
Probabilistic Decomposed Linear Dynamical Systems for Robust Discovery of Latent Neural Dynamics

Yenho Chen^{1,3}, Noga Mudrik⁴, Kyle A. Johnsen³,
Sankaraleengam Alagapan², Adam S. Charles⁴, and Christopher J. Rozell^{1,2}

¹Machine Learning Center, Georgia Institute of Technology

²School of Electrical and Computer Engineering, Georgia Institute of Technology

³Coulter Dept. of Biomedical Engineering, Emory University and Georgia Institute of Technology

⁴Department of Biomedical Engineering, Mathematical Institute for Data Science,
Center for Imaging Science, Kavli Neuroscience Discovery Institute, Johns Hopkins University
yenho@gatech.edu, nmudrik1@jhu.edu, kjohnsen@gatech.edu
sankar.alagapan@gatech.edu, adamsc@jhu.edu, crozell@gatech.edu

Abstract

Time-varying linear state-space models are powerful tools for obtaining mathematically interpretable representations of neural signals. For example, switching and decomposed models describe complex systems using latent variables that evolve according to simple locally linear dynamics. However, existing methods for latent variable estimation are not robust to dynamical noise and system nonlinearity due to noise-sensitive inference procedures and limited model formulations. This can lead to inconsistent results on signals with similar dynamics, limiting the model’s ability to provide scientific insight. In this work, we address these limitations and propose a probabilistic approach to latent variable estimation in decomposed models that improves robustness against dynamical noise. Additionally, we introduce an extended latent dynamics model to improve robustness against system nonlinearities. We evaluate our approach on several synthetic dynamical systems, including an empirically-derived brain-computer interface experiment, and demonstrate more accurate latent variable inference in nonlinear systems with diverse noise conditions. Furthermore, we apply our method to a real-world clinical neurophysiology dataset, illustrating the ability to identify interpretable and coherent structure where previous models cannot. ¹

1 Introduction

A central goal in computational neuroscience is to develop models capable of discovering latent structure within noisy, high-dimensional neural signals. By identifying hidden relationships within neural recordings, we can begin to understand, predict, and control the behaviors of the underlying systems. Modeling neural time-series is challenging due to the range of temporal dynamics present. For example, there may be gradual short-term fluctuations, abrupt shifts in response to external stimuli, and long-term global drifts resulting from changes in baseline activity levels [36, 43, 39, 11, 7].

Although black-box approaches based on deep learning are available [34, 20, 38], their complexity often obscures the relationships learned from the data, making it difficult to extract scientific insights from these models. As a result, practitioners may favor time-varying linear state-space models which offer mathematically interpretable representations by approximating complex dynamics with simple

¹Code is available at:

<https://github.com/siplab-gt/probabilistic-decomposed-linear-dynamical-systems>

locally linear regimes [26]. However, obtaining latent variable estimates that are robust to dynamical noise and system nonlinearity in these state-space models is challenging. When applied to neural time-series, latent variable estimation may become unstable due to inflexible model formulations or noise-sensitive inference procedures. This can incorrectly produce disparate results for signals generated from the same underlying system.

For example, switching linear dynamical systems (SLDS) and related models [42, 44] segment time-series into discrete linear dynamical states, providing a piecewise linear approximation of the underlying system while highlighting coherent groups of activity. However, the assumption of discrete components can be a poor modeling choice for neural signals that contain continuous-valued fluctuations, such as gradual or random changes to the system speed as seen during neural ramping activity [31] or Lévy walk dynamics in the cerebral cortex [27]. As demonstrated in [30] and our experiments in Section 4.3, when applied to real-world datasets, inference of the switching variables can result in rapid, random oscillations between the discrete modes, indicating that the model is unable to identify meaningful structure in the data.

To address the limitations of discrete states, the decomposed linear dynamical systems (dLDS) model [30] learns a dictionary set of linear dynamical regimes, referred to as dynamic operators (DOs), that can be modified and combined through a linear combination of sparse coefficients. By allowing coefficients to be time-varying and continuous-valued, dLDS naturally captures both gradual changes by adjusting the coefficient magnitudes and abrupt shifts by changing the set of active DOs over time. Inference is accomplished by optimizing over a cost function that encourages data reconstruction while also constraining the structure of the dynamic coefficients to be sparse and temporally smooth.

Unfortunately, there are two critical shortcomings that prevent the robust inference of the latent variables in dLDS. First, the cost-based inference procedure is sensitive to noise, because of the regularization term encouraging temporal smoothness. This term sequentially propagates errors from noisy coefficient estimates over the length of the time series. Consequently, the model may produce inconsistent coefficient estimates on similar signals and have poor multi-step inference performance, indicating that the learned dynamics are unable to generalize well beyond a single time step. Second, the original latent dynamics model lacks a method for accurately representing systems with multiple fixed points, causing DO coefficients to oscillate or switch arbitrarily in a way that may not align with the underlying process. To learn an effective decomposed model in practice, we require a strategy that provides robust estimates of latent variables despite the presence of noise and system nonlinearity.

In this work, we address these limitations by introducing the probabilistic decomposed linear dynamical systems (p-dLDS) model. Our approach improves robustness of latent variable estimation while maintaining the richness of a decomposed dynamical systems model. First, we propose a probabilistic inference procedure that reduces the model’s sensitivity to temporal noise by accounting for uncertainty in the latent variable estimates over time. Namely, we introduce time-informed hierarchical variables that encourage both sparse and smooth model coefficients. We devise a variational expectation maximization (vEM) procedure to perform inference and learning over this probabilistic structure. Second, we incorporate a time-varying offset term to model systems that orbit multiple fixed points. While we analytically identify model degeneracies with this formulation, we propose an additive decomposition strategy that prevents convergence to trivial solutions.

Through several synthetic examples, we demonstrate how these contributions lead to improved accuracy and robustness of latent variable estimation despite difficult noise conditions. We extend these results to an empirically-derived brain-computer interface experiment [8], showcasing robustness to highly nonlinear observation functions and the ability to extract meaningful insights from the learned latent variables. Finally, we illustrate how our method effectively identifies interpretable and coherent structure in a clinical neurophysiology dataset where previous models are unsuccessful.

2 Background and Related Work

State Space Models. Our goal is to accurately describe the evolution of high-dimensional time-series data $\mathbf{y}_t \in \mathbb{R}^M$ with the following state-space equations,

$$\begin{aligned} \mathbf{y}_t &= \mathbf{D}\mathbf{x}_t + \mathbf{d} + \boldsymbol{\epsilon}_{y_t}, & \boldsymbol{\epsilon}_{y_t} &\sim \mathcal{N}(0, \boldsymbol{\Sigma}_y), & \text{(Observations)} \\ \mathbf{x}_t &= f_t(\mathbf{x}_{t-1}) + \boldsymbol{\epsilon}_{x_t}, & \boldsymbol{\epsilon}_{x_t} &\sim \mathcal{N}(0, \boldsymbol{\Sigma}_x), & \text{(Dynamics)} \end{aligned} \tag{1}$$

where $\mathbf{x}_t \in \mathbb{R}^N$ is the latent state, $f_t(\cdot)$ is the dynamics function, and $\mathbf{D} \in \mathbb{R}^{M \times N}$ and $\mathbf{d} \in \mathbb{R}^M$ describe a linear observation function. Our work focuses on the case when $N < M$, which compresses high-dimensional signals into a low-dimensional latent space. By choosing f_t to be a time-varying linear operator, we can approximate complex nonlinear dynamics with simple locally-linear components, balancing expressivity with mathematical interpretability. However, learning a time-varying linear operator from data can be challenging, and typically requires additional constraints on the underlying generative model to identify meaningful representations.

Switching Linear Dynamical System (SLDS). SLDS approximates nonlinear systems by introducing a discrete switching variable $z_t = \{1, \dots, K\}$ into the time-varying linear dynamics equation,

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{F}_{z_t} \mathbf{x}_t + \mathbf{b}_{z_t} + \epsilon_{x_t}.$$

At each time step t , the latent state \mathbf{x}_t evolves according to the z_t -th linear regime defined by $\mathbf{F}_{z_t} \in \mathbb{R}^{N \times N}$ and $\mathbf{b}_{z_t} \in \mathbb{R}^N$ while the switching variables evolve according to a Markov matrix. Inference is performed through a vEM algorithm, where the approximate posterior of the latent variables is estimated through coordinate ascent updates over tractable subgraphs. There are many extensions of SLDS, such as rSLDS [26] which modifies its generative behavior by informing the transitions of z_t with x_{t-1} . However, switching models are inherently limited when describing complex signals due to their discrete formulation. For instance, a switched representation is unable to learn that a dynamic regime may exhibit a range of variations. In neural systems, these variations may arise from random spiking processes [37, 40] or systems with randomly distributed speeds [41, 12]. In SLDS, each variation is learned as a separate discrete state, thus obscuring that the learned states are related. Furthermore, the switching formulation cannot adapt the learned system to unseen variations (i.e. different levels of random speeds). This can produce unstable inference behavior, where the switching state oscillates unpredictably or collapses to a single uninformative state.

Decomposed Linear Dynamical Systems (dLDS). dLDS [30] relaxes the discrete formulation by approximating nonlinear and nonstationary signals with a time-varying mixture of linear dynamical systems (LDS) defined by the following equations and constraints,

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{F}_t \mathbf{x}_t + \epsilon_{x_t}, \quad \mathbf{F}_t = \sum_{k=1}^K \mathbf{f}_k c_{t,k}, \quad \text{s.t. } \mathbf{c}_t \text{ is sparse.} \quad (2)$$

Every transition \mathbf{F}_t is decomposed as a linear combination of sparse coefficients $\mathbf{c}_t \in \mathbb{R}^K$ and a dictionary of K DOs $\mathbf{f}_k \in \mathbb{R}^{N \times N}$. Figure 1A shows the corresponding graphical model. Inference of the latent variables is accomplished by solving the Basis Pursuit Denoising with Dynamic Filtering (BPDN-DF) [6] objective sequentially for all t and $\lambda_0, \lambda_1, \lambda_2 > 0$,

$$\hat{\mathbf{x}}_t, \hat{\mathbf{c}}_t = \arg \min_{\mathbf{x}_t, \mathbf{c}_t} \|\mathbf{y}_t - \mathbf{D} \mathbf{x}_t\|_2^2 + \lambda_0 \|\mathbf{x}_t - \hat{\mathbf{x}}_{t-1} - \mathbf{F}_t \hat{\mathbf{x}}_{t-1}\|_2^2 + \lambda_1 \|\mathbf{c}_t\|_1 + \lambda_2 \|\mathbf{c}_t - \hat{\mathbf{c}}_{t-1}\|_2^2.$$

This produces a point estimate of \mathbf{x}_t and \mathbf{c}_t that matches the likelihood function resulting from Equations (1) and (2). In this objective, the dynamic coefficients are encouraged to be sparse through the ℓ_1 penalty and temporally smooth through the ℓ_2 penalty centered around the previous coefficient estimate. However, this approach is sensitive to noise because inference relies on propagating noisy point estimates of $\hat{\mathbf{c}}_{t-1}$ over time. As a result, BPDN-DF may accumulate errors that can lead to significantly different coefficient estimates on signals sampled from the same generative process. Furthermore, the lack of robustness to noise can degrade multi-step inference performance, causing the inferred system to quickly diverge from the true system. This suggests that the inferred latent variables only capture the local activity narrowly and are unable to accurately represent the dynamics beyond a single time-step. Another drawback of dLDS arises from the dynamics model in equation (2) which implicitly assumes that the observed dynamics contain a single fixed point that revolves around the origin. This limits dLDS's ability to model systems that cannot be easily mean-centered such as those with multiple fixed points or nonstationary drifts.

