Make SLDS great again

Khanh-Tung Nguyen-Ba
kn2465@columbia.edu

Abstract

We investigated a solution to improve the performance of the Switching Linear Dynamic System (SLDS) model and its multipopulation extension by using a sequential autoencoder to approximate the motor cortex dynamics and generate denoised firing rates from spike data.

1. Introduction

Recent advancement in recording techniques has posed exciting challenges for neural data scientists to catch up as data has become larger in terms of both size and complexity. Novel modeling methods have been introduced, further insights have been gained, but there has never been a lack of room for improvement.

The Switching Linear Dynamic System (SLDS) [3] model has been shown to be able to capture discrete states that juxtapose well against behavior annotations. A recent multi-population extension of SLDS [4] also allows us to model dynamics between different brain populations. However, these models seem to be prone to give undesirable results when spike data input is noisy as shown in our experiments. Our hypothesis is that SLDS still has some limitation in approximating the internal brain dynamics that it mistook noise for signal, and so we seek out a denoiser that can better capture these dynamics.

Latent Factor Analysis via Dynamical Systems (LFADS) [5] is a sequential extension of the variational autoencoder that is known to be able to infer neural firing rates and latent dynamics well. In this report, we showed that using LFADS as a denoiser has a significant impact on the ability of SLDS to infer behaviorally-relevant latent states.

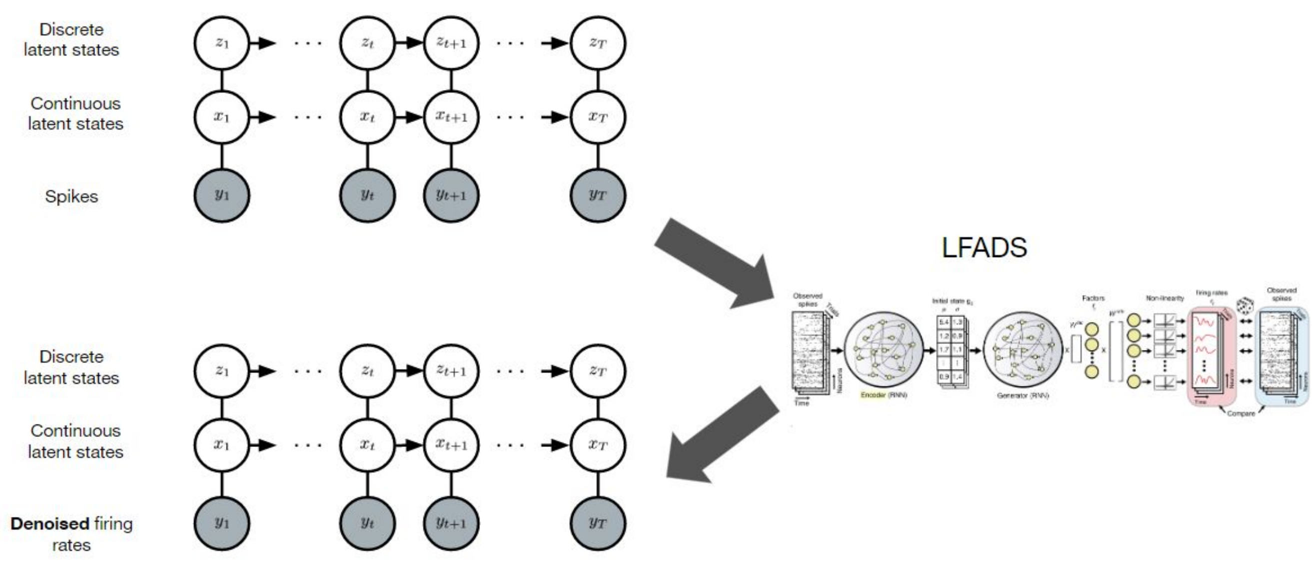


Figure 1: Using LFADS to infer firing rates that are input to SLDS

2. Dataset

The dataset from a sequential reaching task [1,2] includes extracellular recordings from 90 neurons in premotor cortex (PMd), 67 neurons from primary motor cortex (M1) and behavior of a macaque. There are 496 reaches, in each reach the monkey controlled an on-screen cursor and was rewarded for moving the cursor to a reach target. Behavioral data include position, velocity and acceleration. The time bin is 10ms.

