

STA 141A Winter 2025 Final Project:  
From Neural Dynamics to Behavior: Predictive Modeling of  
Visual Decision Outcomes

919257083      Yunze Hang



Wednesday, March 12th, 2025

# Contents

<b>1 Abstract</b>	<b>1</b>
<b>2 Introduction to the Experiment</b>	<b>2</b>
<b>3 Exploratory Data Analysis</b>	<b>3</b>
3.1 Data Structures across Sessions . . . . .	3
3.1.1 Number of Neurons and Number of Trials Each Session . . . . .	3
3.1.2 The Distribution of Stimuli Conditions Across Sessions . . . . .	4
3.1.3 Feedback Types . . . . .	5
3.2 The Neural Activities during Each Trial . . . . .	6
3.2.1 Single Neural Activities during Each Trial . . . . .	6
3.3 The Neural Activity across Trials . . . . .	8
3.4 The Total Spike Counts Across Trials . . . . .	9
3.5 The Homogeneity and Heterogeneity across Sessions and Mice . . . . .	10
3.5.1 Box Plots . . . . .	10
3.5.2 Kruskal-Wallis Test . . . . .	11
<b>4 Data Integration</b>	<b>13</b>
4.1 Data Standardization . . . . .	13
4.2 Feature Extraction . . . . .	14
4.3 Cross-Session Feature Alignment with Harmony . . . . .	15
4.4 Data Integration and Factor Conversion . . . . .	15
<b>5 Model Training and Prediction</b>	<b>16</b>
5.1 LASSO . . . . .	17
5.2 SMOTE . . . . .	19
5.3 KNN . . . . .	19
5.3.1 KNN without SMOTE . . . . .	19
5.3.2 KNN with SMOTE . . . . .	20
5.4 Logistic Regression with SMOTE . . . . .	20
5.5 XGBoost . . . . .	21
<b>6 The Predictive Model used in the Reference</b>	<b>22</b>
<b>7 Conclusion</b>	<b>23</b>
<b>A Tables</b>	<b>25</b>
A.1 Session-Wise Summary Table . . . . .	25

## 1 Abstract

Nowadays, as the development of brain-machine interfaces and other new devices in the field of modern neuroscience, the ability to decode behavioral outcomes from neural activity becomes an important cornerstone. This project studies a subset of data from “*Distributed coding of choice, action and engagement across the mouse brain*” by Steinmetz et al. (2019)[1] in the field of perceptual decision, mainly focusing on 18 experimental sessions across 4 mice performing a visual contrast discrimination task.

During these 18 sessions, 4 mice were presented with bilateral visual stimuli of varying contrasts and rewarded for correctly indicating the side of higher contrast (or do nothing when in zero-contrast situations). Neural activities from the visual cortex was recorded by Neuropixels probes as spike trains, providing a high-dimensional dataset to explore the relationship between sensory input, neural dynamics, and behavioral outcomes.

The main objective is to develop a predictive model that integrates neural activities across 0 to 0.4s and stimulus contrasts to classify trial outcomes. (Success for 1, Failure for 0.) To address the goal, we structure our statistical analysis into the following several procedures:

- 1) **Introduction to the experiment**
- 2) **Exploratory data analysis**
- 3) **Data integration**
- 4) **Model training and prediction**

In the Data integration part, we used PCA to reduce the high dimensional data; In the Model training and prediction part, we applied many different kinds of predictive models including LASSO, KNN, Logistic Regression, XGBoost, with SMOTE. Moreover, we learnt from the reference[1] and found a new statistical method with Kernel Functions.

This work not only advances methodological frameworks for analyzing high-dimensional and large-scale neural datasets but also contributes to understanding how visual cortex activity encodes perceptual decision.

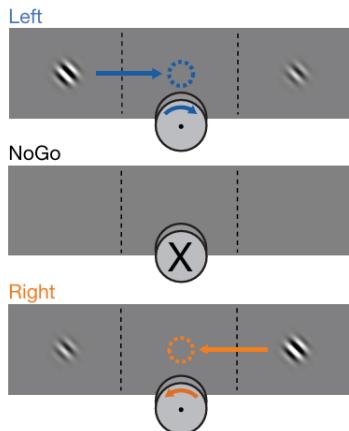
### Key Words

- Neuroscience
- Perceptual Decision
- PCA
- Harmony Algorithm
- LASSO
- KNN
- Logistic Regression
- SMOTE
- XGBoost

## 2 Introduction to the Experiment

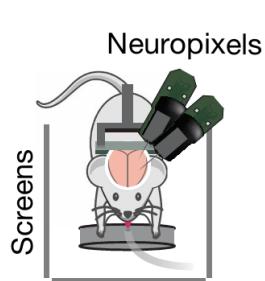
The experiment analyzed in this study investigates the neural correlations of perceptual decision in mice by doing a visual contrast discrimination task, originally designed by Steinmetz et al. (2019)[1].

On each trial, visual stimuli of varying contrast (0, 0.25, 0.5, or 1) could appear on the left side, right side, both sides or neither side. Mice earned a water reward by turning a wheel with their forepaws to indicate which side had highest contrast (Fig.1). If neither stimulus was present, they earned a reward for making a third type of response: keeping the wheel still for 1.5s. If left and right stimuli had equal non-zero contrast, the mice were rewarded randomly for left or right turns.

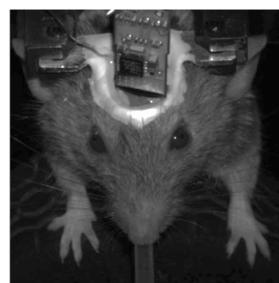


**Fig 1.** Mice earned water rewards by turning a wheel to indicate which of two visual gratings had higher contrast, or by not turning if no stimulus was presented. When stimuli had equal contrast, a left or right choice was rewarded with 50% probability. Grey rectangles indicate the three computer screens surrounding the mouse. Arrows (not visible to the mouse) indicate the rewarded wheel turn direction and the coupled movement of the visual stimulus (X indicates reward for no turn), and the coloured dashed circle (not visible to the mouse) indicates the stimulus location at which a reward was delivered.

The total data in ref[1] was recorded by Neuropixels probes, with approximately 30,000 neurons in 42 brain regions of mice performing a visual discrimination task.



**Fig 2.** Mouse and the Neuropixels



**Fig 3.** The Picture in Real Experiment

Moreover, each session's data was a nested list has the following elements:

```
names(sessions[[5]])
'contrast_left' 'contrast_right' 'feedback_type' 'mouse_name' 'brain_area'
'date_exp' 'spks' 'time'
```

Our main goal is to build up a predictive model for the outcomes (1 for success,  $-1$  for failure). We'll start our task step by step, firstly exploring data, then reducing the dimension, finally training our models and finding the most optimal one (model selection).