Sparse Bayesian Learning with Dynamic Filtering. Sparsity is achieved in probabilistic models through hierarchical scale-mixture priors [10, 2, 4]. To integrate dynamical information into probabilistic sparse signal inference, previous work [33] proposes the Sparse Bayesian Learning with Dynamic Filtering (SBL-DF) framework where the following hierarchical model is defined,

$$p(\mathbf{x}_t, \mathbf{c}_t, \gamma_t) = p(\mathbf{x}_t | \mathbf{c}_t) \prod_{k=1}^K p(c_{t,k} | \gamma_{t,k}) p(\gamma_{t,k} | a_{t,k}, b_{t,k}). \quad (3)$$

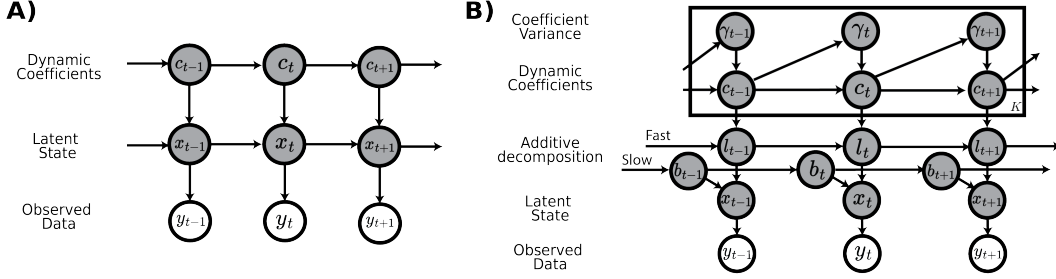


Figure 1: **(A)** Graphical model of dLDS. **(B)** p-dLDS includes hierarchical variables for probabilistic sparse inference and reparameterizes the latent space to include a time-varying offset term.

Here, $p(\mathbf{x}_t | \mathbf{c}_t) = \mathcal{N}(\Phi \mathbf{c}_t, \lambda_t \mathbf{I})$ specifies the likelihood, where Φ is a measurement matrix. Sparsity is encouraged through the zero-mean Gaussian priors $p(c_{t,k} | \gamma_{t,k}) = \mathcal{N}(0, \gamma_{t,k})$ independently placed on each element of the sparse vector. The variance parameters $\gamma_{t,k}$ are defined by an inverse gamma hyperprior $p(\gamma_{t,k} | \alpha_{t,k}, \beta_{t,k}) = \mathcal{IG}_{\gamma_{t,k}}(\alpha_{t,k}, \beta_{t,k})$ with shape parameters $\alpha_{t,k}$ and $\beta_{t,k}$. When marginalizing over $\gamma_{t,k}$'s, we see that $p(c_{t,k} | \alpha_{t,k}, \beta_{t,k})$ becomes a t-distribution known for its high kurtosis, which is essential for producing sparse solutions. To propagate dynamics information, the past estimate \hat{c}_{t-1} informs the hyperprior parameters of the next estimate such that $b_{t,k}/a_{t,k} = c_{t-1,k}^2$.

3 Probabilistic Decomposed Linear Dynamical Systems

We build upon dLDS and propose a probabilistic decomposed linear dynamical systems (p-dLDS) model. Rather than propagating noisy point estimates of the latent variables during inference, we improve robustness by marginalizing over uncertainty with respect to time. Additionally, we propose a tractable method for extending the dynamics model to systems with multiple fixed points.

3.1 Time-varying offset term

Note that for any parameter setting, the dLDS local dynamics (eq. (2)) reduces to a linear dynamical system (LDS) that is characterized by a single fixed point centered around the origin. Yet, real-world dynamical systems often consist of much more complicated behaviors. Nonlinearities can cause a signal to navigate through multiple fixed points throughout its trajectory, while nonstationarities may change the behavior of the system entirely with new fixed points emerging or disappearing. Simple preprocessing measures, such as mean-centering the data, are inadequate to account for these behaviors. To enable robustness against these behaviors, we introduce a time-varying offset term $\mathbf{b}_t \in \mathbb{R}^N$ into Equation (2) as a flexible way to account for dynamics not readily captured by the original dLDS latent dynamics model.

Lemma 1. *Let the transition between any two state vectors $\mathbf{x}_t, \mathbf{x}_{t+1} \in \mathbb{R}^N$ be defined by the linear dynamics matrix $\mathbf{F}_t \in \mathbb{R}^{N \times N}$ and the dynamics offset $\mathbf{b}_t \in \mathbb{R}^N$. For any $\lambda > 0$, the objective,*

$$\arg \min_{\mathbf{F}_t, \mathbf{b}_t} \|\mathbf{x}_{t+1} - \mathbf{x}_t - \mathbf{F}_t \mathbf{x}_t - \mathbf{b}_t\|_2^2 + \lambda \|\mathbf{F}_t\|_2^2,$$

is minimized when $\mathbf{F}_t = \mathbf{0}$ and $\mathbf{b}_t = \mathbf{x}_{t+1} - \mathbf{x}_t$.

This result, proven in Appendix B.1, reveals that introducing a time-varying offset term makes inference of the dynamics a degenerate problem. While the solution in Lemma 1 minimizes the objective, it fails to capture any meaningful structure in \mathbf{F}_t as the result of \mathbf{b}_t being unconstrained. To prevent the convergence to these trivial solutions, we decompose the latent state space as,

$$\mathbf{x}_t = \mathbf{l}_t + \mathbf{b}_t, \quad (4)$$

where \mathbf{l}_t captures fast dynamics and \mathbf{b}_t captures slow-varying trend behavior. These latent variables follow the dynamic equations $\mathbf{l}_{t+1} = \mathbf{l}_t + \mathbf{F}_t \mathbf{l}_t + \epsilon_{l,t+1}$ and $\mathbf{b}_{t+1} = \mathbf{b}_t + \epsilon_{b,t+1}$ where $\epsilon_l, \epsilon_b \in \mathbb{R}^N$ represent noise sampled from $\epsilon_b \sim \mathcal{N}(0, \Sigma_b)$ and $\epsilon_l \sim \mathcal{N}(0, \Sigma_l)$ respectively.

3.2 Probabilistic Time-Informed Sparsity

In decomposed models, we aim to achieve two goals simultaneously: sparsity and smoothness of coefficients over time. Motivated by this, we incorporate dynamics-informed probabilistic structure. First, we assume that each coefficient evolves independently of the others. Second, we introduce a hierarchical variance parameter $\gamma_{t,k}$ that controls the sparsity for each $c_{t,k}$. Moreover, we introduce dynamics information during sparse inference by encouraging a similar active support set in consecutive time slices through the variance hyperpriors. Put together, the resulting coefficient transition density in p-dLDS becomes,

$$p(\mathbf{c}_t, \boldsymbol{\gamma}_t | \mathbf{c}_{t-1}) := p(\mathbf{c}_t | \mathbf{c}_{t-1}, \boldsymbol{\gamma}_t) p(\boldsymbol{\gamma}_t | \mathbf{c}_{t-1}) = \prod_{k=1}^K p(c_{t,k} | c_{t-1,k}, \gamma_{t,k}) p(\gamma_{t,k} | c_{t-1,k}). \quad (5)$$

We define the first term on the right-hand side with the following functional form,

$$p(c_{t,k} | c_{t-1,k}, \gamma_t) \propto \exp\left(-\frac{c_{t,k}^2}{2\gamma_{t,k}} - \frac{(c_{t,k} - c_{t-1,k})^2}{2\sigma_{t-1,k}^2}\right) \propto \mathcal{N}(c_{t-1,k}, \sigma_{t-1,k}^2) \mathcal{N}(0, \gamma_{t,k}). \quad (6)$$

This density captures the constraints of sparsity and smoothness for the inferred coefficients $c_{t,k}$. When the variance around zero $\gamma_{t,k}$ is small, this structure promotes sparsity by shrinking coefficient values towards zero. Conversely, when the variance around the previous time step $\sigma_{t-1,k}^2$ is small, it encourages smoothness by shrinking coefficients towards the previous value. While the idea of combining two shrinkage effects in a single density has been explored in previous works [15, 5, 24, 18], those approaches generally require manual balancing of the two penalties. In contrast, we devise a procedure in the following section that estimates these variance parameters automatically during inference and learning.

The second density on the right-hand side from equation (5) is defined similarly to the hyperprior in SBL-DF. (i.e., $p(\gamma_{t,k} | c_{t-1,k}) = \mathcal{IG}(\xi, \xi c_{t-1,k}^2)$ where ξ weighs the influence of the dynamics when estimating $\gamma_{t,k}$). The resulting graphical model is shown in Figure 1B. We note that since the value of the previous coefficient is squared, the overall prior placed on the inverse gamma density follows a χ^2 distribution.

