A Comparative Empirical Study of Relative Embedding Alignment in Neural Dynamical System Forecasters

Deniz Kucukahmetler KUCUKAHM@CBS.MPG.DE
Max Planck Institute for Human Cognitive and Brain Sciences, Leipzig, Germany
SECAI: School of Embedded Composite Artificial Intelligence, Dresden/Leipzig, Germany

Maximilian Jean Hemmann
Leipzig University, Leipzig, Germany

Julian Mosig von Aehrenfeld[†]
Leipzig University, Leipzig, Germany

Maximilian Amthor[†]
Leipzig University, Leipzig, Germany

Christian Deubel[†]
Leipzig University, Leipzig, Germany

Nico Scherf
Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI), Dresden/Leipzig, Germany

Diaaeldin Taha
Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany

[†] **Equal contribution.**

Editors: List of editors' names

Abstract

We study representation alignment in neural forecasters using anchor-based, geometry-agnostic *relative embeddings* that remove rotational and scaling ambiguities, enabling robust cross-seed and cross-architecture comparisons. Across diverse periodic, quasi-periodic, and chaotic systems and a range of forecasters (MLPs, RNNs, transformers, Neural ODE/ Koopman, ESNs), we find consistent family-level patterns: MLPs align with MLPs, RNNs align strongly, transformers align least with others, and ESNs show reduced alignment on several chaotic systems. Alignment generally tracks forecasting accuracy—higher similarity predicts lower multi-step MSE—yet strong performance can occur with weaker alignment (notably for transformers). Relative embeddings thus provide a practical, reproducible basis for comparing learned dynamics.

Keywords: dynamical systems, relative representations, latent representations, forecasting

© 2025 D. Kucukahmetler, M.J. Hemmann, J.M.v. Aehrenfeld[†], M. Amthor[†], C. Deubel[†], N. Scherf & D. Taha.

1. Introduction

Neural forecasters are widely used for modeling time-evolving processes, making it essential to understand how they represent dynamics internally and whether those representations align with human goals. Dynamical systems theory—from Poincaré to modern hyperbolic dynamics—provides the basis for this study [1; 22]. Canonical benchmarks such as the Lorenz-63 attractor [14], logistic map [17], Hopf oscillators, the double pendulum, and reduced-order cylinder wakes [3] span periodic, quasi-periodic, and chaotic regimes. Models ranging from reservoir computers [20] and recurrent neural networks (RNNs) [26; 10] to transformers [25], latent ODEs [5], and Koopman autoencoders [15] are routinely evaluated on these systems, situating our work within the data-driven forecasting tradition rooted in nonlinear time-series analysis [24]. Yet latent spaces are unstable across seeds and architectures (rotations, scalings, geometry shifts), complicating cross-model comparisons (Appendix Figure 3).

Comparing learned dynamics therefore requires robust alignment tools. RSA [12], Procrustes [9], and CKA [11] are influential but can be geometry-dependent, brittle across runs, or impose restrictive mapping assumptions. Structured alternatives include topological conjugacy [2], anchor-based relative representations [18], atlas-style latent-space merging [6], stitching across modalities and policies [19; 21], and topology/spectral refinements [8; 7], alongside landmark-based alignments [16] and product-space decompositions [4]. We adopt relative, anchor-based, geometry-agnostic embeddings that remove rotational/scaling freedoms and yield reproducible alignment across seeds, architectures, and systems (Figure 3). This quantifies "representational families," reveals systematic patterns across multi-layer perceptrons (MLPs), RNNs, transformers, and echo-state networks (ESNs), and provides a practical signal correlated with forecasting accuracy—including cases where high accuracy coexists with low alignment.

2. Method

Representational alignment framework Following Sucholutsky et al. [23], a *representational alignment experiment* specifies *data*, *systems*, *measurements*, *embeddings*, and a *similarity metric*.

Data We generate multistep trajectories from seven canonical systems spanning periodic, quasi-periodic, and chaotic dynamics in continuous or discrete time: Lorenz–63 (3D chaotic ODE), stable limit cycle (2D), double pendulum (4D Hamiltonian chaos), Hopf normal form (2D), logistic map (1D), a fluid cylinder-wake dataset using the top three POD coefficients [3], and a weakly coupled 6D skew-product built from chaotic founders (Lorenz–63/Rössler/Chen) with parameter jitter and unidirectional coupling (see [13]). Each system provides independent train/val/test trajectories of length T (z-scored per channel using train statistics).

