LEARNING STRUCTURED NEURAL DYNAMICS FROM SINGLE TRIAL POPULATION RECORDING

Josue Nassar, Scott W. Linderman[‡]*, Yuan Zhao, Mónica Bugallo*[*]*, Il Memming Park*[†]

Stony Brook University, Stony Brook, NY 11794-2350
[‡]Columbia University, New York, NY 10027
{josue.nassar, yuan.zhao, monica.bugallo, memming.park}@stonybrook.edu, [‡]scott.linderman@columbia.edu

ABSTRACT

To understand the complex nonlinear dynamics of neural circuits, we fit a structured state-space model called *tree-structured recurrent switching linear dynamical system* (TrSLDS) to noisy high-dimensional neural time series. TrSLDS is a multi-scale hierarchical generative model for the state-space dynamics where each node of the latent tree captures locally linear dynamics. TrSLDS can be learned efficiently and in a fully Bayesian manner using Gibbs sampling. We showcase TrSLDS' potential of inferring low-dimensional interpretable dynamical systems on a variety of examples.

***Index Terms*—** state-space model, dynamical system, statistical neuroscience, Bayesian inference, population spike trains

I. INTRODUCTION

Statistical neuroscience has been pushing the boundary of what can be inferred bottom-up from neural data. Recent research has suggested that high-dimensional population dynamics of neurons are well approximated by dynamics restricted to a low-dimensional manifold in many cases [1]. Learning the hidden population dynamics is key to understanding the inner workings of neural systems [2].

As higher-dimensional population recordings become available, the question arises of whether we can learn the underlying dynamical law of population dynamics solely based on the recorded neural activities [3], [4]. While modern machine learning techniques have been successful in learning latent dynamical models of neural populations, these techniques often trade interpretability for predictive power [5], [6]. These methods essentially become "black boxes," making it difficult for neuroscientists to decipher their inner workings and understand the computational principles implemented by neural systems. In contrast, linear dynamical systems (LDS) are very interpretable, but sadly, the limited expressive power of LDS can only capture trivial neural dynamics.

To increase the flexibility of the LDS, one strategy is to partition the underlying latent space into multiple regions, each equipped with an LDS. This combination of locally linear dynamics can represent a much richer class of dynamical systems while retaining interpretability. Tree-structured recurrent switching linear dynamical systems (TrSLDS) [7] accomplishes this by partitioning the latent space using tree-structured stick-breaking. TrSLDS leverages a hierarchical LDS prior to enforce structural smoothness of the dynamics. The learned model is a binary tree where each node represents a spatially constrained LDS, providing a multi-scale view of the full dynamics with increasing complexity for deeper levels of the tree allowing the neuroscientist to examine the dynamics at different levels of resolution.

We showcase the expressive power of TrSLDS on two examples with known population dynamics. The first one being a synthetic spike train data where the underlying dynamics is the FitzHugh-Nagumo oscillator [8]. We then fit TrSLDS to the spiking neural network of [9] where the learned effective 2-dimensional dynamics recapitulate the theoretically derived reduction of the high-dimensional spiking neural network model [10].

II. BACKGROUND

Let $x_t \in \mathbb{R}^{d_x}$ and $y_t \in \mathbb{R}^{d_y}$ denote the latent state and the observation of the system at time t respectively. The system can be described using a state-space model:

$$x_t = f(x_{t-1}, w_t; \Theta), \quad w_t \sim \mathrm{F}_w \quad \textit{(state dynamics)} \quad (1)$$
$$y_t = g(x_t, v_t; \Psi), \qquad v_t \sim \mathrm{F}_v \qquad \textit{(observation)} \quad (2)$$

where Θ denotes the dynamics parameters, Ψ denotes the emission (observation) parameters, and w_t and v_t are the state and observation noises respectively. For simplicity, we restrict the state dynamics to be of the form:

$$x_t = f(x_{t-1}; \Theta) + w_t, \quad w_t \sim \mathcal{N}(0, Q). \quad (3)$$

In neuroscience applications, where the observations are spike trains recorded from d_y neurons, the data are multi-

