Hierarchical Latent Variable Models for Neural Data Analysis

Courtney Cheung c6cheung@ucsd.edu
Sha Lei s1lei@ucsd.edu
Yutian Shi yus029@ucsd.edu
Shuyu Wang shw043@ucsd.edu

Mikio Aoi maoi@ucsd.edu

Abstract

When the brain receives a stimulus, the associated information is encoded to inform a neural response. The goal of neural coding is to describe the sensory stimuli responsible for neural and behavioral responses, which can be measured through electrode arrays and imaging techniques. Decoding such measurements has several uses, including the development of brain-computer interface (BCI) prosthetics, which assist in and repair motor functions. Our research builds upon established dimensional reduction techniques and Bayesian inference in order to create machine learning models for decoding that account for both temporal correlations and the non-negative support of neural activity data. Specifically, we aim to learn latent neural trajectories (lower dimensional systems underlying observed neural behavior) by applying variational Gaussian process factor analysis (vLGP) to spike train data recorded from studies on decision-making in mice. This modification of GPFA assumes that spike counts are poisson-distributed, and consequently utilizes variational inference to address the issue of non-conjugate priors in the latent space by maximizing the evidence-based lower bound (ELBo). Furthermore, we analyze the correlation between neural activity in two brain regions – the Superior Colliculus Deep Gray Layer and the Superior Colliculus Intermediate White Layer – during a decision making task, using Probabilistic Canonical Correlation Analysis (pCCA).

Code: https://github.com/courtneyacheung/Hierarchical-Latent-Variable-Models-for-Neural-Data-Analysis

1 Introduction

1.1 Literature

The work on neural spike sorting has evolved substantially, emerging after new techniques that allow monitoring hundreds of neurons at the same time in a sub-millisecond (Rey, Pedreira and Quiroga 2015). Nevertheless, due to the large amount of data that needs to be processed, new algorithms that could handle the vast neural data while filtering useful parameters are mandatory. Our approach seeks to innovate by integrating these historical strengths while mitigating their weaknesses when performing the dimensional-reduction task, aiming for better capture of the dynamic of neural trajectories in the neural population activity.

In the work on maximum likelihood factor analysis, statistics researchers Donald Rubin and Dorothy Thayer introduced the EM Algorithm. This algorithm comprises two essential steps in each iteration: the E step (expectation step) and the M step (maximization step). The algorithm ends when it converges to a maximum likelihood. The advantage of the algorithm is that it is applicable to a broad spectrum of problems and usually converges to a local maximum (Rubin and Thayer 1982). This method proves particularly valuable in scenarios where multiple maximums need to be identified, which is common in factor analysis (Rubin and Thayer 1982). In terms of implementation, the code for each iteration is relatively straightforward and boasts a low computational cost. However, it is worth noting a limitation: the method's performance diminishes when dealing with medium-sized samples (Rubin and Thayer 1982). Consequently, while the EM algorithm stands as a reliable tool, its effectiveness may vary depending on the specific problem at hand. Further research might be necessary to determine its suitability for particular applications.

In the paper "Gaussian-Process Factor Analysis for Low-Dimensional Single-Trial Analysis of Neural Population Activity", the authors introduced the Gaussian-process factor analysis (GPFA) for extracting neural trajectories. This method can help to combine smoothing and dimensionality reduction operations (Yu et al. 2008). During their experiment, they showed GPFA improved the prediction of their study by using metrics to measure how others can predict each neuron compared to two-stage methods because the two-stage methods have no use of time-label information. They also explained how GPFA is valuable for linking spiking activity across neural populations when it knows the time course (Yu et al. 2008). The Gaussian process enables the incorporation of correlation structures among low-dimensional states. This approach helps when dealing with systems that want to undergo smooth temporal evolution, and allows the extraction of a smooth low-dimensional neural trajectory, which can be seen as a balance between the low-dimensional representation of each data point achieved by factor analysis (FA) and the ambition to combine data points using continuous, smooth functions (Yu et al. 2008).

In addition to GPFA for single-region analysis, there has also been research utilizing methods for multi-region analyses (Keeley et al. 2020). Two such statistical methods include regression-based models and shared latent variable models. Regression-based models "fit a model that predicts activity in one brain area using the activity from one or more other

brain areas" and shared latent variable models "model the response fluctuations in a large population of neurons as arising from a small number of hidden or "latent" sources. For our purposes, we are specifically interested in the shared latent variables models. Previous models that have been used for multi-region data analysis are variants of Canonical Correlation (CCA), such as Probabilistic CCA and Bayesian CCA. We will further explore pCCA in the methods section.

1.2 Motivation

We want to build a model that makes use of the advantages of GPFA and EM algorithms – namely, that GPFA works well with temporal data and that the EM algorithm is efficient for factor analysis, where multiple maximums need to be identified. Because Gaussian distributions produce positive and negative draws, a more appropriate modification to GPFA will use a Poisson distribution to reflect the true non-negative support of our data. We would also like apply pCCA to two distinct brain regions in order to explore whether there is a relationship between neural activity in those regions during decision making tasks.