Systems: encoder–decoder forecasters Given an input window $\mathbf{x}_{t-L+1:t} \in \mathbb{R}^{L\times d}$, the model predicts the next H states $\widehat{\mathbf{x}}_{t+1:t+H} \in \mathbb{R}^{H\times d}$ via $\widehat{\mathbf{x}}_{t+1:t+H} = \psi_{\theta_d}\big(\mathcal{P}_\Theta\big(\phi_{\theta_e}(\mathbf{x}_{t-L+1:t})\big)\big)$, with encoder $\phi_{\theta_e} : \mathbb{R}^{L\times d} \to \mathbb{R}^k$, latent propagator $\mathcal{P}_\Theta : \mathbb{R}^k \to \mathbb{R}^k$, and decoder $\psi_{\theta_d} : \mathbb{R}^k \to \mathbb{R}^{H\times d}$. We instantiate $\mathcal{P}_\Theta$ as: (a) identity (one-shot MLP); (b) RNN propagators, including (i) a standard latent gated recurrent unit (GRU) update $\mathbf{z}_{k+1} = \mathrm{GRU}_\Theta(\mathbf{z}_k)$ and (ii) an autoregressive GRU forecaster where the hidden state is updated based on both the previous state and the model's own decoded output; (c) causal transformer; (d) Neural ODE integrated for H steps; (e) linear Koopman update $\mathbf{z}_{k+1} = K\mathbf{z}_k$ for H steps. As a reservoir baseline, we use an echo-state network with fixed sparse reservoir and ridge-regression readout (no BPTT).

Measurements: latent representations Training a given architecture with different seeds or swapping architectures yields a family of encoders $\{\phi^{(s)}_{\theta_e^{(s)}}\}_{s=1}^{S}$ whose latent spaces need not align. For each input window, we take $\mathbf{z} = \phi_{\theta_e}(\mathbf{x}_{t-L+1:t}) \in \mathbb{R}^k$ as the measurement.

Embeddings: anchor-based relative embeddings Each encoder produces latent vectors $\mathbf{z}_j = \phi_{\theta_e}(\mathbf{x}_{t_j-L+1:t_j}) \in \mathbb{R}^k$, which are first z-scored feature-wise across the dataset. A fixed subset $\mathcal{A} = \{\mathbf{a}_i\}_{i=1}^{m} \subset \{\mathbf{z}_j\}_{j=1}^{N}$ serves as anchors, and each normalized latent yields a *relative embedding*

$$\mathbf{r}_{\text{rel}}(\mathbf{z}) = \big(\text{sim}(\mathbf{z}, \mathbf{a}_1), \ldots, \text{sim}(\mathbf{z}, \mathbf{a}_m)\big),$$

where $\text{sim}(\cdot, \cdot)$ denotes a similarity function introduced in the next subsection. This produces, for each forecaster, a matrix $\mathbf{R}_{\text{rel}} \in \mathbb{R}^{N \times m}$ whose rows correspond to datapoints and columns to anchors. We fix the number of anchors $K = 80$ to balance variance (see Appendix C for details).

Similarity metrics between two models We quantify the similarity between two encoders $\phi^{(1)}_{\theta_e^{(1)}}$ and $\phi^{(2)}_{\theta_e^{(2)}}$ over a dataset $\mathcal{V}$ using *cosine similarity*. Between the encoders' relative embeddings $\mathbf{r}^{(1)}_{\text{rel}}$ and $\mathbf{r}^{(2)}_{\text{rel}}$, the alignment score is

$$\alpha_{\cos}\Big(\phi^{(1)}_{\theta_e^{(1)}}, \phi^{(2)}_{\theta_e^{(2)}}; \mathcal{V}\Big) = \frac{1}{|\mathcal{V}|} \sum_{\mathbf{z} \in \mathcal{V}} \frac{\big\langle \mathbf{r}^{(1)}_{\text{rel}}(\mathbf{z}),\, \mathbf{r}^{(2)}_{\text{rel}}(\mathbf{z}) \big\rangle}{\|\mathbf{r}^{(1)}_{\text{rel}}(\mathbf{z})\|_2 \, \|\mathbf{r}^{(2)}_{\text{rel}}(\mathbf{z})\|_2}.$$

This measure captures how consistently the two encoders place samples in relative position to a shared set of anchors.

