Topological Out-of-Domain Generalization in Dynamical Systems Reconstruction

Georg Trede[1,2,†,*], Charlotte Ricarda Doll[1,2,†,*], Elias Weber[1,2], and Daniel Durstewitz[1,2,3,4,†]

[1]Dept. of Theoretical Neuroscience, Central Institute of Mental Health, Mannheim, Germany
[2]Faculty of Physics and Astronomy, Heidelberg University, Germany
[3]Interdisciplinary Center for Scientific Computing (IWR), Heidelberg University, Germany
[4]ELLIS Unit Heidelberg, Germany
[†]Correspondence to: {georg.trede, charlotte.doll, daniel.durstewitz}@zi-mannheim.de
[*]Equal contribution

Abstract

Predicting the behavior of dynamical systems (DS) beyond the dynamical and parameter regimes observed in training is a pivotal and essentially unresolved problem in scientific ML. It is central to any good scientific theory, which we expect to be able to make predictions about regimes not covered by currently available data. Recent hierarchical and hyper-network guided approaches for DS reconstruction (DSR) enable training on many DS simultaneously, and revealed that extracted latent features are often related to crucial control parameters of the underlying DS that varied across the training corpus. However, true out-of-domain forecasting abilities of these models, e.g., across tipping points, remain limited, and fine-tuning, or even full model retraining, on time series from the new dynamical regime is usually required. Here, we mathematically analyze the root of these limitations in previous model formulations and identify three core shortcomings rooted in a mismatch between structural assumptions of the reconstruction model and typical properties of physical systems. We propose a combination of remedies for these shortcomings, most importantly *feature splitting*, and furthermore derive a bound on the extrapolation error. We demonstrate empirically that our techniques allow for accurate zero-shot prediction into new dynamical regimes outside the observed training regime, as, e.g., encountered across tipping points.[1]

1 Introduction

From turbulent flows and climate systems to ecosystems and neural population activity, much of science is concerned with understanding and predicting the behavior of complex dynamical systems (DS) [10, 21, 24]. A variety of data-driven ML approaches have been proposed toward this goal of dynamical systems reconstruction (DSR) in recent years, ranging from library-based methods like sparse identification of nonlinear dynamical systems (SINDy) [11, 14], methods based on Neural Ordinary Differential Equations (Neural ODEs) [16, 36, 2, 42], Koopman operators [12, 55, 76, 4], Reservoir Computers [61, 63], and physics-informed models [65, 17], to methods employing Recurrent Neural Networks (RNNs) [31, 73, 19, 30] for approximating the DS' underlying flow map. However, most of these require custom training on any observed DS from scratch and do not easily generalize to new systems, parameter configurations, or dynamical regimes [27]. In contrast, from a good scientific model built based on only limited data, we would expect that it can predict the behavior of the underlying DS under novel parameter configurations within potentially novel

[1]Code is available at `https://github.com/DurstewitzLab/TopOODG`.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

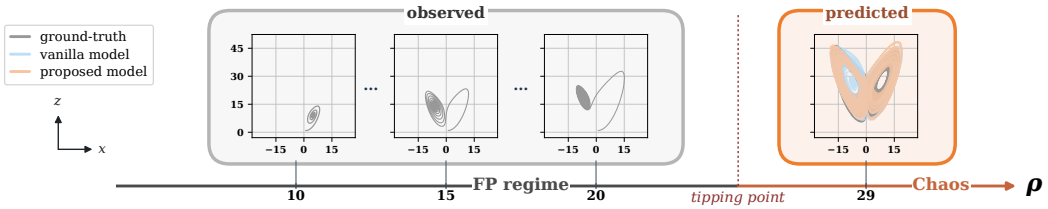


Figure 1: Example of the Lorenz system with training data in the fixed-point (FP) regime (gray) and extrapolation to chaos (orange) as control parameter ρ is varied.

dynamical regimes not previously encountered (Fig. 1). For instance, a good biophysical model of the nervous system, even if constructed from biophysical principles derived solely from healthy tissue, would be able to forecast that certain parameter changes lead to epileptic activity [34]. This is a hallmark of scientific theories that is currently unmatched by data-driven ML/AI approaches [75].

Earlier work on forecasting *non-autonomous* systems has shown some promise in predicting previously unseen dynamics across bifurcations [60, 46]. However, these approaches typically rely on specific assumptions about the data available, which are empirically often not met, and explicitly couple the dynamical variables to the progression of time, thus do not enable the independent exploration of ranges of control parameters. Ultimately, to achieve true out-of-domain (OOD) generalization (beyond the observed parameter range), reconstructing isolated systems is not sufficient [27], but DSR models must be able to capture entire *families* of DS across a range of varying control parameters. Indeed, in the last few years, several such frameworks for training simultaneously across many DS have surfaced [77, 41, 9, 71, 32]. However, so far, none of them solves the fundamental problem of reliably extrapolating beyond the parameter regime observed in training.

Here, we provide a mathematical analysis of the hierarchical DSR frameworks recently proposed by Brenner et al. [9] and Huh et al. [32], identifying the root cause of the limitations discussed above. Hierarchical or meta-learning DSR frameworks differentiate between group-level parameters learned across all observed DS, and a small number of latent features that parameterize system-specific DSR models [77, 41, 71, 32]. It has been observed empirically that after training on a larger set of DS, the model's latent features map (surprisingly often linearly) onto crucial control parameters that varied within the underlying population of DS [9, 32]. While, in theory, this should enable OOD forecasts, in practice fine-tuning or retraining on additional data from the new dynamical regime was still required [9, 41]. This indicates that there must be some mismatch between the model's and the true physical system's structural properties.

We identify three fundamental structural discrepancies that explain these OOD failures: (1) the model Jacobians commonly depend densely on latent features, whereas real systems often exhibit sparse control-parameter dependence; (2) important geometrical properties of the true system's state space vary in a different way with changes in control parameters than is the case in the DSR model (Fig. 2); and (3) the process of time discretization in discrete-time DSR models may induce nonlinear parameter dependencies not present in the true physical DS. By fixing the first two issues and establishing a bound on the extrapolation error for the third, we show empirically that we are able to significantly improve *zero-shot* out-of-domain generalization (OODG) across tipping points. We further demonstrate that the general recipe developed here can also be incorporated into other DSR approaches beyond RNNs and Neural ODEs. Our core contribution is thus a detailed theoretical analysis and solution strategy that significantly improves OODG in DSR without any specific data requirements or strong structural priors.

2 Related work

Dynamical Systems Reconstruction (DSR) DSR aims to infer generative surrogate models that capture the invariant (long-term) topological and statistical properties of a true data-generating process [10, 21]. While library-based approaches like SINDy [11] offer high interpretability, they usually fail on systems which are not covered by their library terms [30]. Alternatively, universal approximators of DS such as RNNs [78, 19, 52] and Neural ODEs [16, 36, 2, 42] provide high expressivity. However, training these highly parameterized models on chaotic trajectories is inher-

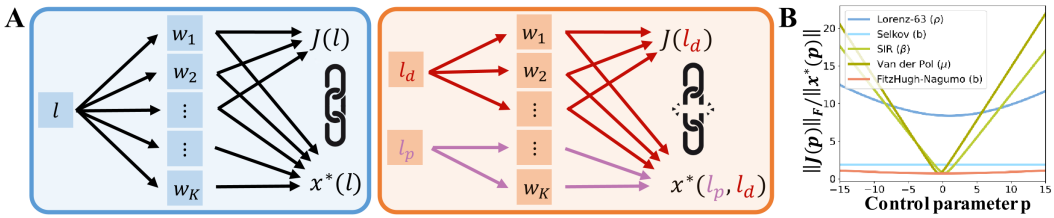


Figure 2: Illustration of hierarchical DSR models. **(A)** Vanilla hierarchical model (left) uses a single feature vector $\boldsymbol{l}$ to obtain weights $\boldsymbol{w}$ of specific DSR model instantiations, which entangles the scaling of the Jacobian $\boldsymbol{J}(\boldsymbol{l})$ with the fixed point (FP) location $\boldsymbol{x}^*(\boldsymbol{l})$ as $\boldsymbol{l}$ changes. In contrast, utilizing independent features (right) $\boldsymbol{l}_d$ and $\boldsymbol{l}_p$ for the local dynamics and FP position enables the decoupled scaling observed in physical systems. **(B)** Examples of differential Jacobian-vs.-location scaling behavior with control parameter p for different 'textbook' examples of DS. $\|\cdot\|_F$ denotes the Frobenius norm; the Jacobian is evaluated at a constant point $\boldsymbol{x}_0$.

ently unstable, necessitating advanced control-theoretic strategies [52, 30] or special regularization criteria [33, 64] that ensure long-term statistics and attractor geometries are correctly recovered.

Extrapolating across bifurcations and tipping points In DSR, true OODG means that models should be able to extrapolate to domains *topologically* different from the training regime, such as different basins of attraction of a multistable DS or novel parameter regimes that drive a DS across a tipping point [3, 27]. Several studies designed *non-autonomous* (explicitly time-dependent) DSR models based on reservoir computers [38, 60, 46] for forecasting dynamical regimes beyond tipping points [3], often enhancing performance through specifically designed training curricula [60] or physics informed approaches [47]. However, these approaches remain limited as they often rely on particular assumptions about the structure of the DS or of the data (e.g., by assuming control parameters are explicitly available in training; [38]), or by exploiting the behavior of trajectories near critical points or other topological dependencies among dynamical regimes [60, 32]. Similarly, approaches like SINDyCP [56] achieve cross-bifurcation extrapolation only via a prespecified library of parameter dependencies. This restricts the application of all these methods to scenarios where the (often strong) assumptions are met, leaving the more general problem of predicting across tipping points unresolved.

Learning parameterized DS families While the DSR models described above are typically custom-trained on a system at hand, more recent DSR foundation models [29] and frameworks for learning entire families of DS were developed, based on hierarchical and/or hyper-network approaches [28, 74]. These typically separate shared physical laws from environment-specific parameters via low-dimensional context vectors [77, 41, 58] or employ meta-learning across diverse DS families [71, 57, 9, 44]. Importantly in the present context, it has been observed that in hierarchical models [9, 32] unsupervised latent features often correlate strongly with the true underlying control parameters, potentially providing a structural basis for knowledge transfer into new dynamical domains.

3 Background and problem setup

3.1 Dynamical Systems

Following standard DS textbooks [1, 26, 45], let $T \subseteq \mathbb{R}$ denote continuous time and $X \subseteq \mathbb{R}^M$ the state space of a DS. A DS is defined as a triple (T, X, Φ_t), where $\Phi_t : T \times X \to X$ is the flow map which describes the evolution of the DS' state across time $t \in T$ starting from $\boldsymbol{x} \in X$. For the common setting of continuous-time DS, the flow map may be thought of as the solution operator of a DS described by ODEs with parameters $\boldsymbol{p} \in \mathbb{R}^P$,

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{p}). \tag{1}$$

While $\boldsymbol{p}$ may be continuous, in any empirical setting we only have access to discrete samples, forming a family of DS $\{\mathcal{S}^{(j)}\}_{j=1}^{J}$ with parameters $\boldsymbol{p}^{(j)}$.

To clarify the idea of topological OODG, we need the concept of topological equivalence [62, 22]:

Definition 1 (Topological equivalence and conjugacy) *Let $(\mathbb{R}, A, \Phi)$ and $(\mathbb{R}, B, \Psi)$ be two DS. These are said to be topologically equivalent if there is a homeomorphism $h : A \to B$ such that for each $\boldsymbol{x}_0 \in A$ we have $h(\Phi_t(\boldsymbol{x}_0)) = \Psi_\tau(h(\boldsymbol{x}_0))$ with $\frac{d\tau(t,\boldsymbol{x}_0)}{dt} > 0$ everywhere (i.e., if h preserves the direction of flow). They are said to be topologically conjugate, if $\tau(t, \boldsymbol{x}_0) = t$ (i.e., if the parameterization by time is the same).*

By *topological* OODG we thus mean that the model is able to generalize to dynamical regimes *not topologically equivalent* to those seen in training. For instance, a bifurcation – by definition – implies a topological change in the state space properties [62].

3.2 Target families

For many physical systems the dependence on control parameters – rates, couplings, forcings, concentrations – is affine, as $\boldsymbol{p}$ enters multiplicatively into the interactions with mechanistic terms in the underlying equations. Canonical examples in this class include the Lorenz-63 [49] and Lorenz-96 [50] models of atmospheric convection, the van-der-Pol [13], Lotka–Volterra [51], and Selkov [67] oscillators, the FitzHugh–Nagumo neuron model [23], the SIR epidemic model [37], or almost any chemical reaction or population ecology system [53, 54]. Mathematically these systems follow a specific form of Eq. 1, given by

$$\dot{\boldsymbol{x}} = \boldsymbol{f}_0(\boldsymbol{x}) + p\, \boldsymbol{f}_1(\boldsymbol{x}) \qquad \boldsymbol{J}_{\text{cont}}(\boldsymbol{x}, p) := \frac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{x}} = \boldsymbol{J}_0(\boldsymbol{x}) + p\, \boldsymbol{J}_1(\boldsymbol{x}), \tag{2}$$

where $\boldsymbol{x} \in \mathbb{R}^M$, $\boldsymbol{f}_0, \boldsymbol{f}_1 : \mathbb{R}^M \to \mathbb{R}^M$. For analytical clarity, we focus primarily on the scalar case ($p \in \mathbb{R}$), but we discuss the generalization to multi-dimensional parameters in Appx. A.1. Note that DS where p enters nonlinearly (e.g., as p^2 or $\sqrt{p}$) into the vector field equations can still be analyzed within our framework after reparameterization ($p \to q := p^2$ or $p \to q := \sqrt{p}$, see Appx. A.1).

3.3 Hierarchical DSR with affine feature maps

To learn an entire family of DS, rather than fitting separate models for each realization, hierarchical and context-conditioned approaches are the most common choice. Building on context-aware dynamics models [9, 41, 71, 77], we represent the parameters of a system-specific model via a shared affine mapping of low-dimensional features $\boldsymbol{l}^{(j)} \in \mathbb{R}^{1\times L}$, which may be learned [77, 9, 32] from, or directly provided by, the context, while the mapping itself is defined by group-level parameters $\boldsymbol{\theta}_c$, $\boldsymbol{\theta}_v$:

$$\boldsymbol{\theta}^{(j)} = \boldsymbol{\theta}_c + \boldsymbol{l}^{(j)}\boldsymbol{\theta}_v = \boldsymbol{\theta}_c + \sum_{i=1}^{L} l_i^{(j)} \boldsymbol{\theta}_{v,i}, \tag{3}$$

Here, $\boldsymbol{\theta}_c \in \mathbb{R}^Q$ (centered parameters) and $\boldsymbol{\theta}_v \in \mathbb{R}^{L\times Q}$ (variation inducing weights) are shared across all systems, i.e., are independent of $\boldsymbol{l}$. $\boldsymbol{\theta}_v$ acts as a constant linear map that determines how changes in the latent features $\boldsymbol{l}$ translate into shifts in the model parameters $\boldsymbol{\theta}^{(j)}$. Furthermore, it has been demonstrated [41, 9, 71, 32] that such learned features are commonly connected to the true underlying control parameter p via some injective function $\boldsymbol{g}$:

$$\boldsymbol{l} = \boldsymbol{g}(p), \qquad \boldsymbol{g} : \mathbb{R} \to \mathbb{R}^{1\times L}. \tag{4}$$

While these hierarchical approaches, which separate system-specific from group-level parameters, generally yield strong DSR performance and have a number of nice properties [7, 9, 32, 41, 77], they meet fundamental structural limitations w.r.t. OODG that are as yet largely unexplored. To analyze these limitations, we consider two representative architectural classes for modeling families of DS. These reflect two complementary perspectives on DSR: learning the flow map directly in discrete time using RNNs, and learning the underlying vector field in continuous time using Neural ODEs. For analytical clarity, we focus primarily on the scalar feature case ($L = 1$), but we discuss the generalization to multi-dimensional features in Section 4.2.

RNNs (discrete-time flow approximation):

$$\boldsymbol{z}_{t+1} = \boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{z}_t) = \boldsymbol{W}\,\boldsymbol{\psi}(\boldsymbol{z}_t) + \boldsymbol{h}\,, \qquad \boldsymbol{J}_{\boldsymbol{\theta}}^{\text{RNN}}(\boldsymbol{z}_t) := \partial \boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{z}_t)/\partial \boldsymbol{z}_t = \boldsymbol{W}\,\text{diag}\big(\boldsymbol{\psi}'(\boldsymbol{z}_t)\big) \tag{5}$$

Here, the mapping $\boldsymbol{F_\theta}$ represents a learned approximation of the flow map Φ acting on a latent state $\boldsymbol{z}_t \in \mathbb{R}^N$ at the sampled times $t_i = i\Delta t$, with weight parameters $\boldsymbol{W} \in \mathbb{R}^{N\times N}$, bias term $\boldsymbol{h} \in \mathbb{R}^N$, and a differentiable and element-wise nonlinearity ψ. We assume a minimal single-layer architecture for simplicity and as common in the DSR literature [52, 30, 8], although the theoretical results generalize to more complex networks with hidden layers (Appx. A.2).

