# Behavior-dLDS: A decomposed linear dynamical systems model for neural activity partially constrained by behavior

**Anonymous authors**
**Paper under double-blind review**

## Abstract

Brain-wide recordings of large-scale networks of neurons now provide an unprecedented view into how the brain drives behavior. However, brain activity contains both information directly related to behavior as well as the potential for many internal computations. Moreover, observable behavior is executed not only by the brain, but also by the spinal cord and peripheral nervous system. Behavior is a coarse-grained product of neural activity, and we thus take the view that it can be best represented by lower-dimensional latent neural dynamics. Capturing this indirect relationship while disambiguating behavior-generating networks from internal computations running in parallel requires new modeling approaches that can embody the parallel and distributed nature of large-scale neural populations. We thus present behavior-decomposed linear dynamical systems (b-dLDS) to disentangle simultaneously recorded subsystems and identify how the latent neural subsystems relate to behavior. We demonstrate the ability of b-dLDS to decouple behavioral vs. internal computations on controlled, simulated data, showing improvements over a state-of-the-art model that uses behavior to supervise all dynamics based on behavior. We then show that b-dLDS can further scale up to tens of thousands of neurons by applying our model to a large-scale recording of a zebrafish hindbrain during the complex positional homeostasis behavior, wherein b-dLDS highlights behavior-related dynamic connectivity networks.

## 1 Introduction

Brains are instrumental in generating and controlling behavior. However, the neural activity that ultimately generates behavior is complex and can be nonlinearly or indirectly related to behavior (Urai et al., 2021). These neural processes may not be time-aligned with behavior (Bondy et al., 2025), and are distributed and overlapping or even compositional in time and space in the brain; at the same time, not every behavior is encoded equally in every subnetwork of neurons (Meshulam et al., 2025; Kashefi et al., 2025; Affan & Scott, 2022; Mu et al., 2020; Wang et al.; Mendoza-Halliday et al., 2024; Chen et al., 2024a). Moreover, these functions may evolve throughout a task (Aoi et al., 2020) and may occur on different timescales (Shi et al., 2025). On the other hand, this neural activity also encodes many other functions such as hunger and circadian rhythms, task-irrelevant sensory information, higher-order cognitive functions, and emotional responses. As neural recording technologies continue to grow and evolve (Steinmetz et al., 2021; Demas et al., 2021; Yang et al., 2024), the extent of the brain that is captured is larger and more likely to include these many parallel functions that need to be disentangled in order to study the activity underlying the behavior of interest (Urai et al., 2021).

Models that aim to identify meaningful temporal components of neural activity can be behavior-agnostic or behavior-aware. Behavior-agnostic models (Sussillo et al., 2016; Linderman et al., 2016; Mudrik et al., 2024a) use only the neural activity to learn components that can then be correlated *post hoc* to other known information, such as behavior, to "translate" the model variables to address task-relevant questions, such as "Does this model component turn on when the animal enters a reward-seeking state?" However, behavior-aware models can more clearly align learned model components with known behavior variables. Behavior-aware models can also come in two flavors: models that directly relate instantaneous neural activity to instantaneous behavior in a shared latent space (Schneider et al., 2023; Sani et al., 2021a), or models that connect the latent dynamics to behavior (Geadah et al., 2025). This last class of models is important because it accounts for the fact that the relationship between neural activity and behavior

is complex and indirect as described above. Instead, the latent dynamics capture the simultaneous coarse-grained processes the brain "runs," some or all of which may be related to behavior (Karigo & Charles, 2025).

More specific to neural dynamics, existing models live at the two extremes of completely supervised, or completely unsupervised. At the former end of the spectrum are models such as dynamic causal modeling (DCM), and the more recent conditionally linear dynamical systems (CLDS), which consider a linear decomposition of the neural dynamics wherein the coefficients of the decomposition are completely a function of external or experimental variables (e.g., behavior) (Geadah et al., 2025; Friston et al., 2003). At the other end are models such as decomposed linear dynamical systems (dLDS) which, while they do similarly decompose the dynamics into a linear function of basis dynamics, do not tie the dynamics to external variables at all (Mudrik et al., 2024a; Yezerets et al., 2025; Chen et al., 2024b; Mudrik et al., 2025). Instead, the dynamics operators (DOs)—which are learned from the data itself—are assumed to be sparsely represented in the basis, forming a dictionary of dynamics that induces temporal independence of when the different dynamics components are used over time.

Here we present a much needed middle ground between these two approaches: behavior-dLDS (b-dLDS), a model that explicitly ties a small subset of DOs to behavior variables over time (Fig. 1). Since the DOs are the fundamental processes, including behavior, that the brain "runs," and the dynamics coefficients describe which DOs are active at a given time, we relate the behavior traces to the dynamics coefficient traces, where the latent neural dynamics generate behavior and other functions (arrow from $\boldsymbol{c}$ to $\boldsymbol{b}$ in Fig. 1).

b-dLDS is comparable to other works (Geadah et al., 2025; Friston et al., 2003) that use external inputs to condition the model-estimated neural dynamics, based on the assumption that the latent dynamics generate all the neural activity, and the neural activity generates the behavior. The dynamics are thus the best coarse-grained representation of the true underlying processes at play. However, b-dLDS is unique in that it achieves this by requiring the dynamics coefficients to reconstruct the behavior traces via an additional regularization term, rather than conditioning the dynamics directly on the external inputs (e.g., behavior).

Moreover, a Frobenius norm on the mapping from dynamics coefficients to external behavior data in b-dLDS means that it is possible to bias the model to learn that this relationship is sparse. This means that some DOs generate behavior, while the rest of the DOs can generate other brain processes, such as decision-making, homeostasis, etc.

In this paper we introduce the b-dLDS model and demonstrate its efficacy in distinguishing between latent neural dynamics that generate behavior vs. other internal computations. We present new regularization terms on the inference and update steps to enable mapping to behavior and to encourage sparsity in the dynamics-behavior mapping. We then demonstrate through simulations that not only can b-dLDS learn a sparse mapping between behavior and dynamics coefficients, but in fact this sparse mapping makes the dynamics it learns more accurate than those learned by CLDS, a model strongly conditioned on behavior. Finally, we present an example of b-dLDS applied to zebrafish hindbrain data, which encodes information in its dynamics related to swimming (the observable behavior) as well as internal computations related to how much to respond to displacement in a current (Yang et al., 2022). b-dLDS can identify dynamics that align with behavior and those that do not.

## 2 Background and Related Work

In neuroscience, a variety of methods have been used to deal with the high dimensionality and complexity of brain-wide or large neural population data and its evolution in time. Dimensionality reduction-only approaches, e.g., principal component analysis (PCA), non-negative matrix factorization (NMF), and independent component analysis (ICA), are unsupervised approaches that enable the remapping of data to dimensions that can be correlated to known features of the experiment or system *post hoc*. More complex approaches, such as Similarity-driven Building-Block Inference using Graphs across States (SiBBlInGS) (Mudrik et al., 2024b) and model-based targeted dimensionality reduction (mTDR) (Aoi et al., 2020), incorporate more supervision from external data, e.g., behavioral task variables. However, matrix factorization-based methods cannot capture complex temporal interactions intrinsic to neural computation.

Generalized linear models (GLMs) fit coefficients that describe explicit relationships between time series data and known external factors (Park et al., 2014; Ashwood et al., 2022). However, GLMs model statistical relationships that are often assumed static, rather than accounting for changing relationships over time, e.g., due to changes in the animal's state or adaptation. Hidden Markov Model GLMs (HMM-GLMs) were thus introduced to learn discrete

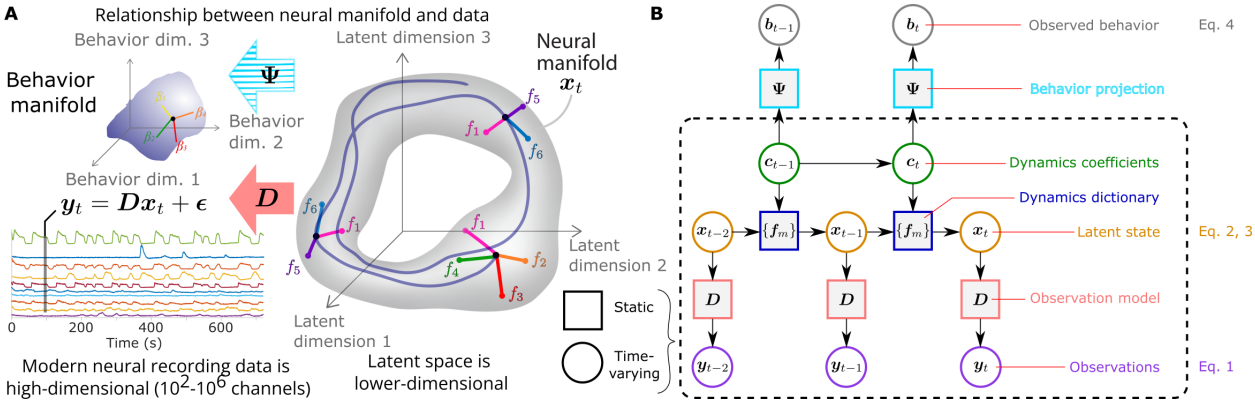



Figure 1: **behavior-dLDS architecture. A:** The latent, lower-dimensional neural manifold represents the processes that generate observable neural activity, mapped to the ambient space via $\boldsymbol{D}$, and behavior, via $\boldsymbol{\Psi}$. **B:** The established dLDS model, inside the dashed lines, is updated here with a mapping from dynamics coefficients $\boldsymbol{c}$ to behavior $\boldsymbol{b}$ via $\boldsymbol{\Psi}$. This contributes another regularization term to the joint inference of latent states $\boldsymbol{x}$ and dynamics coefficients $\boldsymbol{c}$.

system states that toggle between different GLM models. However, HMM-GLMs, as with GLMS, still model the system as a single, monolithic circuit, and do not identify independent subsystems.

Latent dynamical systems model higher-dimensional, potentially noisy data as being generated by a lower-dimensional system of latent variables (Pandarinath et al., 2018a). Dynamical systems can model the relationship between internal states and other external factors, e.g., behavior. Finally, while often assumed to be static, latent dynamics may also evolve over time (nonstationarity) when the system itself is malleable (e.g., adaptation, learning, or representational drift in neuroscience).

**Dynamical systems in neuroscience:** In neuroscience, dynamical systems models have included both linear and nonlinear variants. Most such models (including linear (Sani et al., 2021a; Golub et al., 2013; Churchland et al., 2012); nonlinear (Pandarinath et al., 2018b; Keshtkaran & Pandarinath, 2019; Sussillo et al., 2015; Sani et al., 2021b; Kleinman et al., 2021); and switching (Ghahramani & Hinton, 1996; Murphy, 1998; Fox et al., 2008; Linderman et al., 2017; Nassar et al., 2018) variants) treat all the data as coming from a single system, similar to the GLM. Mathematically, if the $N$-dimensional system state at time $t$ is $\boldsymbol{x}_t \in \mathbb{R}^N$, then $\boldsymbol{x}_t \approx g(\boldsymbol{x}_{t-1})$ for some function $g : \mathbb{R}^N \to \mathbb{R}^N$ that may be nonlinear or even time-varying.

More recently, the need to model larger-scale recordings that contain multiple biological subsystems has required new models that can identify these meaningful subsystems by decomposing the dynamics into combinations of systems (Friston et al., 2003; Mudrik et al., 2024a; Yezerets et al., 2025; Chen et al., 2024b; Mudrik et al., 2025; Geadah et al., 2025). In these models, the dynamics are linear at each time point, i.e., $\boldsymbol{x}_t \approx \boldsymbol{F}_t \boldsymbol{x}_{t-1}$ for a dynamics transition matrix $\boldsymbol{F}_t \in \mathbb{R}^{N \times N}$. Each $\boldsymbol{F}_t$ can be expanded into a basis as $\boldsymbol{F}_t = \sum_{m=1}^{M} \boldsymbol{f}_m c_{mt}$, where the basis elements $\boldsymbol{f}_m \in \mathbb{R}^{N \times N}$ do not change over time, however the expansion coefficients (called dynamics coefficients) $c_{mt}$ do. By varying the dynamics coefficients, this model class can capture nonlinear and nonstationary behavior, providing a flexible descriptive framework while maintaining a local sense of interpretability through the linear formulation at each time point.

Two classes of such decomposed linear dynamics models have emerged. On one end is the work in dynamic causal modeling (DCM) (Friston et al., 2003) and conditionally linear dynamical systems (CLDS) (Geadah et al., 2025), wherein the dynamics coefficients $c_{mt}$ are functions of external or experimental variables (e.g., behavioral quantities). In this case the basis for the dynamics is typically fixed, and it is the mapping from behavioral variables $\boldsymbol{b}_t$ to dynamics coefficients $\boldsymbol{c}_t$ that is learned. On the other end is the completely unsupervised decomposed linear dynamical systems (dLDS) model (Mudrik et al., 2024a; Yezerets et al., 2025) and extensions thereof (Mudrik et al., 2025; Chen et al., 2024b), which take a completely unsupervised approach to learning the dynamics basis (or dictionary) from data. In this setting the $c_{mt}$ are assumed unknown latent variables with sparse statistics. Sparsity in this case encourages the learned basis to be statistically in terms of when they are active, i.e., they truly represent "different" processes. This model can be optimized via an expectation-maximization (EM) procedure, which results in a dictionary-learning-like iterative algorithm that alternates between fitting the dynamics coefficients and the dynamics basis elements.

**Other models of brain and behavior:** Building on the advances in dynamical systems, further work has sought to devise joint models of neural activity and behavior. In the Preferential Subspace Identification (PSID) algorithm (Sani et al., 2021a), behavior and neural activity come from a shared latent state space, which is split into neural- and behavior-related subspaces. In other words, the underlying state generating neural activity and behavior is assumed to be the same at each time point; however, this may not necessarily be true if there is a hierarchical relationship between the latent dynamical systems, the behaviors they represent, and neural activity they generate. It is possible that only a subset of the dynamics represent behavior-related processes, while another subset relate to internal cognitive variables not observed in the external input. Moreover, in the underdetermined case where there are more neurons than time points, PSID cannot compute the matrix inverse. In order to model such data, one is obliged first to perform dimensionality reduction such as PCA, which runs counter to the aims of b-dLDS.