3. Results

Relative representations provide a common basis across architectures. Figure 1 illustrates that anchor-based *relative* embeddings reduce geometric arbitrariness (rotations, scalings) in latent spaces, making cross-architecture comparisons more interpretable. With colors indicating distinct model labels, the relative space clarifies similarities and differences across models in a common coordinate system.

Model–model alignment structure. Cross-model similarity in Figure 1 (pairwise alignment heatmaps; cosine similarity of relative embeddings) reveals consistent family structure across systems: (i) in all systems, the *MLP family* (plain MLP, Koopman–MLP, Neural-ODE–MLP) forms a cluster; (ii) the *RNN family* (GRU, autoregressive GRU, Koopman–GRU, Neural-ODE–GRU) is well-aligned in all systems *except* the Logistic Map, where alignment weakens; (iii) the ESN baseline exhibits noticeably lower alignment in Lorenz, Double Pendulum, and the random skew-product; (iv) the *transformer family* tends to align less with other families—most prominently in Double Pendulum and Lorenz—suggesting a different inductive bias in how context is summarized for forecasting. Overall, these patterns indicate that architectural choices induce reproducible representational geometries within families, while some dynamics (e.g., Logistic Map) challenge specific families (RNNs).

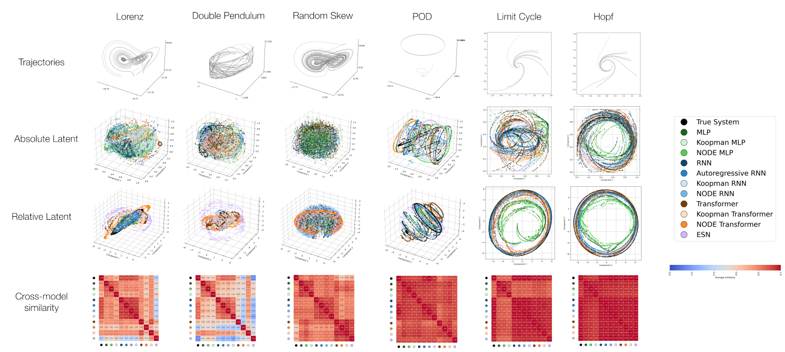

Figure 1: **Trajectories, embeddings, and cross-model alignment.** Columns show six systems: Lorenz, double pendulum, random skew, POD (cylinder wake), limit cycle, and Hopf. Rows: (top) training trajectories in state space; each gray shade denotes a distinct input trajectory; (second) absolute latent embeddings from each forecaster; (third) relative latent embeddings after anchor-based standardization (visualized with PCA; we plot the first 2 or 3 components, depending on the system); (bottom) cosine-similarity heatmaps between relative embeddings for all forecaster pairs, averaged over five seeds. Relative embeddings reduce geometric variability across models, making alignment directly comparable across systems.

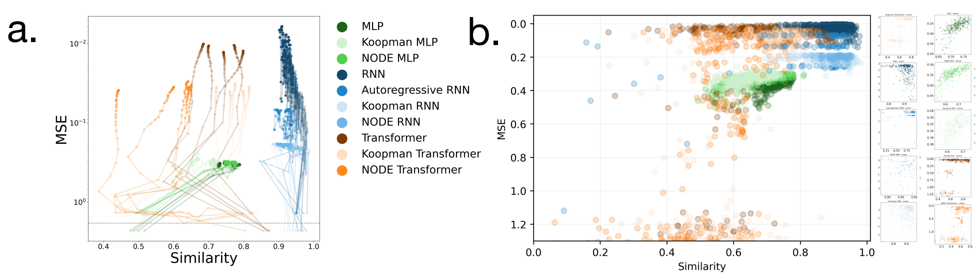

Figure 2: **Alignment correlates with forecasting performance.** (a) Test MSE versus representational similarity with the Lorenz system across all tuned models during training. Lines of the same color correspond to different seeds of the same model; opacity increases as training progresses. Models with higher alignment generally achieve lower error. A random baseline error of 1.25 (untrained model predictions averaged over 20 seeds) is marked for reference. (b) Each point represents a trained model used in hyperparameter tuning. Main panel: validation loss versus representational similarity aggregated across all individual models shown in the right inset. A consistent positive correlation indicates that relative-embedding alignment provides a useful proxy for forecasting quality.

