (1) converging to Γ_k, i.e. $\boldsymbol{z}_1 \in \mathcal{B}_{\Gamma_k}$. Since $\mathcal{O}_{\boldsymbol{z}_1}$ and Γ_k have the same largest Lyapunov exponent, we have

$$\lambda_{\mathcal{O}_{\boldsymbol{z}_1}} = \lim_{T\to\infty} \frac{1}{T} \ln \left\| J_T \, J_{T-1} \, \cdots \, J_2 \right\| = \lambda_{\Gamma_k} < 0, \tag{10}$$

and hence for $\boldsymbol{z}_1 \in \mathcal{B}_{\Gamma_k}$

$$\lim_{T\to\infty} \left\| \frac{\partial \boldsymbol{z}_T}{\partial \boldsymbol{z}_1} \right\| = \lim_{T\to\infty} \left\| J_T \, J_{T-1} \, \cdots \, J_2 \right\| = 0. \tag{11}$$

(ii) & (iii) See Appx. A.2.1. □

Remark 1. *The result of Theorem 1 part (i) will be generally true for any first-order-Markovian recursive map (1), but the conclusions in part (ii) may hinge on its specific definition.*

Remark 2. *None of the results above and throughout sect. 3 require the dynamics to be autonomous, the theory applies whether there is external input or not. In fact, mathematically, non-autonomous (externally forced) systems can always be rewritten as autonomous dynamical systems [3, 71, 97], see Appx. A.1.1 for details.*

The results above ensure that loss gradients will not diverge (explode) as $T \to \infty$ in RNNs that are "well-behaved" in the sense that they converge to a fixed point or cycle (see Fig. 1a). This is a generalization of the results given in Theorem 1 in Schmidt et al. [80], where this was shown only a) for the specific class of PLRNNs and b) for specific constraints imposed on the eigenvalue spectrum of the RNN's Jacobians which were relaxed in our theorem above.

While our treatment above is centered on the "exploding-gradients" case, various architectural modifications or regularization techniques can ensure that gradients do not vanish either, i.e. remain bounded from below as well. This was established, for instance, in Schmidt et al. [80] for PLRNNs using 'manifold attractor regularization'. In Appx. A.2.1 we show that the results from Theorem 2 from Schmidt et al. [80] on *doubly* bounded (from below and above) loss gradients can indeed be extended to the more general case covered by Theorem 1 above.

3.3 Chaotic dynamics

We will now consider the all-important chaotic case. Let F be a recursive map and $\mathcal{O}_{\boldsymbol{z}_1} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3, \cdots\}$ be an orbit of F. The orbit is chaotic if (i) it is not asymptotically periodic and (ii) has at least one positive Lyapunov exponent [26, 60]. If the system's invariant set is *bounded*, condition (ii) is considered a standard signature of chaos, as in this case two nearby orbits separate exponentially fast, but at the same time their mutual separation cannot go to infinity so that there are also folds. The following theorem states the sufficient condition for exploding gradients:

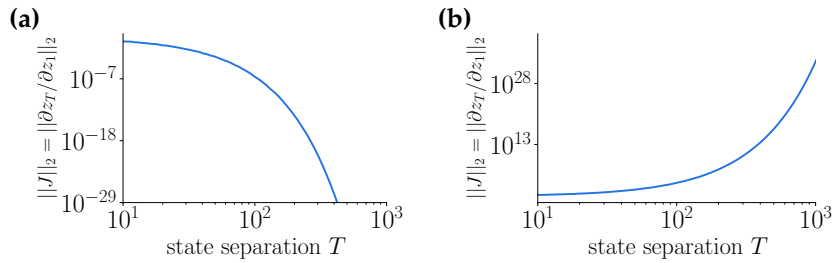


Figure 1: Illustration of the exploding gradient problem when training RNNs on dynamical systems. Jacobians (a) decay away across time separation T for an RNN trained on a simple cycle (cf. Thm. 1), but (b) quickly shoot through the roof when training was performed on chaotic time series (Lorenz system; cf. Thm. 2). Note the doubly-logarithmic scale of these graphs.

Theorem 2. *Suppose that an RNN $F_{\boldsymbol{\theta}} \in \mathcal{R}$ (parameterized by $\boldsymbol{\theta}$) has a chaotic attractor Γ^* with $\mathcal{B}_{\Gamma^*}$ as its basin of attraction. Then, for almost every orbit with $\boldsymbol{z}_1 \in \mathcal{B}_{\Gamma^*}$, (i) the Jacobians connecting temporally distal states $\boldsymbol{z}_T$ and $\boldsymbol{z}_t$ ($T \gg t$), $\frac{\partial \boldsymbol{z}_T}{\partial \boldsymbol{z}_t}$, will exponentially explode for $T \to \infty$, and (ii) the tangent vector $\frac{\partial \boldsymbol{z}_T}{\partial \theta}$ and so the gradients of the loss function, $\frac{\partial \mathcal{L}_T}{\partial \theta}$, will diverge as $T \to \infty$.*

Proof. Let the RNN $F_{\boldsymbol{\theta}} \in \mathcal{R}$ have a chaotic orbit denoted by $\Gamma^* = \{\boldsymbol{z}_1^*, \boldsymbol{z}_2^*, \cdots, \boldsymbol{z}_T^*, \cdots\}$. Then, denoting by J_T^* the Jacobian of (1) at $\boldsymbol{z}_T^* \in \Gamma^*$, the largest Lyapunov exponent of Γ^* is given by

$$\lambda = \lim_{T\to\infty} \frac{1}{T} \ln \left\| J_T^* \, J_{T-1}^* \cdots J_2^* \right\| . \tag{12}$$

Since Γ^* is chaotic, so $\lambda > 0$. Hence, from (12), it is concluded that

$$\lim_{T\to\infty} \left\| J_T^* \, J_{T-1}^* \cdots J_2^* \right\| = \lim_{T\to\infty} \left\| \frac{\partial \boldsymbol{z}_T^*}{\partial \boldsymbol{z}_t^*} \right\| = \infty, \quad T \gg t. \tag{13}$$

Now, according to Oseledec's multiplicative ergodic Theorem, almost all the points in the basin of attraction of Γ^* have the same largest Lyapunov exponent λ. Thus, (13) holds for almost every $\boldsymbol{z}_1 \in \mathcal{B}_{\Gamma^*}$.

(ii) See Appx. A.2.2. □

Remark 3. *The first part of Theorem 2 holds for all first-order-Markovian recursive maps (1). Note that for LSTMs, $\frac{\partial \boldsymbol{z}_T}{\partial \boldsymbol{z}_t}$ ($\boldsymbol{z} := (\boldsymbol{h}, \boldsymbol{c})^\top$) denotes the full Jacobian of both hidden and cell states.*

We collect some further mathematical results and remarks related to Theorem 2 in Appx. A.3.1.

Hence, the essential result is that for all popular RNNs $\mathcal{R}$ and activation functions, loss gradients will inevitably diverge if the RNN latent states converge to a chaotic attractor (as illustrated in Fig. 1b).

3.4 Quasi-periodicity

Quasi-periodicity is a long-term behavior which occurs on a torus and, superficially, bears some similarity to chaos in the sense that, strictly speaking, orbits are also *aperiodic*. That is, as $T \to \infty$, trajectories will never close up with themselves. Moreover, every trajectory becomes arbitrarily close to any point on the torus, that is, it is dense. One important difference between quasi-periodic and chaotic systems is, however, that in a quasi-periodic system, as time passes, two close initial conditions are *linearly* diverging, while in a chaotic system the divergence is exponential.

Theorem 3. *Assume that an RNN $F_{\boldsymbol{\theta}} \in \mathcal{R}$ (parameterized by $\boldsymbol{\theta}$) has a quasi-periodic attractor Γ with $\mathcal{B}_\Gamma$ as its basin of attraction. Then, for every $\boldsymbol{z}_1 \in \mathcal{B}_\Gamma$*

$$\forall\, 0 < \epsilon < 1 \;\; \exists\, T_0 > 1 \;\; s.t. \;\; \forall\, T \geq T_0 \implies$$

$$(1-\epsilon)^{T-1} < \left\| \frac{\partial \boldsymbol{z}_T}{\partial \boldsymbol{z}_1} \right\| < (1+\epsilon)^{T-1}. \tag{14}$$

Proof. See Appx. A.2.3. □

According to Theorem 3, for every orbit converging to a quasi-periodic attractor, the Jacobians $\frac{\partial \boldsymbol{z}_T}{\partial \boldsymbol{z}_t}$ may diverge or vanish as $T \to \infty$, but this will *not* occur exponentially fast as $T \to \infty$. Thus, even for bounded non-chaotic RNNs we may sometimes stumble into the problem of diverging gradients. Although this may be a less common scenario, we point out it may occur if we train RNNs on real data from oscillatory systems with incommensurate frequencies, as for instance encountered in electronic engineering.

In Appx. A.3.2 we have collected further mathematical results on the connection between RNN dynamics and loss gradients that hold regardless of the RNN's limiting behavior.

4 Empirical evaluation

Our theoretical results imply that chaotic time series pose a principle challenge for RNN training that cannot easily be circumvented through specifically designed architectures, constraints, or regularization criteria. If the underlying DS we aim to capture is chaotic, loss gradients propagated back in time will inevitably explode. Hence we need to curtail gradients in an ideal way. The issue arises especially in scientific ML where time series from chaotic systems are ubiquitous and the aim is to *reconstruct* the generating DS with its limiting behavior. Thus, our exposition will focus on this area.

4.1 Training on systems with exploding gradients by sparse teacher forcing

To illustrate the connections between theory and RNN training, we revive the old idea of TF [95] as a mechanism for truncating error gradients and keeping model-generated trajectories on track while training. However, we would like to do this such that important information about the system dynamics does not get lost, for which Lyapunov theory offers some guidance. Specifically, we should not force the system back onto the true trajectory all or most of the time (as in "classical TF"), but should effectively "re-calibrate" it only at certain time points chosen wisely according to the system's local divergence rates. This procedure will be referred to as *sparsely forced BPTT* in the following. Assume we want to train an RNN with hidden states $\boldsymbol{z}_t \in \mathbb{R}^M$ and linear (or affine) output layer on a time-series $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_T\}$ generated by a chaotic system.[3] The linear output layer $\hat{\boldsymbol{x}}_t = \boldsymbol{B}\boldsymbol{z}_t$, $\boldsymbol{B} \in \mathbb{R}^{N \times M}$, maps the RNN hidden states into the observation space. This allows us to modify the original TF procedure by constructing a control series $\{\tilde{\boldsymbol{z}}_1, \tilde{\boldsymbol{z}}_2, \cdots, \tilde{\boldsymbol{z}}_T\}$ from the observations by "inverting" the linear output mapping [4]

$$\tilde{\boldsymbol{z}}_t = (\boldsymbol{B}^\mathsf{T}\boldsymbol{B})^{-1}\boldsymbol{B}^\mathsf{T}\boldsymbol{x}_t. \tag{15}$$

The idea is to supply this control signal only sparsely, separated by the learning interval τ between consecutive forcings. Hence, defining $\mathcal{T} = \{n\tau + 1\}_{n \in \mathbb{N}_0}$ as the set of all time points at which we force the RNN onto the 'true' values, the RNN updates can be written as

$$\boldsymbol{z}_{t+1} = \begin{cases} RNN(\tilde{\boldsymbol{z}}_t) & \text{if } t \in \mathcal{T} \\ RNN(\boldsymbol{z}_t) & \text{else} \end{cases}. \tag{16}$$

This forcing is applied *after* calculation of the loss, such that $\mathcal{L}_t = \|\boldsymbol{x}_t - \boldsymbol{B}\boldsymbol{z}_t\|_2^2$ irrespective of whether t is in $\mathcal{T}$ or not (and of course it is applied only during training, not at test time!). Replacing hidden states $\boldsymbol{z}_t$ with their teacher-forced signals $\tilde{\boldsymbol{z}}_t$ simply breaks divergence between true and predicted trajectories at time points $t \in \mathcal{T}$, and also cuts off the Jacobians by breaking the temporal contingency (for details see Appx. A.7). The learning interval τ hence controls how many time steps are included in the gradient calculation and has to be chosen with care such as to balance the effects of exploding gradients vs. those of losing relevant time scales and long-term dependencies. While it is general wisdom that an optimal batch size will facilitate training,[5] the point here is thus much

[3] Note that in DS reconstruction one usually considers the data as observations (*unsupervised* problem).

[4] To ensure invertibility, one could add a regularizer $\lambda\mathbf{I}$ to $\boldsymbol{B}^\mathsf{T}\boldsymbol{B}$ in eqn. (15), as in ridge regression, but we did not find this necessary in any of our examples.

[5] It is also reminiscent of truncated BPTT, but with the all-important differences that we suggest 1) a theoretically informed choice of the optimal 'truncation length' (forcing interval) and 2) a specific procedure for replacing current latent states by control values. As shown in sect. 4.2 & 4.3, both these aspects are indeed crucial to avoid diverging gradients and trajectories whilst not loosing relevant longer time scales.

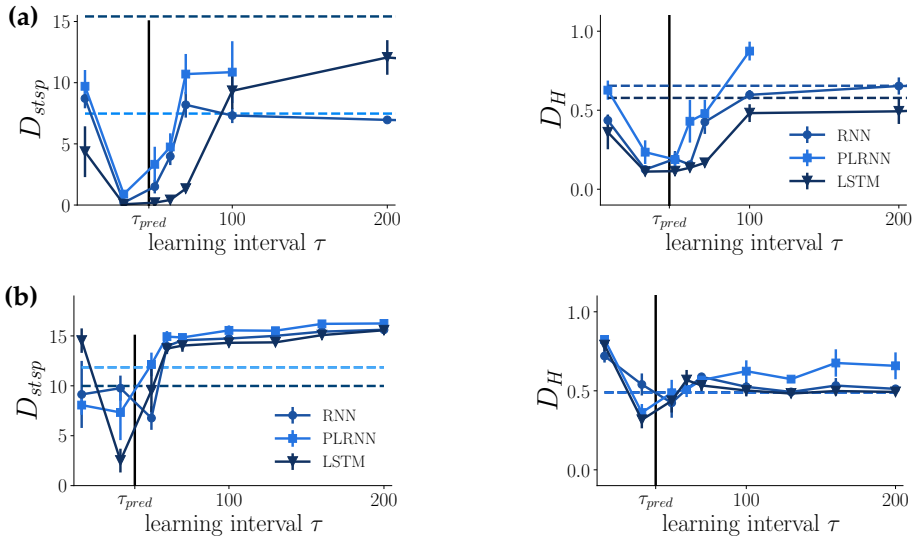


Figure 2: Overlap in attractor geometry (D_{stsp}, lower = better) and dimension-wise comparison of power-spectra (D_H, lower = better) against learning interval τ for (a) the Lorenz and (b) the chaotically forced Duffing oscillator. Continuous lines = sparsely forced BPTT. Dashed lines = classical BPTT with gradient clipping. Prediction time indicated vertically in black.

more specific: Ideally τ should be chosen in accordance with the system's Lyapunov spectrum, for instance based on the *predictability time* [7]

$$\tau_{\text{pred}} = \frac{\ln 2}{\lambda_{\max}}. \tag{17}$$

Various open-source packages exist for calculating the maximal Lyapunov exponent $\lambda_{\max}$ from empirical time series data (e.g., Julia: DynamicalSystems.jl [12], C++: TISEAN [30]), potentially after delay-embedding the data (see Appx. A.4 for more details). Note that this needs to be done on the *empirical* data and only once *before* RNN training, as $\lambda_{\max}$ is an invariant characteristic of the empirical system that we are aiming to reconstruct by our RNN (i.e., after successful training the RNN should have the same invariant properties as the underlying DS). We also emphasize that such a simple recipe for addressing the exploding gradient problem is based on modifying the training routine, and is thus in principle applicable to any model architecture.

4.2 Example 1: Lorenz system and externally forced Duffing oscillator in chaotic regime

Let us illustrate these ideas on two classical textbook examples of chaotic DS, the chaotic Lorenz attractor as an autonomous system, and the chaotically forced Duffing oscillator as an example with explicit external input (see Appx. A.4 for details). Trajectories were repeatedly drawn from these systems, on which we trained a PLRNN, a vanilla RNN with $\tanh$ activation function, and a LSTM by stochastic gradient descent (SGD) to minimize the MSE loss between predicted and actual observations. As optimizer we used Adam [47] from PyTorch [68] with a learning rate of 0.001. For all models, training proceeded solely by *sparsely forced BPTT* and did not employ gradient clipping or any other technique that may interfere with optimal loss truncation.