1.3 Data

The International Brain Laboratory (IBL) collects data that records the neural activity across the entire mouse brain when it has a decision-making task. The dataset contains various data from 115 mice, 354 sessions, and 295501 neurons. Neural data is simultaneously acquired from up to two probes that record during each session (N=354).

In our project, we carefully process the data by using high-quality clusters/units (groups of spikes) that meet certain standards (the label variable of clusters equals 1). The data of each session includes details about the number of spikes, clusters, and trials. For every signal trial, a table is created, offering insights into trial events. This table, with dimensions equal to the number of trials, contains various variables related to trial parameters and mouse behavior.

These variables include 'firstMovement_times,' denoting the time of the initial movement detection during the trial, with NaN indicating no movement detected. The 'choice' variable signifies the response, with -1 indicating a counterclockwise wheel turn, +1 for a clockwise turn, and 0 indicating no wheel movement. Additionally, 'contrastRight' denotes the contrast of stimuli presented on the right side of the screen, while 'probabilityLeft' indicates the probability of the stimulus on the left side (with the probability on the right being 1 minus 'probabilityLeft').

For the cluster data, we can get variables such as brain region and overall firing rate.

Studying the neural process behind decision-making is a basic and interesting aspect of neuroscience. The IBL dataset provides an opportunity to study the basic aspects of decision-related neural activities from patterns of trajectory and their correlation with behavioral outcomes. By exploring how to encode data, researchers can reveal the complex

process behind decision-making. This dataset can also provide insights into neural variables, such as stimulus and reward clues, and show the representation and processing of key information by data.

1.3.1 Data Acquisition

- We first load the raw neural data with the Open Neurophysiology Environment (ONE) library, a standard framework for loading neural data.
- ONE, The Open Neurophysiology Environment, standardizes, searches, and shares neurophysiology data. It establishes conventions for data storage, including cross-references and time synchronization. The protocol offers an API for searching and loading datasets from local machines or remote servers. We use search functions to help us search mice brain data based on sessions (experimental sessions) or insertions (Neuropixels recordings).
- SpikeSorting loader helps to load spike sorting data for a given probe insertion.

1.3.2 Data Cleaning

Candidate Regions
The empirical data from IBL is recorded from brain activity in different brain regions of mice. These mice are given visual stimuli with varying levels of contrast as well as rewards and punishments based upon the correctness of their response to each stimulus. The major task involved the mice' motion control on how to turn the wheel and decision-making process determining the degree to which they should turn the wheel. We pick the Midbrain region, Superior Colliculus, as the candidate regions, as it has the motor functions for controlling the eyes and brain's orientation to the stimulus (Gandhi and Katnani 2011). We selected Superior Colliculus Deep Gray Layer (SCdg) and Superior Colliculus Intermediate White Layer (SCiw) as the two candidate regions, and we will explore whether they are activated simultaneously and the patterns of the neuron activities in these two regions under different task conditions.

Probe Selection
We wanted to find high quality data. Specifically, a probe insertion that collects a relatively large number of neurons in our researched brain region. To achieve this, we searched for all potential probe insertions that contain our candidate regions. When searching the data, we used the built-in function in one API "one.search", and we chose the particular region by inputting the acronym name of the region. On average, one brain region may contain approximately 100 to 200 clusters of neurons. Then, we counted the number of good clusters of our researched brain region within each probe and found probes with the most high-quality data. When filtering the data, we use the built-in function in one API "one.search", and we choose the particular region by inputting the acronym name of the region. While doing multi-region analysis, we also ensured that the data was collected from the same experiment and the same subject.

Spike Sorting and Cluster Selection
After finding the probe with the most high-quality data, we loaded in the spikes, clusters, and trials data for the respective probe. We then performed spike sorting, and filtered the spike data by only including spikes from good clusters. A cluster is a group of spikes captured from the same neuron or very similar neurons. We defined our good clusters as clusters that fulfill at least two of the three quality control metrics:

1. `slidingRP_viol = 1`
2. `noise_cutoff < 20`
3. `amp_median > 50`

`slidingRP_viol` is a binary metric which determines whether there is an acceptable level of refractory period violations by using a sliding refractory period. When the variable is 1, it means that the cluster has an acceptable level of refractory period and a decent firing rate. `noise_cutoff` is a metric to determine whether a unit's amplitude distribution is cut off (at floor), without assuming a Gaussian distribution. `amp_median` is a metric that computes the median of the spike amplitudes. By only including spikes from clusters that fulfill at least two of these requirements (clusters with the `label` variable greater or equal to $\frac{2}{3}$), we obtained spike data with higher quality.

Data Query and Variable Extraction
Since a probe may contain clusters from multiple regions, we filter our data to only include spikes from our researched regions. We also extract useful variables from the trials dataset, including time variables, contrast, choice, and accuracy, which will be useful for analyzing cell activities under different conditions.