Performance versus alignment. Figure 2a relates test performance to alignment with the true system on Lorenz, a chaotic and widely used benchmark; results for the other systems are in the appendix. We observe family-specific training trajectories. *RNNs* begin with comparatively high alignment and remain stable through training, while their test error decreases steadily. *MLPs* start with lower alignment that increases as training proceeds, tracking improvements in error; this manifests as transparent (early) points moving towards higher similarity and lower MSE. *Transformers* display lower and more variable alignment across seeds (including Koopman- and ODE-augmented variants), yet often achieve competitive or superior forecasting error—frequently surpassing the MLP family and often rivaling GRU variants. This underscores that high alignment is *helpful but not strictly necessary* for strong forecasting: transformers can realize good accuracy with a representational geometry that aligns less to the ground-truth relative space.

To probe robustness beyond training trajectories, we aggregate models generated during hyperparameter tuning (each point is a trained model used in the tuning process) in Figure 2b. Within several architectures we observe a positive association between representational similarity and forecasting accuracy, though the strength of this association is family- and system-dependent.

4. Discussion and conclusion

Anchor-based *relative* embeddings offer a geometry-agnostic, stable basis for comparing neural forecasters across seven canonical systems and diverse architectures. Anchor-based relative embeddings provide a geometry-agnostic and stable basis for comparing neural forecasters across architectures and dynamical regimes. Alignment typically correlates with forecasting accuracy yet admits family-specific exceptions (notably transformers), making it a complementary audit signal to validation error. Future work should probe noise, partial observability, and targeted interpretability to explain family-level differences.

References

[1] Vladimir I Arnold. *Mathematical methods of classical mechanics*, volume 60. Springer-Verlag, 1989.

[2] Arthur Bizzi, Lucas Nissenbaum, and João M Pereira. Neural conjugate flows: A physics-informed architecture with flow structure. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025. To appear.

[3] 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*, 113(15):3932–3937, 2016.

[4] Irene Cannistraci, Luca Moschella, Marco Fumero, Valentino Maiorca, and Emanuele Rodolà. From bricks to bridges: Product of invariances to enhance latent space communication. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=vngVydDWft.

[5] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[6] Donato Crisostomi, Irene Cannistraci, Luca Moschella, Pietro Barbiero, Marco Ciccone, Pietro Liò, and Emanuele Rodolà. From charts to atlas: Merging latent spaces into one. *arXiv preprint arXiv:2311.06547*, 2023.

[7] Marco Fumero, Marco Pegoraro, Valentino Maiorca, Francesco Locatello, and Emanuele Rodolà. Latent functional maps: a spectral framework for representation alignment, 2025. URL https://arxiv.org/abs/2406.14183.

[8] Alejandro García-Castellanos, Giovanni Luca Marchetti, Danica Kragic, and Martina Scolamiero. Relative representations: Topological and geometric perspectives. In *UniReps: 2nd Edition of the Workshop on Unifying Representations in Neural Models*, 2024. URL https://openreview.net/forum?id=RDfkKNoET5.

[9] John C Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[11] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019.

[12] Nikolaus Kriegeskorte, Marieke Mur, and Peter A Bandettini. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:4, 2008.

[13] Jeffrey Lai, Anthony Bao, and William Gilpin. Panda: A pretrained forecast model for universal representation of chaotic dynamics, 2025. URL https://arxiv.org/abs/2505.13755.

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

[15] Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1):4950, 2018.

[16] Valentino Maiorca, Luca Moschella, Antonio Norelli, Marco Fumero, Francesco Locatello, and Emanuele Rodolà. Latent space translation via semantic alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[17] Robert M May. Simple mathematical models with very complicated dynamics. *Nature*, 261(5560):459–467, 1976.

[18] Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. Relative representations enable zero-shot latent space communication. In *The Eleventh International Conference on Learning Representations*, 2023.

[19] Antonio Norelli, Marco Fumero, Valentino Maiorca, Luca Moschella, Emanuele Rodola, and Francesco Locatello. Asif: Coupled data turns unimodal models to multimodal without training. *Advances in Neural Information Processing Systems*, 36:15303–15319, 2023.

[20] 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), December 2017. ISSN 1089-7682. doi: 10.1063/1.5010300. URL http://dx.doi.org/10.1063/1.5010300.

[21] Antonio Pio Ricciardi, Valentino Maiorca, Luca Moschella, Riccardo Marin, and Emanuele Rodolà. R3l: Relative representations for reinforcement learning. *arXiv preprint arXiv:2404.12917*, 2024.

