STATE AND FLOW: DIFFERENTIABLE SWITCHING STATE SPACE MODELS FOR DYNAMICAL SYSTEMS RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Dynamical systems reconstruction (DSR) leverages neural surrogate models to recover underlying dynamics from noisy, indirect observations. Complex real-world systems often exhibit both continuous nonlinear flows and discrete regime switching. However, existing DSR approaches tackle these challenges in isolation: Neural Ordinary Differential Equations (NODEs) capture expressive nonlinearities but lack discrete states, whereas recurrent Switching Linear Dynamical Systems (rSLDS) identify discrete regimes but restrict per-regime dynamics to linear transformations. Moreover, while DSR surrogates are intended to be simulated and perturbed, current training and evaluation typically reward reconstruction rather than autonomous generation. To bridge these gaps, we introduce a latent autoencoder framework that emphasizes autonomous rollout generation during training, along with two novel architectures: recurrent differentiable SLDS (rdSLDS), an end-to-end trainable rSLDS, and recurrent Switching Neural ODE (rSNODE), which combines discrete state inference with nonlinear NODE flows. Both models infer discrete states in an unsupervised manner. To evaluate models on systems exhibiting coupled nonlinear and switching behavior, we propose the switching Lorenz system as a new benchmark. Across chaotic, piecewise-linear switching, and chaotic switching benchmarks, rdSLDS recovers the ground truth linear regime structure but rarely converges under chaotic switching, where rSNODE alone reconstructs both the attractor geometry and the correct two-state segmentation. Applied to task-fMRI data, rSNODE generates complete regional time series from short initial windows, preserves their temporal and task-activation structure, and identifies states aligned with broad experimental conditions without using task labels. Combining interpretable switching regimes with the expressive power of NODEs yields a scalable, versatile tool for DSR in neuroimaging and beyond.

1 INTRODUCTION

Many real-world phenomena arise from dynamical systems governed by differential equations (Strogatz, 2024). For simulated systems like the Lorenz attractor (Lorenz, 1963), these equations are known and directly accessible for analysis, but for real-world systems they rarely are: systems of interest are often not directly observable, nonlinear, high-dimensional, and corrupted by noise (Brunton et al., 2016). Further, many systems switch between qualitatively different dynamical regimes over time due to external inputs, internal noise, or intrinsic state-dependent behavior (Franović et al., 2024).

Dynamical systems reconstruction (DSR) leverages machine learning to build surrogate models that mirror a system's true dynamics inferred from indirect, noisy observations alone (Chen et al., 2024; Durstewitz et al., 2023). This surrogate can then be simulated, interrogated, and perturbed to recover properties such as attractors, limit cycles, and bifurcation structure without access to governing equations. DSR is commonly performed with Recurrent Neural Networks (RNNs) (Durstewitz et al., 2023; Koppe et al., 2019) or Neural ODEs (NODEs) (Chen et al., 2019), both flexible nonlinear function approximators, or with recurrent Switching Linear Dynamical Systems (rSLDS) (Linderman et al., 2017), which are restricted to linear dynamics but offer an interpretable segmentation into discrete states.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Neural systems are a major target for DSR but exemplify the challenges above: their dynamics are nonlinear (Izhikevich, 2007; Hodgkin & Huxley, 1952), span interacting spatial and temporal scales (Breakspear, 2017), and shift across tasks, cognitive demands, sleep, and pathology (Vidaurre & Smith, 2017). Measurements, however, remain partial: invasive recordings sample only small neural populations, whereas fMRI provides spatially extensive but noisy, indirect observations of neural activity (Logothetis, 2008). DSR methods have been applied to these neural measurements with promising results, both for modeling continuous dynamics and for recovering latent state structure (Koppe et al., 2019; Volkmann et al., 2024; Nair et al., 2023; Durstewitz et al., 2023).

These applications, however, inherit four gaps in the underlying DSR methods, which persist on simulated and real-world data alike: 1) many models encode the full observation sequence at both training and inference time, so they never act as true generative surrogates (see Koppe et al. (2019)); 2) standard rSLDS implementations scale poorly and are not end-to-end differentiable; 3) a capacity gap remains between expressive but unstructured NODE or RNN approaches and the structured but limited rSLDS; 4) existing benchmarks probe nonlinear dynamics and discrete switching only in isolation.

This work addresses these four gaps. We introduce a latent autoencoder rollout framework that confines the encoder to inferring the initial latent state, after which the model generates trajectories autonomously. Within this framework, we present recurrent differentiable SLDS (rdSLDS), which uses a Gumbel-Softmax relaxation to make rSLDS end-to-end differentiable and scalable. Building on this, we introduce the recurrent switching Neural ODE (rSNODE), which parametrizes each regime with a nonlinear NODE flow, combining interpretable switching structure with flexible continuous-time dynamics. We benchmark NODE, rdSLDS, and rSNODE on the chaotic Lorenz attractor, NASCAR, and a newly introduced switching Lorenz system that switches between two chaotic attractors. We further evaluate the models on task fMRI data, assessing their ability to reconstruct neural dynamics and to recover discrete functional states.

2 Background and Related Work

2.1 Neural ODEs

NODEs (Chen et al., 2019) parametrize the derivative of a system's state with a neural network, $\frac{d\mathbf{x}}{dt} = f_\theta(\mathbf{x})$, defining a continuous-time trajectory via an ODE solver:

$$\mathbf{x}(t_1) = \mathbf{x}(t_0) + \int_{t_0}^{t_1} f_\theta(\mathbf{x}(t))\, dt. \tag{1}$$

Latent ODEs (Chen et al., 2019) embed NODE in a Variational Autoencoder: an encoder infers an approximate posterior $q_\phi(\mathbf{x}_{t_0} \mid \mathbf{y}_{0:T})$ over the initial state, the ODE evolves $\mathbf{x}(t)$ forward over $[t_0, t_T]$, and a decoder maps the trajectory back to $\hat{\mathbf{y}}(t)$. Training maximizes the ELBO:

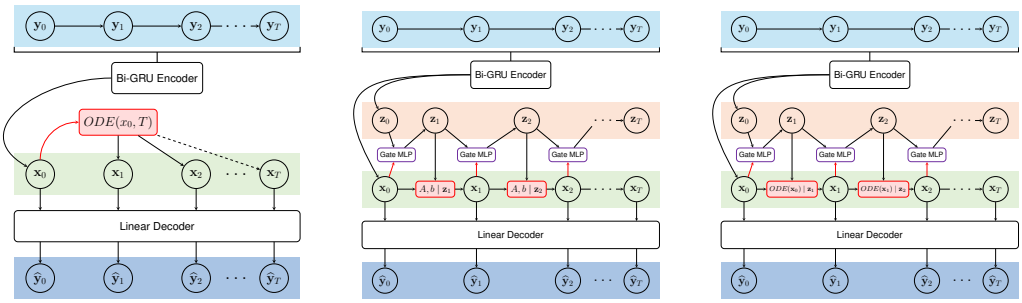
$$\mathcal{L} = \mathbb{E}_{q_\phi}\left[\sum_t \log p_\theta(\mathbf{y}_t \mid \mathbf{x}_t)\right] - \mathrm{KL}[q_\phi(\mathbf{x}_{t_0}) \,\|\, p(\mathbf{x}_{t_0})] \tag{2}$$

NODE-based models (Han et al., 2024; Cai et al., 2023; Chang et al., 2024) and related RNN architectures, such as PLRNNs (Koppe et al., 2019; Durstewitz et al., 2023; Kramer et al., 2022), are commonly used in neuroscience DSR applications; however, they typically lack switching mechanisms altogether. Meanwhile, Zhang & Saxena's Switching RNN (Zhang & Saxena, 2024) relies on encoder inputs during inference without generative evaluation, and DynaMix (Hemmer & Durstewitz, 2025; Brenner et al., 2025) mixes RNN dynamics via gating without explicit discrete states. Other extensions, such as Neural Event ODEs (Chen et al., 2021) and latent segmented ODEs (Shi & Morris, 2021), treat switching as exogenous and event-driven. Instead, rSNODE jointly learns endogenous switching through a gating network alongside nonlinear per-regime dynamics.

2.2 Recurrent Switching Linear Dynamical Systems

The rSLDS (Linderman et al., 2017) models time series with a piecewise-linear continuous latent state $\mathbf{x}_t \in \mathbb{R}^d$ that evolves under one of K linear regimes, selected by a discrete variable $z_t \in \{1, \ldots, K\}$,

$$\mathbf{x}_t = A_{z_t}\mathbf{x}_{t-1} + \mathbf{b}_{z_t} + \boldsymbol{\epsilon}_t, \qquad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, Q_{z_t}) \tag{3}$$



(a) NODE (b) rdSLDS (c) rSNODE

Figure 1: **Model diagrams. a)** NODE: an encoder maps $\mathbf{y}_{0:T}$ to an initial latent $\mathbf{x}_0$, from which an ODE solver integrates $\mathbf{x}_{0:T}$ autonomously; a linear decoder reconstructs $\hat{\mathbf{y}}_{0:T}$. **b)** rdSLDS: the encoder additionally infers a discrete latent $\mathbf{z}_0$; a gating network selects the active regime at each step, and $\mathbf{x}_t$ is updated as a Gumbel-Softmax-weighted mixture of K linear transitions $\mathbf{A}_k, \mathbf{b}_k$. **c)** rSNODE: identical to rdSLDS, but each regime's transition is a dedicated NODE $f_\theta^{(k)}$ rather than a linear map. All three models share the same latent autoencoder rollout framework: the encoder infers only the initial state, after which generation proceeds autonomously.

where z_t itself depends on z_{t-1} and $\mathbf{x}_{t-1}$ (the "recurrent" connection). Inference is typically performed via Pólya-Gamma augmentation or structured mean-field variational inference, fitting the model to $\mathbf{y}_t$ through a Gaussian or Poisson emission model (Linderman et al., 2017). Hu et al. (2025) propose a still-linear Gaussian-process SLDS with smooth regime transitions; and Karniol-Tambour et al. (2024) model nonlinear switching between brain regions but, unlike rSNODE, do not address autonomous long-horizon generation. Wang et al. (2025) use Gumbel-Softmax switching for observation-space forecasting rather than latent DSR. Unlike these, rdSLDS makes the rSLDS model end-to-end trainable while emphasizing generation, and rSNODE extends it further to nonlinear per-regime dynamics, combining capabilities none of the above provides.

3 Methods

3.1 Latent Autoencoder Rollout Framework

Existing DSR approaches often encode the full observation sequence during training rather than emphasizing autonomous generation (see Koppe et al. (2019)), while relying on supervised or pre-biased priors for regime assignment rather than unsupervised recovery (Linderman et al., 2017). We address these limitations by introducing a latent autoencoder rollout framework, shared by rdSLDS and rSNODE, that confines the encoder strictly to inferring initial latent states. The model then generates trajectories fully autonomously without further access to observations, thereby acting as a true surrogate dynamical system while recovering discrete regime structure in an unsupervised manner.

Both models maintain a continuous latent state $\mathbf{x}_t \in \mathbb{R}^d$ and a discrete latent switching variable $\mathbf{z}_t \in \Delta^{K-1}$, where K is the number of regimes. A bidirectional GRU encoder maps an observed window $\mathbf{y}_{0:T_w}$ to initial values $(\mathbf{x}_0, \mathbf{z}_0)$, from which the generative model rolls out $(\mathbf{x}_{0:T_w}, \mathbf{z}_{0:T_w})$ autoregressively. A decoder then maps $\mathbf{x}_{0:T_w}$ back to observation space $\hat{\mathbf{y}}_{0:T_w}$. The rollout length is unconstrained at inference: the model is initialized from a window of length T_w and generates for T steps, with $T \gg T_w$ in all evaluations (Section 4). Discrete regime assignments are made differentiable via the Gumbel-Softmax (Jang et al., 2016) estimator with temperature τ, annealed from τ_{start} to τ_{end} over training to sharpen assignments toward one-hot vectors. The training objective is to maximize a modified β-ELBO:

$$\mathcal{L} = \mathbb{E}_q \left[\sum_t \log p_\theta(\mathbf{y}_t \mid \mathbf{x}_t) \right] - \beta(\mathrm{KL}[q(\mathbf{x}_0) \,\|\, p(\mathbf{x}_0)] + \mathrm{KL}[q(z_0) \,\|\, p(z_0)]) \tag{4}$$

where $p(\mathbf{x}_0) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, $p(z_0) = \mathrm{Cat}(1/K)$, and β is linearly annealed from 0 to $\beta_{\max}$.

3.2 Recurrent Differentiable SLDS

The rdSLDS implements the rSLDS generative model by Linderman et al. (2017) within the framework above. The encoder infers

$$\boldsymbol{\mu}_{q_0},\, \log \boldsymbol{\sigma}^2_{q_0},\, q(z_0) = \mathrm{Enc}_\phi(\mathbf{y}_{0:T_w}) \tag{5}$$

and the initial continuous state is sampled via reparametrization as $\mathbf{x}_0 = \boldsymbol{\mu}_{q_0} + \boldsymbol{\sigma}_{q_0} \odot \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, with $q(z_0)$ giving the probability distribution over the initial discrete state. At each rollout step, a gating network g_ψ conditioned on the previous continuous ("recurrent") and discrete latents selects the active regime

$$\tilde{\mathbf{z}}_t = \mathrm{GumbelSoftmax}(g_\psi(\mathbf{x}_{t-1}, \mathbf{z}_{t-1}),\, \tau) \tag{6}$$

and the continuous state is updated as a soft mixture of K linear transitions

$$\mathbf{x}_t = \sum_{k=1}^{K} \tilde{z}_t^{(k)}\, (A_k\, \mathbf{x}_{t-1} + \mathbf{b}_k) \tag{7}$$

where $A_k \in \mathbb{R}^{d\times d}$ and $\mathbf{b}_k \in \mathbb{R}^d$ are learnable per-regime transition matrices and biases, initialized close to the identity.

3.3 Recurrent Switching Neural ODE

The rSNODE extends the rdSLDS by replacing linear per-regime transition matrices with NODE flows while retaining an identical encoder architecture for inferring initial discrete and continuous latents. This preserves the interpretable switching structure while substantially increasing capacity to model complex nonlinear dynamics. Rather than applying a linear map $A_k\mathbf{x}_{t-1}+\mathbf{b}_k$ per regime, each of the K regimes is governed by a dedicated n-layer NODE $f_\theta^{(k)}$ with regime-specific parameters $\theta^{(k)}$. All K ODEs are integrated in parallel over a unit time step, yielding K candidate next states

$$\mathbf{x}_t^{(k)} = \mathbf{x}_{t-1} + \int_{t-1}^{t} f_\theta^{(k)}(\mathbf{x}(s))\, ds, \quad k = 1, \ldots, K \tag{8}$$

The continuous latent state is updated as a Gumbel-Softmax weighted mixture of K ODE solutions (analogous to Equation 7):

$$\mathbf{x}_t = \sum_{k=1}^{K} \tilde{z}_t^{(k)}\, \mathbf{x}_t^{(k)} \tag{9}$$

where $\tilde{\mathbf{z}}_t$ is computed according to Equation 6. The regime-specific functions $f_\theta^{(k)}$ share an identical architecture while maintaining distinct parameters, implemented as a batched operation over all K regimes to ensure constant wall-clock training time across values of K (see Table 1). The rSNODE is trained using the same objective function and annealing schedule as the rdSLDS (Equation 4). Gradients flow via backpropagation through the ODE solver steps (Chen et al., 2019) and the Gumbel-Softmax weights, maintaining end-to-end differentiability across the full model.

4 Results

4.1 Reconstruction of benchmark dynamical systems

We evaluate NODE, rdSLDS, and rSNODE on three simulated benchmarks. For each benchmark, models are trained on 4 trajectories and evaluated on 1 unseen validation trajectory across 15 distinct training seeds; metrics are averaged across all 5 trajectories per seed, as train and validation performances were consistently equivalent. Runs whose rollouts diverge are excluded from metric aggregation, and the number of converged seeds N is reported per condition (Appendix B.7.6). Across all experiments, models share identical hyperparameters (Table 2) and rely on unsupervised discrete state assignment, using ground truth labels strictly for post-hoc evaluation. Statistical comparisons use Bonferroni-corrected Mann-Whitney U tests (Appendix B.7.4). Main evaluations report trajectory Maximum Mean Discrepancy (MMD; Appendix B.7.1), as MSE is non-meaningful for chaotic systems; we provide MSE, MLE, and PSC in Table 4.