[*]MB was supported by NSF CCF-1617986.
[†]YZ and IMP were supported by NSF IIS-1734910, NIH/NIBIB EB026946, and Thomas Hartman Center for Parkinsons Research (#6424). JN was supported by NSF LSAMP Bridge to the Doctorate (HRD-1612689).

variate binary time series. Thus, we model the observations as

$$y_{j,t} \sim \text{Bernoulli}(\eta_{j,t}), \quad \eta_{j,t} = c_j^\top x_t + b_j, \tag{4}$$

where $c_j \in \mathbb{R}^{d_x}$, and $b_j \in \mathbb{R}, \forall j \in \{1, \ldots, d_y\}$. The goal is to learn a low-dimensional ($d_x \ll d_y$) representation of the system state evolution, represented by $x_{0:T}$, and the underlying dynamical law of the latent state, represented by $f(\cdot; \Theta)$ from the observed spike trains $y_{1:T}$.

When $f(\cdot; \Theta)$ is assumed to be a linear function, the latent states and the parameters can be efficiently learned [11]. While very well researched, the expressive power of an LDS is limited. An alternative is to fit a set of LDSs, which is the approach taken by switching linear dynamical systems (SLDS) [12], [13].

II-A. Switching Linear Dynamical Systems

SLDS approximate nonlinear dynamics by switching between a finite set of LDSs. An additional discrete latent state $z_t \in \{1, \ldots, K\}$ determines the linear dynamics at time t,

$$x_t = x_{t-1} + A_{z_t} x_{t-1} + B_{z_t} + w_t, \quad w_t \sim \mathcal{N}(0, Q_{z_t}) \tag{5}$$

where $A_k, Q_k \in \mathbb{R}^{d_x \times d_x}$ and $B_k \in \mathbb{R}^{d_x}$ for $k = 1, \ldots, K$. Typically, z_t is modeled to have Markovian dynamics. The conditionally linear dynamics allow for fast and efficient inference of the model and can utilize the tools developed for linear systems [14]. The assumption of Markovian dynamics for the discrete latent states severely limits SLDS' generative capacity, preventing it from learning the underlying dynamics [15].

II-B. Recurrent Switching Linear Dynamical Systems

Recurrent switching linear dynamical systems (rSLDS) [15] are an extension of SLDS where the transition density of the discrete latent state depends on the previous location in the continuous latent space

$$z_t | x_{t-1}, \{R, r\} \sim \pi_{SB}\left(\nu_t\right), \quad \nu_t = R x_{t-1} + r, \tag{6}$$

where $R \in \mathbb{R}^{(K-1) \times d_x}$ and $r \in \mathbb{R}^{K-1}$ represents hyperplanes. $\pi_{SB} : \mathbb{R}^{K-1} \to [0, 1]^K$ maps from the reals to the probability simplex via stick-breaking:

$$\pi_{SB}(\nu) = \left(\pi_{SB}^{(1)}(\nu), \cdots, \pi_{SB}^{(K)}(\nu)\right), \tag{7}$$

$$\pi_{SB}^{(k)} = \sigma(\nu_k) \prod_{j<k} \sigma\left(-\nu_j\right), \tag{8}$$

for $k = 1, \ldots, K-1$ and $\pi_{SB}^{(K)} = \prod_{k=1}^{K-1} \sigma\left(-\nu_k\right)$ where ν_k is the kth component of ν and $\sigma(\nu) = (1 + e^{-\nu})^{-1}$ is the logistic function. By including this recurrence in the transition density of z_t, the rSLDS partitions the latent space into K sections, where each section follows its own linear dynamics. It is through this combination of locally linear dynamical systems that the rSLDS approximates (3); the partitioning of the space allows for a more interpretable visualization of the underlying dynamics. While rSLDS can effectively learn the underlying dynamics, its dependence on stick-breaking poses problems for learning.

III. TREE-STRUCTURED RECURRENT SWITCHING LINEAR DYNAMICAL SYSTEMS

TrSLDS [7] is an extension of rSLDS which, like rSLDS, also uses conditionally linear dynamics but uses tree-structured stick-breaking instead of sequential stick-breaking. TrSLDS also imposes a hierarchical prior over the per node LDS, allowing for a multi-scale view. In the next two sections, we briefly explain the tree-structured stick-breaking and the hierarchical prior (see [7] for more details).

