Open access • Posted Content • DOI:10.1101/2021.01.13.426570

A large-scale neural network training framework for generalized estimation of single-trial population dynamics — Source link

Mohammad Reza Keshtkaran, Andrew R. Sedler, Raeed H. Chowdhury, Raeed H. Chowdhury ...+9 more authors

Institutions: Georgia Institute of Technology, Northwestern University, University of Pittsburgh, University of California, San Diego ...+2 more institutions

Published on: 15 Jan 2021 - bioRxiv (Cold Spring Harbor Laboratory)

Topics: Deep learning, Artificial neural network and Population

A large-scale neural network training framework for generalized estimation of single-trial population dynamics

Mohammad Reza Keshtkaran[1,+], Andrew R. Sedler[2,1,+], Raeed H. Chowdhury[3,4,^], Raghav Tandon[1,^], Diya Basrai[1,5], Sarah L. Nguyen[6], Hansem Sohn[7], Mehrdad Jazayeri[7], Lee E. Miller[8,4,9,10], Chethan Pandarinath[1,11,2*]

1. Wallace H. Coulter Department of Biomedical Engineering, Emory University and Georgia Institute of Technology, Atlanta, GA, USA
2. Center for Machine Learning, Georgia Institute of Technology, Atlanta, GA, USA
3. Department of Biomedical Engineering, Northwestern University, Evanston, IL, USA
4. Department of Bioengineering, University of Pittsburgh, Pittsburgh, PA, USA
5. Physiology and Neuroscience, University of California San Diego, La Jolla, CA, USA
6. College of Computing, Georgia Institute of Technology, Atlanta, GA, USA
7. Department of Brain and Cognitive Sciences, McGovern Institute for Brain Research, Massachusetts Institute of Technology, Cambridge, MA, USA
8. Department of Physiology, Northwestern University, Chicago, IL, USA
9. Department of Physical Medicine and Rehabilitation, Northwestern University, Chicago, IL, USA
10. Shirley Ryan AbilityLab, Chicago, IL, USA
11. Department of Neurosurgery, Emory University, Atlanta, GA, USA

+ Authors contributed equally
^ Authors contributed equally
* Correspondence: chethan [at] gatech.edu

Abstract

Large-scale recordings of neural activity are providing new opportunities to study network-level dynamics. However, the sheer volume of data and its dynamical complexity are critical barriers to uncovering and interpreting these dynamics. Deep learning methods are a promising approach due to their ability to uncover meaningful relationships from large, complex, and noisy datasets. When applied to high-D spiking data from motor cortex (M1) during stereotyped behaviors, they offer improvements in the ability to uncover dynamics and their relation to subjects' behaviors on a millisecond timescale. However, applying such methods to less-structured behaviors, or in brain areas that are not well-modeled by autonomous dynamics, is far more challenging, because deep learning methods often require careful hand-tuning of complex model hyperparameters (HPs). Here we demonstrate AutoLFADS, a large-scale, automated model-tuning framework that can characterize dynamics in diverse brain areas without regard to behavior. AutoLFADS uses distributed computing to train dozens of models simultaneously while using evolutionary algorithms to tune HPs in a completely unsupervised way. This enables accurate inference of dynamics out-of-the-box on a variety of datasets, including data from M1 during stereotyped and free-paced reaching, somatosensory cortex during reaching with perturbations, and frontal cortex during cognitive timing tasks. We present a cloud software package and comprehensive tutorials that enable new users to apply the method without needing dedicated computing resources.

Introduction

Ongoing advances in neural interfacing technologies are enabling simultaneous monitoring of the activity of large neural populations across a wide array of brain areas and behaviors (1–5). Such technologies may fundamentally change the questions we can address about computations within a neural population, allowing neuroscientists to shift focus from understanding how individual neurons' activity relates to externally-measurable or controllable parameters, toward understanding how neurons within a network coordinate their activity to perform computations underlying those behaviors. A natural method for interpreting these complex, high-dimensional datasets is that of neural population dynamics (6–8). The dynamical systems framework centers on uncovering coordinated patterns of activation across a neural population and characterizing how these patterns change over time. Knowledge of these hidden dynamics has provided new insights into how neural populations implement the computations necessary for motor, sensory, and cognitive processes (9–15).

A focus on population dynamics could also facilitate a shift away from reliance on stereotyped behaviors and trial-averaged neural responses. Standard approaches must typically average activity across trials, sacrificing single trial interpretability for robustness against what is perceived as noise in single trials. However, as articulated by Cunningham and Yu (16): "If the neural activity is not a direct function of externally measurable or controllable variables (for example, if activity is more a reflection of internal processing than stimulus drive or measurable behavior), the time course of neural responses may differ substantially on nominally identical trials." This may be especially true of non-primary cortical areas, and cognitively demanding tasks that involve decision-making, allocation of attention, or varying levels of motivation.

To move beyond this bottleneck, high-time resolution single-trial analyses are essential. These can be enabled by a combination of neural population recordings and novel analytical tools like those proposed here. Single-trial, population-level analyses benefit from two principles of the dynamical systems view: first, that simultaneously recorded neurons are not independent, but rather exhibit coordinated patterns of activation that reflect the state of the overall network rather than individual neurons. Second, the coordinated patterns evolve over time in ways that are largely predictable based on the population's internal dynamics. Thus, while it may be challenging to accurately estimate the network's state based solely on activity observed at a single time point, knowledge of how the state evolves can constrain an estimate at any given time point.

Several approaches have been developed to infer latent dynamical structure from neural population activity on individual trials, including a growing number that leverage artificial neural networks (17–22). One such method, latent factor analysis via dynamical systems (LFADS) (22,20) achieved precise inference of motor cortical firing rates on single trials of stereotyped behaviors, enabling accurate prediction of subjects' behaviors on a moment-by-moment, millisecond timescale (20). Further, in tasks with unpredictable events, a modified network architecture enabled inference of dynamical perturbations that corresponded to how subjects ultimately responded to the unpredictable events.

Though highly effective, artificial neural networks, including LFADS, typically have many thousands of parameters, and potentially dozens of non-trainable hyperparameters (HPs) that need to be tuned to achieve good performance. HPs include architecture parameters like the type, dimensionality, and number of various layers, as well as regularization and optimization parameters. Until recently, the HP optimization problem was typically addressed by an iterative manual process, a random search, or some combination of the two. In the past several years, a host of more advanced approaches promises to eliminate the tedious work and domain knowledge required for manual tuning while performing better and more efficiently than random search (23–25). The form and variety of possible neuroscientific datasets present unique challenges that make HP optimization a particularly impactful problem (26). Thus, bringing efficient HP search algorithms to neuroscience could allow more effective experimentation with models based on artificial neural networks, like LFADS.

Here we present AutoLFADS, a framework for large-scale, automated model tuning that enables accurate single-trial inference of neural population dynamics across a range of brain areas and behaviors. We evaluate AutoLFADS using data from three cortical regions: primary motor and dorsal premotor cortex (M1/PMd), somatosensory cortex area 2, and dorsomedial frontal cortex (DMFC). The tasks span a mix of functions where population activity can be well-modeled by autonomous dynamics (e.g., pre-planned reaching movements, estimation of elapsed time) and those for which population activity is responsive to external inputs (e.g., mechanical perturbations, unexpected appearance of reaching targets, variable timing cues).

Using this broad range of datasets, we show that AutoLFADS achieves high-time resolution, single-trial inference of neural population dynamics, surpassing LFADS in all scenarios tested. Remarkably, AutoLFADS does this in a completely unsupervised manner that does not depend on the knowledge of the tasks, subjects' behaviors, or brain areas. In all applications, the method is applied "out of the box" without careful adjustment for each dataset. We believe these capabilities greatly extend the range of neuroscientific applications for which accurate inference of single-trial population dynamics should be achievable, and substantially lower the barrier to entry for applying these methods. Finally, we present a cloud software package and comprehensive tutorials to enable new users without machine learning expertise or dedicated computing resources to apply AutoLFADS successfully.

Results

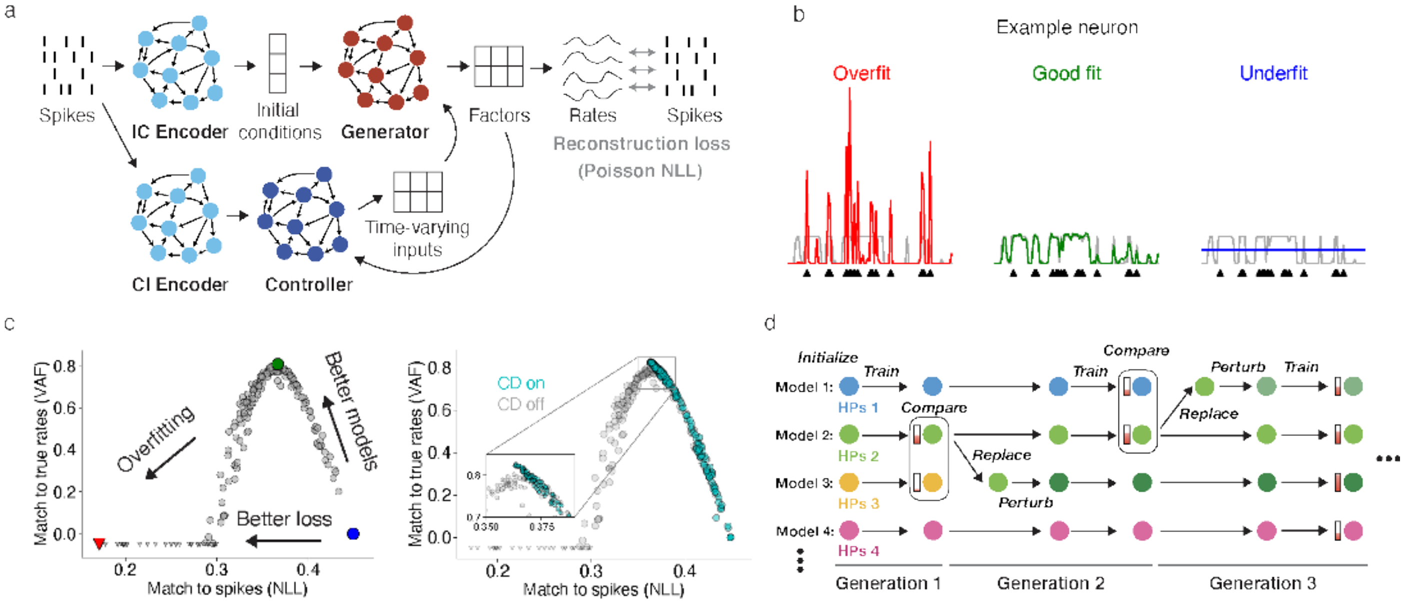

Fig1 | AutoLFADS combines a novel neural network regularization method with a large-scale framework for automated hyperparameter optimization. (a) Schematic of the LFADS architecture, showing how the generative model infers the firing rates that underlie the observed spikes. (b) Examples of LFADS-inferred rates (colored) and the corresponding synthetic input data (spikes, shown as black triangles) and data-generating distribution (ground truth rates, shown as gray traces) for three fitting modes. (c) Left: performance of 200 LFADS models with random HPs in matching the spikes and the known rates of a synthetic dataset, measured by negative log-likelihood (NLL) and variance accounted for (VAF) respectively. Colored points indicate the models that produced the rates in the previous panel. Right: same as previous, but for models trained with CD. (d) Schematic of the PBT approach to HP optimization. Each colored circle represents an LFADS model with a certain HP configuration and partially filled bars represent model performance. Models are trained for fixed intervals (generations), between which poorly-performing models are replaced by copies of better-performing models with perturbed HPs.