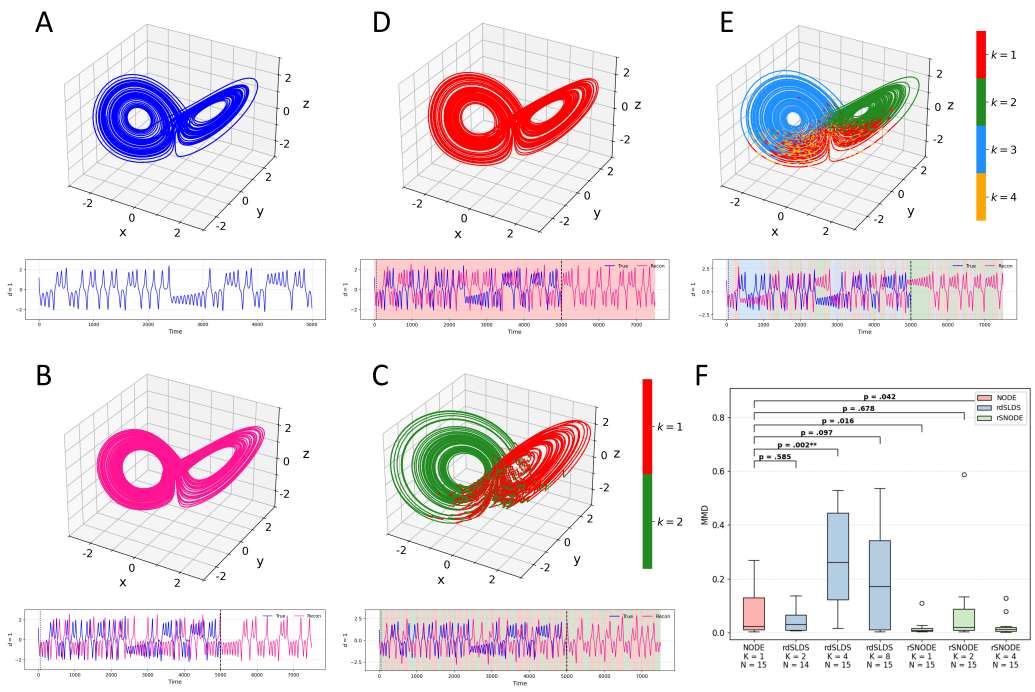


Figure 2: **Chaotic Lorenz results.** All models successfully reconstruct the chaotic Lorenz dynamics. **A.** True 3D attractor (top) and $d = 1$ trajectory, $T = 5000$ (bottom). **B–D.** NODE, rdSLDS ($K = 2$), and rSNODE ($K = 1$) reconstructions, initialized from a $T_w = 50$ encoded window (dotted) and unrolled to $T = 7500$, beyond the training length (dashed); all three recover the butterfly attractor. **E.** rSNODE ($K = 4$): nonlinear per-regime dynamics render the state segmentation arbitrary. **F.** MMD across 15 seeds (N converged) for NODE, rdSLDS ($K \in \{2, 4, 8\}$), rSNODE ($K \in \{1, 2, 4\}$).

4.1.1 Chaotic Lorenz

We examine the models' ability to reconstruct highly nonlinear single-regime systems on the widely used chaotic Lorenz benchmark. The Lorenz system (Lorenz, 1963), modeling atmospheric convection, is a canonical benchmark for chaotic dynamics, governed by:

$$\dot{u} = \sigma(v - u), \quad \dot{v} = u(\rho - w) - v, \quad \dot{w} = uv - \beta w \tag{10}$$

with standard parameters $\sigma = 10$, $\rho = 28$, $\beta = 8/3$. Its positive Lyapunov exponent makes it a challenging test of a model's capacity to reconstruct chaotic geometry, while the butterfly attractor allows for visual assessment of reconstruction performance.

Figure 2 shows reconstructed phase portraits and MMD across seeds for all three models. NODE and rSNODE ($K = 1$) both accurately reconstruct the attractor with comparable MMD, confirming that rSNODE reduces to a standard NODE at $K = 1$. rdSLDS requires at least $K = 2$ states, each dominating one wing; performance degrades with larger K, indicating two linear regimes are optimal. All models generate chaotic dynamics, with average MLE matching the true value of $\lambda \approx 0.9$ (Table 4). Unlike rdSLDS, rSNODE maintains reconstruction quality with $K > 1$ while roughly favoring a two-state segmentation (Figure 10); MMD plateaus at $K = 2$ for rdSLDS and at $K = 1$ for rSNODE (Figure 9). Results are qualitatively unchanged when substantial Gaussian observation noise ($\sigma_{obs} = 0.5$) is added to the trajectories: all models recover the clean attractor (Appendix C.2).

4.1.2 NASCAR

Linear switching architectures such as rdSLDS are naturally suited to reconstructing systems with piecewise-linear dynamics; therefore, we evaluate our models on the linear switching NASCAR benchmark (Linderman et al., 2017). This benchmark features 2D latent dynamics switching across four linear regimes arranged around an oval track, mapped to a 10-dimensional observation space

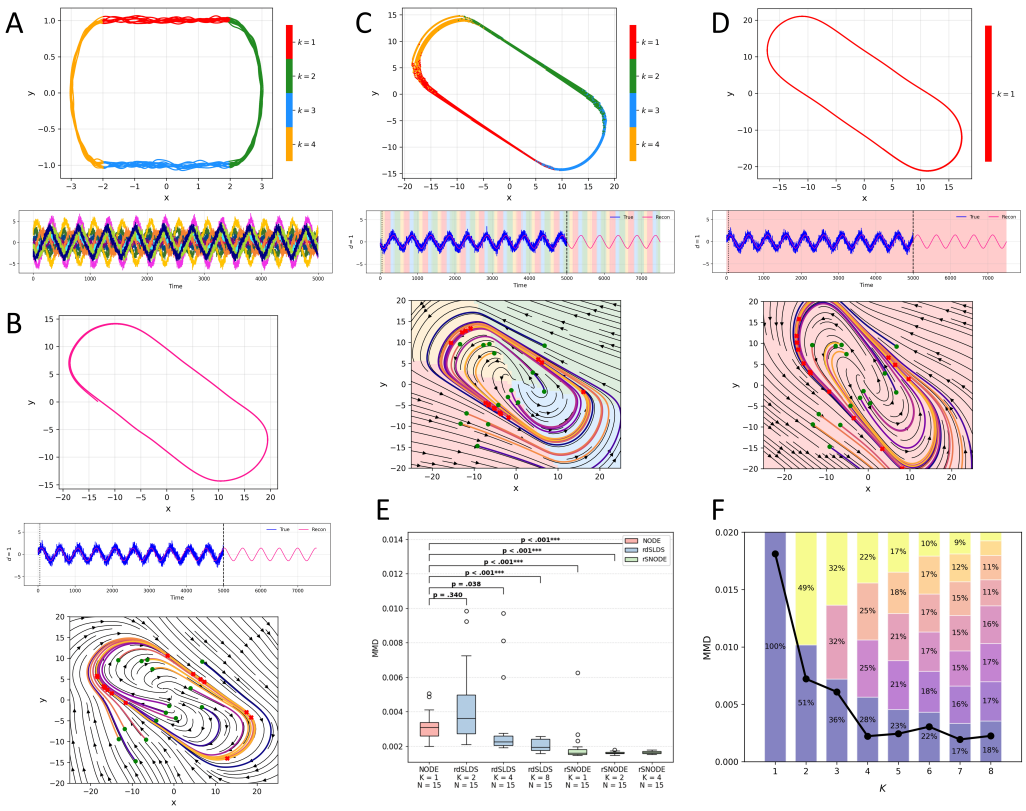


Figure 3: **NASCAR results.** Piecewise-linear rdSLDS alone discovers the correct 4-regime segmentation while faithfully reconstructing the trajectories. **A.** True 2D latent dynamics with 4 linear regimes (top) and 10D noisy observations ($\sigma_{obs} = 0.5$). **B–D.** NODE, rdSLDS ($K = 4$), and rSNODE ($K = 1$) reconstructions (middle), recovered latent limit cycles (top), and latent unrolls from random initial states (bottom); only rdSLDS recovers the ground truth 4-regime segmentation (top, C). **E.** MMD across 15 seeds (N converged) for NODE, rdSLDS ($K \in \{2, 4, 8\}$), rSNODE ($K \in \{1, 2, 4\}$). **F.** MMD and state usage vs. K for rdSLDS, averaged over 3 seeds (see Figure 13 for the rSNODE ablation).

with Gaussian noise $\sigma_{\text{obs}} = 0.5$ (Figure 3A, Appendix B.6.3). Its known ground truth state sequence provides an ideal benchmark for evaluating both discrete regime segmentation and continuous trajectory reconstruction.

All three models achieve competitive reconstruction quality (Figure 3E) but differ in recovering the underlying state structure. NODE reconstructs the dynamics accurately but provides no state segmentation. Only rdSLDS with $K \geq 4$ recovers the correct four-regime structure, matching ground truth and plateauing at $K = 4$ (Figure 3F); with $K = 2$ rdSLDS performs significantly worse than all other models. rSNODE learns the correct latent trajectory for any K but does not discover a meaningful segmentation, instead favoring an arbitrary two-state split (Figure 14), as its nonlinear dynamics with $K = 1$ suffice to represent the full trajectory. Generative rollouts from all models reproduce the NASCAR limit cycle for $T = 5000$ steps, even when setting the initial condition to latent states not sampled during training.

4.1.3 Switching Lorenz

The previous benchmarks each tested either nonlinear or linear-switching reconstruction in isolation. To evaluate rSNODE's ability to capture *nonlinear switching* systems specifically, we introduce the switching Lorenz benchmark: two chaotic Lorenz attractors, generating a three-wing chaotic

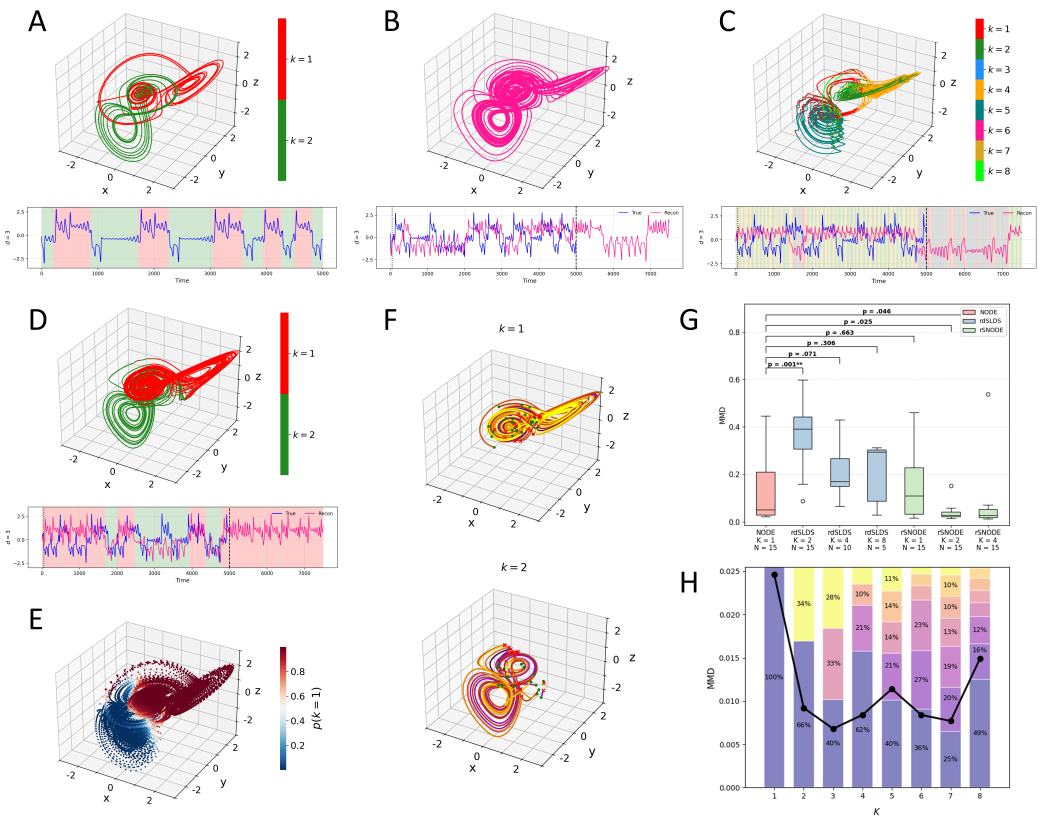


Figure 4: **Switching Lorenz results.** By combining discrete states with continuous nonlinear dynamics, the rSNODE accurately identifies overlapping chaotic regimes and outperforms both non-switching NODE and piecewise-linear rdSLDS. **A.** True 3D attractor: two chaotic Lorenz systems, sharing an overlapping wing. **B.** NODE reconstructs the attractor beyond the training length (dashed) but merges both regimes, missing the 2-state structure. **C.** rdSLDS ($K = 8$) requires many states, giving sharp, fragmented transitions. **D.** rSNODE ($K = 2$) accurately reconstructs the 3-wing structure and recovers the correct 2-state segmentation. **E.** rSNODE's gating network separates the two regimes, with intermediate state probability ($p = 0.5$) in the overlap region. **F.** Per-regime ODE rollouts ($k = 1, 2$) recover the two individual Lorenz attractors. **G.** MMD across 15 seeds (N converged); rSNODE ($K \geq 2$) achieves the lowest MMD. **H.** MMD and state usage vs. K for rSNODE, averaged over 3 seeds (see Figure 15 for the rdSLDS ablation).

structure that switches between the two attractors at two fixed crossover spheres (Appendix B.6.4). This benchmark requires both flexible nonlinear within-regime dynamics and discrete switching.

rSNODE with $K = 2$ achieves the most stable and accurate reconstruction across all models. NODE merges both attractors into a single continuous system, achieving good reconstruction but failing to capture the regime boundary (Figure 4B). rdSLDS fails to provide meaningful reconstructions for most seeds across all tested K; successful runs confirm the system can in principle be decomposed into linear states for sufficiently high K (Figure 4C). rSNODE with $K = 1$ behaves like NODE, merging both regimes, while $K = 2$ recovers the correct geometry and a gating boundary that closely tracks the true switching locations (Figure 4E), consistently across seeds; MMD plateaus for $K \geq 2$ (Figure 4H). Per-regime ODE rollouts recover the two individual Lorenz attractors, including phase-space regions never sampled during training (Figure 4F), confirming that the learned dynamics generalize beyond the training distribution.

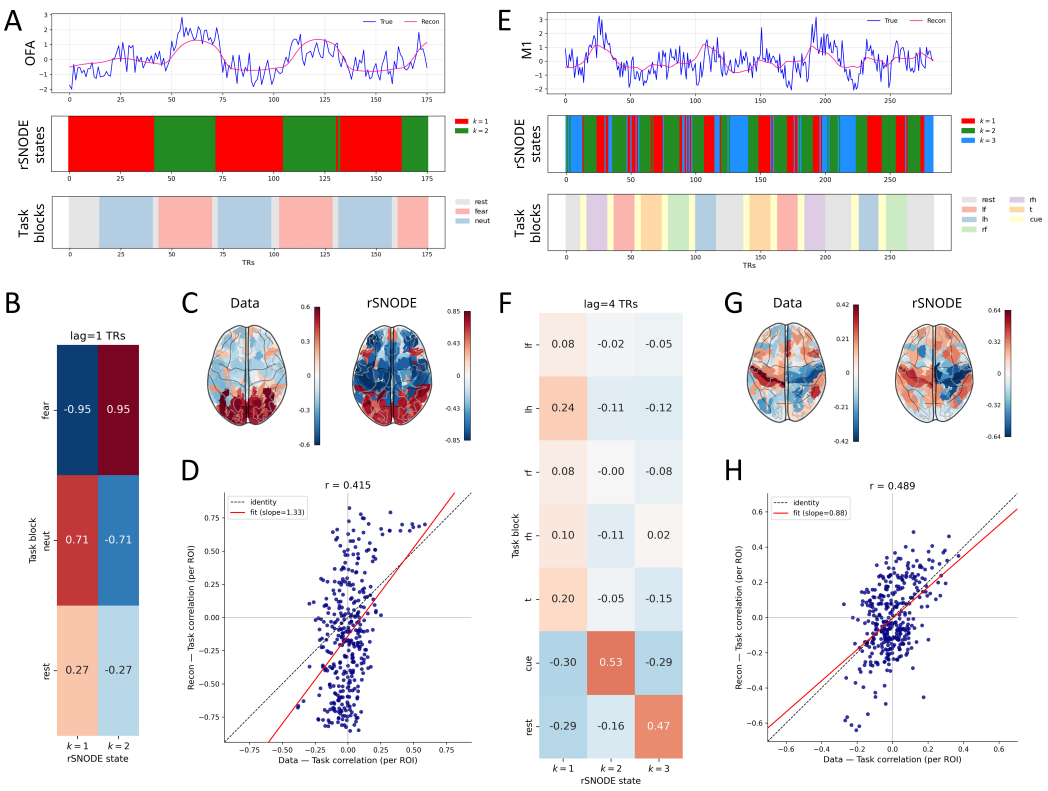


Figure 5: **Task fMRI results.** rSNODE successfully reconstructs task fMRI data, with discrete states corresponding to task blocks. All panels show one representative subject; additional subjects are shown in Appendix D. Given a $T_w = 50$ encoded window, rSNODE reconstructs ROI trajectories for EMOTION (**A**, top; $K = 2$, OFA = occipital face area) and MOTOR (**E**, top; $K = 3$, M1 = primary motor area). **A.** (middle, bottom) Recovered EMOTION states align with the *fear*/*neut* task blocks (*rest* is grouped with *neut*). **B.** Cross-correlation at optimal lag confirms state-block alignment. **C, D.** Task activations (*fear–neut* contrast) and true-vs-reconstructed ROI-task correlation match the true data. **E.** (middle, bottom) MOTOR: $K = 3$ states correspond to *cue*, *rest*, and pooled motion blocks (*lf/lh/rf/rh/t*). **F.** Cross-correlation at optimal lag. **G, H.** Task activations ($(rh + rf) - (lh + lf)$ contrast) and ROI-task correlation confirm high fidelity.

