Deep inference of latent dynamics with spatio-temporal super-resolution using selective backpropagation through time

Anonymous Author(s)
Affiliation
Address
email

Abstract

Modern neural interfaces allow access to the activity of up to a million neurons within brain circuits. However, bandwidth limits often create a trade-off between greater spatial sampling (more channels or pixels) and the temporal frequency of sampling. Here we demonstrate that it is possible to obtain spatio-temporal super-resolution in neuronal time series by exploiting relationships among neurons, embedded in latent low-dimensional population dynamics. Our novel neural network training strategy, selective backpropagation through time (SBTT), enables learning of deep generative models of latent dynamics from data in which the set of observed variables changes at each time step. The resulting models are able to infer activity for missing samples by combining observations with learned latent dynamics. We test SBTT applied to sequential autoencoders and demonstrate more efficient and higher-fidelity characterization of neural population dynamics in electrophysiological and calcium imaging data. In electrophysiology, SBTT enables accurate inference of neuronal population dynamics with lower interface bandwidths, providing an avenue to significant power savings for implanted neuroelectronic interfaces. In applications to two-photon calcium imaging, SBTT accurately uncovers high-frequency temporal structure underlying neural population activity, substantially outperforming the current state-of-the-art. Finally, we demonstrate that performance could be further improved by using limited, high-bandwidth sampling to pretrain dynamics models, and then using SBTT to adapt these models for sparsely-sampled data.

1 Introduction

Modern systems neuroscientists have access to the activity of many thousands to potentially millions of neurons via multi-photon calcium imaging and high-density silicon probes [1–4]. Such interfaces provide a qualitatively different picture of brain activity than was achievable even a decade ago.

However, neural interfaces increasingly face a trade-off – the number of neurons that can be accessed (capacity) is often far greater than the number that is simultaneously monitored (bandwidth). For example, with 2-photon calcium imaging (2p; **Fig. 1a**, *top*), hundreds to thousands of neurons are serially scanned by a laser that traverses the field of view, resulting in different neurons being sampled at different times within an imaging frame. As a consequence, a trade-off exists between the size of the field-of-view (and hence the number of neurons monitored), the sampling frequency, and the signal-to-noise with which each neuron is sampled. Whereas current analysis methods treat 2p data as if all neurons within a field-of-view were sampled at the same time at the imaging frame rate, the fact that each neuron is sampled at staggered, known times within the frame could be employed to increase the time resolution.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

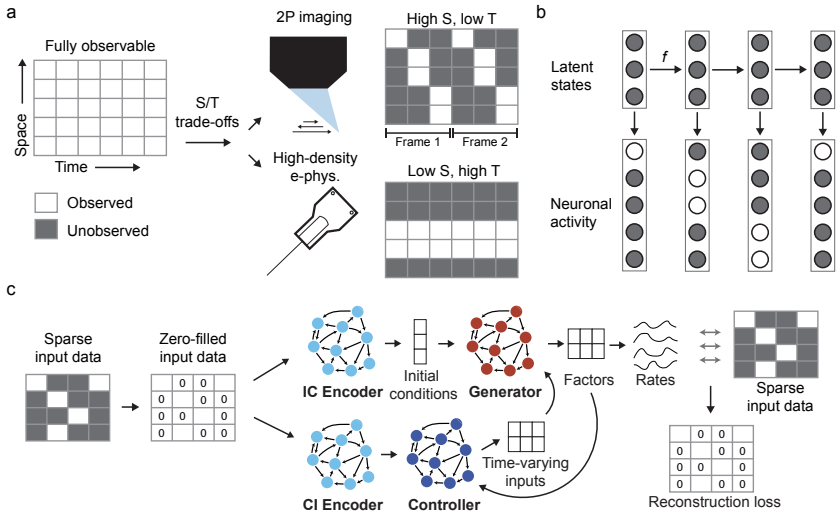

Figure 1: Exploiting space-time trade-offs in neural interfaces using SBTT. (a) In 2-photon calcium imaging (top), individual neurons are serially scanned at a low frame rate, resulting in staggered sample times. In modern electrophysiological recordings (bottom), bandwidth or power constraints prevent simultaneous monitoring of all recording sites. (b) Observed neuronal activity reflects latent, low-dimensional dynamics (captured by the function f). (c) SBTT applied to a sequential autoencoder for inferring latent dynamics from neural population activity.

Electrophysiological interfaces face similar trade-offs (**Fig. 1a**, *bottom*). With groundbreaking high-density probes such as Neuropixels and Neuroseeker [3–5], simultaneous monitoring of all recording sites is either not currently possible or limits the signal-to-noise ratio, so users typically monitor a selected subset of sites within a given recording session. For example, Neuropixels 2.0 probes contain up to 5120 electrodes, 384 of which can be recorded simultaneously [4]. In other situations, power constraints might make it preferable to restrict the number of channels that are simultaneously monitored, such as in wireless or fully-implanted applications where battery life and heat dissipation are key challenges [6–8]. As newer interfacing strategies provide a pathway to hundreds of thousands of channels for revolutionary brain-machine interfaces [9, 10], neural data processing strategies that can leverage dynamic deployment of recording bandwidth might allow substantial power savings.

Solutions to these space-time trade-offs may come from the structure of neural activity itself. A large body of work suggests that the activity of individual neurons within a large population is not independent, but instead is coordinated through a lower-dimensional, latent state that evolves with stereotyped temporal structure (**Fig. 1b**). We can represent the state at time t as a vector $\boldsymbol{x}_t \in \mathbb{R}^D$ that evolves according to dynamics captured by a function f such that $\boldsymbol{x}_{t+1} \approx f(\boldsymbol{x}_t)$. Rather than directly observing the latent state $\boldsymbol{x}_t$, we observe neural activity that we represent as $\boldsymbol{y}_t \in \mathbb{R}^N$, where $\boldsymbol{y}_t \approx h(\boldsymbol{x}_t)$ for some function h. Due both to the fact that f imposes a significant amount of structure on the trajectory of the $\boldsymbol{x}_t$'s and the fact that we typically expect the dimension D of $\boldsymbol{x}_t$ to be far smaller than the number of possible observations N, one might expect that it should be possible to estimate the $\boldsymbol{x}_t$'s without observing every neuron at every time step (i.e., measuring only some of the elements of each $\boldsymbol{y}_t$), just as we generally infer latent states from only a fraction of the neurons in a given area. If so, principled exploitation of the space-time trade-off of neural interfaces might achieve higher-fidelity or more bandwidth-efficient characterization of neural population activity.

