Using Latent Variable Models to Predict Mouse Behavior

Saket Arora
University of California, San Diego
La Jolla, CA 92093
s2arora@ucsd.edu

Aryan Singh
University of California, San Diego
La Jolla, CA 92093
ars001@ucsd.edu

Jad Makki
University of California, San Diego
La Jolla, CA 92093
jmakki@ucsd.edu

Rishabh Viswanathan
University of California, San Diego
La Jolla, CA 92093
rviswana@ucsd.edu

Abstract

Dimensionality reduction, typically applied to data in order to make it more usable and generally easier to model, is often used to understand neural spike data, which is typically very complex due to its high-dimensional nature. Through latent factor modeling, we can effectively apply dimensionality reduction to this data and create lower dimensional representations of it. With data compiled from the International Brain Lab, we applied a Variational Latent Gaussian Process model (vLGP) in order to extract latent trajectories from the data collected within the motor cortex. These trajectories are separated into four distinct categories, representing the activity from the designated brain region when the mouse moves the wheel in one direction correctly in response to the stimulus, or when the mouse moves the wheel in the incorrect direction. These trajectories reveal linear separability between the correct decisions in different directions (when the mouse turns the wheel clockwise correctly versus when the mouse turns the wheel counter-clockwise correctly). Comparing this model to Gaussian Process Factor Analysis (GPFA), another dimensionality reduction technique - reveals that vlGP produces cleaner and more interpretable trajectories while also being more scalable to larger amounts of data without running into memory issues.

1 Introduction

The brain is an intricate and complex system that deals with hundreds of thousands of simultaneous processes at the same time. In order to truly understand the brain we must carefully observe it, knowing that we only have access to input stimuli and output of the neurons we are observing. The actual calculations themselves, computed by the "Neural Network" of the brain, are unseen by us, and thus must be inferred to gain a clearer understanding of what the brain is truly doing.

The reality of the situation is that there are up to billions of neurons in the brain. While we cannot observe them all, the ones we can observe yield data in very high dimensions. This is where the importance of dimensionality reduction comes in. There are two main reasons why reducing the dimensionality is extremely useful when dealing with neural data. The first, most obvious reason is that a smaller and less noisy subspace is simply much easier to understand and internalize than the raw data in huge dimensions. Secondly, dimensionality reduction is a key tool in understanding the true nature of the neural circuit and its computations as it allows us to describe and simplify a set of complex variables in order to explore the unobserved, lower dimensional, variables that may explain

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

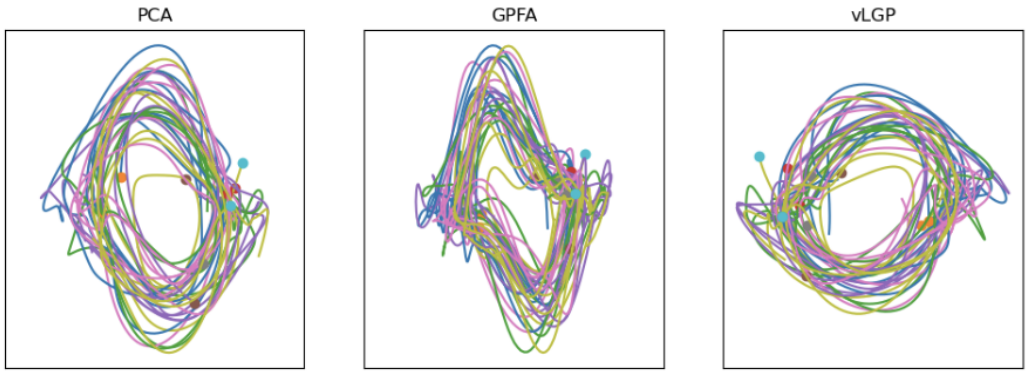
their relation. Last quarter we focused on gathering the necessary tools in order to create models and conduct meaningful analysis from neural data, by developing factor analysis and Gaussian Process Factor analysis models for use on dummy data.

In this project, we aim to apply these methods, including an additional method, Variational Latent Gaussian Process, on real-world neuron data, collected from the International Brain Lab, in order to find relationships between mouse behavior and neuron activity.