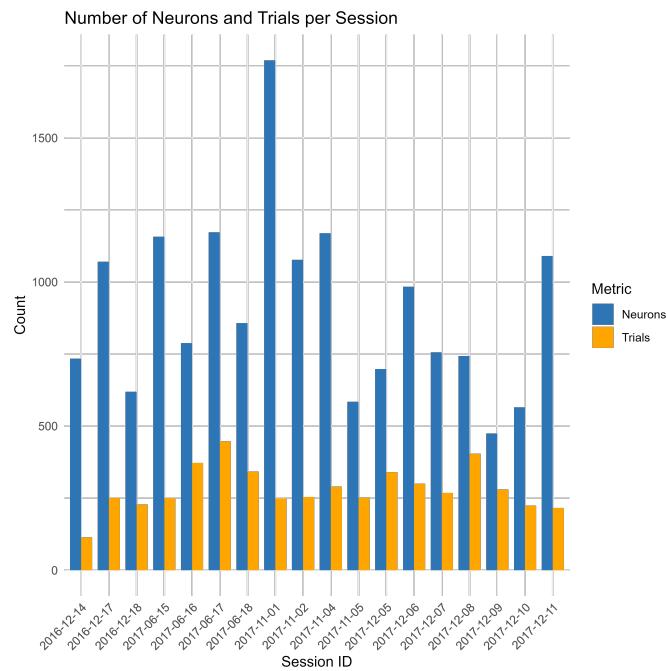
### 3 Exploratory Data Analysis

In this part, we will explore the features of the data sets in order to build our prediction model.

#### 3.1 Data Structures across Sessions

##### 3.1.1 Number of Neurons and Number of Trials Each Session

Firstly, with the data on hand, we are curious about how many neurons and how many trials for each session. After coding, the visualized bar plot is shown below, where the blue bars represent the number of neurons for each session and the yellow ones represent the number of trials for each session. There are 18 sessions in total, as we mentioned in the last section.

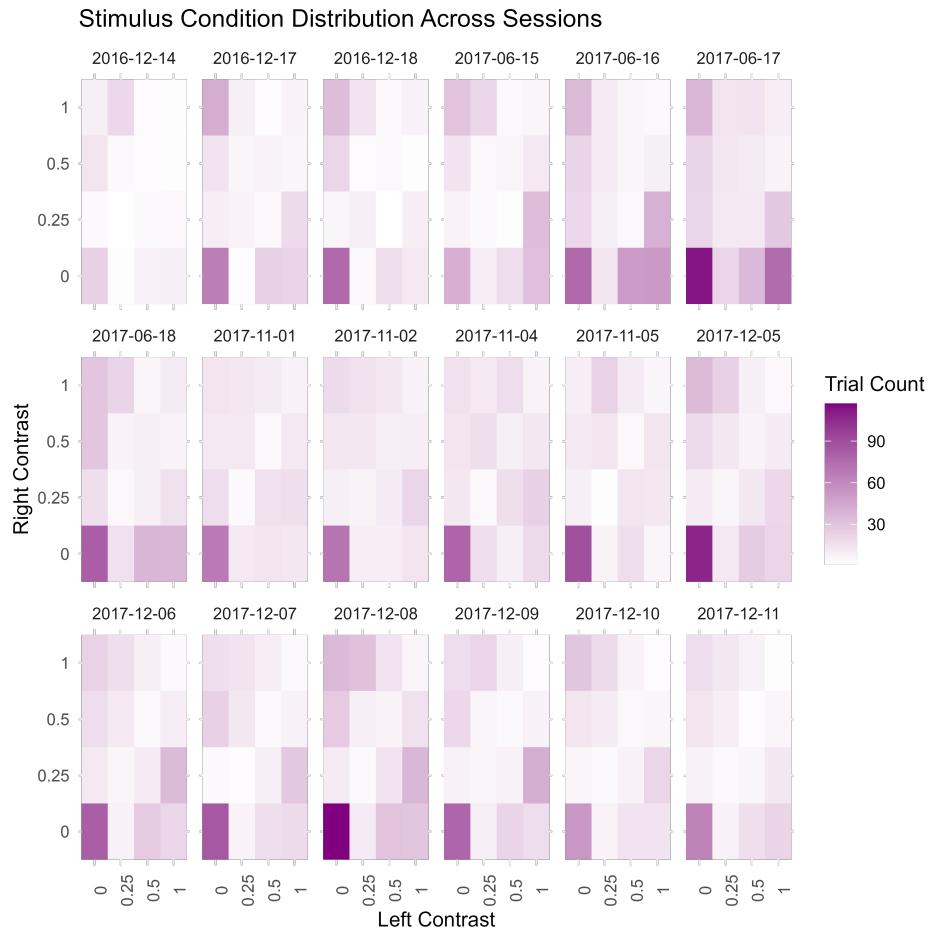


**Fig 4.** The Number of Neurons and the Number of Trials

The means for those statistics above are: 906 for the number of Neurons and 282 for the number of Trials over 18 different sessions. Those numbers are enough for us to get a well performed predictive model. More details are shown in the table, see Appendix Table 3.

### 3.1.2 The Distribution of Stimuli Conditions Across Sessions

The stimuli conditions are also presented by different 18 heat maps as follows.



**Fig 5.** The Distribution of Stimuli Conditions Across Sessions

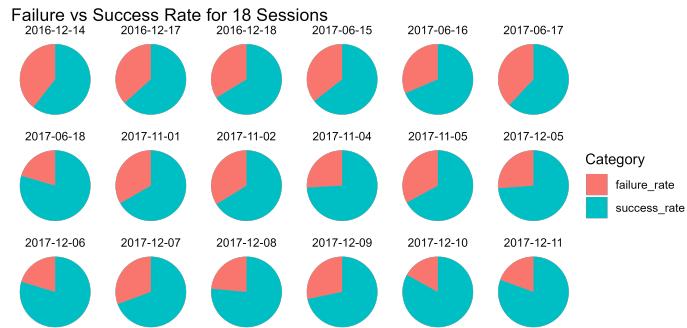
As we can see, the high frequency of trials with  $(\text{contrast\_left}, \text{contrast\_right}) = (0, 0)$  in the statistics is likely a deliberate feature of the experimental design. In the experiment, the  $(0, 0)$  condition, which means no signal on any screens, represents a **no-stimuli trial**, where it is typically to test whether the mouse can correctly inhibit its behavior and reaction in the absence of sensory cues.

