Optimal Recurrent Network Topologies for Dynamical Systems Reconstruction

Christoph Jürgen Hemmer [1 2] **Manuel Brenner** [1 2] **Florian Hess** [1 2] **Daniel Durstewitz** [1 2 3]

Abstract

In dynamical systems reconstruction (DSR) we seek to infer from time series measurements a generative model of the underlying dynamical process. This is a prime objective in any scientific discipline, where we are particularly interested in parsimonious models with a low parameter load. A common strategy here is parameter pruning, removing all parameters with small weights. However, here we find this strategy does not work for DSR, where even low magnitude parameters can contribute considerably to the system dynamics. On the other hand, it is well known that many natural systems which generate complex dynamics, like the brain or ecological networks, have a sparse topology with comparatively few links. Inspired by this, we show that *geometric pruning*, where in contrast to magnitude-based pruning weights with a low contribution to an attractor's geometrical structure are removed, indeed manages to reduce parameter load substantially without significantly hampering DSR quality. We further find that the networks resulting from geometric pruning have a specific type of topology, and that this topology, and not the magnitude of weights, is what is most crucial to performance. We provide an algorithm that automatically generates such topologies which can be used as priors for generative modeling of dynamical systems by RNNs, and compare it to other well studied topologies like small-world or scale-free networks.

1. Introduction

In scientific settings we are commonly interested in the dynamical rules that govern the temporal evolution of an observed system. Recent data-driven deep learning approaches aim to infer (approximate) these from time series measurements of the system under study, often based on recurrent neural networks (RNNs) like LSTMs (Vlachas et al., 2018; Hochreiter & Schmidhuber, 1997), reservoir computers (Pathak et al., 2018; Platt et al., 2021), piecewise linear RNNs (Durstewitz, 2017; Brenner et al., 2022; Hess et al., 2023), or neural ordinary differential equations (Neural ODEs; Chen et al. (2018); Ko et al. (2023)). Producing a generative model of the dynamical process that underlies the observed time series, including its geometrical structure in state space and its invariant (long-term) statistics, we call dynamical systems (DS) reconstruction (DSR).

In science we usually prefer small models, with as few relevant parameters as possible, to reduce training and simulation times, and to ease subsequent model analysis (interpretability). Magnitude-based parameter pruning is a well established strategy in deep learning to carve out such a low-dimensional (in parameter space) network structure (Blalock et al., 2020). It is based on the insight that successfully trained models often contain a 'winning ticket', a small subnetwork with performance almost equal to that of the full large network (Frankle & Carbin, 2019). Here, however, we demonstrate that magnitude-based parameter tuning does not work well for DSR. Instead, we find that parameters even small in relative size could substantially influence the dynamics of the trained model (Fig. 1). This might be related to the fact that in dynamical systems neighborhood relations (topological structure) are often more important. Natural systems, like the brain (Bullmore & Sporns, 2009; Pajevic & Plenz, 2012), climate systems (Tziperman et al., 1997), or ecological and social networks (Watts & Strogatz, 1998), rarely follow an "all-to-all" connectivity, but have a well defined topology owing to physical and spatial constraints on the system. This suggests that deep learning models trained on time series from natural systems may inherit similar organizing principles, i.e. a low-dimensional parameter representation is conceivable, but it may be less related to the relative size of parameters.

[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, Heidelberg, Germany. Correspondence to: Christoph Hemmer <Christoph.Hemmer@zi-mannheim.de>, Daniel Durstewitz <Daniel.Durstewitz@zi-mannheim.de>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Here we demonstrate that this is indeed the case. In contrast to pruning by size, selecting parameters based on their relevance for the invariant geometrical structure of the dynamical system enabled to profoundly sparsify recurrent neural networks (RNNs) without considerably affecting performance. Such *geometry-pruned* RNNs turned out to bear a specific topology, which *in itself* was largely sufficient for performance-preserving sparsification. We further find that while more 'traditional' models of network topology, like the Watts-Strogatz (Watts & Strogatz, 1998) or Albert-Barabási (Barabási & Albert, 1999) model, can also produce efficiency gains beyond magnitude-based or random pruning, these are not quite as profound as those obtained through our procedure.

2. Related Work

Dynamical Systems Reconstruction (DSR) In DSR we aim to obtain from time series data a generative model that is at least topologically conjugate to the flow of the underlying true system on the domain observed (Durstewitz et al., 2023), with the same invariant long-term properties, including attractor geometry and temporal characteristics (as assessed, e.g., through power spectra or auto-correlation functions; Wood (2010); Platt et al. (2021); Brenner et al. (2022); Mikhaeil et al. (2022); Platt et al. (2023)). The forecasting ability of a model on its own is usually not considered a viable performance criterion for DSR models, because in chaotic systems nearby trajectories diverge exponentially fast and hence there is only a limited prediction horizon (Wood, 2010; Koppe et al., 2019). Various architectures and training algorithms have been proposed for DSR, based on RNNs like LSTMs (Vlachas et al., 2018; Hochreiter & Schmidhuber, 1997), piecewise-linear RNNs (PLRNNs; Koppe et al. (2019); Brenner et al. (2022); Hess et al. (2023)), or reservoir computers (Pathak et al., 2018; Platt et al., 2021), based on library methods (Brunton et al., 2016; Champion et al., 2019; Messenger & Bortz, 2021), or based on neural ordinary differential equations (Neural ODEs; Chen et al. (2018); Karlsson & Svanström (2019); Alvarez et al. (2020); Ko et al. (2023)). A variety of different training strategies has been suggested for DSR models, e.g. based on Expectation-Maximization (Voss et al., 2004; Koppe et al., 2019) or on variational inference (Kramer et al., 2022). However, the to date most effective training algorithms rely on Backpropagation Through Time (BPTT) (Rumelhart et al., 1986) combined with control-theoretic forms of teacher forcing (TF; Williams & Zipser (1989)), like sparse (Mikhaeil et al., 2022; Brenner et al., 2022) or generalized (Hess et al., 2023) TF, that guarantee optimal trajectory and gradient flows whilst training, and avoid exploding gradients even on chaotic systems. The role of RNN topology in DSR has, however, not been studied yet, providing one novel contribution of the present work.

Pruning and Lottery Ticket Hypothesis Linked to the double descent phenomenon (Belkin et al., 2019), modern deep learning models are often extensively over-parameterized, surpassing the interpolation threshold, which leads to improved network trainability and expressiveness. While smaller models with fewer parameters are generally preferable for computational and memory reasons, and to enhance interpretability, this becomes almost imperative for these strongly over-parameterized systems. LeCun et al. (1989) is one of the first studies that explored pruning networks in order to remove redundancies in parameters. Since then it has been shown numerous times that in deep learning architectures a non-trivial number of parameters remains essentially unexploited (Han et al., 2015b;a), and a variety of different pruning techniques have been developed (Blalock et al., 2020), for instance, Hessian based approaches (Hassibi & Stork, 1992) or structured pruning (He & Xiao, 2023). The most common and straightforward procedure, however, remains pruning parameters based on their absolute value (Blalock et al., 2020). This can be done in different ways, for instance, pruning in a single step (Liu et al., 2018) or iteratively (Han et al., 2015c; Zhang et al., 2021). Hope that this may work more generally is based on the so-called lottery ticket hypothesis (LTH), which states (Frankle & Carbin, 2019): *"A randomly-initialized, dense neural network contains a subnetwork that is initialized such that – when trained in isolation – it can match the test accuracy of the original network after training for at most the same number of iterations."* Empirically this was verified for feed-forward neural networks in image classification using iterative magnitude pruning (Frankle & Carbin, 2019; Girish et al., 2021), and has spawned a range of theoretical and empirical follow-up studies (Malach et al., 2020b; Orseau et al., 2020; Zhang et al., 2021; Sreenivasan et al., 2022; Burkholz et al., 2022). Burkholz et al. (2022) even suggests the existence of universal LTs that are winning tickets across tasks. However, so far there is comparatively little work on the LTH and pruning of RNNs (Yu et al., 2019; Liu et al., 2021; Chatzikonstantinou et al., 2021).

Network Topology of Real World Systems Many complex real-world systems have a characteristic network topology (Watts & Strogatz, 1998; Albert & Barabási, 2002), for instance, a small-world topology shared by many biological and social systems (Kleinberg, 2000; Amaral et al., 2000; Bassett & Bullmore, 2006; Rubinov et al., 2009; Muldoon et al., 2016), or a scale-free topology observed in brain anatomy and dynamics (Beggs & Plenz, 2003; Eguiluz et al., 2005; van den Heuvel et al., 2008; Rubinov et al., 2009). Scale-free networks are characterized by the existence of central nodes or 'hubs', possibly linked to a hierarchical organization in the system (Ravasz & Barabási, 2003). While sometimes the specific network topology may simply be a result of the underlying physics (e.g., spatially restricted interaction patterns or physical barriers; Jiang & Claramunt (2004)), in other instances they may bear specific functional or biological advantages like minimizing wiring costs while optimizing information transfer (Bullmore & Sporns, 2012), or increasing efficiency (Zhang et al., 2009b).

Network Topology in RNNs Given the importance of topological structure in real-world networks, inferring topology directly from observed data is of particular interest (Shandilya & Timme, 2011; Wang et al., 2016), although not our focus here. Regarding the reverse direction, Emmert-Streib (2006) were among the first to point out that RNN topology also influences learning dynamics. This has been particularly well studied in the context of reservoir computers (RCs; Jaeger & Haas (2004)). Dutoit et al. (2009) observed that pruning connections in the reservoir's output layer leads to improved generalization. Yin & Meng (2012) adapt the structure of the dynamical reservoir to mirror that of cortical networks, while Carroll & Pecora (2019) study the influence of directedness of edges in RCs more generally. Other authors (Li et al., 2020; Junior et al., 2020; Dale et al., 2021) directly studied the impact of specific topologies, like hub structure, directed acyclic graphs, or Erdős–Rényi graphs, on RC performance. Han et al. (2022) study generalization bounds for RCs initialized with different graph structures. In contrast to our work here, however, all these studies impose a specific, previously defined topology to begin with, and do not examine structure or properties that arise through training or pruning (in fact, recall that in RCs the recurrent connectivity is *fixed* and not altered throughout training). This question thus remains largely unexplored, especially in the context of DSR. Small-world topology has, however, been observed in task-optimized *feedforward* networks like MLPs or CNNs (You et al., 2020), associated with superior performance.

3. Methodological Setting

3.1. DSR Model and Training

For our numerical studies we focus on a well established SOTA model and training algorithm for DSR, the piecewise linear recurrent neural network (PLRNN) first introduced in (Durstewitz, 2017), but also checked LSTMs and vanilla RNNs to highlight that our results are more general. The PLRNN is defined by

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

where $\mathcal{M}$ is a mean-centering operation (see Appx. A.1.1 and Brenner et al. (2022)), and with element-wise nonlinearity $\phi(\bullet) = \mathrm{ReLU}(\bullet) = \max(0, \bullet)$. The model describes the temporal evolution of an M-dimensional latent state vector, $\boldsymbol{z}_t \in \mathbb{R}^M$, with linear self-connections in diagonal matrix $\boldsymbol{A} \in \mathbb{R}^{M\times M}$, full weight matrix $\boldsymbol{W} \in \mathbb{R}^{M\times M}$, bias term $\boldsymbol{h} \in \mathbb{R}^M$, and possibly external inputs $\boldsymbol{s}_t \in \mathbb{R}^K$ weighted by $\boldsymbol{C} \in \mathbb{R}^{M\times K}$ (the benchmarks we explore in here, however, are all autonomous DS with $\boldsymbol{s}_t = 0 \ \forall t$).[1] The latent PLRNN is linked to the actually observed time series $\boldsymbol{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_T\}$ through a decoder (observation) model, in the simplest case given by a linear layer:

$$\boldsymbol{x}_t = G_{\boldsymbol{\lambda}}(\boldsymbol{z}_t) = \boldsymbol{B}\boldsymbol{z}_t \ , \tag{2}$$

where $\boldsymbol{B} \in \mathbb{R}^{N\times M}$. Numerous variations on this basic model have been introduced and benchmarked on DSR problems in the literature (Koppe et al., 2019; Brenner et al., 2022; Hess et al., 2023), but here we will stick to this most basic form to enhance interpretability of our results in graph-theoretical language (but see Appx. A.1 for further details).[2] Of major importance in scientific settings, a crucial advantage of model Eqn. 1 is that it allows for an equivalent continuous-time formulation (i.e., as a system of ODEs; Monfared & Durstewitz (2020)), and all its fixed points and cycles can be exactly determined by efficient, often linear-time, algorithms (Eisenmann et al., 2023).

In DSR, we would like to capture long-term statistical and geometrical properties of the underlying DS, beyond mere short-term forecasts (Platt et al., 2021; Durstewitz et al., 2023; Platt et al., 2023). It turns out that the actual training algorithm is much more important for this than the RNN architecture (Mikhaeil et al., 2022; Hess et al., 2023). In particular, efficient training routines often implement control-theoretic ideas like sparse (Mikhaeil et al., 2022) or generalized (Hess et al., 2023) teacher forcing (STF, GTF) that manage the exploding-&-vanishing gradient problem even for chaotic systems with diverging trajectories. To keep things simple, here we employ STF (Mikhaeil et al., 2022; Brenner et al., 2022) with an identity mapping for the observation model, i.e.

$$\boldsymbol{x}_t = \boldsymbol{\mathcal{I}}\boldsymbol{z}_t, \qquad \mathcal{I}_{kl} = \begin{cases} 1, & \text{if } k = l \text{ and } k \leq N \\ 0, & \text{else} \end{cases} . \tag{3}$$

STF then replaces the latent states $z_{t',k}, k \leq N$, with observations $x_{t',k}$ sparsely at strategically chosen time points $t' \in \mathcal{T} = \{n\tau + 1\}$ with $n \in \mathbb{N}$ (Mikhaeil et al., 2022), thereby re-calibrating trajectories during training such that relevant time scales are captured yet too wild divergence and exploding gradients are prevented (for details see Appx. A.1; note that no such forcing is used in the actual test or generation phase).

[1] Also, mathematically, a non-autonomous system can always be equivalently rewritten as an autonomous system (Zhang et al., 2009a).

[2] In fact, as shown in Brenner et al. (2022) and Hess et al. (2023), more complex PLRNN variants can be reformulated in terms of Eqn. 1.

3.2. Weight Pruning

We implement weight pruning (LeCun et al., 1989) by applying a mask $\boldsymbol{m}$ to the weight matrix $\boldsymbol{W}$ using element-wise multiplication:

$$\boldsymbol{z}_t = \boldsymbol{A}\boldsymbol{z}_{t-1} + (\boldsymbol{m} \odot \boldsymbol{W})\phi(\mathcal{M}(\boldsymbol{z}_{t-1})) + \boldsymbol{h}\,, \qquad (4)$$

where $\boldsymbol{m} \in \{0, 1\}^{M \times M}$ represents the network topology. Pseudo-code for the iterative pruning procedure, retaining initial parameters $\boldsymbol{\theta}_0$ but updating the mask in each iteration, is given in Algorithm 1.

