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

Deniz Kucukahmetler[*][†][‡] Maximilian Jean Hemmann[§] Julian Mosig von Aehrenfeld[§][¶]
Maximilian Amthor[§][¶] Christian Deubel[§][¶] Nico Scherf[*][‖] Diaaeldin Taha[**]

Abstract

We study neural forecasters for dynamical systems through the lens of representational alignment. We introduce anchor-based, geometry-agnostic *relative embeddings* that remove rotational and scaling ambiguities, enabling robust cross-seed and cross-architecture comparison. Across diverse periodic, quasi-periodic, and chaotic systems, we observe consistent family-level patterns: MLPs align with MLPs, RNNs with RNNs, and ESNs show reduced alignment on chaotic dynamics, while Transformers often align weakly but still perform well. Alignment generally correlates with forecasting accuracy, yet high accuracy can coexist with low alignment. Relative embeddings thus offer a simple, reproducible basis for comparing learned dynamics.

1 Introduction

Neural forecasters are widely used to model time-evolving processes, yet their internal representations are often unstable across seeds and architectures. Rotations, scalings, and other geometric distortions obscure whether different models capture equivalent dynamical invariants, complicating comparisons across systems. Robust alignment tools are therefore needed.

Classical approaches such as RSA [11], CKA [10], or Procrustes [8] provide useful baselines but remain geometry-dependent and often brittle across runs. Alternatives span conjugacy [1], relative/anchor methods [17], latent-space merging [5], stitching [18, 20], spectral/topological refinements [6, 7], landmark alignment [15], and product-space decompositions [3].

We adopt anchor-based, geometry-agnostic relative embeddings [17] that remove rotational and scaling freedoms and provide reproducible alignment across seeds, architectures, and systems. Using seven canonical periodic, quasi-periodic, and chaotic benchmarks [2, 13, 16], we compare a diverse set of forecasters— MLPs, recurrent neural networks (RNNs) [9, 23], transformers[22], Neural ODEs [4]/Koopman models [14], and echo-state networks (ESNs) [19](Figure 1). Our results reveal reproducible family-level structure in representational geometry and show that alignment generally tracks forecasting accuracy, though strong accuracy can also emerge with weaker alignment (notably for Transformers). Although our focus is on representation alignment, this connects directly to the goals of differentiable systems: building interpretable and reproducible models of dynamics, where understanding latent representations is as important as forecasting accuracy.

[*] Corresponding author: `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.

[§] Leipzig University, Leipzig, Germany.

[¶] Equal contribution.

[‖] Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI), Dresden/Leipzig, Germany.

[**] Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany.

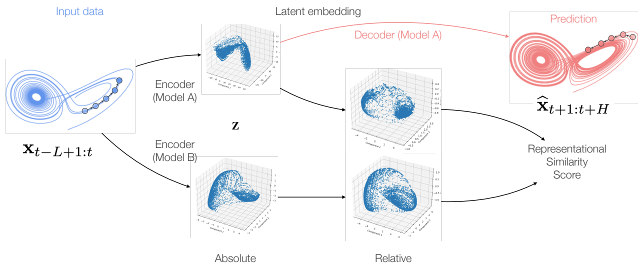

Figure 1: **Overview of forecasting and representational alignment.** An encoder–propagator–decoder maps past states $\mathbf{x}_{t-L+1:t}$ to latent $\mathbf{z}$ and predicts future states $\widehat{\mathbf{x}}_{t+1:t+H}$. We compare models by transforming absolute latents into anchor-based relative embeddings and computing similarity scores between them.

2 Method

Representational alignment framework Following Sucholutsky et al. [21], a *representational alignment experiment* specifies *data*, *systems*, *measurements*, *embeddings*, and a *similarity metric*. In our study: (i) **Data**: simulated trajectories from seven dynamical systems; (ii) **Systems (models)**: encoder–decoder forecasters trained under different seeds/architectures; (iii) **Measurement**: encoder latents $\mathbf{z} = \phi_{\theta_e}(\mathbf{x}_{t-L+1:t}) \in \mathbb{R}^k$ from input windows of shape $\mathbb{R}^{L\times d}$; (iv) **Embeddings**: anchor-based *relative* embeddings built from z-scored anchor similarities; (v) **Similarity**: mean cosine similarity between (1) two encoders' and (2) mean cosine similarity between encoder vs true system.

