Decomposed Linear Dynamical Systems (dLDS) for identifying the latent dynamics underlying high-dimensional time-series

Anonymous Authors[1]

Abstract

Learning interpretable representations of neural population dynamics is a crucial step to understanding how brain activity relates to behavior. Models of neural dynamics often focus on either low-dimensional projections that overlook the temporal relationships within the data, oversimplify the dynamics to linear and stationary patterns, or provide un-interpretable representations. Here, we consider dynamical systems as representative of flows on a low-dimensional manifold, and propose a new decomposed Linear Dynamical Systems (dLDS) model that captures complex nonstationary dynamics. dLDS models the latent state's evolution as following a sparse combination of simple interpretable components identified through a dictionary learning procedure. Importantly, the decomposed nature of the dynamics enables identifying overlapping co-active processes—-a feature unavailable to other methods. Through several examples, we demonstrate our model's ability to learn interpretable representations of multiple systems and demix population dynamics of multiple sub-networks. Finally, when applying our model to neural recordings of *C. elegans*, we identified unique patterns of dynamics emerging across behavioral states, which are obscured by other methods.

1. Introduction

The past decade has seen emergence of new technologies that enable the simultaneous recording of hundreds of neurons (Demas et al., 2021; Steinmetz et al., 2021), requiring the development of new computational models capable of modeling the dynamics of large neuronal populations (Saxena & Cunningham, 2019). Existing models often rely on either dimensionality reduction or dynamical systems modeling. Dimensionality reduction primarily addresses the identification of a small number of degrees of freedom that characterize a time series recording, however often treat individual time points independently. Traditional dimensionality-reduction methods (e.g., PCA, ICA, POD (Berkooz et al., 1993)), and their variants (e.g., (Yu et al., 2008; Wu et al., 2017)), can be viewed as decomposing time signals into a linear summation of components with time-changing coefficients. However, these methods are not tailored to capture the dynamical properties of the system. These methods have recently given way to more flexible descriptions of dimensionality reduction. Particularly, recent methods rely on the manifold hypothesis (Wu et al., 2017; Nieh et al., 2021; Gallego et al., 2017; Cunningham & Yu, 2014; Benisty et al., 2021; Mishne et al., 2016), including local embeddings (Balasubramanian & Schwartz, 2002; Roweis & Saul, 2000) and variational auto-encoders (Han et al., 2019). This *neural manifold* assumption (i.e., that instantaneous neural activity patterns lie on a low-dimensional manifold) removes the linearity assumption and thus enables the identification of correlated activity that corresponds to a potentially much lower-dimensional geometric structure. Similar to the linear dimensionality reduction methods, however, nonlinear methods also do not explicitly capture the temporal nature of the data.

Dynamical systems models, on the other hand, focus on capturing the temporal relationships within the data but often do not consider the geometric structure within it. Existing dynamical systems models offer provide either an interpretable, but over-simplified, representation that cannot capture the complexities in real-world dynamics (Churchland et al., 2012; Pillow et al., 2008), or, alternatively, powerful, yet un-interpretable, representations (e.g., RNNs (Pandarinath et al., 2018; Keshtkaran & Pandarinath, 2019)).

Newer models, as Switched Linear Dynamical Systems (SLDS) and its recurrent (rSLDS) variants (Ackerson & Fu, 1970; Chang & Athans, 1978; Hamilton, 1990; Bar-Shalom & Li, 1990; Ghahramani & Hinton, 1996; Murphy, 1998; Fox et al., 2008; Linderman et al., 2017; Nassar et al., 2018) solves the interpretability issue by capturing system's transi-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the Geometry-grounded Representation Learning and Generative Modeling at ICML 2024. Do not distribute.

tions over time between a discrete set of linear systems based on Markov process. Although SLDS and its extensions can discover latent dynamics and non-stationarities, they inherently cannot capture multiple co-occurring processes or overlapping subsystems (e.g., simultaneous observation of slow-timescale and faster-timescale systems).

Consequently, there is a critical need to develop methods that remain interpretable while maintaining the ability to capture rich non-linear structures, here termed "expressivity". In particular, we currently lack methods that integrate the manifold hypothesis into a dynamical systems model, thus maintaining both expressivity and interpretability. As a prime example, one challenge is the identification of multiple co-active dynamics subnetworks underlying the data whose overall effect is potentially non-linear and non-stationary.

To address this limitation, we choose to draw inspiration from sparse coding. Sparse coding assumes a form of efficiency in representation under a linear generative model (Olshausen & Field, 1996; Aharon et al., 2006). In the sparse coding representation, each data point y_k can be linearly generated from a latent vector x_k such that $y_k = Dx_k + \epsilon$, where the matrix D contains representational features as its elements and ϵ is some zero-mean noise. For any y_k, only a few features (columns of D) are required to construct the data point (i.e., the number of non-zero entries in x_k for any y_k is small). D is unknown and must be learned from data via "dictionary learning". Dictionary learning is a variational approach that includes iteratively inferring the sparse vector x_k and updating the dictionary D by gradient descent.

2. decomposed Linear Dynamical Systems

Here, we introduce dLDS that captures high-dimensional data as a sparse time-varying decomposition of dynamical flows on a low-dimensional manifold, thus enabling the modeling of complex behaviors while maintaining interpretability.