It may serve the following propose:

- 1) **Task Rule Learning:** By frequently presenting no-stimulus trials, the experiment tries its best to ensure that every mouse learns the rule of task instead of turning the wheel left or right randomly. It's fundamental for the experiment.
- 2) **Distinguish from Random Moves:** With frequently no-stimulus trials, we can study the behavior of neurons when there's no stimuli.
- 3) **Controlling the Difficulty:** Designing experiment with more no-stimulus trials help the tasks more easier to understand, ensuring that the mice do not develop bias towards always turning the wheel.

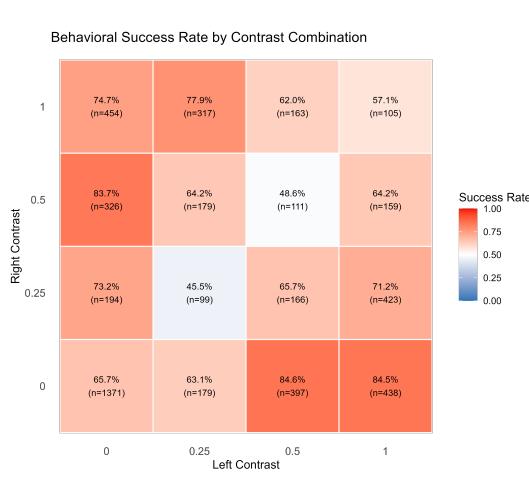
### 3.1.3 Feedback Types

As the experiment mentioned in the last section, there are two feedback types in total: Success and Failure. We plotted the pie chart for each session as follows.

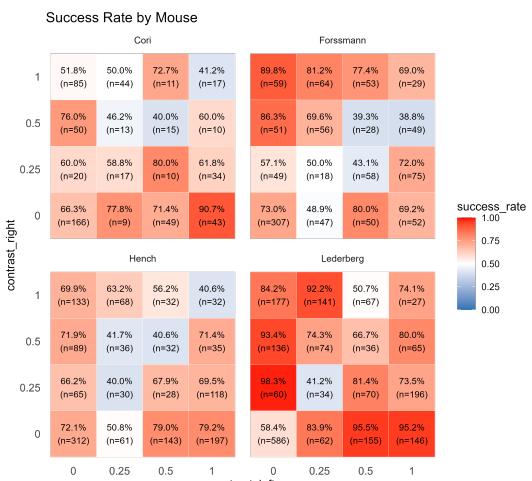


**Fig 6.** The Success Rate and the Failure Rate Pie Chart for Every Session

Moreover, we can explore the successful heat map for all trials and for each mouse, as the heat maps shown below.



**Fig 7.** The Population Heat Map



**Fig 8.** The Heat Map for Each Mouse

As we can see, for easy cases, the success rate is usually higher than 50%, but for more difficult cases, the color of blocks turns to blue and white, which means less or around 50%.

## 3.2 The Neural Activities during Each Trial

Now, with a preliminary understanding of our data, we now dive into the neural activities during each trial, then across trials in the next section. This part of analysis explores the patterns of neural activities each trial. Firstly, we take one special neural and plotted the spike counts of it over time bins, then we plotted the neural population's average activity.

### 3.2.1 Single Neural Activities during Each Trial

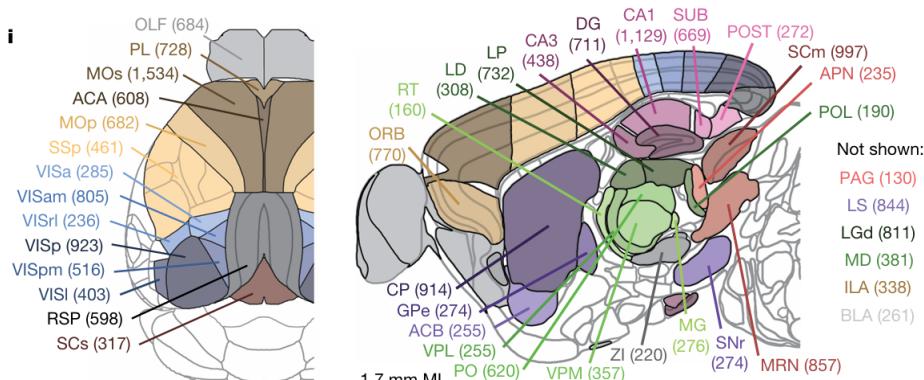
We take the Neuron with id 1 in the dataset as our observing object. It has 203240 rows in total, with 6 columns which has the names: 'session\_id', 'mouse', 'brain\_area', 'trial\_id', 'neuron\_id', 'time\_bin', 'spike\_count'. And the first 6 rows are shown in the table below.

session_id	mouse	brain_area	trial_id	neuron_id	time_bin	spike_count
2016-12-14	Cori	ACA	1	1	1	0
2016-12-14	Cori	ACA	1	1	15	0
2016-12-14	Cori	ACA	1	1	29	0
2016-12-14	Cori	ACA	1	1	3	0
2016-12-14	Cori	ACA	1	1	17	0
2016-12-14	Cori	ACA	1	1	31	0

**Tab 1.** Neural Activity Data

As we can see, this neuron has the 1st number from the experiment held on 2016-12-14, which is from ACA brain area, meaning Anterior Cingulate Area, and it does not have any spike in time bin 1, 15, 29, 3, 17, 31, i.e. no signal during the six time intervals: 0-0.01s, 0.02-0.03s, 0.14-0.15s, 0.16-0.17s, 0.28-0.29s, 0.30-0.31s in the 1st trial, with a mouse named "Cori".

The different brain areas are shown in the graph below from the Ref[1] which originally from Ref[2], where the ACA can be easily found.



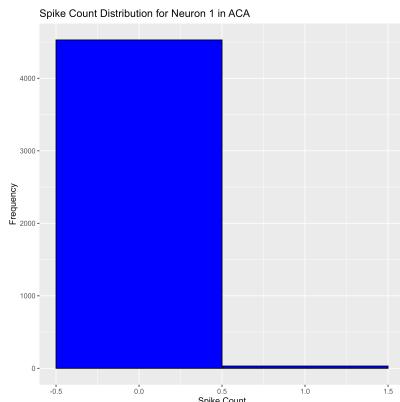
**Fig 9.** Different Brain Area for Mouse

Moreover, we should notice that the “neuron\_id” is only unique during one session, see the corresponding table below.