Algorithm 1 Pruning algorithm

```
Input  : Initial model $f(\boldsymbol{x}; \boldsymbol{\theta}_0 = \{\boldsymbol{A}_0, \boldsymbol{W}_0, \boldsymbol{h}_0\})$
         with initial parameters $\boldsymbol{\theta}_0$
Output : Mask $\boldsymbol{m}$ of pruned network
Initialize $m^1_{ij} = 1 \; \forall i, j \in [1, ..., M]$
for $k \leftarrow 1$ to $n$ do
    1. Train model $f\left(\boldsymbol{x}; \boldsymbol{\theta}_0 = \{\boldsymbol{A}_0, \boldsymbol{m}^k \odot \boldsymbol{W}_0, \boldsymbol{h}_0\}\right)$
       for $j$ epochs, yielding parameters $\boldsymbol{\theta}$
    2. Remove $p\%$ of parameters $w_{ij} \in \boldsymbol{\theta}$ based on
       their contribution $I_{w_{ij}}$ to model performance,
       resulting in mask $\boldsymbol{m}^{k+1}$
    3. Reset parameters to $\boldsymbol{\theta_0}$
end
```

Traditionally in pruning procedures, importance I_{θ_i} of a parameter is simply measured by its absolute magnitude, i.e. $I_{\theta_i} = |\theta_i|$. As we find below that weight magnitude is only weakly correlated with DSR performance, we introduce *geometric pruning* as a means to examine 1) whether a significant sparsification of the network is possible in this context, and 2) which parameters do have a significant impact. In geometric pruning we iteratively, using the same protocol as in Algorithm 1, remove those connections that have the least impact on attractor geometry in state space (Fig. 1). Since in DSR we are interested in obtaining a generative model that has the same long-term temporal behavior and geometrical structure in state space as the true underlying DS, this is a direct indicator of DSR quality. Formally we define it through the same measure that has been used to assess geometrical agreement, a Kullback-Leibler (KL) divergence in the system's *state space* (Koppe et al., 2019; Hess et al., 2023), namely

$$\begin{aligned} I_{\theta_i} = \big| &\mathrm{KL}(p_{true}(\boldsymbol{x}) \| p^{-i}_{gen}(\boldsymbol{x}|\boldsymbol{z})) \\ &- \mathrm{KL}(p_{true}(\boldsymbol{x}) \| p_{gen}(\boldsymbol{x}|\boldsymbol{z}))\big|\,, \end{aligned} \qquad (5)$$

where $p_{true}(\boldsymbol{x})$ is the limit set distribution across state space of the ground truth trajectories, $p_{gen}(\boldsymbol{x}|\boldsymbol{z})$ the corresponding full model-generated trajectory distribution, and $p^{-i}_{gen}(\boldsymbol{x}|\boldsymbol{z})$ the model generated distribution with parameter θ_i removed. In low-dimensional spaces this KL divergence can be approximated by simply discretizing (binning) the space, while in higher-dimensional spaces a Gaussian mixture model approximation is usually employed (see Appx. A.3 and Brenner et al. (2022) for details). As illustrated in Fig. 1, in geometric pruning I_{θ_i} naturally picks out those weight parameters which contribute the least to attractor geometry. Computing this measure is generally costly and pruning through this procedure may thus not always be feasible in practical settings. However, here it mainly served to study resulting network topologies, based on which initialization templates can be constructed (see sect. 4.4).

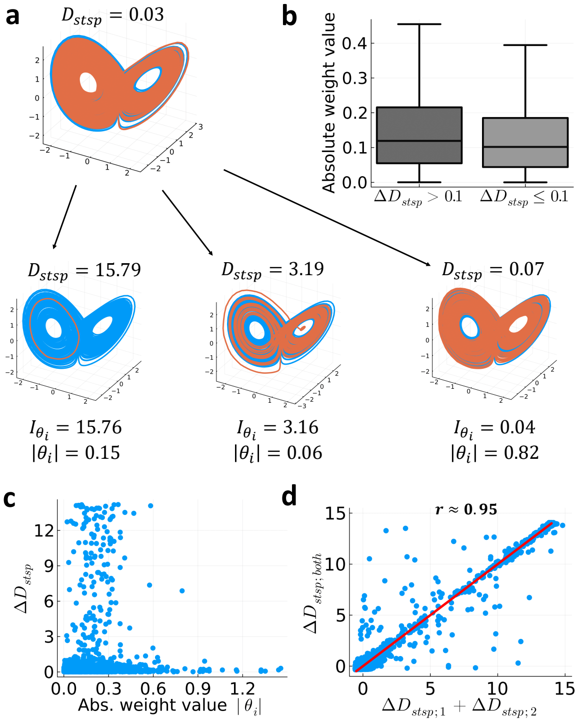


Figure 1. **a**) Illustration of geometry-based pruning. Top shows the (ground truth) iconic Lorenz-63 (Lorenz, 1963) chaotic attractor (blue) and an optimal PLRNN reconstruction (red), while below three reconstructions are shown with a single weight parameter removed with high (leftmost), medium (center) or low (rightmost) influence on attractor geometry. Measure for geometrical (dis)agreement (D_{stsp}) on top of each graph, and geometric importance score and magnitude of pruned parameter indicated below. **b**) Weight parameters with large ($\Delta D_{\text{stsp}} > 0.1$) vs. low ($\Delta D_{\text{stsp}} \leq 0.1$) impact on geometrical reconstruction quality do not substantially differ in absolute magnitude. **c**) Change in geometrical disagreement (ΔD_{stsp}) vs. weight magnitude for PLRNNs trained on the Lorenz-63. Note there is no discernible trend for larger weights to associate with stronger effects on attractor geometry. **d**) The effects of weight removal on ΔD_{stsp} are largely additive, with simultaneous removal of two weights having about the same effect as the sum of the individual weight effects.

3.3. Analysis of Network Topology

A graph $G = (V, E)$ consists of a set of nodes $V(G) = \{v_i\}$, $i \in 1, ..., n$, and a set of edges $E(G) = \{e_{ij}\}$ (or links) between nodes (Diestel, 2005). Network topology focuses on abstract structural properties of such graphs, as specified through the adjacency matrix $\boldsymbol{A}^{\text{adj}}$, which codes the existing links (and their direction) between any two nodes. In our case it is given by the pruning mask $\boldsymbol{A}^{\text{adj}} = \boldsymbol{m}$. The type of graph and its tendency to form highly connected hubs is determined by its degree distribution $P(k)$, where the degree k refers to the number of connections a node receives (Diestel, 2005). The two most important statistical quantities describing the properties of a graph, which we will use here, are, first, the mean average path length L defined as the minimum path length between two nodes averaged over all pairs of nodes (Watts & Strogatz, 1998):

$$L(G) = \frac{1}{n(n-1)} \sum_{i \neq j} d(v_i, v_j) \ , \tag{6}$$

where $d(v_i, v_j)$ is the geodesic distance between node v_i and node v_j in a graph with n nodes. Second, the *clustering* of nodes is calculated as (Fagiolo, 2007)

$$C(G) = \frac{1}{n} \sum_{i=1}^{n} \frac{(\boldsymbol{A}^{\text{adj}} + (\boldsymbol{A}^{\text{adj}})^T)^3_{ii}}{2T_i} \ , \tag{7}$$

where $\boldsymbol{A}^{\text{adj}}$ is the adjacency matrix of the graph. The clustering $C(G)$ gives the ratio between the number of all triangle connections (i.e., where two neighbors of a chosen node v_i are also directly connected) and the total theoretically possible number of triangles T_i. Details on these measures can be found in Appx. A.5.

Based on such a topological characterization of network graphs obtained through geometric pruning of trained PLRNNs, in sect. 4.4 we derive an algorithm that creates an adjacency matrix $\boldsymbol{A}^{\text{adj}}$ with the desired properties which can be used as a mask $\boldsymbol{m}$ in Eqn. 4. Fig. 2 illustrates the general approach. We compare its reconstruction performance with common network structures from graph theory, like the Erdős–Rényi model (Erdös & Rényi, 1959) generating random graphs, the Watts-Strogatz model (Watts & Strogatz, 1998) known for its small-world properties, or the Barabási-Albert model (Barabási & Albert, 1999; Albert & Barabási, 2002) producing central hub topologies.

4. Results

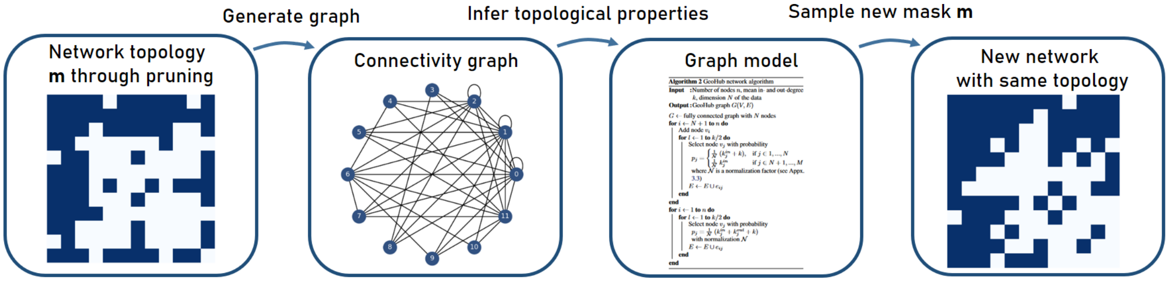
4.1. Performance Evaluation

We used well-established performance criteria to evaluate the DS reconstruction quality of trained networks (Koppe et al., 2019; Brenner et al., 2022; Hess et al., 2023). Because of exponential trajectory divergence in chaotic systems, mean-squared prediction errors are only of limited use as they may become large quickly even for well-trained systems (Wood, 2010; Mikhaeil et al., 2022).[3] Instead, we focus on the *geometrical* agreement between true and reconstructed attractors as quantified through a Kullback-Leibler divergence (D_{stsp}) as suggested in (Koppe et al. (2019); see also Appx. A.3), and on the long-term *temporal* agreement between true and reconstructed (generated) time series assessed by the average dimension-wise Hellinger distance (D_{H}) between true and reconstructed power spectra (equivalently, autocorrelation coefficients may be used; see Appx. A.3 for more details). Note that while D_{stsp} and D_{H} will be correlated in good reconstructions, they assess fundamentally different, complementary aspects of the dynamics.

4.2. Geometry-Based, but not Magnitude-Based, Pruning Allows for Substantial Reduction in Network Size

Figs. 1b & c show that there is hardly any (or at most very small) difference in the absolute magnitude of PLRNN connection weights contributing substantially vs. essentially non-contributing to geometric reconstruction quality (see Fig. A6b for a further example), regardless of whether several weights are removed individually or simultaneously (Fig. 1d, Fig. A6a). This raises the question of whether the parameter size of RNNs trained for DSR can be reduced beyond what would be expected by just random removal of connections. Using geometric pruning, however, we found that reductions by up to 95% for some systems are indeed possible (Fig. 3). More specifically, we evaluated DSR performance on several DS benchmarks for three different iterative pruning protocols (Algorithm 1; Fig. 3). First, the *Lorenz-63* model of atmospheric convection, proposed by Edward Lorenz (Lorenz, 1963), produces a chaotic attractor with iconic butterfly-wing structure (Fig. 1) and is probably the most commonly employed benchmark in this whole literature. Second, we use a simplified biophysical model of a *bursting neuron* (Durstewitz, 2009) which produces fast spiking (action potential) activity on top of a slow oscillation (see Fig. A3), thus featuring two widely different time scales. It has been previously used to evaluate DSR models (Brenner et al., 2022), and can also generate chaotic activity within some parameter regimes (Durstewitz, 2009), but was employed here as an example of a system with a complex (multi-period) limit cycle as in previous work. Third, as a real-world example, we used *human electrocardiogram (ECG)* data bearing signatures of chaos, with a positive maximum Lyapunov exponent (see Hess et al. (2023)). We also tested DSR on the *Rössler* attractor (Rössler, 1976), a simplification of the Lorenz-63 system, on the Lorenz-63

[3] Counterintuitively, as exemplified in Koppe et al. (2019), they can even become *larger* for perfect than for poorer reconstructions, because in the longer run only the mean may be well predictable for chaotic systems.



```
Algorithm 2 GeoHub network algorithm
Input   :Number of nodes $n$, mean in- and out-degree $k$, dimension $N$ of the data
Output :GeoHub graph $G(V, E)$
$G \leftarrow$ fully connected graph with $N$ nodes
for $i \leftarrow N + 1$ to $n$ do
    Add node $v_i$
    for $l \leftarrow 1$ to $k/2$ do
        Select node $v_j$ with probability
        $p_j = \begin{cases} \frac{1}{\mathcal{N}}\left(k_j^{in} + k\right), & \text{if } j \in 1, \dots, N \\ \frac{1}{\mathcal{N}}\, k_j^{in} & \text{if } j \in N+1, \dots, M \end{cases}$
        where $\mathcal{N}$ is a normalization factor (see Appx. 3.3)
        $E \leftarrow E \cup e_{ij}$
    end
end
for $i \leftarrow 1$ to $n$ do
    for $l \leftarrow 1$ to $k/2$ do
        Select node $v_j$ with probability
        $p_j = \frac{1}{\mathcal{N}}\left(k_j^{in} + k_j^{out} + k\right)$
        with normalization $\mathcal{N}$
        $E \leftarrow E \cup e_{ij}$
    end
end
```

Figure 2. Approach for translating graph-topological properties of trained networks into a general scheme to be used as topological prior.

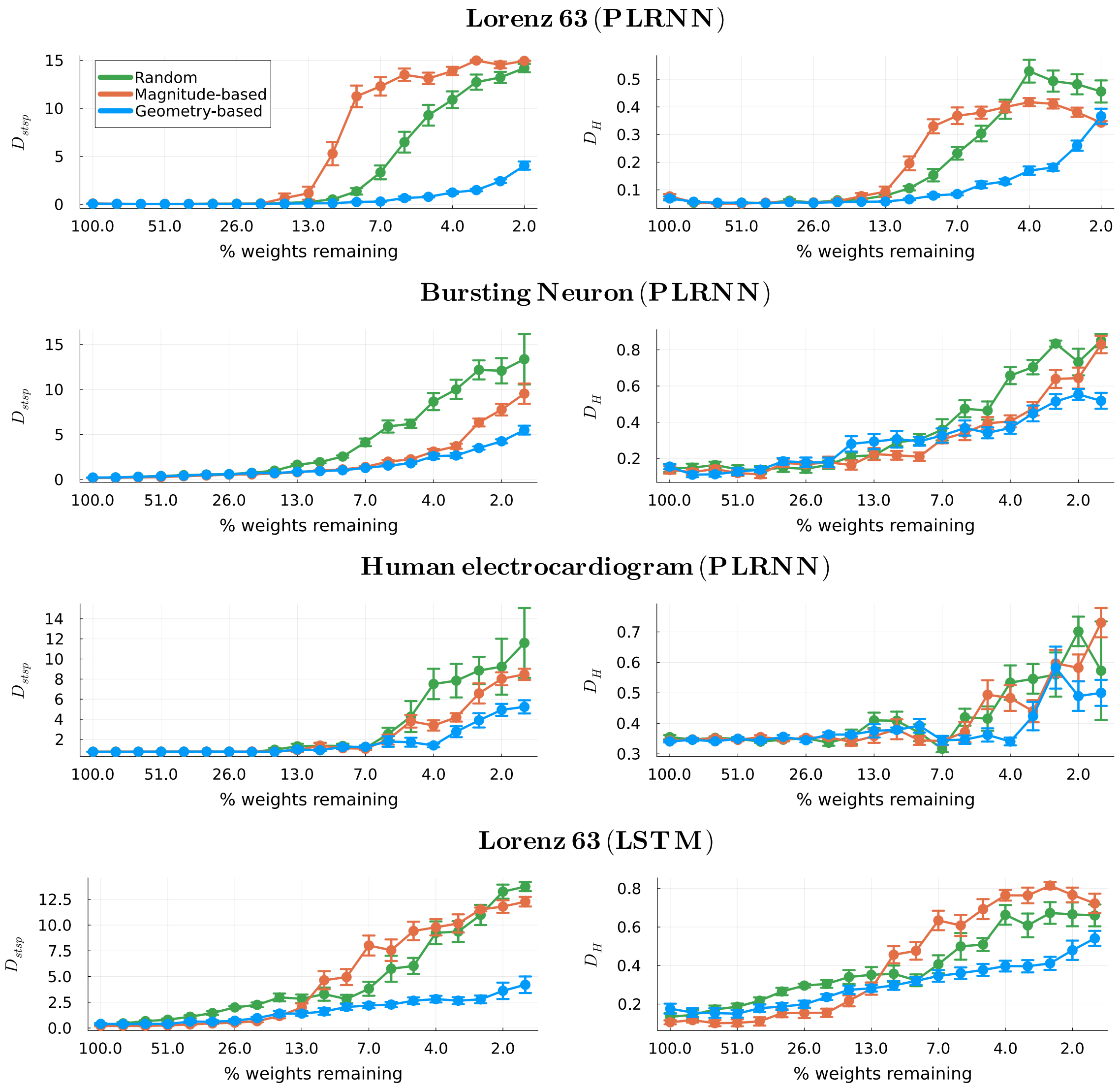


Figure 3. Quantification of DS reconstruction quality in terms of attractor geometry disagreement (D_{stsp}, left column) and disagreement in long-term temporal structure (D_{H}, right column) as a function of network pruning (x-axis, exponential scale) and different pruning criteria. Error bars = SEM.

system with high levels of observation noise (25%), and on the *Lorenz-96* model (Lorenz, 1996), a higher-dimensional spatial extension of the earlier Lorenz-63 model which also produces (highly) chaotic behavior for the parameter setup chosen here (we use a $5d$ system in our experiments, Fig. A2; see Appx. A.2 for details on all models, and Appx. A.1 for detailed hyper-parameter settings used in RNN training).

