Flow-field inference from neural data using deep recurrent networks

Timothy Doyeon Kim [1 2] **Thomas Zhihao Luo** [1] **Tankut Can** [3 4] **Kamesh Krishnamurthy** [1 5]
Jonathan W. Pillow [1] **Carlos D. Brody** [1 6]

Abstract

Neural computations underlying processes such as decision-making, working memory, and motor control are thought to emerge from neural population dynamics. But estimating these dynamics remains a significant challenge. Here we introduce Flow-field Inference from Neural Data using deep Recurrent networks (FINDR), an unsupervised deep learning method for inferring low-dimensional, nonlinear, stochastic dynamics underlying neural population activity. Using spike train data from frontal brain regions of rats performing an auditory decision-making task, we demonstrate that FINDR performs competitively with existing methods in capturing the heterogeneous responses of individual neurons. When trained to disentangle task-relevant and irrelevant activity, FINDR uncovers interpretable low-dimensional dynamics. These dynamics can be visualized as flow fields and attractors, enabling direct tests of attractor-based theories of neural computation. We suggest FINDR as a powerful method for revealing the low-dimensional task-relevant dynamics of neural populations and their associated computations.

1. Introduction

How do neurons work together in large populations to solve a task? Experimental evidence suggests that neural population activity lies on a low-dimensional manifold across multiple brain regions in different species, including rodents, monkeys, humans, and even nematodes (e.g., Kato et al. (2015); Nieh et al. (2021); Churchland et al. (2012); Pandarinath et al. (2015); Safaie et al. (2023)). One influential premise in systems neuroscience is that the low-dimensional dynamics on this manifold mediate the computations performed by neural populations (Vyas et al., 2020; Duncker & Sahani, 2021).

Several methods—here referred to as "neural population dynamics inference methods"—have been developed to infer these dynamics directly from neural population activity measured from an animal performing a computational task. These methods typically make certain assumptions about the dynamics, either to facilitate inference or to increase model capacity. For example, the dynamics are assumed to be autonomous (Duncker et al., 2019), linear (Macke et al., 2011; Gao et al., 2016), switching linear (Linderman et al., 2017; Nassar et al., 2019; Zoltowski et al., 2020), deterministic except at specific time points (Pandarinath et al., 2018; Kim et al., 2021; Keshtkaran et al., 2022), one-dimensional (Genkin et al., 2021), or high-dimensional (Pandarinath et al., 2018; Keshtkaran et al., 2022).

While these methods can capture neural population activity effectively, there are two key areas where improvements can be made. Here, we propose to make these improvements with a new method called FINDR (Flow-field Inference from Neural Data using deep Recurrent networks; Section 2).

1) We relax the assumptions on dynamics by using a gated multilayer perception (MLP) as FINDR's dynamics model (Kim et al., 2023), enabling it to learn a wide range of complex dynamical systems, including those that are difficult to learn using existing methods. In Section 3.1, we construct an example of a dynamical system that is difficult to learn with existing methods, and show that FINDR can accurately infer its dynamics from simulated spike trains generated by this system.

2) The complex response patterns observed in real neurons may be decomposed into task-relevant and -irrelevant components (Rigotti et al., 2013; Gallego et al., 2018; Sani et al., 2021). However, many currently available methods do not explicitly take this into account. FINDR distinguishes between the task-relevant and -irrelevant dynamics, and infers

[1] Princeton Neuroscience Institute, Princeton, NJ [2] Present address: Allen Institute & University of Washington, Seattle, WA [3] School of Natural Sciences, Institute for Advanced Study, Princeton, NJ [4] Present address: Department of Physics, Emory University, GA [5] Joseph Henry Laboratories of Physics, Princeton University, Princeton, NJ [6] Howard Hughes Medical Institute, Princeton University, Princeton, NJ. Correspondence to: Timothy Doyeon Kim <tdkim@princeton.edu>, Carlos D. Brody <brody@princeton.edu>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

them separately. This allows us to more easily interpret the inferred dynamics, and demonstrate a link between the dynamics and task computation. We show this by applying FINDR to spike trains from rat frontal cortical regions involved in decision-making (Sections 3.2– 3.3). We find that task-relevant dynamics in these regions are low-dimensional (Extended Data Figure 3), and even though we can capture the heterogeneous responses of individual neurons well using the existing approaches when we use a sufficient number of latent dimensions in these models, we show that a prominent alternative deep learning-based method infers dynamics that are inconsistent across different training and test splits of the same dataset, whereas FINDR infers dynamics that are consistent and also low-dimensional (Section 3.3).

Because FINDR can represent neural population activity in low dimensions, even as low as two or three dimensions, we can explicitly visualize the flow field (or the velocity vector field) underlying neural population activity. We show that when we visualize the flow field formed by the frontal cortical neural population during decision-making, we consistently see two slow points, one associated with stimulus favoring a leftward choice and the other favoring a rightward choice (Section 3.3).

While we showcase our method with a neural population dataset in decision-making, we expect this method to be applicable to a wide range of neuroscience datasets.

2. Methods

Neural population dynamics inference methods 1) compress the activity of a large population of neurons at time t to an abstract low-dimensional representation, and 2) learn the "rules" of how this representation evolves over time. We call these rules the *dynamics* of the system (Figure 1a).

Given population spike train data from N neurons in an animal performing an experimental task, FINDR achieves 1) by inferring their underlying neural population firing rates $\boldsymbol{\lambda}_t \in \mathbb{R}^N_{\geq 0}$ at time step t ($= 1, 2, 3, ..., T$, where T is the total number of steps taken in a trial) using a low-dimensional task-relevant latent representation $\boldsymbol{z}_t \in \mathbb{R}^L$. This map from $\boldsymbol{z}_t$ to the firing rates $\boldsymbol{\lambda}_t$ is given by

$$\boldsymbol{\lambda}_t = \text{softplus}(\boldsymbol{C}\boldsymbol{z}_t + \boldsymbol{d}_t), \tag{1}$$

where $\boldsymbol{C} \in \mathbb{R}^{N \times L}$ is a loading matrix and $\boldsymbol{d}_t \in \mathbb{R}^N$ is some task-irrelevant time-varying bias. Here, the softplus nonlinearity prevents the firing rates from being negative. Instead of Equation (1), we could have used a more general mapping $\boldsymbol{\lambda}_t = \text{softplus}(\Psi_\kappa(\boldsymbol{z}_t))$, where Ψ is a differentiable map with parameters κ, and could, for example, be a deep neural network. However, for simplicity and for interpretability, we confine our map to be affine. Notably, if $\boldsymbol{C}$ is semi-orthogonal, this makes the distance and angle in the latent space $\mathbb{R}^L$ equivalent to the distance and angle in the inverse-softplus rate space $\mathbb{R}^N$ (see Section 2.2). Also, by learning a task-irrelevant time-varying representation $\boldsymbol{d}_t$, we encourage the low-dimensional representation $\boldsymbol{z}_t$ to be more task-relevant, and therefore more interpretable. The observed population spike counts $\boldsymbol{y}_t$ at time bin t of width Δt are modeled by $\boldsymbol{y}_t \sim \text{Poisson}(\Delta t \boldsymbol{\lambda}_t)$.

To achieve 2), FINDR models the dynamics of the latent representation $\boldsymbol{z}_t$ as a stochastic differential equation (SDE) discretized with the Euler-Maruyama method:

$$\boldsymbol{z}_t = \boldsymbol{z}_{t-1} + \frac{\Delta t}{\tau}\mu(\boldsymbol{z}_{t-1}, \boldsymbol{u}_t) + \frac{\sqrt{\Delta t}}{\tau}\boldsymbol{\xi}_t. \tag{2}$$

Here, τ is a fixed time constant of the SDE, and $\boldsymbol{u}_t \in \mathbb{R}^M$ is the external input to the system at t, typically a set of task variables that the experimenter has control over. The drift function μ describes the "rules" used by $\boldsymbol{z}_t$ to evolve over time. Therefore, μ represents the dynamics used by the neural population while the animal is performing the task. We can visualize the dynamics by plotting the flow field (or the velocity vector field) of μ (Figure 1).

In Equation (2), we model potential noise that may be present in the dynamics, represented as $\boldsymbol{\xi}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$. Here, the noise covariance $\boldsymbol{\Sigma}$ is a diagonal matrix, with each element a learnable parameter in the model. Noise $\boldsymbol{\xi}_t$ can be a crucial component in modeling neural computation. As an example, in perceptual decision-making, we observe that even when an animal is presented with identical stimuli, the animal's choice behavior can vary from trial to trial. If we are given spiking observations from a neural population that represents the animal's choice, we can model variability in the animal's behavior with noise in the dynamics of the neural population. Poisson noise does not model noise in the *dynamics* and is not sufficient to capture this variability.

The drift function μ in Equation (2) is parameterized by

$$\mu(\boldsymbol{z}, \boldsymbol{u}) = \sigma(G(\boldsymbol{z}, \boldsymbol{u})) \odot [-\boldsymbol{z} + F(\boldsymbol{z}, \boldsymbol{u})], \tag{3}$$

where σ is the sigmoid function that acts element-wise, and F and G are multilayer perceptions (MLPs). Function μ parameterized this way is practically more expressive and trainable compared to models without the gating network $\sigma(G(\boldsymbol{z}, \boldsymbol{u}))$ (Kim et al., 2023).

FINDR achieves 1) and 2) by finding the optimal parameters for the networks F and G in the drift function μ, the diagonal noise covariance $\boldsymbol{\Sigma}$, the loading matrix $\boldsymbol{C}$, and the time-varying bias $\boldsymbol{d}_t$ that best capture the observed $\boldsymbol{y}_t$. We will denote these parameters as $\Theta = \{\theta_F, \theta_G, \boldsymbol{\Sigma}, \boldsymbol{C}, \boldsymbol{d}_{1:T}\}$.

2.1. FINDR optimization

To obtain the optimal parameters Θ^*, we minimize the negative log-likelihood of the spike trains given Θ:

$$\mathcal{L} = -\log p_\Theta(\boldsymbol{y}_{1:T}). \tag{4}$$

We approach this problem by first optimizing for $\boldsymbol{d}_{1:T}$ and then optimizing the rest of the parameters $\theta = \Theta \backslash \boldsymbol{d}_{1:T}$. To obtain the optimal $\boldsymbol{d}_t = [\boldsymbol{d}_t^{(1)}; \boldsymbol{d}_t^{(2)}; ...; \boldsymbol{d}_t^{(n)}; ...; \boldsymbol{d}_t^{(N)}]$, we fit a linear basis function model (Bishop, 2007) for each neuron n with

$$\boldsymbol{d}_t^{(n)} = \sum_{j=1}^{D_n} w_j \varphi_j(t), \tag{5}$$

where $\boldsymbol{\varphi} = \{\varphi_1, \varphi_2, ..., \varphi_{D_n}\}$ is a set of raised cosine basis functions (Pillow et al., 2005; 2008; Park et al., 2014). In this setup, we are approximating Equation (4) as a problem of minimizing the mean squared error (MSE) between the observed $\boldsymbol{y}_t^{(n)}$ and $\boldsymbol{d}_t^{(n)}$ for all time steps t and all neurons n. This time-varying bias $\boldsymbol{d}_t^{(n)}$ is meant to capture fluctuations in an individual neuron n's firing rate *within* and *across* trials of the task that are not directly relevant to performing the task itself. See Appendix A.1.1 for details.

After fitting this linear basis function model to obtain $\boldsymbol{d}_t$, we proceed to optimize θ. Minimizing Equation (4) directly with respect to θ can be computationally expensive. Therefore, we instead compute an approximate upper bound of $\mathcal{L}$, which we denote as $\tilde{\mathcal{L}}$ (see Appendix A.1.2 for derivation; Kingma & Welling (2014); Chung et al. (2015); Krishnan et al. (2017)):

$$\begin{aligned}\tilde{\mathcal{L}} = &\sum_{t=1}^{T} \Big[\log p_\theta(\boldsymbol{y}_t | \tilde{\boldsymbol{z}}_t, \boldsymbol{d}_t) \\ &- \beta D_{\text{KL}} \left(q_\phi(\boldsymbol{z}_t | \tilde{\boldsymbol{z}}_{t-1}, \boldsymbol{e}_t) || p_\theta(\boldsymbol{z}_t | \tilde{\boldsymbol{z}}_{t-1}, \boldsymbol{u}_t) \right) \Big],\end{aligned} \tag{6}$$

where $\tilde{\boldsymbol{z}}_t \sim q_\phi(\boldsymbol{z}_t | \tilde{\boldsymbol{z}}_{t-1}, \boldsymbol{e}_t)$, and $\boldsymbol{e}_t$ is some time-varying representation that summarizes the entire sequence $[\boldsymbol{u}_{1:T}; \boldsymbol{y}_{1:T}]$. q is the variational approximation of the posterior $p(\boldsymbol{z}_t | \boldsymbol{z}_{t-1}, \boldsymbol{u}_{1:T}, \boldsymbol{y}_{1:T})$, with ϕ being the variational parameters of q. $\tilde{\boldsymbol{z}}$ is a sample from the variational posterior q.

Minimizing $\tilde{\mathcal{L}}$ has two effects: optimizing neural activity reconstruction (due to the first term in Equation (6)) and inferring the low-dimensional flow field that generated the neural activity (due to the second term in Equation (6)). The coefficient β in front of the second term is an important hyperparameter that determines the trade-off between the accuracy of the reconstructed neural activity and the discovered flow field. When β is too low, the reconstructed neural activity may be accurate, but it becomes unlikely that the inferred latent trajectory $\tilde{\boldsymbol{z}}_{1:T}$ is generated from the flow field $\mu(\boldsymbol{z}, \boldsymbol{u})$. If β is too high, the inferred latent trajectory $\tilde{\boldsymbol{z}}_{1:T}$ becomes more irrelevant to the observed neural activity, but it becomes highly likely that it is generated from our inferred flow field $\mu(\boldsymbol{z}, \boldsymbol{u})$. It has been observed that using $\beta > 1$ can help arrive at interpretable latent representations (Higgins et al., 2017; Burgess et al., 2018). We let $\beta = 2$ to put slightly more weight on the vector-field inference at the cost of less accurate reconstruction of neural activity. However, as we will show in Section 3.2, we find that under many conditions, FINDR outperforms existing methods in reconstructing neural population activity.

We minimize $\tilde{\mathcal{L}}$ by training a sequential variational autoencoder (VAE), where we model $p_\theta(\boldsymbol{y}_t | \boldsymbol{z}_t, \boldsymbol{d}_t)$ in the first term of Equation (6) as $\boldsymbol{y}_t \sim \text{Poisson}(\Delta t \boldsymbol{\lambda}_t)$ with $\boldsymbol{\lambda}_t$ given by Equation (1). We model $p_\theta(\boldsymbol{z}_t | \boldsymbol{z}_{t-1}, \boldsymbol{u}_t)$ in the second term with Equations (2–3). Finally, we model $q_\phi(\boldsymbol{z}_t | \boldsymbol{z}_{t-1}, \boldsymbol{e}_t)$ with

$$\begin{aligned}\boldsymbol{z}_t &= \boldsymbol{z}_{t-1} + \frac{\Delta t}{\tau} \nu(\boldsymbol{z}_{t-1}, \boldsymbol{u}_t, \boldsymbol{e}_t) + \frac{\sqrt{\Delta t}}{\tau} \boldsymbol{\xi}_t, \\ \nu(\boldsymbol{z}, \boldsymbol{u}, \boldsymbol{e}) &= \sigma(\tilde{G}(\boldsymbol{z}, \boldsymbol{u}, \boldsymbol{e})) \odot \left[-\boldsymbol{z} + \tilde{F}(\boldsymbol{z}, \boldsymbol{u}, \boldsymbol{e}) \right],\end{aligned} \tag{7}$$

where $\tilde{F}$ and $\tilde{G}$ are MLPs (separate from F and G), and $\boldsymbol{e}_t$ is the output at every time step t of a bidirectional gated recurrent unit (GRU) (Cho et al., 2014) that gets the sequence $[\boldsymbol{u}_t; \boldsymbol{y}_t]$ at t as its input. When we model the variational posterior q this way, the KL divergence term in Equation (6) has an analytical form. Note that in principle, having $\boldsymbol{u}$ in addition to $\boldsymbol{e}$ as input to ν is redundant. Nevertheless, we choose to have $\boldsymbol{u}$ as an extra input to ν because we find that this parameterization is empirically superior to the model without $\boldsymbol{u}$ in terms of performance. Figure 1b presents a graphical overview of the FINDR model, and Figure 1c shows schematics of the model architecture.