To our knowledge, no methods have demonstrated inference of dynamics from data in which the set of neurons being monitored changes dynamically at short intervals. To address this challenge, we introduce *selective backpropagation through time* (SBTT; **Fig. 1c**), a method to train deep generative models of latent dynamics from data where the identity of observed variables varies from sample to sample. Here we explore applications of SBTT to state space modeling of neural population activity that obeys low-dimensional dynamics.

This paper is organized as follows. Section 2 provides an overview of related work. Section 3 details SBTT and its integration with sequential autoencoders for modeling neural population dynamics. Section 4 demonstrates the effectiveness of this solution in achieving more efficient and higher-fidelity inference of latent dynamics in applications to electrophysiological and calcium imaging data.

2 Related work

There is a long and rich literature on methods for system identification, particularly in the case of *linear* dynamical systems. The last several years have witnessed a burst of activity in establishing a more robust theoretical understanding of when and how well these methods work. Particularly relevant to our approach, [11] shows that under suitable conditions on the dynamical system, performing gradient descent on a similar loss function to that considered in this work can provably recover the parameters of the system despite the nonconvexity of the problem. Additional guarantees are provided in [12–16] which make varying assumptions on the underlying dynamics and the observation function, the existence of an observable control input, and the stochasticity of the dynamical system. Adversarial noise models are further considered in [17, 18]. We emphasize, however, that all of the above works limit their focus to *linear* dynamical systems where the observations are *fully sampled*, i.e., where all of $\boldsymbol{y}_t = \boldsymbol{H}\boldsymbol{x}_t$ is measured for all t.

In the case of a linear observation model ($\boldsymbol{y}_t = \boldsymbol{H}\boldsymbol{x}_t$) but where we observe only a subset of the elements of each $\boldsymbol{y}_t$, the problem is reminiscent of the *low-rank matrix completion* problem [19]. Specifically, by letting $\boldsymbol{Y}$ and $\boldsymbol{X}$ denote the matrices whose columns are given by the $\boldsymbol{y}_t$ and $\boldsymbol{x}_t$ respectively, we can write $\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{X}$. If $D \ll N$, this is a low-rank matrix, and hence could be recovered from a random sampling of $O(D \log N)$ elements of each column of $\boldsymbol{Y}$ [19]. However, this strategy essentially assumes that there is no relationship between the $\boldsymbol{x}_t$ – one would expect to obtain significant improvements by exploiting the dynamical structure among the $\boldsymbol{x}_t$ imposed by f. Indeed, in [20, 21] the authors show that if the dynamics f are *known*, then it is possible to significantly reduce the sampling requirements. However, the question of *learning* such an f from undersampled observations has again not been addressed in this literature.

In some application domains, there have been hints in this direction. In particular, in the related contexts of recommendation systems [22, 23] and student knowledge tracking [24, 25] there have been successful empirical efforts aimed at learning dynamical systems for modeling how user preferences/knowledge change over time. While such approaches have also had to confront the issue of missing observations (items that are not rated or questions that are not answered), they are aided by the existence of rich sources of additional metadata (e.g., tags) that lead to fundamentally different approaches than what we take here.

Within our application domain, a variety of methods have been developed to infer latent dynamical structure from neural population activity on individual trials, including those based on Gaussian processes [26–29], linear [30–32] and switching linear dynamical systems [33–35], and nonlinear dynamical systems such as recurrent neural networks [36–39], hidden Markov models [40], neural ODEs [41], and transformers [42]. Variants of these methods accommodate cases where the particular observed neurons change over long time periods (e.g., over the course of days) [36, 43, 44], but these are not appropriate for cases where neurons are intermittently sampled on short timescales. As described below, several of these methods would be amenable to using SBTT to adapt to intermittent sampling, as SBTT should be applicable to any neural network architecture that learns weights via backpropagation through time.

3 Selective backpropagation through time

3.1 Overview

SBTT is a learning rule for updating the weights of a neural network that allows backpropagation of loss for the portions of data that are present while preventing missing data from corrupting the gradient signal. The technique optimizes the model to reconstruct observed data and extrapolate to the unobserved data. SBTT is related to other approaches that augment network inputs and cost functions to reflect different subsets of the data matrix across samples, in particular coordinated dropout [37], masked language modeling [45], and DeepInterpolation [46]. These previous approaches split fully-observed data into two portions - a portion that is provided at the input to the network, and a portion

that is used to compute loss at the output. SBTT uses a similar strategy to accommodate missing data, by zero-filling missing input points and aggregating only losses for observed data points at the output.

3.2 Illustration with a simple linear dynamical system

We begin by describing our approach in the context of a simple linear dynamical system. In the case where we have no (observable) inputs, we can model a linear dynamical system as

$$\begin{aligned} \boldsymbol{x}_{t+1} &= \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{w}_t \\ \boldsymbol{y}_t &= \boldsymbol{H}\boldsymbol{x}_t + \boldsymbol{z}_t. \end{aligned}$$

Here, $\boldsymbol{x} \in \mathbb{R}^D$ represents a hidden state, $\boldsymbol{y} \in \mathbb{R}^N$ represents our observations, and $\boldsymbol{w}_t$ and $\boldsymbol{z}_t$ represent noise. The matrix $\boldsymbol{A}$ models the dynamics of the hidden state, and $\boldsymbol{H}$ models the observation function of our system. In this setting, our task is to learn the parameters $\boldsymbol{A}$ and $\boldsymbol{H}$ given the observations $\boldsymbol{y}_0, \ldots, \boldsymbol{y}_{T-1}$ as well as the initial system state $\boldsymbol{x}_0$.