Neural ODEs (continuous-time vector field approximation):

$$\dot{\boldsymbol{z}}_t = \boldsymbol{f_\theta}(\boldsymbol{z}_t) = \boldsymbol{W}\psi(\boldsymbol{z}_t) + \boldsymbol{h}\,, \qquad \boldsymbol{J}_{\boldsymbol{\theta}}^{\text{NODE}}(\boldsymbol{z}_t) := \partial \boldsymbol{f_\theta}/\partial \boldsymbol{z}_t = \boldsymbol{W}\,\mathrm{diag}\big(\psi'(\boldsymbol{z}_t)\big) \tag{6}$$

In contrast to RNNs, Neural ODEs parameterize the vector field $\boldsymbol{f_\theta}$ that generates the continuous-time dynamics. For direct comparability, we use the same network structure as for the RNN (Eq. 5).[2]

Applying the feature mapping (Eq. 3) to the RNN and Neural ODE, we obtain the following affine-in-l Jacobian, which governs its local growth attributes, called **dynamical scaling** in the following:

$$\boldsymbol{J_\theta} = (\boldsymbol{W}_c + l\boldsymbol{W}_v)\mathrm{diag}\big(\psi'(\boldsymbol{z}_t)\big)\,, \qquad \frac{\partial \boldsymbol{J_\theta}}{\partial l} = \boldsymbol{W}_v \mathrm{diag}\big(\psi'(\boldsymbol{z}_t)\big). \tag{7}$$

A key observation now is how the location of the fixed points (FPs) $\boldsymbol{z}^*(l)$ depends on the latent feature, which we call **positional scaling** in the following (referring to a mere *translation* of the vector field that does not alter the local Jacobian). For the discrete-time RNN, the FP must satisfy $\boldsymbol{z}^*(l) = \boldsymbol{F_\theta}(\boldsymbol{z}^*(l), l)$. Differentiating both sides with respect to l yields:

$$\frac{d\boldsymbol{z}^*}{dl} = \frac{\partial \boldsymbol{F_\theta}}{\partial \boldsymbol{z}}\frac{d\boldsymbol{z}^*}{dl} + \frac{\partial \boldsymbol{F_\theta}}{\partial l} \implies \frac{d\boldsymbol{z}^*}{dl} = (\boldsymbol{I} - \boldsymbol{J}(l))^{-1}\,\frac{\partial \boldsymbol{F}_\theta}{\partial l}\bigg|_{\boldsymbol{z}^*}\,, \tag{8}$$

where $\frac{\partial \boldsymbol{F_\theta}}{\partial l}$ does not explicitly depend on l (Appx. A.2), and we assume $(\boldsymbol{I} - \boldsymbol{J}(l))$ to be invertible, i.e., the Jacobian $\boldsymbol{J}(l)$ has no eigenvalue of $+1$, since such an eigenvalue would make the matrix singular and prevent the fixed point from being finite and isolated. For the continuous-time Neural ODE, the equilibrium condition $\boldsymbol{0} = \boldsymbol{f_\theta}(\boldsymbol{z}^*(l), l)$ leads to an analogous implicit derivative (assuming invertibility of $\boldsymbol{J}(l)$):

$$\boldsymbol{0} = \frac{\partial \boldsymbol{f_\theta}}{\partial \boldsymbol{z}}\frac{d\boldsymbol{z}^*}{dl} + \frac{\partial \boldsymbol{f}_\theta}{\partial l} \implies \frac{d\boldsymbol{z}^*}{dl} = -\boldsymbol{J}(l)^{-1}\,\frac{\partial \boldsymbol{f_\theta}}{\partial l}\bigg|_{\boldsymbol{z}^*} \tag{9}$$

Thus, for both cases the positional scaling depends on the inverse of the Jacobian $\boldsymbol{J}(l)$ (or $\boldsymbol{I}-\boldsymbol{J}(l)$ respectively), which is itself a function of l. This means that the shift in FP location is mathematically entangled with the change in the system's Jacobian and thus Lyapunov spectrum, which determines the rate of expansion or contraction along different state space directions. This is the basis for the structural bottleneck we formally analyze in Section 4.2.

4 Structural analysis of affine hierarchical models

4.1 Structural overparameterization and Jacobian sparsity

In physical systems, a control parameter p typically governs a specific, isolated mechanism, resulting in a sparse derivative of the Jacobian (although there are also exceptions, like coupled oscillator systems [15]). For instance, the Rayleigh number ρ affects a single entry of the Lorenz-63 Jacobian [49], the infection rate β influences two entries in SIR models [37], and other models such as the Selkov oscillator [67] or the Lorenz-96 system [50] have control parameters representing constant inflow or forcing which do not enter the Jacobian at all (i.e., $\boldsymbol{J}_1 = \boldsymbol{0}$). Likewise, in all of the most common bifurcation normal forms the dependence on control parameters is extremely low-dimensional [45]. In contrast, unconstrained affine feature mappings can introduce spurious, dense parameter dependencies that corrupt the topological structure when extrapolated.

Proposition 1 (Perturbations due to dense affine parameter-dependence) *Let the true Jacobian's dependence on p be affine, $\boldsymbol{J}_{true}(p) = \boldsymbol{J}_0 + p\boldsymbol{J}_1$. We assume $\boldsymbol{J}_1 \in \mathcal{S}$, where $\mathcal{S} \subset \mathbb{R}^{N\times N}$ is a k-dimensional subspace consisting of matrices that share a fixed, sparse support pattern ($k \ll N^2$), and $\mathcal{S}^\perp$ is its complement. If the DSR model's feature-coupling matrix $\boldsymbol{\theta}_v$ (Eq. 3) is full-rank and*

[2] Note that although in Eq. 6 $\boldsymbol{z}$ depends continuously on time, for notational consistency we write t as an index.

unregularized, it yields $(N^2 - k)$ unconstrained degrees of freedom within the model's feature-dependent Jacobian $\hat{\boldsymbol{J}}_1$. These spurious components lie orthogonal to the true sparse matrix subspace $\mathcal{S}$, allowing an unbounded $\mathcal{O}(|p|)$ spectral divergence outside the training domain.

Proof sketch (full proof in Appx. G.1.) The model's learned feature-dependent Jacobian term $\hat{\boldsymbol{J}}_1$ can be decomposed into the sum of $\hat{\boldsymbol{J}}_1^{\parallel} \in \mathcal{S}$ and $\hat{\boldsymbol{J}}_1^{\perp} \in \mathcal{S}^{\perp}$. Empirical risk minimization (ERM) on finite training data determines $\hat{\boldsymbol{J}}_1^{\parallel}$, but leaves the (N^2-k)-dimensional orthogonal complement $\hat{\boldsymbol{J}}_1^{\perp}$ underdetermined. If left unregularized, dense weight matrices $\boldsymbol{\theta}_v$ (Eq. 3) with $\hat{\boldsymbol{J}}_1^{\perp} \neq 0$ perturb $\hat{\boldsymbol{J}}_1$ out of the k-dimensional target space $\mathcal{S}$. By the Bauer–Fike theorem [5], moving the feature $l \propto p$ outside of the training domain produces linear amplification of such perturbations, which permits spurious eigenvalues to cross the imaginary axis or unit circle and may cause false bifurcations.

Solution: Sparsity and low-rank constraints To directly prevent the orthogonal perturbation established in Proposition 1, we must restrict the dimensionality of the learned feature-coupling Jacobian $\hat{\boldsymbol{J}}_1$. Because true physical systems typically restrict control parameter effects to specific, sparse mechanistic pathways ($k \ll N^2$), dense parameter dependencies in the DSR model formulation allowing $\hat{\boldsymbol{J}}_1^{\perp} \neq 0$ should be actively penalized. Specifically, we propose a low-rank constraint (e.g., with target rank k') jointly with an $\mathcal{L}_1$ regularization on the feature-coupling parameters $\boldsymbol{\theta}_v$ (Eq. 3) which is added to the training objective:

$$\mathcal{L}_{\text{reg}} = \lambda_{\text{reg}} \|\boldsymbol{\theta}_v\|_1, \qquad \text{where } \boldsymbol{\theta}_v = \boldsymbol{U}\boldsymbol{V}^{\top};\ \boldsymbol{U} \in \mathbb{R}^{L \times k'},\ \boldsymbol{V} \in \mathbb{R}^{Q \times k'} \tag{10}$$

Unlike standard ERM, which leaves excess degrees of freedom underdetermined, incorporating this sparsity prior into the training objective actively penalizes the model for populating the physically spurious (N^2-k)-dimensional orthogonal complement $\hat{\boldsymbol{J}}_1^{\perp}$, and thereby suppresses the unbounded $\mathcal{O}(|p|)$ spectral divergence Proposition 1 suggests. Crucially, we leave the centered parameters $\boldsymbol{\theta}_c$ (Eq. 3) unregularized. As $\boldsymbol{\theta}_c$ only affects the feature-independent $\hat{\boldsymbol{J}}_0$ term of the Jacobian and does not interact with the feature l, it does not contribute to the perturbation responsible for the spectral divergence. This targeted regularization thus encourages the training algorithm to align the model's feature-coupling weights almost exclusively with the true sparse matrix subspace.

Empirical validation Figure 3A shows a comparison of single-hidden-layer Neural ODEs (Appx. C.2) with and without the proposed sparsity and low-rank constraints, trained on the Lorenz-63 fixed-point regime. For the purpose of evaluating the fit between predicted and true dynamics for unseen control values, we fit an algebraic function mapping the true control parameter to the learned latent features and use this to extrapolate into the new domain (see Appx. E for details, and for how to achieve extrapolation in empirical settings w/o any such access to true control parameters).[3]

While both models exhibit comparable in-domain performance (measured via Wasserstein-1 distance; Appx. F), their out-of-domain behavior drastically differs. The unconstrained model, exhibiting a dense Jacobian-feature dependence $\partial \boldsymbol{J}_\theta / \partial \rho$ (Fig. 3B, top), suffers from the predicted spectral divergence and fails at the transition to chaos ($\rho \approx 24.06$). Conversely, the regularized model maintains a sparser dependence (Fig. 3B, bottom), successfully suppressing spurious behavior and recreating the ground truth dynamics far beyond the training boundaries. Similar behavior was observed for discrete-time RNNs, Eq. 5 (Appx. C.1), see Fig. 7 & Appx. B.2.

In practice, the required sparsity and low-rank constraints usually would need to be determined via in-domain grid search. If, for instance, the control parameter dependence is truly dense, strong sparsity regularization may be counterproductive, as we empirically confirmed for the example of the coupled Stuart-Landau oscillator system [15] (see Appx. B.6).

4.2 Compatibility of positional and dynamical scaling

For real physical systems undergoing changes in control parameters, the location of attractors (like stable FPs) and the rates of con-/divergence as determined by the Jacobians often vary independently from each other (Fig. 2B). However, forcing all model parameters to depend on a single feature (Eq. 3) algebraically entangles these two properties, as made explicit by the following theorem, which provides a condition for the compatibility of model and true system:

[3] Note that access to control parameters is only assumed here for the purpose of evaluation, but not required in training.

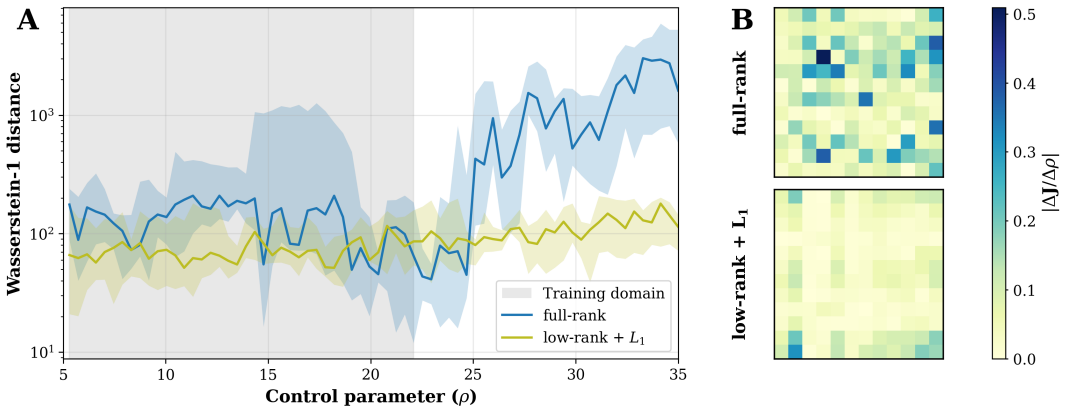


Figure 3: Comparison of low-rank-&-sparsity-regularized DSR model with vanilla hierarchization trained on the Lorenz-63 DS (Appx. D.1) with control parameter ρ in the two-fixed-point regime, and extrapolated into the chaotic regime. **(A)** Disagreement in attractor geometry as assessed by Wasserstein-1 distance between true and model-generated bifurcation graphs (Appx. F) for the hierarchical Neural ODE with and w/o low-rank & sparsity constraints (Eq. 10) as a function of ρ. Curves are medians across $n = 10$ trained models with error bands indicating the interquartile range. **(B)** Change of model Jacobian entries with ρ for the unconstrained (top) and the low-rank-&-sparsity-regularized model (bottom).

Theorem 1 (Compatibility of positional and dynamical scaling under feature sharing) *Consider a DSR model following Eqs. 3-6 with identity observation mapping, $\boldsymbol{x} = \boldsymbol{z}$, and positional scaling $\partial \boldsymbol{z}^*/\partial l$ given by Eqs. 8 & 9. Let the true system have an isolated FP $\boldsymbol{x}^*_{\text{true}}(p)$ that is differentiable w.r.t. p. Further assume $l = g(p)$ (Eq. 4) is differentiable, and the model correctly captures the true local Jacobian, $\boldsymbol{J}_{\text{model}}(\boldsymbol{x}^*_{\text{true}}(p), g(p)) = \boldsymbol{J}_{\text{true}}(p)$, throughout the considered parameter interval. Then, the model matches the location of the true FP for all p throughout the considered interval if and only if it matches the true FP at one parameter value p_0 and*

$$\frac{\mathrm{d}\boldsymbol{x}^*_{\text{true}}}{\mathrm{d}p} = \begin{cases} -g'(p)\, \boldsymbol{J}_{\text{true}}(p)^{-1} \left[\boldsymbol{W}_v\, \psi(\boldsymbol{x}^*_{\text{true}}(p)) + \boldsymbol{h}_v\right], & \textit{Neural ODE,} \\ g'(p)\, \left(\boldsymbol{I} - \boldsymbol{J}_{\text{true}}(p)\right)^{-1} \left[\boldsymbol{W}_v\, \psi(\boldsymbol{x}^*_{\text{true}}(p)) + \boldsymbol{h}_v\right], & \textit{RNN.} \end{cases} \tag{11}$$

Thus, after the shared feature $l = g(p)$ has been determined such as to capture the dynamical scaling through $\boldsymbol{J}_{\text{true}}(p)$, the model can capture the positional scaling only if the true FP obeys this compatibility relation, requiring a specific change of FP location with p determined by $\boldsymbol{W}_v, \boldsymbol{h}_v$, the model architecture, and ψ. Only if this relation holds, both dependencies can be represented exactly; if it does not, a single shared affine feature cannot capture both simultaneously.

Proof sketch (full proof in Appx. G.2.) Differentiate the model FP residual (i.e., $\boldsymbol{r}_{\text{NODE}}(p) := f_{\boldsymbol{\theta}}(\boldsymbol{x}^*(p), p)$ or $\boldsymbol{r}_{\text{RNN}}(p) := \boldsymbol{x}^*(p) - F_{\boldsymbol{\theta}}(\boldsymbol{x}^*(p), p)$) w.r.t. p, evaluated at $\boldsymbol{x}^*_{\text{true}}(p)$. Matching the Jacobians and assuming affine feature mapping yields $\boldsymbol{J}_{\text{true}}\boldsymbol{x}^{*\prime}_{\text{true}} + g'(\boldsymbol{W}_v\psi(\boldsymbol{x}^*_{\text{true}}) + \boldsymbol{h}_v)$ for the Neural ODE, and $(\boldsymbol{I}-\boldsymbol{J}_{\text{true}})\boldsymbol{x}^{*\prime}_{\text{true}} - g'(\boldsymbol{W}_v\psi(\boldsymbol{x}^*_{\text{true}}) + \boldsymbol{h}_v)$ for the RNN. Under the stated invertibility assumptions, the relevant derivative vanishes exactly when the compatibility relation holds. Since the residual is zero at p_0, it vanishes throughout the interval if and only if its derivative does.

Corollary 1.1 (Local FP mismatch under incompatibility) *(Proof in Appx. G.3) Under Theorem 1, let p_b be a training boundary where the model matches the true FP, and assume the model FP residual has a bounded second derivative within a one-sided OOD neighborhood of p_b. If the compatibility relation (Eq. 11) fails at p_b, the model FP residual at $\boldsymbol{x}^*_{\text{true}}(p_b + \Delta p)$ satisfies $\|\boldsymbol{r}(p_b + \Delta p)\|_2 = \Theta(|\Delta p|)$ as $\Delta p \to 0$ on the OOD side. Thus exact FP matching fails locally, which in turn can cause a failure of topologically equivalent reconstruction.*

Proposition 2 (Conditional OOD amplification of orthogonal feature leakage) *(Proof in Appx. G.4) Assume the affine mapping (Eq. 3) with a multi-dimensional feature vector $\boldsymbol{l} \in \mathbb{R}^L$, and a ground truth system governed by a scalar control parameter p. Let the true $N \times N$-dimensional Jacobian variation lie in a matrix subspace $\mathcal{S} \subseteq \mathbb{R}^{N\times N}$, and let a model with $L > 1$ affine features*

satisfy

$$\boldsymbol{J}_{\text{model}}\big(\boldsymbol{l}(p)\big) = \hat{\boldsymbol{J}}_0 + \sum_{i=1}^{L} l_i(p)\, \hat{\boldsymbol{J}}_{1,i}. \tag{12}$$

Assume that the model matches the true Jacobian at a training boundary p_b, where it can be decomposed as

$$\hat{\boldsymbol{J}}_{1,i} = \hat{\boldsymbol{J}}^{\parallel}_{1,i} + \hat{\boldsymbol{J}}^{\perp}_{1,i}, \qquad \hat{\boldsymbol{J}}^{\parallel}_{1,i} \in \mathcal{S}, \quad \hat{\boldsymbol{J}}^{\perp}_{1,i} \in \mathcal{S}^{\perp}, \tag{13}$$

and let $\Delta\boldsymbol{l}(p) = \boldsymbol{l}(p) - \boldsymbol{l}(p_b)$. Suppose non-degeneracy, i.e., that for some $\beta > 0$,

$$\left\| \sum_{i=1}^{L} \Delta l_i(p)\, \hat{\boldsymbol{J}}^{\perp}_{1,i} \right\|_F \geq \beta \left\| \Delta\boldsymbol{l}(p) \right\|_2 \tag{14}$$

along the considered OOD feature path. Then

$$\left\| \boldsymbol{J}_{\text{model}}\big(\boldsymbol{l}(p)\big) - \boldsymbol{J}_{\text{true}}(p) \right\|_F \geq \beta \left\| \Delta\boldsymbol{l}(p) \right\|_2. \tag{15}$$

Consequently, if $\|\Delta\boldsymbol{l}(p)\|_2 \to \infty$ OOD, the Jacobian mismatch is unbounded. Multiple feature dimensions do not by themselves prevent this leakage: the stated non-degeneracy condition excludes vanishing or mutually canceling orthogonal feature directions.

Solution: Orthogonal feature splitting To resolve the mismatch between model and true system established in Theorem 1 without creating the spectral divergence through unconstrained overparameterization (see Proposition 2), we introduce *feature splitting*. Instead of projecting a single latent feature onto all parameters or naively expanding the feature dimension, we structurally partition the parameter space into two mutually exclusive sets.