We train FINDR with mini-batch gradient descent with warm restarts (Loshchilov & Hutter, 2017), where the gradient of the loss $\tilde{\mathcal{L}}$ is taken with respect to both θ and the variational parameters ϕ (which include parameters of the bidirectional GRU, $\tilde{F}$, and $\tilde{G}$) (see Section A.1.5 for details). The gradient is obtained by backprogation through time (BPTT).

2.2. Identifiability and interpreting the learned latent representation

Following Wang et al. (2021), we define latent variable $\boldsymbol{z}$ as identifiable if $\boldsymbol{z}_1 \neq \boldsymbol{z}_2$ implies that $p_\theta(\boldsymbol{y}|\boldsymbol{z}_1, \boldsymbol{d}) \neq p_\theta(\boldsymbol{y}|\boldsymbol{z}_2, \boldsymbol{d})$ in the first term of Equation (6). Since we use a linear projection $\boldsymbol{C}$ for $p_\theta(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{d})$, this is satisfied if $\boldsymbol{C}$ is injective. Note that $\boldsymbol{z}$ is identifiable only up to a linear transformation because for any invertible $\boldsymbol{A} \in \mathbb{R}^{L \times L}$, we can re-write $\boldsymbol{C}\boldsymbol{z}$ such that $\boldsymbol{C}\boldsymbol{z} = \boldsymbol{C}\boldsymbol{A}\boldsymbol{A}^{-1}\boldsymbol{z} = \boldsymbol{C}'\boldsymbol{z}'$, where $\boldsymbol{C}' = \boldsymbol{C}\boldsymbol{A}$ and $\boldsymbol{z}' = \boldsymbol{A}^{-1}\boldsymbol{z}$. Therefore, for all models with $p_\theta(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{d}) = \text{softplus}(\boldsymbol{C}\boldsymbol{z} + \boldsymbol{d})$ as in Equation (1),

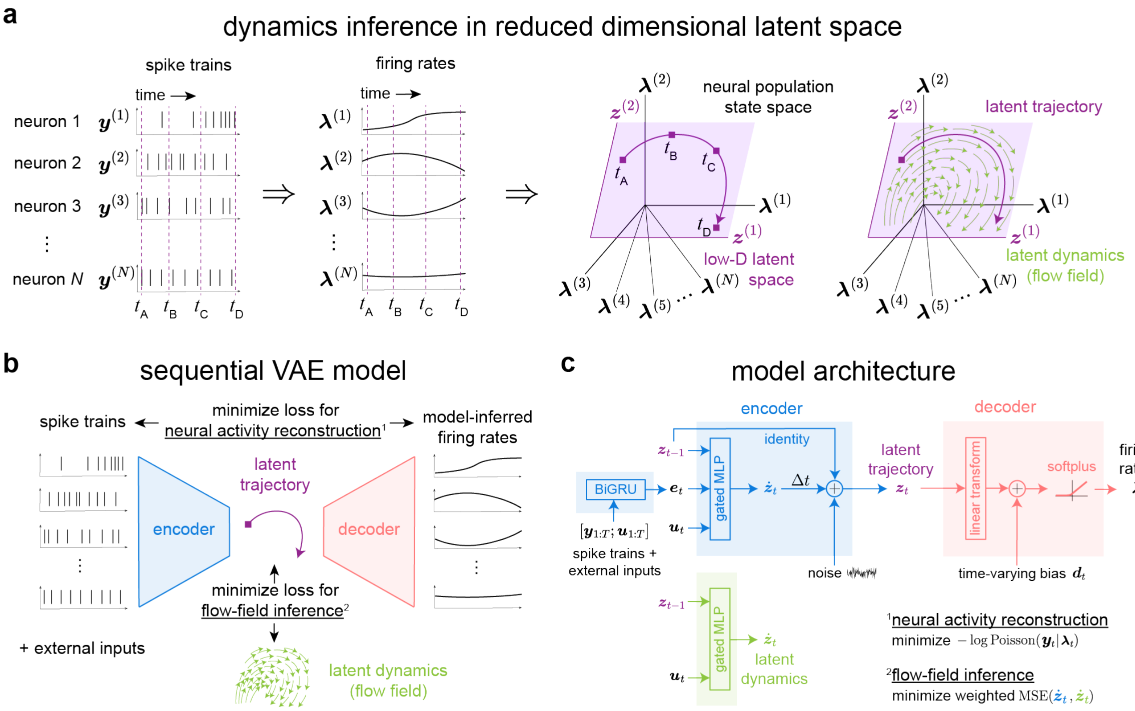

a
dynamics inference in reduced dimensional latent space
spike trains
time →
neuron 1 $\boldsymbol{y}^{(1)}$
neuron 2 $\boldsymbol{y}^{(2)}$
neuron 3 $\boldsymbol{y}^{(3)}$
neuron N $\boldsymbol{y}^{(N)}$
t_A t_B t_C t_D
⇒
firing rates
time →
$\boldsymbol{\lambda}^{(1)}$
$\boldsymbol{\lambda}^{(2)}$
$\boldsymbol{\lambda}^{(3)}$
$\boldsymbol{\lambda}^{(N)}$
t_A t_B t_C t_D
⇒
neural population state space
$\boldsymbol{z}^{(2)}$
$\boldsymbol{z}^{(1)}$
t_A t_B t_C t_D
$\boldsymbol{\lambda}^{(1)}$ $\boldsymbol{\lambda}^{(2)}$ $\boldsymbol{\lambda}^{(3)}$ $\boldsymbol{\lambda}^{(4)}$ $\boldsymbol{\lambda}^{(5)}$ … $\boldsymbol{\lambda}^{(N)}$
low-D latent space
latent trajectory
latent dynamics (flow field)
b
sequential VAE model
spike trains
minimize loss for neural activity reconstruction¹
model-inferred firing rates
encoder
latent trajectory
decoder
minimize loss for flow-field inference²
+ external inputs
latent dynamics (flow field)
c
model architecture
encoder
identity
$\boldsymbol{z}_{t-1}$
BiGRU
$\boldsymbol{e}_t$
$\boldsymbol{u}_t$
gated MLP
$\dot{\boldsymbol{z}}_t$
Δt
$[\boldsymbol{y}_{1:T}; \boldsymbol{u}_{1:T}]$
spike trains + external inputs
noise
latent trajectory
$\boldsymbol{z}_t$
decoder
linear transform
softplus
firing rates
$\boldsymbol{\lambda}_t$
time-varying bias $\boldsymbol{d}_t$
$\boldsymbol{z}_{t-1}$
$\boldsymbol{u}_t$
gated MLP
$\dot{\boldsymbol{z}}_t$
latent dynamics
¹neural activity reconstruction
minimize $-\log \mathrm{Poisson}(\boldsymbol{y}_t | \boldsymbol{\lambda}_t)$
²flow-field inference
minimize weighted $\mathrm{MSE}(\dot{\boldsymbol{z}}_t, \dot{\boldsymbol{z}}_t)$

Figure 1. A graphical description of FINDR. **a**, FINDR infers the firing rates $\boldsymbol{\lambda}$ underlying N-dimensional spike trains $\boldsymbol{y}$ and compresses $\boldsymbol{\lambda}$ to L-dimensional representation $\boldsymbol{z}$ ($L \ll N$). Based on the trajectory formed by $\boldsymbol{z}$ over time ("latent trajectory" over $t_A \to t_B \to t_C \to t_D$), FINDR infers the dynamics of $\boldsymbol{z}$, more specifically the "flow field" showing the rules of *how* $\boldsymbol{z}$ moves in the latent space. **b**, FINDR is a sequential VAE that models the probability density functions in the loss $\tilde{\mathcal{L}}$ in Equation (6). FINDR's encoder network models the variational posterior $q_\phi(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{e}_t)$, and the decoder network models $p(\boldsymbol{y}_t|\boldsymbol{z}_t, \boldsymbol{d}_t)$. An additional network models $p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{u}_t)$ to infer the flow field. FINDR minimizes a loss for neural activity reconstruction (the first term in Equation (6)) and a loss for flow-field inference (the second term in Equation (6)). **c**, The encoder network (in blue) processes data with a bidirectional GRU and infers the single-trial latent trajectories using a gated neural SDE discretized with the Euler-Maruyama scheme (Equation (7)). The decoder network (in red) transforms the latent trajectories to firing rates (Equation (1)). FINDR reconstructs the observed neural activity using the firing rates from the low-dimensional latent trajectories and the time-varying bias. At the same time, FINDR infers the most likely flow field that generated the inferred latent trajectories using a gated MLP (in green).

including FINDR, and those with a rectifying nonlinearity other than the softplus, we perform singular value decomposition (SVD) on $\boldsymbol{C} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$, where $\boldsymbol{U} \in \mathbb{R}^{N\times L}$ is a semi-orthogonal matrix, $\boldsymbol{S} \in \mathbb{R}^{L\times L}$ is a diagonal matrix with its entries populated by the singular values and $\boldsymbol{V} \in \mathbb{R}^{L\times L}$ is an orthogonal matrix. Then, we set $\boldsymbol{C} \leftarrow \boldsymbol{U}$ and $\boldsymbol{z} \leftarrow \boldsymbol{S}\boldsymbol{V}^\top\boldsymbol{z}$ post-training. This makes the distance and angle in the new latent space $\mathbb{R}^L$ and the distance and angle in the inverse-softplus rate space $\mathbb{R}^N$ the same, i.e., $||\boldsymbol{C}\boldsymbol{z}||^2 = \boldsymbol{z}^\top\boldsymbol{C}^\top\boldsymbol{C}\boldsymbol{z} = \boldsymbol{z}^\top\boldsymbol{z} = ||\boldsymbol{z}||^2$ for all $\boldsymbol{z}$. We then rotate the latent space of $\boldsymbol{z}$ so that the axes are its principal components. The latents $\boldsymbol{z}$ after this post-training transformation (effectively equivalent to a model with a semi-orthogonal $\boldsymbol{C}$) are thus identifiable up to an orthogonal transformation. We did not place soft or hard constraints on $\boldsymbol{C}$ to be semi-orthogonal during training, as this worsened performance as reported in other contexts (Vorontsov et al., 2017). See Appendices A.2–A.3 for more details.

3. Experimental Results

3.1. FINDR can accurately infer approximately continuous attractors in synthetic neural populations

To examine the validity of FINDR, we generated simulated population spike trains from a known low-dimensional dynamical system and checked whether FINDR can infer latent dynamics that are similar to the ground truth. The low-dimensional dynamical system we use is inspired by the "n-bit flip-flop task" (Sussillo & Barak, 2013). In this task, the system receives transient pulse inputs from n different channels and needs to memorize the value of the most recent pulse in each channel. It is known that a dynamical system can use attractors to solve this task and that the attractor structure of the system reflects the statistics of the pulses (Kim et al., 2023). For example, in Figure 2a, we let the system memorize the pulse values from two channels, where the pulse value in channel 1, c_1, and the value in channel 2, c_2, are constrained to satisfy $1 \leq \sqrt{c_1^2 + c_2^2} \leq 2$. For the system to have robust memory of the pulses in the two channels, it should form a 2-dimensional continuous attractor that has the shape of a disk (Kim et al., 2023).

We simulated our data from 500 different Poisson spiking neurons, with the task-irrelevant dynamics being a constant bias with around 5 spikes/s of firing rates (Figure 2b). The latent trajectories $\boldsymbol{z}_{\text{true}}$ trace, with some small noise, the optimal solution to the task, given external inputs $\boldsymbol{u}_{\text{true}}$ (Figure 2a). Then we asked whether FINDR, given the neural population activity and external inputs $\boldsymbol{u}_{\text{true}}$, can reconstruct a 2-dimensional disk attractor (Figure 2c-d) needed to solve this task. This task was constructed such that methods that assume autonomous dynamics (e.g., (Duncker et al., 2019)) or linear dynamics (e.g., (Macke et al., 2011)) cannot discover the disk attractor. Methods assuming switching linear dynamics (e.g., Linderman et al. (2017)) may have difficulty approximating the disk with a few interpretable discrete states because the latent space is typically partitioned linearly in these models. Methods learning a high-dimensional dynamical system (e.g., Pandarinath et al. (2018)) may have difficulty inferring the true latent dimensionality (in this case $L = 2$), and may use dynamical features not present in the data (Sedler et al., 2023).

To identify whether FINDR correctly captures the true latent dimensionality ($L = 2$) of the population spike trains, we trained multiple FINDR models, each assuming different latent dimensions ($L = 1, 2, ..., 6$), on 600 trials of the simulated population spike trains. For each of the FINDR models assuming different latent dimensions, we did a grid search over the hyperparameters (see Section A.1.6 for details) and found the best-performing model by evaluating the normalized log-likelihood score (Pei et al., 2021) on 200 validation trials not used during training. We find that the log-likelihood, evaluated on 200 test trials (which are separate from the validation trials), saturates around $L = 2$ (Figure 2c). Consistent with this result, we also find that when we do principal component analysis (PCA) on the FINDR-inferred latent trajectories $\boldsymbol{z}$, we see that two principal components (PCs) are sufficient to explain more than 99% of the variance in each model that assumes a different L $(= 2, 3, ..., 6)$ (Figure 2d). Furthermore, when we project the flow field inferred from FINDR onto the first two PCs, we find an approximate disk attractor across FINDR models assuming $L = 2, 3, ..., 6$ (Figure 2d).

When we do similar analyses with simulated population spike trains generated from a 2-dimensional system with a continuous attractor that has the shape of a rectangle, we find that FINDR discovers these structures (Extended Data Figure 1). Furthermore, for the dynamical system with a rectangular attractor, the width and length of the rectangle were roughly preserved in the inferred latent representation (Extended Data Figure 1c). We find similar results for a 3-dimensional dynamical system with a continuous attractor of the rectangular prism shape (Extended Data Figure 1d-g). These results suggest that FINDR accurately infers latent dynamics with attractors of different geometries and dimensionalities, while preserving distance in latent space.