SBTT is a variation of standard back-propagation where loss terms attributed to missing observations are ignored when computing back-propagation updates. Concretely speaking, consider a linear recurrent network that can learn this linear model using a least squares loss

$$\mathcal{L} = \frac{1}{T}\sum_{t=0}^{T-1} \frac{1}{2}\|\boldsymbol{y}_t - \boldsymbol{H}\boldsymbol{x}_t\|_2^2.$$

If the observation vector $\boldsymbol{y}_t$ contains a missing entry at index i, the least squares loss would not contain the $(y_t^i - (Hx_t)^i)^2$ term, where the superscript i represents the ith index of a vector. If $\boldsymbol{o}_t = \boldsymbol{H}\boldsymbol{x}_t$ is taken to be the output of the recurrent network at time step t, then the loss with respect to the outputs of the network is

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{o}_t} = \frac{1}{T}(\boldsymbol{o}_t - \boldsymbol{y}_t). \tag{1}$$

SBTT requires that loss terms, and subsequently loss gradients, related to missing observations are ignored. This means that elements in the gradient vector (1) are ignored and set to 0 at indices i where the corresponding observations, y_t^i, are missing. This gradient is then back-propagated through time to obtain gradients with respect to model parameters $\boldsymbol{A}$ and $\boldsymbol{H}$ as shown below

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{H}} = \sum_{t=0}^{T-1} \frac{\partial \mathcal{L}}{\partial \boldsymbol{o}_t}(\boldsymbol{x}_t)^\intercal, \frac{\partial \mathcal{L}}{\partial \boldsymbol{A}} = \sum_{t=1}^{T-1} \frac{\partial \mathcal{L}}{\partial \boldsymbol{x}_t}\boldsymbol{x}_{t-1}^\intercal,$$

where $\frac{\partial L}{\partial \boldsymbol{x}_t}$ is recursively computed using back-propagation through time:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{x}_t} = \boldsymbol{A}^\intercal \frac{\partial \mathcal{L}}{\partial \boldsymbol{x}_{t+1}} + \boldsymbol{H}^\intercal \frac{\partial \mathcal{L}}{\partial \boldsymbol{o}_t}.$$

These parameters can then be updated using gradient descent.

3.3 Integration with a deep generative model of neural population dynamics

Here we will demonstrate the use of SBTT with a recently developed framework for inferring nonlinear latent dynamics from neural population recordings. This framework, latent factor analysis via dynamical systems (LFADS), is a sequential variational auto-encoder (SVAE), detailed in [36]. LFADS models single-trial latent dynamics by learning the initial state of the dynamical system, the dynamical rules that govern state evolution, and any time-varying inputs that cannot be explained by the dynamics (i.e., in the case of a non-autonomous dynamical system). Briefly, a bidirectional RNN encoder operates on the neural spiking sequence $\mathbf{y}(t)$ and produces a conditional distribution over initial condition $\mathbf{z}$, $Q(\mathbf{z}|\mathbf{y}(t))$. A Kullback-Leibler (KL) divergence penalty is applied as a regularizer for divergence between the uninformative prior $P(\mathbf{z})$ and $Q(\mathbf{z}|\mathbf{y}(t))$. The initial condition is then drawn from $Q(\mathbf{z}|\mathbf{y}(t))$ and used as the initial state for a generator RNN, which learns to approximate the dynamical rules underlying the neural data. A controller RNN takes as input the state of the generator at each time step, along with a time-varying encoding of $\mathbf{y}(t)$ (produced by a second bidirectional RNN encoder), and injects a time-varying input $\mathbf{u}(t)$ into the generator. Similar

to $\mathbf{z}$, $\mathbf{u}(t)$ is drawn from a parameterized time-varying distribution of $Q(\mathbf{u}(t)|\mathbf{x}(t))$ produced by the controller. 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 in turn mapped to the firing rates of the neurons using a linear mapping followed by an exponential nonlinearity. LFADS assumes a Possion emissions model for the observed spiking activity. The optimization objective combines the reconstruction cost of the observed spiking activity (i.e., the Poisson likelihood of the observed spiking activity given the rates produced by the generator network), the KL penalties described above, and L2 regularization penalties on the weights of the recurrent networks. During training, network weights are optimized using stochastic gradient descent and backpropagation through time.

Throughout this paper we use population-based training along with coordinated dropout, together known as AutoLFADS, to optimize our models [37, 39, 47]. This framework is essential for achieving reliably high-performing LFADS models, regardless of dataset statistics. Hyperparameters, search ranges, and training details are given in the supplement.

4 Experiments

4.1 High performance with limited bandwidth on primate electrophysiological recordings

A key target application of AutoLFADS with SBTT (A-SBTT) is to enable reduced sampling of electrodes: either to enable recording from larger populations of electrodes with limited bandwidth (such as with Neuropixels), or to reduce power consumption (such as for fully-implantable brain-machine interfaces). To investigate the performance of AutoLFADS models trained with SBTT, we started with a large and well-characterized dataset containing electrophysiological recordings from macaque primary motor and dorsal premotor cortex (M1/PMd) [48]. The data were collected during a delayed reaching task, in which the monkey made both straight and curved reaches from a center position, around virtual barriers (the maze), to one of 108 possible target positions. The dataset consisted of 2296 trials with 202 sorted units aligned to movement onset in a window from 250 ms before to 450 ms after this point. Spike counts were binned at 10 ms (70 bins). We held out 50 randomly selected units from modeling to use for evaluation of inferred latent factors. We simulated various missing data scenarios for the remaining 152 units by randomly masking a fraction of the observations at each time step for each trial (**Fig. 2a**, *top*).

For each of the masked datasets, we used A-SBTT to robustly train neural dynamics models. Latent factors and firing rates were inferred for all time steps, despite the missing (masked) observations. Even with 70% dropped samples, the inferred firing rates showed structure comparable to the model of fully observed data (**Fig. 2a**, *bottom*).

To determine whether the models were able to capture biologically relevant information from sparsely sampled data, we evaluated the inferred latent factors in terms of their ability to predict hand velocity (**Fig. 2b**) and the spiking activity of held-out units (**Fig. 2c**). As a baseline, we trained a Gaussian Process Factor Analysis (GPFA) model (40 latent dimensions, 20 ms bins) on the fully observed dataset [26, 49]. GPFA is a commonly-used and versatile method for extracting latent structure from neural population activity. We trained simple linear decoders to predict hand velocity from the inferred latent factors with an 80 ms delay (50/50, trial-wise train-test split), and evaluated using the coefficient of determination, averaged over x- and y-dimensions. For A-SBTT, decoding performance showed a minimal decline until around 80% of the data had been dropped, with some models outperforming the GPFA baseline using as little as 15% of the original data (**Fig. 2b**). To measure how well the models captured the population structure, we trained generalized linear models (GLMs) [50, 51] to predict the spikes for the held out units and evaluated fit quality using pseudo-R^2 (pR^2). Similar to the decoding results, we found that A-SBTT captured population structure significantly better than fully observed GPFA, and that the information content of the factors declined slowly until about 80% missing samples (**Fig. 2c**).