4.2 Reconstruction of task fMRI data

We further tested the models on task fMRI data from the Human Connectome Project (HCP) (Van Essen et al., 2012). Models were trained on EMOTION and MOTOR tasks across 50 subjects (40 training, 10 validation; 2 phase-encoding directions per subject) using HCP-preprocessed BOLD signals (Glasser et al., 2013) parcellated into 360 Glasser ROIs (Glasser et al., 2016) (details in Appendix B.6.5). Quantitative metrics and state-alignment analyses are reported across 5 random seeds for 5 subjects (3 training, 2 validation), using the hyperparameters described in Table 3. Reconstructions and state alignments for additional training and validation subjects are provided in Appendix D and do not differ qualitatively. Because ground truth dynamics are unknown for real neural data, models are evaluated on ROI timeseries reconstruction quality and alignment between recovered discrete states and experimental task conditions. Task labels were strictly withheld during training and inference. Table 5 reports these quantitative metrics (MMD, MSE, PSC) for rSNODE alongside baseline evaluations for NODE and rdSLDS (detailed in Appendices D.4 and D.5).

For the EMOTION task, optimal performance was achieved with $D_L = 3$ latent dimensions and $K = 2$ states. Cross-correlation analysis (Appendix B.7.5) shows the recovered states align strongly with the task blocks (Figure 5A,B): state 1 with *neut* and state 2 with *fear*, while the short *rest* intervals mostly grouped into *neut*. With $K = 3$, the model could not consistently assign a distinct *rest* state

(Figure 19). rSNODE also reconstructs the full $T = 176$ ROI timeseries given only a $T_w = 50$ window, while preserving the temporal frequency structure (PSC = 0.591 ± 0.013) and reproducing the true brain activation profile (Figure 5C,D). rdSLDS results with identical hyperparameters (Figure 22) are highly comparable.

For the MOTOR task, rSNODE achieves the most stable segmentation of the 7 task blocks with $D_L = 18$ and $K = 3$, with states corresponding to *rest*, *cue*, and all motion blocks pooled together, without separating left and right motor tasks (Figure 5E,F). With $K = 7$, rSNODE relied mainly on ~ 3 of the states, with inconsistent and arbitrary use of the remaining states (Figure 19). rSNODE also reconstructs the full $T = 284$ ROI timeseries from only a $T_w = 50$ window, while preserving the temporal frequency structure (PSC = 0.659 ± 0.009) and reproducing the true brain activation profile (Figure 5G,H). rdSLDS results (Figure 23) are again highly comparable.

5 Discussion

Current DSR methods lack end-to-end trainable latent models capable of reconstructing the nonlinear switching dynamics assumed to underlie many complex real-world systems (Franović et al., 2024). We addressed this gap with rdSLDS and rSNODE, two fully differentiable architectures trained end-to-end within a latent autoencoder rollout framework. By confining the encoder strictly to inferring initial latent states, our models act as true generative surrogates: they infer discrete regime structure in an unsupervised manner and generate long-horizon trajectory rollouts autonomously given only an initial state. Benchmarking on chaotic, piecewise-linear, and chaotic switching systems, alongside task fMRI data, demonstrates that combining structured discrete switching with continuous neural flows enables faithful reconstruction across diverse dynamical regimes.

Our experiments reveal a central trade-off between discrete-state interpretability and dynamical complexity, where increasing continuous model capacity often comes at the expense of meaningful regime segmentation. While a single-regime NODE ($K = 1$) captures complex global state-space geometry well, it lacks discrete segmentation mechanisms entirely. Conversely, rdSLDS recovers linear switching regimes accurately on NASCAR, but diverges on most seeds under chaotic switching across all tested K, indicating that piecewise-linear regimes cannot accommodate chaotic within-regime dynamics. rSNODE bridges this divide by combining nonlinear per-regime flows with discrete gating, achieving the lowest MMD on the switching Lorenz system while recovering the correct two-state segmentation. However, on simpler benchmarks such as NASCAR or chaotic Lorenz, rSNODE is over-parameterized relative to the underlying dynamics and favors a less interpretable state split while still fitting trajectories accurately. When ground truth is unavailable, a segmentation that is consistent across seeds and values of K can therefore be read as evidence of genuine discrete structure rather than an artifact of model capacity. Integrating all K vector fields is batched across regimes and incurs no measurable wall-clock cost at the scales considered here (Table 1). The benefits of this parallelization are expected to diminish only at substantially larger K or latent dimensionality.

When evaluated on task fMRI data, rSNODE accurately reconstructs regional timeseries and identifies discrete states aligning with experimental task conditions without receiving task labels. However, rdSLDS achieves comparable trajectory reconstruction quality, indicating that these macroscopic task-relevant dynamics can be captured via approximately linear transitions (Nozari et al., 2024; Boynton et al., 1996). Furthermore, both tasks could in principle be reconstructed with $K = 1$ given a sufficiently expressive ODE, underscoring the general capacity/sparsity trade-off: overly powerful per-regime ODEs allow the model to ignore discrete regime transitions, whereas under-parameterized ODEs can yield arbitrary or spurious state segmentations.

Crucially, user-defined choices such as K, ODE capacity, and selecting between linear (rdSLDS) and nonlinear (rSNODE) vector fields remain primary limitations of our current architectures. As observed on the NASCAR benchmark, rSNODE's tendency to over-parameterize simpler systems highlights that automatic model complexity selection remains an open challenge in DSR. Incorporating automatic relevance determination, non-parametric priors, or sparsity penalties over both K and ODE architectures (Geadah, 2024) represents a promising avenue toward unified models capable of diagnosing both state structure and dynamical complexity autonomously. Ultimately, we demonstrate that combining structured discrete switching with flexible continuous dynamics, trained explicitly for autonomous generative rollout, provides a principled direction for dynamical systems reconstruction from high-dimensional, noisy observations across scientific domains.

AI USE STATEMENT

In this work, we used generative AI tools for providing feedback on research methodology. We have not used generative AI tools to generate synthetic datasets, help develop theoretical models or conceptual frameworks, propose or refine hypotheses, implement methods, clean and reformat datasets, support qualitative and thematic data analysis, or interpret results. Formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, and assisting with translation are not applicable to this work. Additionally, we used generative AI tools for brainstorming, editing and debugging code, and adhering to formatting guidelines. We have reviewed all AI-assisted work, and all AI-generated code was verified and tested. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work reanalyzes de-identified task fMRI data from the Human Connectome Project (HCP), a publicly available dataset collected under institutional ethics approval and informed consent obtained by the original HCP consortium; no new human subjects data were collected for this study, and data were used in accordance with the HCP Data Use Terms. All other experiments use simulated benchmark dynamical systems (chaotic Lorenz, NASCAR, switching Lorenz) and raise no human-subjects or dual-use concerns. We see no other ethical concerns pertaining to this work.

REPRODUCIBILITY STATEMENT

We describe our latent autoencoder rollout framework, the rdSLDS and rSNODE generative models, and the training objective in detail in Section 3.1–3.3. Full architectural and training hyperparameters for all models, on both the benchmark systems and the task fMRI data, are listed in Appendix B.4 and B.5. The data generation procedure for each benchmark dynamical system (chaotic Lorenz, noisy Lorenz, NASCAR, and the newly introduced switching Lorenz) is given in Appendix B.6.1–B.6.4, and the HCP task fMRI preprocessing and parcellation pipeline is described in Appendix B.6.5. All evaluation metrics (MMD, MSE, MLE, PSC) and statistical tests are defined in Appendix B.7. Code for NODE, rdSLDS, rSNODE, and all benchmark experiments, along with the benchmark dynamical systems data and generation scripts, will be made publicly available upon publication. Task fMRI data are third-party data obtained from the HCP and are publicly available at https://www.humanconnectome.org, subject to the HCP Data Use Terms.

ACKNOWLEDGMENTS

Task fMRI data were provided by the Human Connectome Project, WU-Minn Consortium (Principal Investigators: David Van Essen and Kamil Ugurbil; 1U54MH091657) funded by the 16 NIH Institutes and Centers that support the NIH Blueprint for Neuroscience Research; and by the McDonnell Center for Systems Neuroscience at Washington University.

REFERENCES

Deanna M. Barch, Gregory C. Burgess, Michael P. Harms, Steven E. Petersen, Bradley L. Schlaggar, Maurizio Corbetta, Matthew F. Glasser, Sandra Curtiss, Sachin Dixit, Cindy Feldt, Dan Nolan, Edward Bryant, Tucker Hartley, Owen Footer, James M. Bjork, Russ Poldrack, Steve Smith, Heidi Johansen-Berg, Abraham Z. Snyder, and David C. Van Essen. Function in the human connectome: Task-fMRI and individual differences in behavior. *NeuroImage*, 80:169–189, 10 2013. ISSN 1053-8119. doi: 10.1016/J.NEUROIMAGE.2013.05.033.

Geoffrey M. Boynton, Stephen A. Engel, Gary H. Glover, and David J. Heeger. Linear Systems Analysis of Functional Magnetic Resonance Imaging in Human V1. *Journal of Neuroscience*, 16(13):4207–4221, 7 1996. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.16-13-04207.1996. URL https://www.jneurosci.org/content/16/13/4207https://www.jneurosci.org/content/16/13/4207.abstract.

Michael Breakspear. Dynamic models of large-scale brain activity, 2 2017. ISSN 15461726.

Manuel Brenner, Elias Weber, Georgia Koppe, and Daniel Durstewitz. Learning Interpretable Hierarchical Dynamical Systems Models from Time Series Data. 2 2025. URL http://arxiv.org/abs/2410.04814.

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 of the United States of America*, 113(15):3932–3937, 4 2016. ISSN 10916490. doi: https://doi.org/10.1073/pnas.1517384113.

Hongmin Cai, Tingting Dan, Zhuobin Huang, and Guorong Wu. OSR-NET: Ordinary Differential Equation-Based Brain State Recognition Neural Network. In *Proceedings - International Symposium on Biomedical Imaging*, volume 2023-April. IEEE Computer Society, 2023. ISBN 9781665473583. doi: 10.1109/ISBI53787.2023.10230734.

Yin Jui Chang, Yuan I. Chen, Hannah M. Stealey, Yi Zhao, Hung Yun Lu, Enrique Contreras-Hernandez, Megan N. Baker, Edward Castillo, Hsin Chih Yeh, and Samantha R. Santacruz. Multiscale effective connectivity analysis of brain activity using neural ordinary differential equations. *PLoS ONE*, 19(12), 12 2024. ISSN 19326203. doi: 10.1371/journal.pone.0314268.

Jiarui Chen, Anastasia Benedyk, Alexander Moldavski, Heike Tost, Andreas Meyer-Lindenberg, Urs Braun, Daniel Durstewitz, Georgia Koppe, and Emanuel Schwarz. Quantifying brain-functional dynamics using deep dynamical systems: Technical considerations. *iScience*, 27(8), 8 2024. ISSN 25890042. doi: 10.1016/j.isci.2024.110545.

Ricky Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural Ordinary Differential Equations. 12 2019. URL http://arxiv.org/abs/1806.07366.

Ricky Chen, Brandon Amos, and Maximilian Nickel. Learning Neural Event Functions for Ordinary Differential Equations. 10 2021. URL http://arxiv.org/abs/2011.03902.

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, 11 2023. ISSN 14710048. doi: 10.1038/s41583-023-00740-7.

Igor Franović, Sebastian Eydam, and Deniz Eroglu. Regime switching in coupled nonlinear systems: Sources, prediction, and control—Minireview and perspective on the Focus Issue. *Chaos*, 34(12), 12 2024. ISSN 10897682. doi: 10.1063/5.0247498/3323324. URL /aip/cha/article/34/12/120401/3323324/Regime-switching-in-coupled-nonlinear-systems.

Victor Geadah. Parsing Neural Dynamics with Infinite Recurrent Switching Linear Dynamical Systems. Technical report, 2024.

Matthew F. Glasser, Stamatios N. Sotiropoulos, J. Anthony Wilson, Timothy S. Coalson, Bruce Fischl, Jesper L. Andersson, Junqian Xu, Saad Jbabdi, Matthew Webster, Jonathan R. Polimeni, David C. Van Essen, and Mark Jenkinson. The minimal preprocessing pipelines for the Human Connectome Project. *NeuroImage*, 80:105–124, 10 2013. ISSN 1053-8119. doi: 10.1016/J.NEUROIMAGE.2013.04.127.

Matthew F. Glasser, Timothy S. Coalson, Emma C. Robinson, Carl D. Hacker, John Harwell, Essa Yacoub, Kamil Ugurbil, Jesper Andersson, Christian F. Beckmann, Mark Jenkinson, Stephen M. Smith, and David C. Van Essen. A multi-modal parcellation of human cerebral cortex. *Nature 2016 536:7615*, 536(7615):171–178, 7 2016. ISSN 1476-4687. doi: 10.1038/nature18933. URL https://www.nature.com/articles/nature18933.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 3 2012. doi: 10.5555/2188385.2188410. URL https://dl.acm.org/doi/10.5555/2188385.2188410.

Kaiqiao Han, Yi Yang, Zijie Huang, Xuan Kan, Ying Guo, Yang Yang, Lifang He, Liang Zhan, Yizhou Sun, Wei Wang, and Carl Yang. BrainODE: Dynamic Brain Signal Analysis via Graph-Aided Neural Ordinary Differential Equations. In *BHI 2024 - IEEE-EMBS International Conference on Biomedical and Health Informatics, Proceedings*. Institute of Electrical and Electronics Engineers Inc., 2024. ISBN 9798350351552. doi: 10.1109/BHI62660.2024.10913768.

Christoph Jürgen Hemmer and Daniel Durstewitz. True Zero-Shot Inference of Dynamical Systems Preserving Long-Term Statistics. 1 2025. URL http://arxiv.org/abs/2505.13192.

A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4):500, 8 1952. ISSN 14697793. doi: 10.1113/JPHYSIOL.1952.SP004764. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC1392413/.

Amber Hu, David Zoltowski, Aditya Nair, David Anderson, Lea Duncker, and Scott Linderman. Modeling Latent Neural Dynamics with Gaussian Process Switching Linear Dynamical Systems. 1 2025. URL http://arxiv.org/abs/2408.03330.

Eugene M Izhikevich. *Dynamical systems in neuroscience*. MIT press, 2007. ISBN 9780262090438.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical Reparameterization with Gumbel-Softmax. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 11 2016. URL https://arxiv.org/abs/1611.01144v5.

Orren Karniol-Tambour, David M Zoltowski, E Mika Diamanti, Lucas Pinto, Carlos D Brody, David W Tank, and Jonathan W Pillow. Modeling state-dependent communication be-tween brain regions with switching nonlinear dynamical systems. In *International Conference on Learning Representations*, 2024.

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), 2019. ISSN 15537358. doi: 10.1371/journal.pcbi.1007263.

Daniel Kramer, Philine Lou Bommer, Carlo Tombolini, Georgia Koppe, and Daniel Durstewitz. Reconstructing Nonlinear Dynamical Systems from Multi-Modal Time Series. 7 2022. URL http://arxiv.org/abs/2111.02922.

Scott W Linderman, Matthew J Johnson, Andrew C Miller, Ryan P Adams David M Blei Liam Paninski Harvard, and Google Brain. Bayesian Learning and Inference in Recurrent Switching Linear Dynamical Systems, 4 2017. ISSN 2640-3498. URL https://proceedings.mlr.press/v54/linderman17a.html.

Nikos K. Logothetis. What we can do and what we cannot do with fMRI. *Nature 2008 453:7197*, 453(7197):869–878, 6 2008. ISSN 1476-4687. doi: 10.1038/nature06976. URL https://www.nature.com/articles/nature06976.

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

Henry B. Mann and Donald R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pp. 50–60, 1947.

Aditya Nair, Tomomi Karigo, Bin Yang, Surya Ganguli, Mark J. Schnitzer, Scott W. Linderman, David J. Anderson, and Ann Kennedy. An approximate line attractor in the hypothalamus encodes an aggressive state. *Cell*, 186(1):178–193, 1 2023. ISSN 10974172. doi: 10.1016/j.cell.2022.11.027.

Erfan Nozari, Maxwell A Bertolero, Jennifer Stiso, Lorenzo Caciagli, Eli J Cornblath, Xiaosong He, Arun S Mahadevan, George J Pappas, and Dani S Bassett. Macroscopic resting-state brain dynamics are best described by linear models. *Nature Biomedical Engineering*, 8:68–84, 2024. doi: 10.1038/s41551-023-01117-y. URL https://doi.org/10.1038/s41551-023-01117-y.

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, 5 1993. ISSN 0167-2789. doi: 10.1016/0167-2789(93)90009-P.

Ruian Shi and Quaid Morris. Segmenting Hybrid Trajectories using Latent ODEs. Technical report, 2021.

Steven H. Strogatz. Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering, Third Edition. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering, Third Edition*, pp. 1–601, 1 2024. doi: https://doi.org/10.1201/9780429398490. URL https://www.taylorfrancis.com/books/mono/10.1201/9780429398490/nonlinear-dynamics-chaos-steven-strogatz.