3.2. FINDR performs competitively against existing methods in capturing real neural population responses

To evaluate FINDR's performance relative to existing methods in predicting the heterogenous responses of individual held-out neurons, we applied FINDR, switching linear dynamical systems model (SLDS; Linderman et al. (2017)), recurrent switching linear dynamical systems model (rSLDS; Linderman et al. (2017)), autoLFADS (Keshtkaran et al., 2022; Sedler & Pandarinath, 2023), and Gaussian Process Factor Analysis model (GPFA; Yu et al. (2008)) to a dataset comprising 67 choice-selective neurons, selected from a larger population of 464 simultaneously recorded neurons from dorsomedial frontal cortex (dmFC) and medial prefrontal cortex (mPFC) of a rat engaged in a decision-making task across 448 trials (Luo et al., 2023). On each trial, the rat listens to two simultaneous, randomly timed auditory click trains played from loudspeakers on its left and right. At the end of the stimulus, it turns to the side that had the greater total number of clicks for water reward. The spike trains and the auditory click times given to the models were aligned to the stimulus onset (Figure 3a; for a more detailed description of the task and the selection criteria for neurons, see Luo et al. (2023)). The spike trains and auditory click times were binned using a bin width of 10ms, and the binned click times from the left and the right speakers were provided to the model as a 2-dimensional external input $\boldsymbol{u}$.

We held out 13 neurons (about 20%) from this dataset, and partitioned the dataset into 5 different folds, each containing a subset of trials in random order. Following Pei et al. (2021), we held out 20% of the neurons by supplying the encoder with the remaining 80% of the held-in neurons. Then, the decoder in Equation 1 reconstructed all neurons. We used 3 of these folds for training, 1 fold for validation, and the remaining 1 fold for testing. We evaluated the 5-fold cross-validated log-likelihood of held-out neural activity to measure model performance. This is a standard metric for evaluation known as co-smoothing (Macke et al., 2011; Pei et al., 2021). When we assess model performance on 5-fold cross-validated co-smoothing across latent dimensions from $L = 1$ to $L = 6$, we find that the log-likelihood for FINDR is consistently higher than existing models when $L < 3$ (Figure 3b; left). We also use the 5-fold cross-validated coefficient of determination (R^2) to evaluate the

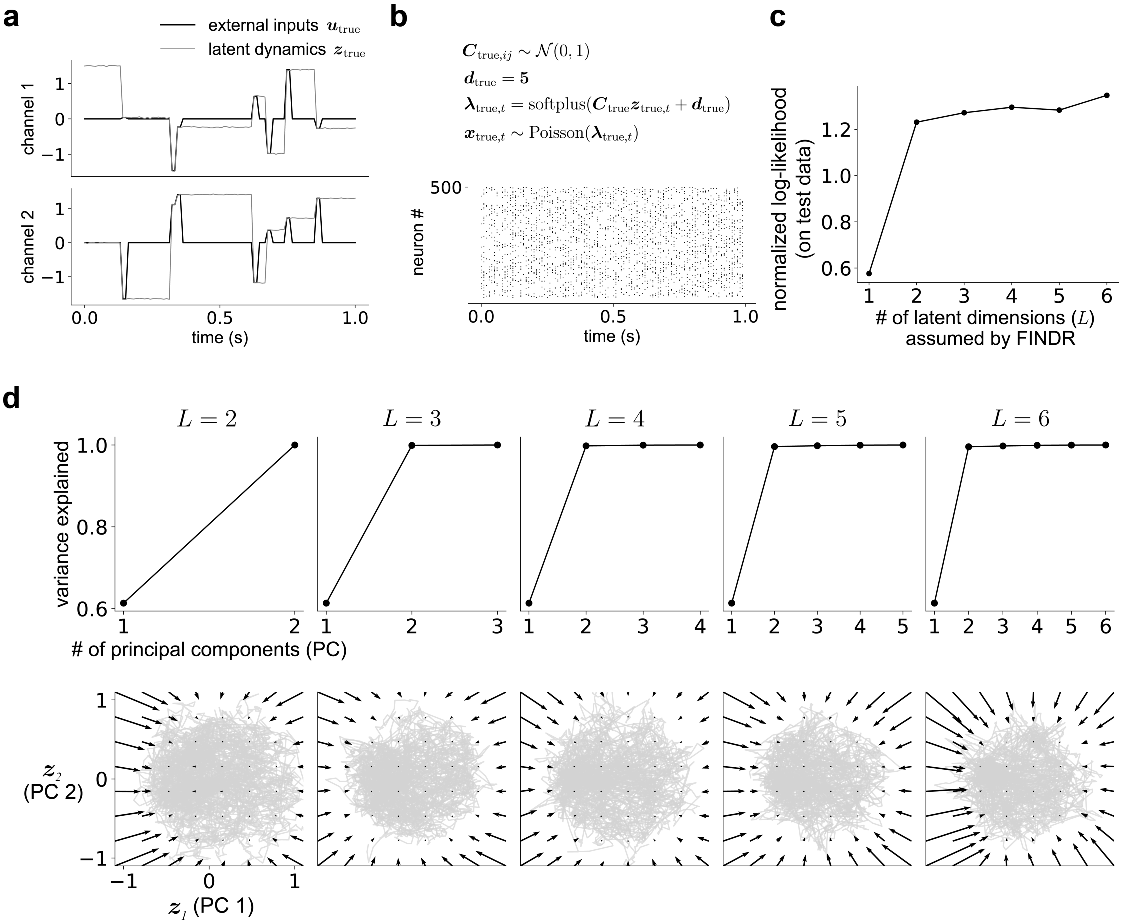


Figure 2. FINDR infers approximately continuous attractors that reflect how the synthetic neural population stores external inputs. **a**, An example trial with external inputs $\boldsymbol{u}_{\text{true}}$ in each channel shown in black, and the state trajectory of the dynamical system that maintains the previous pulse value shown in gray. **b**, An example trial of the simulated population spike trains. **c**, normalized log-likelihood evaluated on the test data as a function of the latent dimensions (L) assumed by FINDR. **d**, Principal component analysis (PCA) on the inferred latent trajectories across FINDR models assuming $L = 2, 3, ..., 6$ show that the first two PCs are sufficient to capture most of the variance in the latent trajectories. The inferred flow fields projected onto the first two PCs are also consistent across FINDR models assuming $L = 2, 3, ..., 6$. Gray lines represent latent trajectories.

match between observed and model-predicted peristimulus time histograms (PSTHs) of held-out individual neurons (Figure 3b; right). We included further details on model evaluation metrics in Appendix A.4.

Figure 3c shows example neurons' activity averaged across trials sorted by whether the stimulus favors a leftward or a rightward choice (evidence-sign conditioned PSTH), and the held-out prediction of this conditioned PSTH from FINDR with $L = 6$. The R^2 between the observed PSTH and FINDR's 5-fold cross-validated prediction is computed for each held-out neuron, and shown as a histogram, with markers indicating where the example neurons fall within this distribution (Figure 3c; with neuron 2 being the median of this distribution and what is shown as the blue circle in the right panel of Figure 3b for $L = 6$). We repeat this procedure for SLDS, rSLDS, autoLFADS, and GPFA across different latent dimensions, and find that FINDR consistently outperforms competing models under this metric (Figure 3d, Extended Data Figure 2). These results together suggest that FINDR can predict held-out neural population activity from latent dynamics with fewer dimensions compared to existing methods.

3.3. FINDR reveals dynamically consistent latent representations across cross-validation folds

That FINDR can capture neural population activity using a latent representation with a lower number of dimensions compared to existing methods allows us to inspect the nature

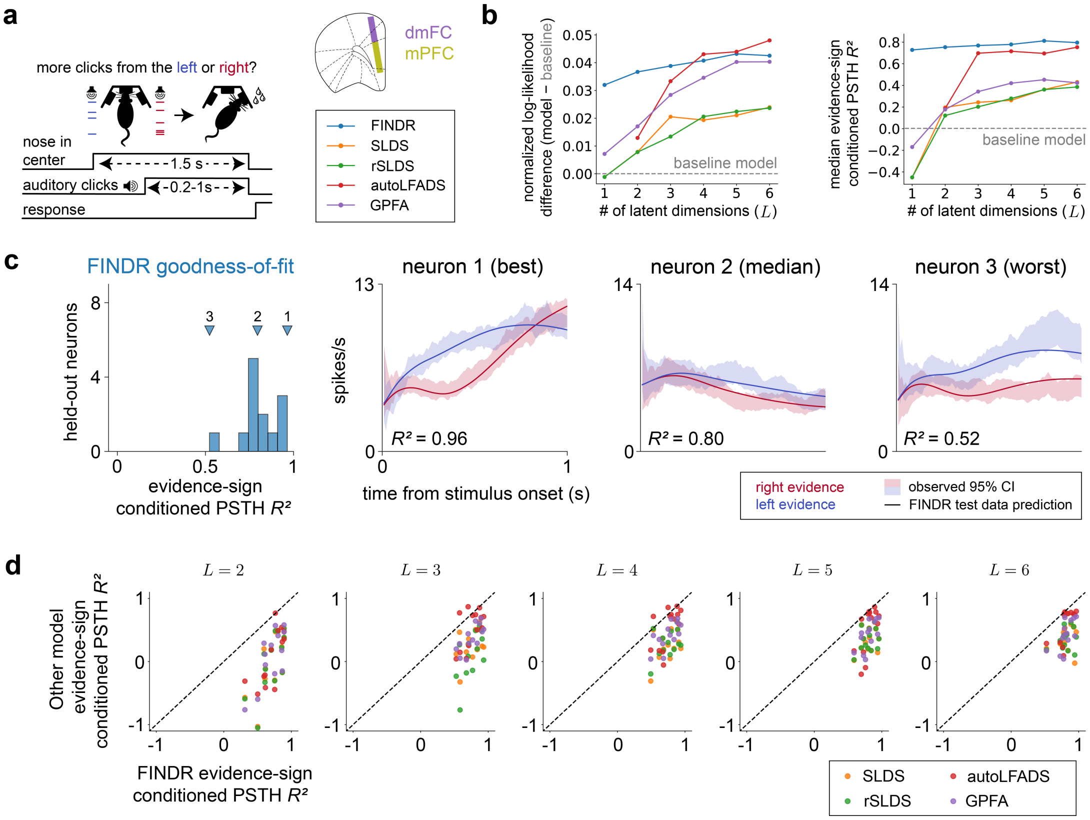


Figure 3. FINDR outperforms SLDS, rSLDS, autoLFADS, and GPFA in reconstructing the neural population activity of a rat performing perceptual decision-making. **a**, Neurons from dorsomedial frontal cortex (dmFC) and medial prefrontal cortex (mPFC) were recorded while the rat listened to a stream of click trains from its left and right sides of the operant chamber, and oriented to the side that had more clicks to receive water reward. **b**, 5-fold cross-validated normalized log-likelihood difference score (Pei et al. (2021); also known as "co-smoothing"; left) and 5-fold cross-validated median evidence-sign conditioned PSTH R^2 (right) on held-out neurons across different latent dimensions for FINDR, SLDS, rSLDS, autoLFADS, and GPFA. For SLDS and rSLDS, we consider only the best-performing model among models assuming the number of discrete latent states $= \{1, 2, 3, 4\}$. For autoLFADS, L corresponds to the factor dimension, not the size of the generator recurrent neural network (RNN). AutoLFADS with $L = 1$ fails to train. The baseline model here is defined as a constant firing rate model with the constant being the mean of the observed neural activity. **c**, FINDR with $L = 6$ captures the complex trial-averaged temporal profiles of individual neurons in mPFC and dmFC. The goodness-of-fit is measured using the R^2. **d**, We perform an analysis similar to **c** for other models and compare the R^2 obtained from these models and the R^2 from FINDR.

of the learned representation in an intuitive and interpretable manner. When we compute the R^2 between the observed and FINDR-reconstructed PSTHs for all neurons in the dataset in Figure 3a, we find that two latent dimensions are sufficient to describe the data (Extended Data Figure 3). Therefore, we focus on $L = 2$ for analyses of the learned dynamics.

In Figure 4b, we explicitly visualize the flow field inferred by FINDR with $L = 2$, and evaluate consistency in a manner similar in spirit to Genkin & Engel (2020). We find that FINDR learns dynamical representations that are consistent across 5 different folds (i.e., different training and test splits of the same data). Specifically, across all folds, we find that they consistently have two slow points. We see that on average, given enough time (4–5s), the latent state falls into the slow point in the upper part of the state space when the stimulus favors rightward choice and that it falls into the slow point in the bottom part of the state space when the stimulus favors leftward choice (Extended Data Figure 4). Furthermore, we find that the learned representations were similar for the FINDR model with the second-best set of hyperparameters, and across different random seeds for network initialization and the order in which the mini-batches of the dataset were supplied to the model during training (Extended Data Figure 7). All of our analyses with FINDR in this Section involved learning the task-relevant and -irrelevant components separately (Section 2.1). We find that we get less consistent and interpretable dynamical representations when we do not learn the task-irrelevant component (Extended Data Figure 5).

Do we find similar slow-point structures across folds for alternative dynamical models? We evaluate this in autoLFADS, which performs similarly to FINDR in predicting neural responses (Figure 4a). For each fold's autoLFADS, we ran the trained generator forward in time for 5s, starting from the initial conditions inferred from the encoder. We found that while the majority of autoLFADS states reached approximate steady-states by 5s, they did not form two clusters as would be expected from bistable attractors, both for folds 1 and 2 (Extended Data Figure 8).

To quantify consistency across folds, we sorted single-trial trajectories by evidence sign and computed the trial average of each group. Then, we calculated Pearson's $|r|$ of these trajectories between fold 1 and 2 for each latent axis and took the average of $|r|$ across the axes (we will refer to this metric as $\langle |r| \rangle$). Note that the axes here are defined by the principal components of the latents. With this, FINDR folds were consistent by 0.99, while autoLFADS folds were consistent by 0.53. The consistency score for autoLFADS increased to 0.99 if we linearly transform autoLFADS fold 1 to match fold 2. However, this linear transform stretches and rotates the latent space, so the distance in latent space is no longer preserved in the neural space (ignoring softplus). This suggests that while autoLFADS is *topologically* consistent, it is not *geometrically* consistent as in FINDR. To further quantify *dynamical* consistency, i.e., to quantify whether we consistently find two slow points across folds, we evaluated whether the distribution of the fold-1 states in approximate steady-state match the distribution of the fold-2 states. We affine transformed the autoLFADS latent trajectories from 4–5s in fold 1 to match those in fold 2, and when we computed $\langle |r| \rangle$, we found that they were not consistent ($\langle |r| \rangle = 0.22$). This suggests that even when the autoLFADS factor trajectories across folds are topologically consistent ($\langle |r| \rangle = 0.99$), this does not guarantee that the underlying dynamics that generated the trajectories by autoLFADS are consistent. Using a similar procedure for FINDR, we found $\langle |r| \rangle = 0.94$, consistent with the visualization in Extended Data Figure 4.

We additionally performed analyses with a prominent latent representation learning method, CEBRA (Schneider et al., 2023), and found that the parts of the state space traversed by the trajectories depend on evidence strength (Figure 4d). We also saw that the first two folds were topologically consistent by $\langle |r| \rangle = 0.99$. However, CEBRA and related methods (e.g., Chen et al. (2025)) do not learn dynamical representations (i.e., Equation (2)), and, unlike FINDR, it is difficult to perform fixed-point analysis on the latents or evaluate dynamical consistency. Another key distinction between FINDR and CEBRA is that, for this dataset, we find that sensory inputs perturb dynamics roughly along PC 1 in the latent space of FINDR, but this would be difficult to know using CEBRA.

Similar to the analysis in Figure 2d, when we fit FINDR models that assume different latent dimensions ($L = 2, 3, ..., 6$), and project the inferred latent trajectories to the first two PCs, we also see consistency across dimensions (Extended Data Figure 6). This suggests that FINDR can discover consistent and interpretable task-relevant latent dynamics.