3.3 Inference and Learning

The joint distribution of p-dLDS is given by,

$$p(\mathbf{x}, \mathbf{y}, \mathbf{c}, \boldsymbol{\gamma} | \theta) = p(\mathbf{x}_1) \left[\prod_{t=1}^T p(\mathbf{y}_t | \mathbf{x}_t) \right] \left[\prod_{t=1}^{T-1} p(\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{c}_t) \left[\prod_{k=1}^K p(c_{t+1,k} | c_{t,k}, \gamma_{t+1,k}) p(\gamma_{t+1,k} | c_{t,k}) \right] \right], \quad (7)$$

where we denote $\mathbf{x} = \mathbf{x}_{1:T}$ for brevity. Exact posterior inference is intractable due to the nonconjugacy introduced by incorporating time-informed sparsity-inducing structure into the graphical model. As a result, we devise a variational expectation maximization (vEM) procedure where the approximate posterior is factorized as

$$p(\mathbf{x}, \mathbf{c}, \boldsymbol{\gamma} | \mathbf{y}, \theta) \approx q(\mathbf{x}) q(\mathbf{c}, \boldsymbol{\gamma}).$$

Here, the parameters are given by $\theta \in \{\mathbf{f}_{1:K}, \mathbf{D}, \mathbf{d}, \boldsymbol{\Sigma}_y, \boldsymbol{\Sigma}_x, \boldsymbol{\Sigma}_c\}$. Our approach contrasts with BPDN-DF, which estimates latent variables through separate ℓ_1 problems at each point in time. Instead, we preserve the time-dependence structure within each class of latent variables and leverage efficient inference algorithms that marginalize over uncertainty with respect to time. In general, we seek to maximize the variational lower bound,

$$\mathcal{L}_q(\theta) = \mathbb{E}_{q(\mathbf{x})q(\mathbf{c}, \boldsymbol{\gamma})} [\log p(\mathbf{y}, \mathbf{x}, \mathbf{c}, \boldsymbol{\gamma} | \theta) - \log q(\mathbf{x})q(\mathbf{c}, \boldsymbol{\gamma})],$$

with coordinate ascent updates on the latent state posterior, the dynamics coefficients posterior, and the model parameters.

Updating Latent State Posterior. The optimal coordinate ascent variational update is given by,

$$q(\mathbf{x}) \propto \exp\left(\mathbb{E}_{q(\mathbf{c}, \gamma)}[\log p(\mathbf{x}, \mathbf{y}, \mathbf{c}, \gamma|\theta)]\right). \quad (8)$$

Our assumed decomposition in equation (4) allows us to define the latent state transition density as $p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{c}_t) = p(\mathbf{x}_{t+1} = \mathbf{l}_{t+1} + \mathbf{b}_{t+1}) = \mathcal{N}(\mathbf{x}_{t+1}; \mathbf{l}_t + \mathbf{F}_t \mathbf{l}_t + \mathbf{b}_t)$. Substituting this into equations (7) and (8), we get that the optimal coordinate ascent approximate posterior becomes,

$$q(\mathbf{x}) = \mathcal{N}(\mathbf{x}_1; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \left[\prod_{t=2}^T \mathcal{N}(\mathbf{y}_t; \mathbf{D}\mathbf{x}_t + \mathbf{d}, \boldsymbol{\Sigma}_y) \right] \left[\prod_{t=1}^T \mathcal{N}(\mathbf{x}_t; \mathbf{l}_{t-1} + \mathbf{F}_{t-1}\mathbf{l}_{t-1} + \mathbf{b}_t, \boldsymbol{\Sigma}_x) \right], \quad (9)$$

where, $\boldsymbol{\mu}_1$ and $\boldsymbol{\Sigma}_1$ are the mean and covariance of the initial state.

Lemma 2. Let $\mathbf{l}, \mathbf{b} \in \mathbb{R}^N$ be independent random variables such that $\mathbf{l} \sim p(\mathbf{l})$ and $\mathbf{b} \sim p(\mathbf{b})$. Their sum $\mathbf{x} = \mathbf{l} + \mathbf{b}$ is distributed according to $\mathcal{N}(\boldsymbol{\mu}_l + \boldsymbol{\mu}_b, \boldsymbol{\Sigma}_l + \boldsymbol{\Sigma}_b)$ when 1) $p(\mathbf{b}) = \mathcal{N}(\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b)$ and $p(\mathbf{l}) = \mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)$ and when 2) $p(\mathbf{b}) = \delta(\mathbf{b} - \boldsymbol{\mu}_b)$ and $p(\mathbf{l}) = \mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l + \boldsymbol{\Sigma}_b)$.

We leverage Lemma 2 to reparameterize the state space trajectories into a deterministic and a stochastic component. The deterministic component captures the slow-moving offset density $q(\mathbf{b}) = \prod_{t=1}^T \delta(\mathbf{b}_t - \hat{\mathbf{b}}_t)$, where $\hat{\mathbf{b}}_t$ is estimated using a moving average of window size S which can be efficiently parallelized. The remaining dynamics are captured in the stochastic component. We define the family of variational distributions to be the class of linear Gaussian state space models, such that $q(\mathbf{l}) = \mathcal{N}(\mathbf{l}_1, \boldsymbol{\Sigma}_1) \prod_{t=2}^T \mathcal{N}(\mathbf{l}_t; \mathbf{l}_{t-1} + \mathbf{F}_{t-1}\mathbf{l}_{t-1}, \boldsymbol{\Sigma}_x)$. Conditioned on estimates $\hat{\mathbf{b}}$ and samples of \mathbf{c} , the optimal coordinate ascent variational update for $q(\mathbf{l})$ is efficiently computed using the Kalman Smoother [35]. We provide a full derivation of the update rule and Lemma 2 in Appendix B.2.

Updating Dynamics Coefficient Posterior. Sparse probabilistic representations introduce non-Gaussian factors which prevent closed-form message passing inference. Specifically, nonconjugacy arises from the inverse gamma term $p(\gamma_{t+1,k}|c_{t,k})$ since it is parameterized by $c_{t,k}^2 \sim \chi^2$. Moreover, the posterior distribution over the coefficients is highly multi-modal as a result of the implicit t-distribution in our hierarchical model. To update the coefficient posteriors, we propose a three-step procedure, where we factorize $q(\mathbf{c}, \gamma) = q(\mathbf{c})q(\gamma)$. First, we obtain an initial estimate of the variational distributions using SBL-DF. Second, we update $q(c_{t,k}) = \mathcal{N}(c_{t,k}^*, \hat{\gamma}_{t,k})$ using stochastic gradient descent (SGD) over

$$\mathbf{c}^* = \arg \max_{\mathbf{c}} \sum_{t=1}^{T-1} \log p(\hat{\mathbf{l}}_{t+1}|\hat{\mathbf{l}}_t, \mathbf{c}_t) + \log p(\mathbf{c}_{t+1}|\mathbf{c}_t, \hat{\gamma}_{t+1}) + \log p(\hat{\gamma}_{t+1}|\mathbf{c}_t), \quad (10)$$

where we have estimated the expectation in the optimal coordinate ascent update rule using samples from $\hat{\gamma} \sim q(\gamma)$ and $\hat{\mathbf{l}} \sim q(\mathbf{l})$. To retain coefficient sparsity, we only update coefficients within the active support set. In our work, this is defined as coefficients that have an initial estimate of $|c_{t,k}| > \eta$ where $\eta = 10^{-4}$. Finally, we update $q(\gamma)$ based on closed form conjugacy rules.

Update Parameters. Given our updated posteriors of the latent variables, we proceed to update the model parameters based on the ELBO,

$$\theta^* = \arg \max_{\theta} \mathbb{E}_{q(\mathbf{x})q(\mathbf{c})q(\gamma)} [p(\mathbf{y}, \mathbf{c}, \mathbf{x}, \gamma|\theta) - \log q(\mathbf{x})q(\mathbf{c})q(\gamma)] \approx \arg \max_{\theta} p(\mathbf{y}, \hat{\mathbf{c}}, \hat{\mathbf{x}}, \hat{\gamma}|\theta), \quad (11)$$

where we estimate the expectation with samples from our variational distributions and drop terms not dependent on θ . We use SGD to update all model parameters, which is possible when we assume that the covariance matrices have diagonal structure.

4 Results

We demonstrate p-dLDS in a variety of synthetic examples, highlighting improved robustness to noise and system nonlinearity. Additionally, we apply our model to a clinical neurophysiology dataset, revealing interpretable patterns where previous methods fail. We compare our method against SLDS, rSLDS, and dLDS as described in Section 2. All datasets are split 50:50 for training and testing. Due to space constraints, we provide full descriptions of the simulation setup in Appendix C and metric definitions in Appendix D.

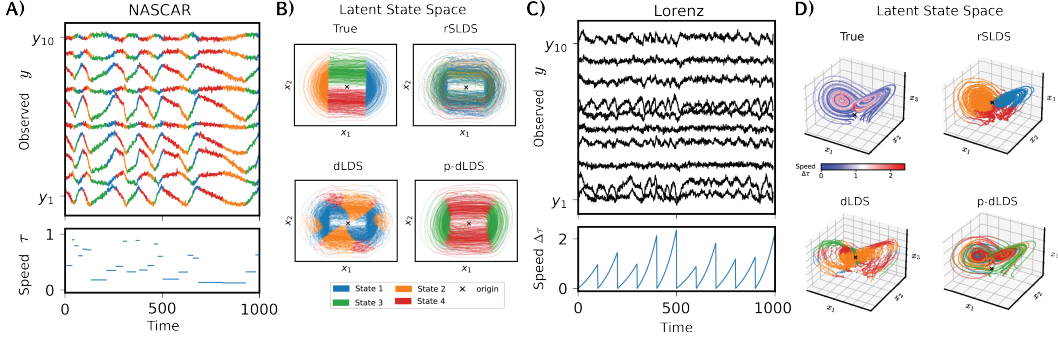


Figure 2: **Probabilistic model and offset term reduce estimation errors.** (A) Example trial from the NASCAR experiment colored by the true switching labels (not provided during training). Each track segment has a random speed τ . (B) Inferred state space, colored by discrete state or dominant coefficients. p-dLDS identifies correct track segments. (C) Example trial from the Lorenz experiment. The speed ramps according to the time intervals $\Delta\tau$ in an ODE solver. (D) Inferred state space, colored by the dominant coefficients. The time-varying offset term allows p-dLDS coefficients to switch according to the true speed and accurately model the two fixed points in the opposing lobes.

Table 1: Metrics for synthetic dynamical systems. Bold means best performance. (\uparrow) indicates higher score is better while (\downarrow) indicates that lower is better. \times indicates that value diverged towards $-\infty$. Switch events for decomposed models are defined as times where the active set of DOs change from the previous time step.

Model	NASCAR			Lorenz		
	Dynamics MSE (\downarrow) ($\times 10^{-3}$)	Switch MSE (\downarrow) ($\times 10^{-3}$)	100-step R^2 (\uparrow)	Dynamics MSE (\downarrow)	Switch MSE (\downarrow)	100-step R^2 (\uparrow)
SLDS	0.0995	12.89	0.184	0.431	0.0204	-3.47
rSLDS	0.1065	13.17	0.238	0.304	0.0208	-11.54
dLDS	123.19	13.28	\times	1.123	0.1529	\times
p-dLDS (ours)	0.033	7.34	0.450	0.141	0.0137	0.418

4.1 Synthetic Dynamical Systems

NASCAR with Random Speeds. We evaluate our inference procedure on the NASCAR dataset [26, 32, 23]. Since this system is easily mean-centered, we can isolate the effect of our proposed inference procedure as offset terms are not necessary. To make this dataset more realistic, we introduce speed variability into the ground truth dynamics as opposed to having a perfect constant speed at all time. Specifically, whenever a trajectory enters a new segment of the track, the system experiences a random change in speed. We trained all models on 30 trials, each consisting of 1000 time steps (Fig. 2A) with randomly sampled initial points, and a randomly constructed 10-dimensional linear observation matrix (see Appendix C.1 for full details). Performance is evaluated on 30 held-out trials where the ground truth switching states are defined by the different segments of the track. In all models, we set the $M = 10$, $N = 2$ and $K = 4$ for DOs or switching states. In our experiments, we define the "discrete states" for decomposed models as the DO state with the largest coefficient magnitude.