Let $Y = [y_1, ..., y_T] \in \mathbb{R}^{k \times T}$ be a sequence of T k-dimensional observations $y_1, ..., y_T$. A loading matrix $D \in \mathbb{R}^{k \times p}$ links each observation vector y_t to its underlying latent state, $x_t \in \mathbb{R}^p$, such that $y_t = Dx_t + \epsilon_t$, where ϵ_t is an *i.i.d* noise. Critical to dLDS is modeling the temporal evolution of latent states along the underlying geometry. We assume x_t lies on a d-dimensional manifold $\mathcal{M} \subset \mathbb{R}^p$, governed by flows that guide it according to its per-point tangent space (Fig. 1).

An appropriate discrete-time model is $x_t = F_t x_{t-1} + \nu_t$ where F_t is a local transformation between consecutive time-points and $\nu_t \sim \mathcal{N}(0, \sigma_\nu)$ is noise. Moreover, as the tangent space is a subspace, the operator F_t can be decomposed at each time point into a linear combination of M dictionary elements $\{f_m\}$ ($f_m \in \mathbb{R}^{p \times p}$ for all $m = 1 \dots M$) which we refer to as dynamic operators (DOs). The DOs span the space of possible local motions at different points on the manifold. Through their time-changing sparse linear decomposition $F_t = \sum_{m=1}^{M} f_m c_{mt}$, with the coefficients $c_t = [c_{1t}, ..., c_{Mt}]^T$, they dictate the movement along the manifold by $x_t = F_t x_{t-1} + \nu_t = \sum_{m=1}^{M} f_m c_{mt} x_{t-1} + \nu_t$.

We distinguish between model parameters (D, $\{f_m\}_{m=1:M}$) and model coefficients (x_t, c_t). The parameters define the global latent geometry, while the coefficients dictate specific trajectories.

Training. We frame the learning as optimization over

$$\begin{aligned}\{\widehat{x}_t, \widehat{c}_t\}_{t=1}^T = \arg\min_{\{x_t, c_t\}} \Bigg[& \sum_{t=1}^{T} \|y_t - Dx_t\|_2^2 \\ & + \sum_{t=2}^{T} \lambda_0 \left\| x_t - \sum_{m=1}^{M} f_m c_{mt} x_{t-1} \right\|_2^2 \\ & + \sum_{t=1}^{T} (\lambda_1 \|x_t\|_1 + \lambda_2 \|c_t\|_1) \\ & + \sum_{t=2}^{T} \lambda_3 \|c_t - c_{t-1}\|_2^2 \Bigg].\end{aligned}$$

The first and second terms are data and dynamics fidelity, the third and fourth terms encourage sparsity in latent state and dynamics coefficient, while the fifth term is for temporal smoothness. λ_0, λ_1, λ_2, and λ_3 are scalar regularization weights. To optimize this objective, we follow the dictionary learning literature (Olshausen & Field, 1996) and perform coordinate descent, alternating between estimating latent variables x and c, and updating model parameters $\{f_m\}_{m=1:M}$ and D.

Inference of Latent Variables x and c. First, we derive an efficient process for inferring x_t and c_t given the data and an estimate of the model parameters, assuming isotropic Gaussian noise in the observation and dynamics model. For inferring $\{x_t\}_{t=1}^T, \{c_t\}_{t=1}^T$ in every time point, we adopt the recent Basis Pursuit De-Noising with Dynamical Filtering (BPDN-DF) (Charles et al., 2016) approach for dynamic filtering of sparse signals. Inferring all coefficients at each time step reduces to LASSO

$$\begin{aligned}\widehat{x}_t, \widehat{c}_t = \arg\min_{x_t, c_t} \Bigg[& \|y_t - Dx_t\|_2^2 \\ & + \lambda_0 \left\| x_t - (\sum_{m=1}^{M} f_m c_{mt}) x_{t-1} \right\|_2^2 + \lambda_1 \|x_t\|_1 \\ & + \lambda_2 \|c_t\|_1 + \lambda_3 \|c_t - \widehat{c}_{t-1}\|_2^2 \Bigg].\end{aligned}$$