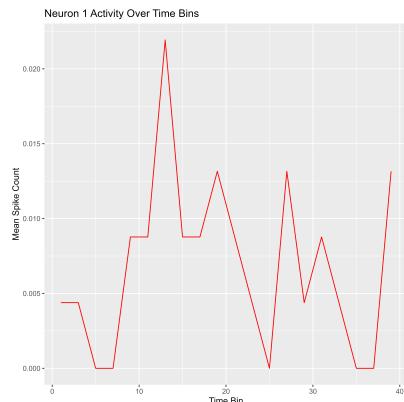
session_id	neuron_id	brain_area
2016-12-14	1	ACA
2016-12-17	1	CA1
2016-12-18	1	DG
2017-06-15	1	ILA
2017-06-16	1	TT
2017-06-17	1	MB
2017-06-18	1	MOp
2017-11-01	1	LGd
:	1	:
2017-12-09	1	SSs
2017-12-10	1	root
2017-12-11	1	CP

**Tab 2.** Neuron and Brain Area Mapping

Now, taking the 1st neuron on 2016-12-14, which located at ACA brain area on a mouse named “Cori” as our example.



**Fig 10.** The Frequency of Spike Counts

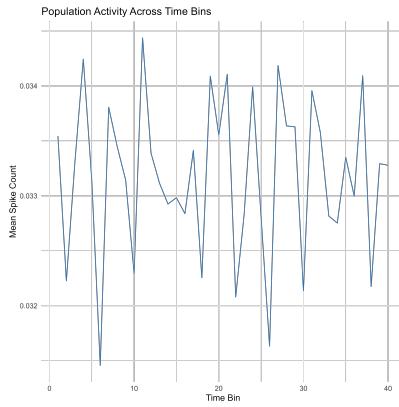


**Fig 11.** The Mean Spike Counts v.s. Time Bins

The two graphs above are the histogram of spike counts and the mean spike counts v.s. time bins from 1 to 40. As we can see, this specific neuron is hard to get activated with a frequency less than 0.03, and usually it is activated around 10 – 15 bins, corresponding to 0.1s – 0.15s in each trial.

### 3.3 The Neural Activity across Trials

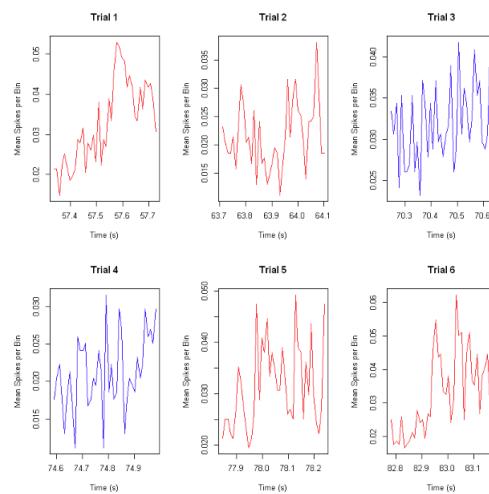
We examined the neural activity patterns of a single neuron (Neuron 1) over 40 time bins during a visual discrimination task, as shown below.



Firstly, as shown in the plot, the activity was recorded as spike trains, which were binned into 40 time intervals (time bins for 0 – 0.4s) to quantify the number of spikes in each interval. The mean spike count for each time bin was calculated across trials to generate the activity profile of the neuron over time.

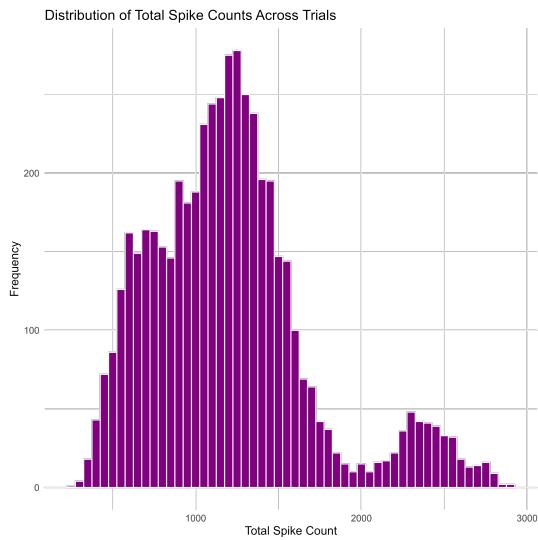
The activity plot for Neuron 1 shows significant **fluctuations** across the 40 time bins. The neuron exhibits periods of high activity with peak mean spike counts reaching approximately 0.02 spikes/bin, interspersed with periods of low or no activity. Notable peaks occur around time bins 12, 22, 32, and 40, may suggesting potential responses to specific events or stimuli in the task timeline. The neuron's activity demonstrates a non-stationary pattern, indicating dynamic responses that vary over the course of the trial.

Also, we plotted the mean spike counts in different trials, colored with red and blue to represent the outcomes. (red for success and blue for failure)



**Fig 12.** Mean Spike Counts in Different Trials

### 3.4 The Total Spike Counts Across Trials



The histogram above depicts the distribution of total spike counts across trials for a neuron. The data shows a right-skewed distribution, with a concentration of trials having lower spike counts and a long tail extending towards higher spike counts. This suggests that while most trials result in relatively few spikes, there are occasional trials with significantly higher spike activity.

The distribution's asymmetry indicates variability in neural response across trials, potentially reflecting different stimuli conditions, behavioral outcomes, or natural variability in neural firing. The spread of the data points around the central tendency provides insight into the reliability and consistency of the neuron's firing pattern during the experimental tasks.

### 3.5 The Homogeneity and Heterogeneity across Sessions and Mice

Now, let's dive into the study for the homogeneity and heterogeneity across sessions and mice.

We firstly studied the participate situation for each mouse, as the table shows, Cori engaged 3 sessions, Forssmann engaged 4 sessions and so on.

Mouse Name	Number of Sessions
Cori	3
Forssmann	4
Hench	4
Lederberg	7

Moreover, for each session, we found that the number of neurons is unchanged for every trial in the specific session. With the above information, we can now study the homogeneity and heterogeneity across sessions and mice.

#### 3.5.1 Box Plots

Firstly, we drawn the box plot for 4 mice and the number of neurons and the success rate in each session to visualize the homogeneity across mice.