Fig. 3 illustrates DSR performance on the Lorenz-63, the bursting neuron, and the ECG benchmarks as a function of network size, i.e. percentage of pruned parameters, for magnitude- compared to geometry-based pruning (results for all other benchmarks are in Fig. A8). As a baseline, we also included a random pruning protocol, where parameters for removal were just chosen at random. In agreement with the observations in Fig. 1, we found that, on average, for all benchmarks, and for both the geometrical and temporal (dis)agreement measure, the effects of magnitude-based pruning essentially were not that much different than if weights were removed just randomly (statistically, by repeated measures ANOVAs, the differences were indeed insignificant in 5/10 comparisons across both D_{stsp} and D_{H}, and significant in the remaining 5/10 cases, but with random better than magnitude in one of these). This confirms that the absolute size of a weight parameter is less indicative of its contribution to RNN performance. Yet, using geometric pruning, network size in general could be reduced substantially beyond that of random pruning (significantly so in all 10/10 cases, $p < 0.03$), with sometimes just about 5% of the weight parameters sufficient to optimally reconstruct the ground truth DS. This was apparent not only in the geometrical measure D_{stsp} (left column in Fig. 3), but also in the temporal (dis)agreement measure D_{H} (right column in Fig. 3) which was not a criterion used in the pruning process (as well as in short-term prediction errors, see Fig. A10).

Furthermore, these effects were present across networks of different initial sizes (Fig. A7): Specifically, while the LTH suggests starting with strongly over-parameterized systems to enhance the chances for a winning ticket, which then naturally can be substantially pruned down (Frankle & Carbin, 2019; Frankle et al., 2020; Malach et al., 2020a), the differences between geometry- and magnitude-based pruning persisted in much smaller networks (Fig. A7). Finally, similar results were obtained for other types of RNN architectures (LSTMs: Fig. 3, bottom; vanilla RNNs: Fig. A9), implying that these observations are not specific to PLRNNs but more general. For LSTMs we furthermore observed that geometrical pruning identified the relevance of the model's different weight matrices to the performance, leading to interpretable results in terms of the inter-cell connectivity (Fig. A12). We conclude that a substantial reduction in parameter set size is indeed possible, but not so much based on the more traditional criterion of weight magnitude (Blalock et al., 2020).

4.3. Network Topology, not Weight Configuration is Essential to Performance

The LTH poses that it is the topology of an embedded subnetwork $\boldsymbol{m}$ in conjunction with a specific random initialization of model parameters $\boldsymbol{\theta}_0$ of this subnetwork which is crucial for its success. The fact that absolute weight magnitude plays less of a role in performance already sheds doubt on this idea in the context of DSR. To more explicitly test this and disentangle the contributions of mask $\boldsymbol{m}$ and weights $\boldsymbol{\theta}_0$ to the DSR of geometrically pruned networks, we re-sampled network parameters $\boldsymbol{\theta}_* \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$ with a fixed mask $\boldsymbol{m}$ from the very same distribution, from which the initial estimate $\boldsymbol{\theta}_0$ had been drawn, and compared this to the standard LTH case where $\boldsymbol{\theta}_0$ is fixed after the initial draw. We found that the influence of the network topology, given by the mask $\boldsymbol{m}$, far outweighed the importance of the specific initial weight vector $\boldsymbol{\theta}_0$: Redrawing $\boldsymbol{\theta}_*$ from scratch vs. fixing it to the initial $\boldsymbol{\theta}_0$ did not make much difference for DSR performance (Fig. 4; see also Fig. A11 for the same results on D_H), highlighting the crucial role network topology plays in the context of DSR. This is good news: in the 'classical' LTH, masks and weight distributions are tied in a specific way and therefore hard to disentangle. This in turn implies that the specific configuration that led to the winning ticket is difficult to reverse-engineer, and hence computationally costly iterative pruning schemes are required. However, given that in our case performance gains are primarily driven by topological structure and not parameter distribution, this structure can be distilled from trained RNNs and reverse-engineered with tools well-known from graph theory, as discussed next.

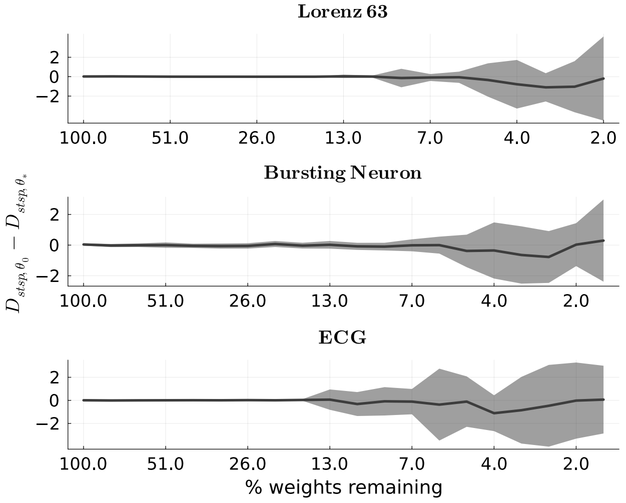


Figure 4. Difference in D_{stsp} when using the initial weights $\boldsymbol{\theta}_0$ and reinitialized weights $\boldsymbol{\theta}_*$ shows there is no strong or consistent influence of the specific weight initialization. Error bands = standard deviation.

4.4. Distilling Network Topology for Enhanced DSR Training

Next, we analyzed the topological properties of geometry-pruned networks in order to identify the crucial features that led to superior performance.[4] We find that these topologies contain both hub-type as well as small-world characteristics (Fig. 5; details in Appx. A.5, see Fig. A18 for specific examples of the different network topologies): As typical for small-world networks like the Watts-Strogatz model, geometrically pruned RNNs were characterized by a small average path length L (Fig. 5c; see Fig. A16 for similar effects in larger networks) as well as a high clustering coefficient C (Fig. 5d, Fig. A16). At the same time, as in scale-free networks like the Barabási-Albert model (Fig. A18), geometrically pruned RNNs bear a hub-like structure with a few highly connected network nodes (Fig. 5a, Fig. A16). We combined these features into an algorithm (Algorithm 2) that automatically produces RNN connectivity structures with these desired properties, which we call ‘GeoHub’ (for geometrically-pruned-hub network).

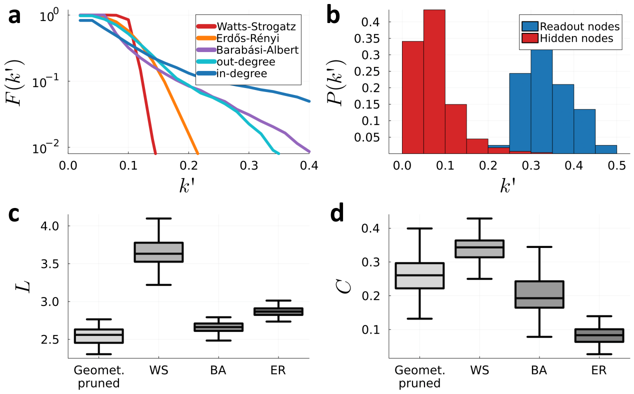


Figure 5. Graph properties of geometrically pruned, Barabási-Albert (BA), Watts-Strogatz (WS), and Erdős–Rényi (ER) networks, with 92.4% of parameters removed and averaged across all datasets with $M = 50$. **a**) Cumulative degree distribution $F(k')$ as a function of normalized degree $k' = \frac{k}{n-1}$, separated according to in- and out-degree (for geometrically pruned network). **b**) Comparison of degree distributions $P(k')$ for readout vs. hidden nodes of geometrically pruned networks. **c**) Average path lengths L for all four network topologies. Note that Erdős–Rényi graphs are not a naive baseline here, but are also known to have small path length (Watts & Strogatz, 1998). **d**) Clustering coefficients C for the same. See also Fig. A16.

In Fig. 6 we compare the DSR performance of RNNs trained with GeoHub topology to those based on the classical Watts-Strogatz and Barabási-Albert models. As evident, GeoHub-based networks perform best on all three benchmark setups employed in this comparison, whether chaotic (Lorenz-63), complex but non-chaotic (bursting neuron), or real-world ECG data, closely followed by RNNs with a Barabási-Albert graph (see Fig. A13 for comparisons on other benchmarks). As a further baseline, we also included RNNs based on an Erdős–Rényi random graph in this comparison. Surprisingly, although small-world features appeared to be necessary to move DSR performance beyond that obtained by the scale-free Barabási-Albert structure alone, a pure small-world structure (Watts-Strogatz model) actually appeared to *diminish* performance compared to the Erdős–Rényi graph models. Finally, we observed that RNNs initialized with optimal topology do not only outperform other graph structures, but also train significantly faster, i.e. reach satisfying DSR performance in fewer epochs than other topologies, as illustrated in Fig. 7.

Algorithm 2 GeoHub network algorithm

```
Input  : Number of nodes $n$, mean in- and out-degree
         $k$, dimension $N$ of the data
Output : GeoHub graph $G(V, E)$
$G \leftarrow$ fully connected graph with $N$ nodes
Add $n - N$ nodes
for $i \leftarrow 1$ to $n$ do
    for $l \leftarrow 1$ to $k/2$ do
        Select node $v_j$ with probability
          $p_j = \begin{cases} \frac{1}{\mathcal{N}}\,(k_j^{in} + k/2), & \text{if } j \in 1, ..., N \\ \frac{1}{\mathcal{N}}\,k_j^{in} & \text{if } j \in N+1, ..., n \end{cases}$
          where $\mathcal{N}$ is a normalization factor (see Appx. A.5)
        $E \leftarrow E \cup \{e_{ij}\}$
    end
end
for $i \leftarrow 1$ to $n$ do
    for $l \leftarrow 1$ to $k/2$ do
        Select node $v_j$ with probability
          $p_j = \frac{1}{\mathcal{N}}\,(k_j^{in} + k_j^{out} + k/4)$
          with normalization $\mathcal{N}$
        $E \leftarrow E \cup \{e_{ji}\}$
    end
end
```

In summary, our results show that the best performing graph model is the one which replicates the topology empirically obtained through geometry-based pruning, and that initializing based on this topology alone yields sparse networks with performance rivaling that of fully connected RNNs.

5. Conclusions

In this work we reported on a surprising observation: Setting up the *right network topology alone* is sufficient to produce

[4] Recall that well trained but unpruned RNNs always have full, all-to-all connectivity.

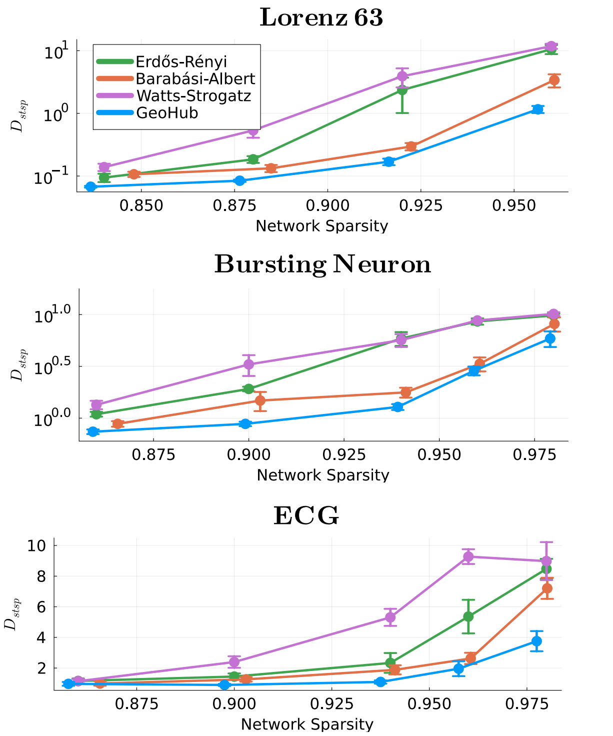


Figure 6. Reconstruction results in terms of state space divergence D_{stsp} as a function of network sparsity $s = 1 - \frac{|\boldsymbol{m}|}{|\boldsymbol{W}|}$. Erdős–Rényi, Barabási-Albert, and Watts-Strogatz graph algorithms are described in detail in A.6. Error bars = SEM.

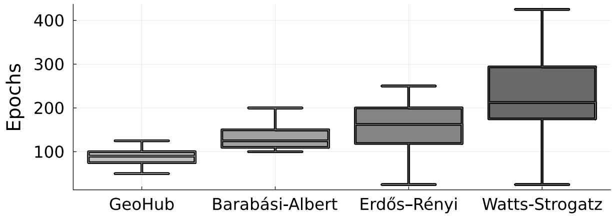


Figure 7. Epoch at which reasonable DSR performance ($D_{\text{stsp}} < 1.0$) is obtained for the different network topologies when trained on the Lorenz-63. Each network contains ≈ 300 parameters.

a highly sparsified generative RNN which recapitulates the geometrical and temporal properties of observed DS about as well as a fully parameterized network. Beyond proper network topology, neither the specific parameter initialization, nor their absolute magnitude, were that crucial to DSR performance. Such networks, as obtained by geometric pruning, with a mixture of hub-type and small-world characteristics, not only profoundly reduce parameter and memory load, but also train much faster than comparably sized RNNs with random topology. Since we observed this for a diverse set of benchmarks, both chaotic and non-chaotic, low- and higher-dimensional, evolving on multiple time scales, as well as for different types of RNNs, these results appear to be more general, although follow-up studies to confirm this for a wider range of systems and NN architectures are desirable.

Our results suggest a new type of LTH (Frankle & Carbin, 2019) in the context of DSR, where the winning tickets in largely over-parameterized networks depend less on specific weights but more on abstract topological features. It remains to be examined why exactly this is the case, how widely this holds, and whether the type of topology will be similar across different systems. For instance, the LTH so far has been mainly explored in the context of feed-forward architectures like CNNs. Is recurrence in the network a crucial feature and do our results therefore also generalize to other sequence and time series models? Or is it specific to the DSR problem, where we also want to capture geometric and long-term temporal properties of the data-generating system? Since many real-world systems (on which many of the benchmarks are based) have specific topological properties due to physical constraints, like scale-free or small-worldness, it is also conceivable that for DSR optimal RNN structure to some extent mirrors this empirical observation. It would be interesting to explore whether these results hold beyond the natural science domain, in areas like NLP for instance.

Finally, note that geometric pruning here mainly served as a tool to examine topological factors important in DSR. However, computationally efficient implementations of it, which make this technique directly applicable, are also conceivable, e.g. based on efficient (dimension-wise and parallelizable) proxies for D_{stsp}. There is also room for improvement, for instance by incorporating invariant temporal structure into the pruning process.

Software and Data

All code created is available at `https://github.com/DurstewitzLab/RNNtopoDSR`.

Acknowledgements

This work was funded by the German Research Foundation (DFG) through individual grant Du 354/15-1 to DD, and within Germany's Excellence Strategy EXC 2181/1 – 390900948 (STRUCTURES).

Impact Statement

This paper presents work whose goal is to advance the field of theoretical Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Albert, R. and Barabási, A.-L. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, 2002. doi: 10.1103/revmodphys.74.47. URL https://doi.org/10.1103%2Frevmodphys.74.47.

Albert, R. and Barabási, A.-L. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

Alvarez, V. M. M., Roşca, R., and Fălcuţescu, C. G. DyNODE: Neural Ordinary Differential Equations for Dynamics Modeling in Continuous Control, September 2020. URL http://arxiv.org/abs/2009.04278. arXiv:2009.04278 [cs, eess, stat].

Amaral, L. A. N., Scala, A., Barthelemy, M., and Stanley, H. E. Classes of small-world networks. *Proceedings of the national academy of sciences*, 97(21):11149–11152, 2000.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Barabási, A.-L. and Albert, R. Emergence of scaling in random networks. *Science*, 286(5439): 509–512, 1999. doi: 10.1126/science.286.5439.509. URL http://www.sciencemag.org/cgi/content/abstract/286/5439/509.

Bassett, D. S. and Bullmore, E. Small-world brain networks. *The neuroscientist*, 12(6):512–523, 2006.