LFADS architecture

The LFADS architecture (**Fig. 1a**) has been detailed previously (20,22,26). Briefly, LFADS is based on the idea that the evolution of a neural population's activity in time can be modeled as a non-autonomous dynamical system, i.e., a dynamical system whose state evolution is influenced by both internal dynamics and external inputs. This dynamical system is approximated by a recurrent neural network (RNN) known as the generator. Observed spiking activity from each neuron is assumed to reflect an underlying firing rate that is linked to the state of the generator at each timestep. Separately, to enable modeling of input-driven dynamical systems, time-varying inputs are inferred by a controller RNN, which receives as input an encoding of the spike count data as well as the generator's output at the previous time step. This architecture is a modification of a sequential variational autoencoder (VAE) (22,27,28). When training the model, the objective is to maximize a lower bound on the Poisson likelihood of the observed spiking activity given the inferred rates (see *Methods* for details).

It is imperative to regularize the model properly in order to extract useful spike rates (**Fig. 1b**) (26). This can be achieved through HP optimization. The two main classes of LFADS HPs are those that set the network architecture (e.g., number of units in each RNN, dimensionality of initial conditions, inputs, and factors), and those that control regularization and training (e.g., L2 penalties, scaling factors for KL penalties, dropout probability, and learning rate; described in *Methods*). The optimal values of these HPs could depend on various factors such as dataset size, dynamical structure underlying the activity of the brain region being modeled, and the behavioral task.

A critical challenge for autoencoders is that automatic HP searches face a type of overfitting that is particularly hard to address (26). Given enough capacity, the model can find a trivial solution where it simply passes individual spikes from the input to the output firing rates, akin to an identity transformation of the input without modeling any meaningful structure underlying the data (**Fig. 1b**). Importantly, such pathological overfitting is not detectable by standard validation likelihood, as the failure mode also results in high likelihood and poor modeling of validation data. We performed a 200-model random search over a space of KL, L2, and dropout regularization HPs that was empirically determined to yield both underfitting and overfitting models on a synthetic dataset (see *Methods* for a description of the dataset). Models that appear to have the best likelihoods actually exhibit poor inference of underlying firing rates, indicating a type of pathological overfitting (**Fig. 1c**, left). This phenomenon is also consistently observed on real data throughout this paper: better validation loss did not indicate better performance for any of our decoding or PSTH-based metrics.

The lack of a reliable validation metric has prevented automated HP searches because it is unclear how one should select between models when underlying firing rates are unavailable or non-existent. To address this issue, we developed a novel regularization technique called coordinated dropout (CD) that forces the network to model only structure that is shared across neurons (26). After applying CD, we repeated the previous test on synthetic data using 200 LFADS models from the same HP search space, and found that they no longer overfit spikes (**Fig. 1c**, right). CD restored the correspondence between model quality assessed from matching spikes (validation likelihood) and matching rates, allowing the former to be used as a surrogate when the latter is not available.

The premise of this paper is that this reliable validation metric should enable large-scale HP searches and fully-automated selection of high-performing neuroscientific models despite having no access to ground truth firing rates. To test this, we needed an efficient HP search strategy. We chose a recent method based on parallel search called Population Based Training (PBT; **Fig. 1d**) (25,29). PBT distributes training across dozens of models simultaneously, and uses evolutionary algorithms to tune HPs over many generations. Because PBT distributes model training over many workers, it matches the scalability of parallel search methods such as random or grid search, while achieving higher performance with the same amount of computational resources (25,29).

These two key modifications - a novel regularization strategy (CD) that results in a reliable validation metric, and an efficient approach to HP optimization (PBT) - yield a large-scale, automated framework for model tuning, which we refer to as AutoLFADS. In the following sections, we test the performance of AutoLFADS on previously characterized datasets, as well as novel ones. We start by evaluating AutoLFADS using data from M1/PMd in a structured reaching task to investigate the model's performance on a well-characterized dataset that had been previously used to benchmark the performance of LFADS (20,26). On this data, we demonstrate that proper HP tuning leads to models that consistently outperform LFADS and that this gap grows substantially when data are limited. Next, we move to assessing the ability of AutoLFADS to approximate input-driven dynamics, using data from M1 in a random target task, data from area 2 in a reaching task with mechanical perturbations, and data from DMFC in a cognitive timing task. In each case, by several metrics, AutoLFADS consistently achieves better results than random searches that used three times the computational resources, despite performing model selection in a completely unsupervised fashion.

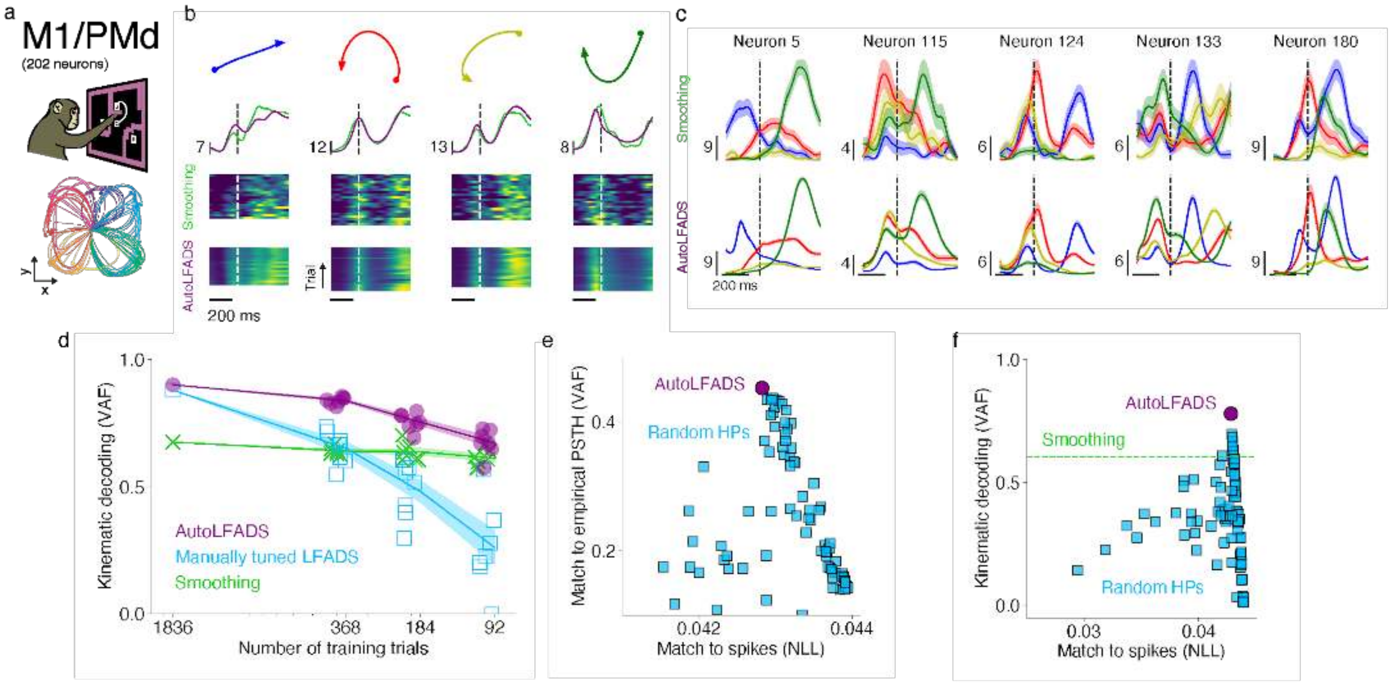

Fig2 | Application of AutoLFADS to data from motor cortex. (a) Schematic of the maze task (top), and representative reach trajectories across 108 total conditions, colored by target location (bottom). (b) Average reach trajectories (top), PSTHs (second row) and single-trial firing rates (bottom) obtained by smoothing (Gaussian kernel, 30 ms s.d.) or AutoLFADS for a single neuron across 4 reach conditions. All data is modeled at 2 ms bins. Dashed lines indicate movement onset and vertical scale bars denote rates (spikes/s). (c) PSTHs produced by smoothing spikes (top) or by applying AutoLFADS (bottom), for 5 example neurons. Shaded regions are standard errors. Movement onset and rate scales are denoted as in the previous panel. (d) Performance in decoding reaching kinematics (arm velocities) as a function of training dataset size. Trial counts exclude the 20% of trials for each dataset size that were held-out for model evaluation. We decoded X and Y arm velocities from smoothed spikes, rates inferred by LFADS with manually-tuned hyperparameters (HPs), and rates inferred by AutoLFADS. Accuracy was quantified by VAF. Lines and shading denote mean +/- standard error across 7 models trained on randomly-drawn subsets of the full dataset. (e) Performance in replicating the empirical PSTHs computed on all trials using rates inferred from a 184-trial training set using AutoLFADS and LFADS with random HPs (100 models). (f) Hand velocity decoding performance for firing rates from a 184-trial training set (same models as in (e)).

AutoLFADS outperforms original LFADS when applied on benchmark data from M1/PMd

We first evaluated AutoLFADS on data from motor cortex during a highly stereotyped behavior, which was used to assess the original LFADS method (20). We used 202 neurons simultaneously recorded from M1 and PMd during a maze reaching task (see *Methods*) in which a monkey made a variety of straight and curved reaches after a delay period following target presentation (**Fig. 2a**; dataset consisted of 2296 individual reach trials spanning 108 reach types). Previous analyses of the delayed reaching paradigm demonstrated that activity during the movement period is well modeled as an autonomous dynamical system (10,20). In this abstract model, the temporal evolution of the neural population's activity is predictable based on the state it reaches during the delay period. Therefore, previous work modeled these data with a simplified LFADS configuration which could only approximate autonomous dynamics (20). However, this simplified model is not applicable more broadly to situations in which both autonomous dynamics and external inputs might be needed to describe neural activity. Therefore, in this paper we do not constrain the network architecture to only model autonomous dynamics for any applications tested, to determine whether AutoLFADS can automatically adjust the degree to which autonomous dynamics and inputs are needed to model the data.

AutoLFADS operates on unlabeled segments of binned spiking data and infers firing rates for each neuron in an unsupervised manner. Consistent with previous applications of LFADS on this dataset (20,26), the firing rates inferred by AutoLFADS for 2 ms bins exhibited clear and consistent structure on individual trials (**Fig. 2b,** bottom). We also verified that these firing rates captured features of the neural responses revealed by averaging across trials, a common method of de-noising neural activity (**Fig. 2b**, second row, and **Fig. 2c**).

A generalizable method should be able to perform well across the broad range of dataset sizes typical of neuroscience experiments. To test this, we compared AutoLFADS and manually-tuned LFADS models that were trained using either the full dataset (2296 trials), or randomly sampled subsets containing 5, 10, and 20% of the trials. We first tested the degree to which the representations produced by the models were informative about observable behavior, which we quantified by decoding the monkey's hand velocity from the inferred rates using optimal linear estimation (**Fig 2d**). At the largest dataset size, decoding performance for AutoLFADS and manually-tuned LFADS was comparable. This result fits with standard intuition that performance is less sensitive to HPs when sufficient data are available. However, for all three reduced dataset sizes, the AutoLFADS outperformed the manually-tuned model ($p<0.05$ for all three sizes, paired, one-tailed Student's t-test).

While this result is promising, the difference in robustness to dataset size between AutoLFADS and LFADS could have resulted from a particularly poor selection of HPs during manual tuning. To control for this possibility, we chose one of the smaller data subsets (184 trials) and trained 100 additional LFADS models with randomly-selected HPs. We evaluated the models' performance in two ways: how accurately the models replicated the empirical trial-averaged firing rates (PSTHs; **Fig. 2e**), and how accurately arm velocity could be decoded from inferred rates (**Fig. 2f**). While the LFADS models achieved a broad range of performance, models with better validation likelihoods did not achieve better inference of firing rates, mirroring our earlier findings with synthetic data (**Fig. 1c**). Thus it is unclear how one could select amongst the LFADS models with random HPs without some supervised intervention. In contrast, the single AutoLFADS model, chosen in a completely unsupervised fashion, outperformed all LFADS models for both performance metrics.

Taken together, these results show that even if one performed a random search and then selected a model using a supervised approach (e.g., based on reconstruction of empirical PSTHs or decoding accuracy), its performance would still be substantially lower than that of AutoLFADS. Additionally, this validation - i.e., that the unsupervised approach produces high-performing models - provides evidence that even in cases where such supervision is unavailable (e.g., settings that lack clear task structure or measurement of behavioral variables), AutoLFADS models will still be high performing.

