In The Name Of God

Sharif University of Technology

Department of Electrical Engineering

Neuroscience Final Project

by
Alireza Garegoori Motlagh
98102176
&
Ali Nourian
98102527

Professor :
Dr. Karbalei Aghajan

semester 1399-2

Contents

1 Introduction with Main Article **3**
- 1.1 The Main Goal of The Paper . . . 3
- 1.2 Complex Neurons . . . 3
- 1.3 Spike-Triggered Correlations Analysis . . . 3
- 1.4 Control Correlation Matrix . . . 4
- 1.5 Segregation Between Two Types of Visual Features . . . 4

2 Introduction with Dateset **5**
- 2.1 Information in the header of .sa0 and .log files . . . 5
- 2.2 Func_ReadData . . . 5
- 2.3 Spike-count rate . . . 6
- 2.4 Func_StimuliExtraction . . . 7
- 2.5 tview.m . . . 7

3 Spike-Triggered Average Method **9**
- 3.1 Finding receptive field by STA . . . 9
- 3.2 p-value matrix (Optional) . . . 10
- 3.3 STA Correlation . . . 11
- 3.4 t-test on Spike and Control (Optional) . . . 12
- 3.5 ON-OFF threshold . . . 12
- 3.6 Results of sec.3.1 to sec.3.5 for some neurons . . . 13
- 3.7 Simple or Complex Neuron . . . 15

4 Spike-Triggered Correlation Method **16**
- 4.1 Receptive Fields Using Eigenvectors . . . 16
- 4.2 Confidence Interval . . . 17
- 4.3 Results of sec.4.1 and sec.4.2 . . . 17
- 4.4 Histogram of Joint Distributions . . . 18
- 4.5 . . . 19
- 4.6 Results of sec.4.1 to sec.4.5 . . . 20
- 4.7 Comparison between STA and STC . . . 24

5 A Question:) **25**

1 Introduction with Main Article

1.1 The Main Goal of The Paper

An important goal in studying the receptive-field properties of visual neurons is to understand how they respond to complex spatiotemporal inputs, including those encountered in natural scenes.

Several methods have been used to define basis sets for the efficient representation of visual stimuli, including principal component analysis (PCA), independent component analysis, and/or analysis based on sparse coding.

For neurons with a linear stimulus–response relationship, relevant visual features can be identified by estimating their linear receptive fields using a spike-triggered average of the stimulus ensemble (also called "reverse correlation") . This method has been widely used to measure the spatiotemporal receptive fields of neurons in the early visual pathway; the resulting receptive fields can largely account for the neuronal responses to complex spatiotemporal stimuli. However, in the visual cortex most of the neurons are complex cells with nonlinear stimulus–response relationships that cannot be characterized with the spike-triggered average.

In the present study, we have used spike-triggered correlation analysis of the stimulus ensemble to construct the basis set for each complex cell.

1.2 Complex Neurons

Unit isolation was based on the cluster analysis of waveforms and the presence of a refractory period determined from the autocorrelograms. Cells were classified as simple if their receptive fields had clear on and off sub-regions and if the ratio of the first harmonic to the DC component of the response to an optimally oriented drifting grating was >1. All other cells were classified as complex. Among the 61 complex cells recorded, one was excluded from analysis because of its low firing rate in response to random-bar stimuli (<1 spike per second).

1.3 Spike-Triggered Correlations Analysis

In general, if certain features in the visual stimuli affect the firing probability of the cell, the spike-triggered stimulus ensemble should exhibit a different probability distribution from the entire stimulus ensemble. Although a change in the probability distribution can be reflected in a change in the first-order (mean), second-order (variance), or higher-order moments, the correlation analysis aims to identify features with changed variance. Because PCA results in a set of components with their variance ranking from the highest to the lowest, it is ideally suited for the identification of features with outstanding variance. Practically, identification of relevant features was achieved by finding eigenvalues of the spike-triggered correlation matrix that were significantly different from the eigenvalues of the control correlation matrix (computed by randomly sampling the entire stimulus ensemble).

The spike-triggered correlation matrix was computed as follows:

$$C_{mn} = \frac{1}{N} \sum_{i=1}^{N} S_m(i) S_n(i)$$

where Sm(i) and Sn(i) are the mth and nth parameters of the stimulus pattern preceding the ith spike, respectively, and N is the total number of spikes in the response. Eigenvalues and eigenvectors of this spike-triggered correlation matrix were then computed. To compute each control correlation matrix, we generated a random spike train with the same number of spikes as in the recorded response but with random spike timing; the correlation matrix was computed based on this simulated random spike train.

1.4 Control Correlation Matrix

As we mentioned in the previous question, spike-triggered ensemble exhibits significantly different eigenvalues from the control correlation matrix. The article suggests that by using this idea and finding the projection of each trigger on the eigenvectors, as well as the projection of each spike-triggered ensemble on these eigenvectors, we can come up with a threshold to determine the type of a trigger.

1.5 Segregation Between Two Types of Visual Features

For most (47 of 60) of the complex cells studied, we found two significant eigenvectors that exhibit a much larger eigenvalue than others.

These two eigenvectors exhibited separate on and off spatial sub-regions, resembling the receptive fields of simple cells. In a few cases (3 of 60), we found only one significant eigenvector for each complex cell; these vectors also exhibited spatiotemporal profiles resembling simple cell receptive fields. In the remaining cases, more than two eigenvalues reached significance. However, these additional eigenvectors (corresponding to third, fourth, ..., largest eigenvalues) tended to exhibit much less spatiotemporal structure than the first two eigenvectors, and their eigenvalues were much smaller, suggesting less functional importance

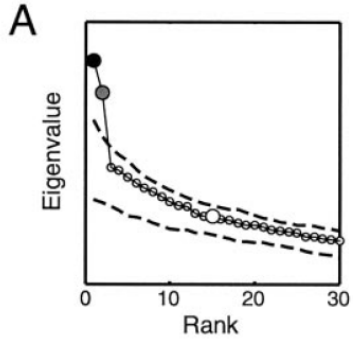

Figure 1

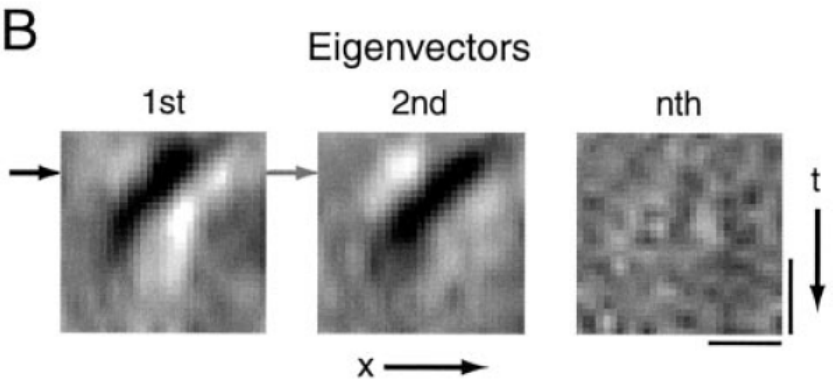

Figure 2

2 Introduction with Dateset

2.1 Information in the header of .sa0 and .log files

These information are :
1. FileInfo which includes data file information including :
   - name
   - type
   - date
   - creator
   - version.
2. DataInfo :which includes the parameters below :
   - ID : the unique name of the data
   - DataForm : data format
   - Channel
   - SampleRate : equals to 10KHz
   - Some features of time signal
   - Some thresholds
   - and other datas which were not used in this project.