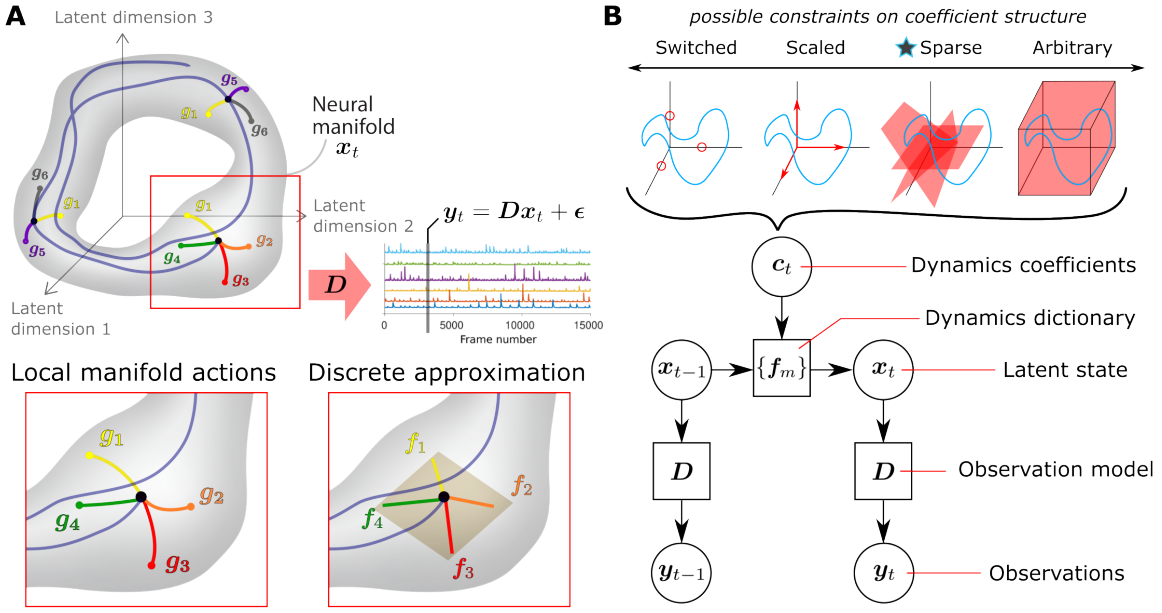

Figure 1. **dLDS illustration. A:** Trajectories along the manifold are guided by local DOs. The latent state x_t is indirectly observed through the observation model D. The transport space is learned through a discretized approximation $\{f_m\}_{m=1:M}$. **B:** dLDS enables capturing non-stationary dynamics, with coefficients that can flexibly adjust the contribution of different DOs over time.

Updating parameters. We update the model parameters $(D, \{f_m\})$ by gradient descent in every iteration $\widehat{D} \leftarrow \Pi_{C_D}(D - \eta_D \nabla_D \sum_{t=1}^{T} \|y_t - Dx_t\|_2^2)$ and $\widehat{f}_m \leftarrow \Pi_{C_f}(\widehat{f}_m - \eta_f \nabla_{f_m} \Sigma_{t=2}^{T} \|x_t - \sum_{m=1}^{M} c_{mt} f_m x_{t-1}\|_2^2)$ where $\prod_{C_D}$ and Π_{C_f} are projections of D's columns and each f_m spectral radius, respectively, to have a unit-norm.

In lower-dimensional data settings, we also consider the specific-case scenario where the observation matrix D is the identity matrix (i.e., $y_t = x_t$). This provides direct observations of manifold points and reduces computational complexity. We also intermittently perturb model parameters to avoid local minima.

2.1. Experiments

We showcase our model's to recover shared underlying operators from synthetic, low-dimensional, nonlinear systems in the case of changing stability regimes (Fig. 2), independent but simultaneously observed systems (Fig. 3), and obscured patterns in real-world data (Fig. 4).

dLDS flexibly models different stability regimes. We consider a simple transition of a spiral system that shifts from a stable, decaying regime to an unstable, expanding regime, following $x_t = \begin{cases} 0.99 f x_{t-1} & \text{if } 0 < t \leq \frac{T}{2}, \\ \frac{1}{0.99} f x_{t-1} & \text{if } \frac{T}{2} < t \leq T \end{cases}$

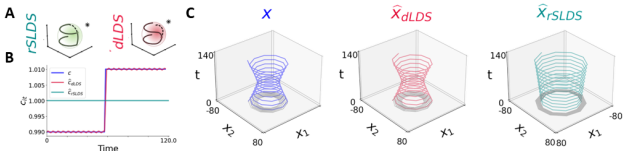

Figure 2. **dLDS captures changes in system stability. A:** Schematic behavior of dLDS and switching dynamics in modeling transitions. **B:** The generated coefficients (blue) vs. dLDS' recovered coefficients (red), and rSLDS coefficients (teal). **C:** Ground truth dynamics (blue) vs. the recovered dynamics by dLDS (red) and rSLDS (teal).

where $f = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$ with $\theta = \frac{\pi}{5}$. (Fig. 2A). We first found that dLDS manages to well capture the system (Fig. 2C red), in contrast to the switching model that is uncapable of capturing a changing direction when constrained to a single dynamic operator (Fig. 2C teal).Moreover, dLDS can further recover the ground truth dynamic operator with high precision, as well as its dynamic coefficient (Fig. 2B).

dLDS disentangles simultaneously observed systems. An additional benefit of dLDS is the ability to account for multiplexed sub-systems within the same recorded data. We focus on the case of two neural populations, each encompasses multiple unobserved sub-circuits, can be presented simultaneously or alone, with varying activation levels. We simulate a ten-dimensional state x_t with two populations (first

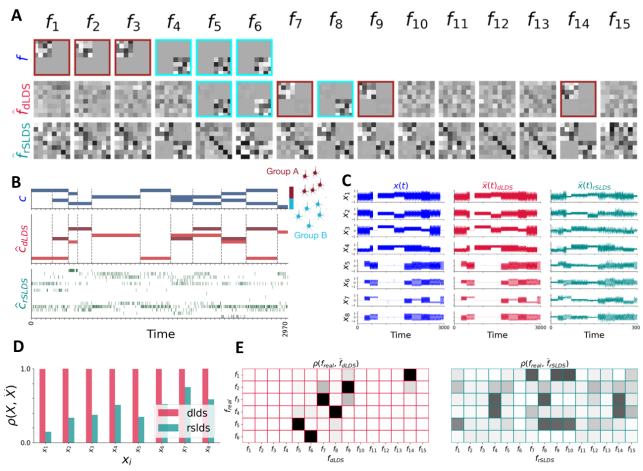

Figure 3. **dLDS identifies independently evolving groups from combined time series. A:** Ground truth DOs (top), DOs recovered by dLDS (middle), DOs recovered by rSLDS (bottom). **B:** Ground truth coefficients (top), dLDS recovered coefficients (middle, accurate recovery of structure), and rSLDS recovered coefficients (bottom). **C:** Generated dynamics (left), dLDS reconstruction (middle), rSLDS reconstruction (right). **D:** Correlations between ground truth data and reconstructions of dLDS vs. rSLDS. **E:** Correlations of ground truth DOs (rows) with recovered DOs (columns), for dLDS (left) vs. rSLDS.

vs. last five elements). We generate six ground truth DOs, three for each population (maroon vs. cyan, Fig. 3A). Each population switches between LDSs or goes silent (Fig. 3C blue). This process is repeated 50 times with random initial states and switch patterns.