AutoLFADS uncovers population dynamics without structured trials

To-date, most efforts to tie dynamics to neural computations have used experiments where subjects perform constrained tasks with repeated, highly structured trials. For example, motor cortical dynamics are often framed as a computational engine to link the processes of motor preparation and execution (6–8). To interrogate these dynamics, most studies use a delayed-reaching paradigm that creates explicit pre-movement and movement periods. However, constrained behaviors may have multiple drawbacks in studying dynamics. First, it is unclear whether such artificial paradigms are good proxies for everyday behaviors. Second, highly constrained, repeated behaviors might impose artificial limits on the properties of the uncovered dynamics, such as the measured dimensionality of the neural population activity (30). Even outside of movement neuroscience, the requirement that we conduct many repetitions of constrained tasks significantly hinders our ability to study a rich sample of the dynamics of a given neural population. Accurate inference of neural dynamics without these constraints could facilitate dynamics-based analyses of richer datasets that are more reflective of the brain's natural behavior.

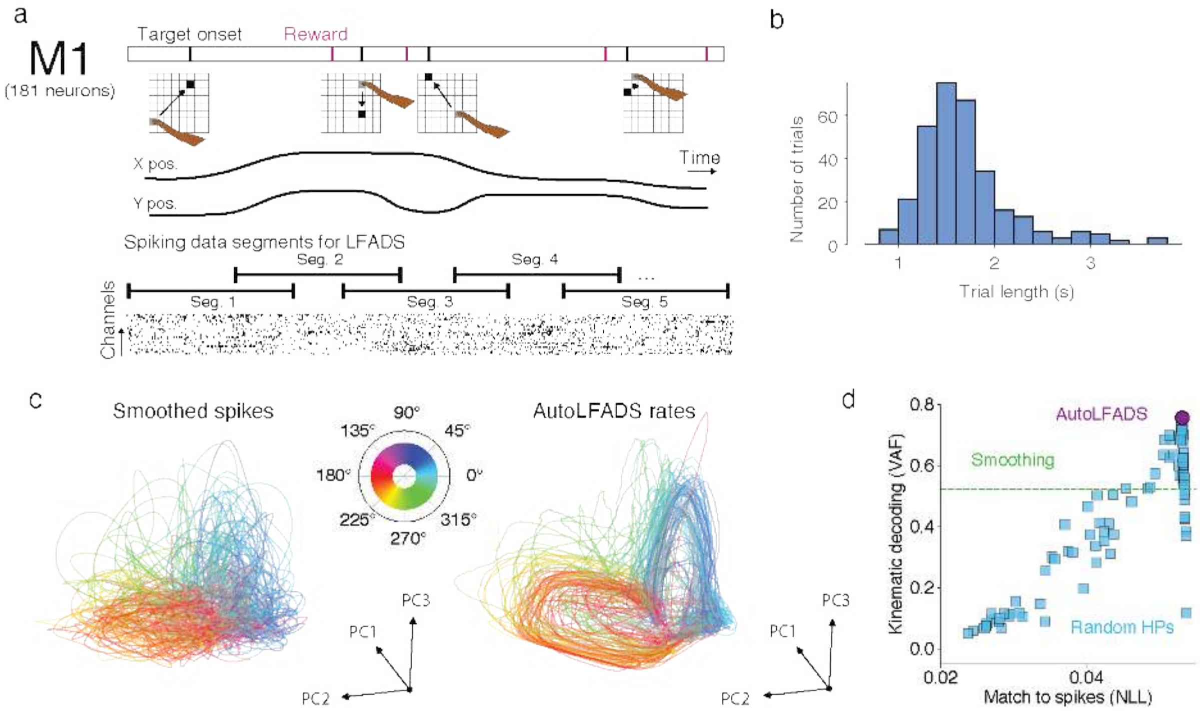

Fig3 | Modeling neural activity in M1 without knowledge of trial or task information. (a) Top: Schematic of the random target task, which lacks stereotyped trial structure and delay periods. Bottom: Continuous neural activity (spiking data) recorded during back-to-back reaching trials was divided into 600 ms segments with 200 ms of overlap between adjacent segments. After modeling by AutoLFADS, the inferred firing rates from different segments were merged together to create a continuous segment, using a weighted average of data at overlapping timepoints. (b) Distributions of trial lengths (time between onsets of successive targets) for 313 total trials. (c) Subspaces of neural activity extracted using PCA and colored by angle to the target. Left: 3D subspace that captures the most variance in smoothed spiking activity. Center: Subspace that captures the most variance in AutoLFADS rates. (d) Accuracy in decoding hand velocity from firing rates inferred by smoothing, 100 LFADS models with random HPs, and AutoLFADS.

In order to provide access to a much broader range of experimental data, we tested whether AutoLFADS could model data without regard to trial structure. We applied AutoLFADS to neural activity from a monkey performing a continuous, self-paced random target reaching task (**Fig. 3a**, top) (31), in which each movement started and ended at a random position, and movements were highly variable in duration (**Fig. 3b**). Analysis of data without consistent temporal structure repeated across trials is challenging, as trial-averaging is not feasible. Even the available single-trial analytical methods have typically relied on strong simplifying assumptions that are not applicable to less-structured tasks. For example, previous efforts to uncover motor cortical dynamics during single reaches have been able to consider only brief data segments that begin with the arm at a consistent starting point, and relied on behavioral events such as target or movement onset to align trials before analysis (17,20,26,32–36).

Like most machine learning algorithms, AutoLFADS operates on discrete, fixed-length segments of neural data. To create these segments from a task with highly variable timing, we chopped an approximately 9 minute window of continuous neural data into 600 ms segments with 200 ms of overlap (**Fig. 3a**, bottom) without regard to trial boundaries. After modeling with AutoLFADS, we merged inferred firing rates from individual segments, which yielded inferred rates for the original continuous window. We then analyzed the inferred rates by aligning the data to movement onset for each trial (see *Methods*). Even though the dataset was modeled without the use of trial information, inferred firing rates during the reconstructed trials exhibited consistent progression in an underlying state space, with clear structure that corresponded with the monkey's reach direction on each trial (**Fig. 3c**, right). Further, the inferred firing rates were highly informative about moment-by-moment details of the measured reaching movements: AutoLFADS enabled decoding of continuous hand velocities with substantially higher accuracy than did smoothing (R^2 of 0.76 for AutoLFADS v. 0.52 for smoothing), and it also outperformed all LFADS models with random HPs (**Fig. 3d**).

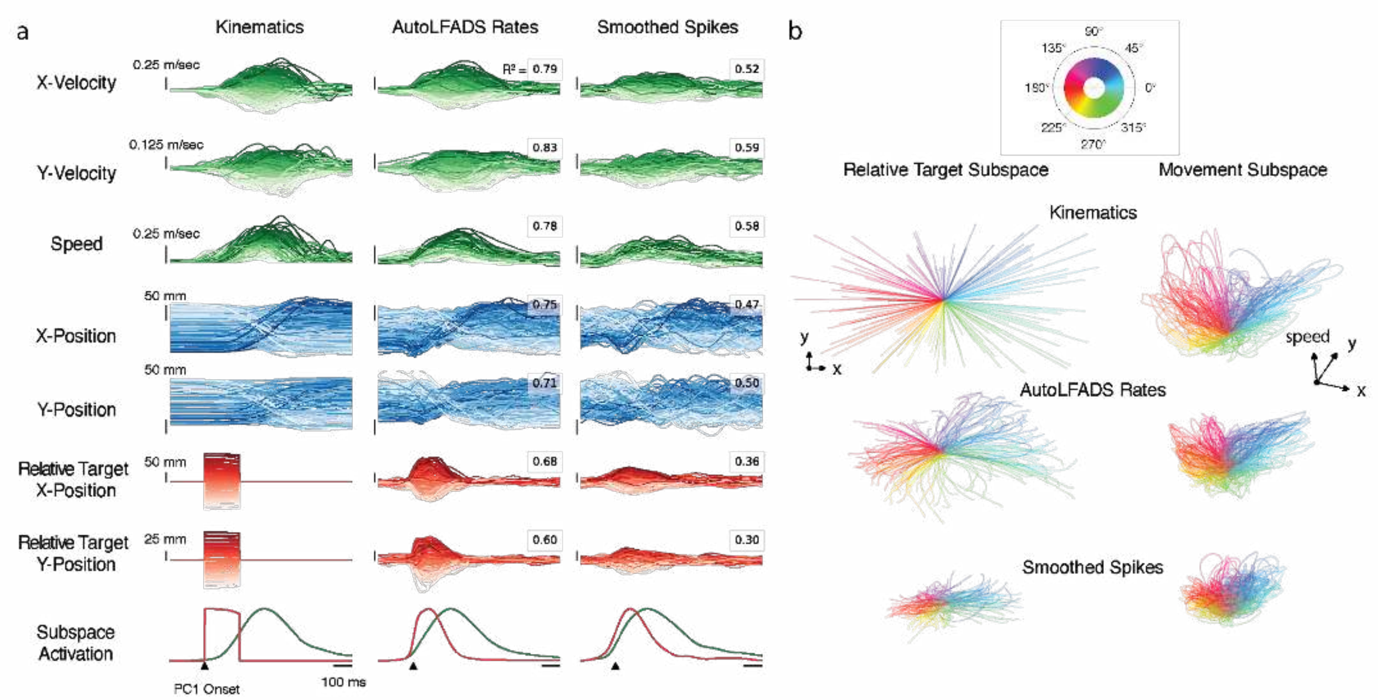

Fig4 | Inferred firing rates contain neural subspaces that are informative about movement kinematics and reach targets. (a) Kinematic and relative target variables and their corresponding neural representations, uncovered via linear regression. The quality of each projection is quantified by accuracy in decoding kinematic and target variables (R^2). Plots are colored by x and y distance to target, except for speed which is colored by peak speed. Bottom row represents the normalized activation of movement (green) and relative target (red) subspaces, illustrating the more transient activation in the target subspace. (b) Movement and relative target subspaces plotted as 3D trajectories and colored by angle to target.

In support of the hypothesis that AutoLFADS is picking up on meaningful dynamics that occurred throughout the session, we found that the firing rates inferred by AutoLFADS were informative of the previously-hypothesized computational role of motor cortical dynamics - i.e., linking the process of movement preparation and execution - despite the model being trained without information about the monkey's behavior (**Fig. 4**). In particular, firing rates contained subspaces that were highly informative about hand position, hand velocity, and reach target on individual trials (**Fig. 4a**) and showed clear structure relative to the task (**Fig. 4b**). To find the subspaces, we used linear regression to project neural activity onto variables related to movement goals (reach target) and movement details (position, velocity and speed). Notably, the subspace reflecting reach target was transiently active around the time of movement execution, consistent with previous studies that have demonstrated the presence of preparatory activity in motor cortex, yet revealed without an explicit preparatory period. It is likely that the rates inferred by AutoLFADS also contain yet undiscovered subspaces and representations that can be explored in this same dataset without experiments explicitly designed to reveal them. Thus, AutoLFADS has the potential to greatly improve the utility and versatility of rich behavioral datasets via a unique unsupervised modeling process.