D. C. Van Essen, K. Ugurbil, E. Auerbach, D. Barch, T. E.J. Behrens, R. Bucholz, A. Chang, L. Chen, M. Corbetta, S. W. Curtiss, S. Della Penna, D. Feinberg, M. F. Glasser, N. Harel, A. C. Heath, L. Larson-Prior, D. Marcus, G. Michalareas, S. Moeller, R. Oostenveld, S. E. Petersen, F. Prior, B. L. Schlaggar, S. M. Smith, A. Z. Snyder, J. Xu, and E. Yacoub. The Human Connectome Project: A data acquisition perspective, 10 2012. ISSN 10538119.

D Vidaurre and SM Smith. Brain network dynamics are hierarchically organized in time. *Proceedings of the National Academy of Sciences of the United States of America*, 114(48):12827–12832, 11 2017. doi: 10.1073/pnas.1705120114. URL https://www.pnas.org/doi/abs/10.1073/pnas.1705120114.

Eric Volkmann, Alena Brändle, Daniel Durstewitz, and Georgia Koppe. A scalable generative model for dynamical system reconstruction from neuroimaging data. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/9304fe9848ca4db73c79ac0bac414aef-Abstract-Conference.html.

Yiliu Wang, Timothy Doyeon Kim, Eric Shea-Brown, and Uygar Sümbül. Interpretable time series analysis with Gumbel dynamics. 9 2025. URL http://arxiv.org/abs/2509.21578.

Peter D. Welch. The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, 1967. ISSN 00189278. doi: 10.1109/TAU.1967.1161901.

Y Zhang and S Saxena. Inference of neural dynamics using switching recurrent neural networks. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/ed8baaf9059a5cee3ffe56cedbf26f69-Abstract-Conference.html.

A Extended Discussion: Design Choices and Future Directions

Design choices. Three choices mattered most in practice. Training with autonomous latent rollout, rather than teacher forcing or continuous encoder access, is what makes the learned vector field self-consistent over long horizons, the key methodological departure from standard autoencoder training. Further, chaotic reconstruction was highly sensitive to temporal resolution: training at a coarser sampling interval ($\Delta t = 0.1$) failed consistently even with zero observation noise, since coarse sampling distorts chaotic topology far more than additive noise does. Finally, a simple linear decoder matched or exceeded MLP, convolutional, and HRF-based decoders (Koppe et al., 2019) in our tests, indicating that the main representational burden lies in the latent dynamics rather than the observation model; all four decoders remain available in our codebase.

A heuristic for model selection. Based on our empirical findings, we propose the following heuristic. NODE is best suited for continuous systems operating within a single regime. rdSLDS is appropriate for systems assumed to exhibit piecewise-linear switching dynamics. rSNODE is best suited for complex systems with multiple overlapping nonlinear regimes where a single continuous vector field is insufficient. To assess the complexity and regime structure of an unknown system, we suggest the following diagnostic procedure: train rdSLDS with a generous latent dimensionality and $K \leq 8$ across multiple seeds. Consistent convergence suggests the system is suitable for piecewise-linear approximation, while widespread flatline or divergence failures suggest highly nonlinear dynamics that exceed the capacity of linear regime transitions. In the latter case, rSNODE should be trained across a range of K values, with $K = 1$ serving as a non-switching NODE baseline. While rdSLDS can in principle approximate any system given enough K, high-K segmentations of nonlinear systems were highly unstable over time and across seeds; a state segmentation that is consistent across seeds and values of K can be interpreted as evidence of meaningful underlying discrete structure, rather than an artifact of model capacity.

Extended future work. Beyond the directions noted in Section 5, future iterations should also validate these architectures on complex, continuous naturalistic paradigms, such as movie-watching fMRI, which offer a middle ground of shared cognitive processing across subjects without rigid block constraints, as well as on resting-state fMRI, where the primary objective is the pure unsupervised discovery of intrinsic brain networks. Extending these evaluations to neural data with significantly higher temporal resolution, such as EEG, MEG, or direct electrophysiological recordings, will be essential to test whether rSNODE and rdSLDS can scale to capture fast, sub-second continuous manifold trajectories and rapid state transitions.

B Methods

All models were implemented in Python 3.12.3 using PyTorch 2.10.0 (CUDA 12.8) and trained on a Linux workstation running Ubuntu 24.04 with an NVIDIA Tesla T4 GPU (16GB VRAM). ODE integration was performed using the `torchdiffeq` library (Chen et al., 2019). Random seeds were set globally across Python, NumPy, and PyTorch to ensure reproducibility. Code and a structured environment file (`requirements.txt`) will be made publicly available upon publication.

B.1 Neural ODE implementation

The latent NODE used for benchmarking was implemented following the standard latent NODE definition as described in Section 2.1. Parallel to the rdSLDS and rSNODE architectures described in Sections 3.2 and 3.3, a window of observations $\mathbf{y}_{0:T}$ is encoded into the mean and variance of the initial continuous latent state $\boldsymbol{\mu}_{q_0}, \boldsymbol{\sigma}_{q_0}$ from which the initial state $\mathbf{x}_0$ is sampled via the reparametrization trick. Starting from $\mathbf{x}_0$ the continuous latent trajectory is integrated to $\mathbf{x}_T$ using `odeint`. The full continuous latent trajectory $\mathbf{x}_{0:T}$ is decoded back to observation space to generate the reconstructed observations $\hat{\mathbf{y}}_{0:T}$. The latent NODE has no discrete latents.

B.2 Loss Function Details

The reconstruction term uses a Gaussian log-likelihood:

$$\log p_\theta(\mathbf{y}_t \mid \mathbf{x}_t) = -\frac{D}{2}\log(2\pi\sigma_y^2) - \frac{1}{2\sigma_y^2}\|\mathbf{y}_t - \hat{\mathbf{y}}_t\|^2 \tag{11}$$

where σ_y is a learnable observation noise parameter and $\hat{\mathbf{y}}_t = \text{Dec}_\theta(\mathbf{x}_t)$. The KL terms are computed analytically:

$$\text{KL}[q(\mathbf{x}_0)\|p(\mathbf{x}_0)] = \frac{1}{2}\sum_d \left(\sigma_{q,d}^2 + \mu_{q,d}^2 - 1 - \log\sigma_{q,d}^2\right) \tag{12}$$

$$\text{KL}[q(\mathbf{z}_0)\|p(\mathbf{z}_0)] = \sum_k q_k \log\frac{q_k}{1/K} \tag{13}$$

B.3 Training time and number of parameters

Table 1: Model complexity and training time on benchmark dynamical systems. Training times are reported for 1000 epochs on chaotic Lorenz with a dataset of size $[5, 5000, 3]$, a stride and window size of 50 and batch size 512 on a single NVIDIA Tesla T4 GPU (16GB VRAM). Parameter counts are reported for the default configuration used in benchmark experiments, see Table 2.

	NODE	**rdSLDS** ($K=2$)	**rdSLDS** ($K=4$)	**rSNODE** ($K=1$)	**rSNODE** ($K=4$)
Training time (min)	5	1	1	4	4
Time per epoch (s)	0.3	0.06	0.06	0.24	0.24
Encoder params	101,766	102,024	102,282	101,895	102,282
Dynamics params	4,611	538	820	4996	19,216
Decoder params	13	13	13	13	13
Total params	106,390	102,575	103,115	106,904	121,511

B.4 BENCHMARK HYPERPARAMETERS

Table 2: Training and model hyperparameters for results on benchmark systems.

Category	**Hyperparameter**	**NODE**	**rdSLDS**	**rSNODE**
Encoder / Decoder	Architecture	Bi-GRU	Bi-GRU	Bi-GRU
	Encoder layers	2	2	2
	Encoder hidden dim	64	64	64
	Decoder type	Linear	Linear	Linear
	Initial $\log \sigma_y$, (clamp)	$-4.0, [-5, -1]$	$-4.0, [-5, -1]$	$-4.0, [-5, -1]$
ODE	Hidden dim	64	–	64
	Layers	2	–	2
	Activation function	GELU	–	GELU
	Solver	RK4	–	RK4
	Tolerance (abs/rel)	$10^{-5}/10^{-3}$	–	$10^{-5}/10^{-3}$
Gate	Hidden dim	–	64	64
	Layers	–	1	1
	Activation function	–	Tanh	Tanh
Training	Epochs	1000	1000	1000
	Learning rate	10^{-3}	10^{-3}	10^{-3}
	Weight decay	10^{-5}	10^{-5}	10^{-5}
	Batch size	512	512	512
	KL anneal epochs	20–60%	20–60%	20–60%
	$\beta_{\max}$	10^{-3}	10^{-3}	10^{-3}
	Gumbel τ	–	$1.0 \to 0.1$	$1.0 \to 0.1$
	Window size	50	50	50
	Stride	50	50	50

B.5 TASK FMRI HYPERPARAMETERS

Table 3: Training and model hyperparameters for results on task fMRI data. For EMOTION in red and MOTOR in blue

Category	Hyperparameter	NODE	rdSLDS	rSNODE
Encoder / Decoder	Architecture	Bi-GRU	Bi-GRU	Bi-GRU
	Encoder layers	2	2	2
	Encoder hidden dim	256	256	256
	Decoder type	Linear	Linear	Linear
	Initial $\log \sigma_y$ (clamp)	$-4.0, [-5, -1]$	$-4.0, [-5, -1]$	$-4.0, [-5, -1]$
ODE	Hidden dim	256, 64	–	256, 64
	Layers	2, 1	–	2, 1
	Activation function	GELU	–	GELU
	Solver	RK4	–	RK4
	Tolerance (abs/rel)	$10^{-5}/10^{-3}$	–	$10^{-5}/10^{-3}$
Gate	Hidden dim	–	64	64
	Layers	–	1	1
	Activation function	–	Tanh	Tanh
Training	Epochs	500, 400	500, 400	500, 400
	Learning rate	$1.5 * 10^{-3}$, 10^{-3}	$1.5 * 10^{-3}$, 10^{-3}	$1.5 * 10^{-3}$, 10^{-3}
	Weight decay	10^{-4}	10^{-4}	10^{-4}
	Batch size	512, 128	512, 128	512, 128
	KL anneal epochs	20–60%	20–60%	20–60%
	$\beta_{\max}$	10^{-3}	10^{-3}	10^{-3}
	Gumbel τ	–	$1.0 \to 0.1$	$1.0 \to 0.1$
	Window size	50	50	50
	Stride	5, 10	5, 10	5, 10

B.6 DATA

B.6.1 CHAOTIC LORENZ BENCHMARK

The Lorenz system (Lorenz, 1963) is a three-dimensional continuous-time dynamical system governed by:

$$\dot{u} = \sigma(v - u), \quad \dot{v} = u(\rho - w) - v, \quad \dot{w} = uv - \beta w \tag{14}$$

with standard parameters $\sigma = 10$, $\rho = 28$, $\beta = 8/3$, which place the system in the chaotic regime with a positive maximal Lyapunov exponent of $\lambda \approx 0.9$. Five trajectories are generated using a fourth-order Runge-Kutta (RK4) integrator with timestep $\Delta t = 0.01$ and length $T = 5000$ steps, corresponding to 50 time units. Initial conditions are sampled with small random perturbations around the reference point $(8.0, -19.0, 30.0)^\top$, ensuring distinct trajectories that nonetheless explore the same attractor. The dataset is split into four training trajectories and one validation trajectory. Models receive an observation window of $T_w = 50$ timesteps and are evaluated on generative rollouts of $T = 7500$ steps initialized from this window. The three-dimensional observations are used directly as both input and reconstruction target, with no additional noise or observation transform applied.

B.6.2 NOISY LORENZ BENCHMARK

The noisy Lorenz benchmark data was generated identically to the chaotic Lorenz data described in B.6.1. Additionally, after generating the smooth trajectories, Gaussian observation noise of $\sigma_{obs} = 0.5$ was added to the trajectories resulting in strong perturbations. Models were trained on the noisy data.

B.6.3 NASCAR BENCHMARK

The NASCAR benchmark (adapted from (Linderman et al., 2017)) consists of two-dimensional latent dynamics that switch between four distinct linear regimes arranged around an oval track. The four regimes correspond to: two straight segments (top and bottom) and two semicircular turns (left and right), each occupying one quarter of the cycle. The latent trajectory traces the oval continuously, switching regime at each quarter boundary. Trajectories are generated by tiling the base cycle over $T = 5000$ timesteps with $\Delta t = 0.1$, corresponding to 10 full laps of length $T_c = 500$. Gaussian noise with $\sigma_{\text{lat}} = 0.1$ is added to the latent trajectory and smoothed with a Gaussian filter ($\sigma_{\text{smooth}} = 5$) to produce continuous, realistic dynamics. Five runs are generated with distinct random starting offsets along the track, sampled uniformly from $\{0, \ldots, 499\}$, ensuring that each run begins at a different point in the cycle. The dataset is split into four training trajectories and one validation trajectory. The two-dimensional latent trajectory $\mathbf{x}_t \in \mathbb{R}^2$ is projected to a $D_{\text{obs}} = 10$-dimensional observation space via a fixed random linear map $W \in \mathbb{R}^{10\times 2}$, shared across all runs (fixed seed), with additive Gaussian observation noise $\sigma_{\text{obs}} = 0.5$:

$$\mathbf{y}_t = W\mathbf{x}_t + \boldsymbol{\epsilon}_t, \qquad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \sigma_{\text{obs}}^2 \mathbf{I}) \tag{15}$$

Models observe only $\mathbf{y}_t$ and must recover both the latent dynamics $\mathbf{x}_t$ and the ground truth regime assignments $z_t \in \{1, 2, 3, 4\}$.

B.6.4 SWITCHING LORENZ BENCHMARK

The switching Lorenz system is a novel benchmark introduced in this work to specifically evaluate models that combine nonlinear within-regime dynamics with discrete switching. It consists of two coupled standard Lorenz attractors governed by the equations in Section 4.1.1 with parameters $\sigma = 10$, $\rho = 28$, $\beta = 8/3$, one of which is transformed via a fixed rotation and translation. The system maintains a single hidden trajectory $\mathbf{u}_t \in \mathbb{R}^3$ that always evolves according to standard Lorenz dynamics, and a discrete regime variable $z_t \in \{0, 1\}$ that determines how the hidden state is mapped to the observed trajectory $\mathbf{x}_t$. In regime $z_t = 0$, the observation directly mirrors the hidden state $\mathbf{x}_t = \mathbf{u}_t$. In regime $z_t = 1$, a fixed rotation matrix R and translation vector $\mathbf{T}$ are applied:

$$\mathbf{x}_t = R\,\mathbf{u}_t + \mathbf{T}, \qquad R = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad \mathbf{T} = (-45, 0, 0)^\top \tag{16}$$

producing a second attractor that is rotated and translated relative to the first. Regime switching is triggered geometrically: the trajectory switches from regime 0 to 1 when it enters a sphere of radius $r = 6$ centred at $\mathbf{p}_{0\to 1} = (-15, -15, 20)^\top$ in observation space, and from regime 1 to 0 when it enters a sphere of radius $r = 6$ centred at $\mathbf{p}_{1\to 0} = (-10, 15, 5)^\top$ (see Figure 6). At each switching event, the hidden state is transformed to the coordinate frame of the new regime to ensure continuity of the underlying dynamics. The resulting observed trajectory produces a three-wing attractor in which the two Lorenz attractors share a common crossover region, with ground truth regime labels available for evaluation. Trajectories are generated using RK4 integration with $\Delta t = 0.01$ for $T = 5000$ steps, yielding 5 runs with different initial conditions for training and validation.

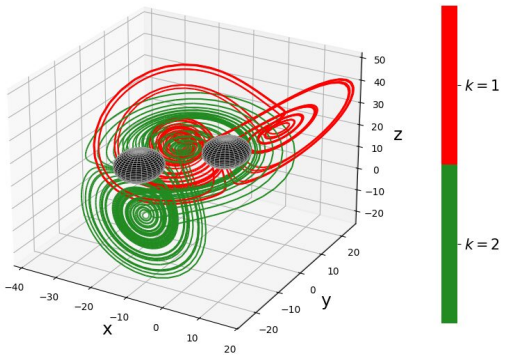


Figure 6: 3-wing attractor of the 3D switching Lorenz benchmark system with switching spheres centered at $\mathbf{p}_{0\to1} = (-15, -15, 20)^\top$ and $\mathbf{p}_{1\to0} = (-10, 15, 5)^\top$.

B.6.5 HCP TASK FMRI DATA

Dataset. We use task fMRI data from the Human Connectome Project (HCP) (Van Essen et al., 2012), publicly available via the BALSA neuroimaging repository (https://balsa.wustl.edu/). Data from the first 50 subjects of the HCP 100 Unrelated Subjects release are used, across two tasks: EMOTION and MOTOR. Each subject contributes two acquisitions corresponding to opposing phase-encoding directions (LR and RL), yielding 100 datasets per task. All data were obtained already preprocessed using the HCP minimal preprocessing pipeline (Glasser et al., 2013), which includes gradient distortion correction, motion correction, and registration to MNI space.

Parcellation. Cortical BOLD timeseries were extracted using the Glasser multi-modal parcellation atlas (Glasser et al., 2016), comprising 360 cortical regions of interest (ROIs). Parcellation was performed using the Connectome Workbench `wb_command -cifti-parcellate` tool, which computes the mean timeseries within each parcel from the dense CIFTI dtseries files.