III-A. Tree-Structured Stick-Breaking

Let $\mathcal{T}$ be a balanced binary tree with nodes $\{\epsilon, 1, \cdots, N\}$. Each node n has a parent node denoted by $\text{par}(n)$ with the exception of the root node, ϵ, which has no parent. Every internal node n is the parent of two children, $\text{left}(n)$ and $\text{right}(n)$. Let $\text{child}(n) = \{\text{left}(n), \text{right}(n)\}$ denote the set of children for internal node n. Let $\mathcal{Z} \subseteq \mathcal{T}$ denote the set of leaf nodes, which have no children. Let $\text{depth}(n)$ denote the depth of a node n in the tree, with $\text{depth}(\epsilon) = 0$.

At time instant t, z_t is chosen by starting at the root node and traversing down the tree until one of the K leaf nodes are reached. The traversal is done through a sequence of left/right choices by the internal node where the choices as modeled as random variables; the traversal can be viewed as a stick breaking process. We start at the root node with a unit-length stick $\pi_\epsilon = 1$, which we divide between its two children. The left child receives a fraction $\pi_{\text{left}(\epsilon)} = \sigma(\nu_\epsilon)$ and the right child receives the remainder $\pi_{\text{right}(\epsilon)} = 1 - \sigma(\nu_\epsilon)$ such that $\nu_\epsilon \in \mathbb{R}$ specifies the left/right balance. This process is repeated recursively, subdividing π_n into two pieces at each internal node until we reach the leaves of the tree. The stick assigned to each node is thus,

$$\pi_n = \begin{cases} \sigma(\nu_{\text{par}(n)}), & n = \text{left}(\text{par}(n)) \text{ and } n \neq \epsilon, \\ 1 - \sigma(\nu_{\text{par}(n)}), & n = \text{right}(\text{par}(n)) \text{ and } n \neq \epsilon, \\ 1, & n = \epsilon. \end{cases}$$

We incorporate this into the TrSLDS by allowing ν_n to be a function of the continuous latent state

$$\nu_n(x_{t-1}, R_n, r_n) = R_n^T x_{t-1} + r_n, \tag{9}$$

where R_n and r_n specify a linear hyperplane in the continuous latent state space. As the continuous latent state x_{t-1} evolves, the left/right choices become more or less probable. This in turn changes the probability distribution $\pi_k(x_{t-1}, \Gamma, \mathcal{T})$ over the K leaf nodes, where $\Gamma = \{R_n, r_n\}_{n \in \mathcal{T}}$. In the TrSLDS, these leaf nodes correspond to the discrete latent states of the model, such that for each leaf node k,

$$p\left(z_t = k \mid x_{t-1}, \Gamma, \mathcal{T}\right) = \pi_k(x_{t-1}, \Gamma, \mathcal{T}). \tag{10}$$