Our analyses suggest that FINDR representations are consistent across folds—not only topologically but also geometrically—and reveal dynamical consistency, specifically two slow points associated with left/right choices. Among all methods tested, only FINDR achieves both 1) strong performance on neural data, and 2) discovery of consistent low-D dynamical representation, with interpretable slow points. While we find empirical evidence that FINDR can discover consistent representations, consistency is not theoretically guaranteed and should be verified empirically on new datasets.

4. Discussion

With recent advances that enable recording from a large population of neurons (Jun et al., 2017; Steinmetz et al., 2021; Stringer et al., 2019; Siegle et al., 2021), many methods (e.g., Macke et al. (2011); Linderman et al. (2017); Pandarinath et al. (2018)) have been introduced to infer the dynamics underlying the recorded population activity (e.g., Vinograd et al. (2024); Liu et al. (2024)). Benchmarks (Pei et al., 2021; Versteeg et al., 2025) have also been introduced to facilitate rigorous and standardized comparisons between these methods.

FINDR improves on the previous methods by enabling flexible inference of stochastic, nonlinear latent dynamics, while also separating task-relevant and -irrelevant components. In a population spike train dataset from frontal cortex of the rat during perceptual decision-making, these two features allowed FINDR to capture the complex, heterogeneous neuronal responses, and discover dynamically consistent and interpretable latent representations that we can easily relate to the task computation.

FINDR is a general dynamical inference framework that goes beyond the particular use case shown here (perceptual decision-making), and supports broader application to other types of neural computation, such as orientation tuning in visual cortex (Hubel & Wiesel, 1959) or rotational trajectories in motor cortex during movement (Churchland et al., 2012). The precise dynamics underlying orientation tuning or motor control remain unclear (Khona & Fiete, 2022). FINDR provides a new tool for investigating dynamics in

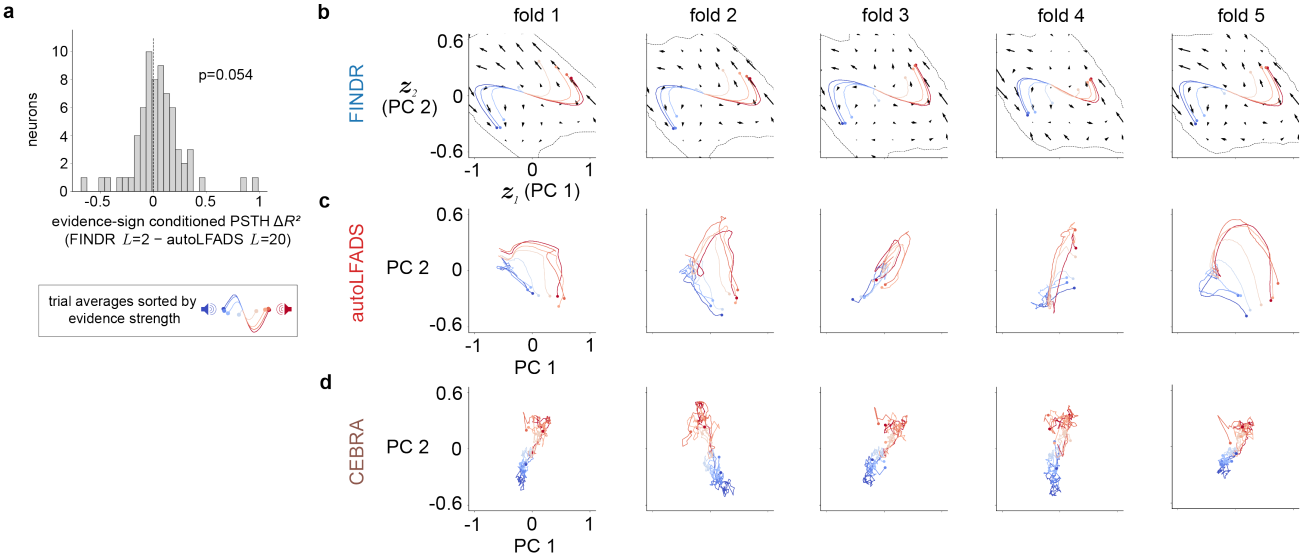


Figure 4. FINDR reveals slow points in the latent dynamics of frontal cortex during perceptual decision-making, consistently across different training and test splits of the same data. **a**, FINDR with $L = 2$ and autoLFADS with $L = 20$ are similar in how well they capture the heterogeneous responses of individual neurons, when we evaluate their performance using the 5-fold cross-validated evidence-sign conditioned PSTH R^2. **b**, To show how the decision process depends on the input, trials are sorted by their evidence strength, and we show the trial-averaged trajectories as colored lines. FINDR-inferred flow fields consistently reveal slow points across 5 different folds used for computing the PSTH R^2 in **a**. To focus on the portion of the state space that is relevant to computation, we show only the region occupied by at least 50 of 5000 simulated single-trial latent trajectories—generated from the learned dynamics in Equation (2) for 1s—outlined by the dotted line. **c**, Trial-averaged trajectories from autoLFADS, projected onto the first two PCs. **d**, Trial-averaged trajectories from CEBRA, projected onto the first two PCs.

these domains.

While we expect FINDR to be generally applicable to a broad range of neural population data, its performance may vary across datasets. As a deep learning-based model, FINDR performs best with datasets that have a large number of simultaneously recorded neurons and many repeated trials. While the exact neuron and trial count that give good performance may vary depending on the dynamics in the dataset and the firing rates, generally an increase in the number of neurons should make FINDR's estimate of each single-trial dynamical trajectory more accurate, while an increase in the number of trials should make FINDR's estimate of the flow field more accurate, because FINDR has more latent trajectories that traverse the latent space to infer the flow field from. Future work could incorporate uncertainty estimates of the inferred flow field, as demonstrated by the recent Gaussian process-based switching linear dynamical system model (Hu et al., 2024).

Additional extensions may include inferring external inputs (Schimel et al., 2022), developing more expressive decoders (Gao et al., 2016; Versteeg et al., 2024; Abbaspourazad et al., 2024), supporting online learning of nonlinear dynamics (Zhao & Park, 2020), or integrating the learning of representational similarity between dynamical systems (Gosztolai et al., 2025) into the framework presented here.

The low-dimensional dynamical representation of neural population activity discovered by FINDR can serve as a bridge between the more fine-grained, neuronal-level representation and the higher-level, algorithmic representation of task computation. Future work could strengthen these cross-level connections. For example, recent studies have started to examine the relationship between the low-rank connectivity structure of a recurrent neural network (RNN) and its low-dimensional dynamics when trained on tasks and neural data (Valente et al., 2022; Pals et al., 2024; Langdon & Engel, 2025). In Luo et al. (2023), the authors show that low-dimensional dynamical representations inferred from data can facilitate the design of parsimonious, algorithmic-level models of neural computation, such as their multi-mode drift-diffusion model (MMDDM). Integrating these levels of description into a unified framework is an interesting future direction. These directions highlight how new neural population dynamics inference methods like FINDR are a promising approach for bridging the gap between neuronal-level mechanistic descriptions and the higher-level algorithmic descriptions of neural function.

Acknowledgements

We thank Jesse Kaminsky, Belle Liu, Chethan Pandarinath, Andrew Sedler, Chris Versteeg, and Iman Wahle for discussions. We thank Jonathan Halverson for help with using the Princeton HPC clusters. This work was supported by grants from NIH R01MH108358, the Simons Collaboration on the Global Brain (SCGB AWD543027), and a U19 NIHNINDS BRAIN Initiative Award (5U19NS104648).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning and Computational Neuroscience. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Abbaspourazad, H., Erturk, E., Pesaran, B., and Shanechi, M. M. Dynamical flexible inference of nonlinear latent factors and structures in neural population activity. *Nature Biomedical Engineering*, 2024.

Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2007.

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. Understanding disentangling in β-VAE. *arXiv*, 2018.

Chen, C., Yang, Z., and Wang, X. Neural embeddings rank: Aligning 3d latent dynamics with movements. *Advances in Neural Information Processing Systems*, 2025.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural Ordinary Differential Equations. *Advances in Neural Information Processing Systems*, 2018.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv*, 2014.

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A., and Bengio, Y. A recurrent latent variable model for sequential data. *Advances in Neural Information Processing Systems*, 2015.

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

Duncker, L. and Sahani, M. Dynamics on the manifold: Identifying computational dynamical activity from neural population recordings. *Current Opinion in Neurobiology*, 70:163–170, 2021.

Duncker, L., Bohner, G., Boussard, J., and Sahani, M. Learning interpretable continuous-time models of latent stochastic dynamical systems. *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Elfwing, S., Uchibe, E., and Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 2018.

Gallego, J. A., Perich, M. G., Naufel, S. N., Ethier, C., Solla, S. A., and Miller, L. E. Cortical population activity within a preserved neural manifold underlies multiple motor behaviors. *Nature Communications*, 2018.

Gao, Y., Archer, E. W., Paninski, L., and Cunningham, J. P. Linear dynamical neural population models through nonlinear embeddings. *Advances in Neural Information Processing Systems*, 2016.

Genkin, M. and Engel, T. A. Moving beyond generalization to accurate interpretation of flexible models. *Nature Machine Intelligence*, 2020.

Genkin, M., Hughes, O., and Engel, T. A. Learning nonstationary langevin dynamics from stochastic observations of latent trajectories. *Nature Communications*, 2021.

Gholami, A., Keutzer, K., and Biros, G. ANODE: Unconditionally Accurate Memory-Efficient Gradients for Neural ODEs. *International Joint Conferences on Artificial Intelligence*, 2019.

Gosztolai, A., Peach, R. L., Arnaudon, A., Barahona, M., and Vandergheynst, P. MARBLE: interpretable representations of neural population dynamics using geometric deep learning. *Nature Methods*, 2025.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-VAE: Learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations*, 2017.

Hu, A., Zoltowski, D. M., Nair, A., Anderson, D., Duncker, L., and Linderman, S. W. Modeling latent neural dynamics with gaussian process switching linear dynamical systems. *Advances in Neural Information Processing Systems*, 2024.

Hubel, D. H. and Wiesel, T. N. Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology*, 148(3):574–591, 1959.

Jun, J. J., Steinmetz, N. A., Siegle, J. H., et al. Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 551:232–236, 2017.

Kato, S., Kaplan, H. S., Schrödel, T., Skora, S., Lindsay, T. H., Yemini, E., Lockery, S., and Zimmer, M. Global brain dynamics embed the motor command sequence of caenorhabditis elegans. *Cell*, 163(3):656–669, 2015.

Keshtkaran, M. R., Sedler, A. R., Chowdhury, R. H., Tandon, R., Basrai, D., Nguyen, S. L., Sohn, H., Jazayeri, M., Miller, L. E., and Pandarinath, C. A large-scale neural network training framework for generalized estimation of single-trial population dynamics. *Nature Methods*, 19: 1572–1577, 2022.

Khona, M. and Fiete, I. R. Attractor and integrator networks in the brain. *Nature Reviews Neuroscience*, 2022.

Kim, T. D., Luo, T. Z., Pillow, J. W., and Brody, C. D. Inferring latent dynamics underlying neural population activity via neural differential equations. *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Kim, T. D., Can, T., and Krishnamurthy, K. Trainability, Expressivity and Interpretability in Gated Neural ODEs. *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *International Conference on Learning Representations*, 2014.

Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. Self-normalizing neural networks. *Advances in Neural Information Processing Systems*, 2017.

Krishnan, R. G., Shalit, U., and Sontag, D. Structured inference networks for nonlinear state space models. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Langdon, C. and Engel, T. A. Latent circuit inference from heterogeneous neural responses during cognitive tasks. *Nature Neuroscience*, 2025.

Li, X., Wong, T.-K. L., Chen, R. T. Q., and Duvenaud, D. Scalable gradients for stochastic differential equations. *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, 2020.

Linderman, S., Johnson, M., Miller, A., Adams, R., Blei, D., and Paninski, L. Bayesian Learning and Inference in Recurrent Switching Linear Dynamical Systems. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017.

Liu, M., Nair, A., Coria, N., Linderman, S. W., and Anderson, D. J. Encoding of female mating dynamics by a hypothalamic line attractor. *Nature*, 2024.

Loshchilov, I. and Hutter, F. SGDR: Stochastic Gradient Descent with Warm Restarts. *arXiv*, 2017.

Luo, T. Z., Kim, T. D., Gupta, D., Bondy, A. G., Kopec, C. D., Elliot, V. A., DePasquale, B., and Brody, C. D. Transitions in dynamical regime and neural mode underlie perceptual decision-making. *bioRxiv*, 2023.

Macke, J. H., Buesing, L., Cunningham, J. P., Yu, B. M., Shenoy, K. V., and Sahani, M. Empirical models of spiking in neural populations. *Advances in Neural Information Processing Systems*, 2011.

Nassar, J., Linderman, S. W., Bugallo, M., and Park, I. M. Tree-structured recurrent switching linear dynamical systems for multi-scale modeling. *International Conference on Learning Representations*, 2019.

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

Onken, D. and Ruthotto, L. Discretize-optimize vs. optimize-discretize for time-series regression and continuous normalizing flows. *arXiv*, 2020.

Pals, M., Sağtekin, A. E., Pei, F. C., Gloeckler, M., and Macke, J. H. Inferring stochastic low-rank recurrent neural networks from neural data. *Advances in Neural Information Processing Systems*, 2024.

Pandarinath, C., Gilja, V., Blabe, C. H., Nuyujukian, P., Sarma, A. A., Sorice, B. L., Eskandar, E. N., Hochberg, L. R., Henderson, J. M., and Shenoy, K. V. Neural population dynamics in human motor cortex during movements in people with als. *eLife*, 4:e07436, 2015.

Pandarinath, C., O'Shea, D. J., Collins, J., et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature Methods*, 15:805–815, 2018.

Park, I. M., Meister, M. L., Huk, A. C., and Pillow, J. W. Encoding and decoding in parietal cortex during sensorimotor decision-making. *Nature neuroscience*, 17(10): 1395–1403, 2014.

Pei, F., Ye, J., Zoltowski, D., Wu, A., Chowdhury, R. H., Sohn, H., O'Doherty, J. E., Shenoy, K. V., Kaufman, M. T., Churchland, M., Jazayeri, M., Miller, L. E., Pillow, J., Park, I. M., Dyer, E. L., and Pandarinath, C. Neural Latents Benchmark '21: Evaluating latent variable models of neural population activity. *Advances in Neural Information Processing Systems*, 2021.

Pillow, J. W., Paninski, L., Uzzell, V. J., Simoncelli, E. P., and Chichilnisky, E. J. Prediction and decoding of retinal ganglion cell responses with a probabilistic spiking model. *Journal of Neuroscience*, 25(47):11003–11013, 2005.

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

Ramachandran, P., Zoph, B., and Le, Q. V. Searching for activation functions. *arXiv*, 2017.

Rigotti, M., Barak, O., Warden, M. R., Wang, X.-J., Daw, N. D., K., M. E., and Fusi, S. The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497:585–590, 2013.

Safaie, M., Chang, J. C., Park, J., Miller, L. E., Dudman, J. T., G., P. M., and Gallego, J. A. Preserved neural dynamics across animals performing similar behaviour. *Nature*, 2023.

Sani, O. G., Abbaspourazad, H., Wong, Y. T., Bijan, P., and Shanechi, M. M. Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification. *Nature Neuroscience*, 2021.

Schimel, M., Kao, T. C., Jensen, K. T., and Hennequin, G. ilqr-vae: control-based learning of input-driven dynamics with applications to neural data. *International Conference on Learning Representations*, 2022.

Schneider, S., Lee, J. H., and Mathis, M. W. Learnable latent embeddings for joint behavioural and neural analysis. *Nature*, 2023.