Preprocessing. Following parcellation, each ROI timeseries was independently z-score normalized across time:

$$\tilde{y}_{t,d} = \frac{y_{t,d} - \mu_d}{\sigma_d + \epsilon} \tag{17}$$

where μ_d and σ_d are the mean and standard deviation of ROI d across all timepoints, and $\epsilon = 10^{-8}$ prevents division by zero. Normalization is applied independently per subject and encoding direction. The resulting data are stored as tensors of shape $[N \times T \times D]$ where $N = 100$ (50 subjects $\times$ 2 directions), T is the number of TRs per task (EMOTION: $T = 176$, MOTOR: $T = 284$), and $D = 360$ ROIs.

Tasks. The EMOTION task consists of two task blocks, *fear* and *neut*, alternated in a blocked design, with short *rest* intervals in between, arranged as depicted in Figure 7. In the *neut* block, participants were asked to match one of two presented shapes to a reference shape; in the *fear* block, participants were asked to match fearful or angry faces to a reference face. For more details on the task, see Barch et al. (2013). The MOTOR task consists of seven task blocks, arranged in a blocked design. In each encoding direction, each task block is repeated twice. Motion task blocks are preceded by short *cue* blocks, which signal the participant which body part to move in the following block. The motion blocks are: left hand (*lh*), right hand (*rh*), left foot (*lf*), right foot (*rf*), and tongue (*t*). Further, each dataset contains two *rest* blocks. The ordering of the blocks differs between LR and RL encoding directions (see Figure 8). For more details on the task, see Barch et al. (2013).

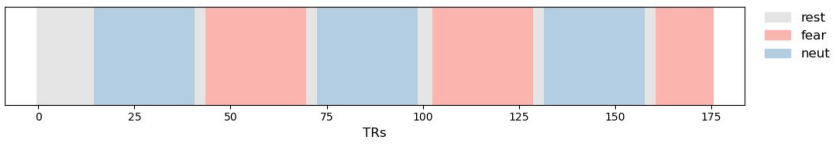


Figure 7: EMOTION task blocks, alternating presentation of shapes in the *neut* block, and fearful faces in the *fear* block.

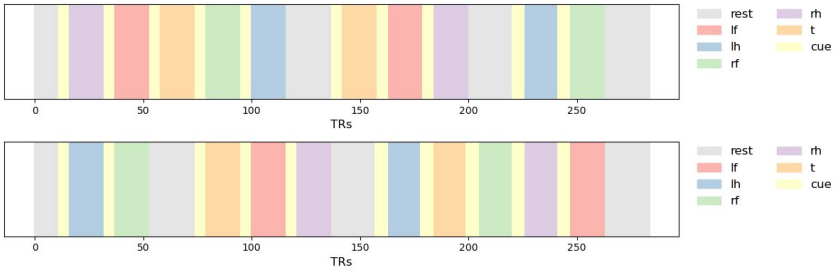


Figure 8: MOTOR task blocks, top LR encoding, bottom RL encoding direction. *Cue* blocks signal the motion participants should perform in the following block, distinguishing left hand (*lh*), right hand (*rh*), left foot (*lf*), right foot (*rf*), and tongue (*t*).

Training setup. For model training, each subject's timeseries is segmented into overlapping windows of length $T_w = 50$ TRs with stride $s = 5$ for EMOTION and $s = 10$ for MOTOR, independently per subject and encoding direction to prevent cross-subject data leakage. The dataset is split into 80 training datasets (40 subjects $\times$ 2 directions) and 20 validation datasets (10 subjects $\times$ 2 directions). For the EMOTION task, the two encoding directions share identical task block structure; for the MOTOR task, block ordering differs between LR and RL. For both tasks, the two encoding directions are treated as separate datasets.

Experimental design vectors. Task condition onset times and durations were obtained from the HCP event files (EVs) and converted to binary TR-level indicator vectors using the recorded onset and duration of each condition block, with TR = 0.72s. The resulting condition vectors are used solely for post-hoc evaluation of state-condition alignment and are never provided to the model during training.

B.7 Experiment Methods

Model reconstructions are evaluated using four complementary metrics. The primary evaluation metric is the Maximum Mean Discrepancy (MMD), which measures the distance between the distributions of true and generated trajectories and is therefore robust to the phase offsets inherent in chaotic systems where exact pointwise agreement is not expected. The Mean Squared Error (MSE) is reported as a secondary metric:

$$\text{MSE} = \frac{1}{T \cdot D} \sum_{t=1}^{T} \sum_{d=1}^{D} \left(\mathbf{y}_{t,d} - \hat{\mathbf{y}}_{t,d}\right)^2 \tag{18}$$

While MSE provides a simple pointwise reconstruction measure, it is less informative for chaotic systems where trajectories diverge exponentially despite following the same attractor geometry. We additionally report the Maximal Lyapunov Exponent (MLE) for chaotic systems to assess whether reconstructed trajectories preserve the chaotic character of the true system, and the Power Spectrum Correlation (PSC) across all benchmark systems to evaluate whether the temporal frequency structure of the dynamics is captured. Each metric is described in detail in Appendices B.7.1, B.7.2, and B.7.3.

For the analyses reported in the main text, metrics were averaged over the 5 runs in each dataset for each seed. The results reported in Table 4 are average and standard deviation averaged across seeds.

B.7.1 Maximum Mean Discrepancy

Model reconstruction quality is evaluated using the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012), a kernel-based statistical test that measures the distance between two distributions by comparing their embeddings in a reproducing kernel Hilbert space (RKHS). Given samples $\mathbf{x}_{1:N} \sim p$ and $\mathbf{y}_{1:M} \sim q$, the unbiased MMD^2 estimator is:

$$\widehat{\text{MMD}}^2(p,q) = \frac{1}{N(N-1)} \sum_{i \neq i'} k(\mathbf{x}_i, \mathbf{x}_{i'}) + \frac{1}{M(M-1)} \sum_{j \neq j'} k(\mathbf{y}_j, \mathbf{y}_{j'}) - \frac{2}{NM} \sum_{i,j} k(\mathbf{x}_i, \mathbf{y}_j) \quad (19)$$

where $k(\cdot,\cdot)$ is a Gaussian RBF kernel. We use a multi-scale kernel with three bandwidths $\sigma \in \{0.5\hat{\sigma}, \hat{\sigma}, 2\hat{\sigma}\}$, where $\hat{\sigma} = \sqrt{\tilde{d}/2}$ is the median heuristic bandwidth estimated from the median pairwise squared distance $\tilde{d}$ of the combined sample. Averaging over multiple bandwidths improves robustness to the scale of the distribution (Gretton et al., 2012). MMD is computed between the true trajectory and the model's generative rollout of $T = 5000$ timesteps, initialized from a single encoder-inferred window of $T_w = 50$ timesteps. Since MMD operates on the full multivariate trajectory, it captures differences in both marginal distributions and temporal correlations between true and generated dynamics. The unbiased estimator can take small negative values due to sampling noise; these are clamped to zero.

B.7.2 Maximal Lyapunov Exponent Estimation

The maximal Lyapunov exponent (MLE) is estimated directly from multivariate trajectories using a variant of the Rosenstein et al. (1993) method, without time-delay embedding or dimensionality reduction. Given a trajectory $\mathbf{x}_{1:T} \in \mathbb{R}^{T \times d}$, for each timepoint i the nearest spatial neighbor $j^*(i)$ is identified using a kd-tree, subject to a Theiler window requiring a minimum temporal separation of $|i - j^*(i)| > w$ to exclude temporally correlated points that are not true independent realizations of nearby initial conditions. The divergence of each neighboring pair is tracked forward in time:

$$\Delta(i,k) = \log \frac{\|\mathbf{x}_{i+k} - \mathbf{x}_{j^*(i)+k}\|}{\|\mathbf{x}_i - \mathbf{x}_{j^*(i)}\|}, \quad k = 1, \ldots, L \quad (20)$$

The mean divergence across all valid pairs is then computed, and the MLE is estimated as the slope of a linear fit to the mean divergence curve over a fixed interval $[k_{\text{start}}, k_{\text{end}}]$ that excludes the initial alignment phase:

$$\hat{\lambda} = \frac{1}{\Delta t} \cdot \text{slope}\big(\overline{\Delta}(\cdot,k),\ k \in [k_{\text{start}}, k_{\text{end}}]\big) \quad (21)$$

where Δt is the integration timestep. In all experiments we use a Theiler window of $w = 100$ steps, trajectory length $L = 100$ steps, and fit the slope over $k \in [60, 100]$ to focus on the asymptotic linear divergence regime. The MLE is estimated separately for the true system trajectory and each model's generative rollout, and a positive MLE confirms that the reconstructed dynamics are chaotic.

B.7.3 Power Spectrum Correlation

The power spectrum correlation (PSC) measures the degree to which a model's generative rollout preserves the frequency content of the true system. For each dimension $d \in \{1, \ldots, D\}$, the power spectral density (PSD) is estimated using Welch's method (Welch, 1967), which provides a robust estimate by averaging over overlapping periodograms. The Pearson correlation between the log-transformed PSDs of the true and reconstructed signals is then computed:

$$\text{PSC}_d = \text{corr}\Big(\log(1 + \hat{S}_d^{\text{true}}(f)),\ \log(1 + \hat{S}_d^{\text{recon}}(f))\Big) \quad (22)$$

where $\hat{S}_d(f)$ denotes the estimated PSD at frequency f for dimension d. The log transform is applied to compress the dynamic range of the power spectrum, which spans several orders of magnitude, improving the sensitivity of the correlation to differences across the full frequency range. The final PSC score is the mean correlation across all D dimensions:

$$\text{PSC} = \frac{1}{D} \sum_{d=1}^{D} \text{PSC}_d \quad (23)$$

A PSC of 1 indicates a perfect match in frequency content; a PSC of 0 indicates no correlation. Unlike MSE, PSC is insensitive to phase offsets and evaluates whether the model captures the temporal structure of the dynamics rather than exact pointwise agreement.

B.7.4 STATISTICAL TESTING

Statistical comparisons between models are performed using the Wilcoxon rank-sum test (equivalent to the Mann-Whitney U test) (Mann & Whitney, 1947), a non-parametric test for the null hypothesis that two independent samples are drawn from the same distribution. The test makes no assumptions about the normality of the underlying distributions, making it appropriate for MMD values over $n = 15$ seeds where normality cannot be assumed. For each benchmark system, pairwise comparisons are performed between all model configurations of interest; reported p-values are Bonferroni-corrected for the number of comparisons per system to control the family-wise error rate. All tests are two-sided with significance threshold $\alpha = 0.05$. Significance in the boxplots is evaluated based on the Bonferroni corrected α for 6 tests, $\alpha = 0.0083$.

B.7.5 TASK FMRI ANALYSES

To investigate the alignment between the rSNODE generated states and the task blocks, we computed their maximum cross-correlation across lags from -10 to 10, to account for small misalignment in the generated states. Reported is the cross-correlation at the best lag. The glass brain plots were generated by correlating the generated time-series (shifted by the best lag) for each ROI with the task blocks, and plotting the correlation value for each ROI on the brain. For reference, the glass brain with correlations between the bandpass-filtered task fMRI data and task blocks per ROI are provided (with zero lag). The alignment between the ROI task activations for generated and true data is visualized with a scatterplot, plotting the per-ROI trajectory-task correlation for the model generated data against those of the true data.

B.7.6 CONVERGENCE

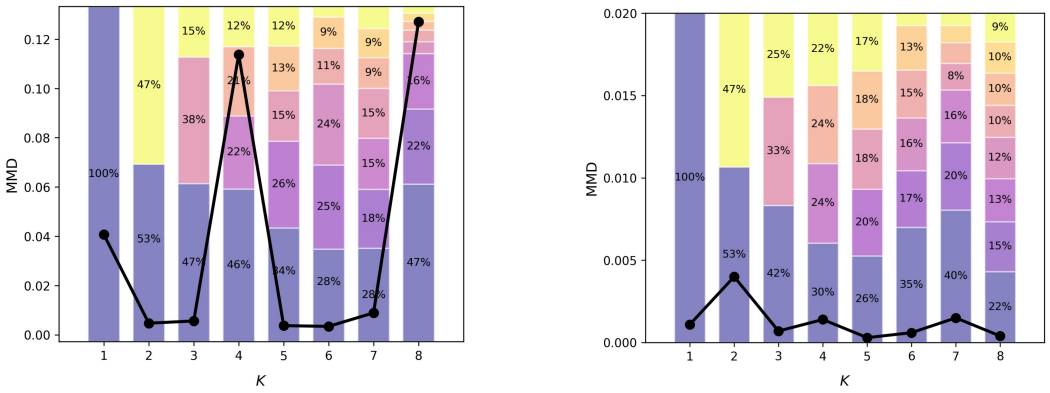
A run is counted as converged if its generative rollout remains numerically finite, i.e. contains no NaN or infinite values; diverging runs are excluded from all metric aggregation. The criterion is applied identically across models and values of K and involves no threshold on reconstruction quality, so it cannot favor any particular model. The number of converged seeds N is reported per condition in the corresponding figure. Because only failures are discarded, aggregates are computed over each model's successful runs, making comparisons conservative with respect to the models that fail most often.

C BENCHMARK RESULTS

Table 4: Mean and standard deviation for Maximum Mean Discrepancy (MMD), Mean Squared Error (MSE), Maximal Lyapunov Exponent (MLE), and Power Spectrum Correlation (PSC) across seeds and runs; reported for NODE, rdSLDS ($K \in \{2, 4\}$; the large mean and standard deviation for MSE of rdSLDS $K = 2$ on the switching Lorenz is due to a single exploding outlier), and rSNODE ($K \in \{1, 2\}$) for the benchmark datasets. MLE is only reported for the chaotic systems. The hyperparameters used for training the models are reported in Table 2.

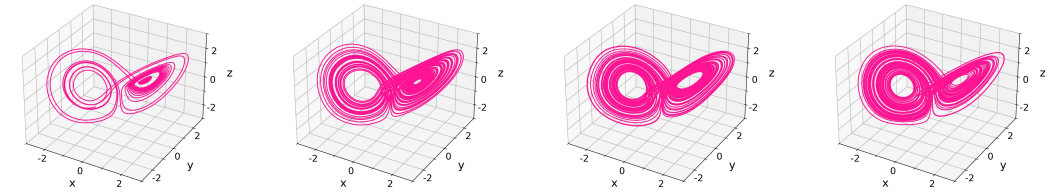
Dataset	**Metric**	**NODE**	**rdSLDS** ($K = 2$)	**rdSLDS** ($K = 4$)	**rSNODE** ($K = 1$)	**rSNODE** ($K = 2$)
Chaotic Lorenz	MMD ↓	0.072 ± 0.079	0.046 ± 0.044	0.271 ± 0.186	0.017 ± 0.026	0.078 ± 0.148
	MSE ↓	1.879 ± 0.109	1.976 ± 0.172	1.730 ± 0.165	1.876 ± 0.074	1.895 ± 0.106
	MLE	0.841 ± 0.213	0.967 ± 0.526	0.883 ± 0.500	0.934 ± 0.074	0.945 ± 0.185
	PSC ↑	0.980 ± 0.033	0.965 ± 0.024	0.887 ± 0.076	0.997 ± 0.002	0.972 ± 0.045
Noisy Lorenz	MMD ↓	0.043 ± 0.062	0.068 ± 0.131	0.247 ± 0.152	0.021 ± 0.024	0.049 ± 0.136
	MSE ↓	2.218 ± 0.098	2.295 ± 0.243	2.004 ± 0.168	2.205 ± 0.083	2.129 ± 0.156
	MLE	0.800 ± 0.236	1.335 ± 0.715	0.887 ± 0.503	0.929 ± 0.097	0.983 ± 0.278
	PSC ↑	0.981 ± 0.023	0.964 ± 0.033	0.893 ± 0.077	0.992 ± 0.003	0.985 ± 0.029
Switching Lorenz	MMD ↓	0.137 ± 0.156	0.379 ± 0.138	0.209 ± 0.111	0.155 ± 0.142	0.038 ± 0.033
	MSE ↓	1.934 ± 0.200	28.566 ± 103.760	1.980 ± 0.185	2.072 ± 0.162	1.943 ± 0.082
	MLE	0.976 ± 0.579	0.718 ± 0.702	1.170 ± 0.411	0.754 ± 0.585	1.553 ± 0.152
	PSC ↑	0.957 ± 0.053	0.832 ± 0.079	0.905 ± 0.036	0.956 ± 0.053	0.991 ± 0.002
NASCAR	MMD ↓	0.003 ± 0.0009	0.004 ± 0.002	0.003 ± 0.002	0.002 ± 0.001	0.002 ± 0.0001
	MSE ↓	6.276 ± 3.846	3.155 ± 3.036	0.511 ± 0.299	0.747 ± 0.693	0.623 ± 0.401
	PSC ↑	0.990 ± 0.000	0.990 ± 0.0002	0.991 ± 0.0001	0.991 ± 0.0001	0.991 ± 0.0001

C.1 CHAOTIC LORENZ

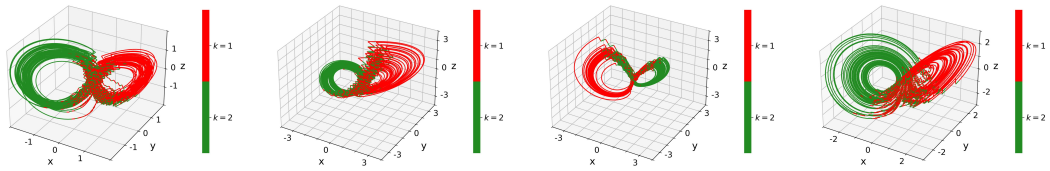


(a) rdSLDS (b) rSNODE

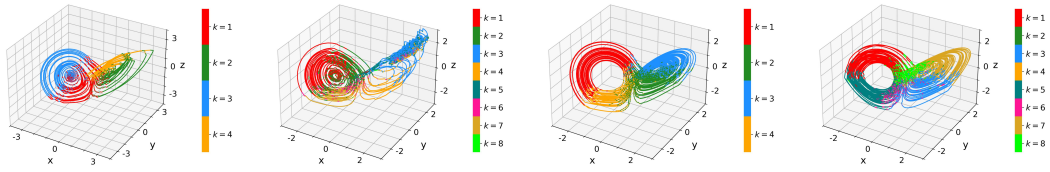
Figure 9: K ablation for $K \in \{1, \ldots, 8\}$ on chaotic Lorenz for a) rdSLDS and b) rSNODE.



