

Improved interpretability in LFADS models using a learned, context-dependent per-trial bias

Nishal P. Shah* Rice University
Benyamin Abramovich Krasa Stanford University
Erin Kunz Stanford University
Nick Hahn Stanford University
Foram Kamdar Stanford University
Donald Avansino Stanford University
Leigh R. Hochberg Massachusetts General Hospital
 Providence VA Medical Center
 Brown University
Jaimie M. Henderson Stanford University
David Sussillo Stanford University

Abstract

The computation-through-dynamics perspective argues that biological neural circuits process information via the continuous evolution of their internal states. Inspired by this perspective, Latent Factor Activity using Dynamical systems (LFADS, [1]) identifies a generative model consistent with the neural activity recordings. LFADS models neural dynamics with a recurrent neural network (RNN) generator, which results in excellent fit to the data. However, it has been difficult to understand the dynamics of the LFADS generator. In this work, we show that this poor interpretability arises in part because the generator implements complex, multi-stable dynamics.

We introduce a simple modification to LFADS that ameliorates issues with interpretability by providing an inferred per-trial bias (modeled as a constant input) to the RNN generator, enabling it to contextually adapt a simpler dynamical system to individual trials. In both simulated neural recordings from pendulum oscillations and real recordings during arm movements in nonhuman primates, we observed that the standard LFADS learned complex, multi-stable dynamics, whereas the modified LFADS learned easier-to-understand contextual dynamics. This enabled direct analysis of the generator, which reproduced at a single-trial level previous results shown only through more complex analyses at the trial average. Finally, we applied the per-trial inferred bias LFADS model to human intracortical brain computer interface recordings during attempted finger movements and speech. We show that modifying neural dynamics using linear operations of the per-trial bias addresses non-stationarity and identifies the extent of behavioral variability, problems known to plague BCI. We call our modification to LFADS as “contextual LFADS”.

1 Introduction

The dynamical systems perspective is increasingly being adopted to understand how neuronal circuits in the brain represent and compute information [2–4]. This perspective states that neural activity embeds the evolution of a latent state, which starts from an initial condition and evolves according

*Work done at Stanford University. Corresponding author. Email: ns157@rice.edu

to recurrent dynamics of the neural circuit as well as inputs from other brain regions. A complete characterization of neuronal activity under this framework entails estimating the dynamical system, as well as the initial conditions and input that drive it. While the latent states give an interpretable description of the high-dimensional neural activity, the components of the dynamical system provide mechanistic interpretation of the neural circuit.

When it comes to estimating such a system, linear dynamical systems provide a simple and interpretable model in terms of the separable modes that evolve independently. While easy to understand, the linear models may not be the best fit to neural data. Subsequently, variants of the linear model with discrete switching [5] or smooth interpolations [6, 7] improve model fit while still being interpretable. However, these approaches impose assumptions about how dynamics vary across task conditions, embedding those assumptions directly into the model architecture.

Alternatively, flexible models such as RNNs [1, 8, 9] or transformers [10–14] do not make such assumptions and achieve state-of-the-art accuracy in fitting to neural data. A prominent example of this is LFADS [1], an unsupervised sequential auto-encoder for neural time series. LFADS uses a recurrent neural network (RNN) generator to model the neural population dynamics, extracting a set of latent factors that summarize high-dimensional neural spiking on single trials. By making an explicit dynamical systems assumption, LFADS infers smooth latent trajectories that capture the neural computations within each trial, leading to improved single-trial representations for downstream decoding or analysis. In practice, LFADS models have shown state-of-the-art performance in denoising neural data and inferring latent firing rates, providing researchers with a window into the “neural manifold” underlying complex behaviors [1, 15, 16].

Interpreting the dynamics learned by an RNN typically requires a reverse engineering step in which one identifies fixed points and analyzes the linearized dynamics around them [17, 18]. While this approach can in principle provide mechanistic insights [19], it has yielded limited success in practice for LFADS thanks to LFADS generators often developing complex, *multi-stable dynamics*. This in turn makes fixed-point finding and linearization far less useful.

Here we propose a straight-forward modification that gives the LFADS generator access to an inferred low-dimensional per-trial bias, modeled as an additional constant input. This bias parameterizes a (low-dimensional) family of RNNs and thereby subtly adapts the RNN dynamics on a per-trial basis, resulting in *contextual dynamics* that are often simpler to interpret than the complex, often multi-stable dynamics of the original LFADS architecture. Further, the per-trial bias gives a low-dimensional embedding of the family of dynamical systems (parameterized by each bias), which can be used for additional analyses as well as for better BCI decoder estimation or improved neural data analysis. The main contributions of our work are:

1. We identify complex, often multi-stable dynamics as a fundamental barrier in using LFADS to understand dynamics in neural recordings.
2. We introduce a new modification of LFADS with a per-trial bias for learning contextual dynamics, which are often easier to interpret. The modification is straightforward and can be easily incorporated into existing implementations [20–22].
3. In recordings from monkey motor cortex during reaching, we show that our model recovers known structure in the neural dynamics and reveals novel, condition-dependent changes in oscillation frequency at the single-trial level [23].
4. We study the per-trial bias embedding space and show that simple geometrical operations of per-trial bias can mitigate the effects of non-stationarity and identify the extent of behavioral variability in human intracortical BCI recordings.

2 Related works

Task RNNs [24–26] are artificial recurrent networks trained to solve cognitive tasks in simulation, and are useful to build hypotheses about how the brain can solve these tasks. Task conditions (such as stimulus, task type or task period) are often provided to the RNNs using constant inputs, similar to the per-trial bias in our model. Similar to our analysis, Driscoll et al., 2024 [24] perform interpolation operation between the constant inputs (bias) for two different conditions and study how the dynamics change smoothly with the interpolation.

Several approaches have sought to improve the interpretability of RNNs fit to neural recordings by introducing structural constraints. One popular strategy constrains the recurrent dynamics to evolve within a low-dimensional subspace, enabling analytical insights into their behavior [27–31]. However, such low-rank constraints may limit biological realism. Other work has leveraged RNNs with piecewise-linear activations [32–37], which allow tractable analysis of fixed points and cycles. Neural ODE models, which learn continuous-time dynamics by predicting state derivatives rather than next states, have also been shown to enhance interpretability [38, 39]. In contrast to these structural modifications, our approach retains standard nonlinear RNN modules such as GRUs [40], and instead improves interpretability by constraining the per-trial contextual bias to vary along a low-dimensional manifold. This approach assumes that while the neural activity within each trial may be high-dimensional, the across-condition variability is captured compactly in a lower-dimensional space.

Finally, Jacobian Switching Linear Dynamical System (JSLDS, [9]) improves interpretability by co-training the LFADS RNN with a switching linear dynamical system. At each time point, this model identifies a single fixed point for linearizing the dynamics and this fixed point can change over the course of a trial. In contrast, our approach can learn more complex structure (multiple fixed points) that remains fixed for the course of the trial.

3 Methods

We begin by reviewing the standard LFADS model and then introduce our proposed modification incorporating a per-trial contextual bias (Figure 1).