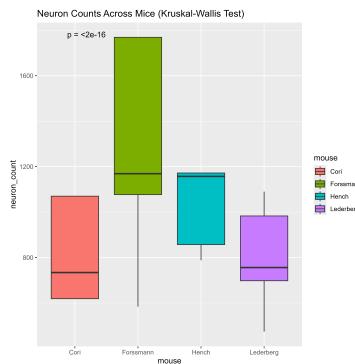


Fig 13. The Neuron Count

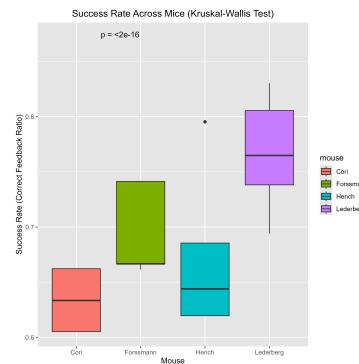


Fig 14. The Success Rate

Also, the box plots for the mean left/right contrast are also shown below.

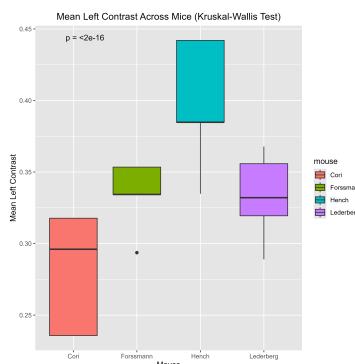


Fig 15. The Mean Left Contrast

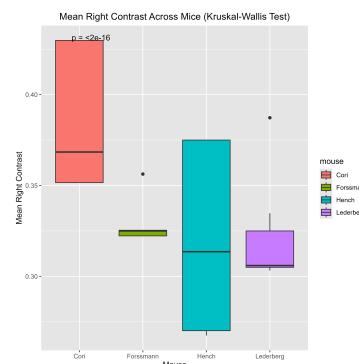


Fig 16. The Mean Right Contrast

As we can see, there are clear gap between medians of them four. Concluding with the preliminary results, we now perform a K-W test on the data.

### 3.5.2 Kruskal-Wallis Test

The Kruskal-Wallis (KW) test we performed aims to assess **the homogeneity of neuron counts, mean left contrast, mean right contrast, and the correct feedback ratio across four mice (Forssmann, Hench, Lederberg, Cori)**. The results shown below.

```
$mean_left_contrast

Kruskal-Wallis rank sum test

data: mean_left_contrast by mouse
Kruskal-Wallis chi-squared = 9049.4, df = 3, p-value < 2.2e-16


$mean_right_contrast

Kruskal-Wallis rank sum test

data: mean_right_contrast by mouse
Kruskal-Wallis chi-squared = 4807.9, df = 3, p-value < 2.2e-16


$neuron_count

Kruskal-Wallis rank sum test

data: neuron_count by mouse
Kruskal-Wallis chi-squared = 6081.6, df = 3, p-value < 2.2e-16


$correct_feedback_ratio

Kruskal-Wallis rank sum test

data: correct_feedback_ratio by mouse
Kruskal-Wallis chi-squared = 9488, df = 3, p-value < 2.2e-16
```

**Analysis of Kruskal-Wallis Test Results:**

Group by mouse, the K-W test was conducted to examine the differences across multiple experimental sessions for the following four key perspectives:

**1) Mean Left Contrast:**

As we can see, the  $\chi^2$  value is 9049.4 with  $df = 3$  and  $p$ -value less than  $2.2 \times 10^{-16}$ . Then we can be sure than there **is a highly significant difference** ( $p < 2.2 \times 10^{-16}$ ) in the mean left contrast among mice. This suggests left stimulus conditions were not balanced across mice, potentially due to inconsistent stimulus parameter settings in the experimental design.

**2) Mean Right Contrast:**

In the same way, the  $\chi^2$  value is 4807.9 with  $df = 3$  and  $p$ -value less than  $2.2 \times 10^{-16}$  as well. Although the magnitude of this difference is smaller than the Mean Left Contrast, it's enough to say the right stimulus conditions were not balanced across mice as well.

**3) Neuron Count:**

As the box plot shown before, we have already noticed that there's a huge difference among mice. With the K-W test results, we found the  $\chi^2$  value is 6081.6 with  $df = 3$  and the  $p$ -value less than  $2.2 \times 10^{-16}$ . This variability reflects recording quality or biological differences among individual mice.

**4) Correct Feedback Ratio:**

The outcomes also have a highly significant difference with  $\chi^2 = 9488$  and the same less enough  $p$ -value, indicating substantial variability in task performance. These differences of outcomes could be attributed to differences in learning ability, motivation levels, or inconsistencies in experimental conditions.

Above all, the Kruskal-Wallis test results reveal significant differences across mice for all four key metrics analyzed: Mean Left Contrast, Mean Right Contrast, Neuron Count, and Correct Feedback Ratio with all  $p$ -values less than  $2.2 \times 10^{-16}$ . These findings suggest substantial variability in both experimental conditions and individual mouse performance, which may significantly impact the interpretability and generalizability of the results.

Moreover, in the predictive model that we are going to build, the KW Test results suggest that we need to take the followings into consideration:

- 1) Standardization and Normalization plays a crucial role in the data integration.
- 2) Model Generalization: Given the variability across mice, it is important to validate models using **cross-validation** techniques that account for individual differences.

## 4 Data Integration

In this part, we need to propose an approach to combine data across trials, for the convenience to build up a predictive model in part 3. It is the bridge for data analysis and predict modeling. By standardizing the neural activity data and extracting key features through **PCA**, we create a consistent representation of neural patterns across sessions. The above data integration process ensures that the final dataset captures both within-session and cross-session variability, providing a robust foundation for developing predictive models in Part 3. The main tasks are constructed as follows:

Task	Description
Data Standardization	Normalize neural activity data across sessions
Feature Extraction	Use PCA to reduce dimensionality
Cross-Session Feature Alignment	Identify common patterns and align features across different sessions
Data Integration	Combine data from all sessions

In the last two subsections, we introduced new statistic method applied in Biology, named the **Harmony Algorithm**, which adjusted for batch effects (i.e. Session-Specific recording conditions).