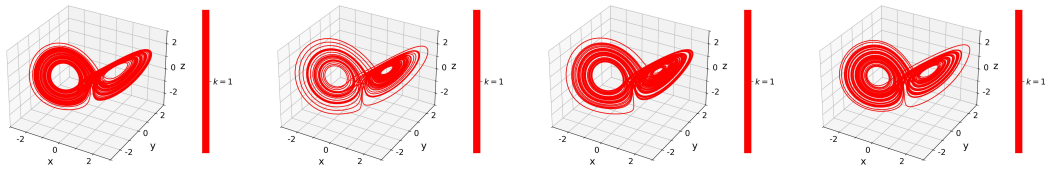
(a) NODE, Seed = 24 (b) NODE, Seed = 42 (c) NODE, Seed = 60 (d) NODE, Seed = 90



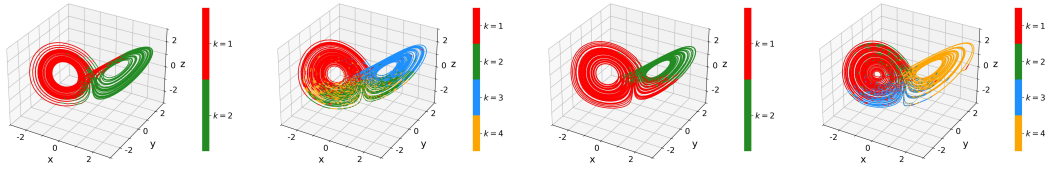
(e) rdSLDS, Seed = 24 (f) rdSLDS, Seed = 42 (g) rdSLDS, Seed = 60 (h) rdSLDS, Seed = 90



(i) rdSLDS, Seed = 24 (j) rdSLDS, Seed = 24 (k) rdSLDS, Seed = 42 (l) rdSLDS, Seed = 42



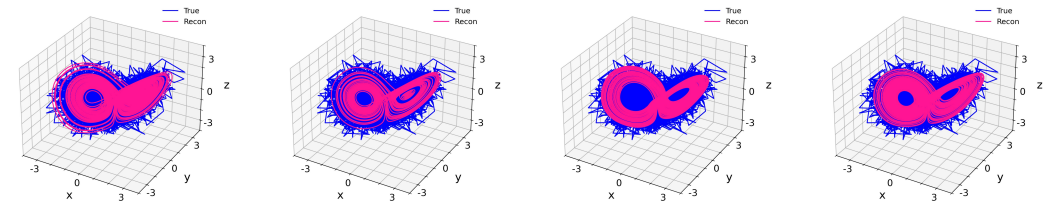
(m) rSNODE, Seed = 24 (n) rSNODE, Seed = 42 (o) rSNODE, Seed = 60 (p) rSNODE, Seed = 90



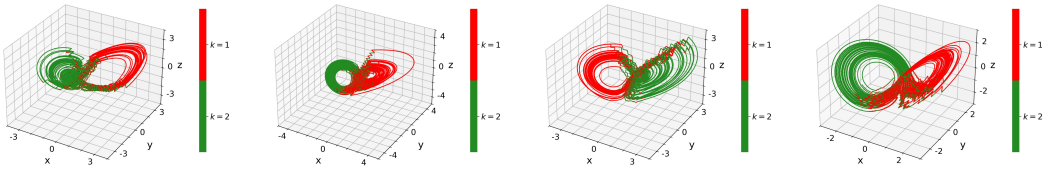
(q) rSNODE, Seed = 42 (r) rSNODE, Seed = 24 (s) rSNODE, Seed = 42 (t) rSNODE, Seed = 42

Figure 10: 3D reconstructions of the chaotic Lorenz system across models, seeds, and K.

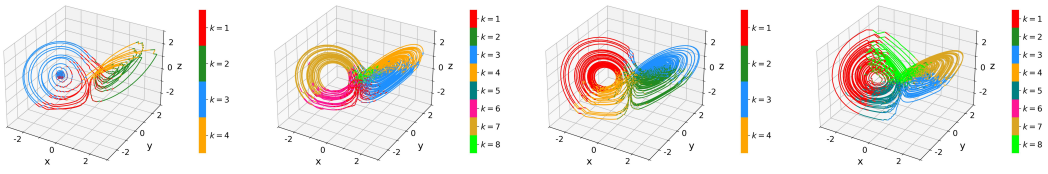
C.2 NOISY CHAOTIC LORENZ



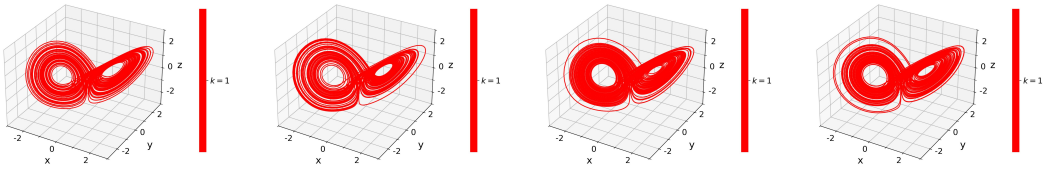
(a) NODE, Seed = 24 (b) NODE, Seed = 42 (c) NODE, Seed = 60 (d) NODE, Seed = 90



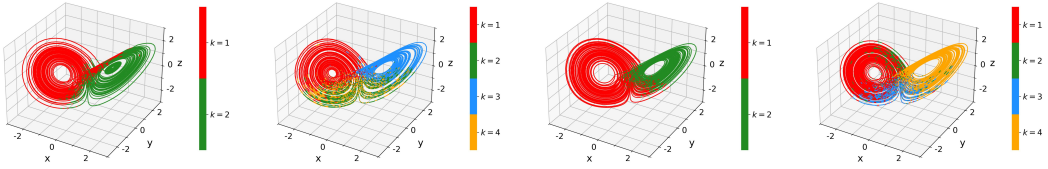
(e) rdSLDS, Seed = 24 (f) rdSLDS, Seed = 42 (g) rdSLDS, Seed = 60 (h) rdSLDS, Seed = 90



(i) rdSLDS, Seed = 24 (j) rdSLDS, Seed = 60 (k) rdSLDS, Seed = 42 (l) rdSLDS, Seed = 42



(m) rSNODE, Seed = 24 (n) rSNODE, Seed = 42 (o) rSNODE, Seed = 60 (p) rSNODE, Seed = 90



(q) rSNODE, Seed = 42 (r) rSNODE, Seed = 24 (s) rSNODE, Seed = 42 (t) rSNODE, Seed = 42

Figure 11: 3D reconstructions of the noisy chaotic Lorenz system with $\sigma_{obs} = 0.5$ observation noise across models, seeds, and K. All models recover the clean Lorenz attractor, with results parallel to those on the chaotic Lorenz system without noise.

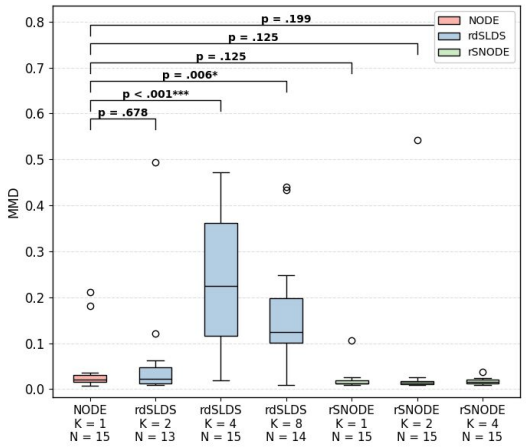


Figure 12: MMD distributions across 15 seeds for the noisy chaotic Lorenz system. The nonlinear NODE and rSNODE models consistently achieve the lowest MMD. Results are highly similar to those for the chaotic Lorenz system without noise shown in Figure 2.

C.3 NASCAR

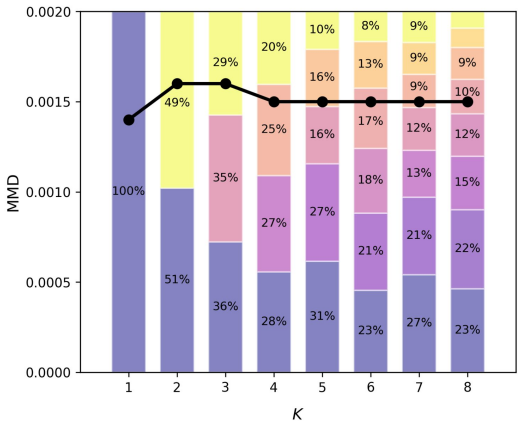
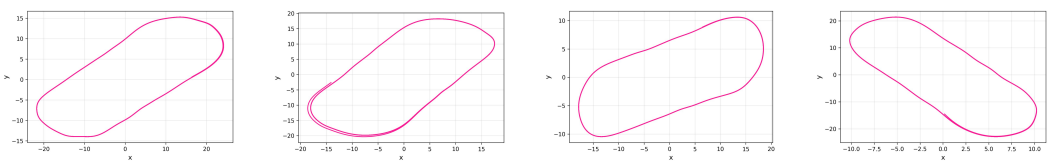
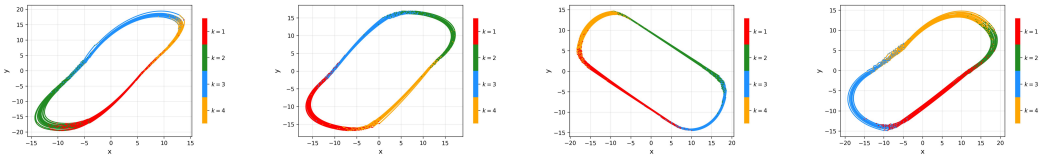


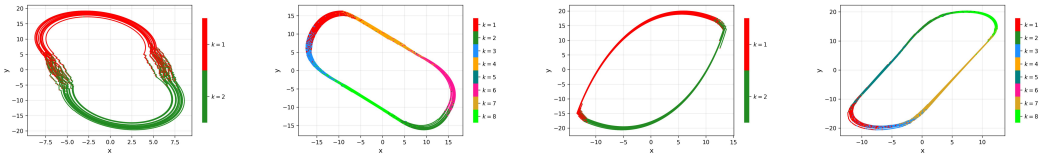
Figure 13: K ablation for $K \in \{1, \ldots, 8\}$ on NASCAR for rSNODE.



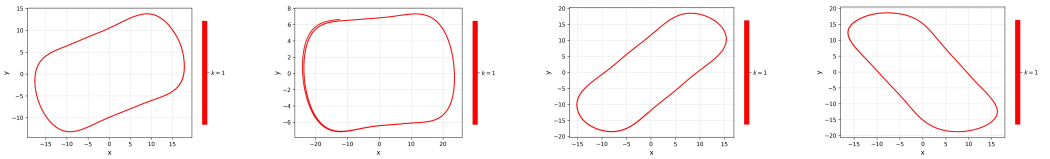
(a) NODE, Seed = 24 (b) NODE, Seed = 42 (c) NODE, Seed = 60 (d) NODE, Seed = 90



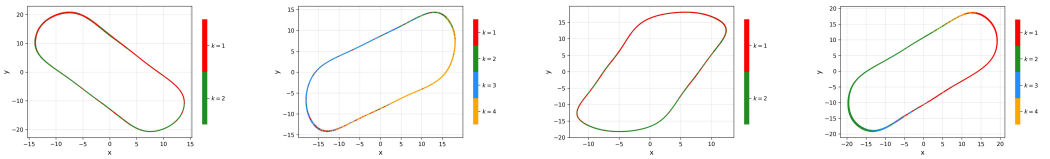
(e) rdSLDS, Seed = 24 (f) rdSLDS, Seed = 42 (g) rdSLDS, Seed = 60 (h) rdSLDS, Seed = 90



(i) rdSLDS, Seed = 24 (j) rdSLDS, Seed = 24 (k) rdSLDS, Seed = 42 (l) rdSLDS, Seed = 42



(m) rSNODE, Seed = 24 (n) rSNODE, Seed = 42 (o) rSNODE, Seed = 60 (p) rSNODE, Seed = 90



(q) rSNODE, Seed = 42 (r) rSNODE, Seed = 24 (s) rSNODE, Seed = 42 (t) rSNODE, Seed = 42

Figure 14: 2D reconstructions of the NASCAR system across models, seeds, and K.

C.4 SWITCHING LORENZ

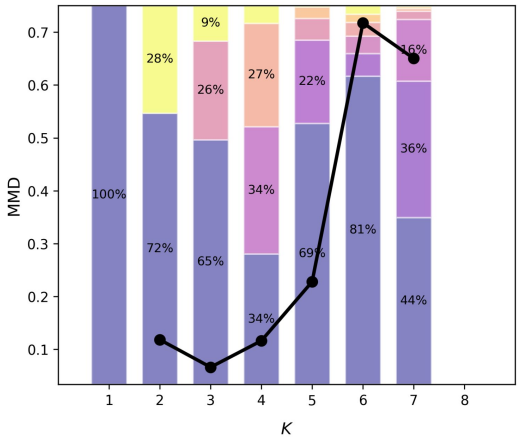
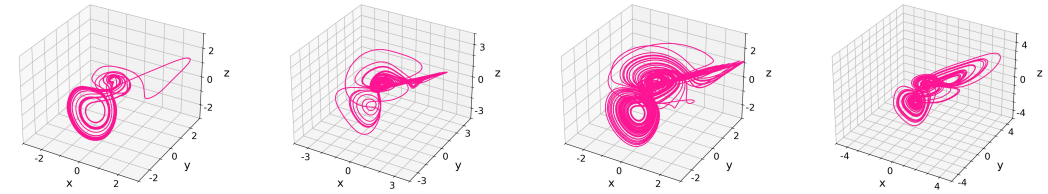
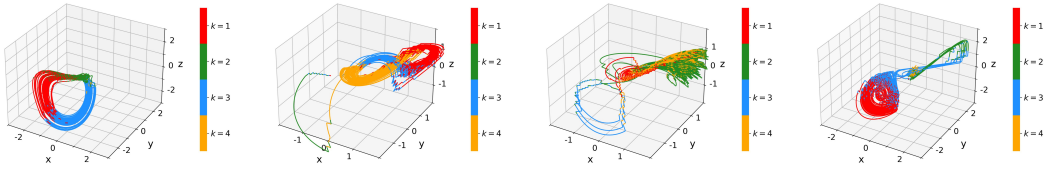


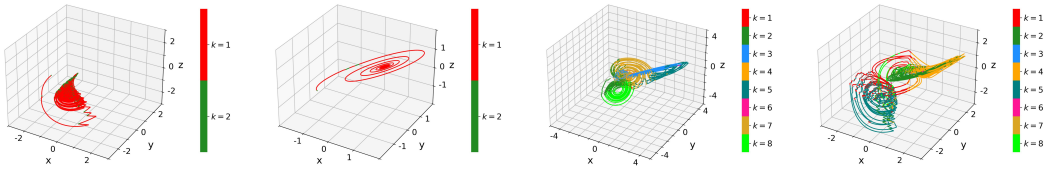
Figure 15: K ablation for $K \in \{1, \ldots, 8\}$ on switching Lorenz system for rdSLDS across 3 seeds. For $K = 1$ and $K = 8$ none of the 3 seeds converged.