In nonlinear DS reconstruction, we are mainly interested in reproducing *invariant* properties of the underlying system such as the attractor geometry (or topology [78, 82]) or the frequency composition (i.e., time-independent properties), while measures like ahead-prediction errors are less meaningful especially on chaotic time series [50, 96]. Thus, in evaluating training performance, here we follow Koppe et al. [50] in using a Kullback-Leibler divergence D_{stsp} to quantify the agreement between observed and generated probability distributions across state-space to asses the overlap in attractor geometry (Appx. A.5). Moreover, we calculate the dimension-wise Hellinger distance D_H between power spectra to quantify the temporal agreement of the observed and generated time-series (Appx. A.5).

Fig. 2 shows the dependence of the reconstruction quality on the learning interval τ for all RNN architectures on (a) the Lorenz and (b) the externally forced Duffing system. Fig. 3 provides particular examples of reconstructions for τ chosen too small, too large, or about right.

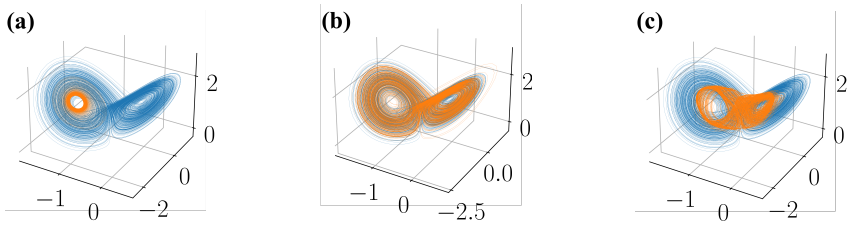


Figure 3: Lorenz attractor (blue) and example reconstructions by an LSTM (orange) trained with a learning interval (a) chosen too small ($\tau = 5$), (b) chosen optimally ($\tau = 30$), and (c) chosen too large ($\tau = 200$). See Fig. 14 for a vanilla RNN example.

For all models we find a system-dependent range for the optimal learning interval that agrees well with the predictability time defined in eqn. (17), where estimates for the maximal Lyapunov exponent were taken from the literature [24, 72]. As a reference, dashed lines represent the reconstruction performance for all architectures when trained with classical BPTT and gradient clipping. The training procedure was the same as for sparsely forced BPTT, except that instead of supplying a control-signal, gradients were normalized to 1 prior to each parameter update (see Fig. 13 for a more systematic evaluation of different clipping procedures and thresholds). As evidenced by the much worse performance, gradient clipping does not effectively address the EVGP, *even for LSTMs*. As further shown in Fig. 10 in Appx. A.6.4, using the optimal (or any other) window length τ but resetting the initial condition for each chunk to either zero or the last forward-iterated state $F_{\boldsymbol{\theta}}(\boldsymbol{z}_{t-1})$ (instead of its control value $\tilde{\boldsymbol{z}}_t$) equally destroys performance. This suggests that neither mere gradient normalization nor simple windowing are sufficient, but will wipe out essential information about the dynamics.

In Appx. A.6 we collect further results on the chaotic Rössler attractor (Fig. 6 & 7), high-dimensional Mackey-Glass equations (Fig. 8), and the Lorenz attractor with partial observations (Fig. 9).

4.3 Example 2: Chaotic weather data

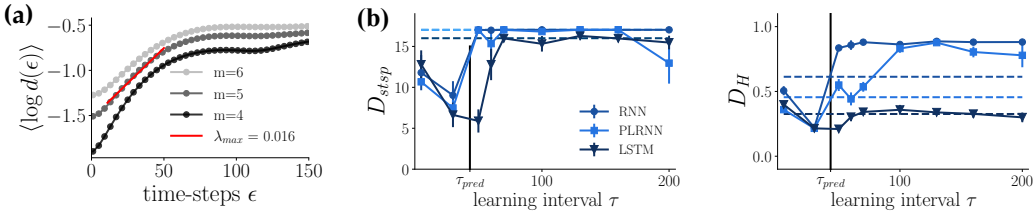


Figure 4: (a) The maximal Lyapunov exponent was determined as the slope of the average log-divergence of nearest neighbors in embedding space (m = embedding dimension). (b) Reconstruction quality assessed by attractor overlap (lower = better) and dimension-wise comparison of power-spectra (D_H, lower = better). Black vertical lines = τ_{pred}.

As for an empirical example, we trained all RNNs (vanilla RNN, PLRNN, LSTM) on a temperature time series recorded at the Weather Station at the Max Planck Institute for Biogeochemistry in Jena, Germany. To expose the chaotic behavior and obtain a robust estimate of the maximal Lyapunov exponent, trends and yearly cycles were removed, and nonlinear noise-reduction was performed ([43]; Appx. A.4). The maximal Lyapunov exponent was determined with the *TISEAN* package [30], as shown in Figure 4 (a). The value obtained is in close agreement with the literature ($\lambda_{max} \approx 0.017$ [62]).

Figure 4 shows that also for these empirical data the optimal training interval τ agrees well with the predictability time, eqn. (17), for all trained RNNs. Furthermore, as was the case for the DS benchmarks, gradient clipping was not able to satisfactorily tackle the EVGP, even when paired with

architectures like LSTMs explicitly designed for alleviating this problem. Similar results are reported for another real-world dataset, electroencephalogram (EEG) recordings, in Appx. A.6.5.

5 Discussion and conclusions

In this paper we proved that RNN dynamics and loss gradients are intimately related for all major types of RNNs and activation functions. If the RNN is "well behaved" in the sense that its dynamics converges to a fixed point or cycle, loss gradients will remain bounded, and established remedies [35, 80] can be used to refrain them from vanishing. However, if the dynamics are chaotic, gradients will always explode. This constitutes a *principle* problem in RNN training that cannot easily be mastered through architectural design or gradient clipping. This is because to avoid exploding gradients while training on time series from chaotic systems, one either needs to constrain the RNN so much that chaotic behavior is completely disabled to begin with (i.e., ultimately by forcing all Lyapunov exponents to be smaller or equal to zero), implying a very poor fit to such data. Or one needs to be a bit more lenient and thereby allow for the possibility of exploding gradients (as LSTMs or PLRNNs in fact do). This problem is furthermore practically highly relevant, as most time series we encounter in nature, and many from man-made systems as well, are inherently chaotic.

While we do not offer a full solution to this problem here, we suggest it might be tackled in training by taking a system's local divergence rates as measured through the Lyapunov spectrum into account. Hence, rather than conquering the EVGP by structural design or specific constraints or regularization terms, we recommend to put the focus more on the training process itself. We illustrated this point empirically using *sparsely forced BPTT*, a training technique that pulls trajectories back on track at times determined by the maximal Lyapunov exponent. Doing so leads to optimal reconstruction results for a variety of simulated and real-world benchmarks, regardless of the specific RNN architecture employed in training.

As noted in sect. 4.1, fairly standard packages are available for computing maximal Lyapunov exponents from data. Some background knowledge, as provided in classical textbooks (e.g. Ch. 5 in [42]), may be required for properly reading the output from these packages: Essentially, one would be looking be for a linear scaling region as in Figs. 4a & 12a, ignoring both the initial noise transient as well as the plateau caused by reaching the full attractor extent. If unsure about the exact value, a moderate amount of jittering around the estimated mean value may help (see Appx. Fig. 15). A further interesting direction for improvement might be to regulate the forcing interval through an annealing procedure [1, 2], for instance starting at $\tau = 1$ and ramping up to $\tau = \tau_{\text{pred}}$ throughout training, similar as in adaptive schemes [5].[6] The idea here would be to first get the short-term behavior right, and then challenge the system more and more for longer time spans until the predictability time is reached.

We stress that our goal here above all was to provide a mathematically grounded perspective on the problem, with the empirical section focused on elucidating the practical implications of the theoretical results. We believe that a more thorough theoretical understanding is important and needed for guiding future research into more powerful training procedures that avoid exploding gradients *without* compromising expressiveness. In our application examples, we developed the case from the perspective of scientific machine learning, which by now is a broad area in its own right with huge societal relevance (e.g., climate or epidemiological time series), and where the reconstruction of geometrical or topological (invariant) properties is important, beyond mere prediction. Nevertheless, we believe that our theoretical results will also have implications for other domains, like NLP [36]. While scientific time series problems traditionally have been extensively considered from a DS perspective (e.g., [42]), much more groundwork is needed, however, in areas like NLP, where, for instance, it may not even be immediately clear how to best define a Lyapunov spectrum.

All code from this paper is available at `https://github.com/DurstewitzLab/ChaosRNN`.

Acknowledgements

This work was funded by the German Research Foundation (DFG) under Germany's Excellence Strategy – EXC-2181 – 390900948 (STRUCTURES), and through grant Du 354/10-1 to DD.

[6] We thank one of the referees for pointing this out.

References

[1] H. D. I. Abarbanel. *Predicting the Future: Completing Models of Observed Complex Systems*. Understanding Complex Systems. Springer-Verlag, 2013.

[2] H. D. I. Abarbanel, P. J. Rozdeba, and S. Shirman. "Machine Learning: Deepest Learning as Statistical Data Assimilation Problems". In: *Neural Computation* 30.8 (2018), pp. 2025–2055.

[3] K. T. Alligood, T. D. Sauer, and J. A. Yorke. *Chaos: An Introduction to Dynamical Systems*. Springer, New York, NY, 1996.

[4] M. Arjovsky, A. Shah, and Y. Bengio. "Unitary Evolution Recurrent Neural Networks". In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning*. Vol. 48. ICML'16. New York, NY, USA: JMLR, 2016, pp. 1120–1128.

[5] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer. "Scheduled sampling for sequence prediction with recurrent Neural networks". In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'15. Cambridge, MA, USA: MIT Press, 2015, pp. 1171–1179.

[6] Y. Bengio, P. Simard, and P. Frasconi. "Learning long-term dependencies with gradient descent is difficult". In: *IEEE Transactions on Neural Networks* 5.2 (1994), pp. 157–166.

[7] B. P. Bezruchko and D. A. Smirnov. *Extracting Knowledge From Time Series: An Introduction to Nonlinear Empirical Modeling*. Springer Series in Synergetics. Berlin Heidelberg: Springer-Verlag, 2010. DOI: 10.1007/978-3-642-12601-7.

[8] M. Brenner, F. Hess, J. M. Mikhaeil, L. F. Bereska, Z. Monfared, P.-C. Kuo, and D. Durstewitz. "Tractable Dendritic RNNs for Reconstructing Nonlinear Dynamical Systems". In: *Proceedings of the 39th International Conference on Machine Learning*. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 2292–2320.

[9] B. Chang, M. Chen, E. Haber, and E. H. Chi. "AntisymmetricRNN: A Dynamical System View on Recurrent Neural Networks". In: *International Conference on Learning Representations*. 2019.

[10] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. "On the Properties of Neural Machine Translation: Encoder–Decoder Approaches". In: *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 103–111.

[11] K. Cho, B. van Merriënboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*. 2014.

[12] G. Datseris. "DynamicalSystems.jl: A Julia software library for chaos and nonlinear dynamics". In: *Journal of Open Source Software* 3.23 (2018), p. 598.

[13] K. Doya. "Bifurcations in the learning of recurrent neural networks". In: *[Proceedings] 1992 IEEE International Symposium on Circuits and Systems*. Vol. 6. 1992, pp. 2777–2780.

[14] J. Duarte, C. Januário, N. Martins, and J. Sardanyés. "Quantifying chaos for ecological stoichiometry". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 20.3 (Sept. 2010), p. 033105.

[15] G. Duffing. *Forced vibrations with variable natural frequencies and their technical significance*. F. Vieweg und Sohn, Braunschweig, 1918.

[16] D. Durstewitz. *Advanced Data Analysis in Neuroscience: Integrating Statistical and Computational Models*. Bernstein Series in Computational Neuroscience. Springer International Publishing, 2017.

[17] D. Durstewitz and T. Gabriel. "Dynamical Basis of Irregular Spiking in NMDA-Driven Prefrontal Cortex Neurons". In: *Cerebral Cortex* 17.4 (2007), pp. 894–908.

[18] R. Engelken, F. Wolf, and L. F. Abbott. "Lyapunov spectra of chaotic recurrent neural networks". In: *arXiv:2006.02427 [nlin, q-bio]* (2020).

[19] N. B. Erichson, O. Azencot, A. Queiruga, L. Hodgkinson, and M. W. Mahoney. "Lipschitz Recurrent Neural Networks". In: *International Conference on Learning Representations*. 2021.

[20] M. Faggini. "Chaotic time series analysis in economics: Balance and perspectives". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 24.4 (2014). Publisher: American Institute of Physics, p. 042101.

[21] R. J. Field and L. Györgyi. *Chaos in Chemistry and Biochemistry*. World Scientific, 1993.

[22] S. Fusi, W. F. Asaad, E. K. Miller, and X.-J. Wang. "A neural circuit model of flexible sensorimotor mapping: learning and forgetting on multiple timescales". In: *Neuron* 54.2 (2007), pp. 319–333.

[23] Z. Ghahramani and S. Roweis. "Learning Nonlinear Dynamical Systems Using an EM Algorithm". In: *Advances in Neural Information Processing Systems*. Vol. 11. MIT Press, 1999.

[24] W. Gilpin. "Chaos as an interpretable benchmark for forecasting and data-driven modelling". In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. 2021.

[25] L. Glass and M. C. Mackey. "Pathological conditions resulting from instabilities in physiological control systems". In: *Annals of the New York Academy of Sciences* 316 (1979), pp. 214–235.

[26] P. A. Glendinning and D. J. W. Simpson. "A constructive approach to robust chaos using invariant manifolds and expanding cones". In: *Discrete & Continuous Dynamical Systems* 41.7 (2021), pp. 3367–3387.

[27] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. "PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals". In: *Circulation* 101.23 (2000), e215–e220.

[28] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[29] A. Graves, G. Wayne, M. Reynolds, and et al. "Hybrid computing using a neural network with dynamic external memory". In: *Nature* 538 (2016), pp. 471 –476.

[30] R. Hegger, H. Kantz, and T. Schreiber. "Practical implementation of nonlinear time series methods: The TISEAN package". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 9.2 (1999), pp. 413–435. DOI: 10.1063/1.166424.

[31] K. Helfrich, D. Willmott, and Q. Ye. "Orthogonal Recurrent Neural Networks with Scaled Cayley Transform". In: *International Conference on Machine Learning*. PMLR, 2018, pp. 1969–1978.

[32] M. Henaff, A. Szlam, and Y. LeCun. "Recurrent Orthogonal Networks and Long-Memory Tasks". In: *International Conference on Machine Learning*. PMLR, 2016, pp. 2034–2042.

[33] D. Hernandez, A. K. Moretti, Z. Wei, S. Saxena, J. Cunningham, and L. Paninski. "Nonlinear Evolution via Spatially-Dependent Linear Dynamics for Electrophysiology and Calcium Data". In: *arXiv preprint arXiv:1811.02459* (2020).

[34] S. Hochreiter. *Untersuchungen zu dynamischen neuronalen Netzen*. Diploma thesis. Institut für Informatik, Technische Universität München. 1991.

[35] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[36] K. Inoue, S. Ohara, Y. Kuniyoshi, and K. Nakajima. "Transient Chaos in BERT". In: *arXiv:2106.03181 [nlin]* (2021).

[37] L. Jing, C. Gulcehre, J. Peurifoy, Y. Shen, M. Tegmark, M. Soljacic, and Y. Bengio. "Gated Orthogonal Recurrent Units: On Learning to Forget". In: *Neural Computation* 31.4 (2019), pp. 765–783.

[38] A. Kag and V. Saligrama. "Time Adaptive Recurrent Neural Network". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021, pp. 15144–15153.

[39] A. Kag and V. Saligrama. "Training Recurrent Neural Networks via Forward Propagation Through Time". In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 5189–5200.

[40] A. Kag, Z. Zhang, and V. Saligrama. "RNNs Incrementally Evolving on an Equilibrium Manifold: A Panacea for Vanishing and Exploding Gradients?" In: *International Conference on Learning Representations*. 2020.

[41] L. Kamdjeu Kengne, J. R. Mboupda Pone, and H. B. Fotsin. "On the dynamics of chaotic circuits based on memristive diode-bridge with variable symmetry: A case study". In: *Chaos, Solitons & Fractals* 145 (2021), p. 110795.

[42] H. Kantz and T. Schreiber. *Nonlinear Time Series Analysis*. 2nd ed. Cambridge University Press, 2003.

[43] H. Kantz, T. Schreiber, I. Hoffmann, T. Buzug, G. Pfister, L. G. Flepp, J. Simonet, R. Badii, and E. Brun. "Nonlinear noise reduction: A case study on experimental data". In: *Physical Review E* 48 (1993), pp. 1529–1538.

[44] M. B. Kennel, R. Brown, and H. D. I. Abarbanel. "Determining embedding dimension for phase-space reconstruction using a geometrical construction". In: *Phys. Rev. A* 45 (1992).