Figure 2B shows that changes in the system speed mask the true transition behavior between segments of the track in rSLDS. Moreover, dLDS identifies coherent segments, but inappropriately learns a different switching pattern for outer and inner edges of the track. In contrast, p-dLDS identifies a switching pattern most consistent with the true track segments despite the presence of noise and randomness in the system's speed. We note that while there are four true segments, decomposed models form a more parsimonious representation by identifying similar behaviors in different track segments such as in both edges and curves.

Table 1 summarizes our quantitative evaluations on three metrics: 1) the mean squared error (MSE) between the learned and ground truth latent dynamics, 2) the MSE between the inferred and true switch rate to determine agreement of the discrete switching behavior, and 3) the 100-step inference R^2 to demonstrate that the learned system generalizes beyond a single step on held-out data. (See Appendix D for mathematical definitions). We see that p-dLDS broadly outperforms existing methods in all metrics and significantly improves inference for decomposed models.

Lorenz System with Random Ramping. Next, we consider the Lorenz system, a chaotic nonlinear system with multiple fixed points, exploring the effect of the offset term. The system is described by the differential equation, $\dot{\mathbf{x}} = [\sigma(x_2 - x_1), x_1(\rho - x_3) - x_2, x_1x_2 - \beta x_3]^\top$ where the parameters $\rho = 28$, $\beta = 8/3$, and $\sigma = 10$ define a chaotic attractor with two opposing lobes (Fig. 2D). We introduce continuous fluctuations in the underlying dynamics by randomly ramping the system’s speed throughout each trajectory. This is accomplished by adjusting the evaluation time intervals given to an ODE solver. Similar to before, we randomly construct a linear observation function with $M = 10$ and train models on 30 randomly constructed trials with 1000 time points each (see Appendix C.2). Furthermore, we define the ground truth switch events as the time points when the signal transitions between the two lobes in addition to the moments when a ramping period concludes. All models are trained with a latent space of $N = 3$ and $K = 4$ states or DOs.

In figure 2D, we see that rSLDS does not distinguish between the different speeds along the outer and inner sections of the attractor. Instead, the discrete states obscure the continuum of speeds by incorrectly grouping all activity in each lobe into a single regime. Furthermore, we observe that dLDS is limited without an offset term, unable to accurately represent multiple fixed points. Instead of aligning with the two attractor lobes, transitions in the dominant coefficients occur radially relative to the origin and fail to reconstruct the two orbiting fixed points. Conversely, p-dLDS’s offset term enables learning a system where coefficients better match the true geometry. This representation correctly recovers differences between the outer and inner sections of the attractor while also accurately reconstructing the two orbiting fixed points. Moreover, this leads to improved estimation of latent dynamics, a switching rate that agrees with the true system, and improved multistep inference performance as shown in Table 1.

4.2 Simulated Motor Cortex Data in a Reaching Task

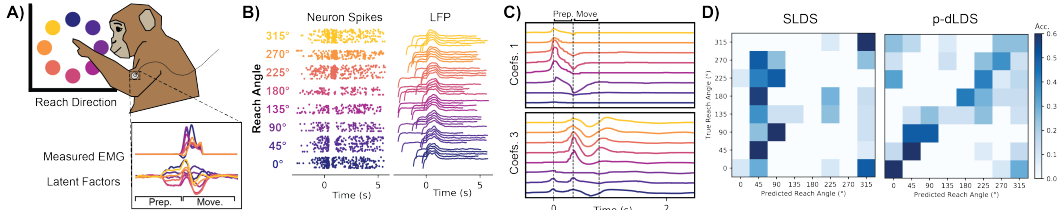


Figure 3: **p-dLDS efficiently captures changes in dynamics.** (A) Latent factors are computed from empirical EMG of the reaching experiment in [8]. Dynamics are characterized by a preparatory and movement phase. (B) Synthetic spikes and LFPs are generated using the `ws1fp` package [17, 16] (C) The trial-averaged coefficients for p-dLDS smoothly vary with reaching angle. DO 1 captures preparatory dynamics while DO 3 captures movement dynamics. (D) Confusion matrix for linear classification of reach directions. p-dLDS predictions closely align to true diagonal.

We now turn to an empirically-derived synthetic experiment related to brain-computer interfaces, where the dynamics and observation functions are nonlinear and derived from analysis of neural data. Our focus is on the reaching task, a neuroscience experiment designed to study motor control in non-human primates [19, 22]. In this experiment, the subjects are trained to reach towards visually cued targets, while neural activity is recorded from motor-related areas such as electromyography (EMG) data from arm muscles. Each trial consists of two distinct phases: preparation and movement. In the preparation phase, the subject plans its movement while keeping their arm still. In the movement phase, the subject physically reaches towards the target. The goal of this experiment is to decode reach intention from neural data. We construct a dataset by first simulating a spiking neural network with known latent factors [8] trained to reproduce empirical EMG signals from the center-out reach

Table 2: Inference performance for the reaching experiment (see Figure 3) on a held-out test set. Top-1 and Top-3 accuracies are obtained by predicting reach directions from latent variable features using linear classifiers. State and Dynamics MSE are computed with respect to true latent variables.

Model	Top-1 Acc. (\uparrow)	Top-3 Acc. (\uparrow)	State MSE (\downarrow) ($\times 10^{-1}$)	Dynamics MSE (\downarrow) ($\times 10^{-2}$)
SLDS	38.46	57.69	0.5289	0.3942
rSLDS	12.82	32.05	0.5503	292.41
dLDS	10.25	39.74	0.6742	35.680
pdLDS (ours)	42.31	70.51	0.4061	0.0567

task in [22]. Spikes are then converted to 50-channel local field potentials (LFP) recordings via a weighted, delayed sum of synaptic currents (see Figure 3A and B) [29, 17, 16]. Our dataset contains 150 6-second trials sampled at 250 hz, where each trial represents one out of eight reach directions visually cued at a random start time. PCA identifies that three components captures 98% of the variance. Thus, we set $M = 50$, $N = 3$, and $K = 4$ DOs or discrete states.

Figure 3C shows the trial-averaged DO coefficients from p-dLDS, which change smoothly and cyclically according to the true reach angle. Additionally, the DOs appear to differentiate between the two distinct dynamical regimes, where the activity of f_1 and f_3 localize to the preparatory and movement phase respectively. Quantitatively, we compute the linear classification accuracy of the reach angles using the average state activity over time as features (see Appendix D.5). Figure 3D shows that classifiers built from SLDS features fail to capture the full continuum of reach angles. This limitation occurs because the discrete switching states are unable to efficiently capture the diversity of activity present in the LFPs, which arise from randomness inherent in the spike sampling process. Consequently, the inferred features from SLDS generalize poorly to held-out data. In contrast, the p-dLDS classifier predictions recover the full spectrum of reach angles since features are naturally continuous and the inferred coefficients can adjust the learned DOs to accurately capture the activity in the held out data. Table 2 shows that p-dLDS outperforms all other models in state and dynamics reconstruction as well as the top-1 and top-3 reach classification accuracy.

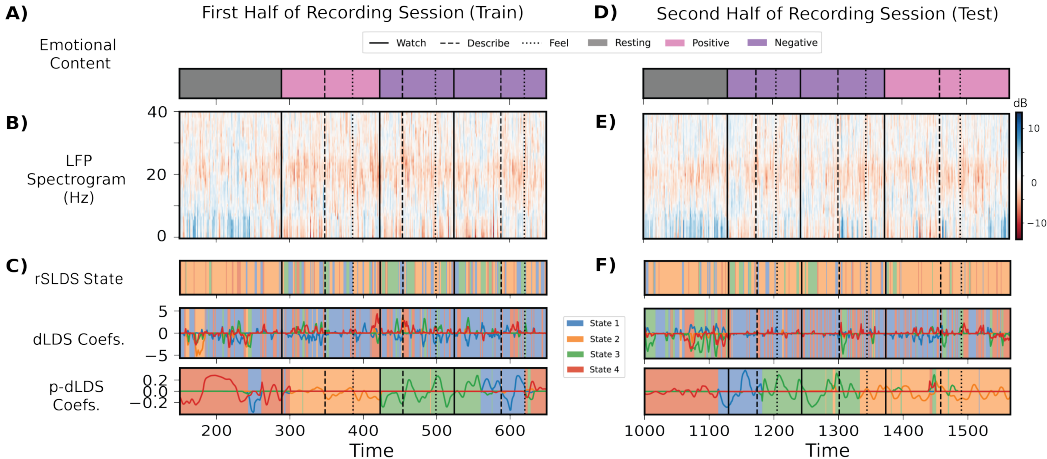


Figure 4: **Learned system discovers coherent structure in clinical data.** (A, D) LFP data was collected on patients watching videos with different emotional content. (B, E) LFP spectrograms are 40-dimensional signals where each channel represents a particular frequency. (C, F) Inferred states and coefficients shows that rSLDS and dLDS exhibit unpredictable switching behavior. In contrast, p-dLDS captures smooth coefficients and identifies DOs that align with the trial’s emotional content. The learned patterns broadly generalize to the held-out data.

4.3 Clinical Neurophysiology Data

We demonstrate p-dLDS on LFP recordings from the subcallosal cingulate cortex (SCC) in patients with treatment-resistant depression (ClinicalTrials.gov identifier NCT01984710). Subjects are asked to watch videos with different emotional content (positive, negative, and neutral), describe the videos, and then discuss how the video made them feel. SCC dynamics have been previously shown to provide a quantitative signal for the presence of emotional content [14] and depression recovery [1]. Thus, we hypothesize that the underlying dynamics may provide information about emotional changes throughout the experiment. We apply p-dLDS to a single patient’s LFP spectrogram data (Fig. 4B, E) within the 0-40 Hz frequency range ($M = 40$). PCA indicates that the first 7 components explain 90% of the variance. Therefore, we train a model with $N = 7$ latent dimension and $K = 4$ DOs.