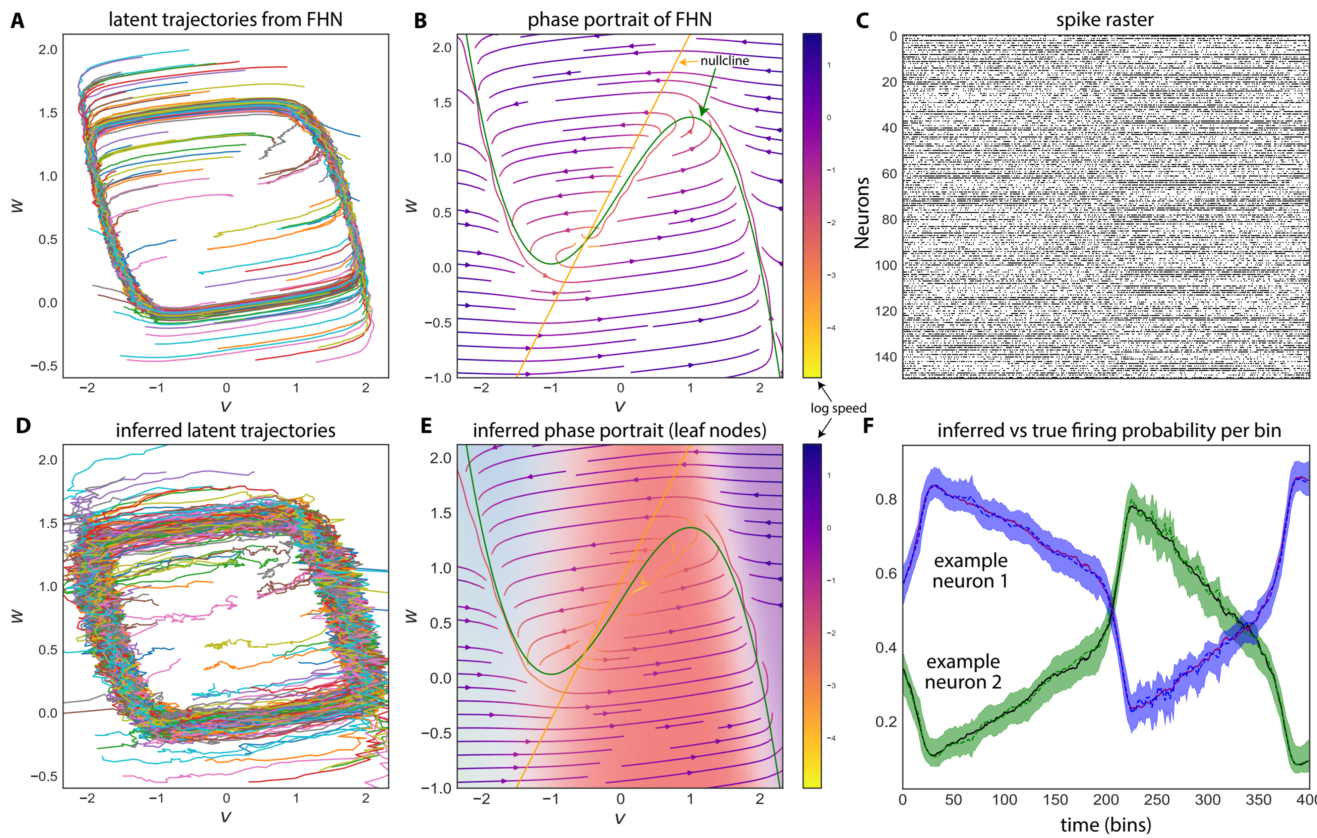

Fig. 1. **(A, D)** TrSLDS is able to infer the latent states. **(B, E)** True and learned phase portraits where the green and yellow lines are the nullclines of the system. The background color showcases the partitioning learned by the model where the darker the color is, the higher the probability of ending up in that discrete state. **(C)** Raster plot of a single trajectory. **(F)** The inferred firing rate (dotted line) and the true firing rate (solid line) are displayed for two neurons. Error bars denote $\pm$ 3 standard deviations under the posterior.

III-B. Hierarchical Dynamics Prior

To obtain a multi-scale view of the dynamics, TrSLDS leverages a hierarchical LDS prior to enforce structural smoothness of the dynamics.

Let $\{A_n, B_n\}$ be the dynamics parameters associated with node n. These internal dynamics serve as a link between the leaf node dynamics via a hierarchical prior,

$$\{A_n, B_n\} | \{A_{\text{par}(n)}, B_{\text{par}(n)}\} \sim \mathcal{N}(\{A_{\text{par}(n)}, B_{\text{par}(n)}\}, \Sigma_n).$$

The prior on the root node is $\{A_\epsilon, b_\epsilon\} \sim \mathcal{N}(0, \Sigma_\epsilon)$.

It is through this hierarchical tree-structured prior that TrSLDS obtains a multi-scale view of the system. Parents are given the task of learning a higher level description of the dynamics over a larger region while children are tasked with learning the nuances of the dynamics. The use of hierarchical priors also allows for neighboring sections of latent space to share common underlying dynamics inherited from their parent. Once fit, TrSLDS can be queried at different levels, where levels deeper in the tree provide more resolution.