[22] Steven H Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering.* CRC press, 2018.

[23] Ilia Sucholutsky, Lukas Muttenthaler, Adrian Weller, Andi Peng, Andreea Bobu, Been Kim, Bradley C Love, Erin Grant, Iris Groen, Jascha Achterberg, et al. Getting aligned on representational alignment. *arXiv preprint arXiv:2310.13018*, 2023.

[24] Floris Takens. Detecting strange attractors in turbulence. In *Dynamical systems and turbulence, Warwick 1980*, pages 366–381. Springer, 1981.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[26] Pantelis R Vlachas, Jaideep Pathak, Brian R Hunt, Themistoklis P Sapsis, Michelle Girvan, Edward Ott, and Petros Koumoutsakos. Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. *Neural Networks*, 126:191–217, 2020.

Appendix A. Experimental setup

Dynamical systems. Seven systems as above; splits are disjoint in initial conditions, and channels are z-scored using train statistics.

Models and training. We evaluate encoder–decoder forecasters of the form:
(1) MLP–MLP, (2) GRU–GRU, (3) autoregressive GRU–autoregressive GRU, (4) Transformer–Transformer. Architectures (1), (3), and (4) are additionally tested with latent propagation via Neural ODEs or Koopman operators. As a non-gradient baseline, we include an echo-state network (ESN) with fixed sparse reservoir and ridge-regression readout. We optimize with Adam and early stopping on validation MSE (patience 20); model widths, dropout, and k are given in the appendix.

Evaluation. Forecast accuracy is reported as MSE averaged over the H-step horizon ($H = 50$). Representational alignment is measured with $\alpha_{\cos}$ on a held-out set of windows using $K = 80$ shared anchors.

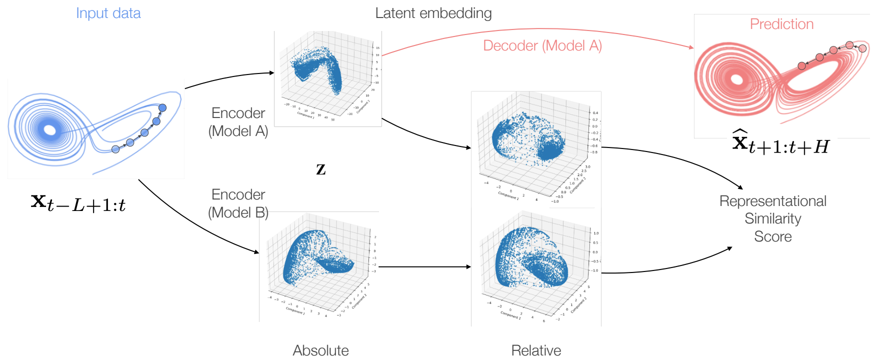

Figure 3: **Overview of forecasting and representational alignment.** Encoder–propagator–decoder forecasters take an input window of L past states $\mathbf{x}_{t-L+1:t}$, embed it into a latent vector $\mathbf{z}$, and decode a prediction of the next H states $\widehat{\mathbf{x}}_{t+1:t+H}$. To compare different models, we compute absolute latent embeddings from data, transform them into anchor-based relative embeddings, and quantify alignment between models (e.g. Model A vs. Model B) using representational similarity scores.

Appendix B. Dynamical systems

We assess our models on seven representative systems. Unless noted otherwise, each system provides 20 trajectories for training, 20 for validation and 20 for testing, with T=500 time steps per trajectory. All channels are z-scored using statistics from the training split; no external noise is added.

Lorenz–63 (3-D chaotic ODE). $\dot{x} = \sigma(y - x),\ \dot{y} = x(\rho - z) - y,\ \dot{z} = xy - \beta z$, with $\sigma = 10,\ \rho = 28,\ \beta = 8/3$. Initial states are sampled from $[-20, 20]^3$ and integrated with Dormand–Prince (RK45) at $\Delta t = 0.01$. Its compact phase space and positive Lyapunov exponent (≈ 0.91) make it a classical multi-step-forecast benchmark.

Stable limit cycle (2-D radial–spiral ODE). $\dot{r} = \mu(R - r),\ \dot{\theta} = \omega,\ (x, y) = (r\cos\theta,\ r\sin\theta)$, with $\mu = 1,\ R = 1,\ \omega = 1$. Trajectories start from $r_0 \sim \mathcal{U}[0, 20]$ and $\theta_0 \sim \mathcal{U}[0, 2\pi]$; integration uses RK45 with $\Delta t = 0.01$.