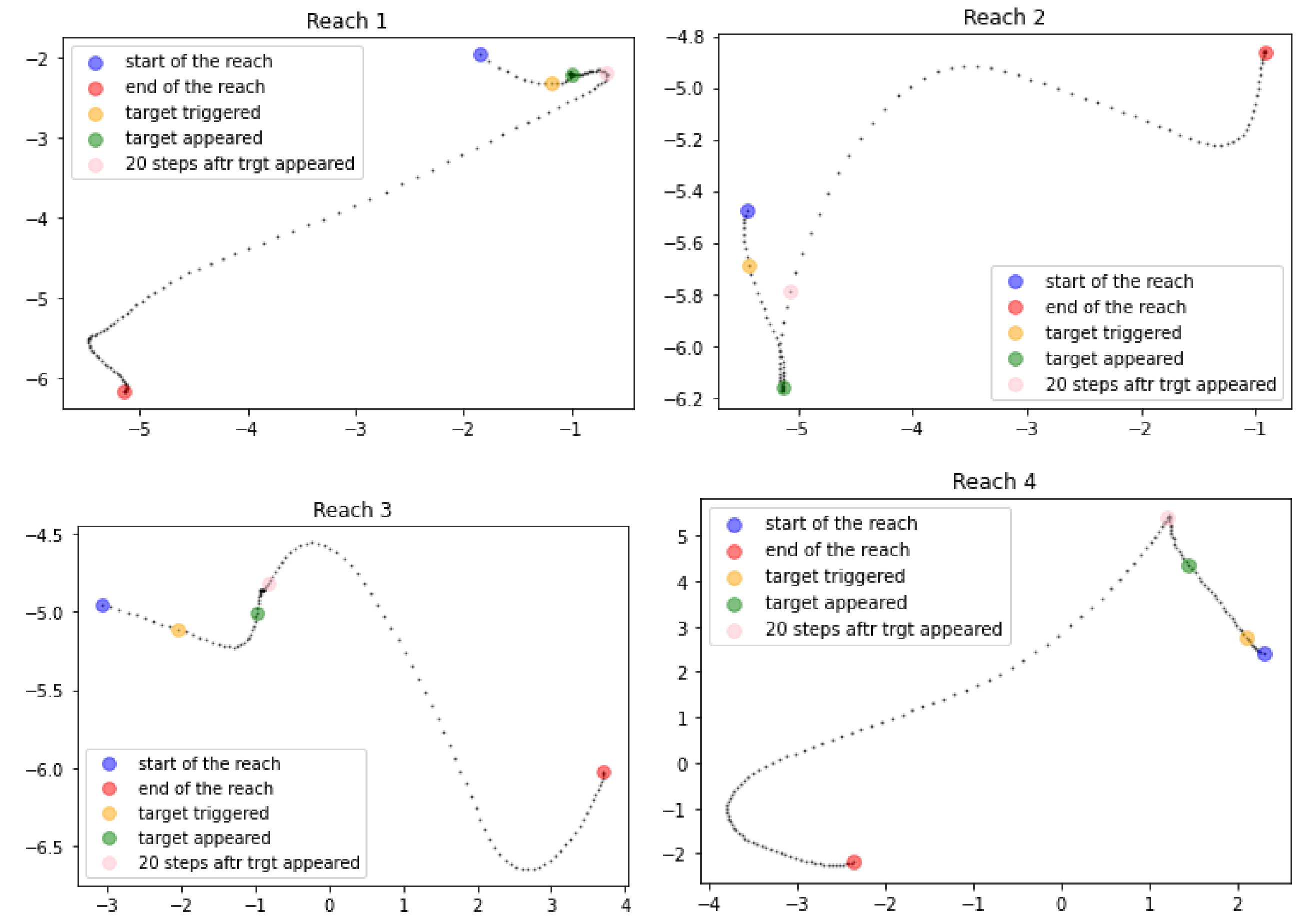


Figure 2: Visualization of the first 4 reaches and behavioral annotations

3. The challenge

Results from running SLDS on each neural population shows either ill-defined states with highly