In contrast to rSLDS, dLDS recovered a basis of operators localized to the relevant population that match the ground truth (Fig. 3A, middle row) with a correlation close to 1 (Fig. 3E, left). The identified coefficients also captured correctly when each subsystem switched (Fig. 3B, red). In contrast, rSLDS learned dynamical systems with both upper and lower diagonal blocks active, indicating it learned combined dynamics of the subsystems and failed to capture the underlying independence between them (Fig. 3A, bottom).

dLDS identifies latent dynamics in *C. elegans* data. Finally, we apply dLDS to "whole brain" *C. elegans* calcium imaging recordings from (Kato et al., 2015; Zimmer, 2021) (Fig. 4). The data (Kato et al., 2015) we considered include the immobilized worms' pirouetting behavior under four states (forward crawl, reverse crawl, sustained reverse crawl, and post-reversal turn). dLDS revealed obscured differences in the neural dynamics during different behavioral states, and that the map D from the latent space to the neural observations was able to highlight neurons that heavily overlapped with known neurons of interest. Moreover, we identified obscured within-state evolving dynamics (Fig. 4),

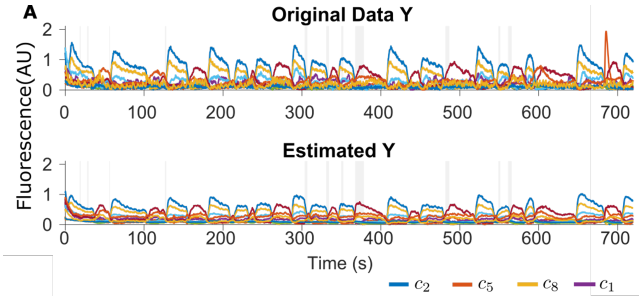

Figure 4. Results on *c. elegans* show that our model captures real-world non-stationary dynamics.

a feature unavailable to rSLDS, which may indicate a gradual change in the worm's internal state or behavior in the middle of these discretely-labeled behavioral states.

Additional experiments. We further tested dLDS in additional settings in the Appendix.

3. Summary and Future work:

Summary. Here, we present a manifold-flow-inspired model of learning decomposed Linear Dynamical Systems (dLDS). dLDS expands on the idea of switching linear systems to a model where linear combinations of a finite dictionary of systems can represent a richer set of dynamics. The examples we present highlight a series of both simple and more complex systems. We demonstrate that even simple systems represent plausible system behaviors in real data that are not captured well by existing techniques. Our model thus estimates dynamical systems that while are locally linear at each point, whose parameters change over time, thus enabling the capture of overall non-linear and non-stationary behavior. dLDS thus enable to model complex patterns while maintaining interpretability through the globality of the DOs and the sparsity applied on each time point. This way, it also enables the identification of overlapping networks active simultaneously as well as the smooth transition between dynamic behaviors. It further benefits from the efficiency, sparsity, and convergence guarantees of the BPDN-DF algorithm (Charles et al., 2016), which filters forward in time.

Limitations and future work: dLDS requires tuning several hyperparameters, and an important future target is to develop approaches to auto-set them. In addition, our work assumes linearity of the observation model, but non-linear projections might better fit certain cases. We acknowledge that advancements in modeling may aim to establish a more explicit relationship between c and x or the inclusion of a driving term in the dynamics equation, however assert that such endeavors go beyond the core model's scope.

References

Ackerson, G. A. and Fu, K. S. On State Estimation in Switching Environments. *IEEE Transactions on Automatic Control*, AC-15(1):10–17, 1970. ISSN 15582523. doi: 10.1109/TAC.1970.1099359.

Aharon, M., Elad, M., and Bruckstein, A. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

Balasubramanian, M. and Schwartz, E. L. The isomap algorithm and topological stability. *Science*, 295(5552): 7–7, 2002.

Bar-Shalom, Y. and Li, X.-R. *Estimation and tracking*. Artech House, Boston, MA, 1990.

Becker, S. R., Candès, E. J., and Grant, M. C. Templates for convex cone problems with applications to sparse signal recovery. *Mathematical programming computation*, 3(3): 165–218, 2011.

Benisty, H., Moberly, A. H., Lohani, S., Barson, D., Coifman, R. R., Mishne, G., Cardin, J. A., and Higley, M. J. Rapid fluctuations in functional connectivity of cortical networks encode spontaneous behavior. *bioRxiv*, 2021. doi: 10.1101/2021.08.15.456390.

Berkooz, G., Holmes, P., and Lumley, J. L. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual review of fluid mechanics*, 25(1):539–575, 1993.

Chang, C.-B. and Athans, M. State estimation for discrete systems. *IEEE Transactions on Aerospace and Electronic Systems*, 14(3):418–425, 1978. ISSN 02755823. doi: 10.5711/1082598316323.