We note here a model, Consistent EmBeddings of high-dimensional Recordings using Auxiliary variables (CEBRA), that jointly embeds neural activity and behavior into a shared latent space and models this over time (Schneider et al., 2023). However, CEBRA is not a dynamical systems model, nor is it linear, thus lacking the interpretability offered by PSID, CLDS, and b-dLDS.

In comparison, decomposed linear dynamical systems (dLDS) models, including b-dLDS, which we introduce in this work, generate a decomposition of dynamical systems, allowing multiple systems to be co-active at the same time. dLDS-family models are, to our knowledge, also unique in the space of dynamical systems models for their application to large-scale neural data, capturing up to thousands of neurons (results and prior work in Mudrik et al. (2025)). This capacity is possible because dLDS-family models do not require the calculation of high-dimensional covariance matrices and can leverage efficient sparse-state filtering algorithms (Charles et al., 2016).

## 3 Behavior-dLDS Model

We begin by modeling the observed data $\boldsymbol{y}_t \in \mathbb{R}^P$, in this case neural data, as representable by a linear latent variable model $\boldsymbol{x}_t \in \mathbb{R}^N$ at each time point $t$. We assume that $\boldsymbol{x}_t$ is lower-dimensional ($N < P$) and can describe the data through the linear generative model

$$\boldsymbol{y}_t = \boldsymbol{D}\boldsymbol{x}_t + \epsilon_y, \tag{1}$$

where $\boldsymbol{D} \in \mathbb{R}^{P \times N}$ is the observation matrix, and $\epsilon_y$ represents independent observation noise, which we model as Gaussian. We focus on a linear latent space since we want to preserve the geometry of the neural dynamics, and nonlinear transformations, e.g., (variational) autoencoders, can warp data and obfuscate important features.

We next model the dynamics in the latent space $\boldsymbol{x}_t$ with a time-varying linear dynamical system

$$\boldsymbol{x}_t = \boldsymbol{F}_t \boldsymbol{x}_{t-1} + \epsilon_x, \tag{2}$$

where we model $\boldsymbol{F}_t$ as a function of a linear combination of dynamics operators (DOs) $\boldsymbol{f}_m$, weighted by their corresponding dynamics coefficients at each time point $c_{mt}$:

$$\boldsymbol{F}_t = \sum_{m=1}^{M} \boldsymbol{f}_m c_{mt}, \tag{3}$$

and $\epsilon_x$ is the dynamics innovations error, or the error in modeling the dynamics, which we also model as Gaussian.

To complement the dynamics model we consider the observations of behavioral variables as the time series $\boldsymbol{b}_t \in \mathbb{R}^K$. To account for the fact that the current neural state $\boldsymbol{x}_t$ might fluctuate too quickly to capture the timescale of behavior, we instead model the behavior as being generated by the dynamics coefficients $\boldsymbol{c}_t = [c_{1t}, ..., c_{Mt}]^T \in \mathbb{R}^M$. Specifically, we model the behavior through a complementary linear generative model

$$\boldsymbol{b}_t = \boldsymbol{\Psi}\boldsymbol{c}_t + \boldsymbol{\epsilon}_b. \tag{4}$$

where $\boldsymbol{\Psi} \in \mathbb{R}^{K \times M}$ projects the current dynamics into the behavior and $\boldsymbol{\epsilon}_b$ is *i.i.d.* Gaussian measurement noise.

While some of the dynamics observed in brain-wide data are assumed to be related to the observed behavior $\boldsymbol{b}_t$, there are some behavioral quantities that might not have been measured by the experimenter, as well as growing evidence

for internal computations that unfold in parallel to behavior-related activity. To model this effect, we assume that $\boldsymbol{\Psi}$ is *incomplete* in that only a portion of the $\boldsymbol{c}_t$ variables (say for $m \in \Gamma \subseteq [1, .., M]$ with $|\Gamma| = M' < M$) actually relate to $\boldsymbol{b}_t$. This means that while the model has appropriate information to orient $c_{mt}$ for $m \in \Gamma$, $c_{mt}$ for $m$ in the complement to $\Gamma$, $\Gamma^C$, require additional regularization. This challenge can be cast as a type of missing regressor problem, which can be mended through appropriate regularization over $\boldsymbol{c}_t$ (Ibrahim et al., 2005; Gauthier et al., 2022). Specifically, we introduce regularization such that the minimum number of elements of $\boldsymbol{c}_t$ are nonzero at each $t$. This parsimoniousness is achieved through sparsity regularization via a Laplacian prior over $\boldsymbol{c}_t$: $p(\boldsymbol{c}_t) = (\lambda/2)^M e^{-\lambda\|\boldsymbol{c}_t\|_1}$.

## 4 Model Inference

Inference under this model estimates the model parameters $\boldsymbol{D}$, $\boldsymbol{\Psi}$, and $\boldsymbol{f}_m$ for $m \in [1, ..., M]$, which we combine into the parameter variable $\theta = \{\boldsymbol{D}, \boldsymbol{\Psi}, \{\boldsymbol{f}_m\}_{m=1,...,M}\}$. Ideally we would like to take a Maximum Likelihood (ML) approach to find

$$\widehat{\theta} = \arg\min_{\theta} - \log p(\{\boldsymbol{y}_t\}_{t=1,...,T}|\theta). \tag{5}$$

However, $\boldsymbol{c}_t$ represents missing information, i.e., we cannot compute the likelihood $p(\{\boldsymbol{y}_t\}_{t=1,...,T}|\theta)$ directly. Instead, we perform expectation-maximization, an iterative procedure for estimating model parameters and updating their inferred coefficients accordingly until convergence. Specifically, as with other dictionary learning algorithms (Olshausen & Field, 1996; Geadah et al., 2024; Mudrik et al., 2024a), the EM algorithm takes a dirac-delta approximation to the posterior, resulting in a MAP estimation over the latent variables $\boldsymbol{x}_t$, and $\boldsymbol{c}_t$ conditioned on $\theta$, and then an optimization over $\theta$.

**E-step: inferring $\boldsymbol{c}_t$ and $\boldsymbol{x}_t$.** Based on an initial set of parameters $\theta$ (or updated $\theta$ in subsequent iterations), the E-step optimizes the most probable parameters for $\boldsymbol{c}_t$ and $\boldsymbol{x}_t$ given the data and $\theta$:

$$\arg\max_{\boldsymbol{c},\boldsymbol{x}} p\left(\{\{\boldsymbol{c}_t\}_{t\in[1,T]}\{\boldsymbol{x}_t\}_{t\in[1,T]}|\{\boldsymbol{y}_t\}_{t\in[1,T]}, \theta\right). \tag{6}$$

This inference problem can be extremely high-dimensional, especially for large quantities of data and long time series. For efficiency we subsample the data to perform a batch update. Furthermore, we factor the estimation problem over time steps and use dynamic filtering (BPDN-DF) (Charles et al., 2016) to iteratively estimate for each time $t$ the latent states $\boldsymbol{x}_t$ and the dynamics coefficients $\boldsymbol{c}_t$ given the estimates at the previous time step $\widehat{\boldsymbol{x}}_{t-1}$ and $\widehat{\boldsymbol{c}}_{t-1}$:

$$\begin{aligned}\arg\min_{\boldsymbol{x}_t,\boldsymbol{c}_t} \Bigg[ &\|\boldsymbol{y_t} - \boldsymbol{D}\boldsymbol{x}_t\|_2^2 + \lambda_0 \left\|\boldsymbol{x}_t - \widetilde{\boldsymbol{F}}_t\boldsymbol{c}_t\right\|_2^2 + \lambda_1\|\boldsymbol{x}_t\|_1 \\ &+\lambda_2\|\boldsymbol{c}_t\|_1 + \lambda_3\|\boldsymbol{c}_t - \widehat{\boldsymbol{c}}_{t-1}\|_2^2 + \lambda_4\left\|\boldsymbol{b}_t - \boldsymbol{\Psi}\boldsymbol{c}_t\right\|_2^2\Bigg].\end{aligned} \tag{7}$$

The BPDN-DF procedure is known to converge (Charles & Rozell, 2014; Charles et al., 2016), providing an accurate representation of the latent states via a single pass over time.

**M-step: Updating $\theta$.** The parameter set $\theta$ consists of $M + 2$ matrices ($M$ DOs, the mapping from $\boldsymbol{y}$ to $\boldsymbol{x}$, and the mapping from $\boldsymbol{c}$ to $\boldsymbol{b}$). As the datasets are large and the E-step is only run on a random subset of the data, rather than completely optimize $\theta$ between each E-step, i.e.,

$$\widehat{\theta} = \arg\max_{\theta} p(\{\boldsymbol{y}_t\}_{t\in T}|\{\boldsymbol{c}_t\}_{t\in T}, \{\boldsymbol{x}_t\}_{t\in T}, \theta), \tag{8}$$

we instead take a (noisy) gradient step over the approximate log-likelihood given the inferred values of $\boldsymbol{c}_t$, $\boldsymbol{x}_t$:

$$\mathcal{L} = \|\boldsymbol{y_t} - \boldsymbol{D}\boldsymbol{x}_t\|_2^2 + \lambda_0 \left\|\boldsymbol{x}_t - \sum_{l=1}^{L} \boldsymbol{f}_l c_{lt}\boldsymbol{x}_{t-1}\right\|_2^2. \tag{9}$$

The updates for $\boldsymbol{D}$, $\boldsymbol{F}$, and $\boldsymbol{\Psi}$ are then computed via an effective stochastic projected gradient descent. For $\boldsymbol{D}$ we update the estimate as

$$\begin{aligned}\widehat{\boldsymbol{D}} &\leftarrow \Pi_{C_D}(\boldsymbol{D} - \eta_D \nabla_{\boldsymbol{D}} \sum_{t=1}^{T} (\boldsymbol{y}_t - \boldsymbol{D}\boldsymbol{x}_t)^2) \\ &= \Pi_{C_D}(\boldsymbol{D} + \eta_D \sum_{t=1}^{T} (\boldsymbol{y}_t - \boldsymbol{D}\boldsymbol{x}_t)\boldsymbol{x}_t^T)),\end{aligned} \tag{10}$$

where $\Pi_{C_D}$ is an optional projection onto any constraint sets (e.g., non-negativity). Similarly, the gradient over each $\boldsymbol{f}_m$ can be computed as

$$\widehat{\boldsymbol{f}}_m = \Pi_{C_{f_m}} \left( \widehat{\boldsymbol{f}}_m + \eta_f \sum_{t=2}^{T} \left( \boldsymbol{c}_{mt} \left( \boldsymbol{x}_t - \widetilde{\boldsymbol{F}}_t \boldsymbol{c}_t \right) \boldsymbol{x}_{t-1}^T \right) \right), \tag{11}$$

where $\Pi_{C_f}$ is an optional constraint set over $\boldsymbol{f}$ and $\eta_f$ is the step size. Finally $\boldsymbol{\Psi}$ is updated similarly as

$$\widehat{\boldsymbol{\Psi}} \leftarrow \Pi_{C_\Psi}(\boldsymbol{\Psi} - \widetilde{\eta}_\Psi \nabla_{\boldsymbol{D}} \sum_{t=1}^{T} (\boldsymbol{b}_t - \boldsymbol{\Psi}\boldsymbol{c}_t)^2), \tag{12}$$

for constraint set projection $\Pi_{C_\Psi}$ and step size $\widetilde{\eta}_\Psi$. For $\boldsymbol{\Psi}$ we find that rather than setting the step size to be constant (or through a data-agnostic schedule), it is more efficient to rescale a base step size $\eta_\Psi$ based on the norm of the dynamics coefficients as $\widetilde{\eta}_\Psi = \eta_\Psi / \sum_{t=1}^{T} (\boldsymbol{c}_t)^2$. This rescaling makes sure that the step size is appropriately sized in cases when there are many behavior outputs but only one or a few dynamics coefficients generating them (which we term the "sparse suspected" option).

We also use the Frobenius norm option to allow columns of $\boldsymbol{\Psi}$ with very small values to be regularized to 0, which is especially useful in cases where there are many behavior outputs but only one or a few dynamics coefficients generating them, as in the simulations presented below:

$$\begin{aligned}\widehat{\boldsymbol{\Psi}} \leftarrow \Psi_{C_\Psi}(\boldsymbol{\Psi} - \eta_{\Psi,rescaled} \nabla_{\boldsymbol{D}} \sum_{t=1}^{T} (\boldsymbol{b}_t - \boldsymbol{\Psi}\boldsymbol{c}_t)^2 \\ + \lambda_{Frobenius} \sum_{m=1}^{M} \|\boldsymbol{\Psi}_m\|_F^2),\end{aligned} \tag{13}$$

where $m = \{1, ..., M'\}$ are the columns of $\boldsymbol{\Psi}$ corresponding to the dynamics coefficients. By contrast, in the neural data, we expect that multiple dynamics coefficients participate in generating the one- or few-dimensional observed behavior, so the Frobenius norm and "sparse suspected" options are not used.

## 5 Results

**b-dLDS learns which subset of dynamics generated behavior in simulation.** Mudrik et al. (2024a) showed that dLDS could recover true coefficients and operators from a simulation with two sets of independent systems. b-dLDS can do the same, but it can *also* recover the relationship between simulated behavior and dynamics coefficients.

We present four cases: a simplified, sparse case, without added noise, where all of the behavior is generated from one dynamics coefficient and thus tied to only one group of simulated "latent neural dimensions" (Fig. 2); a more complex but still sparse case, where two latent neural dimensions account for the behavior (Supp. Fig. 1); a complex case, where all dynamics coefficients contribute to generating the behavior (Supp. Fig. 2); and a case with one dynamics coefficient used to generate behavior, with added noise (Supp. Fig. 3).