frequent switching on PMd, or lack of any discernible states on M1. For details on the model set-up, please refer to Table 1 in the Appendix.

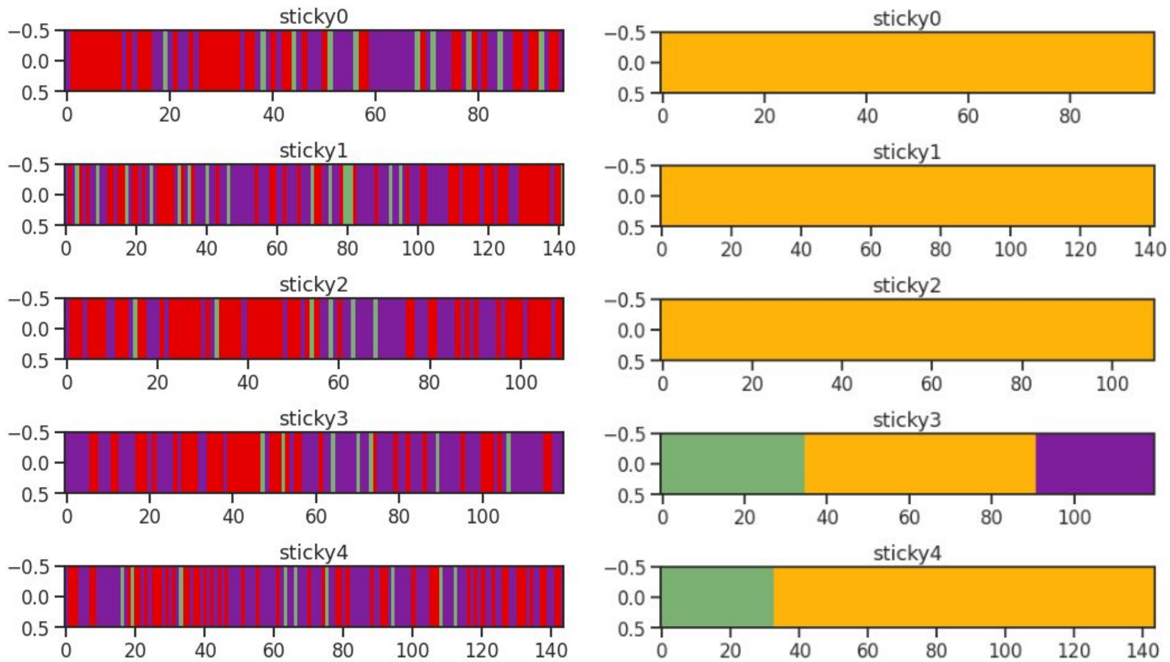
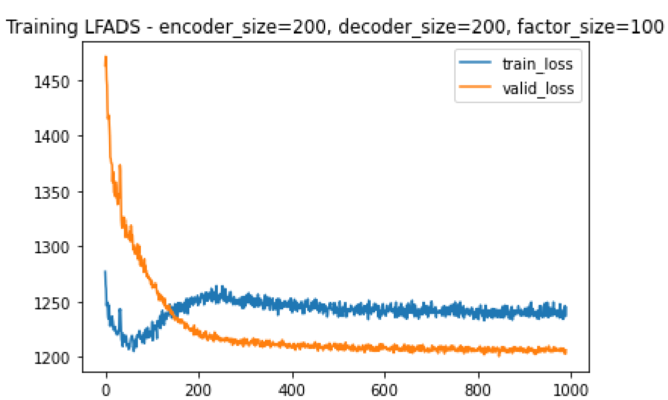