Charles, A. S., Balavoine, A., and Rozell, C. J. Dynamic filtering of time-varying sparse signals via ℓ_1 minimization. *IEEE Transactions on Signal Processing*, 64(21): 5644–5656, 2016.

Churchland, M. M., Cunningham, J. P., Kaufman, M. T., Foster, J. D., Nuyujukian, P., Ryu, S. I., and Shenoy, K. V. Neural population dynamics during reaching. *Nature*, 487 (7405):51–56, 2012.

Cunningham, J. P. and Yu, B. M. Dimensionality reduction for large-scale neural recordings. *Nature neuroscience*, 17(11):1500–1509, 2014.

Demas, J., Manley, J., Tejera, F., Barber, K., Kim, H., Traub, F. M., Chen, B., and Vaziri, A. High-speed, cortex-wide volumetric recording of neuroactivity at cellular resolution using light beads microscopy. *Nature Methods*, 18 (9):1103–1111, 2021.

Fox, E., Sudderth, E., Jordan, M., and Willsky, A. Nonparametric bayesian learning of switching linear dynamical systems. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L. (eds.), *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008. URL https://proceedings.neurips.cc/paper/2008/file/950a4152c2b4aa3ad78bdd6b366cc179-Paper.pdf.

Gallego, J. A., Perich, M. G., Miller, L. E., and Solla, S. A. Neural manifolds for the control of movement. *Neuron*, 94(5):978–984, 2017.

Ghahramani, Z. and Hinton, G. E. Switching state-space models. Technical report, King's College Road, Toronto M5S 3H5, 1996.

Hamilton, J. D. Analysis of time series subject to changes in regime. *Journal of Econometrics*, 45(1-2):39–70, 1990. ISSN 03044076. doi: 10.1016/0304-4076(90)90093-9.

Han, K., Wen, H., Shi, J., Lu, K.-H., Zhang, Y., Fu, D., and Liu, Z. Variational autoencoder: An unsupervised model for encoding and decoding fmri activity in visual cortex. *NeuroImage*, 198:125–136, 2019.

Kato, S., Kaplan, H. S., Schrödel, T., Skora, S., Lindsay, T. H., Yemini, E., Lockery, S., and Zimmer, M. Global Brain Dynamics Embed the Motor Command Sequence of Caenorhabditis elegans. *Cell*, 163(3):656–669, 2015. ISSN 10974172. doi: 10.1016/j.cell.2015.09.034.

Keshtkaran, M. R. and Pandarinath, C. Enabling hyperparameter optimization in sequential autoencoders for spiking neural data. *Advances in Neural Information Processing Systems*, 32, 2019.

Linderman, S., Johnson, M., Miller, A., Adams, R., Blei, D., and Paninski, L. Bayesian learning and inference in recurrent switching linear dynamical systems. In *Artificial Intelligence and Statistics*, pp. 914–922. PMLR, 2017.

Mishne, G., Talmon, R., Meir, R., Schiller, J., Lavzin, M., Dubin, U., and Coifman, R. R. Hierarchical coupled-geometry analysis for neuronal structure and activity pattern discovery. *IEEE Journal of Selected Topics in Signal Processing*, 10(7):1238–1253, 2016. doi: 10.1109/JSTSP.2016.2602061.

Murphy, K. P. Switching Kalman filters. Technical report, Technical Report, UC Berkeley, 1998.

Nassar, J., Linderman, S. W., Bugallo, M., and Park, I. M. Tree-structured recurrent switching linear dynamical systems for multi-scale modeling. *arXiv preprint arXiv:1811.12386*, 2018.

Nieh, E. H., Schottdorf, M., Freeman, N. W., Low, R. J., Lewallen, S., Koay, S. A., Pinto, L., Gauthier, J. L., Brody, C. D., and Tank, D. W. Geometry of abstract learned knowledge in the hippocampus. *Nature*, 595(7865):80–84, 2021.

Olshausen, B. A. and Field, D. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(13):607–609, June 1996.

Pandarinath, C., O'Shea, D. J., Collins, J., Jozefowicz, R., Stavisky, S. D., Kao, J. C., Trautmann, E. M., Kaufman, M. T., Ryu, S. I., Hochberg, L. R., et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10):805–815, 2018.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Pillow, J. W., Shlens, J., Paninski, L., Sher, A., Litke, A. M., Chichilnisky, E., and Simoncelli, E. P. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454(7207):995–999, 2008.

Ravasi, M. and Vasconcelos, I. Pylops—a linear-operator python library for scalable algebra and optimization. *SoftwareX*, 11:100361, 2020.

Roweis, S. T. and Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500): 2323–2326, 2000.

Saxena, S. and Cunningham, J. P. Towards the neural population doctrine. *Current opinion in neurobiology*, 55: 103–111, 2019.

Steinmetz, N. A., Aydin, C., Lebedeva, A., Okun, M., Pachitariu, M., Bauza, M., Beau, M., Bhagat, J., Böhm, C., Broux, M., et al. Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings. *Science*, 372(6539):eabf4588, 2021.

Wu, A., Roy, N. A., Keeley, S., and Pillow, J. W. Gaussian process based nonlinear latent structure discovery in multivariate spike train data. *Advances in neural information processing systems*, 30, 2017.

Yu, B. M., Cunningham, J. P., Santhanam, G., Ryu, S., Shenoy, K. V., and Sahani, M. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Advances in neural information processing systems*, 21, 2008.