Double pendulum (4-D Hamiltonian chaos). Two unit-mass, unit-length links move under gravity $g = 9.81$. Angles are initialised in $[-20^\circ, 20^\circ]$ and angular velocities in $[-1, 1]$. Dynamics are solved with RK45 at $\Delta t = 0.01$. Energy conservation and a Lyapunov exponent of ≈ 1.5 test a model's ability to capture chaotic yet nearly conservative motion.

Hopf normal form (2-D near-critical oscillation). $\dot{x} = \mu x - \omega y - (x^2 + y^2)x,\ \dot{y} = \omega x + \mu y - (x^2 + y^2)y$, with $\mu = 0,\ \omega = 1$. Starting points $(x_0, y_0) \sim \mathcal{U}[-2, 2]^2$ spiral onto a unit-radius limit cycle; $\Delta t = 0.01$ with RK45.

Logistic map (1-D near-onset discrete chaos). $x_{t+1} = 3.57\, x_t(1 - x_t)$ with $x_0 \sim \mathcal{U}(0, 1)$; sequences of length T=500 are recorded at an effective step $\Delta t = 0.1$.

Fluid wake behind a cylinder (POD coefficients; $d = 3$). We adopt the three leading Proper-Orthogonal-Decomposition coefficients from [3] (Re = 100, Strouhal ≈ 0.16).

We supply 10 trajectories per split, each of T=500 snapshots sampled at $\Delta t = 0.2$; only z-score normalisation is applied.

Skew-product of 3-D chaotic founders (6-D weakly coupled ODE). Following [13], select two founders from {Lorenz–63, Rössler, Chen}, jitter parameters by multiplicative log-normal noise ($\log s \sim \mathcal{N}(0, 0.15^2)$, sign preserved), and couple them in a skew-product: the first 3-D system $x \in \mathbb{R}^3$ drives the second $y \in \mathbb{R}^3$ via a weak injection into the first response coordinate. Writing $\dot{x} = f_a(x; p_a)$ and $\dot{y} = f_b(y; p_b)$ for the founders with jittered parameters,

$$\dot{x} = f_a(x; p_a), \qquad \dot{y} = f_b(y; p_b) + \varepsilon\, e_1\, x_1, \quad \varepsilon = 0.05,\ e_1 = (1, 0, 0)^\top.$$

Founder templates and nominal seeds:

$$\begin{aligned} \text{Lorenz–63: } & \dot{x} = \sigma(y - x), \quad \dot{y} = x(\rho - z) - y, \quad \dot{z} = xy - \beta z, \\ & (\sigma, \rho, \beta) = (10, 28, \tfrac{8}{3}), \quad x_0 = (1, 1, 1) \end{aligned}$$

$$\begin{aligned} \text{Rössler: } & \dot{x} = -y - z,\ \dot{y} = x + ay, \\ & \dot{z} = b + z(x - c);\ (a, b, c) = (0.2, 0.2, 5.7),\ x_0 = (0.1, 0, 0), \end{aligned}$$

$$\begin{aligned} \text{Chen: } & \dot{x} = a(y - x), \quad \dot{y} = (c - a)x - xz + cy, \quad \dot{z} = xy - bz, \\ & (a, b, c) = (35, 3, 28), \quad x_0 = (-10, 0, 37). \end{aligned}$$

A single skew system is sampled once per dataset; train/val/test splits then differ only by initial conditions. Initial states jitter the concatenated founder seeds $z_0 = [x_0; y_0]$ with i.i.d. Gaussian noise of scale 0.1. Trajectories are integrated with DOP853 at the dataset step Δt (absolute tolerance 10^{-8}, relative 10^{-6}). We discard an initial warm-up fraction (default 10%) and keep the next T steps. Runs are rejected if any state is non-finite, the radius exceeds 10^6, or the summed channel variance falls below 10^{-6}; on rejection we resample once.

Appendix C. Anchor ablations

Computation. We compute the relative embedding

$$r_{\text{rel}}(x) = \big(r_1(x), \ldots, r_m(x)\big), \quad r_i(x) = \frac{\text{sim}\big(\phi(x), \phi(a_i)\big) - \mu_i}{\sigma_i},$$

where a_i is the i-th anchor, μ_i and σ_i are the mean and standard deviation of $\text{sim}\big(\phi(\cdot), \phi(a_i)\big)$ over V, and sim is the similarity used in the main text.