[45] G. Kerg, K. Goyette, M. Puelma Touzel, G. Gidel, E. Vorontsov, Y. Bengio, and G. Lajoie. "Non-normal Recurrent Neural Network (nnRNN): learning long time dependencies while improving expressivity with transient dynamics". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[46] M. Kesmia, S. Boughaba, and S. Jacquir. "Control of continuous dynamical systems modeling physiological states". In: *Chaos, Solitons & Fractals* 136 (2020), p. 109805.

[47] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations*. Ed. by Y. Bengio and Y. LeCun. ICLR, 2015.

[48] D. P. Kingma and M. Welling. "Auto-Encoding Variational Bayes". In: *Proceedings of the 2nd International Conference on Learning Representations*. 2014.

[49] J. Z. Kolter and G. Manek. "Learning Stable Deep Dynamics Models". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[50] G. Koppe, H. Toutounji, P. Kirsch, S. Lis, and D. Durstewitz. "Identifying nonlinear dynamical systems via generative recurrent neural networks with applications to fMRI". In: *PLOS Computational Biology* 15.8 (2019), pp. 1–35.

[51] D. Kramer, P. L. Bommer, C. Tombolini, G. Koppe, and D. Durstewitz. "Reconstructing Nonlinear Dynamical Systems from Multi-Modal Time Series". In: *Proceedings the 39th International Conference on Machine Learning*. Vol. 162. PMLR, 2022.

[52] J. Laskar and P. Robutel. "The chaotic obliquity of the planets". In: *Nature* 361.6413 (1993), pp. 608–612. (Visited on 09/10/2021).

[53] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao. "Independently Recurrent Neural Network (IndRNN): Building A Longer and Deeper RNN". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5457–5466.

[54] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. "Fourier Neural Operator for Parametric Partial Differential Equations". In: *International Conference on Learning Representations*. 2021.

[55] Z. C. Lipton, J. Berkowitz, and C. Elkan. "A Critical Review of Recurrent Neural Networks for Sequence Learning". In: *arXiv:1506.00019 [cs]* (2015).

[56] E. N. Lorenz. "Deterministic Nonperiodic Flow". In: *Journal of the Atmospheric Sciences* 20.2 (1963), pp. 130–141.

[57] N. Maheswaranathan, A. Williams, M. Golub, S. Ganguli, and D. Sussillo. "Reverse engineering recurrent networks for sentiment classification reveals line attractor dynamics". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[58] S. Mangiarotti, M. Peyre, Y. Zhang, M. Huc, F. Roger, and Y. Kerr. "Chaos theory applied to the outbreak of COVID-19: an ancillary approach to decision making in pandemic context". In: *Epidemiology and Infection* 148 (2020), e95.

[59] R. M. May. "Chaos and the Dynamics of Biological Populations". In: *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 413.1844 (1987), pp. 27–44.

[60] J. D. Meiss. *Differential Dynamical Systems*. Society for Industrial and Applied Mathematics, 2007.

[61] J. Menick, E. Elsen, U. Evci, S. Osindero, K. Simonyan, and A. Graves. "Practical Real Time Recurrent Learning with a Sparse Approximation". In: *International Conference on Learning Representations*. 2021.

[62] H. Millán, B. Ghanbarian-Alavijeh, and I. García-Fornaris. "Nonlinear dynamics of mean daily temperature and dewpoint time series at Babolsar, Iran, 1961–2005". In: *Atmospheric Research*. Clouds, Aerosols and Radiation 98.1 (2010), pp. 89–101. DOI: 10.1016/j.atmosres.2010.06.001.

[63] Z. Monfared and D. Durstewitz. "Existence of n-cycles and border-collision bifurcations in piecewise-linear continuous maps with applications to recurrent neural networks". In: *Nonlinear Dynamics* 101.2 (2020), pp. 1037–1052.

[64] Z. Monfared and D. Durstewitz. "Transformation of ReLU-based recurrent neural networks from discrete-time to continuous-time". In: *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 2020, pp. 6999–7009.

[65] L. F. Olsen and H. Degn. "Chaos in an enzyme reaction". In: *Nature* 267.5607 (1977), pp. 177–178.

[66] E. Orhan and X. Pitkow. "Improved memory in recurrent neural networks with sequential non-normal dynamics". In: *International Conference on Learning Representations*. ICLR, 2020.

[67] R. Pascanu, T. Mikolov, and Y. Bengio. "On the Difficulty of Training Recurrent Neural Networks". In: *Proceedings of the 30th International Conference on International Conference on Machine Learning*. Vol. 28. ICML'13. JMLR, 2013, III–1310–III–1318.

[68] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. "Automatic differentiation in PyTorch". In: *31st Conference on Neural Information Processing Systems*. NIPS, 2017.

[69] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott. "Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach". In: *Phys. Rev. Lett.* 120 (2018), p. 024102. DOI: 10.1103/PhysRevLett.120.024102.

[70] B. Pearlmutter. *Dynamic recurrent neural networks*. Carnegie Mellon University. 1990.

[71] L. Perko. *Differential Equations and Dynamical Systems*. Vol. 7. Springer, New York, NY, 2001.

[72] M. T. Rosenstein, J. J. Collins, and C. J. De Luca. "A practical method for calculating largest Lyapunov exponents from small data sets". In: *Physica D: Nonlinear Phenomena* 65.1 (1993), pp. 117–134. DOI: 10.1016/0167-2789(93)90009-P.

[73] T. K. Rusch and S. Mishra. "Coupled Oscillatory Recurrent Neural Network (coRNN): An accurate and (gradient) stable architecture for learning long time dependencies". In: *International Conference on Learning Representations*. ICLR, 2021.

[74] T. K. Rusch and S. Mishra. "UnICORNN: A recurrent model for learning very long time dependencies". In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. PMLR, 2021, pp. 9168–9178.

[75] E. Russo and D. Durstewitz. "Cell assemblies at multiple time scales with arbitrary lag constellations". In: *eLife* 6 (2017). Ed. by M. Howard, e19428.

[76] O. E. Rössler. "An equation for continuous chaos". In: *Physics Letters A* 57.5 (1976), pp. 397–398.

[77] S. Sabanal and M. Nakagawa. "The fractal properties of vocal sounds and their application in the speech recognition model". In: *Chaos, Solitons & Fractals* 7.11 (1996), pp. 1825–1843.

[78] T. Sauer, J. A. Yorke, and M. Casdagli. "Embedology". In: *Journal of Statistical Physics* 65.3 (1991), pp. 579–616.

[79] G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw. "BCI2000: a general-purpose brain-computer interface (BCI) system". In: *IEEE transactions on bio-medical engineering* 51.6 (2004), pp. 1034–1043.

[80] D. Schmidt, G. Koppe, Z. Monfared, M. Beutelspacher, and D. Durstewitz. "Identifying nonlinear dynamical systems with multiple time scales and long-range dependencies". In: *International Conference on Learning Representations*. ICLR, 2021.

[81] B. Sivakumar. “Chaos theory in geophysics: past, present and future”. In: *Chaos, Solitons & Fractals*. Fractals in Geophysics 19.2 (2004), pp. 441–462.

[82] F. Takens. “Detecting strange attractors in turbulence”. In: *Dynamical Systems and Turbulence, Warwick 1980*. Springer Berlin Heidelberg, 1981, pp. 366–381.

[83] R. Tchitnga, B. A. Mezatio, T. Fonzin Fozin, R. Kengne, P. H. Louodop Fotso, and A. Fomethe. “A novel hyperchaotic three-component oscillator operating at high frequency”. In: *Chaos, Solitons & Fractals* 118 (2019), pp. 166–180.

[84] D. J. Thomson. “Time series analysis of Holocene climate data”. In: *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences* 330.1615 (1990), pp. 601–616.

[85] T. H. Trinh, A. M. Dai, M.-T. Luong, and Q. V. Le. “Learning Longer-term Dependencies in RNNs with Auxiliary Losses”. In: *arXiv:1803.00144v3* (2018).

[86] A. P. Trischler and G. M. D’Eleuterio. “Synthesis of recurrent neural networks for dynamical system simulation”. In: *Neural Networks* 80 (2016), pp. 67–78.

[87] P. Turchin and A. D. Taylor. “Complex Dynamics in ecological Time Series”. In: *Ecology* 73.1 (1992), pp. 289–305.

[88] E. Tziperman, H. Scher, S. E. Zebiak, and M. A. Cane. “Controlling Spatiotemporal Chaos in a Realistic El Niño Prediction Model”. In: *Physical Review Letters* 79 (1997), pp. 1034–1037.

[89] P. R. Vlachas, G. Arampatzis, C. Uhler, and P. Koumoutsakos. “Learning the Effective Dynamics of Complex Multiscale Systems”. In: *arXiv:2006.13431 [nlin, physics:physics]* (2020).

[90] P. R. Vlachas, W. Byeon, Z. Y. Wan, T. P. Sapsis, and P. Koumoutsakos. “Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 474.2213 (2018), p. 20170844.

[91] R. Vogt, M. Puelma Touzel, E. Shlizerman, and G. Lajoie. “On Lyapunov Exponents for RNNs: Understanding Information Propagation Using Dynamical Systems Tools”. In: *Frontiers in Applied Mathematics and Statistics* 8 (2022).

[92] H. U. Voss, J. Timmer, and J. Kurths. “Nonlinear dynamical system identification from uncertain and indirect measurements”. In: *International Journal of Bifurcation and Chaos* 14.06 (2004), pp. 1905–1933.

[93] C. van Vreeswijk and H. Sompolinsky. “Chaos in Neuronal Networks with Balanced Excitatory and Inhibitory Activity”. In: *Science* 274.5293 (1996), pp. 1724–1726.

[94] J. H. M. Wedderburn. *Lectures on Matrices*. New York: American mathematical society, New York : Dover Publications, 1964.

[95] R. J. Williams and D. Zipser. “A Learning Algorithm for Continually Running Fully Recurrent Neural Networks”. In: *Neural Computation* 1.2 (1989), pp. 270–280.

[96] S. N. Wood. “Statistical inference for noisy nonlinear ecological dynamic systems”. In: *Nature* 466.7310 (2010).

[97] H. Zhang, D. Liu, and Z. Wang. *Controlling Chaos*. Springer, London, 2009.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Sec. 5.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] We provide comprehensive proofs of all presented theorems in the Appendix.
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code will be made public after publication.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

A.1 Theorems: Preliminaries

A.1.1 Transforming non-autonomous into autonomous discrete-time DS

Following [97], and based on similar reasoning as for continuous time (ODE-based) DS [3, 71], let us consider the non-autonomous discrete-time DS

$$\boldsymbol{x}_{t+1} \;=\; F(\boldsymbol{x}_t, t), \qquad \boldsymbol{x} \in \mathbb{R}^n. \tag{18}$$

Defining $\boldsymbol{z}_t \;=\; (\boldsymbol{x}_t, t)^\mathsf{T}$ and $G(\boldsymbol{z}_t) \;=\; (F(\boldsymbol{x}_t, t), t+1)^\mathsf{T}$, system (18) can be rewritten as the autonomous system

$$\boldsymbol{z}_{t+1} = G(\boldsymbol{z}_t), \qquad \boldsymbol{z} \in \mathbb{R}^{n+1}. \tag{19}$$

Hence, in all our theoretical treatment we can confine our attention to systems of the form (19).

A.1.2 RNN derivatives

Considering the loss function $\mathcal{L} \;=\; \sum_{t=1}^{T} \mathcal{L}_t$ of an RNN $F_{\boldsymbol{\theta}} \in \mathcal{R}$ parameterized by $\boldsymbol{\theta}$, we have

$$\frac{\partial \mathcal{L}}{\partial \theta} \;=\; \sum_{t=1}^{T} \frac{\partial \mathcal{L}_t}{\partial \theta}, \tag{20}$$

where

$$\frac{\partial \mathcal{L}_t}{\partial \theta} \;=\; \frac{\partial \mathcal{L}_t}{\partial \boldsymbol{z}_t} \frac{\partial \boldsymbol{z}_t}{\partial \theta}. \tag{21}$$

The tangent vector $\frac{\partial \boldsymbol{z}_T}{\partial \theta}$ has the form

$$\frac{\partial \boldsymbol{z}_T}{\partial \theta} \;=\; \frac{\partial^+ \boldsymbol{z}_T}{\partial \theta} \;+\; \sum_{t=1}^{T-2} \left(\prod_{r=0}^{t-1} \boldsymbol{J}_{T-r} \right) \frac{\partial^+ \boldsymbol{z}_{T-t}}{\partial \theta}, \tag{22}$$

where ∂^+ denotes the immediate partial derivative. Since for an RNN $F_{\boldsymbol{\theta}} \in \mathcal{R}$ the activation function is element-wise, with θ the m-th element of a parameter vector $\boldsymbol{\theta}$ (or belonging to the m-th row of a parameter matrix $\boldsymbol{\theta}$), we have

$$\frac{\partial^+ \boldsymbol{z}_T}{\partial \theta} \;=\; \begin{pmatrix} 0 & \cdots & 0 & \frac{\partial^+ z_{m,T}}{\partial \theta} & 0 & \cdots & 0 \end{pmatrix}^\mathsf{T}. \tag{23}$$

For instance, let $\boldsymbol{\theta} \;=\; \boldsymbol{W}$ be a weight matrix, then

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}} \;=\; \begin{pmatrix} \frac{\partial \mathcal{L}}{\partial w_{11}} & \frac{\partial \mathcal{L}}{\partial w_{12}} & \cdots & \frac{\partial \mathcal{L}}{\partial w_{1M}} \\ \frac{\partial \mathcal{L}}{\partial w_{21}} & \frac{\partial \mathcal{L}}{\partial w_{22}} & \cdots & \frac{\partial \mathcal{L}}{\partial w_{2M}} \\ \vdots & & & \\ \frac{\partial \mathcal{L}}{\partial w_{M1}} & \frac{\partial \mathcal{L}}{\partial w_{M2}} & \cdots & \frac{\partial \mathcal{L}}{\partial w_{MM}} \end{pmatrix}. \tag{24}$$

In this case, for the standard RNN we have

$$\frac{\partial^+ \boldsymbol{z}_T}{\partial w_{mk}} \;=\; \begin{pmatrix} 0 & \cdots & 0 & z_{k,T-1}\, \xi_{mk}(\boldsymbol{z}_{T-1}) & 0 & \cdots & 0 \end{pmatrix}^\mathsf{T} \;=\; \mathbf{1}_{(m,k)}\, \xi_{mk}(\boldsymbol{z}_{T-1})\, \boldsymbol{z}_{T-1}, \tag{25}$$

where $\xi_{mk}(\boldsymbol{z}_{T-1}) \;=\; f'_{w_{m,k}}\big(\sum_{j=1}^{M} w_{mj}\, z_{j,T-1} \;+\; \sum_{j=1}^{M} b_{mj}\, s_{j,T} + h_m \big)$, and $f'_{w_{m,k}}$ stands for the derivative of f with respect to $w_{m,k}$.

Therefore, for standard RNNs, (22) becomes

$$\frac{\partial \boldsymbol{z}_T}{\partial w_{mk}} \;=\; \mathbf{1}_{(m,k)}\, \xi_{mk}(\boldsymbol{z}_{T-1})\, \boldsymbol{z}_{T-1} \;+\; \sum_{t=1}^{T-2} \left(\prod_{r=0}^{t-1} \boldsymbol{J}_{T-r} \right) \mathbf{1}_{(m,k)}\, \xi_{mk}(\boldsymbol{z}_{T-t-1})\, \boldsymbol{z}_{T-t-1}. \tag{26}$$

A.1.3 Piecewise-linear RNN (PLRNN)

The PLRNN has the generic form [50, 80]

$$\boldsymbol{z}_t \,=\, F(\boldsymbol{z}_{t-1}) \,=\, \boldsymbol{A}\,\boldsymbol{z}_{t-1} \,+\, \boldsymbol{W}\phi(\boldsymbol{z}_{t-1}) \,+\, \boldsymbol{C}\boldsymbol{s}_t \,+\, \boldsymbol{h} + \boldsymbol{\varepsilon}_t, \tag{27}$$

where $\phi(\boldsymbol{z}_{t-1}) = \max(\boldsymbol{z}_{t-1}, 0)$ is the element-wise rectified linear unit (ReLU) function, $\boldsymbol{z}_t \in \mathbb{R}^M$ is the neural state vector, $\boldsymbol{A} \in \mathbb{R}^{M\times M}$ is a diagonal matrix of auto-regression weights, $\boldsymbol{W} \in \mathbb{R}^{M\times M}$ is a matrix of connection weights, $\boldsymbol{h} \in \mathbb{R}^M$ is the bias vector, $\boldsymbol{s}_t \in \mathbb{R}^K$ the external input weighted by $\boldsymbol{C} \in \mathbb{R}^{M\times K}$, and $\boldsymbol{\varepsilon}_t \,\sim\, \mathcal{N}(0, \boldsymbol{\Sigma})$ a Gaussian noise term with diagonal covariance matrix $\boldsymbol{\Sigma}$.