### 4.1 Data Standardization

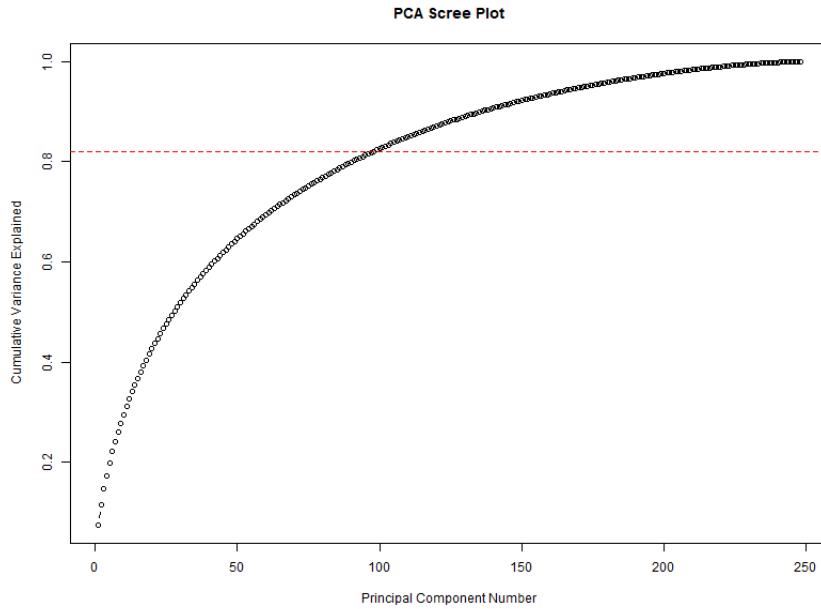
To enable cross-session comparability and model generalizability, we performed session-wise standardization on neural activity data. For each session, raw spike trains were aggregated into interpretable features **by averaging firing rates across brain areas (e.g., VIsP, ACA) and time windows (0–0.1s, 0.1–0.2s, 0.2–0.3s, 0.3–0.4s)**. These features were then standardized within each session using Z-score normalization (with mean = 0, SD = 1) to eliminate baseline variations caused by session-specific recording conditions or biological variability. Part of the results are shown below.(Window 1 to 4 represents the time 0-0.1s to 0.3-0.4s respectively).

```
VISa_window1_mean VISa_window2_mean VISa_window3_mean VISa_window4_mean
1      -6.304825e-17      1.538807e-16      1.504693e-17     -7.965557e-17
root_window1_mean root_window2_mean root_window3_mean root_window4_mean
1      1.211779e-16      1.575218e-16      1.566183e-16      2.004767e-16

VISa_window1_sd VISa_window2_sd VISa_window3_sd VISa_window4_sd
1                  1                  1                  1                  1
root_window1_sd root_window2_sd root_window3_sd root_window4_sd
1                  1                  1                  1                  1
```

## 4.2 Feature Extraction

In this sub-part, we aim to use PCA to reduce the high dimensional data and extract the principal components with the highest variance in the data. With the top 100 PCs, we found they yields 82% of the accumulated total variance, indicating that approximately half of the variability in the high-dimensional spatiotemporal features can be represented in this reduced subspace.



**Fig 17.** PCA Scree Plot

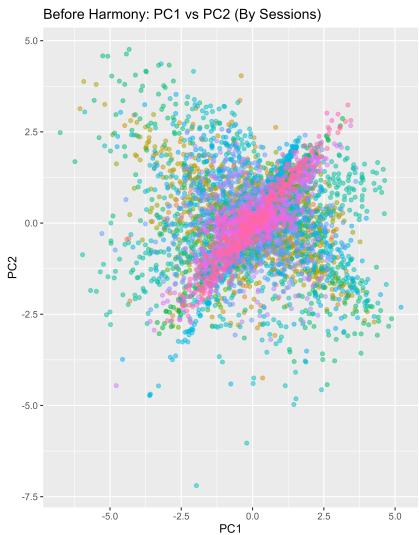
Moreover, the MSE between the original standardized data and the reconstructed data using top 30 PCs is **0.02703899**, suggesting relatively faithful preservation of the original neural activity patterns despite significant dimensionality reduction.

MSE = 0.02703899

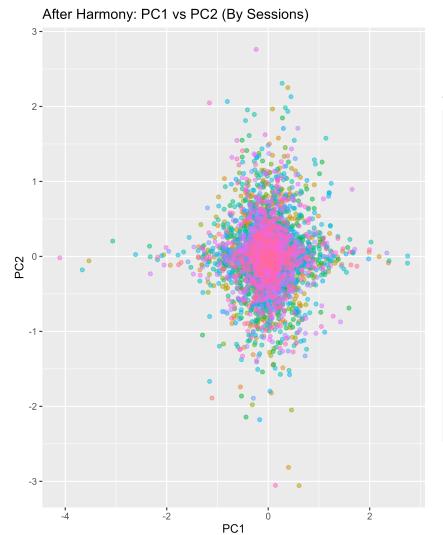
### 4.3 Cross-Session Feature Alignment with Harmony

For harmonizing data across sessions, the Harmony Algorithm was employed. This method adjusted for batch effects, which is exactly dealing with the session-specific recording conditions by aligning the distribution of PCs across sessions while preserving biologically meaningful signals (e.g., stimulus-response relationships). Post-alignment visualization confirmed reduced session-driven clustering in the PC space, enhancing cross-session comparability.

After coding, we got the following graphs below:



**Fig 18.** PC1 vs PC2 before Harmony



**Fig 19.** PC1 vs PC2 after Harmony

### 4.4 Data Integration and Factor Conversion

In the final section of part 2, we let the harmonized dataset integrated into a unified proper structure, with left contrast and right contrast, and the behavioral outcomes (feedback: 1 for success and  $-1$  for failure). Also, we take the categorical variables like session\_id and mouse\_name encoded as factors and put them into the unified structure as well, ensuring handling the predictive model in the next part properly.

The final integrated dataset balances dimensionality reduction, batch correction, and interpretability, providing a robust foundation for training generalized classification models.

## 5 Model Training and Prediction

Now, we dive into the part for predictive modeling. We already had the standardized, PCA-reduced, and Harmony-integrated data into training and test sets. The prediction set consisted of 100 randomly selected samples from session1 and session18, respectively.

We employed three machine learning models for prediction, including LASSO regression, K-Nearest Neighbors (KNN), and Logistic Regression. In the preliminary results of the LASSO model, we identified **a severe class imbalance issue**, which led to the model's inability to predict the minority class (-1) effectively. Specifically, the confusion matrix showed that all predictions for the minority class were 0, while the majority class (1) achieved high prediction accuracy. However, the overall model's Kappa value was 0, indicating that the model failed to effectively distinguish between the two classes.

To address the class imbalance issue, we applied the Synthetic Minority Over-sampling Technique (SMOTE) to the training data. After SMOTE processing, the number of minority class samples in the training set increased to 4158, while the majority class samples numbered 3365, resulting in a more balanced ratio between the two classes.

