A Minimal Interpretable Architecture for Zero-Shot Reconstruction of Dynamical Systems

Christoph Jürgen Hemmer[2,1,3]*, **Florian Plaswig**[1,3]*, **Daniel Durstewitz**[1,2,3,4]
[1]Dept. of Theoretical Neuroscience, Central Institute of Mental Health, Mannheim, Germany
[2]Interdisciplinary Center for Scientific Computing (IWR), Heidelberg, Germany
[3]Faculty of Physics and Astronomy, Heidelberg University, Heidelberg, Germany
[4]ELLIS Unit Heidelberg, Heidelberg, Germany

Abstract

Recent foundation models (FMs) for zero-shot reconstruction of dynamical systems (DS) achieve strong out-of-domain generalization but provide little insight into the *mechanisms* that underlie their forecasts. Such an understanding could help to strip down overladen FM architectures to their bare essence and expose the minimal requirements for in-context learning in the DS domain. Toward this goal, here we iteratively reduce a recent powerful SOTA model for DS reconstruction, DynaMix [46], to a minimal interpretable two-parameter form, which we call *DynaBase*. DynaBase produces forecasts through a linear blend of the current latent state and the nearest in-context neighbor and its temporal successor. Surprisingly, despite its extreme simplicity, DynaBase produces highly competitive zero-shot DS reconstructions across chaotic, cyclic, and real-world systems, with a negligible parameter load, many orders of magnitude below that of other FMs. Even more, this extreme simplicity permits direct model optimization on DS reconstruction measures, as well as closed-form one-step analytical solutions on prediction MSE. Theoretical and empirical analysis of DynaBase further leads to a 1-parameter family of maps, with the context-parroting algorithm of [100] recovered at one end, and chaotic (divergent but bounded) behavior at the other. We further show how different training strategies lead to models either optimal for short-term prediction or for DS reconstruction. Thus, DynaBase not only exposes the minimal mechanisms required for producing zero-shot DS reconstruction, but also reconciles within an accessible mathematical frame divergent observations in the literature.

1 Introduction

Many, if not most, natural and engineered systems, from chemical and molecular processes to climate, ecosystems, brain activity, or stock markets, are naturally described as dynamical systems (DS) [32, 40, 67, 88, 10, 22, 64]. Reconstructing these DS (DSR) in a data-driven way from just time series observations is of fundamental interest to all sciences, which seek tractable models that enable mechanistic insight into the underlying dynamical processes. DSR, hence, aims for *generative surrogate models* of the underlying dynamics which not only forecast a few steps ahead, but—importantly—reproduce *long-term statistical* properties of the DS in question, such as its attractor geometry (ergodic distribution) and power spectrum [15, 32, 41, 44, 76, 20, 18, 33]. While traditionally DSR involved custom-training ML/AI models on each particular dataset [14, 48, 44, 75, 76], inspired by the success of in-context learning in large language and time-series foundation models

*These authors contributed equally to this work.
Corresponding authors: {christoph.hemmer, daniel.durstewitz}@zi-mannheim.de
Code: https://github.com/DurstewitzLab/DynaBase

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

(FMs) [4, 5, 11], a new line of research explored whether zero-shot generalization to previously unseen DS is also feasible [46, 57].

However, as for most if not all FMs, it is currently utterly unclear how SOTA DSR FMs like DynaMix [46] achieve their zero-shot out-of-domain generalization, and what the essential architectural requirements for this are. Insight into how AI models solve their problems is of general interest, but is particularly important in scientific and medical domains where the focus is on mechanistic understanding [78, 32]. Here we tackle this question in the context of DSR by reverse-engineering the pretrained DynaMix model. We produce a series of systematic ablations and simplifications that largely retain, or even improve on, the model's zero-shot DSR performance. We ultimately arrive at an algebraically minimal form consisting just of a linear combination of the model's previous state, the nearest neighbor from the context signal, and its immediate temporal successor. A natural *symmetry* requirement on the context points further condenses the model into a two-parameter form. We show that this recursive map, *DynaBase*, can generate point, cyclic, and chaotic attractors, and allows for closed-form one-step least-squares fitting to time series data. Theoretical and empirical analysis of DynaBase trained directly for DSR reveals that optimal zero-shot solutions essentially lie along a 1-parameter curve with truly chaotic dynamics at one end and 'context parroting' of a time series snippet at the other, a strategy recently observed in transformer-based time series FMs like Chronos [100, 99]. Our observations are in line with a recent literature that demonstrates that tiny ML/AI models, many orders of magnitude below the parameter load of modern FMs, often achieve comparable, or even widely superior performance, on a range of tasks from cognitive and behavioral modeling [58] to challenging reasoning problems [92, 51].

In a nutshell, our main results are:

- **Minimal zero-shot DSR model:** Through iterative simplification of DynaMix [46] we create a minimal 2-parameter map, DynaBase, which exhibits competitive zero-shot DSR performance at essentially zero training costs.
- **Mechanistic insight:** The model's two coefficients (α, β) parametrize a continuous spectrum of dynamical behaviors, with optimal DSR solutions essentially driven to a 1-parameter subspace. This minimal form achieves its strong performance essentially by 1) closely following the context signal while 2) regulating local divergence rates through a single parameter α, enabling a spectrum from point attraction and stable cycles to chaos.
- **Impact of training strategy:** We further demonstrate that training using either a standard ahead-prediction MSE loss vs. a specific DSR loss yields fundamentally different solutions: Training for ahead-prediction often collapses to simple context parroting and hence does not recover the true dynamics, especially for chaotic systems, which is only achieved by DSR-specific training.

2 Related work

Dynamical systems reconstruction (DSR) DSR aims to learn generative models from time series data that capture the underlying system's dynamical long-term behavior, i.e. its long-term or "climate" statistics and state space geometry [76, 32, 48, 65, 41, 20]. Custom-trained models for this purpose build on library-based methods like SINDy [19, 60], reservoir computing [73, 75, 76, 23, 45], neural ODEs [25, 3, 54], Koopman operators [63, 70, 6, 68, 21, 94], or on various types of RNNs [87, 31, 90, 14, 79, 48, 47, 15, 13]. More important than the particular architecture appears to be the training strategy: sparse or generalized teacher forcing [65, 48] are control-theoretic training techniques that enable long trajectory roll-outs even for chaotic DS, while specific regularization methods based on a system's invariant measures have been designed to enforce agreement in long-term statistics [38, 76, 50]. All of these methods, however, require purpose-training on each new system, and out-of-domain generalization remains a key challenge [44].

Zero-shot DSR and time-series foundation models (FMs) Multi-environment [98, 53], hypernetwork-guided [89], or hierarchical [16] approaches have been used to train DSR models on many DS simultaneously, but all of these still require fine-tuning on new systems or parameter configurations not seen in training. More recently, inspired by the success of in-context inference in LLMs [17, 39, 1], a line of general-purpose time series (TS) FMs pretrained on large corpora of real and synthetic data has emerged, which forecast TS from a short observed context snippet zero-shot. Most of these,

like Chronos [4, 5], TimesFM [29], or Panda [57], are based on transformers, with a few exceptions that engage RNNs like the xLSTM [8] or Mamba [42] as their core. All of these, however, lack the ability to reproduce a system's long-term behavior as required in DSR [44, 46, 33]. They mostly either converge to just fixed points in the limit, or cyclically repeat the context [99, 100], and thus fail to capture DS with inherently chaotic dynamics [46]. DynaMix [46] closes this gap by pretraining, based on DSR-specific training techniques [65], a mixture of AL-RNN [15] experts, weighted by a context-dependent gating network. It delivers competitive short-term forecasts and SOTA long-term performance, while being relatively lightweight with $\sim 10^4$ parameters.

Mechanisms of in-context learning What is at the root of in-context learning has been intensely studied in the past few years [17, 39, 1], and hypotheses range from transformers implementing gradient-descent in-context [7, 91, 59, 96] to mechanisms that more or less boil down to a simple remixing of their memory contents [84, 93, 95]. For DSR FMs, the analogous problem remains largely open but—at the same time—appears more tractable due to the more constrained and mathematically better defined [44] nature of the question. Recent studies using Chronos suggested that TS FMs may long-term forecast DS by the simple mechanism of context parroting [99], a well known pattern in LLMs [69, 35, 26]. Based on this phenomenon, Zhang and Gilpin [100] design a 'context parroting algorithm' which takes the last D values of the context as a delay-embedded query, retrieves the best-matching motif elsewhere in the context, and copies the subsequent fragment as forecast. They show that this strategy works surprisingly well and actually outperforms most TS FMs. But context parroting, as exhibited by Chronos or conceptualized in this form, can only reproduce *discrete-cyclic* activity and thus *inherently* fails to capture defining signatures of chaotic systems, like positive Lyapunov exponents and rather broad power spectra [46, 33]. A superficially similar principle is employed in classical nonlinear forecasting algorithms that retrieve from a library (history) locally neighboring trajectories whose next time step predictions are either just averaged or based on which a locally linear model is fit [36, 52]. Crucially, in contrast to mere context parroting, this neighborhood may change, however, with each prediction step taken. Through iterative reduction of DynaMix we arrive at a mathematical form that contains these classes of algorithms as special cases.

3 Methods

3.1 Background: DynaMix

DynaMix [46] is a DSR foundation model based on a mixture-of-experts architecture, using $j = 1 \ldots J$ almost-linear RNNs (AL-RNNs) [15] as experts,

$$\boldsymbol{z}_{t+1}^j = F_j^{\text{exp}}(\boldsymbol{z}_t) = \boldsymbol{A}_j \boldsymbol{z}_t + \boldsymbol{W}_j \Phi^*(\boldsymbol{z}_t) + \boldsymbol{h}_j \,, \tag{1}$$

where $\boldsymbol{z}_t^j \in \mathbb{R}^M$ is an M-dimensional latent state, $\Phi^*(\boldsymbol{z}_t) := [\boldsymbol{z}_{1:M-P,t},\ \text{ReLU}(\boldsymbol{z}_{M-P+1:M,t})]^\top$ is the activation function with $M-P$ linear units and P ReLUs, and $\boldsymbol{A}_j \in \text{diag}(\mathbb{R}^M)$, $\boldsymbol{W}_j \in \mathbb{R}^{M\times M}$, and $\boldsymbol{h}_j \in \mathbb{R}^M$ are learnable parameters. The different experts are combined into an overall next state prediction by $\boldsymbol{z}_{t+1} = \sum_{j=1}^J w_{j,t}^{exp} \cdot \boldsymbol{z}_{t+1}^j$, where the weights $w_{j,t}^{exp} \in (0,1)$ are determined by a gating network through

$$\boldsymbol{w}_t^{exp}(\boldsymbol{z}_t, \tilde{\boldsymbol{C}}\boldsymbol{w}_t^{att}) = \sigma\left(\frac{\text{MLP}(\tilde{\boldsymbol{C}}\boldsymbol{w}_t^{att}, \boldsymbol{z}_t)}{\tau_{\text{exp}}}\right) \in \mathbb{R}^J \,, \tag{2}$$

where τ_{exp} is a temperature parameter and $\tilde{\boldsymbol{C}} = \text{CNN}(\boldsymbol{C})$ are temporal features obtained from the provided context signal $\boldsymbol{C} = \{\boldsymbol{c}_t\} \in \mathbb{R}^{N\times T_C}$ through a CNN. Attention weights $\boldsymbol{w}_t^{att}$ at each time step t are in turn computed based on a distance between current latent state $\boldsymbol{z}_t$ and the context $\boldsymbol{C}$ as

$$\boldsymbol{w}_t^{att}(\boldsymbol{z}_t, \boldsymbol{C}) = \sigma\left(\frac{\left|\boldsymbol{C} - (\boldsymbol{D}\boldsymbol{z}_t + \boldsymbol{\epsilon})\,\boldsymbol{1}_{T_C}^\top\right|^\top \boldsymbol{1}_N}{\tau_{\text{att}}}\right) \in \mathbb{R}^{T_C} \,, \tag{3}$$

where τ_{att} is another temperature, $\boldsymbol{D} \in \mathbb{R}^{N\times M}$ a learnable matrix, $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$ exploration noise, and $\boldsymbol{1}_{\{T_C,N\}}$ are column vectors of ones of length T_C, N, respectively. In [46], DynaMix was trained on $\approx 6 \times 10^5$ trajectories from 34 $3d$ DS in the cyclic or chaotic regime using control-theoretic training techniques that ensure DSR criteria are met [65, 48]. After pretraining, given any context time series snippet $\boldsymbol{C}$, DynaMix is charged with forecasting the long-term trajectory evolution of the underlying DS without any retraining or fine-tuning.

3.2 Reducing DynaMix

At a high level, DynaMix can be compactly expressed as a single RNN of the following form

$$\boldsymbol{z}_{t+1} = F_{\text{DynaMix}}(\boldsymbol{z}_t, \boldsymbol{C}) := F^{\text{exp}}(\boldsymbol{z}_t) \cdot \boldsymbol{w}_t^{exp}(\boldsymbol{z}_t, \tilde{\boldsymbol{C}}\boldsymbol{w}_t^{att}) \tag{4}$$

with (see [46])

$$\tilde{\boldsymbol{C}}\boldsymbol{w}_t^{att} = \text{CNN}(\boldsymbol{C}) \cdot \boldsymbol{w}_t^{\text{att}}(\boldsymbol{z}_t, \boldsymbol{C}) \; . \tag{5}$$

Note that eq. 4 is a weighted sum of *piecewise-linear* systems F_j^{exp}, where, however, the weights $\boldsymbol{w}_t^{exp}(\boldsymbol{z}_t, \tilde{\boldsymbol{C}}\boldsymbol{w}_t^{att})$ themselves are nonlinear functions of the current state and the context. To reduce this model, we first replace the *almost-linear* experts F_j^{exp} (eq. 1) by a *fully linear* form, without loss in performance as confirmed by ablations (Table 1). Regarding the expert weights $\boldsymbol{w}^{exp}$, it has previously been demonstrated [46] that the MLP in eq. 2 can also be replaced by a linear layer without significant loss in performance. Likewise, we observe that the output nonlinearity σ (softmax; eq. 2) is nearly linear over its relevant operating domain and hence replace it by a linear function as well (Table 1). This leaves us with a representation that is bilinear in $\boldsymbol{z}_t$.

We assume that this bilinear form on a compact domain can be mimicked by a universal ReLU-based MLP approximator [97]. Since the MLP is ReLU-based, for each given input it performs a linear mapping, and hence its 'degree of non-linearity' depends on how many of these linear subregions (polytopes) are actively engaged. We observe that assuming just a single linear region, i.e. a fully linear MLP, does not degrade performance either (Table 1). Thus, $\tilde{\boldsymbol{C}} = \text{CNN}(\boldsymbol{C})$ and $\boldsymbol{w}_t^{att}(\boldsymbol{z}_t, \boldsymbol{C})$ remain as the only nonlinearities, and the whole model can be expressed as:

$$\boldsymbol{z}_{t+1} = F_{\text{Linear}}(\boldsymbol{z}_t, \boldsymbol{C}) = \boldsymbol{A}\boldsymbol{z}_t + \boldsymbol{B}(\text{CNN}(\boldsymbol{C})\boldsymbol{w}_t^{att}(\boldsymbol{z}_t, \boldsymbol{C})) + \boldsymbol{h} \; . \tag{6}$$

Next, for $\tau_{\text{att}} \to 0$, $\boldsymbol{w}_t^{att}(\boldsymbol{z}_t, \boldsymbol{C})$ essentially boils down to a nearest neighbor selector. Indeed, empirically $\tau_{\text{att}} \approx 0.06 \ll 1$, and replacing it by $\tau_{\text{att}} \to 0$ does not degrade DynaMix performance (Table 1). Further assuming a CNN kernel size of 2 and noticing that for a single linear CNN layer (as used in DynaMix) the CNN operations can be combined with $\boldsymbol{B}$ in eq. 6, we thus obtain

$$f_{\text{Linear}}(\boldsymbol{z}_t, \boldsymbol{C}) = \boldsymbol{A}\boldsymbol{z}_t + \boldsymbol{B}^{(1)}\boldsymbol{c}_{\tau(\boldsymbol{z}_t)} + \boldsymbol{B}^{(2)}\boldsymbol{c}_{\tau(\boldsymbol{z}_t)+1} + \tilde{\boldsymbol{h}} \tag{7}$$

$$\tau(\boldsymbol{z}_t) = \underset{s<T_C}{\arg\min} \|\boldsymbol{c}_s - \boldsymbol{z}_t\|_2 \tag{8}$$

where $\boldsymbol{A}, \boldsymbol{B}^{(1)}, \boldsymbol{B}^{(2)} \in \mathbb{R}^{M\times M}$ and $\tilde{\boldsymbol{h}} \in \mathbb{R}^M$. We call this form a *recursive affine nearest-neighbor (NN) map*. See Appx. A.3 for more details on the reduction.