Beggs, J. M. and Plenz, D. Neuronal avalanches in neocortical circuits. *The Journal of Neuroscience: The Official Journal of the Society for Neuroscience*, 23(35): 11167–11177, December 2003. ISSN 1529-2401. doi: 10.1523/JNEUROSCI.23-35-11167.2003.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019. ISSN 1091-6490. doi: 10.1073/pnas.1903070116. URL http://dx.doi.org/10.1073/pnas.1903070116.

Blalock, D., Gonzalez Ortiz, J. J., Frankle, J., and Guttag, J. What is the state of neural network pruning? *Proceedings of machine learning and systems*, 2:129–146, 2020.

Brenner, M., Hess, F., Mikhaeil, J. M., Bereska, L., Monfared, Z., Kuo, P.-C., and Durstewitz, D. Tractable dendritic rnns for reconstructing nonlinear dynamical systems. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 2292–2320. PMLR, 2022.

Brunton, S. L., Proctor, J. L., and Kutz, J. N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

Bullmore, E. and Sporns, O. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186–198, 2009.

Bullmore, E. and Sporns, O. The economy of brain network organization. *Nature reviews neuroscience*, 13(5):336–349, 2012.

Burkholz, R., Laha, N., Mukherjee, R., and Gotovos, A. On the Existence of Universal Lottery Tickets, March 2022. URL http://arxiv.org/abs/2111.11146. arXiv:2111.11146 [cs, stat].

Carroll, T. L. and Pecora, L. M. Network Structure Effects in Reservoir Computers. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(8):083130, August 2019. ISSN 1054-1500, 1089-7682. doi: 10.1063/1.5097686. URL http://arxiv.org/abs/1903.12487. arXiv:1903.12487 [nlin].

Champion, K., Lusch, B., Kutz, J. N., and Brunton, S. L. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, November 2019. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1906995116. URL http://arxiv.org/abs/1904.02107. arXiv:1904.02107 [stat].

Chatzikonstantinou, C., Konstantinidis, D., Dimitropoulos, K., and Daras, P. Recurrent neural network pruning using dynamical systems and iterative fine-tuning. *Neural Networks*, 143:475–488, 2021. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2021.07.001. URL https://www.sciencedirect.com/science/article/pii/S0893608021002641.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Dale, M., O'Keefe, S., Sebald, A., Stepney, S., and Trefzer, M. A. Reservoir computing quality: connectivity and topology. *Natural Computing*, 20(2):205–216, June 2021. ISSN 1572-9796. doi: 10.1007/s11047-020-09823-1. URL https://doi.org/10.1007/s11047-020-09823-1.

Diestel, R. *Graph Theory (Graduate Texts in Mathematics)*. Springer, 2005. ISBN 3540261826.

Durstewitz, D. Implications of synaptic biophysics for recurrent network dynamics and active memory. *Neural Networks*, 22(8):1189–1200, 2009. ISSN 0893-6080.

doi: https://doi.org/10.1016/j.neunet.2009.07.016. URL https://www.sciencedirect.com/science/article/pii/S0893608009001622. Cortical Microcircuits.

Durstewitz, D. A state space approach for piecewise-linear recurrent neural networks for identifying computational dynamics from neural measurements. *PLOS Computational Biology*, 13(6):1–33, 06 2017. doi: 10.1371/journal.pcbi.1005542. URL https://doi.org/10.1371/journal.pcbi.1005542.

Durstewitz, D., Koppe, G., and Thurm, M. I. Reconstructing computational system dynamics from neural data with recurrent neural networks. *Nature Reviews Neuroscience*, pp. 1–18, 2023.

Dutoit, X., Schrauwen, B., Van Campenhout, J., Stroobandt, D., Van Brussel, H., and Nuttin, M. Pruning and regularization in reservoir computing. *Neurocomputing*, 72(7):1534–1546, March 2009. ISSN 0925-2312. doi: 10.1016/j.neucom.2008.12.020. URL https://www.sciencedirect.com/science/article/pii/S0925231209000186.

Eguiluz, V. M., Chialvo, D. R., Cecchi, G. A., Baliki, M., and Apkarian, A. V. Scale-free brain functional networks. *Physical review letters*, 94(1):018102, 2005.

Eisenmann, L., Monfared, Z., Göring, N. A., and Durstewitz, D. Bifurcations and loss jumps in RNN training. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Emmert-Streib, F. Influence of the neural network topology on the learning dynamics. *Neurocomputing*, 69(10):1179–1182, June 2006. ISSN 0925-2312. doi: 10.1016/j.neucom.2005.12.070. URL https://www.sciencedirect.com/science/article/pii/S0925231205003966.

Erdös, P. and Rényi, A. On random graphs i. *Publicationes Mathematicae Debrecen*, 6:290, 1959.

Fagiolo, G. Clustering in complex directed networks. *Phys. Rev. E*, 76:026107, 2007. doi: 10.1103/PhysRevE.76.026107. URL https://link.aps.org/doi/10.1103/PhysRevE.76.026107.

Floyd, R. W. Algorithm 97: Shortest path. *Communications of the ACM*, 5:345, 1962. URL https://api.semanticscholar.org/CorpusID:2003382.

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJl-b3RcF7.

Frankle, J., Dziugaite, G. K., Roy, D., and Carbin, M. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pp. 3259–3269. PMLR, 2020.

Girish, S., Maiya, S. R., Gupta, K., Chen, H., Davis, L., and Shrivastava, A. The Lottery Ticket Hypothesis for Object Recognition. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 762–771, Nashville, TN, USA, June 2021. IEEE. ISBN 978-1-66544-509-2. doi: 10.1109/CVPR46437.2021.00082. URL https://ieeexplore.ieee.org/document/9578168/.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015a.

Han, S., Pool, J., Tran, J., and Dally, W. Learning both weights and connections for efficient neural network. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015b. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/ae0eb3eed39d2bcef4622b2499a05fe6-Paper.pdf.

Han, S., Pool, J., Tran, J., and Dally, W. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015c.

Han, X., Zhao, Y., and Small, M. A tighter generalization bound for reservoir computing. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 32(4):043115, April 2022. ISSN 1054-1500. doi: 10.1063/5.0082258. URL https://doi.org/10.1063/5.0082258.

Hassibi, B. and Stork, D. Second order derivatives for network pruning: Optimal brain surgeon. In Hanson, S., Cowan, J., and Giles, C. (eds.), *Advances in Neural Information Processing Systems*, volume 5. Morgan-Kaufmann, 1992. URL https://proceedings.neurips.cc/paper_files/paper/1992/file/303ed4c69846ab36c2904d3ba8573050-Paper.pdf.

He, Y. and Xiao, L. Structured pruning for deep convolutional neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2023. ISSN 1939-3539. doi: 10.1109/tpami.2023.3334614. URL http://dx.doi.org/10.1109/TPAMI.2023.3334614.

Hess, F., Monfared, Z., Brenner, M., and Durstewitz, D. Generalized teacher forcing for learning chaotic dynamics. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 13017–13049. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/hess23a.html.

Hochreiter, S. and Schmidhuber, J. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Jaeger, H. and Haas, H. Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. *Science*, 304 (5667):78–80, April 2004. doi: 10.1126/science.1091277. URL https://www.science.org/doi/10.1126/science.1091277. Publisher: American Association for the Advancement of Science.

Jiang, B. and Claramunt, C. Topological analysis of urban street networks. *Environment and Planning B: Planning and design*, 31(1):151–162, 2004.

Junior, L. O., Stelzer, F., and Zhao, L. Clustered Echo State Networks for Signal Observation and Frequency Filtering. In *Anais do Symposium on Knowledge Discovery, Mining and Learning (KDMiLe)*, pp. 25–32. SBC, October 2020. doi: 10.5753/kdmile.2020.11955. URL https://sol.sbc.org.br/index.php/kdmile/article/view/11955. ISSN: 2763-8944.

Karlsson, D. and Svanström, O. Modelling Dynamical Systems Using Neural Ordinary Differential Equations, 2019.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015. URL http://arxiv.org/abs/1412.6980.

Kleinberg, J. The small-world phenomenon: An algorithmic perspective. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pp. 163–170, 2000.

Ko, J.-H., Koh, H., Park, N., and Jhe, W. Homotopy-based training of neuralodes for accurate dynamics discovery. *Advances in Neural Information Processing Systems*, 36: 64725–64752, 2023.

Koppe, G., Toutounji, H., Kirsch, P., Lis, S., and Durstewitz, D. Identifying nonlinear dynamical systems via generative recurrent neural networks with applications to fmri. *PLOS Computational Biology*, 15(8):1–35, 2019. doi: 10.1371/journal.pcbi.1007263. URL https://doi.org/10.1371/journal.pcbi.1007263.

Kramer, D., Bommer, P. L., Tombolini, C., Koppe, G., and Durstewitz, D. Reconstructing nonlinear dynamical systems from multi-modal time series. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 11613–11633. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/kramer22a.html.

LeCun, Y., Denker, J., and Solla, S. Optimal brain damage. In Touretzky, D. (ed.), *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1989. URL https://proceedings.neurips.cc/paper_files/paper/1989/file/6c9882bbac1c7093bd25041881277658-Paper.pdf.

Li, F., Li, Y., and Wang, X. Echo State Network with Hub Property. In Deng, Z. (ed.), *Proceedings of 2019 Chinese Intelligent Automation Conference*, Lecture Notes in Electrical Engineering, pp. 537–544, Singapore, 2020. Springer. ISBN 978-981-329-050-1. doi: 10.1007/978-981-32-9050-1_61.

Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Han, J. On the variance of the adaptive learning rate and beyond. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkgz2aEKDr.

Liu, S., Mocanu, D. C., Pei, Y., and Pechenizkiy, M. Selfish sparse rnn training. In *International Conference on Machine Learning*, pp. 6893–6904. PMLR, 2021.

Liu, Z., Sun, M., Zhou, T., Huang, G., and Darrell, T. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270*, 2018.

Lorenz, E. N. Deterministic nonperiodic flow. *Journal of Atmospheric Sciences*, 20(2):130 – 141, 1963. doi: https://doi.org/10.1175/1520-0469(1963)020⟨0130:DNF⟩2.0.CO;2. URL https://journals.ametsoc.org/view/journals/atsc/20/2/1520-0469_1963_020_0130_dnf_2_0_co_2.xml.

Lorenz, E. N. Predictability: A problem partly solved. In *Proc. Seminar on predictability*, volume 1. Reading, 1996.

Malach, E., Yehudai, G., Shalev-Schwartz, S., and Shamir, O. Proving the lottery ticket hypothesis: Pruning is all you need. In *International Conference on Machine Learning*, pp. 6682–6691. PMLR, 2020a.

Malach, E., Yehudai, G., Shalev-Schwartz, S., and Shamir, O. Proving the Lottery Ticket Hypothesis: Pruning is All You Need. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 6682–6691. PMLR, November 2020b. URL https://proceedings.mlr.press/v119/malach20a.html. ISSN: 2640-3498.

Messenger, D. A. and Bortz, D. M. Weak SINDy: Galerkin-Based Data-Driven Model Selection. *Multiscale Modeling & Simulation*, 19(3):1474–1497, January 2021. ISSN 1540-3459. doi: 10.1137/20M1343166. URL https://epubs.siam.org/doi/10.1137/20M1343166. Publisher: Society for Industrial and Applied Mathematics.

Mikhaeil, J., Monfared, Z., and Durstewitz, D. On the difficulty of learning chaotic dynamics with rnns. *Advances in Neural Information Processing Systems*, 35: 11297–11312, 2022.

Monfared, Z. and Durstewitz, D. Transformation of ReLU-based recurrent neural networks from discrete-time to continuous-time. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6999–7009. PMLR, 2020. URL https://proceedings.mlr.press/v119/monfared20a.html.

Muldoon, S. F., Bridgeford, E. W., and Bassett, D. S. Small-world propensity and weighted brain networks. *Scientific reports*, 6(1):22057, 2016.

NEAL, Z. P. How small is it? comparing indices of small worldliness. *Network Science*, 5(1):30–44, 2017. doi: 10.1017/nws.2017.5.

Orseau, L., Hutter, M., and Rivasplata, O. Logarithmic Pruning is All You Need. In *Advances in Neural Information Processing Systems*, volume 33, pp. 2925–2934. Curran Associates, Inc., 2020.

Pajevic, S. and Plenz, D. The organization of strong links in complex networks. *Nature Physics*, 8(5):429–436, 2012.

Pathak, J., Hunt, B., Girvan, M., Lu, Z., and Ott, E. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Physical review letters*, 120(2):024102, 2018.

Platt, J. A., Wong, A., Clark, R., Penny, S. G., and Abarbanel, H. D. Robust forecasting using predictive generalized synchronization in reservoir computing. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(12), 2021.

Platt, J. A., Penny, S. G., Smith, T. A., Chen, T.-C., and Abarbanel, H. D. Constraining chaos: Enforcing dynamical invariants in the training of reservoir computers. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 33 (10), 2023.

Ravasz, E. and Barabási, A.-L. Hierarchical organization in complex networks. *Physical Review E*, 67(2): 026112, February 2003. doi: 10.1103/PhysRevE.67.026112. URL https://link.aps.org/doi/10.1103/PhysRevE.67.026112. Publisher: American Physical Society.

Reiss, A., Indlekofer, I., Schmidt, P., and Van Laerhoven, K. Deep ppg: Large-scale heart rate estimation with convolutional neural networks. *Sensors*, 19(14):3079, 2019.

Rubinov, M., Knock, S. A., Stam, C. J., Micheloyannis, S., Harris, A. W., Williams, L. M., and Breakspear, M. Small-world properties of nonlinear brain activity in schizophrenia. *Human brain mapping*, 30(2):403–416, 2009.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

Rössler, O. An equation for continuous chaos. *Physics Letters A*, 57(5):397–398, 1976. ISSN 0375-9601. doi: https://doi.org/10.1016/0375-9601(76)90101-8. URL https://www.sciencedirect.com/science/article/pii/0375960176901018.

Shandilya, S. G. and Timme, M. Inferring network topology from complex dynamics. *New Journal of Physics*, 13(1): 013004, January 2011. ISSN 1367-2630. doi: 10.1088/1367-2630/13/1/013004. URL https://dx.doi.org/10.1088/1367-2630/13/1/013004.

Sreenivasan, K., Sohn, J.-y., Yang, L., Grinde, M., Nagle, A., Wang, H., Xing, E., Lee, K., and Papailiopoulos, D. Rare gems: Finding lottery tickets at initialization. *Advances in neural information processing systems*, 35: 14529–14540, 2022.

Stoer, J. and Bulirsch, R. *Introduction to numerical analysis*. Texts in applied mathematics. Springer, 2002. ISBN 9780387954523.

Talathi, S. S. and Vartak, A. Improving performance of recurrent neural network with relu nonlinearity. In *Proceedings of the 4th International Conference on Learning Representations*, 2016. URL http://arxiv.org/abs/1511.03771.

Tziperman, E., Scher, H., Zebiak, S. E., and Cane, M. A. Controlling spatiotemporal chaos in a realistic el niño prediction model. *Phys. Rev. Lett.*, 79: 1034–1037, Aug 1997. doi: 10.1103/PhysRevLett.79.1034. URL https://link.aps.org/doi/10.1103/PhysRevLett.79.1034.

van den Heuvel, M. P., Stam, C. J., Boersma, M., and Pol, H. H. Small-world and scale-free organization of voxel-based resting-state functional connectivity in the human brain. *Neuroimage*, 43(3):528–539, 2008.

Vlachas, P. R., Byeon, W., Wan, Z. Y., Sapsis, T. P., and Koumoutsakos, P. Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2213): 20170844, 2018.

Voss, H. U., Timmer, J., and Kurths, J. Nonlinear dynamical system identification from uncertain and indirect measurements. *International Journal of Bifurcation and Chaos*, 14(06):1905–1933, June 2004. ISSN 0218-1274. doi: 10.1142/S0218127404010345. URL https://www.worldscientific.com/doi/abs/10.1142/S0218127404010345. Publisher: World Scientific Publishing Co.

Wang, Y., Wu, X., Feng, H., Lu, J., and Lü, J. Topology inference of uncertain complex dynamical networks and its applications in hidden nodes detection. *Science China Technological Sciences*, 59(8):1232–1243, August 2016. ISSN 1869-1900. doi: 10.1007/s11431-016-6050-1. URL https://doi.org/10.1007/s11431-016-6050-1.