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 [2], and a weakly coupled 6D skew-product built from chaotic founders (Lorenz–63/Rössler/Chen) with parameter jitter and unidirectional coupling (see [12]). Each system provides independent train/val/test trajectories of length T (z-scored per channel using train statistics).

Models: 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; (b) Neural ODE integrated for H steps; (c) 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 back propagation through time; no-BPTT). A summary of the corresponding propagators is given in Table 1.

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 autoencoders 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

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

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

3 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 using Adam and apply early stopping on the validation MSE (patience of 5 after 20 epochs).

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 (Appendix C).

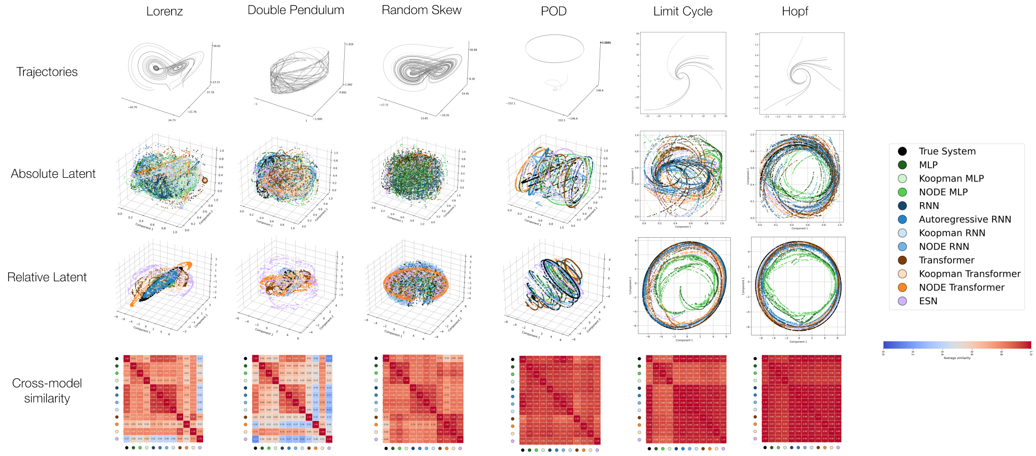

Figure 2: **Trajectories, embeddings, and cross-model alignment.** Six systems (Lorenz, double pendulum, skew-product; trajectories can differ markedly across seeds due to sensitivity to initial conditions; POD wake, limit cycle, Hopf). Rows show trajectories, absolute embeddings, relative embeddings (PCA), and cross-model similarity heatmaps (avg. over five seeds). Relative embeddings reduce geometric variability, enabling direct comparison across forecasters.

4 Results

Relative representations provide a common basis across architectures. Figure 2 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 2 (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).

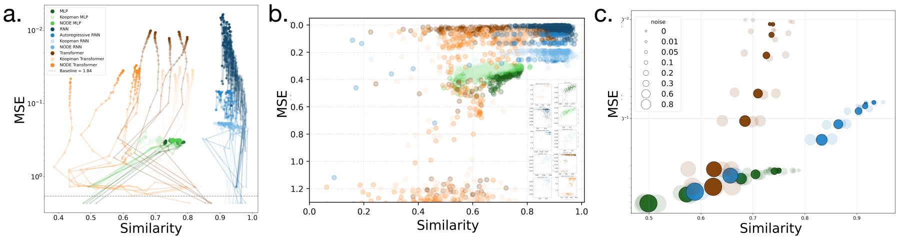

Figure 3: **Alignment correlates with forecasting performance.** (a) On Lorenz, higher similarity generally predicts lower error across seeds. (b) Tuning runs show a consistent similarity–accuracy relationship across models. (c) Added Gaussian noise increases MSE and decreases alignment (bubble size = noise level).