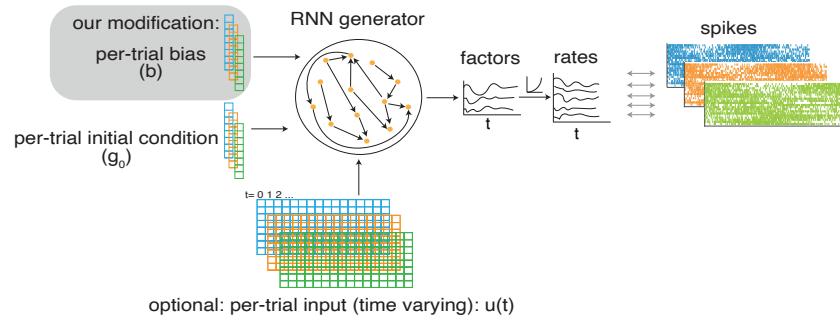


Figure 1: **The contextual LFADS model.** Recorded spikes are modeled as samples from a Poisson firing rate. Log-firing rate is linearly read out from the RNN state with an optional bottleneck of low-dimensional factors. In the standard LFADS, the RNN starts from an initial condition and optionally driven by a low-dimensional time-varying input (both inferred from data from encoders, not shown). Our modification of LFADS provides an additional inferred per-trial bias (gray box) which is modeled as a constant input to the RNN.

3.1 The Standard Latent Factor Analysis via Dynamical Systems (LFADS) Model

LFADS [1] is a sequential variational autoencoder designed to infer smooth latent dynamics from noisy, high-dimensional neural spike recordings. The model represents each trial’s neural activity as the output of a nonlinear dynamical system, simulated by an RNN. For trial number i , the RNN generates latent states $\mathbf{g}_{i,t}$ from an initial condition $\mathbf{g}_{i,0}$ and optional inferred inputs $\mathbf{u}_{i,t}$ (both are outputs of separate probabilistic encoders, not shown).

For trial i with observed spike counts $\mathbf{x}_{i,1:T}$, the generative process is:

$$\mathbf{g}_{i,0} \sim Q_{enc,g}(\mathbf{g}_0 | \mathbf{x}_{i,1:T}) \quad (\text{inferred initial condition}) \quad (1)$$

$$\mathbf{u}_{i,t} \sim Q_{enc,u}(\mathbf{u}_t | \mathbf{x}_{i,1:T}) \quad (\text{optional inferred input}) \quad (2)$$

$$\mathbf{g}_{i,t} = \text{RNN}_{\text{gen}}(\mathbf{g}_{i,t-1}, \mathbf{u}_{i,t}) \quad (3)$$

$$\mathbf{f}_{i,t} = W_{\text{fac}} \mathbf{g}_{i,t} + \mathbf{b}_{\text{fac}} \quad (4)$$

$$\mathbf{r}_{i,t} = \exp(W_{\text{rate}} \mathbf{f}_{i,t}) \quad (5)$$

$$\mathbf{x}_{i,t} \sim \text{Poisson}(\mathbf{r}_{i,t}) \quad (\text{observed spikes}) \quad (6)$$

where the optional time-varying factors $\mathbf{f}_{i,t}$ provide a low-dimensional representation of population activity, and $\mathbf{r}_{i,t}$ are the Poisson firing rates. The encoders ($Q_{enc,g}$, $Q_{enc,u}$) are bidirectional RNNs followed by linear mapping to a lower dimension. We use reparametrization trick to generate samples for initial condition and time-varying input. For example, the initial condition encoder predicts mean ($\mathbf{g}_{i,0,\mu}$) and standard-deviation ($\mathbf{g}_{i,0,\sigma}$) and samples are generated by $\mathbf{g}_{i,0} = \mathbf{g}_{i,0,\mu} + \epsilon \mathbf{g}_{i,0,\sigma}$ with ϵ distributed as standard Gaussian. See Pandarinath et al. 2018 [1] for further details.

3.2 Modification of LFADS with a Per-Trial Bias

To improve interpretability and reduce model complexity, we introduce a per-trial contextual bias $\mathbf{b}_i$. The per-trial bias is modeled as a constant input to the RNN, in addition to the optional time-varying inferred inputs $\mathbf{u}_{i,t}$. The RNN update equations are given by:

$$\mathbf{g}_{i,t+1} = \text{RNN}_{\text{gen}}(\mathbf{g}_{i,t}, [\mathbf{u}_{i,t}, \mathbf{b}_i]) \quad (7)$$

where the input and bias are concatenated ($[\mathbf{u}_{i,t}, \mathbf{b}_i]$) at each time-step. Note that the per-trial bias is inferred by an encoder from the observed spike train:

$$\mathbf{b}_i \sim Q_{enc,b}(\mathbf{b} | \mathbf{x}_{i,1:T}) \quad (\text{inferred per-trial bias}) \quad (8)$$

Similar to the encoder for initial condition, the bias-encoder ($Q_{enc,b}$) is a bidirectional RNN, and reparameterization trick is used to generate samples. The per-trial bias is typically lower dimensional (dimensionality ≤ 10) than the RNN hidden state dimensions. The training objective is presented in Appendix section 6.1, details on model fitting is presented in Appendix section 6.2 and hyperparameters are listed in Appendix section 6.4.

The per-trial bias allows the generator to implement trial-specific dynamics via the (potentially smooth) modulation of the dynamical landscape. Although the mean of the time-varying input could, in principle, capture similar information as the per-trial bias, the full complexity of a time-varying inferred input is often unnecessary for many analyses. Moreover, commonly used priors on the input—such as Gaussian or smooth autoregressive priors—can obscure the mean, making it more difficult to learn explicitly. Empirically, the result is often more interpretable using an inferred bias, where each $\mathbf{b}_i$ moves a single fixed point rather than relying on multi-stable autonomous dynamics or time-varying dynamics resulting from time-varying inputs.

4 Results

4.1 Interpretable Dynamics in a Simulated Pendulum

To evaluate the interpretability of the proposed method, we simulated a system with known ground truth dynamics: a family of damped pendulums (Figure 2A). This example is constructed to make the point that while a single pendulum should result in simple dynamics around a single fixed point, a family of pendulums will result in multi-stable dynamics. When modeling neural data with LFADS we will not know which scenario we are in.

The pendulum’s state was represented as $s = (\theta, \dot{\theta})$, where θ denotes the angle. Starting from an initial state $s_0 = (\theta_0, 0)$, the state evolved according to the nonlinear differential equation given by $ml^2\ddot{\theta} + d\dot{\theta} + mgl \sin \theta = 0$, with $m = 1$, $l = 5$ and $d = 1$. The activity of 192 simulated neurons (n) embedded the state of the pendulum using a mapping $n \sim \text{Poisson}(\max(Ks + \mu, 0))$. We simulated a family of pendulums under 8 distinct gravity conditions by changing the values of g from 0.5 to 4.5 in steps of 0.5. For each condition, we generated 125 trials of 300 sample duration, with 75 used for training, 25 for validation and 25 for testing.

We trained both the standard LFADS model and the contextual LFADS model incorporating per-trial biases on the simulated dataset. As the ground-truth dynamics are autonomous, we did not have the optional time-varying input for both models. We observed that the both models fit the data well and showed oscillations in the hidden state of the generator (Figure 2B,C).