4.2 Recovery of high frequency features in simulated 2P calcium imaging data

High-frequency features of neural responses are generally assumed to be lost in 2P imaging due to limited scanning speeds and indicator kinetics. We hypothesized that some of the loss is actually due to standard 2P data processing, which discards information regarding sub-frame sampling time of

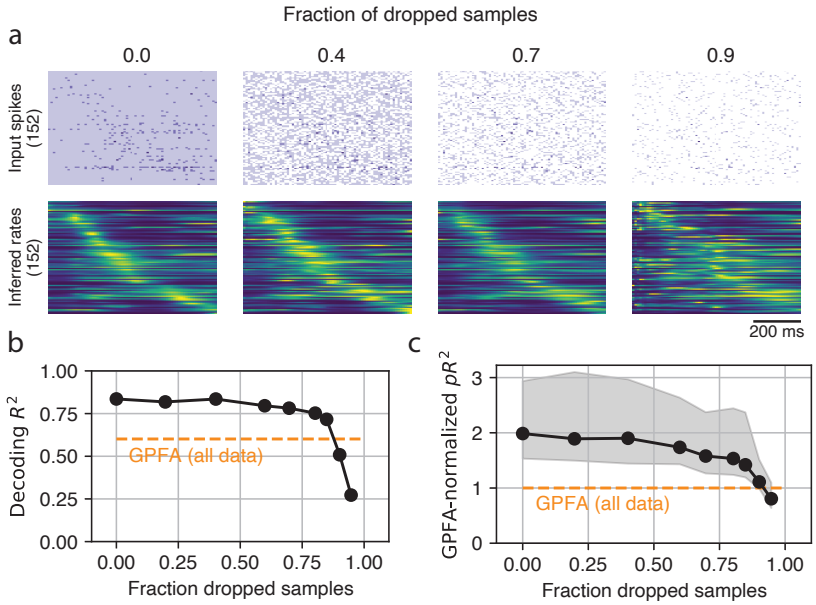

Figure 2: SBTT allows inference of latent dynamics from M1/PMd electrophysiology data with sparse observations. (a) Spike count input and inferred rate output of LFADS for the same example trial with increasingly sparse observations. Masked data are shown in white, observed zeroes are shown in light purple, and nonzero spike counts are shown in darker shades. Units are sorted by timing of firing rate peaks for the fully sampled model. (b) Accuracy of linear hand velocity decoding from inferred latent factors. (c) Quality of GLM fits from inferred latent factors to 50 held-out units. pR^2 values for each held-out unit are normalized to the corresponding values achieved by the GPFA baseline. Points denote the median across all units. Shaded areas depict the 25th and 75th quantiles.

individual neurons, and that SBTT could recover some of this information. The inherently staggered sampling of neurons due to raster scanning can be treated as a time series with missing values and higher temporal resolution than the frame rate. We tested SBTT on both simulated and real calcium imaging data. In both cases, we adapted AutoLFADS to better account for the statistics of deconvolved calcium activity (AutoLFADS-ZIG, see supplement) by substituting the underlying Poisson emission model with a Zero-inflated Gamma distribution [52]. In our experiments we compared three methods: AutoLFADS-ZIG with SBTT (A-ZIG-SBTT), a standard frame-resolution version of AutoLFADS-ZIG without SBTT (A-ZIG-FR), and Gaussian smoothing of deconvolved calcium activity.

We generated artificial 2P data from a population of simulated neurons (278 neurons) whose firing rates were linked to the state of an underlying Lorenz system [28, 53] (see supplement). To assess the ability to reconstruct latent dynamics at different frequencies, we simulated Lorenz systems with different speeds. For each Lorenz system we report the Z dimension power spectrum peak, which contains the most concentrated and highest frequencies. Fluorescence traces were simulated from the spike trains using an order 1 autoregressive model followed by a non-linearity and injected with 4 sources of noise (see supplement). Firing rates were simulated at 100Hz, and a "location" was randomly chosen for each simulated neuron, such that sampling times for different neurons were staggered to simulate 2p laser scanning sampling times. This produced fluorescence traces with one of three possible associated phases (0,11,22ms) and overall sample rate 33 Hz. We deconvolved neural activity from the fluorescence traces using the OASIS algorithm [54] as implemented in the CaImAn package [55].

For A-ZIG-SBTT we used the sub-frame phase information to generate intermittently-sampled data. In contrast, for both A-ZIG-FR and Gaussian smoothing, we discarded phase information and collapsed samples into a single time bin per frame, as is standard in 2p imaging data processing. To evaluate the performance in recovering the ground truth Lorenz states, we trained a mapping from the output of each method (i.e., the inferred event rates from A-ZIG-SBTT and A-ZIG-FR, and

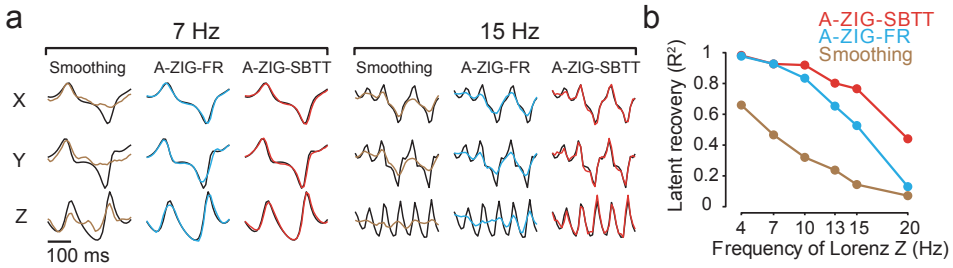

Figure 3: SBTT improves inference of high-frequency dynamics from simulated 2P data with known dynamical structure. (a) True and inferred Lorenz latent states (X/Y/Z dimensions) for a single example trial from Lorenz systems simulated at two different frequencies (7Hz and 15Hz). Black: ground truth. Colored: inferred. (c) Performance in estimating the Lorenz Z dimension as a function of Lorenz speed was quantified by variance explained (R^2) for all three methods. The speed of the Lorenz dynamics was quantified based on the peak location of the power spectra of Lorenz Z dimension, with a sampling frequency of 100Hz.

smoothed deconvolved events by Gaussian smoothing; signals were interpolated to 100 Hz for the latter methods) to the ground truth Lorenz states using cross-validated ridge regression. We used R^2 between the true and inferred Lorenz states as a metric of performance.