Sedler, A. R. and Pandarinath, C. lfads-torch: A modular and extensible implementation of latent factor analysis via dynamical systems. *arXiv*, 2023.

Sedler, A. R., Versteeg, C., and Pandarinath, C. Expressive architectures enhance interpretability of dynamics-based neural population models. *Neurons, Behavior, Data analysis and Theory*, 2023.

Siegle, J. H., Jia, X., Durand, S., et al. Survey of spiking in the mouse visual system reveals functional hierarchy. *Nature*, 592:86–92, 2021.

Steinmetz, N. A., Aydin, C., Lebedeva, A., et al. Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings. *Science*, 372(6539):eabf4588, 2021.

Stringer, C., Pachitariu, M., Steinmetz, N., Reddy, C. B., Carandini, M., and Harris, K. D. Spontaneous behaviors drive multidimensional, brainwide activity. *Science*, 364 (6437):255–258, 2019.

Sussillo, D. and Barak, O. Opening the Black Box: Low-Dimensional Dynamics in High-Dimensional Recurrent Neural Networks. *Neural Computation*, 25(3):626–649, 2013.

Valente, A., Pillow, J. W., and Ostojic, S. Extracting computational mechanisms from neural data using low-rank rnns. *Advances in Neural Information Processing Systems*, 2022.

Versteeg, C., Sedler, A. R., McCart, J. D., and Pandarinath, C. Expressive dynamics models with nonlinear injective readouts enable reliable recovery of latent features from neural activity. *Proceedings of the 2nd NeurIPS Workshop on Symmetry and Geometry in Neural Representations*, 2024.

Versteeg, C., McCart, J. D., Ostrow, M., Zoltowski, D. M., Washington, C. B., Driscoll, L., Codol, O., Michaels, J. A., Linderman, S. W., Sussillo, D., and Pandarinath, C. Computation-through-dynamics benchmark: Simulated datasets and quality metrics for dynamical models of neural activity. *bioRxiv*, 2025.

Vinograd, A., Nair, A., Kim, J. H., Linderman, S. W., and Anderson, D. J. Causal evidence of a line attractor encoding an affective state. *Nature*, 2024.

Vorontsov, E., Trabelsi, C., Kadoury, S., and Pal, C. On orthogonality and learning recurrent networks with long term dependencies. *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Vyas, S., Golub, M. D., Sussillo, D., and Shenoy, K. V. Computation through neural population dynamics. *Annual Review of Neuroscience*, 43:249–275, 2020.

Wang, Y., Blei, D., and Cunningham, J. P. Posterior collapse and latent variable non-identifiability. *Advances in Neural Information Processing Systems*, 2021.

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

Zhao, Y. and Park, I. M. Variational online learning of neural dynamics. *Frontiers in Computational Neuroscience*, 14: 71, 2020.

Zoltowski, D., Pillow, J., and Linderman, S. A general recurrent state space framework for modeling neural dynamics during decision-making. *Proceedings of the 37th International Conference on Machine Learning*, 2020.

A. Appendix

A.1. Model Specification

Dynamics in a neural population may be associated with not only the task that the animal performs but also other factors that are not relevant to the task. Therefore, in FINDR, we distinguish between the task-irrelevant dynamics $\boldsymbol{d}_t$ and task-relevant low-dimensional dynamics $\boldsymbol{z}_t$ and use a separate inference procedure for the two. These latent variables and task stimulus $\boldsymbol{u}_t$ generate spike trains $\boldsymbol{y}_t$ from N recorded neurons. We assume that the total length of a trial is $\mathcal{T}$, and we bin $\mathcal{T}$ into T bins, with each of the bins having the same width Δt. We assume that there are a total of M trials in an experimental session. We denote $t \in \{1, ..., T\}$ representing the t-th time bin within a trial, and $m \in \{1, ..., M\}$ representing the m-th trial. In all tasks that we consider in this work, $\mathcal{T} = 1$ second, with $\Delta t = 0.01$ second. In Section 3.3, we only analyzed the epoch of a trial from stimulus onset to right before movement initiation, which was variable across trials and less than or equal to 1 second. In Section 3.2, for each trial, we looked at the first 1 second from stimulus onset. Thus, if the stimulus duration was less than 1 second on a given trial, we also considered spiking activity during movement. We fixed the lengths of all trials to be the same in this Section to allow comparisons to existing models (datasets with variable trial lengths are not yet supported in some model implementations we consider).

A.1.1. Inference of task-irrelevant dynamics

To model task-irrelevant fluctuations in an individual neuron's firing rate *across* a total of M trials, we first compute each neuron n's average firing rate for each trial m with

$$\bar{\boldsymbol{y}}_m^{(n)} = \frac{\text{\# of spikes in neuron } n \text{ in trial } m}{\text{duration of trial } m \text{ (in seconds)}}. \tag{8}$$

Here we let the n-th element of the N-dimensional vector $\bar{\mathbf{y}}_m$ to be $\bar{\boldsymbol{y}}_m^{(n)}$. We partition M randomized trials into 5 equal subsets, where $3/5$ of the M trials are used for training ($\mathcal{D}_{\text{train}}$), $1/5$ for validation ($\mathcal{D}_{\text{val}}$), and $1/5$ for testing ($\mathcal{D}_{\text{test}}$). For each neuron n, we fit a linear basis function model (Bishop, 2007) with the following loss function:

$$\min_{w_{1:D_n^{\text{across}}}^{\text{across},(n)}} \sum_{m \in \mathcal{D}_{\text{train}}} ||\bar{\boldsymbol{y}}_m^{(n)} - \boldsymbol{d}_m^{\text{across},(n)}||^2 + p^{\text{across}} \sum_{j=1}^{D_n^{\text{across}}} ||w_j^{\text{across},(n)}||^2, \tag{9}$$

where

$$\begin{aligned}
\boldsymbol{d}_m^{\text{across},(n)} &= \sum_{j=1}^{D_n^{\text{across}}} w_j^{\text{across},(n)} \varphi_j^{\text{across},(n)}(m), \\
\varphi_j^{\text{across},(n)}(m) &= \frac{1}{2}\left(\cos\left(H\left(\frac{\pi(D_n^{\text{across}} - 1)}{2S_{\max}}(S(m) - \text{disp}^{\text{across}}[j])\right)\right) + 1\right), \\
\text{disp}^{\text{across}}[j] &= \text{linspace}(S_{\min}, S_{\max}, D_n^{\text{across}})[j].
\end{aligned} \tag{10}$$

Here,

$$H(x) = \begin{cases} \pi, & \text{if } \pi < x \\ x, & \text{if } -\pi \le x \le \pi \\ -\pi & \text{if } x < -\pi \end{cases}, \tag{11}$$

and $S(m)$ is the time stamp of the onset of trial m (which lies between the beginning of the session, $S_{\min} = 0$s, and $S_{\max} = 10,000$s, and rounded to the nearest second; we assume no session runs more than $S_{\max}$). Here, linspace is a function that returns D_n^{across} evenly spaced numbers over the interval $(S_{\min}, S_{\max})$. The hyperparameters $D_n^{\text{across}} \in \{4, 5, ..., 10\}$ and $p^{\text{across}} \in \{0.1, 0.1^2, 0.1^3, 0.1^4, 0.1^5\}$ are optimized by evaluating the loss on $\mathcal{D}_{\text{val}}$.

To model task-irrelevant fluctuations in an individual neuron's firing rate *within* each trial, we again fit, for each neuron n, a linear basis function model (Bishop, 2007) with the following loss function:

$$\min_{w_{1:D_n^{\text{within}}}^{\text{within},(n)}} \sum_{m \in \mathcal{D}_{\text{train}}} ||\boldsymbol{y}_{m,t}^{(n)} - \boldsymbol{d}_m^{*,\text{across},(n)} - \boldsymbol{d}_t^{\text{within},(n)}||^2 + p^{\text{within}} \sum_{j=1}^{D_n^{\text{within}}} ||w_j^{\text{within},(n)}||^2, \tag{12}$$

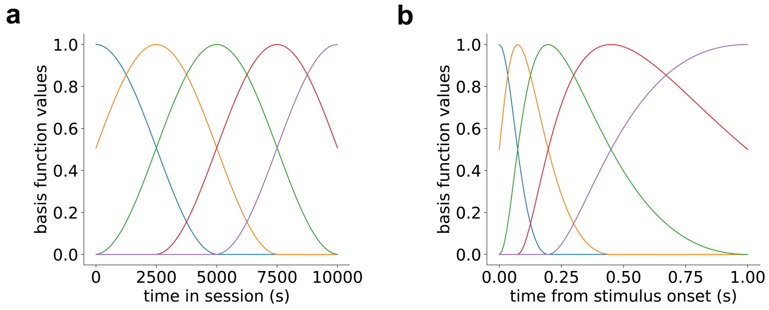


Figure 5. Examples of raised cosine basis functions (Pillow et al., 2005; 2008; Park et al., 2014) used in the inference of task-irrelevant dynamics. **a**, $\varphi_j^{\text{across},(n)}(m)$ when $D_n^{\text{across}} = 5$. **b**, $\varphi_j^{\text{within},(n)}(t)$ when $D_n^{\text{within}} = 5$.

where

$$\begin{aligned}
\boldsymbol{d}_t^{\text{within},(n)} &= \sum_{j=1}^{D_n^{\text{within}}} w_j^{\text{within},(n)} \varphi_j^{\text{within},(n)}(t), \\
\varphi_j^{\text{within},(n)}(t) &= \frac{1}{2}\left(\cos\left(H\left(\frac{\pi(D_n^{\text{within}}-1)}{2\,\text{arsinh}(11.88)}(\text{arsinh}(0.12(t-1)) - \text{disp}^{\text{within}}[j])\right)\right)+1\right), \text{ for } t \in \{1,...,T\}, \\
\text{disp}^{\text{within}}[j] &= \text{linspace}(0, \text{arsinh}(11.88), D_n^{\text{within}})[j].
\end{aligned} \tag{13}$$

Here, $\boldsymbol{d}_m^{*,\text{across},(n)}$ is the solution to Equation (9). The hyperparameters $D_n^{\text{within}} \in \{5, 6, ..., 10\}$ and $p^{\text{within}} \in \{0.1, 0.1^2, 0.1^3, 0.1^4, 0.1^5\}$ are optimized by evaluating the loss on $\mathcal{D}_{\text{val}}$. After obtaining the solution of Equation (12), $\boldsymbol{d}_t^{*,\text{within},(n)}$, we set $\boldsymbol{d}_{m,t}^{(n)} = \boldsymbol{d}_m^{*,\text{across},(n)} + \boldsymbol{d}_t^{*,\text{within},(n)}$. Once we do this procedure for all neurons n, we get $\boldsymbol{d}_{m,t}$, which is used in the task-relevant dynamics inference in Appendix A.1.2.

A.1.2. Inference of task-relevant dynamics

For simplicity, going forward we suppress m in our notation whenever we can. To maximize the log-likelihood of observing the population spike trains $\boldsymbol{y}$ given the task-related external inputs $\boldsymbol{u}$, the task-irrelevant baseline inputs $\boldsymbol{d}$ and the model parameters θ, we need to compute

$$\log p_\theta(\boldsymbol{y}_{1:T}|\boldsymbol{u}_{1:T}, \boldsymbol{d}_{1:T}) = \log\left(\int \prod_{t=1}^{T} p_\theta(\boldsymbol{y}_t|\boldsymbol{z}_t, \boldsymbol{d}_t) p_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{u}_t) d\boldsymbol{z}\right). \tag{14}$$

Here, $p_\theta(\boldsymbol{y}_t|\boldsymbol{z}_t, \boldsymbol{d}_t)$ is given by Equation (1). The term $p_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{u}_t)$ is given by Equation (2). We call Equation (2) the prior process. Here, we assume that the initial condition of the latent $\boldsymbol{z}_0 = \boldsymbol{0}$. Equation (14) does not have a closed-form solution, and computing this quantity can be computationally expensive. Therefore, we instead compute the evidence lower bound (ELBO) (Kingma & Welling, 2014):

$$\log p_\theta(\boldsymbol{y}_{1:T}|\boldsymbol{u}_{1:T}, \boldsymbol{d}_{1:T}) \geq \mathbb{E}_{q_\phi}\left[\log p_\theta(\boldsymbol{y}_{1:T}|\boldsymbol{z}_{1:T}, \boldsymbol{d}_{1:T})\right] - D_{\text{KL}}\left(q_\phi(\boldsymbol{z}_{1:T}|\boldsymbol{y}_{1:T}, \boldsymbol{u}_{1:T}) \| p_\theta(\boldsymbol{z}_{1:T}|\boldsymbol{u}_{1:T})\right), \tag{15}$$

by introducing a variational posterior q_ϕ. Here, D_{KL} is the Kullback-Leibler divergence between the prior p_θ and the variational posterior q_ϕ. We can further decompose the first term into:

$$\mathbb{E}_{q_\phi}\left[\log p_\theta(\boldsymbol{y}_{1:T}|\boldsymbol{z}_{1:T}, \boldsymbol{d}_{1:T})\right] = \mathbb{E}_{q_\phi}\left[\sum_{t=1}^{T} \log p_\theta(\boldsymbol{y}_t|\boldsymbol{z}_t, \boldsymbol{d}_t)\right]. \tag{16}$$

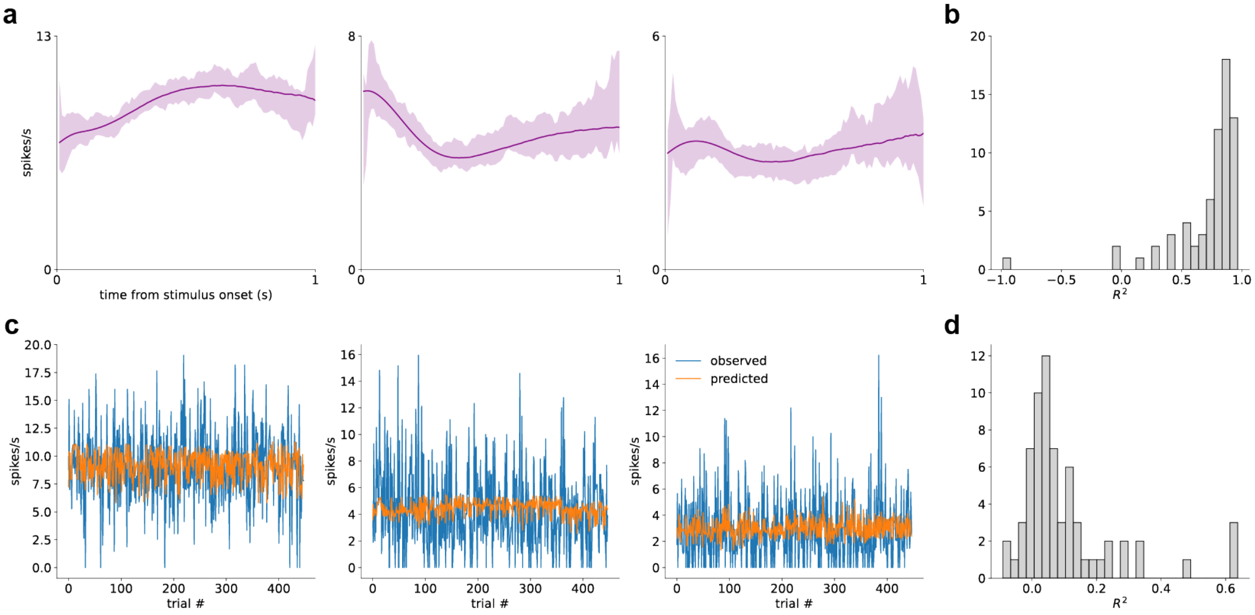


Figure 6. Estimated $\boldsymbol{d}_t^{\text{within}}$ closely matches the observed PSTH, and estimated $\boldsymbol{d}_m^{\text{across}}$ matches the true across-trial firing rate fluctuations. **a**, 5-fold cross-validated PSTH reconstruction from FINDR (in bold line) plotted against observed PSTH (in shading; 95% confidence interval) for three example neurons. **b**, Histogram of the cross-validated R^2 between the observed and FINDR PSTHs (median=0.82). **c**, 5-fold cross-validated firing rate prediction from FINDR plotted against the observed firing rate across trials for three example neurons. **d**, Histogram of the cross-validated R^2 between the observed and FINDR firing rates across trials (median=0.05). For the majority of neurons, having a time-varying baseline across trials instead of a constant baseline improves goodness-of-fit.