In Figure 4, rSLDS and dLDS produces a high degree of state oscillations making it difficult to identify time intervals with consistent emotional content. In contrast, p-dLDS infers coherent structure that corresponds to changes in emotional content in the trial. For example, f_4 (red) coincides with resting, f_2 (orange) with positive videos, and f_1 and f_3 (blue and green) to negative videos (Fig. 4C). Importantly, this structure persists even on held out data from the second half of the session (Fig. 4 F). We note this preliminary analysis on a single subject isn’t intended to make a claim about specific neurophysiological responses to emotional content in this brain region, but generally highlights that p-dLDS identifies meaningful dynamical modes where previous models are unable to.

5 Conclusion

In this work, we present a probabilistic decomposed linear dynamical systems model that can be used to discover meaningful representations in neural signals. By marginalizing over uncertainty in latent variable estimates and incorporating an offset into the dynamics, we enhance robustness and improve a variety of performance metrics. Some areas of future work includes exploiting structure in the offsets to automatically identify window size and extending the probabilistic model to include more complicated emissions distributions, such as the Poisson likelihood commonly used to model neural spiking data [28].

Acknowledgements and Disclosure of Funding Sources

Y.C. and C.R. were funded by the James S. McDonnell Foundation (grant number 22002039), with Y.C. being further funded by National Institutes of Health (grant number 2T32EB025816), and C.R. being further funded by the Julian T. Hightower Chair. Y.C. and K.J. were part of the Georgia Tech/Emory NIH/NIBIB Training Program in Computational Neural Engineering (T32EB025816). N.M. was funded by The Kavli Foundation NeuroData Discovery award. A.S.C. were partially supported by the NSF CAREER Award (2340338) and a Johns Hopkins Bridge Grant. S.A. is supported by the National Center for Advancing Translational Sciences of the National Institutes of Health (Award Number UL1TR002378 and KL2TR002381). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

- [1] Sankaraleengam Alagapan, Ki Sueng Choi, Stephen Heisig, Patricio Riva-Posse, Andrea Crowell, Vineet Tiruvadi, Mosadoluwa Obatusin, Ashan Veerakumar, Allison C. Waters, Robert E. Gross, Sinead Quinn, Lydia Denison, Matthew O’Shaughnessy, Marissa Connor, Gregory Canal, Jungho Cha, Rachel Hershenberg, Tanya Nauvel, Faical Isbaine, Muhammad Furqan Afzal, Martijn Figee, Brian H. Kopell, Robert Butera, Helen S. Mayberg, and Christopher J. Rozell. Cingulate dynamics track depression recovery with deep brain stimulation. *Nature*, Sep 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06541-3. URL <https://doi.org/10.1038/s41586-023-06541-3>.
- [2] Kyoung-hwa Bae and Bani K Mallick. Gene selection using a two-level hierarchical bayesian model. *Bioinformatics*, 20(18):3423–3430, 2004.
- [3] Leonard Berrada, Andrew Zisserman, and M Pawan Kumar. Smooth loss functions for deep top-k classification. *arXiv preprint arXiv:1802.07595*, 2018.

- [4] Carlos M Carvalho, Nicholas G Polson, and James G Scott. The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480, 2010.
- [5] George Casella, Malay Ghosh, Jeff Gill, and Minjung Kyung. Penalized regression, standard errors, and bayesian lassos. 2010.
- [6] Adam S Charles, Aurele Balavoine, and Christopher J Rozell. Dynamic filtering of time-varying sparse signals via ℓ_1 minimization. *IEEE Transactions on Signal Processing*, 64(21):5644–5656, 2016.
- [7] Gustavo Deco, Morten L Kringelbach, Viktor K Jirsa, and Petra Ritter. The dynamics of resting fluctuations in the brain: metastability and its dynamical cortical core. *Scientific reports*, 7(1):3095, 2017.
- [8] Brian DePasquale, David Sussillo, LF Abbott, and Mark M Churchland. The centrality of population-level factors to network computation is demonstrated by a versatile approach for training spiking networks. *Neuron*, 111(5):631–649, 2023.
- [9] John R Dormand and Peter J Prince. A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980.
- [10] Mário AT Figueiredo. Adaptive sparseness for supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 25(9):1150–1159, 2003.
- [11] Michael D Fox and Marcus E Raichle. Spontaneous fluctuations in brain activity observed with functional magnetic resonance imaging. *Nature reviews neuroscience*, 8(9):700–711, 2007.
- [12] Zachary T Goodman, Jason S Nomi, Salome Kornfeld, Taylor Bolt, Roger A Saumure, Celia Romero, Sierra A Bainter, and Lucina Q Uddin. Brain signal variability and executive functions across the life span. *Network Neuroscience*, pages 1–15, 2024.
- [13] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.
- [14] Julius Huebl, Christof Brücke, Angela Merkl, Malek Bajbouj, Gerd-Helge Schneider, and Andrea A. Kühn. Processing of emotional stimuli is reflected by modulations of beta band activity in the subgenual anterior cingulate cortex in patients with treatment resistant depression. *Social Cognitive and Affective Neuroscience*, 11(8):1290–1298, 03 2016. ISSN 1749-5016. doi: 10.1093/scan/nsw038. URL <https://doi.org/10.1093/scan/nsw038>.
- [15] Kaoru Irie. Bayesian dynamic fused lasso. *arXiv preprint arXiv:1905.12275*, 2019.
- [16] Kyle Johnsen, AlissaW0921, and oliviaklemmer. Siplab-gt/wslfp: V0.2.1. Zenodo, March 2024. URL <https://zenodo.org/records/10819758>.
- [17] Kyle A Johnsen, Nathanael A Cruzado, Adam A Willats, and Christopher J Rozell. Cleo: A testbed for bridging model and experiment by simulating closed-loop stimulation, electrode recording, and optogenetics. *bioRxiv*, pages 2023–01, 2023.
- [18] Yuko Kakikawa, Kaito Shimamura, and Shuichi Kawano. Bayesian fused lasso modeling via horseshoe prior. *Japanese Journal of Statistics and Data Science*, 6(2):705–727, 2023.
- [19] Matthew T Kaufman, Mark M Churchland, and Krishna V Shenoy. The roles of monkey m1 neuron classes in movement preparation and execution. *Journal of neurophysiology*, 110(4):817–825, 2013.
- [20] Timothy Doyeon Kim, Thomas Zhihao Luo, Jonathan W. Pillow, and Carlos D. Brody. Inferring latent dynamics underlying neural population activity via neural differential equations. *Proceedings of the 38th International Conference on Machine Learning*, 2021.
- [21] Maksim Lapin, Matthias Hein, and Bernt Schiele. Loss functions for top-k error: Analysis and insights. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1468–1477, 2016.

- [22] A. H. Lara, J. P. Cunningham, and M. M. Churchland. Different population dynamics in the supplementary motor area and motor cortex during reaching. *Nature Communications*, 9(1):2754, July 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-05146-z. URL <https://www.nature.com/articles/s41467-018-05146-z>.
- [23] Hyun Dong Lee, Andrew Warrington, Joshua I Glaser, and Scott W Linderman. Switching autoregressive low-rank tensor models. *arXiv preprint arXiv:2306.03291*, 2023.
- [24] Qing Li and Nan Lin. The bayesian elastic net. 2010.
- [25] Scott Linderman, Benjamin Antin, David Zoltowski, and Joshua Glaser. SSM: Bayesian Learning and Inference for State Space Models, October 2020. URL <https://github.com/lindermanlab/ssm>.
- [26] Scott W Linderman, Andrew C Miller, Ryan P Adams, David M Blei, Liam Paninski, and Matthew J Johnson. Recurrent switching linear dynamical systems. *arXiv preprint arXiv:1610.08466*, 2016.
- [27] Yuxi Liu, Xian Long, Paul R Martin, Samuel G Solomon, and Pulin Gong. Lévy walk dynamics explain gamma burst patterns in primate cerebral cortex. *Communications Biology*, 4(1):739, 2021.
- [28] Jakob H Macke, Lars Buesing, John P Cunningham, Byron M Yu, Krishna V Shenoy, and Maneesh Sahani. Empirical models of spiking in neural populations. *Advances in neural information processing systems*, 24, 2011.
- [29] Alberto Mazzoni, Henrik Lindén, Hermann Cuntz, Anders Lansner, Stefano Panzeri, and Gaute T. Einevoll. Computing the Local Field Potential (LFP) from Integrate-and-Fire Network Models. *PLOS Computational Biology*, 11(12):e1004584, 2015. ISSN 1553-7358. doi: 10.1371/JOURNAL.PCBI.1004584. URL <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004584>.
- [30] Noga Mudrik, Yenho Chen, Eva Yezerets, Christopher J Rozell, and Adam S Charles. Decomposed linear dynamical systems (dlds) for learning the latent components of neural dynamics. *Journal of Machine Learning Research*, 25(59):1–44, 2024.
- [31] Nandakumar S Narayanan. Ramping activity is a cortical mechanism of temporal control of action. *Current opinion in behavioral sciences*, 8:226–230, 2016.
- [32] Josue Nassar, Scott W. Linderman, Monica Bugallo, and Il Memming Park. Tree-structured recurrent switching linear dynamical systems for multi-scale modeling. In *International Conference on Learning Representations (ICLR)*, 2019. URL <https://openreview.net/pdf?id=HkzRQhR9YX>.
- [33] Matthew R O’Shaughnessy, Mark A Davenport, and Christopher J Rozell. Sparse bayesian learning with dynamic filtering for inference of time-varying sparse signals. *IEEE Transactions on Signal Processing*, 68:388–403, 2019.
- [34] Chethan Pandarinath, Daniel J O’Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D Stavisky, Jonathan C Kao, Eric M Trautmann, Matthew T Kaufman, Stephen I Ryu, Leigh R Hochberg, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10):805–815, 2018.
- [35] Herbert E Rauch, F Tung, and Charlotte T Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8):1445–1450, 1965.
- [36] Maria Ribeiro and Miguel Castelo-Branco. Slow fluctuations in ongoing brain activity decrease in amplitude with ageing yet their impact on task-related evoked responses is dissociable from behavior. *Elife*, 11:e75722, 2022.
- [37] Michael E Rule, Carlos Vargas-Irwin, John P Donoghue, and Wilson Truccolo. Contribution of lfp dynamics to single-neuron spiking variability in motor cortex during movement execution. *Frontiers in systems neuroscience*, 9:89, 2015.