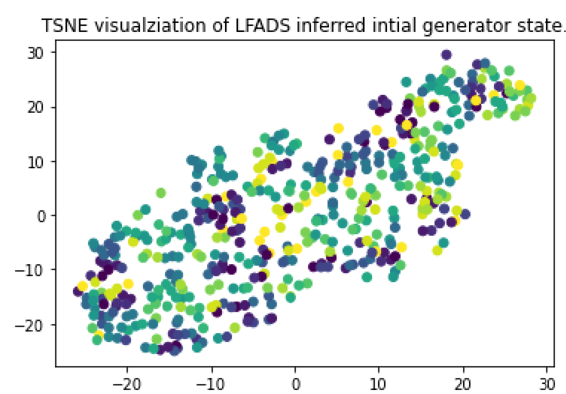
Figure 2: SLDS results on spike data from PMd (left) and M1 (right). Result from 5 different trials.

4. Training LFADS

We split 496 reaches into a training set (90%) and a validation set (10%). Our empirical LFADS set-up was trained until convergence after 1000 epochs. For details on the set-up of the LFADS model, please see Table 2 in the Appendix.

To test whether the initial state vector **g_0** was able to capture any behaviorally relevant structure, following the same analysis in [5], we applied the non-linear dimensionality reduction technique t-distributed stochastic neighbor embedding to reduce the dimensionality of the initial state vectors to 2. Each point represented one reach color-coded by the angle of the target of the upcoming reach. The result was bad but can be explained by the nature of the experiment. Since there was no holding period, the money could start moving the cursor as soon as the target was on, the condition-invariant component that reflected a chosen movement [6] was a relatively transient signal. When we empirically chose the 200ms period after the target was on (it can be seen from Figure 2 that the movement onset was around the pink point) and train LFADS only on this period, we received much better clustering results.