Choice of K anchors. We estimate alignment as a function of the number of anchors K. For each $K \in \{1, 2, 3, 4, 5, 6, 8, 16, 32, 64, 80, 128, 512, 800, 999\}$ we repeat the procedure 30 times with fresh random anchor draws. Estimates stabilize for $K \geq 16$; we set $K = 80$ to balance variance and compute time 4.

Random baseline (disjoint anchors). As a control, we re-estimate alignment using disjoint anchor sets across the two spaces. This collapses alignment to near zero, confirming the necessity of shared anchors [18].

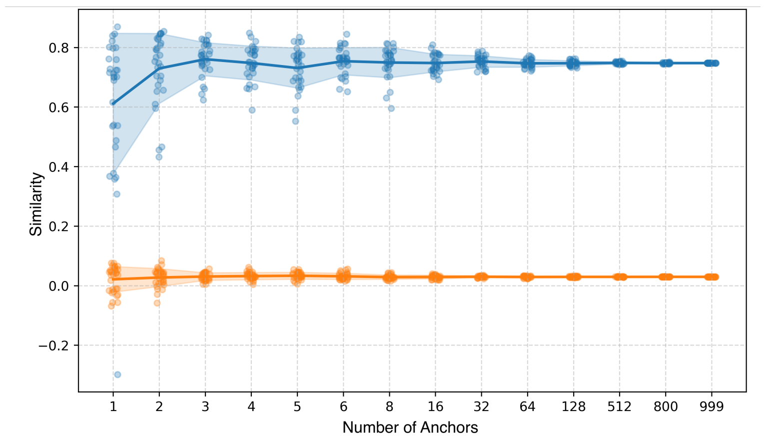

Figure 4: **Anchor ablation and baseline.** (Blue) Alignment vs. number of anchors K; lines show mean over 30 repeats. Stabilization occurs for $K \geq 16$; we choose $K = 80$ (vertical marker) for the main experiments. (Orange) Random baseline with disjoint anchor sets across spaces, yielding near-zero alignment.

Appendix D. Alignment in different parameter settings

See Figure 6.

Appendix E. Alignment during training of the tuned models

See Figure 5.

Appendix F. Model performances

See Figure 7

Appendix G. Cross-model similarity for logistic map

Additional results complementing Figure 1 are shown in Figure 8.

Appendix H. Compute resources

All experiments were conducted on RAVEN HPC system, equipped with Intel Xeon IceLake-SP processors and NVIDIA A100 GPU nodes interconnected via NVLink.