Temporal Alignment
It is essential to make sure that the data and the signals are synchronized, so we need to check that the time bins are consistent when we compare clusters of cells in different brain regions. By conducting the EDA, we found out that the peak of firing rate appears around 100 milliseconds ahead of the `firstMovement_times` when the mouse turns the wheel, and the fluctuation of the firing rate ceased at approximately 1s after. Therefore, we select the time window for a single trial to be in `firstMovement_times` - 0.1 and `firstMovement_times` +1, and thus making sure there are no time intervals overlapping or redundant spike data counted in.

1.3.3 Exploratory Data Analysis (EDA)

We performed EDA to find the relevant variables for our subsequent data analysis. We also conducted statistical analyses to identify patterns, correlations, or significant changes in neural activity based on different stimulus conditions. First, we analyzed the following.

- Raster: a visual representation that shows the timing and occurrence of neural spikes over time. Each vertical line represents a spike and the horizontal line represents

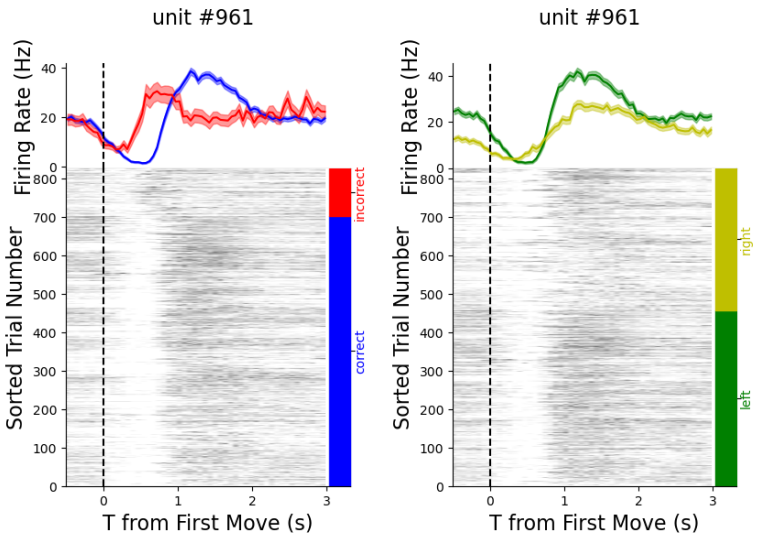
time. The Raster plot displays the firing patterns of neurons and is useful for analyzing the neural activities in the brain.

- Peri-Stimulus Time Histogram (PSTH): illustrates the neural firing rate in response to a stimulus over time. It provides the temporal pattern and strength of neural response which helps to analyze how neurons react to specific stimuli.
- Firing rate: the number of times that a neuron releases electrical energy in a given time period.

We plotted multiple images to analyze individual clusters in the SCdg and SCiw regions of the mouse brain. We expect that (1) when we align the spikes using "First Movement Time", there will be a correlation to whether the mouse moved the wheel left or right, and (2) when we align the spikes using "Feedback Time" there will be a correlation to whether the mouse got the task correct versus incorrect.

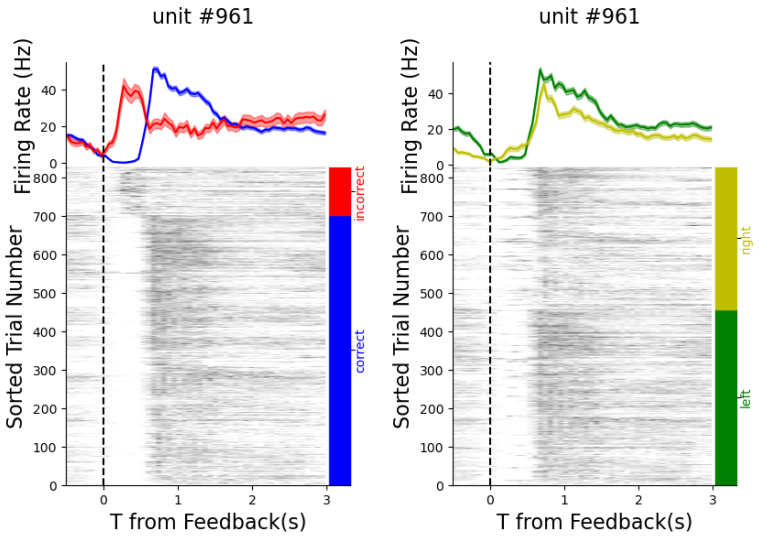
For the SCdg region: In Figure 1, certain features of the images align with our conjectures about the original spike data. For instance, there's a significant change in the firing rates of feedback and decision variables after the first_movement time, indicating the mouse's response (the wheel rotation and reward delivery) to visual stimulus. In Figure 2, the Feedback image also meets expectations. For example, compared to the first_movement time image, the change in firing rate occurs earlier and is closer to x=0. However, unexpectedly, the decision image continues to exhibit noticeable changes after the feedback time, suggesting that the mouse continues to react to left and right decisions after receiving the outcome. Moreover, between the two images, we observe that the changes in 'incorrect' occur earlier than 'correct', and the peak firing rate of 'correct' is higher than 'incorrect'. This suggests that the mouse may respond more quickly to errors but more intensely to correct responses.