1.1 Literature Review and Prior Work

Due to the nature of the problem, many studies have been done on dimensionality reduction techniques and methodologies for analyzing neural data. Yu et al. [2] explain that "neural responses are a reflection of internal processing rather than external stimuli drive", and that in behavioral tasks that involve perception, decision making, etc. are best analyzed on a trial-by-trial basis, rather than averaging neural data across trials. They introduce the GPFA model, which "unifies smoothing and dimensionality reduction operations in a common, probabilistic framework. Zhao, Yuan, and Memming [3] take this one step further, explaining that compared to previous methods such as GPFA, vLGP achieves a substantially higher performance for predicting omitted spike trains. Compared to previous methods, vLGP is faster and yields better predictability at a fine time scale to reveal hidden neural dynamics from large-scale neural recordings.

After our work last quarter, we realized that GPFA would have some shortcomings when used on the IBL data. Since the dataset is more closely aligned with a Poission distribution, the Gaussian observations from the GPFA model were not optimal for reducing dimensionality in the neural data we were feeding it. Namely, we started running into issues with scalability, with GPFA unable to fit or infer if given all the data at once. Thus, we switched to Variational Latent Gaussian Processes, which aim to remedy this problem by using Poisson observations, which operate much better in small(millisecond) time ranges. The following figure shows the difference in calculated trajectories between PCA, GPFA, and vLGP on dummy data:



1.2 Dataset

The data analyzed were collected by the International Brain Lab, which aimed to produce standardized and reproducible measurements of decision-making in mice [1]. Their experiment involved mice trained in a task to move a vertical grating, which randomly appeared at $-35°$, $-0°$, or $35°$ azimuth, to the center of the screen using a wheel. During this task, Neuropixel probes in the mouse's brain collect electrophysiology measurements. See figure 1 for the experimental setup.

The position of the probes was determined such that they cover all major brain areas of the left hemisphere, and the right cerebellum, since connections between the cerebrum and cerebellum cross hemispheres. See figure 2 for the probe locations. The raw data consists of voltage potentials across 384 channels over time. Action potentials, referred to as spikes, were determined from this raw data. Using the waveform of the spikes, IBL used a modified version of the Kilosort 2.5 algorithm to find clusters, which are individual neurons or multiple neurons that always fire together, referred to as units. Each probe was precisely placed according to the Allen Mouse Brain Atlas, thus the location of every channel is known, allowing us to plot the spikes over time and their depths. See figure 3.

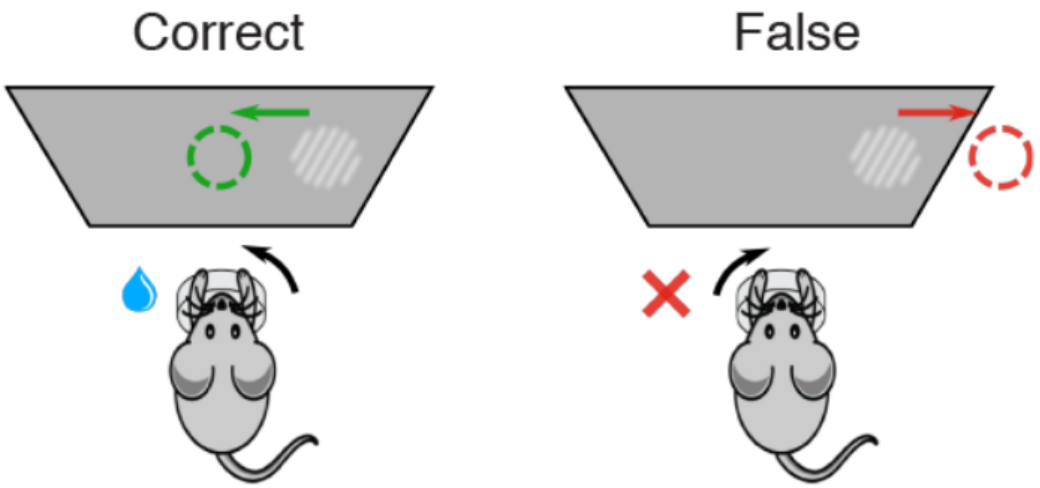


Figure 1: Diagram of the experimental setup showing vertical grating, and a correct vs incorrect response.

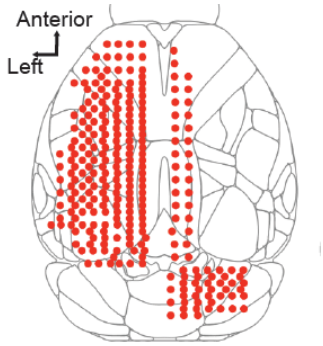


Figure 2: Coordinates of where Neuropixel probes were inserted.