2.2 Func_ReadData

In this part we write a function which takes the code of the neuron in string, and then gives us a struct that contains the vector of events(times when spike has occurred) and the header of that experiment.

This function first gets all the files which matched to the code in the input; separates the events and header of the files, Then puts all of them in a struct and gives it as output.

The code of this function is as follows :

```
function Output = Func_ReadData(nCode)
   str = ['Data\Spike_and_Log_Files\',nCode,'\'];
   filesname = strcat(str,nCode,'*','msq1D.sa0');

   files = dir(filesname);

   Output = struct;
   for i=1:length(files)
      currentFile = files(i);
      [events,hdr] = fget_spk(strcat(str,currentFile.name),'hdr');
      Output(i).events = events;
      Output(i).hdr = hdr;
   end
end
```

'fget_spk' is the function for getting the events and header by the neuron code which exists in the 'MatlabFunctions' folder.

2.3 Spike-count rate

Now we should calculate the spike-count rate of a neuron. The function below takes the struct which prepared in the section 2.2 and calculates the spike-count rate for each events and then gives the averages of these rates. The code of this function is as follows :

```
function rate = getSpikeCountRate(input)
   Fs = 59.721395;          % refer to the paper
   N = 32767;               % numbers of stimuli
   rate = 0;
   L = length(input);
   for i=1:L
      event = input(i).events;
      temp = length(event) / (N / Fs);
      rate = rate + (1/L * temp);
   end
end
```

By reading all the neurons code and save them in a cell and using the function above, we calculate the spike-count rate for all neurons in a loop an save them in a vector. Then we plot the histogram of rates of the neurons. the result is as follows :

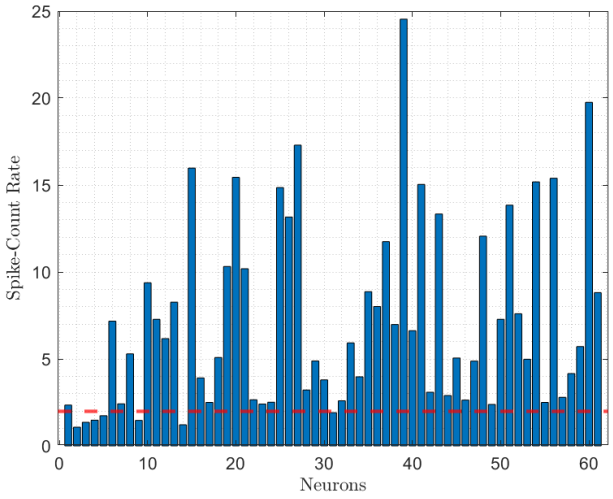

Figure 3

As it mentioned, the neurons with spike-count rate less than 2 must be removed form dataset. In this case we set the threshold - as it showed in figure 3 - at 2 and saved the neurons with the spike-count rate more than 2 in the.

The neurons which were removed has the following codes :

```
Removed neurons code :
'000413.b03'
'000413.b04'
'000413.b05'
'000418.a01'
'000420.b02'
'000524.c01'
'000907.f07'
```

2.4 Func_StimuliExtraction

In this part we want to epoch the data 'msq1D.mat'. The result should be $16 \times 16 \times N$ where N is the number of spikes that occurred. when a spike occurs, we takes the 16 latest stimuli - that we assume they make the neuron to spike - and each stimulus has 16 white and black bars. So for each spike we has 16×16 stimuli. Therefore the epoch is $16 \times 16 \times N$.
The function 'Func_StimuliExtraction' takes the data and the events(times when spike occurred) and gives the epoch which explained above.

The code of this function is as follows :

```
function SpikeTriggeredStimuli = Func_StimuliExtraction(events, trigger)
   N = length(events);
   SpikeTriggeredStimuli = zeros(16,16,N);
   Fs = 59.721395;
   ind = ceil( (events*Fs) / 1e04);

   for i=1:N
      if (ind(i)>15 && ind(i)<32767)
         SpikeTriggeredStimuli(:,:,i) = trigger(ind(i)-15:ind(i),:);
      end
   end
end
```

2.5 tview.m

The function 'tview.m' takes the file name for the tuning stimulus and gives the result of the tuning experiment.In the following, we can see the output of this function to some samples neurons :