The KL term can also be decomposed into:

$$
\begin{aligned}
D_{\text{KL}}\left(q_\phi(\boldsymbol{z}_{1:T}|\boldsymbol{y}_{1:T},\boldsymbol{u}_{1:T})||p_\theta(\boldsymbol{z}_{1:T}|\boldsymbol{u}_{1:T})\right) &= \int d\boldsymbol{z}_{1:T} q_\phi(\boldsymbol{z}_{1:T}|\boldsymbol{y}_{1:T},\boldsymbol{u}_{1:T}) \log \frac{q_\phi(\boldsymbol{z}_{1:T}|\boldsymbol{y}_{1:T},\boldsymbol{u}_{1:T})}{p_\theta(\boldsymbol{z}_{1:T}|\boldsymbol{u}_{1:T})} \\
= \sum_{t=1}^{T} \left[\mathbb{E}_{q_\phi}\left[D_{\text{KL}}\left(q_\phi(\boldsymbol{z}_t|\boldsymbol{z}_{t-1},\boldsymbol{y}_{1:T},\boldsymbol{u}_{1:T})||p_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{t-1},,\boldsymbol{u}_t)\right)\right]\right].
\end{aligned}
\tag{17}
$$

Therefore, our objective becomes:

$$
\log p_\theta(\boldsymbol{y}_{1:T}|\boldsymbol{u}_{1:T},\boldsymbol{d}_{1:T}) \geq \sum_{t=1}^{T} \mathbb{E}_{q_\phi}\Big[\log p_\theta(\boldsymbol{y}_t|\boldsymbol{z}_t,\boldsymbol{d}_t) - D_{\text{KL}}\left(q_\phi(\boldsymbol{z}_t|\boldsymbol{z}_{t-1},\boldsymbol{y}_{1:T},\boldsymbol{u}_{1:T})\|p_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{t-1},\boldsymbol{u}_t)\right)\Big]. \tag{18}
$$

We specify q_ϕ with Equation (7), and call Equation (7) the posterior process. The initial condition $\boldsymbol{z}_0$ is again set to be $\mathbf{0}$ here. We also let the diagonal matrix $\boldsymbol{\Sigma}$ be the same in both the prior and posterior processes to make the KL divergence finite (Li et al., 2020). $\boldsymbol{e}_t \in \mathbb{R}^{2H_{\text{RNN}}}$ encodes representations of $\boldsymbol{y}_{1:T}$ and $\boldsymbol{u}_{1:T}$ using a bidirectional GRU (Cho et al., 2014):

$$
\mathbf{f}_t = \text{GRU}_{\mathbf{f},\phi}\left(\mathbf{f}_{t-1},\boldsymbol{y}_t,\boldsymbol{u}_t\right), \quad \mathbf{b}_t = \text{GRU}_{\mathbf{b},\phi}\left(\mathbf{b}_{t+1},\boldsymbol{y}_t,\boldsymbol{u}_t\right), \quad \boldsymbol{e}_t = [\mathbf{f}_t;\mathbf{b}_t], \tag{19}
$$

where $\mathbf{f}_t \in \mathbb{R}^{H_{\text{RNN}}}$ and $\mathbf{b}_t \in \mathbb{R}^{H_{\text{RNN}}}$. To train FINDR, we compute the gradient of

$$
\tilde{\mathcal{L}}(\theta,\phi) = \sum_{t=1}^{T}\Big[-\log p_\theta(\boldsymbol{y}_t|\tilde{\boldsymbol{z}}_t,\boldsymbol{d}_t) + \beta D_{\text{KL}}\left(q_\phi(\boldsymbol{z}_t|\tilde{\boldsymbol{z}}_{t-1},\boldsymbol{e}_t)\|p_\theta(\boldsymbol{z}_t|\tilde{\boldsymbol{z}}_{t-1},\boldsymbol{u}_t)\right)\Big], \tag{20}
$$

with respect to θ and ϕ using backpropagation through time (BPTT), where we sampled $\tilde{\boldsymbol{z}}_k$ once from $q_\phi(\boldsymbol{z}_k|\boldsymbol{z}_{t-1},\boldsymbol{e}_t)$ for $t \in \{1,...,T\}$. Note that we included β to the KL term in our loss $\tilde{\mathcal{L}}(\theta,\phi)$.

With the m included back into our notation, Equation (20) can be re-written as

$$\begin{aligned}\mathcal{L}(\theta,\phi) = &\sum_m \sum_t -\log p_\theta(\boldsymbol{y}_{m,t}|\tilde{\boldsymbol{z}}_{m,t}, \boldsymbol{d}_{m,t}) \\ &+ \sum_m \sum_t \beta D_{\mathrm{KL}}\left(q_\phi(\boldsymbol{z}_{m,t}|\tilde{\boldsymbol{z}}_{m,t-1}, \boldsymbol{y}_{m,1:T}, \boldsymbol{u}_{m,1:T}) || p_\theta\left(\boldsymbol{z}_{m,t}|\tilde{\boldsymbol{z}}_{m,t-1}, \boldsymbol{u}_{m,t}\right)\right),\end{aligned} \tag{21}$$

where

$$\sum_m \sum_t \beta D_{\mathrm{KL}}\left(q_\phi(\boldsymbol{z}_{m,t}|\tilde{\boldsymbol{z}}_{m,t-1}, \boldsymbol{y}_{m,1:T}, \boldsymbol{u}_{m,1:T}) || p_\theta\left(\boldsymbol{z}_{m,t}|\tilde{\boldsymbol{z}}_{m,t-1}, \boldsymbol{u}_{m,t}\right)\right) \tag{22}$$

is equal to:

$$\sum_m \sum_t \beta \Delta t \frac{1}{2}\left(\nu-\mu\right)^\top \boldsymbol{\Sigma}^{-1}\left(\nu-\mu\right), \tag{23}$$

where the arguments of the functions ν and μ are $(\tilde{\boldsymbol{z}}_{m,t-1}, \boldsymbol{e}_{m,t}, \boldsymbol{u}_{m,t})$ and $(\tilde{\boldsymbol{z}}_{m,t-1}, \boldsymbol{u}_{m,t})$, respectively.

A.1.3. Model architecture

For G and F in Equation (3), we use

$$\begin{aligned} G(\boldsymbol{z},\boldsymbol{u}) &= W_{G,0}\boldsymbol{z} + V_{G,0}\boldsymbol{u} + \boldsymbol{b}_{G,0} \\ F(\boldsymbol{z},\boldsymbol{u}) &= W_{F,1}\mathrm{SiLU}(W_{F,0}\boldsymbol{z} + V_{F,0}\boldsymbol{u} + \boldsymbol{b}_{F,0}) + \boldsymbol{b}_{F,1} \\ \boldsymbol{\Sigma} &= \sigma\left(\mathrm{diag}(\boldsymbol{s})\right), \end{aligned} \tag{24}$$

Similarly, for $\tilde{G}$ and $\tilde{F}$ in Equation (7), we use

$$\begin{aligned} \tilde{G}(\boldsymbol{z},\boldsymbol{u},\boldsymbol{e}) &= W_{\tilde{G},0}\boldsymbol{z} + V_{\tilde{G},0}\boldsymbol{u} + U_{\tilde{G},0}\boldsymbol{e} + \boldsymbol{b}_{\tilde{G},0} \\ \tilde{F}(\boldsymbol{z},\boldsymbol{u},\boldsymbol{e}) &= W_{\tilde{F},1}\mathrm{SiLU}(W_{\tilde{F},0}\boldsymbol{z} + V_{\tilde{F},0}\boldsymbol{u} + U_{\tilde{F},0}\boldsymbol{e} + \boldsymbol{b}_{\tilde{F},0}) + \boldsymbol{b}_{\tilde{F},1} \end{aligned} \tag{25}$$

where $W_{G,0}, W_{\tilde{G},0} \in \mathbb{R}^{L\times L}$, $V_{G,0}, V_{\tilde{G},0} \in \mathbb{R}^{L\times \dim(\boldsymbol{u})}$, $U_{G,0}, U_{\tilde{G},0} \in \mathbb{R}^{L\times 2H_{\text{RNN}}}$, $W_{F,0}, W_{\tilde{F},0} \in \mathbb{R}^{H_{\text{FNN}}\times L}$, $W_{F,1}, W_{\tilde{F},1} \in \mathbb{R}^{L\times H_{\text{FNN}}}$, $V_{F,0}, V_{\tilde{F},0} \in \mathbb{R}^{H_{\text{FNN}}\times \dim(\boldsymbol{u})}$, $U_{\tilde{F},0} \in \mathbb{R}^{H_{\text{FNN}}\times 2H_{\text{RNN}}}$, $\boldsymbol{b}_{G,0}, \boldsymbol{b}_{\tilde{G},0} \in \mathbb{R}^L$, $\boldsymbol{b}_{F,0}, \boldsymbol{b}_{\tilde{F},0} \in \mathbb{R}^{H_{\text{FNN}}}$, $\boldsymbol{b}_{F,1}, \boldsymbol{b}_{\tilde{F},1} \in \mathbb{R}^L$, and $\boldsymbol{s} \in \mathbb{R}^L$ are trainable parameters. Here, $2H_{\text{RNN}}$ is a hyperparameter that determines the size of the bidirectional GRU, and H_{FNN} is a hyperparameter that determines the width of the hidden layer. In Equation (19), $\mathbf{f}_0$ and $\mathbf{b}_{T+1}$ are trainable parameters. We let

$$p_\theta(\boldsymbol{y}_{m,t}|\tilde{\boldsymbol{z}}_{m,t}, \boldsymbol{d}_{m,t}) = \text{Poisson}(\Delta t \boldsymbol{\lambda}_{m,t} = \Delta t \cdot \text{softplus}(\boldsymbol{C}\tilde{\boldsymbol{z}}_{m,t} + \boldsymbol{d}_{m,t})), \tag{26}$$

where $\boldsymbol{C} \in \mathbb{R}^{N\times L}$ is trainable. For all FNNs in this paper, we use the SiLU (a.k.a. swish) activation function (Ramachandran et al., 2017; Elfwing et al., 2018).

A.1.4. Initialization

We use the initialization scheme in Kim et al. (2023) for the kernels that transform the hidden states in our gated MLPs. We use the orthogonal initializer for the kernels that transform the hidden states in the GRUs. We use the Lecun normal initializer (Klambauer et al., 2017) for the kernels that transform the input in both the GRUs and gated MLPs. Biases are i.i.d. normal with variance 10^{-6}.

A.1.5. Optimization

To train this model, we use the discrete adjoint sensitivity (i.e., standard backpropagation through time) to compute the gradient of $\mathcal{L}$ in Equation (21) with respect to $\{\theta, \phi\}$. A few studies (Gholami et al., 2019; Onken & Ruthotto, 2020) show that the discrete adjoint sensitivity produces more accurate gradients than the continuous adjoint sensitivity used in Chen et al. (2018). We train for a total of 3000 epochs and minimize loss using mini-batch gradient descent with warm restart (Loshchilov & Hutter, 2017). The learning rate increases from 0 to η linearly for 10 epochs every $D_{\text{cycle}_i} = 2^{i-1}D$ epochs, where i goes from 1 to i_{end}. After the 10 epochs, the learning rate decays in a cosine manner, where at D_{cycle_i}, the learning rate becomes 0. i_{end} is determined by the minimum $\sum_{i=1}^{i_{\text{end}}} D_{\text{cycle}_i}$ which is greater than or equal to 3000. D is set to be 200.

A.1.6. Hyperparameter grid-search

For each of the 5-folds in a single experimental session dataset, we do a grid search on the parameters $(\eta, H_{\text{FNN}}, H_{\text{RNN}})$ to identify the model that performs best when the objective is evaluated using the validation dataset. Here, $\eta \in \{10^{-2.0}, 10^{-1.625}, 10^{-1.25}, 10^{-0.875}, 10^{-0.5}\}$, $H_{\text{FNN}} \in \{30, 50, 100\}$, and $H_{\text{RNN}} \in \{50, 100, 200\}$. H_{FNN} is the number of hidden units in FNNs F_θ and F_ϕ, where both networks had a single hidden layer. H_{RNN} represents the number of units for both $\text{GRU}_{\mathbf{f},\phi}$ and $\text{GRU}_{\mathbf{b},\phi}$. Thus the total number of units for the bidirectional GRU is $2H_{\text{RNN}}$.

A.1.7. Fixed hyperparameters

We train FINDR for a total of 3000 epochs. For the first 100 epochs, we train only the first 30 time bins of the trials. Then for the next 200 epochs, we train only the first 50 time bins of the trials. Afterward, we fit all time bins in the trials. We set the coefficient of the ℓ_2 regularization on the weights of all model parameters to be 10^{-7}. We let F_θ and F_ϕ be an FNN with a single hidden layer. We set the time constant $\tau = 0.1$s. We set $\beta = 2$. We set the number of trials in a mini-batch to be 25. We set the momentum in mini-batch gradient descent to be 0.9. We perform annealing to the KL term in Equation (22). Specifically, the KL term is multiplied by $1 - 0.99^{\text{iteration \#}}$.

A.2. Identifiability of FINDR

We use the definition of latent identifiability in Wang et al. (2021) in the context of FINDR. FINDR is non-identifiable when $p_\theta(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{d}) = p_\theta(\boldsymbol{y}|\tilde{\boldsymbol{z}}, \boldsymbol{d})$ for all possible $\boldsymbol{z}$ and $\tilde{\boldsymbol{z}}$. For the latent variable to be identifiable, the sufficient condition is for $p_\theta(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{d})$ to be an injective function of $\boldsymbol{z}$ for some θ at the end of model training (Wang et al., 2021). In FINDR, when $\operatorname{rank}(\boldsymbol{C}) = L$, Equation (1) becomes injective. Therefore, we check at the end of model training whether the rank of the learned $\boldsymbol{C}$ matches L, and consider only models that satisfy $\operatorname{rank}(\boldsymbol{C}) = L$ at the end of training.

A.3. Post-Modeling Analysis

While we do not constrain $\boldsymbol{C}$ in Equation (26) to be semi-orthogonal (i.e., $\boldsymbol{C}^\top \boldsymbol{C} = \boldsymbol{I}$) during the training procedure, a semi-orthogonal $\boldsymbol{C}$ may be desired because distance and angle in the latent space $\mathbb{R}^L$ are distance and angle in the inverse-softplus rate space $\mathbb{R}^N$. More precisely, $||\boldsymbol{C}\boldsymbol{z}||^2 = \boldsymbol{z}^\top \boldsymbol{C}^\top \boldsymbol{C}\boldsymbol{z} = \boldsymbol{z}^\top \boldsymbol{z} = ||\boldsymbol{z}||^2$ for all $\boldsymbol{z}$. Having a semi-orthogonal, and therefore a distance-preserving, map $\boldsymbol{C}$ would make the latent trajectories inferred by FINDR more interpretable. We do not put either the soft or hard constraints on $\boldsymbol{C}$ to be semi-orthogonal because having such constraints is known to worsen performance in other contexts (Vorontsov et al., 2017). We also find this to be true when we add soft constraints to enforce orthogonality.