During the task, the mouse is simultaneously shown the vertical grating, and a 5 kHz sine wave was played for 100ms. The mouse is given a reward of 1.5 μL water if it correctly completes the task, and a noise burst is played if it incorrectly completes the task.

Our analyses were focused on the Primary Motor Cortex (MOp) of a single mouse across hundreds of trials. The motor cortex was selected because latent trajectories should be more distinguishable between different mouse behaviors, which would allow us to classify the latent trajectories, and eventually predict the mouse's behavior. We attempt to differentiate trajectories between 4 different trial types, the wheel turned clockwise incorrectly, and the wheel turned counter-clockwise incorrectly.

2 Methods

Raw electrophysiology data is very high dimensional and contains a lot of noisy, spiky activity. Due to this, it must be heavily processed before the accurate neural trajectories can be extracted. We utilized two important methods in our study that are able to both process and model our complex datasets: Gaussian Process Factor Analysis and Variational Latent Gaussian Processes.

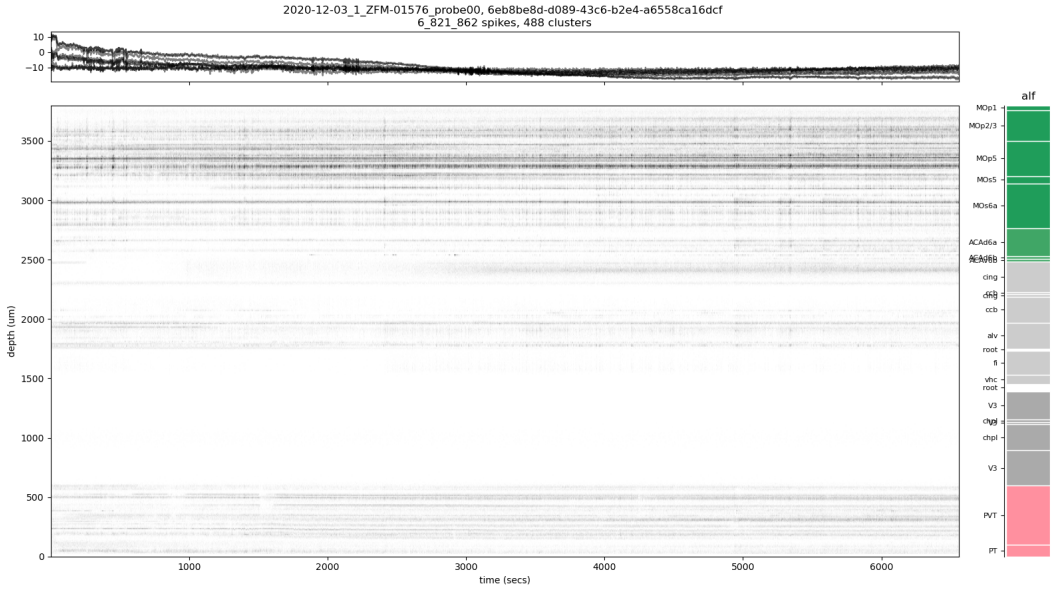


Figure 3: Raster plot of Spikes over Time vs Depth with corresponding brain region acronyms on the right side.

2.1 Gaussian Process Factor Analysis (GPFA)

Gaussian Process Factor Analysis (GPFA) is a probabilistic modeling technique used to infer low-dimensional structure from high-dimensional time series data. The method is based on the idea of representing each high-dimensional time series as a linear combination of a small number of underlying latent factors that vary smoothly over time.

GPFA models the observed data, y, as a linear combination of low-dimensional latent variables, z, using a Gaussian process. The latent variables are modeled as a Gaussian process with mean, m, and covariance function, $k_z(z_i, z_j)$. The observed data is modeled as a linear combination of the latent variables, given by $y = Cz + \epsilon$, where C is a loading matrix and ϵ is a noise term.