Equation (27) can be rewritten as

$$\boldsymbol{z}_t \,=\, (\boldsymbol{A} + \boldsymbol{W}\boldsymbol{D}_{\Omega(t-1)})\boldsymbol{z}_{t-1} \,+\, \boldsymbol{C}\boldsymbol{s}_t \,+\, \boldsymbol{h} + \boldsymbol{\varepsilon}_t \,=:\, \boldsymbol{W}_{\Omega(t-1)}\,\boldsymbol{z}_{t-1} \,+\, \boldsymbol{C}\boldsymbol{s}_t \,+\, \boldsymbol{h} + \boldsymbol{\varepsilon}_t, \tag{28}$$

where $\boldsymbol{D}_{\Omega(t)} \coloneqq \mathrm{diag}(\boldsymbol{d}_{\Omega(t)})$ is a diagonal matrix with $\boldsymbol{d}_{\Omega(t)} \coloneqq (d_1, d_2, \cdots, d_M)$ an indicator vector such that $d_m(z_{m,t}) =: d_m = 1$ whenever $z_{m,t} > 0$, and zeros otherwise.

For the PLRNN (28) we have

$$\boldsymbol{J}_t \,=\, \frac{\partial \boldsymbol{z}_t}{\partial \boldsymbol{z}_{t-1}} \,=\, \boldsymbol{W}_{\Omega(t-1)}, \tag{29}$$

and $\left\|\boldsymbol{W}_{\Omega(t-1)}\right\| \,\leq\, \|\boldsymbol{A}\| \,+\, \|\boldsymbol{W}\|$.

Furthermore, the derivatives (22) for the PLRNN (28) are

$$\frac{\partial \boldsymbol{z}_T}{\partial w_{mk}} \,=\, \boldsymbol{1}_{(m,k)}\boldsymbol{D}_{\Omega(T-1)}\,\boldsymbol{z}_{T-1} + \sum_{j=2}^{T-1}\,\left(\prod_{i=1}^{j-1}\boldsymbol{W}_{\Omega(T-i)}\right)\boldsymbol{1}_{(m,k)}\boldsymbol{D}_{\Omega(T-j)}\,\boldsymbol{z}_{T-j}. \tag{30}$$

A.1.4 Long Short-Term Memory (LSTM)

The LSTM is defined by the equations

$$\begin{aligned}
\boldsymbol{i}_t \,&=\, \sigma\big(\boldsymbol{W}_{ii}\boldsymbol{s}_t \,+\, \boldsymbol{W}_{hi}\boldsymbol{h}_{t-1} \,+\, \boldsymbol{b}_i\big)\\
\boldsymbol{f}_t \,&=\, \sigma\big(\boldsymbol{W}_{if}\boldsymbol{s}_t \,+\, \boldsymbol{W}_{hf}\boldsymbol{h}_{t-1} \,+\, \boldsymbol{b}_f\big)\\
\boldsymbol{g}_t \,&=\, \tanh\big(\boldsymbol{W}_{ig}\boldsymbol{s}_t \,+\, \boldsymbol{W}_{hg}\boldsymbol{h}_{t-1} \,+\, \boldsymbol{b}_g\big)\\
\boldsymbol{o}_t \,&=\, \sigma\big(\boldsymbol{W}_{io}\boldsymbol{s}_t \,+\, \boldsymbol{W}_{ho}\boldsymbol{h}_{t-1} \,+\, \boldsymbol{b}_o\big)\\
\boldsymbol{c}_t \,&=\, \boldsymbol{f}_t \,\odot\, \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \,\odot\, \boldsymbol{g}_t\\
\boldsymbol{h}_t \,&=\, \boldsymbol{o}_t \,\odot\, \tanh\left(\boldsymbol{c}_t\right)
\end{aligned} \tag{31}$$

where $\{\boldsymbol{s}_t\}$ is the input sequence, $\boldsymbol{W}$ denotes weight matrices, $\boldsymbol{b}$ bias terms, $\boldsymbol{i}_t, \boldsymbol{f}_t, \boldsymbol{g}_t, \boldsymbol{o}_t$ demonstrate the input, forget, cell, and output gates, $\boldsymbol{h}_t$ and $\boldsymbol{c}_t$ are the hidden and cell states at time t respectively, σ is the sigmoid activation function, and $\odot$ represents the element-wise (Hadamard) product (see [29, 35, 90] for further information on LSTMs).

Defining $\boldsymbol{z}_t \,:=\, (\boldsymbol{h}_t, \boldsymbol{c}_t)^{\mathsf{T}}$, the LSTM (31) can be represented as the first-order recursive map

$$\boldsymbol{z}_t \,=\, F_{\boldsymbol{\theta}}(\boldsymbol{z}_{t-1}) \,=\, \begin{pmatrix} \boldsymbol{o}_t \,\odot\, \tanh\left(\boldsymbol{f}_t \,\odot\, \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \,\odot\, \boldsymbol{g}_t\right) \\ \boldsymbol{f}_t \,\odot\, \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \,\odot\, \boldsymbol{g}_t \end{pmatrix}. \tag{32}$$

The term $\frac{\partial \mathcal{L}_t}{\partial \theta}$ in (20) for some LSTM parameter θ can be written as

$$\frac{\partial \mathcal{L}_t}{\partial \theta} \,=\, \sum_{r=1}^{t}\frac{\partial \mathcal{L}_t}{\partial \boldsymbol{h}_t}\,\frac{\partial \boldsymbol{h}_t}{\partial \boldsymbol{z}_t}\,\frac{\partial \boldsymbol{z}_t}{\partial \boldsymbol{z}_r}\,\frac{\partial \boldsymbol{z}_r}{\partial \theta}. \tag{33}$$

A necessary condition for LSTMs to have a chaotic orbit is given by:

Proposition 1. *Let the LSTM given by (31) have a chaotic attractor Γ^* with $\mathcal{B}_{\Gamma^*}$ as its basin of attraction. Then for every $\boldsymbol{z}_1 = (\boldsymbol{h}_1, \boldsymbol{c}_1)^\mathsf{T} \in \mathcal{B}_{\Gamma^*}$*

$$\gamma := \lim_{T\to\infty} \sqrt[T]{\left\| \begin{pmatrix} \frac{\partial \boldsymbol{h}_T}{\partial \boldsymbol{h}_1} & \frac{\partial \boldsymbol{h}_T}{\partial \boldsymbol{c}_1} \\ \frac{\partial \boldsymbol{c}_T}{\partial \boldsymbol{h}_1} & \frac{\partial \boldsymbol{c}_T}{\partial \boldsymbol{c}_1} \end{pmatrix} \right\|} > 1. \tag{34}$$

Proof. The Jacobian matrix of (32) for $t > 1$ can be written in the block form

$$\frac{\partial \boldsymbol{z}_t}{\partial \boldsymbol{z}_{t-1}} = J_t = \begin{pmatrix} \frac{\partial \boldsymbol{h}_t}{\partial \boldsymbol{h}_{t-1}} & \frac{\partial \boldsymbol{h}_t}{\partial \boldsymbol{c}_{t-1}} \\ \frac{\partial \boldsymbol{c}_t}{\partial \boldsymbol{h}_{t-1}} & \frac{\partial \boldsymbol{c}_t}{\partial \boldsymbol{c}_{t-1}} \end{pmatrix}. \tag{35}$$

Further, due to the chain rule, we have

$$\begin{aligned} J_t\, J_{t-1} &= \begin{pmatrix} \frac{\partial \boldsymbol{h}_t}{\partial \boldsymbol{h}_{t-1}} \frac{\partial \boldsymbol{h}_{t-1}}{\partial \boldsymbol{h}_{t-2}} + \frac{\partial \boldsymbol{h}_t}{\partial \boldsymbol{c}_{t-1}} \frac{\partial \boldsymbol{c}_{t-1}}{\partial \boldsymbol{h}_{t-2}} & \frac{\partial \boldsymbol{h}_t}{\partial \boldsymbol{h}_{t-1}} \frac{\partial \boldsymbol{h}_{t-1}}{\partial \boldsymbol{c}_{t-2}} + \frac{\partial \boldsymbol{h}_t}{\partial \boldsymbol{c}_{t-1}} \frac{\partial \boldsymbol{c}_{t-1}}{\partial \boldsymbol{c}_{t-2}} \\ \frac{\partial \boldsymbol{c}_t}{\partial \boldsymbol{h}_{t-1}} \frac{\partial \boldsymbol{h}_{t-1}}{\partial \boldsymbol{h}_{t-2}} + \frac{\partial \boldsymbol{c}_t}{\partial \boldsymbol{c}_{t-1}} \frac{\partial \boldsymbol{c}_{t-1}}{\partial \boldsymbol{h}_{t-2}} & \frac{\partial \boldsymbol{c}_t}{\partial \boldsymbol{h}_{t-1}} \frac{\partial \boldsymbol{h}_{t-1}}{\partial \boldsymbol{c}_{t-2}} + \frac{\partial \boldsymbol{c}_t}{\partial \boldsymbol{c}_{t-1}} \frac{\partial \boldsymbol{c}_{t-1}}{\partial \boldsymbol{c}_{t-2}} \end{pmatrix} \\ &= \begin{pmatrix} \frac{\partial \boldsymbol{h}_t}{\partial \boldsymbol{h}_{t-2}} & \frac{\partial \boldsymbol{h}_t}{\partial \boldsymbol{c}_{t-2}} \\ \frac{\partial \boldsymbol{c}_t}{\partial \boldsymbol{h}_{t-2}} & \frac{\partial \boldsymbol{c}_t}{\partial \boldsymbol{c}_{t-2}} \end{pmatrix}, \end{aligned} \tag{36}$$

and by induction we obtain

$$\frac{\partial \boldsymbol{z}_t}{\partial \boldsymbol{z}_1} = J_t\, J_{t-1}\, J_{t-2} \cdots J_2 = \begin{pmatrix} \frac{\partial \boldsymbol{h}_t}{\partial \boldsymbol{h}_1} & \frac{\partial \boldsymbol{h}_t}{\partial \boldsymbol{c}_1} \\ \frac{\partial \boldsymbol{c}_t}{\partial \boldsymbol{h}_1} & \frac{\partial \boldsymbol{c}_t}{\partial \boldsymbol{c}_1} \end{pmatrix}. \tag{37}$$

Now assume that (32) has a chaotic orbit given by

$$\Gamma^* = \{\boldsymbol{z}_1^*, \boldsymbol{z}_2^*, \cdots, \boldsymbol{z}_T^*, \cdots\}. \tag{38}$$

According to (37), the largest Lyapunov exponent of Γ^* is given by

$$\lambda_{\Gamma^*} = \lim_{T\to\infty} \frac{1}{T} \ln \left\| J_T^*\, J_{T-1}^* \cdots J_2^* \right\| = \lim_{T\to\infty} \frac{1}{T} \ln \left\| \begin{pmatrix} \frac{\partial \boldsymbol{h}_T^*}{\partial \boldsymbol{h}_1^*} & \frac{\partial \boldsymbol{h}_T^*}{\partial \boldsymbol{c}_1^*} \\ \frac{\partial \boldsymbol{c}_T^*}{\partial \boldsymbol{h}_1^*} & \frac{\partial \boldsymbol{c}_T^*}{\partial \boldsymbol{c}_1^*}, \end{pmatrix} \right\|.$$

Since Γ^* is chaotic, so $\lambda_{\Gamma^*} > 0$, which gives

$$\lim_{T\to\infty} \sqrt[T]{\left\| \begin{pmatrix} \frac{\partial \boldsymbol{h}_T^*}{\partial \boldsymbol{h}_1^*} & \frac{\partial \boldsymbol{h}_T^*}{\partial \boldsymbol{c}_1^*} \\ \frac{\partial \boldsymbol{c}_T^*}{\partial \boldsymbol{h}_1^*} & \frac{\partial \boldsymbol{c}_T^*}{\partial \boldsymbol{c}_1^*} \end{pmatrix} \right\|} > 1. \tag{39}$$

Based on Oseledec's multiplicative ergodic Theorem, (39) holds for every $\boldsymbol{z}_1 \in \mathcal{B}_{\Gamma^*}$. This completes the proof. □

A.1.5 Gated Recurrent Unit (GRU)

A GRU network is defined by the equations

$$\begin{aligned} \boldsymbol{z}_t &= \sigma\big(\boldsymbol{W}_z\, \boldsymbol{s}_t + \boldsymbol{U}_z \boldsymbol{h}_{t-1} + \boldsymbol{b}_z\big) \\ \boldsymbol{r}_t &= \sigma\big(\boldsymbol{W}_r\, \boldsymbol{s}_t + \boldsymbol{U}_r \boldsymbol{h}_{t-1} + \boldsymbol{b}_r\big) \\ \boldsymbol{h}_t &= (1 - \boldsymbol{z}_t) \odot \tanh\big(\boldsymbol{W}_h\, \boldsymbol{s}_t + \boldsymbol{U}_h(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}) + \boldsymbol{b}_h\big) + \boldsymbol{z}_t \odot \boldsymbol{h}_{t-1}, \end{aligned} \tag{40}$$

where $\boldsymbol{r}_t$ represents the reset gate, $\boldsymbol{z}_t$ the update gate, $\boldsymbol{s}_t$ and $\boldsymbol{h}_t$ denote the inputs and the hidden state respectively, $\boldsymbol{W}_z, \boldsymbol{W}_r, \boldsymbol{W}_h \in \mathbb{R}^{M\times N}$ and $\boldsymbol{U}_z, \boldsymbol{U}_r, \boldsymbol{U}_h \in \mathbb{R}^{M\times M}$ are weight matrices, $\boldsymbol{b}_z, \boldsymbol{b}_r, \boldsymbol{b}_h \in \mathbb{R}^M$ are bias vectors, and σ is the element-wise logistic sigmoid function (for more details about GRUs see [10]).

A.1.6 Unitary evolution RNN (uRNN)

The uRNN, proposed in [4], is defined as the nonlinear DS

$$\boldsymbol{z}_t \,=\, \sigma_{\boldsymbol{b}}\big(\boldsymbol{W}\boldsymbol{z}_{t-1} \,+\, \boldsymbol{V}\boldsymbol{s}_t\big), \tag{41}$$

for which $\boldsymbol{W} \in U(M)$ is an unitary matrix, $\boldsymbol{V} \in \mathbb{C}^{M\times N}$, $\boldsymbol{b} \in \mathbb{R}^M$ is the bias parameter, $\boldsymbol{s}_t$ is the real- or complex-valued input of dimension N, and

$$[\sigma_{\boldsymbol{b}}(\boldsymbol{z})]_i \,=\, [\sigma_{\text{modReLU}}(\boldsymbol{z})]_i \,=\, \begin{cases} \big(|z_i| + b_i\big)\frac{z_i}{|z_i|} & \text{if } |z_i| + b_i \,\geq\, 0 \\ 0 & \text{if } |z_i| + b_i \,<\, 0 \end{cases} . \tag{42}$$

Proposition 2. *The uRNN given by (41) cannot have any chaotic orbit.*

Proof. For any arbitrary orbit $\mathcal{O}_{\boldsymbol{z}_1}$ of (41) we have

$$\|J_T\, J_{T-1}\, \cdots\, J_2\| \,=\, \left\|\prod_{k=0}^{T-2} \boldsymbol{D}_{T-k}\, \boldsymbol{W}^{\mathsf{T}}\right\|, \tag{43}$$

where $\boldsymbol{D}_t \,=\, diag\Big(\sigma'_{\boldsymbol{b}}\big(\boldsymbol{W}\boldsymbol{z}_{t-1} \,+\, \boldsymbol{V}\boldsymbol{s}_t\big)\Big)$. Since $\boldsymbol{W}$ is unitary and so a norm preserving matrix, it is concluded that

$$\left\|\prod_{k=0}^{T-2} \boldsymbol{D}_{T-k}\, \boldsymbol{W}^{\mathsf{T}}\right\| \,\leq\, \prod_{k=0}^{T-2} \big\|\boldsymbol{D}_{T-k}\, \boldsymbol{W}^{\mathsf{T}}\big\| \,=\, \prod_{k=0}^{T-2} \|\boldsymbol{D}_{T-k}\| \,=\, 1, \tag{44}$$

which implies

$$\lambda_{max} \,=\, \lim_{T\to\infty} \frac{1}{T}\, \ln \|J_T\, J_{T-1}\, \cdots\, J_2\| \,\leq\, 0. \tag{45}$$

This rules out the existence of chaos (since $\lambda_{max} \,>\, 0$ is a necessary condition for $\mathcal{O}_{\boldsymbol{z}_1}$ to be chaotic). □

Note that, more generally, any RNN which is constrained such as to exhibit global convergence to a fixed point or cycle, by definition must have a maximum Lyapunov exponent $\lambda_{max} \leq 0$ (in accordance with Theorem 1), hence cannot exhibit chaotic behavior by definition.