For the SCiw region, through Figure 3 and 4, we observe that the trends in spike data are similar to SCdg. However, in the feedback image, the response of 'incorrect' doesn't precede 'correct' significantly. Compared to SCdg, the difference in response time between 'incorrect' and 'correct' has diminished. Additionally, the difference in firing rates between left and right choices has also decreased. There's a significant improvement in the overlap of line plots between the two images. This suggests that the SCiw region may not respond to 'incorrect' as promptly as SCdg and exhibits a more stable reaction to left and right decisions.



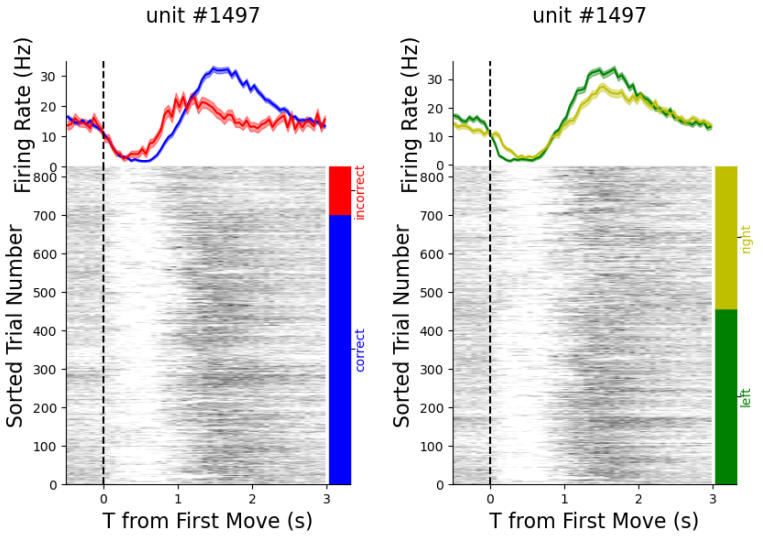
(a) feedback (b) decision

Figure 1: firstMovement_times–SCdg



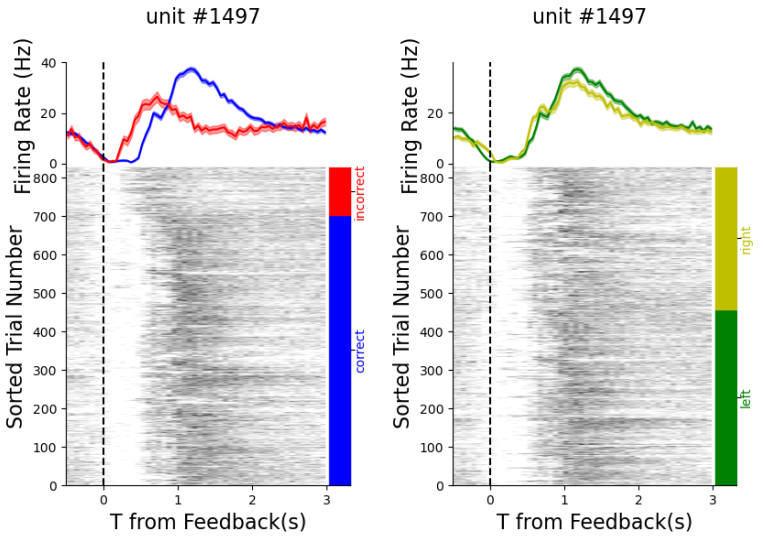
(a) feedback (b) decision

Figure 2: Feedback_times–SCdg



(a) feedback (b) decision

Figure 3: firstMovement_times–SCiw



(a) feedback (b) decision

Figure 4: feedback_times–SCiw

2 Methods

2.1 Single-Region Analysis Approach

We first use the variational latent gaussian process model (vLGP) (Zhao and Park 2017), which relaxes the assumption of linear dynamics in order to reflect the more accurate non-linear dynamics of the brain. This method differs from GPFA in that it uses "a point-process observation model with self-history dependence rather than an instantaneous gaussian observation model" (Zhao and Park 2017). vLGP sequentially integrates evidence into the model. Furthermore, it assumes that $y_n(t)$, the activity of a neuron at a certain time, is drawn from the poisson distribution. However with this assumption, the prior and posterior are non-conjugate. This means that there is no closed-form empirical Bayes solution, and we need to make a guess as to what the posterior of the latent process is. Thus, we use variational inference to minimize the difference between our guessed posterior and our real posterior. By maximmizing the evidence-based lower bound (ELBo), the model minimizes the Kullback-Leibler divergence, and therefore the difference between the guessed and real posterior (Zhao and Park 2017).