Zimmer, M. Kato2015 whole brain imaging data, January 2021. URL osf.io/2395t.

A. Duffing Dynamics and Linearization Components Recovery

As an example of the application of our model's abilities, we demonstrate its capability in recovering the latent components of a linearization of the Duffing oscillator. The Duffing oscillator is a classical example of a nonlinear, second-order differential equation that exhibits chaotic behavior. It is described by the following equation:

$$\ddot{x} + \delta\dot{x} + \alpha x + \beta x^3 = \gamma\cos(\omega t), \tag{1}$$

where α, β, and γ are parameters that control the behavior of the oscillator, and $\gamma\cos(\omega t)$ represents a periodic driving force with amplitude γ and frequency ω.

To apply our model to the Duffing oscillator, we define $y = \dot{x}$. We can then rewrite the Duffing oscillator equation in terms of y as follows:

$$\dot{y} + \delta y + \alpha x + \beta x^3 = \gamma\cos(\omega t). \tag{2}$$

We can represent the Duffing oscillator using a decomposed linear dynamical systems model by discretizing the system over time. We use the state vector $[x_t, y_t]$ to represent the state of the system at time t. The discrete-time model of the Duffing oscillator is given by the following equation:

$$\begin{bmatrix}\dot{x}_t \\ \dot{y}_t\end{bmatrix} = \begin{bmatrix}0 & 1 \\ -\alpha - \beta x_t^2 & -\delta\end{bmatrix}\begin{bmatrix}x_t \\ y_t\end{bmatrix} + \begin{bmatrix}0 \\ \gamma\cos(\omega t)\end{bmatrix}. \tag{3}$$

We can then apply our model to this discrete-time Duffing oscillator equation. We estimate the transition matrix A_t by fitting our model to the time series data. We can decompose A_t into two matrices, $F_t = F_{t,1} + F_{t,2}$, where $F_{t,1}$ and $F_{t,2}$ correspond to distinct physical processes.

Specifically, the matrix $F_{t,1} = \begin{bmatrix}1 & \Delta t \\ -\alpha\Delta t & 1 - \delta\Delta t\end{bmatrix}$ describes the linear dynamics of the system, while $F_{t,2} = \begin{bmatrix}0 & 0 \\ -\beta x_t^2\Delta t & 0\end{bmatrix}$ describes the nonlinearity of the system.

Using these matrices, we can write the discrete-time Duffing oscillator equation in matrix multiplication form as follows:

$$\begin{bmatrix}x_{t+1} \\ y_{t+1}\end{bmatrix} = \left(\begin{bmatrix}1 & \Delta t \\ -\alpha\Delta t & 1\end{bmatrix} + \begin{bmatrix}0 & 0 \\ -\beta x_t^2\Delta t & 0\end{bmatrix}\right)\begin{bmatrix}x_t \\ y_t\end{bmatrix} + b_t \tag{4}$$

where $b_t = \begin{bmatrix}0 \\ -\cos(t)\end{bmatrix}$, $\cos(t)$ represents the cosine function evaluated at time t. By estimating the matrices $F_{t,1}$ and $F_{t,2}$ using our model, we can recover the underlying linear and nonlinear components of the Duffing oscillator dynamics.

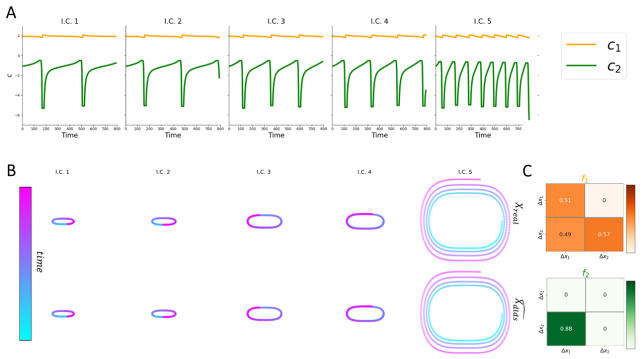

Figure 5. **Results of dLDS analysis on the Duffing oscillator.** A) Time traces (orange and green) of each dynamic operators (DO) for different initial conditions (I.C.) in separate subplots. B) Comparison between the real dynamics of the Duffing oscillator and the reconstructed dynamics by dLDS for each I.C. The color represents time. C) Heatmaps of each DO, where the first DO shows the main diagonal and other non-zero elements, as its coefficient in A is 2, while the second DO shows the lower left diagonal element and all other elements are zero.

By applying dLDS to the Duffing oscillator, we highlight the model's capacity to reconstruct nonlinear dynamics using basic linear components whose coefficients change over time and to recover the ground-truth basic linear elements that underlie these complex dynamics. These results emphasize the potential of dLDS for extracting valuable insights from intricate systems, by facilitating the reconstruction of nonlinear dynamics with basic linear components, and by retrieving the underlying basic linear elements.

B. A note on LASSO solvers

We found that the correlations between $f_m \widehat{x}_t$ could be large for certain time points. These correlations meant that some ℓ_1 regularize least-squares solvers would exhibit instability during learning. We noticed this in particular for the primary LASSO functions in both MATLAB and Python. We found that instead, the SPGL1 solver of the pylops package (Ravasi & Vasconcelos, 2020) was more robust in Python, and the TFOCS software [1] (Becker et al., 2011) was more robust in MATLAB. While TFOCS solves the LASSO program directly, SPGL1 solves a slightly modified version:

$$\hat{c}_t = \arg\min_{c} \left\| x_{t+1} - \sum_{m=1}^{M} f_m c_m(t) * x_t \right\|_2 \text{ s.t. } ||c_t||_1 \leq \tau. \tag{5}$$

Importantly, despite presenting only the SPGL1 results in the above paper, the Python code enables the user to choose from a wide range of solvers (including FISTA, ISTA, Sklearn LASSO (Pedregosa et al., 2011), OMP), and the decision of which solver to use is up to the user and should depend on the data properties and the user's goals in running the model.