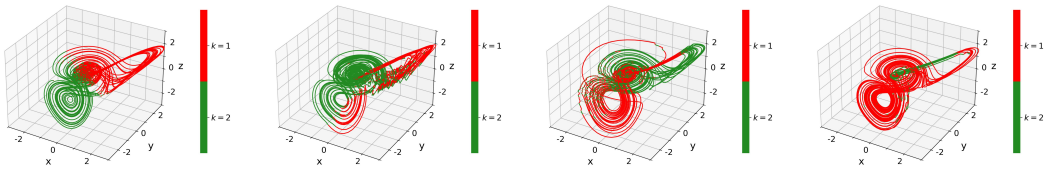
(a) NODE, Seed = 24 (b) NODE, Seed = 42 (c) NODE, Seed = 60 (d) NODE, Seed = 90



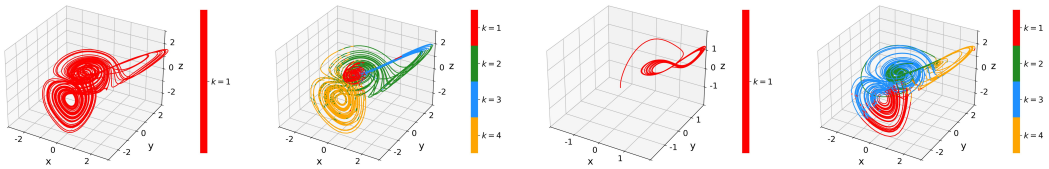
(e) rdSLDS, Seed = 24 (f) rdSLDS, Seed = 42 (g) rdSLDS, Seed = 90 (h) rdSLDS, Seed = 276



(i) rdSLDS, Seed = 24 (j) rdSLDS, Seed = 42 (k) rdSLDS, Seed = 101 (l) rdSLDS, Seed = 397



(m) rSNODE, Seed = 24 (n) rSNODE, Seed = 42 (o) rSNODE, Seed = 60 (p) rSNODE, Seed = 90



(q) rSNODE, Seed = 42 (r) rSNODE, Seed = 24 (s) rSNODE, Seed = 42 (t) rSNODE, Seed = 42

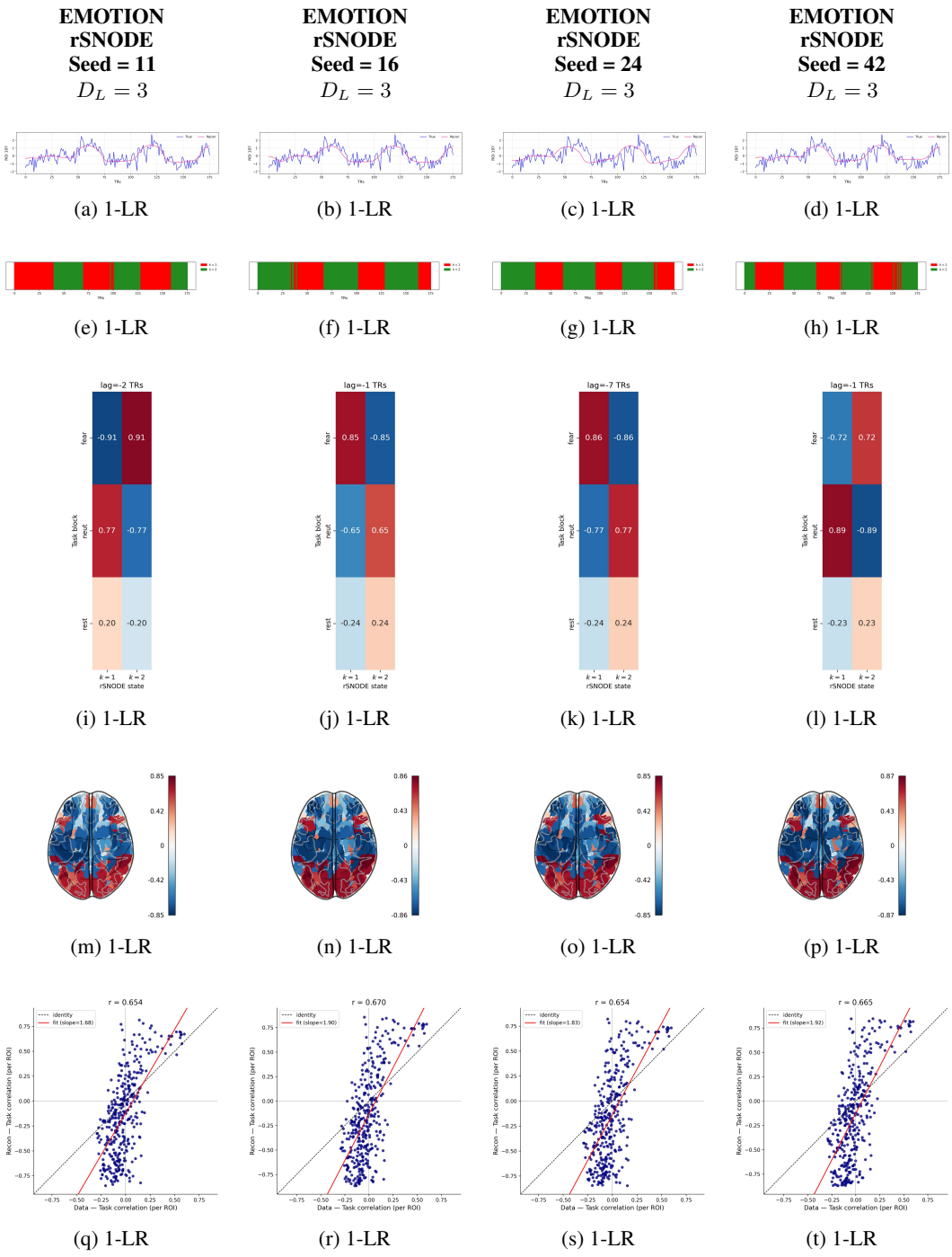
Figure 16: 3D reconstructions of the switching Lorenz system across models, seeds, and K. Illustrative reconstructions; where the default seed diverged, another converged seed is shown. Quantitative results (Figures 2–4) use the same fixed 15 seeds for all models.

D TASK FMRI RESULTS

Table 5: Task fMRI reconstruction metrics for NODE, rdSLDS and rSNODE for EMOTION and MOTOR tasks. Reported are the mean and standard deviation of Maximum Mean Discrepancy (MMD), Mean Squared Error (MSE), and Power Spectrum Correlation (PSC) averaged across seeds (5 seeds for rdSLDS and rSNODE, 3 seeds for NODE). The hyperparameters used for training the models are reported in Table 3.

Dataset	**Metric**	**NODE** ($D_L = 3$), ($D_L = 18$)	**rdSLDS** ($D_L = 6$), ($D_L = 36$)	**rSNODE** ($D_L = 3$), ($D_L = 18$)
EMOTION, $K = 2$	MMD ↓	0.319 ± 0.0003	0.284 ± 0.011	0.316 ± 0.007
	MSE ↓	0.956 ± 0.002	0.968 ± 0.021	0.957 ± 0.012
	PSC ↑	0.581 ± 0.018	0.622 ± 0.009	0.591 ± 0.013
MOTOR, $K = 3$	MMD ↓	0.144 ± 0.028	0.271 ± 0.112	0.249 ± 0.016
	MSE ↓	1.248 ± 0.003	1.010 ± 0.028	1.007 ± 0.011
	PSC ↑	0.683 ± 0.011	0.565 ± 0.236	0.659 ± 0.009

D.1 rSNODE EMOTION RESULTS ACROSS SEEDS AND SUBJECTS



(a) 1-LR (b) 1-LR (c) 1-LR (d) 1-LR

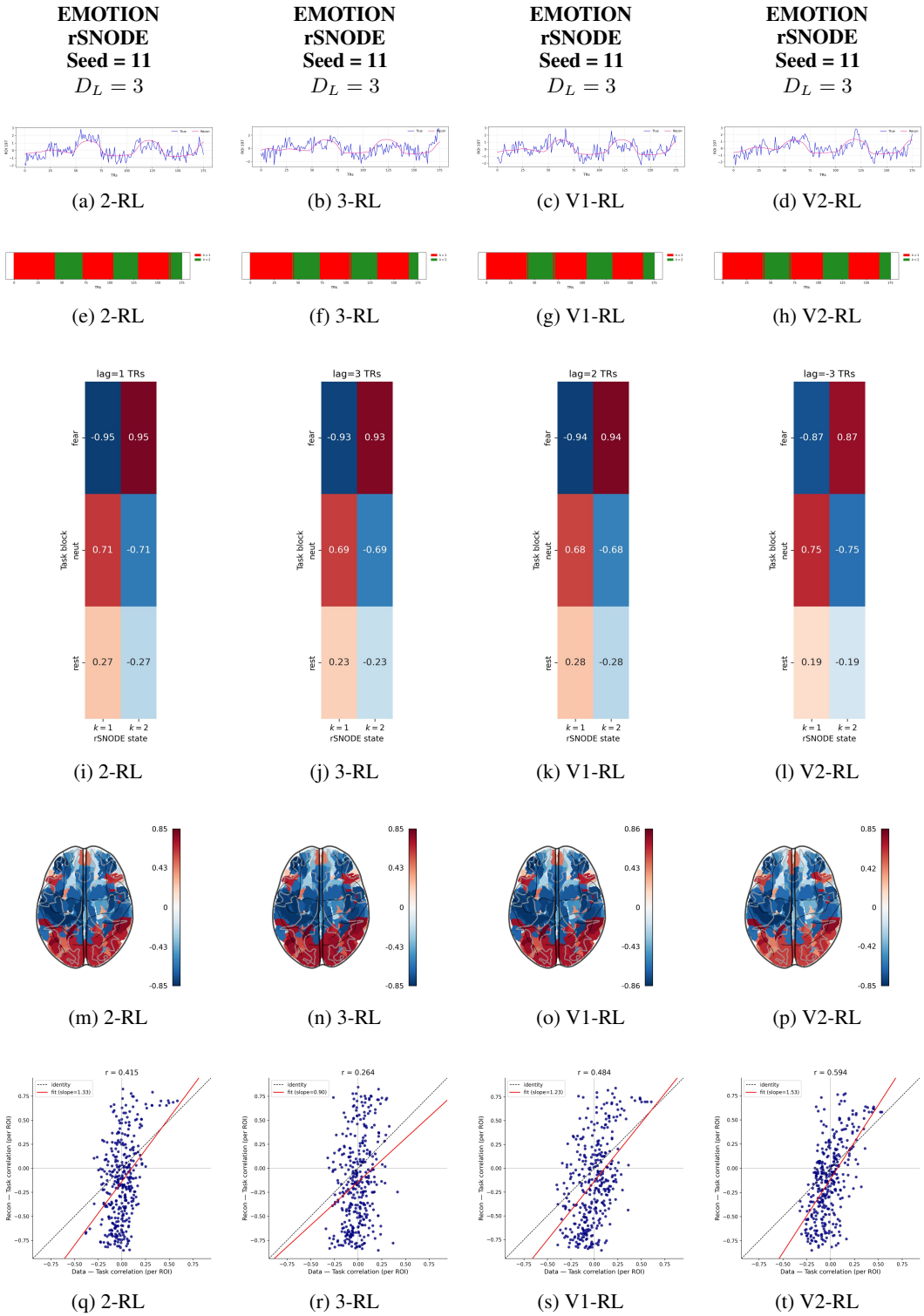
(e) 1-LR (f) 1-LR (g) 1-LR (h) 1-LR

(i) 1-LR (j) 1-LR (k) 1-LR (l) 1-LR

(m) 1-LR (n) 1-LR (o) 1-LR (p) 1-LR

(q) 1-LR (r) 1-LR (s) 1-LR (t) 1-LR

Figure 17: Reconstructed trajectories, generated states, state-task correlations, ROI activations, and ROI activation correlations for the EMOTION task for rSNODE across seeds.



(a) 2-RL (b) 3-RL (c) V1-RL (d) V2-RL

(e) 2-RL (f) 3-RL (g) V1-RL (h) V2-RL

(i) 2-RL (j) 3-RL (k) V1-RL (l) V2-RL

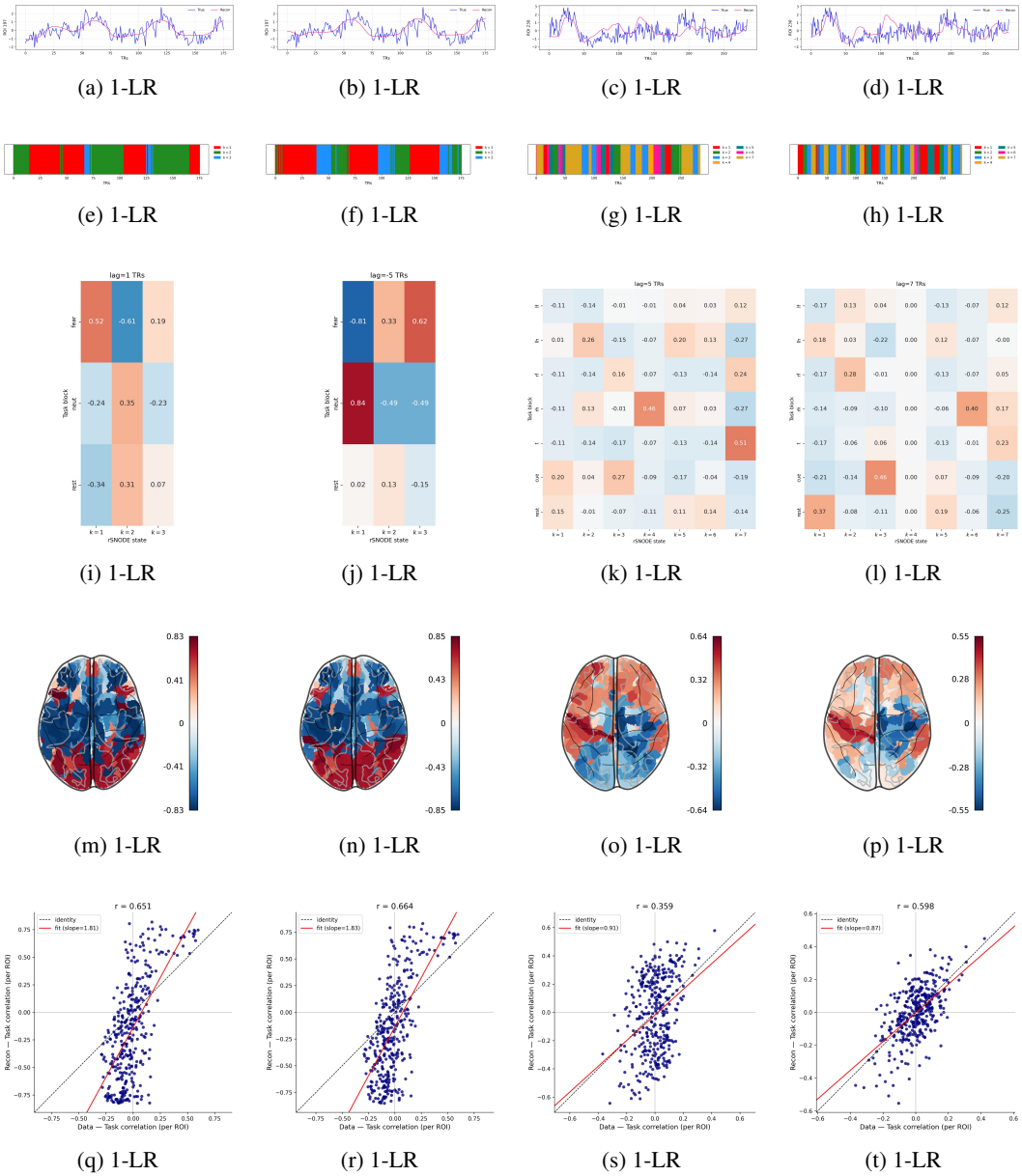
(m) 2-RL (n) 3-RL (o) V1-RL (p) V2-RL

(q) 2-RL (r) 3-RL (s) V1-RL (t) V2-RL

Figure 18: Reconstructed trajectories, generated states, state-task correlations, ROI activations, and ROI activation correlations for the EMOTION task for rSNODE across subjects and encoding directions (labeled as subject number - encoding direction; V = validation subject).

D.2 EMOTION WITH 3 STATES AND MOTOR WITH 7 STATES

EMOTION rSNODE Seed = 8 $D_L = 3$	**EMOTION rSNODE Seed = 11** $D_L = 3$	**MOTOR rSNODE Seed = 8** $D_L = 18$	**MOTOR rSNODE Seed = 11** $D_L = 18$



(a) 1-LR (b) 1-LR (c) 1-LR (d) 1-LR

(e) 1-LR (f) 1-LR (g) 1-LR (h) 1-LR

(i) 1-LR (j) 1-LR (k) 1-LR (l) 1-LR

(m) 1-LR (n) 1-LR (o) 1-LR (p) 1-LR

(q) 1-LR (r) 1-LR (s) 1-LR (t) 1-LR

Figure 19: Reconstructed trajectories, generated states, state-task correlations, ROI activations, and ROI activation correlations for the EMOTION and MOTOR task for rSNODE K equal to the number of task blocks across seeds.