- [38] Steffen Schneider, Jin Hwa Lee, and Mackenzie Weygandt Mathis. Learnable latent embeddings for joint behavioural and neural analysis. *Nature*, 617(7960):360–368, 2023.
- [39] James M Shine, Patrick G Bissett, Peter T Bell, Oluwasanmi Koyejo, Joshua H Balsters, Krzysztof J Gorgolewski, Craig A Moodie, and Russell A Poldrack. The dynamics of functional brain networks: integrated network states during cognitive task performance. *Neuron*, 92(2): 544–554, 2016.
- [40] Chuanmeizhi Wang, Bijan Pesaran, and Maryam M Shanechi. Modeling multiscale causal interactions between spiking and field potential signals during behavior. *Journal of neural engineering*, 19(2):026001, 2022.
- [41] Leonhard Waschke, Niels A Kloosterman, Jonas Obleser, and Douglas D Garrett. Behavior needs neural variability. *Neuron*, 109(5):751–766, 2021.
- [42] Wei Wu, Michael J Black, David Mumford, Yun Gao, Elie Bienenstock, and John P Donoghue. Modeling and decoding motor cortical activity using a switching kalman filter. *IEEE transactions on biomedical engineering*, 51(6):933–942, 2004.
- [43] Andrew Zalesky, Alex Fornito, Luca Cocchi, Leonardo L Gollo, and Michael Breakspear. Time-resolved resting-state brain networks. *Proceedings of the National Academy of Sciences*, 111(28):10341–10346, 2014.
- [44] David M Zoltowski, Jonathan W Pillow, and Scott W Linderman. Unifying and generalizing models of neural dynamics during decision-making. *arXiv preprint arXiv:2001.04571*, 2020.

A Appendix / supplemental material

A.1 p-dLDS Algorithm

Algorithm 1 describes the proposed inference algorithm. In our experiments, we set $n = 1$ and $\eta = 10^{-4}$ and observe that the model converges. Below we use the notation hat notation for latent variable estimates or samples and the variable itself to represent the parameters of the variational distributions.

Algorithm 1 Variational EM for Probabilistic dLDS

Require: M observation dimension, N latent state dimension, K number of dynamic operators, S moving average window size, ξ SBL-DF trade-off parameter, n number of samples to estimate expectations, η sparsity threshold, θ model parameters.

// Initialize parameters

$\mathbf{c}_t \leftarrow \mathbf{0}$

$D_{i,j} \sim \mathcal{N}(0, \sigma^2)$

$f_{k,i,j} \sim \mathcal{N}(0, \sigma^2)$

$\hat{\mathbf{x}}_t \leftarrow D^+ \mathbf{y}_t$

▷ Initialize latent state with PCA

while ELBO has not converged **do**

// Update Latent State Posterior

$\hat{\mathbf{b}}_{1:T} \leftarrow \text{MovingAverage}_S(\hat{\mathbf{x}}_{1:T})$

$\hat{\mathbf{c}}_t \sim q(\mathbf{c}_t)$

$\hat{\mathbf{F}}_t \leftarrow \sum_{k=1}^K f_k \hat{\mathbf{c}}_{k,t}$

$\mathbf{l}_{1:T}, \Sigma_x \leftarrow \text{KalmanSmoother}(\mathbf{y}_{1:T}, \hat{\mathbf{b}}_{1:T}, \hat{\mathbf{F}}_{1:T}, \theta)$

// Update Coefficient Posterior

Initialize $q(\mathbf{c})$ and $q(\gamma)$ jointly with SBL-DF.

Update $q(c_{t,k})$ with SGD over equation (10) for densities where $|c_{t,k}| > \eta$.

Update $q(\gamma_t) \leftarrow \mathcal{IG}(\xi + \frac{n}{2}, \xi c_{t-1,k}^2 + \frac{\sum_{i=1}^n (\tilde{c}_{t,k,i} - c_{t,k})^2}{2})$

// Update Parameters

Update θ with SGD over equation (11).

end while

B Latent Variable Inference

B.1 Lemma 1 Derivation

Lemma 1. Let the transition between any two state vectors $\mathbf{x}_t, \mathbf{x}_{t+1} \in \mathbb{R}^N$ be defined by the linear dynamics matrix $\mathbf{F}_t \in \mathbb{R}^{N \times N}$ and the dynamics offset $\mathbf{b}_t \in \mathbb{R}^N$. For any $\lambda > 0$, the objective,

$$\arg \min_{\mathbf{F}_t, \mathbf{b}_t} \|\mathbf{x}_{t+1} - \mathbf{x}_t - \mathbf{F}_t \mathbf{x}_t - \mathbf{b}_t\|_2^2 + \lambda \|\mathbf{F}_t\|_2^2,$$

is minimized when $\mathbf{F}_t = \mathbf{0}$ and $\mathbf{b}_t = \mathbf{x}_{t+1} - \mathbf{x}_t$.

Proof. Let $\mathbf{r}_t = \mathbf{x}_{t+1} - \mathbf{x}_t$. We can rewrite the reconstruction objective in the following form,

$$\arg \min_{\mathbf{F}_t, \mathbf{b}_t} \|\mathbf{r}_t - \mathbf{F}_t \mathbf{x}_t - \mathbf{b}_t\|_2^2 + \lambda \|\mathbf{F}_t\|_2^2.$$

This objective is identical to the standard ridge regression with an unpenalized intercept term [13]. The solution is obtained by first centering the data, and then solving for the parameters using the solution for the standard Tikhonov regression. Below, we define the centered data as $\tilde{\mathbf{x}}_t$ and $\tilde{\mathbf{r}}_t$ for inputs and outputs respectively. Finally, we can use these values to obtain the following estimates of the parameters,

$$\hat{\mathbf{b}}_t = \boldsymbol{\mu}_t \quad \hat{\mathbf{F}}_t = (\tilde{\mathbf{x}}_t^\top \tilde{\mathbf{x}}_t + \lambda \mathbf{I})^{-1} \tilde{\mathbf{x}}_t^\top \tilde{\mathbf{r}}_t$$

However, when there is only a single datapoint, we get that $\tilde{\mathbf{x}}_t = 0$, which results in $\hat{\mathbf{F}} = 0$. \square

This result arises from having only a single observation for any dynamic transition, which leads to a singular design matrix. Although we can improve our estimate of \mathbf{F}_t by collecting more samples along a given trajectory, this is impractical when dealing with naturalistic time-series. For instance, it may be infeasible to collect more data from the exact same initial condition in a naturalistic environment due to noise in the experimental setup. In chaotic systems, minor deviations can lead to drastically different outcomes over long time horizons. Even if it were possible to precisely control for the initial condition of the signal, the presence of dynamical noise can cause initially aligned time series to quickly drift out of alignment. Consequently, it is not uncommon to observe a single transition between any two time points, as it is not guaranteed that events across multiple trials will be well-aligned.

B.2 Lemma 2 Derivation

Lemma 2. Let $\mathbf{l}, \mathbf{b} \in \mathbb{R}^N$ be independent random variables such that $\mathbf{l} \sim p(\mathbf{l})$ and $\mathbf{b} \sim p(\mathbf{b})$. Their sum $\mathbf{x} = \mathbf{l} + \mathbf{b}$ is distributed according to $\mathcal{N}(\boldsymbol{\mu}_l + \boldsymbol{\mu}_b, \boldsymbol{\Sigma}_l + \boldsymbol{\Sigma}_b)$ when 1) $p(\mathbf{b}) = \mathcal{N}(\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b)$ and $p(\mathbf{l}) = \mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)$ and when 2) $p(\mathbf{b}) = \delta(\mathbf{b} - \boldsymbol{\mu}_b)$ and $p(\mathbf{l}) = \mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l + \boldsymbol{\Sigma}_b)$.

Proof. Case 1. Let $\mathbf{l} \sim \mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)$ and $\mathbf{b} \sim \mathcal{N}(\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b)$. The sum of normal random variables follows a distribution that results from convolving their individual distributions,

$$\begin{aligned} q(\mathbf{x}) &= q(\mathbf{l} + \mathbf{b}) \\ &= q(\mathbf{l}) * q(\mathbf{b}) \\ &= \mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l) * \mathcal{N}(\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b) \\ &= \mathcal{N}(\boldsymbol{\mu}_l + \boldsymbol{\mu}_b, \boldsymbol{\Sigma}_l + \boldsymbol{\Sigma}_b) \end{aligned}$$

This is a standard result from probability theory.

Case 2. Now let $\mathbf{l} \sim \mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l + \boldsymbol{\Sigma}_b)$ and $\mathbf{b} \sim \delta(\mathbf{b} - \boldsymbol{\mu}_b)$. Similarly, the distribution of the sum of these variables is distributed according to their convolution,

$$\begin{aligned} q(\mathbf{x}) &= q(\mathbf{l}) * q(\mathbf{b}) \\ &= \mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l + \boldsymbol{\Sigma}_b) * \delta(\mathbf{b} - \boldsymbol{\mu}_b) \\ &= \int_{-\infty}^{\infty} \mathcal{N}(\mathbf{x} - \boldsymbol{\tau}; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l + \boldsymbol{\Sigma}_b) \delta(\boldsymbol{\tau} - \boldsymbol{\mu}_b) d\boldsymbol{\tau} \\ &= \mathcal{N}(\mathbf{x} + \boldsymbol{\mu}_b; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l + \boldsymbol{\Sigma}_b) \\ &= \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_l + \boldsymbol{\mu}_b, \boldsymbol{\Sigma}_l + \boldsymbol{\Sigma}_b), \end{aligned}$$

where the fourth line is the result of the sifting property of delta distributions. Since the final distribution in Case 1 and Case 2 are identical, we complete the proof. \square

B.3 Optimal $q(\mathbf{x})$ Update

The optimal coordinate ascent variational update is given by the following equation,

$$\begin{aligned} \log q^*(\mathbf{x}) &\propto \mathbb{E}_{q(\mathbf{c}, \boldsymbol{\gamma})} [\log p(\mathbf{x}, \mathbf{c}, \mathbf{y}, \boldsymbol{\gamma} | \boldsymbol{\theta})] \\ &= \mathbb{E}_{q(\mathbf{c}, \boldsymbol{\gamma})} [\log p(\mathbf{l}_1 | \boldsymbol{\theta}) + \sum_{t=2}^T \log p(\mathbf{l}_t | \mathbf{l}_{t-1}, \mathbf{c}_t, \boldsymbol{\theta}) + \sum_{t=1}^T \log p(\mathbf{y}_t | \mathbf{l}_t + \mathbf{b}_t, \boldsymbol{\theta})] + C. \end{aligned} \quad (12)$$

Conditioned on estimates of $\mathbf{b}_{1:T}$ and samples of $\mathbf{c}_{1:T}$, the factor graph of equation (12) corresponds exactly to a time-varying Linear Gaussian State Space Model. Thus we can leverage the efficient inference algorithms such as the Kalman filter and RTS smoother when computing the marginals of the variational distribution of $\mathbf{l}_{1:T}$.