2.1.1 GPFA

In our project, we have high-dimensional neural data. By using the GPFA model, we can reduce the data to the lower dimension and capture the variations in neural activities over time. Thus, we can decode data to predict conditions from neural trajectories. Compared to the PCA, GPFA is a more general method that considers the time dependence of neural activities. GPFA also addresses the problem that the directions learned in PCA tend to be dominated by the most active neurons. Additionally, it also includes an explicit noise model. These properties enable it to perform better than the traditional methods, including better performance in adjusting the degree of smoothness, taking account of various timescales, and distinguishing spike noises (Yu et al. 2008).

2.1.2 Why Poisson Observation?

In the traditional Gaussian-process factor analysis, the observed data are assumed to be generated from a Gaussian distribution centered around the latent variables, while the gaussian distribution will produce noise values that are equally likely to be positive or negative centered around 0. However, our observed spike count neuron data are typically non-Gaussian and skewed (Keeley et al. 2020). Moreover, the spike trains are naturally non-negative count data. Therefore, to better fit the dynamics of the neural activity, we will apply the poisson distribution since it is a natural choice for modeling count data and avoid the negative values.

2.1.3 Variational Inference

Variational inference is a method that approximates the complex true posterior p(x|y) with a simpler distribution q(x) (Zhao and Park 2017). It is an essential foundation for us to construct our latent variable model.

2.1.4 Kullback-Leibler Divergence & Evidence Lower Bound

To measure the approximation of variational distribution q(x) to the true posterior, the Kullback-Leibler (KL) divergence was introduced (Zhao and Park 2017).

$$D_{KL}(q||p) = \int_{-\infty}^{\infty} q(x) \log \frac{q(x)}{p(x|y)} dx$$

When the KL divergence is equal to 0, q(x) is exactly the same as the true posterior. Thus, minimizing the KL divergence helps us to find the optimal q(x) that best approximates the actual distribution.

$$\log p(y) = L(q) + D_{KL}(q||p)$$

According to Zhao, minimizing the KL divergence is the same as maximizing the evidence lower bound $L(q)$ (ELBO), a lower bound of the log marginal likelihood (2016). Thus, q(x) can be optimized by maximizing ELBO.

2.1.5 vLGP

By combining the advantages of variational inference, the Variational Latent Gaussian Process model (vLGP) becomes more efficient and practical compared to previous methods (Zhao and Park 2017). This approach is especially beneficial for high-level neural activities such as decision-making, due to its flexibility in inferring latent trajectories for each trial (Zhao and Park 2017). Since we want to explore mice' decision making process, which is high-level, vLGP will be an ideal starting point for us to learn the low dimensional latent trajectory. Additionally, compared to GPFA, vLGP is applicable to more diverse spike train observations (Zhao and Park 2017). Additionally, vLGP is considerably efficient compared to PLDS, another common dimensional reduction method (Zhao and Park 2017).

The data structure of vLGP consists of the basic unit of a single trial which is spike training data y, design matrix of regressors x, length scale of time bins dt. A single trial in our study is defined as the event that the mouse receives the stimulus and response to that stimulus by correctly or incorrectly turning the wheel back to the center of the screen. While the spike train data is a three dimensional array that records each clusters' spike rates in each trial along the timescale. Therefore, each trial is split up into n time points that are calculated by the length scale t, and count the spikes for each cluster within that time interval to form

the spike train y. And then each trial's data is appended into a session that is constructed by length scale dt. And the data are split into train and test data, and we run the vLGP model by the training data, and verified the pattern by the testing data.

2.2 Multi-Region Analysis Approach

After fitting our data to the vLGP model, we would like to explore multi-region analysis to examine variability shared between regions. This will give us insight into how activity in brain regions may be correlated versus distinct during a given task. For this analysis, we will use Probabilistic Canonical Correlation (pCCA). Thus, our overall hierarchical latent variable model will involve learning two layers of latent neural trajectories:

Data → vLGP (latent variable layer 1) → pCCA (latent variable layer 2)

2.2.1 CCA

To understand pCCA, we will first discuss the concept of Canonical Correlation Analysis (CCA). CCA was first introduced by Harold Hotelling in 1936 (wik 2024). Its objective is to take two sets of data and analyze the correlation between them as a whole. Say we want to perform CCA using datasets, X_a and X_b. We first have to summarize each dataset by taking a linear combination of their respective variables. Doing so creates two canonical variables, Z_a and Z_b. The goal of CCA is to then show the correlation between datasets, X_a and X_b, by finding weights, W_a and W_b, that produce projections/canonical variables, $Z_a = X_a W_a$ and $Z_b = X_b W_b$, such that Z_a and Z_b are most correlated. This means that the weights must satisfy the following constraints: (1) the angle between the projected vectors must be minimized, (2) the projected vectors are unit length, and (3) the projected vectors are orthogonal. Solving for W_b will then become a standard eigenvector problem dependent on X_a and X_b's covariance and cross-covariance matrices (Gunderson 2018a).