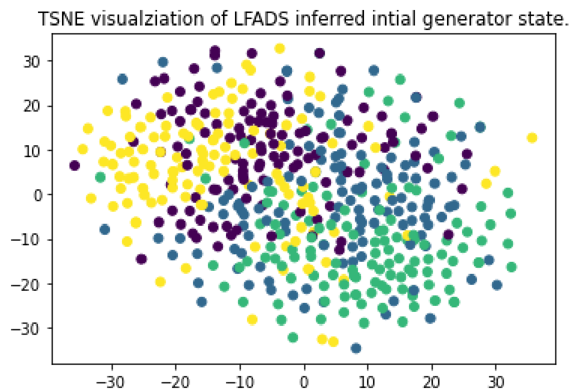
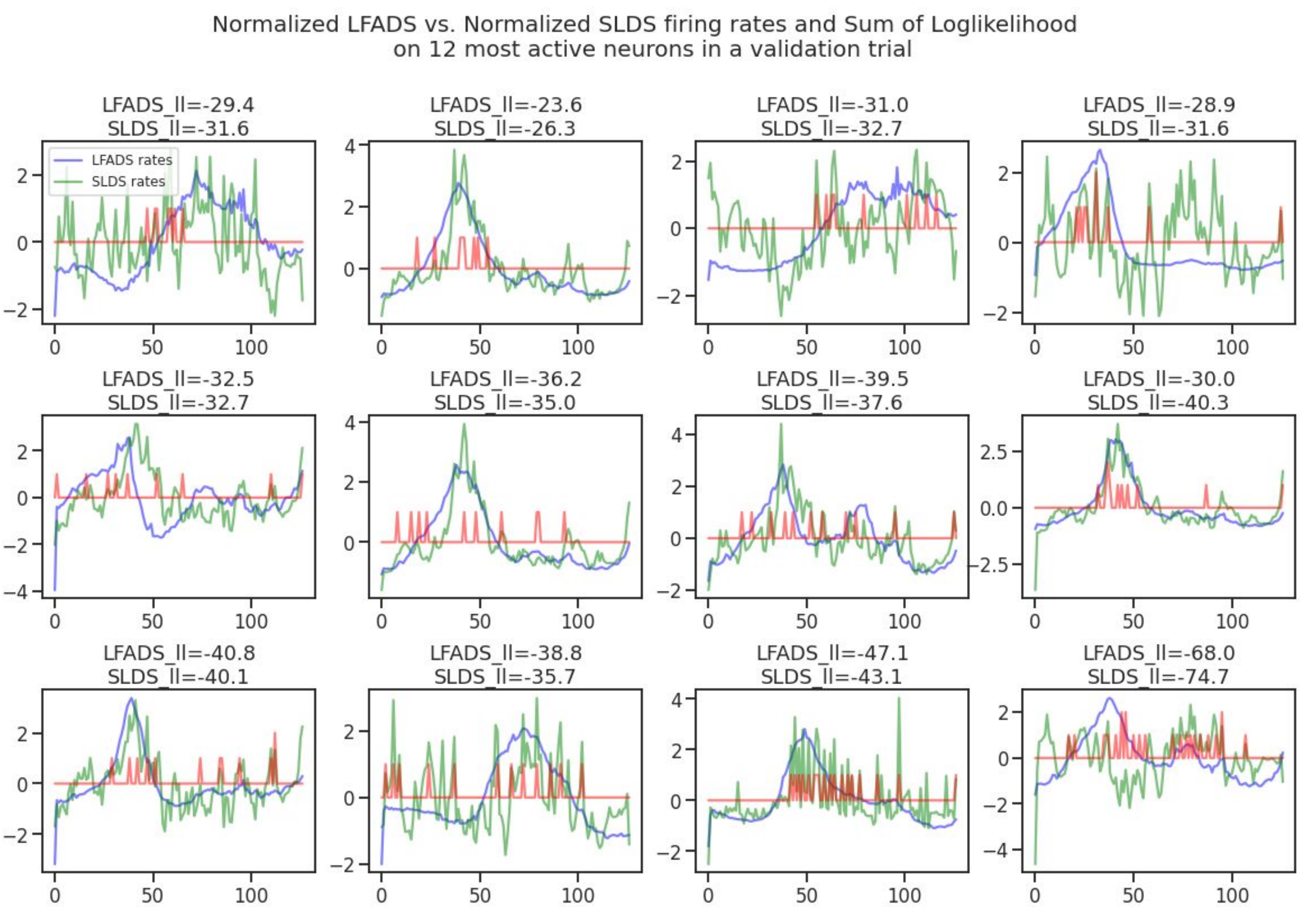


Figure 3: Train and validation loss result from training LFADS (Top Left),
2D-projection of the initial states of LFADS showed bad clustering result (Top Right),
Improved clustering result when training LFADS on only 200ms after target is on (Bottom)

5. Results

5.1. LFADS-inferred firing rates validation

For validation, we used the trained LFADS model to infer firing rates in the validation trials, and then compared those against the SLDS firing rates and spike data. SLDS firing rates are visually choppier, and do not necessarily fit the spike data better. The sum of log likelihood are indeed higher for LFADS on 7 of the 12 most active neurons in a sample trial as shown in Figure 4. Similar results can be seen for the 12 least active neurons in the same trial.