Fig. 3a illustrates true and predicted Lorenz states for two example trials. The performance of smoothing and A-FR dropped substantially for higher Lorenz state frequencies, while A-SBTT maintained reasonable estimates ($R^2 \approx 0.8$) up to 15Hz (**Fig. 3a & b**) and never dropped below 0.4 in the range of tested frequencies.

4.3 Improved representation of hand kinematics in mouse 2P calcium imaging data

We next applied SBTT to real 2P calcium imaging data we collected from motor cortex in a mouse performing a forelimb water grab task. The dataset comprised 475 trials in which the mouse was cued by a tone to reach to a left or right spout and retrieve a droplet of water with its right forepaw. Pyramidal cells expressing the GCaMP6f calcium indicator were imaged with a two-photon microscope at a 31 Hz frame rate, and a subset of 439 modulated neurons within the field-of-view (FOV) were considered for analysis (FOV shown in **Fig. 4a**, *left*; example calcium traces in **Fig. 4a**, *right*). The mouse's forepaw position was tracked in 3D at 150 Hz with stereo cameras and DeepLabCut [56]. Calcium events were deconvolved with OASIS [54, 55].

2P data for A-ZIG-SBTT were processed analogously to the simulations, using neuron locations within the FOV to inform the intermittent sampling times. Trials represented a window spanning 200 ms before to 800 ms after the mouse's reach onset. This resulted in 100 time points per trial for A-ZIG-SBTT, and 31 time points per trial for A-ZIG-FR and Gaussian smoothing. For both A-ZIG-SBTT and A-ZIG-FR, trials were split into 80/20 train/validation.

To compare representations inferred by A-ZIG-SBTT and A-ZIG-FR, we first evaluated how closely the single-trial event rates inferred for each neuron resembled that neuron's peri-stimulus time histogram (PSTH). PSTHs were calculated by taking the average of the Gaussian-smoothed deconvolved events across trials within each experimental condition. Because the mouse's reaches were not stereotyped to each spout (i.e., left or right), we subgrouped trials into 4 finer conditions based on forepaw Z position during the reach. A-ZIG-SBTT single-trial event rates were more strongly correlated with neurons' PSTHs compared to those inferred by A-ZIG-FR (**Fig. 4b**).

We next decoded the mouse's single-trial forepaw kinematics (position and velocity) based on each model's output. Decoding was performed using ridge regression with 5-fold cross validation. We used R^2 between the true and predicted hand positions and velocities as a metric of performance. R^2 was averaged across XYZ behavioral dimensions and all 5 folds of the test sets. Decoding using A-ZIG-SBTT inferred rates outperformed results from smoothing deconvolved events, or from the A-ZIG-FR inferred rates (**Fig. 4c**). Because the improvement of decoding performance for position is modest, we further assessed how the improvement was distributed as a function of temporal frequency. We computed the coherence between the true and decoded positions for each method

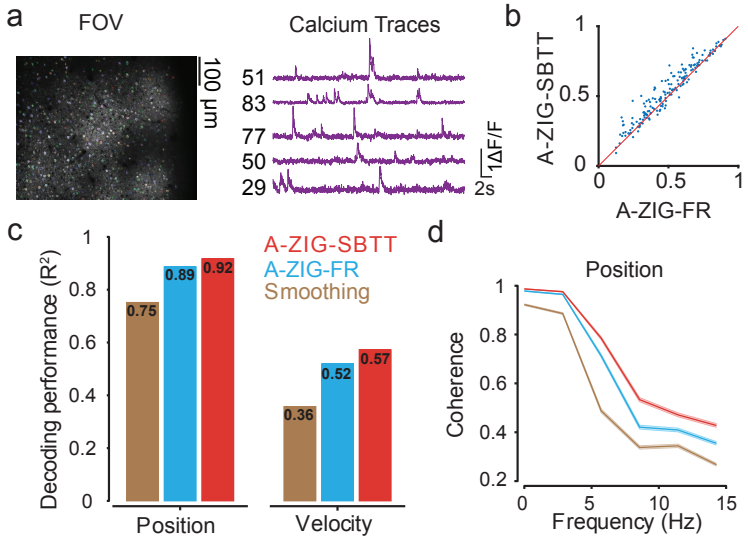

Figure 4: SBTT improves inference of latent dynamics from mouse 2P calcium imaging data. (a) Left: an example field-of-view (FOV), colored by neurons. Right: calcium traces (dF/F) from a single trial for 5 example neurons. (b) Performance of capturing empirical PSTHs was quantified by computing the correlation coefficient r between the inferred single-trial event rates and empirical PSTHs, comparing A-ZIG-FR vs A-ZIG-SB. Each point represents an individual neuron. (c) Decoding performance was quantified by computing the R^2 between the true and decoded position (left) and velocity (right) across all trials. (d) Quality of reconstructing the kinematics across frequencies was quantified by measuring coherence between the true and decoded position for all three methods.

(**Fig. 4d**). Consistent with the simulations, A-ZIG-SBTT predictions showed higher coherence with true position than predictions from other methods, with improvements more prominent at higher frequencies (5-15Hz).

4.4 Using high-bandwidth observations to improve performance in low-bandwidth conditions

In implantable or wireless applications, using the device's full interface bandwidth might incur significant power costs, which would burden users with frequent battery recharging. However, it may be possible to leverage high-bandwidth recordings from limited time periods to learn models of latent dynamics, and then switch to low-bandwidth modes for subsequent long-term operation, in order to minimize ongoing power use. Such an approach is enabled by the stability of latent dynamics over months to years [36, 44, 57].

We tested these ideas on the same electrophysiological dataset described in section 4.1. After training AutoLFADS models on the fully sampled data, we retrained the initial condition and controller input encoders using SBTT on each of the sparsely sampled datasets. The weights for the rest of the network remained fixed. In this way, the dynamical rules learned from the fully sampled data are maintained, while the mappings from data to the initial conditions and controller inputs are adapted for sparse data. Retraining the encoding networks in this way (**Fig. 5**, "Retrained sparse") maintained performance to high levels of missing data, outperforming AutoLFADS trained on fully observed data but run with missing data (**Fig. 5**, "Trained full") or training directly on sparsely-sampled data (**Fig. 5**, "Trained sparse", same as in **Fig. 2b**). These results show that dynamics models are learned most accurately on fully observed data, but that the learned dynamics can be used to model sparsely sampled data if models are adapted to the sparser domain using SBTT.