2.2.2 pCCA

pCCA is the probabilistic interpretation of CCA (Gunderson 2018b). In this setting, we assume that there is a shared latent variable, z, which comes from the normal distribution, $z \sim \mathcal{N}(0, I_r)$. z generates two sets of n observations of each random variable, $X_a|z \sim \mathcal{N}(W_a z + \mu_a, \psi_a)$ and $X_b|z \sim \mathcal{N}(W_b z + \mu_b, \psi_b)$. Renaming W to Λ, we have $x_a = \Lambda_a z + u_a$ and $x_b = \Lambda_b z + u_b$. Using a process akin to the EM algorithm, we update $\Lambda^* = (\sum_{i=1}^n x E_{z|x}[z|x_i]^\intercal)(\sum_{i=1}^n E_{z|x}[zz^\intercal|x_i])^{-1}$ and $\psi^* = \frac{1}{n} diag(\sum_{i=1}^n x_i x_i^\intercal - \Lambda^* E_{z|x}[z|x_i] x_i^\intercal)$, where $x = \begin{bmatrix} x_a \\ x_b \end{bmatrix}$, $\Lambda = \begin{bmatrix} \Lambda_a \\ \Lambda_b \end{bmatrix}$, and $u = \begin{bmatrix} u_a \\ u_b \end{bmatrix}$. The table below shows how properties of CCA translate to assumptions in PCCA.

CCA Property	pCCA Assumption
Correlate z_a and z_b	Shared latent variable, $z \sim \mathcal{N}(0, I_r)$
Canonical variables are orthogonal	Latent variables are independent with an isotropic covariance matrix
Canonical variables have unit length	z has unit variance

As input to our pCCA model, we reshaped the latent trajectories we previously learned from vLGP for each of the two regions.Since pCCA model could learn the shared variability of latent variable of two regions; therefore, we need to ensure that the two regions exist in the same subject (animal) for which there is signal. In the next section, we will explore two different candidate regions for which there are distinct neural trajectories corresponding to different decisions and simultaneously recorded data in both regions.

3 Results

3.1 Training vLGP Model

3.1.1 Superior Colliculus Deep Gray Layer (SCdg)

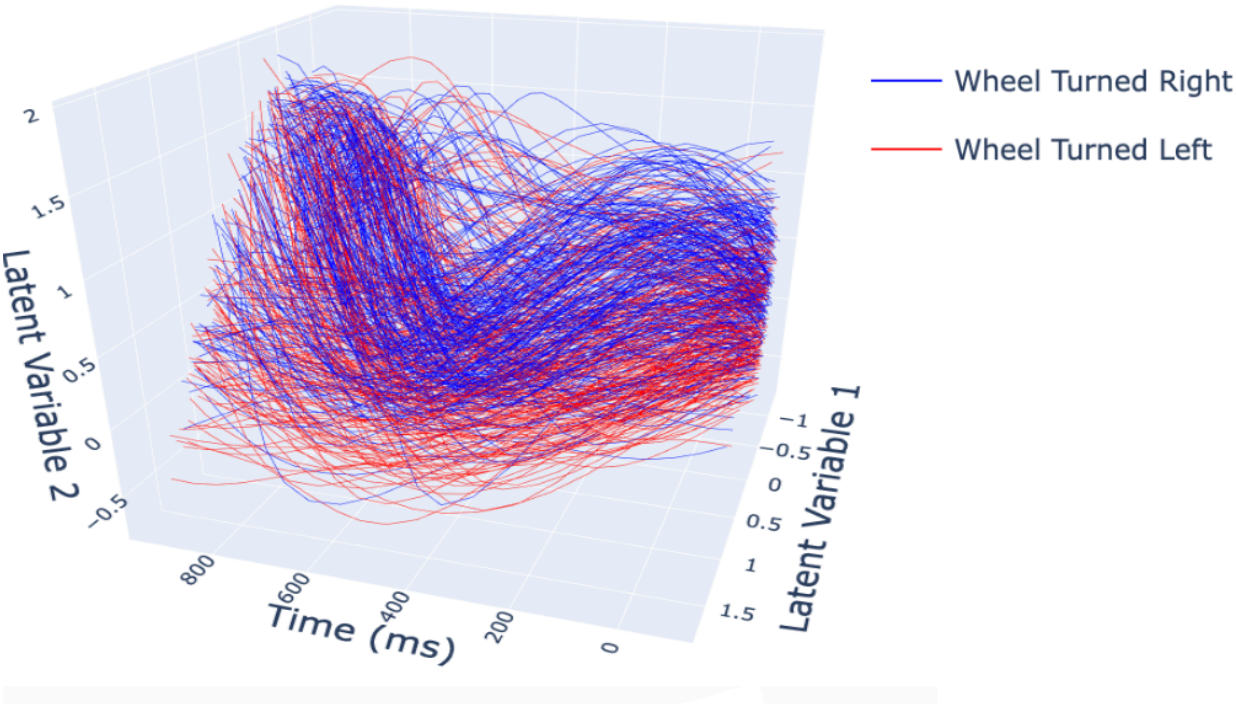


Figure 5: Neural trajectories of SCdg region for mice when turning the wheel correctly