Warshall, S. A theorem on boolean matrices. *J. ACM*, 9:11–12, 1962. URL https://api.semanticscholar.org/CorpusID:33763989.

Watts, D. J. and Strogatz, S. H. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998. doi: 10.1038/30918.

Williams, R. J. and Zipser, D. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

Wood, S. N. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466(7310):1102–1104, August 2010. ISSN 1476-4687. doi: 10.1038/nature09319. URL https://www.nature.com/articles/nature09319. Number: 7310 Publisher: Nature Publishing Group.

Yin, J. and Meng, Y. Self-organizing reservior computing with dynamically regulated cortical neural networks. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, June 2012. doi: 10.1109/IJCNN.2012.6252772. URL https://ieeexplore.ieee.org/document/6252772. ISSN: 2161-4407.

You, J., Leskovec, J., He, K., and Xie, S. Graph structure of neural networks. In *International Conference on Machine Learning*, pp. 10881–10891. PMLR, 2020.

Yu, H., Edunov, S., Tian, Y., and Morcos, A. S. Playing the lottery with rewards and multiple languages: lottery tickets in rl and nlp. *arXiv preprint arXiv:1906.02768*, 2019.

Zhang, H., Liu, D., and Wang, Z. *Controlling chaos: suppression, synchronization and chaotification*. Springer Science & Business Media, 2009a.

Zhang, S., Wang, M., Liu, S., Chen, P.-Y., and Xiong, J. Why Lottery Ticket Wins? A Theoretical Perspective of Sample Complexity on Sparse Neural Networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 2707–2720. Curran Associates, Inc., 2021.

Zhang, Z., Lin, Y., Gao, S., Zhou, S., Guan, J., and Li, M. Trapping in scale-free networks with hierarchical organization of modularity. *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, 80(5 Pt 1):051120, November 2009b. ISSN 1550-2376. doi: 10.1103/PhysRevE.80.051120.

A. Appendix

A.1. Methodological Details

A.1.1. Mean-centered PLRNN

Layer normalization is often beneficial for training RNNs (Ba et al., 2016) and has been modified for PLRNNs in order to retain their piecewise linear structure (Brenner et al., 2022). Brenner et al. (2022) observed that mean-centering before applying the nonlinearity at each time step is already sufficient to obtain the usual performance boosts, implemented as

$$\mathcal{M}(\boldsymbol{z}_{t-1}) = \boldsymbol{z}_{t-1} - \mu_{t-1} = \boldsymbol{z}_{t-1} - \mathbf{1}\frac{1}{M}\sum_{i=1}^{M} \boldsymbol{z}_{t-1,i} \ , \tag{8}$$

where $\mathbf{1} \in \mathbb{R}^M$ is a vector of ones. Since mean-centering is linear, it can be rewritten as a matrix multiplication with the latent state vector,

$$\mathcal{M}(\boldsymbol{z}_{t-1}) = \boldsymbol{M}\boldsymbol{z}_{t-1} = \frac{1}{M}\begin{pmatrix} M-1 & -1 & \dots & -1 \\ -1 & M-1 & \dots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \dots & M-1 \end{pmatrix} \boldsymbol{z}_{t-1} \ . \tag{9}$$

A.1.2. BPTT + identity-TF

Training RNNs via BPTT runs into the exploding & vanishing gradient problem, which is aggravated when training on chaotic systems (Mikhaeil et al., 2022). Mikhaeil et al. (2022); Brenner et al. (2022) suggested STF as a remedy, which in the case of a direct (identity) mapping from a subset of latent states to the observed time series values takes a particularly simple form. Let $\boldsymbol{X} = \{\boldsymbol{x}_t\}$ be the observed time series, and $\boldsymbol{Z} = \{\boldsymbol{z}_t\}$ the RNN latent states. We then create a control series $\tilde{\boldsymbol{Z}}$ by inverting the observation model, which in identity-TF simply comes down to

$$\tilde{z}_{t,k} = \begin{cases} x_{t,k}, & \text{for } k \leq N \\ z_{t,k}, & \text{for } k > N \end{cases} , \tag{10}$$

i.e. the $\tilde{\boldsymbol{z}}_t$ are just the latent states with the first N components replaced by the actual observations. In STF, the latent states $\boldsymbol{z}_t$ are replaced by the control states $\tilde{\boldsymbol{z}}_t$ sparsely at times $\mathcal{T} = \{n\tau + 1\}$, $n \in \mathbb{N}$, separated by an interval τ:

$$\boldsymbol{z}_t = \begin{cases} PLRNN(\tilde{\boldsymbol{z}}_{t-1}), & \text{if } t \in \mathcal{T} \\ PLRNN(\boldsymbol{z}_{t-1}), & \text{else} \end{cases} , \tag{11}$$

where this forcing is always applied *after* calculating the loss. To allow the system to capture relevant time scales while avoiding divergence, ideally the forcing interval τ is chosen according to the predictability time based on the system's maximum Lyapunov exponent (Mikhaeil et al., 2022), but here we simply determined the optimal τ by grid search as in (Brenner et al., 2022). Importantly, STF is only applied during model training and not at test time.

A.1.3. Training Protocol

Given a time series $\{\boldsymbol{x}_{1:T}\}$ from a DS, we train the model using BPTT + identity-STF. For each training epoch we sample several subsequences of length $\tilde{T}$, $\tilde{x}^{(p)}_{1:\tilde{T}} = x_{t_p:t_p+\tilde{T}}$ where $t_p \in [1, T - \tilde{T}]$ is chosen randomly. These subsequences $\left\{\tilde{\boldsymbol{x}}^{(p)}_{1:\tilde{T}}\right\}_{p=1}^{S}$ are then arranged into a batch of size S. On each sequence, the PLRNN is initialized with the first forcing signal $\tilde{\boldsymbol{z}}_1^{(p)}$, and from there forward-iterated in time, yielding predictions $\left\{\hat{\tilde{\boldsymbol{x}}}^{(p)}_{2:\tilde{T}}\right\} = \left\{\boldsymbol{z}^{(p)}_{2:\tilde{T},1:N}\right\}$ using Eqn. 1. The loss is then computed as the MSE between predicted and ground truth time series

$$\mathcal{L}_{\text{MSE}}\left(\left\{\tilde{\boldsymbol{x}}^{(p)}_{2:\tilde{T}}\right\}, \left\{\hat{\tilde{\boldsymbol{x}}}^{(p)}_{2:\tilde{T}}\right\}\right) = \frac{1}{S(\tilde{T}-1)}\sum_{p=1}^{S}\sum_{t=2}^{\tilde{T}} \left\|\tilde{\boldsymbol{x}}^{(p)}_t - \hat{\tilde{\boldsymbol{x}}}^{(p)}_t\right\|_2^2 \ . \tag{12}$$

We took rectified adaptive moment estimation (RADAM) (Liu et al., 2020) as the optimizer, using $L = 50$ batches of size $S = 16$ in each epoch. We chose $M = \{50, 100, 100, 50, 100\}$, $\tau = \{16, 10, 5, 8, 8\}$, $T = \{200, 50, 50, 300, 200\}$,

$\eta_{\text{start}} = \{10^{-2}, 10^{-3}, 10^{-3}, 5 \cdot 10^{-3}, 5 \cdot 10^{-3}\}$, and $epochs = \{2000, 3000, 4000, 3000, 3000\}$ for the $\{$Lorenz-63, ECG, Bursting Neuron, Rössler, Lorenz-96$\}$, respectively, and $\eta_{\text{end}} = 10^{-5}$ for all settings. Parameters in $\boldsymbol{W}$ were initialized using a Gaussian initialization with $\sigma = 0.01$, $\boldsymbol{h}$ simply as a vector of zeros, and $\boldsymbol{A}$ as the diagonal of a normalized positive-definite random matrix (Brenner et al., 2022; Talathi & Vartak, 2016). Across all training epochs of a given run, we consistently (for all comparisons and protocols) selected the model with the lowest D_{stsp}. Failed trainings (yielding NaN entries) were discarded. Failures in training were indeed much more common for the random and magnitude-based pruning protocols, further reinforcing our points about the importance of network graph topology.

The vanilla RNN used is given by

$$\boldsymbol{z}_t = \phi(\boldsymbol{W}\boldsymbol{z}_{t-1} + \boldsymbol{C}\boldsymbol{x}_t + \boldsymbol{b}) \ . \tag{13}$$

The LSTM architecture is defined by (Hochreiter & Schmidhuber, 1997)

$$\begin{aligned}
\boldsymbol{f}_t &= \sigma(\boldsymbol{W}_f\boldsymbol{z}_{t-1} + \boldsymbol{C}_f\boldsymbol{x}_{t-1} + \boldsymbol{b}_f) \ , \\
\boldsymbol{i}_t &= \sigma(\boldsymbol{W}_i\boldsymbol{z}_{t-1} + \boldsymbol{C}_i\boldsymbol{x}_{t-1} + \boldsymbol{b}_i) \ , \\
\boldsymbol{o}_t &= \sigma(\boldsymbol{W}_o\boldsymbol{z}_{t-1} + \boldsymbol{C}_o\boldsymbol{x}_{t-1} + \boldsymbol{b}_o) \ , \\
\tilde{\boldsymbol{c}}_t &= \tanh(\boldsymbol{W}_c\boldsymbol{z}_{t-1} + \boldsymbol{C}_c\boldsymbol{x}_{t-1} + \boldsymbol{b}_c) \ , \\
\boldsymbol{c}_t &= \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \tilde{\boldsymbol{c}}_t \ , \\
\boldsymbol{z}_t &= \boldsymbol{o}_t \odot \tanh(\boldsymbol{c}_t) \ ,
\end{aligned} \tag{14}$$

where $\sigma(\cdot)$ is the standard sigmoid function. For both these architectures, a linear observation model

$$\boldsymbol{x}_t = \boldsymbol{B}\boldsymbol{z}_t + \boldsymbol{h} \tag{15}$$

was used. Models were trained via standard BPTT (Rumelhart et al., 1986) with MSE loss, using the ADAM optimizer (Kingma & Ba, 2015). For both architectures batch size was $S = 16$, with $L = 20$ batches per epoch, $M = 50$, $T = 200$, $\eta_{\text{start}} = 10^{-3}$, $\eta_{\text{end}} = 10^{-6}$, and $epochs = 500$. All weight parameters were initialized using a Gaussian scheme with $\sigma = 0.01$, and biases as vectors of zeros. The $\boldsymbol{B}$ matrix of the observation model was initialized using glorot-uniform initialization (Glorot & Bengio, 2010). Like for the PLRNN, pruning was applied to all weight matrices $\boldsymbol{W}$, $\boldsymbol{C}$ (for vanilla RNN), and $\boldsymbol{W}_f$, $\boldsymbol{W}_i$, $\boldsymbol{W}_o$, $\boldsymbol{W}_c$, $\boldsymbol{C}_f$, $\boldsymbol{C}_i$, $\boldsymbol{C}_o$, $\boldsymbol{C}_c$ (for LSTM).

A.2. Details on the Benchmark Datasets

From all benchmark systems, as detailed below, trajectories of 10^5 time steps were drawn for training, all dimensions were individually standardized, and Gaussian observation noise was added (Lorenz-63: 5%, Lorenz-96: 1%, Bursting Neuron: 2%, Rössler: 5%, ECG: 5%). All systems (except for the human ECG data) were numerically integrated using a fourth-order Runge-Kutta scheme (Stoer & Bulirsch, 2002).

A.2.1. Lorenz-63 System

The Lorenz-63 system, introduced by Edward Lorenz in 1963 (Lorenz, 1963) as a model of atmospheric convection, is given by

$$\begin{aligned}\frac{\mathrm{d}x}{\mathrm{d}t} &= \sigma(y-x) \\ \frac{\mathrm{d}y}{\mathrm{d}t} &= x(\rho-z)-y \\ \frac{\mathrm{d}z}{\mathrm{d}t} &= xy-\beta z,\end{aligned} \tag{16}$$

where σ, ρ, β, are parameters that control the behavior of the system (set to $\sigma = 10$, $\beta = \frac{8}{3}$, and $\rho = 28$ here, within the chaotic regime). The Lorenz-63 is one of the most popular examples in chaos theory and in the literature on dynamical systems reconstruction. Here we solved this system with integration time step $\Delta t = 0.01$. See Fig. A1 for an illustration.

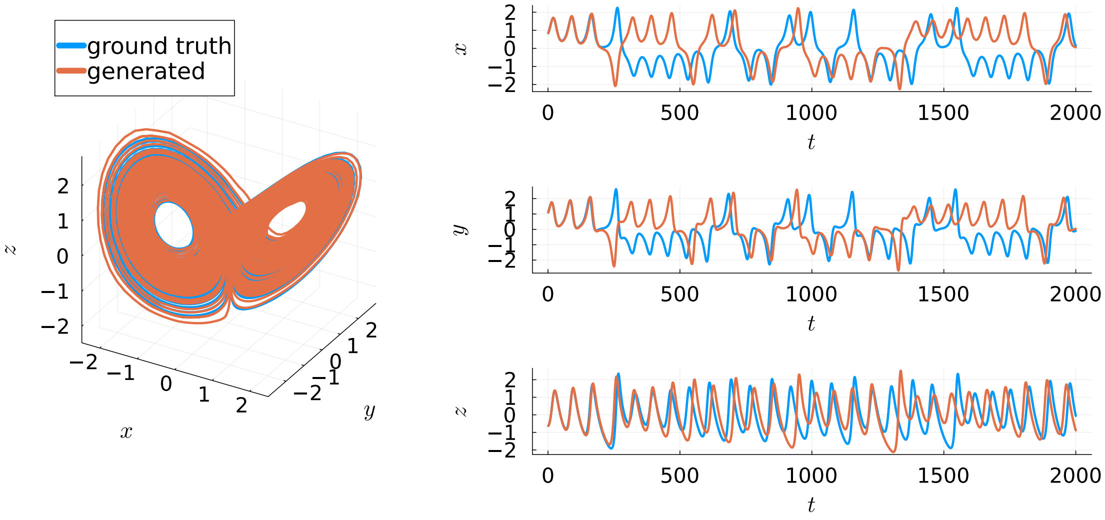


Figure A1. State space (left) and time graphs (right) for Lorenz-63 system with $\sigma = 10$, $\beta = \frac{8}{3}$ and $\rho = 28$ (blue), and a PLRNN reconstruction (red). Note that despite the essentially perfect reconstruction in state space, the Lorenz system's positive Lyapunov exponent causes the true and reconstructed trajectories to eventually diverge (yet their temporal structure remains the same).

A.2.2. Lorenz-96 System

The spatially extended Lorenz-96 system (Lorenz, 1996) is defined by

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = (x_{i+1} - x_{i-2})x_{i-1} - x_i + F, \tag{17}$$

with system variables x_i, $i = 1, ..., N$, and forcing term F (where $F = 8$ puts the system into the chaotic regime). Furthermore, cyclic boundary conditions are assumed with $x_{-1} = x_{N-1}$, $x_0 = x_N$, $x_{N+1} = x_1$, and $\Delta t = 0.04$ is used for numerical integration. Fig. A2 provides an illustration.