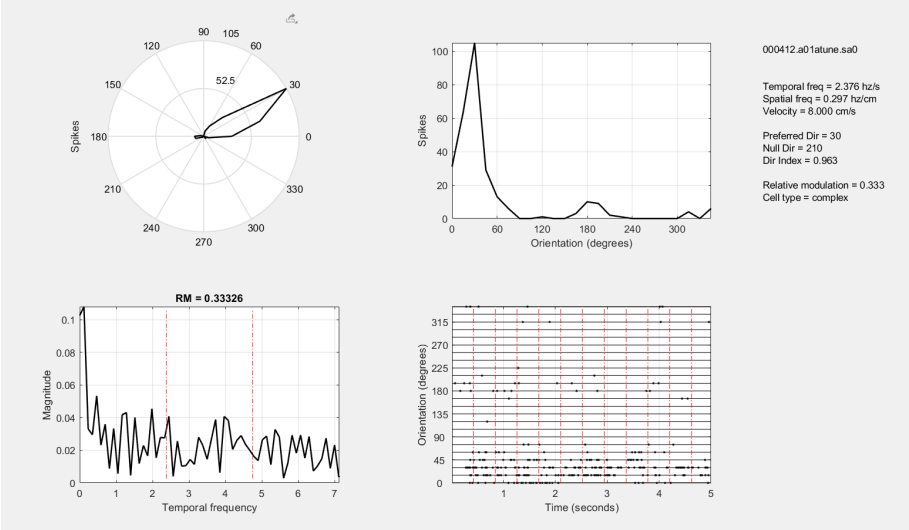

Figure 4: neuron code : 000412.a01atune

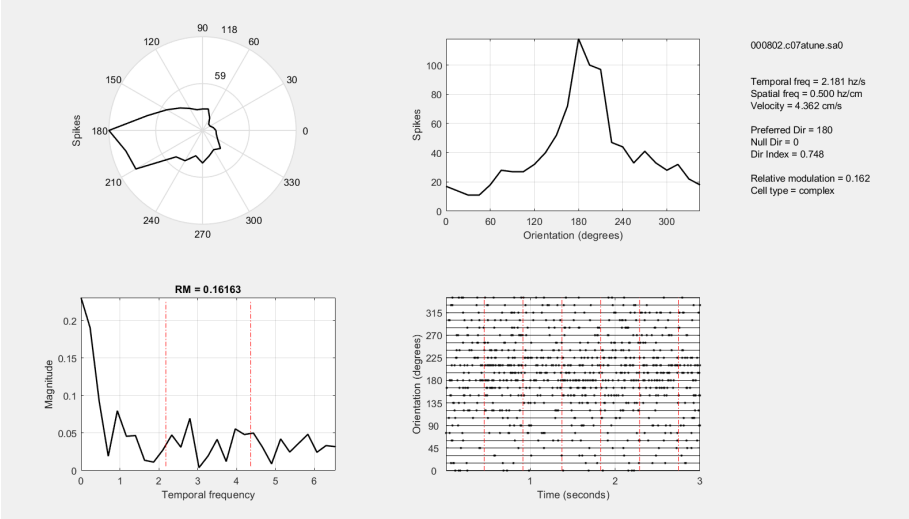

Figure 5: neuron code : 000802.c07atune

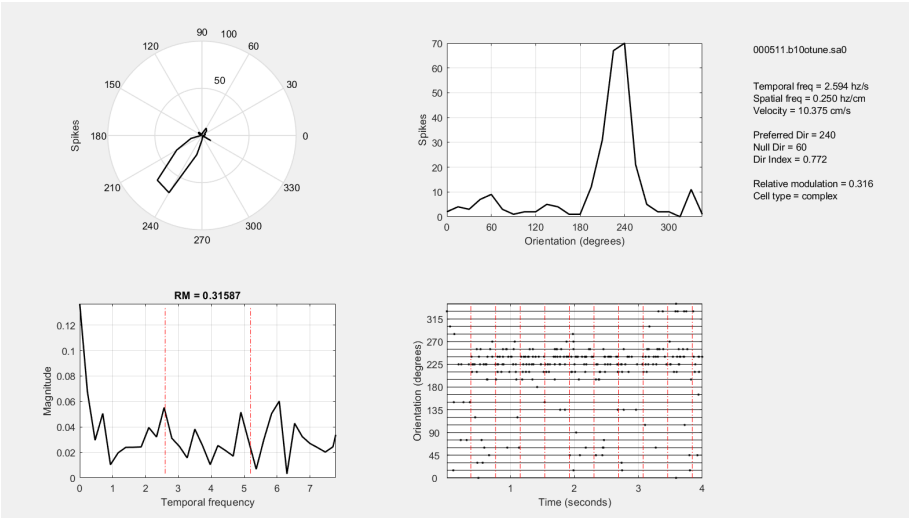

Figure 6: neuron code : 000511.b10otune

As we see in the figures above to find the best orientation they test the stimulus in different orientations and record the response of the neuron. The result is as the stimulus becomes closer to the orientation that is in the receptive field of the neuron, the neuron starts spiking and as it placed in the receptive filed of the neuron, the neuron has the most firing rate. Therefore by looking at the spiking in different orientations we can find the best orientation and the receptive filed of the neuron. For example for figure 4, as we see, the best orientation is 30 degrees and for figure 5 this is 100 degrees. But an interesting difference between these two subjects is the area of their receptive fields. The first neuron has the smaller receptive filed so it spikes for smaller range os orientations that is closer to it's best orientation while the second one spikes for larger range of orientations.

3 Spike-Triggered Average Method

3.1 Finding receptive field by STA

As the result of section 2.4, by averaging in third Dimension of the epoch we get the receptive field of the neuron. Then we show this by 'imshow' in the range of $[-1, 1]$. The code and the result is as follow :

* for all the parts we worked on the neuron '000601.c05'.

```
load 'msq1D.mat';
Fs = 59:721395;

Output = Func_ReadData('000601.c05');

temp = []; Stimuli = []; Max = [];
for i=1:length(Output)
   event = Output(i).events;
   temp = Func_StimuliExtraction(event, msq1D);
   Stimuli = cat(3,Stimuli,temp);
   Max = cat(1,Max,max(Output(i).events));
end

receptive_field = mean(Stimuli,3);
imshow(receptive_field,[-1 1]); xlabel('Spatial'); ylabel('Temporal');
```

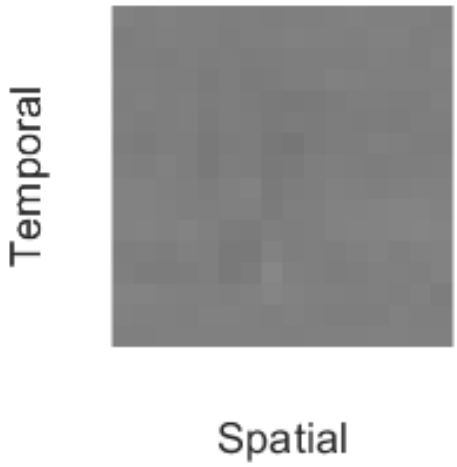

Figure 7: receptive field (STA)

3.2 p-value matrix (Optional)

In this part we want to apply t-test to each index of the STA matrix. For doing this we use a loop on the size of third dimension of our epoch and for each element $a_{i,j}$ of 16×16 matrix of stimuli, we define a vector that has $a_{i,j}$ of all the stimuli. then we apply the t-test function in this vector and get the 'h' and the 'p-value' of this vector and put the 'p-value' in the index of i, j in p-value matrix. The code and the result is as follows :

```
p_value = zeros(16,16) ;
for i = 1:16
   for j = 1: 16
      [a, p_value(i,j)] = ttest(Stimuli(i,j,:)) ;
   end
end
figure;
subplot(121); imshow(receptive_field,[-1 1]);
subplot(122); imshow(1-p_value);
```

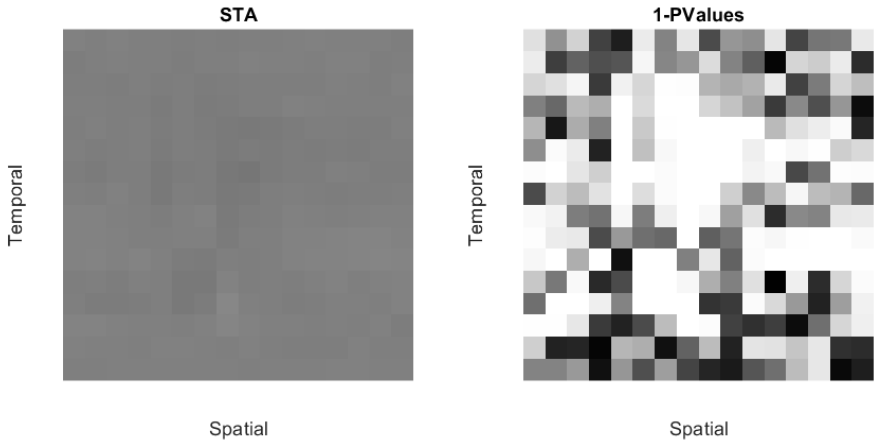

Figure 8: receptive field (STA)

3.3 STA Correlation

In the section before we calculate STA and now we want to calculate the correlation of each stimulus with STA. As we know to STA and each stimulus are like a vector in 256D space. So for calculating the correlation we just have to calculate the inner product of the stimulus and the STA. After calculating this parameter for all the stimuli we should create the 'Spike' vector by these values. Also we should create the 'Control' vector in the same way but from random spikes.