C Generating Synthetic Examples

C.1 Noisy NASCAR

NASCAR data is generated by partitioning the two-dimensional state space into four regions according to the rules,

$$Z(\mathbf{x}) = \begin{cases} 1, & x_1 > 1 \\ 2, & x_1 < -1 \\ 3, & -1 \leq x_1 \leq 1, x_2 \geq 0 \\ 4, & -1 \leq x_1 \leq 1, x_2 < 0, \end{cases}$$

where $Z(\mathbf{x})$ is the ground truth switching state function that depends on the particular location \mathbf{x} . The ground truth dynamics matrices are defined as,

$$\mathbf{A}(\mathbf{x}) = \begin{cases} \begin{bmatrix} 0 & 0.1 \\ -0.1 & 0 \end{bmatrix}, & \text{when } Z(\mathbf{x}) = 1 \text{ or } 2 \\ \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, & \text{when } Z(\mathbf{x}) = 3 \text{ or } 4, \end{cases}$$

and ground truth offsets are defined as,

$$\mathbf{b}(\mathbf{x}) = \begin{cases} [0 \ 0.005]^\top, & \text{when } Z(\mathbf{x}) = 1 \\ [0 \ -0.005]^\top, & \text{when } Z(\mathbf{x}) = 2 \\ [0.1 \ 0]^\top, & \text{when } Z(\mathbf{x}) = 3 \\ [-0.1 \ 0]^\top, & \text{when } Z(\mathbf{x}) = 4. \end{cases}$$

Given the current location in state space \mathbf{x}_t , we can transition to the next point using the continuous time dynamics equation

$$\mathbf{x}_t = \expm(\tau \mathbf{A}_{Z(\mathbf{x}_t)}) \mathbf{x}_{t-1} + \tau \mathbf{b}_{Z(\mathbf{x}_t)} + \boldsymbol{\nu}_t,$$

where each entry of the process noise is sampled from $\nu_{t,i} \sim \mathcal{N}(0, 10^{-4})$. To modulate the speed of the system, we uniformly sample a speed constant $\tau \in [0.1, 1]$, which is applied throughout each segment of the track. We use the continuous time formulation over the discrete-time formulation to ensure that changes to the speed do not distort the shape of the original system's state space. To generate noisy observations, we construct a linear emissions matrix with random variables such that each entry is given by $D_{i,j} \sim \mathcal{N}(0, 1)$.

C.2 Ramping Lorenz

In order to modulate the speed of the Lorenz system, we adjust the evaluation time points of an ODE integrator, specifically Runge-Kutta of the order 5(4) (RK54) as implemented in scipy's `solve_ivp` [9]. Ramping activity is generated randomly with the following procedure,

1. Uniformly sample an evaluation interval length $\tau \in [0.25, 1.5]$.
2. Construct a vector \tilde{T} that consists n evenly spaced numbers over the interval $[0, \tau]$. In our experiments, we set n to be 100.
3. Perform the transformation $\exp(\tilde{T}) - 1$ to obtain a vector of ramped evaluation times.
4. Plug in the transformed evaluation times into the RK45 Solver to obtain latent trajectories.

Similar to the NASCAR experiment, we generate noisy observation from a randomly constructed linear emissions matrix such that each entry is given by $D_{i,j} \sim \mathcal{N}(0, 1)$.

C.3 Simulated Monkey Reaching Task

Our dataset is constructed from publicly available data and code from the center-out reach task in [22, 8]. We obtain latent factors from spiking networks that are trained to reproduce empirically measured EMG signals, given a 3-dimensional input that specifies the go input and the reach angle. In our experiments, these factors are considered ground truth. Our trained factor-based spiking network then generates spiking activity for 1200 neurons. Synaptic currents are used as inputs into the Weighted Sum of synaptic currents LFP proxy method (WSLFP) [29], as implemented in the `wslfp` Python package [17, 16]. As WSLFP is a function of the relative location of neurons and electrodes, we place neurons randomly within a 5 mm by 10 mm by 1 mm region and electrodes in a grid centered in this region. The result is a multi-channel LFP dataset with nonlinear dynamics and measurements characteristic of systems neuroscience.

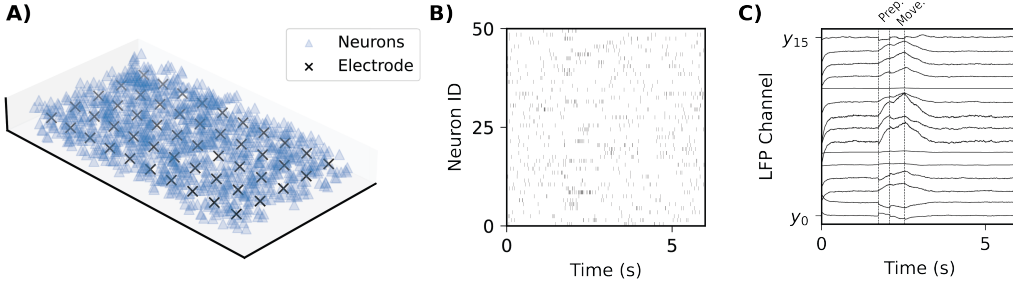


Figure 5: **Empirically-Derived Reach Experiment.** (A) 1,200 neurons are randomly placed into a 5 mm by 10 mm by 1 mm region. Electrodes are placed in a grid centered in this region (B) Spiking activity for a subset of neurons in an example trial produced from a factor-based spiking network. (C) First 15 channels in a simulated multi-channel LFP recording. Preparatory and Movement phases are marked by the dotted lines.

D Evaluation Metrics

D.1 Multi-step Inference

The multi-step inference performance is computed with the following R-squared metric,

$$R_k^2 = 1 - \frac{\sum_{t=0}^{T-k} \|\mathbf{y}_{t+k} - \hat{\mathbf{y}}_{t+k}\|_2^2}{\sum_{t=0}^{T-k} \|\mathbf{y}_{t+k} - \bar{\mathbf{y}}\|_2^2}, \quad (13)$$

where k is the number of steps from the initial condition, $\bar{\mathbf{y}}$ is the mean estimator for each trajectory and $\hat{\mathbf{y}}_{t+k}$ is the model prediction after applying the inferred dynamics for k steps. When testing, model parameters such as the dynamics and observation matrices are frozen, while specific latent variables are estimated based on the held-out data. In Table 1, we show results for $k = 100$.

D.2 Inferred Dynamics Error

We measure the accuracy of the latent dynamics with the mean squared error (MSE) of the inferred speed, defined as,

$$\text{MSE}_{\text{speed}} = \frac{1}{T-1} \sum_{t=1}^{T-1} \|\dot{\mathbf{x}}_t - U\hat{\mathbf{x}}_t\|_2^2, \quad (14)$$

where the true speed $\dot{\mathbf{x}}_t = \mathbf{x}_{t+1} - \mathbf{x}_t$ is computed from the denoised ground truth latent state, and the predicted speed $\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t$ is computed using the model's 1-step prediction. Since latent trajectories are only identifiable up to a linear transformation, we align the inferred trajectories with the true trajectories using a least squares fit before computing this score. More specifically, we

find the optimal linear transformation $U \in \mathbb{R}^{N \times N}$ between the estimated and true states across all trajectories by solving,

$$\hat{U} = \arg \min_U \frac{1}{T} \sum_{t=1}^T \|\mathbf{x}_t - U \hat{\mathbf{x}}_t\|. \quad (15)$$

D.3 Inferred Latent State Space Error

Similarly, we measure the accuracy of the latent state space by computing the MSE after a linear alignment between trajectories from the inferred and true state space. We use this metric only for the reaching example, since the true observation function is a complex nonlinear function,

$$\text{MSE}_{\text{state}} = \frac{1}{T} \sum_{t=1}^T \|\mathbf{x}_t - U \hat{\mathbf{x}}_t\|_2^2. \quad (16)$$

The linear alignment $U \in \mathbb{R}^{N \times N}$ between the estimated and true states across all trajectories is computed by solving the least squares problem in equation (15).

D.4 Inferred switching rate error

Evaluating the accuracy of the switching behavior is a more difficult task. In fact, developing a procedure that matches predicted switch times with true switch times can lead to a complicated optimal transport procedure. To simplify the evaluation of switching times, we marginalize over time, and compare only the MSE of the switch rate defined as,

$$\text{MSE}_{\text{switch}} = \frac{1}{m} \sum_{i=1}^m \|r_i - \hat{r}_i\|_2^2, \quad (17)$$

where m is the number of trials, r_i is the true switch rate for the i th trajectory, and $\hat{r}_i = \frac{1}{T} \sum_{t=1}^T \mathbf{1}\{z_t \neq z_{t-1}\}$ is the predicted switch rate. Intuitively, \hat{r}_i is the number of times that the state or dominant DO changes between consecutive time points normalized by the length of the interval T . In switching models, switch events are defined as a time point where the current inferred dynamical state differs from the state in the previous time step. Similarly in decomposed models, switch events are defined as time points where the active set of DOs change from the previous time step.

In the NASCAR example, r_i is defined with the number of transitions between ground truth segments. In the Lorenz example, r_i is defined by the number of times that the trajectory switches between the two lobes in addition to the number of ramping periods.

D.5 Reaching Classification Accuracy

We quantitatively evaluate the reaching experiment with a classification task. Here, we want to determine whether the learned systems can be used to distinguish between different reach directions. Recall that switched models infer a switching variable for each time point where $z_t \in \{1, \dots, K\}$ while decomposed models infer a coefficient vector $\mathbf{c}_t \in \mathbb{R}^K$. Rather than viewing z_t as an index, we can equivalently view it as a one-hot encoded vector $z_t \in \{0, 1\}^K$ which describes whether a particular switching state is active at any given time. This matches the dimensionality of the variables in both switched and decomposed systems.

For simplicity, we focus on linear logistic regression classifiers in our experiment. If we let the inputs be z_t and \mathbf{c}_t directly, then our classifiers quickly overfits since there are many more input features than trials. Specifically, the number of features scales linearly with the number of time points and systems $\mathcal{O}(TK)$. Instead, we marginalize over time and compute features from the estimated latent variables by averaging state activity over time. In switched models, this is the average one-hot encoding value over time. Similarly, this is the average coefficient value in decomposed models. However, for each dynamical state, we compute separate features for positive and negative coefficient values to prevent interference between them. In this setup, the input (feature) dimensionality scales according to $\mathcal{O}(K)$ while the output dimensionality of the linear classifiers are the reaching directions. For all

classifiers, we perform a grid search over the values $\{10^i\}_{i=-4}^4$ to identify an appropriate amount of L2 regularization. Top-k accuracies are a standard metric in machine learning [21, 3] and computed using the estimated class probabilities from the logistic regression classifier.

E Additional Results

E.1 Synthetic Dynamical Systems

Figure 6A demonstrates that our inference procedure converges to a local optimum while Figure 6B shows a full sweep of the multi-step inference metric. Tables 1 in the main paper reports the final value. For completeness, we include Tables 3 and 4 which reports the means across 5 seeds of each model, and includes the standard deviations in parenthesis.