These four cases correspond to three simulations: one where all 10 simulated behavior traces are random multiples of the first simulated dynamics coefficient; one with two overlapping groups of simulated behavior traces, where traces

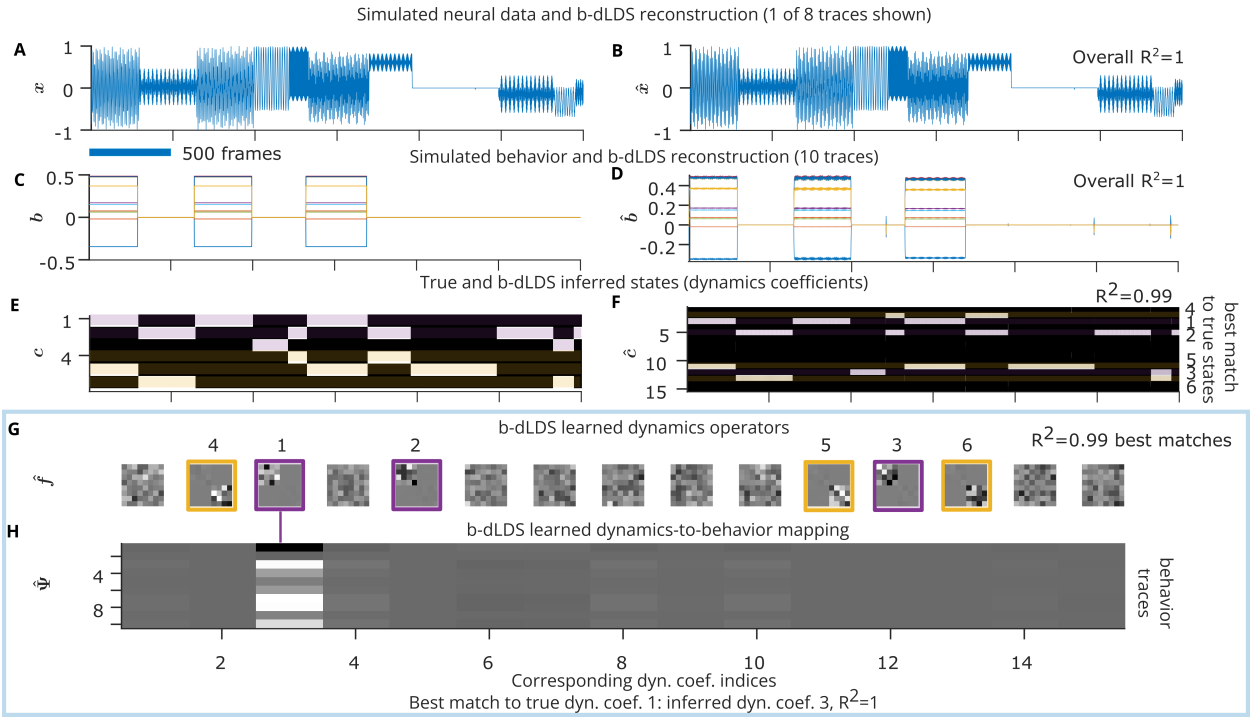



Figure 2: **Two-independent-systems simulation with 10 behavior traces generated as scaled versions of the first ground truth dynamics coefficient.** First trial (out of 50 randomly generated from the same dynamics operators) shown. $R^2$ calculated over all 50 samples. **A,B:** "Neural" ground truth and reconstruction by b-dLDS. Signal is generated from a set of 6 ground truth dynamics operators and corresponding dynamics coefficients, with $\boldsymbol{D} = \boldsymbol{I}$. **C,D:** Simulated behavior and reconstruction from b-dLDS, where behavior is simulated from the ground truth dynamics coefficients. **E:** True dynamics coefficients used to generate the data. **F:** Learned dynamics coefficients (absolute value). **G:** Learned dynamics operators (1-15). **H:** Learned $\boldsymbol{\Psi}$, where each column corresponds to a dynamics operator and dynamics coefficient. Only column 3 is nonzero, and it perfectly reconstructs column 1 of the ground truth $\boldsymbol{\Psi}$. This is the correct column to reconstruct because the first ground truth dynamics coefficient was used to generate the simulated behavior traces.

1-8 are tied via $\boldsymbol{\Psi}$ to the first dynamics coefficient and 3-10 are tied to the second dynamics coefficient; and one where all 10 simulated behavior traces are mapped via a simulated $\boldsymbol{\Psi}$ that is a completely random linear combination of all 6 simulated dynamics coefficients. In the first scenario, one column of the learned $\boldsymbol{\Psi}$ should match the simulated $\boldsymbol{\Psi}$ with a high $R^2$. In the second scenario, two columns of $\boldsymbol{\Psi}$ should match. In the third, six columns should match. Note that in each simulation, there are two independent systems, either regulated by upper-left corner (purple) or lower-right corner (yellow) ground truth DOs (corresponding to ground truth states 1-3 or 4-6), with no overlap between the systems. Moreover, these simulations are further simplified by the fact that the latent states are the observed states, i.e., $\boldsymbol{D} = \boldsymbol{I}$. In Figure 2 and Supplementary Figures 1, 2, and 3, we show that behavior-dLDS learns which dynamics coefficient or coefficients were used to generate simulated behavior through the variable $\boldsymbol{\Psi}$ (Appendix Section "Experiment Details").

We further compared to the most similar and recent method, CLDS (Geadah et al., 2025), which can learn dynamics conditioned on external input (Fig. 3). We tested CLDS on the simulation where only 1 ground truth dynamics coefficient was used to generate only 1 behavior trace. All of the other dynamics coefficients were independent of the behavior. In addition, we included gradations in the values of the true dynamics coefficients in the "on" state (rather than just 1 or 0). We show that while b-dLDS was able to reconstruct the ground truth dynamics (linear combination of $\boldsymbol{f}$ and $\boldsymbol{c}_t$), CLDS, whose dynamics matrix is represented by the variable $\boldsymbol{A}_t$, could not because CLDS is so strongly conditioned on the one behavior it sees. This simulation is comparable to a large-scale recording where one region's activity is strongly correlated with the behavior, but the other regions in the recording are not.

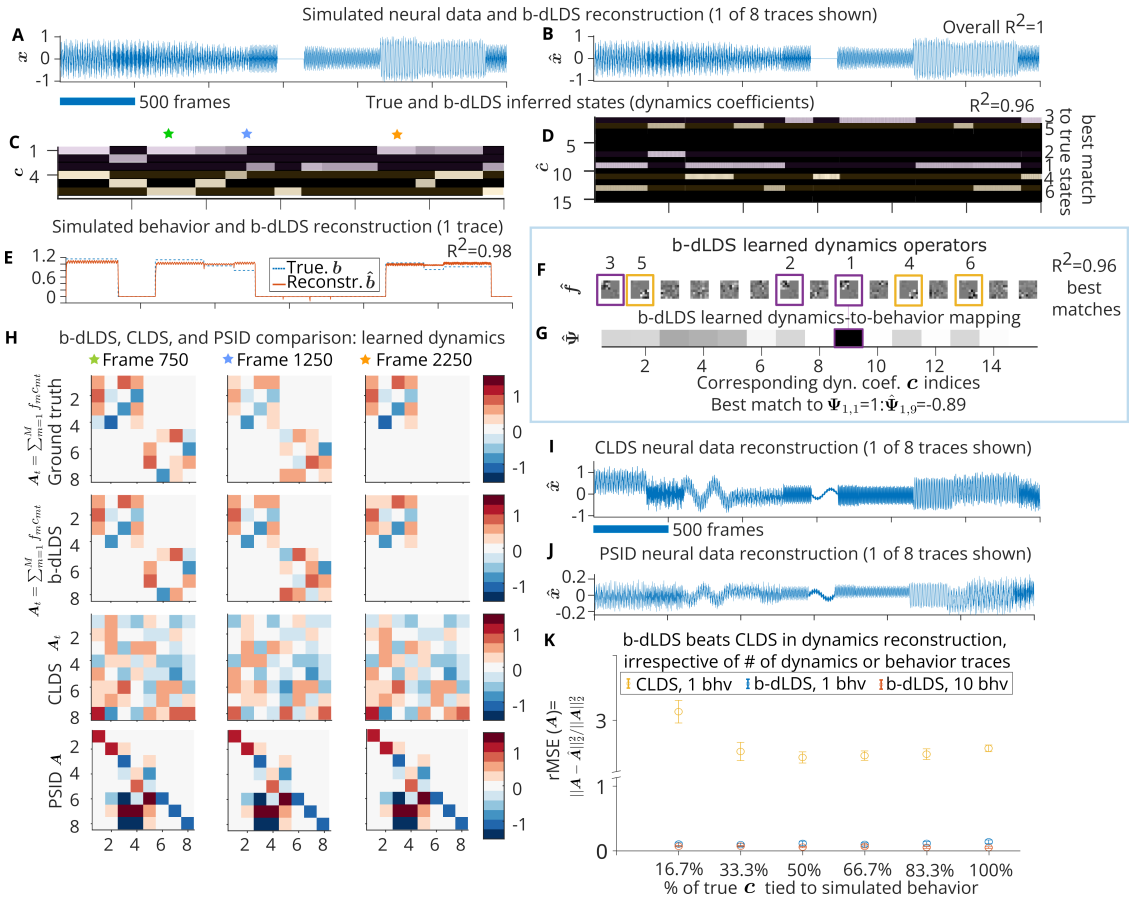



Figure 3: **b-dLDS vs. CLDS and PSID.** CLDS learns only behavior-conditioned dynamics, while b-dLDS learns both behavior-generating and intrinsic ones. PSID learns only a shared latent neural-behavioral space, while b-dLDS learns a hierarchical relationship between latent neural states, latent dynamics, and behavior. (Note: CLDS requires an 80/20 train/test split, so all models were only trained/learned on 40 out of 50 simulated trials. Results shown are on the same 40 "train" trials across models.) **A-G:** b-dLDS, 2 independent systems simulation, 1 trials out of 40 shown, $\boldsymbol{D} = \boldsymbol{I}$, with simulated behavior generated as a linear combination of ground truth dynamics coefficients. $R^2$ calculated across 40 trials. **A, B:** 1 trace out of 8 shown, data and reconstruction. **E:** True dynamics coefficients $\boldsymbol{c}$. Stars match selected time points in **H**. Note that at the first two time points, one top-left (purple) dynamics operator and one bottom-right (yellow) dynamics operator are active, while at the third time point, only a top-left dynamics operator is active. **D:** b-dLDS-inferred $\boldsymbol{c}$. The b-dLDS model is initialized with more than enough dynamics operators (15 learned vs. 6 ground truth operators). Noisy dynamics operators in **F** correspond to unused (black) dynamics coefficients in **D**. **E:** Behavior and reconstruction (1 trace). **G:** $\boldsymbol{\Psi}$ is a row vector with one dynamics coefficient mapping to one behavior (true value: 1, learned value: 0.8765, other true values 0, learned near 0). **H:** CLDS learns an 8x8 dynamics matrix $\boldsymbol{A}_t$, while b-dLDS learns $\boldsymbol{f}$ and $\boldsymbol{c}$, which can be combined via $\sum_{m=1}^{M} f_m c_{mt}$ to create an equivalent matrix. We show 3 example time points here as described above. b-dLDS learns the block-diagonal dynamics matrices, while CLDS does not. The full comparison across time is shown in Figure 4. PSID learns a dynamics matrix $\boldsymbol{A}$ that describes the relationship between latent states. **I:** CLDS neural data reconstruction (1st trace). **J:** PSID neural data reconstruction (1st trace). **K:** b-dLDS achieves lower relative mean squared error than CLDS on dynamics reconstruction, regardless of the number of dynamics or the number of behaviors. CLDS used more than 1 TB of RAM when we tried to run it on 10 behavior traces.

We then compared CLDS to b-dLDS on this same simulation across multiple settings, with 10 random seeds in each. The full set of ground truth dynamics coefficients is 6, so that is the maximum possible number of nonzero columns of $\boldsymbol{\Psi}$. We created simulated datasets with 1 to 6 nonzero columns of $\boldsymbol{\Psi}$ linearly combined to generate 1 simulated behavior. b-dLDS achieved lower relative mean squared error (relative MSE) on the reconstructed dynamics matrix than CLDS did, regardless of whether all or only a subset of the true dynamics generated the behavior (Fig. 3K). CLDS

was once again so strongly conditioned on the behavior that it could not learn the intrinsic dynamics, even when all of the intrinsic dynamics were linearly combined to generate the behavior.

Finally, we compared b-dLDS, which ties behavior to dynamics, to PSID, which instead ties behavior to the latent state (Fig. 3H,J). Because PSID uses a shared neural and behavior latent state, it cannot handle this simulated case, where the neural activity is rapidly oscillatory, while the latent neural dynamics and the behavior are scaled versions of on-off states. PSID is overly biased by the neural activity in this case, correctly registering the time of switches in dynamics, but incorrectly predicting oscillatory behavior outputs instead of smooth states.

**b-dLDS highlights behavior-related dynamic connectivity across the zebrafish hindbrain:** Learning $\mathbf{\Psi}$ from data enables us to distinguish between behaviorally-related dynamics coefficients, and those related to other internal neural computations. Here we present b-dLDS applied to zebrafish data from work by Yang et al. (2022). This dataset features one-dimensional motor information as the behavior trace, and approximately 13000 neurons recorded over 12 minutes in a single zebrafish hindbrain (Appendix Section "Zebrafish Data"). We used b-dLDS to plot dynamic connectivity maps (Appendix Section "Dynamic Connectivity Mapping") corresponding to DOs whose dynamics coefficients have the strongest vs. weakest coefficients in the learned $\mathbf{\Psi}$ (Fig. 4 vs. Supp. Fig. 5). In a second version of the model, we used both the motor signal and trial types as a four-dimensional behavior input to the model (Supp. Figs. 6, 7, 8, and 9).

We observed differences in the dynamics coefficients as well as the connectivity their operators represent. The weakest $\mathbf{\Psi}$ dynamics coefficients appeared rarely and sharply throughout the session. By contrast, the strongest coefficients either appeared throughout the entire session correlated strongly with behavior or separated the session into early and late behavior patterns. These patterns may indicate nonstationarity.

When incorporating trial type information, the model learned different relationships between the motor signal and each trial type. While the most strongly motor-related dynamics coefficients were present throughout the recording, the trial type-related dynamics coefficients split up the recordings in time, indicating nonstationarity.

The dynamic connectivity maps generated from the DOs corresponding to the dynamics coefficients selected above were corroborated by Yang et al. (2022), who showed that these layers of the hindbrain feature the self-location encoding neurons of the medulla oblongata (SLO-MO) (Fig. 4). In DO 8, hindbrain-wide connectivity was highly correlated in time with behavior. Conversely, DOs 10, 18, and 25 acted more locally, both in terms of connectivity and temporally. Interestingly, the dynamic connectivity maps with the weakest $\mathbf{\Psi}$ coefficients (Supp. Fig. 5) shared some key nodes and connections with the dynamic connectivity maps with the strongest $\mathbf{\Psi}$ coefficients, but the temporal patterns were distinct, highlighting a few sharp, sparse moments of strong recurrent dynamic connectivity both laterally and along the anterior-posterior axis.