Before we interpret the latent trajectories $\boldsymbol{z}$ and the inferred flow field $\mu_\theta(\boldsymbol{z}, \boldsymbol{u})$, we perform singular value decomposition (SVD) on $\boldsymbol{C} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$, where $\boldsymbol{U} \in \mathbb{R}^{N \times L}$ is a semi-orthogonal matrix, $\boldsymbol{S} \in \mathbb{R}^{L \times L}$ is a diagonal matrix with its entries populated by the singular values and $\boldsymbol{V} \in \mathbb{R}^{L \times L}$ is an orthogonal matrix. Then, we apply a transformation $\tilde{\boldsymbol{z}} = \boldsymbol{S}\boldsymbol{V}^\top \boldsymbol{z}$. We next perform principal component analysis (PCA) on $\tilde{\boldsymbol{z}}$ so that the first component of the transformed $\bar{\boldsymbol{z}} = \boldsymbol{U}_{\text{pca}}\tilde{\boldsymbol{z}} + \boldsymbol{b}_{\text{pca}}$ corresponds to the first PC, and the L-th component of the transformed $\bar{\boldsymbol{z}}$ corresponds to the L-th PC. This transformation by PCA is rigid. Therefore, the distance and angle in the space and axes given by $\bar{\boldsymbol{z}}$ are still the distance and angle in the inverse-softplus rate space. The transformed flow field $\bar{\mu}_\theta$ in the space of $\bar{\boldsymbol{z}}$ is given by:

$$
\begin{aligned}
\boldsymbol{A} &= \boldsymbol{U}_{\text{pca}}\boldsymbol{S}\boldsymbol{V}^\top \\
\bar{\boldsymbol{z}} &= \boldsymbol{A}\boldsymbol{z} + \boldsymbol{b}_{\text{pca}} \\
\bar{\mu}_\theta(\bar{\boldsymbol{z}}, \boldsymbol{u}) &= \boldsymbol{A}\mu_\theta(\boldsymbol{A}^{-1}(\boldsymbol{z} - \boldsymbol{b}_{\text{pca}}), \boldsymbol{u}).
\end{aligned}
\tag{27}
$$

In all of our analyses that show the latent trajectories and flow fields inferred by FINDR, we plot the transformed $\bar{\boldsymbol{z}}$ and $\bar{\mu}_\theta(\bar{\boldsymbol{z}}, \boldsymbol{u})$. To project the flow field $\bar{\mu}_\theta(\bar{\boldsymbol{z}}, \boldsymbol{u})$ onto the first two PCs, we assume that the third and later components of $\bar{\boldsymbol{z}}$ are zero, and consider the first two components of $\dot{\bar{\boldsymbol{z}}} = \bar{\mu}_\theta(\bar{\boldsymbol{z}}, \boldsymbol{u})$.

A.4. Model Evaluation Metrics

In Section 3.2, the normalized log-likelihood difference score (NLL) is defined as:

$$
\text{NLL} = \mathbb{E}_t[\text{Poisson log-likelihood}(\boldsymbol{\lambda}_t, \boldsymbol{y}_t) - \text{Poisson log-likelihood}(\overline{\boldsymbol{\lambda}}_{1:T}, \boldsymbol{y}_t)], \tag{28}
$$

similar to the definition in Pei et al. (2021). Here, $\overline{\boldsymbol{\lambda}}_{1:T}$ is the mean firing rate estimated from $\boldsymbol{y}_{1:T}$. We use another model evaluation metric called evidence-conditioned peristimulus time histogram (PSTH) R^2, which is defined for each neuron as:

$$\begin{aligned} R^2 &= 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}, \\ SS_{\text{res}} &= \sum_t \left[\left(\text{PSTH}^R_{\text{obs},t} - \text{PSTH}^R_{\text{pred},t}\right)^2 + \left(\text{PSTH}^L_{\text{obs},t} - \text{PSTH}^L_{\text{pred},t}\right)^2 \right], \\ SS_{\text{tot}} &= \sum_t \left[\left(\text{PSTH}^R_{\text{obs},t} - \mathbb{E}_t\left[\text{PSTH}^R_{\text{obs},t}\right]\right)^2 + \left(\text{PSTH}^L_{\text{obs},t} - \mathbb{E}_t\left[\text{PSTH}^L_{\text{obs},t}\right]\right)^2 \right]. \end{aligned} \tag{29}$$

For a given neuron, we binned the spike train with a 10ms bin-width, and then convolved it with a causal Gaussian linear filter with a standard deviation of 0.1s and a width of 0.3s. Then, PSTHs were computed by averaging this smoothed spike train across trials. Here, PSTH^L_* represents the average firing rate computed from trials where there were more left clicks than right clicks, and PSTH^R_* represents the average firing rate computed from trials where there were more right clicks than left clicks. We computed the PSTHs from the observed and FINDR-predicted firing rates ($* \in \{\text{obs}, \text{pred}\}$) to compute the R^2.

A.5. Fitting FINDR to Synthetic Datasets with Autonomous Dynamics

We found that FINDR correctly captures autonomous limit cycle dynamics in a synthetic dataset (Figure 7). To produce spike trains from the limit cycle dynamics (Figure 7a), we we used similar settings as the datasets in Section 3.1 and Extended Data Figure 1 (but with 500 trials, 80 neurons, and a constant bias of 5 spikes/s).

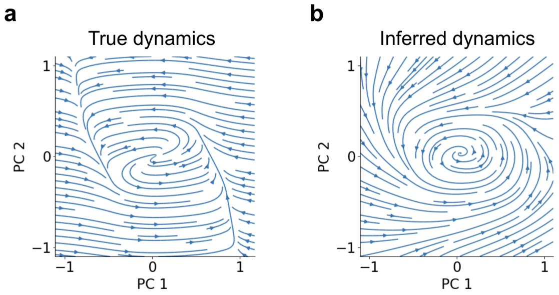


Figure 7. FINDR correctly captures autonomous limit cycle dynamics from simulated spike train data. **a**, Ground truth dynamics used to generate the spike trains. **b**, Dynamics inferred from FINDR. The inferred dynamics captures limit cycle.

B. Software and Data

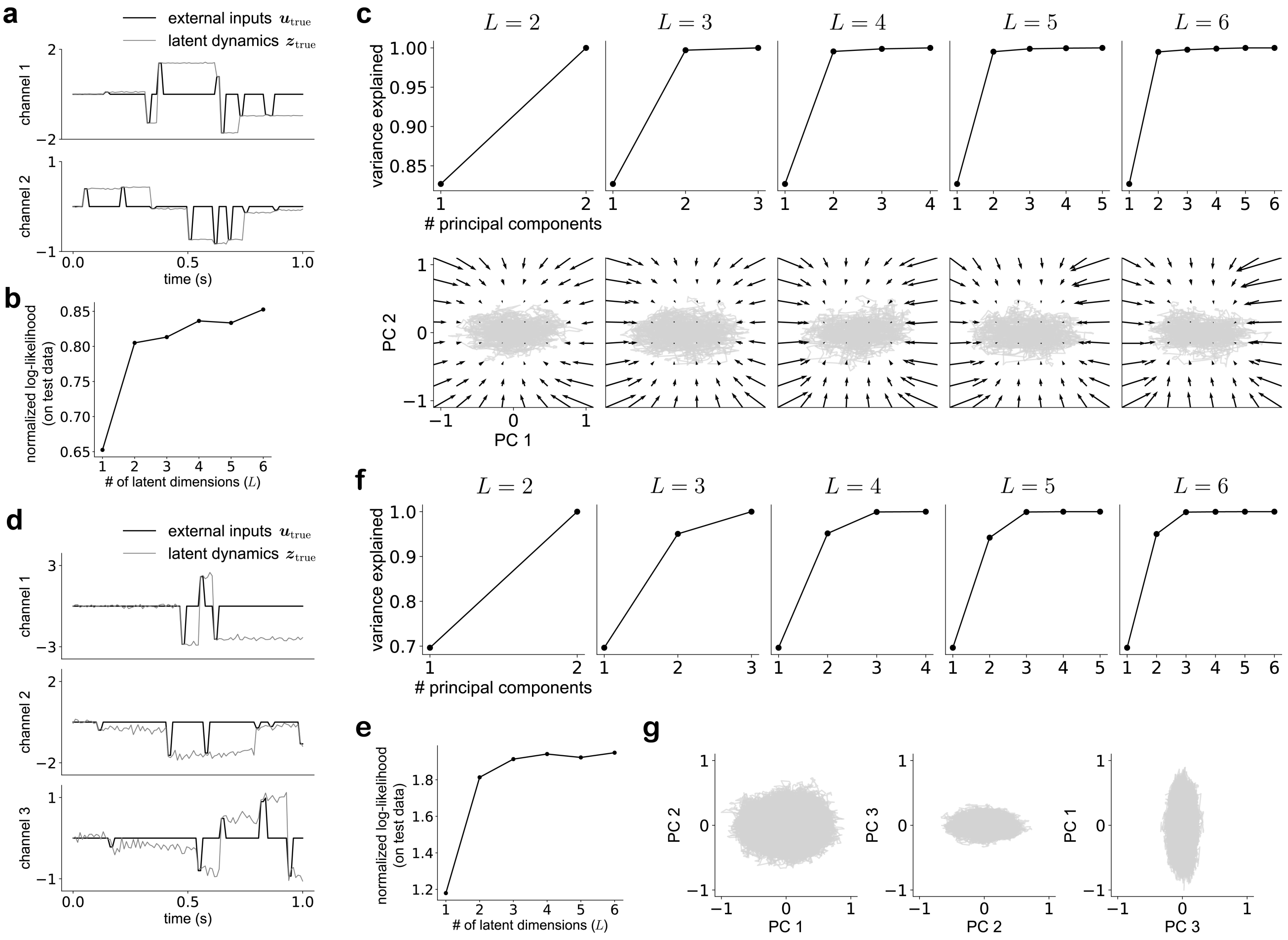
Our code is available as a GitHub repository: `https://github.com/Brody-Lab/findr`.

For SLDS and rSLDS, we used code from `https://github.com/lindermanlab/ssm`. For autoLFADS, we used code from `https://github.com/arsedler9/lfads-torch`, with hyperparameter search configurations in configs/pbt.yaml. For GPFA, we used Elephant: `https://github.com/NeuralEnsemble/elephant`. For CEBRA, we used code from `https://github.com/AdaptiveMotorControlLab/cebra`. We fit a Euclidean-distance CEBRA-Time model using hyperparameters from `https://cebra.ai/docs/demo_notebooks/CEBRA_best_practices.html#Items-to-consider`, but with changes to three hyperparameters (`model_architecture="offset10-model-mse"`, `max_iterations=1000`, `output_dimension=2`).

C. Author Contributions

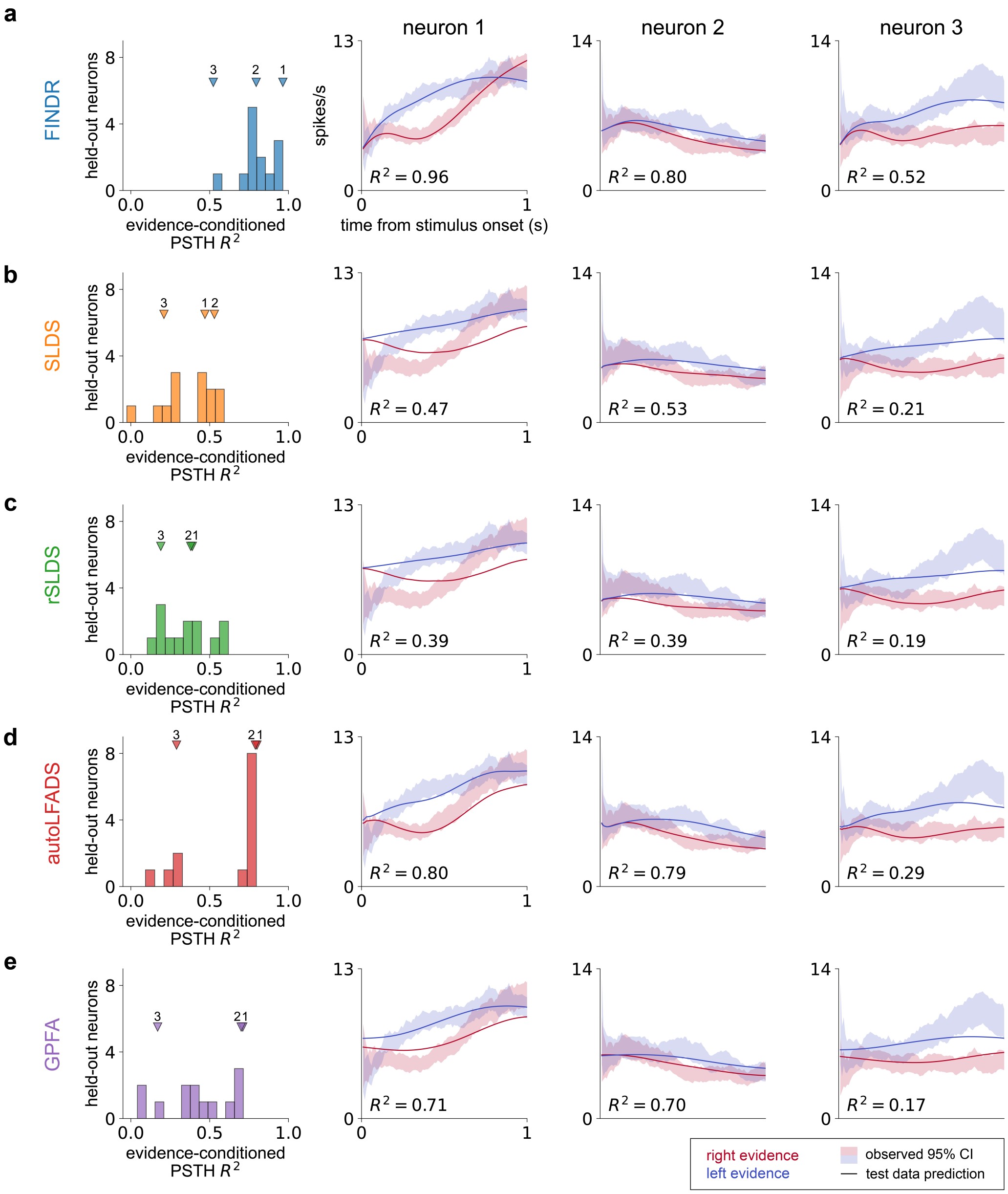
T.D.K. conceptualized the method. T.D.K. and T.Z.L. developed the inference method for task-irrelevant dynamics. T.D.K. developed the inference method for task-relevant dynamics. T.Z.L. collected data. T.D.K., T.C., and K.K. developed the gated multilayer perceptron (MLP) used in this method. T.D.K. implemented the method as a software package. T.D.K. wrote the manuscript after discussions among all authors. J.W.P. and C.D.B. supervised the project.