3.3 DynaBase

In DSR, two vector fields that differ only by a rigid coordinate transformation and possibly invertible coordinate changes by some matrix $\boldsymbol{T}$ describe the same underlying flow, yielding *topologically conjugate* reconstructions [74]. Recognizing this invariance, we can state the following result:

Theorem 1 (Uniqueness of recursive affine NN maps)**.** *Any nearest-neighbor recursive map of the form eq. 7 that is affine in $\boldsymbol{z}_t$, $\boldsymbol{c}_1$, and $\boldsymbol{c}_2$, equivariant under $O(N)$ rotations and reflections, and translation equivariant, belongs to the two-parameter linear family $f_{\alpha,\beta}$,*

$$\boldsymbol{z}_{t+1} = f_{\alpha,\beta}(\boldsymbol{z}_t) = \alpha\, \boldsymbol{z}_t + \beta\, \boldsymbol{c}_1 + (1-\alpha-\beta)\, \boldsymbol{c}_2. \tag{9}$$

Proof. See Appx. B. □

Exploiting Thm. 1, we obtain a minimal model, which we call *DynaBase*, as

$$\boldsymbol{z}_{t+1} = f_{\alpha,\beta}(\boldsymbol{z}_t) = \alpha\, \boldsymbol{z}_t + \beta\, \boldsymbol{c}_{\tau(\boldsymbol{z}_t)} + \gamma\, \boldsymbol{c}_{\tau(\boldsymbol{z}_t)+1}, \qquad \tau(\boldsymbol{z}_t) = \arg\min_{s<T_C} \|\boldsymbol{c}_s - \boldsymbol{z}_t\|_2, \tag{10}$$

with $\alpha, \beta, \gamma \in \mathbb{R}$ and $\alpha + \beta + \gamma = 1$. Thus, $\gamma = 1 - \alpha - \beta$ leaves $(\alpha, \beta) \in \mathbb{R}^2$ as the only free parameters of the minimal model (formalized in Thm. 1). If the data were noise-free, a natural *self-consistency* condition would be given by

$$\boldsymbol{c}_{i+1} = f_{\alpha,\beta}(\boldsymbol{c}_i) \; \forall i \in [1, T_C - 1] \; , \tag{11}$$

which guarantees that whenever $\boldsymbol{z}_t$ coincides with a context point $\boldsymbol{c}_i$, update eq. 10 maps $\boldsymbol{z}_t \mapsto \boldsymbol{c}_{i+1}$ exactly. Using this reasonable assumption, the affine NN map could be simplified further by imposing $\alpha = -\beta \implies \gamma = 1$, such that the whole construction depends only on a single parameter:

$$\boldsymbol{z}_{t+1} = f_\alpha(\boldsymbol{z}_t) = \alpha\, (\boldsymbol{z}_t - \boldsymbol{c}_{\tau(\boldsymbol{z}_t)}) + \boldsymbol{c}_{\tau(\boldsymbol{z}_t)+1} \; , \tag{12}$$

where the local rate of con- or divergence is determined solely by α.

3.4 Training

For training DynaBase, we assume we are given N-dimensional training data $\boldsymbol{X} \in \mathbb{R}^{N \times T}$, of which we define the first T_C time steps as the context signal $\boldsymbol{C} \in \mathbb{R}^{N \times T_C}$. Given the model's simplicity, it allows for straightforward training, where we assume $\boldsymbol{z}_t = \boldsymbol{x}_t$ (i.e., the dynamics is directly defined in the observation space). Specifically, we explore the two following approaches:

Linear regression As DynaBase is linear in parameters, its one-step-ahead prediction error loss is convex and we can optimize it by least-squares regression in one step using the reparameterized form

$$\boldsymbol{z}_{t+1} - \boldsymbol{c}_{\tau(\boldsymbol{z}_t)+1} = \alpha \left(\boldsymbol{z}_t - \boldsymbol{c}_{\tau(\boldsymbol{z}_t)+1}\right) + \beta \left(\boldsymbol{c}_{\tau(\boldsymbol{z}_t)} - \boldsymbol{c}_{\tau(\boldsymbol{z}_t)+1}\right). \tag{13}$$

Given a training sequence, nearest neighbors $(\boldsymbol{c}_{\tau(\boldsymbol{x}_t)}, \boldsymbol{c}_{\tau(\boldsymbol{x}_t)+1})\ \forall \boldsymbol{x}_i \in \boldsymbol{X}$ are simply precomputed based on eq. 8, and ordinary-least-squares is used to solve for (α, β) in closed form. No backpropagation or teacher forcing is required; fitting is embarrassingly cheap compared to conventional FM pretraining.

Grid search However, linear regression only optimizes for short term forecasts, and hence might not be optimal for DSR problems which require capturing long-term behavior. We therefore investigate a second strategy, where the loss is given through a long-term reconstruction measure [55, 65]

$$\text{Loss}(\boldsymbol{X}, \hat{\boldsymbol{X}}) = D_{\text{stsp}}(p_{\text{true}}(\boldsymbol{X}) \| p_{\text{pred}}(\hat{\boldsymbol{X}}))\ , \tag{14}$$

i.e. the Kullback-Leibler divergence between the true and model-generated trajectory distributions based on long-term rollouts $\hat{\boldsymbol{X}}$ of DynaBase. D_{stsp} is a common measure in the DSR literature to assess the agreement in true and reconstructed attractor geometries [55, 65, 41, 48, 71, 99, 57]. Since the model has only 2 trainable parameters, searching for an optimal solution by grid search is fairly straightforward and inexpensive. In Appx. A.7 we furthermore show that using the sliced Wasserstein distance instead as optimization criterion, yet evaluating performance on D_{stsp}, gives similar results.

4 Results

4.1 Theoretical results

We now provide a theoretical characterization of the forecasting properties of DynaBase.

Structural characterization By Thm. 1, DynaBase is a unique nearest-neighbor affine family compatible with the symmetry assumptions in Sec. 3.3. We use this structural result as the starting point for the two main mechanism-level consequences below.

Theorem 2 (Reduction to context parroting)**.** *Let $\boldsymbol{C}_{1:T_C} \in \mathbb{R}^{1 \times T_C}$ be a context sequence and $\overline{\boldsymbol{C}}_{1:T_C-D+1} \in \mathbb{R}^{D \times (T_C-D+1)}$ its delay embedding with dimension D, defined as $\overline{\boldsymbol{c}}_s = (c_s, c_{s+1}, \ldots, c_{s+D-1})$. Let f_0 denote DynaBase with $\alpha = 0$ (hence $\beta = 0, \gamma = 1$) on the embedded context $\overline{\boldsymbol{C}}$, and consider the discrete-time system $\boldsymbol{z}_{t+1} = f_0(\boldsymbol{z}_t)$ with initial condition $\boldsymbol{z}_0 = \overline{\boldsymbol{c}}_{T_C-D+1}$ and $\tau(\boldsymbol{z}_0) \in [1, T_C - 2D]$. Then the projection $x_t = z_{D,t}$ exactly reproduces the context-parroting algorithm of [100].*

Proof. See Appx. B. □

Theorem 2 shows that the context-parroting baseline of [100] is recovered for free from DynaBase by simply setting $(\alpha, \beta) = (0, 0)$ and delay-embedding the context. Any DynaBase model with $\alpha \neq 0$ is therefore strictly more expressive than parroting in the sense that a larger class of DS can be represented.

Given that the map f_α is piecewise-affine, we can also derive an exact continuous-time analogue [66] $\boldsymbol{\zeta}(t)$, such that $\boldsymbol{\zeta}(kh) = \boldsymbol{z}_k \forall k \in \mathbb{Z}, h \in \mathbb{R}$, by integrating the equations

$$\frac{\mathrm{d}\boldsymbol{\zeta}}{\mathrm{d}t} = \frac{\log \alpha}{h}\boldsymbol{\zeta} - \frac{1}{h}\frac{\log \alpha}{1-\alpha}(\boldsymbol{c}_{\tau(\zeta)+1} - \alpha \boldsymbol{c}_{\tau(\zeta)}) \tag{15}$$

where we denote the discrete timestep size of f_α as h and $\frac{\log \alpha}{1-\alpha}|_{\alpha=1}$ is defined by its continuous extension. Notably, this limit *does not exist* for $\alpha = 0$ (context parroting), and is only defined in

the complex number space for $\alpha < 0$. For $\alpha = 0$, the map collapses onto $\boldsymbol{C}$ at every step, i.e. hops between a *discrete set of points*, which is not reproducible by a continuous flow unless $\boldsymbol{C}$ contains exactly one (fixed) point. Importantly, context parroting is therefore an *inherently discrete* special case of DynaBase, *defined on a measure-zero set*, which cannot reproduce true limit cycles or chaotic behavior. A detailed derivation is given in Appx. A.6.

Dynamical characterization We now characterize the dynamical properties of the self-consistent DynaBase – specifically how the parameter α governs the transition from trivial to genuinely chaotic behavior.

Theorem 3 (Long-term dynamics)**.** *For $\alpha = 0$, every orbit of DynaBase converges after finitely many steps onto a terminal cycle contained in the context sequence, and thus has max. Lyapunov exponent $\lambda_{max} < 0$.*

Proof. See Appx. B. □

The case $\alpha = 0$ is therefore dynamically trivial: all orbits eventually collapse to $\boldsymbol{C}$; context parroting is thus *structurally incapable of producing chaos*.

Theorem 4 (Boundedness)**.** *Let f_α be the self-consistent DynaBase map with parameter $\alpha \geq 0$ and context sequence $\boldsymbol{C}_{1:T_C} \subset \mathbb{R}^N$, and let $d_C(z) := \min_{1 \leq s \leq T_C - 1} \|z - c_s\|_2$. Then $\alpha \in [0, 1)$ is a sufficient but not a necessary condition for the orbits of f_α to remain bounded, while for $\alpha = 1$ this holds if the context is closed, i.e., $d_{\boldsymbol{C}}(\boldsymbol{c}_{T_C}) = 0$. More precisely, $\alpha \in [0, 1]$ guarantees $d_{\boldsymbol{C}}(\boldsymbol{z}_{t+1}) \leq \alpha\, d_{\boldsymbol{C}}(\boldsymbol{z}_t) + d_{\boldsymbol{C}}(\boldsymbol{c}_{T_C})$ for all t, where the last term vanishes for closed contexts. For $\alpha > 1$, there exist Voronoi parcellations by $\boldsymbol{C}$ which allow for strictly chaotic dynamics, i.e. such that $\lambda_{\max} > 0$ and the dynamics remains bounded.*

Proof. See Appx. B. □

The theorem thus establishes that $\alpha \in [0, 1)$ is sufficient for the orbit to remain confined to – or track – the context sequence, as distances to $\boldsymbol{C}$ remain bounded. For $\alpha = 1$, distances to $\boldsymbol{C}$ are non-increasing whenever the context is closed. For $\alpha > 1$, nearby trajectories diverge almost everywhere. If the arrangement of Voronoi cells through the context $\boldsymbol{C}$ is, however, such that its dynamics still remains bounded, it will almost surely follow a chaotic attractor [2, 43]. An example is discussed in the proof to Thm. 4 and illustrated in Fig. 2c.

4.2 Empirical forecasting comparison

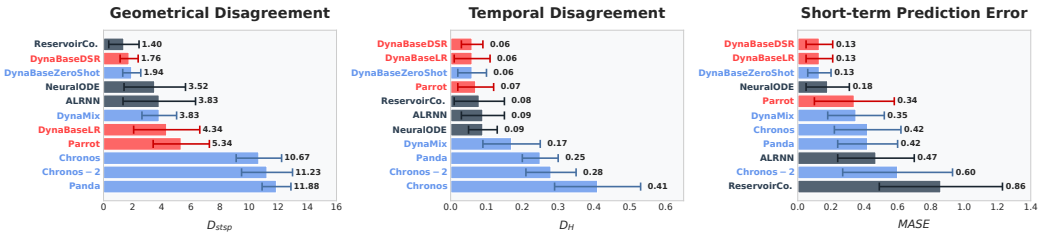

Figure 1: DSR performance across all 54 test DS used in [46]. Minimal models (red: DynaBase training variants and pure context parroting), custom-trained DSR models (gray), and pretrained zero-shot foundation models including a zero-shot variant of DynaBase (blue) are compared for the same context length $T_C = 2000$ (values for custom-trained and foundation models taken from [46]). Median±MAD of D_{stsp} (left, geometrical disagreement), D_H (center, temporal disagreement) and MASE (right, 10-step prediction error).

We evaluate DynaBase on the same DSR test set as in [46], reporting the same three complementary measures, two of them specific to DSR (see Appx. A.2 for definitions): the disagreement D_{stsp} in state space geometry, the long-term temporal disagreement D_H, and the short-term mean absolute scaled error (MASE). To evaluate the long-term statistics D_{stsp} and D_H, each system is forecast for $T = 10{,}000$ time steps, following [46] and in agreement more generally with the DSR literature

[48, 76, 33] (note that point forecasts become meaningless for chaotic systems beyond a few Lyapunov times). As introduced in Sec. 3.4, we fit DynaBase by either (i) closed-form one-step linear regression, optimizing the one-step MSE in eq. 13, or (ii) directly for DSR over the grid of parameter values. As baselines we include context parroting as reported in [99, 100], DynaMix [46], Panda [57] and Chronos [4, 5] as FMs, and three custom-trained DSR baselines (AL-RNN [15], reservoir computers [72], and Neural ODEs [25]) trained directly on the context signal provided to the FMs. In Appx. A.8 we furthermore report comparisons to 'classical' library-based nonlinear forecasting and local approximation methods [52, 36].

As shown in Fig. 1, DynaBase generally performs best, with the variant optimized for DSR (ii) better than the one trained for ahead-prediction (i) on the long-term measure D_{stsp} (while the differences in MASE and D_H are not statistically significant, Wilcoxon signed-rank test, $p > 0.1$). However, for longer (but not too long) prediction horizons, optimizing for ahead-prediction achieves lower MASE than DSR-optimization (Table 3). This difference illustrates that, not unexpectedly, the choice of *training objective* determines the model's forecasting strategy: a one-step MSE penalizes any local deviations from the ground truth, and hence tends to produce smaller α values enabling the map to track the immediate next step tightly, yielding better short-term predictions. In contrast, the grid optimization strategy (ii) directly targets the invariant attractor geometry, outperforming (i) on DSR (see also Appx. A.7, where we further show this also holds when optimizing for the Wasserstein distance yet evaluating on D_{stsp}). In either case, training DynaBase with its few parameters is extremely cheap and rapid.