## 6 Discussion, Limitations, and Future Work

Here we present behavior-dLDS (b-dLDS), which seeks to identify the complementary behavior-related and behavior-independent subsystems underlying rich whole-brain recordings during behavior. As a dLDS-family model, the learned DOs can be coactive in a variety of linear combinations, describing a highly expressive range of dynamics while maintaining the interpretability of linear systems and the ability to match up specific dynamics with temporal epochs/behaviors via sparsity.

Crucially, b-dLDS differs from other models in that it does not assume that the latent neural state, as defined by $\boldsymbol{x}_t$, directly generates behavior on a moment-by-moment basis. Instead, the "state" that controls behavior is the configuration of the DOs that map connectivity in the neural space that is either behavior-related or non-behavior-related, and how these maps change throughout the recordings: early, late, evolving throughout each trial, etc. Unlike defining brain state as the instantaneous neural firing, we treat it as the longer-term configuration of the latent dynamics (Karigo & Charles, 2025). Basis expansions of the dynamics like b-dLDS and CLDS seek to capture these complex interactions at longer timescales and relate the *trends* in neural activity to *trends* in behavior, rather than a time-point-by-time-point representational model.

A limitation of b-dLDS, like all models that use regularization, is that it can be biased by that regularization. For example, b-dLDS might be biased toward having more or fewer behavior-tied dynamics coefficients by selecting extreme regularization parameter settings for behavior reconstruction or the sparsity of $\mathbf{\Psi}$. However, in more moderate

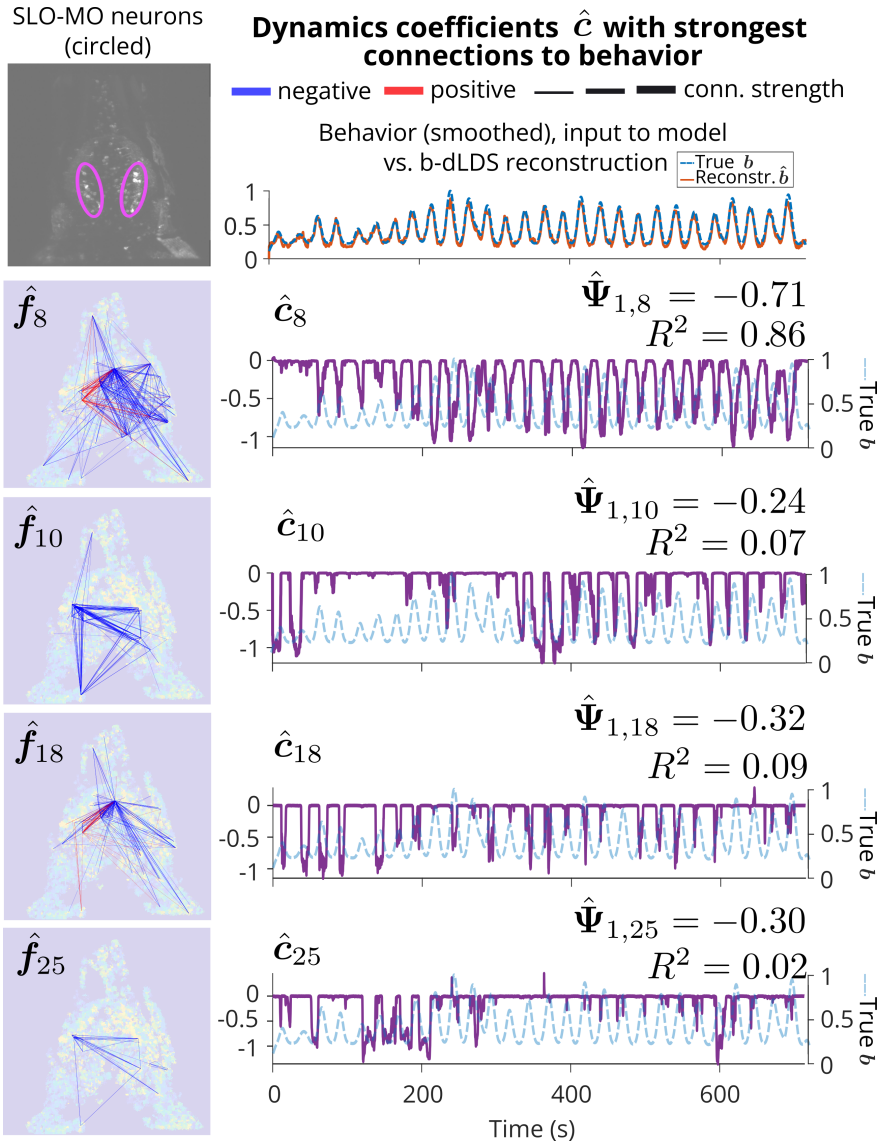



Figure 4: **Four strongest $\Psi$ connections between dynamics coefficients and motor activity in a zebrafish model with one behavior trace.** Overall model $R^2 = 0.90$. Dynamic connectivity maps for dynamics operators and corresponding dynamics coefficients shown. Motor-aligned dynamics are utilized throughout the recording or a mix of earlier and/or later in the recording.

regularization parameter setting ranges, the regularization on the behavior vs. dynamics coefficients balance each other, to some extent; the bias toward inferring a sparse set of dynamics coefficients to describe the data counteracts the bias toward perfectly reconstructing the behavior.

A key benefit of b-dLDS is the relative computational efficiency: other comparable methods that relate neural dynamics to behavior cannot scale to the tens-of-thousands of neurons needed to model zebrafish data. Both CLDS and PSIDencountered memory and size errors, limiting their applicability to the data. This benefit enabled b-dLDS to identify dynamic connectivity maps tied to behavior that revealed the evolution of connections to the SLO-MO region throughout positional homeostasis. As neural datasets continue to grow, models will need to scale even higher, to hundreds of thousands or millions of neurons, in order to map brain-wide circuitry.

In conclusion, b-dLDS is a conceptual reframing of the relationship between brain activity and behavior that separates behavior- and non-behavior-related neural dynamics. b-dLDS can accurately model 13000-dimensional zebrafish hindbrain calcium imaging data and identify which latent neural dynamics are most likely behavior-generating. This

approach enables future work analyzing dynamic connectivity at key time points to compare internal computations to active swimming. The modularity of the b-dLDS framework could also be used to create new dLDS-family models with hierarchical relationships between observed data, e.g., sensory inputs, and latent model components in the style of b-dLDS.

## References

Rifqi O Affan and Benjamin B Scott. Everything, everywhere, all at once: Functional specialization and distributed coding in the cerebral cortex. *Neuron*, 110:2361–2362, 2022. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2022.07.006. URL https://www.sciencedirect.com/science/article/pii/S0896627322006481.

Mikio C. Aoi, Valerio Mante, and Jonathan W. Pillow. Prefrontal cortex exhibits multidimensional dynamic encoding during decision-making. *Nature Neuroscience*, 23:1410–1420, 11 2020. ISSN 15461726. doi: 10.1038/s41593-020-0696-5.

Zoe C. Ashwood, Nicholas A. Roy, Iris R. Stone, Anne E. Urai, Anne K. Churchland, Alexandre Pouget, and Jonathan W. Pillow. Mice alternate between discrete strategies during perceptual decision-making. *Nature Neuroscience*, 25, 2022. ISSN 1097-6256. doi: 10.1038/s41593-021-01007-z.

Adrian G Bondy, Julie A Charlton, Thomas Zhihao Luo, Charles D Kopec, Wynne M Stagnaro, Sarah Jo C Venditto, Laura Lynch, Sanjeev Janarthanan, Stefan N Oline, Timothy D Harris, and Carlos D Brody. Brain-wide coordination of internal signals during decision-making, 2025. URL http://biorxiv.org/lookup/doi/10.1101/2024.08.21.609044.

Adam S Charles and Christopher J Rozell. Convergence of basis pursuit de-noising with dynamic filtering. In *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 374–378. IEEE, 2014.

Adam S Charles, Aurele Balavoine, and Christopher J Rozell. Dynamic filtering of time-varying sparse signals via $\ell_1$ minimization. *IEEE Transactions on Signal Processing*, 64(21):5644–5656, 2016.

Susu Chen, Yi Liu, Ziyue Aiden Wang, Jennifer Colonell, Liu D. Liu, Han Hou, Nai-Wen Tien, Tim Wang, Timothy Harris, Shaul Druckmann, Nuo Li, and Karel Svoboda. Brain-wide neural activity underlying memory-guided movement. *Cell*, 187:676–691.e16, 2 2024a. ISSN 00928674. doi: 10.1016/j.cell.2023.12.035. URL https://linkinghub.elsevier.com/retrieve/pii/S0092867423014459.

Yenho Chen, Noga Mudrik, Kyle A Johnsen, Sankaraleengam Alagapan, Adam S Charles, and Christopher Rozell. Probabilistic decomposed linear dynamical systems for robust discovery of latent neural dynamics. *Advances in Neural Information Processing Systems*, 37:104443–104470, 2024b.

Mark M Churchland, John P Cunningham, Matthew T Kaufman, Justin D Foster, Paul Nuyujukian, Stephen I Ryu, and Krishna V Shenoy. Neural population dynamics during reaching. *Nature*, 487(7405):51–56, 2012.

Jeffrey Demas, Jason Manley, Frank Tejera, Kevin Barber, Hyewon Kim, Francisca Martínez Traub, Brandon Chen, and Alipasha Vaziri. High-speed, cortex-wide volumetric recording of neuroactivity at cellular resolution using light beads microscopy. *Nature Methods*, 18(9):1103–1111, 2021.

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

K. J. Friston, L. Harrison, and W. Penny. Dynamic causal modelling. *NeuroImage*, 19:1273–1302, 8 2003. ISSN 10538119. doi: 10.1016/S1053-8119(03)00202-7.

Jeffrey L Gauthier, Sue Ann Koay, Edward H Nieh, David W Tank, Jonathan W Pillow, and Adam S Charles. Detecting and correcting false transients in calcium imaging. *Nature methods*, 19(4):470–478, 2022.

Victor Geadah, Gabriel Barello, Daniel Greenidge, Adam S Charles, and Jonathan W Pillow. Sparse-coding variational autoencoders. *Neural computation*, 36(12):2571–2601, 2024.

Victor Geadah, Amin Nejatbakhsh, David Lipshutz, Jonathan W. Pillow, and Alex H. Williams. Modeling neural activity with conditionally linear dynamical systems. 2 2025. URL `http://arxiv.org/abs/2502.18347`.

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

Matthew Golub, Steven Chase, and Byron Yu. Learning an internal dynamics model from control demonstration. In *International Conference on Machine Learning*, pp. 606–614. PMLR, 2013.

Joseph G Ibrahim, Ming-Hui Chen, Stuart R Lipsitz, and Amy H Herring. Missing-data methods for generalized linear models: A comparative review. *Journal of the American Statistical Association*, 100(469):332–346, 2005.

Tomomi Karigo and Adam S. Charles. Towards a multi-dimensional understanding of brain states. *Manuscript under review*, 2025.

Mehrdad Kashefi, Jonathan A Michaels, Rhonda Kersten, Jonathan C Lau, Joern Diedrichsen, and J. Andrew Pruszynski. Compositional neural dynamics during reaching, 9 2025. URL `http://biorxiv.org/lookup/doi/10.1101/2025.09.04.674069`.

Mohammad Reza Keshtkaran and Chethan Pandarinath. Enabling hyperparameter optimization in sequential autoencoders for spiking neural data. *Advances in Neural Information Processing Systems*, 32, 2019.

Michael Kleinman, Chandramouli Chandrasekaran, and Jonathan Kao. A mechanistic multi-area recurrent network model of decision-making. *Advances in Neural Information Processing Systems*, 34, 2021.

Scott Linderman, Matthew Johnson, Andrew Miller, Ryan Adams, David Blei, and Liam Paninski. Bayesian learning and inference in recurrent switching linear dynamical systems. In *Artificial Intelligence and Statistics*, pp. 914–922. PMLR, 2017.

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.

Diego Mendoza-Halliday, Haoran Xu, Frederico A.C. Azevedo, and Robert Desimone. Dissociable neuronal substrates of visual feature attention and working memory. *Neuron*, 112:850–863.e6, 3 2024. ISSN 10974199. doi: 10.1016/j.neuron.2023.12.007.

Leenoy Meshulam et al. A brain-wide map of neural activity during complex behaviour. *Nature*, 645(8079):177–191, 9 2025. ISSN 0028-0836. doi: 10.1038/s41586-025-09235-0. URL `https://www.nature.com/articles/s41586-025-09235-0`.

Yu Mu, Sujatha Narayan, Brett D. Mensh, and Misha B. Ahrens. Brain-wide, scale-wide physiology underlying behavioral flexibility in zebrafish. *Current Opinion in Neurobiology*, 64:151–160, 2020. ISSN 18736882. doi: 10.1016/j.conb.2020.08.013.

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, 2024a.

Noga Mudrik, Gal Mishne, and Adam S. Charles. SiBBlInGS: Similarity-driven building-block inference using graphs across states. 6 2024b. URL `http://arxiv.org/abs/2306.04817`.

Noga Mudrik, Ryan Ly, Oliver Ruebel, and Adam Shabti Charles. CREIMBO: Cross-regional ensemble interactions in multi-view brain observations. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

Josue Nassar, Scott W Linderman, Monica Bugallo, and Il Memming Park. Tree-structured recurrent switching linear dynamical systems for multi-scale modeling. *arXiv preprint arXiv:1811.12386*, 2018.

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

Chethan Pandarinath, K Cora Ames, Abigail A Russo, Ali Farshchian, Lee E Miller, Eva L Dyer, and Jonathan C Kao. Latent factors and dynamics in motor cortex and their application to brain–machine interfaces. *Journal of Neuroscience*, 38(44):9390–9401, 2018a.

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, 2018b.

Il Memming Park, Miriam L.R. Meister, Alexander C. Huk, and Jonathan W. Pillow. Encoding and decoding in parietal cortex during sensorimotor decision-making. *Nature Neuroscience*, 17:1395–1403, 10 2014. ISSN 15461726. doi: 10.1038/nn.3800.