Here is the function which we write to calculate the 'Spike' and 'Control' :

```
function [control,spike] = getControlAndSpike(Stimuli,Max,msq1D)
   control = [] ;
   spike = [] ;
   STA = mean(Stimuli,3);

   for i=1:length(Stimuli)
      temp = sum((STA).*(Stimuli(:,:,i)),'all') ;
      spike = cat(1,spike,temp) ;
   end

   control_events = ceil(max(Max).*rand(1,length(Stimuli)));
   control_matrix3D = Func_StimuliExtraction(control_events,msq1D) ;
   for i=1:length(control_matrix3D)
      temp = sum(STA .* control_matrix3D(:,:,i),'all');
      control = cat(1,control,temp);
   end
end
```

The result of ploting the histogram of the 'Spike' and 'Control' :

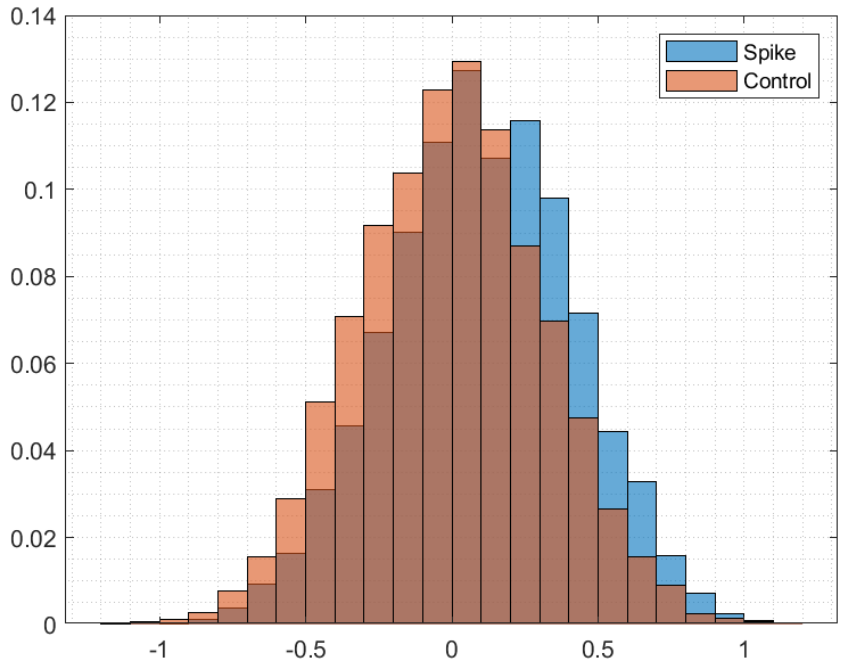

Figure 9

3.4 t-test on Spike and Control (Optional)

Now we want to apply t-test to 'Spike' and 'Control' vectors that we calculate in the section before. The code is :

```
[h_spike, p_spike] = ttest(Spike);
[h_control, p_control] = ttest(Control);

fprintf('h_spike = %d,\t p_spike = %f\n\n',h_spike, p_spike);
fprintf('h_control = %d,\t p_control = %f\n\n',h_control, p_control);
```

And the result is :

```
h_spike = 1,  p_spike = 0.000000

h_control = 0, p_control = 0.767331
```

As we expect, 'h' for 'Spike' is 1 and the p-value is almost zero. Also the 'h' for 'Control' is zero and the p-value is close to 1. It Shows that the correlation between 'Spike' and the STA is not by chance and there is a significant correlation between them. Also we can conclude the upside for 'Control' vector.

3.5 ON-OFF threshold

We have two gaussian distributions with overlap. The intersection of these two gaussian distributions is where the probability of classified the random stimuli in stimulation on the condition of spike is equal to the probability of classified the random stimuli in random stimuli. We take this value as the threshold. So we can't properly recognize 50 percent of the stimulation on the condition of spike.

The code below calculate the threshold and the probability of STA success.

```
mu1 = mean(Spike);
mu2 = mean(Control);

sigma1 = var(Spike);
sigma2 = var(Control);

dist1 = @(x) 1/sqrt(2*sigma1^2*pi) * exp(-(x-mu1).^2 / (2*sigma1^2));
dist2 = @(x) 1/sqrt(2*sigma2^2*pi) * exp(-(x-mu2).^2 / (2*sigma2^2));

threshold = fzero(@(x) dist1(x)-dist2(x), rand * (mu1-mu2) + (mu1+mu2));
STA_Success_Percent =
    length(find((dist1(Spike)-dist2(Spike))>0))/length(Spike);

fprintf('threshold=%f\n\n',threshold);
fprintf('STA_Success_Percent=%f\n\n',STA_Success_Percent);
```

Result :

```
threshold=0.051138

STA_Success_Percent=0.554424
```

3.6 Results of sec.3.1 to sec.3.5 for some neurons

In the following, we put the result of section 3.1 to section 3.5 for some neurons.

* Neuron code : '000412.a01'

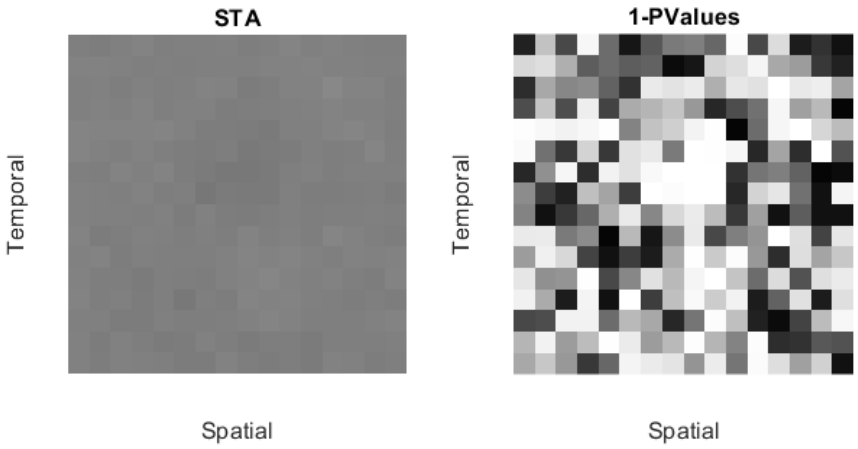

Figure 10

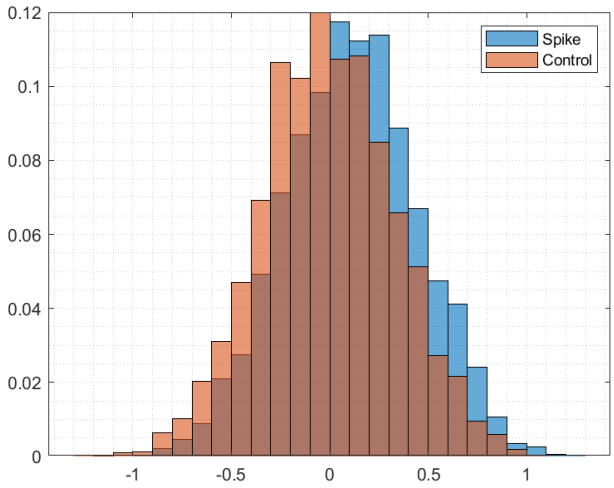

Figure 11

```
h_spike = 1,  p_spike = 0.000000

h_control = 0, p_control = 0.146340

threshold=0.052577

STA_Success_Percent=0.567242
```

* Neuron code : '000513.d11'