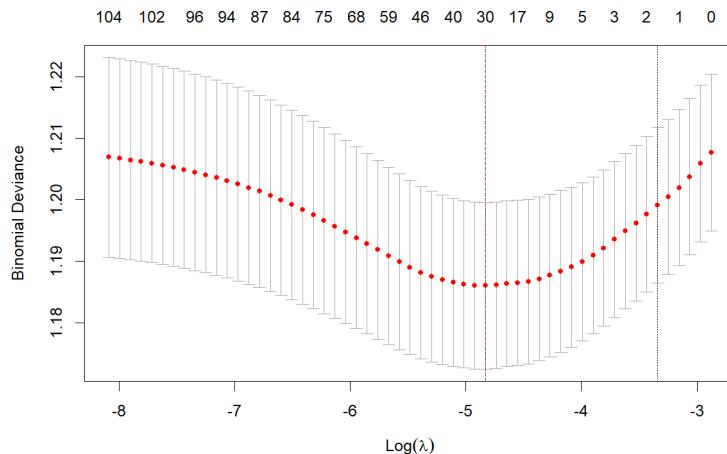
Subsequently, we retrained and predicted using the LASSO, KNN, and Logistic Regression models. However, the performance of the LASSO model remained suboptimal. Although the class imbalance issue was mitigated, the model's predictive capability did not improve significantly. The KNN model showed some improvement after SMOTE processing, but its F1 Score was only 0.147, indicating that the model's predictive ability for the minority class was still weak.

In further analysis, we observed that the KNN model's confusion matrix after SMOTE processing showed high prediction accuracy for the majority class (1), but its ability to predict the minority class (-1) remained insufficient. Specifically, the KNN model achieved an F1 Score of 0.404878, which, although improved compared to the pre-SMOTE results, still fell short of the desired level. The Logistic Regression model performed relatively better after SMOTE processing, with an F1 Score of 0.3754045, suggesting enhanced predictive capability on the balanced dataset, though there is still room for improvement.

Finally, we attempted to use XGBoost combined with the SMOTE method for prediction. The XGBoost model demonstrated significantly better performance than the other models after SMOTE processing. Its confusion matrix revealed high prediction accuracy for the majority class (1), along with improved predictive capability for the minority class (-1). The XGBoost model achieved an F1 Score of 0.3743316 and an ROC-AUC value of 0.6068303, indicating that the model exhibited balanced predictive performance on the dataset and outperformed the other models in overall performance.

## 5.1 LASSO

For the LASSO part, we trained the model with train set and found the best lambda (See the Fig below)



**Fig 20.** LASSO CV Plot

And the best lambda turns out to be:

best\_lambda = 0.00798771359450813.

Then, we constructed the LASSO Model with the best lambda (lambda\_min) and predicted the predict set from session 1 and session 18. The Confusion Matrix and other results are shown below.

### Confusion Matrix and Statistics

#### Reference

Prediction	-1	1
-1	0	0
1	87	243

Accuracy : 0.7364

95% CI : (0.6853, 0.7831)

As we can see, the results predicted all data to success 1. Though the accuracy of the model is 0.7364, which is high enough, but it is identical to the no information rate (the accuracy achieved by always predicting the majority class). This indicates that the model's performance is **no better than random guessing** based on the majority class.

We then took a look at the distribution of training set and found:

```
The training set distribution:  
y_train  
0    1  
1386 3365  
The predicting set distribution:  
y_test  
0    1  
87  243
```

The results clearly indicate that the LASSO model struggles with the **class imbalance problem**. The model's inability to predict the minority class (-1) and its reliance on the majority class (1) for achieving accuracy suggest that the data distribution is heavily skewed. This imbalance leads to biased predictions, where the model prioritizes the majority class at the expense of the minority class.

By searching and studying new method to address this issue, we found **SMOTE (Synthetic Minority Over-sampling Technique)** . It can generate synthetic samples of the minority class to balance the distribution, then improving the model's ability to learn and predict both classes more efficiently.

## 5.2 SMOTE

We applied SMOTE Algorithm, having the result:

```
y_train_balanced  
-1      1  
4158 3365
```

Now with more confident, we tried to use LASSO with SMOTE once again, but also with an suboptimal result. Thus, we decided to apply other models.

## 5.3 KNN

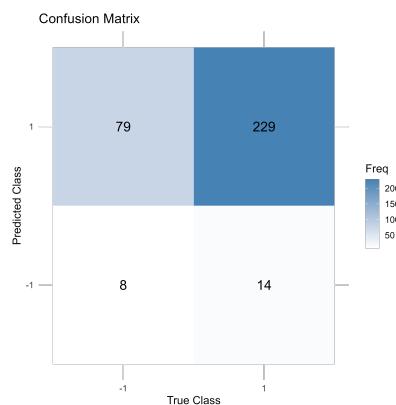
### 5.3.1 KNN without SMOTE

Firstly, we trained a KNN model with  $k \in \{3, 5, 7, 9, 11, 13, 15\}$  as candidates. The package “caret” will help us to find out the best performed  $k$  with 10 folds.

The result of KNN without SMOTE is shown below.

```
Confusion Matrix and Statistics  
  
Reference  
Prediction -1      1  
-1      8     14  
1       79    229  
  
Accuracy : 0.7182  
95% CI : (0.6663, 0.7661)
```

The confusion matrix is also visualized below.



**Fig 21.** Confusion Matrix

Moreover, we compute F1 score of it, 0.147, which means the predictive model has small ability to generalize with more situations. Then we turn to KNN with SMOTE.

### 5.3.2 KNN with SMOTE

With SMOTE, the KNN model's result also shown below.

```
Confusion Matrix and Statistics
Reference
Prediction -1    1
-1    83  240
1     4    3
Accuracy : 0.2606
95% CI  : (0.2141, 0.3115)
```

The accuracy dropped to a bad situation, but with a higher *F1* score: 0.404878.

## 5.4 Logistic Regression with SMOTE

```
Confusion Matrix and Statistics
Reference
Prediction -1    1
-1    58  164
1     29  79
Accuracy : 0.4152
95% CI  : (0.3614, 0.4704)
```

As we can see, the accuracy gets higher, 41.52%, and the *F1* score is 0.3754045.