The goal of GPFA is to infer the latent variables and the loading matrix given the observed data. This can be done using maximum likelihood estimation by maximizing the log-likelihood of the data, given by:

$$\log p(y \mid \mathbf{C}, \mathbf{m}, \mathbf{k_z}) = -\frac{1}{2}\mathbf{y}^T\mathbf{K}_y^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}_y| - \frac{n}{2}\log(2\pi)$$

where $K_y = CC^T + \sigma^2 I$ is the covariance matrix of the observed data and σ^2 is the noise variance. The optimization problem can be solved using iterative algorithms, such as the expectation-maximization (EM) algorithm or the gradient descent algorithm.[2]

The solution provides estimates of the latent variables and the loading matrix, which were used to uncover patterns in our mouse's brain activity.

2.2 Variational Latent Gaussian Processes (vLGP)

Variational Latent Gaussian Process (vLGP) is an extension of Gaussian Process Factor Analysis that incorporates a variational inference framework to allow for more efficient and scalable inference. This also allows for a more flexible approach to modeling time-series data. For example, in our case the prior distribution fed into our model is Gaussian while the posterior distribution that is solved for is Poisson. This allows us to create a model more closely aligned with real world neural data.

The goal of the vLGP is to infer the posterior distribution, $q(z|x)$, over the latent variables given the observed data. This is done using variational inference by minimizing the objective function, also

known as the evidence lower bound (ELBO), given by:

$$ELBO = -D_{KL}(q(z|x)||p(z)) + E_{q(z|x)}[log(p(y|z,x))]$$

where D_{KL} is the Kullback-Leibler divergence, which measures the difference between two distributions, and E is the expected value. The first term in the ELBO encourages the approximate posterior, $q(z|x)$, to be close to the prior, $p(z)$, while the second term represents the negative log-likelihood of the data given the latent variables [3].

The solution to the optimization problem provides estimates of the latent variables, which can be used to reconstruct the hidden patterns in the data. For the purposes of our project, vLGP is used to extract neural trajectories, which are the underlying patterns in neural activity that reflect how the brain processes information.

2.3 Data Preparation for vlGP and GPFA

Prior to running the vlGP model, we spent a great deal of time preparing the data and getting it into a format which would allow vLGP to produce good trajectories. The first step in this process was to filter our data by the spikes produced in our brain regions of interest - the primary motor cortex. Once we obtained the spikes within our regions of interest, we filtered once more to keep only the "good spikes", which are the spikes which pass all three of the metric tests conducted by IBL themselves. At this point, we had significantly reduced the amount of spikes being analyzed, leaving us with a cleaner, yet very informative data set to work with. After filtering, we worked on getting the data into the correct format to pass into the model. This consisted of two steps: aligning our spikes, and binning them appropriately. We chose to align the spikes based on the mouse's first movement, analyzing the spikes from 100 milliseconds before the first movement up to 1000 ms after this movement. This method of alignment gave us important insight into how the neuron activity and latent variables change throughout the entire process of the mouse making a decision, from the mouse thinking about the decision before moving, the mouse moving the wheel, and the potential neuron activity after the mouse's first movement. After aligning our spike times, we binned each trial by 50 milliseconds - where the index of each row represented time, and each column was a specific neuron, with entries representing the number of times in which that particular neuron fired between bins. The motivation for binning in this format is that, upon passing multiple trials - each binned by 50ms - we are able to examine neural spike activity within each trial, which is revealed by each individual trajectory for a specific decision. With this in mind, we separated the trials into two categories: when the mouse rotated the wheel clockwise and when the mouse rotated the wheel counterclockwise.

To prepare our data for GPFA, we underwent the same initial filtering process to obtain the appropriate neurons within our brain regions of interest. We also aligned our spikes and binned them in the same way. However in order to fit the model, we had to convert our data into SpikeTrain objects, which take in the trial start time, end time, and an array of times in which the neurons fired over the length of a trial. These SpikeTrain objects were passed into the GPFA model, and from this we were able to generate trajectories. Similarly to vlGP, we separated the data into two categories mentioned above, allowing us to observe the difference in activity in our regions of interest between the decision to turn the wheel left versus right.

2.4 Classification

After extracting trajectories from our data preparation process, our aim was to determine the time at which we could best observe a clear separation between the trajectories of the two different trial types, counterclockwise and clockwise wheel rotation. We determined this time by plotting our trajectories in a 3-dimensional space and testing on data from multiple different mice, who all exhibited unique trajectories. Eventually, upon achieving trajectories that had an interval of clear separation between categories, we were able to build our Logistic Regression classification model.