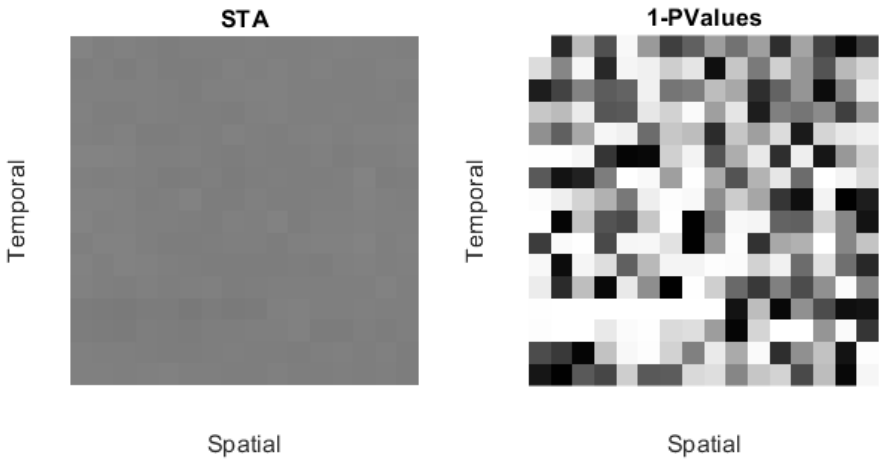

Figure 12

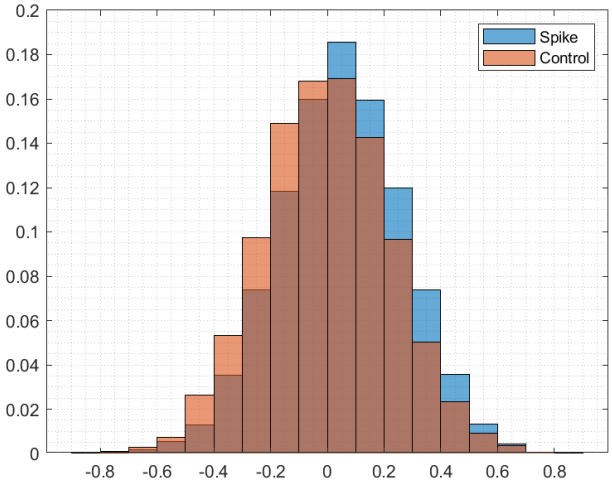

Figure 13

```
h_spike = 1,  p_spike = 0.000000

h_control = 0, p_control = 0.163741

threshold=0.021890

STA_Success_Percent=0.548555
```

* Neuron code : '020306.A.a02'

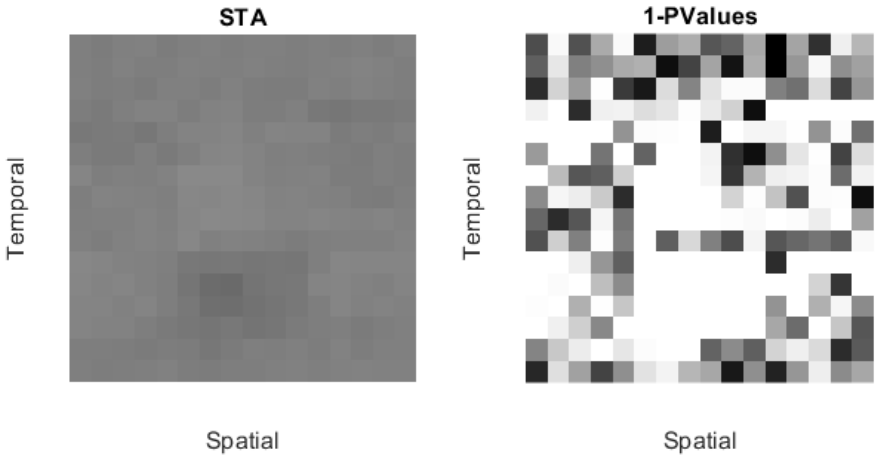

Figure 14

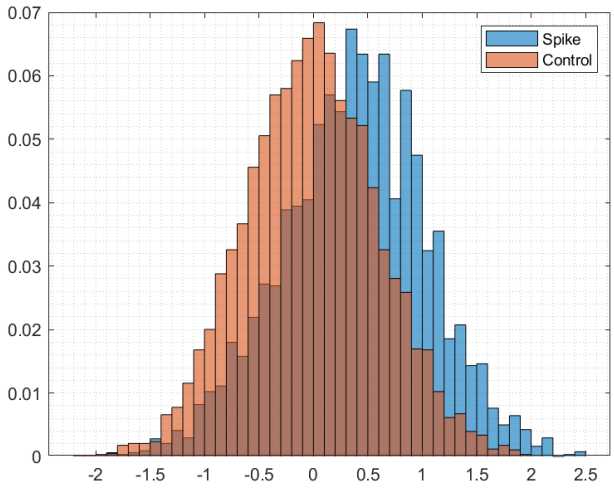

Figure 15

```
h_spike = 1,  p_spike = 0.000000

h_control = 0, p_control = 0.051300

threshold=0.233061

STA_Success_Percent=0.598539
```

3.7 Simple or Complex Neuron

As we see in the figures in sections before, the output of the STA method is generally random and it matches to the receptive field just in very small cases . So the claim of the article that tells the neuros are complex, is true.

4 Spike-Triggered Correlation Method

4.1 Receptive Fields Using Eigenvectors

As stated in section 1.3, Spike-Triggered Correlation matrix is computed as follows:

$$C_{mn} = \frac{1}{N} \sum_{i=1}^{N} S_m(i) S_n(i)$$

After implementing a method to get Spike-Triggered Correlation, we write a function called SpatioTemporal.m to find the Spike-Triggered Correlation and finds the receptive fields which are simply the eigenvectors of spike-triggered correlation matrix. We then plot the top 3 receptive fields and increase the contrast of the image for better visualization.

* Neuron code = '000412.a01'

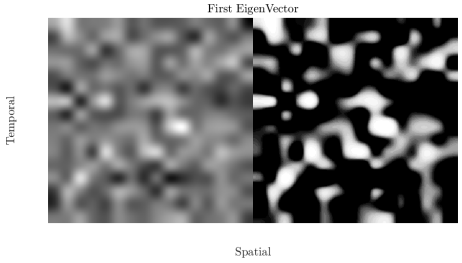

Figure 16

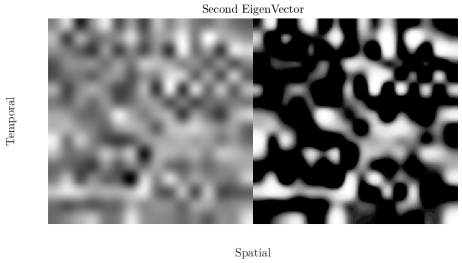

Figure 17

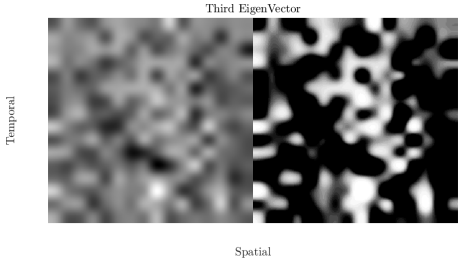

Figure 18

4.2 Confidence Interval

By creating the control matrix, we compute the eigenvalues of both correlation matrix and control matrix. Then using ±5.2 SD confidence interval, we may plot the results, which we get:

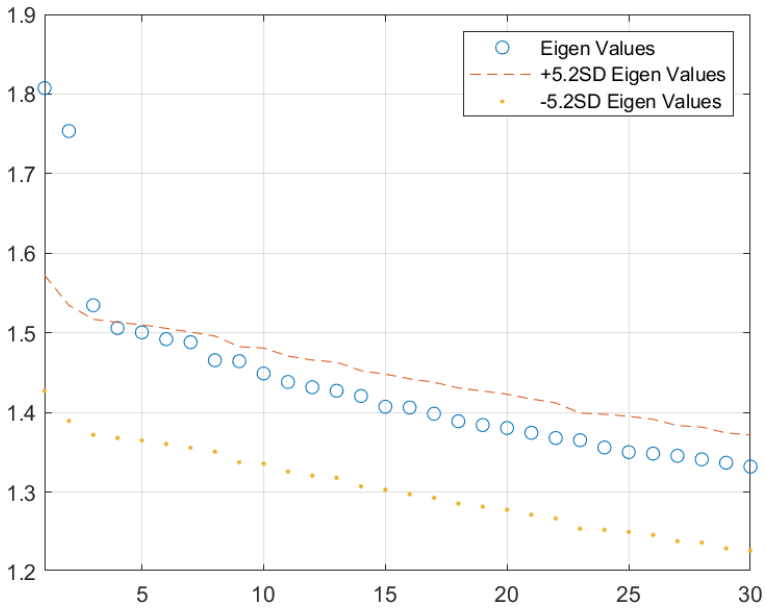

Figure 19

As we observe from the above figure, there are two significant eigenvalues for the spike-triggered correlation matrix. These are known as meaningful(significant) eigenvalues of STC matrix and their corresponding eigenvectors are used as principal components for this neuron.