Extended Data Figure 1



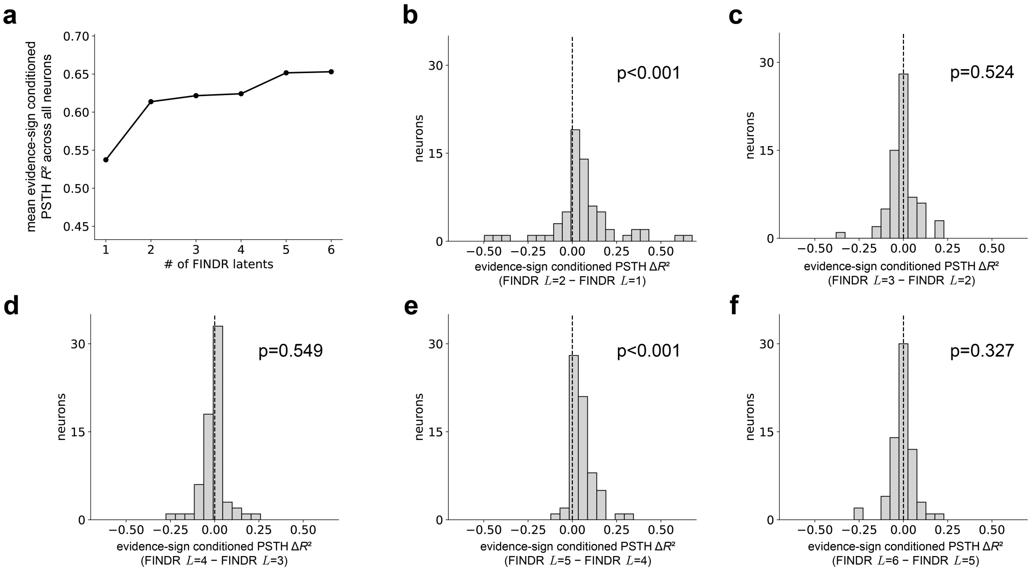
Extended analysis related to Figure 2. **a**, We generate transient pulse inputs from 2 independent channels and let a 2-dimensional system maintain the value of the most recent pulse in each channel. The pulse value in channel 1 (c_1) satisfies $-2 \leq c_1 \leq 2$ and the pulse value in channel 2 (c_2) satisfies $-1 \leq c_2 \leq 1$. **b–c**, Analysis similar to Figure 2c-d. **d**, We generate transient pulse inputs from 3 independent channels and let a 3-dimensional system maintain (with some noise) the value of the most recent pulse in each channel. The pulse value in channel 1 (c_1) satisfies $-3 \leq c_1 \leq 3$, the pulse value in channel 2 (c_2) satisfies $-2 \leq c_2 \leq 2$, and the pulse value in channel 3 (c_3) satisfies $-1 \leq c_2 \leq 1$. **e–f**, Analysis similar to Figure 2c-d. **g**, The range of values that the latent trajectory $\boldsymbol{z}$ takes along PC 1 is larger than the range of values that $\boldsymbol{z}$ takes along PC 2 and PC 3. The range of values that $\boldsymbol{z}$ takes along PC 2 is larger than the range of values that $\boldsymbol{z}$ takes along PC 3. This suggests that distance in this latent space is preserved and reflects the statistics of the pulses.

Extended Data Figure 2



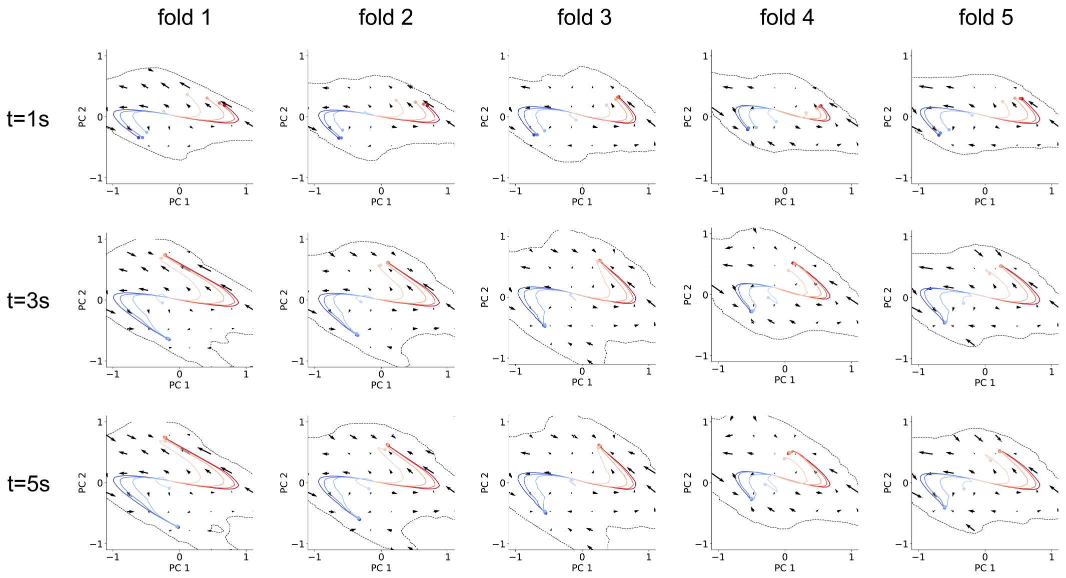
Extended analysis related to Figure 3c. **a**, Same as Figure 3c. **b**, Analysis similar to Figure 3c, but for SLDS ($L = 6$). **c**, Analysis similar to Figure 3c, but for rSLDS ($L = 6$). **d**, Analysis similar to Figure 3c, but for autoLFADS ($L = 6$). **e**, Analysis similar to Figure 3c, but for GPFA ($L = 6$).

Extended Data Figure 3



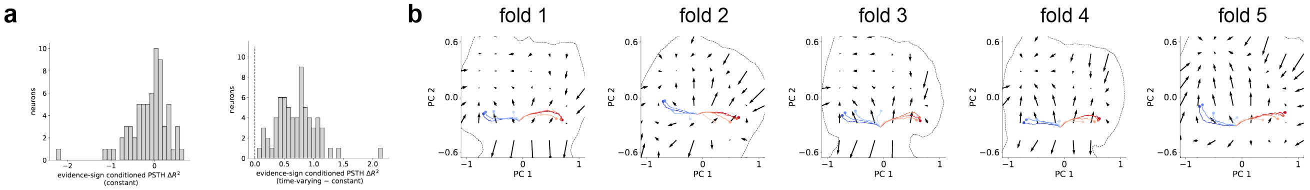
Extended analysis related to Figure 4. FINDR with $L = 2$ is sufficient to describe the data. To show this, we split the dataset into 5 different folds, where each fold contains a subset of trials in random order. We train FINDR on 3 of the folds, validate on 1 fold, and test its performance on the remaining 1 fold. **a**, We compute the 5-fold cross-validated evidence-sign conditioned PSTH R^2 for all neurons in the dataset and take the mean. We see an "elbow" at $L = 2$. **b**, We compute the evidence-sign conditioned PSTH R^2's for FINDR assuming $L = 2$ and $L = 1$. Then we take the difference between the R^2 obtained from FINDR with $L = 2$ and R^2 obtained from FINDR with $L = 1$ for each neuron. We find that FINDR with $L = 2$ performs significantly better than FINDR with $L = 1$ (Wilcoxon signed-rank test, $p < 0.001$). **c**, Analysis similar **b**, but for FINDR with $L = 3$ and $L = 2$. **d**, Analysis similar **b**, but for FINDR with $L = 4$ and $L = 3$. **e**, Analysis similar **b**, but for FINDR with $L = 5$ and $L = 4$. **f**, Analysis similar **b**, but for FINDR with $L = 6$ and $L = 5$.

Extended Data Figure 4



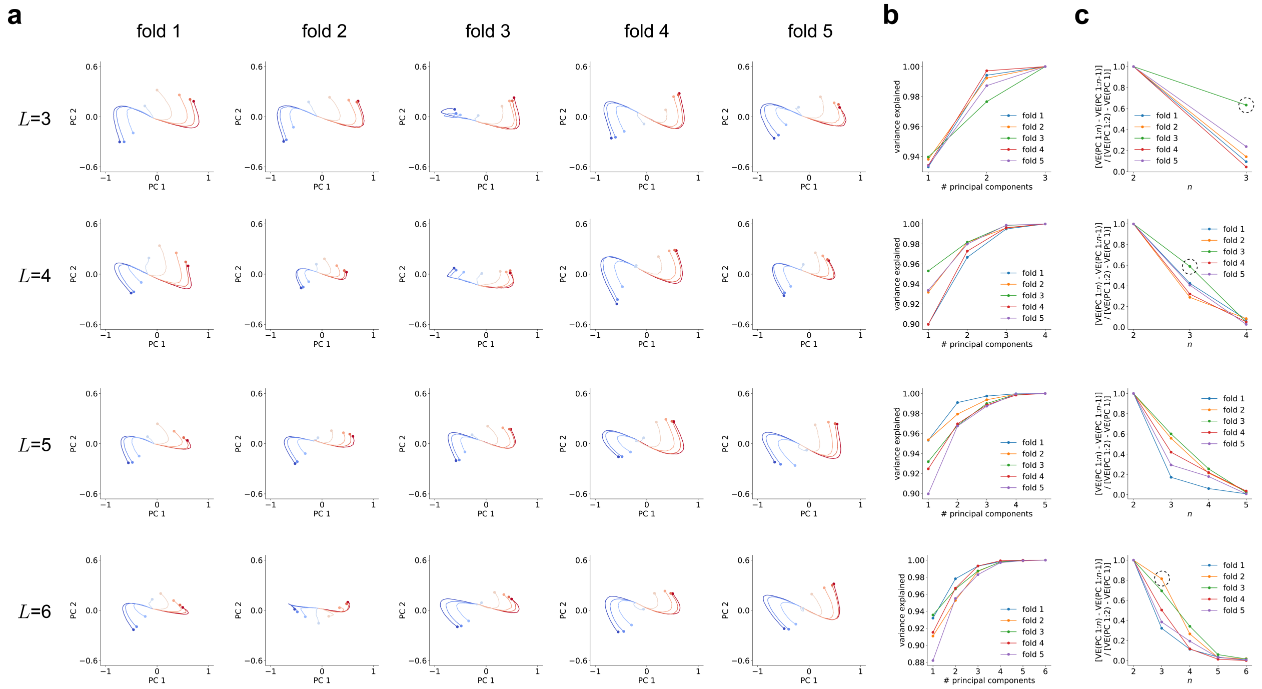
Extended analysis related to Figure 4. While the maximum duration of the auditory stimulus is 1 second in the data, and we fit FINDR only to the stimulus period, we can run the model for more than 1 second. When the model was run for $3-5$ seconds, the trial-averaged trajectories reached a steady state either at a point in the upper region of the state space or a point in the bottom region of the state space. What point the trajectories reached depended on the sign of the evidence (either left or right evidence), and suggested that these slow points might maintain the memory of the choice. Here, inside the dotted line represents part of the state space visited by single-trial trajectories, similar to Figure 4.

Extended Data Figure 5



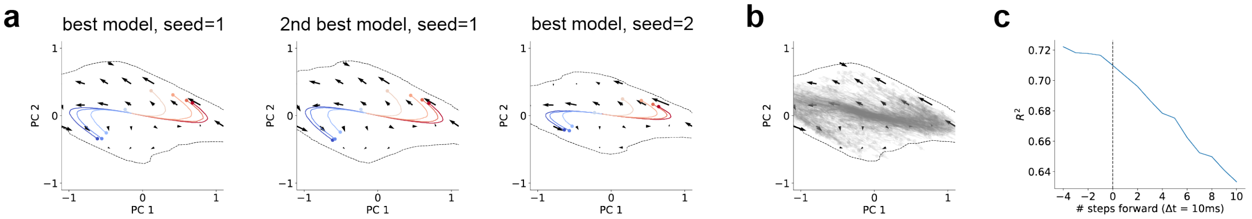
Extended analysis related to Figure 4. When the time-varying task-irrelevant within- and across-trial dynamics are not learned, and instead $\boldsymbol{d}_t$ was set to be a constant bias $\boldsymbol{d}$ for all t and all trials, **a**, we find that evidence-sign conditioned PSTH R^2's are significantly lower for this model compared to the FINDR model that learns the task-irrelevant dynamics. **b**, We also find that the latent dynamics learned by FINDR are less consistent across folds. Here, inside the dotted line represents part of the state space visited by single-trial trajectories, similar to Figure 4.

Extended Data Figure 6



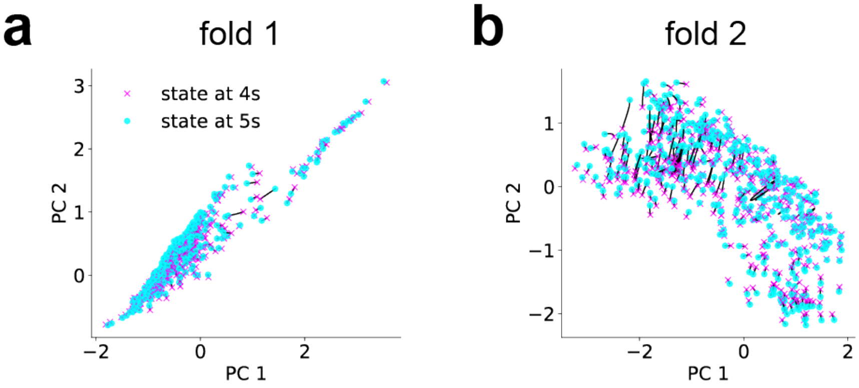
Extended analysis related to Figure 4. The latent trajectories discovered by FINDR with $L > 2$ are consistent across folds, and span mainly the first two PCs. **a**, As in Figure 4b, we plot the trial-averaged trajectories sorted by their evidence strength, with red indicating leftward evidence and blue indicating rightward evidence. The trial-averaged trajectories are projected onto the first two PCs. **b**, The first two PCs of the latent trajectories explain 95% of the variance. **c**, The latent trajectories in **a** are mostly consistent across folds and across dimensions, with an “S”-shape to the trajectories. For those conditions where there was no prominent S-shape (e.g., fold 3 of $L = 3$, fold 3 of $L = 4$, and fold 2 of $L = 6$), the variance explained by the 3rd component was relatively higher than the other folds (as indicated by black dotted circles in **c**), and we could find the “S”-shaped trajectories in the 3-dimensional PC space.

Extended Data Figure 7



Extended analysis related to Figure 4. **a**, We find that the second-best hyperparameter choice gives a representation consistent with the best choice. We also find a consistent representation when we train the model with a different initialization. **b**, The "confidence heatmap" showing the single-trial trajectories generated from the posterior. We expect that the dynamics inferred around regions traversed by more trajectories will be more accurate compared to regions traversed less. **c**, To directly assess how well FINDR extrapolates from training data, we trained FINDR to the same dataset used in Figure 4, but held out the last 0.1s of each trial (= 10 time steps). We then computed 5-fold cross-validated R^2 for each time step between $\boldsymbol{z}$ from the full vs. the held-out models.

Extended Data Figure 8



Extended analysis related to Figure 4. **a**, For fold 1, we ran the trained generator forward in time for 5s, starting from the initial conditions inferred from the encoder (448 initial conditions, because there were a total of 448 trials in the dataset in Section 3.3). We found that the autoLFADS states at 4s and states at 5s are close to each other, suggesting they may have reached approximate steady-states (black line indicates displacement from state at 4s to state at 5s). However, we found that the states did not form two clusters as would be expected from bistable attractors. The states are visualized with the first two PCs. **b**, We performed the same analysis for fold 2.