3.1.2 Superior Colliculus Intermediate White Layer

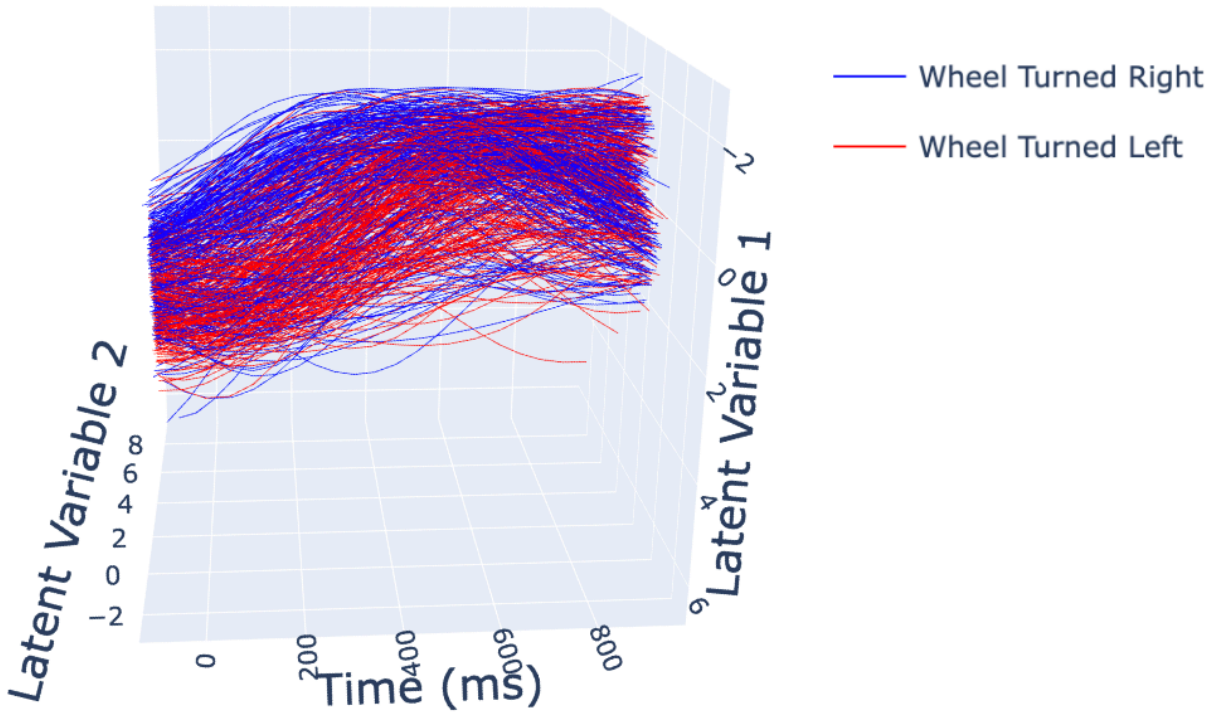


Figure 6: Neural trajectories of SCiw region for mice when turning the wheel correctly

3.2 Training pCCA Model

Next, we fit the pCCA model on the Superior Colliculus Deep Gray Layer datasets and Intermediate White Layer datasets that were produced from the vLGP model. Then, fine-tune the number of latent variable learnt by the pCCA model to testify whether the model operates correctly.

4 Conclusions and Discussions

From the two regions' neural trajectories generated from the vLGP, there are clear separations between wheels turning right and wheels turning left in the brain regions Superior Colliculus Deep Gray Layer and Intermediate White Layer, which means that there are different neural activity patterns under left and right direction conditions. However, we cannot make a conclusion about whether one region is driven by another when both are activated.

Increase in the number of latent variables in pCCA model leads to smaller RMSE between the actual latent variable and estimated latent variable from the two regions, which conforms to the rules that with more latent variables, the model captures the complexity of the latent variable better.