4.3 Results of sec.4.1 and sec.4.2

As we are looking for vectors in which the principal components can be inferred, as the paper suggests, we are using PCA. We found that there are 2 eigenvalues which stood out from the rest and were significant; So the values on their direction have the most variance and other values can be calculated using linear combination of this principal components. As the paper says, we use the eigenvectors which are out of the confidence interval and their variance is significantly large or small. Form the above figure 2-3 of such eigenvalues can be recognized.

4.4 Histogram of Joint Distributions

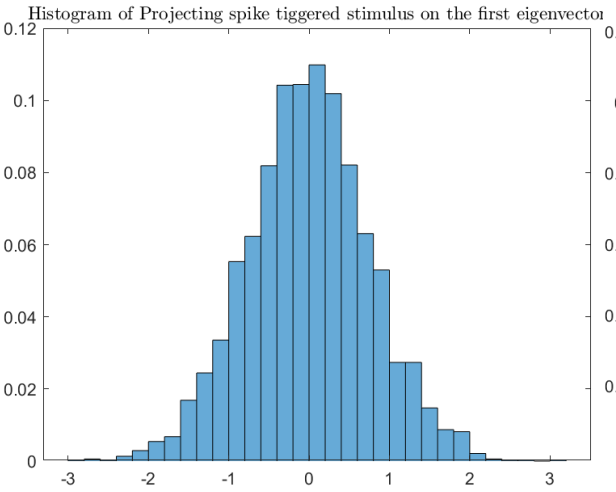

Figure 20

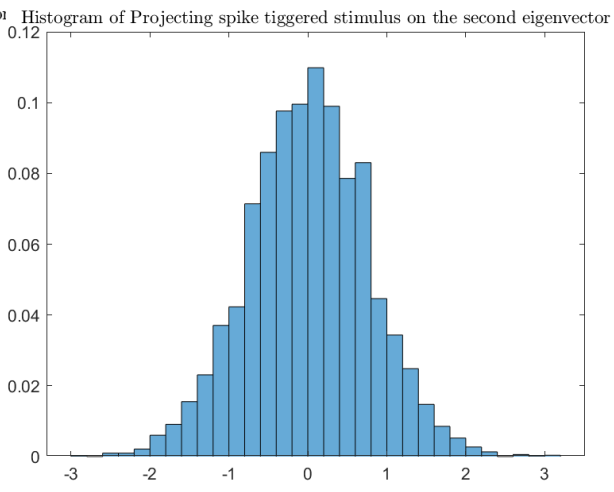

Figure 21

As it can be seen from the above figures, the histograms of projecting the spike-triggered stimulus on the first and second significant eigenvectors (principal components) approximately have a gaussian form with mean 0.

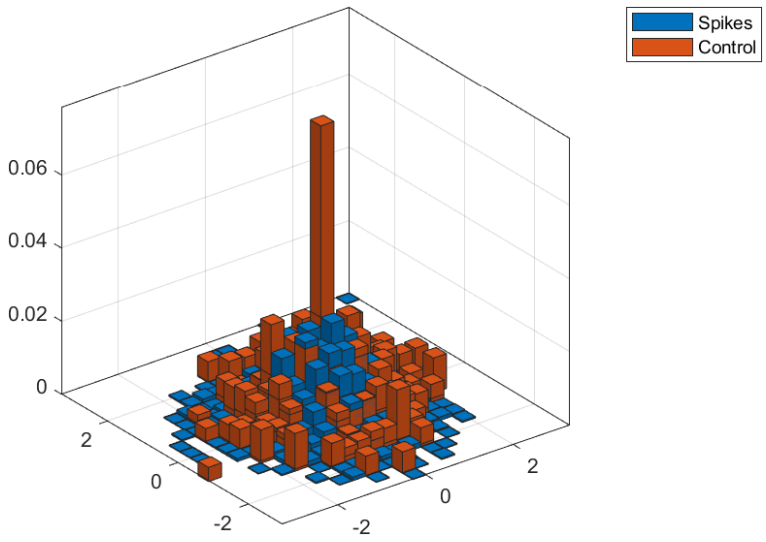

Figure 22

However, plotting the joint histograms is not the same as it was expected; since there is an almost random and somehow same(except the peak value for control matrix) joint distribution. Even by changing the way that control matrix is calculated we still have no meaningful result and the spikes joint histogram is totally under control ensemble:

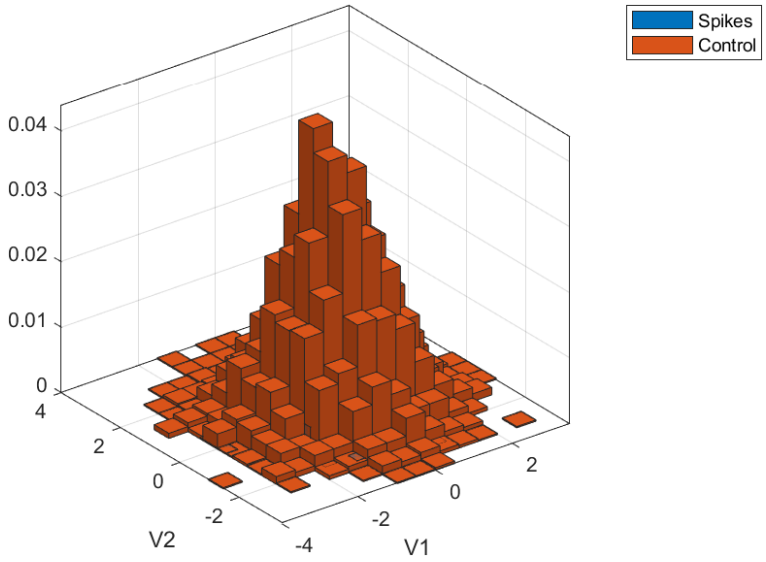

Figure 23

4.5

4.6 Results of sec.4.1 to sec.4.5

* Neuron code :'000601.c05'