To understand the learned dynamics, we “reversed engineered” the learned RNN to identify approximate fixed points of the state update function [17, 18]. Specifically, the fixed points were

identified as neural states where $\mathbf{g}^* \approx \text{RNN}_{\text{gen}}(\mathbf{g}^*, \mathbf{u}_{i,t} = 0)$ for the standard LFADS and $\mathbf{g}^* \approx \text{RNN}_{\text{gen}}(\mathbf{g}^*, [\mathbf{u}_{i,t} = 0, \mathbf{b}_i])$ for the LFADS model with per-trial bias. Details of fixed point finding are presented in Appendix (Section 6.3).

We observed that the standard LFADS model implemented a cluster of fixed points which were shared across all conditions (Figure 2B, gray), with each fixed point associated with dynamics of a distinct oscillatory frequency. Consequently, the initial condition of the RNN not only determined the initial state of the pendulum, but also had to implicitly select the correct pendulum from the family of pendulums. To accomplish this, the RNN implemented a **multi-stable dynamical** landscape.

In contrast, we observed that the LFADS model with per-trial biases implemented a single fixed point per trial. Within each condition, both the inferred bias and the corresponding fixed-point location remained consistent across trials (Figure 2C). As the gravity level varied, the fixed point location and bias moved smoothly along a one-dimensional manifold, effectively recovering the underlying scalar parameter. This structure is described as **contextual dynamics**, in which the RNN dynamics are explicitly modulated by inferred contextual inputs, leading to an interpretable representation of condition-specific variability. In the next section, we extend these observations to real data.

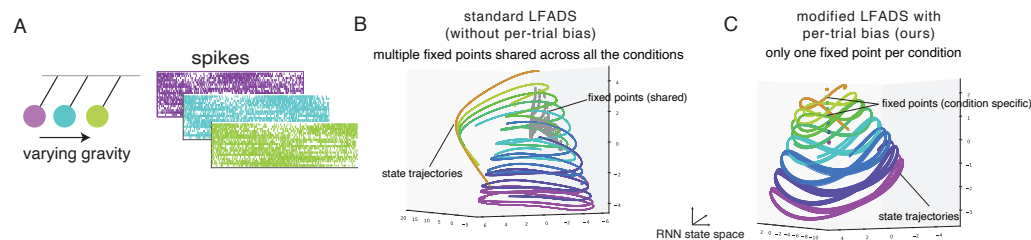


Figure 2: **Simulated pendulum.** (A) Task conditions correspond to distinct values of gravity, which changes the oscillation frequency. Neural activity embeds oscillations of the pendulum. Multiple trials generated for each gravity condition. (B) Standard LFADS. Colored lines indicate RNN hidden state trajectories (projected to top three principal components), colors indicate oscillation frequency/gravity. Gray dots indicate fixed points, which are shared across all conditions. (C) Same as (B) for contextual LFADS with inferred per-trial bias. The dots indicate fixed points, and for a given task condition only the fixed points with the corresponding color are present.

4.2 Neural dynamics during reaching task in monkeys

We further evaluated contextual LFADS on the “Maze dataset” [23], in which a monkey performed straight and curved reaches while sitting in front of a computer screen (Figure 3A,B). The dataset consists of 107 reach conditions, with 14 – 39 trials per condition (average 18.3 trials per condition). Each trial spans 180 ms before to 450 ms seconds after the go cue.

As with the pendulum simulation, we observed that both standard and contextual LFADS models identified similar latent dynamics and achieved comparable kinematic reconstruction accuracy (linear regression, Standard LFADS: 0.842, SD: 0.008 vs Contextual LFADS: 0.844, SD: 0.012). The standard LFADS model exhibited multi-stable dynamics, characterized by a shared cluster of fixed points across conditions (Figure 3C). In contrast, the modified LFADS model with per-trial bias exhibited contextual dynamics, with one fixed point per condition, leading to a more interpretable organization of trial-to-trial variability (Figure 3D).

A key scientific question that can be addressed with this dataset is which aspects of neural dynamics are invariant across reach conditions and which are condition-dependent. Churchland and Shenoy, 2012 [23] argued that starting from a reach-specific preparatory activity (corresponding to initial condition), neural activity follows shared condition-invariant dynamics. Recently, Sabatini and Kauffman, 2024 [41] showed that all properties of the dynamics except rotation frequency are condition-specific. We revisit this question using the contextual LFADS model.

We observe that the inferred per-trial biases captured the circular geometry of reach directions, with biases corresponding to similar directions located nearby (Figure 4A). As the bias varied smoothly around the circle, the associated fixed point locations varied smoothly in the RNN state space (Figure 4B).

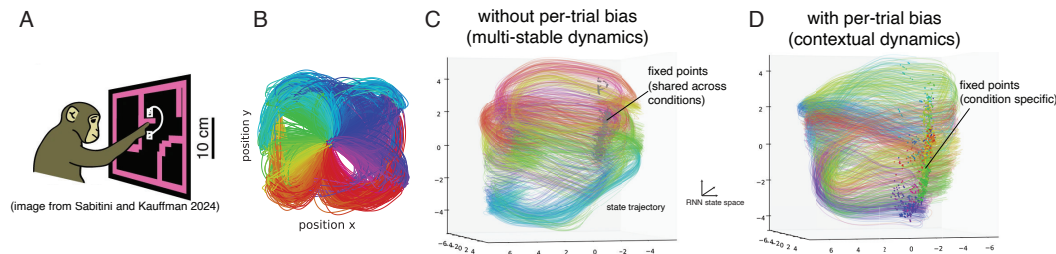


Figure 3: Maze task. (A) Task setup showing monkey reaches around obstacles to reach targets, resulting in straight or curved reaches. Figure from [41]. (B) 2D arm trajectories. Colors indicate initial reach direction. (C) RNN hidden state visualization (along top 3 PCs) for the standard LFADS model. Lines indicate state trajectory, colors indicate initial reach direction (same as (B)). Gray dots indicate the shared fixed points across all conditions. (D) Same as (C), for contextual LFADS with inferred per-trial bias. Colored dots indicate fixed point locations. For a given trial, only a single fixed point was active (shown in corresponding color).