D.3 rSNODE MOTOR RESULTS ACROSS SEEDS AND SUBJECTS

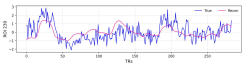
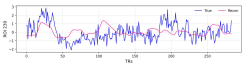
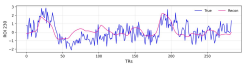
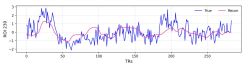
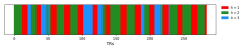
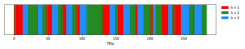
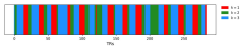
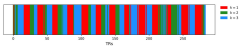
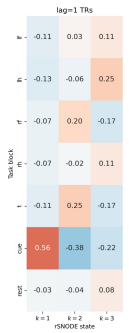
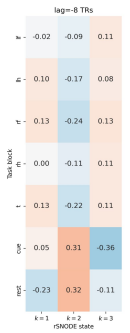
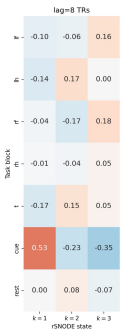
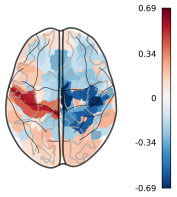
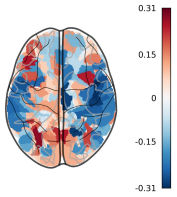
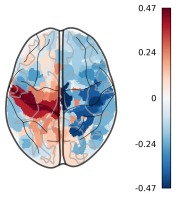
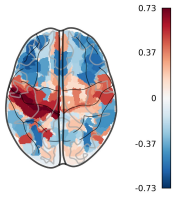
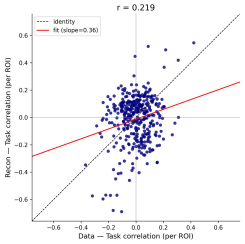
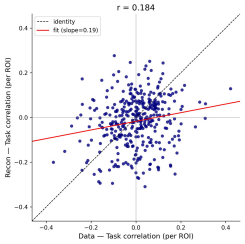
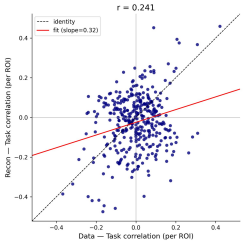
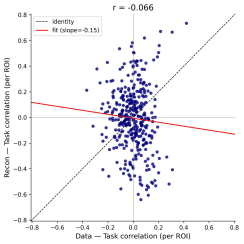
MOTOR rSNODE Seed = 11 $D_L = 18$	**MOTOR rSNODE Seed = 16** $D_L = 18$	**MOTOR rSNODE Seed = 24** $D_L = 18$	**MOTOR rSNODE Seed = 42** $D_L = 18$
			
(a) 1-LR	(b) 1-LR	(c) 1-LR	(d) 1-LR
			
(e) 1-LR	(f) 1-LR	(g) 1-LR	(h) 1-LR
			
(i) 1-LR	(j) 1-LR	(k) 1-LR	(l) 1-LR
			
(m) 1-LR	(n) 1-LR	(o) 1-LR	(p) 1-LR
			
(q) 1-LR	(r) 1-LR	(s) 1-LR	(t) 1-LR

Figure 20: Reconstructed trajectories, generated states, state-task correlations, ROI activations, and ROI activation correlations for the MOTOR task for rSNODE across seeds.

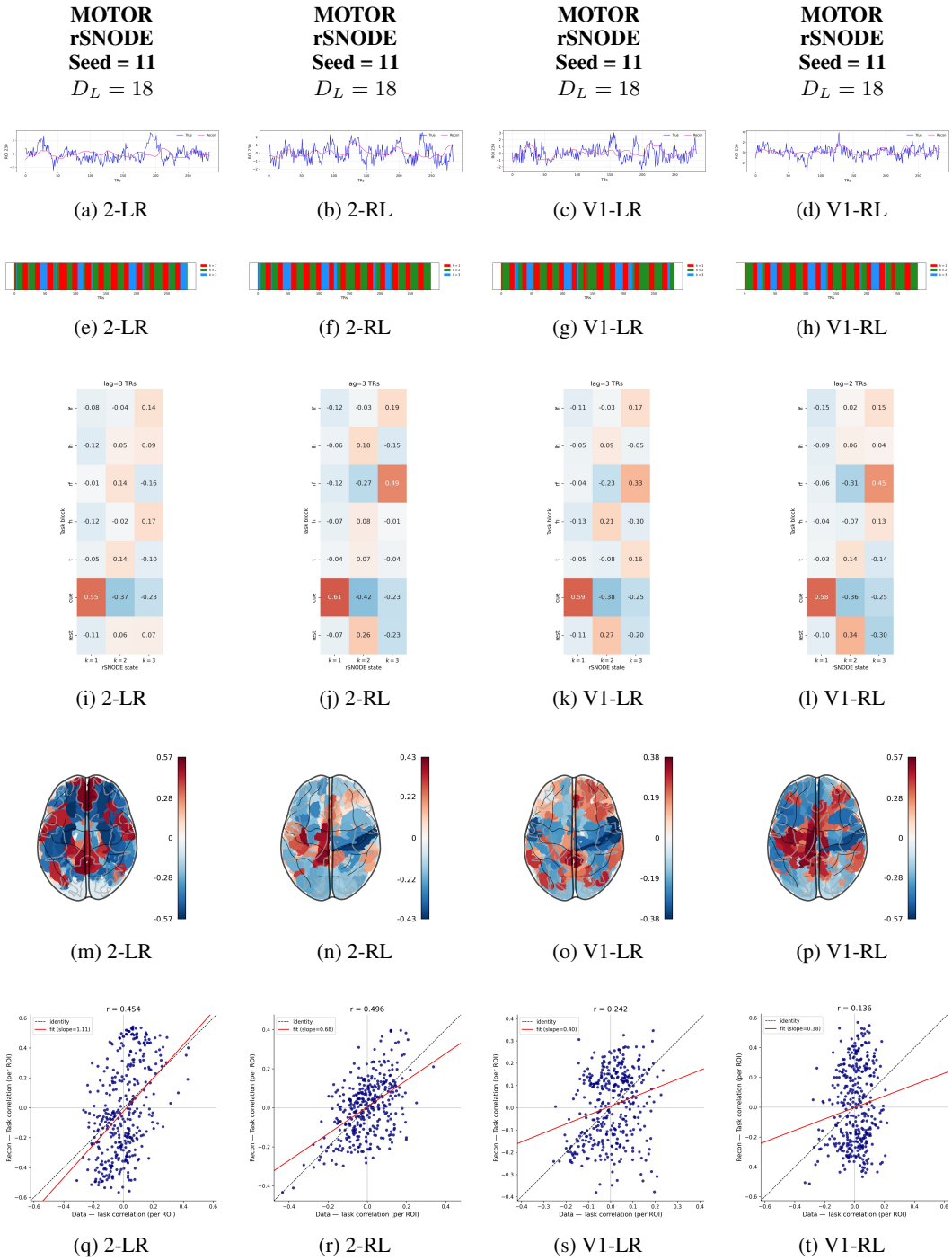


Figure 21: Reconstructed trajectories, generated states, state-task correlations, ROI activations, and ROI activation correlations for the MOTOR task for rSNODE across subjects and encoding directions (labeled as subject number - encoding direction; V = validation subject).

D.4 rdSLDS RESULTS

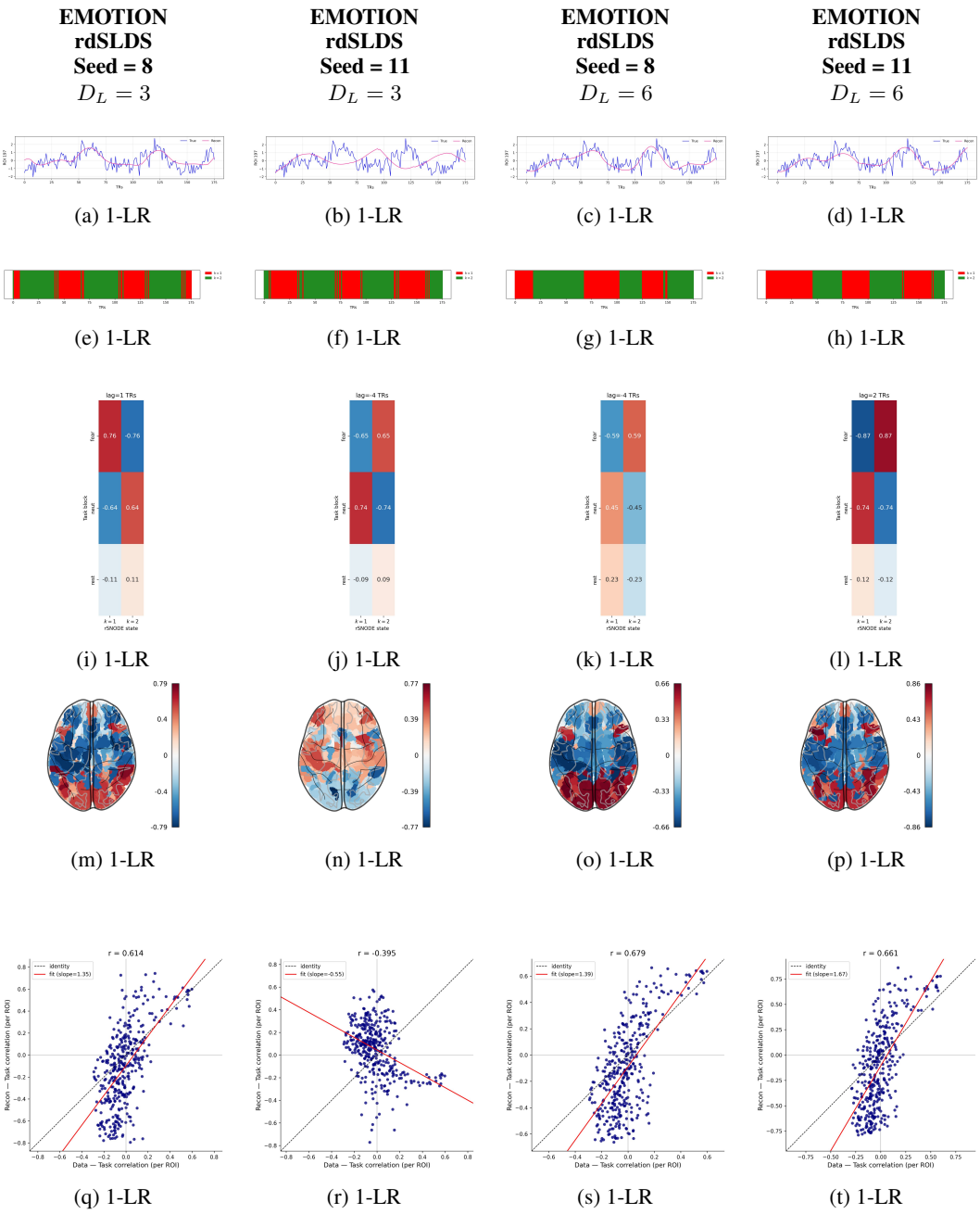


Figure 22: Reconstructed trajectories, generated states, state-task correlations, ROI activations, and ROI activation correlations for the EMOTION task for rdSLDS across seeds and latent dimensionality D_L.

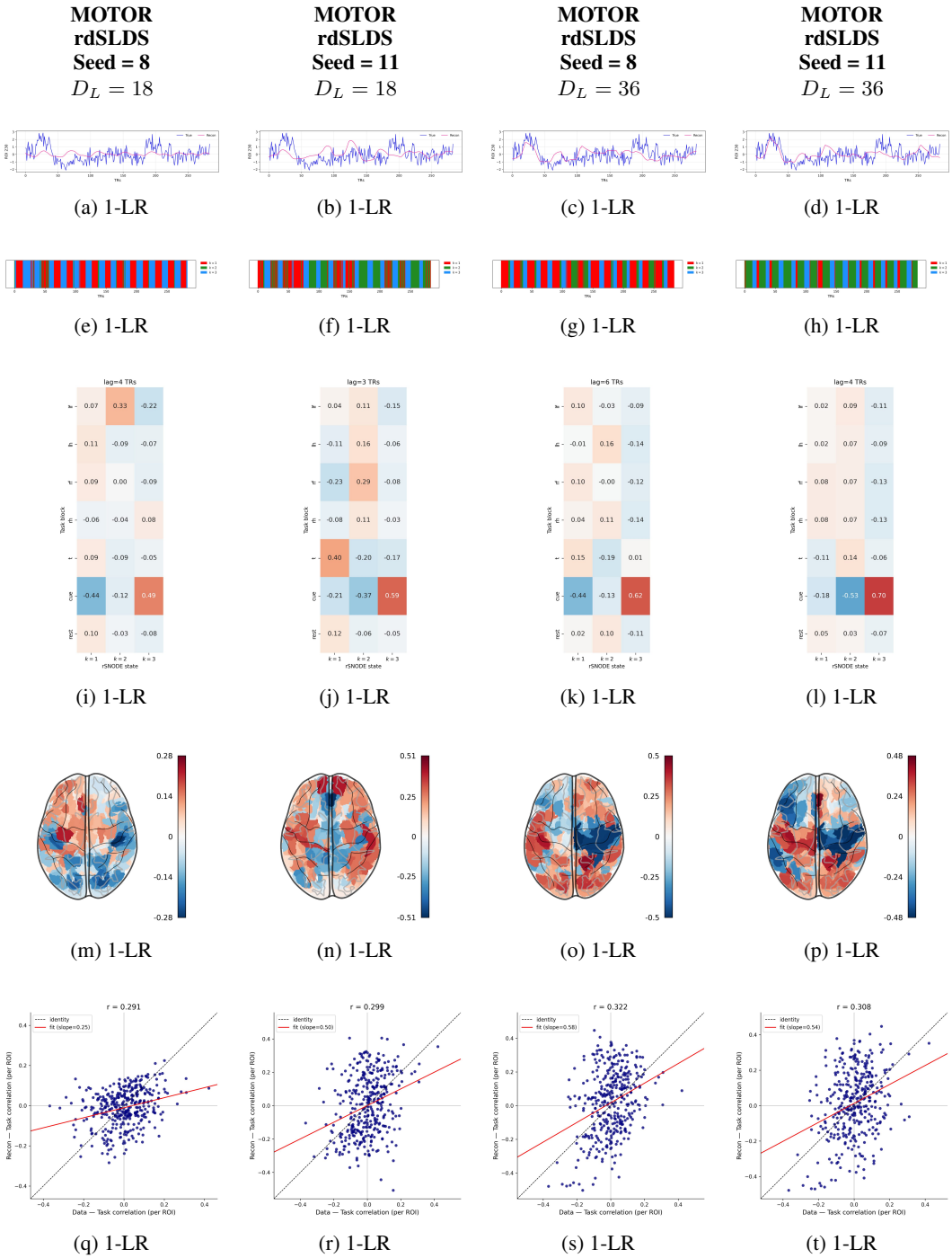
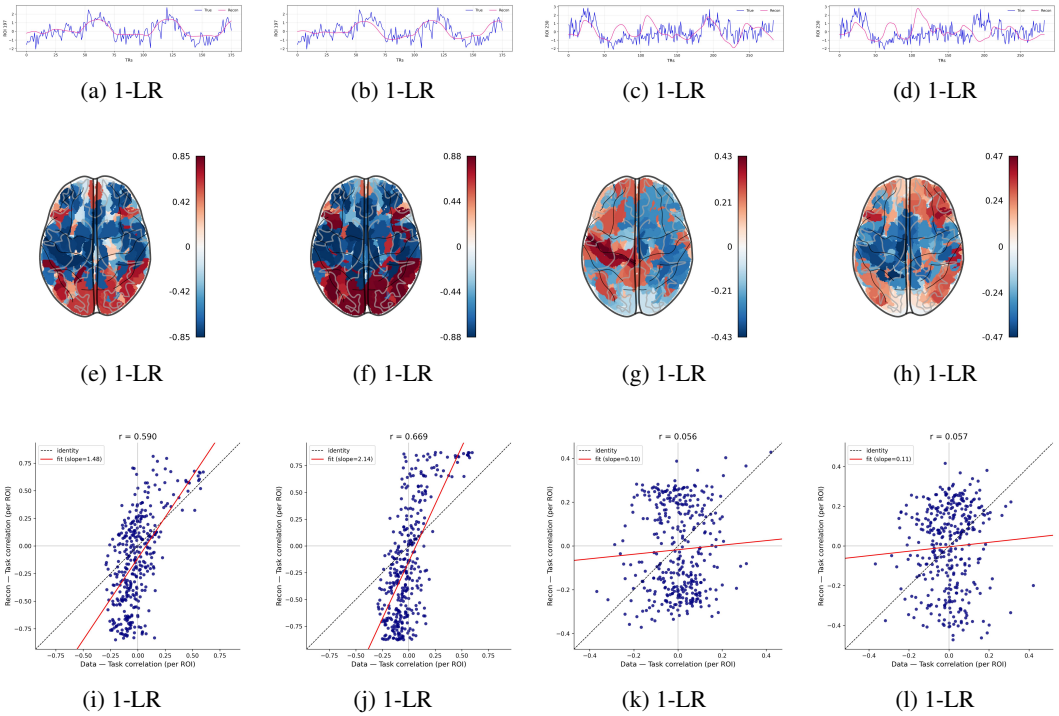


Figure 23: Reconstructed trajectories, generated states, state-task correlations, ROI activations, and ROI activation correlations for the MOTOR task for rdSLDS across seeds and latent dimensionality D_L.

D.5 NODE EMOTION AND MOTOR RESULTS

EMOTION NODE Seed = 8 $D_L = 3$	**EMOTION NODE Seed = 11** $D_L = 3$	**MOTOR NODE Seed = 8** $D_L = 18$	**MOTOR NODE Seed = 11** $D_L = 18$



(a) 1-LR (b) 1-LR (c) 1-LR (d) 1-LR

(e) 1-LR (f) 1-LR (g) 1-LR (h) 1-LR

(i) 1-LR (j) 1-LR (k) 1-LR (l) 1-LR

Figure 24: Reconstructed trajectories, generated states, state-task correlations, ROI activations, and ROI activation correlations for the EMOTION and MOTOR tasks for NODE across seeds. The model was trained using the hyperparameters described in Table 3.