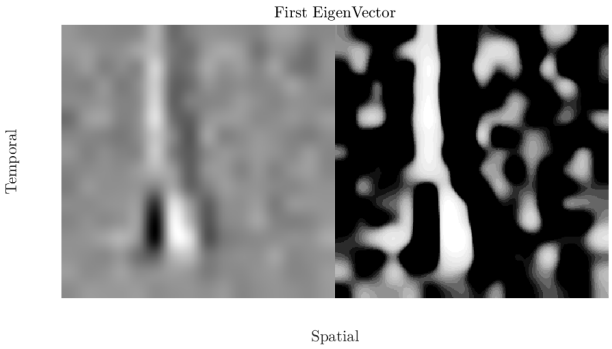

Figure 24

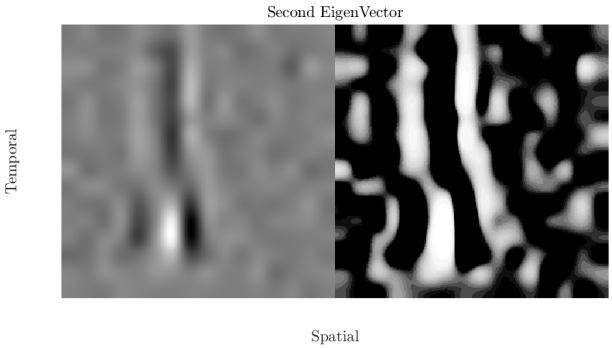

Figure 25

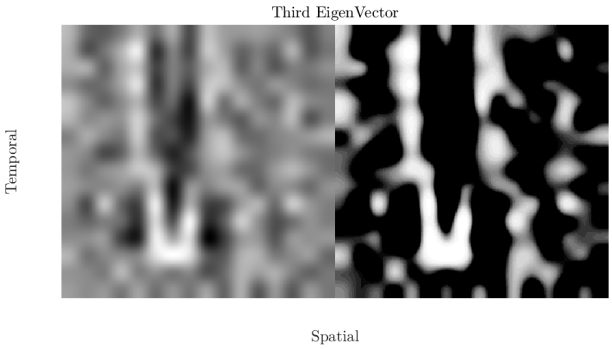

Figure 26

As it can be seen, on and off regions are obvious for this neuron.

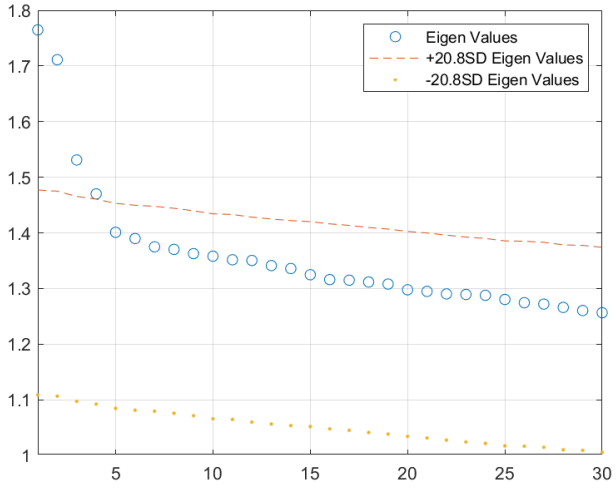

Figure 27

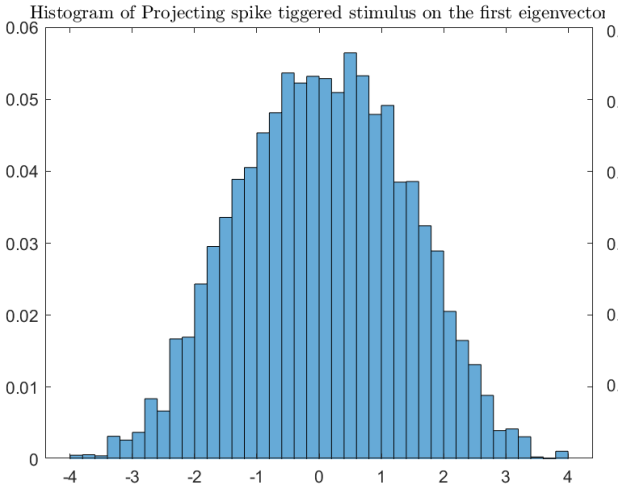

Figure 28

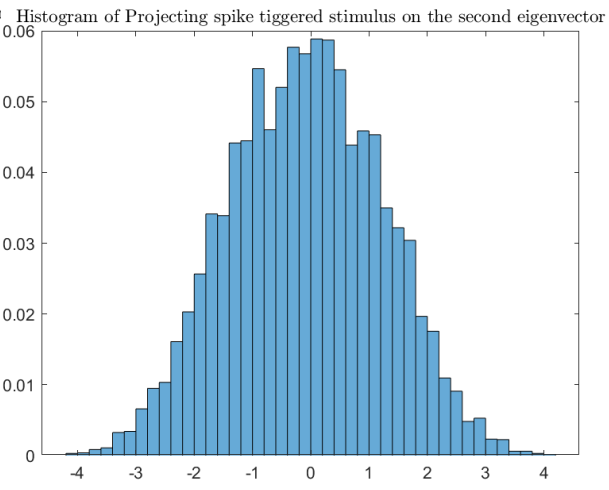

Figure 29

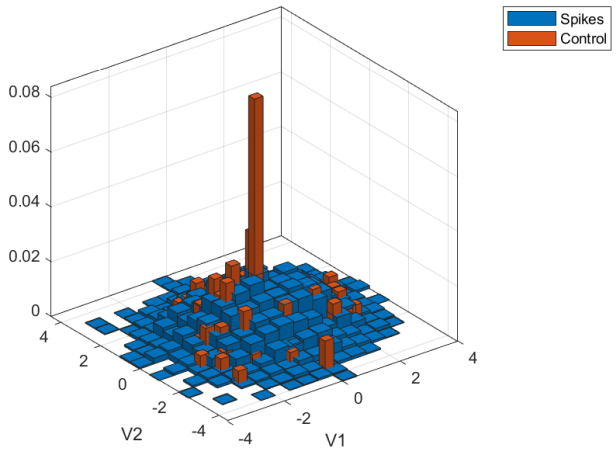

Figure 30

* Neuron code :'000824.g04'