5 Discussion

We introduced SBTT, a novel approach for learning latent dynamics from irregularly or sparsely sampled time series data. In experiments on real electrophysiology data from macaque motor cortex,

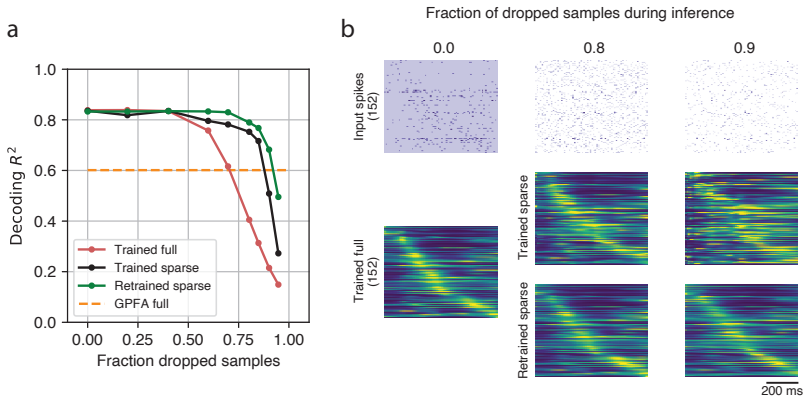

Figure 5: Retraining full-data LFADS encoders on sparse data improves decoding performance. (a) Hand velocity decoding performance in function of dropped samples (as in **Fig. 2b**). "Trained full" indicates training on fully observed data and inference on sparse data. "Trained sparse" indicates training and inference on sparse data. "Retrained sparse" indicates training on fully observed data, followed by encoder retraining and inference on sparse data. (b) Spike count input and inferred rate output of LFADS. Conventions are as in **Fig. 2a**.

we show that models trained with SBTT learn biologically relevant neural dynamics with up to 80% masked training data. On data from a synthetic 2P calcium imaging simulation, we show that models trained with SBTT capture high frequency features of the latent dynamics that are not captured at frame resolution. We also showed improved behavioral decoding performance on real 2P imaging data from mouse M1. Finally, we demonstrate that retraining the early layers of a full-data model on sparse datasets using SBTT can substantially improve decoding performance at the most challenging sparsity levels, outperforming models trained on the sparse data alone. Taken together, these results clearly show that SBTT is a valuable technique for training models with irregularly or sparsely sampled time series data.

5.1 Limitations

Though we made an effort to characterize performance across multiple potential applications, it remains untested how this approach would generalize to other experimental settings (microscopes, calcium indicators, expression levels), model systems, and brain areas or tasks with more complex or higher-dimensional dynamics [39], so we are optimistic that these properties will extend to AutoLFADS models that use SBTT. Applications to brain-machine interfaces await incorporation of neural network-based dynamics models into closed-loop, real time systems. We also note that hardware implementations of intermittent sampling for electrophysiology are still largely unexplored, and might incur time or power costs when switching between channels. This might change the point at which intermittent sampling is beneficial from a power or performance perspective. We hope that this work indicates new directions for future generations of recording hardware that focus on high interface capacities and rapid switching between contacts.

5.2 Broader impact

Our results could pave the way to substantially decreased power consumption for fully-implantable brain-machine interfaces. Ultimately, this should result in more reliable and less burdensome assistive devices for people with disabilities. Further, expanding the information that can be gathered through a given recording bandwidth has scientific implications, and could enable neuroscientists to ask new questions via larger-scale studies of the brain.

Like any resource-intensive technology, this technique has the potential to increase inequity by only benefiting those who can afford the most advanced neural interfaces. Efforts to deploy such technologies should weigh input from ethicists to ensure that everyone benefits from these scientific innovations [58, 59].

References

[1] Carsen Stringer, Marius Pachitariu, Nicholas Steinmetz, Matteo Carandini, and Kenneth D Harris. High-dimensional geometry of population responses in visual cortex. *Nature*, 571(7765): 361–365, 2019.

[2] Jeffery Demas, Jason Manley, Frank Tejera, Hyewon Kim, Francisca Martinez Traub, Brandon Chen, and Alipasha Vaziri. High-speed, cortex-wide volumetric recording of neuroactivity at cellular resolution using light beads microscopy. *bioRxiv*, 2021.

[3] James J Jun, Nicholas A Steinmetz, Joshua H Siegle, Daniel J Denman, Marius Bauza, Brian Barbarits, Albert K Lee, Costas A Anastassiou, Alexandru Andrei, Çağatay Aydın, et al. Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 551(7679): 232–236, 2017.

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

[5] Bogdan C Raducanu, Refet F Yazicioglu, Carolina M Lopez, Marco Ballini, Jan Putzeys, Shiwei Wang, Alexandru Andrei, Veronique Rochus, Marleen Welkenhuysen, Nick van Helleputte, et al. Time multiplexed active neural probe with 1356 parallel recording sites. *Sensors*, 17(10): 2388, 2017.

[6] Henrique Miranda, Vikash Gilja, Cindy A Chestek, Krishna V Shenoy, and Teresa H Meng. Hermesd: A high-rate long-range wireless transmission system for simultaneous multichannel neural recording applications. *IEEE Transactions on Biomedical Circuits and Systems*, 4(3): 181–191, 2010.

[7] David A Borton, Ming Yin, Juan Aceros, and Arto Nurmikko. An implantable wireless neural interface for recording cortical circuit dynamics in moving primates. *Journal of neural engineering*, 10(2):026010, 2013.

[8] John D Simeral, Thomas Hosman, Jad Saab, Sharlene N Flesher, Marco Vilela, Brian Franco, Jessica Kelemen, David M Brandman, John G Ciancibello, Paymon G Rezaii, et al. Home use of a percutaneous wireless intracortical brain-computer interface by individuals with tetraplegia. *IEEE Transactions on Biomedical Engineering*, 2021.