A.2 Theorems: Proofs

A.2.1 Proof of theorem 1, parts (ii) & (iii)

Proof. (ii) If $\boldsymbol{J}$ is the Jordan normal form of $\prod_{s=0}^{k-1} J_{t^{*k}-s}$, then $\prod_{s=0}^{k-1} J_{t^{*k}-s} \,=\, \boldsymbol{P}\,\boldsymbol{J}\,\boldsymbol{P}^{-1}$, where

$$\boldsymbol{J} \,=\, \begin{pmatrix} \boldsymbol{J}_{m_1}(\lambda_1) & 0 & 0 & \cdots & 0 \\ 0 & \boldsymbol{J}_{m_2}(\lambda_2) & 0 & \cdots & 0 \\ \vdots & \dots & \ddots & \dots & \vdots \\ 0 & \dots & 0 & \boldsymbol{J}_{m_{p-1}}(\lambda_{p-1}) & 0 \\ 0 & \dots & \dots & 0 & \boldsymbol{J}_{m_p}(\lambda_p) \end{pmatrix}, \tag{46}$$

and m_i is the algebraic multiplicity of each eigenvalue λ_i. Since $\rho\big(\prod_{s=0}^{k-1} J_{t^{*k}-s}\big) \,<\, 1$, so the eigenvalue λ_i associated with each Jordan block satisfies $|\lambda_i| < 1\,(i = 1, \cdots, p)$. Moreover, every $m_i \times m_i$ Jordan block has the form

$$\boldsymbol{J}_{m_i}(\lambda_i) \,=\, \begin{pmatrix} \lambda_i & 1 & 0 & \cdots & 0 \\ 0 & \lambda_i & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_i & 1 \\ 0 & 0 & \dots & 0 & \lambda_i \end{pmatrix}. \tag{47}$$

Accordingly

$$\left\| \left(\prod_{s=0}^{k-1} J_{t^{*k}-s} \right)^j \right\| \;=\; \left\| \boldsymbol{P}\, \boldsymbol{J}^j\, \boldsymbol{P}^{-1} \right\| \;\leq\; p\, \left\| \boldsymbol{J}^j \right\|, \tag{48}$$

in which $p = \|\boldsymbol{P}\| \left\|\boldsymbol{P}^{-1}\right\|$. Furthermore, for $j \in \mathbb{N}$, $\boldsymbol{J}^j$ is a block diagonal matrix of the form

$$\boldsymbol{J}^j \;=\; \begin{pmatrix} \boldsymbol{J}^j_{m_1}(\lambda_1) & 0 & 0 & \cdots & 0 \\ 0 & \boldsymbol{J}^j_{m_2}(\lambda_2) & 0 & \cdots & 0 \\ \vdots & \dots & \ddots & \dots & \vdots \\ 0 & \dots & 0 & \boldsymbol{J}^j_{m_{p-1}}(\lambda_{p-1}) & 0 \\ 0 & \dots & \dots & 0 & \boldsymbol{J}^j_{m_p}(\lambda_p) \end{pmatrix}, \tag{49}$$

in which every $m_i \times m_i$ Jordan block has the form

$$\boldsymbol{J}^j_{m_i}(\lambda_i) \;=\; \begin{pmatrix} \lambda_i^j & \binom{j}{1}\lambda_i^{j-1} & \binom{j}{2}\lambda_i^{j-2} & \cdots & \binom{j}{m_i-1}\lambda_i^{j-m_i+1} \\ 0 & \lambda_i^j & \binom{j}{1}\lambda_i^{j-1} & \cdots & \binom{j}{m_i-2}\lambda_i^{j-m_i+2} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_i^j & \binom{j}{1}\lambda_i^{j-1} \\ 0 & 0 & \dots & 0 & \lambda_i^j \end{pmatrix}. \tag{50}$$

In addition, for every block $\boldsymbol{J}^j_{m_i}(\lambda_i)$, we have

$$\begin{aligned} \left\|\boldsymbol{J}^j_{m_i}(\lambda_i)\right\| \;&\leq\; \sqrt{m_i}\, \left\|\boldsymbol{J}^j_{m_i}(\lambda_i)\right\|_\infty \;=\; \sqrt{m_i} \sum_{q=1}^{m_i} \left| \left(\boldsymbol{J}^j_{m_i}(\lambda_i)\right)_{1q} \right| \\ &=\; \sqrt{m_i} \sum_{q=1}^{m_i} \binom{j}{q-1} |\lambda_i|^{j-q+1} \;=\; |\lambda_i|^j \sqrt{m_i} \left(|\lambda_i|^{1-m_i} \sum_{q=1}^{m_i} \binom{j}{q-1} |\lambda_i|^{m_i-q} \right) \\ &\leq\; |\lambda_i|^j\, j^{\,m_i}\, \sqrt{m_i} \left(|\lambda_i|^{1-m_i} \sum_{q=1}^{m_i} |\lambda_i|^{m_i-q} \right) \;=:\; |\lambda_i|^j\, j^{\,m_i}\, N_{\lambda_i}. \end{aligned} \tag{51}$$

Moreover, for any $1 < \tilde{r}_i < \frac{1}{|\lambda_i|}$, there exists some l_i such that $j^{\,m_i} < \tilde{r}_i^{\,j}$ for $j \geq l_i$. This means for $j \geq l_i$

$$\left\|\boldsymbol{J}^j_{m_i}(\lambda_i)\right\| \;\leq\; N_{\lambda_i}\, |\tilde{r}_i\, \lambda_i|^j, \tag{52}$$

such that $|\tilde{r}_i\, \lambda_i| = \tilde{r}_i |\lambda_i| < 1$.

Besides, for $\boldsymbol{J}^j = \boldsymbol{J}^j_{m_1}(\lambda_1) \oplus \boldsymbol{J}^j_{m_2}(\lambda_2) \oplus \cdots \oplus \boldsymbol{J}^j_{m_p}(\lambda_p)$

$$\left\|\boldsymbol{J}^j\right\| \;=\; \max_{1\leq i\leq p} \left\|\boldsymbol{J}^j_{m_i}(\lambda_i)\right\| \;=:\; \left\|\boldsymbol{J}^j_{m}(\lambda)\right\|. \tag{53}$$

Hence, from (48), (52) and (53), it is deduced that for $j \geq l$

$$\left\| \left(\prod_{s=0}^{k-1} J_{t^{*k}-s} \right)^j \right\| \;\leq\; p\, N_\lambda\, |\tilde{r}\, \lambda|^j \;=:\; \bar{p}\, r^j, \tag{54}$$

in which $r = |\tilde{r}\, \lambda| < 1$.

Furthermore, let for Γ_k

$$\max_{T\geq 1}\Big\{\,\|\boldsymbol{J}_T^*\|\,\Big\} \;=\; \max_{0\leq s\leq k-1}\Big\{\,\|J_{t^{*k}-s}\|\,\Big\} \;=\; \bar{m},$$

$$\max_{T\geq 1}\Big\{\,\Big\|\frac{\partial^+\boldsymbol{z}_T}{\partial\theta}\Big\|\,\Big\} \;=\; \max_{0\leq s\leq k-1}\Big\{\,\Big\|\frac{\partial^+\boldsymbol{z}_{t^{*k}-s}}{\partial\theta}\Big\|\,\Big\} \;=\; \xi,$$

$$\max_{T\geq 1}\Big\{\,\|\boldsymbol{z}_T\|\,\Big\} \;=\; \max_{0\leq s\leq k-1}\Big\{\,\|\boldsymbol{z}_{t^{*k}-s}\|\,\Big\} \;=\; \bar{q}. \tag{55}$$

Hence, defining $\boldsymbol{z}_0 \;=\; 0$, for this k-cycle

$$\begin{aligned}\Big\|\frac{\partial\boldsymbol{z}_T}{\partial\theta}\Big\| &= \Big\|\frac{\partial^+\boldsymbol{z}_T}{\partial\theta} \;+\; \sum_{t=1}^{T-2}\Big(\prod_{r=0}^{t-1}\boldsymbol{J}_{T-r}^*\Big)\frac{\partial^+\boldsymbol{z}_{T-t}}{\partial\theta}\Big\| \\ &= \Big\|\frac{\partial^+\boldsymbol{z}_T}{\partial\theta} \;+\; \sum_{t=1}^{T-1}\Big(\prod_{r=0}^{t-1}\boldsymbol{J}_{T-r}^*\Big)\frac{\partial^+\boldsymbol{z}_{T-t}}{\partial\theta}\Big\| \\ &\leq\; \bar{q}\,\xi\Big(1 \;+\; \sum_{t=1}^{T-1}\Big\|\prod_{r=0}^{t-1}\boldsymbol{J}_{T-r}^*\Big\|\Big).\end{aligned} \tag{56}$$

On the other hand, for $T \;=\; kj$, from (54) and (55) we have

$$\begin{aligned}\sum_{t=1}^{T-1}\Big\|\prod_{r=0}^{t-1}\boldsymbol{J}_{T-r}^*\Big\| &= \sum_{t=1}^{kj-1}\Big\|\prod_{r=0}^{t-1}\boldsymbol{J}_{kj-r}^*\Big\| = \sum_{t=1}^{k-1}\Big\|\prod_{r=0}^{t-1}\boldsymbol{J}_{kj-r}^*\Big\| + \sum_{t=k}^{2k-1}\Big\|\prod_{r=0}^{t-1}\boldsymbol{J}_{kj-r}^*\Big\| \\ &\quad + \sum_{t=2k}^{3k-1}\Big\|\prod_{r=0}^{t-1}\boldsymbol{J}_{kj-r}^*\Big\| + \cdots + \sum_{t=(j-2)k}^{(j-1)k-1}\Big\|\prod_{r=0}^{t-1}\boldsymbol{J}_{kj-r}^*\Big\| + \sum_{t=(j-1)k}^{kj-1}\Big\|\prod_{r=0}^{t-1}\boldsymbol{J}_{kj-r}^*\Big\| \\ &= \sum_{t=1}^{k-1}\Big\|\prod_{r=0}^{t-1}\boldsymbol{J}_{kj-r}^*\Big\| + \sum_{i=2}^{j}\sum_{t=(i-1)k}^{ik-1}\Big\|\prod_{r=0}^{t-1}\boldsymbol{J}_{kj-r}^*\Big\| \\ &\leq\; \big(\bar{m}+\bar{m}^2+\cdots+\bar{m}^{k-1}\big) \;+\; \sum_{i=2}^{j}\bar{p}\,\big(1+\bar{m}+\bar{m}^2+\cdots+\bar{m}^{k-1}\big)\,r^{i-1}.\end{aligned} \tag{57}$$

Thus, considering $\big(\bar{m}+\bar{m}^2+\cdots+\bar{m}^{k-1}\big) \;=\; \mathcal{M}$, it is deduced that

$$\lim_{T\to\infty}\Big\|\frac{\partial\boldsymbol{z}_T}{\partial\theta}\Big\| \;=\; \lim_{j\to\infty}\Big\|\frac{\partial\boldsymbol{z}_{kj}}{\partial\theta}\Big\| \;\leq\; \bar{q}\,\xi\big(1 \;+\; \mathcal{M} + \frac{\bar{p}\,r(1+\mathcal{M})}{1-r}\big) \;=\; \bar{\mathcal{M}} \;<\; \infty, \tag{58}$$

which, by (21), implies $\frac{\partial\mathcal{L}_T}{\partial\theta}$ will be bounded for $T\to\infty$.

(iii) Consider the PLRNN given by (27), where for simplicity we ignore the external inputs and noise terms. Let $\{\boldsymbol{z}_{t_1},\boldsymbol{z}_{t_2},\boldsymbol{z}_{t_3},\ldots\}$ be an orbit which converges to Γ_k. Hence

$$\lim_{n\to\infty} d(\boldsymbol{z}_{t_n},\Gamma_k) \;=\; 0, \tag{59}$$

which implies there exists a neighborhood U of Γ_k and k sub-sequences $\{\boldsymbol{z}_{t_{km}}\}_{m=1}^{\infty}$, $\{\boldsymbol{z}_{t_{km+1}}\}_{m=1}^{\infty}$, $\cdots$, $\{\boldsymbol{z}_{t_{km+(k-1)}}\}_{m=1}^{\infty}$ of the sequence $\{\boldsymbol{z}_{t_n}\}_{n=1}^{\infty}$ such that all these sub-sequences belong to U and

a) $\boldsymbol{z}_{t_{km+s}} \;=\; F^k(\boldsymbol{z}_{t_{k(m-1)+s}}), s=0,1,2,\cdots,k-1,$

b) $\lim_{m\to\infty} \boldsymbol{z}_{t_{km+s}} = \boldsymbol{z}_{t^{*k}-s}, s = 0, 1, 2, \cdots, k-1,$

c) for every $\boldsymbol{z}_{t_n} \in U$ there is some $s \in \{0, 1, 2, \cdots, k-1\}$ such that $\boldsymbol{z}_{t_n} \in \{\boldsymbol{z}_{t_{km+s}}\}_{m=1}^{\infty}$.

In this case, for every $\boldsymbol{z}_{t_n} \in U$ with $\boldsymbol{z}_{t_n} \in \{\boldsymbol{z}_{t_{km+s}}\}_{m=1}^{\infty}$, there exists some $\tilde{n} \in \mathbb{N}$ such that $\boldsymbol{z}_{t_n} = \boldsymbol{z}_{t_{k\tilde{n}+s}}$ and $\lim_{\tilde{n}\to\infty} \boldsymbol{z}_{t_{k\tilde{n}+s}} = \boldsymbol{z}_{t^{*k}-s}$. Therefore, continuity of F results in

$$\lim_{\tilde{n}\to\infty} F(\boldsymbol{z}_{t_{k\tilde{n}+s}}) = F(\boldsymbol{z}_{t^{*k}-s}), \tag{60}$$

and so by (28)

$$\lim_{\tilde{n}\to\infty} \left(\boldsymbol{W}_{\Omega(t_{k\tilde{n}+s})}\, \boldsymbol{z}_{t_{k\tilde{n}+s}} + \boldsymbol{h}\right) = \boldsymbol{W}_{\Omega(t^{*k}-s)}\, \boldsymbol{z}_{t^{*k}-s} + \boldsymbol{h}, \tag{61}$$

which implies

$$\lim_{\tilde{n}\to\infty} \boldsymbol{W}_{\Omega(t_{k\tilde{n}+s})}\, \boldsymbol{z}_{t_{k\tilde{n}+s}} = \boldsymbol{W}_{\Omega(t^{*k}-s)}\, \boldsymbol{z}_{t^{*k}-s}. \tag{62}$$

Assuming $\lim_{\tilde{n}\to\infty} \boldsymbol{W}_{\Omega(t_{k\tilde{n}+s})} = \boldsymbol{L}$, since (62) holds for every $\boldsymbol{z}_{t^{*k}-s}$, substituting $\boldsymbol{z}_{t^{*k}-s} = \boldsymbol{e}_1^{\mathsf{T}} = (1, 0, \cdots, 0)^T$ in (62), we can prove that the first column of $\boldsymbol{L}$ equals the first column of $\boldsymbol{W}_{\Omega(t^{*k}-s)}$. Performing the same procedure for $\boldsymbol{z}_{t^{*k}-s} = \boldsymbol{e}_i^{\mathsf{T}}$, $i = 2, 3, \cdots, M$, yields

$$\lim_{\tilde{n}\to\infty} \boldsymbol{W}_{\Omega(t_{k\tilde{n}+s})} = \boldsymbol{W}_{\Omega(t^{*k}-s)}. \tag{63}$$

According to (59), U contains an infinite number of terms of the sequence $\{\boldsymbol{z}_{t_n}\}_{n=1}^{\infty}$, i.e.

$$\exists N \in \mathbb{N} \;\; s.t. \quad n \geq N \implies \boldsymbol{z}_{t_n} \in U. \tag{64}$$