III-C. Bayesian Inference

The linear dynamic matrices Θ, the hyperplanes Γ, the emission parameters Ψ, the continuous latent states $x_{0:T}$ and the discrete latent states $z_{1:T}$ must be inferred from the data. Under the Bayesian framework, this corresponds to computing the posterior,

$$p\left(x_{0:T}, z_{0:T}, \Theta, \Psi, \Gamma | y_{1:T}\right) = \frac{p\left(x_{0:T}, z_{1:T}, \Theta, \Psi, \Gamma, y_{1:T}\right)}{p\left(y_{1:T}\right)}. \tag{11}$$

The parameters and latent states are learned via Gibbs sampling [16] to obtain samples from the posterior distribution described in (11). To allow for fast and closed form conditional posteriors, we augment the model with Pólya-gamma auxiliary variables [17]; details of the sampling can be found in [7].

IV. EXPERIMENTS

We demonstrate the potential of TrSLDS by testing it on a couple of example systems. The first, FitzHugh-Nagumo (FHN), is a two-dimensional nonlinear oscillation system.

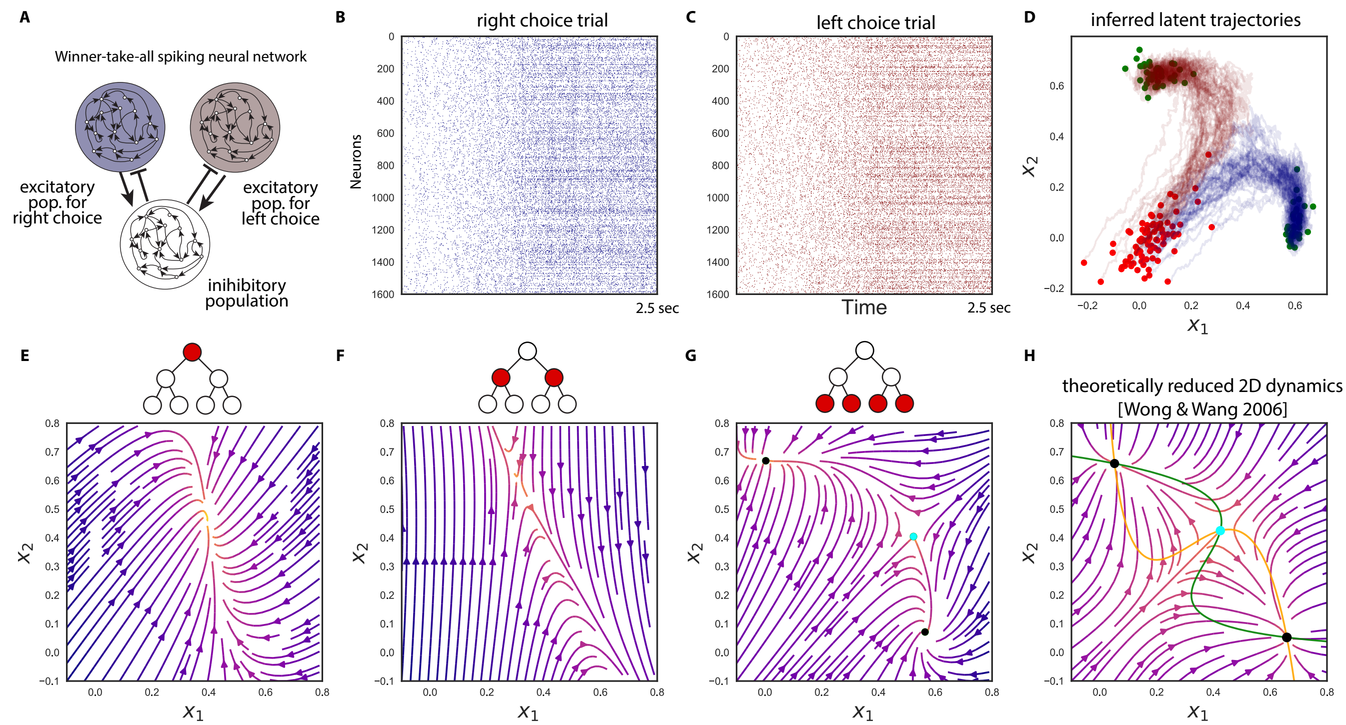

Fig. 2. TrSLDS on Winner-take-all spiking neural network. (**A**) Overview of the connectivity structure of the spiking neural network [9]. (**B,C**) raster plots of excitatory neurons for 2 random trials. (**D**) the latent trajectories converge to either one of the two of sinks at the end of trial (green). Each trajectory is colored by their final choice. (**E-F**) Dynamics captured by each level provide a multi-scale view. **(H)** Manual 2-dimensional reduction of the spiking neural network dynamics given the full specification [10].

Synthetic spike trains are generated from a Bernoulli model, as described in (4), to showcase that TrSLDS can indeed learn the low-dimensional latent states and the underlying dynamics from high-dimensional spike trains through the Pólya-gamma augumentation of the observation model. The next example is the winner-take-all spiking neural network of [9] where it has been shown that the population dynamics follow a two-dimensional dynamical system [10].

IV-A. FitzHugh-Nagumo with Bernoulli observations

The FHN model is a 2-dimensional reduction of the Hodgkin-Huxley model which is completely described by the following system of differential equations [8]:

$$\dot{v} = v - \frac{v^3}{3} - w + I_{ext}, \qquad \tau \dot{w} = v + a - bw. \qquad (12)$$

We set the parameters to $a = 0.7$, $b = 0.8$, $\tau = 12.5$, and $I_{ext} \sim \mathcal{N}(0.7, 0.04)$. The model was trained on 100 trajectories where each trajectory consisted of 400 data points with 150 Bernoulli neurons. We set the number of leaf nodes to be 4 and ran Gibbs for 500 samples.