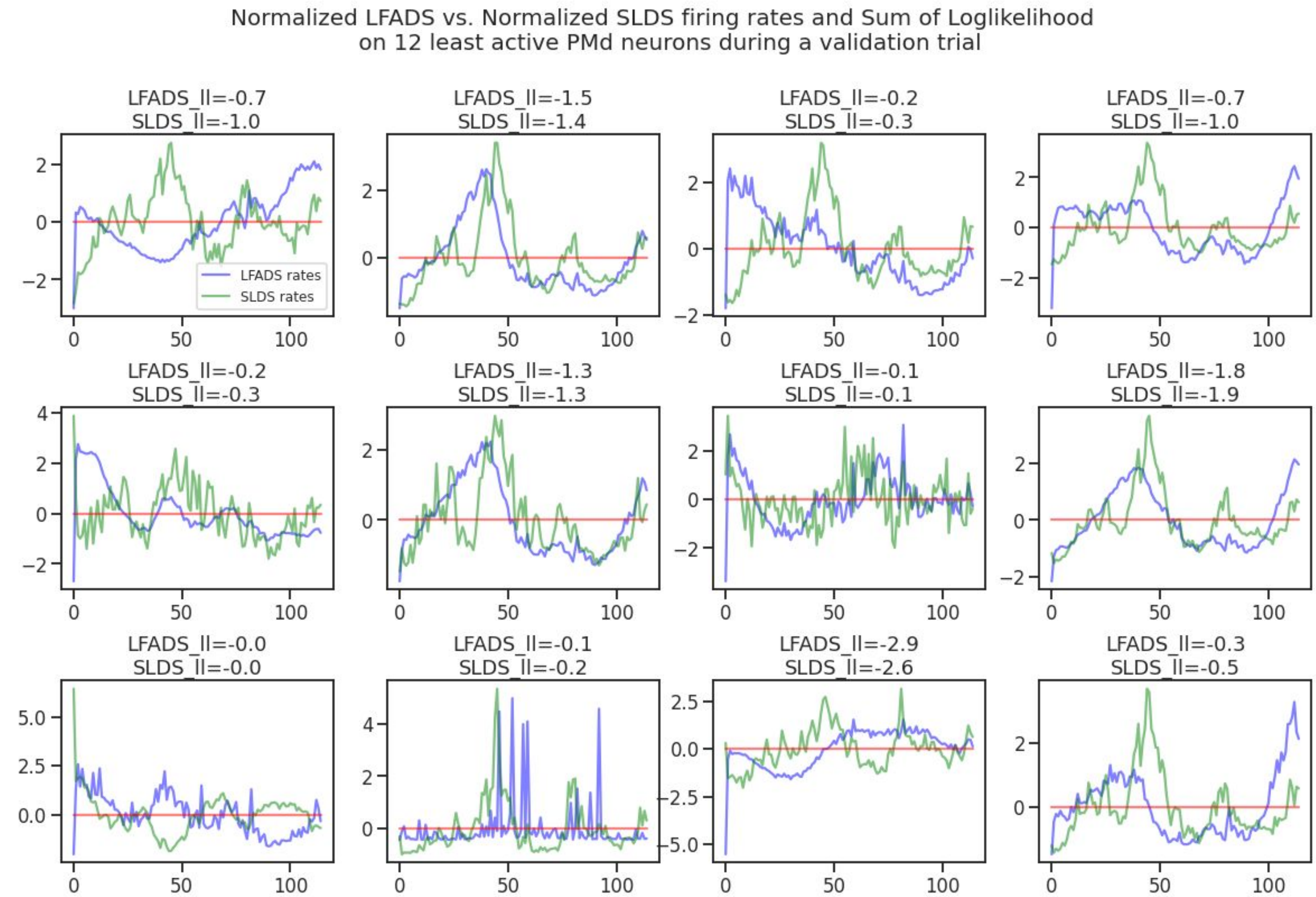
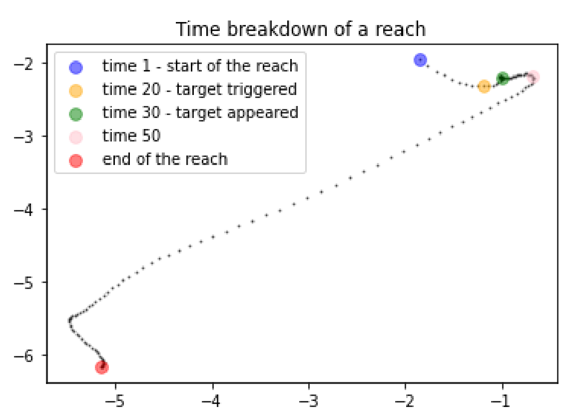