Suppose that $\boldsymbol{z}_{t_n} \in U$ for some $n \geq N$. Thus, there exists some $s \in \{0, 1, 2, \cdots, k-1\}$ such that $\boldsymbol{z}_{t_n} \in \{\boldsymbol{z}_{t_{km+s}}\}_{m=1}^{\infty}$. Without loss of generality let $s = 0$. Hence, there is some $\tilde{n} \in \mathbb{N}$ such that $\boldsymbol{z}_{t_n} = \boldsymbol{z}_{t_{k\tilde{n}}}$ and $\lim_{\tilde{n}\to\infty} \boldsymbol{z}_{t_{k\tilde{n}}} = \boldsymbol{z}_{t^{*k}}$. In this case, moving forward in time gives

$$\begin{aligned}
&\boldsymbol{z}_{t_n} = \boldsymbol{z}_{t_{k\tilde{n}}} \;\; \left(\boldsymbol{z}_{t_n} \in \{\boldsymbol{z}_{t_{km}}\}_{m=1}^{\infty}\right), && \lim_{\tilde{n}\to\infty} \boldsymbol{z}_{t_{k\tilde{n}}} = \boldsymbol{z}_{t^{*k}},\\
&\boldsymbol{z}_{t_{n+1}} = \boldsymbol{z}_{t_{k\tilde{n}+1}} \;\; \left(\boldsymbol{z}_{t_{n+1}} \in \{\boldsymbol{z}_{t_{km+1}}\}_{m=1}^{\infty}\right), && \lim_{\tilde{n}\to\infty} \boldsymbol{z}_{t_{k\tilde{n}+1}} = \boldsymbol{z}_{t^{*k}-1},\\
&\boldsymbol{z}_{t_{n+2}} = \boldsymbol{z}_{t_{k\tilde{n}+2}} \;\; \left(\boldsymbol{z}_{t_{n+2}} \in \{\boldsymbol{z}_{t_{km+2}}\}_{m=1}^{\infty}\right), && \lim_{\tilde{n}\to\infty} \boldsymbol{z}_{t_{k\tilde{n}+2}} = \boldsymbol{z}_{t^{*k}-2},\\
&\vdots\\
&\boldsymbol{z}_{t_{n+k-1}} = \boldsymbol{z}_{t_{k\tilde{n}+k-1}} \;\; \left(\boldsymbol{z}_{t_{n+(k-1)}} \in \{\boldsymbol{z}_{t_{km+k-1}}\}_{m=1}^{\infty}\right), && \lim_{\tilde{n}\to\infty} \boldsymbol{z}_{t_{k\tilde{n}+k-1}} = \boldsymbol{z}_{t^{*k}-(k-1)},\\
&\boldsymbol{z}_{t_{n+k}} = \boldsymbol{z}_{t_{k(\tilde{n}+1)}} \;\; \left(\boldsymbol{z}_{t_{n+k}} \in \{\boldsymbol{z}_{t_{km}}\}_{m=1}^{\infty}\right), && \lim_{\tilde{n}\to\infty} \boldsymbol{z}_{t_{k(\tilde{n}+1)}} = \boldsymbol{z}_{t^{*k}},\\
&\boldsymbol{z}_{t_{n+k+1}} = \boldsymbol{z}_{t_{k(\tilde{n}+1)+1}} \;\; \left(\boldsymbol{z}_{t_{n+k+1}} \in \{\boldsymbol{z}_{t_{km+1}}\}_{m=1}^{\infty}\right), && \lim_{\tilde{n}\to\infty} \boldsymbol{z}_{t_{k(\tilde{n}+1)+1}} = \boldsymbol{z}_{t^{*k}-1},\\
&\vdots\\
&\boldsymbol{z}_{t_{n+2k-1}} = \boldsymbol{z}_{t_{k(\tilde{n}+1)+k-1}} \;\; \left(\boldsymbol{z}_{t_{n+2k-1}} \in \{\boldsymbol{z}_{t_{km+k-1}}\}_{m=1}^{\infty}\right), && \lim_{\tilde{n}\to\infty} \boldsymbol{z}_{t_{k(\tilde{n}+1)+k-1}} = \boldsymbol{z}_{t^{*k}-(k-1)},\\
&\boldsymbol{z}_{t_{n+2k}} = \boldsymbol{z}_{t_{k(\tilde{n}+2)}} \;\; \left(\boldsymbol{z}_{t_{n+2k}} \in \{\boldsymbol{z}_{t_{km}}\}_{m=1}^{\infty}\right), && \lim_{\tilde{n}\to\infty} \boldsymbol{z}_{t_{k(\tilde{n}+2)}} = \boldsymbol{z}_{t^{*k}},\\
&\vdots
\end{aligned} \tag{65}$$

Consequently, for $n \geq N$ and $j \in \mathbb{N}$, we can write

$$\begin{aligned}
&\prod_{i=0}^{kj-1} \boldsymbol{W}_{\Omega(t_{n+kj-1-i})} \\
&= \Big(\prod_{i=1}^{k} \boldsymbol{W}_{\Omega(t_{k(\tilde{n}+j)+k-i})}\Big)\Big(\prod_{i=1}^{k} \boldsymbol{W}_{\Omega(t_{k(\tilde{n}+j-1)+k-i})}\Big)\cdots\Big(\prod_{i=1}^{k} \boldsymbol{W}_{\Omega(t_{k(\tilde{n})+k-i})}\Big) \\
&= \prod_{l=0}^{j}\prod_{i=1}^{k} \boldsymbol{W}_{\Omega(t_{k(\tilde{n}+j-l)+k-i})}. \qquad (66)
\end{aligned}$$

On the other hand, in equation (28), there are different configurations for matrix $\boldsymbol{D}_{\Omega(t-1)}$ and hence different forms for matrix $\boldsymbol{W}_{\Omega(t_{k\tilde{n}+s})}$. In this case, the phase space of the system is divided into different sub-regions by some borders; see [63, 64] for more details. Also, since the system (28) is a linear map in each sub-region, the k periodic points of Γ_k must belong to different sub-regions (at least two different sub-regions). Accordingly, based on (63) and (65), there exists some $\tilde{N} \in \mathbb{N}$ such that for every $\tilde{n} \geq \tilde{N}$ both $\boldsymbol{z}_{t_{k\tilde{n}+s}}$ and $\boldsymbol{z}_{t^{*k}-s}$ belong to the same sub-region and so the matrices $\boldsymbol{W}_{\Omega(t_{k\tilde{n}+s})}$ and $\boldsymbol{W}_{\Omega(t^{*k}-s)}$ $(s \in \{0, 1, 2, \cdots, k-1\})$ are identical. Hence, for $n \geq N$, $\tilde{n} \geq \tilde{N}$ and $j \in \mathbb{N}$, equation (66) becomes

$$\prod_{i=0}^{kj-1} \boldsymbol{W}_{\Omega(t_{n+kj-1-i})} = \prod_{l=0}^{j}\prod_{i=1}^{k} \boldsymbol{W}_{\Omega(t_{k(\tilde{n}+j-l)+k-i})} = \Big(\prod_{s=0}^{k-1} \boldsymbol{W}_{\Omega(t^{*k}-s)}\Big)^{j}. \qquad (67)$$

Therefore, similar to the part (ii), we can prove for every $\boldsymbol{z}_1 \in \mathcal{B}_{\Gamma_k}$, $\frac{\partial \boldsymbol{z}_T}{\partial \theta}$ and $\frac{\partial \mathcal{L}_T}{\partial \theta}$ will also remain bounded. □

A.2.2 Proof of theorem 2, part (ii)

Proof. (ii) Let for every $T > 2$

$$\boldsymbol{L}_T := J_T^* \, J_{T-1}^* \cdots J_2^*. \qquad (68)$$

$\{\boldsymbol{L}_T\}_{T\in\mathbb{N},\, T>2}$ is a sequence of matrices $\boldsymbol{L}_T = \big[l_{ij}^{(T)}\big]_{1\leq i,j\leq M}$ and, due to (13), $\lim_{T\to\infty} \|\boldsymbol{L}_T\| = \infty$. Hence, there is at least one sub-sequence $\{l_{mk}^{(T_n)}\}_{T_n\in\mathbb{N},\, T_n>2}$ (for some $m, k \in \{1, 2, \cdots, M\}$) such that $\lim_{T_n\to\infty} l_{mk}^{(T_n)} = \infty$.

On the other hand

$$\frac{\partial \boldsymbol{z}_T^*}{\partial \theta} = \frac{\partial^+ \boldsymbol{z}_T^*}{\partial \theta} + \sum_{t=1}^{T-2}\Big(\prod_{r=0}^{t-1} \boldsymbol{J}_{T-r}^*\Big)\frac{\partial^+ \boldsymbol{z}_{T-t}^*}{\partial \theta}. \qquad (69)$$

Moreover, there exists some $N > 2$ such that (for $t = T - N + 1$)

$$\frac{\partial^+ \boldsymbol{z}_{N-1}^*}{\partial \theta} \neq 0. \qquad (70)$$

For θ as the k-th element of a parameter vector $\boldsymbol{\theta}$ (or belonging to the k-th row of a parameter matrix $\boldsymbol{\theta}$), the term

$$\Big(\prod_{r=0}^{T-N} \boldsymbol{J}_{T-r}^*\Big)\frac{\partial^+ \boldsymbol{z}_{N-1}^*}{\partial \theta} \qquad (71)$$

is a vector in which the i-th element is $l_{ik}^{(T)} \, \frac{\partial^+ \boldsymbol{z}_{k,N-1}^*}{\partial \theta}$.

Since $\lim_{T_n\to\infty} l_{mk}^{(T_n)} = \infty$, due to (70) $\lim_{T_n\to\infty} l_{mk}^{(T_n)} \, \frac{\partial^+ \boldsymbol{z}_{k,N-1}^*}{\partial \theta} = \infty$, which implies $\frac{\partial \boldsymbol{z}_T^*}{\partial \theta}$ will diverge as $T \to \infty$. Similarly, by (21), we can prove $\frac{\partial \mathcal{L}_T^*}{\partial \theta}$ is divergent for $T \to \infty$.

By Oseledec's multiplicative ergodic Theorem, the results also hold for every $\boldsymbol{z}_1 \in \mathcal{B}_{\Gamma^*}$. □

A.2.3 Proof of theorem 3

Proof. Let $\Gamma = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \dots \boldsymbol{z}_T, \cdots\}$ be a quasi-periodic attractor. Then, the largest Lyapunov exponent of Γ is

$$\lambda = \lim_{T\to\infty} \frac{1}{T} \ln \|J_T\, J_{T-1} \cdots J_2\| = \lim_{T\to\infty} \frac{1}{T} \ln \left\| \frac{\partial \boldsymbol{z}_T}{\partial \boldsymbol{z}_1} \right\| = 0. \tag{72}$$

We prove for every $0 < \epsilon < 1$

$$\lim_{T\to\infty} (1-\epsilon)^{T-1} < \lim_{T\to\infty} \left\| \frac{\partial \boldsymbol{z}_T}{\partial \boldsymbol{z}_1} \right\| < \lim_{T\to\infty} (1+\epsilon)^{T-1}. \tag{73}$$

For this purpose, we show $\forall\, 0 < \epsilon < 1$

(I) $\lim_{T\to\infty}(1-\epsilon)^{T-1} < \lim_{T\to\infty} \left\| \frac{\partial \boldsymbol{z}_T}{\partial \boldsymbol{z}_1} \right\|$, and

(II) $\lim_{T\to\infty} \left\| \frac{\partial \boldsymbol{z}_T}{\partial \boldsymbol{z}_1} \right\| < \lim_{T\to\infty}(1+\epsilon)^{T-1}$.

Assume for the sake of contradiction that (I) does not hold. Then there exists some $0 < \epsilon < 1$ such that

$$\lim_{T\to\infty} (1-\epsilon)^{T-1} \geq \lim_{T\to\infty} \left\| \frac{\partial \boldsymbol{z}_T}{\partial \boldsymbol{z}_1} \right\|. \tag{74}$$

Therefore

$$\exists\, T_0 > 1 \;\; s.t. \;\; \forall\, T \geq T_0 \implies (1-\epsilon)^{T-1} \geq \left\| \frac{\partial \boldsymbol{z}_T}{\partial \boldsymbol{z}_1} \right\|, \tag{75}$$

and so

$$\exists\, T_0 > 1 \;\; s.t. \;\; \forall\, T \geq T_0 \implies \frac{\ln(1-\epsilon)^{T-1}}{T-1} \geq \frac{\ln \left\| \frac{\partial \boldsymbol{z}_T}{\partial \boldsymbol{z}_1} \right\|}{T-1}. \tag{76}$$

Consequently, due to (72), for $T \to \infty$ we have $\ln(1-\epsilon) \geq 0$. This implies $\epsilon \leq 0$, which is a contradiction.

Similarly if we assume (II) is not true, then there exists some $0 < \epsilon < 1$ such that

$$\lim_{T\to\infty} \left\| \frac{\partial \boldsymbol{z}_T}{\partial \boldsymbol{z}_1} \right\| \geq \lim_{T\to\infty} (1+\epsilon)^{T-1}. \tag{77}$$

Thereby

$$\exists\, T_0 > 1 \;\; s.t. \;\; \forall\, T \geq T_0 \implies \left\| \frac{\partial \boldsymbol{z}_T}{\partial \boldsymbol{z}_1} \right\| \geq (1+\epsilon)^{T-1}, \tag{78}$$

and thus

$$\exists\, T_0 > 1 \;\; s.t. \;\; \forall\, T \geq T_0 \implies \frac{\ln \left\| \frac{\partial \boldsymbol{z}_T}{\partial \boldsymbol{z}_1} \right\|}{T-1} \geq \frac{\ln(1+\epsilon)^{T-1}}{T-1}. \tag{79}$$

This means $\ln(1+\epsilon) \leq 0$ as $T \to \infty$, i.e. $\epsilon \leq 0$, which is a contradiction.

Therefore (14) holds for Γ and also, according to Oseledec's multiplicative ergodic Theorem, for every $\boldsymbol{z}_1$ in the basin of attraction of Γ. □

A.3 Additional results on relation between dynamics and gradients

A.3.1 Further results and remarks related to Theorem 2

Remark 4. *The result of Theorem 2 also holds for unstable orbits* $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3, \cdots\}$ *with positive largest Lyapunov exponent. Trivially, for such orbits that diverge to infinity (unbounded latent states) gradients of the loss function will explode as* $T \to \infty$*.*

Remark 5. *For RNNs with ReLU activation functions there are finite compartments in the phase space each with a different functional form. In such a case, to define the largest Lyapunov exponent of* Γ^**, in the proof of Theorem 2 we assume that* Γ^* *never maps to the points of the borders.*

Based on Theorem 2, we can also formulate the necessary conditions for chaos and diverging gradients in standard RNNs with particular activation functions by considering the norms of their recurrence matrix, for which the following Corollary provides the basis:

Corollary 1. *Let for a standard RNN*

$$\left\| diag\big(f'(\boldsymbol{W}\boldsymbol{z}_{t-1} + \boldsymbol{B}\boldsymbol{s}_t + \boldsymbol{h})\big) \right\| \leq \gamma < \infty. \tag{80}$$

If the RNN is chaotic, then $\|\boldsymbol{W}\| \, \gamma > 1$.

Proof. Assume for the sake of contradiction that $\|\boldsymbol{W}\| \, \gamma \leq 1$. From

$$\begin{aligned} \left\| \prod_{2<t\leq T} \boldsymbol{W} diag\big(f'(\boldsymbol{W}\boldsymbol{z}_{t-1} + \boldsymbol{B}\boldsymbol{s}_t + \boldsymbol{h})\big) \right\| &\leq \prod_{2<t\leq T} \left\| \boldsymbol{W} diag\big(f'(\boldsymbol{W}\boldsymbol{z}_{t-1} + \boldsymbol{B}\boldsymbol{s}_t + \boldsymbol{h})\big) \right\| \\ &\leq (\|\boldsymbol{W}\| \, \gamma)^{T-2}, \end{aligned} \tag{81}$$

it is concluded that $\lim_{T\to\infty} \left\| \prod_{2<t\leq T} \boldsymbol{W} diag\big(f'(\boldsymbol{W}\boldsymbol{z}_{t-1} + \boldsymbol{B}\boldsymbol{s}_t + \boldsymbol{h})\big) \right\| < \infty$, which contradicts (13). This means $\|\boldsymbol{W}\| \, \gamma > 1$ is a necessary condition for the standard RNN to be chaotic. □

Remark 6. *For RNN with the tanh and sigmoid activation functions* $\gamma = 1$ *and* $\gamma = \frac{1}{4}$*, respectively. Thus, by Corollary 1, the necessary conditions for chaos in these two cases are* $\|\boldsymbol{W}\| > 1$ *and* $\|\boldsymbol{W}\| > 4$*, respectively.*

A.3.2 Other connections between dynamics and gradients

There is a direct link between the norms of the Jacobians of the RNN along trajectories and the EVGP. By observing this link, we can formulate some general conditions that will have implications for the behavior of the gradients regardless of the limiting behavior of the RNN, as collected in the following theorem:

Theorem 4. *Let* $\mathcal{O}_{\boldsymbol{z}_1} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots \boldsymbol{z}_T, \cdots\}$ *be a sequence (orbit) generated by an RNN* $F_{\boldsymbol{\theta}} \in \mathcal{R}$ *parameterized by* $\boldsymbol{\theta}$*, and* $\boldsymbol{P}_T := \boldsymbol{J}_T - \boldsymbol{I}, \; T = 2, 3, \cdots$.