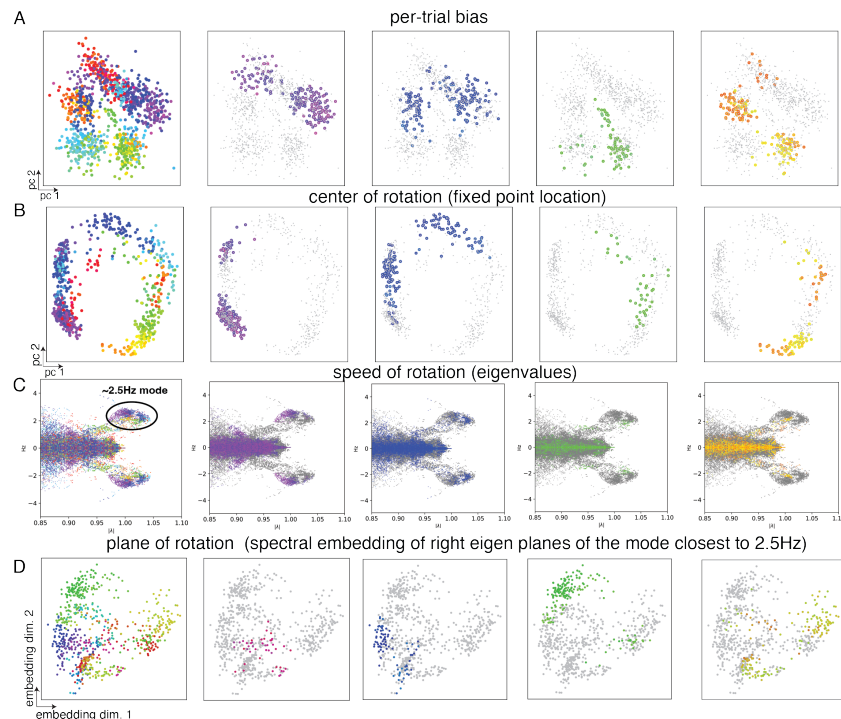


Figure 4: Contextual dynamics in the Maze task. Reach condition–dependent variation in different components of the dynamics system. Each dot represents a single trial. Color indicates the initial reach direction across all trials. In the first column, reach direction of all trials is indicated. In the subsequent columns, only trials within with a narrow range of reach angles is indicated (by color), while other trials are in gray. (A) Per-trial bias, projected onto its top two principal components. (B) Fixed-point location, projected onto its top two principal components. (C) Oscillation frequency, visualized via the real and imaginary parts of the eigenvalues across trials. (D) Two-dimensional spectral embedding of planes of rotation for the dominant oscillatory mode near 2.5 Hz. Similarity between trials was quantified based on how well the RNN state trajectory of one trial could be reconstructed using the eigen-planes from another. Similarity of center of rotation (B) and plane of rotation (D) for similar reach directions is consistent with [41]. The condition-specific changes in eigenvalues (C) has not been reported before.

Next, we studied the linearized dynamics around the fixed points, approximating $RNN_{\text{gen}}(\mathbf{x}, [\mathbf{u}_{i,t} = 0, \mathbf{b}_i]) \approx A(\mathbf{b}_i)(\mathbf{x} - \mathbf{x}_{\text{fixed point}}(\mathbf{b}_i)) + \mathbf{x}_{\text{fixed point}}(\mathbf{b}_i)$. Across different per-trial biases, we compared eigenvectors and eigenvalues of A . The real and imaginary components of the eigenvalue indicate the decay and frequency of rotations, and the eigen-planes (span of a pair of conjugate eigenvectors) indicate the planes in which these rotations lie.

Across different trials, there was a consistent 2.5 Hz oscillation mode in the eigenvalues (see the cluster in Figure 4C). The plane spanned by eigenvectors with eigenvalue closest to the 2.5 Hz frequency varied continuously with reach direction (Figure 4D). A modest but consistent modulation of the eigenvalue (oscillation frequency) near the 2.5 Hz mode was also detected across conditions (Figure 4C). While condition-dependent changes in fixed-point locations and eigen-planes have been previously reported [41], the eigenvalue modulations were not captured by earlier analyses relying on trial averaged data. Our single-trial analysis reveals a more nuanced view: all dynamical properties exhibit condition-specific variation, though the degree of variability differs across them (Table 1). We emphasize that contextual LFADS aided us in arriving at these results through no specific knowledge or hypotheses concerning these data (beyond trial structure).

Dynamical system property	Churchland & Shenoy, 2012 [23]	Sabatini & Kauffman, 2024 [41]	contextual LFADS (ours)
Preparation activity (initial condition)	Condition specific	Condition specific	Condition specific
Center of rotation (fixed-point location)	Condition invariant	Condition specific	Condition specific
Plane of rotation (eigen-planes)	Condition invariant	Condition specific	Condition specific
Speed of rotation (eigenvalues)	Condition invariant	Condition invariant	Condition specific (mild variation of frequency)

Table 1: Dynamical systems understanding of the neural activity in Maze reaching dataset. Note that contextual LFADS uses single-trial analysis whereas the other two use trial-averaged analysis.

4.3 Data augmentation using contextual LFADS makes iBCI decoders robust to non-stationarities

A major challenge in intracortical brain-computer interface (BCI) studies is the issue of non-stationarity, where neural recordings corresponding to the same intended movements drift over time. To investigate whether contextual LFADS could mitigate the effects of non-stationarity, we analyzed data from a participant ‘T5’ (enrolled in the BrainGate2 clinical trial) with two 96-channel Utah arrays surgically placed in the hand-knob region of the precentral gyrus. During the experiment, the participant attempted combinatorial movements of two finger groups: the thumb (one degree of freedom) and all other fingers (second degree of freedom). Each group was independently cued to flex, extend, or remain at rest. Trials alternated between transitions from rest to a cued gesture and back to rest. A total of 966 trials were collected, 52-54 trials for each of the 16 movements, and 107 trials without movement.

We focus on the goal of learning a linear finger position (continuous valued) decoder for the final quarter of the trials collected during a recording session. First, we evaluated the performance of a decoder trained on Gaussian-smoothed neural activity. Using the trials only from the final quarter yielded significantly lower accuracy compared to using all trials (R^2 of 0.24 vs. 0.46, Table 2), likely due to reduction in training data despite of an improved match in data distribution.

Training and testing the decoder on LFADS-inferred firing rates improved prediction accuracy, as expected from prior work [1]. Moreover, the discrepancy between training exclusively on the final quarter versus all trials was reduced (0.67 vs. 0.66, Table 2), suggesting that improved smoothing yields decoders that are robust to non-stationarities.

Finally, we used contextual LFADS model to transform data collected early in the session to later in the session. Analysis of the top two principal components of the inferred biases revealed a drift over the course of the session (Figure 5C). To compensate for this drift, per-trial biases and initial condition from early-session trials were systematically shifted to match the corresponding parameters for later-session trials.

Specifically, we first applied the bias-encoder to estimate the bias for trials in training partition. Next, we estimated the average bias of training trials collected early in the session $\mathbf{b}_{\text{early}}$ and estimated the average bias of training trials collected late in the session $\mathbf{b}_{\text{late}}$. Finally, for each trial collected earlier in the session, we corrected their bias with $\mathbf{b}_{i,\text{corrected}} = \mathbf{b}_i - \mathbf{b}_{\text{early}} + \mathbf{b}_{\text{late}}$. We applied a similar transformation for correcting initial conditions. No changes were applied to the time-varying inputs. Further details are provided in Appendix (Section 6.5).

The shifted biases and initial conditions were then passed through the RNN to generate drift corrected neural activity, augmenting the decoder training corpus. For a given trial, as the bias moved along the non-stationarity axis, the RNN activity and the fixed points moved along the non-stationarity axis in the RNN state space (Figure 5C, D). The augmented firing rates were then combined with non-augmented firing rates for trials later in the session to estimate the decoder.

Qualitative comparison showed that the transformed neural activity closely resembled that of a late-session trial with identical cued movement (Figure 5 E). As a result, combining LFADS-inferred rates from these transformed trials with LFADS-inferred rates from trials collected later in the session resulted in the highest decoding accuracy for the later trials ($R^2=0.70$, Table 2).