Figure 4: Normalized LFADS vs Normalized SLDS firing rates for 12 most active PMd neurons in a validation trial (Top), Normalized LFADS vs Normalized SLDS firing rates for 12 least active PMd neurons in a validation trial (Bottom)

5.2 SLDS result

Finally, we applied SLDS models on the LFADS-inferred firing rates, and observed significant improvement on both the single-population and multi-population models. The inferred discrete states also aligned well against behavioral data. As shown in Figure 5 Top, the trials are aligned so that at the start of each trial, the monkey was reaching the previous trial's target. Once the cursor got close enough to the previous target at time 20, the new target was triggered (not shown), and then appeared at time 30. The monkey could start moving to the new target right at time 30, we see that it usually took 100-200ms (10-20 time steps) before the cursor started moving to the target's direction.



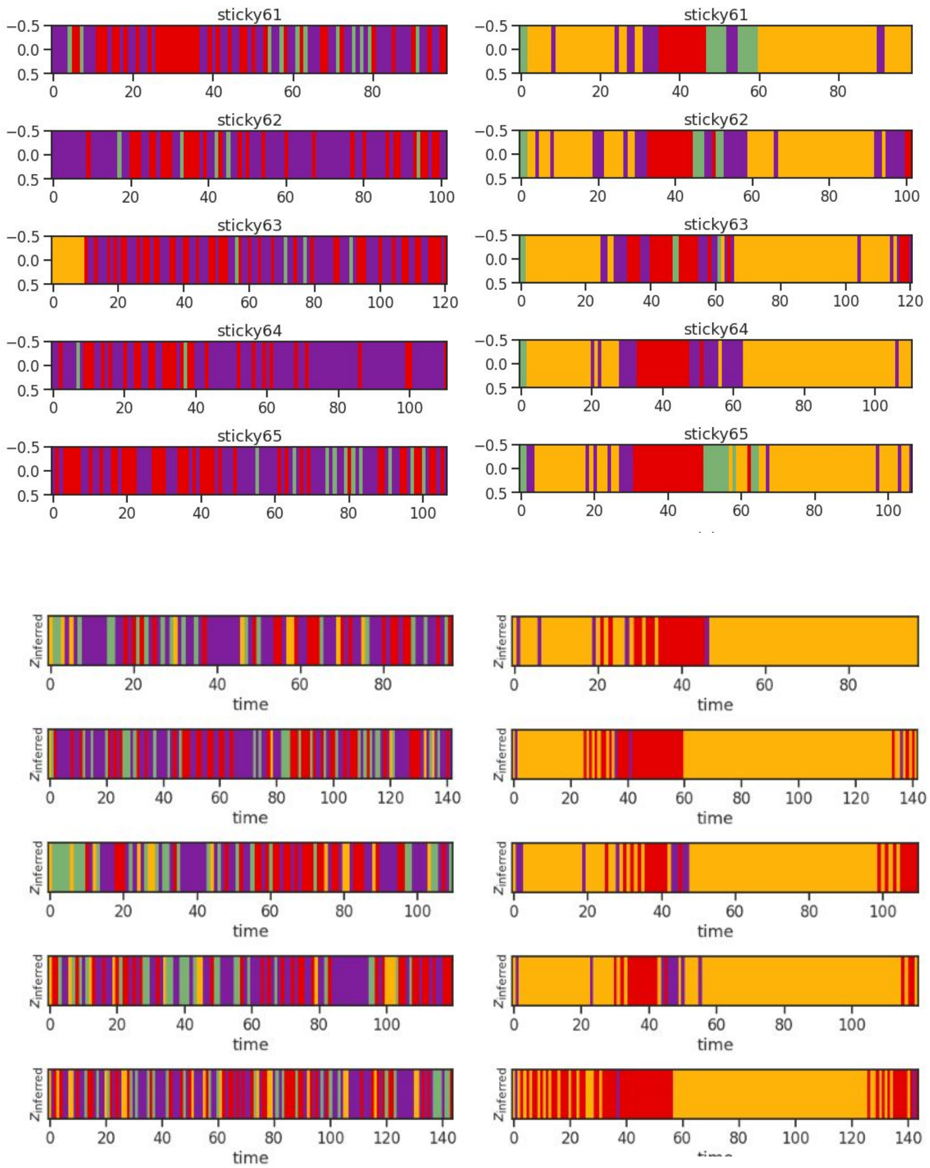


Figure 4: Time breakdown of a reach (Top)
SLDS result before and after using LFADS firing rates from PMd (Middle Left and Right),
mprSLDS result before and after using LFADS firing rates from PMd and M1 (Bottom Left and Right).
Results from 5 different trials.

6. Conclusion

In this report, we showed that using LFADS as a denoiser has a significant impact on how well the SLDS models can infer behaviorally relevant discrete states. Apparently, the SLDS model still has some limitation in approximating the motor cortex dynamics that it allows too much noise to be captured. We can think of LFADS as an external extension to make SLDS more robust, but one future direction is to extend the model internally.