- *(i) Assume that* $\mathcal{O}_{\boldsymbol{z}_1}$ *is an orbit for which* $\left\| \frac{\partial^+ \boldsymbol{z}_T}{\partial \theta} \right\| \leq \xi \;\; \forall t$*. If* $\sum_{T=2}^{\infty} \|\boldsymbol{J}_T\| < \infty$*, then the Jacobian* $\frac{\partial \boldsymbol{z}_T}{\partial \boldsymbol{z}_1}$*, the tangent vector* $\frac{\partial \boldsymbol{z}_T}{\partial \theta}$ *and thus the gradient of the loss function,* $\frac{\partial \mathcal{L}_T}{\partial \theta}$*, will be bounded for* $T \to \infty$*.*
- *(ii) If* $\sum_{T=2}^{\infty} \|\boldsymbol{P}_T\| < \infty$*, then the Jacobian* $\frac{\partial \boldsymbol{z}_T}{\partial \boldsymbol{z}_1}$ *will neither vanish nor explode as* $T \to \infty$*.*
- *(iii) Let* $\|\boldsymbol{J}_T\| \neq 0, \; T \geq 2$*, and* $\sum_{T=2}^{\infty} \ln \|\boldsymbol{J}_T\|$ *diverge to* $-\infty$*, then the Jacobian* $\frac{\partial \boldsymbol{z}_T}{\partial \boldsymbol{z}_1}$ *vanishes as* T *tends to infinity.*

Part (i) of Theorem 4 relaxes some of the conditions required in Theorem 1 for bounded gradients by imposing a Lipschitz condition on the immediate derivatives. Part (ii) generalizes conditions satisfied, for instance, in orthogonal (unitary) RNNs [4, 32] or fully regularized PLRNNs [80].

Proof. Let $\|.\|$ be any matrix norm satisfying $\|\boldsymbol{A}_1 \boldsymbol{A}_2\| \leq \|\boldsymbol{A}_1\| \, \|\boldsymbol{A}_2\|$.

(i) By boundedness of $\frac{\partial^+ \boldsymbol{z}_T}{\partial \theta}$ we have

$$\left\| \frac{\partial \boldsymbol{z}_T}{\partial \theta} \right\| = \left\| \frac{\partial^+ \boldsymbol{z}_T}{\partial \theta} + \sum_{t=1}^{T-2} \Big(\prod_{r=0}^{t-1} \boldsymbol{J}_{T-r} \Big) \frac{\partial^+ \boldsymbol{z}_{T-t}}{\partial \theta} \right\|$$

$$\leq \xi \Big(1 + \sum_{t=1}^{T-2} \left\| \prod_{r=0}^{t-1} \boldsymbol{J}_{T-r} \right\| \Big) \leq \xi \Big(1 + \sum_{t=1}^{T-2} \prod_{r=0}^{t-1} \|\boldsymbol{J}_{T-r}\| \Big). \tag{82}$$

Moreover,

$$1 + \sum_{t=1}^{T-2} \prod_{r=0}^{t-1} \|\boldsymbol{J}_{T-r}\| \leq 1 + \sum_p \|\boldsymbol{J}_p\| + \sum_{p<q} \|\boldsymbol{J}_p\| \, \|\boldsymbol{J}_q\| + \sum_{p<q<r} \|\boldsymbol{J}_p\| \, \|\boldsymbol{J}_q\| \, \|\boldsymbol{J}_r\| + \cdots$$

$$= \big(1 + \|\boldsymbol{J}_T\|\big)\big(1 + \|\boldsymbol{J}_{T-1}\|\big) \cdots \big(1 + \|\boldsymbol{J}_2\|\big) =: \prod_{t=2}^{T} \big(1 + \|\boldsymbol{J}_t\|\big). \tag{83}$$

Since $\sum_{T=2}^{\infty} \|\boldsymbol{J}_T\|$ converges, according to [94], the infinite products $\prod_{T=2}^{\infty} \big(1 + \|\boldsymbol{J}_T\|\big)$ in (83) converge to a finite number $\tilde{\mathcal{K}} \neq 0$. Consequently, by (82) and (83)

$$\lim_{T \to \infty} \left\| \frac{\partial \boldsymbol{z}_T}{\partial \theta} \right\| \leq \tilde{\mathcal{K}} < \infty, \tag{84}$$

which implies $\frac{\partial \mathcal{L}_T}{\partial \theta}$ will be bounded for $T \to \infty$.

Furthermore

$$\lim_{T \to \infty} \left\| \frac{\partial \boldsymbol{z}_T}{\partial \boldsymbol{z}_1} \right\| \leq \prod_{T=2}^{\infty} \|\boldsymbol{J}_T\| := \lim_{T \to \infty} \Big(\|\boldsymbol{J}_T\| \, \|\boldsymbol{J}_{T-1}\| \cdots \|\boldsymbol{J}_2\| \Big) \leq \prod_{T=2}^{\infty} \big(1 + \|\boldsymbol{J}_T\|\big) \leq \tilde{\mathcal{K}}, \tag{85}$$

which completes the proof.

(ii) Since $\sum_{T=1}^{\infty} \|\boldsymbol{P}_T\| < \infty$, due to [94] the infinite product

$$\prod_{T=2}^{\infty} \big(\boldsymbol{I} + \boldsymbol{P}_T\big) = \prod_{T=2}^{\infty} \boldsymbol{J}_T := \lim_{T \to \infty} \boldsymbol{J}_T \, \boldsymbol{J}_{T-1} \cdots \boldsymbol{J}_2, \tag{86}$$

converges to a matrix $\boldsymbol{K} \neq \boldsymbol{O}$, which implies

$$0 < \lim_{T \to \infty} \left\| \frac{\partial \boldsymbol{z}_T}{\partial \boldsymbol{z}_1} \right\| = \|\boldsymbol{K}\| < \infty. \tag{87}$$

(iii) For $\|\boldsymbol{J}_T\| \neq 0, \; T \geq 2$, we have

$$0 \leq \left\| \frac{\partial \boldsymbol{z}_T}{\partial \boldsymbol{z}_1} \right\| \leq \|\boldsymbol{J}_T\| \, \|\boldsymbol{J}_{T-1}\| \cdots \|\boldsymbol{J}_2\|$$

$$= e^{\ln \|\boldsymbol{J}_T\|} e^{\ln \|\boldsymbol{J}_{T-1}\|} \cdots e^{\ln \|\boldsymbol{J}_2\|} = e^{\sum_{t=2}^{T} \ln \|\boldsymbol{J}_t\|}. \tag{88}$$

Hence if $\sum_{T=2}^{\infty} \ln \|\boldsymbol{J}_T\| \to -\infty$, then

$$\lim_{T \to \infty} \frac{\partial \boldsymbol{z}_T}{\partial \boldsymbol{z}_1} = \boldsymbol{O}. \tag{89}$$

□

A.4 Empirical evaluation: Datasets

Lorenz attractor The Lorenz system [56] is a simplified model for atmospheric convection, given by

$$\begin{aligned} \frac{\mathrm{d}x}{\mathrm{d}t} &= \sigma(y - x), \\ \frac{\mathrm{d}y}{\mathrm{d}t} &= x(\rho - z) - y, \\ \frac{\mathrm{d}z}{\mathrm{d}t} &= xy - \beta z. \end{aligned} \tag{90}$$

The system is of particular interest for its chaotic regime and was studied here for $\sigma = 16$, $\rho = 45.92$ and $\beta = 4$. For these parameters the Lorenz system is known to have a maximal Lyapunov exponent $\lambda_{\max} = 1.5$ [72]. To generate a time series, the ODEs were integrated with a step size $\Delta t = 0.01$ using `scipy.integrate`. Accordingly, the prediction time is $\tau_{pred} = \frac{\ln(2)}{\Delta t\ \lambda_{max}} = 46.2$.

Duffing oscillator The Duffing oscillator [15] is an example of a periodically forced oscillator with nonlinear elasticity

$$\ddot{x} + \delta\dot{x} + \beta x + \alpha x^3 = \gamma\cos(\omega t). \tag{91}$$

Note that this system is non-autonomous, that is externally forced due to the r.h.s. of eqn. 91. The following parameters were chosen to arrive at a chaotically forced oscillator: $\alpha = 1.0$, $\beta = -1.0$, $\delta = 0.1$, $\gamma = 0.35$, and $\omega = 1.4$. For these parameters the Duffing oscillator has a maximum Lyapunov exponent of $\lambda_{max} = 0.0995$. The dataset used here was created with the code from [24] as a three dimensional embedding with step size $\Delta t = 0.17$. The prediction time is $\tau_{pred} = 39.28$.

Rössler system Another prime textbook example for a chaotic system is the Rössler system [76] given by:

$$\begin{aligned} \frac{dx}{dt} &= -y - z, \\ \frac{dy}{dt} &= x + ay, \\ \frac{dz}{dt} &= b + z(x - c). \end{aligned} \tag{92}$$

For the parameters $a = 0.15$, $b = 0.2$ and $c = 10$, the maximal Lyapunov exponent is $\lambda_{\max} = 0.09$ [72]. To arrive at a time series, a step size of $\Delta t = 0.1$ was chosen for integration. This gives us a prediction time of $\tau_{pred} = 77.0$ for this system.

Mackey-Glass equation The Mackey-Glass equation [25] is a nonlinear time delay differential equation

$$\dot{x} = \beta\frac{x_\rho}{1 + x_\rho^n} - \gamma x \quad \text{with } \beta, \gamma, \rho > 0. \tag{93}$$

Here x_ρ represents the value of the variable x at time $t - \rho$ (note that strictly, mathematically, this makes the system infinite-dimensional). Choosing the parameters to be $\beta = 2$, $\gamma = 1.0$, $n = 9.65$, and $\rho = 2.0$, leads to chaotic behavior with a maximum Lyapunov exponent of $\lambda_{max} = 0.21$. The dataset was created as a 10-dimensional embedding with the code from [24] using $\Delta t = 0.04$. This yields a prediction time of $\tau_{pred} = 82.2$.

Empirical temperature time series This time series was recorded at the Weather Station at the Max Planck Institute for Biogeochemistry in Jena, Germany, spanning the time period between 2009 and 2016, and reassembled by François Chollet for the book *Deep Learning with Python*. The data set can be accessed at https://www.kaggle.com/pankrzysiu/weather-archive-jena.
To expose the underlying chaotic dynamics of the time series, trends and yearly cycles were removed, and nonlinear noise-reduction was performed (using `ghkss` from *TISEAN*, see also [43]). Fig. 5 (a) shows a snippet of the temperature data in comparison with the de-noised time-series. High-frequency

noise was further reduced through Gaussian kernel smoothing ($\sigma = 200$), and the resulting time series was sub-sampled (every 5th data point was retained). Fig. 5 (b) clearly reveals a fractional dimension of $D_{eff} = 2.8$ for the de-noised and smoothed time-series. This strongly suggests that the dynamics governing the time series are chaotic. We created a time delay embedding [42] with $m = 5$ (estimated by the false nearest neighbor technique, see [44]) and delay $\Delta t = 500$ (obtained as the first minimum of the mutual information). The first three embedding dimensions are shown in Fig. 5(c). The maximal Lyapunov exponent of this time series was determined with `lyap_r` from *TISEAN* [30] to be $\lambda_{\max} = 0.016$, see Fig. 4(a). This value is in close agreement with the literature [62]. The predictability time of this system is estimated to be $\tau_{pred} = 43.3$.

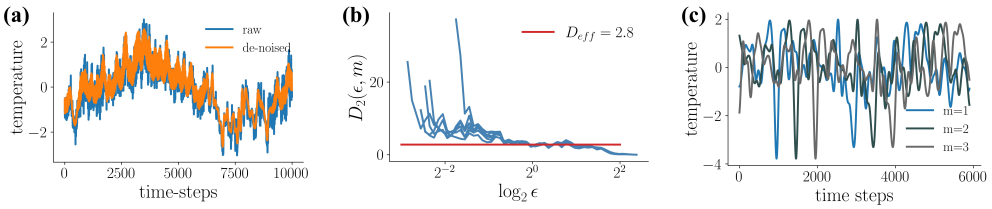


Figure 5: (a) Snippet of the original temperature data and de-noised time series. (b) Blue lines show the local slopes of the correlation sums for embedding dimensions $m \in \{5, \ldots, 10\}$. The convergence of these estimates in m reveals a fractional dimension indicated by the plateau. (c) First three dimensions of the time-delay embedding series as used for training.

All datasets used were standardized (i.e., centered with unit variance) prior to training.

A.5 Empirical evaluation: measures of reconstruction quality

Attractor overlap To asses the geometrical similarity of the chaotic attractor produced by the RNN to the one underlying the observations, we calculate the Kullback-Leibler divergence of the ground truth distribution $p_{\text{true}}(\boldsymbol{x})$ and the distribution $p_{\text{gen}}(\boldsymbol{x}|\boldsymbol{z})$ generated by RNN simulation. To do so in practice, we employ a binning approximation (see [50])

$$D_{\text{stsp}}\left(p_{\text{true}}(\mathbf{x}), p_{\text{gen}}(\mathbf{x} \mid \mathbf{z})\right) \approx \sum_{k=1}^{K} \hat{p}_{\text{true}}^{(k)}(\mathbf{x}) \log\left(\frac{\hat{p}_{\text{true}}^{(k)}(\mathbf{x})}{\hat{p}_{\text{gen}}^{(k)}(\mathbf{x} \mid \mathbf{z})}\right),$$

where K is the total number of bins, and $\hat{p}_{\text{true}}^{(k)}(\mathbf{x})$ and $\hat{p}_{\text{gen}}^{(k)}(\mathbf{x} \mid \mathbf{z})$ are estimates obtained as relative frequencies through sampling trajectories from the observed time-series and the trained RNN, respectively.

Hellinger distance between power spectra Since in DS reconstruction we mainly aim to capture invariant and time-independent properties of the underlying system, besides the geometrical agreement, we compare the similarity in true and RNN-reconstructed power spectra. To do so, we generate a time series of length $100,000$ from the RNN and calculate its dimension-wise power spectra $S(\omega)$ using the fast Fourier transform (`scipy.fft`). By standardizing all trajectories prior to Fourier transforming them, we have $\int_{-\infty}^{\infty} S(\omega) = 1$ due to the Plancherel theorem. This allows us to compare two power spectra, $S(\omega)$ and $P(\omega)$, with the Hellinger distance

$$H(S(\omega), P(\omega)) = \sqrt{1 - \int_{-\infty}^{\infty} \sqrt{S(\omega)P(\omega)}\, d\omega} \;\in\; [0, 1]. \tag{94}$$

To reduce the influence of noise we apply Gaussian kernel smoothing. The Hellinger distances between observed and generated spectra for all dimensions are then averaged to give the reported overall distance D_H.

A.6 Further empirical evaluations

A.6.1 Reconstruction: Rössler System

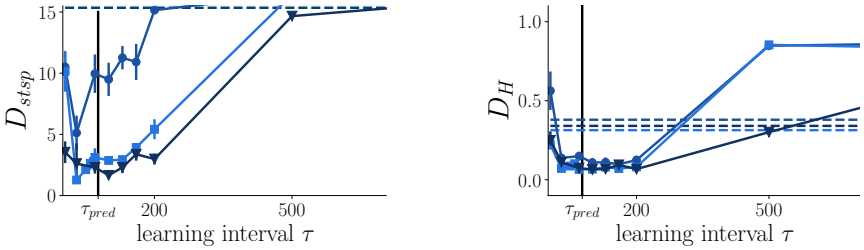


Figure 6: Overlap in attractor geometry (D_{stsp}, lower = better) and dimension-wise comparison of power-spectra (D_H, lower = better) against learning interval τ for the Rössler attractor. Continuous lines = sparsely forced BPTT. Dashed lines = classical BPTT with gradient clipping. Prediction time indicated vertically in black.