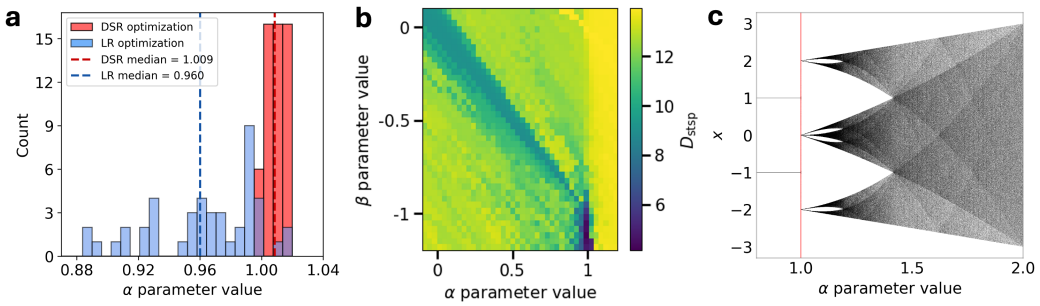

Figure 2: **a**) Comparison of distributions of estimated α values across all 54 chaotic test DS [46] for the two training objectives (sect. 3.4). Closed-form minimization of the 1-step MSE yields $\alpha < 1$, outside of the chaotic regime. In contrast, optimization for DSR places most α values firmly (and correctly) into the chaotic regime, with a median $\alpha \approx 1.01$. **b**) DSR loss landscape (optimization for D_{stsp}) across (α, β) shows a valley at $\alpha = -\beta$ in agreement with the self-consistency condition, as well as a minimum at $\alpha > 1$. **c**) Bifurcation diagram of the self-consistent DynaBase map f_α for the $1d$ context sequence $\boldsymbol{C} = (-1, 1, -1)$ (the boundedness of this map was shown in the proof of Thm. 4). The graph shows the map's long-term behavior (attractor structure) as a function of α, obtained by drawing 1,000 random initial conditions from the interval $[-2, 2]$ and discarding transients. The red line ($\alpha = 1$) marks the transition from a stable 2-cycle $(-1, +1)$ to the chaotic regime.

The distribution of estimated α values across the test systems in Fig. 2**a** makes this point explicit. The two training objectives lead to qualitatively different parameter regimes: linear regression on the 1-step MSE places the map mostly into the parroting/ semi-cyclic regime with $\alpha < 1$, while optimizing directly on D_{stsp}, in contrast, produces mostly chaotic behavior (in accordance with the ground-truth DS) with $\alpha > 1$. This implies that only DSR-based training has correctly inferred the chaotic ground truth systems, as further confirmed by a high correlation ($r \approx 0.90$) between max. Lyapunov exponents of the GT DS and the DynaBase map when fitted by DSR, but not when fitted by least-squares ($r \approx 0.18$), Fig. 3**a**. Moreover, the loss landscape reveals an elongated valley along the $\alpha = -\beta$ axis (Fig. 2**b**), providing empirical support for the self-consistency condition in eq. 12, which reduces DynaBase to a one-parameter map. The bifurcation graph in Fig. 2**c** further illustrates on a synthetic toy example the cyclic behavior for $\alpha < 1$ and emergence of chaos for $\alpha > 1$.

Another interesting observation is that for DSR-specific training, α tightly concentrates on a single value near $\alpha \approx 1.01$, across the diverse set of DS used. This suggests that, at least on this benchmark, the α-parameter may be *fixed a priori*. We test this by estimating α on a *separate training set* (similar to the one used for training DynaMix in [46]), obtaining $\alpha = 1.006$ close to the median in

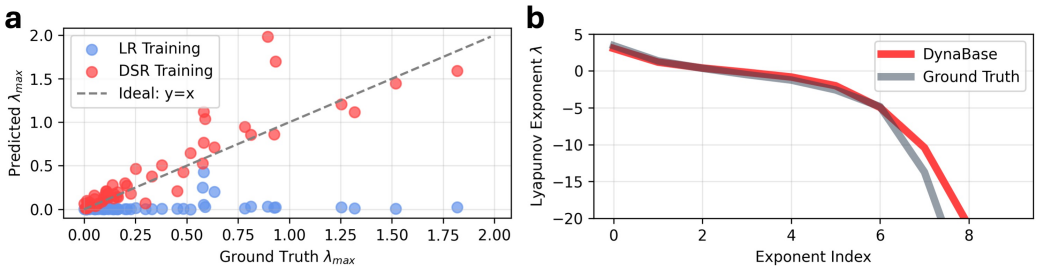

Figure 3: **a**) Lyapunov exponents estimated via the Rosenstein algorithm (see Appx. A.2) across test DS for both linear regression (blue) and DSR-based training (red) vs. ground truth exponents. Only DSR training produces long-term dynamics that closely follows that of the ground truth, including chaotic attractors with $\lambda_{max} > 0$. **b**) Full Lyapunov spectrum of multiscale Lorenz-96 for DynaBase and ground truth, estimated via the Sano-Sawada method.

Fig. 2**a**, which leads to a *parameter-free zero-shot forecaster* that requires no DS-specific fitting. Applying this to the context signals from the test set, we still achieve performance comparable to the custom-trained models (see Fig. 1, and Figs. 8-11 for zero-shot reconstructions). In Tables 4-7 we, moreover, confirm that the estimated α value as well as DynaBase' performance is rather robust w.r.t. different scalings of the dimensions, context lengths, partial observations, and various noise levels.

Given that DynaBase only uses a single parameter α to regulate local divergence rates, how well could it represent the full Lyapunov spectra of high-dimensional systems? We tested this by applying the zero-shot DynaBase version to the 10-dimensional, hyperchaotic two-scale Lorenz-96 system, with two positive Lyapunov exponents and slow and fast variables. As shown in Fig. 13 and Table 2, DynaBase reconstructs the hyperchaotic attractor and captures both time scales equally well ($D_H = 0.13$ for slow and $D_H = 0.11$ for fast variables). Even more surprisingly, the reconstructed Lyapunov spectrum closely matches that of the ground truth across its whole range (Fig. 3**b**).[1]

Finally, we assessed DynaBase on long-term forecasting of *real-world TS*, following the setup in [33]. The results in Fig. 4 and Table 8 indicate that DynaBase often even outperforms DynaMix on these partially observed, noisy time series, preserving their long-term behavior.

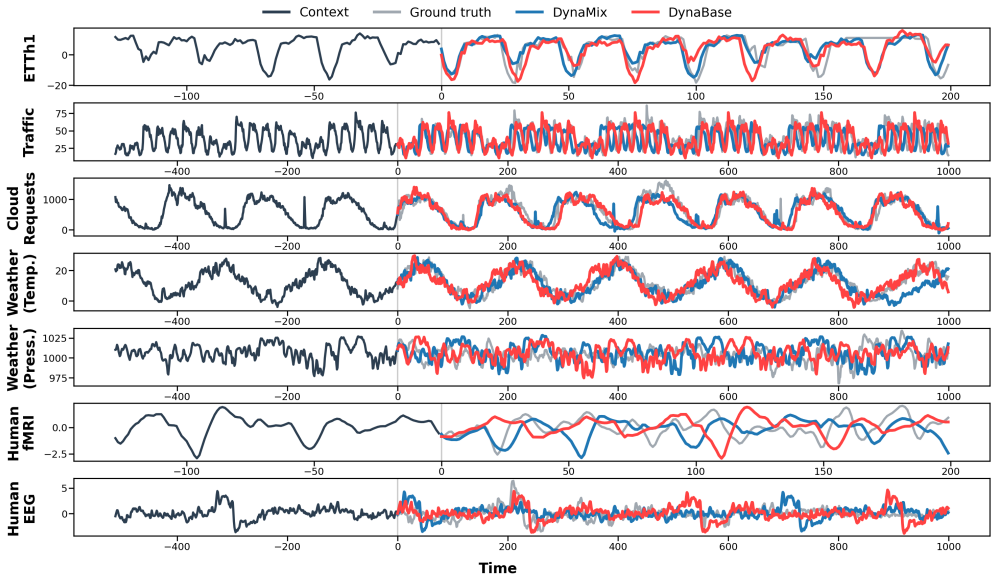

Figure 4: TS forecasts on $6d$ delay-embedded real-world time series for DynaBase (red) with α custom-trained, and DynaMix (blue). Context in dark gray.

[1] Note that con-/divergence rates also depend on the piecewise Voronoi parcellation [27], not just on parameter α.

4.3 Loss landscape tracks dynamics

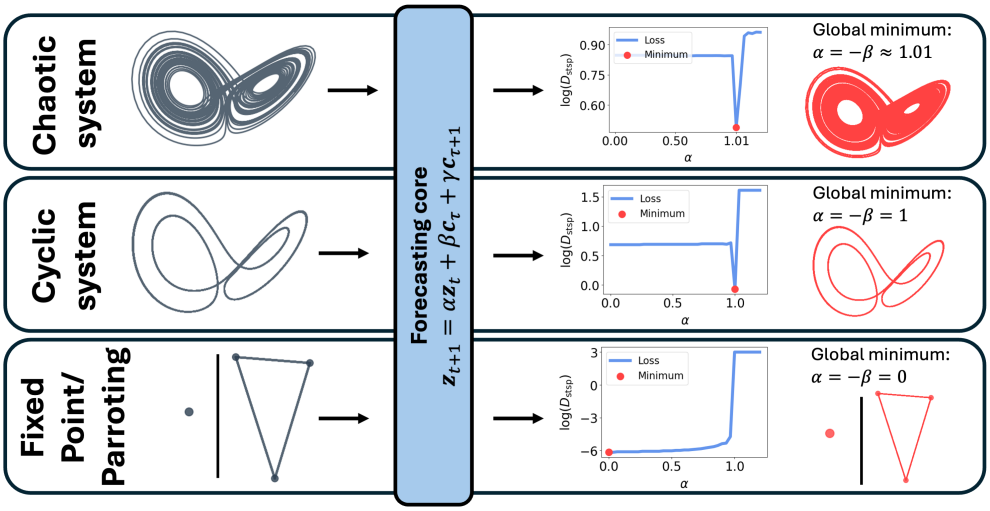

Figure 5: DynaBase' loss landscape reflects the dynamics of the target system when trained for DSR. **Top:** Lorenz-63 in chaotic regime (gray). The geometrical misalignment D_{stsp} across α exhibits a global minimum (red dot) at $\alpha \approx 1.01$, which produces a valid reconstruction of the attractor (red). **Center:** Lorenz-63 in cyclic regime. The D_{stsp} minimum occurs at $\alpha = 1$. **Bottom:** A fixed point or cyclic point (discrete cycle) (gray). D_{stsp} is minimized at $\alpha = 0$, corresponding to context parroting.

To examine how DynaBase' parameters map onto the ground truth system's dynamical regime, we study the DSR loss landscape on a representative chaotic, cyclic and fixed point system (Fig. 5). For these different dynamical regimes, the loss landscape exhibits distinct global minima which correspond to different dynamical mechanisms:

- **Context parroting** $\alpha = -\beta = 0$: the map reduces to $\boldsymbol{z}_{t+1} = \boldsymbol{c}_{\tau(\boldsymbol{z}_t)+1}$, i.e. the state snaps onto the successor of its nearest context neighbor. For a delay-embedded context signal, this exactly reproduces the context-parroting algorithm of Zhang and Gilpin [100] (see Thm. 2).
- **Flow mimicry** $\alpha = -\beta = 1$: the map reduces to $\boldsymbol{z}_{t+1} - \boldsymbol{z}_t = \boldsymbol{c}_{\tau(\boldsymbol{z}_t)+1} - \boldsymbol{c}_{\tau(\boldsymbol{z}_t)}$, a forward-Euler step driven by the local finite difference $\dot{\boldsymbol{c}} \approx (\boldsymbol{c}_{\tau(\boldsymbol{z}_t)+1} - \boldsymbol{c}_{\tau(\boldsymbol{z}_t)})/h$ of the underlying vector field as sampled from the context. This yields a 'zero-friction' direction in state space (neither con- nor divergence), as along the limit cycle ($\lambda_{\max} = 0$) of a continuous-time DS.
- **Chaos** $\alpha > 1$: One obtains local divergence almost everywhere and, if bounded, produces chaotic trajectories (cf. Fig. 14 for Lorenz-63 reconstructions along a bifurcation diagram).

The nearest-neighbor lookup $\tau(\cdot)$ induces a Voronoi tessellation of the state space (Fig. 6**a**). Inside each cell $V_s = \{\boldsymbol{z} : \tau(\boldsymbol{z}) = s\}$, DynaBase is a strictly affine map $\boldsymbol{z}_{t+1} = \alpha\, \boldsymbol{z}_t + \tilde{\boldsymbol{c}}_s$, where $\tilde{\boldsymbol{c}}_s = \beta \boldsymbol{c}_s + \gamma \boldsymbol{c}_{s+1}$ is fixed by the cell, hence belongs to the class of well-studied switching piecewise-linear (PL) systems [28, 85]. Within each cell V_s, the model's Jacobian is simply given by $\alpha \boldsymbol{I}$, i.e. a single scalar α controls the local con- or divergence almost everywhere in state space. With $\alpha \approx 1.01$ just slightly above 1 (Fig. 2), DynaBase diverges almost everywhere but still stays sufficiently close to the cell centers to create a chaotic *attractor*, despite not having a nonlinear activation function (in line with Thm. 3 & Thm. 4). For context parroting with $\alpha = 0$ (Thm. 2), on the other hand, DynaBase snaps onto a cyclic point that perfectly traces out the empirically observed context trajectory. However, a cyclic point would result in this case even if the observed trajectory snippet comes from a chaotic system (Fig. 15).

Pure context parroting also prevents generalization to new initial conditions of an observed DS, a feature of DynaMix demonstrated in [46]. This is illustrated for the Selkov DS in Fig. 6**b**: For $\alpha = 0$, DynaBase strictly follows the context points and thus fails to map the vector field anywhere outside the context. On the other hand, pure context mimicry, f_1, while reproducing the $\lambda_{\max} = 0$ direction *on* a limit cycle, fails to capture the convergence when started *off* the stable cycle, since the finite difference $\boldsymbol{c}_{\tau(\boldsymbol{z}_t)+1} - \boldsymbol{c}_{\tau(\boldsymbol{z}_t)}$ is always evaluated at the nearest context point. Fitting α via eq. 14 on

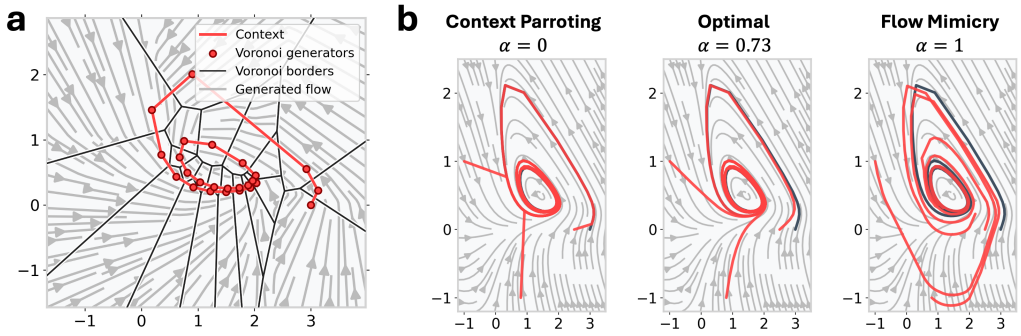

Figure 6: **a**) Voronoi tessellation for the periodic Selkov DS induced by the nearest-neighbor lookup $\tau(\cdot)$ in DynaBase. The context $\boldsymbol{C}_{1:T_C}$ partitions the state space into cells $V_s = \{\boldsymbol{z} : \tau(\boldsymbol{z}) = s\}$, within which DynaBase is described by a strictly affine map $\boldsymbol{z}_{t+1} = \alpha\, \boldsymbol{z}_t + \tilde{\boldsymbol{c}}_s$. **b**) Out-of-context forecasting on the 2d Selkov system. When the initial condition lies outside of the context-covered region, neither parroting ($\alpha = 0$, left) nor pure flow mimicry ($\alpha = 1$, center) recovers the true vector field. Refitting α from a single additional short trajectory via eq. 13 produces a local flow estimate that aligns better with the true vector field (center).

a trajectory bit that contains a transient, however, yields a better approximation of the surrounding vector field outside the observed context while still reproducing the cycle (Fig. 6**b**, center).

5 Conclusion

We reduced DynaMix, a mixture-of-experts DSR foundation model, to *DynaBase*, a two-parameter nearest-neighbor affine recurrence that can be further reduced to a $1d$ family of maps by imposing a consistency condition. DynaBase exposes the minimal mechanisms necessary for zero-shot DSR by 1) closely following the context via its nearest-neighbor selector, thus generating TS with similar geometrical and temporal characteristics as represented in the data, while 2) regulating local divergence rates through a single parameter α, thus enabling a spectrum of context-guided dynamical behaviors from point attraction ($\alpha < 1$) and limit cycles ($\alpha = 1$) to chaos ($\alpha > 1$). Despite this minimal form, DynaBase matches or exceeds substantially larger architectures on both long-horizon DSR measures and short-term predictions.