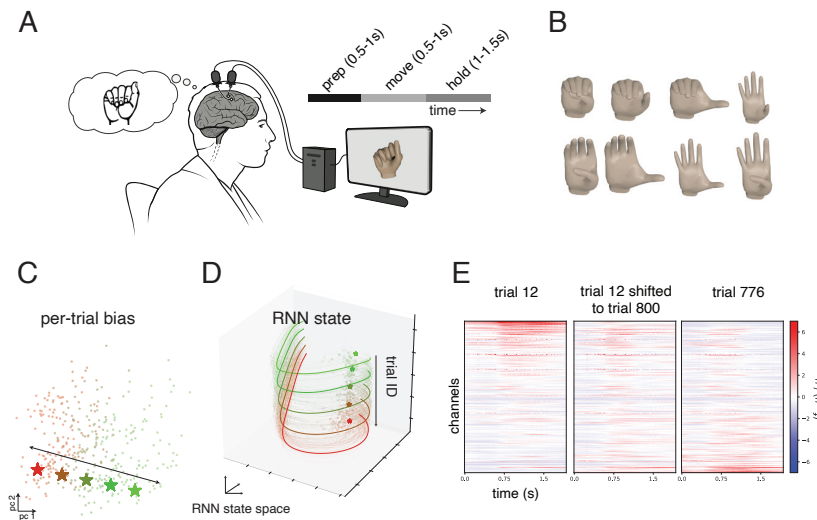


Figure 5: Using contextual LFADS for data augmentation improves decoder robustness to non-stationarities. (A) Intracortical BCI participant attempted finger movements as neural activity was recorded using Utah arrays. (B) The corpus of finger movement targets, with two finger groups (thumb and all other fingers constrained together) cued independently to be flexed/extended/stay at rest. (C) Two dimensional PCA of per-trial bias, with trials (dots) colored by the trial number. The direction of most correlated with trial number is indicated. (D) Three dimensional PCA of RNN state. Colors indicate trial number. Moving the bias (stars in panel C) moves the RNN state and fixed point (stars in panel D and solid lines). (E) Visualizations of smoothed neural activity (relative to mean for each channel) for trial 12 (left), trial 12 shifted to trial 800 (middle) and trial 776 (trial closest to 800, with same movement as trial 12, right).

Method	Finger position decoding (R^2)
Gaussian smoothed, later trials	0.24 +/- 0.06
Gaussian smoothed, all trials	0.46 +/- 0.02
LFADS smoothed, all trials	0.67 +/- 0.04
LFADS smoothed, later trials	0.66 +/- 0.05
LFADS smoothed, augmented trials	0.70 +/- 0.04

Table 2: Using contextual LFADS for data augmentation improves decoder accuracy in the presence of non-stationarities. Performance of “LFADS smoothed, augmented trials” is significantly better than “LFADS smoothed, later trials” (paired t-test, $p < 0.01$).

4.4 Per-trial bias helps identify variability in speed of putative behavior

Another potential source of variability in neural recordings with iBCI participants is the variability in the execution speed of a given movement across trials. We tested if the contextual LFADS model (without inferred inputs) could separate trials according to movement speed. Neural activity (multi-unit threshold crossings) was recorded from two 64-channel Utah arrays implanted in the ventral precentral gyrus of an BrainGate2 iBCI participant ('T12') as they attempted to speak 39 phonemes, (Figure 6A) with 20 trials each (data taken from [42]).

Projecting the neural activity along the second principal component of the condition invariant signal (i.e. neural activity over channels and time, averaged across all trials) and sorting them according to the order in which they were collected did not show any systematic structure in variation across trials (Figure 6C). Assuming that changes in the behavioral speed changes the speed of traversal of neural activity, we identified a direction in the per-trial bias space that directly affected the speed of neural trajectory traversal. For a given speed-up factor α (with $\alpha < 1$ for slowdown and $\alpha > 1$ for speedup), we identified a shift in per-trial bias space ($\mathbf{b}_{\text{speed},\alpha}$) that accelerates the neural state updates by solving the following optimization problem: $\mathbf{b}_{\text{speed},\alpha} = \arg \min_{\Delta \mathbf{b}} \sum_{i,t} [(\text{RNN}_{\text{gen}}(\mathbf{g}_{i,t}, [\mathbf{u}_{i,t} = 0, \mathbf{b}_i + \Delta \mathbf{b}]) - \mathbf{g}_{i,t}) - \alpha(\text{RNN}_{\text{gen}}(\mathbf{g}_{i,t}, [\mathbf{u}_{i,t} = 0, \mathbf{b}_i]) - \mathbf{g}_{i,t})]^2$. Intuitively, we seek a shift in per-trial bias ($\Delta \mathbf{b}$) that scales the one-step RNN updates by a factor α .

We identified a common direction for speed ($\vec{\mathbf{b}}_{\text{speed}}$) as the left singular vector of the matrix with columns corresponding to $\mathbf{b}_{\text{speed},\alpha}$ for 7 regularly spaced α between 0.7 and 1.3 (except $\alpha = 1$). Sorting the trials according to their projection along this direction (i.e. sorting by $\mathbf{b}_i^T \vec{\mathbf{b}}_{\text{speed}}$) ordered them by their putative speeds in behavior, as seen from the visualization along the first principal component of the neural activity (Figure 6B,D). Hence, the contextual LFADS model helps identify the extent of behavioral variability in the neural recordings.

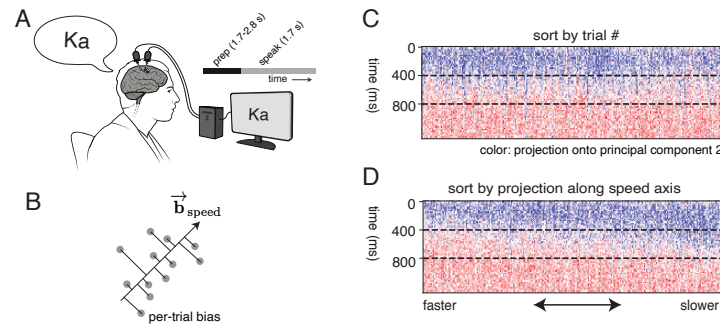


Figure 6: Using contextual LFADS to sort trials according to the speed of neural activity. (A) Intracortical BCI participant attempted to speak phonemes as neural activity was recorded using Utah arrays. (B) Visualization of projection of per-trial bias along a speed axis. (C, D) Visualization of first principal component of neural activity across trials, sorted by trial number (C) and projection along speed axis (D).

5 Discussion

We propose a modification of LFADS with an additional inferred per-trial bias provided to the RNN generator. This modification led to interpretable, contextual dynamics in the learned RNN in synthetic (Figure 2) and real monkey maze data which led to new scientific insights (Figures 3, 4). As low-dimensional bias summarizes neural dynamics, simple geometrical operations in the bias space enabled us counter non-stationarity (Figure 5) and identify the extent of behavioral variability (Figure 6) in human intracortical BCI recordings. While there may be other specialized or bespoke methods that may perform better for each of these problems, it is important to note that the method proposed in this paper is unsupervised and each of these problems is addressed in post-hoc analyses.

A key empirical observation was that contextual LFADS typically yielded a single fixed point per trial, facilitating interpretation of the inferred dynamics. However, this outcome is not guaranteed by the model. In principle, the network could learn more complex dynamics even in the presence of a per-trial bias. What we expect—and what we observed—is that introducing a contextual bias reduces the burden on the initial condition and time-varying input to encode condition-specific structure.

Instead, the bias allows the model to parameterize a family of generators, often simplifying the resulting dynamics.