Performance versus alignment. Figure 3a 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 (see Appendix D). 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 Appendix D. 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.

Stitching models using relative representations. In Table 2 we compare stitching models trained on relative representations versus models trained on absolute spaces, across architectures. We find that within model family (MLP and Transformer) relative representations offer a great advantage over absolute representations. Across family stitching (e.g., MLP Encoder stitched to Transformer Decoder), however, benefits less due to architectural incompatibilities. Notice that we exclude the RNN family. That is because RNNs rely heavily on hidden states, which are neither given by MLPs nor by Transformers.

5 Discussion and conclusion

Anchor-based *relative* embeddings offer a geometry-agnostic, stable basis for comparing neural forecasters across seven canonical systems and diverse architectures. Alignment to a system-specific relative space generally correlates with lower multi-step MSE, yet strong performance can arise with weaker alignment (notably for Transformers). Practically, shared-anchor relative spaces can provide a lightweight audit signal across seeds/architectures and complement validation error for model selection and training diagnostics. Limitations include the lack of deeper, model-specific analyses (e.g., ablations or interpretability probes). Each model–system pair warrants targeted follow-up to characterize its representation-learning behavior, especially under added noise, perturbations, and partial observability. These results contribute to the broader goals of differentiable systems by providing tools to analyze latent flows in a geometry-agnostic way. Relative embeddings can thus complement classical system identification, offering a path to more interpretable and reproducible neural dynamical models. For the Differentiable Systems community, our framework highlights how representation-level diagnostics can act as lightweight probes of learned dynamics, bridging machine learning forecasters with system-theoretic analysis.

References

[1] 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.

[2] 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.

[3] 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`.

[4] 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.

[5] 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.

[6] 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`.

[7] 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`.

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

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

[10] 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.

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

[12] 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`.

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

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

[15] 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.

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

[17] 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.

[18] 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.

[19] 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`.

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

[21] 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.

[22] 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.

[23] 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.

A Dynamical systems

We assess our models on seven representative systems. Unless noted otherwise, each system provides 10 trajectories for training, 10 for validation and 10 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 [2] (Re = 100, Strouhal $\approx$ 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 [12], 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:

$$\begin{aligned} \text{Lorenz–63: } & \dot{x} = \sigma(y - x),\ \dot{y} = x(\rho - z) - y, \\ & \dot{z} = xy - \beta z;\ (\sigma, \rho, \beta) = (10, 28, 8/3),\ x_0 = (1, 1, 1), \\ \text{Rössler: } & \dot{x} = -y - z,\ \dot{y} = x + ay,\ \dot{z} = b + z(x - c); \\ & (a, b, c) = (0.2, 0.2, 5.7),\ x_0 = (0.1, 0, 0), \\ \text{Chen: } & \dot{x} = a(y - x),\ \dot{y} = (c - a)x - xz + cy, \\ & \dot{z} = xy - bz;\ (a, b, c) = (35, 3, 28),\ x_0 = (-10, 0, 37). \end{aligned}$$

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.

B Propagators

Table 1: Encoder–Propagator–Decoder decomposition across model families.

Model	Encoder	Propagator	Decoder
MLP	MLP (feed-forward)	Identity ($\mathcal{P}(\mathbf{z}) = \mathbf{z}$)	MLP (feed-forward)
RNN (GRU)	GRU encoder (last hidden state)	Identity	MLP (feed-forward)
Autoregressive RNN	GRU encoder (last hidden state)	Identity	GRU decoder (autoregressive)
Transformer	Transformer encoder (causal attention)	Identity	Transformer decoder (causal attention)
Neural ODE (MLP/RNN/Transformer)	Same encoder as base model	Latent ODE $\dot{\mathbf{z}} = f_\Theta(\mathbf{z}, t)$	Same decoder as base model
Koopman (MLP/RNN/Transformer)	Same encoder as base model	Linear Koopman operator K : $\mathbf{z}_{k+1} = K\mathbf{z}_k$	Same decoder as base model
Echo-State Network (ESN)	None (random reservoir)	Reservoir recurrence $\mathbf{r}_{k+1} = \tanh(W\mathbf{r}_k + U\mathbf{x}_k)$	Linear readout