Analysis of this minimal map and its training provides a number of interesting insights into the minimal requirements TS and DSR FMs may need to achieve zero-shot DS forecasts. First, $\alpha = 0$ exactly reproduces context parroting as observed in [99, 100], but this mechanism is inherently unable to reproduce chaotic or even true limit cycle behavior. Second, for $\alpha > 0$ this affine nearest-neighbor form admits the whole spectrum from convergence to fixed points or cycles to strictly chaotic activity for $\alpha > 1$, thus being sufficiently expressive to approximate a large range of DS. Third, for $\alpha \approx 1.01$ the map can even operate in zero-shot mode, at least on a common DSR benchmark set, exposing the profound role the context alone plays in structuring the forecast. Fourth, optimizing for ahead-predictions biases the map toward convergence to fixed or cyclic points, even when the underlying DS is chaotic, explaining why this training objective is suboptimal if DSR is the goal [65, 48, 76, 50]. More generally, with DynaBase we have established a formal framework that enables to study mechanisms of DSR in FMs and beyond in detail, creating an instrument for mathematical analysis of context-based forecasting and DSR methods.

Limitations DynaBase is a minimal 2-parameter model. Although it highlights the fact that complex architectures with tens of thousands to many millions of parameters may not be needed for DSR and time series forecasting, the assumptions on which its reduction is based come at a cost in terms of the diversity of DS and vector field topologies that can be learned, as well as the accuracy with which geometrical properties of the vector field can be captured (as apparent from the remaining mismatch in Fig. 6**b**). Currently, DynaBase may be viewed more as a research tool that allows for detailed mathematical analysis of context-based DSR FMs. A more in-depth study of DynaBase, e.g. its DS approximation capabilities or analytical computation of the full Lyapunov spectrum, which requires the concept of a saltation operator [30, 27], is still pending.

Acknowledgements

This work was supported by individual grants Du 354/15-1 (# 502196519) and Du 354/18-1 (# 567025973) from the German Research Foundation (DFG).

References

[1] Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? Investigations with linear models. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=0g0X4H8yN4I.

[2] 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.

[3] 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.

[4] Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Bernie Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=gerNCVqqtR.

[5] Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael Bohlke-Schneider. Chronos-2: From univariate to universal forecasting, 2025. URL https://arxiv.org/abs/2510.15821.

[6] 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.

[7] Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 57125–57211. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/b2e63e36c57e153b9015fece2352a9f9-Paper-Conference.pdf.

[8] Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xlstm: Extended long short-term memory. *Advances in Neural Information Processing Systems*, 37: 107547–107603, 2024.

[9] 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.

[10] Upinder S. Bhalla and Ravi Iyengar. Emergent properties of networks of biological signaling pathways. *Science*, 283(5400):381–387, 1999. doi: 10.1126/science.283.5400.381.

[11] Sathya Kamesh Bhethanabhotla, Omar Swelam, Julien Siems, David Salinas, and Frank Hutter. Mamba4cast: Efficient zero-shot time series forecasting with state space models. *arXiv preprint arXiv:2410.09385*, 2024.

[12] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1): 22–45, 2015.

[13] Alena Brändle, Lukas Eisenmann, Florian Götz, and Daniel Durstewitz. Continuous-time piecewise-linear recurrent neural networks. *arXiv preprint arXiv:2602.15649*, 2026.

[14] Manuel Brenner, Florian Hess, Jonas M. Mikhaeil, Leonard F. Bereska, Zahra Monfared, Po-Chen Kuo, and Daniel Durstewitz. Tractable Dendritic RNNs for Reconstructing Nonlinear Dynamical Systems. In *Proceedings of the 39th International Conference on Machine Learning*, pages 2292–2320. PMLR, June 2022. URL https://proceedings.mlr.press/v162/brenner22a.html. ISSN: 2640-3498.

[15] 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.

[16] 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.

[17] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

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

[19] 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.

[20] Steven L. Brunton, Bingni W. Brunton, Joshua L. Proctor, Eurika Kaiser, and J. Nathan Kutz. Chaos as an intermittently forced linear system. *Nature Communications*, 8(1):19, May 2017. ISSN 2041-1723. doi: 10.1038/s41467-017-00030-8. URL https://www.nature.com/articles/s41467-017-00030-8. Publisher: Nature Publishing Group.

[21] 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.

[22] Gyorgy Buzsaki. *Rhythms of the Brain*. Oxford University Press, August 2006. ISBN 978-0-19-804125-2. Google-Books-ID: ldz58irprjYC.

[23] Thomas L. Carroll and Louis M. Pecora. Network Structure Effects in Reservoir Computers. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(8):083130, August 2019. ISSN 1054-1500, 1089-7682. doi: 10.1063/1.5097686.

[24] Martin Casdagli. Nonlinear prediction of chaotic time series. *Physica D: Nonlinear Phenomena*, 35(3):335–356, 1989.

[25] 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.

[26] Siyu Chen, Heejune Sheen, Tianhao Wang, and Zhuoran Yang. Unveiling induction heads: Provable training dynamics and feature learning in transformers. *Advances in Neural Information Processing Systems*, 37:66479–66567, 2024.

[27] Stephen Coombes, Mustafa Şayli, R"udiger Thul, Rachel Nicks, Mason A. Porter, and Yi Ming Lai. Oscillatory networks: Insights from piecewise-linear modeling. *SIAM Review*, 66(4):619–679, 2024. doi: 10.1137/22M1534365.

[28] J. Daafouz, P. Riedinger, and C. Iung. Stability analysis and control synthesis for switched systems: a switched Lyapunov function approach. *IEEE Transactions on Automatic Control*, 47(11):1883–1887, November 2002. ISSN 1558-2523. doi: 10.1109/TAC.2002.804474. URL https://ieeexplore.ieee.org/document/1047016. Conference Name: IEEE Transactions on Automatic Control.

[29] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*, 2024.

[30] Mario di Bernardo, Chris J. Budd, Alan R. Champneys, and Piotr Kowalczyk. *Piecewise-Smooth Dynamical Systems: Theory and Applications*, volume 163 of *Applied Mathematical Sciences*. Springer, London, 2008. ISBN 978-1-84628-039-9. doi: 10.1007/978-1-84628-708-4.

[31] 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.

[32] 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.

[33] 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. *arXiv preprint arXiv:2602.16864*, 2026.

[34] J-P Eckmann, S Oliffson Kamphorst, David Ruelle, and Sergio Ciliberto. Liapunov exponents from time series. *Physical Review A*, 34(6):4971, 1986.

[35] Ezra Edelman, Nikolaos Tsilivis, Benjamin L Edelman, Eran Malach, and Surbhi Goel. The evolution of statistical induction heads: In-context learning markov chains. *Advances in neural information processing systems*, 37:64273–64311, 2024.

[36] J. Doyne Farmer and John J. Sidorowich. Predicting chaotic time series. *Physical Review Letters*, 59(8):845–848, August 1987. doi: 10.1103/PhysRevLett.59.845. URL https://link.aps.org/doi/10.1103/PhysRevLett.59.845. Publisher: American Physical Society.

[37] Jerome H Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, 3(3):209–226, 1977.

[38] L. Fumagalli, K. Lüdge, J. de Wiljes, H. Haario, and L. Jaurigue. Data-driven performance measures using global properties of attractors for testing black-box surrogate models of chaotic systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 35(11):113121, 11 2025. ISSN 1054-1500. doi: 10.1063/5.0283424. URL https://doi.org/10.1063/5.0283424.

[39] Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

[40] William Gilpin. Chaos as an interpretable benchmark for forecasting and data-driven modelling. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2022. URL https://openreview.net/forum?id=enYjtbjYJrf.

[41] William Gilpin. Model scale versus domain knowledge in statistical forecasting of chaotic systems. *Physical Review Research*, 5(4):043252, 2023.

[42] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First conference on language modeling*, 2024.

[43] 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.

[44] 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.

[45] Xinyu Han, Yi Zhao, and Michael Small. A tighter generalization bound for reservoir computing. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 32(4):043115, April 2022. ISSN 1054-1500. doi: 10.1063/5.0082258.

[46] 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.

[47] Christoph Jürgen Hemmer, Manuel Brenner, Florian Hess, and Daniel Durstewitz. Optimal Recurrent Network Topologies for Dynamical Systems Reconstruction. In *Proceedings of the 41st International Conference on Machine Learning*, pages 18174–18204. PMLR, July 2024. URL https://proceedings.mlr.press/v235/hemmer24a.html. ISSN: 2640-3498.

[48] 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.

[49] Hansika Hewamalage, Klaus Ackermann, and Christoph Bergmeir. Forecast evaluation for data scientists: common pitfalls and best practices. *Data Mining and Knowledge Discovery*, 37(2):788–832, 2023. URL https://link.springer.com/content/pdf/10.1007/s10618-022-00894-5.pdf?pdf=button. (IF 4.8, Q2 in "Computer Science, Artificial Intelligence", JCR 2022).

[50] 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.

[51] Alexia Jolicoeur-Martineau. Less is more: Recursive reasoning with tiny networks, 2025. URL https://arxiv.org/abs/2510.04871.

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

[53] 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.

[54] 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.

[55] Georgia Koppe, Hazem Toutounji, Peter Kirsch, Stefanie Lis, and Daniel Durstewitz. Identifying nonlinear dynamical systems via generative recurrent neural networks with applications to fMRI. *PLOS Computational Biology*, 15(8):e1007263, 2019. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1007263.

[56] Daniel Kramer, Philine L Bommer, Carlo Tombolini, Georgia Koppe, and Daniel Durstewitz. Reconstructing nonlinear dynamical systems from multi-modal time series. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 11613–11633. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/kramer22a.html.

[57] Jeffrey Lai, Anthony Bao, and William Gilpin. Panda: A pretrained forecast model for universal representation of chaotic dynamics. *arXiv preprint arXiv:2505.13755*, 2025.

[58] Ji-An Li, Marcus K. Benna, and Matthew G. Mattar. Discovering cognitive strategies with tiny recurrent neural networks. *Nature*, 644:993–1001, 2025. doi: 10.1038/s41586-025-09142-4.

[59] Ziqian Lin and Kangwook Lee. Dual operating modes of in-context learning. In *Forty-first International Conference on Machine Learning*, 2024.

[60] Jean-Christophe Loiseau and Steven L. Brunton. Constrained sparse Galerkin regression. *Journal of Fluid Mechanics*, 838:42–67, March 2018. ISSN 0022-1120, 1469-7645. doi: 10.1017/jfm.2017.823. Publisher: Cambridge University Press.

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

[62] Edward N Lorenz. Predictability: A problem partly solved. In *Proc. Seminar on predictability*, volume 1, pages 1–18. Reading, 1996.

[63] Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nat Commun*, 9(1):4950, December 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-07210-0. URL http://arxiv.org/abs/1712.09707. arXiv: 1712.09707.

[64] Benoit Mandelbrot and Richard L Hudson. *The Misbehavior of Markets: A fractal view of financial turbulence*. Basic books, 2007.

[65] 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.

[66] Zahra Monfared and Daniel Durstewitz. Transformation of ReLU-based recurrent neural networks from discrete-time to continuous-time. In *Proceedings of the 37th International Conference on Machine Learning*, 2020. URL http://proceedings.mlr.press/v119/monfared20a.html.

[67] Peter J. Mumby, Alan Hastings, and Helen J. Edwards. Thresholds and the resilience of caribbean coral reefs. *Nature*, 450(7166):98–101, 2007. doi: 10.1038/nature06252. URL https://doi.org/10.1038/nature06252.

[68] 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.

[69] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

[70] Samuel E. Otto and Clarence W. Rowley. Linearly recurrent autoencoder networks for learning dynamics. *SIAM Journal on Applied Dynamical Systems*, 18(1):558–593, 2019. doi: 10.1137/18M1177846. URL https://doi.org/10.1137/18M1177846.

[71] Matthijs Pals, A Erdem Sağtekin, Felix Pei, Manuel Gloeckler, and Jakob H Macke. Inferring stochastic low-rank recurrent neural networks from neural data. 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 18225–18264. Curran Associates, Inc., 2024. doi: 10.52202/079017-0579.

[72] 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.

[73] 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.

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

[75] 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.

[76] 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.

[77] Michael T Rosenstein, James J Collins, and Carlo J De Luca. A practical method for calculating largest lyapunov exponents from small data sets. *Physica D: Nonlinear Phenomena*, 65(1-2): 117–134, 1993.

[78] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019. doi: 10.1038/s42256-019-0048-x.

[79] T Konstantin Rusch, Siddhartha Mishra, N Benjamin Erichson, and Michael W Mahoney. Long expressive memory for sequence modeling. In *International Conference on Learning Representations*, 2022.

[80] Masaki Sano and Yasuji Sawada. Measurement of the lyapunov spectrum from a chaotic time series. *Physical review letters*, 55(10):1082, 1985.

[81] Tim Sauer, James A Yorke, and Martin Casdagli. Embedology. *Journal of statistical Physics*, 65(3):579–616, 1991.

[82] Gerwin Schalk, Dennis J. McFarland, Thilo Hinterberger, Niels Birbaumer, and Jonathan R. Wolpaw. BCI2000: a general-purpose brain-computer interface (BCI) system. *IEEE transactions on bio-medical engineering*, 51(6):1034–1043, 2000. ISSN 0018-9294. doi: 10.1109/TBME.2004.827072.

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

[84] Lingfeng Shen, Aayush Mishra, and Daniel Khashabi. Position: Do pretrained transformers learn in-context by gradient descent? In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 44712–44740. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/shen24d.html.

[85] Zhendong Sun. *Switched Linear Systems: Control and Design*. Springer Science & Business Media, March 2006. ISBN 978-1-84628-131-0. Google-Books-ID: u4GArZN1bmsC.

[86] Floris Takens. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick 1980*, volume 898, pages 366–381. Springer, 1981. ISBN 978-3-540-11171-9 978-3-540-38945-3. URL http://link.springer.com/10.1007/BFb0091924.

[87] Adam P. Trischler and Gabriele M.T. D'Eleuterio. Synthesis of recurrent neural networks for dynamical system simulation. *Neural Networks*, 80:67–78, 2016. ISSN 08936080. doi: 10.1016/j.neunet.2016.04.001. URL https://linkinghub.elsevier.com/retrieve/pii/S0893608016300314.

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

[89] Ayesha Vermani, Josue Nassar, Hyungju Jeon, Matthew Dowling, and Il Memming Park. Meta-dynamical state space models for integrative neural data analysis. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=SRpq50BpED.

[90] 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. *Proc. R. Soc. A.*, 474(2213):20170844, 2018. ISSN 1364-5021, 1471-2946. doi: 10.1098/rspa.2017.0844. URL https://royalsocietypublishing.org/doi/10.1098/rspa.2017.0844.

[91] Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 35151–35174. PMLR, 2023. URL https://proceedings.mlr.press/v202/von-oswald23a.html.

[92] Guan Wang, Jin Li, Yuhao Sun, Xing Chen, Changling Liu, Yue Wu, Meng Lu, Sen Song, and Yasin Abbasi Yadkori. Hierarchical reasoning model, 2025. URL https://arxiv.org/abs/2506.21734.

[93] Qixun Wang, Yifei Wang, Xianghua Ying, and Yisen Wang. Can in-context learning really generalize to out-of-distribution tasks? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=INe4otjryz.

[94] 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].

[95] Weimin Wu, Teng-Yun Hsiao, Jerry Yao-Chieh Hu, Wenxin Zhang, and Han Liu. In-context learning as conditioned associative memory retrieval. 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 67300–67325. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/wu25k.html.

[96] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

[97] Dmitry Yarotsky. Error bounds for approximations with deep relu networks. *Neural networks*, 94:103–114, 2017.

[98] 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.