Fig. 1 displays the power of TrSLDS. From the synthetic spike trains, TrSLDS is able to infer the latent continuous states. It also accurately predicts the firing rates of the neurons in the system. The phase portrait learned by TrSLDS is an accurate approximation of the true phase portrait, solidifying TrSLDS' ability to generate data that is similar to the true system.

IV-B. Winner-Take-All Spiking Neural Network

Next, we fit TrSLDS to the winner-take-all spiking neural network [9]. The network is comprised of 2000 neurons, of which 1600 are excitatory and the rest are inhibitory. Among the excitatory neurons, 480 are selective to the binary choice. In [10], they theoretically reduced the spiking neural network to a system of 2 differential equations. The corresponding vector field in Fig. 2 show that the system is composed of 3 fixed points; 2 sinks corresponding to the possible choices and an unstable node between the them. We trained TrSLDS on 80 trajectories from 150 subsampled tunned excitatory neurons from the simulated spiking neural network. We set the number of leaf nodes to be 4 and ran Gibbs for 250 iterations.

From Fig. 2, we see the multi-scale view TrSLDS obtains of the underlying dynamics. The root node corresponds to an LDS, which does not have much expressive power (we note that the root node demonstrates what a Poisson linear dynamical system (PLDS) [11] would learn if fitted to this data). Traversing one level deeper into the tree, the model learns the unstable node present between the two sinks (represented by the black dots in Fig. 2H). The leaf nodes refine this, and it is evident from Fig. 2G that TrSLDS is able to learn the dynamics of the underlying model. Fig. 2D showcase that not only is TrSLDS able to learn the latent states but also strengthens the idea that high-dimensional spike train can be represented through a low-dimensional representation.

V. CONCLUSION

We showed that TrSLDS can learn a low-dimensional dynamical system from single-trial population neural data. The learned system is explicitly expressed as a combination of linear dynamical systems, thus increasing interpretability. Moreover, the tree-structure provides a multi-scale view that naturally gives the neruoscientist the option to control the granularity of viewing and understanding. Our approach is distinguished from the mainstream approach in low-dimensional neural trajectory learning where averages are used instead of single-trial data. The averaging is only possible for tightly repeatable trial-based tasks, and even then, the neural trial-to-trial variability would be completely ignored. In contrast, our approach (not unique, but in minority) focuses on learning the dynamical law that drives the single-trial trajectories. This allows us to learn the hidden population dynamics in a data-driven fashion without the assumptions required for trial-averaging.