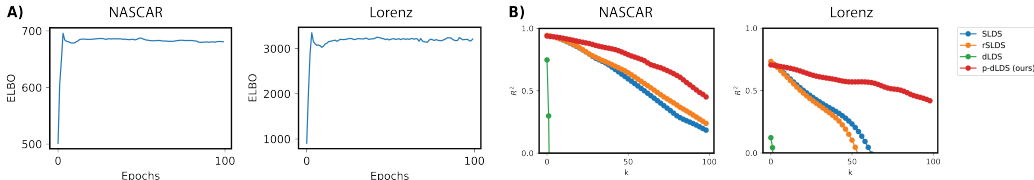


Figure 6: (A) ELBO converges in both synthetic dynamical systems. (B) Multi-step inference where k represents the number of steps. Tables 3 and 4 report the final values.

Table 3: Metrics for NASCAR. Bold means best performance. (\uparrow) indicates higher score is better while (\downarrow) indicates that lower is better. \times indicates that value diverged towards $-\infty$. All MSE values are $\times 10^{-3}$ while R^2 values are not scaled. We report means across 5 seeds and include standard deviation in parenthesis.

Model	Speed MSE (\downarrow)	Switch MSE (\downarrow)	100-step R^2 (\uparrow)
SLDS	0.0995 (0.021)	12.89 (1.30)	0.184 (0.024)
rSLDS	0.1065 (0.024)	13.17 (2.84)	0.238 (0.022)
dLDS	123.19 (23.13)	13.28 (5.31)	\times
p-dLDS (ours)	0.033 (0.009)	7.34 (3.40)	0.450 (0.027)

Table 4: Metrics for Lorenz. Bold means best performance. (\uparrow) indicates higher score is better while (\downarrow) indicates that lower is better. \times indicates that value diverged towards $-\infty$. We report means across 5 seeds and include standard deviation in parenthesis.

Model	Speed MSE (\downarrow)	Switch MSE (\downarrow)	100-step R^2 (\uparrow)
SLDS	0.431 (0.233)	0.0204 (0.007)	-3.47 (1.052)
rSLDS	0.304 (0.040)	0.0208 (0.004)	-11.54 (1.353)
dLDS	1.123 (0.089)	0.1529 (0.070)	\times
p-dLDS (ours)	0.141 (0.015)	0.0137 (0.014)	0.418 (0.079)

E.2 Reaching Task

For each model, we visualize the trial-averaged dynamic regime activity of each reach direction (Fig. 7). In SLDS, this is visualized by considering the discrete states as a one-hot vector over time. When a dynamic regime is active, that state will have a value of 1 while the unactive states will have a value of 0. Thus the trial-averaged value of each state must have a value in the interval $[0, 1]$. In dLDS, we plot the inferred coefficient value without any modification.

Although SLDS correctly identifies preparatory and movement phases using states 4 and 3 respectively, it fails to differentiate dynamics occurring outside of these expected phases, incorrectly grouping

unrelated regions together. Furthermore, the discrete formulation produces very similar activity patterns across all reach angles, obscuring any differences that are present. In dLDS, we observe that the features change smoothly and cyclically with the reach angle. However, the dynamic operator activity do not localize to the preparatory and movement phases due to a limited inference procedure.

Table 5: Inference performance for the reaching experiment (see Figure 3) on a held-out test set. Top-1 and Top-3 accuracies are obtained by predicting reach directions from latent variable features using linear classifiers. State and Dynamics MSE are computed with respect to true latent variables. We report standard deviations in parenthesis across 5 seeds.

Model	Top-1 Acc.	Top-3 Acc.	State MSE ($\times 10^{-1}$)	Dynamics MSE ($\times 10^{-2}$)
SLDS	38.46 (2.84)	57.69 (7.53)	0.5289 (0.13)	0.3942 (0.23)
rSLDS	12.82 (3.05)	32.05 (8.31)	0.5503 (0.23)	292.41 (13.96)
dLDS	10.25 (5.97)	39.74 (10.29)	0.6742 (0.52)	35.680 (5.76)
pdLDS (ours)	42.31 (3.50)	70.51 (6.45)	0.4061 (0.38)	0.0567 (0.04)

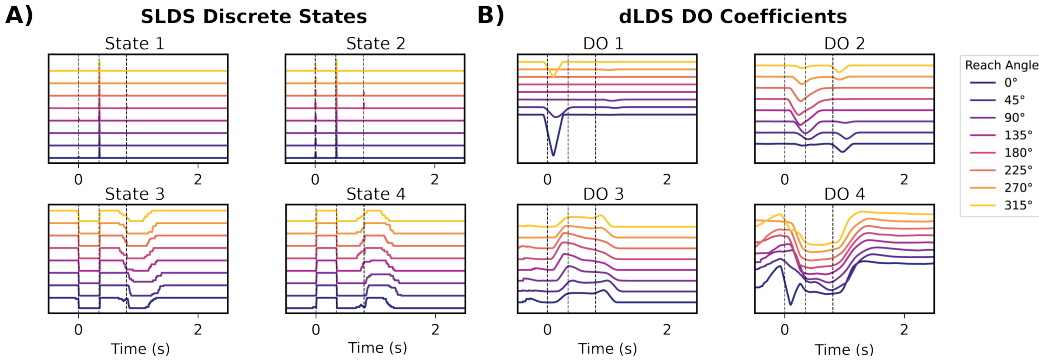


Figure 7: Trial-averaged activity for (A) SLDS discrete states and (B) dLDS DO coefficients for each reach angle. The preparatory and movement phases occur between the dashed lines similar to Figure 3. Time 0 represents the onset of the stimulus.

F Experimental Setup

F.1 Hyperparameter Settings

For switching models, we rely on the `ssm` package which allows for efficient Bayesian inference for a variety of state space models [25]. We set the variational posterior to `structured_meanfield`, and the fitting procedure to `laplace_em` as recommended by the developers. Additionally, we set the distributional form of the dynamics and emissions matrices to Gaussian.

The hyperparameters of dLDS primarily consists of the lagrange multipliers in the BPDN-DF objective including $\lambda_0, \lambda_1, \lambda_2$. We find the optimal value of these hyperparameters using a random search with a fixed budget of 1000 evaluations. For each hyperparameter, we uniformly sample over the log of the interval $[10^{-3}, 10^3]$ and evaluate it against the BPDN-DF objective. For the NASCAR experiment, we found that $\lambda_0 = 1.044, \lambda_1 = 0.254,$ and $\lambda_2 = 0.023$ resulted in the best performance. For the Lorenz experiment, we found that $\lambda_0 = 0.628, \lambda_1 = 2.010,$ and $\lambda_2 = 0.0124$ yielded the best performance.

For p-dLDS, the relevant hyperparameters consists of the SBL-DF dynamics tradeoff ξ , and the offset window size S . We use random search with a budget of 1000 samples to determine the values of S and ξ and fit a separate model for each set of hyperparameters. In the NASCAR experiment, we isolate the effect of the probabilistic inference procedure by setting $S = T$, removing the influence of the time-varying offset term. For ξ , we perform a random search by uniformly sample over the log of the interval $[10^{-3}, 10^3]$ and found that $\xi = 0.945$ was optimal. For the Lorenz experiment, we also optimize for the window size S by uniformly sample a discrete index on the interval $\{2, \dots, T\}$.

For the Lorenz experiment, the optimized hyperparameters are $S = 85$ and $\xi = 8.928$. For the real dataset, the optimal offset is $S = 76$ which is smaller than the timescale of p-dLDS coefficient switching (around 150 time points), suggesting that the same DO dynamics may persist even as the fixed points of the system fluctuates throughout the experiment.

F.2 Hardware Specification

We perform hyperparameter sweep on our institution’s HPC cluster using small-scale CPU resources which consists of Dual Intel Xeon Gold 6226 CPUs. Once hyperparameters have been optimized, it is possible to run each experiment within approximately 2 hours on the 2020 edition of the M1 Macbook Pro.

G Description of Clinical Neurophysiology Data

Data was collected as part of a study investigating deep brain stimulation for treatment-resistant depression (TRD). The study is pre-registered in ClinicalTrials.gov (identifier NCT04106466). The study protocol was approved by the IRB (identifier IRB00066843). Informed consent was obtained from participants before participation in the trial. Patients receive no monetary compensation, but instead have their DBS electrodes and Summit RC+S IPG device provided free of charge. The analysis focused on LFP signals from a single participant with all personally identifiable information removed.

H Limitations

While our proposed method demonstrates strong performance in our experiments, there are many limitations. For instance, our approach does not have a strong mechanism for generating future unseen coefficients. Our assumed coefficient transition model is primarily motivated by our desire to obtain smooth coefficients over time. However, we believe that they may be more complex transition models that can both capture persistent activity in challenging systems while also being an accurate forecaster, such as a deep learning based transition model. Another limitation of our approach is that our method assumes smoothness in the latent space. However, we do not explore the possibility of having sparse structure in the latent space which can be easily accomplished in BPDN-DF by adding an L1 penalty over x .

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: We propose a probabilistic treatment and an extended dynamics formulation in decomposed models (Section 3). We demonstrate that these changes reduce estimation errors and finds coherent structure where previous models fail in many challenging synthetic examples, and a noisy real-world example (Section 4).

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: A limitations section is provided in Appendix H due to space constraints.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We provide proofs of our lemmas in B.1 and B.2.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide synthetic data details in Appendix C, metric definitions in Appendix D, and hyperparameter details in Appendix F.1. Moreover, we release our code with our submission.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide functions to generate the synthetic datasets from our experiments. Unfortunately, we are unable to release the neurophysiological dataset due to request from our collaborators.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We specify that all datasets are split 50:50 for train and test in the main paper. Moreover, we provide optimizer and hyperparameter settings in Appendix F.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We provide standard deviations in Appendix E.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide details about hardware in Appendix F.2.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: Most of our datasets are synthetically generated or derived from publicly available sources, which we refer to throughout the paper. For our real-world experiment, we include the IRB identifier.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: Our proposed model aims to reveal coherent patterns in time-series data, serving as a scientific tool similar to PCA. As a probabilistic technique not designed for content generation or automated decision-making, we do not anticipate direct societal impacts.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our method does not generate synthetic content that is at risk of abuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification:

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: We include details on where we obtained our data (either synthetically generated, or part of an existing dataset) in the main text and appendix. We also point to the commonly package used for SLDS and rSLDS in the appendix.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[Yes\]](#)

Justification: We provide the IRB identifier in Appendix G which includes the informed consent form with experiment details.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[Yes\]](#)

Justification: We include the IRB identifier number in Appendix G, but removed any information related to the institution conducting the experiments.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.