We assign a dynamical feature $l_{\text{dyn}} \in \mathbb{R}$ strictly to parameters that govern local growth rates (the Jacobians), and an independent positional feature $l_{\text{pos}} \in \mathbb{R}$ strictly to parameters that shift the vector field without altering its local derivative:

$$P_{\text{dyn}} = \left\{ \boldsymbol{\theta}_i : \frac{\partial \boldsymbol{J}_{\text{model}}}{\partial \boldsymbol{\theta}_i} \neq \mathbf{0} \right\}, \qquad P_{\text{pos}} = \left\{ \boldsymbol{\theta}_i : \frac{\partial \boldsymbol{J}_{\text{model}}}{\partial \boldsymbol{\theta}_i} = \mathbf{0} \right\} \tag{16}$$

The standard feature mapping (Eq. 3) is thereby split into two affine maps for both parameter classes:

$$\boldsymbol{\theta}_i(l_{\text{dyn}}, l_{\text{pos}}) = \begin{cases} \boldsymbol{\theta}_{c,i} + l_{\text{dyn}} \boldsymbol{\theta}_{v,i}, & \text{if } \boldsymbol{\theta}_i \in P_{\text{dyn}} \\ \boldsymbol{\theta}_{c,i} + l_{\text{pos}} \boldsymbol{\theta}_{v,i}, & \text{if } \boldsymbol{\theta}_i \in P_{\text{pos}} \end{cases} \tag{17}$$

This targeted structural intervention can be automated using auto-differentiation and directly resolves the analyzed architectural bottlenecks. Resolving the previous single-feature prior frees the model to learn completely independent nonlinear relationships to the underlying control parameter (e.g., $l_{\text{dyn}} \propto p$ while $l_{\text{pos}} \propto \sqrt{p - p_{\text{crit}}}$ for Lorenz-63). This naturally accommodates the heterogeneous scaling with physical properties as required for true OOD extrapolation, functioning as a robust structural guardrail across various DSR architectures (including RNNs and Neural ODEs).

Empirical validation Trained on multiple control-parameter values of the Selkov system (Appx. D.2) in the fixed-point regime, both discrete-time and continuous-time models (details in Appx. C) *with* feature splitting significantly outperform standard models without splitting in the topological OOD region according to the Wasserstein-1 distance between true and reconstructed bifurcation diagrams (Fig. 4). They do not only correctly predict the onset of the limit cycle but also the second Hopf bifurcation back to FP dynamics. Furthermore, note that simply increasing the feature dimension of conventional hierarchical models does not have the same effect (Fig. 4C). We further confirmed successful extrapolation to the unseen dynamical regime by comparing the maximum Lyapunov exponent and correlation dimension to those of the ground truth DS, see Appx. B.4. Appx. B.3 provides additional empirical validation of our methodology on the Rössler system undergoing a period-doubling route to chaos, and Appx. B.7 on (real-world) membrane-potential recordings from cortical neurons. We also compared our feature-splitting technique to SINDyCP [56] and Next-Generation Reservoir Computers [46] in Appx. B.5, finding that our approach is at least on par with these two methods, *although both of them – in contrast to ours – require explicit knowledge of true control parameters in training*.

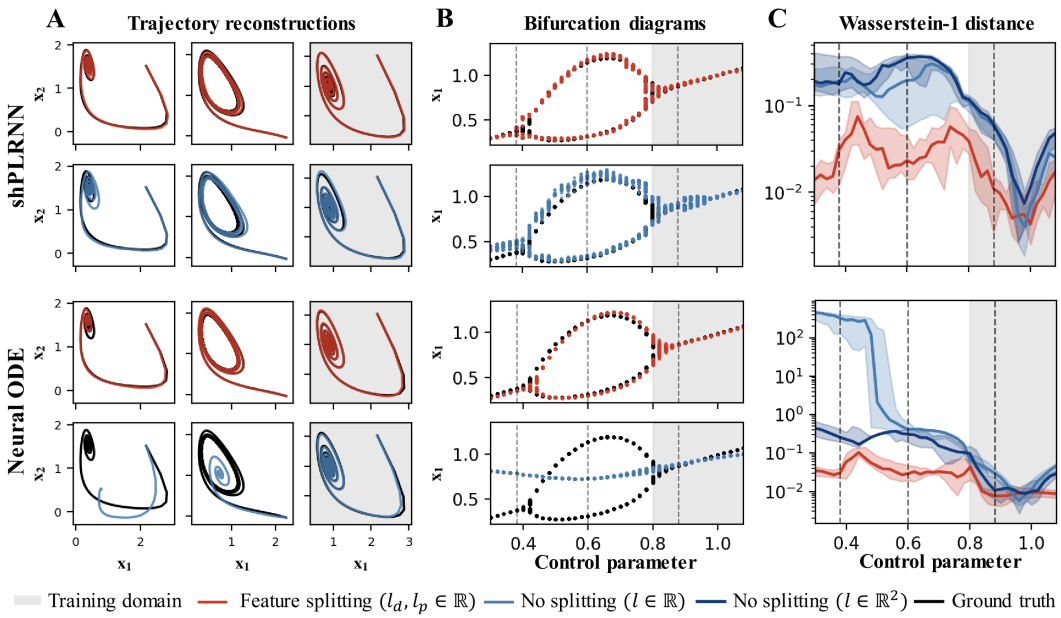


Figure 4: Reconstruction of the Selkov bifurcation structure by models with and without feature splitting. **(A)** Representative trajectory reconstructions at three distinct control parameter values marked by vertical dashed lines in panels B and C. **(B)** Reconstructed bifurcation diagrams (runs with lowest overall Wasserstein-1 distance for each setting). **(C)** Wasserstein-1 distance between true and reconstructed bifurcation diagrams (Appx. F) for models trained with ($l_d, l_p \in \mathbb{R}$) and without ($\boldsymbol{l} \in \mathbb{R}, \mathbb{R}^2$) feature splitting. Solid lines are medians across 10 runs and error bands indicate the interquartile range. The shaded area indicates the training domain.

4.3 Nonlinear scaling constraints in continuous-to-discrete time mapping

Representing a continuous-time parameterized system $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, p)$ as a discrete-time map (as in RNNs and other discrete flow approximations, but not for Neural ODEs) transforms a fundamentally affine parameter dependence into a nonlinear one. Consider a continuous-time system at a control parameter value p_0 and center the parameter at the point of expansion, $\overline{p} := p - p_0$, then:

$$\boldsymbol{f}(\boldsymbol{x}, p) = \boldsymbol{f}_0(\boldsymbol{x}) + \overline{p}\boldsymbol{f}_1(\boldsymbol{x}), \qquad \boldsymbol{J}_{\text{cont}}(\boldsymbol{x}, p) = \boldsymbol{J}_0(\boldsymbol{x}) + \overline{p}\boldsymbol{J}_1(\boldsymbol{x}),$$

where $\boldsymbol{f}_0(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}, p_0)$ and $\boldsymbol{J}_0(\boldsymbol{x}) = \boldsymbol{J}_{\text{cont}}(\boldsymbol{x}, p_0)$. The Jacobian of the true continuous-time flow map $\Phi_{\Delta t}$ can be expanded for small Δt as follows (details in Appx. G.5):

$$\boldsymbol{J}_{\text{disc}}(p, \boldsymbol{x}) = \boldsymbol{I} + (\boldsymbol{J}_0 + \overline{p}\boldsymbol{J}_1)\Delta t + \frac{1}{2}\left[(\boldsymbol{J}_0 + \overline{p}\boldsymbol{J}_1)^2 + \sum_{k=1}^{N}(\boldsymbol{f}_{0,k} + \overline{p}\boldsymbol{f}_{1,k})\left(\frac{\partial \boldsymbol{J}_0}{\partial \boldsymbol{x}_k} + \overline{p}\,\frac{\partial \boldsymbol{J}_1}{\partial \boldsymbol{x}_k}\right)\right]\Delta t^2 + O(\Delta t^3)$$

This true Jacobian $\boldsymbol{J}_{\text{disc}}(\boldsymbol{x}, p)$ thus traces out a curved, quadratic path parameterized by p. The following theorem formalizes the imposed limitation on affine discrete-time DSR models.

Theorem 2 (Geometric mismatch in discrete-time affine models) *Let the true continuous-time system have a strictly affine Jacobian dependence on p (Eq. 2) and consider a discrete-time DSR model with affine feature mapping (Eq. 3), i.e., $\boldsymbol{J}_{\text{model}}(l) = \hat{\boldsymbol{J}}_0 + l\hat{\boldsymbol{J}}_1$, where the feature l (or l_{dyn} in feature splitting models) scales as $l = g(p)$ (Eq. 4), g is three times differentiable and $g'(p_0) \neq 0$ at some p_0 in the training domain. Let the true $\boldsymbol{J}_{\text{disc}}(p, \boldsymbol{x})$ be three times differentiable around p_0, with $\frac{\partial}{\partial p}\boldsymbol{J}_{\text{disc}}(p_0, \boldsymbol{x}) \neq 0$ and suppose the model matches the true $\boldsymbol{J}_{\text{disc}}$ and $\frac{\partial}{\partial p}\boldsymbol{J}_{\text{disc}}$ at p_0. Then*

$$\frac{\mathrm{d}^2}{\mathrm{d}p^2}\boldsymbol{J}_{\text{model}}\big(g(p)\big)\Big|_{p=p_0} = \frac{g''(p_0)}{g'(p_0)}\,\frac{\partial}{\partial p}\boldsymbol{J}_{\text{disc}}(p_0, \boldsymbol{x}). \tag{18}$$

For fixed g, the model can therefore match the second-order parameter dependence if and only if

$$\frac{\partial^2}{\partial p^2}\boldsymbol{J}_{\text{disc}}(p_0, \boldsymbol{x}) = \frac{g''(p_0)}{g'(p_0)} \cdot \frac{\partial}{\partial p}\boldsymbol{J}_{\text{disc}}(p_0, \boldsymbol{x}) \tag{19}$$

If this condition does not hold, the affine-feature model incurs an irreducible local quadratic Jacobian mismatch. For the true continuous-time Jacobian, a sufficient mismatch condition (given $\boldsymbol{J}_1 \neq 0$ and small positive sampling intervals Δt) is

$$\boldsymbol{J}_1^2 + \sum_{k=1}^{N} f_{1,k} \frac{\partial \boldsymbol{J}_1}{\partial \boldsymbol{x}_k} \notin \operatorname{span}\{\boldsymbol{J}_1\}. \tag{20}$$

Thus, a nonlinear scalar feature can absorb curvature parallel to the first-order matrix direction, but it cannot represent curvature in any additional matrix direction.

Proof sketch (full proof in Appx. G.5.) Fix $\boldsymbol{x}$ and let $\boldsymbol{A} := \frac{\partial}{\partial p}\boldsymbol{J}_{\text{disc}}(p_0, \boldsymbol{x})$ and $\boldsymbol{B} := \frac{\partial^2}{\partial p^2}\boldsymbol{J}_{\text{disc}}(p_0, \boldsymbol{x})$. Matching the Jacobian of the true flow map and its first parameter derivative at p_0 gives $g'(p_0)\hat{\boldsymbol{J}}_1 = \boldsymbol{A}$. The second derivative available to the affine-feature model is thus $\frac{g''(p_0)}{g'(p_0)}\boldsymbol{A}$. Taylor expansion yields

$$\boldsymbol{J}_{\text{model}}(g(p)) - \boldsymbol{J}_{\text{disc}}(p, x) = \frac{\bar{p}^2}{2}\left[\frac{g''(p_0)}{g'(p_0)}\boldsymbol{A} - \boldsymbol{B}\right] + O(\bar{p}^3). \tag{21}$$

Hence the fixed-feature model matches the quadratic term if and only if $\boldsymbol{B} = \frac{g''(p_0)}{g'(p_0)}\boldsymbol{A}$. For the affine continuous-time system, the corrected small-Δt expansion gives

$$\boldsymbol{B} = \left[\boldsymbol{J}_1^2 + \sum_{k=1}^{N} f_{1,k}\frac{\partial \boldsymbol{J}_1}{\partial \boldsymbol{x}_k}\right]\Delta t^2 + O(\Delta t^3). \tag{22}$$

Such a mismatch cannot be resolved by the previously introduced solutions of sparsity constraints or feature splitting. However, understanding the structural truncation error allows us to derive an upper bound on the local extrapolation error using estimated in-domain curvature.

Corollary 2.1 (Local extrapolation error) *(Proof in Appx. G.6) Under Theorem 2, fix $\boldsymbol{x}$ and let p_0 be an upper boundary of the training domain. Let $\boldsymbol{E}(p) := \boldsymbol{J}_{\text{model}}(g(p)) - \boldsymbol{J}_{\text{disc}}(\boldsymbol{x}, p)$ satisfy $\boldsymbol{E}(p_0) = \boldsymbol{E}'(p_0) = 0$. Assume $\boldsymbol{E} \in C^3([p_0, p_0 + r_0])$ for some $r_0 > 0$ and $\|\boldsymbol{E}'''(p)\|_F \leq K$ for $p \in [p_0, p_0 + r_0]$. Let $\widehat{\boldsymbol{R}}_{\text{curv}}$ be an estimate of the boundary curvature residual from independent in-domain Jacobian estimates, with error at most η_R. Then, for $0 \leq r \leq r_0$,*

$$\|\boldsymbol{E}(p_0 + r)\|_F \leq \tfrac{1}{2}(\|\widehat{\boldsymbol{R}}_{\text{curv}}\|_F + \eta_R)r^2 + \tfrac{K}{6}r^3. \tag{23}$$

For $\epsilon > 0$, any r for which the right-hand side is at most ϵ gives $\|\boldsymbol{E}(p)\|_F \leq \epsilon$ throughout $[p_0, p_0 + r]$.

With this bound, we can guarantee a maximal Jacobian error for the extrapolation up to r_0, and thus an OOD range throughout which we can safely rely on the predicted extrapolation of the dynamics.

5 Conclusions

We identified three fundamental structural discrepancies that prevent existing feature-based hierarchical DSR approaches from extrapolating reliably across bifurcations: dense Jacobian coupling where real systems are typically sparse, entangled positional and dynamical scaling under single-feature parameterizations (as used in the recent literature [77, 41, 9, 32]), and geometric mismatches introduced by continuous-to-discrete time mappings. By introducing L_1 sparsity regularization and feature splitting (decoupling positional and dynamical features), we resolve the conflicting scaling requirements and restrict the model to physically plausible parameter dependence. We further derive an upper bound on the extrapolation error for discrete-time DSR models. With these amendments, extrapolation beyond the training regime and achieving true topological OODG becomes feasible *without explicit knowledge of the true control parameters* [70, 58, 43, 39], which we usually do not have for complex real-world systems (cf. Appx. E).

Limitations We made the assumption that control parameters enter in affine form (or at least in a single functional form) into the physical equations, which might not always be met. Likewise, the sparseness assumption might not be true for all systems. As stated in Theorem 2, there may also be more principled obstacles when moving from continuous to discrete time, which might be harder to resolve and require further research.

6 Acknowledgments

This work was supported by the German Research Foundation (DFG) through individual grants Du 354/15-1 (# 502196519) and Du 354/18-1 (# 567025973), and within Germanys Excellence Strategy EXC 2181/1 390900948 (STRUCTURES). Further support came from the Federal Ministry of Research, Technology, and Space (BMFTR) through the FEDORA (# 01EQ2403F) research consortium.

References

[1] Kathleen T. Alligood, Tim D. Sauer, and James A. Yorke. *Chaos: An Introduction to Dynamical Systems*. Textbooks in Mathematical Sciences. Springer, 1996. ISBN 978-0-387-94677-1 978-0-387-22492-3. doi: 10.1007/b97589.

[2] Victor M Martinez Alvarez, Rareş Roşca, and Cristian G Fălcuţescu. Dynode: Neural ordinary differential equations for dynamics modeling in continuous control. *arXiv preprint arXiv:2009.04278*, 2020.

[3] Peter Ashwin, Sebastian Wieczorek, Renato Vitolo, and Peter Cox. Tipping points in open systems: bifurcation, noise-induced and rate-dependent examples in the climate system. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370(1962):1166–1184, 2012. doi: 10.1098/rsta.2011.0306. URL `https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2011.0306`.

[4] Omri Azencot, N. Benjamin Erichson, Vanessa Lin, and Michael W. Mahoney. Forecasting Sequential Data using Consistent Koopman Autoencoders. In *Proceedings of the 37th International Conference on Machine Learning*, 2020. URL `http://arxiv.org/abs/2003.02236`.

[5] Friedrich L Bauer and Charles T Fike. Norms and exclusion theorems. *Numerische mathematik*, 2(1):137–141, 1960.

[6] Giancarlo Benettin, Luigi Galgani, Antonio Giorgilli, and Jean-Marie Strelcyn. Lyapunov characteristic exponents for smooth dynamical systems and for hamiltonian systems; a method for computing all of them. part 1: Theory. *Meccanica*, 15(1):9–20, 1980.

[7] Matthieu Blanke and Marc Lelarge. Interpretable meta-learning of physical systems. *arXiv preprint arXiv:2312.00477*, 2023.

[8] Manuel Brenner, Christoph Jürgen Hemmer, Zahra Monfared, and Daniel Durstewitz. Almost-linear rnns yield highly interpretable symbolic codes in dynamical systems reconstruction. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 36829–36868. Curran Associates, Inc., 2024.

[9] Manuel Brenner, Elias Weber, Georgia Koppe, and Daniel Durstewitz. Learning interpretable hierarchical dynamical systems models from time series data. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL `https://openreview.net/forum?id=Vp2OAxMs2s`.

[10] Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2019.

[11] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences USA*, 113(15):3932–3937, 2016. ISSN 0027-8424. doi: 10.1073/pnas.1517384113.

[12] Steven L. Brunton, Marko Budišić, Eurika Kaiser, and J. Nathan Kutz. Modern koopman theory for dynamical systems. *SIAM Review*, 64(2):229–340, 2022. doi: 10.1137/21M1401243. URL `https://doi.org/10.1137/21M1401243`.

[13] Mary Lucy Cartwright. Balthazar van der pol. *J. London Math. Soc*, 35(3):367–376, 1960.

[14] Kathleen Champion, Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences USA*, 116(45):22445–22451, 2019. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1906995116. URL `http://www.pnas.org/lookup/doi/10.1073/pnas.1906995116`.

[15] Kuan-Wei Chen and Ting-Yang Hsiao. Synchronization and hopf bifurcation in stuart–landau networks, 2026. URL https://arxiv.org/abs/2601.10234.

[16] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural Ordinary Differential Equations. In *Advances in Neural Information Processing Systems 31*, 2018. URL http://arxiv.org/abs/1806.07366.

[17] Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific machine learning through physics–informed neural networks: Where we are and whats next. *Journal of Scientific Computing*, 92(3):88, 2022.

[18] Daniel Durstewitz. Implications of synaptic biophysics for recurrent network dynamics and active memory. *Neural Networks*, 22(8):1189–1200, 2009. ISSN 08936080. doi: 10.1016/j.neunet.2009.07.016.

[19] Daniel Durstewitz. A state space approach for piecewise-linear recurrent neural networks for identifying computational dynamics from neural measurements. *PLoS Comput. Biol.*, 13(6): e1005542, 2017. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1005542.

[20] Daniel Durstewitz and Thomas Gabriel. Dynamical Basis of Irregular Spiking in NMDA-Driven Prefrontal Cortex Neurons. *Cerebral Cortex*, 17(4):894–908, April 2007. ISSN 1047-3211. doi: 10.1093/cercor/bhk044. URL https://doi.org/10.1093/cercor/bhk044.