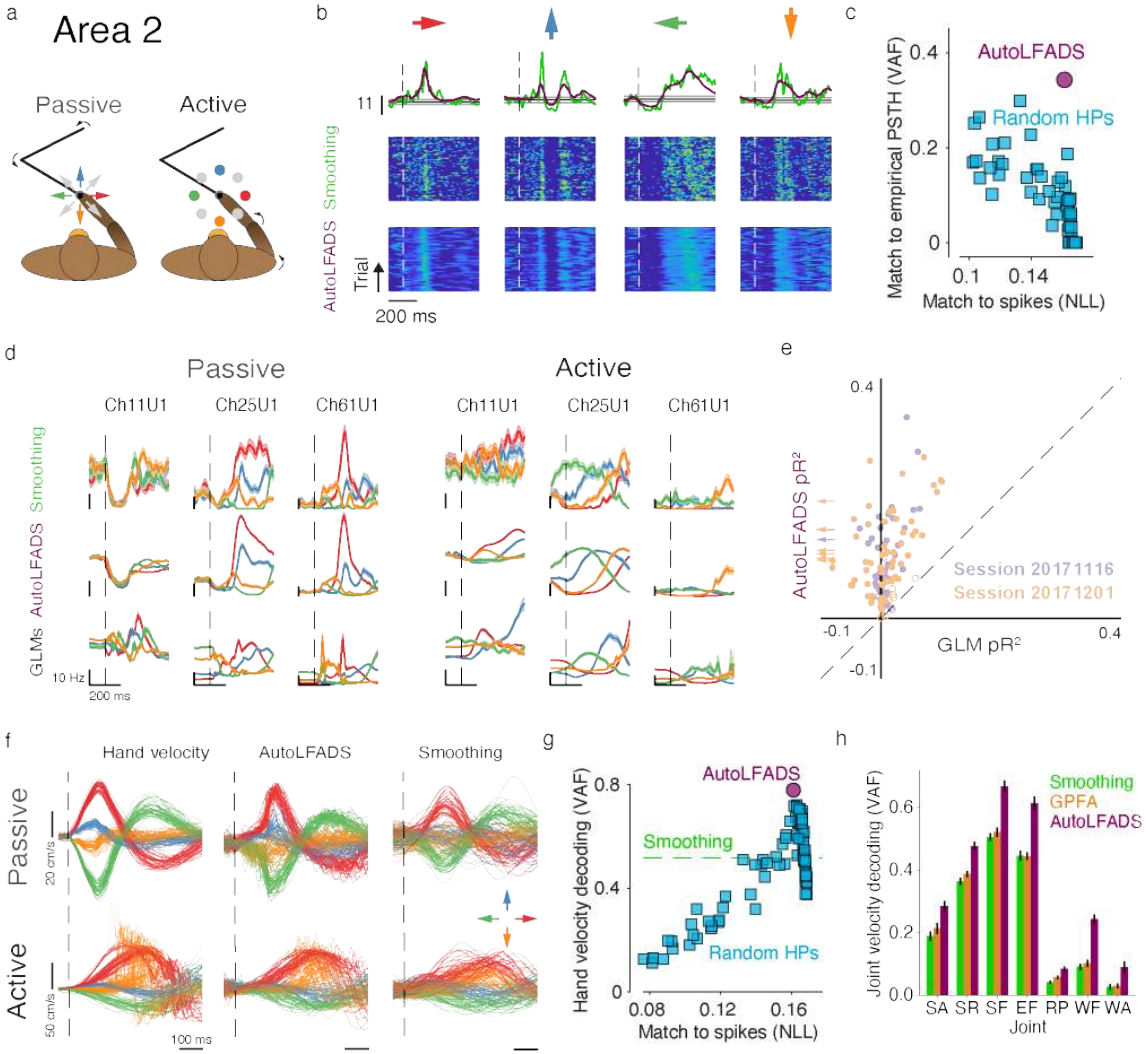

Fig5 | Application of AutoLFADS to data from somatosensory cortex area 2. (a) Schematic of the center-out, bump task showing passive and active conditions. (b) PSTHs and single-trial firing rates for a single neuron across 4 passive perturbation directions. Smoothing was performed using a Gaussian kernel with 10 ms s.d.. Dashed lines indicate movement onset. (c) Comparison of AutoLFADS vs. random search in matching empirical PSTHs. (d) PSTHs produced by smoothing spikes (top), AutoLFADS (middle), or GLM predictions (bottom) for 3 example neurons. (e) Comparison of spike count predictive performance for AutoLFADS and GLMs. Filled circles correspond to neurons for which AutoLFADS pR^2 was significantly higher than GLM pR^2, and open circles correspond to neurons for which there was no significant difference. Arrows (left) indicate neurons for which GLM pR^2 was outside of the plot bounds. (f) Subspace representations of hand x-velocity during active and passive movements extracted from smoothed spikes and rates inferred by AutoLFADS. (g) Comparison of AutoLFADS vs. random search in decoding hand velocity during active trials. (h) Joint angular velocity decoding performance from firing rates inferred using smoothing, Gaussian process factor analysis (GPFA), and AutoLFADS. Error bars denote standard error of the mean. Joint abbreviations: shoulder adduction (SA), shoulder rotation (SR), shoulder flexion (SF), elbow flexion (EF), wrist radial pronation (RP), wrist flexion (WF), and wrist adduction (WA).

AutoLFADS accurately captures single-trial population dynamics in somatosensory cortex

Results from the motor cortical datasets demonstrated that AutoLFADS could produce accurate dynamical models that were robust to training dataset size and generalized well across task conditions, without requiring highly constrained tasks or repeated trials. We next investigated whether AutoLFADS, without manual adjustment, could accurately model dynamics associated with sensory processes. Specifically, we modeled activity in somatosensory area 2 during a reaching task with mechanical perturbation.

Area 2 provides a valuable test case for AutoLFADS generalization. As a sensory area, area 2 receives strong afferent input from cutaneous receptors and muscles and is robustly driven by mechanical perturbations to the arm (37–39). Functionally, area 2 is thought to serve a role in mediating reach-related proprioception (38–41), was recently shown to contain information about whole-arm kinematics (39), and may also receive efferent input from motor areas (38,39,42,43).

In the area 2 experiment (**Fig. 5a**), a monkey used a manipulandum to control a cursor. The task began with a center-hold period where the monkey held the cursor in the center of the screen. During half of the center-hold attempts, the manipulandum randomly perturbed the monkey's arm in one of the eight directions, and the monkey had to re-acquire the central target (passive movement trials). Following the center-hold, the monkey moved to acquire one of eight peripheral targets (active movement trials). The single-trial rates inferred by AutoLFADS for passive trials exhibited clear and structured responses to the unpredictable perturbations (**Fig. 5b**), highlighting the model's ability to approximate input-driven dynamics.

As for M1/PMd, we verified that the rates inferred by AutoLFADS accurately reproduced empirical PSTHs and were informative of task variables. The inferred rates captured the distinct features of PSTHs during active and passive trials, even though no behavioral or task information was provided to the model (**Fig. 5b**; top, and **Fig. 5c**). The rates inferred by AutoLFADS also had a much closer correspondence to the empirical PSTHs during passive trials than LFADS models trained with random HPs (**Fig. 5c**). However, sensory brain regions like area 2 are typically characterized in terms of how neural activity encodes sensory stimuli (37–39). Thus, we examine whether rates inferred by AutoLFADS explain observed spikes better than a typical area 2 neural encoding model, in which neural activity is fit to some function of the state of the arm. We fit a generalized linear model (GLM) for each neuron over both active and passive movements, where the firing rate was solely a function of the position and velocity of the hand, as well as the contact forces with the manipulandum handle (39) (GLM predictions shown in **Fig. 5d**). We then compared the ability of the GLM and AutoLFADS to capture each neuron's observed response using pseudo-R^2 (pR^2), a metric similar to R^2 but adapted for the Poisson statistics of neural firing (44). For the vast majority of neurons across two datasets, AutoLFADS predicted the observed activity significantly better than GLMs ($p<0.05$ for 110/121 neurons, bootstrap; see *Methods*), and there were no neurons for which the GLM produced better predictions than AutoLFADS (**Fig. 5e**).

We used linear decoding to extract subspaces of neural activity that corresponded to x and y hand velocities for both smoothed spikes and rates inferred by AutoLFADS (**Fig. 5f**). The AutoLFADS rates contained subspaces that more clearly separated hand velocities for all active conditions and all passive conditions than smoothing, showing that they are better represented in the modeled dynamics of area 2. Further, single-trial hand velocity decoding from rates inferred by AutoLFADS for active trials was substantially more accurate than that of smoothing, and also more accurate than decoding from the output of any random search model (**Fig. 5g**). On a second dataset that included whole-arm motion tracking, the velocity of all joint angles was decoded from AutoLFADS rates with higher accuracy than from smoothing or GPFA (**Fig. 5h**, right; $p<0.05$ for all joints, paired, one-sided Student's t-Test).

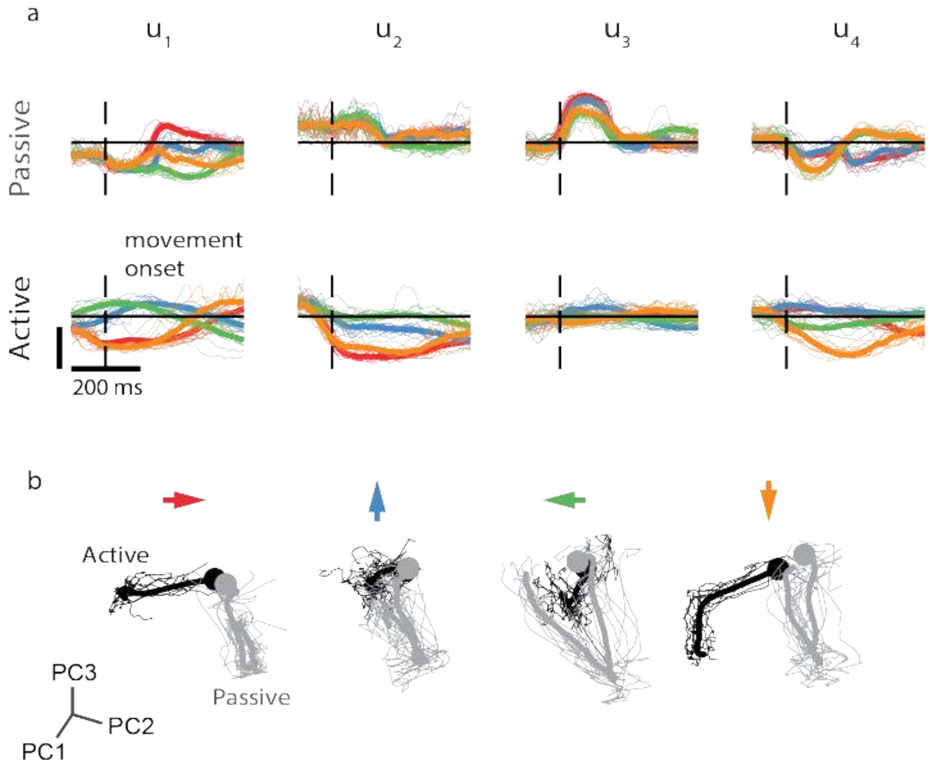

Fig6 | AutoLFADS-inferred inputs for area 2 neural activity. (a) Time-courses of the four inferred generator input dimensions for passive (top) and active (bottom) conditions. Thick line indicates average input trace for each direction, indicated by color, while thin colored lines show input traces for ten randomly chosen trials. Vertical scale bar is A.U. (b) Projection of four-dimensional inputs, from -100 ms to 200 ms around movement onset, into the top three principal components, with separate plots for each movement direction. Darker lines indicate active trials while lighter lines denote passive trials. Large dots indicate average initial input in PC space. Thick and thin lines follow conventions in (a).

Since area 2 plays a significant role in processing sensory inputs, it stands to reason that the inputs inferred by AutoLFADS are important for successfully modeling the area's activity as a dynamical system. If AutoLFADS is successfully modeling area 2 as an input-driven dynamical system, we should expect the inferred inputs to be consistent across trials with the same behavioral conditions. In these experiments, AutoLFADS models the data as fixed-length segments without regard to trial boundaries, so there is no guarantee of the consistency of the meaning of a given input between different trials of the same condition or even within a single trial.

Despite the unsupervised modeling process, AutoLFADS inferred input trajectories that were consistent with the supervised notions of trials, directions, and perturbation types (**Fig. 6a**). Inputs were continuous over the course of a trial, implying that the model was able to pick up on statistical similarities between adjacent segments. The model also produced similar input patterns within a given condition, showing that it was able to detect the statistical patterns of a given condition from arbitrary segments of time during arbitrary trials. Finally, AutoLFADS produced distinct and logically consistent output patterns for active and passive trials. Inputs for abrupt passive movements generally had a much shorter time course that unfolded post-perturbation, while inputs for active trials began before movement and evolved more slowly. Visualization of these inputs highlights AutoLFADS's ability to infer distinct inputs for distinct subsets of the data (**Fig. 6b**).