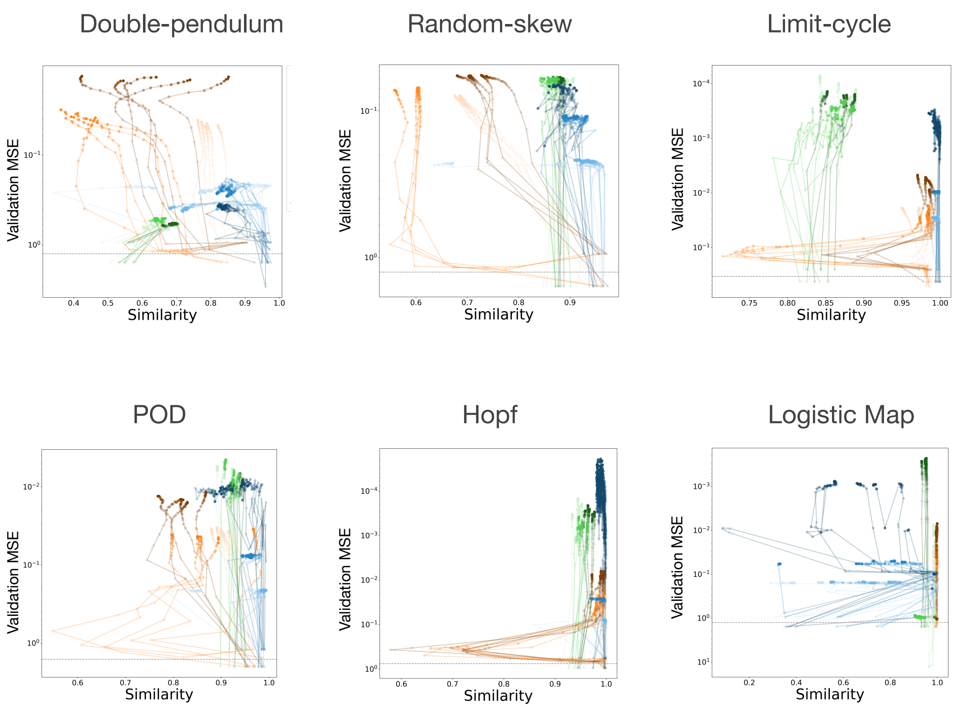

Figure 5: **MSE versus representational similarity.**

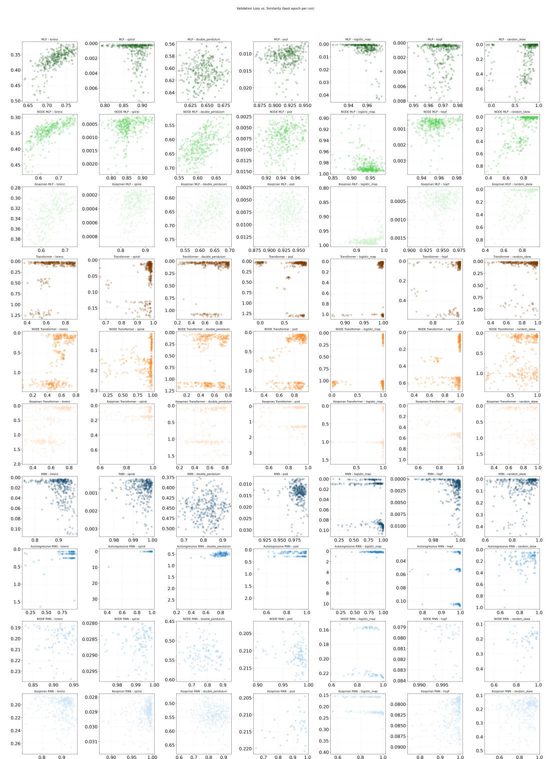

Figure 6: **Individual model alignment scores versus similarity.** Each point represents a distinct model configuration evaluated during training.

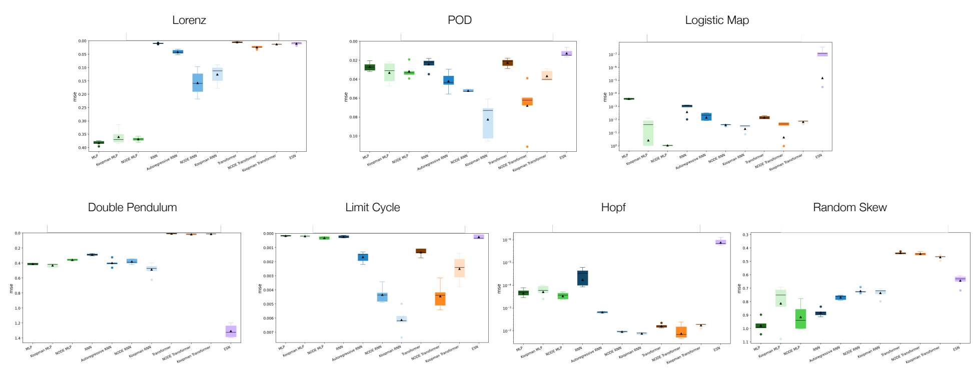

Figure 7: **Tuned model performance by dataset.** For each dataset, box–and–whisker plots compare models: boxes span the interquartile range (25th–75th percentiles), whiskers extend to $1.5 \times \mathrm{IQR}$, and points beyond are outliers. The orange line marks the median, and the green marker denotes the mean.

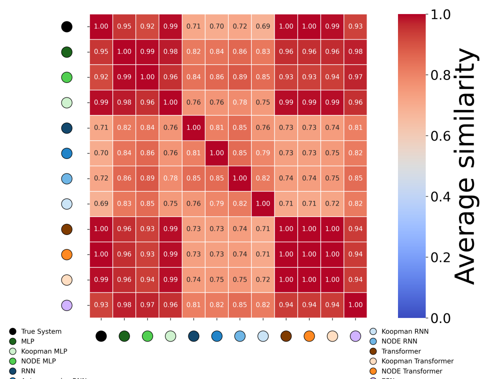

Figure 8: **Cross-Model Similarity of Logistic Map.**