Omid G Sani, Hamidreza Abbaspourazad, Yan T Wong, Bijan Pesaran, and Maryam M Shanechi. Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification. *Nature Neuroscience*, 24(1): 140–149, 2021a.

Omid G Sani, Bijan Pesaran, and Maryam M Shanechi. Where is all the nonlinearity: flexible nonlinear modeling of behaviorally relevant neural dynamics using recurrent neural networks. *bioRxiv*, 2021b.

Steffen Schneider, Jin Hwa Lee, and Mackenzie Weygandt Mathis. Learnable latent embeddings for joint behavioural and neural analysis. *Nature*, 617:360–368, 5 2023. ISSN 14764687. doi: 10.1038/s41586-023-06031-6.

Yan-Liang Shi, Roxana Zeraati, The International Brain Laboratory, Anna Levina, and Tatiana A Engel. Brain-wide organization of intrinsic timescales at single-neuron resolution, 8 2025. URL `http://biorxiv.org/lookup/doi/10.1101/2025.08.30.673281`.

Nicholas A Steinmetz, Cagatay Aydin, Anna Lebedeva, Michael Okun, Marius Pachitariu, Marius Bauza, Maxime Beau, Jai Bhagat, Claudia Böhm, Martijn Broux, et al. Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings. *Science*, 372(6539):eabf4588, 2021.

David Sussillo, Mark M Churchland, Matthew T Kaufman, and Krishna V Shenoy. A neural network that finds a naturalistic solution for the production of muscle activity. *Nature Neuroscience*, 18(7):1025–1033, 2015.

David Sussillo, Rafal Jozefowicz, LF Abbott, and Chethan Pandarinath. Lfads-latent factor analysis via dynamical systems. *arXiv preprint arXiv:1608.06315*, 2016.

Anne E. Urai, Brent Doiron, Andrew M. Leifer, and Anne K. Churchland. Large-scale neural recordings call for new insights to link brain and behavior. 2021. doi: 10.1038/s41593-021-00980-9. URL `http://arxiv.org/abs/2103.14662`.

Ziyue Aiden Wang, Susu Chen, Yi Liu, Dave Liu, Karel Svoboda, Nuo Li, and Shaul Druckmann. Not everything, not everywhere, not all at once: a study of brain-wide encoding of movement. doi: 10.1101/2023.06.08.544257. URL `https://doi.org/10.1101/2023.06.08.544257`.

C. Yang, L. Mammen, B. Kim, M. Li, Drew Robson, and Jennifer Li. Uncovering spatial cognitive maps in zebrafish using brain-wide imaging in freely moving animals. In *5th Interdisciplinary Navigation Symposium (iNAV)*, pp. 9–10, 2024.

En Yang, Maarten F. Zwart, Ben James, Mikail Rubinov, Ziqiang Wei, Sujatha Narayan, Nikita Vladimirov, Brett D. Mensh, James E. Fitzgerald, and Misha B. Ahrens. A brainstem integrator for self-location memory and positional homeostasis in zebrafish. *Cell*, 185:5011–5027.e20, 12 2022. ISSN 10974172. doi: 10.1016/j.cell.2022.11.022.

Eva Yezerets, Noga Mudrik, and Adam S. Charles. Decomposed linear dynamical systems (dLDS) models reveal instantaneous, context-dependent dynamic connectivity in C. elegans. *Communications Biology*, 8:1218, 8 2025. ISSN 2399-3642. doi: 10.1038/s42003-025-08599-3. URL `https://www.nature.com/articles/s42003-025-08599-3`.

# A Dynamic Connectivity Mapping

Dynamic connectivity (Yezerets et al., 2025) is calculated by mapping individual DOs or linear combinations of DOs into the ambient space:

$$Connectivity = \boldsymbol{D}\boldsymbol{F}\boldsymbol{D}^T. \tag{14}$$

Dynamic connectivity is a description of the correlation between individual units (e.g., neurons) states $\boldsymbol{y}$ over time, i.e., the outer product of $\boldsymbol{y}_t$ and $\boldsymbol{y}_{t-1}$, $\mathbf{E}[\mathbf{y_t}\mathbf{y_{t-1}^T}]$. Under the b-dLDS model, we can compute this correlation through the latent state correlations between $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t-1}$ mapped back to the ambient space. Given $\boldsymbol{y}_t = \boldsymbol{D}\boldsymbol{x}_t$ and $\boldsymbol{x}_t = \boldsymbol{F}\boldsymbol{x}_{t-1}$,

$$\begin{aligned}
\mathbf{E}[(\mathbf{y_t} - \boldsymbol{\mu}_\mathbf{y})(\mathbf{y_{t-1}} - \boldsymbol{\mu}_\mathbf{y})^\mathbf{T}] &= \mathbf{E}[(\mathbf{D}(\mathbf{x_t} - \boldsymbol{\mu}_\mathbf{x}))(\mathbf{D}(\mathbf{x_{t-1}} - \boldsymbol{\mu}_\mathbf{x}))^\mathbf{T}] && (15)\\
&= \mathbf{E}[\mathbf{D}\mathbf{x_t}\mathbf{x_{t-1}^T}\mathbf{D^T}] - \mathbf{E}[\mathbf{D}\boldsymbol{\mu}_\mathbf{x}]\mathbf{E}[\mathbf{D}\boldsymbol{\mu}_\mathbf{x}]^\mathbf{T} && (16)\\
&= \mathbf{E}[\mathbf{D}]\mathbf{E}[\mathbf{x_t}\mathbf{x_{t-1}^T}]\mathbf{E}[\mathbf{D^T}] - \mathbf{E}[\mathbf{D}]\mathbf{E}[\boldsymbol{\mu}_\mathbf{x}]\mathbf{E}[\boldsymbol{\mu}_\mathbf{x}]^\mathbf{T}\mathbf{E}[\mathbf{D}]^\mathbf{T} && (17)\\
&= \boldsymbol{D}\boldsymbol{F}\boldsymbol{D}^T - \boldsymbol{D}\boldsymbol{\mu}_x\boldsymbol{\mu}_x^T\boldsymbol{D}^T && (18)\\
&= \boldsymbol{D}\mathbf{E}[\mathbf{F}]\mathbf{E}[\mathbf{x_{t-1}}\mathbf{x_{t-1}^T}]\mathbf{D^T} - \mathbf{D}\boldsymbol{\mu}_\mathbf{x}\boldsymbol{\mu}_\mathbf{x}^\mathbf{T}\mathbf{D^T} && (19)\\
&= \boldsymbol{D}\boldsymbol{F}\boldsymbol{\Sigma}_x\boldsymbol{D}^T - \boldsymbol{D}\boldsymbol{\mu}_x\boldsymbol{\mu}_x^T\boldsymbol{D}^T && (20)\\
&= \boldsymbol{D}\left(\boldsymbol{F}\boldsymbol{\Sigma}_x - \boldsymbol{\mu}_x\boldsymbol{\mu}_x^T\right)\boldsymbol{D}^T, && (21)
\end{aligned}$$

where $\boldsymbol{\mu}_x = \mathbf{E}[\mathbf{x}]$ is the expected value of the latent variables.

Under the assumption that 1) the latent states are isotropically distributed over time, we can further approximate $\Sigma_x \approx I$, and 2) the latent states are centered at zero, i.e., $\boldsymbol{\mu}_x =, 0$, we can simplify the above expression to

$$\mathbf{E}[\mathbf{y_t}\mathbf{y_{t-1}^T}] \approx \mathbf{D}\mathbf{F}\mathbf{D^T}. \tag{22}$$

In cases where the above assumptions are violated, the appropriate corrections in Equation equation 21 can be applied.

Here we define this connectivity such that each entry $i, j$ in each $\boldsymbol{f}$ included in $\boldsymbol{F}$ represents the influence of group $j$ in the latent space at time $t-1$ on group $i$ in the latent space at time $t$. Thus, connections go from columns to rows. As a result, when plotting, we transpose this connectivity matrix.

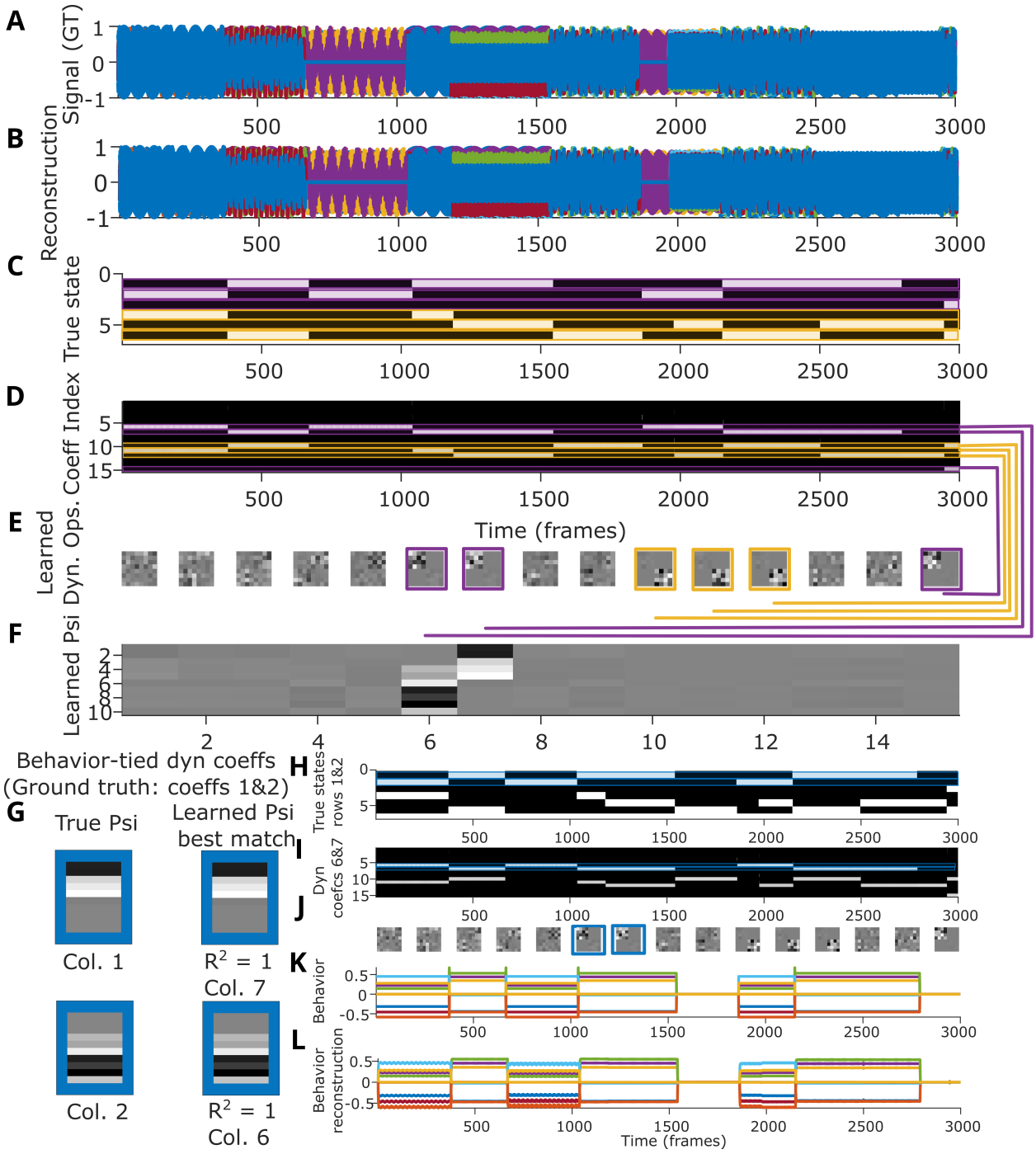
# B Experiment Details

## B.1 Simulation of two independent systems and behavior

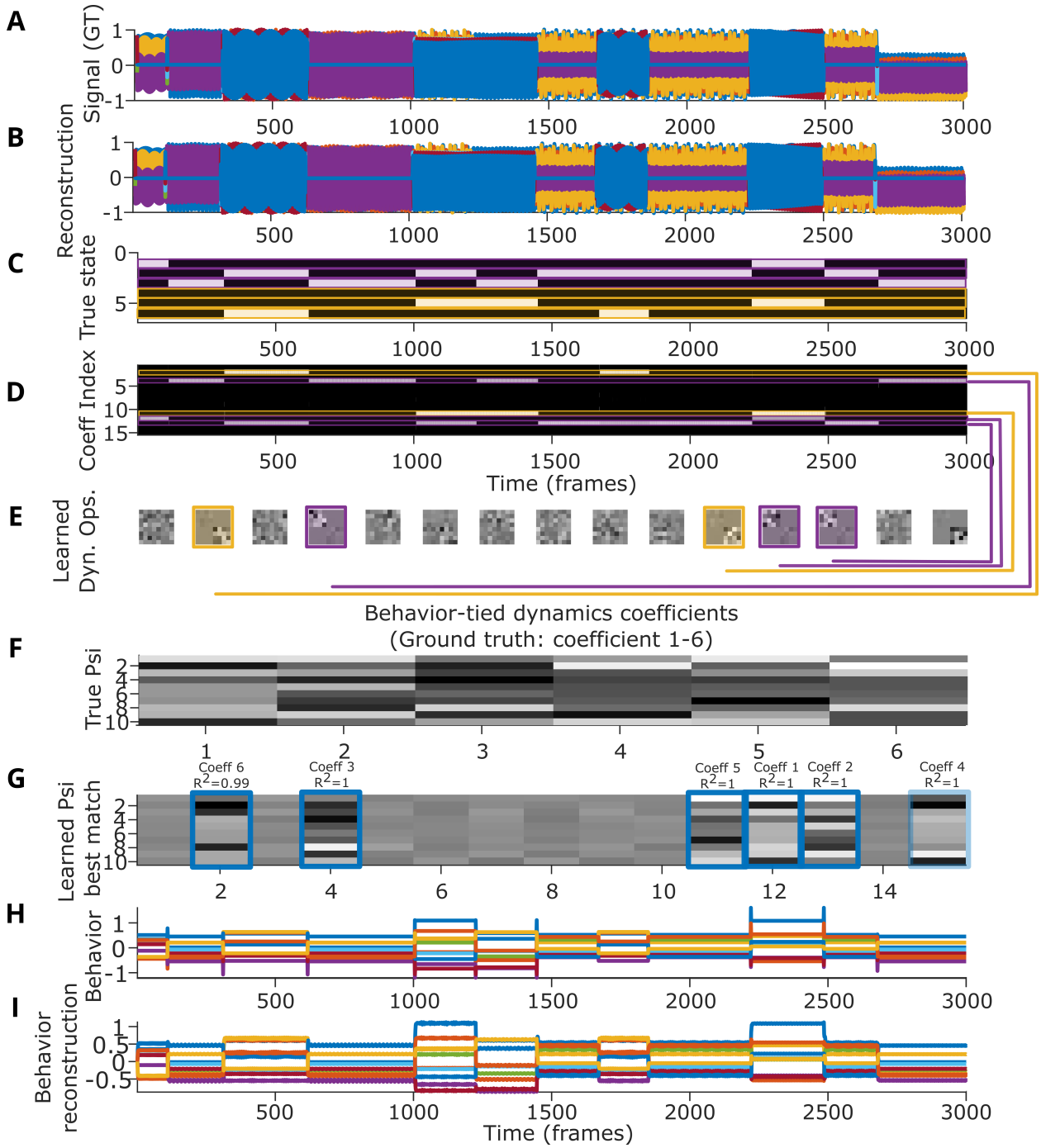
As in Figure 5 in Mudrik et al. (2024a), we test behavior-dLDS on a simulation in which two independent groups of time traces are generated from two non-overlapping sets of operators, unit-normed and applied repetitively over random durations. No more than two operators are active at the same time, and no more than one operator is active for each independent group at the same time. We vary the Gaussian noise levels applied after generating the dynamics operators and dynamics coefficients. Note that $\boldsymbol{D} = \boldsymbol{I}$. In this behavior-dLDS simulation, we also generate simulated behavior $\boldsymbol{b}_t$ from the dynamics coefficients using a randomly generated $\boldsymbol{\Psi}$, where $\boldsymbol{b}_t = \boldsymbol{\Psi}\boldsymbol{c}_t$. (In order to make $\boldsymbol{\Psi}$ only map a subset of the dynamics coefficients, other columns were set to 0 after random initialization.) Supplementary Figures 1, 2, and 3 are shown here; Figure 2 is shown above. The CLDS comparison continues in Supplementary Figure 4.