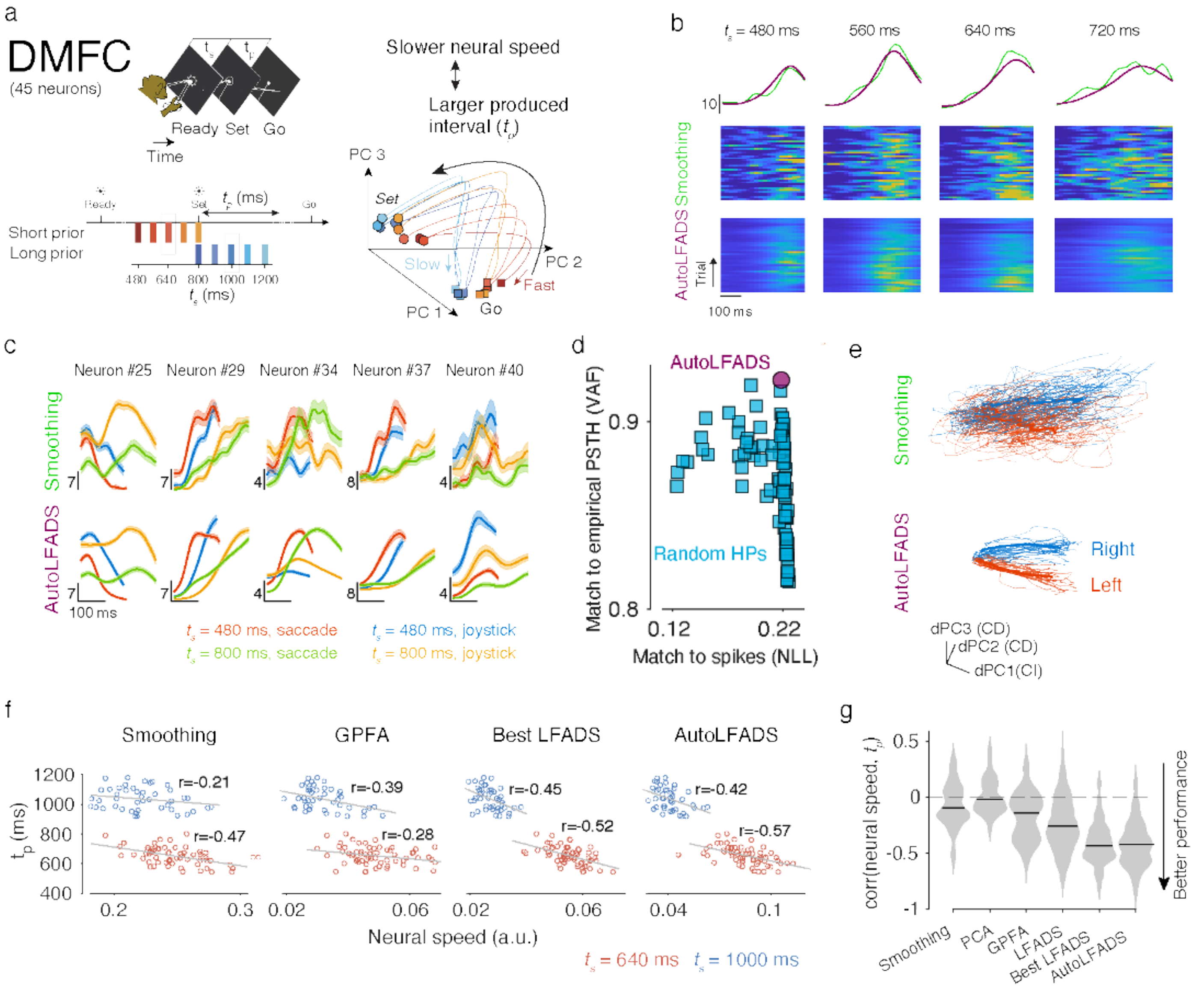

Fig7 | Application of AutoLFADS to data from dorsomedial frontal cortex (DMFC). (a) Top left: the time interval reproduction task. Bottom left: timing conditions used. Right: schematic illustrating the inverse correlation between neural speed and monkey's produced time (t_p). (b) PSTHs and single-trial firing rates for an example neuron during the Set-Go period of leftward saccade trials across 4 different values of t_s (vertical scale bar: spikes/sec). Smoothing was performed using a Gaussian kernel with 25 ms s.d.. (c) PSTHs for 5 example neurons during the Set-Go period of rightward trials for two response modalities and two values of t_s. (d) Performance in replicating the empirical PSTHs. (e) Visualization of low-dimensional trial-averaged and single-trial neural trajectories for the Ready-Set period for left and right joystick trials with t_s of 1000 ms. 30 trials are shown for each condition. dPC: demixed principal component, CI: condition-independent, CD: condition-dependent. (f) Example plots showing correlations between neural speed and behavior (i.e., production time, t_p) for individual trials across two timing intervals (red: 640 ms blue: 1000 ms). Neural speed was obtained based on the firing rates inferred from smoothing, GPFA, the LFADS model with best median speed-t_p correlation across the 40 different task conditions (Best LFADS), and AutoLFADS. (g) Distributions of correlation coefficients across 40 different task conditions. Horizontal lines denote medians. For LFADS, the distribution includes correlation values for all 96 models with random HPs (40x96 values).

AutoLFADS accurately captures single-trial dynamics during cognition

While activity in M1 and area 2 are largely driven by internal dynamics and inputs, respectively, many brain areas depend critically on the confluence of internal dynamics and inputs. To further test the generality of AutoLFADS to these situations, we applied it to data collected from dorsomedial frontal cortex (DMFC) during a cognitive time estimation task. DMFC comprises the supplementary eye field, dorsal supplementary motor area, and presupplementary motor area. It is often considered an intermediate region in the sensorimotor hierarchy (45), interfacing with both low-level sensory and motor (PMd/M1) areas. DMFC activity is less closely tied to the moment-by-moment details of movements than activity in M1 or area 2 - instead, its activity seems to relate to higher-level aspects of motor control, including motor timing (46,47), planning movement sequences (48), learning sensorimotor associations (49) and context-dependent reward modulation (50). However, population dynamics in DMFC are tied to behavioral correlates such as movement production time (15,47,51). This makes DMFC another excellent test case for unsupervised modeling with AutoLFADS.

For this task, the monkey was presented with two visual stimuli ("Ready" and "Set", respectively), separated by sample timing interval t_s. After "Set", the monkey attempted to reproduce the interval by waiting for the same amount of time (t_p) before initiating a movement ("Go") (**Fig. 7a**, left). The movement was either a saccade or joystick manipulation to the left or right depending on the location of a peripheral target. The two response modalities, combined with 10 timing conditions (t_s) and two target locations, led to a total of 40 task conditions.

Consistent with our observations on M1/PMd and area 2 data, AutoLFADS-inferred rates for this dataset showed consistent, denoised structure at the single-trial level (**Fig. 7b**, bottom) and recapitulated the features of neural responses uncovered by trial averaging (**Fig. 7b,** top; **Fig. 7c**). Quantitative comparison of the PSTHs shows that AutoLFADS-inferred rates again achieved a better match to the empirical PSTHs than all of the random search models (**Fig. 7d**), providing further evidence that AutoLFADS can achieve superior models without expert tuning of regularization HPs or supervised model selection criteria. Additionally, when visualized in a low-dimensional space using demixed principal components analysis (dPCA), the AutoLFADS-inferred firing rates showed much greater consistency across trials of a given condition than firing rates computed by smoothing spikes (**Fig. 7e**).

To evaluate the AutoLFADS model beyond its ability to capture trial-averaged responses, we sought to evaluate whether its predicted firing rates were more informative of trial-by-trial timing behaviors than other methods. Previous studies have shown that the monkey's produced time interval (t_p) is negatively correlated to the speed at which the neural trajectories evolve during the Set-Go period (**Fig. 7a**, right) (15,51). To evaluate the correspondence between neural activity and behavior, we estimated neural speeds using representations produced by smoothing spikes, GPFA, principal component analysis (PCA), the best random search model ('Best LFADS', see *Methods* for details), and an AutoLFADS model, and measured the trial-by-trial correlation between the estimated speeds and t_p. Note that selecting the best random search model again required a supervised calculation (t_p correlation) for each model. If a given representation of neural activity is more informative about behavior, we expect a stronger (more negative) correlation between predicted and observed $t_{p.}$

We show correlation values for individual trials across two different values of t_s (**Fig. 7f**), and summarize across all 40 task conditions (**Fig. 7g**). We observed consistent negative correlations between t_p and the estimated neural speed from rates obtained by different methods. Correlations from rates inferred by AutoLFADS were significantly better than all unsupervised approaches (p<0.001, Wilcoxon signed rank test), and comparable with the supervised selection approach ('Best LFADS', p=0.758, Wilcoxon signed rank test), despite using no task information.

Taken together, the area 2 and DMFC results demonstrate that the out-of-the-box, automated inference of neural population dynamics provided by AutoLFADS allows modeling of diverse brain areas, with dynamics that span the continuum from autonomous to input-driven. AutoLFADS provides a powerful framework for generalized inference of input-driven dynamics and enables decoding of simultaneously monitored behavioral variables with unprecedented accuracy. Importantly, the unsupervised approach of AutoLFADS avoids the use of any behavioral data and optimizes only for neural modeling. This allows for modeling when behavioral data is not available and also prevents any behavioral biases from being introduced to the firing rates, resulting in better inference of the brain's inherently generalized representations. This is evident in the high performance of AutoLFADS rates in both PSTH reconstruction and various decoding tasks.

Running AutoLFADS in the Cloud

A key challenge with emerging, computationally-intensive data analysis methods is that the computational infrastructure and expertise necessary to make effective use of these tools is a significant barrier to widespread adoption (52). For example, many labs do not have the resources necessary to train dozens of models in parallel across many GPUs. To address this hurdle, we provide an open-source implementation of AutoLFADS designed to operate on Google Cloud Platform (GCP). Additionally, we provide a comprehensive tutorial to help novice users get started running AutoLFADS on GCP without expert knowledge of cloud computing or machine learning. The tutorial describes how to set up the framework, prepare input data, set up AutoLFADS runs, and load the final results. Users of AutoLFADS on GCP don't need to worry about the upfront hardware and labor costs associated with maintaining a local computing cluster, yet have access to virtually unlimited computation on demand. This framework allows researchers to spend less time doing non-research tasks like dependency management and hyperparameter optimization, while giving them confidence that their models are performing well, regardless of brain area or task. We include links to the code and tutorial in *Code Availability*.

Discussion

The original LFADS work (20) provided a method for inferring latent dynamics, denoised firing rates, and external inputs from large populations of neurons, producing representations that were more informative of behavior than previous approaches (33). However, application of LFADS to neural populations with different dynamics, strong external inputs, or unconstrained behavior would have necessitated time-consuming and subjective manual tuning. In the current work, we show that with robust regularization and efficient hyperparameter tuning it is possible to train high-performing LFADS models for neural spiking datasets with arbitrary size, trial structure, and dynamical complexity. We demonstrated several properties of the AutoLFADS training approach which have broad implications. On the maze task, we showed that AutoLFADS models are more robust to dataset size, opening up new lines of inquiry on smaller datasets and reducing the number of trials that must be conducted in future experiments. Using the random target task, we demonstrated how AutoLFADS needs no task information in order to generate rich dynamical models of neural activity. This enables the study of dynamics during richer tasks and reuse of datasets collected for another purpose. With the perturbed reaching task, we demonstrated the first application of dynamical modeling, as opposed to encoder-based modeling, to the highly input-driven somatosensory area 2. Finally, in the timing task, we showed that AutoLFADS found the appropriate balance between inputs and internal dynamics for a cognitive area by modeling DMFC.