[99] Yuanzhao Zhang and William Gilpin. Zero-shot forecasting of chaotic systems. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=TqYjhJrp9m.

[100] Yuanzhao Zhang and William Gilpin. Context parroting: A simple but tough-to-beat baseline for foundation models in scientific machine learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=EUAXc9Hlvm.

A Appendix

A.1 Methodological details

Training details Training of the DynaBase parameters (α, β) uses a sampled training sequence $\boldsymbol{X}_{\text{train}} \in \mathbb{R}^{N\times T}$ drawn from a DS, where the first T_C steps serve as context signal $\boldsymbol{C}$. For the experiments in Fig. 1, we set $T = 2000$ and $T_C = 1000$. At evaluation time, we supplied the training sequence as context alongside the estimated parameter values (α, β), and assessed performance on a sequence unseen during training and context construction (using the self-consistent DynaBase f_α as defined in eq. 12). This ensures that DynaBase receives exactly the same context and training sequence information as all other comparison methods.

To fit DynaBase via **linear regression**, we compute (α, β) in closed form via ordinary least-squares. Nearest neighbors $\tau(\cdot)$ are precomputed once via ℓ^2-distances over $\boldsymbol{C}$ and the training sequence $\boldsymbol{X}$. To cast this as a standard linear regression problem, we define the response vector $\boldsymbol{b}$ and design matrix $\boldsymbol{A}$ based on eq. 10, for each training sample i

$$\boldsymbol{b}_i = \boldsymbol{z}_{i+1} - \boldsymbol{c}_{\tau(\boldsymbol{z}_i)+1}, \tag{16}$$

$$\boldsymbol{A}_i = \Big[\, \boldsymbol{z}_i - \boldsymbol{c}_{\tau(\boldsymbol{z}_i)+1} \;,\; \boldsymbol{c}_{\tau(\boldsymbol{z}_i)} - \boldsymbol{c}_{\tau(\boldsymbol{z}_i)+1} \,\Big]. \tag{17}$$

The regression then takes the form $\boldsymbol{b} = \boldsymbol{A}[\alpha, \beta]^\top$, and the coefficients are obtained by the standard least-squares solution

$$[\alpha, \beta]^\top = (\boldsymbol{A}^\top \boldsymbol{A})^{-1} \boldsymbol{A}^\top \boldsymbol{b}. \tag{18}$$

In the self-consistent setting of eq. 12, the constraint $\beta = -\alpha$ reduces the two-parameter fit to a single unknown scalar, simplifying $\boldsymbol{A}_i = \boldsymbol{z}_i - \boldsymbol{c}_{\tau(\boldsymbol{z}_i)}$.

To train parameters via **grid search**, we sweep across $\alpha \in [0, 1.2]$ with step size $\Delta\alpha = 0.002$ along the self-consistency line $\beta = -\alpha$. For each candidate α, DynaBase is rolled out for $T = 10{,}000$ steps using the context $\boldsymbol{C}$ and scored by the KL-divergence loss given in eq. 14 between predicted trajectories $\hat{\boldsymbol{X}}$ and the training data. The global minimizer α is returned.

Training and evaluation of DynaBase was performed on a single CPU (AMD EPYC 9655 96-Core).

Comparison methods We compare DynaBase against three groups of baselines, all evaluated on the same 54 test DS and the same input data of length $T = 2000$ as in [46].

Zero-shot foundation models: DynaMix [46], the DSR FM that DynaBase is reduced from (mixture-of-experts of AL-RNNs with context-dependent gating; pretrained on $\sim 6 \times 10^5$ trajectories from 34 DS); Panda [57], a zero-shot DS forecasting model trained on simulated DS data; Chronos-T5-base [4], a transformer FM that quantizes real-valued series into discrete tokens and forecasts autoregressively on a corpus of synthetic and real-world time series; and Chronos-2 [5], its multivariate extension augmenting the T5 backbone with a group-attention mechanism. Evaluations performed as in [46].

Custom-trained DSR models: An AL-RNN [15] (eq. 1), which is used for the experts in DynaMix, a Neural-ODE [25], and a reservoir computer [72]; each trained from scratch on the same $T = 2000$ context segment provided to the foundation models. Evaluations as in [46].

Context parroting: The algorithm of Zhang and Gilpin [100], which retrieves the best-matching motif in $\boldsymbol{C}$ and copies its successor as forecast. As proven in Thm. 2, this baseline corresponds to the $\alpha = 0$ case of DynaBase on a delay-embedded context.

A.2 Performance measures

Long-term metrics To evaluate the *geometric similarity* between true and model-generated reconstructions, we employ a measure, D_{stsp}, based on the Kullback-Leibler (KL) divergence evaluated in state space [55, 65, 41, 71, 99, 57, 46]. This measure quantifies the (mis)match between the ground-truth spatial distribution of trajectory points, $p_{\text{true}}(\boldsymbol{x})$, and the distribution $p_{\text{gen}}(\boldsymbol{x}|\boldsymbol{z})$ of points from trajectories freely generated by the model:

$$D_{\text{stsp}}\big(p_{\text{true}}(\boldsymbol{x}) \,\|\, p_{\text{gen}}(\boldsymbol{x}|\boldsymbol{z})\big) = \int p_{\text{true}}(\boldsymbol{x}) \log \frac{p_{\text{true}}(\boldsymbol{x})}{p_{\text{gen}}(\boldsymbol{x}|\boldsymbol{z})}\, d\boldsymbol{x}. \tag{19}$$

In practice, we approximate this via a discrete binning of the state space into $K = m^N$ bins, where m is the number of bins per dimension and N is the system dimensionality, estimating occupation probabilities through relative frequency of visits $\hat{p}_i$:

$$D_{\text{stsp}} \approx \sum_{i=1}^{K} \hat{p}_{\text{true};i} \log \frac{\hat{p}_{\text{true};i}}{\hat{p}_{\text{gen};i}}. \tag{20}$$

To ensure that the system has reached a steady-state distribution, long trajectories ($T = 10000$) are sampled from the trained model. For the 3d attractors we set $m = 30$ bins per dimension, following [47].

To evaluate long-term *temporal* reconstruction quality, we calculate the Hellinger distance D_H between the power spectra of the ground-truth and model-generated time series [65, 48, 71, 33]. We first apply dimension-wise Fast Fourier Transforms (FFT) to the time series and then smooth the resulting power spectra with a Gaussian kernel (with $\sigma = 20$), followed by normalization to enable comparison across dimensions. High-frequency tails dominated by noise are removed. The Hellinger distance between the smoothed true spectrum $F(\omega)$ and the generated spectrum $G(\omega)$ is given by

$$D_H\big(F(\omega), G(\omega)\big) = \sqrt{1 - \int_{-\infty}^{\infty} \sqrt{F(\omega)\, G(\omega)}\, d\omega} \;\in [0, 1], \tag{21}$$

where values close to 0 indicate high spectral agreement and values close to 1 indicate complete disagreement. D_H is reported as the average across all dimensions. Note that both D_{stsp} and D_H are designed to capture *long-term* reconstruction quality in the limit $T \to \infty$, after transients have settled and the attractor geometry has been traced out reasonably well (which was the case here for $T = 10000$ steps).

Short-term forecast quality For evaluating short-term predictive performance, we use the Mean Absolute Scaled Error (MASE) [49, 46], which is a normalized n-step-ahead prediction error aggregated across all N output dimensions:

$$\text{MASE} = \frac{1}{N} \sum_{i=1}^{N} \frac{\frac{1}{n} \sum_{t=1}^{n} \big| x_{i,t} - \hat{x}_{i,t} \big|}{\frac{1}{T} \sum_{t=1}^{T} \big| x_{i,t} - x_{i,t-1} \big|}, \tag{22}$$

where T denotes the total length of the ground truth time series, n is the forecast horizon, $x_{i,t}$ the true value, and $\hat{x}_{i,t}$ the model prediction for dimension i at time t. We evaluate at $n = 10$ steps.

Lyapunov exponent To numerically estimate the *maximum Lyapunov exponent* λ_{max} from a time series $\boldsymbol{X}$, we use the *Rosenstein algorithm* [77]. For each state $\boldsymbol{x}_i$, its nearest neighbor $\boldsymbol{x}_j$ is identified subject to $|i - j| > l_t$, which excludes direct temporal neighbors on the same trajectory segment. We set $l_t = 100$ (based on the mean periodicity [77]). The divergence between neighboring trajectories is then tracked over time steps $k = 0, \ldots, k_{\max}$,

$$d_i(k) = \|\boldsymbol{x}_{i+k} - \boldsymbol{x}_{j(i)+k}\|_2 \; . \tag{23}$$

Since nearby trajectories diverge as

$$d_i(k) \approx d_i(0)\, e^{\lambda_{\max} k \Delta t} \; , \tag{24}$$

where Δt is the sampling interval, taking the logarithm and averaging over all pairs gives

$$\langle \ln d(k) \rangle \approx \lambda_{\max} k \Delta t + C \; . \tag{25}$$

For chaotic systems, this quantity grows approximately linearly with k for a suitable $k_{\max}$ (here: 100) [52], with slope $\lambda_{\max} > 0$, whereas stable periodic systems yield $\lambda_{\max} = 0$.

When the equations of motion $\dot{\mathbf{x}} = f(\mathbf{x})$ are known, the full Lyapunov spectrum $\{\lambda_i\}$ can be computed via a *QR decomposition-based algorithm* [9], which integrates the system along its variational equations $\dot{Q} = J(\mathbf{x})Q$, where $J(\boldsymbol{x}) = \partial f / \partial \boldsymbol{x}$ is the Jacobian of f and Q is an $n \times n$ matrix

of perturbation vectors initialized to the identity I_n. At regular time intervals, a QR decomposition $Q = \tilde{Q}R$ is performed and the columns of $\tilde{Q}$, which form an orthonormal basis, replace Q. The diagonal entries R_{ii} record the local expansion rates, and the k-th Lyapunov exponent is obtained as the time-averaged logarithmic growth rate,

$$\lambda_k = \lim_{T\to\infty} \frac{1}{T} \sum_j \log \left| R_{kk}^{(j)} \right| , \tag{26}$$

where the sum runs over all QR factorizations performed up to time T. If the governing equations are not available, the full spectrum can instead be estimated numerically from the time series using the method of *Sano and Sawada* [80, 34]. For each state $\boldsymbol{x}_i$, its n_{nn} nearest neighbors $\boldsymbol{x}_j$ (for the Lorenz-96 experiment $n_{\text{nn}} = 100$) are used to fit a local Jacobian $\hat{J}_i$ by least squares based on how the displacements evolve over one time step,

$$\boldsymbol{x}_{j+1} - \boldsymbol{x}_{i+1} \approx \hat{J}_i \left(\boldsymbol{x}_j - \boldsymbol{x}_i \right) . \tag{27}$$

The estimated Jacobians are then used to propagate the perturbation vectors, $Q \leftarrow \hat{J}_i Q$, followed by the same QR re-orthonormalization as above.

A.3 Details on DynaMix reduction

Table 1: Zero-shot performance of DynaMix and model simplifications performed across all 54 test DS used in [46]. DynaMix and simplified models are compared for the same context length $T_C = 2000$. Reported are median±MAD for geometrical disagreement D_{stsp}, temporal disagreement D_H, and 10-step-ahead prediction error MASE.

Model	$D_{\text{stsp}} \downarrow$	$D_H \downarrow$	MASE $\downarrow$
Original DynaMix	3.83 ± 1.19	0.17 ± 0.08	0.35 ± 0.17
Linear experts & gating ($M > N$)	3.10 ± 1.19	0.13 ± 0.07	0.26 ± 0.11
Linear experts & gating ($M = N$)	2.85 ± 1.30	0.13 ± 0.06	0.23 ± 0.10
MLP approximation of bilinear term	3.19 ± 1.33	0.14 ± 0.07	0.34 ± 0.14
Pre-trained DynaMix with $\tau_{\text{att}} \to 0$	2.75 ± 1.05	0.12 ± 0.06	0.40 ± 0.20

Here we provide the supporting evidence and derivation details for the successive reduction of DynaMix to the *recursive affine NN map* eq. 7 described in Sec. 3.2.

Linear approximation Starting from eq. 4, DynaMix is a weighted sum of piecewise-linear experts F_j^{exp} with time-dependent expert weights $\boldsymbol{w}_t^{exp}$. To simplify the architecture we linearize both parts. Specifically, each expert is an almost-linear RNN (eq. 1) with piecewise-linear activation Φ^* on P units and linear mapping for the remaining $M - P$. If we replace Φ^* with the identity, each expert becomes an affine function of $\boldsymbol{z}_t$. The expert weights $\boldsymbol{w}_t^{exp}$ (eq. 2) are nonlinear due to the MLP and a softmax output. As previously observed in [46], the MLP can be replaced by a linear layer without loss in performance. Moreover, across the relevant domain, the softmax σ can be approximated by an affine mapping as well. This makes each expert affine in $\boldsymbol{z}_t$ and each weight affine in $(\boldsymbol{z}_t, \tilde{\boldsymbol{C}}\boldsymbol{w}_t^{att})$, so that their product is bilinear in $\boldsymbol{z}_t$. Table 1 (*Linear experts & gating, $M > N$*) empirically confirms that these approximations leave performance on the chaotic benchmark set unchanged.

Latent dimension The experts in DynaMix operate in an M-dimensional latent space distinct from the N-dimensional observation space, requiring the projection $\boldsymbol{D}$ in eq. 3. Replacing the experts with linear maps removes the need for such a distinction since no nonlinear latent units are required, yielding $M = N$ and $\boldsymbol{D} = \boldsymbol{I}$, such that the model operates entirely in observation space. Table 1 (*Linear experts & gating, $M = N$*) confirms this does not reduce performance on the present test set. Working in a single space is also a prerequisite for the $O(N)$ equivariance exploited in Thm. 1, since rotations and the nearest-neighbor distance must act on the same space in which the state $\boldsymbol{z}_t$ and the context $\boldsymbol{C}$ live. Note we can always perform a classical delay-embedding [86, 81] should this dimensionality not be sufficient to properly represent an empirically observed DS.

Universal approximation of the bilinear term We exploit the universal approximation properties of ReLU-based MLPs [97] to approximate the bilinear form in $\boldsymbol{z}_t$ and $\boldsymbol{w}^{exp}$. Ablations confirm that the MLP can, in turn, itself be reduced to a linear layer (see Table 1), leaving $\tilde{\boldsymbol{C}} = \text{CNN}(\boldsymbol{C})$ and $\boldsymbol{w}_t^{att}(\boldsymbol{z}_t, \boldsymbol{C})$ as the only nonlinearities. This yields the recurrent linearized map form eq. 6.

Nearest-neighbor selector In the publicly available DynaMix checkpoint, the learned attention temperature τ_{att} converged to $|\tau_{\text{att,learned}}| \approx 0.06$, indicating that the soft attention eq. 3 is already operating close to its $\tau_{\text{att}} \to 0$ limit, i.e. effectively selecting a single nearest context point at each step. Two deliberate departures from DynaMix are made in passing to the hard limit. First, DynaMix injects Gaussian exploration noise $\varepsilon \sim \mathcal{N}(0, \boldsymbol{\Sigma})$ into the attention eq. 3; in favor of a minimal and deterministic model, we set $\boldsymbol{\Sigma} = 0$. Second, DynaMix computes context distances with the ℓ^1-norm, whereas we adopt the Euclidean ℓ^2-norm. The invariance of ℓ^2 distances under orthogonal transformations $R \in O(N)$, is used in Thm. 1. Together these choices yield the deterministic selector $\tau(\boldsymbol{z}_t) = \arg\min_s \|\boldsymbol{c}_s - \boldsymbol{z}_t\|_2$.