[21] Daniel Durstewitz, Georgia Koppe, and Max Ingo Thurm. Reconstructing computational system dynamics from neural data with recurrent neural networks. *Nature Reviews. Neuroscience*, 24(11):693–710, November 2023. ISSN 1471-0048. doi: 10.1038/s41583-023-00740-7.

[22] Daniel Durstewitz, Christoph Jürgen Hemmer, Florian Hess, Charlotte Ricarda Doll, and Lukas Eisenmann. Position: Why a dynamical systems perspective is needed to advance time series modeling, 2026. URL https://arxiv.org/abs/2602.16864.

[23] Richard FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical journal*, 1(6):445–466, 1961.

[24] William Gilpin. Generative learning for nonlinear dynamics. *Nature Reviews Physics*, 6(3): 194–206, March 2024. ISSN 2522-5820. doi: 10.1038/s42254-024-00688-2. URL https://www.nature.com/articles/s42254-024-00688-2. Publisher: Nature Publishing Group.

[25] Peter Grassberger and Itamar Procaccia. Measuring the strangeness of strange attractors. *Physica D: nonlinear phenomena*, 9(1-2):189–208, 1983.

[26] John Guckenheimer and Philip Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, volume 42 of *Applied Mathematical Sciences*. Springer, New York, NY, 1983. ISBN 978-1-4612-7020-1 978-1-4612-1140-2. doi: 10.1007/978-1-4612-1140-2. URL http://link.springer.com/10.1007/978-1-4612-1140-2.

[27] Niclas Alexander Göring, Florian Hess, Manuel Brenner, Zahra Monfared, and Daniel Durstewitz. Out-of-Domain Generalization in Dynamical Systems Reconstruction. In *Proceedings of the 41st International Conference on Machine Learning*, pages 16071–16114. PMLR, July 2024. URL https://proceedings.mlr.press/v235/goring24a.html. ISSN: 2640-3498.

[28] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

[29] Christoph Jürgen Hemmer and Daniel Durstewitz. True zero-shot inference of dynamical systems preserving long-term statistics. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=RE97LT26w8.

[30] Florian Hess, Zahra Monfared, Manuel Brenner, and Daniel Durstewitz. Generalized Teacher Forcing for Learning Chaotic Dynamics. In *Proceedings of the 40th International Conference on Machine Learning*, pages 13017–13049. PMLR, July 2023. URL https://proceedings.mlr.press/v202/hess23a.html. ISSN: 2640-3498.

[31] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8): 17351780, nov 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735.

[32] In Huh, Changwook Jeong, and Muhammad Alam. Context-informed neural ODEs unexpectedly identify broken symmetries: Insights from the poincaréhopf theorem. In Aarti Singh,

Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 26311–26337. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/huh25a.html.

[33] Ruoxi Jiang, Peter Y Lu, Elena Orlova, and Rebecca Willett. Training neural operators to preserve invariant measures of chaotic attractors. *Advances in Neural Information Processing Systems*, 36:27645–27669, 2023.

[34] Viktor K. Jirsa, William C. Stacey, Pascale P. Quilichini, Anton I. Ivanov, and Christophe Bernard. On the nature of seizure dynamics. *Brain*, 137(8):2210–2230, August 2014. doi: 10.1093/brain/awu133.

[35] Holger Kantz and Thomas Schreiber. *Nonlinear time series analysis*, volume 7. Cambridge university press, 2004.

[36] Daniel Karlsson and Olle Svanström. Modelling Dynamical Systems Using Neural Ordinary Differential Equations, 2019. URL https://hdl.handle.net/20.500.12380/256887.

[37] William Ogilvy Kermack and Anderson G McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, 115(772):700–721, 1927.

[38] Jason Z. Kim, Zhixin Lu, Erfan Nozari, George J. Pappas, and Danielle S. Bassett. Teaching recurrent neural networks to infer global temporal structure from local examples. *Nature Machine Intelligence*, 3(4):316–323, April 2021. ISSN 2522-5839. doi: 10.1038/s42256-021-00321-2. URL https://www.nature.com/articles/s42256-021-00321-2.

[39] Timothy D. Kim, Thomas Z. Luo, Jonathan W. Pillow, and Carlos Brody. Inferring latent dynamics underlying neural population activity via neural differential equations. In *International Conference on Machine Learning*, pages 5551–5561. PMLR, 2021. URL http://proceedings.mlr.press/v139/kim21h.html. ISSN: 2640-3498.

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

[41] Matthieu Kirchmeyer, Yuan Yin, Jérémie Donà, Nicolas Baskiotis, Alain Rakotomamonjy, and Patrick Gallinari. Generalizing to new physical systems via context-informed dynamics model. In *International Conference on Machine Learning*, pages 11283–11301. PMLR, 2022.

[42] Joon-Hyuk Ko, Hankyul Koh, Nojun Park, and Wonho Jhe. Homotopy-based training of neuralodes for accurate dynamics discovery. *Advances in Neural Information Processing Systems*, 36:64725–64752, 2023.

[43] Ling-Wei Kong, Hua-Wei Fan, Celso Grebogi, and Ying-Cheng Lai. Machine learning prediction of critical transition and system collapse. *Physical Review Research*, 3(1):013090, January 2021. ISSN 2643-1564. doi: 10.1103/PhysRevResearch.3.013090. URL https://link.aps.org/doi/10.1103/PhysRevResearch.3.013090.

[44] Ling-Wei Kong, Yang Weng, Bryan Glaz, Mulugeta Haile, and Ying-Cheng Lai. Reservoir computing as digital twins for nonlinear dynamical systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 33(3):033111, March 2023. ISSN 1054-1500, 1089-7682. doi: 10.1063/5.0138661.

[45] Yuri A. Kuznetsov. *Elements of Applied Bifurcation Theory (2nd Ed.)*. Springer-Verlag, Berlin, Heidelberg, 1998. ISBN 0387983821.

[46] Daniel Köglmayr and Christoph Räth. Extrapolating tipping points and simulating nonstationary dynamics of complex systems using efficient machine learning. *Scientific Reports*, 14(1):507, January 2024. ISSN 2045-2322. doi: 10.1038/s41598-023-50726-9. URL https://www.nature.com/articles/s41598-023-50726-9.

[47] Soon Hoe Lim, Ludovico Theo Giorgini, Woosok Moon, and J. S. Wettlaufer. Predicting critical transitions in multiscale dynamical systems using reservoir computing. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(12):123126, December 2020. ISSN 1054-1500, 1089-7682. doi: 10.1063/5.0023764.

[48] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *Proceedings of the Eighth International Conference on Learning Representations*, 2020. URL https://iclr.cc/virtual_2020/poster_rkgz2aEKDr.html.

[49] Edward N Lorenz. Deterministic nonperiodic flow. *Journal of atmospheric sciences*, 20(2): 130–141, 1963.

[50] Edward N Lorenz. Predictability: A problem partly solved. In *Proc. Seminar on predictability*, volume 1, 1996.

[51] Alfred J Lotka. Analytical note on certain rhythmic relations in organic systems. *Proceedings of the National Academy of Sciences*, 6(7):410–415, 1920.

[52] Jonas Mikhaeil, Zahra Monfared, and Daniel Durstewitz. On the difficulty of learning chaotic dynamics with RNNs. *Advances in Neural Information Processing Systems*, 35:11297–11312, December 2022.

[53] James D. Murray. *Mathematical Biology: I. An Introduction*, volume 17 of *Interdisciplinary Applied Mathematics*. Springer, New York, 3 edition, 2002. ISBN 978-0-387-95223-9.

[54] James D. Murray. *Mathematical Biology: II. Spatial Models and Biomedical Applications*, volume 18 of *Interdisciplinary Applied Mathematics*. Springer, New York, 3 edition, 2003. ISBN 978-0-387-95228-4.

[55] Ilan Naiman and Omri Azencot. A Koopman Approach to Understanding Sequence Neural Models. *arXiv:2102.07824 [cs, math]*, October 2021. URL http://arxiv.org/abs/2102.07824. arXiv: 2102.07824.

[56] Zachary G Nicolaou, Guanyu Huo, Yihui Chen, Steven L Brunton, and J Nathan Kutz. Data-driven discovery and extrapolation of parameterized pattern-forming dynamics. *Physical Review Research*, 5(4):L042017, 2023.

[57] Roussel Desmond Nzoyem, Grant Stevens, Amarpal Sahota, David A. W. Barton, and Tom Deakin. MixER: Better Mixture of Experts Routing for Hierarchical Meta-Learning. In *First Workshop on Scalable Optimization for Efficient and Adaptive Foundation Models*, March 2025.

[58] Shirin Panahi and Ying-Cheng Lai. Adaptable reservoir computing: A paradigm for model-free data-driven prediction of critical transitions in nonlinear dynamical systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 34(5), 2024.

[59] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[60] Dhruvit Patel and Edward Ott. Using machine learning to anticipate tipping points and extrapolate to post-tipping dynamics of non-stationary dynamical systems. *Chaos (Woodbury, N.Y.)*, 33(2):023143, February 2023. ISSN 1089-7682. doi: 10.1063/5.0131787.

[61] Jaideep Pathak, Zhixin Lu, Brian R. Hunt, Michelle Girvan, and Edward Ott. Using Machine Learning to Replicate Chaotic Attractors and Calculate Lyapunov Exponents from Data. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(12):121102, December 2017. ISSN 1054-1500, 1089-7682. doi: 10.1063/1.5010300. URL http://arxiv.org/abs/1710.07313. arXiv: 1710.07313.

[62] Lawrence Perko. *Differential equations and dynamical systems*. Number 7 in Texts in applied mathematics. Springer, New York, 3rd edition, 2001. ISBN 978-0-387-95116-4.

[63] Jason A Platt, Stephen G Penny, Timothy A Smith, Tse-Chun Chen, and Henry DI Abarbanel. A systematic exploration of reservoir computing for forecasting complex spatiotemporal dynamics. *Neural Networks*, 153:530–552, 2022.

[64] Jason A Platt, Stephen G Penny, Timothy A Smith, Tse-Chun Chen, and Henry DI Abarbanel. Constraining chaos: Enforcing dynamical invariants in the training of reservoir computers. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 33(10), 2023.

[65] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, February 2019. ISSN

00219991. doi: 10.1016/j.jcp.2018.10.045. URL https://linkinghub.elsevier.com/retrieve/pii/S0021999118307125.

[66] Dominik Schmidt, Georgia Koppe, Zahra Monfared, Max Beutelspacher, and Daniel Durstewitz. Identifying nonlinear dynamical systems with multiple time scales and long-range dependencies. In *Proceedings of the 9th International Conference on Learning Representations*, 2021. URL http://arxiv.org/abs/1910.03471.

[67] Evgeny Evgenievich SEL'KOV. Self-oscillations in glycolysis 1. a simple kinetic model. *European Journal of Biochemistry*, 4(1):79–86, 1968.

[68] James Theiler. Statistical precision of dimension estimators. *Physical Review A*, 41(6):3038, 1990.

[69] SS Vallender. Calculation of the wasserstein distance between probability distributions on the line. *Theory of Probability & Its Applications*, 18(4):784–786, 1974.

[70] Eva Van Tegelen, George Van Voorn, Ioannis N. Athanasiadis, and Peter Van Heijster. Neural ordinary differential equations for learning and extrapolating system dynamics across bifurcations. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 35(10):101103, October 2025. ISSN 1054-1500, 1089-7682. doi: 10.1063/5.0288264.

[71] Ayesha Vermani, Josue Nassar, Hyungju Jeon, Matthew Dowling, and Il Memming Park. Meta-dynamical state space models for integrative neural data analysis. *arXiv preprint arXiv:2410.05454*, 2024.

[72] Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

[73] Pantelis R Vlachas, Wonmin Byeon, Zhong Y Wan, Themistoklis P Sapsis, and Petros Koumoutsakos. 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.

[74] Johannes Von Oswald, Christian Henning, Benjamin F Grewe, and João Sacramento. Continual learning with hypernetworks. *arXiv preprint arXiv:1906.00695*, 2019.

[75] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S. Yu. Generalizing to Unseen Domains: A Survey on Domain Generalization. *IEEE Transactions on Knowledge and Data Engineering*, 35(8): 8052–8072, August 2023. ISSN 1558-2191. doi: 10.1109/TKDE.2022.3178128. URL https://ieeexplore.ieee.org/document/9782500. Conference Name: IEEE Transactions on Knowledge and Data Engineering.

[76] Rui Wang, Yihe Dong, Sercan Ö Arik, and Rose Yu. Koopman Neural Forecaster for Time Series with Temporal Distribution Shifts, October 2022. URL http://arxiv.org/abs/2210.03675. arXiv:2210.03675 [cs, stat].

[77] Yuan Yin, Ibrahim Ayed, Emmanuel de Bézenac, Nicolas Baskiotis, and Patrick Gallinari. Leads: Learning dynamical systems that generalize across environments. *Advances in Neural Information Processing Systems*, 34:7561–7573, 2021.

[78] David Zipser. Recurrent network model of the neural mechanism of short-term active memory. *Neural Computation*, 3(2):179–193, 1991.

A Additional theoretical results

A.1 Systems with multiple parameters and nonlinear dependencies

The analysis in Sect. 4.2 and 4.3, which established the structural limitations of affine hierarchical models for the scalar parameter case $p \in \mathbb{R}$, extends directly to systems depending on multiple parameters $\boldsymbol{p} = (p_1, \ldots, p_n)^T$ of the form

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{p}) = \boldsymbol{f}_0(\boldsymbol{x}) + \sum_{i=1}^{n} p_i \boldsymbol{f}_i(\boldsymbol{x}), \tag{24}$$

where each p_i enters the system multiplicatively, scaling a corresponding vector field component $\boldsymbol{f}_i(\boldsymbol{x})$. To see this, fix any index k and rewrite the system as

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{p}) = \tilde{\boldsymbol{f}}_k(\boldsymbol{x}, \boldsymbol{p}_k^{\perp}) + p_k \boldsymbol{f}_k(\boldsymbol{x}), \tag{25}$$

where $\boldsymbol{p}_k^{\perp} := (p_1, \ldots, p_{k-1}, p_{k+1}, \ldots, p_n)^T$ collects all parameters except p_k, and $\tilde{\boldsymbol{f}}_k(\boldsymbol{x}, \boldsymbol{p}_k^{\perp}) := \boldsymbol{f}_0(\boldsymbol{x}) + \sum_{i \neq k} p_i \boldsymbol{f}_i(\boldsymbol{x})$ groups all terms independent of p_k. The resulting expression is affine in p_k, with the remaining terms playing the role of a modified baseline vector field. Since this argument holds for any k, all previous derivations apply to each parameter individually, and hence to the full multi-parameter system.

More generally, systems in which parameters appear nonlinearly can be brought into the same affine form by a reparameterization. Consider, for example,

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{p}) = \boldsymbol{f}_0(\boldsymbol{x}) + p_1^{\beta} \boldsymbol{f}_1(\boldsymbol{x}) + p_2 p_3 \boldsymbol{f}_2(\boldsymbol{x}). \tag{26}$$

Defining $q_1 := p_1^{\beta}$ and $q_2 := p_2 p_3$ gives

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{q}) = \boldsymbol{f}_0(\boldsymbol{x}) + q_1 \boldsymbol{f}_1(\boldsymbol{x}) + q_2 \boldsymbol{f}_2(\boldsymbol{x}), \tag{27}$$

which is linear in the transformed parameters $\boldsymbol{q}$. The model therefore effectively learns an embedding $\boldsymbol{l}(\boldsymbol{q})$ rather than $\boldsymbol{l}(\boldsymbol{p})$. Two important limitations arise in this context, however. First, if the reparameterization mapping $\boldsymbol{p} \mapsto \boldsymbol{q}$ is non-injective (e.g., $q_1 = p_1^2$), distinct physical parameter configurations can map to identical dynamics, which breaks the unique identifiability of the original control parameters and violates the injectivity assumed in Eq. 4. Second, when a single original parameter p_i appears in multiple distinct algebraic forms within the system (e.g., when both p_1 and p_1^2 enter separately), each such form requires its own feature dimension. This artificially expands the effective parameter space beyond what the true linear parameter dependence requires. As established in Proposition 2, because standard ERM strictly optimizes for local training loss without structural priors, it leaves these excess dimensions unconstrained, allowing them to retain non-zero weights. Extrapolating outside the training domain can then cause an unbounded growth of the model's spectral radius, potentially introducing spurious bifurcations that are absent from the true system.

A.2 Compatibility conditions for hidden-layer models

The residual-based argument of Theorem 1 extends to hidden-layer architectures by using their corresponding partial feature derivative. Consider the network

$$\boldsymbol{H}_{\boldsymbol{\theta}}(\boldsymbol{z}, l) = \boldsymbol{W}_1(l)\, \varphi(\boldsymbol{W}_2(l)\boldsymbol{z} + \boldsymbol{h}_2(l)) + \boldsymbol{h}_1(l), \tag{28}$$

used either as an RNN transition map or as a Neural ODE vector field. All weights and biases depend affinely on the scalar feature, with $\boldsymbol{W}_i(l) = \boldsymbol{W}_{i,c} + l\boldsymbol{W}_{i,v}$ and $\boldsymbol{h}_i(l) = \boldsymbol{h}_{i,c} + l\boldsymbol{h}_{i,v}$. Define the preactivation $\boldsymbol{u}(\boldsymbol{z}, l) := \boldsymbol{W}_2(l)\boldsymbol{z} + \boldsymbol{h}_2(l)$ and the indicator matrix $\boldsymbol{D}(\boldsymbol{z}, l) := \mathrm{diag}(\varphi'(\boldsymbol{u}(\boldsymbol{z}, l)))$. For continuously differentiable activations, the state Jacobian and the partial feature derivative are

$$\boldsymbol{J}_{\mathrm{model}}(\boldsymbol{z}, l) = \boldsymbol{W}_1(l)\boldsymbol{D}(\boldsymbol{z}, l)\boldsymbol{W}_2(l), \tag{29}$$

$$\begin{aligned} \boldsymbol{b}_l(\boldsymbol{z}, l) &:= \frac{\partial}{\partial l} \boldsymbol{H}_{\boldsymbol{\theta}}(\boldsymbol{z}, l)\Big|_{\boldsymbol{z}} \\ &= \boldsymbol{W}_{1,v}\varphi(\boldsymbol{u}) + \boldsymbol{W}_1(l)\boldsymbol{D}(\boldsymbol{z}, l)\big(\boldsymbol{W}_{2,v}\boldsymbol{z} + \boldsymbol{h}_{2,v}\big) + \boldsymbol{h}_{1,v}. \end{aligned} \tag{30}$$

For piecewise-linear activations φ, the same expressions apply on intervals of l for which the preactivation $\boldsymbol{u}(\boldsymbol{z}^*(l), l)$ along the FP path $\boldsymbol{z}^*(l)$ remains within the same linear subregion of φ, i.e. the indicator matrix $\boldsymbol{D}(\boldsymbol{z}, l)$ is constant.