AutoLFADS inherits some of the flaws of the LFADS model. For example, the linear-exponential-Poisson observation model is likely an oversimplification. However, we used this architecture as a starting point to show that a large-scale hyperparameter search is feasible and beneficial. By enabling large-scale searches, we can be reasonably confident that any performance differences achieved by future architecture changes will be due to real differences in modeling capabilities rather than a simple lack of HP optimization.

AutoLFADS performed well using a simple binary tournament exploitation and perturbation exploration strategies for PBT (25). Future work might investigate alternate exploitation or exploration strategies, or whether more powerful and efficient PBT variants (53) can increase speed and performance of AutoLFADS while lowering computational cost. A current limitation of AutoLFADS is its inability to explore hyperparameters that modify the underlying model architecture. Thus, another avenue for further work lies in combining AutoLFADS with the recent techniques for automated neural architecture search (54).

Though AutoLFADS is much more efficient than previous approaches, it still requires substantial computational resources that may not be available for all potential users. Setting up the requisite software environments can be an additional hurdle. Our GCP implementation allows users to apply AutoLFADS without needing to purchase and maintain a local cluster. We estimate that the compute cost for a typical AutoLFADS run on GCP is between $5-25, depending on dataset and model sizes. We have created detailed tutorials to guide novice users through the setup, model training, and data retrieval processes, making AutoLFADS accessible to anyone who works with neural spiking data.

Taken together, AutoLFADS provides an accessible and extensible framework for generalized inference of single-trial neural dynamics that has the potential to unify the way we study computation through dynamics across brain areas and tasks.

Code Availability

AutoLFADS for GCP can be downloaded from GitHub at `github.com/snel-repo/autolfads` and the tutorial is available at `snel-repo.github.io/autolfads`.

Data Availability

Data will be made available upon reasonable request from the authors. The random target dataset is publicly available at `http://doi.org/10.5281/zenodo.3854034`.

Acknowledgements

We thank K. Shenoy, M. Churchland, M. Kaufman, and S. Ryu for sharing the Monkey J Maze dataset. We also thank J. O'Doherty, M. Cardoso, J. Makin, and P. Sabes for making the random target dataset publicly available. This work was supported by the Emory Neuromodulation and Technology Innovation Center (ENTICe), NSF NCS 1835364, DARPA PA-18-02-04-INI-FP-021, NIH Eunice Kennedy Shriver NICHD K12HD073945, the Alfred P. Sloan Foundation, the Burroughs Wellcome Fund, and the Simons Foundation as part of the Simons-Emory International Consortium on Motor Control (CP), NIH NINDS R01 NS053603, R01 NS095251, and NSF NCS 1835345 (LEM), NSF Graduate Research Fellowships DGE-1650044 (ARS) and DGE-1324585 (RHC), the Center for Sensorimotor Neural Engineering and NARSAD Young Investigator grant from the Brain & Behavior Research Foundation (HS), NIH NINDS NS078127, the Sloan Foundation, the Klingenstein Foundation, the Simons Foundation, the McKnight Foundation, the Center for Sensorimotor Neural Engineering, and the McGovern Institute (MJ).

Author Contributions

	MRK	ARS	RHC	RT	DB	SLN	HS	MJ	LEM	CP
Conceptualization	■	■								■
Funding acquisition									■	■
Investigation	■	■	■				■			■
Resources			■				■	■	■	■
Software	■	■	■	■		■				■
Tutorials				■	■					
Writing	■	■								■
Revision	■	■	■				■	■	■	■

Competing Interests

The authors declare no competing interests.

References

1. Jun JJ, Steinmetz NA, Siegle JH, Denman DJ, Bauza M, Barbarits B, et al. Fully integrated silicon probes for high-density recording of neural activity. Nature. 2017 Nov;551(7679):232–6.

2. Stevenson IH, Kording KP. How advances in neural recording affect data analysis. Nat Neurosci. 2011 Feb;14(2):139–42.
3. Stringer C, Pachitariu M, Steinmetz N, Carandini M, Harris KD. High-dimensional geometry of population responses in visual cortex. Nature. 2019 Jul;571(7765):361–5.
4. Steinmetz NA, Aydin C, Lebedeva A, Okun M, Pachitariu M, Bauza M, et al. Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings. bioRxiv. 2020;
5. Berger M, Agha NS, Gail A. Wireless recording from unrestrained monkeys reveals motor goal encoding beyond immediate reach in frontoparietal cortex. Elife. 2020;9:e51322.
6. Shenoy KV, Sahani M, Churchland MM. Cortical Control of Arm Movements: A Dynamical Systems Perspective. Annu Rev Neurosci. 2013 Jul 8;36(1):337–59.
7. Pandarinath C, Ames KC, Russo AA, Farshchian A, Miller LE, Dyer EL, et al. Latent Factors and Dynamics in Motor Cortex and Their Application to Brain–Machine Interfaces. J Neurosci. 2018 Oct 31;38(44):9390–401.
8. Vyas S, Golub MD, Sussillo D, Shenoy KV. Computation Through Neural Population Dynamics. Annual Review of Neuroscience. 2020;43(1):249–75.
9. Carnevale F, de Lafuente V, Romo R, Barak O, Parga N. Dynamic Control of Response Criterion in Premotor Cortex during Perceptual Detection under Temporal Uncertainty. Neuron. 2015 May;86(4):1067–77.
10. Churchland MM, Cunningham JP, Kaufman MT, Foster JD, Nuyujukian P, Ryu SI, et al. Neural population dynamics during reaching. Nature. 2012 Jul;487(7405):51–6.
11. Harvey CD, Coen P, Tank DW. Choice-specific sequences in parietal cortex during a virtual-navigation decision task. Nature. 2012 Apr;484(7392):62–8.
12. Kobak D, Brendel W, Constantinidis C, Feierstein CE, Kepecs A, Mainen ZF, et al. Demixed principal component analysis of neural population data. eLife. 2016 Apr 12;5:e10989.
13. Mante V, Sussillo D, Shenoy KV, Newsome WT. Context-dependent computation by recurrent dynamics in prefrontal cortex. Nature. 2013 Nov;503(7474):78–84.
14. Pandarinath C, Gilja V, Blabe CH, Nuyujukian P, Sarma AA, Sorice BL, et al. Neural population dynamics in human motor cortex during movements in people with ALS. eLife. 2015 Jun 23;4:e07436.
15. Remington ED, Narain D, Hosseini EA, Jazayeri M. Flexible Sensorimotor Computations through Rapid Reconfiguration of Cortical Dynamics. Neuron. 2018 Jun;98(5):1005-1019.e5.
16. Cunningham JP, Yu BM. Dimensionality reduction for large-scale neural recordings. Nat Neurosci. 2014 Nov;17(11):1500–9.
17. Gao Y, Archer E, Paninski L, Cunningham JP. Linear dynamical neural population models through nonlinear embeddings. arXiv:160508454 [q-bio, stat] [Internet]. 2016 Oct 25 [cited 2020 Aug 13]; Available from: http://arxiv.org/abs/1605.08454
18. Hernandez D, Moretti AK, Wei Z, Saxena S, Cunningham J, Paninski L. A novel variational family for hidden nonlinear markov models. arXiv preprint arXiv:181102459. 2018;
19. Koppe G, Toutounji H, Kirsch P, Lis S, Durstewitz D. Identifying nonlinear dynamical systems via generative recurrent neural networks with applications to fMRI. PLoS computational biology. 2019;15(8):e1007263.
20. Pandarinath C, O'Shea DJ, Collins J, Jozefowicz R, Stavisky SD, Kao JC, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. Nat Methods. 2018 Oct;15(10):805–15.
21. She Q, Wu A. Neural dynamics discovery via gaussian process recurrent neural networks. arXiv preprint arXiv:190700650. 2019;
22. Sussillo D, Jozefowicz R, Abbott L, Pandarinath C. LFADS-latent factor analysis via dynamical systems. arXiv preprint arXiv:160806315. 2016;
23. González J, Dai Z, Hennig P, Lawrence ND. Batch Bayesian Optimization via Local Penalization. arXiv:150508052 [stat] [Internet]. 2015 Oct 14 [cited 2020 Sep 11]; Available from: http://arxiv.org/abs/1505.08052
24. Li L, Jamieson K, Rostamizadeh A, Gonina E, Hardt M, Recht B, et al. A System for Massively Parallel Hyperparameter Tuning. arXiv:181005934 [cs, stat] [Internet]. 2020 Mar 15 [cited 2020 Sep 11]; Available from: http://arxiv.org/abs/1810.05934
25. Jaderberg M, Dalibard V, Osindero S, Czarnecki WM, Donahue J, Razavi A, et al. Population based training of neural networks. arXiv preprint arXiv:171109846. 2017;

26. Keshtkaran MR, Pandarinath C. Enabling hyperparameter optimization in sequential autoencoders for spiking neural data. In: Advances in Neural Information Processing Systems. 2019. p. 15937–47.
27. Chung J, Kastner K, Dinh L, Goel K, Courville AC, Bengio Y. A Recurrent Latent Variable Model for Sequential Data. In: Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R, editors. Advances in Neural Information Processing Systems 28 [Internet]. Curran Associates, Inc.; 2015 [cited 2020 Oct 2]. p. 2980–8. Available from: http://papers.nips.cc/paper/5653-a-recurrent-latent-variable-model-for-sequential-data.pdf
28. Gregor K, Danihelka I, Graves A, Rezende DJ, Wierstra D. DRAW: A Recurrent Neural Network For Image Generation. arXiv:150204623 [cs] [Internet]. 2015 May 20 [cited 2020 Oct 2]; Available from: http://arxiv.org/abs/1502.04623
29. Jaderberg M, Czarnecki WM, Dunning I, Marris L, Lever G, Castaneda AG, et al. Human-level performance in 3D multiplayer games with population-based reinforcement learning. Science. 2019;364(6443):859–65.
30. Gao P, Trautmann E, Yu B, Santhanam G, Ryu S, Shenoy K, et al. A theory of multineuronal dimensionality, dynamics and measurement. BioRxiv. 2017;214262.
31. O'Doherty JE, Cardoso MMB, Makin JG, Sabes PN. Nonhuman Primate Reaching with Multichannel Sensorimotor Cortex Electrophysiology [Internet]. Zenodo; 2020 [cited 2020 Aug 21]. Available from: https://zenodo.org/record/3854034#.Xz_iqpNKhuU
32. Williams AH, Kim TH, Wang F, Vyas S, Ryu SI, Shenoy KV, et al. Unsupervised Discovery of Demixed, Low-Dimensional Neural Dynamics across Multiple Timescales through Tensor Component Analysis. Neuron. 2018 Jun;98(6):1099-1115.e8.
33. Yu BM, Cunningham JP, Santhanam G, Ryu SI, Shenoy KV, Sahani M. Gaussian-Process Factor Analysis for Low-Dimensional Single-Trial Analysis of Neural Population Activity. Journal of Neurophysiology. 2009 Jul;102(1):614–35.
34. Macke JH, Buesing L, Cunningham JP, Yu BM, Shenoy KV, Sahani M. Empirical models of spiking in neural populations. In: Advances in Neural Information Processing Systems. 2011. p. 9.
35. Petreska B, Yu BM, Cunningham JP, Santhanam G, Ryu SI, Shenoy KV, et al. Dynamical segmentation of single trials from population neural data. In: Advances in Neural Information Processing Systems. 2011. p. 9.
36. Buesing L, Macke JH, Sahani M. Spectral learning of linear dynamics from generalised-linear observations with application to neural population data. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ, editors. Advances in Neural Information Processing Systems 25 [Internet]. Curran Associates, Inc.; 2012 [cited 2020 Aug 26]. p. 1682–90. Available from: http://papers.nips.cc/paper/4836-spectral-learning-of-linear-dynamics-from-generalised-linear-observations-with-application-to-neural-population-data.pdf
37. Prud'homme MJ, Kalaska JF. Proprioceptive activity in primate primary somatosensory cortex during active arm reaching movements. J Neurophysiol. 1994 Nov;72(5):2280–301.
38. London BM, Miller LE. Responses of somatosensory area 2 neurons to actively and passively generated limb movements. J Neurophysiol. 2013 Mar;109(6):1505–13.
39. Chowdhury RH, Glaser JI, Miller LE. Area 2 of primary somatosensory cortex encodes kinematics of the whole arm. Makin TR, Gold JI, Makin TR, editors. eLife. 2020 Jan 23;9:e48198.
40. Kaas JH, Nelson RJ, Sur M, Lin CS, Merzenich MM. Multiple representations of the body within the primary somatosensory cortex of primates. Science. 1979 May 4;204(4392):521–3.
41. Jennings VA, Lamour Y, Solis H, Fromm C. Somatosensory cortex activity related to position and force. Journal of Neurophysiology. 1983 May 1;49(5):1216–29.
42. Nelson RJ. Activity of monkey primary somatosensory cortical neurons changes prior to active movement. Brain Res. 1987 Mar 17;406(1–2):402–7.
43. Padberg J, Cooke DF, Cerkevich CM, Kaas JH, Krubitzer L. Cortical connections of area 2 and posterior parietal area 5 in macaque monkeys. J Comp Neurol. 2019 15;527(3):718–37.
44. Cameron AC, Windmeijer FAG. R-Squared Measures for Count Data Regression Models with Applications to Health-Care Utilization. Journal of Business & Economic Statistics. 1996;14(2):209–20.
45. Felleman DJ, Van Essen DC. Distributed hierarchical processing in the primate cerebral cortex. Cereb Cortex. 1991 Feb;1(1):1–47.
46. Mita A, Mushiake H, Shima K, Matsuzaka Y, Tanji J. Interval time coding by neurons in the