[9] Kunal Sahasrabuddhe, Aamir A Khan, Aditya P Singh, Tyler M Stern, Yeena Ng, Aleksandar Tadić, Peter Orel, Chris LaReau, Daniel Pouzzner, Kurtis Nishimura, et al. The argo: A high channel count recording system for neural recording in vivo. *Journal of Neural Engineering*, 18 (1):015002, 2021.

[10] Elon Musk et al. An integrated brain-machine interface platform with thousands of channels. *Journal of medical Internet research*, 21(10):e16194, 2019.

[11] Moritz Hardt, Tengyu Ma, and Benjamin Recht. Gradient descent learns linear dynamical systems. *Journal of Machine Learning Research*, 19(29):1–44, 2018. URL `http://jmlr.org/papers/v19/16-465.html`.

[12] Max Simchowitz, Horia Mania, Stephen Tu, Michael I. Jordan, and Benjamin Recht. Learning without mixing: Towards a sharp analysis of linear system identification. In *Proc. Conference on Learning Theory (COLT)*, volume 75 of *Proceedings of Machine Learning Research*, pages 439–473. PMLR, July 2018.

[13] Elad Hazan, Holden Lee, Karan Singh, Cyril Zhang, and Yi Zhang. Spectral filtering for general linear dynamical systems. In *Adv. in Neural Information Processing Systems (NeurIPS)*, Montréal, Canada, December 2018.

[14] Samet Oymak and Necmiye Ozay. Non-asymptotic identification of lti systems from a single trajectory. In *2019 American Control Conference (ACC)*, pages 5655–5661, 2019. doi: 10.23919/ACC.2019.8814438.

[15] Anastasios Tsiamis and George J. Pappas. Finite sample analysis of stochastic system identification. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 3648–3654, 2019. doi: 10.1109/CDC40024.2019.9029499.

[16] Holden Lee and Cyril Zhang. Robust guarantees for learning an autoregressive filter. In Aryeh Kontorovich and Gergely Neu, editors, *Proceedings of the 31st International Conference on Algorithmic Learning Theory*, volume 117 of *Proceedings of Machine Learning Research*, pages 490–517, San Diego, California, USA, February 2020. PMLR. URL `http://proceedings.mlr.press/v117/lee20a.html`.

[17] Max Simchowitz, Ross Boczar, and Benjamin Recht. Learning linear dynamical systems with semi-parametric least squares. In Alina Beygelzimer and Daniel Hsu, editors, *Proc. Conference on Learning Theory (COLT)*, volume 99 of *Proceedings of Machine Learning Research*, pages 2714–2802, Phoenix, USA, June 2019. PMLR. URL `http://proceedings.mlr.press/v99/simchowitz19a.html`.

[18] Max Simchowitz, Karan Singh, and Elad Hazan. Improper learning for non-stochastic control. In Jacob Abernethy and Shivani Agarwal, editors, *Proc. Conference on Learning Theory (COLT)*, volume 125 of *Proceedings of Machine Learning Research*, pages 3320–3436. PMLR, July 2020. URL `http://proceedings.mlr.press/v125/simchowitz20a.html`.

[19] Mark A. Davenport and Justin Romberg. An overview of low-rank matrix recovery from incomplete observations. *IEEE J. of Selected Topics in Signal Processing*, 10(4):608–622, June 2016.

[20] Liangbei Xu and Mark Davenport. Dynamic matrix recovery from incomplete observations under an exact low-rank constraint. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29. Curran Associates, Inc., 2016. URL `https://proceedings.neurips.cc/paper/2016/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf`.

[21] Liangbei Xu and Mark A. Davenport. Simultaneous recovery of a series of low-rank matrices by locally weighted matrix smoothing. In *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 1–5, 2017. doi: 10.1109/CAMSAP.2017.8313104.

[22] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. November 2015.

[23] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J. Smola, and How Jing. Recurrent recommender networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, pages 495–503, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450346757. doi: 10.1145/3018661.3018689.

[24] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. Deep knowledge tracing. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28. Curran Associates, Inc., 2015. URL `https://proceedings.neurips.cc/paper/2015/file/bac9162b47c56fc8a4d2a519803d51b3-Paper.pdf`.

[25] Liangbei Xu and Mark A. Davenport. Dynamic knowledge embedding and tracing. May 2020.

[26] Byron M Yu, John P Cunningham, Gopal Santhanam, Stephen I Ryu, Krishna V Shenoy, and Maneesh Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. In *Advances in neural information processing systems*, pages 1881–1888, 2009.

[27] Lea Duncker, Gergo Bohner, Julien Boussard, and Maneesh Sahani. Learning interpretable continuous-time models of latent stochastic dynamical systems. In *International Conference on Machine Learning*, pages 1726–1734. PMLR, 2019.

[28] Yuan Zhao and Il Memming Park. Variational latent gaussian process for recovering single-trial dynamics from population spike trains. *Neural computation*, 29(5):1293–1316, 2017.

[29] Anqi Wu, Nicholas A Roy, Stephen Keeley, and Jonathan W Pillow. Gaussian process based nonlinear latent structure discovery in multivariate spike train data. *Advances in neural information processing systems*, 30:3496, 2017.

[30] Jakob H Macke, Lars Buesing, John P Cunningham, Byron M Yu, Krishna V Shenoy, and Maneesh Sahani. Empirical models of spiking in neural populations. In *Advances in Neural Information Processing Systems 24: 25th conference on Neural Information Processing Systems (NIPS 2011)*, pages 1350–1358, 2012.

[31] Yuanjun Gao, Evan Archer, Liam Paninski, and John P Cunningham. Linear dynamical neural population models through nonlinear embeddings. *arXiv preprint arXiv:1605.08454*, 2016.

[32] Jonathan C Kao, Paul Nuyujukian, Stephen I Ryu, Mark M Churchland, John P Cunningham, and Krishna V Shenoy. Single-trial dynamics of motor cortex and their applications to brain-machine interfaces. *Nature communications*, 6(1):1–12, 2015.

[33] Biljana Petreska, Byron M Yu, John P Cunningham, Gopal Santhanam, Stephen Ryu, Krishna V Shenoy, and Maneesh Sahani. Dynamical segmentation of single trials from population neural data. *Advances in neural information processing systems*, 24:756–764, 2011.

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

[35] Joshua I Glaser, Matthew R Whiteway, John P Cunningham, Liam Paninski, and Scott W Linderman. Recurrent switching dynamical systems models for multiple interacting neural populations. *bioRxiv*, 2020.