The time where we observed separation happened to be around 100 milliseconds after the mouse began moving in each trial. The latent feature data from each trial, at the time of separation, was then separated into four categories: clockwise and correct, counterclockwise and correct, clockwise and incorrect, and counterclockwise and incorrect. The aim of the separation at this point was to see if we could achieve separability within each individual trajectory regarding whether that decision was correct or not, and to check the strength of classification between the clockwise and counterclockwise trajectories. This was done using two Logistic Regression classifiers. First we

compared clockwise against counterclockwise, and then we compared clockwise and correct against clockwise and incorrect.

3 Results

After fitting both vlGP and GPFA models with two latent variables, and experimenting with different mice, we obtained the following trajectories plotted over time:

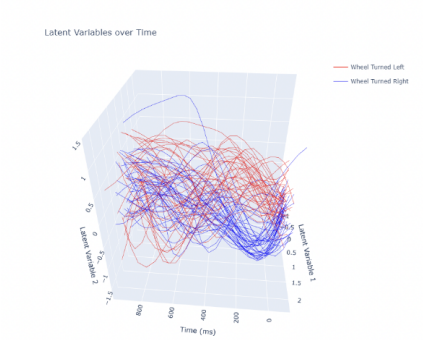


Figure 4: vLGP

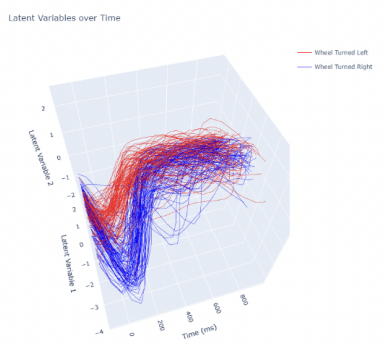


Figure 5: GPFA

Looking at the above figures, we can see that in both models, we achieve separability between the trajectories for left and right movements between 0 and 200 milliseconds for both plots. For both plots, we used data from this interval of separation in order to train our logistic regression model. Upon zooming into the plots, when time is equal to 100 milliseconds, we observe the following for vlGP (Figure 6) and GPFA (Figure 7):

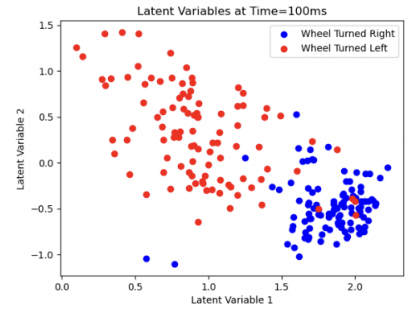


Figure 6: vLGP

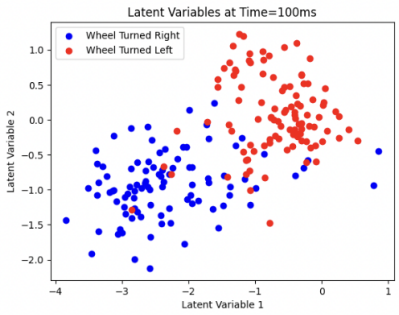


Figure 7: GPFA

Our hypothesis was that, when observing separable trajectories such as our results above, a classification algorithm should accurately predict which way the mouse would turn the wheel based on the brain's neural trajectories. This hypothesis proved true, as our classification algorithm performed very well on test data in both models, achieving 93% balanced accuracy using trajectories from vLGP, and 91% accuracy using trajectories from GPFA.