One limitation of our approach is that it assumes segments of neural activity with relatively stationary dynamics. In more complex tasks, modeling the per-trial bias as a piecewise-constant input could allow the generator to adapt to dynamics across task phases. Another challenge lies in hyperparameter selection—particularly in choosing the dimensionality of the bias—since decoding performance is often unaffected by the bias and interpretability is difficult to quantify.

Acknowledgements

We would like to thank T5, T12 and their care partners. We would like to thank Beverly Davis, Kathy Tsou, and Sandrin Kosasih for administrative support. Support provided by Office of Research and Development, Rehabilitation R&D Service, Department of Veterans Affairs (N9228C, N2864C, A2295R, B6453R, P1155R, LRH); NIH-NIDCD (LRH), R01DC009899 (LRH), NIH-NINDS UH2NS095548 (LRH), NIH-NIDCD U01DC17844 (LRH), American Heart Association (LRH), NIH-NIDCD R01DC014034 (JMH), NIH-NINDS U01NS098968 (JMH), Wu Tsai Neurosciences Institute at Stanford (JMH), Larry and Pamela Garlick (JMH), Milton Safenowitz ALS Fellowship (NPS).

Disclosures

The MGH Translational Research Center has a clinical research support agreement (CRSA) with Ability Neuro, Axoft, Neuralink, Neurobionics, Paradromics, Precision Neuro, Synchron, and Reach Neuro, for which LRH provides consultative input. LRH is a non-compensated member of the Board of Directors of a nonprofit assistive communication device technology foundation (Speak Your Mind Foundation). Mass General Brigham (MGB) is convening the Implantable Brain-Computer Interface Collaborative Community (iBCI-CC); charitable gift agreements to MGB, including those received to date from Paradromics, Synchron, Precision Neuro, Neuralink, and Blackrock Neurotech, support the iBCI-CC, for which LRH provides effort.

JMH is a consultant for Neuralink and Paradromics, is a shareholder in Maplight Therapeutics and Enspire DBS, and is a co-founder and shareholder in Re-EmergeDBS. He is also an inventor on intellectual property licensed by Stanford University to Blackrock Neurotech and Neuralink.

The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health, or the Department of Veterans Affairs, or the United States Government.

CAUTION: Investigational Device. Limited by Federal Law to Investigational Use

References

- [1] Chethan Pandarinath, Daniel J O’Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D Stavisky, Jonathan C Kao, Eric M Trautmann, Matthew T Kaufman, Stephen I Ryu, Leigh R Hochberg, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10):805–815, 2018.
- [2] Saurabh Vyas, Matthew D Golub, David Sussillo, and Krishna V Shenoy. Computation through neural population dynamics. *Annual review of neuroscience*, 43(1):249–275, 2020.
- [3] Krishna V Shenoy and Jonathan C Kao. Measurement, manipulation and modeling of brain-wide neural population dynamics. *Nature communications*, 12(1):633, 2021.
- [4] Mikail Khona and Ila R Fiete. Attractor and integrator networks in the brain. *Nature Reviews Neuroscience*, 23(12):744–766, 2022.
- [5] Scott W Linderman, Andrew C Miller, Ryan P Adams, David M Blei, Liam Paninski, and Matthew J Johnson. Recurrent switching linear dynamical systems. *arXiv preprint arXiv:1610.08466*, 2016.

- [6] Noga Mudrik, Yenho Chen, Eva Yezerets, Christopher J Rozell, and Adam S Charles. Decomposed linear dynamical systems (dlds) for learning the latent components of neural dynamics. *Journal of Machine Learning Research*, 25(59):1–44, 2024.
- [7] Yenho Chen, Noga Mudrik, Kyle A Johnsen, Sankaraleengam Alagapan, Adam S Charles, and Christopher Rozell. Probabilistic decomposed linear dynamical systems for robust discovery of latent neural dynamics. *Advances in Neural Information Processing Systems*, 37:104443–104470, 2024.
- [8] Ding Zhou and Xue-Xin Wei. Learning identifiable and interpretable latent models of high-dimensional neural activity using pi-vae. *Advances in neural information processing systems*, 33:7234–7247, 2020.
- [9] Jimmy Smith, Scott Linderman, and David Sussillo. Reverse engineering recurrent neural networks with jacobian switching linear dynamical systems. *Advances in Neural Information Processing Systems*, 34:16700–16713, 2021.
- [10] Joel Ye and Chethan Pandarinath. Representation learning for neural population activity with neural data transformers. *arXiv preprint arXiv:2108.01210*, 2021.
- [11] Mehdi Azabou, Vinam Arora, Venkataramana Ganesh, Ximeng Mao, Santosh Nachimuthu, Michael Mendelson, Blake Richards, Matthew Perich, Guillaume Lajoie, and Eva Dyer. A unified, scalable framework for neural population decoding. *Advances in Neural Information Processing Systems*, 36:44937–44956, 2023.
- [12] Yizi Zhang, Yanchen Wang, Donato Jiménez-Benetó, Zixuan Wang, Mehdi Azabou, Blake Richards, Renee Tung, Olivier Winter, Eva Dyer, Liam Paninski, et al. Towards a "universal translator" for neural dynamics at single-cell, single-spike resolution. *Advances in Neural Information Processing Systems*, 37:80495–80521, 2024.
- [13] Josue Ortega Caro, Antonio H de O Fonseca, Christopher Averill, Syed A Rizvi, Matteo Rosati, James L Cross, Prateek Mittal, Emanuele Zappala, Daniel Levine, Rahul M Dhodapkar, et al. Brainlm: A foundation model for brain activity recordings. *bioRxiv*, pages 2023–09, 2023.
- [14] Trung Le and Eli Shlizerman. Stndt: Modeling neural population activity with spatiotemporal transformers. *Advances in Neural Information Processing Systems*, 35:17926–17939, 2022.
- [15] Mohammad Reza Keshtkaran, Andrew R Sedler, Raed H Chowdhury, Raghav Tandon, Diya Basrai, Sarah L Nguyen, Hansem Sohn, Mehrdad Jazayeri, Lee E Miller, and Chethan Pandarinath. A large-scale neural network training framework for generalized estimation of single-trial population dynamics. *Nature Methods*, 19(12):1572–1577, 2022.
- [16] Feng Zhu, Harrison A Grier, Raghav Tandon, Changjia Cai, Anjali Agarwal, Andrea Giovannucci, Matthew T Kaufman, and Chethan Pandarinath. A deep learning framework for inference of single-trial neural population dynamics from calcium imaging with subframe temporal resolution. *Nature neuroscience*, 25(12):1724–1734, 2022.
- [17] David Sussillo and Omri Barak. Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural computation*, 25(3):626–649, 2013.
- [18] Matthew D Golub and David Sussillo. Fixedpointfinder: A tensorflow toolbox for identifying and characterizing fixed points in recurrent neural networks. *Journal of Open Source Software*, 3(31):1003, 2018.
- [19] Emily L Sylwestrak, YoungJu Jo, Sam Vesuna, Xiao Wang, Blake Holcomb, Rebecca H Tien, Doo Kyung Kim, Lief Fenno, Charu Ramakrishnan, William E Allen, et al. Cell-type-specific population dynamics of diverse reward computations. *Cell*, 185(19):3568–3587, 2022.
- [20] Google Research. Computation through dynamics. <https://github.com/google-research/computation-thru-dynamics>, 2025. Accessed: 2025-04-28.
- [21] TensorFlow Authors. Lfads: Latent factor analysis via dynamical systems (tensorflow models). <https://github.com/tensorflow/models/tree/master/research/lfads>, 2025. Accessed: 2025-04-28.