[36] 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.

[37] MR Keshtkaran and C Pandarinath. Enabling hyperparameter optimization in sequential autoencoders for spiking neural data. *Advances in Neural Information Processing Systems*, 32, 2019.

[38] Qi She and Anqi Wu. Neural dynamics discovery via gaussian process recurrent neural networks. In *Uncertainty in Artificial Intelligence*, pages 454–464. PMLR, 2020.

[39] Mohammad Reza Keshtkaran, Andrew Robert Sedler, Raeed 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. *bioRxiv*, 2021.

[40] Daniel Hernandez, Antonio Khalil Moretti, Ziqiang Wei, Shreya Saxena, John Cunningham, and Liam Paninski. A novel variational family for hidden nonlinear markov models. *arXiv preprint arXiv:1811.02459*, 2018.

[41] Timothy Doyeon Kim, Thomas Zhihao Luo, Jonathan W. Pillow, and Carlos D. Brody. Inferring latent dynamics underlying neural population activity via neural differential equations. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

[42] Joel Ye and Chethan Pandarinath. Representation learning for neural population activity with neural data transformers. *bioRxiv*, 2021.

[43] Marcel Nonnenmacher, Srinivas C Turaga, and Jakob H Macke. Extracting low-dimensional dynamics from multiple large-scale neural population recordings by learning to predict correlations. *arXiv preprint arXiv:1711.01847*, 2017.

[44] Jonathan C Kao, Stephen I Ryu, and Krishna V Shenoy. Leveraging neural dynamics to extend functional lifetime of brain-machine interfaces. *Scientific reports*, 7(1):1–16, 2017.

[45] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[46] Jerome Lecoq, Michael Oliver, Joshua H Siegle, Natalia Orlova, and Christof Koch. Removing independent noise in systems neuroscience data using deepinterpolation. *bioRxiv*, 2020.

[47] Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, et al. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017.

[48] Mark M Churchland, John P Cunningham, Matthew T Kaufman, Stephen I Ryu, and Krishna V Shenoy. Cortical preparatory activity: representation of movement or first cog in a dynamical machine? *Neuron*, 68(3):387–400, 2010.

[49] M. Denker, A. Yegenoglu, and S. Grün. Collaborative HPC-enabled workflows on the HBP Collaboratory using the Elephant framework. In *Neuroinformatics 2018*, page P19, 2018. doi: 10.12751/incf.ni2018.0019. URL `https://abstracts.g-node.org/conference/NI2018/abstracts#/uuid/023bec4e-0c35-4563-81ce-2c6fac282abd`.

[50] Liam Paninski. Maximum likelihood estimation of cascade point-process neural encoding models. *Network: Computation in Neural Systems*, 15(4):243–262, 2004.

[51] Mainak Jas, Titipat Achakulvisut, Aid Idrizović, Daniel Acuna, Matthew Antalek, Vinicius Marques, Tommy Odland, Ravi Garg, Mayank Agrawal, Yu Umegaki, Peter Foley, Hugo Fernandes, Drew Harris, Beibin Li, Olivier Pieters, Scott Otterson, Giovanni De Toni, Chris Rodgers, Eva Dyer, Matti Hamalainen, Konrad Kording, and Pavan Ramkumar. Pyglmnet: Python implementation of elastic-net regularized generalized linear models. *Journal of Open Source Software*, 5(47):1959, 2020. doi: 10.21105/joss.01959. URL `https://doi.org/10.21105/joss.01959`.

[52] Xue-Xin Wei, Ding Zhou, Andres Grosmark, Zaki Ajabi, Fraser Sparks, Pengcheng Zhou, Mark Brandon, Attila Losonczy, and Liam Paninski. A zero-inflated gamma model for deconvolved calcium imaging traces. *arXiv preprint arXiv:2006.03737*, 2020.

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

[54] Johannes Friedrich, Pengcheng Zhou, and Liam Paninski. Fast online deconvolution of calcium imaging data. *PLoS computational biology*, 13(3):e1005423, 2017.

[55] Andrea Giovannucci, Johannes Friedrich, Pat Gunn, Jérémie Kalfon, Brandon L Brown, Sue Ann Koay, Jiannis Taxidis, Farzaneh Najafi, Jeffrey L Gauthier, Pengcheng Zhou, Baljit S Khakh, David W Tank, Dmitri B Chklovskii, and Eftychios A Pnevmatikakis. CaImAn an open source tool for scalable calcium imaging data analysis. *eLife*, 8:e38173, January 2019. ISSN 2050-084X. doi: 10.7554/eLife.38173. URL `https://doi.org/10.7554/eLife.38173`.

[56] Alexander Mathis, Pranav Mamidanna, Kevin M Cury, Taiga Abe, Venkatesh N Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature neuroscience*, 21(9):1281–1289, 2018.

[57] Juan A Gallego, Matthew G Perich, Raeed H Chowdhury, Sara A Solla, and Lee E Miller. Long-term stability of cortical population dynamics underlying consistent behavior. *Nature neuroscience*, 23(2):260–270, 2020.

[58] Eran Klein, Tim Brown, Matthew Sample, Anjali R Truitt, and Sara Goering. Engineering the brain: ethical issues and the introduction of neural devices. *Hastings Center Report*, 45(6): 26–35, 2015.

[59] Sara Goering, Eran Klein, Laura Specker Sullivan, Anna Wexler, Blaise Agüera y Arcas, Guoqiang Bi, Jose M Carmena, Joseph J Fins, Phoebe Friesen, Jack Gallant, et al. Recommendations for responsible development and application of neurotechnologies. *Neuroethics*, pages 1–22, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Yes, in section 5.1 we describe untested scenarios and suggest potential solutions and future work.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Yes, in section 5.2 we discuss potential positive and negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The real maze and calcium datasets are unreleased, so we will not provide code to model and analyze those datasets. However, we plan to submit data and code to run the synthetic calcium experiments in the supplement. The maze dataset is planned to be publicly released by the experimenters that collected the data before the conference.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] More specific training details will be included in the supplement.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] The AutoLFADS approach applied here uses large-scale hyperparameter optimization, beginning with many networks that are each initialized randomly. In our experience, the approach achieves highly consistent performance for different initializations on a given dataset, so we do not find it necessary to run with multiple random seeds.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Computational resources will be specified in the supplement.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]