C Anchor ablations

Computation. We compute the relative embedding

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

where a_i is the i-th anchor, μ_i and σ_i are the mean and standard deviation of $\operatorname{sim}\big(\phi(\cdot), \phi(a_i)\big)$ 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 Figure 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 [17].

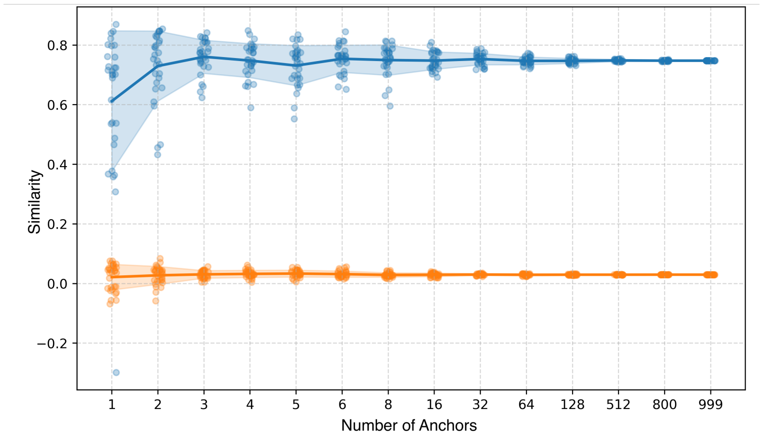

Figure 4: **Anchor ablation and baseline.** (Blue) Alignment vs. number of anchors K; lines show mean over 30 repeats. Stabilization occurs for $K \geq 16$; we choose $K = 80$ (vertical marker) for the main experiments. (Orange) Random baseline with disjoint anchor sets across spaces, yielding near-zero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

D Results on rest of the dynamical systems.

See Figure 5.

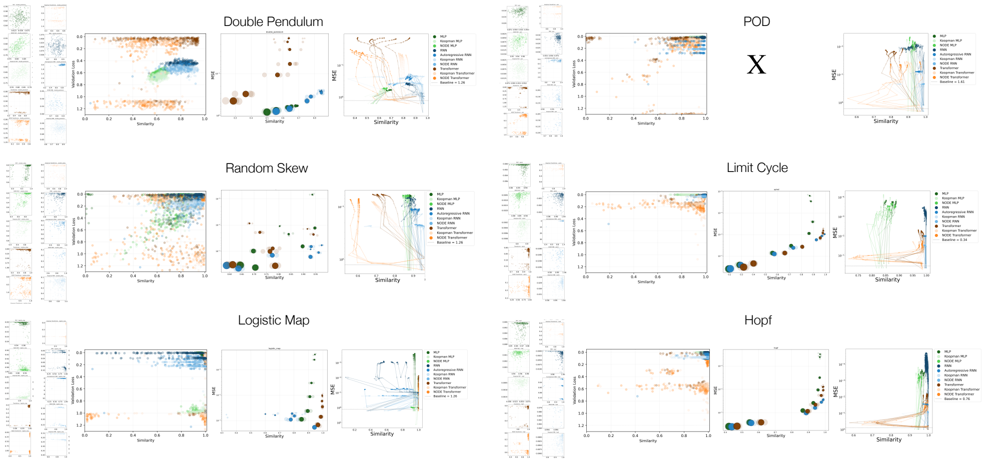

Figure 5: **Family-level alignment across different dynamical systems.** Validation mean-squared error (MSE) and representational similarity are shown for a range of dynamical systems, including the Double Pendulum, Random Skew, Logistic Map, Limit Cycle, and Hopf oscillator. Each system depicts results for model-specific tuning, noise perturbation experiments, and the relationship between representational similarity and predictive performance. (**X**) The noise perturbation experiment was omitted for the POD dataset, as it was obtained from external sources.