CNN expansion To make the recursive map fully explicit in terms of individual context points, we expand the CNN operation. Since DynaMix uses a single linear CNN layer with kernel size 2, for $s \in 1 \dots T_C - 1$ we can write

$$\tilde{\boldsymbol{c}}_s = \text{CNN}(\boldsymbol{C})_s = \boldsymbol{K}^{(1)}\boldsymbol{c}_s + \boldsymbol{K}^{(2)}\boldsymbol{c}_{s+1} + \boldsymbol{b}, \tag{28}$$

where $\boldsymbol{K}^{(1)}, \boldsymbol{K}^{(2)} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{b} \in \mathbb{R}^N$ are learned weights. Substituting into eq. 6 with the nearest-neighbor selector and absorbing $\boldsymbol{B}\boldsymbol{K}^{(1)}$, $\boldsymbol{B}\boldsymbol{K}^{(2)}$ and the bias into $\boldsymbol{B}^{(1)}$, $\boldsymbol{B}^{(2)}$, and $\tilde{\boldsymbol{h}} = \boldsymbol{h} + \boldsymbol{B}\boldsymbol{b}$ respectively, yields the *recursive affine NN map* eq. 7 as stated in the main text.

A.4 Datasets

Our DS dataset consists of simulated time series of length 10^5, produced from $3d$ DS collected in [40]. For the test set, we used the same 54 distinct systems as in [46] to enable a fair comparison among all the baseline models in relation to the results published originally. To evaluate DynaBase in a zero-shot setting, we estimated α using a training set of 30 different chaotic DS, analogous to the setup in [46].

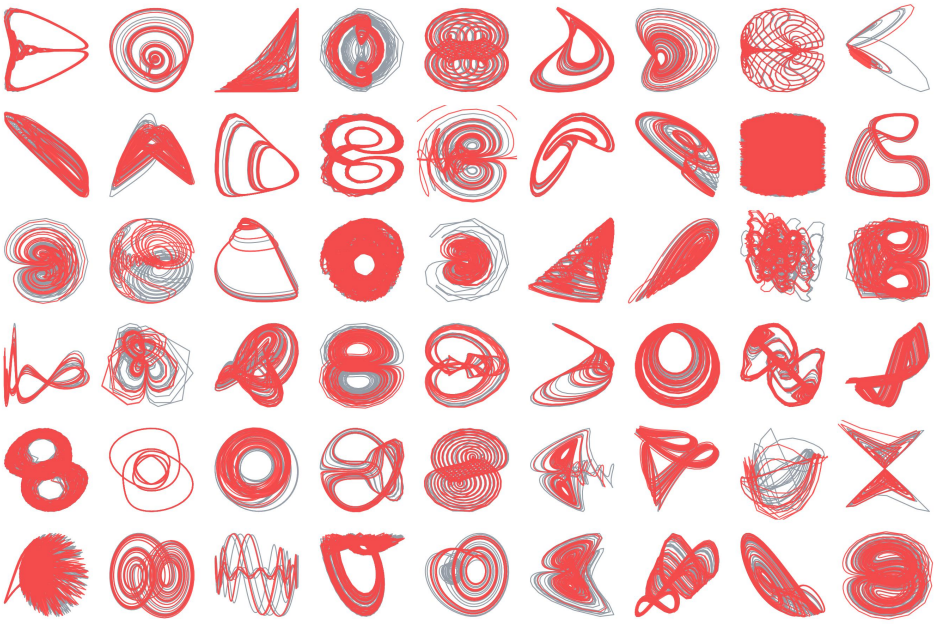

Figure 7: DSR test set of 54 different DS gathered from [40]. Ground truth (gray) and DynaBase reconstructions using a context window of $T_C = 2000$ (red).

In addition, we conduct experiments on the $2d$ Selkov system [83], which models the kinetics of an open monosubstrate enzyme reaction and is defined by the equations

$$\begin{aligned} \frac{dx}{dt} &= -x + ay + x^2 y, \\ \frac{dy}{dt} &= b - ay - x^2 y, \end{aligned} \tag{29}$$

where we chose $a = 0.1$ and $b = 0.5$. The system was solved numerically with integration time step $\Delta t = 0.3$ using `scipy.integrate` with the `RK45` solver.

For Fig. 14 we used the Lorenz-63 attractor [61], originally formulated by Edward Lorenz in 1963 [61] to model atmospheric convection. It is defined by

$$\begin{aligned} \frac{\mathrm{d}x}{\mathrm{d}t} &= \sigma(y - x) \\ \frac{\mathrm{d}y}{\mathrm{d}t} &= x(\rho - z) - y \\ \frac{\mathrm{d}z}{\mathrm{d}t} &= xy - \beta z, \end{aligned} \tag{30}$$

where σ, ρ, β, are control parameters of the system (we fix $\sigma = 10$ and $\rho = 28$, and vary $\beta \in [0, 1]$, producing regimes with point, cyclic, and chaotic attractors). The system was solved numerically with integration time step $\Delta t = 0.02$ using `scipy.integrate` with the `RK45` solver and was standardized afterward.

To test DynaBase on a higher-dimensional system with multiple time scales, we use the two-scale Lorenz-96 model [62], in which slow (X_k) and fast (Y_k) variables interact. We use $K = 5$ slow and corresponding fast variables, resulting in an $N = 10$ dimensional system,

$$\begin{aligned} \frac{\mathrm{d}X_k}{\mathrm{d}t} &= -X_{k-1}(X_{k-2} - X_{k+1}) - X_k + F - \frac{hc}{b} Y_k, \\ \frac{\mathrm{d}Y_k}{\mathrm{d}t} &= -cb\, Y_{k+1}(Y_{k+2} - Y_{k-1}) - c\, Y_k + \frac{hc}{b} X_k, \end{aligned} \tag{31}$$

with cyclic indices $k = 1, \ldots, K$. We set $F = 10$, $h = 1$, $c = 20$, and $b = 10$, yielding a hyperchaotic system with two positive Lyapunov exponents, $\lambda_{\max} \approx 3.6$. The system was solved numerically with a fixed-step fourth-order Runge-Kutta scheme with $\Delta t = 0.02$, and standardized per dimension.

For evaluation on empirical (real-world) TS, we adopt the same TS datasets and evaluation protocol as in [33], spanning multiple domains such as weather signals, traffic, human physiological signals, cloud requests, and the ETTh1 dataset. All data is publicly available and links can be found in [33] (ETTh1: https://github.com/zhouhaoyi/ETDataset; Traffic: https://www.kaggle.com/datasets/fedesoriano/traffic-prediction-dataset/data: Cloud Requests: https://github.com/sir-lab/data-release; Weather: https://www.dwd.de/EN/ourservices/cdc/cdc_ueberblick-klimadaten_en.html; fMRI: [56]; EEG: [82]).

A.5 Further empirical results

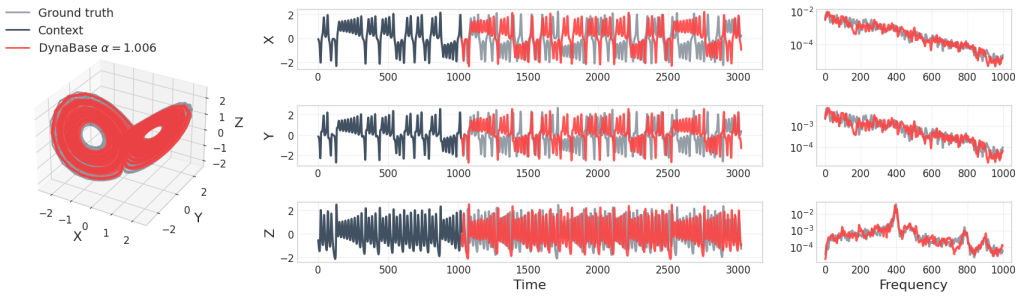

Figure 8: Zero-shot reconstruction of chaotic Lorenz-63 system using DynaBase with $\alpha = 1.006$. Left: State space, center: time graphs, right: power spectrum.

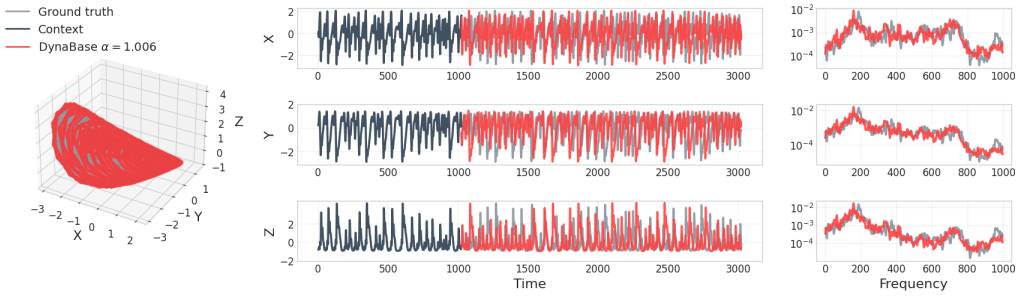

Figure 9: Zero-shot reconstruction of chaotic Sprott-F system using DynaBase with $\alpha = 1.006$. Left: State space, center: time graphs, right: power spectrum.

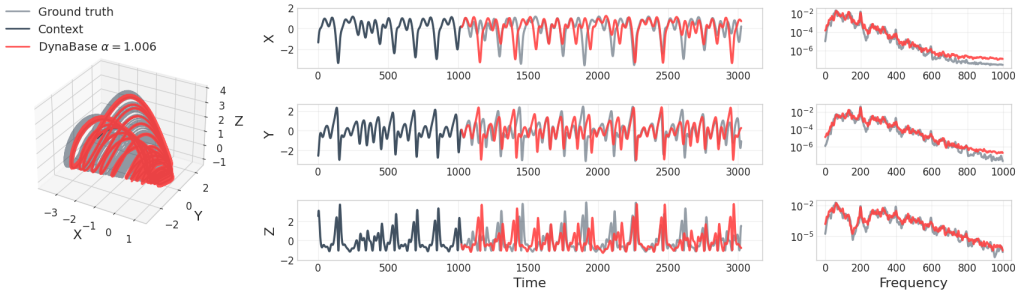

Figure 10: Zero-shot reconstruction of chaotic Sprott-D system using DynaBase with $\alpha = 1.006$. Left: State space, center: time graphs, right: power spectrum.

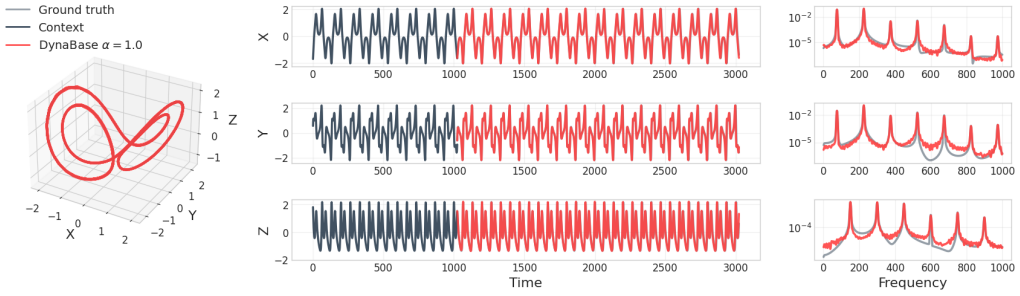

Figure 11: Zero-shot reconstruction using $\alpha = 1$ of cyclic Lorenz-63 system using DynaBase. Left: State space, center: time graphs, right: power spectrum.

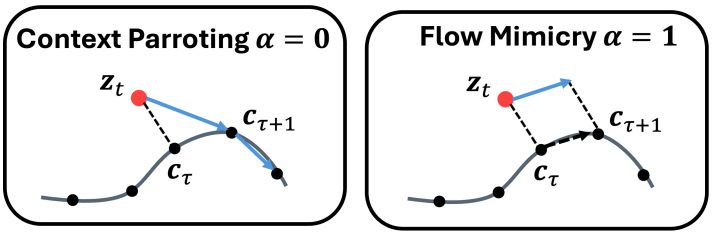

Figure 12: Illustration of context parroting and flow mimicry. **Left:** *context parroting* — at $(\alpha, \beta) = (0, 0)$ the state snaps onto the successor $\boldsymbol{c}_{\tau(\boldsymbol{z}_t)+1}$ of its nearest context neighbor, producing a piecewise-constant map that replays the context. **Right:** *flow mimicry* — at $(\alpha, \beta) = (1, -1)$ the state is updated by adding the finite-difference increment $\boldsymbol{c}_{\tau(\boldsymbol{z}_t)+1} - \boldsymbol{c}_{\tau(\boldsymbol{z}_t)}$ evaluated at the nearest context neighbor, a local Euler step on the context vector field.

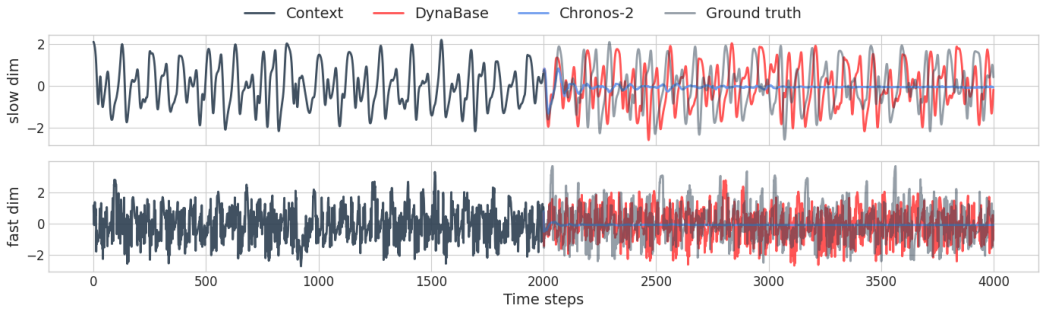

Figure 13: Prediction of the 10d, hyperchaotic, multiscale Lorenz-96 by DynaBase ($\alpha = 1.006$) and Chronos-2. Top shows a slowly and bottom a quickly evolving variable.

Table 2: Zero-shot performance of DynaBase and Chronos-2 on the 10d, hyperchaotic, multiscale Lorenz-96 system for context length $T_C = 2000$. Reported are geometrical disagreement D_{stsp} (due to the high dimensionality, the distributions were estimated via Gaussian mixture models as in [48]), temporal disagreement D_H, and 10-step-ahead prediction MASE.

Model	$D_{\text{stsp}} \downarrow$	$D_H \downarrow$	MASE $\downarrow$
DynaBase	0.19	0.12	1.01
Chronos-2	6.89	0.89	0.69

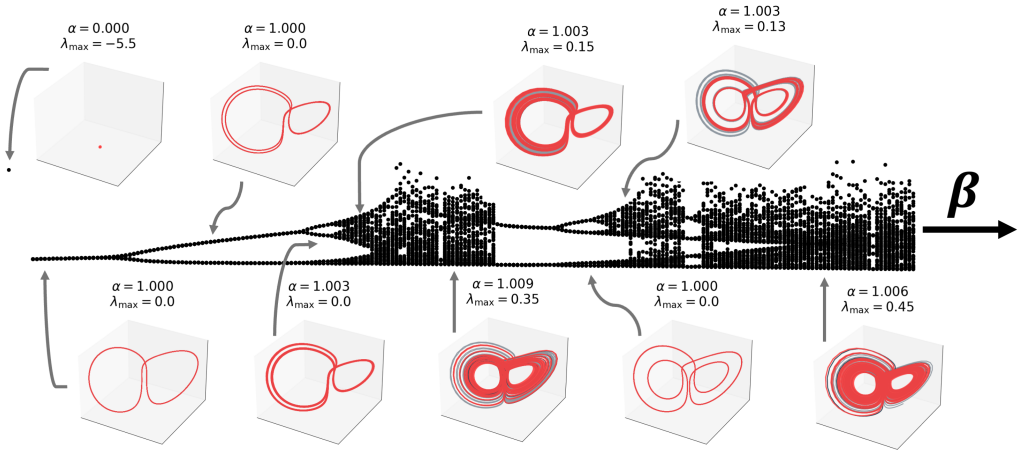

Figure 14: Different reconstructions by DynaBase along a bifurcation diagram of the Lorenz-63 system with system parameters $\sigma = 10$, $\rho = 28$ and $\beta \in [0, 1]$. For each reconstruction, the estimated alpha value together with the GT system's true Lyapunov exponent (estimated via QR decomposition, see Sec. A.2) is indicated.