## B.2 Zebrafish data

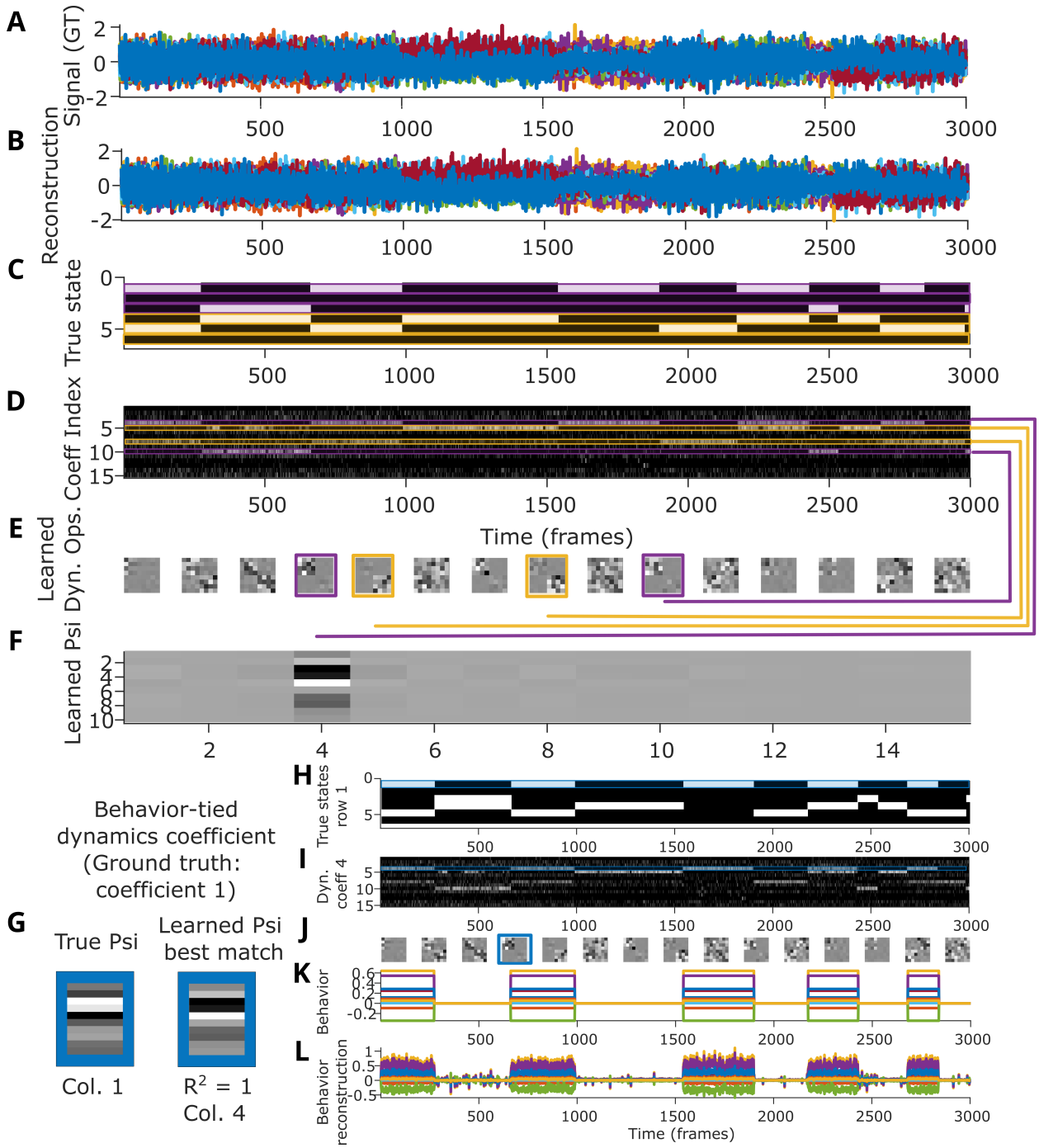
This calcium imaging data was recorded across 9 slices of a single larval zebrafish (*Danio rerio* hindbrain at 6 Hz using lightsheet microscopy by En Yang (Yang et al., 2022) in order to find the self-location encoding medulla oblongata



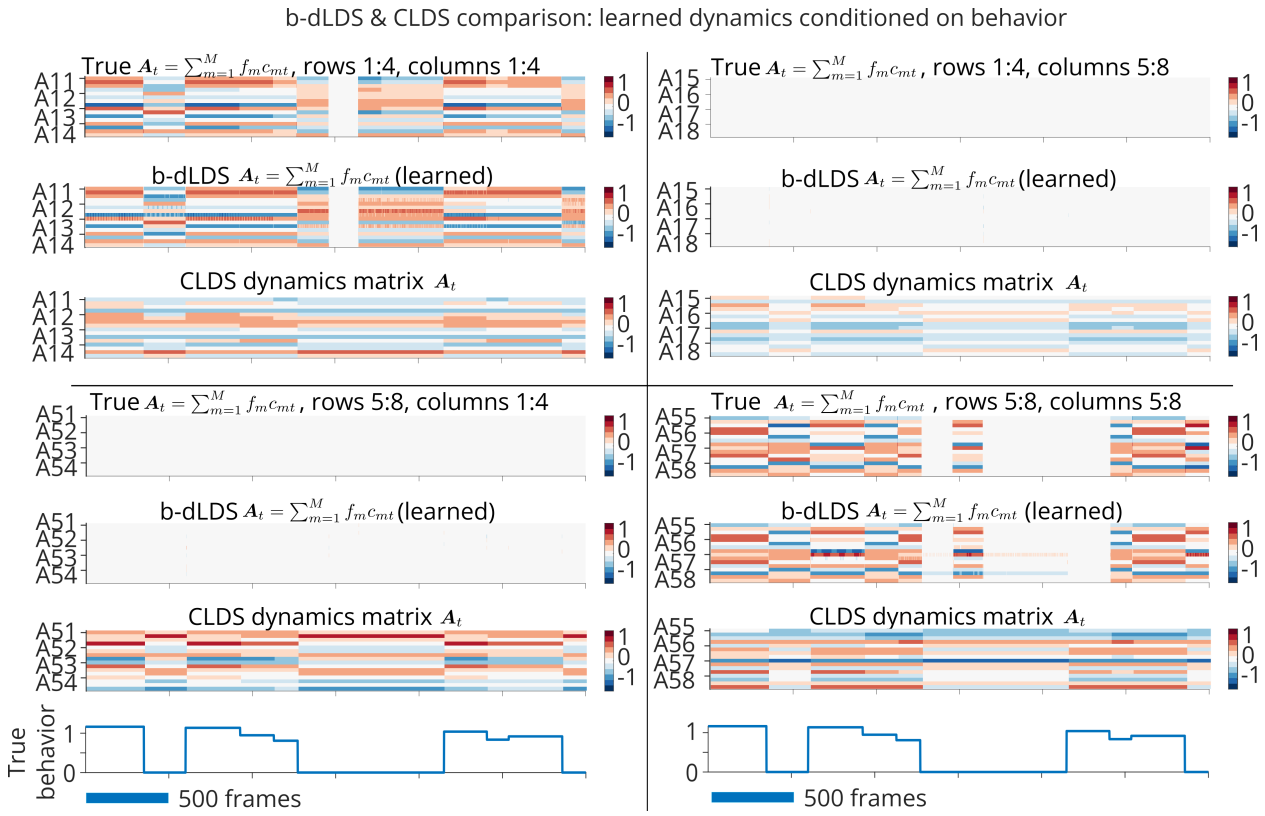

Supplementary Figure 1: **Two-independent-systems simulation with 10 behaviors generated from the first two ground truth states, with some overlap in their combined effects. A:** Ground truth signal fed into the model. **B:** Signal reconstruction from b-dLDS. **C:** True dynamics coefficients used to generate the data. **D:** Learned dynamics coefficients (absolute value). **E:** Learned dynamics operators (1-15). **F:** Learned $\Psi$. **G:** Best reconstruction of the true $\Psi$ (2 columns, other columns all 0). **H:** Highlighted dynamics coefficient used to simulate behavior data. **I:** Highlighted dynamics coefficient corresponding to best match of learned $\Psi$. **J:** Highlighted corresponding dynamics operator. **K:** Simulated behavior ground truth. **L:** Behavior reconstruction.




Supplementary Figure 2: **Two-independent-systems simulation with 10 behaviors generated from a combination of all 6 ground truth states. A:** Ground truth signal fed into the model. **B:** Signal reconstruction from b-dLDS. **C:** True dynamics coefficients used to generate the data - one was randomly initialized as all zeros. **D:** Learned dynamics coefficients (absolute value). **E:** Learned dynamics operators (1-15). **F:** True $\Psi$. **G:** Best reconstruction of the true $\Psi$ (6 columns). **H:** True behavior. **I:** Behavior reconstruction.




Supplementary Figure 3: **Two-independent-systems simulation with 10 behaviors generated from the first ground truth state, with added Gaussian noise in the "neural" data.** **A:** Ground truth signal fed into the model, with Gaussian noise scaled by $\sqrt{0.1}$. **B:** Signal reconstruction from b-dLDS. **C:** True dynamics coefficients used to generate the data - two were randomly initialized as all zeros. **D:** Learned dynamics coefficients (absolute value). **E:** Learned dynamics operators (1-15). **F:** Learned $\mathbf{\Psi}$. **G:** Best reconstruction of the true $\mathbf{\Psi}$ (1 column, other columns all 0). **H:** Highlighted dynamics coefficient used to simulate behavior data. **I:** Highlighted dynamics coefficient corresponding to best match of learned $\mathbf{\Psi}$. **J:** Highlighted corresponding dynamics operator. **K:** Simulated behavior ground truth. **L:** Behavior reconstruction.




Supplementary Figure 4: **CLDS vs. b-dLDS learned dynamics, continued.** CLDS learns an 8x8 dynamics matrix $\boldsymbol{A}_t$, while b-dLDS learns $\boldsymbol{f}$ and $\boldsymbol{c}$, which can be combined via $\sum_{m=1}^{M} f_m c_{mt}$ to create an equivalent matrix. We compare each block of the matrix (top left 4x4 through bottom right 4x4) to the ground truth dynamics from the simulation. b-dLDS closely matches the ground truth and correctly learns that the two off-diagonal corners are 0, indicating 2 independent systems; CLDS instead incorrectly follows the behavior in all dynamics. Each row of each heatmap is an entry of the 4x4 block of the overall 8x8 matrix (A12 means row 1 column 2, 16 rows per heatmap).

neurons (SLO-MO) (14090 neural time traces, 5123 time points; 13098 neural time traces and 4282 time points used after preprocessing in experiment with 1 motor signal (calculated based on electrical recordings from a cluster of neurons as described by Yang et al. (2022)); 14064 neural time traces and 4200 time points used after preprocessing in experiment with 1 motor signal and 3 trial type labels). SLO-MO is hypothesized to form a circuit with the inferior olive, cerebellum, and optic tectum to enable the fish maintain its position in a current (positional homeostasis).

### B.3 Zebrafish data preprocessing

Based on feedback from the Ahrens lab, extreme fluorescence values were capped at +/- 1 a.u. Neural activity was median filtered with window size 5 and behavior was smoothed with a Gaussian with $\sigma = 20$, based on the approximate swim trial durations. The data was then rescaled to calculate delta fluorescence/fluorescence, centered on the mode of each trace and rescaled by a robust standard deviation metric for each trace. A soft normalization of 1 was added to the dF/F rescaling, which proved crucial for improving reconstruction for this and other large-scale zebrafish lightsheet data with b-dLDS. The Ljung-Box test was used to filter out traces with insignificant autocorrelation over a large lag (500 to 1000 time points), since signal traces should show autocorrelation throughout the session given the repetitive nature of the positional homeostasis task, while noise traces should have rapidly decaying autocorrelation. Only the last 4282 out of 5123 time points (after the first 5 complete swim trials) were used for modeling.