E Model performances

See Figure 6

F Cross-model similarity for logistic map

Additional results complementing Figure 2 are shown in Figure 7.

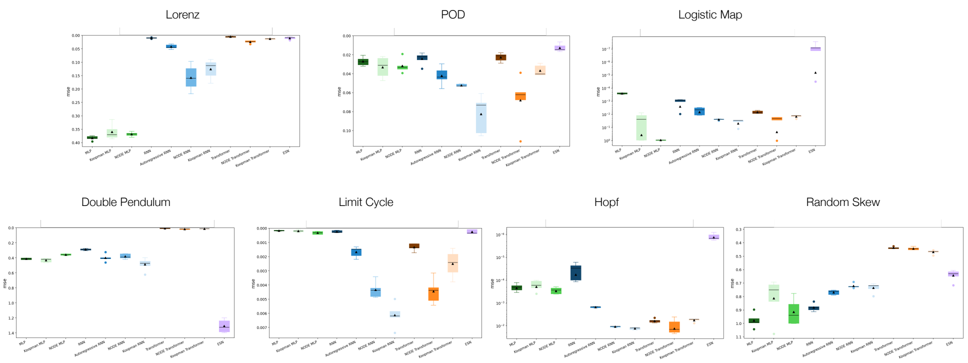

Figure 6: **Performances by test MSE by dataset.**

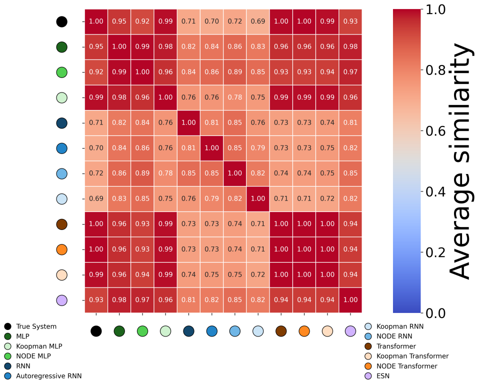

Figure 7: **Cross-Model Similarity of Logistic Map.**

G Stitching Results

enc/dec	MLP		N-MLP		K-MLP		TF		N-TF		K-TF	
	Abs.	Rel.	Abs.	Rel.	Abs.	Rel.	Abs.	Rel.	Abs.	Rel.	Abs.	Rel.
MLP	1.655	**0.383**	2.334	**0.479**	**2.459**	2.818	**0.293**	0.825	1.181	**1.067**	**0.923**	0.988
N-MLP	2.195	**0.404**	3.916	**0.491**	3.511	**3.078**	**0.233**	0.813	**0.545**	1.040	1.235	**0.925**
K-MLP	1.621	**0.753**	2.224	**0.759**	2.523	**0.891**	**0.290**	0.587	1.233	**0.974**	0.835	**0.679**
TF	**1.538**	2.019	2.389	**1.754**	**2.118**	9.517	0.265	**0.043**	1.383	**0.587**	0.599	**0.076**
N-TF	**1.514**	1.780	2.003	**1.580**	**2.039**	7.112	0.184	**0.061**	1.017	**0.757**	0.689	**0.256**
K-TF	**1.590**	2.011	2.095	**1.750**	**2.129**	9.466	0.254	**0.042**	1.325	**0.586**	0.840	**0.075**

Table 2: Cross-architecture average stitching loss (MSE) over encoder–decoder pairs for **absolute** (Abs.) and **relative** (Rel.) stitching. Each decoder column is independently normalized; darkest cell shows highest MSE and lightest shows lowest MSE respectively. Lower value per pair in bold.

H Compute resources

We ran all experiments on the RAVEN HPC cluster, which features Intel Xeon Ice Lake-SP CPUs and NVIDIA A100 GPU nodes connected via NVLink.