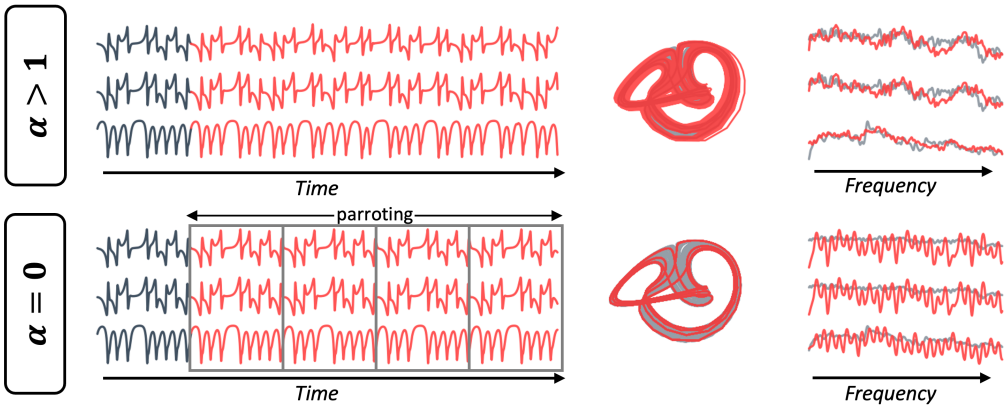

Figure 15: DynaBase forecasting of a chaotic laser system [40], using the same context once with $\alpha > 1$ and once with $\alpha = 0$. Note that despite the system's non-periodic context trajectory, parroting ($\alpha = 0$) incorrectly produces a discrete cycle. Shown are time graphs (left), state space (center), and power spectrum (right).

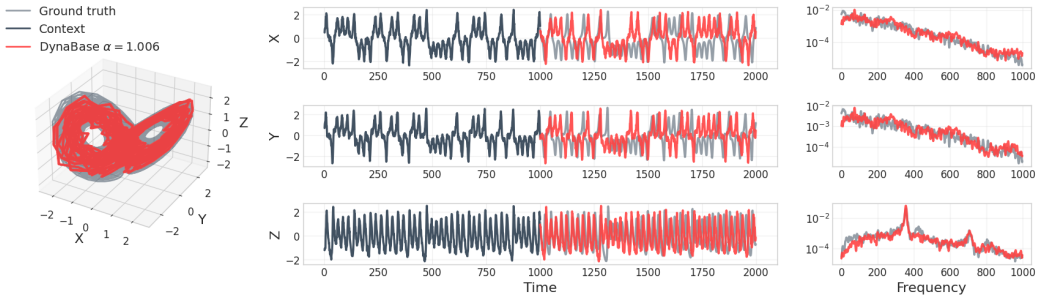

Figure 16: DynaBase reconstruction of Lorenz-63 adding 15% Gaussian noise to the context signal. Left: State space, center: time graphs, right: power spectrum.

Table 3: Short-term forecasting accuracy (MASE, median ± MAD) on the DS test set for three DynaBase training strategies (linear regression (LR), DSR objective, and zero-shot) across various prediction horizons n. Best median values in bold. Note that LR training outperforms the other approaches on this measure for intermediate forecasting horizons, while for larger horizons ($n = 300$) performance differences narrow down again due to the general limitations caused by exponential trajectory divergence in chaotic systems.

Model	MASE ($n = 10$)	MASE ($n = 50$)	MASE ($n = 100$)	MASE ($n = 200$)	MASE ($n = 300$)
DynaBase LR	$\mathbf{0.13 \pm 0.08}$	$\mathbf{0.51 \pm 0.40}$	$\mathbf{0.71 \pm 0.62}$	$\mathbf{1.06 \pm 0.93}$	$\mathbf{2.64 \pm 2.18}$
DynaBase DSR & Gating	$\mathbf{0.13 \pm 0.08}$	0.62 ± 0.48	1.06 ± 0.89	2.22 ± 1.64	2.80 ± 1.96
DynaBase zero-shot	$\mathbf{0.13 \pm 0.07}$	0.61 ± 0.48	0.94 ± 0.80	2.29 ± 1.69	3.07 ± 2.09

Table 4: Zero-shot performance of DynaBase across all 54 test DS used in [46], with the x-dimension scaled by a factor of 2. Reported are median±MAD for geometrical disagreement D_{stsp}, temporal disagreement D_H, 10-step-ahead prediction MASE, and α estimated via grid search.

Model	$D_{\text{stsp}} \downarrow$	$D_H \downarrow$	MASE $\downarrow$	α
DynaBase (original)	1.76 ± 0.62	0.06 ± 0.03	0.13 ± 0.08	1.009 ± 0.006
DynaBase (x-dim. scaled $\times 2$)	1.74 ± 0.60	0.06 ± 0.03	0.14 ± 0.10	1.009 ± 0.006

Table 5: Zero-shot performance of DynaBase across all 54 test DS used in [46] for different context lengths T_C. Reported are median±MAD for geometrical disagreement D_{stsp}, temporal disagreement D_H, 10-step-ahead prediction MASE, and α estimated via grid search.

Model	$D_{\text{stsp}} \downarrow$	$D_H \downarrow$	MASE $\downarrow$	α
DynaBase ($T_C = 1000$)	2.70 ± 1.03	0.07 ± 0.05	0.22 ± 0.16	1.006 ± 0.006
DynaBase ($T_C = 2000$)	1.76 ± 0.62	0.06 ± 0.03	0.13 ± 0.08	1.009 ± 0.006
DynaBase ($T_C = 3000$)	1.31 ± 0.50	0.05 ± 0.03	0.14 ± 0.09	1.009 ± 0.006

Table 6: Zero-shot performance of DynaBase across all 54 test DS used in [46] for partially observed DS, using delay-embedding of only the x-dimension. Reported are median±MAD for geometrical disagreement D_{stsp}, temporal disagreement D_H, 10-step-ahead prediction MASE, and α estimated via grid search.

Model	$D_{\text{stsp}} \downarrow$	$D_H \downarrow$	MASE $\downarrow$	α
DynaBase (fully observed)	1.76 ± 0.62	0.06 ± 0.03	0.13 ± 0.08	1.009 ± 0.006
DynaBase (partially observed)	1.96 ± 1.04	0.06 ± 0.04	0.16 ± 0.12	1.014 ± 0.006

Table 7: Zero-shot performance of DynaBase across all 54 test DS used in [46] for different levels of Gaussian observation noise on the context signal. Reported are median±MAD for geometrical disagreement D_{stsp}, temporal disagreement D_H, 10-step-ahead prediction MASE, and α estimated via grid search.

Model	$D_{\text{stsp}} \downarrow$	$D_H \downarrow$	MASE $\downarrow$	α
DynaBase (0% noise)	1.76 ± 0.62	0.06 ± 0.03	0.13 ± 0.08	1.009 ± 0.006
DynaBase (5% noise)	2.17 ± 0.67	0.07 ± 0.02	0.77 ± 0.39	1.020 ± 0.001
DynaBase (10% noise)	3.09 ± 0.89	0.10 ± 0.02	1.34 ± 0.55	1.017 ± 0.003

Table 8: Forecasting results of DynaMix and DynaBase on different real world time series. Reported are median±MAD (for DynaMix) for geometrical disagreement D_{stsp}, temporal disagreement D_H, 10-step-ahead prediction MASE.

System	**DynaMix**			**DynaBase**		
	D_{stsp}	D_H	MASE	D_{stsp}	D_H	MASE
ETTh1	0.59 ± 0.002	0.15 ± 0.003	$\mathbf{1.55 \pm 0.02}$	**0.12**	**0.13**	1.68
Traffic	0.85 ± 0.02	0.18 ± 0.005	1.20 ± 0.44	**0.18**	**0.12**	**1.09**
Cloud Requests	0.68 ± 0.01	0.07 ± 0.006	$\mathbf{2.64 \pm 0.12}$	**0.05**	**0.05**	3.59
Weather (Temp.)	0.06 ± 0.002	$\mathbf{0.07 \pm 0.003}$	$\mathbf{2.04 \pm 0.07}$	**0.03**	0.08	3.17
Weather (Press.)	0.19 ± 0.004	$\mathbf{0.26 \pm 0.008}$	$\mathbf{4.32 \pm 0.08}$	**0.14**	0.27	6.75
Human fMRI	0.24 ± 0.03	0.20 ± 0.004	$\mathbf{2.03 \pm 0.13}$	**0.20**	**0.19**	3.02
Human EEG	0.20 ± 0.01	0.24 ± 0.02	7.56 ± 0.25	**0.16**	**0.21**	**4.30**

A.6 Further theoretical results

Details on continuous-time limit. We begin by writing eq. 12 explicitly as a piecewise-affine discrete-time system

$$\boldsymbol{z}_{k+1} = f_\alpha(\boldsymbol{z}_k) = \alpha \boldsymbol{I} \boldsymbol{z}_k + (\boldsymbol{c}_{\tau(\boldsymbol{z}_k)+1} - \alpha \boldsymbol{c}_{\tau(\boldsymbol{z}_k)}) \,. \tag{32}$$

For values of $\alpha \in \mathbb{R} \setminus \{0, 1\}$, Thm. 3 in [66] guarantees the existence of an equivalent affine continuous-time system, in the sense that

$$\boldsymbol{\zeta}(t_0) = \boldsymbol{z}_{k_0}, \; \boldsymbol{\zeta}(t_0 + h) = \boldsymbol{z}_{k_0+1} \,, \tag{33}$$

where h denotes the discrete timestep at which observations $\boldsymbol{z}_k$ are sampled. The equivalent continuous-time system is then given by

$$\frac{\mathrm{d}\boldsymbol{\zeta}}{\mathrm{d}t} = \frac{\log \alpha}{h} \boldsymbol{\zeta} - \frac{1}{h} \frac{\log \alpha}{1-\alpha} (\boldsymbol{c}_{\tau(\boldsymbol{\zeta})+1} - \alpha \boldsymbol{c}_{\tau(\boldsymbol{\zeta})}) \,. \tag{34}$$

For the case $\alpha = 1$, we define $\frac{\log \alpha}{1-\alpha}|_{\alpha=1} := -1$ by its continuous extension. Such an extension does, however, not exist for $\alpha = 0$. For $\alpha < 0$, the logarithm is complex-valued. Hence, a real-valued continuous-time limit for f_α exists only for $\alpha > 0$.

Computational complexity The minimal architecture of DynaBase makes it computationally highly efficient for zero-shot inference. By exploiting discrete nearest-neighbor lookup and efficient heuristic search algorithms such as k-d trees [37], the time complexity of running DynaBase autoregressively over T_F forecast steps is

$$\mathcal{T} = \mathcal{O}(T_F \log T_C), \tag{35}$$

with a one-time overhead of $\mathcal{O}(T_C \log T_C)$ for constructing the k-d tree. In higher dimensions, query time degrades toward $\mathcal{O}(T_C)$ in the worst case. Provided the generated trajectory is not retained, the space complexity reduces to

$$\mathcal{S} = \mathcal{O}(T_C). \tag{36}$$

These bounds allow DynaBase to generate long forecasts over extended contexts at low computational cost — in contrast to standard transformer architectures, which incur quadratic time and space complexity in context length.

A.7 Alternative geometric optimization criterion

One may argue that using D_{stsp} for both training and evaluation yields an unfair advantage for DynaBase compared to other methods on this measure. Although this was a deliberate choice to expose the importance of the training criterion (sect. 4.2), and the same might be said about MASE (on which TS FMs are trained) the other way round, here we study how much performance degrades if DynaBase training is performed on a different geometrical criterion than used for evaluation. Specifically, we used the sliced Wasserstein-1 distance D_W between $p_{\text{true}}(\boldsymbol{x})$ and $p_{\text{gen}}(\boldsymbol{x}|\boldsymbol{z})$ [12, 33] for training. It averages one-dimensional Wasserstein-1 distances over $L = 200$ random projections $\boldsymbol{\theta}_l \in \mathbb{S}^{N-1}$:

$$D_W \approx \frac{1}{L} \sum_{l=1}^{L} W_1\big(\boldsymbol{\theta}_l^\top \boldsymbol{x}_{\text{true}}, \, \boldsymbol{\theta}_l^\top \boldsymbol{x}_{\text{gen}}\big). \tag{37}$$

As shown in Table 9, changing the objective to another DSR criterion leads to similar performance, and still outperforms most other models, including training via linear regression.

Table 9: Reconstruction results on whole 54 DS benchmark dataset for different training objectives (linear regression, KL-divergence D_{stsp}, see eq. 14, or Wasserstein-1 distance, see eq. 37) using grid search. Median±MAD.

Model	$D_{\text{stsp}} \downarrow$	$D_H \downarrow$	MASE $\downarrow$	α
DynaBase LR	4.34 ± 2.27	0.06 ± 0.05	0.13 ± 0.08	0.960 ± 0.031
DynaBase DSR (D_{stsp})	1.76 ± 0.62	0.06 ± 0.03	0.13 ± 0.08	1.009 ± 0.006
DynaBase-DSR (Wasserstein)	2.16 ± 1.16	0.06 ± 0.04	0.13 ± 0.08	1.003 ± 0.007

A.8 Comparison to classical nonlinear forecasting and local approximation methods

We further compare DynaBase to classical, library-based local prediction methods for chaotic time series [36, 24, 52]. For all of them we provide the context trajectory ($T_C = 2000$) as a library of states $\boldsymbol{x}_t$ with known successors $\boldsymbol{x}_{t+1}$. These methods are classically iterated to produce finite-horizon point forecasts. Here we run the same recursion as a long free trajectory, feeding each prediction back as the next query state.

Specifically, we test the *one-NN successor*, which takes the successor of the current state's nearest neighbor as the next state prediction, while the *k-NN successor* and *distance-weighted k-NN* average the successors of the k nearest neighbors (here $k = 3$), either uniformly or weighted by inverse distance [36, 52]. *Locally linear regression* instead fits an affine map $\boldsymbol{x}_{t+1} \approx A\boldsymbol{x}_t + \boldsymbol{b}$ to the $k = 2(N + 1)$ nearest neighbors by least squares each single time step and uses that for next-state prediction [36, 24].

As shown in Table 10, DynaBase achieves the best long-term performance (D_{stsp}, D_H), while locally linear regression yields the lowest short-term error. This is expected, as these classical methods were designed for short-term prediction rather than for reconstructing the long-term dynamics of a DS, and all of them lack parameters such as α in DynaBase which can be adjusted to regulate local con- or divergence rates, and they are thus more restricted in the types of global dynamics they can represent given a specific context. We emphasize, however, that DynaBase was not intended to become a new SOTA forecasting and DSR model, but rather was introduced here to distill and understand the minimal mechanisms necessary for zero-shot DSR in DSR FMs like DynaMix.

Table 10: Comparison of DynaBase to classical nonlinear library-based forecasters across all 54 test DS used in [46], for the same context length $T_C = 2000$ (disregarding diverging runs). Reported are median±MAD for geometrical disagreement D_{stsp}, temporal disagreement D_H, and 10-step-ahead prediction MASE.

Model	$D_{\text{stsp}} \downarrow$	$D_H \downarrow$	MASE $\downarrow$
DynaBase	1.76 ± 0.62	0.06 ± 0.03	0.13 ± 0.08
One-NN successor	5.00 ± 2.53	0.06 ± 0.05	0.19 ± 0.13
k-NN successor	9.73 ± 1.47	0.17 ± 0.12	0.24 ± 0.18
Distance-weighted k-NN	8.51 ± 1.99	0.13 ± 0.10	0.23 ± 0.18
Locally linear regression	2.25 ± 1.65	0.11 ± 0.10	0.01 ± 0.01

B Proofs of Theorems

In the following we use the same notation as introduced in Sec. 3.3, with $\boldsymbol{C} = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{T_C}\} \in \mathbb{R}^{N \times T_C}$ a context sequence, $\tau(\boldsymbol{z}) = \arg\min_s \|\boldsymbol{c}_s - \boldsymbol{z}\|_2$ the nearest-neighbor lookup, and

$$f_{\alpha,\beta}(\boldsymbol{z}) = \alpha\boldsymbol{z} + \beta\boldsymbol{c}_{\tau(\boldsymbol{z})} + \gamma\boldsymbol{c}_{\tau(\boldsymbol{z})+1}, \qquad \gamma = 1 - \alpha - \beta, \tag{38}$$

the two-parameter DynaBase map; f_α denotes its self-consistent reduction with $\beta = -\alpha$, $\gamma = 1$. Ties in τ are broken by the smallest index; the tie set has measure zero and does not affect any of the statements below.