### B.4 behavior-dLDS parameters and model selection

#### B.4.1 Simulations

Most key model parameters (Table 1) were maintained from the dLDS simulation in Figure 5 of the dLDS paper (Mudrik et al., 2024a). However, model fitting was drastically improved by implementing stronger regularization on the orthogonality of the dynamics operators ($\lambda_f$) that decays more quickly. While we initially used BayesOpt to find the optimal regularization on the behavior reconstruction ($\lambda_4 = \lambda_{behavior}$), we found that using the same small $\lambda$ as we used for the zebrafish models was a better balance of modeling the dynamics and learning the correct $\mathbf{\Psi}$. Increasing the number of samples and time points per sample also improved reconstruction. Large step size of $\mathbf{\Psi}$ helped to explore the space of $\mathbf{\Psi}$ faster. Parameters were tested iteratively to achieve $R^2_{\mathbf{\Psi}}$ near 1.

Table 1: **Meaning of key b-dLDS parameters, heuristics for setting them, and settings used for 4 simulation model types:** one dynamics coefficient maps to all 10 behaviors; 2 dynamics coefficients map to all 10 behaviors; all dynamics coefficients map to all 10 behaviors; and one dynamics coefficient maps to all 10 behaviors, with added Gaussian noise in the simulated data.

| Parameter | Meaning | How to set | Settings used for 4 simulation types |
|---|---|---|---|
| nF | number of dynamics operators (DOs) | First set to 2x the number of PCs required for 95% variance explained. Then decrease until the median number of active dynamics coefficients starts decreasing. (Goal: minimize the number of DOs while maintaining the inherent dimensionality of the data.) Alternative heuristic: use the number of known behavioral states as a benchmark and double it to get the starting point for the number of DOs, and then start decreasing from there. | 15 |

| | | | |
|---|---|---|---|
| N | number of latent dimensions | Set approximately equal to nF | 8 |
| lambda_val | regularization: sparsity of states (x) | Increase to reduce the number of states active at the same time (moderately sensitive) | 1.0000e-04 |
| lambda_history | regularization: smoothness of states (x) | Increase to encourage states to change smoothly from time point to time point (less sensitive) | 1.0000e-04 |
| lambda_b | regularization: sparsity of dynamics coefficients (c) | Increase to reduce the number of DOs active at the same time (sensitive) | 0.25 |
| lambda_historyb | regularization: smoothness of dynamics coefficients (c) | Increase to encourage DO coefficients to change smoothly from time point to time point | 0.45 |
| max_iters | max number of iterations | Increase if dF not converging to near 0, decrease if converging quickly to save time | 500;5000;500;500;5000 (Fig. 3) |
| F_update | T/F: update dynamics operators | True because operators must be learned here | 1 |
| D_update | T/F: update observation matrix $\boldsymbol{D}$ | False because $\boldsymbol{D} = \boldsymbol{I}$ here | 0 |
| N_ex | number of samples | More samples (and time points) can better learn the data but increases the time required per iteration | 100;100;200;200;100 (Fig. 3) |
| T_s | number of time points per sample | As above | 200 |
| step_d | gradient step size of $\boldsymbol{D}$ update | Start small | 1 |
| step_f | gradient step size of $\boldsymbol{f}$ dictionary update | Start small | 10 |
| step_decay | gradient step decay | Start near 1 | 0.99995 |
| lambda_f | regularization: promotes diversity of dynamics operators | sensitive, start near 0 | 0.1 |
| lambda_f_decay | reduces lambda_f gradually as dynamics operators are learned | Start near 1 | 0.996 |
| solver_type | ‘fista’ or ‘tfocs’ | As the name suggests, fista is significantly faster | ‘fista’ |
| special | ‘’ or ‘no_obs’ | Choose ‘no_obs’ when $\boldsymbol{D} = \boldsymbol{I}$ | ‘no_obs’ |
| behaviordLDS | Use behavior data to regularize b-dLDS model? | 1 or 0 | 1 |
| step_psi | Gradient descent step size for updating $\boldsymbol{\Psi}$ | larger step size explores the space in fewer iterations | 100;10 (Fig. 3) |

| | | | |
|---|---|---|---|
| lambda_behavior | regularization: reconstruction of behavior from dynamics coefficients and $\mathbf{\Psi}$ | start small | 0.1 |
| verysparsebhv | option for improved dictionary update of $\mathbf{\Psi}$ where some dynamics coefficients may not map to behavior at all, resulting in zero values and NaN errors ("sparse suspected") | 1 or 0 (default) | 1 |
| psinorm | option for improved update of $\mathbf{\Psi}$ - Frobenius norm pushes some dynamics coefficients mappings to 0 | 'norm' (default) or 'frob' | 'frob' |
| lambda2 | scaling: Frobenius norm term in dictionary update | start small | 5 |

### B.4.2 Zebrafish

dLDS and behavior-dLDS reconstruction $R^2$ proved particularly sensitive to model size. This was best approximated by the number of principal components required by PCA to achieve an $R^2$ around 0.8-0.9. Thus, we opted to use 50 latent dimensions. We then started with 50 dynamics operators and scaled down from there to 25 operators without loss of $R^2$ or median number of active operators at a time. Regularization coefficients were optimized starting from the coefficients used to model *C. elegans* data in Yezerets et al. (2025). The additional sparsity settings for updating $\mathbf{\Psi}$, e.g., the Frobenius norm, were not used. Settings can be found in Table 2.

Table 2: **Meaning of key b-dLDS parameters, heuristics for setting them, and settings used for two zebrafish model types:** one motor signal input as behavior, or one motor signal plus three trial types as one-hot encodings.

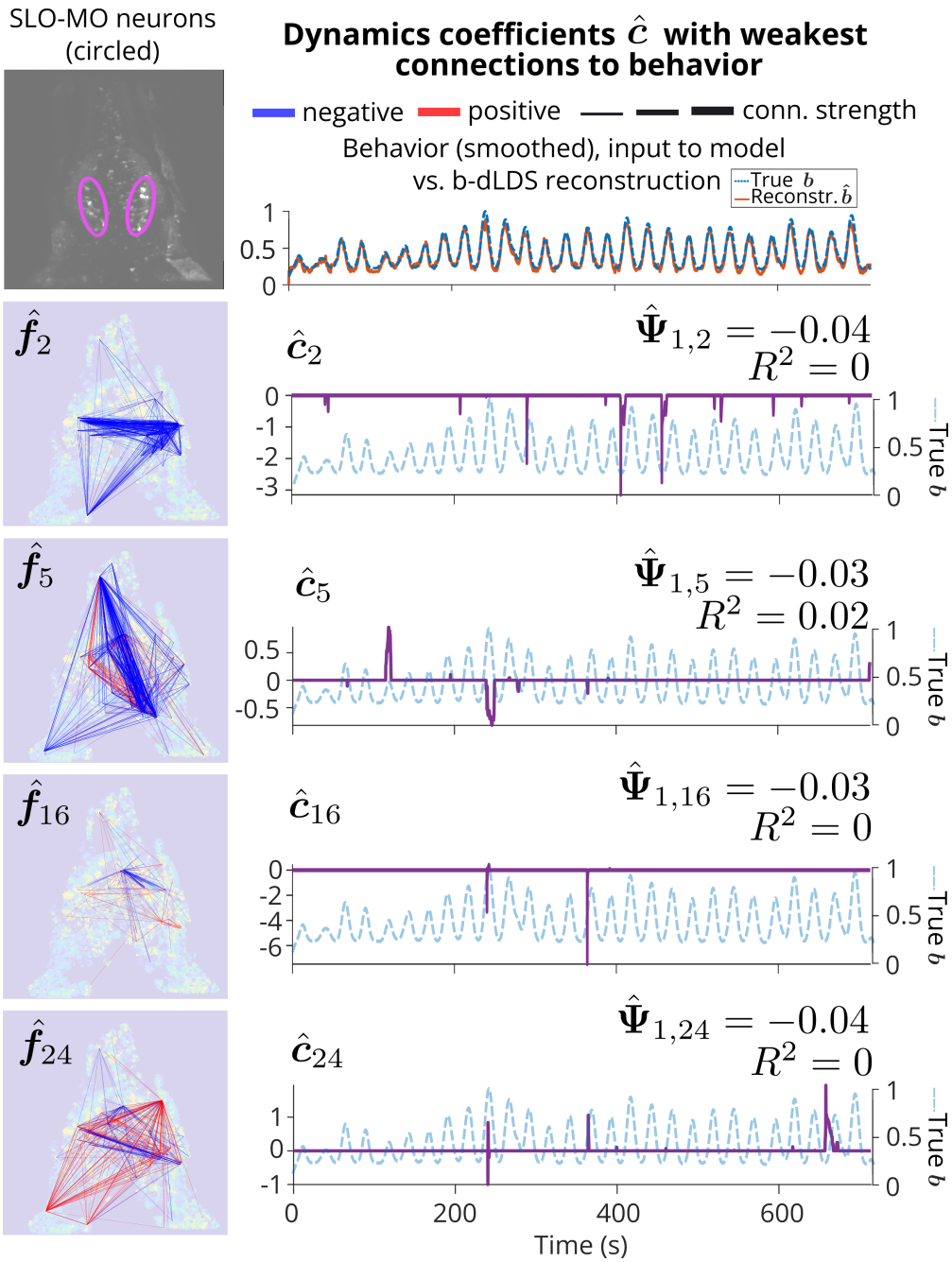
| Parameter | Meaning | How to set | Settings used for zebrafish models (1 vs. 4 behavior inputs) |
|---|---|---|---|

| | | | |
|---|---|---|---|
| nF | number of dynamics operators (DOs) | First set to 2x the number of PCs required for 95% variance explained. Then decrease until the median number of active dynamics coefficients starts decreasing. (Goal: minimize the number of DOs while maintaining the inherent dimensionality of the data.) Alternative heuristic: use the number of known behavioral states as a benchmark and double it to get the starting point for the number of DOs, and then start decreasing from there. | 25 |
| N | number of latent dimensions | Set approximately equal to nF | 50 |
| lambda_val | regularization: sparsity of states (x) | Increase to reduce the number of states active at the same time (moderately sensitive) | 0.05 |
| lambda_history | regularization: smoothness of states (x) | Increase to encourage states to change smoothly from time point to time point (less sensitive) | 0.23 |
| lambda_b | regularization: sparsity of dynamics coefficients (c) | Increase to reduce the number of DOs active at the same time (sensitive) | 0.05 |
| lambda_historyb | regularization: smoothness of dynamics coefficients (c) | Increase to encourage DO coefficients to change smoothly from time point to time point | 0 |
| max_iters | max number of iterations | Increase if dF not converging to near 0, decrease if converging quickly to save time | 5000 |
| F_update | T/F: update dynamics operators | True because operators must be learned here | 1 |
| D_update | T/F: update observation matrix $\boldsymbol{D}$ | True because $\boldsymbol{D} \neq \boldsymbol{I}$ here | 1 |
| N_ex | number of samples | More samples (and time points) can better learn the data but increases the time required per iteration | 40 |
| T_s | number of time points per sample | As above | 30 |
| lambda_f | regularization: promotes diversity of dynamics operators | sensitive, start near 0 | 0.05 |
| solver_type | ‘fista’ or ‘tfocs’ | Although fista is faster to run, tfocs was used for neural data previously in (Yezerets et al., 2025) and produces more regularly-timed dynamics coefficients on neural data | ‘tfocs’ |

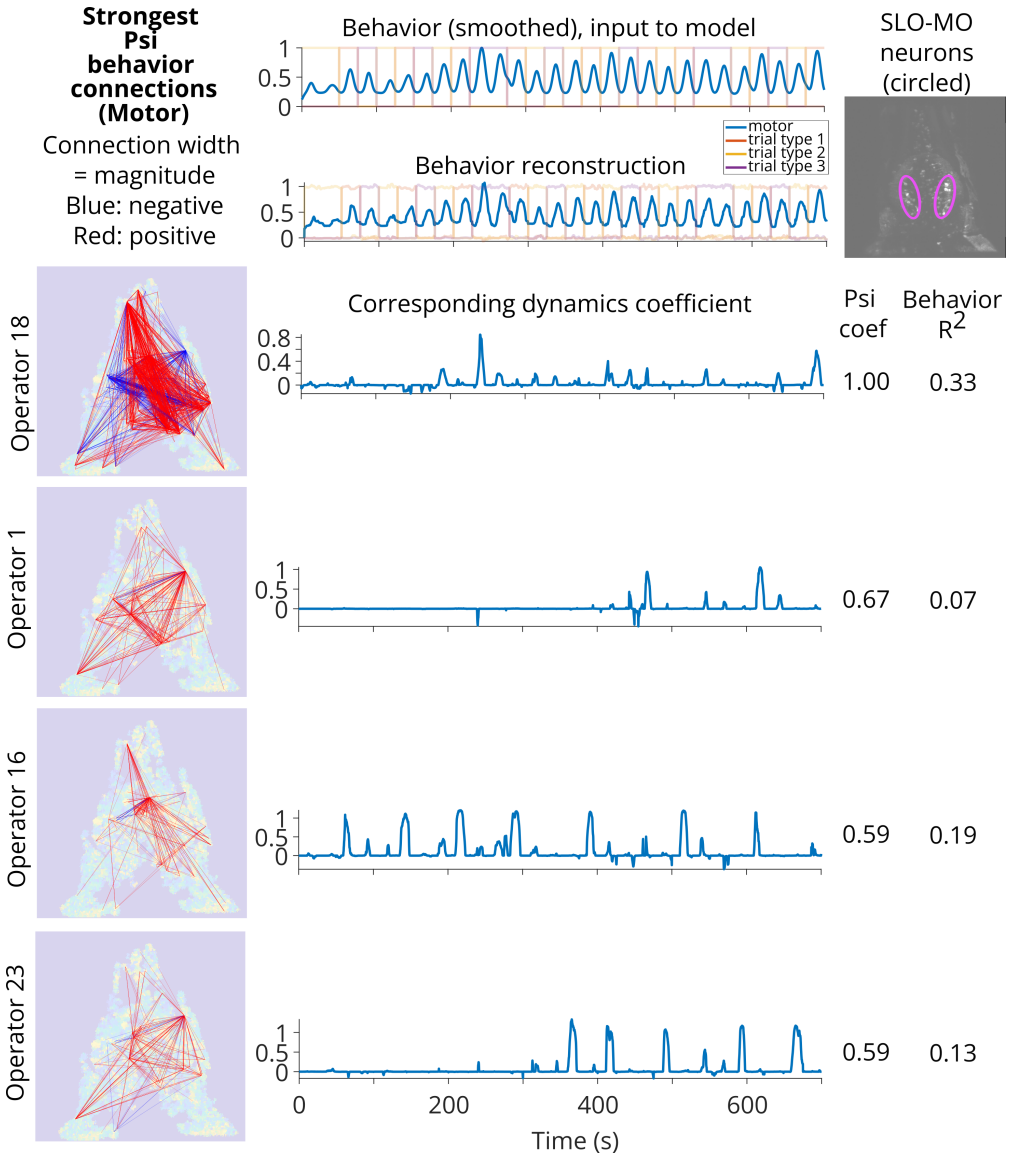
| | | | |
|---|---|---|---|
| behaviordLDS | Use behavior data to regularize b-dLDS model? | 1 or 0 | 1 |
| step_psi | Gradient descent step size for updating $\mathbf{\Psi}$ | larger step size explores the space in fewer iterations | 10 |
| lambda_behavior | regularization: reconstruction of behavior from dynamics coefficients and $\mathbf{\Psi}$ | start small | 1 |