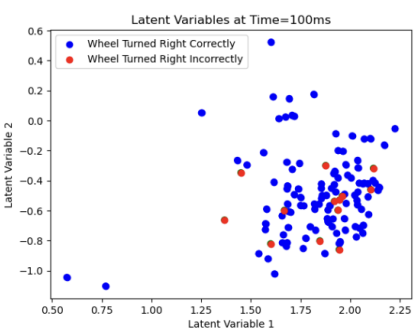


Figure 8: vLGP

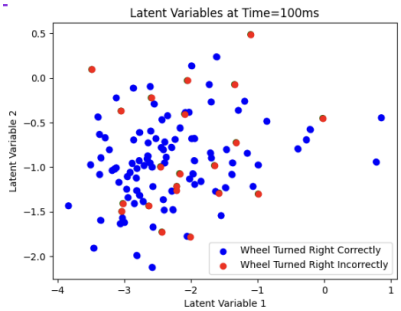


Figure 9: GPFA

We also hypothesized that separating trials by whether or not the mouse was correct in the direction it moved the wheel would not help us achieve separability, and this was proven true upon analysis of correctness at the times when we observed statistical significance. Looking at Figures 8 and 9, we can see that the latent variables behave similarly when the wheel was moved correctly to the right versus incorrectly to the right, and this is likely because the mouse thinks that moving the wheel to the right is the correct decision regardless of whether it actually is or not, and thus the activity observed in the brain is not very different and follows the same properties in both cases. This conclusion is further reinforced by the fact that, upon training the classifier to predict whether or not the mouse's decision to move the wheel was correct, the balanced accuracy of our trained classifier was around 50% for both our vlGP and GPFA trajectories. This reveals that a trained classifier's predictions are no better than a random guess between the two possible categories, and thus the decision to not separate by correctness proved fruitful.

4 Discussion

Our results indicate that using vLGP trajectories to predict decision making is more effective than using GPFA trajectories. There are multiple potential reasons behind why we observed such a difference in performance. For one, raw neural data is very erratic, with action potentials lasting only for 2 to 4 milliseconds, and with many action potentials occurring all at once. Our vLGP model did better at capturing this quality of our data than GPFA which always gives very smooth results. In addition, The slight difference in accuracy between our model trained on vLGP trajectories and our model trained on GPFA trajectories can be attributed to the fact that our trajectories for vLGP had better separability than those of GPFA. The fact that we had better separability for vLGP reveals that the distribution of our data aligns more closely with that of a Poisson distribution than a Gaussian distribution, and therefore we were able to observe the distinction between the decision trajectories more clearly, and the classifier was more accurate in differentiating between categories.

In addition, when it came to prediction, we initially wanted to classify between the four categories of left and correct, left and incorrect, right and correct, and right and incorrect, and we struggled to get separable trajectories for each of these categories. This struggle was present for several reasons, the first being the fact that within the decision trajectories themselves, correct and incorrect decisions exhibit the same behavior, which makes sense due to the fact that the mouse made its decision thinking it was the correct choice, and thus the thought process for choosing right or choosing left is uniform regardless of correctness. Secondly, our analysis was conducted only using data from one brain region at a time, in the mouse where we observed separable trajectories, we only examined the spikes from the motor cortex. The motor cortex is largely responsible for movement, and thus examining only this region does not provide us with insight whether or not the mouse was thinking about whether the decision was correct or not after making it. If we wanted to categorize the trajectories by both correctness and decision, we should have monitored the activity of another brain region at the same time intervals which we analyzed the motor cortex, perhaps the prefrontal cortex, and this would have shed more light as to how neural activity can be different for an incorrect vs. correct decision.

Another important aspect of our investigation was determining the ideal brain region to analyze, which ended up being the motor cortex, which is heavily involved in movement. We initially looked

at the visual field, since the mice are presented with a visual stimulus and are tasked with responding to it. The resulting trajectories, however, did not exhibit any separability, and ultimately a classifier would not have performed well when trained on these trajectories. This could be due to the fact that we did not give enough time before the mouse's first movement when we were aligning our spikes in our data preparation, as we only looked at the data before the first movement for 100 milliseconds, but included spike data from up to 1000 milliseconds after the first movement. This could explain why analyzing the primary motor cortex gave better trajectories. Perhaps if we reversed this alignment, and looked at more data before the first movement, while the mouse was processing the stimulus, we could have achieved more separable results from analyzing the visual field. What's important to take away from this is that when analyzing neural data, we must pick our brain region very carefully, and make sure that the data preparation allows us to maximize results from that region.

5 Appendix

Project Proposal

References

[1] International Brain Laboratory. *Standardized and reproducible measurement of decision-making in mice*. eScholarship, University of California, 2021.

[2] 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. *Journal of Neurophysiology*, 102(1):614–635, 2009. PMID: 19357332.

[3] 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, May 2017.

6 Team contributions

6.1 Aryan

Independently wrote code for GPFA and vLGP. Contributed to poster, report, and website.

6.2 Jad

Independently wrote code for GPFA and vLGP. Contributed to poster, report, and website.

6.3 Saket

Independently wrote code for GPFA and vLGP. Contributed to poster, report, and website.

6.4 Rishabh

Independently wrote code for GPFA and vLGP. Contributed to poster, report, and website.