Under the assumptions of Theorem 1, with the model architecture replaced by Eq. 28, exact FP matching throughout the parameter interval holds if and only if the FP is matched at some parameter value p_0 and

$$\frac{\mathrm{d}\boldsymbol{x}^*_{\mathrm{true}}}{\mathrm{d}p} = \begin{cases} -g'(p)\, \boldsymbol{J}_{\mathrm{true}}(p)^{-1} \boldsymbol{b}_l(\boldsymbol{x}^*_{\mathrm{true}}(p), g(p))\,, & \text{Neural ODE,} \\ g'(p)\, \big(\boldsymbol{I} - \boldsymbol{J}_{\mathrm{true}}(p)\big)^{-1} \boldsymbol{b}_l(\boldsymbol{x}^*_{\mathrm{true}}(p), g(p))\,, & \text{RNN.} \end{cases} \tag{31}$$

Now, differentiating the FP residual w.r.t. p, evaluated at the true FP $\boldsymbol{x}^*_{\mathrm{true}}(p)$ gives

$$\boldsymbol{r}'_{\mathrm{NODE}}(p) = \boldsymbol{J}_{\mathrm{true}}(p)\boldsymbol{x}^{*\prime}_{\mathrm{true}}(p) + g'(p)\boldsymbol{b}_l(\boldsymbol{x}^*_{\mathrm{true}}(p), g(p))\,, \tag{32}$$

$$\boldsymbol{r}'_{\mathrm{RNN}}(p) = \big(\boldsymbol{I} - \boldsymbol{J}_{\mathrm{true}}(p)\big)\boldsymbol{x}^{*\prime}_{\mathrm{true}}(p) - g'(p)\boldsymbol{b}_l(\boldsymbol{x}^*_{\mathrm{true}}(p), g(p))\,. \tag{33}$$

Given invertibility of $\boldsymbol{J}_{\mathrm{true}}$ or $(\boldsymbol{I} - \boldsymbol{J}_{\mathrm{true}})$ respectively, these expressions show that the residual derivative vanishes throughout the interval only if Eq. 31 holds throughout the interval and the FP matches at some parameter value p_0. Then, the residual is constant and matching at one parameter value implies FP matching throughout the interval, exactly as in the proof of Theorem 1.

For the minimal architecture in the main text, $\boldsymbol{H}_{\boldsymbol{\theta}}(\boldsymbol{z}, l) = \boldsymbol{W}(l)\psi(\boldsymbol{z}) + \boldsymbol{h}(l)$, the feature derivative reduces to

$$\frac{\partial}{\partial l}\boldsymbol{H}_{\boldsymbol{\theta}}(\boldsymbol{z}, l) = \boldsymbol{W}_v\psi(\boldsymbol{z}) + \boldsymbol{h}_v, \tag{34}$$

recovering Eq. 11. For the shPLRNN used in our experiments, the additional term $\boldsymbol{A}(l)\boldsymbol{z}$ contributes $\boldsymbol{A}(l)$ to the state Jacobian and $\boldsymbol{A}_v\boldsymbol{z}$ to the partial feature derivative. The same residual argument therefore applies, subject to the corresponding assumptions.

B Additional empirical results

B.1 Interplay between feature splitting and regularization

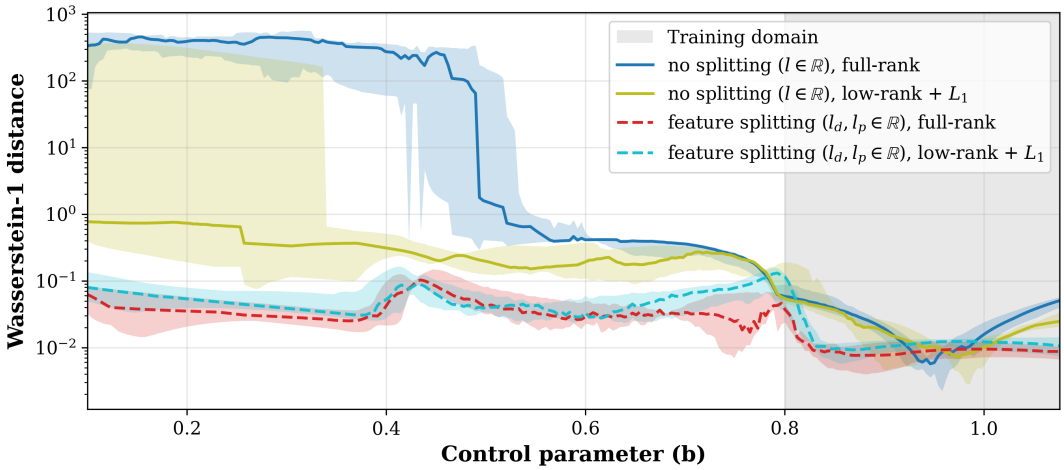


Figure 5: Selkov system: combining feature splitting and low-rank-&-sparsity-regularization. Wasserstein-1 distances between true and reconstructed bifurcation diagrams for hierarchical Neural ODE models with none, one, or both of the modifications of the vanilla hierarchical model. Training data lies in the fixed-point regime; OOD extrapolation covers the limit-cycle regime and the second Hopf bifurcation back to FPs. Solid and dashed lines show medians over 10 runs; envelopes indicate the interquartile range. The gray shaded region indicates the training domain.

Here we examine how feature splitting and low-rank-&-sparsity-regularization interact when combined. For this, two benchmark systems with different extrapolation regimes were examined. We find that for the Selkov system (Fig. 5) feature splitting is the dominant factor: it improves both

in-domain stability near the training boundaries and OOD performance across the full extrapolation range. Adding regularization yields no consistent additional gain. On the Lorenz-63 system extrapolated into the chaotic regime (Fig. 6), the picture reverses: feature splitting alone is insufficient to prevent the full-rank model from diverging, and the combination of regularization and splitting produces the lowest OOD error. Taken together, these results suggest that the two interventions address complementary failure modes (see Section 4).

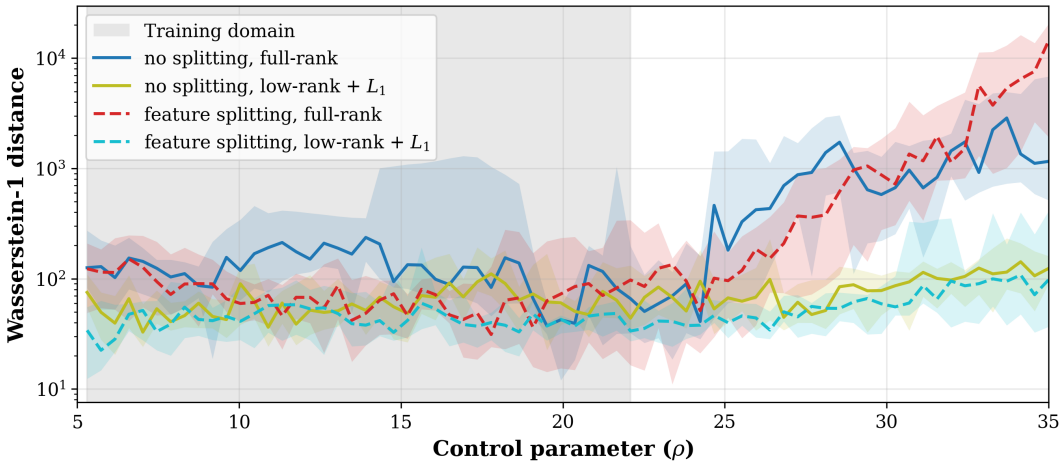


Figure 6: Lorenz-63 system: combining feature splitting and low-rank-&-sparsity-regularization. Same setup as in Fig. 5, with training data in the two-fixed-point regime and OOD extrapolation across the transition to chaos. Solid and dashed lines show medians over 10 runs; envelopes indicate the interquartile range. The gray shaded region indicates the training domain.

B.2 shPLRNN models trained on Lorenz-63

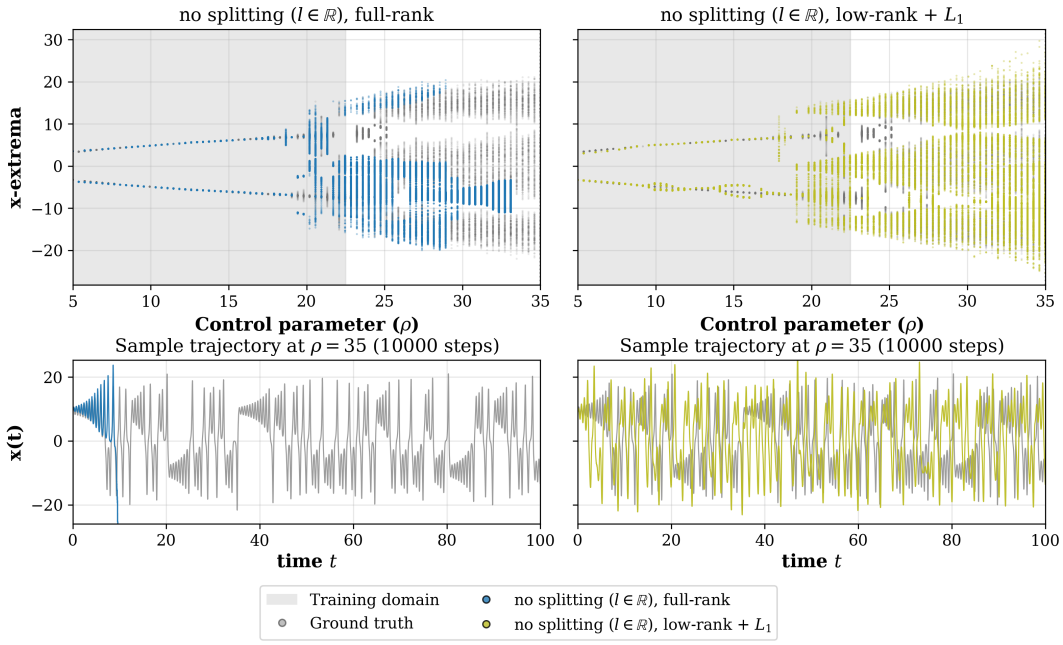


Figure 7: shPLRNN models trained on Lorenz-63, without (left) and with (right) low-rank-&-sparsity regularization. **Top:** reconstructed bifurcation diagrams. **Bottom:** sample trajectories at $\rho = 35$. The regularized model captures the chaotic regime, while the unconstrained model fails to reconstruct it.

A comparison of shPLRNN models with and without low-rank-&-sparsity regularization on the Lorenz-63 DS (Appx. D.1) is presented in Fig. 7, which provides the bifurcation diagram and a sampled trajectory of $10,000$ time steps, with median Wasserstein-1 distance over $n = 10$ trained models per setting. The regularized model recovers the chaotic attractor even at $\rho = 35$, well outside the training domain, whereas the unconstrained model breaks down and converges to a FP, highlighting the need for the low-rank-&-sparsity regularization.

B.3 shPLRNN models trained on Rössler attractor

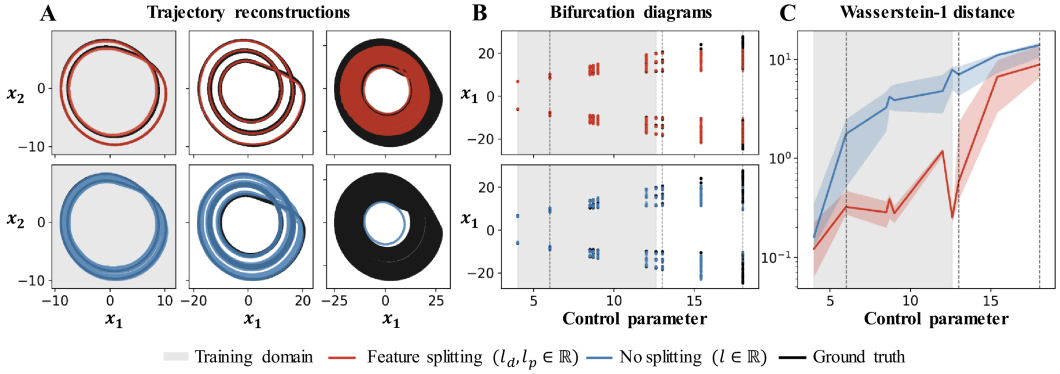


Figure 8: Reconstruction of period-doubling route to chaos in the Rössler system using shPLRNN models with and without feature splitting. **(A)** Representative trajectory reconstructions at three distinct control parameter values marked by vertical dashed lines in panels B and C. **(B)** Reconstructed bifurcation diagrams (runs with lowest overall Wasserstein-1 distance for each setting). **(C)** Wasserstein-1 distance between true and reconstructed bifurcation diagrams (Appx. F) for models trained with and without feature splitting. Solid lines are medians across 10 runs and error bands indicate the interquartile range. The shaded area indicates the training domain.

The shPLRNN models with and without feature splitting were trained on the period-doubling cascade of the Rössler attractor and extrapolated into the chaotic regime. Details about training and test sets are as described in Appx. D.3, the results of the different training runs are shown in Fig. 8. The models without splitting show significantly worse reconstructions of the bifurcation diagram according to the Wasserstein-1 distance than those with feature splitting (panel C). Although this difference on average is less pronounced in the OOD regime, well-trained feature splitting models can correctly extrapolate to the chaotic attractor (for control parameter $c = 18$ in panel A), while *all* vanilla models fall back to period-1 limit cycles.

B.4 OODG verification via maximum Lyapunov exponent and correlation dimension

To further support our results on topological OODG, we investigated models trained on the Selkov system using two additional metrics, namely the maximum Lyapunov exponent (Appx. F.3) and the correlation dimension (Appx. F.2). For the chosen in-domain and OOD control parameter settings the system has either limit cycle or fixed point attractors, hence the maximum Lyapunov exponent is either exactly zero or negative, and the correlation dimension is either exactly one or exactly zero. Figure 9 shows these additional results for the models already introduced in Figure 4. As for the Wasserstein-1 distance on bifurcation graphs, we observe that models with feature splitting outperform those without.

B.5 Comparison to other model architectures

We compared the performance of hierarchical models with feature splitting to that of other architectures developed for similar types of settings, namely control-parameter aware SINDy (SINDyCP) [56] and Next Generation Reservoir Computers (NG-RC) [46]. As shown in Figure 10 for the Selkov system (Appx. D.2), the best OOD results are achieved by the hierarchical models with feature splitting (shPLRNN or Neural ODE) and by SINDyCP. NG-RC and hierarchical models without splitting instead, while performing similarly in-domain, were significantly worse in the OOD regime. While

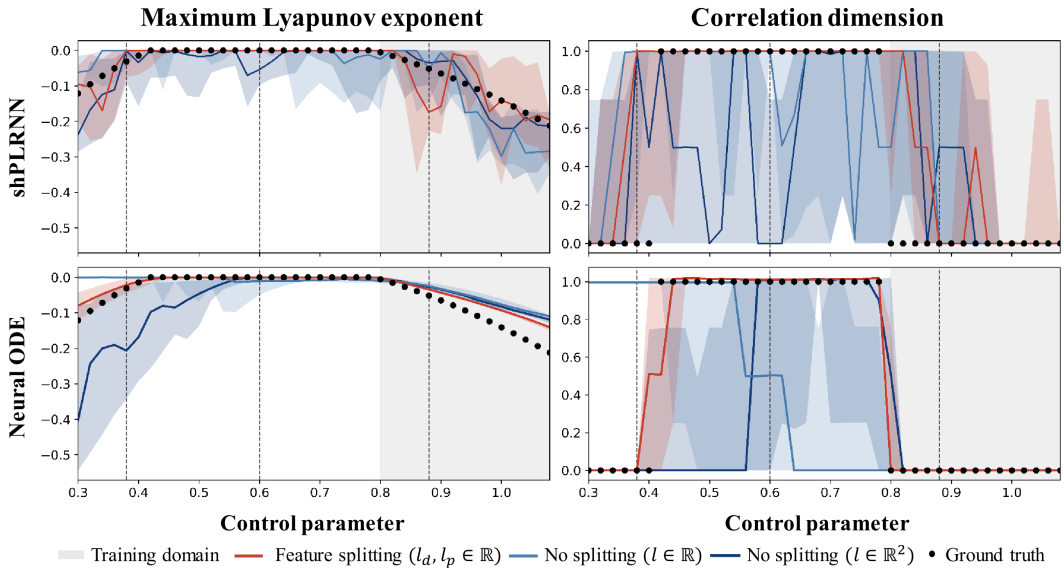


Figure 9: Selkov system: Reconstruction quality of shPLRNN (top row) and Neural ODE (bottom row) in terms of (**A**) maximum Lyapunov exponent, and (**B**) correlation dimension. Models are the same as used for Figure 4, and the vertical dashed lines indicate the same control parameter values highlighted there. Solid lines are the median results over 10 runs, envelopes indicate interquartile ranges.

it thus appears as if SINDyCP could keep up in performance with our feature splitting approach, crucially, in contrast to ours, both SINDyCP and NG-RC *require knowledge of the true control parameter of each trajectory during training*, i.e. information we commonly do not have for complex real world systems, and which our approach is designed to extract from the data as well.

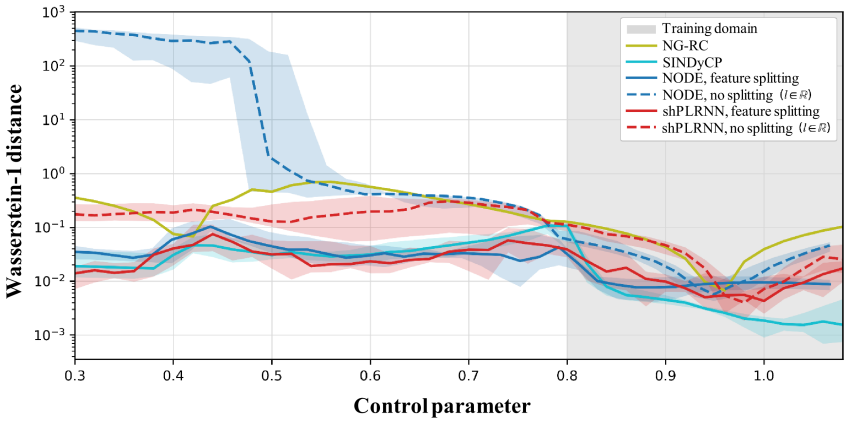


Figure 10: Selkov system: Reconstruction performance of different model architectures in terms of Wasserstein-1 distance between true and model-generated bifurcation diagrams. Note that SINDyCP and NG-RC, in contrast to the hierarchical models, assume the control parameters *to be known* in training.

B.6 Effect of regularization in the case of highly coupled systems

Next, we evaluated the feature splitting and regularization approaches on a highly coupled system [15] which has a dense dependence on a control parameter. As expected, we observe that L_1 regularization and low-rank constraints decrease performance. Nevertheless, feature splitting still proves to be advantageous.