### B.5 Computing infrastructure

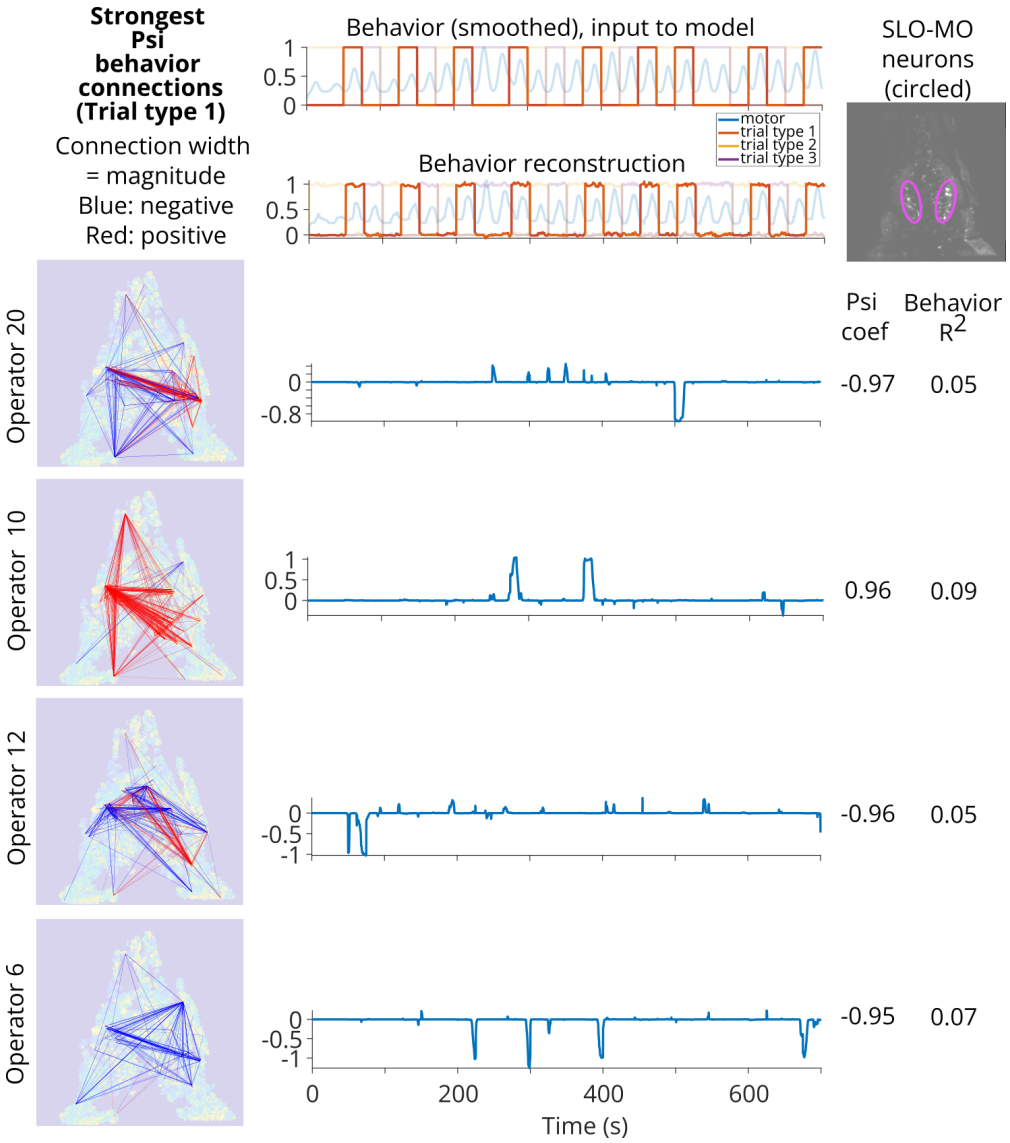
All models shown were run on an internal cluster. Only CPUs were used.



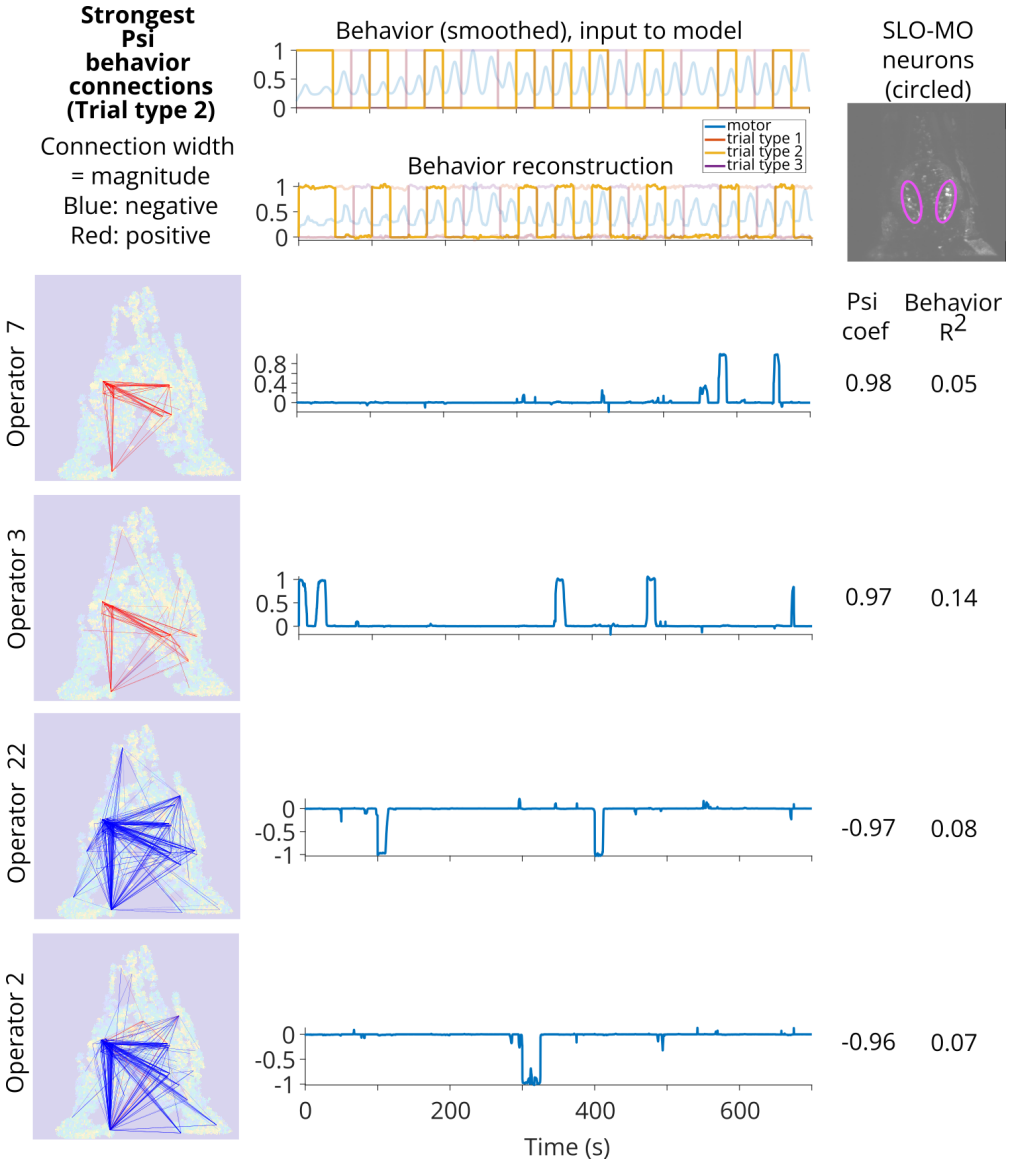

Supplementary Figure 5: **Four weakest $\Psi$ connections between dynamics coefficients and motor activity in a model with one behavior.** Overall model $R^2 = 0.90$. Dynamic connectivity maps and dynamics coefficients shown. The weakest relationships to behavior are those dynamics operators that are rarely utilized throughout the recording.



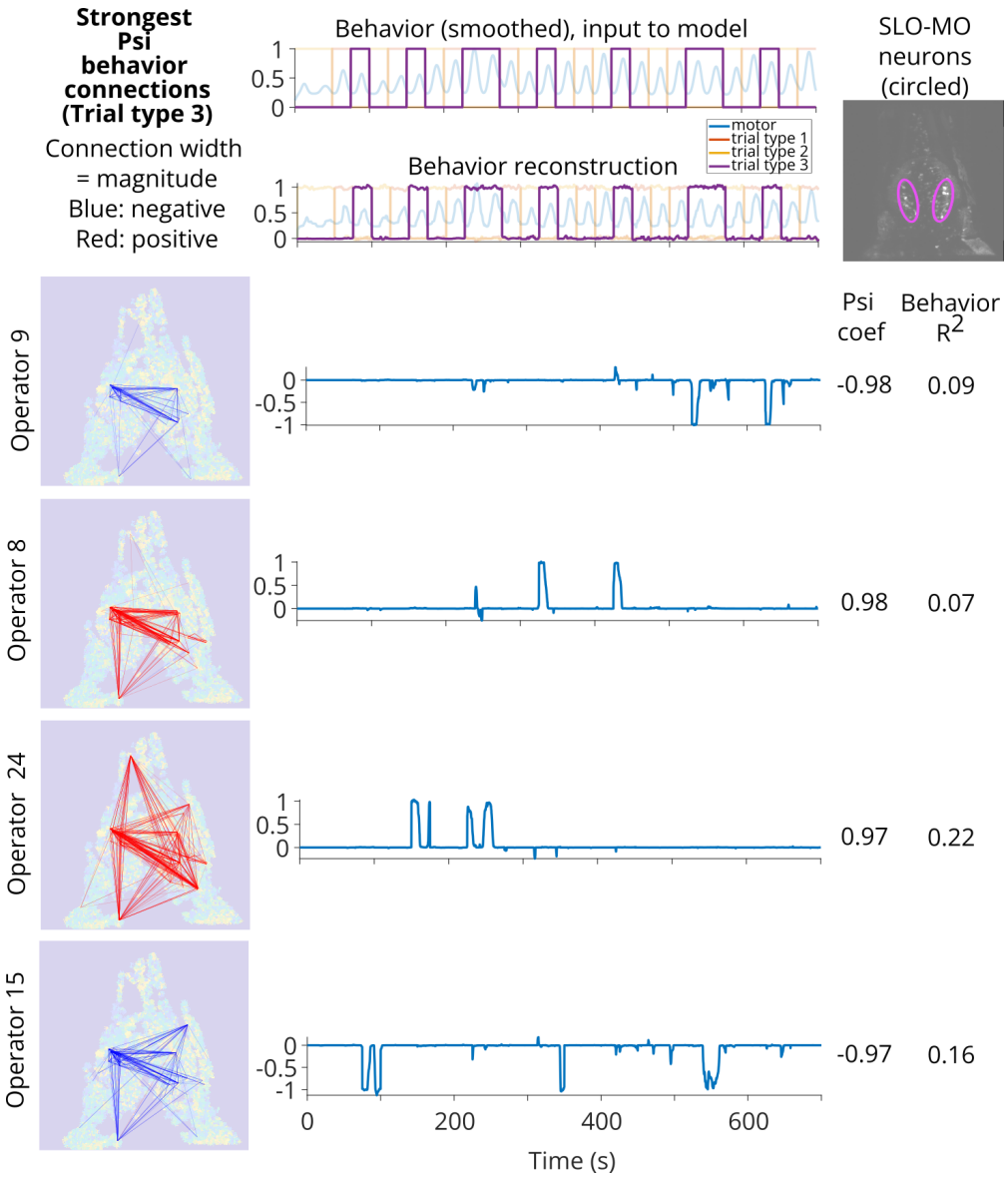

Supplementary Figure 6: **Four strongest $\Psi$ connections between dynamics coefficients and motor activity in a model with four behaviors (motor and 3 trial types based on stimulation patterns, i.e., fish offset by experimenter).** Overall model $R^2 = 0.93$. Dynamic connectivity maps and dynamics coefficients shown. Motor-aligned dynamics are utilized throughout the recording or later in the recording.




Supplementary Figure 7: **Four strongest $\Psi$ connections between dynamics coefficients and swim trial type 1 in a model with four behaviors (motor and 3 trial types based on stimulation patterns between swims).** Overall model $R^2 = 0.93$. Dynamic connectivity maps and dynamics coefficients shown. Swim trial-aligned dynamics cover different time segments of the recording.




Supplementary Figure 8: **Four strongest $\Psi$ connections between dynamics coefficients and swim trial type 2 in a model with four behaviors (motor and 3 trial types based on stimulation patterns between swims).** Overall model $R^2 = 0.93$. Dynamic connectivity maps and dynamics coefficients shown. Swim trial-aligned dynamics cover different time segments of the recording.




Supplementary Figure 9: **Four strongest $\Psi$ connections between dynamics coefficients and swim trial type 3 in a model with four behaviors (motor and 3 trial types based on stimulation patterns between swims).** Overall model $R^2 = 0.93$. Dynamic connectivity maps and dynamics coefficients shown. Swim trial-aligned dynamics cover different time segments of the recording.