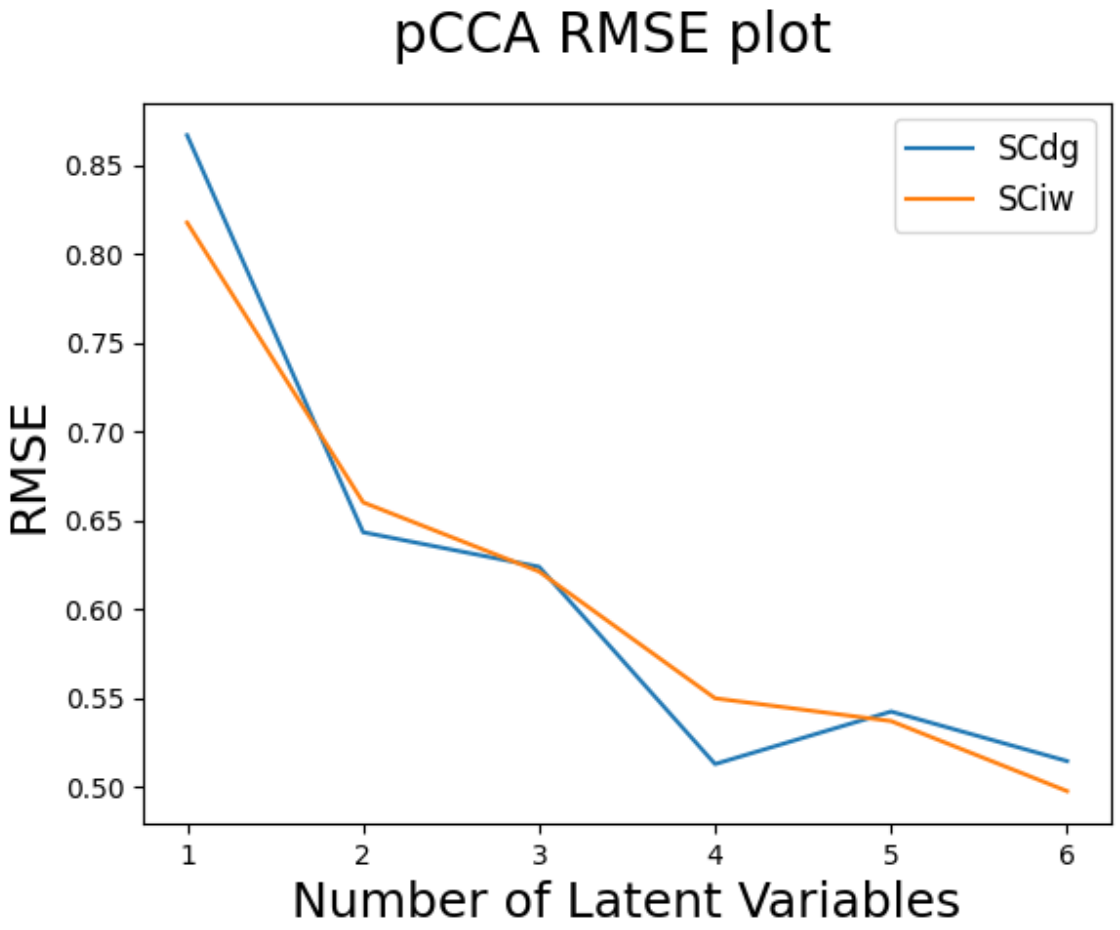


Figure 7: pCCA RMSE plot

For future research, we would like to explore the correlation between regions in more depth by reconstructing the data based on the learned latent variables produced by pCCA. Further research could also include investigating whether there is causation between neural activity in the Superior Colliculus Deep Gray Layer and Intermediate White Layer.

Acknowledgement

We would like to give our most sincere thanks to our mentor Professor Mikio Aoi who provide us guidance and help. We also want to express our appreciation to the members of the previous capstone group in neural data analysis, Aryan Singh, Jad Makki, Saket Arora, and Rishabh Viswanathan, for their invaluable contributions that inspire us to conduct our

research. Finally, we want to give special thanks to our TA Yuyao Wang and instructor Suraj Rampure who support our work and provide suggestions.

References

Gandhi, Neeraj J, and Husam A Katnani. 2011. "Motor functions of the superior colliculus." *Annual review of neuroscience* 34: 205–231

Gunderson, Gregory. 2018a. "Canonical Correlation Analysis in Detail." [Link]

Gunderson, Gregory. 2018b. "Probabilistic Canonical Correlation Analysis in Detail." [Link]

Keeley, Stephen L, David M Zoltowski, Mikio C Aoi, and Jonathan W Pillow. 2020. "Modeling statistical dependencies in multi-region spike train data." *Current Opinion in Neurobiology* 65: 194–202

Keeley, Stephen, David Zoltowski, Yiyi Yu, Spencer Smith, and Jonathan Pillow. 2020. "Efficient non-conjugate Gaussian process factor models for spike count data using polynomial approximations." In *International Conference on Machine Learning*. PMLR

Rey, Hernan Gonzalo, Carlos Pedreira, and Rodrigo Quian Quiroga. 2015. "Past, present and future of spike sorting techniques." *Brain research bulletin* 119: 106–117

Rubin, Donald B, and Dorothy T Thayer. 1982. "EM algorithms for ML factor analysis." *Psychometrika* 47: 69–76

2024. "Canonical Correlation." [Link]

Yu, Byron M, John P Cunningham, Gopal Santhanam, Stephen Ryu, Krishna V Shenoy, and Maneesh Sahani. 2008. "Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity." *Advances in neural information processing systems* 21

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