[1] https://github.com/cvxr/TFOCS

C. Comparison between dLDS and rSLDS for the FHN model

Here we further demonstrate the comparison between dLDS to rSLDS on the FitzHugh–Nagumo model (Fig. 6). In contrast to the rSLDS model, for which the coefficients are binary, in our model the coefficients can take on continuous values. Hence, contrasting the observed coefficients-space spanned by rSLDS (6D) and dLDS (6E,F), in dLDS the dynamics representations are not limited to discrete locations on the axes, but can travel along them, resulting in a more flexible representation without the need to increase the number of dynamical systems learned. Specifically, we identify that while rSLDS learns slightly varying dynamical systems, while dLDS learns reorientations to different axes which more smoothly trade off with each other as the system rotates about the attractor.

Additionally, as the ℓ_1 regularization over c_t in dLDS increases, the coefficients become more similar to those obtained by rSDLS, namely, more restricted to the axes. Thus, modulating the regularization in our model makes possible the creation and exploration of a continuum of representations whose coefficients-space range from switched systems (high regularization) to arbitrary structured (unregularized), as described in Figure 1B.

D. FitzHugh-Nagumo (discrete-time model)

We used the Python discrete code for the FHN case. The iterative model ran until convergence (reconstructed error ¡ 1e-8) or until reaching a maximum of 6,000 iterations. The ground truth for the FHN dynamics was using 1000 samples with time intervals of 0.2 (s.t. $t_{\max} = 200$). We used $M = 2$ dictionary elements, an initial value of $\eta = 30$, while its decay rate over the training iterations was set to $\gamma = 0.99$; The standard-deviation of the perturbations added randomly to each f_m in case of local minimum, was set to 0.1.

For the regularized dynamics case, the hyperparameters of the SPGLl solver, "iter lim", the maximum number of solver iterations in each coefficients updating step, was set to 10, and τ (from (5)) was set to 0.3.

For the unregularized case, the following pseudo-inverse was used for each the updating step of the coefficients in

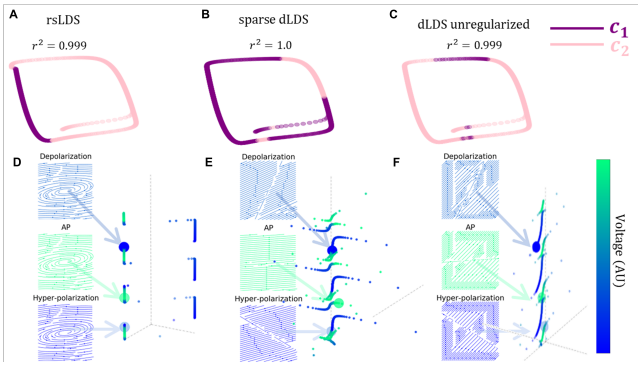

Figure 6. **Learned representations using dLDS and rSLDS for the FHN oscillator.** Note three time points of interest: repolarization, action potential peak, and hyperpolarization. With $M = 2$, dLDS can reconstruct the three distinct states, while rSLDS can only capture two reconstructed states in this case. A, B, C: All three models were able to reconstruct the FHN dynamics. D: The coefficients obtained by the rSLDS are restricted to the axes, resulting in no more than two distinct reconstructed states to describe the action potential cycle. E: Although the coefficients of the regularized dLDS tend to live on the axes due to the sparse regularization, this constraint is softer than of the rSLDS (in which living outside the axes is not possible). F: Coefficients space obtained for the unregularized dLDS model. Most coefficients do not necessarily live on the axes, since no regularization was applied.

each time point and in each iteration:

$$\begin{aligned} \widetilde{F}_t &= [f_1 x_t, f_2 x_t, ..., f_M x_t] \in \mathbb{R}^{(2\times M)} \\ x_{t+1} &= \widetilde{F}_t c_t \\ \widehat{c}_t &= \widetilde{F}_t^{\dagger} x_{t+1}, \end{aligned} \tag{6}$$

where † denotes the pseudo-inverse.

E. Sparse video example

To test the model in a sparse higher-dimensional setting we simulate a single dynamics function is present ($M = 1$) and the sparsity dictionary as the canonical basis ($D = I$). This test will check if our algorithm can accomplish simple system identification as a special case. We modeled the single dynamics function as a permutation matrix concatenated with a scaling matrix, i.e., signal coefficients move around and may be scaled (Fig. 7A). The learned and true models are a very close match, differing by only a permutation and sign change (the same ambiguity present in all dictionary learning methods).

In a more complex simulation, we simulate a dictionary of twelve distinct scaled permutation functions, only two of which are used at any time step (i.e. the sparsity of c_t is two). This system induces complex, highly non-stationary dynamics. Figure 7B,C depicts the results of the learning procedure, demonstrating that the sparsity dictionary is again learned up to a permutation and sign change, and the learned dynamics functions are again close matches to the true dynamics (i.e., we recover 12 scaled permutation matrices).