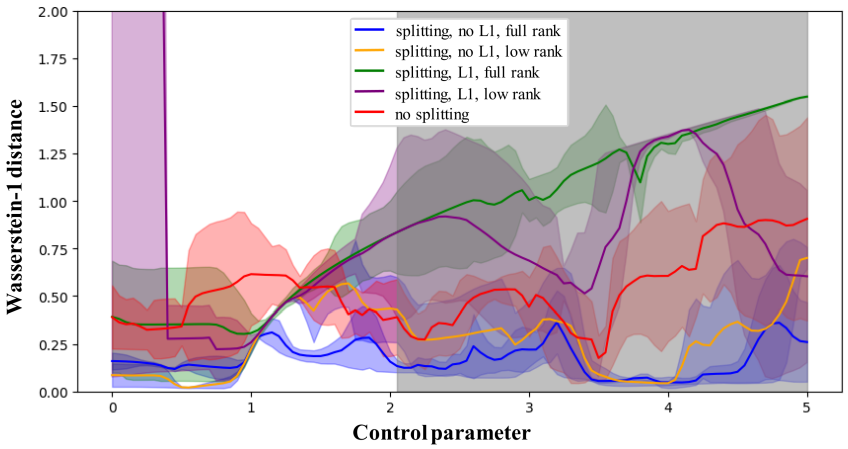


Figure 11: Highly coupled Stuart–Landau system: Reconstruction quality in terms of Wasserstein-1 distance between true and model-generated bifurcation diagrams for different combinations of feature-splitting, low-rank parameterization & L1-regularization.

B.7 Empirical data example: membrane potential recordings from cortical neurons

We trained and evaluated hierarchical models with and without feature splitting on membrane potential recordings from cortical neurons transitioning between irregular spiking and spike-burst regimes. As discussed in [20, 18], the irregular spiking regime bears signatures of chaos (Fig. 12, bottom left), while the bursting activity regime (Fig. 12, bottom right) might be closer to a multi-period limit cycle. These transitions are likely regulated by the concentration of the neurotransmitter NMDA in the bath medium [20], but the precise control parameter is not known.

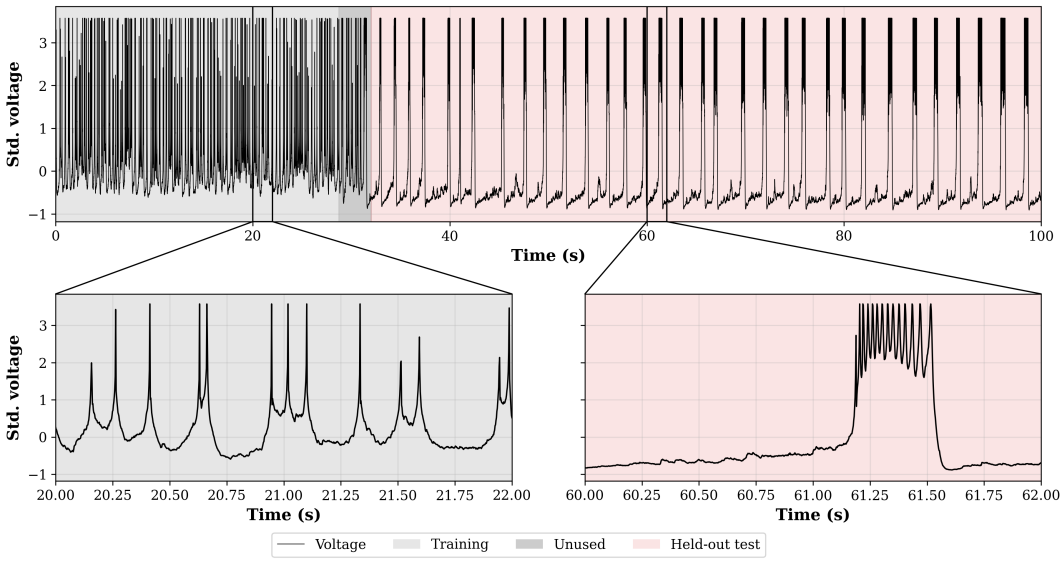


Figure 12: Standardized membrane potential of a cortical neuron transitioning from irregular spiking (training data) to bursting (evaluation data). Data from [20, 18].

Models were trained on the initial irregular-spiking regime, with time-delay embedding ($m = 6$ and $\tau = 4$) and a separate feature vector $\boldsymbol{l}_\tau \in \mathbb{R}$ for each time bin τ of size 1.2 s (resulting in 25 bins for the training data) to accommodate variation in the underlying control parameter over time, i.e. the control parameter was assumed to be approximately constant across timescales of 1.2 s but to vary across larger timescales. For evaluation, segments of 125 time steps from the bursting regime (again time-delay embedded with the same settings) were used to fine-tune the feature vectors $\boldsymbol{l}_{\text{extrap.}}$. These feature vectors were then held fixed, and prediction MSE over 100 time steps was

evaluated on 64 separate sequences from the same bursting regime. Models with feature splitting achieved a lower median MSE of 0.87 (quartiles: 0.80–0.99) than models without feature splitting (median: 1.14; quartiles: 0.98–1.69). While this empirical example thus confirms the advantages of our approach, it also highlights that complex, partially observed, multi-timescale, noisy real-world scenarios with potentially multiple underlying changes remain challenging.

C Architectures of the used models

C.1 Discrete-time model

For our experiments we employed the shallow piecewise-linear RNN (shPLRNN; [30]), given by Eq. 28 (Appx. A.2) with a linear term added, which helps for mapping multiple timescales in the data and stabilizing loss gradients [66]:

$$\boldsymbol{z}_{t+1} = \boldsymbol{A}(l)\boldsymbol{z}_t + \boldsymbol{W}_1(l)\,\varphi(\boldsymbol{W}_2(l)\boldsymbol{z}_t + \boldsymbol{h}_2(l)) + \boldsymbol{h}_1(l), \tag{35}$$

where $\boldsymbol{z}_t \in \mathbb{R}^M$ denotes the latent state at time t, and φ is the ReLU activation function. The matrix $\boldsymbol{A}(l) \in \mathbb{R}^{M\times M}$ is diagonal, i.e. $\boldsymbol{A}(l) = \mathrm{diag}(\boldsymbol{a}(l))$. The weight matrices are $\boldsymbol{W}_1(l) \in \mathbb{R}^{M\times H}$ and $\boldsymbol{W}_2(l) \in \mathbb{R}^{H\times M}$, with corresponding biases $\boldsymbol{h}_1(l) \in \mathbb{R}^M$ and $\boldsymbol{h}_2(l) \in \mathbb{R}^H$. Here, M denotes the latent dimensionality and H the hidden dimensionality.

The latent states are related to the observations via

$$\boldsymbol{x}_t = g(\boldsymbol{z}_t)\,, \tag{36}$$

where a linear mapping $g(\boldsymbol{z}_t) = \boldsymbol{E}\boldsymbol{z}_t$ was used here.

Without feature splitting, all parameters are constructed according to Eq. 3 in Section 3.3, using a single feature l:

$$\boldsymbol{A}(l) = \boldsymbol{A}_c + l\,\boldsymbol{A}_v, \qquad \boldsymbol{W}_1(l) = \boldsymbol{W}_{1,c} + l\,\boldsymbol{W}_{1,v}, \qquad \dots \tag{37}$$

Likewise, with feature splitting, one feature l_{dyn} is used for all parameters that enter the Jacobian and thus contribute to the local growth rates, while another feature l_{pos} is used for parameters only translating the flow without changing its local growth rates (Eq. 16). Thus, the positional feature only affects the first bias term $\boldsymbol{h}_1$, while all other parameters depend on the dynamical feature:

$$\boldsymbol{h}_1(l_{\text{pos}}) = \boldsymbol{h}_{1,c} + l_{\text{pos}}\,\boldsymbol{h}_{1,v}, \qquad \boldsymbol{A}(l_{\text{dyn}}) = \boldsymbol{A}_c + l_{\text{dyn}}\,\boldsymbol{A}_v, \qquad \dots \tag{38}$$

C.2 Continuous-time

The continuous-time model used in our empirical evaluations directly parameterizes the vector field as

$$\dot{\boldsymbol{z}}_t = \boldsymbol{W}_1(l)\boldsymbol{\psi}(\boldsymbol{W}_2(l)\boldsymbol{z}_t + \boldsymbol{h}_2(l)) + \boldsymbol{h}_1(l) \tag{39}$$

where $\boldsymbol{z}_t \in \mathbb{R}^M$ denotes the latent state at time t, and $\boldsymbol{\psi}$ is the $\tanh$ activation function. The weight matrices are $\boldsymbol{W}_1(l) \in \mathbb{R}^{M\times H}$ and $\boldsymbol{W}_2(l) \in \mathbb{R}^{H\times M}$, with the corresponding biases $\boldsymbol{h}_1(l) \in \mathbb{R}^M$ and $\boldsymbol{h}_2(l) \in \mathbb{R}^H$. As in the discrete-time case above, M and H denote latent and hidden dimensionality, respectively. As above, latent states are related to the observations by Eq. 36.

Scaling with features l was implemented the same way as for the discrete-time models (C.1), with the distinction between dynamical and positional parameters given by Eq. 16.

C.3 Hyperparameters & compute resources

For the experiments on the Selkov system, we used $M = 2$ and $H = 128$ for both the discrete-time and continuous-time setups, and trained each model on ground-truth sequences of up to 20 steps with generalized teacher forcing [30]. Training times were approximately 10 min per shPLRNN model (on an AMD EPYC 7713 CPU with 64 cores; with multiple models trained in parallel) and about 10 min per Neural ODE model (on an NVIDIA GeForce RTX 2080 Ti GPU with roughly 2 GB used per model; again, multiple models were trained in parallel). For the reported results on the Lorenz-63 system with the continuous-time setup, we used $M = 12$ and $H = 64$, ground-truth sequences of up to 48 steps and generalized teacher forcing [30]. On the same GPU setup as for the

Selkov system, approximately 30 min were required for training and up to 4 GB of GPU memory per model.

All continuous-time models were optimized using the Adam optimizer [40], and all discrete-time models with RAdam [48], both as implemented in `pytorch` [59].

D Dynamical systems used for benchmarking

For each benchmark data set, we used J_{ID} different control parameter values as training domain. For each system within this domain, we generated K trajectories of length T_{train} for training, and additional K trajectories of length T_{test} for in-domain evaluation. To evaluate out-of-domain (OOD) generalization, we also generated K trajectories of length T_{test} each for J_{OOD} additional systems located outside the training domain. All data were standardized per dimension. To prevent data leakage, the mean and variance were computed exclusively across the training trajectories (spanning all J_{ID} training systems and time points) and subsequently applied to scale the training, in-domain and OOD test sets.

D.1 Lorenz-63 model

The Lorenz-63 model was originally introduced by Lorenz [49] as a simplified model of atmospheric convection. It was one of the earliest examples illustrating that a low-dimensional deterministic ODE with simple nonlinearities (low-order multinomials) can generate chaotic behavior:

$$\dot{x} = \sigma(y - x) \tag{40}$$

$$\dot{y} = x(\rho - z) - y \tag{41}$$

$$\dot{z} = xy - \beta z \tag{42}$$

We took $\sigma = 10$ and $\beta = 8/3$, and for the training set varied the control parameter ρ across $J_{\text{ID}} = 16$ uniformly spaced values in the interval $[5, 22.5]$, where the system converges to one of two FPs, depending on initial condition.

For each parameter setting, we integrated 10 training trajectories up to time $T = 50$ in time steps of $\Delta t = 0.01$ from random uniformly distributed initial conditions $x_0 \sim \mathcal{U}(-20, 20)$, $y_0 \sim \mathcal{U}(-25, 25)$ and $z_0 \sim \mathcal{U}(0, 35)$. As noted above, all training trajectories were standardized dimension-wise.

To test trained models, we created two additional sets of trajectories of length $T = 200$. The in-domain test set draws from the same range of values for ρ as in training, while for the OOD test set we added $J_{\text{OOD}} = 30$ systems with values of $\rho \in [22.5, 35]$, enabling us to evaluate generalization to unseen types of dynamics such as the transition to chaos at $\rho \approx 24.06$.

D.2 Selkov model

The two-dimensional Selkov model [67], which transitions from a stable spiral point to a stable limit cycle through a super-critical Hopf bifurcation, is given by

$$\dot{x} = -x + ay + x^2 y \tag{43}$$

$$\dot{y} = b - ay - x^2 y \, . \tag{44}$$

For training, we kept $a = 0.1$ constant and varied b across $J_{\text{ID}} = 15$ uniformly spaced values in the interval $[0.8, 1.1]$ ($\Delta b = 0.02$), all of which correspond to FP dynamics. Per setting, 10 training trajectories were integrated for $T = 5$ at step size $\Delta t = 0.1$, starting from uniformly distributed initial conditions $(x, y) \sim \mathcal{U}(-1, 3)$. The initial conditions and integration period were deliberately chosen to yield short *transient* trajectories, covering mainly the period of convergence to the fixed points (as the type of fixed point is otherwise impossible to learn from just a constant trajectory initiated in the point itself). All training trajectories were standardized per dimension.

For both test sets (i.e., in-domain and OOD) the integration time was extended to $T = 500$. The OOD test set was created using $J_{\text{OOD}} = 15$ values of $b \in [0.3, 0.8]$, covering both fixed point and limit cycle dynamics.

D.3 Rössler model

As a second chaotic benchmark system, we chose the Rössler attractor. Unlike the Lorenz-63, which becomes chaotic through a homoclinic bifurcation (for the specific parameter configurations we chose), the Rössler attractor has a period-doubling route to chaos. It is defined by the ODE system

$$\dot{x} = -y - z \tag{45}$$
$$\dot{y} = x + ay \tag{46}$$
$$\dot{z} = b + z(x - c). \tag{47}$$

We set $a = b = 0.1$ while varying the parameter c to generate the datasets. As training domain, we chose $c \in [4, 6, 8.5, 8.7, 9, 12, 12.6]$, where the first four parameters result in cycles of periods 1, 2, 4, and 8. At $c = 9$ the system is chaotic and $c = 12, 12.6$ are in a periodic window with periods 3 and 6. Like for the other benchmark datasets, we integrated 10 trajectories from initial conditions $(x, y) \sim \mathcal{U}(-5.0, 5.0)$ for $T = 50$ with $\Delta t = 0.1$ and then standardized them per dimension.

Furthermore, in-domain test trajectories were sampled at the same control parameter values as the training set, while for the OOD test set we chose $c \in [13, 15.4, 18]$ (chaotic, period 5, chaotic). The test trajectories are $T = 1000$ time units long.

E Feature extrapolation strategy

In this section we describe the procedure we used to obtain the feature vectors for the extrapolation regime. For the quantitative comparisons between model reconstructions and true systems at specific control parameter values we require these to be explicitly known, as otherwise a strict ground truth comparison is not possible. We emphasize, however, that 1) this knowledge is *not* required or used in model training, but only for the purpose of exact evaluation, and 2) even without this knowledge, learned model features can be extrapolated in different ways, depending on the information available. Assuming that subjects are (roughly) ordered according to the underlying changing control parameter (which is a reasonable assumption for time-ordered data, as, e.g., in climate systems), the strategy outlined below could still be used by replacing the control parameter value with a variable encoding the passage of time between consecutive datasets and the times to be extrapolated to, or in the simplest case just with the subject index. In the worst case, we may not have any information about control parameters and how they might be related among datasets. In this case, one could first train the hierarchical model with a single feature and then sort subjects by feature magnitude before applying an extrapolation fit. Once an ordering has been determined this way, retraining with more than just one feature could be done.

Given the true control parameters $p^{(j)}$ (for the purpose of evaluation) or a proxy for it (in any real-world setting with unknown control parameters), assume a hierarchical model is trained on J_{ID} systems/datasets with associated parameter values $\{p^{(j)}\}_{j=1}^{J_{\text{ID}}}$, yielding inferred latent features $\{l^{(j)}\}_{j=1}^{J_{\text{ID}}} \subset \mathbb{R}^L$. Predicting the dynamics for unseen control parameters $\{p^{(j')}\}_{j'=1}^{J_{\text{OOD}}}$ requires an estimate of the corresponding latent features $\{l^{(j')}\}_{j'=1}^{J_{\text{OOD}}}$. As discussed in Section 4.2, feature splitting allows l_{dyn} and l_{pos} to assume independent algebraic relationships with p (e.g. $l_{\text{dyn}} \propto p$ while $l_{\text{pos}} \propto \sqrt{p - p_{\text{crit}}}$ near a Hopf bifurcation). Rather than committing to a fixed parametric form *a priori*, we fit each feature (respectively feature dimension for $L > 1$) independently to p and select the most appropriate function class via cross-validation on the training domain.

Candidate models For each feature dimension, we consider three low-parameter functional forms that cover the scaling behaviors expected from the analysis in Section 4.2:

$$\text{(Power law)} \quad l_{\text{PL}}(p) = a\,(p - b)^d + c, \quad b < p_{\text{min}} - \eta\,(p_{\text{max}} - p_{\text{min}}), \tag{48}$$
$$\text{(Signed power law)} \quad l_{\text{SPL}}(p) = a\,\text{sign}(p - b)\,|p - b|^d + c, \tag{49}$$
$$\text{(Parabola)} \quad l_{\text{Parab}}(p) = a\,(p - b)^2 + c, \tag{50}$$

with exponent $d \in [0.5, 1.5]$. The power law models algebraic scaling with a fractional exponent, which is the natural functional form for the square-root-type behavior characteristic of multiple normal forms near local bifurcations. For any fit with $l_{\text{PL}}(p)$, we require that $b < p_{\min} - \eta(p_{\max} -$

$p_{\min}$), where $p_{\min}$ and $p_{\max}$ are the smallest and largest control parameter values (across both in-domain and OOD sets), respectively, and $\eta = 1$ determines the margin. This ensures that all points to which we seek to extrapolate lie safely within the real-valued domain of the function (avoiding complex roots where $p - b < 0$), and that the highly nonlinear region near $p = b$ is avoided. Additionally, we allow fits with a signed power law, which in contrast to the simple power law has full support, as well as a sign change similar to an inflection point at $p = b$. The parabolic fit is incorporated to cover non-monotonic trends which were observed to occur for higher-dimensional feature vectors.

Robust regression Each candidate function is fit by nonlinear least squares with a soft-ℓ_1 loss (`scipy.optimize.least_squares`), which down-weighs points with large residuals and thereby prevents a few aberrant features from dominating the fit. After an initial fit, points with standardized residual $|r_j|/\mathrm{RMSE} > k$ ($k = 2.5$) are flagged as outliers and the model is refit on the remaining points. We cap the fraction of points dropped at 5%.