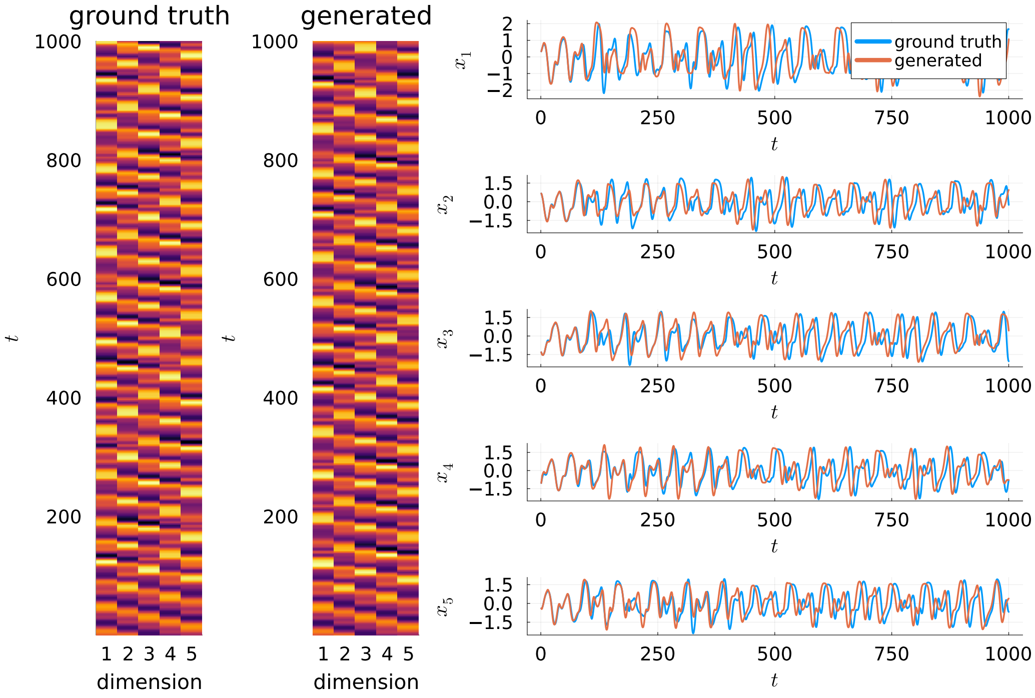


Figure A2. Illustration of the Lorenz-96 system ($N = 5$, $F = 8$), with spatiotemporal evolution on the left and single time series on the right.

A.2.3. BURSTING NEURON MODEL

The simplified 3d biophysical model of a neuron used here is defined by (Durstewitz, 2009)

$$\begin{aligned} -C_m \frac{\mathrm{d}V}{\mathrm{d}t} &= g_L(V - E_L) + g_{N_a} m_\infty(V)(V - E_{N_a}) + g_K n(V - E_K) \\ &+ g_M h(V - E_K) + g_{NMDA}(1 + 0.33e^{-0.0625V})^{-1}(V - E_{NMDA}) \\ \frac{\mathrm{d}h}{\mathrm{d}t} &= \frac{h_\infty(V) - h}{\tau_h} \\ \frac{\mathrm{d}n}{\mathrm{d}t} &= \frac{n_\infty(V) - n}{\tau_n}, \end{aligned} \tag{18}$$

where V is the membrane voltage and n, h, are so-called gating variables controlling current flux through voltage-gated ion channels with vastly different time constants. Specifically, we used a standard set for the parameters which produces bursting activity (fast spikes riding on top of slow oscillations, see Fig. A3):

$$\begin{gathered} C_m = 6\mu F,\ g_L = 8mS,\ E_L = -80mV,\ g_{Na} = 20mS \\ E_{Na} = 60mV,\ V_{hNa} = -20mV,\ k_{Na} = 15,\ g_K = 10mS, \\ E_K = -90mV,\ V_{hK} = -25mV,\ k_K = 5,\ \tau_n = 1ms,\ g_M = 25mS \\ V_{hM} = -15mV,\ k_M = 5,\ \tau_h = 200ms, g_{NMDA} = 10.2mS \end{gathered} \tag{19}$$

The limit values of the ionic gating variables are given by

$$\{m_\infty, n_\infty, h_\infty\} = \sigma\left(\frac{V - \{V_{hNa}, V_{hK}, V_{hM}\}}{\{k_{Na}, k_K, k_M\}}\right), \tag{20}$$

where $\sigma(\cdot)$ is the common sigmoid function.

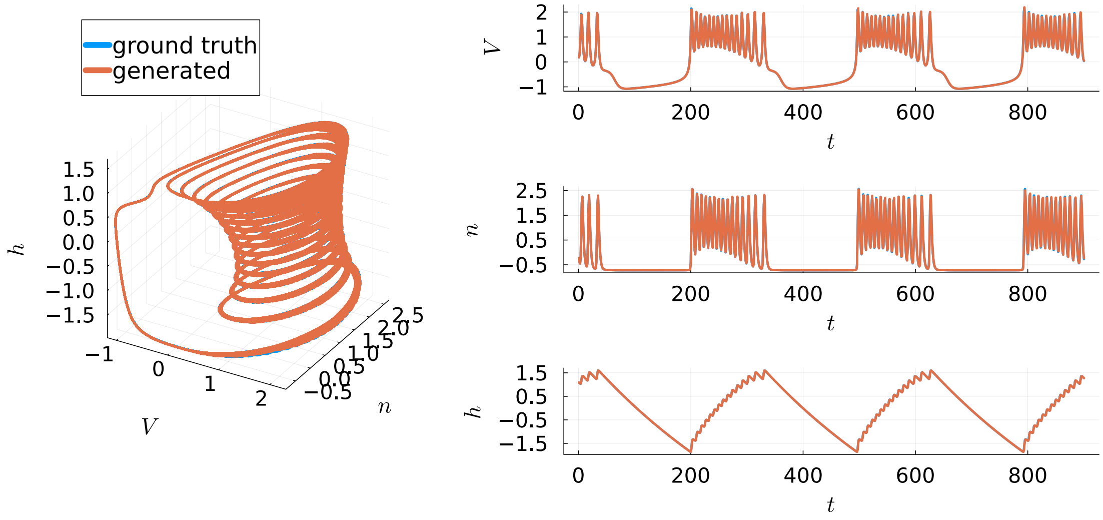


Figure A3. Illustration of the biophysical neuron activity (blue; Durstewitz (2009)) and a reconstruction using the PLRNN (red). Left: state space; right: time graphs. As this system is non-chaotic, true and reconstructed trajectories precisely overlap.

A.2.4. Rössler System

The Rössler system, introduced by Otto Rössler in 1976 (Rössler, 1976) is a model that produces chaotic dynamics with nonlinearity in only one state variable, given by

$$\begin{aligned}\frac{\mathrm{d}x}{\mathrm{d}t} &= -y - z \\ \frac{\mathrm{d}y}{\mathrm{d}t} &= x + ay \\ \frac{\mathrm{d}z}{\mathrm{d}t} &= b + z(x - c),\end{aligned} \tag{21}$$

where a, b, c, are parameters that control the behavior of the system (set to $a = 0.2$, $b = 0.2$, and $c = 5.7$ here, within the chaotic regime). Here we solved this system with integration time step $\Delta t = 0.08$. See Fig. A4 for an illustration.

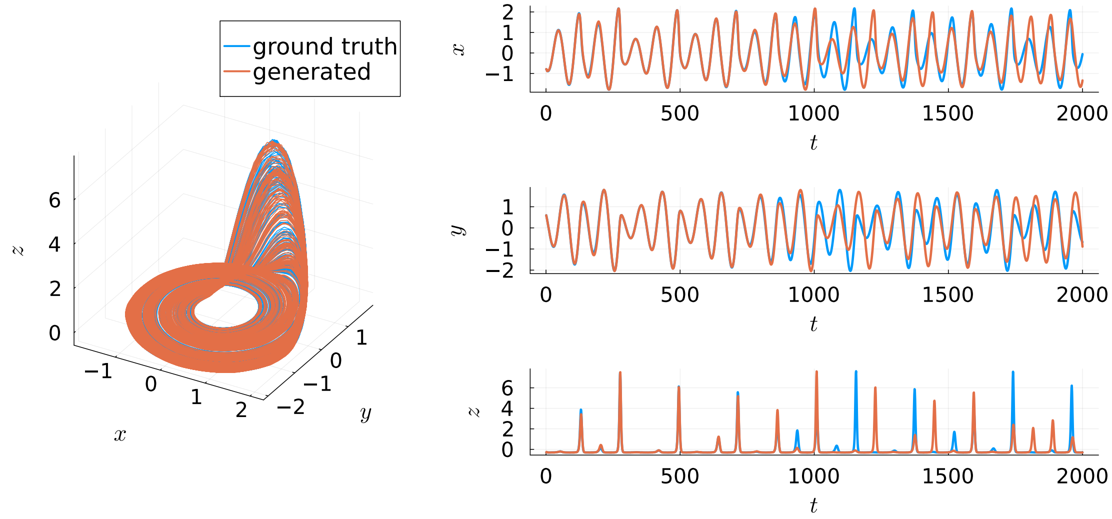


Figure A4. State space (left) and time graphs (right) for Rössler system with $a = 0.2$, $b = 0.2$, and $c = 5.7$ (blue), and a PLRNN reconstruction (red). Note that despite the essentially perfect reconstruction in state space, the Rössler system's positive Lyapunov exponent causes the true and reconstructed trajectories to eventually diverge (yet their temporal structure remains the same).

A.2.5. Electrocardiogram (ECG) Data

Electrocardiogram (ECG) time series were taken from the PPG-DaLiAdataset (Reiss et al., 2019). Using a sampling frequency of $700Hz$, this translates to a recording duration of 600 seconds resulting in a time series spanning $T = 419,973$ time points. The data were initially smoothed by applying a Gaussian filter (with $\sigma = 6$, $l = 8\sigma + 1 = 49$). The time series is then standardized, followed by a delay embedding, using the `DynamicalSystems.jl` Julia library with embedding dimension $m = 5$. In our experiments we use the first $T = 100,000$ samples (approximately 143 seconds).

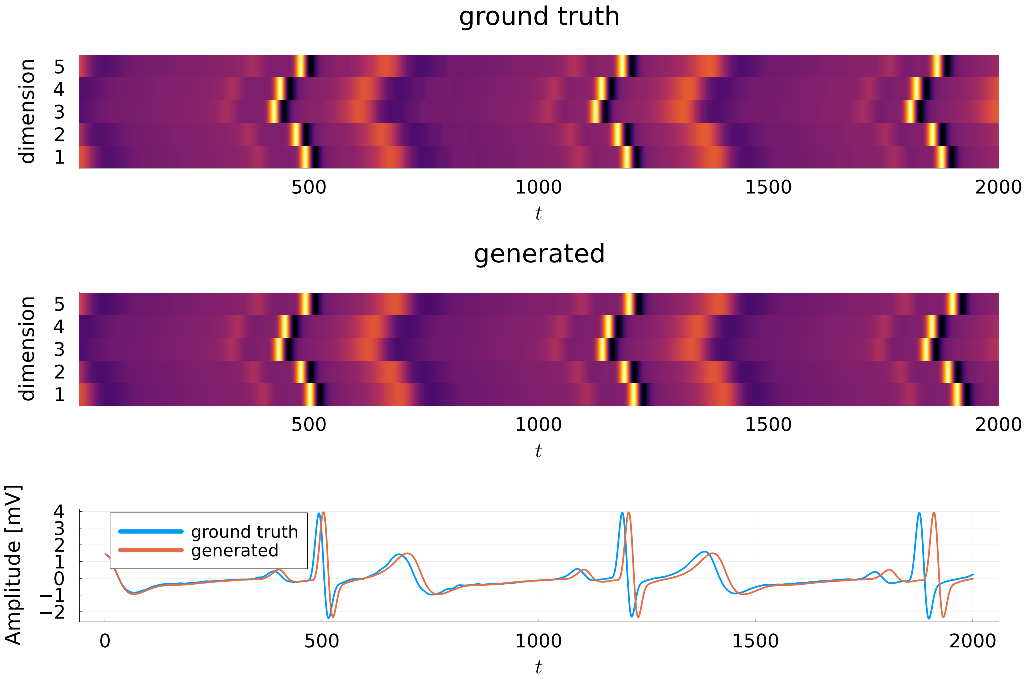


Figure A5. Illustration of a DS reconstruction of human ECG, with spatiotemporal evolution of delay-embedded time series (top panels) and single voltage signal (bottom). Note that the ECG signal is slightly chaotic (Hess et al., 2023), ultimately leading to divergence.

A.3. Evaluation Measures

To measure the geometrical (dis)agreement D_{stsp} between true and model-generated attractors we use a *Kullback-Leibler divergence*, as first suggested in (Koppe et al., 2019). It assesses the overlap between the true distribution $p_{true}(\boldsymbol{x})$ of trajectory points, and the distribution generated by the model $p_{gen}(\boldsymbol{x}|\boldsymbol{z})$, as

$$\text{KL}(p_{true}(\boldsymbol{x}) \| p_{gen}(\boldsymbol{x}|\boldsymbol{z})) = \int p_{true}(\boldsymbol{x}) \log \frac{p_{true}(\boldsymbol{x})}{p_{gen}(\boldsymbol{x}|\boldsymbol{z})} d\boldsymbol{x}. \tag{22}$$

Practically, this is evaluated by binning space into k^N bins, with $k = 30$ bins per dimension for $N = 3$ and $k = 8$ for $N = 5$, estimating the occupation probabilities through the relative frequencies

$$p_i = \frac{n_i}{T}, \tag{23}$$

where n_i is the number of time points falling into bin i, and taking

$$D_{\text{stsp}} \approx \sum_{i=1}^{k^N} p_{true;i} \log \frac{p_{true;i}}{p_{gen;i}}. \tag{24}$$

To assess the agreement in long-term temporal structure between true and reconstructed systems, the *Hellinger distance* D_{H} between power spectra $f_i(\omega)$ and $g_i(\omega)$ of the true and generated time series, respectively, are computed separately for each dimension i (Mikhaeil et al., 2022; Hess et al., 2023). It is defined as

$$H(f_i(\omega), g_i(\omega)) = \sqrt{1 - \int_{-\infty}^{\infty} \sqrt{f_i(\omega) g_i(\omega)}\, d\omega}\,. \tag{25}$$

Power spectra are computed through the Fast Fourier Transform, slightly smoothed using a Gaussian kernel, and normalized (see Hess et al. (2023) for details). The total measure D_{H} is then defined as the average across all dimension-wise distances $H(f_i(\omega), g_i(\omega))$.