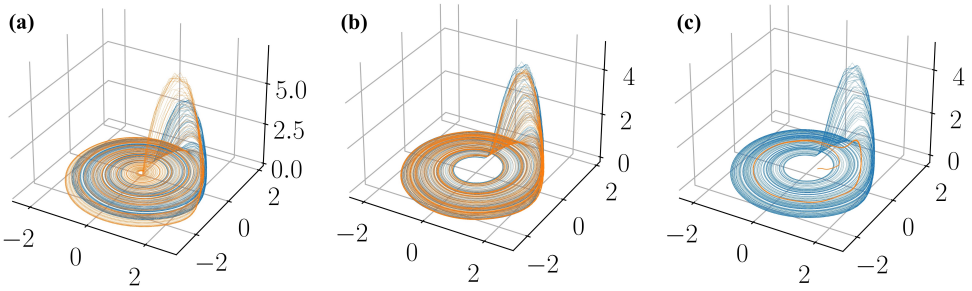


Figure 7: The Rössler attractor (blue) and reconstruction by a LSTM (orange) trained with a learning interval (a) chosen too small ($\tau = 5$), (b) chosen optimally ($\tau = 30$), and (c) chosen too large ($\tau = 200$).

A.6.2 Reconstruction: High-dimensional Mackey-Glass system

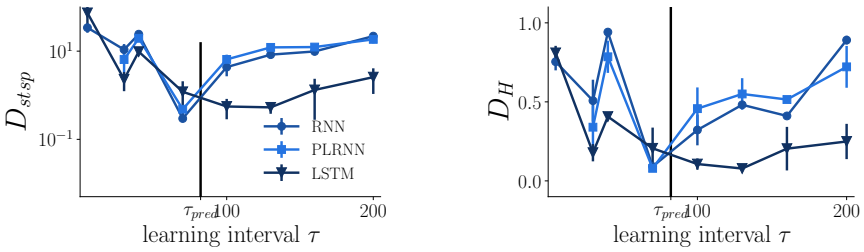


Figure 8: Overlap in attractor geometry (D_{stsp}, lower = better) and dimension-wise comparison of power-spectra (D_H, lower = better) against learning interval τ for the 10d Mackey-Glass system. Continuous lines = sparsely forced BPTT. Prediction time indicated vertically in black.

A.6.3 Reconstruction: Partially observed Lorenz System

For this evaluation we trained models only on the variables $\{y, z\}$ of the Lorenz system, eqn. 90. In order to compute the attractor overlap (D_{stsp}) in the true state space, however, after training the observation matrix $\boldsymbol{B}$ was recomputed by linearly regressing the first 10 latent states onto the first 10 observations from all three Lorenz variables in eqn. 90.

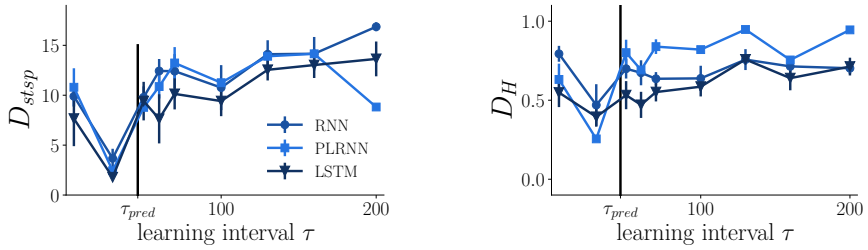


Figure 9: Overlap in attractor geometry (D_{stsp}, lower = better) and dimension-wise comparison of power-spectra (D_H, lower = better) against learning interval τ for the partially observed Lorenz system. Continuous lines = sparsely forced BPTT. Prediction time indicated vertically in black.

A.6.4 Other initialization procedures: Truncated BPTT with zero resetting or forward-iterated states

A common procedure in training RNNs is partitioning the time series into chunks of length τ (as we did based on the Lyapunov spectrum), but then simply resetting the hidden states $\boldsymbol{z}_{1(k)}$ at the beginning of each chunk (window) k to $\mathbf{0}$, or forward-iterating them from the previous chunk $k-1$, i.e. $\boldsymbol{z}_{1(k)} = F_{\boldsymbol{\theta}}(\boldsymbol{z}_{\tau(k-1)})$. Formally this would mean that we do not force the trajectory back on track as in our approach, but instead may either kick it off track (zero resetting) or just let it freely evolve whilst still truncating the gradients (forward-iterating). To illustrate this, here we trained an LSTM on chunks (windows) with a length given by the optimal τ ($\tau_{opt} = 30$ for the Lorenz system), but then initialized the hidden states to either 0 or to the forward-iterated last state at the beginning of each window. The performance obtained with zero-resetting is indicated by the dashed line in Fig. 10a below, while the performance with forward-iterated states is shown in Fig. 10c. As another control, we also checked dependence on window length (without forcing) in Fig. 10b.

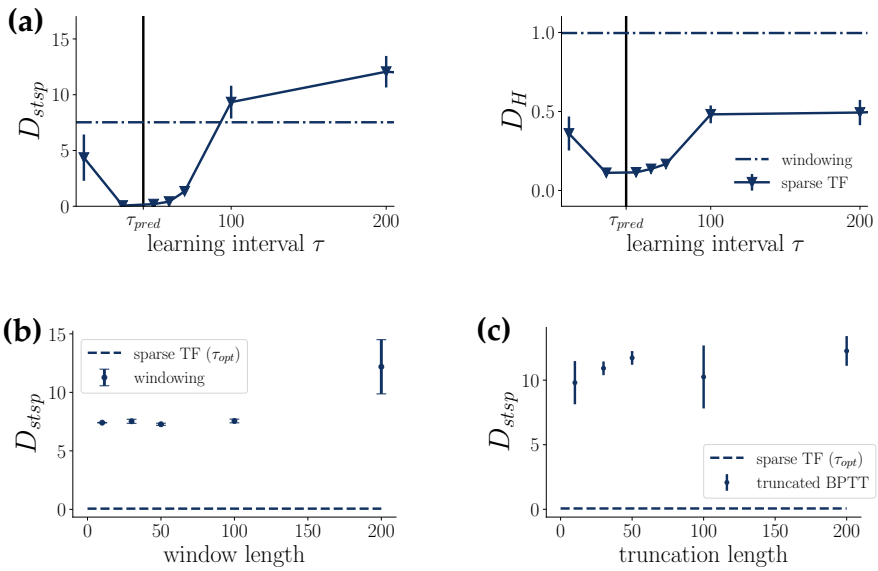


Figure 10: (a) Overlap in attractor geometry (D_{stsp}, lower = better) and dimension-wise comparison of power-spectra (D_H, lower = better) against learning interval τ for the Lorenz system. Continuous lines = sparsely forced BPTT. Dashed-dotted lines = windowing without forcing (choosing windows according to the optimal prediction time, but resetting hidden states to zero rather than its TF control value). Prediction time indicated vertically in black. (b) Dependence of geometrical reconstruction quality on window length. Without forcing, the window length hardly has any bearing on reconstruction quality. (c) Same as (b) but with initial states of each window k forward-iterated from the previous window's state, $\boldsymbol{z}_{1(k)} = F_{\boldsymbol{\theta}}(\boldsymbol{z}_{\tau(k-1)})$, instead of zero resetting.

A.6.5 Electroencephalogram (EEG) data

We used EEG data recorded by Schalk et al. [79] and provided on PhysioNet [27], from which we took the baseline recording of the first patient for our analysis. Preprocessing was performed as outlined above for the temperature time-series, i.e. we applied nonlinear noise-reduction (see Fig.11 (a)) and Gaussian kernel smoothing ($\sigma = 5$). Fig. 11 (b) indicates a fractional dimension $D_{eff} = 2.5$ for the de-noised and smoothed times series. We created a time delay embedding with an embedding dimension of $m = 10$ and a delay time of $\Delta t = 40$. The maximal Lyapunov exponent for this time series was determined to be $\lambda_{max} = 0.017$, see Fig. 12 (a). With this, we obtain a predictability time $\tau_{pred} = 40.77$.

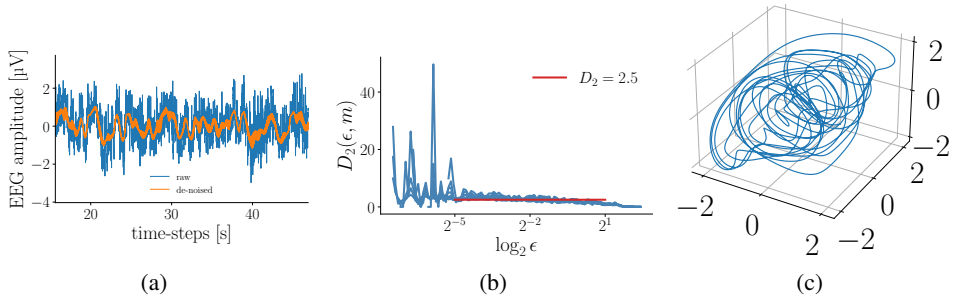


Figure 11: (a) Snippet of the original EEG data and de-noised time series. (b) Blue lines show the local slopes of the correlation sums for embedding dimensions $m \in \{5, \ldots, 15\}$. The convergence of these estimates in m reveals a fractional dimension indicated by the plateau. (c) First three dimensions of the time-delay embedding series as used for training.

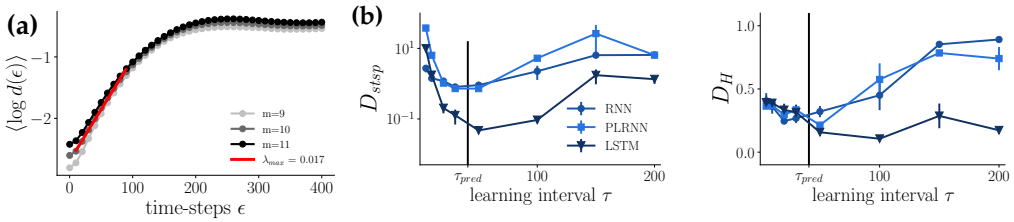


Figure 12: (a) The maximal Lyapunov exponent was determined as the slope of the average log-divergence of nearest neighbors in embedding space (m = embedding dimension). (b) Reconstruction quality assessed by attractor overlap (lower = better) and dimension-wise comparison of power-spectra (D_H, lower = better). Black vertical lines = τ_{pred}.

A.6.6 Miscellaneous additional results

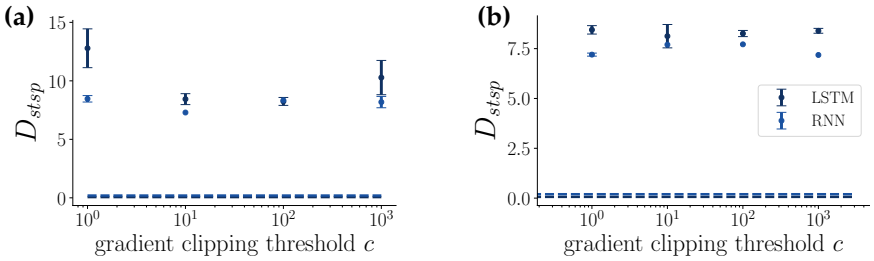
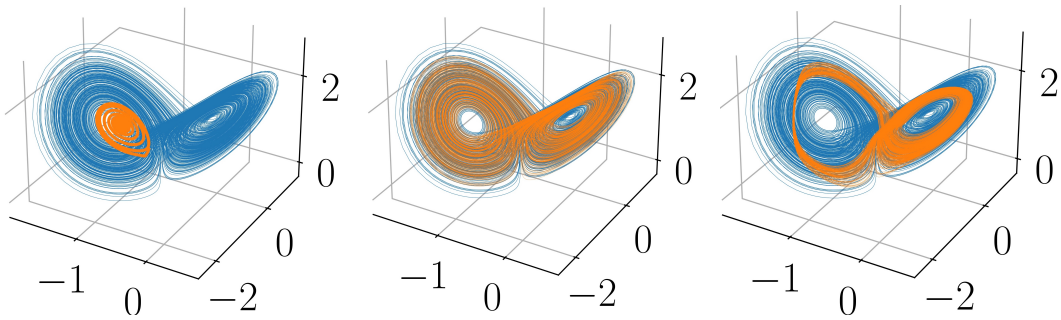


Figure 13: Dependence of geometrical reconstruction quality (D_{stsp}) on the Lorenz system for various clipping thresholds in classical BPTT. (a) Gradient clipping by constraining the Euclidean norm to c. (b) Gradient clipping by constraining the max (infinity) norm to c. For comparison, in both graphs the values obtained for sparse teacher forcing with optimal forcing interval τ_{pred} are shown as dashed lines.



(a) learning interval τ too small (b) learning interval τ optimal (c) learning interval τ too large

Figure 14: Same as Fig. 3 for vanilla RNNs. Although, as this graph confirms, with sparsely forced BPTT training of vanilla RNNs on chaotic systems becomes feasible, generally they were somewhat harder to train than the other RNN architectures (likely due to their known problems with long-range dependencies).

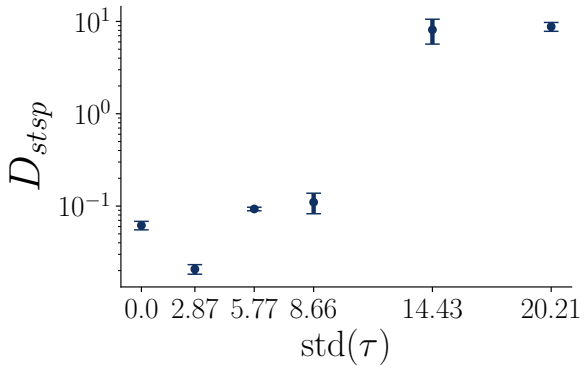


Figure 15: Teacher forcing for LSTM with the learning interval τ drawn uniformly random around the optimal value ($\tau_{opt} = 30$) with standard deviation std(τ) (for std(τ) > 8.66 the interval becomes asymmetric, however, due to the lower bound at $\tau = 1$). As std(τ) is increased, performance generally degrades. A little jittering around the optimal interval τ_{pred} may potentially help, however (as more commonly observed in various machine learning procedures).

A.7 Sparsely forced BPTT

Loss truncation One implicit consequence of the teacher forcing, eqn. (16), is the interruption of the hidden-to-hidden connections at these time points. More specifically, if the system is forced at time $t \in \mathcal{T}$, then there is no connection between $\boldsymbol{z}_t$ and $\boldsymbol{z}_{t+1}$, that is

$$\boldsymbol{J}_{t+1} = \frac{\partial \boldsymbol{z}_{t+1}}{\partial \boldsymbol{z}_t} = \frac{\partial RNN(\tilde{\boldsymbol{z}}_t)}{\partial \boldsymbol{z}_t} = 0. \tag{95}$$

To see how these vanishing Jacobians truncate the loss gradients w.r.t to some parameter θ, let us focus on the loss gradients immediately after the forcing,

$$\begin{aligned}
\frac{\partial \mathcal{L}_{t+1}}{\partial \theta} &= \frac{\partial \mathcal{L}_{t+1}}{\partial \boldsymbol{z}_{t+1}} \sum_{k=1}^{t+1} \frac{\partial \boldsymbol{z}_{t+1}}{\partial \boldsymbol{z}_k} \frac{\partial^+ \boldsymbol{z}_k}{\partial \theta} \\
&= \frac{\partial \mathcal{L}_{t+1}}{\partial \boldsymbol{z}_{t+1}} \Big(\frac{\partial^+ \boldsymbol{z}_{t+1}}{\partial \theta} + \sum_{k=1}^{t} \underbrace{\frac{\partial \boldsymbol{z}_{t+1}}{\partial \boldsymbol{z}_k}}_{=0\text{ , because of (95)}} \frac{\partial^+ \boldsymbol{z}_k}{\partial \theta} \Big) \\
&= \frac{\partial \mathcal{L}_{t+1}}{\partial \boldsymbol{z}_{t+1}} \frac{\partial^+ \boldsymbol{z}_{t+1}}{\partial \theta}.
\end{aligned} \tag{96}$$

Eqn. (96) shows that sparsely forced BPTT implicitly truncates the loss gradients because it interrupts the hidden-to-hidden connection from $\boldsymbol{z}_t$ to $\boldsymbol{z}_{t+1}$ for $t \in \mathcal{T}$. More generally, defining $\widetilde{t} := \max\{t' \in$

$\mathcal{T} : t' \leq t\}$, the overall loss gradients are truncated to

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \theta} &= \sum_{t=1}^{T} \frac{\partial \mathcal{L}_t}{\partial \boldsymbol{z}_t} \sum_{k=1}^{t} \frac{\partial \boldsymbol{z}_t}{\partial \boldsymbol{z}_k} \frac{\partial^+ \boldsymbol{z}_k}{\partial \theta} \\ &\overset{\text{tr.}}{=} \sum_{t=1}^{T} \frac{\partial \mathcal{L}_t}{\partial \boldsymbol{z}_t} \sum_{k=\widetilde{t}}^{t} \frac{\partial \boldsymbol{z}_t}{\partial \boldsymbol{z}_k} \frac{\partial^+ \boldsymbol{z}_k}{\partial \theta}. \end{aligned} \tag{97}$$