presupplementary and supplementary motor areas. Nat Neurosci. 2009 Apr;12(4):502–7.
47. Wang J, Narain D, Hosseini EA, Jazayeri M. Flexible timing by temporal scaling of cortical responses. Nat Neurosci. 2018;21(1):102–10.
48. Lu X, Matsuzawa M, Hikosaka O. A neural correlate of oculomotor sequences in supplementary eye field. Neuron. 2002 Apr 11;34(2):317–25.
49. Chen LL, Wise SP. Neuronal activity in the supplementary eye field during acquisition of conditional oculomotor associations. J Neurophysiol. 1995 Mar;73(3):1101–21.
50. Stuphorn V. The role of supplementary eye field in goal-directed behavior. J Physiol Paris. 2015 Jun;109(1–3):118–28.
51. Sohn H, Narain D, Meirhaeghe N, Jazayeri M. Bayesian Computation through Cortical Latent Dynamics. Neuron. 2019 Sep 4;103(5):934-947.e5.
52. Abe T, Kinsella I, Saxena S, Paninski L, Cunningham JP. Neuroscience Cloud Analysis As a Service. bioRxiv. 2020 Jun 12;2020.06.11.146746.
53. Li A, Spyra O, Perel S, Dalibard V, Jaderberg M, Gu C, et al. A Generalized Framework for Population Based Training. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining [Internet]. Anchorage AK USA: ACM; 2019 [cited 2020 Dec 17]. p. 1791–9. Available from: https://dl.acm.org/doi/10.1145/3292500.3330649
54. Elsken T, Metzen JH, Hutter F. Neural Architecture Search: A Survey. arXiv:180805377 [cs, stat] [Internet]. 2019 Apr 26 [cited 2020 Dec 17]; Available from: http://arxiv.org/abs/1808.05377
55. Kaufman MT, Seely JS, Sussillo D, Ryu SI, Shenoy KV, Churchland MM. The Largest Response Component in the Motor Cortex Reflects Movement Timing but Not Movement Type. eNeuro [Internet]. 2016 Jul 1 [cited 2020 Dec 28];3(4). Available from: https://www.eneuro.org/content/3/4/ENEURO.0085-16.2016
56. Perich MG, Gallego JA, Miller LE. A Neural Population Mechanism for Rapid Learning. Neuron. 2018 Nov 21;100(4):964-976.e7.
57. Sohn H, Narain D, Meirhaeghe N, Jazayeri M. Bayesian computation through cortical latent dynamics. bioRxiv. 2018 Nov 8;465419.

Methods

LFADS architecture and training

A detailed overview of the LFADS model is given in (20). Briefly: at the input to the model, a pair of bidirectional RNN encoders read over the spike sequence and produce initial conditions for the generator RNN and time-varying inputs for the controller RNN. All RNNs were implemented using gated recurrent unit (GRU) cells. At each time step, the generator state evolves with input from the controller and the controller receives delayed feedback from the generator. The generator states are linearly mapped to factors, which are mapped to the firing rates of the original neurons using a linear mapping followed by an exponential. The optimization objective is to minimize the negative log-likelihood of the data given the inferred firing rates, and includes KL and L2 regularization penalties.

Identical architecture and training hyperparameter values were used for most runs, with a few deviations. We used a generator dimension of 100, initial condition dimension of 100 (50 for area 2 runs), initial condition encoder dimension of 100, factor dimension of 40, controller and controller input encoder dimension of 80 (64 for DMFC runs), and controller output dimension of 4 (10 for overfitting runs).

We used the Adam optimizer with an initial learning rate of 0.01 and, for non-AutoLFADS runs, decayed the learning rate by a factor of 0.95 after every 6 consecutive epochs with no improvement to the validation loss. Training was halted for these runs when the learning rate reached 1e-5. The loss was scaled by a factor of 1e4 immediately before optimization for numerical stability. GRU cell hidden states were clipped at 5 and the global gradient norm was clipped at 200 to avoid occasional pathological training.

We used a trainable mean initialized to 0 and fixed variance of 0.1 for the Gaussian initial condition prior and set a minimum allowable variance of 1e-4 for the initial condition posterior. The controller output prior was autoregressive with a trainable autocorrelation tau and noise variance, initialized to 10 and 0.1, respectively.

Memory usage for RNNs is highly dependent on the sequence length, so batch size was varied accordingly (100 for maze and random target datasets, 500 for synthetic and area 2 datasets, and 300/400 for the DMFC dataset). KL and L2 regularization penalties were linearly ramped to their full weight during the first 80 epochs for most runs to avoid local minima induced by high initial regularization penalties. Exceptions were the runs on synthetic data, which were ramped over 70 epochs and random searches on area 2 and DMFC datasets, which used step-wise ramping over the first 400 steps.

Random searches and AutoLFADS runs used the architecture parameters described above, along with regularization HPs sampled from ranges (or initialized with constant values) given in **Supp. Table 2**. Most runs used a default set of ranges, with a few exceptions outlined in the table. Dropout was sampled from a uniform distribution and KL and L2 weight HPs were sampled from log-uniform distributions.

During PBT, weights were used to control maximum and minimum perturbation magnitudes for different HPs (e.g. a weight of 0.3 results in perturbation factors between 0.7 and 1.3). The dropout and CD HPs used a weight of 0.3 and KL and L2 penalty HPs used a weight of 0.8. CD rate, dropout rate, and learning rate were limited to their specified ranges, while the KL and L2 penalties could be perturbed outside of the initial ranges. Each generation of PBT consisted of 50 training epochs. AutoLFADS training was stopped when the best smoothed validation NLL improved by less than 0.05% over the course of four generations.

Validation NLL was exponentially smoothed with $\alpha = 0.7$ during training. For non-AutoLFADS runs, the model checkpoint with the lowest smoothed validation NLL was used for inference. For AutoLFADS runs, the checkpoint with the lowest smoothed validation NLL in the last epoch of any generation was used for inference. Firing rates were inferred 50 times for each model using different samples from initial condition and controller output posteriors. These estimates were then averaged, resulting in the final inferred rates for each model.

Overfitting on synthetic data

Synthetic data were generated using a 2-input chaotic vanilla RNN ($\gamma = 1.5$) as described in the original LFADS work (20,22). The only modification was that the inputs were white Gaussian noise. In brief, the 50-unit RNN was run for 1 second (100 time steps) starting from 400 different initial conditions to generate ground-truth Poisson rates for each condition. These distributions were sampled 10 times for each condition, resulting in 4000 spiking trials. Of these trials, 80% (3200 trials) were used for LFADS training and the final 20% (800 trials) were used for validation.

We sampled 200 HP combinations from the distributions specified in **Supp. Table 2** and used them to train LFADS models on the synthetic dataset. We then trained 200 additional models with the same set of HPs using a CD rate of 0.3 (i.e., using 70% of data as input and remaining 30% for likelihood evaluation) (26). The coefficient of determination between inferred and ground truth rates was computed across all samples and neurons on the 800-sample validation set.

M1 maze task

We used the previously-collected maze dataset (55) described in detail in the original LFADS work (20). Briefly, a male macaque monkey performed a two-dimensional center-out reaching task by guiding a cursor to a target without touching any virtual barriers while neural activity was recorded via two 92-electrode arrays implanted into M1 and dorsal PMd. The full dataset consisted of 2,296 trials, 108 reach conditions, and 202 single units.

The spiking data were binned at 1 ms and smoothed by convolution with a Gaussian kernel (30 ms s.d.). Hand velocities were computed using second order accurate central differences from hand position at 1kHz. An antialiasing filter was applied to hand velocities and all data were then resampled to 2 ms. Trials were created by aligning the data to 250 ms before and 450 ms after movement onset, as calculated in the original paper.

Datasets of varying sizes were created for LFADS by randomly selecting trials with 20, 10, and 5% of the original dataset using seven fixed seeds, and then splitting each of these into 80/20 training and validation sets for LFADS (22 total, including the full dataset). As a baseline for each data subset, we trained LFADS models with fixed HPs that had been previously found to result in high-performing models for this dataset, with the exception of controller input encoder and controller dimensionalities (see *LFADS architecture and training* and **Supp. Table 2**). We increased the dimensionality of these components to allow improved generalization to the datasets from more input-driven areas while keeping the architecture consistent across all datasets. We also trained AutoLFADS models (40 workers) on each subset using the search space given in **Supp. Table 2**. Additionally, we ran a random search using 100 HPs sampled from the AutoLFADS search space on one of the 230-trial datasets.

We used rates from spike smoothing, manually tuned LFADS models, random search LFADS models, and AutoLFADS models to predict x and y hand velocity delayed by 90 ms using ridge regression with a regularization penalty of $\lambda = 1$. Each data subset was further split into 80/20 training and validation sets for decoding. To account for the difficulty of modeling the first few time points of each trial with LFADS, we discarded data from the first 50 ms of each trial and did not use that data for model evaluation. Decoding performance was evaluated by computing the coefficient of determination for predicted and true velocity across all trials for each velocity dimension. The result was then averaged across the two velocity dimensions.

To evaluate PSTH reconstruction for random search and AutoLFADS models, we first computed the empirical PSTHs by averaging smoothed spikes from the full 2296-trial dataset across all 108 conditions. We then computed model PSTHs by averaging inferred rates across conditions for all trials in the 230-trial subset. We computed the coefficient of determination between model-inferred PSTHs and empirical PSTHs for each neuron across all conditions in the subset. We then averaged the result across all neurons.

M1 random target task

The random target dataset consists of neural recordings and hand position data recorded from macaque M1 during a self-paced, sequential reaching task between random elements of a grid (31). For our experiments, we used only the first 30% (approx. 9 minutes) of the dataset recorded from Indy on 04/26/2016.

We started with sorted units obtained from M1 and binned their spike times at 1 ms. To avoid artifacts in which the same spikes appeared on multiple channels, we computed cross-correlations between all pairs of neurons over the first 10 sec and removed individual correlated neurons ($n = 34$) by highest firing rate until there were no pairs with correlation above 0.0625, resulting in 181 uncorrelated neurons. The position data were provided at 250 Hz, so we upsampled these data to 1 kHz using cubic interpolation. We smoothed the spikes by convolving with a Gaussian kernel (50 ms s.d.), applied an antialiasing filter to hand velocities, and downsampled to 2 ms. The continuous neural spiking data were chopped into overlapping segments of length 600 ms, where each segment shared its last 200 ms with the first 200 ms of the next. The resulting 1321 segments were split into 80/20 training and validation sets for LFADS, where the validation segments were chosen in blocks of 3 to minimize the overlap between training and validation subsets.