- [22] Andrew R Sedler and Chethan Pandarinath. Ifads-torch: A modular and extensible implementation of latent factor analysis via dynamical systems. *arXiv preprint arXiv:2309.01230*, 2023.
- [23] Mark M Churchland, John P Cunningham, Matthew T Kaufman, Justin D Foster, Paul Nuyujukian, Stephen I Ryu, and Krishna V Shenoy. Neural population dynamics during reaching. *Nature*, 487(7405):51–56, 2012.
- [24] Laura N Driscoll, Krishna Shenoy, and David Sussillo. Flexible multitask computation in recurrent networks utilizes shared dynamical motifs. *Nature Neuroscience*, 27(7):1349–1363, 2024.
- [25] Guangyu Robert Yang, Michael W Cole, and Kanaka Rajan. How to study the neural mechanisms of multiple tasks. *Current opinion in behavioral sciences*, 29:134–143, 2019.
- [26] Guangyu Robert Yang, Madhura R Joglekar, H Francis Song, William T Newsome, and Xiao-Jing Wang. Task representations in neural networks trained to perform many cognitive tasks. *Nature neuroscience*, 22(2):297–306, 2019.
- [27] Francesca Mastrogiuseppe and Srdjan Ostojic. Linking connectivity, dynamics, and computations in low-rank recurrent neural networks. *Neuron*, 99(3):609–623, 2018.
- [28] Friedrich Schuessler, Alexis Dubreuil, Francesca Mastrogiuseppe, Srdjan Ostojic, and Omri Barak. Dynamics of random recurrent networks with correlated low-rank structure. *Physical Review Research*, 2(1):013111, 2020.
- [29] Manuel Beiran, Alexis Dubreuil, Adrian Valente, Francesca Mastrogiuseppe, and Srdjan Ostojic. Shaping dynamics with multiple populations in low-rank recurrent networks. *Neural computation*, 33(6):1572–1615, 2021.
- [30] Alexis Dubreuil, Adrian Valente, Manuel Beiran, Francesca Mastrogiuseppe, and Srdjan Ostojic. The role of population structure in computations through neural dynamics. *Nature neuroscience*, 25(6):783–794, 2022.
- [31] Matthijs Pals, A Erdem Sağtekin, Felix Pei, Manuel Gloeckler, and Jakob H Macke. Inferring stochastic low-rank recurrent neural networks from neural data. *arXiv preprint arXiv:2406.16749*, 2024.
- [32] Daniel Durstewitz. A state space approach for piecewise-linear recurrent neural networks for identifying computational dynamics from neural measurements. *PLoS computational biology*, 13(6):e1005542, 2017.
- [33] Florian Hess, Zahra Monfared, Manuel Brenner, and Daniel Durstewitz. Generalized teacher forcing for learning chaotic dynamics. *arXiv preprint arXiv:2306.04406*, 2023.
- [34] Carina Curto, Jesse Geneson, and Katherine Morrison. Fixed points of competitive threshold-linear networks. *Neural computation*, 31(1):94–155, 2019.
- [35] Manuel Brenner, Florian Hess, Jonas M Mikhaeil, Leonard F Bereska, Zahra Monfared, Po-Chen Kuo, and Daniel Durstewitz. Tractable dendritic rnns for reconstructing nonlinear dynamical systems. In *International conference on machine learning*, pages 2292–2320. Pmlr, 2022.
- [36] Lukas Eisenmann, Zahra Monfared, Niclas Göring, and Daniel Durstewitz. Bifurcations and loss jumps in rnn training. *Advances in Neural Information Processing Systems*, 36:70511–70547, 2023.
- [37] Katherine Morrison, Anda Degeratu, Vladimir Itskov, and Carina Curto. Diversity of emergent dynamics in competitive threshold-linear networks. *arXiv preprint arXiv:1605.04463*, 2016.
- [38] Timothy D Kim, Thomas Z Luo, Jonathan W Pillow, and Carlos D Brody. Inferring latent dynamics underlying neural population activity via neural differential equations. In *International Conference on Machine Learning*, pages 5551–5561. PMLR, 2021.

- [39] Andrew R Sedler, Christopher Versteeg, and Chethan Pandarinath. Expressive architectures enhance interpretability of dynamics-based neural population models. *Neurons, behavior, data analysis, and theory*, 2023:10–51628, 2023.
- [40] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
- [41] David A Sabatini and Matthew T Kaufman. Reach-dependent reorientation of rotational dynamics in motor cortex. *Nature Communications*, 15(1):7007, 2024.
- [42] Francis R Willett, Erin M Kunz, Chaoferi Fan, Donald T Avansino, Guy H Wilson, Eun Young Choi, Foram Kamdar, Matthew F Glasser, Leigh R Hochberg, Shaul Druckmann, et al. A high-performance speech neuroprosthesis. *Nature*, 620(7976):1031–1036, 2023.

6 APPENDIX

6.1 Variational Training Objective

The model is trained to maximize the evidence lower bound (ELBO) on the log-likelihood, summed over all data samples. The ELBO of the observed data sample i is given by:

$$\begin{aligned} \mathcal{L}^i = & \mathbb{E}_{Q(\mathbf{g}_{i,0}, \mathbf{u}_{i,t}, \mathbf{b}_i | \mathbf{x}_{i,1:T})} \log P(\mathbf{x}_{i,1:T} | \mathbf{g}_{i,0}, \mathbf{u}_{i,t}, \mathbf{b}_i) - D_{\text{KL}}(Q_{\text{enc},g}(\mathbf{g}_{i,0} | \mathbf{x}_{i,1:T}) \| P(\mathbf{g}_0)) \\ & - \sum_{t=1}^{t=T} D_{\text{KL}}(Q_{\text{enc},u}(\mathbf{u}_{i,t} | \mathbf{x}_{i,1:T}) \| P(\mathbf{u})) - D_{\text{KL}}(Q_{\text{enc},b}(\mathbf{b}_i | \mathbf{x}_{i,1:T}) \| P(\mathbf{b})) \end{aligned}$$

where the expectation is using the factorized distribution $Q(\mathbf{g}_{i,0}, \mathbf{u}_{i,t}, \mathbf{b}_i | \mathbf{x}_{i,1:T}) = Q_{\text{enc},g}(\mathbf{g}_{i,0} | \mathbf{x}_{i,1:T}) Q_{\text{enc},b}(\mathbf{b}_i | \mathbf{x}_{i,1:T}) \prod_t Q_{\text{enc},u}(\mathbf{u}_{i,t} | \mathbf{x}_{i,1:T})$. We assume isotropic Gaussian priors $P(\mathbf{g}_0), P(\mathbf{u}), P(\mathbf{b})$. Posterior distributions are parametrized by encoder networks. The data likelihood term corresponds to the Poisson log-likelihood of observed spike counts under generated rates.

6.2 Implementation and Training

Both the standard and modified LFADS models are implemented in Tensorflow 2 using GRU cells for the generator network. We use bidirectional RNN encoders for $\mathbf{g}_0, \mathbf{u}_{i,t}, \mathbf{b}_i$, and KL annealing during training. The dimensionality of the per-trial bias vector $\mathbf{u}_i$ is typically constrained to 2–10 dimensions.