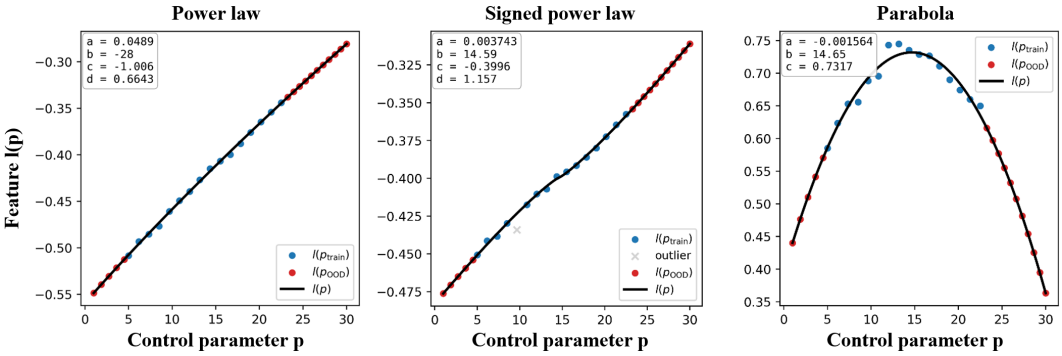


Figure 13: Empirical examples of extrapolation fits obtained from the three candidate models: power law (left), signed power law (center), and parabola (right). Blue markers correspond to in-domain training data, red markers to OOD samples, gray markers indicate discarded outliers, and the black curve represents the fitted function.

Model selection by cross-validation The three candidate function classes are compared by cross-validation: across $S = 5$ splits, we hold out a fraction $\tau = 0.2$ of all systems, fit each candidate function (with the same outlier handling) on the remaining systems, and evaluate the mean absolute error on the held-out systems,

$$\mathrm{MAE}_{\mathrm{CV}}^{(m)} = \frac{1}{S}\sum_{s=1}^{S}\frac{1}{|J_s|}\sum_{j\in J_s}\left|l^{(m,s)}(p^{(j)}) - l^{(j)}\right|, \tag{51}$$

for $m \in \{\mathrm{PL}, \mathrm{SPL}, \mathrm{Parab}\}$, where J_s is the set of indices of the held-out systems in split s, and $|J_s|$ denotes the size of this set. We use MAE rather than MSE to prevent single badly captured systems from dominating the comparison. The winning candidate function class $m^* = \arg\min_m \mathrm{MAE}_{\mathrm{CV}}^{(m)}$ is then re-fit on the full set of J_{ID} systems (excluding outliers) to produce the final extrapolated features at the requested OOD parameters $\{p^{(j')}\}_{j'=1}^{J_{\mathrm{OOD}}}$.

F Evaluation metrics

F.1 Wasserstein distance

To quantify reconstruction quality, we compare the bifurcation diagrams of the true and reconstructed systems by treating them as empirical distributions and measuring the discrepancy between them. In practice, we obtain bifurcation diagrams by evaluating the long-term behavior of predicted and true test trajectories for each system (in-domain and OOD), cutting off any transients and extracting local minima and maxima in the x_1 dimension. This yields two empirical samples

$$A = \{a_i\}_{i=1}^{n}, \qquad B = \{b_j\}_{j=1}^{m}, \tag{52}$$

from the true and reconstructed systems at each control parameter configuration, which we treat as one-dimensional empirical distributions on $\mathbb{R}$. For trajectories converging to a FP, the last 500 time points from the trajectory are used as the sample from the distribution. Crucially, the Wasserstein-1 distance is well-defined even when comparing singular distributions (e.g., isolated points) to continuous distributions (e.g., chaotic attractors), as it relies on optimal transport costs rather than overlapping support [69, 72], making it suitable for evaluating topological changes across bifurcations. For probability measures μ, ν on $\mathbb{R}$ with finite first moments, the Wasserstein-1 distance is defined as

$$W_1(\mu, \nu) = \int_0^1 \left| Q_\mu(q) - Q_\nu(q) \right| dq, \tag{53}$$

where Q_μ, Q_ν are the quantile functions of μ and ν [72].

This formulation motivates a simple empirical estimator. Let A and B be sorted in ascending order and let $\hat{Q}_A, \hat{Q}_B$ denote their empirical quantile functions, obtained by linear interpolation between the sample points. Evaluating both on a common uniform grid $\{q_k = (k-1)/(K-1)\}_{k=1}^K$ with $K = \max(n, m)$, we estimate

$$\widehat{W}_1(A, B) = \frac{1}{K} \sum_{k=1}^{K} \left| \hat{Q}_A(q_k) - \hat{Q}_B(q_k) \right|. \tag{54}$$

Choosing $K = \max(n, m)$ ensures that the larger of the two samples is used at full resolution while the smaller is up-sampled by linear interpolation of its quantile function; non-finite values in either sample are removed before sorting.

F.2 Correlation dimension

The correlation dimension introduced by Grassberger and Procaccia [25] is a measure of the fractal geometry of a set, but still yields the standard Euclidean integer-valued dimensionality for simple geometrical objects (like points or closed curves). The correlation sum of points $\{\boldsymbol{x}_t\}_{t=1}^T$ in some vector space is given by the number of all pairs of points that are less than some distance ϵ apart, i.e.,

$$C(\epsilon) = \frac{2}{T(T-1)} \sum_{i=1}^{T} \sum_{j=i+1}^{T} \Theta(\epsilon - \|\boldsymbol{x}_i - \boldsymbol{x}_j\|), \tag{55}$$

where $\Theta(\cdot)$ denotes the Heaviside step function. The correlation dimension is then defined as

$$D = \lim_{\epsilon \to 0} \lim_{N \to \infty} d(N, \epsilon), \qquad d(N, \epsilon) = \frac{\partial \ln C(\epsilon, N)}{\partial \ln \epsilon}. \tag{56}$$

Intuitively, for every point we count the number of neighboring points within a sphere of radius ϵ and then average all these counts. These counts $C(\epsilon, N)$ are then plotted as a function of radius ϵ in log-log coordinates, with the slope of the line fit for $\epsilon \to 0, N \to \infty$ yielding the correlation dimension.

For proper estimation of the correlation dimension as a geometrical measure in state space, we need to remove points which are purely temporal neighbors (due to the auto-correlation) [68]:

$$C_{\text{est}}(\epsilon) = \frac{2}{(T - t_{\min})(T - t_{\min} - 1)} \sum_{i=1}^{N} \sum_{j=i+1+t_{\min}}^{T} \Theta(\epsilon - \|\boldsymbol{x}_i - \boldsymbol{x}_j\|). \tag{57}$$

Generally, if a system is only partially observed, an optimal temporal delay embedding of the observed variables is required to compute this measure [35]. However, in all benchmark systems used here all dimensions of the DS are observed and therefore delay embedding is not required.

The correlation dimension is computed using the python library `nolitsa` (`https://github.com/manu-mannattil/nolitsa`). For a 2500 time unit trajectory (we chose time units to accommodate for varying integration step sizes for different systems) we discard the first 1500 time units as transient and apply the `c2`-method (Theiler window $t_{\min} = 10$ time units; 100 ϵ configurations; Euclidean distance) for the correlation sum and the `d2`-method for the respective slopes. Finally, we inspect finite slopes $C_{\text{est}}(\epsilon) \in [10^{-3}, 10^{-1}]$ and pick the slope of the 15-point-wide window with the least curvature.

F.3 Maximum Lyapunov exponent

Consider a discrete-time DS $\boldsymbol{x}_{t+1} = \boldsymbol{F}(\boldsymbol{x}_t)$, $\boldsymbol{x} \in \mathbb{R}^n$, with Jacobian $\boldsymbol{J}_t = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}_t}$. Considering the locally linearized dynamics, an infinitesimal perturbation δ_t of the state $\boldsymbol{x}_t$ evolves according to $\delta_{t+1} = \boldsymbol{J}_t \delta_t$ across one step and as $\delta_T = \boldsymbol{J}_{T-1}\boldsymbol{J}_{T-2}\cdots\boldsymbol{J}_0\delta_0$ after T steps. The maximum Lyapunov exponent is then given by [1, 35]

$$\lambda_{\max} = \lim_{T\to\infty} \frac{1}{T} \log \frac{\|\delta_T\|}{\|\delta_0\|} = \lim_{T\to\infty} \frac{1}{T} \log \sigma_{\max} \left(\prod_{t=0}^{T-1} \boldsymbol{J}_{T-t} \right), \tag{58}$$

where $\sigma_{\max}$ is the largest singular value of the Jacobian product matrix. The maximum Lyapunov exponent therefore describes the maximum rate of divergence in any direction of tangent space. For attraction to a fixed point one has $\lambda_{\max} < 0$, for divergence or chaotic attractors one has $\lambda_{\max} > 0$, and for an attracting limit cycle $\lambda_{\max} = 0$ along the closed orbit.

For numerical calculation of the maximum Lyapunov exponent, we use the Benettin QR algorithm [6] applied to a discrete map formulation $\boldsymbol{x}_{t+1} = \boldsymbol{F}(\boldsymbol{x}_t)$ of each model with time step Δt. The shPLRNN is a discrete map by construction, and its Jacobian known analytically. For the ground-truth ODE systems and the Neural ODE models, we define $\boldsymbol{F}$ as one step of the fourth-order Runge–Kutta (RK4) scheme, and the tangent dynamics are obtained by integrating the variational equation $\dot{\boldsymbol{\delta}} = \boldsymbol{J}(\boldsymbol{x})\,\boldsymbol{\delta}$ with the same scheme, which is equivalent to using the Jacobian $\boldsymbol{J}_t = \partial \boldsymbol{F}/\partial \boldsymbol{x}\,|_{\boldsymbol{x}_t}$ of the RK4 map.

Starting from an orthonormal frame $\boldsymbol{Q}_0 \in \mathbb{R}^{n\times n}$, the frame is propagated along the trajectory as $\boldsymbol{Q}_{t+1} = \boldsymbol{J}_t \boldsymbol{Q}_t$ and re-orthonormalized by a QR decomposition every $m = 5$ steps,

$$\boldsymbol{J}_{km+m-1}\cdots\boldsymbol{J}_{km}\,\boldsymbol{Q}_{km} = \boldsymbol{Q}_{(k+1)m}\,\boldsymbol{R}^{(k)}, \qquad k = 0, 1, \ldots, K-1. \tag{59}$$

The diagonal entries of the upper-triangular factors $\boldsymbol{R}^{(k)}$ contain the local expansion rates. The maximum Lyapunov exponent is the time average of the leading diagonal entry,

$$\lambda_{\max} \approx \frac{1}{Km\Delta t} \sum_{k=0}^{K-1} \ln \left| R_{11}^{(k)} \right|, \tag{60}$$

Dividing by Δt expresses all exponents per physical time unit, making all three estimates, ground-truth, Neural ODE and shPLRNN, directly comparable. In order to compute these estimates, we discard an initial transient of 1000 time units. The estimate is then refined across consecutive time windows until the running value of $\lambda_{\max}$ changes by less than a relative tolerance of 10^{-3} over three consecutive windows, up to a maximum integration horizon of 20000 time units.

G Proofs

G.1 Proof of Proposition 1

Proof. Let $\mathcal{S} \subset \mathbb{R}^{N\times N}$ denote the k-dimensional subspace of matrices representing the true physical sparsity pattern. By assumption, the true parameter-dependent part of the Jacobian satisfies $\boldsymbol{J}_1 \in \mathcal{S}$. The model uses an affine Jacobian parameterization of the form

$$\boldsymbol{J}_{\text{model}}(l) = \hat{\boldsymbol{J}}_0 + l\,\hat{\boldsymbol{J}}_1,$$

where l is the learned latent feature.

To demonstrate that extrapolation failures are a structural artifact of this affine parameterization rather than a consequence of poor training, we consider an ideal in-domain scenario. Suppose that the model perfectly captures the true parameter dependence during training, such that the learned feature scales strictly linearly with the control parameter, $l = \alpha p$ for some scalar $\alpha \neq 0$. We now decompose the learned variational matrix $\hat{\boldsymbol{J}}_1$ into its components parallel and orthogonal to $\mathcal{S}$ with respect to the Frobenius inner product:

$$\hat{\boldsymbol{J}}_1 = \hat{\boldsymbol{J}}_1^{\parallel} + \hat{\boldsymbol{J}}_1^{\perp}, \qquad \hat{\boldsymbol{J}}_1^{\parallel} \in \mathcal{S}, \quad \hat{\boldsymbol{J}}_1^{\perp} \in \mathcal{S}^{\perp}.$$

Here, $\hat{\boldsymbol{J}}_1^{\parallel}$ is the component aligned with the true physical parameter dependence, whereas $\hat{\boldsymbol{J}}_1^{\perp}$ lies in directions along which there is no variation in the true system.

Because the feature-coupling weights $\boldsymbol{\theta}_v$ in Eq. 3 are full-rank and no structural prior or sparsity-promoting regularization is imposed, empirical risk minimization (ERM) only constrains the projection of $\hat{\boldsymbol{J}}_1$ onto the true subspace $\mathcal{S}$. The orthogonal component $\hat{\boldsymbol{J}}_1^{\perp}$, which lives in the (N^2-k)-dimensional complement $\mathcal{S}^{\perp}$, therefore remains under-determined. Hence, generically, the learned solution may not satisfy $\hat{\boldsymbol{J}}_1^{\perp}=\mathbf{0}$; we write

$$\|\hat{\boldsymbol{J}}_1^{\perp}\|_F=\epsilon>0.$$

Now consider the Jacobian extrapolation error outside the training domain,

$$\boldsymbol{E}(p):=\boldsymbol{J}_{\mathrm{model}}(\alpha p)-\boldsymbol{J}_{\mathrm{true}}(p).$$

Even under the favorable assumption of perfect in-domain recovery of the physically meaningful part, namely

$$\hat{\boldsymbol{J}}_0=\boldsymbol{J}_0,\qquad \alpha\hat{\boldsymbol{J}}_1^{\parallel}=\boldsymbol{J}_1,$$

the remaining error is exactly the orthogonal component:

$$\boldsymbol{E}(p)=\big(\hat{\boldsymbol{J}}_0+\alpha p(\hat{\boldsymbol{J}}_1^{\parallel}+\hat{\boldsymbol{J}}_1^{\perp})\big)-\big(\boldsymbol{J}_0+p\boldsymbol{J}_1\big)=\alpha p\,\hat{\boldsymbol{J}}_1^{\perp}.$$

Thus, the extrapolation error grows linearly in $|p|$ whenever $\hat{\boldsymbol{J}}_1^{\perp}\neq 0$, and this growth occurs entirely in directions orthogonal to the true sparse parameter subspace $\mathcal{S}$.

To understand how this growing matrix error corrupts the reconstructed dynamics, we must evaluate the system's eigenvalue spectrum, which determines local stability and the occurrence of local bifurcations. We require a mechanism to translate the spatial matrix error into spectral drift. The Bauer–Fike theorem [5] provides this mathematical bridge, bounding how far the eigenvalues of a perturbed matrix can deviate from their true values based on the magnitude of the perturbation. Assuming that the true Jacobian $\boldsymbol{J}_{\mathrm{true}}(p)$ is diagonalizable, i.e.,

$$\boldsymbol{J}_{\mathrm{true}}(p)=\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1},$$

the theorem says that for any eigenvalue $\tilde{\lambda}$ of the extrapolated model Jacobian $\boldsymbol{J}_{\mathrm{model}}(\alpha p)$, there exists an eigenvalue λ of $\boldsymbol{J}_{\mathrm{true}}(p)$ such that

$$|\tilde{\lambda}-\lambda|\le\kappa(\boldsymbol{V})\|\boldsymbol{E}(p)\|_2\le\kappa(\boldsymbol{V})\,|\alpha p|\,\|\hat{\boldsymbol{J}}_1^{\perp}\|_2,$$

where $\kappa(\boldsymbol{V})=\|\boldsymbol{V}\|_2\|\boldsymbol{V}^{-1}\|_2$ is the condition number of the true eigenvector matrix. Using $\|\hat{\boldsymbol{J}}_1^{\perp}\|_2\le\|\hat{\boldsymbol{J}}_1^{\perp}\|_F=\epsilon$, we obtain

$$|\tilde{\lambda}-\lambda|\le\kappa(\boldsymbol{V})\,|\alpha p|\,\epsilon.$$

Therefore, as soon as $\epsilon>0$, the spectral deviation is allowed to grow as $\mathcal{O}(|p|)$ outside of the training domain. Because this unconstrained spectral drift acts in directions completely decoupled from the true physical mechanism, it can push eigenvalues across the imaginary axis or unit circle in a non-physical way, thereby inducing spurious local bifurcations. □

G.2 Proof of Theorem 1

Proof. We consider the Neural ODE and RNN separately.

Neural ODE: Define the FP residual along the true FP $\boldsymbol{x}^*_{\mathrm{true}}(p)$ by

$$\boldsymbol{r}_{\mathrm{NODE}}(p):=\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}^*_{\mathrm{true}}(p),g(p))\,. \tag{61}$$

The model captures the true FP at p precisely when $\boldsymbol{r}_{\mathrm{NODE}}(p)=\mathbf{0}$. By the chain rule,

$$\boldsymbol{r}'_{\mathrm{NODE}}(p)=\frac{\partial\boldsymbol{f}_{\boldsymbol{\theta}}}{\partial\boldsymbol{z}}(\boldsymbol{x}^*_{\mathrm{true}}(p),g(p))\,\frac{\mathrm{d}\boldsymbol{x}^*_{\mathrm{true}}}{\mathrm{d}p}+\frac{\partial\boldsymbol{f}_{\boldsymbol{\theta}}}{\partial l}\,(\boldsymbol{x}^*_{\mathrm{true}}(p),g(p))\,g'(p). \tag{62}$$

The assumed Jacobian match gives

$$\frac{\partial \boldsymbol{f}_\theta}{\partial \boldsymbol{z}}\left(\boldsymbol{x}^*_{\mathrm{true}}(p), g(p)\right) = \boldsymbol{J}_{\mathrm{true}}(p). \tag{63}$$

Moreover, because

$$\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z}, l) = (\boldsymbol{W}_c + l\boldsymbol{W}_v)\boldsymbol{\psi}(\boldsymbol{z}) + \boldsymbol{h}_c + l\boldsymbol{h}_v, \tag{64}$$

differentiation with respect to l while holding $\boldsymbol{z}$ fixed yields

$$\frac{\partial \boldsymbol{f}_{\boldsymbol{\theta}}}{\partial l}(\boldsymbol{z}, l) = \boldsymbol{W}_v\boldsymbol{\psi}(\boldsymbol{z}) + \boldsymbol{h}_v. \tag{65}$$

Consequently,

$$\boldsymbol{r}'_{\mathrm{NODE}}(p) = \boldsymbol{J}_{\mathrm{true}}(p)\frac{\mathrm{d}\boldsymbol{x}^*_{\mathrm{true}}}{\mathrm{d}p} + g'(p)\left[\boldsymbol{W}_v\boldsymbol{\psi}\left(\boldsymbol{x}^*_{\mathrm{true}}(p)\right) + \boldsymbol{h}_v\right]. \tag{66}$$

Necessity: If the model is to capture the true FP throughout the interval, then $\boldsymbol{r}_{\mathrm{NODE}}(p) = \boldsymbol{0}$ throughout that interval, and hence $\boldsymbol{r}'_{\mathrm{NODE}}(p) = \boldsymbol{0}$. Therefore,

$$\boldsymbol{J}_{\mathrm{true}}(p)\frac{\mathrm{d}\boldsymbol{x}^*_{\mathrm{true}}}{\mathrm{d}p} = -g'(p)\left[\boldsymbol{W}_v\boldsymbol{\psi}\left(\boldsymbol{x}^*_{\mathrm{true}}(p)\right) + \boldsymbol{h}_v\right]. \tag{67}$$

As $\boldsymbol{J}_{\mathrm{true}}(p)$ by assumption is invertible, multiplication by $\boldsymbol{J}_{\mathrm{true}}(p)^{-1}$ gives

$$\frac{\mathrm{d}\boldsymbol{x}^*_{\mathrm{true}}}{\mathrm{d}p} = -g'(p)\boldsymbol{J}_{\mathrm{true}}(p)^{-1}\left[\boldsymbol{W}_v\boldsymbol{\psi}\left(\boldsymbol{x}^*_{\mathrm{true}}(p)\right) + \boldsymbol{h}_v\right]. \tag{68}$$

Thus the compatibility relation is necessary.