## 5.5 XGBoost

Now we apply a new statistic method called XGBoost as our predictive model. After training, the results are

```
Confusion Matrix and Statistics
```

```
Reference
```

```
Prediction -1 1
```

```
-1 35 65
```

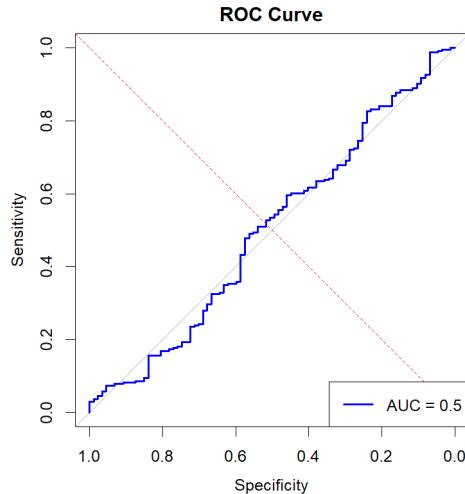
```
1 52 178
```

```
Accuracy : 0.6455
```

```
95% CI : (0.5912, 0.6971)
```

```
F1 Score: 0.3743316
```

And the ROC curve is also shown below. Although the accuracy is not the best, the F1 Score



**Fig 22.** ROC Curve

and the accuracy has a better performance all together, which have tried our best.

## 6 The Predictive Model used in the Reference

In the reference [1], a **kernel method** was employed to analyze the firing patterns of individual neurons: The activity of neuron  $n$  at time  $t$  is modeled by the following equation:

$$f_n(t) = \sum_c \sum_{t_s \in S_c} K_{c,n}(t - t_s) + \sum_{t_m \in M} (K_{m,n}(t - t_m) + D_m K_{D,n}(t - t_m)),$$

where  $K_{c,n}$ ,  $K_{m,n}$ , and  $K_{D,n}$  are kernel functions,  $S_c$  and  $M$  are sets of event times, and  $D_m$  is a scaling parameter.

This approach involved fitting a set of kernel functions to the neural activity, time-locked to specific task events such as **stimulus presentation** and **movement onset**. The kernel functions were designed to capture variations in neural activity driven by different task variables, including visual stimuli, actions, and choices. Specifically, **six vision kernels** were fitted to represent the amplitude and timing of neural responses to different visual contrast levels, while **a single action kernel** captured non-specific movement-related activity. Additionally, **a choice kernel** was introduced to distinguish between left and right movement decisions. To determine the significance of these kernels, **a nested testing framework** was used, where models with and without specific kernels were compared to assess their predictive power. See more details in ref[1][4][5][6].

This kernel-based approach provided a systematic way to identify how neurons encode task-relevant information, revealing the distributed nature of neural representations underlying behavior. The method's ability to capture both average activity patterns and trial-to-trial variability made it a powerful tool for understanding the functional roles of neurons in complex behavioral tasks.

However, due to time constraints and the complexity involved, we are unable to delve deeper into this sophisticated model based on kernel fitting at this stage. Kernel fitting requires precise time-locked analysis of neuronal activity, as well as substantial computational resources and data preprocessing efforts, which are beyond the scope of our current research. Therefore, we were unable to replicate the full model presented in the paper. However, this does not hinder our understanding and application of the methodological principles outlined in the article. In the future, with more resources and time, we can further attempt to implement and validate this approach.

## 7 Conclusion

This project aimed to develop a predictive model to classify behavioral outcomes (success or failure) based on neural activity and stimulus conditions in a visual discrimination task performed by mice.

Through a comprehensive analysis, we explored the structure of the data, integrated neural activity across sessions, and trained multiple machine learning models to predict trial outcomes.

Key findings include the identification of significant variability in neural activity and task performance across sessions and mice, as highlighted by the box plot and the Kruskal-Wallis test. To address this variability, we employed data standardization, PCA for dimensionality reduction, and the Harmony algorithm for cross-session feature alignment. These steps ensured the robustness of our predictive models, which included LASSO, KNN, Logistic Regression, and XGBoost, with SMOTE used to mitigate class imbalance.

The results demonstrate that neural activity in the visual cortex encodes information relevant to perceptual decision-making, and our models successfully captured these relationships. The integration of advanced statistical and machine learning techniques not only improved model performance but also provided insights into the neural mechanisms underlying decision-making.

This work contributes to the growing field of neuroscience by offering a methodological framework for analyzing high-dimensional neural datasets and advancing our understanding of how sensory input and neural dynamics translate into behavior. Future research could explore additional neural regions, refine model architectures, and investigate the generalizability of these findings to other decision-making tasks.

## References

- [1] Steinmetz N A, Zatka-Haas P, Carandini M, et al. Distributed coding of choice, action and engagement across the mouse brain[J]. *Nature*, 2019, 576(7786): 266-273.
- [2] Brain diagrams were derived from the Allen Mouse Brain Common Coordinate Framework (version 3 (2017); downloaded from [http://download.alleninstitute.org/informatics-archive/current-release/mouse\\_ccf/](http://download.alleninstitute.org/informatics-archive/current-release/mouse_ccf/))
- [3] Jun, J. J. et al. Fully integrated silicon probes for high-density recording of neural activity. *Nature* 551, 232–236 (2017).
- [4] Park, I. M., Meister, M. L. R., Huk, A. C. & Pillow, J. W. Encoding and decoding in parietal cortex during sensorimotor decision-making. *Nat. Neurosci.* 17, 1395–1403 (2014).
- [5] Ashe, J. & Georgopoulos, A. P. Movement parameters and neural activity in motor cortex and area 5. *Cereb. Cortex* 4, 590–600 (1994).
- [6] Paninski, L., Shoham, S., Fellows, M. R., Hatsopoulos, N. G. & Donoghue, J. P. Superlinear population encoding of dynamic hand trajectory in primary motor cortex. *J. Neurosci.* 24, 8551–8561 (2004).

## A Tables

### A.1 Session-Wise Summary Table

session_id	mouse	n_neurons	n_trials	success_rate
2016-12-14	Cori	734	114	0.6053
2016-12-17	Cori	1070	251	0.6335
2016-12-18	Cori	619	228	0.6623
2017-11-01	Forssmann	1769	249	0.6667
2017-11-02	Forssmann	1077	254	0.6614
2017-11-04	Forssmann	1169	290	0.7414
2017-11-05	Forssmann	584	252	0.6706
2017-06-15	Hench	1157	250	0.6440
2017-06-16	Hench	788	372	0.6855
2017-06-17	Hench	1172	447	0.6197
2017-06-18	Hench	857	342	0.7953
2017-12-05	Lederberg	698	340	0.7382
2017-12-06	Lederberg	983	300	0.7967
2017-12-07	Lederberg	756	268	0.6940
2017-12-08	Lederberg	743	404	0.7649
2017-12-09	Lederberg	474	280	0.7179
2017-12-10	Lederberg	565	224	0.8304
2017-12-11	Lederberg	1090	216	0.8056

**Tab 3.** Session-wise Summary