Another direction for future research is how to set up LFADS in a methodological way. This will alleviate the considerable pain in handcrafting the right elements for the model to work.

7. Acknowledgement

We are very grateful for Professor Liam Paninski's guidance throughout the project.

8. References

[1] Lawlor, P.N., Perich, M.G., Miller, L., Kording, K.P. Linear-Nonlinear-Time-WarpPoisson models of neural activity. J Comput Neurosci (2018). https://doi.org/10.1007/s10827-018-0696-6
[2] Matthew G. Perich, Patrick N. Lawlor, Konrad P. Kording, Lee E. Miller (2018). Extracellular neural recordings from macaque primary and dorsal premotor motor cortex during a sequential reaching task. . CRCNS.org. http://dx.doi.org/10.6080/ K0FT8J72
[3] Linderman, S. W., Johnson, M. J., Miller, A. C., Adams David M Blei Liam Paninski Harvard, R. P., & Brain, G. (n.d.). Bayesian Learning and Inference in Recurrent Switching Linear Dynamical Systems.
[4] Glaser, J. I., Whiteway, M., Cunningham, J. P., Paninski, L., & Linderman, S. (2020). Recurrent Switching Dynamical Systems Models for Multiple Interacting Neural Populations. NeurIPS, 1–11.
[5] Sussillo, D., Jozefowicz, R., Abbott, L. F., & Pandarinath, C. (2016). LFADS - Latent Factor Analysis via Dynamical Systems. http://arxiv.org/abs/1608.06315
[6] Kaufman, M. T., Seely, J. S., Sussillo, D., Ryu, S. I., Shenoy, K. V., & Churchland, M. M. (2016). The largest response component in the motor cortex reflects movement timing but not movement type. ENeuro, 3(4). https://doi.org/10.1523/ENEURO.0085-16.2016

9. Appendix

Table 1 - SLDS set up

dynamics	gaussian
emissions	poisson before, gaussian after denoising
method	laplace_em
variational_posterior	structured_meanfield
transitions	sticky

Table 2 - LFADS set up

encoder_dim	200
decoder_dim	200
factor_dim	20
learning_rate	0.01
learning_rate_decay	0.95