The chopped segments were used to train an AutoLFADS model and to run a random search using 100 HPs sampled from the AutoLFADS search space. After modeling, the chopped data were merged using a quadratic weighting of overlapping regions that placed more weight on the rates inferred at the ends of the segments. The merging technique weighted the ends of segments as $w = 1 - x^2$ and the beginnings of segments as $1 - w$, with x ranging from 0 to 1 across the overlapping points. After weights were applied, overlapping points were summed, resulting in a continuous ~9-minute stretch of modeled data.

We computed hand velocity from position using second-order accurate central differences and introduced a 120 ms delay between neural data and kinematics. We used ridge regression ($\lambda = 1e - 5$) to predict hand velocity across the continuous data using smoothed spikes, random search LFADS rates, and AutoLFADS rates. We computed coefficient of determination for each velocity dimension individually and then averaged the two velocity dimensions to compute decoding performance.

To prepare the data for subspace visualization, the continuous activity for each neuron was soft-normalized by subtracting its mean and dividing by its 90th quantile plus an offset of 0.01. Trials were identified in the continuous data as the intervals over which target positions were constant (314 trials). To identify valid trials, we computed the normalized distance from the final position. Trials were removed if the cursor exceeded 5% of this original distance or overshot by 5%. Thresholds ($n = 100$) were also created between 25 and 95% of the distance and trials were removed if they crossed any of those thresholds more than once. We then computed an alignment point at 90% of the distance from the final position for the remaining trials and labeled it as movement onset (227 trials). For each of these trials, data were aligned to 400 ms before and 500 ms after movement onset. The first principal component of AutoLFADS rates during aligned trials was computed and activation during the first 100 ms of each trial was normalized to $[0,1]$. Trials were rejected if activation peaked after 100 ms or the starting activation was more than 3 standard deviations from the mean. The PC1 onset alignment point was calculated as the first time that activity in the first principal component crossed 50% of its maximum in the first 100 ms (192 trials). This alignment point was used for all neural subspace analyses.

Movement-relevant subspaces were extracted by ridge regression from neural activity onto x-velocity, y-velocity, and speed. Similarly, position-relevant subspaces involved regression from neural activity onto x-position and y-position. For movement and position subspaces, neural and behavioral data were aligned to 200 ms before and 1000 ms after PC1 onset. Target subspaces were computed by regressing neural activity onto time series that represented relative target positions. As with the movement and position subspaces, the time series spanned 200 ms before to 1000 ms after PC1 onset. A boxcar window was used to confine the relative target position information to the time period spanning 0 to 200 ms after PC1 onset, and the rest of the window was zero-filled. For kinematic prediction from neural subspaces, we used a delay of 120 ms and 80/20 trial-wise training and validation split. For each behavioral variable and neural data type, a 5-fold cross-validated grid search ($n = 100$) was used on training data to find the best-performing regularization across orders of magnitude between 1e-5 and 1e4.

Single subspace dimensions were aligned to 200 ms before and 850 ms after PC1 onset for plotting. Subspace activations were calculated by computing the norm of activations across all dimensions of the subspace and then rescaling the min and max activations to 0 and 1, respectively. Multidimensional subspace plots for the movement subspace were aligned to 180 ms before and 620 ms after PC1 onset and for target subspace 180 ms before and 20 ms after.

Area 2 bump task

The sensory dataset consisted of two recording sessions during which a monkey moved a manipulandum to direct a cursor towards one of eight targets (active trials). During passive trials, the manipulandum induced a mechanical perturbation to the monkey's hand prior to the reach. Activity was recorded via an intracortical electrode array embedded in Brodmann's area 2 of the somatosensory cortex. For the second session, joint angles were calculated from motion tracking data collected throughout the session. The first session was used for PSTH, GLM, subspace, and velocity decoding analyses and the second session was only used for pseudo-R^2 comparison to GLM and joint angle decoding. More details on the task and dataset are given in the original paper (39).

For both sessions, only sorted units were used. Spikes were binned at 1 ms and neurons that were correlated over the first 1000 sec were removed ($n = 2$ for each session) as described for the random target task, resulting in 53 and 68 neurons in the first and second sessions, respectively. Spikes were then rebinned to 5 ms and the continuous data were chopped into 500 ms segments with 200 ms of overlap. Segments that did not include data from rewarded trials were discarded (kept 9,626 for the first session and 7,038 for the second session). A subset of the segments (30%) were further split into training and validation data (80/20) for LFADS. An AutoLFADS model (32 workers) was trained on each session and a random search (96 models) was performed on the first session. After modeling, LFADS rates were then reassembled into their continuous form, with linear merging of overlapping data points.

Empirical PSTHs were computed by convolving spikes binned at 1 ms with a half-Gaussian (10 ms s.d.), rebinning to 5 ms, and then averaging across all trials within a condition. LFADS PSTHs were computed by similarly averaging LFADS rates. Passive trials were aligned 100 ms before and 500 ms after the time of perturbation, and active trials were aligned to the same window around an acceleration-based movement onset (39). Neurons with firing rates lower than 1 Hz were excluded from the PSTH analysis. To quantitatively evaluate PSTH reconstruction, the coefficient of determination was computed for each neuron and passive condition in the four cardinal directions, and these numbers were averaged for each model.

As a baseline for how well AutoLFADS could reconstruct neural activity, we fit generalized linear models (GLMs) to each individual neuron's firing rate, based on the position and velocity of and forces on the hand (see Chowdhury et al., 2020 for details of the hand kinematic-force GLM). Notably, in addition to fitting GLMs using the concurrent behavioral covariates, we also added 10 bins of behavioral history (50 ms) to the GLM covariates, increasing the number of GLM parameters almost tenfold. Furthermore, because we wanted to find the performance ceiling of a behavioral-encoder-based GLMs to compare with the dynamics-based AutoLFADS, we purposefully did not cross-validate the GLMs. Instead, we simply evaluated GLM fits on data used to train the model.

To evaluate AutoLFADS and GLMs individually, we used the pseudo-R^2 (pR^2), a goodness-of-fit metric adapted for the Poisson-like statistics of neural activity. Like variance-accounted-for and R^2, pR^2 has a maximum value of 1 when a model perfectly predicts the data, and a value of 0 when a model predicts as well as a single parameter mean model. Negative values indicate predictions that are worse than a mean model. For each neuron, we compared the pR^2 of the AutoLFADS model to that of the GLM (Fig 5e). To determine statistically whether AutoLFADS performed better than GLMs, we used the relative-pR^2 (rpR^2) metric, which compares the two models against each other, rather than to a mean model (see Perich et al., 2018 for full description of pR^2 and rpR^2). In this case, a rpR^2 value above 0 indicated that AutoLFADS outperformed the GLM (indicated by filled circles in Fig 5e). We assessed significance using a bootstrapping procedure, after fitting both AutoLFADS and GLMs on the data. On each bootstrap iteration, we drew a number of trials from the session (with replacement) equal to the total number of trials in the session, evaluating the rpR^2 on this set of trials as one bootstrap sample. We repeated this procedure 100 times. We defined neurons for which at least 95 of these rpR^2 samples were greater than 0 as neurons that were predicted better by AutoLFADS than a GLM. Likewise, neurons for which at least 95 of these samples were below 0 would have been defined as neurons predicted better by GLM (though there were no neurons with this result).

For the subspace analysis, spikes were smoothed by convolution with a Gaussian (50 ms s.d.) and then rebinned to 50 ms. Neural activity was scaled using the same soft-normalization approach outlined for the random target task subspace analysis. Movement onset was calculated using the acceleration-based movement onset approach for both active and passive trials. For decoder training, trials were aligned to 100 ms before to 600 ms after movement onset. For plotting, trials were aligned to 50 ms before and 600 ms after movement onset. The data for successful reaches in the four cardinal directions was divided into 80/20 trial-wise training and validation partitions. Separate ridge regression models were trained to predict each hand velocity dimension for active and passive trials using neural activity delayed by 50 ms (total 4 decoders). The regularization penalty was determined through a 5-fold cross validated grid search of 25 values from the same range as the random target task subspace decoders.

For hand velocity decoding, spikes during active trials were smoothed by convolution with a half-Gaussian (50 ms s.d.) and neural activity was delayed by 100 ms relative to kinematics. The data were aligned to 200 ms before and 1200 ms after movement onset and trials were split into 80/20 training and validation sets. Simple regression was used to estimate kinematics from neural activity and the coefficient of determination was computed and averaged across x- and y-velocity.

GPFA was performed on segments from all rewarded trials using a latent dimension of 20 and Gaussian smoothing kernel (30 ms s.d.). Decoding data were extracted by aligning data from active trials to 200 ms before and 500 ms after movement onset. Data were split into 80/20 training and validation sets and neural activity was lagged 100 ms behind kinematics. Ridge regression ($\lambda = 0.001$) was used to decode all joint angle velocities from smoothed spikes (half-Gaussian, 50 ms kernel s.d.), rates inferred by GPFA, and rates inferred by AutoLFADS.

DMFC timing task

The cognitive dataset consisted of one session of recordings from the dorsomedial frontal cortex (DMFC) while a monkey performed a time interval reproduction task. The monkey was presented with a "Ready" visual stimulus to indicate the start of the interval and a second "Set" visual stimulus to indicate the end of the sample timing interval, t_s. Following the Set stimulus, the monkey made a response ("Go") so that the production interval (t_p) between Set and Go matches the corresponding t_s. The animal responded with either a saccadic eye movement or a joystick manipulation to the left or right depending on the location of a peripheral target. The two response modalities, combined with 10 timing conditions

(t_s) and two target locations, led to a total of 40 task conditions. A more detailed description of the task is available in the original paper (57).

To prepare the data for LFADS, the spikes from sorted units were binned at 20 ms. To avoid artifacts from correlated spiking activity, we computed cross-correlations between all pairs of neurons for the duration of the experiment and sequentially removed individual neurons ($n = 8$) by the number of above-threshold correlations until there were no pairs with correlation above 0.2, resulting in 45 uncorrelated neurons. Data between the "Ready" cue and the trial end was chopped into 2600 ms segments with no overlap. The first chop for each trial was randomly offset by between 0 and 100 ms to break any link between trial start times and chop start times. The resulting neural data segments (1659 total) were split into 80/20 training and validation sets for LFADS. An AutoLFADS model (32 workers) and random search (96 models) were trained on these segments (see **Supp. Table 2**).

For all analyses of smoothed spikes, smoothing was performed by convolving with a Gaussian kernel (widths described below) at 1 ms resolution.

Empirical PSTHs were computed by trial-averaging smoothed spikes (25 ms kernel s.d., 20 ms bins) within each of the 40 conditions. LFADS PSTHs were computed by similarly averaging LFADS rates. The coefficient of determination was computed between inferred and empirical PSTHs across all neurons and time steps during the "Ready-Set" and "Set-Go" periods for each condition and then averaged across periods and conditions.

To visualize low-dimensional neural trajectories, demixed principal component analysis (dPCA; Kobak et al., 2016) was performed on smoothed spikes (40 ms kernel s.d., 20 ms bins) and AutoLFADS rates during the "Ready-Set" period. The two conditions used were rightward and leftward hand movements with $t_s = 1000\, ms$.

Besides LFADS/AutoLFADS, three alternate methods were applied for speed-tp correlation comparisons: spike smoothing, GPFA, and PCA. For spike smoothing, analyses were performed by smoothing with a 40 ms s.d.. For GPFA, a model was trained on the concatenated training and validation sets with a latent dimension of 9. Principal component analysis (PCA) was performed on smoothed spikes (40 ms kernel s.d., 20 ms bins) and 5-7 top PCs that explained more than 75% of data variance across conditions were included in the later analysis.

Neural speed was calculated by computing distances between consecutive time bins in a multidimensional state space and then averaging the distances across the time bins for the production epoch. The number of dimensions used to compute the neural speed was 45, 5-7, 9, and 45 for smoothing, PCA, GPFA and LFADS, respectively. The Pearson's correlation coefficient between neural speed and the produced time interval was computed across trials within each condition.