REFERENCES

[1] Matthew D Golub, Patrick T Sadtler, Emily R Oby, Kristin M Quick, Stephen I Ryu, Elizabeth C Tyler-Kabara, Aaron P Batista, Steven M Chase, and Byron M Yu, "Learning by neural reassociation," *Nat. Neurosci.*, vol. 21, no. 4, pp. 607–616, Apr. 2018.

[2] Abigail A Russo, Sean R Bittner, Sean M Perkins, Jeffrey S Seely, Brian M London, Antonio H Lara, Andrew Miri, Najja J Marshall, Adam Kohn, Thomas M Jessell, Laurence F Abbott, John P Cunningham, and Mark M Churchland, "Motor cortex embeds muscle-like commands in an untangled population response," *Neuron*, vol. 97, no. 4, pp. 953–966.e8, Feb. 2018.

[3] Yuan Zhao and Il Memming Park, "Interpretable nonlinear dynamic modeling of neural trajectories," in *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[4] Scott W Linderman and Samuel J Gershman, "Using computational theory to constrain statistical models of neural data," *Curr. Opin. Neurobiol.*, vol. 46, pp. 14–24, Oct. 2017.

[5] Chethan Pandarinath, Daniel J OShea, 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*, p. 1, 2018.

[6] Roger Frigola, Yutian Chen, and Carl Edward Rasmussen, "Variational Gaussian Process State-Space Models," in *Advances in Neural Information Processing Systems 27*, Z Ghahramani, M Welling, C Cortes, N D Lawrence, and K Q Weinberger, Eds., pp. 3680–3688. Curran Associates, Inc., 2014.

[7] Josue Nassar, Scott W. Linderman, Monica Bugallo, and Il Memming Park, "Tree-structured recurrent switching linear dynamical systems for Multi- Scale Modeling," *ArXiv e-prints*, p. arXiv:1811.12386, Nov. 2018.

[8] Eugene M Izhikevich, *Dynamical Systems in Neuroscience*, MIT press, 2007.

[9] Xiao-Jing Wang, "Probabilistic decision making by slow reverberation in cortical circuits," *Neuron*, vol. 36, no. 5, pp. 955–968, 2002.

[10] Kong-Fatt Wong and Xiao-Jing Wang, "A recurrent network mechanism of time integration in perceptual decisions," *Journal of Neuroscience*, vol. 26, no. 4, pp. 1314–1328, 2006.

[11] Jakob H Macke, Lars Buesing, John P Cunningham, M Yu Byron, Krishna V Shenoy, and Maneesh Sahani, "Empirical models of spiking in neural populations," in *Advances in Neural Nnformation Processing Systems*, 2011, pp. 1350–1358.

[12] Chaw-Bing Chang and Michael Athans, "State estimation for discrete systems with switching parameters," *IEEE Transactions on Aerospace and Electronic Systems*, , no. 3, pp. 418–425, 1978.

[13] Zoubin Ghahramani and Geoffrey E Hinton, "Switching state-space models," Tech. Rep., University of Toronto, 1996.

[14] Simon S Haykin, *Kalman Filtering and Neural Networks*, John Wiley & Sons, Inc., New York, NY, USA, 2001.

[15] Scott Linderman, Matthew Johnson, Andrew Miller, Ryan Adams, David Blei, and Liam Paninski, "Bayesian Learning and Inference in Recurrent Switching Linear Dynamical Systems," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, Aarti Singh and Jerry Zhu, Eds., Fort Lauderdale, FL, USA, 9 2017, vol. 54 of *Proceedings of Machine Learning Research*, pp. 914–922, PMLR.

[16] Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng, *Handbook of Markov Chain Monte Carlo*, CRC press, 2011.

[17] Nicholas G Polson, James G Scott, and Jesse Windle, "Bayesian Inference for Logistic Models Using Pólya-Gamma Latent Variables," *Journal of the American Statistical Association*, vol. 108, no. 504, pp. 1339–1349, 2013.