A.4. Further Results

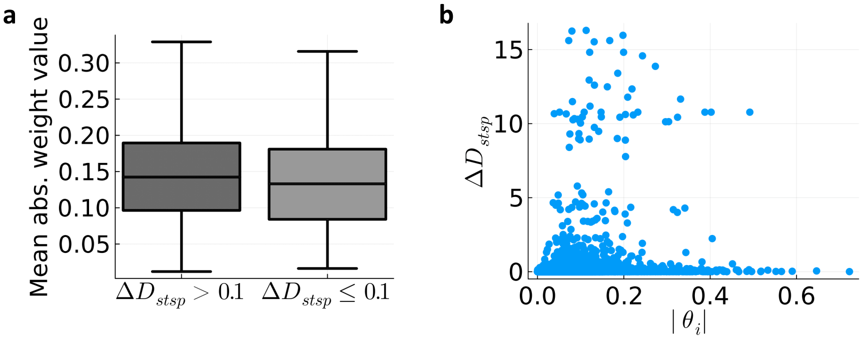
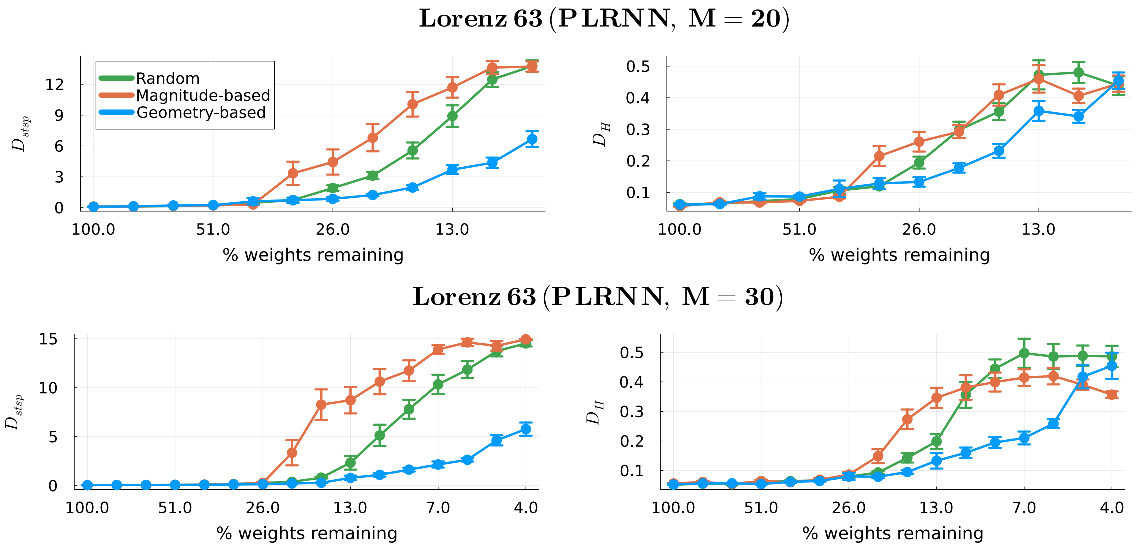
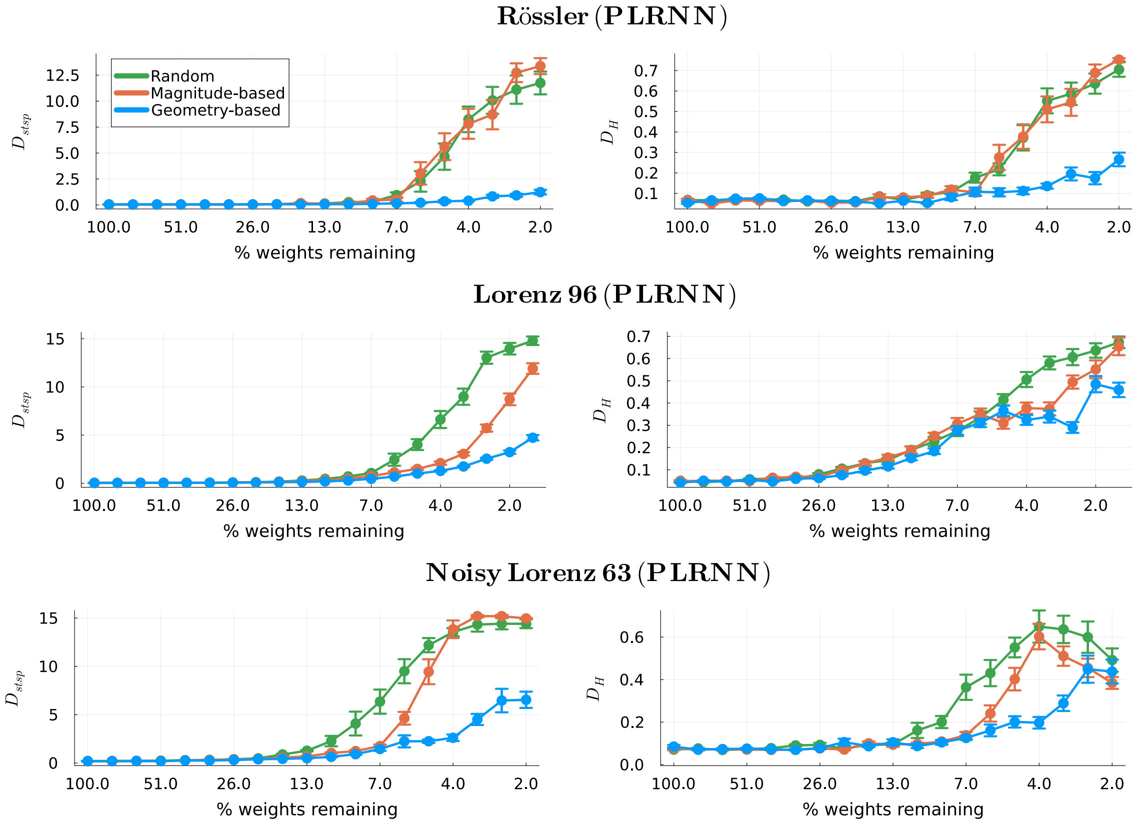


Figure A6. **a**) Weights with large ($\Delta D_{\text{stsp}} > 0.1$) vs. low ($\Delta D_{\text{stsp}} \leq 0.1$) impact on geometrical reconstruction quality when removing 3 weights simultaneously in each iteration. **b**) Change in geometrical agreement (ΔD_{stsp}) as a function of pruned weight magnitude for PLRNNs trained on human ECG, cf. Fig 1c.


Lorenz 63 (PLRNN, M = 20)

Lorenz 63 (PLRNN, M = 30)

Figure A7. Same as Fig. 3 for PLRNNs of different (smaller) network (latent space) size (Fig. 3 was produced for $M = 50$). Error bars = SEM.


Rössler (PLRNN)

Lorenz 96 (PLRNN)

Noisy Lorenz 63 (PLRNN)

Figure A8. Same as Fig. 3 for the chaotic Rössler system (top row), the chaotic Lorenz-96 system (center), and the chaotic Lorenz-63 system with high (25%) noise level (bottom). Error bars = SEM.

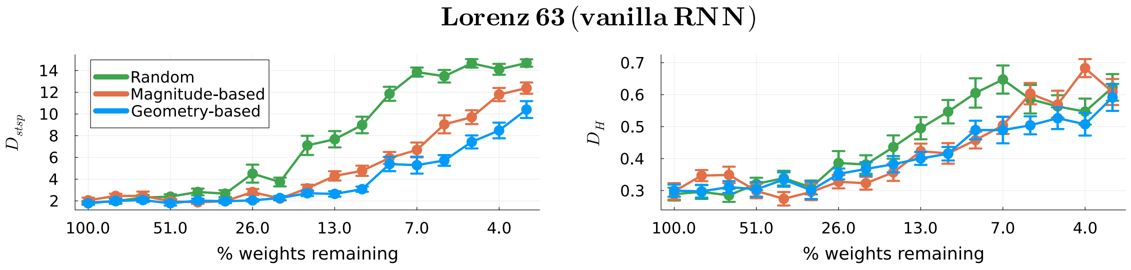


Figure A9. Same as Fig. 3 for a vanilla RNN, $\boldsymbol{z}_t = \phi(\boldsymbol{W}\boldsymbol{z}_{t-1} + \boldsymbol{C}\boldsymbol{x}_t + \boldsymbol{b})$ with $\hat{\boldsymbol{x}}_t = \boldsymbol{B}\boldsymbol{z}_t + \boldsymbol{h}$. Error bars = SEM.

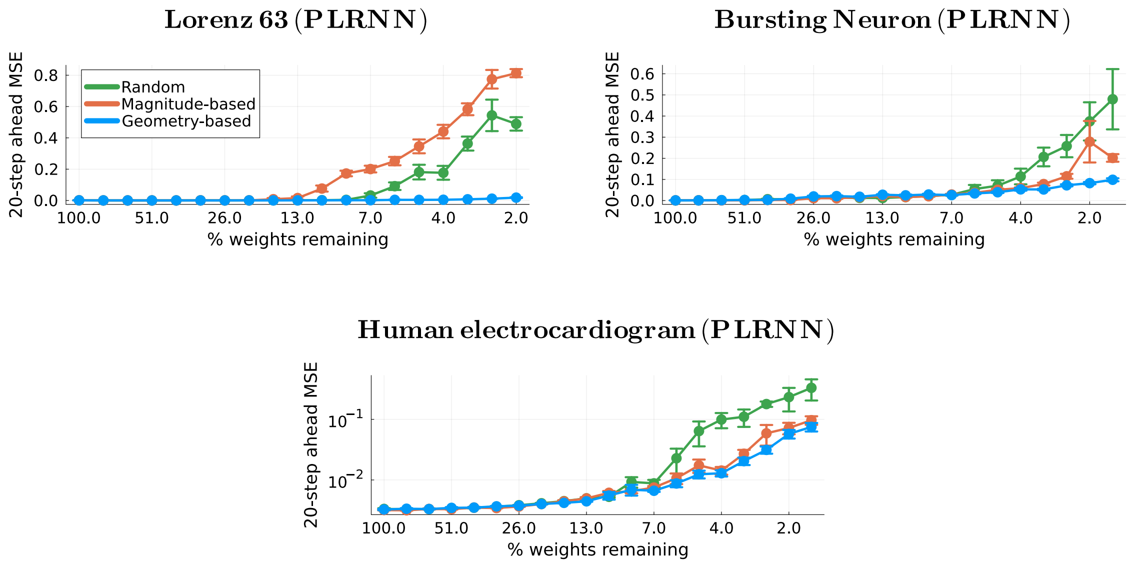


Figure A10. Quantification of DS reconstruction quality in terms of the mean-squared 20-step-ahead prediction error as a function of network pruning (x-axis, exponential scale) and different pruning criteria for the Lorenz-63, bursting neuron, and ECG. Error bars = SEM.

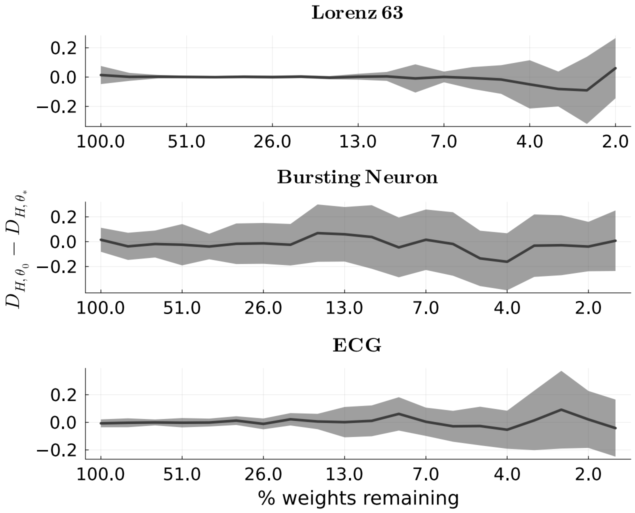


Figure A11. Same as Fig. 4 for D_{H}: Difference in D_{H} when using the initial weights $\boldsymbol{\theta}_0$ and reinitialized weights $\boldsymbol{\theta}_*$ shows there is no notable influence of the specific weight initialization.

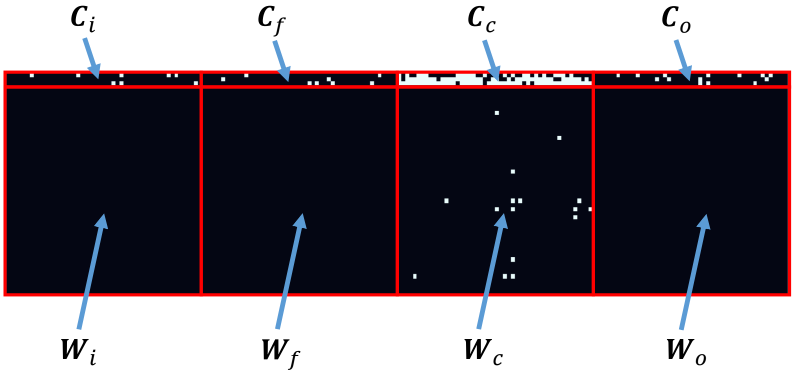


Figure A12. LSTM pruning masks for all types of LSTM weight matrices (input, forget, cell state, and output): Geometry-based pruning is able to identify the relevance of different weight matrices to the performance, therefore excluding $\boldsymbol{W}_i,\ \boldsymbol{W}_f,\ \boldsymbol{W}_o$ from the model, leading to interpretable pruning results in terms of inter-cell connections and their links to observable outputs.

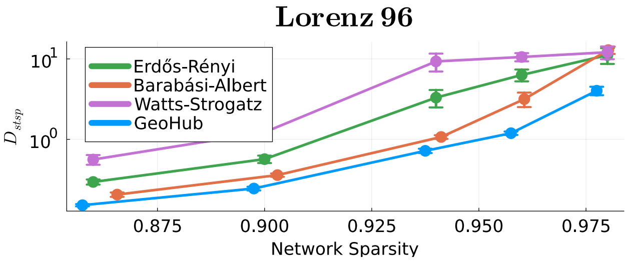


Figure A13. Reconstruction results in terms of state space divergence D_{stsp} as a function of network sparsity $s = 1 - \frac{|\boldsymbol{m}|}{|\boldsymbol{W}|}$ for Erdős–Rényi, Barabási-Albert, Watts-Strogatz and GeoHub graph algorithms. Error bars = SEM.

A.5. Network Topology

A.5.1. Graph Properties

When characterizing graphs, we seek properties that are preserved under different isomorphisms. A fundamental characteristic is the degree distribution $P(k)$, where the degree is the number of connections associated with a node, which may be further distinguished into incoming (k^{in}) and outgoing connections (k^{out}). Two major types of graphs are *single-scale* and *scale-free* graphs, where the latter is surprisingly common in many real-world systems (Barabási & Albert, 1999). While single-scale graphs are often well captured by the binomial or Poisson distribution, $P(k)$ in scale-free graphs follows a power law

$$P(k) \propto k^{-\gamma} \ . \tag{26}$$

Such graphs are characterized by the presence of many nodes with low and a few nodes with high degrees, often referred to as hubs.

Many algorithms exist for finding the shortest path between two nodes in a graph, as required to compute the mean average path length $L(G)$ in Eqn. 6. Here we use the Floyd-Warshall algorithm (Floyd, 1962; Warshall, 1962). While originally the clustering coefficient $C(G)$, as used in Eqn. 7, was defined only for undirected graphs (Watts & Strogatz, 1998), here we use the formulation proposed in (Fagiolo, 2007) which considers the more general case with directed edges. When two neighbors of a node v_i are connected by any type of edge, they form a triangle. The number of directed triangles t_i formed by all neighbors of v_i can be calculated from the adjacency matrix $\boldsymbol{A}^{\text{adj}}$ as (Fagiolo, 2007)

$$t_i = \frac{1}{2}\sum_j \sum_h (A^{\text{adj}}_{ij} + A^{\text{adj}}_{ji})(A^{\text{adj}}_{ih} + A^{\text{adj}}_{hi})(A^{\text{adj}}_{jh} + A^{\text{adj}}_{hj}) = \frac{1}{2}\left(\boldsymbol{A}^{\text{adj}} + \left(\boldsymbol{A}^{\text{adj}}\right)^T\right)^3_{ii} \ . \tag{27}$$

The total number of possible triangles T_i is given by (Fagiolo, 2007)

$$T_i = k_i^{tot}(k_i^{tot} - 1) - 2k_i^{\leftrightarrow} \ , \tag{28}$$

where $k_i^{tot} = k_i^{out} + k_i^{in}$ is the total degree and $k_i^{\leftrightarrow}$ is the number of bilateral edges connected to node v_i. The ratio of these two quantities yields the clustering coefficient of a node v_i,

$$C_i = \frac{(\boldsymbol{A}^{\text{adj}} + (\boldsymbol{A}^{\text{adj}})^T)^3_{ii}}{2[d_i^{tot}(d_i^{tot} - 1) - 2d_i^{\leftrightarrow}]}, \tag{29}$$

and the average across all nodes yields Eqn. 7 in sect. 3.3 (Watts & Strogatz, 1998). Intuitively, this quantity measures how likely it is that any two neighbors of a given node are also immediate neighbors. Based on these quantities, Watts & Strogatz (1998) introduced the idea of small-world graphs, which are characterized by a high clustering coefficient yet short average path length. To measure the small-worldness of any graph, these characteristics can be combined in a single measure called the small-world index SWI (NEAL, 2017). Its definition requires a reference, for which Erdős–Rényi random graphs (Erdös & Rényi, 1959) and ring lattice graphs are used. Random graphs have a short average path length L_r and a low clustering coefficient C_r, while, on the contrary, lattice graphs have a high average path length L_l and high clustering coefficient C_l. Based on these one defines

$$\text{SWI} = \frac{L - L_l}{L_r - L_l} \cdot \frac{C - C_r}{C_l - C_r} \; . \tag{30}$$

This measure is thus normalized and further clipped into the range $0 \leq \text{SWI} \leq 1$, where a value close to 1 indicates higher small-worldness.

A.5.2. Graphs Obtained from Geometry-based Pruning

The graph structure of pruned PLRNNs is given through their pruning masks $\boldsymbol{m}$. Investigating the complementary cumulative degree distribution in $\boldsymbol{m}$ on a logarithmic scale (Fig. 5a) reveals a scale-free distribution, indicating the existence of hubs. Those hubs are primarily associated with PLRNN nodes that directly link to the outputs (observations) via the identity mapping used here (see Eqn. 3 and sect. 3.1), see Fig. 5b. The difference between the k^{in}- and the k^{out}-distribution furthermore indicates directedness of the edges, i.e. directed information flow. The pruned PLRNNs also exhibit properties of small-world graphs with a higher clustering C and a path length L not exceeding the one of random graphs (Fig. A14 left), and as evidenced by the SWI (Eqn. 30; Fig. A14 right). Algorithm 2 implements a procedure that respects all these properties, namely directedness in edges, hub nodes preferentially associated with in-going connections and, mainly, for all output units, and small-worldness.