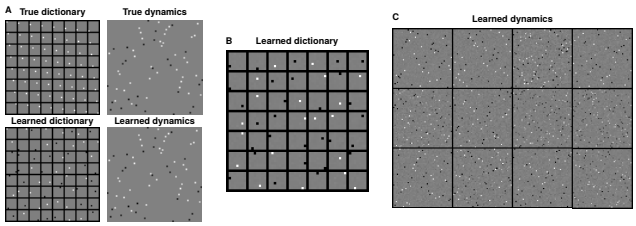

Figure 7. **Pixel permutation example.** A: Example test set consisting of a sparse number of pixels being permuted via an unknown permutation matrix. dLDS recovers in this case both the pixel-sparse dictionary as well as the ground truth permutation matrix. B: For a more complex example where multiple permutation matrices may be used (sometimes in tandem to split or merge pixels), the correct pixel-sparse dictionary is again learned. C: For the same example as B, the set of permutation matrices is learned, capturing the underlying dynamics.

F. BBC video example

To test dLDS on higher-dimensional real data, we learn a dynamics dictionary for natural video sequences. For computational considerations we restricted our algorithm to learn representations of 12x12 pixel patches, and learned a 4x overcomplete sparsity dictionary concurrently with 25 576x576 DOs. As no ground truth is available for video sequences, we instead qualitatively explore learned dictionaries. First we note that the sparsity dictionary recovered the expected Gabor-like statistics for image patches (Olshausen & Field, 1996; Aharon et al., 2006) (Fig. 8A). This result matches the intuition that the spatial statistics are not qualitatively changed by including the temporal model. To assess the dynamics we note that despite the high-dimensional nature of the data, the learned dynamics were relatively low-dimensional (rank 2-10), with one exception that had almost full rank (Fig. 8B). Additionally, the top eigenvectors tend to be correlated, but not overly so. The correlations cluster around ~ 0.2 with some correlations as high as 0.8 (Fig. 8C). This indicates that the learned functions are neither independent nor identical. Thus, interactions between the dLDS DOs permit flexible nonlinear behaviors.

Three such examples are shown in Figure 8D-F. First we project a single frame forward by a combination of two overlapping dynamics, f_1 and f_5. As the weight on f_1 is reduced and that on f_5 increased, the projection changes from exaggerating the linear feature in the top-left to inverting the image with an emphasis on the bottom right. Similar effects

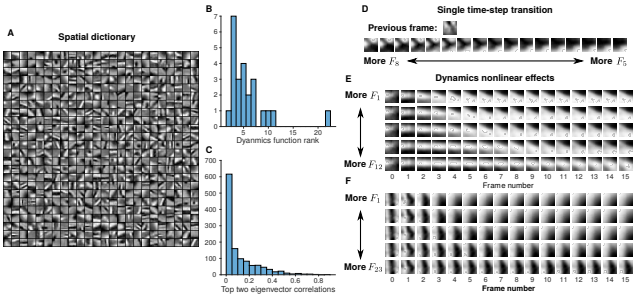

Figure 8. **Results of dynamics learning on natural image patches.** A: The spatial dictionary interestingly retains the Gabor-like structure seen in previous static dictionary learning algorithms. B: Learned dynamics are low rank. The typical rank of each f_m (aside from one almost full-rank function) ranges from 2-10. C: the correlations between the top two eigenvectors show that the dynamics are mostly non-aligned, yet overlap, allowing for second-order effects when combining dynamics. D: Linear combinations of dictionary elements can achieve nonlinear effects. Starting from the same frame, the next frame changes continuously between two possible next frames as the fraction of each dynamic function used is swept from completely using f_8 to f_5. E,F: Examples of dynamics combinations that achieve nonlinear effects. For each of changing from using more of f_1 and f_{12} and f_1 and F_{23}, the overall effect (rotation/expansion and outward expansion respectively) happens with faster or slower speeds.

appear in iterated dynamics projections, for example changing the weights on f_1 and f_{12}. This combination effectively rotates a bar over time, and the speed of rotation depends on the amount of f_1 vs. f_{12} in the linear combination. This type of speed modulation is impossible in a switched model unless one mode for each speed is included, which is untenable for a continuum of speeds. Similarly when f_1 and f_{23} are traded off, a vertical bar slowly has the bottom edge expand to encompass the bottom-right corner, again with different speeds depending on the ratio chosen.

G. Invariance of the model to transformations in the latent state

Consider the base model complete with the observation equation and decomposed dynamics,

$$y_t = Dx_t, \qquad x_t 11 = \left[\sum_{m=1}^{M} f_m c_{mt}\right] x_{t-1}. \quad (7)$$

For any learned model, we can always define a transformation of the latent space via an invertible matrix U such that

$$z_t = U^{-1} x_t \qquad x_t = U z_t. \quad (8)$$

This transformation results in an equivalent solution

$$y_t = DUz_t, \qquad z_t = \left[U^{-1}\sum_{m=1}^{M} f_m U c_{mt}\right] z_{t-1}, \quad (9)$$

i.e., an equivalent set of parameters $\widetilde{D} = DU$ and $\widetilde{f}_m = U^{-1}\sum_{m=1}^{M} f_m U$ result in the same sequence of dynamics but in a transformed latent space. One way to prevent the rotational ambiguity is to assume structure over the latent space, such as we implement via sparsity over x_t, which enables us to learn the correct representation (up to a permutation and sign-flip) of observation model (Fig. 7).