Model optimization uses Adam with gradient clipping and early stopping. Monkey maze data was binned at 10 ms resolution and human iBCI data was binned at 15ms resolution. We regularize the generator dynamics via L2 weight penalties and restrict $\mathbf{u}_i$ through KL divergence and dimensionality limits.

6.3 Fixed Point and Linear Dynamics Analysis

To interpret the dynamical system learned by the RNN generator, we identify fixed points or slow points and analyze the linearized dynamics around them [17, 18]. A fixed point $\mathbf{g}^*$ of the generator dynamics satisfies:

$$\mathbf{g}^* = \text{RNN}_{\text{gen}}(\mathbf{g}^*, [\mathbf{u}_{i,t} = 0, \mathbf{b}_i]) \quad (9)$$

Since RNNs do not always yield exact algebraic solutions for fixed points, we identify *slow points* by minimizing the norm of the velocity vector:

$$\hat{\mathbf{g}} = \arg \min_{\mathbf{g}} \|\mathbf{g} - \text{RNN}_{\text{gen}}(\mathbf{g}, [\mathbf{u}_{i,t} = 0, \mathbf{b}_i])\|^2 \quad (10)$$

Optimization is performed using gradient descent from various initializations, seeded from latent states during real trials. Points for which the update norm falls below a fixed threshold (e.g., 10^{-7}) are considered valid slow points.

To study local behavior near a slow point, we compute the Jacobian of the update function:

$$\mathbf{J}(\mathbf{b}_i) = \left. \frac{\partial \text{RNN}_{\text{gen}}(\mathbf{g}, [\mathbf{u}_{i,t} = 0, \mathbf{b}_i])}{\partial \mathbf{g}} \right|_{\mathbf{g}=\hat{\mathbf{g}}} \quad (11)$$

The eigenvalues and eigenvectors of $\mathbf{J}(\mathbf{b}_i)$ characterize the linearized dynamics:

- The real part of each eigenvalue indicates the stability of the corresponding mode (negative implies damping).
- Complex conjugate pairs indicate oscillatory modes, with imaginary parts determining the frequency and real parts controlling the rate of growth or decay.

- Eigenvectors (or generalized eigenspaces) define the oscillation plane or direction.

This analysis reveals how different trial conditions (via $\mathbf{b}_i$) induce shifts in fixed point locations, the orientation of eigenplanes, and oscillation frequencies, as shown in the pendulum and monkey reaching data.

6.4 Hyperparameters

All hyperparameters were set by optimizing ELBO on validation partition or manual quasi-optimization. The hyperparameters chosen for different analyses are listed in Table 3.

Hyperparameter	Simulated Pendulum	Macaque Maze dataset	Human iBCI finger movement	Human iBCI Speech dataset
Initial condition(I.C.) dimension	5	128	10	128
I.C. encoder hidden dimension	128	128	128	128
Time-varying (T.V.) input dimension	N/A	N/A	1	N/A
T.V. encoder hidden dimension	N/A	N/A	128	N/A
T.V. Input smoothness α	N/A	N/A	0	N/A
Bias dimension	5	10	10	10
Bias encoder hidden dimension	128	128	128	128
Generator hidden dimension	200	200	200	128
Dropout rate	0	0	0	0
Factor dimensions	40	40	40	40
L2 loss on trainable parameters	0	0	1	0.1

Table 3: Hyperparameters used to fit the contextual LFADS model for each dataset.

6.5 Data augmentation for addressing non-stationarities in neural recordings

We outline the details of the non-stationarity correction method outlined in Section 4.3. The neural data consists of 966 trials with 52-54 trials for each of the 16 movements, and 107 trials without a movement.

Movements were cued with a continuous video of finger animation. For each trial, markers showed the final position of each finger and the fingers slowly moved from initial position to the target positions.

Cued target finger positions consist of two degrees of freedom (thumb vs all other fingers constrained to move together), and each finger group was cued independently to be flexed/extended/remain at rest. Trials alternated between fingers going from resting position to a target position and back. The iBCI participant was instructed to attempt to move his own fingers to follow the movement of the fingers in the animation. Multi-unit threshold crossings were recorded concurrent to attempted movements.

A linear finger-position decoder takes threshold crossings ($\mathbf{x}_t$) and predicts continuous finger positions ($\mathbf{p}_t$) at the corresponding time point: $\hat{\mathbf{p}}_t = A_{\text{decoder}}\mathbf{x}_t + b_{\text{decoder}}$. Our goal is to learn the decoder for the last quarter of the trials (trial numbers 724 – 966).

To evaluate different methods, data was randomly split into training (60% of trials), validation (20% of trials) and testing (rest of the 20% of trials). Note that trials in the target duration (trials 724 – 966) were present in both training and testing partitions.

Below we detail the steps for using contextual LFADS for data augmentation, by correcting the bias for each trial.

1. Identify the per-trial bias by applying the bias-encoder $\mathbf{b}_i$.
2. Divide the training trials into two sets based on trial ID, the non-target set $\mathcal{S}_1 = [0 : 724]$ and the target set $\mathcal{S}_2 = [724 : 966]$.
3. Compute the average bias for each set: $\mathbf{b}_{\mathcal{S}_1} = \frac{1}{|\mathcal{S}_1|} \sum_{i \in \mathcal{S}_1} \mathbf{b}_i$; and $\mathbf{b}_{\mathcal{S}_2} = \frac{1}{|\mathcal{S}_2|} \sum_{i \in \mathcal{S}_2} \mathbf{b}_i$.
4. Apply bias correction for each trial i in the non-target set $\mathcal{S}_1$: $\mathbf{b}_{i,\text{corrected}} = \mathbf{b}_i - \mathbf{b}_{\mathcal{S}_1} + \mathbf{b}_{\mathcal{S}_2}$. No change applied to trials in target set $\mathcal{S}_2$.

We applied identical steps for correcting the initial condition (yielding $\mathbf{g}_{0,i,\text{corrected}}$) for trials in the non-target set $\mathcal{S}_1$.

The drift corrected neural activity was generated starting from the corrected initial condition as follows:

$$\mathbf{g}_{i,t} = \mathbf{g}_{0,i,\text{corrected}} \quad (12)$$

$$\mathbf{g}_{i,t} = \text{RNN}_{\text{gen}}(\mathbf{g}_{i,t-1}, [\mathbf{u}_{i,t}, \mathbf{b}_{i,\text{corrected}}]) \quad (13)$$

$$\mathbf{f}_{i,t} = W_{\text{fac}} \mathbf{g}_{i,t} + \mathbf{b}_{\text{fac}} \quad (14)$$

$$\mathbf{r}_{i,t} = \exp(W_{\text{rate}} \mathbf{f}_{i,t}) \quad (15)$$

$$\mathbf{x}_{i,t,\text{corrected}} \sim \text{Poisson}(\mathbf{r}_{i,t}) \quad (16)$$

Finally, the corrected firing rate samples for trials in $\mathcal{S}_1$ were combined with uncorrected firing rate samples for trials in $\mathcal{S}_2$ to estimate the finger-position decoder. The decoder was subsequently evaluated on LFADS-smoothed firing rate on the test data.