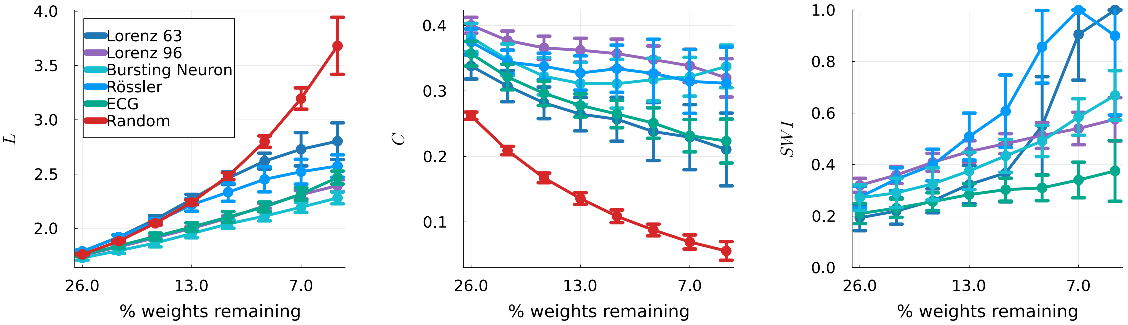


Figure A14. Average path length L (left), clustering coefficient C (center) and SWI (right) for the PLRNN topologies obtained through geometry-based pruning, as a function of the % of weights remaining, and for the different benchmarks used in here. Random graph for comparison in red. Note that x-axes are on an exponential scale for better visualization.

In more detail, since the network topologies obtained by pruning highlight an important difference between readout nodes and hidden nodes, we start constructing the graph by a fully connected model with readout dimension N. Due to the strong hub characteristics we expand this network based on the preferential attachment mechanism of the Barabási-Albert model. To endow specifically the readout nodes with hub-like features, the probability of connecting to these nodes is increased by a term $k_{mean}^{in} = \frac{1}{n} \sum_{i=1}^{n} k_i^{in}$ as observed empirically. To introduce directness within the graph, we further decouple the generation of incoming and outgoing edges, establishing incoming edges first. The probability for a random node connecting to node v_i is thus given by

$$p_i^{in} = \begin{cases} \frac{1}{\mathcal{N}} \ \left(k_i^{in} + \frac{1}{2}k_{mean}^{in}\right), & \text{if } i \in 1, ..., N \\ \frac{1}{\mathcal{N}} \ k_i^{in} & \text{if } i \in N+1, ..., n \end{cases} \tag{31}$$

where the normalization is $\mathcal{N} = \sum_{j=1}^{n} k_j^{in} + \frac{N}{2} k_{mean}^{in}$. In a next step, for each node outgoing edges are created, taking the already established structure into account. To prevent strong hub formation in outgoing edges, a term proportional to $k_{mean}^{out} = \frac{1}{n}\sum_{i=1}^{n} k_i^{out}$, related to the empirically observed out-degree, is added, equalizing the connection probabilities p_i. Specifically, the probability for a random node to receive a connection from node v_i is given by

$$p_i^{out} = \frac{1}{\mathcal{N}} \left(k_i^{in} + k_i^{out} + \frac{1}{4} k_{mean}^{out}\right) , \tag{32}$$

where the normalization is $\mathcal{N} = \sum_{j=1}^{n}(k_j^{in} + k_j^{out}) + \frac{n}{4} k_{mean}^{out}$. (For numerical stability, a constant 0.05 is added to each k_i^{in} and k_i^{out}.) The algorithm is described in detail in Algorithm 2. The user needs to specify a hyperparameter k that determines the number of edges connected to a node, based on which the mean in-degree k_{mean}^{in} and mean out-degree k_{mean}^{out} required for calculating the probabilities is exactly given, making the calculations tractable. Using this model, we are able to generate graphs that fulfill the characteristics found empirically as described further above.

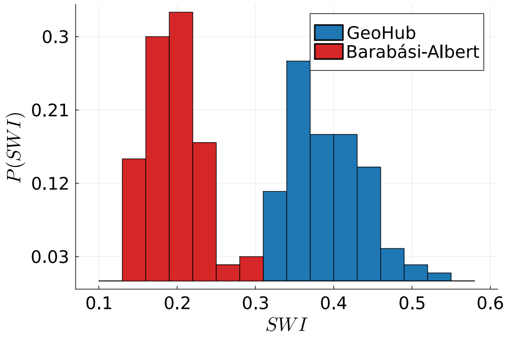


Figure A15. Distribution of the SWI, Eqn. 30, for the GeoHub graph (blue) and the Barabási-Albert model (red), for $n = 100$ and $s \approx 90\%$.

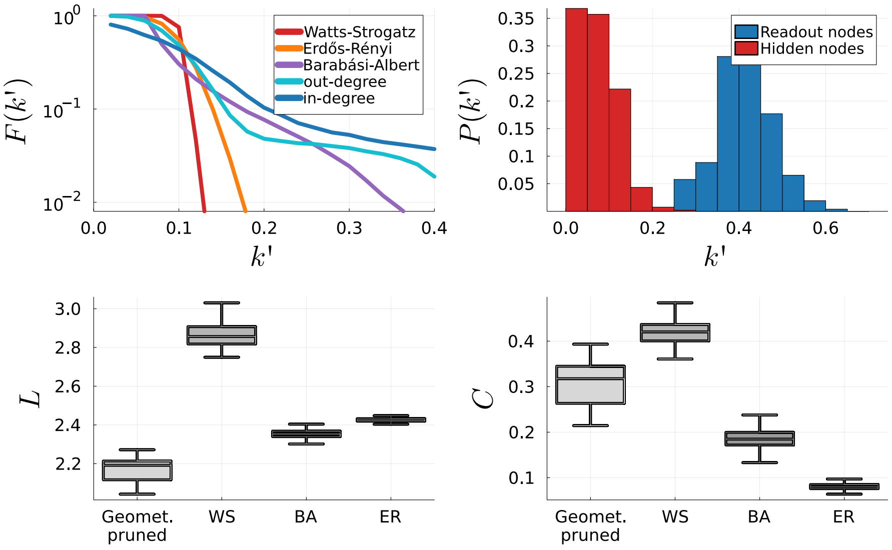


Figure A16. Same as Fig. 5 for network size $M = 100$.

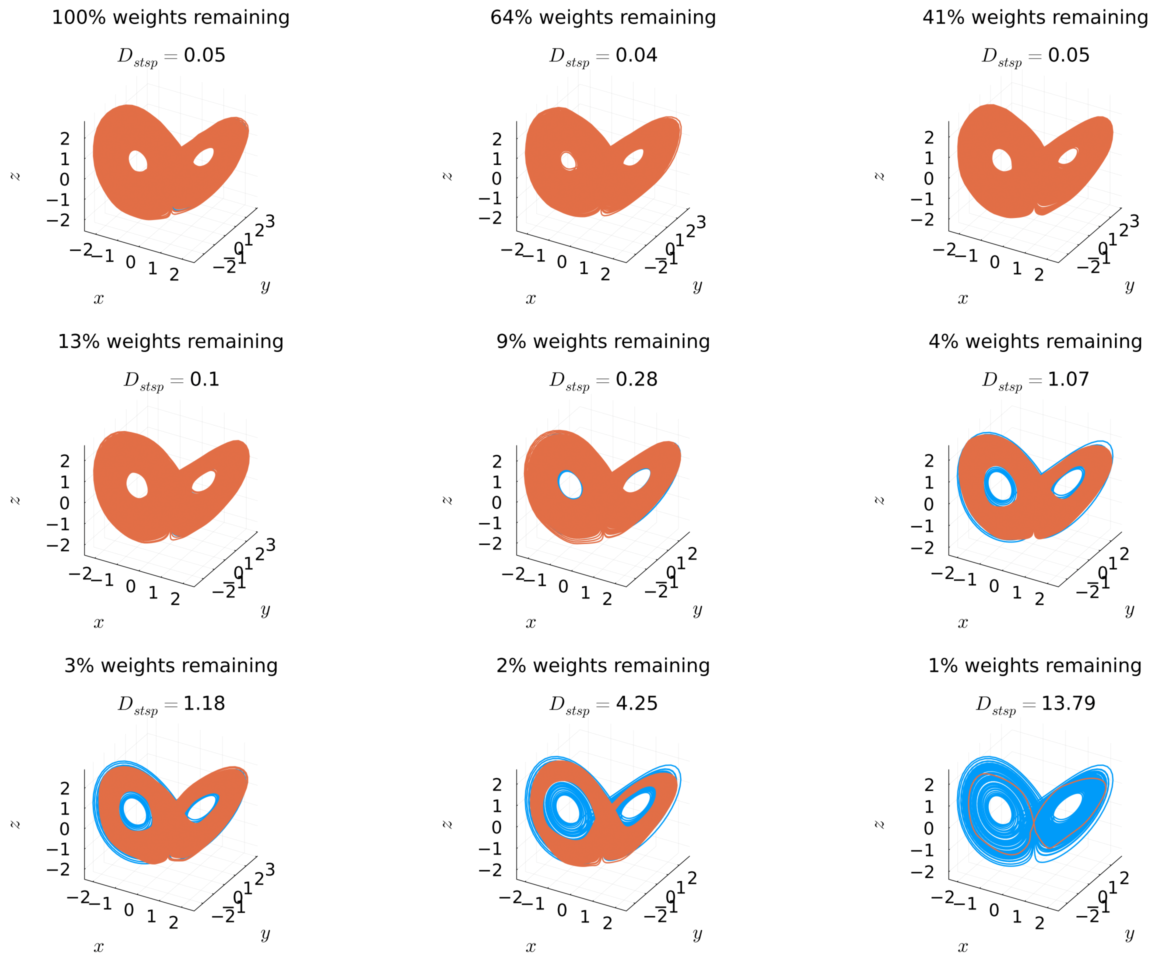


Figure A17. Example reconstructions of the Lorenz-63 at different levels of geometry-based parameter pruning, with an indication of reconstruction quality as measured by D_{stsp} on top.

A.6. Graph Algorithms

A.6.1. Erdős-Rényi Model

Algorithm A1 produces an undirected Erdős-Rényi random graph, where edges are chosen uniformly at random. Required inputs are the number of nodes n and the requested number of edges k.

Algorithm A1 Erdős–Rényi model

```
Input  : Number of nodes $n$, number of edges $k$
Output : Erdős-Rényi graph $G(V, E)$
$G \leftarrow$ empty graph with $|V| = n$ and $|E| = 0$
while $|E| < k$ do
    Choose a random pair of nodes $v_i$ and $v_j$ from $G$
    if $e_{ij} \notin E$ then
        $E = E \cup \{e_{ij}\}$
    end
end
```

A.6.2. Watts-Strogatz Model

The Watts-Strogatz model yields undirected small-world graphs. For constructing such a graph the number of nodes n, the desired degree k, and the rewiring probability p need to be specified.

Algorithm A2 Watts-Strogatz

```
Input  : Number of nodes $n$, number of edges $k$, rewiring probability $p$
Output : Watts-Strogatz graph $G(V, E)$
$G \leftarrow$ ring lattice with $n$ nodes, each connected to its $k$ nearest neighbors
for $i \leftarrow 1$ to $n$ do
    for $j \leftarrow i + 1$ to $i + k/2$ do
        $G \leftarrow$ rewire edge $e_{ij}$ with probability $p$
    end
end
```

A.6.3. Barabási-Albert Model

The Barabási-Albert algorithm generates undirected graphs with a preferential attachment mechanism. The probability p_i of adding an undirected edge to a node v_i is given by

$$p_i = \frac{k_i}{\sum_{j=1}^{n} k_j} \ , \tag{33}$$

which leads to hub-like structures with scale-free degree distribution. The algorithm starts with a fully connected graph with n nodes, and then iteratively adds $n - k$ nodes which are attached to the k existing nodes with probability Eqn. 33.

Algorithm A3 Barabási-Albert

```
Input  : Number of nodes $n$, number of edges $k$
Output : Barabási-Albert graph $G(V, E)$

$G \leftarrow$ graph with $k$ nodes fully connected
for $i \leftarrow k + 1$ to $n$ do
    Add node $v_i$
    for $l \leftarrow 1$ to $k$ do
        Select node $v_j$ with probability $p_j = \frac{k_j}{\sum_h k_h}$
        $E \leftarrow E \cup \{e_{ij}\}$
    end
end
```

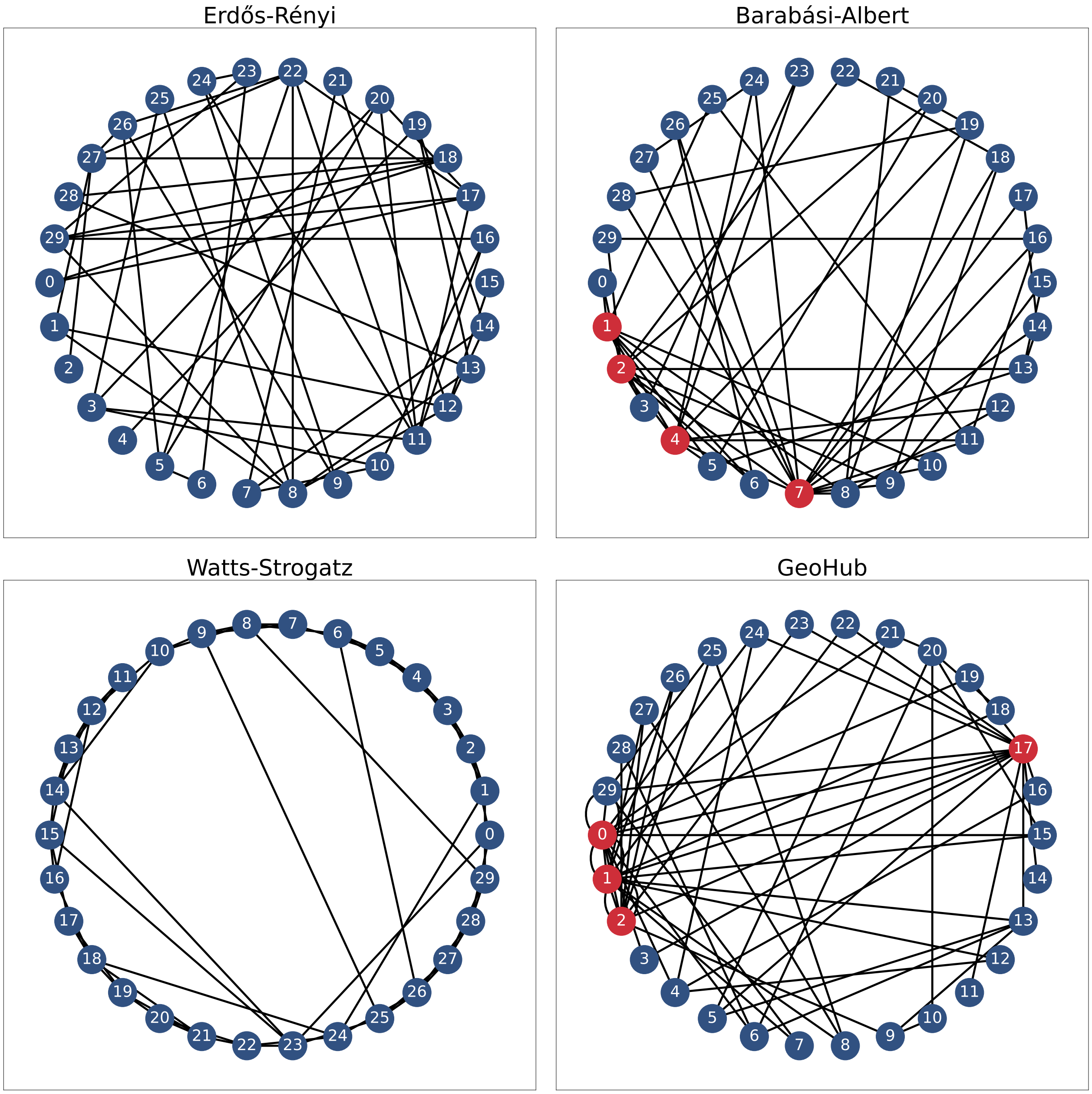


Figure A18. Example graph topologies with network sparsity of 85%. Hubs with ≥ 6 connections are marked in red.