B.1 Proof of Theorem 1 (Uniqueness of form)

Proof. We start from the affine ansatz eq. 7,

$$f_{\text{Linear}}(\boldsymbol{z}, \boldsymbol{C}) = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{B}^{(1)}\boldsymbol{c}_{\tau(\boldsymbol{z})} + \boldsymbol{B}^{(2)}\boldsymbol{c}_{\tau(\boldsymbol{z})+1} + \tilde{\boldsymbol{h}}, \tag{39}$$

with $\boldsymbol{A}, \boldsymbol{B}^{(1)}, \boldsymbol{B}^{(2)} \in \mathbb{R}^{N \times N}$ and $\boldsymbol{h} \in \mathbb{R}^N$, and impose the two structural assumptions in turn.

Orthogonal equivariance. $O(N)$-equivariance requires $Q\, f_{\text{Linear}}(\boldsymbol{z}, \boldsymbol{C}) = f_{\text{Linear}}(Q\boldsymbol{z}, Q\boldsymbol{C})$ for all $Q \in O(N)$, where $Q\boldsymbol{C} := \{Q\boldsymbol{c}_s\}_s$. Since τ is itself $O(N)$-equivariant, substituting eq. 39 and matching coefficients of the independent vectors $\boldsymbol{z}, \boldsymbol{c}_{\tau(\boldsymbol{z})}, \boldsymbol{c}_{\tau(\boldsymbol{z})+1}$ gives

$$Q\boldsymbol{A} = \boldsymbol{A}Q, \quad Q\boldsymbol{B}^{(1)} = \boldsymbol{B}^{(1)}Q, \quad Q\boldsymbol{B}^{(2)} = \boldsymbol{B}^{(2)}Q, \quad Q\boldsymbol{h} = \boldsymbol{h}, \qquad \forall Q \in O(N). \tag{40}$$

The standard representation of $O(N)$ on $\mathbb{R}^N$ is irreducible for $N \geq 2$, so by Schur's lemma every commuting matrix is a scalar multiple of the identity, and the only $O(N)$-fixed vector is $\mathbf{0}$:

$$\boldsymbol{A} = \alpha \boldsymbol{I}, \quad \boldsymbol{B}^{(1)} = \beta \boldsymbol{I}, \quad \boldsymbol{B}^{(2)} = \gamma \boldsymbol{I}, \quad \boldsymbol{h} = \mathbf{0}, \qquad \alpha, \beta, \gamma \in \mathbb{R}. \tag{41}$$

Hence eq. 39 reduces to $f(\boldsymbol{z}) = \alpha \boldsymbol{z} + \beta \boldsymbol{c}_{\tau(\boldsymbol{z})} + \gamma \boldsymbol{c}_{\tau(\boldsymbol{z})+1}$.

Translation equivariance. For any $\boldsymbol{v} \in \mathbb{R}^N$, applying the joint shift $\boldsymbol{z} \mapsto \boldsymbol{z} + \boldsymbol{v}$, $\boldsymbol{C} \mapsto \boldsymbol{C} + \boldsymbol{v}$ leaves τ invariant. Substituting into the previous form,

$$f(\boldsymbol{z} + \boldsymbol{v}; \boldsymbol{C} + \boldsymbol{v}) = \alpha(\boldsymbol{z} + \boldsymbol{v}) + \beta(\boldsymbol{c}_{\tau(\boldsymbol{z})} + \boldsymbol{v}) + \gamma(\boldsymbol{c}_{\tau(\boldsymbol{z})+1} + \boldsymbol{v}) = f(\boldsymbol{z}; \boldsymbol{C}) + (\alpha + \beta + \gamma)\boldsymbol{v}. \tag{42}$$

Translation equivariance demands $f(\boldsymbol{z} + \boldsymbol{v}; \boldsymbol{C} + \boldsymbol{v}) = f(\boldsymbol{z}; \boldsymbol{C}) + \boldsymbol{v}$ for all $\boldsymbol{v}$, hence $\alpha + \beta + \gamma = 1$. Setting $\gamma = 1 - \alpha - \beta$ yields the two-parameter family

$$f_{\alpha,\beta}(\boldsymbol{z}) = \alpha \boldsymbol{z} + \beta \boldsymbol{c}_{\tau(\boldsymbol{z})} + (1 - \alpha - \beta)\, \boldsymbol{c}_{\tau(\boldsymbol{z})+1}. \tag{43}$$

□

B.2 Proof of Theorem 2 (Reduction to context parroting)

Proof. For $\alpha = \beta = 0$, the DynaBase map on the delay-embedded context $\overline{\boldsymbol{C}}_{1:T_C-D+1} \in \mathbb{R}^{D \times (T_C - D + 1)}$ reduces to the deterministic lookup

$$\boldsymbol{z}_{t+1} = f_0(\boldsymbol{z}_t) = \overline{\boldsymbol{c}}_{\tau(\boldsymbol{z}_t)+1}, \qquad \tau(\boldsymbol{z}) = \operatorname*{arg\,min}_{1 \leq s \leq T_C - D} \|\overline{\boldsymbol{c}}_s - \boldsymbol{z}\|_2. \tag{44}$$

Initial step. Starting from $\boldsymbol{z}_0 = \overline{\boldsymbol{c}}_{T_C-D+1}$ (the D most recent context entries), the first application of eq. 44, assuming the final D steps are omitted for $\tau(\boldsymbol{z}_0)$ (see [100]), yields

$$s^\star := \tau(\boldsymbol{z}_0) = \operatorname*{arg\,min}_{1 \leq s \leq T_C - 2D} \|\overline{\boldsymbol{c}}_s - \overline{\boldsymbol{c}}_{T_C-D+1}\|_2. \tag{45}$$

Since $\overline{\boldsymbol{c}}_s = (c_s, \ldots, c_{s+D-1})$, this distance is the Euclidean distance between two D-length motifs of the original context—identical to the motif-matching step of [100, Algorithm 1]. Hence $s^\star$ coincides with the best-matching motif index produced by context parroting.

Induction step. Suppose $\boldsymbol{z}_k = \overline{\boldsymbol{c}}_{s^\star+k}$ with $s^\star + k \leq T_C - D$. Then $\tau(\boldsymbol{z}_k) = s^\star + k$ at zero distance, and

$$\boldsymbol{z}_{k+1} = \overline{\boldsymbol{c}}_{s^\star+k+1} = (c_{s^\star+k+1}, \ldots, c_{s^\star+k+D}). \tag{46}$$

By induction, $\boldsymbol{z}_t = \overline{\boldsymbol{c}}_{s^\star+t}$ for all $0 \leq t \leq T_C - D - s^\star$. Extracting observations from the delay embedded vectors results in

$$x_t = z_{D,t} = c_{s^\star+t+D-1}, \tag{47}$$

which is exactly the forecast emitted by [100, Algorithm 1] after locating the best-matching motif $s^\star$.

Wrap-around. Once $s^\star + t$ exceeds $T_C - D$, both algorithms treat the current tail as a new query and repeat the motif search. Both procedures execute the same operation, so the equivalence extends to arbitrary forecast horizons. □

B.3 Proof of Theorem 3 (Long-term dynamics)

Proof. For $\alpha = 0$, f_α reduces to $f_0(\boldsymbol{z}) = \boldsymbol{c}_{\tau(\boldsymbol{z})+1}$, with finite range $\{\boldsymbol{c}_2, \ldots, \boldsymbol{c}_{T_C}\}$. After one iteration every orbit lies in this finite set, on which f_0 acts as the deterministic graph map $\boldsymbol{c}_s \mapsto \boldsymbol{c}_{\tau(\boldsymbol{c}_s)+1} = \boldsymbol{c}_{s+1}$ for $s < T_C$. Each node has out-degree exactly one, so iteration on a finite directed functional graph must enter a cycle in at most $T_C - 1$ steps and remain periodic thereafter. The differential of f_0 vanishes on the interior of every Voronoi cell (the map is piecewise constant), so the Lyapunov exponent *within each cell* is $\lambda_{\max} = -\infty < 0$ everywhere. Since the saltation operator $\mathcal{S}_{s,s+1} = \mathbf{I}$ between two Voronoi cells V_s and V_{s+1} is always finite and nonzero for $\alpha = 0$, we obtain $\lambda_{\max} = -\infty$ globally. □

B.4 Proof of Theorem 4 (Boundedness)

Proof. For $\boldsymbol{z} \in \mathbb{R}^N$ define the distance to the context points available to the nearest-neighbor lookup by

$$d_{\boldsymbol{C}}(\boldsymbol{z}) := \min_{1 \le s \le T_C - 1} \|\boldsymbol{c}_s - \boldsymbol{z}\|_2 = \left\|\boldsymbol{z} - \boldsymbol{c}_{\tau(\boldsymbol{z})}\right\|_2. \tag{48}$$

Sufficiency ($\alpha \in [0, 1)$, and $\alpha = 1$ for closed contexts). Fix any $\boldsymbol{z}_t$ and let $\boldsymbol{z}_{t+1} = f_\alpha(\boldsymbol{z}_t) = \alpha(\boldsymbol{z}_t - \boldsymbol{c}_{\tau(\boldsymbol{z}_t)}) + \boldsymbol{c}_{\tau(\boldsymbol{z}_t)+1}$ as in eq. 12. If $\tau(\boldsymbol{z}_t) + 1 \le T_C - 1$, then $\boldsymbol{c}_{\tau(\boldsymbol{z}_t)+1}$ is itself a lookup point and

$$d_{\boldsymbol{C}}(\boldsymbol{z}_{t+1}) \le \left\|\boldsymbol{z}_{t+1} - \boldsymbol{c}_{\tau(\boldsymbol{z}_t)+1}\right\|_2 = \alpha \left\|\boldsymbol{z}_t - \boldsymbol{c}_{\tau(\boldsymbol{z}_t)}\right\|_2 = \alpha\, d_{\boldsymbol{C}}(\boldsymbol{z}_t). \tag{49}$$

If $\tau(\boldsymbol{z}_t) + 1 = T_C$, choose $k \le T_C - 1$ with $\|\boldsymbol{c}_{T_C} - \boldsymbol{c}_k\|_2 = d_{\boldsymbol{C}}(\boldsymbol{c}_{T_C})$. By the triangle inequality we obtain

$$d_{\boldsymbol{C}}(\boldsymbol{z}_{t+1}) \le \|\boldsymbol{z}_{t+1} - \boldsymbol{c}_{T_C}\|_2 + \|\boldsymbol{c}_{T_C} - \boldsymbol{c}_k\|_2 = \alpha\, d_{\boldsymbol{C}}(\boldsymbol{z}_t) + d_{\boldsymbol{C}}(\boldsymbol{c}_{T_C}). \tag{50}$$

Hence, in both cases, the following holds:

$$d_{\boldsymbol{C}}(\boldsymbol{z}_{t+1}) \le \alpha\, d_{\boldsymbol{C}}(\boldsymbol{z}_t) + d_{\boldsymbol{C}}(\boldsymbol{c}_{T_C}). \tag{51}$$

For closed contexts, $d_{\boldsymbol{C}}(\boldsymbol{c}_{T_C}) = 0$ and this gives $d_{\boldsymbol{C}}(\boldsymbol{z}_{t+1}) \le \alpha\, d_{\boldsymbol{C}}(\boldsymbol{z}_t) \le d_{\boldsymbol{C}}(\boldsymbol{z}_t)$ for $\alpha \in [0, 1]$, so by induction $d_{\boldsymbol{C}}(\boldsymbol{z}_t) \le d_{\boldsymbol{C}}(\boldsymbol{z}_0) =: R$ for all $t \ge 0$. For general contexts and $\alpha \in [0, 1)$, induction gives $d_{\boldsymbol{C}}(\boldsymbol{z}_t) \le \alpha^t d_{\boldsymbol{C}}(\boldsymbol{z}_0) + (1 - \alpha^t)\frac{d_{\boldsymbol{C}}(\boldsymbol{c}_{T_C})}{1-\alpha} \le \max\left\{d_{\boldsymbol{C}}(\boldsymbol{z}_0), \frac{d_{\boldsymbol{C}}(\boldsymbol{c}_{T_C})}{1-\alpha}\right\} =: R$ for all $t \ge 0$. In both cases,

$$\|\boldsymbol{z}_t\|_2 \le \max_{1 \le s \le T_C - 1} \|\boldsymbol{c}_s\|_2 + R < \infty, \tag{52}$$

and the orbit is bounded. For $\alpha = 1$, the closeness condition cannot be dropped: for $\boldsymbol{C} = (0, 1, 2)$, every cell has increment $c_{s+1} - c_s = 1$, so $f_1(z) = z + 1$ and all orbits diverge. It is, however, not necessary either: for $\boldsymbol{C} = (0, 1, \varepsilon)$ with $0 < \varepsilon \le \frac{1}{2}$, f_1 maps $[\varepsilon - \frac{1}{2}, \frac{3}{2}]$ onto itself and all other orbits enter this interval after finitely many steps, although $d_{\boldsymbol{C}}(\boldsymbol{c}_{T_C}) = \varepsilon > 0$, i.e., boundedness depends on the arrangement of the Voronoi cells.

Non-necessity ($\alpha > 1$ counterexample). Let $N = 1$, $\alpha = 2$, and $\boldsymbol{C} = (c_1, c_2, c_3) = (-1, 1, -1)$. The induced Voronoi cells are $V_1 = \{z \le 0\}$ and $V_2 = \{z > 0\}$, with successors $c_2 = 1$ and $c_3 = -1$, respectively. Substituting into eq. 12 gives the piecewise-affine map

$$f_2(z) = \begin{cases} 2z + 3, & z \le 0, \\ 2z - 3, & z > 0. \end{cases} \tag{53}$$

A direct check shows $f_2([-3, 3]) \subseteq [-3, 3]$: for $z \in [-3, 0]$ we have $2z + 3 \in [-3, 3]$, and for $z \in (0, 3]$ we have $2z - 3 \in (-3, 3]$. Hence any orbit with $z_0 \in [-3, 3]$ remains in $[-3, 3]$ for all $t \ge 0$, proving bounded dynamics for $\alpha = 2 > 1$. This map is topologically conjugate to the doubling/sawtooth map $x_{t+1} = 2x_t \bmod 1$, whose Lyapunov exponent equals $\log 2$ on its invariant set, and hence by virtue of this conjugacy inherits the chaotic dynamics of the sawtooth map. □

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: All results referenced in the abstract and introduction are fully reported and supported in later sections of the paper.

   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: Limitations are addressed in the conclusion or directly in the respective sections.

   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 paper includes formal proofs of all theorems in Appx. B.

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 model architecture, along with the training and evaluation procedures, is described in the main text, with further details and hyper-parameter settings in Appx. A.1-A.4.

   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: A link to a github repository is included, which provides the implementation of the network architecture used to obtain the main results reported in the paper (https://github.com/DurstewitzLab/DynaBase).

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: All settings are provided in the main text or in sections A.1-A.4.

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: Errors are reported for numerical results, as well as statistical significance tests where needed.

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: Details on computational resources are provided in Sect. A.1.

   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: All datasets used are publicly available. This work targets improved interpretability of machine learning models, which we believe carries primarily positive ethical implications. While no specific concerns have been identified, the breadth of potential applications means misuse cannot be entirely ruled out.

   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: As basic research, we anticipate no direct negative 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: We do not foresee a high risk for misuse.

    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: [Yes]

    Justification: All novel code, models, and results were created by us. Baseline models, datasets and evaluations are publicly available and properly credited.

    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: [Yes]

    Justification: A documented version of the code is provided.

    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: The paper does not involve crowdsourcing or research with human subjects.

    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: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - 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: The methods developed in this paper did 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.