Sufficiency: Conversely, suppose the compatibility relation holds. Substitution into the expression for $\boldsymbol{r}'_{\mathrm{NODE}}(p)$ yields $\boldsymbol{r}'_{\mathrm{NODE}}(p) = \boldsymbol{0}$ throughout the interval, i.e. $\boldsymbol{r}_{\mathrm{NODE}}$ is constant across the interval. The assumed FP match at p_0 gives $\boldsymbol{r}_{\mathrm{NODE}}(p_0) = \boldsymbol{0}$, so the constant is zero and

$$\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}^*_{\mathrm{true}}(p), g(p)) = \boldsymbol{0} \tag{69}$$

throughout the interval. The model therefore captures the complete true FP dependence.

RNN: For the discrete-time model, define

$$\boldsymbol{r}_{\mathrm{RNN}}(p) := \boldsymbol{x}^*_{\mathrm{true}}(p) - \boldsymbol{F}_{\boldsymbol{\theta}}\left(\boldsymbol{x}^*_{\mathrm{true}}(p), g(p)\right). \tag{70}$$

The model captures the true FP precisely when $\boldsymbol{r}_{\mathrm{RNN}}(p) = \boldsymbol{0}$. Differentiating gives

$$\boldsymbol{r}'_{\mathrm{RNN}}(p) = \frac{\mathrm{d}\boldsymbol{x}^*_{\mathrm{true}}}{\mathrm{d}p} - \frac{\partial \boldsymbol{F}_{\boldsymbol{\theta}}}{\partial \boldsymbol{z}}\left(\boldsymbol{x}^*_{\mathrm{true}}(p), g(p)\right)\frac{\mathrm{d}\boldsymbol{x}^*_{\mathrm{true}}}{\mathrm{d}p} \tag{71}$$

$$- \frac{\partial \boldsymbol{F}_{\boldsymbol{\theta}}}{\partial l}(\boldsymbol{x}^*_{\mathrm{true}}(p), g(p))\, g'(p). \tag{72}$$

Using the Jacobian match and the affine dependence on l,

$$\boldsymbol{r}'_{\mathrm{RNN}}(p) = \left(\boldsymbol{I} - \boldsymbol{J}_{\mathrm{true}}(p)\right)\frac{\mathrm{d}\boldsymbol{x}^*_{\mathrm{true}}}{\mathrm{d}p} - g'(p)\left[\boldsymbol{W}_v\boldsymbol{\psi}\left(\boldsymbol{x}^*_{\mathrm{true}}(p)\right) + \boldsymbol{h}_v\right]. \tag{73}$$

If the FP is captured throughout the interval, then this derivative vanishes. Since $\boldsymbol{I} - \boldsymbol{J}_{\mathrm{true}}(p)$ is assumed to be invertible, we obtain

$$\frac{\mathrm{d}\boldsymbol{x}^*_{\mathrm{true}}}{\mathrm{d}p} = g'(p)\left(\boldsymbol{I} - \boldsymbol{J}_{\mathrm{true}}(p)\right)^{-1}\left[\boldsymbol{W}_v\boldsymbol{\psi}\left(\boldsymbol{x}^*_{\mathrm{true}}(p)\right) + \boldsymbol{h}_v\right]. \tag{74}$$

This proves necessity. Conversely, if this relation holds, then $\boldsymbol{r}'_{\mathrm{RNN}}(p) = \boldsymbol{0}$. Hence $\boldsymbol{r}_{\mathrm{RNN}}$ is constant, and the assumed match $\boldsymbol{r}_{\mathrm{RNN}}(p_0) = \boldsymbol{0}$ implies $\boldsymbol{r}_{\mathrm{RNN}}(p) = \boldsymbol{0}$ throughout the whole interval. This proves sufficiency. □

G.3 Proof of Corollary 1.1

Proof. Write $\boldsymbol{x}^*(p) := \boldsymbol{x}^*_{\text{true}}(p)$. For the Neural ODE, let $\boldsymbol{r}_{\text{NODE}}(p) := \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}^*(p), g(p))$; for the RNN, let $\boldsymbol{r}_{\text{RNN}}(p) := \boldsymbol{x}^*(p) - \boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{x}^*(p), g(p))$. Both residuals vanish at p_b. Differentiating at the true FP and using the Jacobian match in Theorem 1 gives

$$\boldsymbol{r}'_{\text{NODE}}(p_b) = \boldsymbol{J}_{\text{true}}(p_b)\boldsymbol{x}^{*\prime}(p_b) + g'(p_b)\big[\boldsymbol{W}_v\boldsymbol{\psi}(\boldsymbol{x}^*(p_b)) + \boldsymbol{h}_v\big], \tag{75}$$

$$\boldsymbol{r}'_{\text{RNN}}(p_b) = \big(\boldsymbol{I} - \boldsymbol{J}_{\text{true}}(p_b)\big)\boldsymbol{x}^{*\prime}(p_b) - g'(p_b)\big[\boldsymbol{W}_v\boldsymbol{\psi}(\boldsymbol{x}^*(p_b)) + \boldsymbol{h}_v\big]. \tag{76}$$

Under the invertibility assumptions of Theorem 1, the relevant derivative vanishes exactly when its compatibility relation holds at p_b. Hence, failure of that relation gives $\boldsymbol{r}'(p_b) \neq \boldsymbol{0}$. Differentiability and $\boldsymbol{r}(p_b) = \boldsymbol{0}$ imply

$$\boldsymbol{r}(p_b + \Delta p) = \boldsymbol{r}'(p_b)\Delta p + O(|\Delta p|^2), \tag{77}$$

where $\boldsymbol{r}$ denotes the residual of the model under consideration. Taking norms yields the stated $\Theta(|\Delta p|)$ bound, so the true FP is not a model FP for all sufficiently small nonzero OOD increments. □

G.4 Proof of Proposition 2

Proof. Let $\boldsymbol{E}(p) := \boldsymbol{J}_{\text{model}}(l(p)) - \boldsymbol{J}_{\text{true}}(p)$. Since the model matches the true Jacobian at p_b,

$$\boldsymbol{J}_{\text{model}}(l(p_b)) = \boldsymbol{J}_{\text{true}}(p_b), \tag{78}$$

it follows that

$$\boldsymbol{E}(p) = [\boldsymbol{J}_{\text{model}}(l(p)) - \boldsymbol{J}_{\text{model}}(l(p_b))] - [\boldsymbol{J}_{\text{true}}(p) - \boldsymbol{J}_{\text{true}}(p_b)] \tag{79}$$

$$= \sum_{i=1}^{L} \Delta l_i(p)\hat{\boldsymbol{J}}_{1,i} - [\boldsymbol{J}_{\text{true}}(p) - \boldsymbol{J}_{\text{true}}(p_b)]\,. \tag{80}$$

Let $P_{\mathcal{S}^\perp}$ denote the Frobenius-orthogonal projection onto $\mathcal{S}^\perp$. By assumption,

$$\boldsymbol{J}_{\text{true}}(p) - \boldsymbol{J}_{\text{true}}(p_b) \in \mathcal{S}, \tag{81}$$

so its projection onto $\mathcal{S}^\perp$ is zero. Therefore,

$$P_{\mathcal{S}^\perp}\boldsymbol{E}(p) = \sum_{i=1}^{L} \Delta l_i(p)\hat{\boldsymbol{J}}^{\perp}_{1,i}. \tag{82}$$

Orthogonal projection cannot increase the Frobenius norm. Hence,

$$\|\boldsymbol{E}(p)\|_F \geq \|P_{\mathcal{S}^\perp}\boldsymbol{E}(p)\|_F \tag{83}$$

$$= \left\| \sum_{i=1}^{L} \Delta l_i(p)\hat{\boldsymbol{J}}^{\perp}_{1,i} \right\|_F \tag{84}$$

$$\geq \beta\|\Delta l(p)\|_2, \tag{85}$$

where the last step is exactly the stated non-degeneracy assumption. Thus,

$$\|\boldsymbol{J}_{\text{model}}(l(p)) - \boldsymbol{J}_{\text{true}}(p)\|_F \geq \beta\|\Delta l(p)\|_2. \tag{86}$$

If $\|\Delta l(p)\|_2 \to \infty$ along the OOD path, the right-hand side diverges, and therefore so does the Jacobian mismatch. □

G.5 Proof of Theorem 2

Proof. Fix $\boldsymbol{x}$, write $\bar{p} := p - p_0$, and let $\boldsymbol{A} := \frac{\partial}{\partial p}\boldsymbol{J}_{\text{disc}}(p_0, \boldsymbol{x})$ and $\boldsymbol{B} := \frac{\partial^2}{\partial p^2}\boldsymbol{J}_{\text{disc}}(p_0, \boldsymbol{x})$. By the assumed differentiability,

$$\boldsymbol{J}_{\text{disc}}(p, \boldsymbol{x}) = \boldsymbol{J}_{\text{disc}}(p_0, \boldsymbol{x}) + \bar{p}\,\boldsymbol{A} + \tfrac{1}{2}\bar{p}^2\boldsymbol{B} + O(\bar{p}^3), \tag{87}$$

$$g(p) = g(p_0) + \bar{p}\,g'(p_0) + \tfrac{1}{2}\bar{p}^2 g''(p_0) + O(\bar{p}^3). \tag{88}$$

Matching the value and first parameter derivative at p_0 gives

$$\hat{\boldsymbol{J}}_0 + g(p_0)\hat{\boldsymbol{J}}_1 = \boldsymbol{J}_{\text{disc}}(p_0, \boldsymbol{x}), \qquad g'(p_0)\hat{\boldsymbol{J}}_1 = \boldsymbol{A}. \tag{89}$$

Since $g'(p_0) \neq 0$, the model's second derivative is $g''(p_0)\hat{\boldsymbol{J}}_1 = \frac{g''(p_0)}{g'(p_0)}\boldsymbol{A}$. Subtracting the two expansions yields

$$\boldsymbol{J}_{\text{model}}(g(p)) - \boldsymbol{J}_{\text{disc}}(p, \boldsymbol{x}) = \tfrac{1}{2}\bar{p}^2 \left[\frac{g''(p_0)}{g'(p_0)}\boldsymbol{A} - \boldsymbol{B}\right] + O(\bar{p}^3). \tag{90}$$

Thus, the equality in the theorem is necessary and sufficient for second-order matching at fixed g. If $\boldsymbol{B} \notin \operatorname{span}\{\boldsymbol{A}\}$, no scalar feature curvature can make the quadratic term vanish.

For the system in Eq. 2, centered at p_0, write $\boldsymbol{f}(\boldsymbol{x}, p) = \boldsymbol{f}_0(\boldsymbol{x}) + \bar{p}\,\boldsymbol{f}_1(\boldsymbol{x})$ and $\boldsymbol{J}_{\text{cont}}(\boldsymbol{x}, p) = \boldsymbol{J}_0(\boldsymbol{x}) + \bar{p}\,\boldsymbol{J}_1(\boldsymbol{x})$. Let $\boldsymbol{M}(t)$ be the Jacobian of the true flow from $\boldsymbol{x}$. It satisfies $\dot{\boldsymbol{M}}(t) = \boldsymbol{J}_{\text{cont}}(\Phi_t(\boldsymbol{x}), p)\boldsymbol{M}(t)$ and $\boldsymbol{M}(0) = \boldsymbol{I}$. Differentiating once more at $t = 0$ gives

$$\ddot{\boldsymbol{M}}(0) = \boldsymbol{J}_{\text{cont}}(\boldsymbol{x}, p)^2 + \sum_{k=1}^{N} f_k(\boldsymbol{x}, p)\, \partial_{x_k} \boldsymbol{J}_{\text{cont}}(\boldsymbol{x}, p). \tag{91}$$

Expanding $\boldsymbol{M}(\Delta t) = \boldsymbol{J}_{\text{disc}}(p, \boldsymbol{x})$ and differentiating w.r.t. p therefore yields

$$\boldsymbol{A} = \boldsymbol{J}_1 \Delta t + O(\Delta t^2), \qquad \boldsymbol{B} = \left[\boldsymbol{J}_1^2 + \sum_{k=1}^{N} f_{1,k}\, \frac{\partial \boldsymbol{J}_1}{\partial \boldsymbol{x}_k}\right] \Delta t^2 + O(\Delta t^3). \tag{92}$$

Here the local smoothness of the flow is assumed to be sufficient for the remaining $O(\Delta t^3)$ term to stay at this order after differentiating twice w.r.t. p. If $\boldsymbol{J}_1(\boldsymbol{x}) \neq \boldsymbol{0}$ and the matrix part enclosed by square brackets lies outside $\operatorname{span}\{\boldsymbol{J}_1(\boldsymbol{x})\}$, then $\boldsymbol{B} \notin \operatorname{span}\{\boldsymbol{A}\}$ for all sufficiently small Δt, because $\boldsymbol{A}/\Delta t \to \boldsymbol{J}_1$ and $\boldsymbol{B}/\Delta t^2$ tends to the matrix part enclosed by brackets. □

G.6 Proof of Corollary 2.1

Fix $\boldsymbol{x}$ and write $\boldsymbol{J}(p) := \boldsymbol{J}_{\text{disc}}(\boldsymbol{x}, p)$. Choose $h > 0$ such that $p_0, p_0 - h, p_0 - 2h$ lie in the training domain. Let $\boldsymbol{Y}_j$ estimate $\boldsymbol{J}(p_0 - jh)$ with $\|\boldsymbol{Y}_j - \boldsymbol{J}(p_0 - jh)\|_F \leq \delta$ for $j = 0, 1, 2$, and assume $\|\boldsymbol{J}'''(p)\|_F \leq M_3$ on $[p_0 - 2h, p_0]$. Since tangent matching gives $\boldsymbol{J}'(p_0) = g'(p_0)\hat{\boldsymbol{J}}_1$, define

$$\widehat{\boldsymbol{R}}_{\text{curv}} := \frac{\boldsymbol{Y}_0 - 2\boldsymbol{Y}_1 + \boldsymbol{Y}_2}{h^2} - g''(p_0)\hat{\boldsymbol{J}}_1, \tag{93}$$

$$\boldsymbol{R}_{\text{curv}} := \boldsymbol{J}''(p_0) - g''(p_0)\hat{\boldsymbol{J}}_1, \tag{94}$$

$$\eta_R := M_3 h + \frac{4\delta}{h^2}. \tag{95}$$

Proof: Two applications of the fundamental theorem of calculus give

$$\frac{\boldsymbol{J}(p_0) - 2\boldsymbol{J}(p_0 - h) + \boldsymbol{J}(p_0 - 2h)}{h^2} = \frac{1}{h^2}\int_0^h \int_0^h \boldsymbol{J}''(p_0 - s - t)\,\mathrm{d}t\,\mathrm{d}s. \tag{96}$$

The bound on $\boldsymbol{J}'''$ makes the difference from $\boldsymbol{J}''(p_0)$ at most $M_3 h$; the three estimation errors contribute at most $4\delta/h^2$. Thus $\|\widehat{\boldsymbol{R}}_{\text{curv}} - \boldsymbol{R}_{\text{curv}}\|_F \leq \eta_R$.

Value and tangent matching give $\boldsymbol{E}(p_0) = \boldsymbol{E}'(p_0) = \boldsymbol{0}$ and $\boldsymbol{E}''(p_0) = -\boldsymbol{R}_{\text{curv}}$. Taylor's theorem and the assumed OOD bound yield

$$\left\|\boldsymbol{E}(p_0 + r) + \tfrac{1}{2}r^2 \boldsymbol{R}_{\text{curv}}\right\|_F \leq \tfrac{K}{6} r^3. \tag{97}$$

Since $\|\boldsymbol{R}_{\text{curv}}\|_F \leq \|\widehat{\boldsymbol{R}}_{\text{curv}}\|_F + \eta_R$, the triangle inequality yields the stated bound. Its right-hand side is nondecreasing in $r \geq 0$, proving the final claim. □

It is important to note that the $\boldsymbol{Y}_j$ must provide independent estimates of the Jacobians of the true flow maps, i.e. must not come from a single hierarchical DSR model where estimates for different control parameter values would depend on each other via the shared group-level parameters. The in-domain error bounds and the OOD bound on $\boldsymbol{E}'''$ are additional assumptions beyond Theorem 2.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state our main contributions: identifying structural causes of OOD failures, proposing sparsity regularization and feature splitting, and deriving a bound on reliable extrapolation. The claims are aligned with the presented theory and experiments.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We explicitly discuss a key limitation: affine discrete-time models still have an irreducible geometric mismatch, and the discussion notes that the approach does not fully remove architectural limits. This is stated both in the main text and conclusion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The theoretical results are accompanied by stated assumptions and full proofs in the appendix. Theorems, propositions, and corollaries are numbered and cross-referenced.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The data setup and general model classes are described in the main text and appendix. The new architecture, constraints and regularizations used during training are well described.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is provided on `https://github.com/DurstewitzLab/TopOODG`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The appendix specifies the benchmark systems, parameter ranges, trajectory counts, integration times, time steps, and standardization procedure, which allows reproducing the results. Exact model architectures, optimizers, hyperparameters etc. are stated.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The figures report results over n=10 runs and show medians with IQR envelopes, which is appropriate variability information for the main experiments.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The compute resources used for this work are stated in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work fully conforms with the Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

Justification: Predicting dynamical systems across bifurcations is not considered to have an immediate societal impact.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The results of this paper do not pose a risk.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [N/A]

Justification: No assets owned by others were used.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not introduce new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Neither crowdsourcing nor research with human subjects were involved in this paper.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Not required for this work.

    Guidelines:

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: Our core methods do not involve LLMs.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.