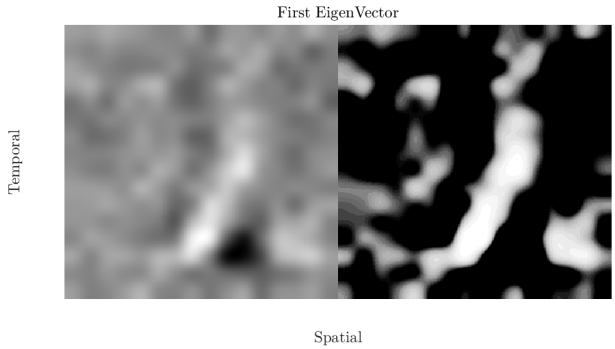

Figure 31

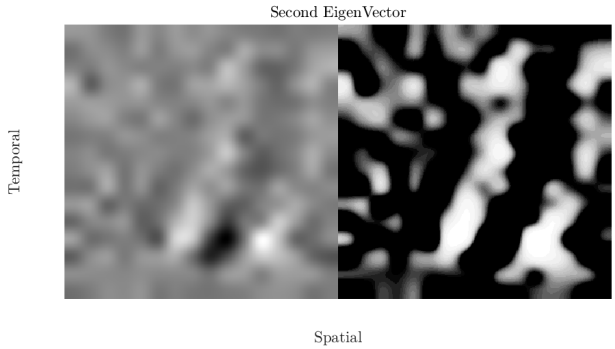

Figure 32

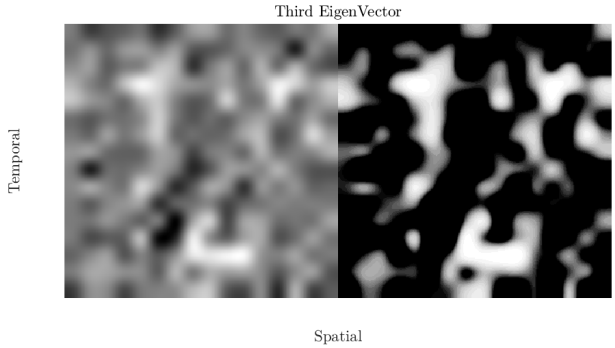

Figure 33

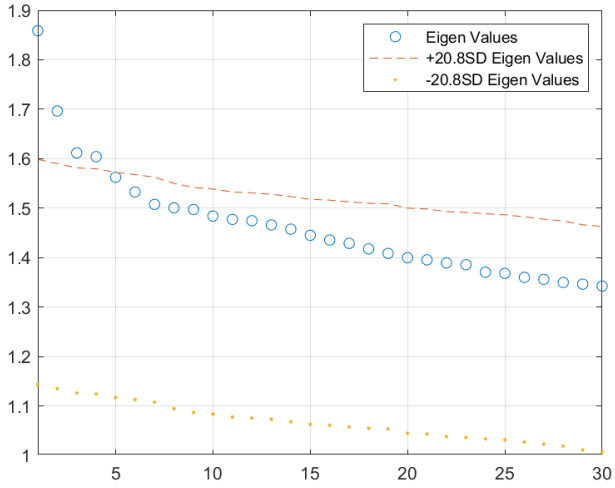

Figure 34

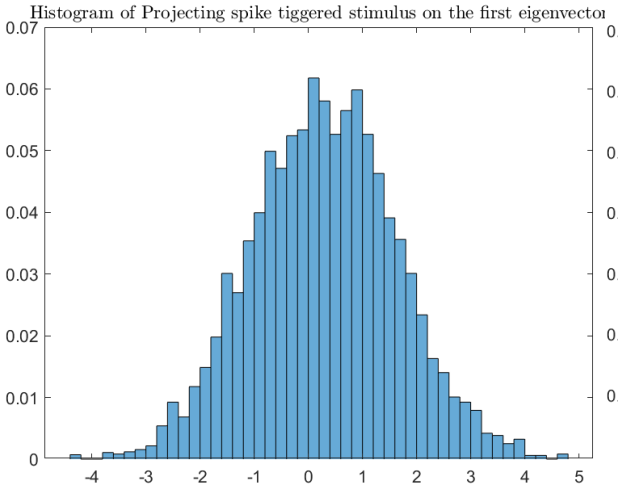

Figure 35

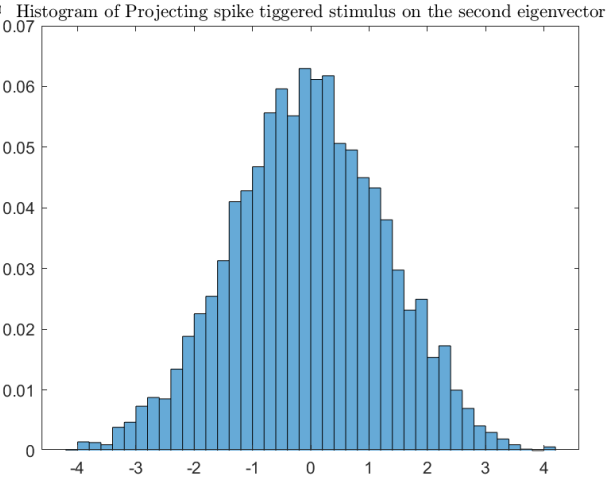

Figure 36

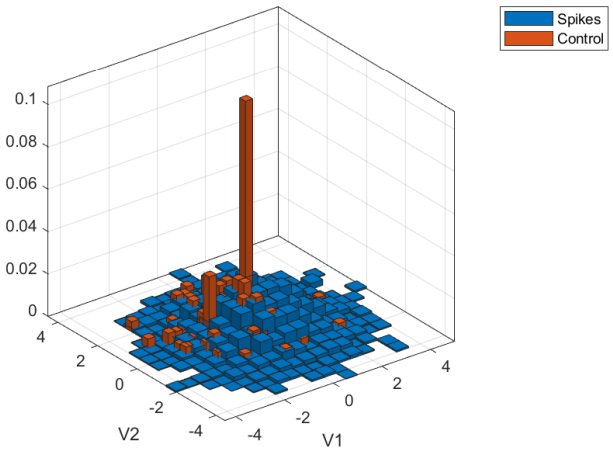

Figure 37

4.7 Comparison between STA and STC

In the spike-triggered correlation matrix, many neurons contained 2 or 3 significant eigenvectors in which the variance is much different when any stimuli is projected on their corresponding eigenvectors and hence they are used as principal components of analysis. However, in the STA analysis, the eigenvalues were generally random and contained no significance factor that could stood out from the rest; but in the STC analysis, there were significant eigenvalues with properties that were discussed. Also, STC analysis based on PCA was much more efficient for recognizing the receptive fields of each neuron. So, as the paper suggests, neurons were not simple but there were labeled as "complex" because of this property.

5 A Question:)

Based on the Hypothesis Testing and the use of P-Values in this project, and the undeniable rule of Statistics in a wide range of Neuroscience fields, we designed the following question:

Suppose we would like to test the hypothesis that at least 10% neurons of a large neuronal network are triggered by a specific stimuli and so, they have a common receptive field. We collect a random sample of 225 neurons and 21 of them had that same pattern for their receptive fields. Based on the given information:

(a) State the null and alternative hypothesis.

(b) Obtain a test statistic and a P-value.

(c) State the conclusion at the $\alpha = 0.05$ level.

Solution :

(a) We would like to test the hypotheses that $H_0 : \theta >= 0.1$ and $H_1 : \theta < 0.1$, which, since equality gives us a worse case scenario, can be simplified to :

$$H_0 : \theta = \theta_0 = 0.1$$

$$H_1 : \theta < \theta_0$$

(b) If we let $X_i = 1$ if the i^{th} neuron has the same receptive field, and 0 otherwise, we see that $X_i \sim Bern(\theta)$, so that $E[X_i] = \theta$ and $Var[X_i] = \theta(1-\theta)$. Now, under the null hypothesis, and under the CLT (since n is large), I have that

$$\frac{\bar{X}_n - \theta_0 n}{\sqrt{n\theta_0(1-\theta_0)}} \sim N(0,1)$$

This is a convenient test statistic to use since I have its distribution, and since if the alternative hypothesis is true, $\bar{X}$ will be large (and so will the statistic). This suggests the following test : if

$$\frac{\bar{X}_n - \theta_0 n}{\sqrt{n\theta_0(1-\theta_0)}} < c$$

then reject the null hypothesis in favor of the alternative hypothesis, while if

$$\frac{\bar{X}_n - \theta_0 n}{\sqrt{n\theta_0(1-\theta_0)}} \geq c$$

fail to reject the null hypothesis.

Calculating the value of the statistic for the particular instance of the data that we have collected :

$$w_1 = \frac{21 - 0.1 \times 225}{\sqrt{225 \times 0.1 \times (1-0.1)}} \approx -0.33$$

Now, the p-value is the probability of making a Type I error when the test threshold, c, is set to be w_1 :

$$\begin{aligned} p-\text{value} &= P(\text{Type I error with} c = w_1) \\ &= P(\text{Reject} H_0 \text{with} c = w_1 | H_0) \\ &= P(\frac{\bar{X}_n - \theta_0 n}{\sqrt{n\theta_0(1-\theta_0)}} < w_1 | H_0) \\ &= \Phi(w_1) \approx 0.37 \end{aligned}$$

(c) Since the p-value is the lowest significance level α that results in rejecting the null hypothesis, at a level of $\alpha = 0.05$, we cannot reject the null hypothesis.