

به نام خدا



دانشگاه صنعتی شریف

دانشکده مهندسی برق

علوم اعصاب یادگیری، شناخت و حافظه

پروژه‌ی پایانی درس: بررسی نورون‌های پیچیده غشر بینایی گربه

استاد: دکتر کربلایی آقاجان

امیرحسین برقراری ۹۷۱۰۴۸۹۴

مازیار شمسی‌پور ۹۸۱۰۱۸۴۴

۱۴۰۰ تیر ۲۷

۱ آشنایی با هدف پژوهش

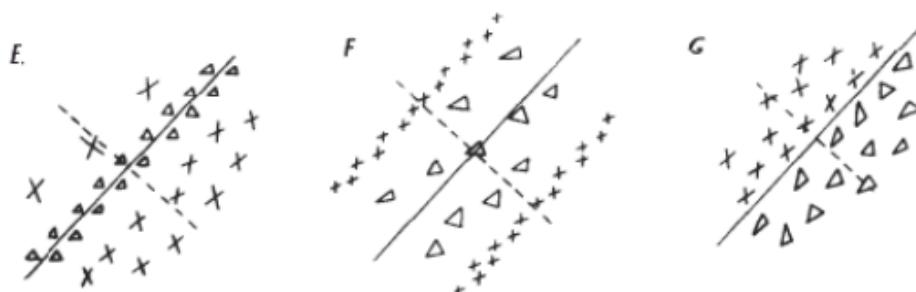
۱.۱ هدف پژوهش

هدف این مقاله مدل سازی receptive field نورون‌های پیچیده غش بصری^۱ می‌باشد. در واقع با اعمال تحریک‌های تصادفی spatiotemporal از توزیع $p(s)$ و مشاهده‌ی پاسخ نورون‌ها می‌خواهیم راستایی را بدست بیاوریم که در آن $p(s|r)$ تفاوت معناداری با توزیع اولیه‌ی تحریک‌ها داشته باشد. تفاوت این مقاله با پژوهش‌های قبل خود آن است که به دلیل رفتار غیر-خطی نورون‌های پیچیده‌ی موجود در Visual cortex نمی‌توانیم از آنالیز‌های خطی spike-triggered average که پیش از این کارگشا بودند استفاده کنیم. این مقاله روش spike-triggered correlation analyses را پیشنهاد می‌دهد که بر پایه‌ی Wiener Kernel طراحی شده است.

نهایتاً با این پژوهش با بررسی‌هایی که انجام می‌دهد ادعا می‌کند که می‌توان با این روش پایه‌ای برای تحریک‌ها ارائه داد که تعداد کمی از آن‌ها مشخص کننده‌ی ویژگی‌های مرتبط و تعداد زیادی مربوط به ویژگی‌های پوچ می‌باشند.

۲.۱ نورون‌های «پیچیده»

این مفهوم اولین بار در مقاله‌ی نوبلیست Hubel and Wiesel مطرح شده است. نورون‌های ساده نورون‌هایی هستند که رابطه‌ی تحریک-پاسخ آنها نسبت به زمان خطی می‌باشد. این موضوع باعث به وجود آمدن مناطق ON-OFF در Receptive Field نورون مورد نظر می‌شود. به عکس دست نوشته‌ی موجود در مقاله‌ی مذکور برای جزئیات بیشتر توجه کنید.



شکل ۱: مناطق ON-OFF حوزه‌ی دریافتی نورون‌های ساده^۲

همچنین در بخش Material and Methods آمده است که اگر نسبت هارمونیک اول به مقدار DC حوزه‌ی دریافتی بزرگتر از یک باشد نورون را ساده می‌نامیم. روشن است که نورون‌هایی که از تعاریف بالا پیروی نکنند ساده نبوده و پیچیده‌اند.

Visual cortex^۱

EVOLUTION OF IDEAS ON THE PRIMARY VISUAL CORTEX, 1955-1978. BY DAVID H. HUBLE^۲

٣.١ STC analysis

می‌دانیم که اگر یک ویژگی در تحریک باعث تغییر احتمال اسپایک زدن شود، می‌دانیم که اگه تحریک‌ها را به فضای آن spike-triggered منتقل کنیم باید تفاوت قابل توجهی بین مقادیر $p(s|r)$ و $p(s)$ باشد.^۳

روش‌های مبتنی بر correlation با هدف بررسی تغییر این توزیع احتمالات با توجه به تغییرات گشتاور مرتبه‌ی دوم آنها (واریانس) طراحی می‌شوند.

در این بین روش PCA به این خاطر که پایه‌ای در اختیار ما قرار می‌دهد که راستاهای بیشترین تا کمترین واریانس می‌باشند، بسیار مناسب کار ما خواهد بود و می‌تواند ویژگی‌هایی را که در آن واریانس تحریک‌ها بسیار بالا می‌باشد را در اختیار ما بگذارد.

روش spike-triggered analysis مبتنی بر موارد فوق مراحل زیر را انجام می‌دهد:

۱. ابتدا با فرض اینکه حافظه‌ی نورون‌ها بیشتر از ۱۶ فریم یا ۲۶۸ میلیثانیه نخواهد بود؛ پترن‌های تحریک را متشکل از ۱۶ نوار رنگی در ۱۶ فریم و در فضای ۲۵۶ بعدی تعریف می‌کنیم.

۲. ماتریس spike-triggered correlation را به صورت زیر تعریف می‌کنیم:

$$C = \frac{1}{N} \sum_{i=1}^N S(i)^T S(i)$$

که در آن $S(i)$ بردار ۲۵۶ بعدی در زامین باریست که نورون در طول آزمایش اسپایک زده است و N تعداد کل اسپایک‌ها در طول آزمایش است.

۳. بردارویژه‌ها و مقدارویژه‌های این ماتریس محاسبه شده و با توجه به اندازه‌ی مقدارویژه‌ها رتبه‌بندی می‌شوند.

با توجه به PCA می‌دانیم که حاصل راستاهایی خواهد بود که در آن واریانس تحریک‌هایی که موجب اسپایک می‌شوند از زیاد به کم مرتب شده‌اند.

٤.١ اعتبارسنجی نتایج

برای اعتبارسنجی مشاهدات روش spike-triggered correlation روشی که مقاله استفاده می‌کند این است که دنباله‌ای از اسپایک‌های تصادفی با تعداد اسپایک‌های مساوی با N تولید می‌کند و ماتریس correlation تحریک‌هایی که موجب این اسپایک‌های فرضی تصادفی شدند را ایجاد می‌کنیم. هدف این است که control correlation matrix را محاسبه کنیم که به این معناست

$$p(\text{response} | \text{stimulus})^3$$

که روشی که در قبل پیش گرفتیم را برای داده‌هایی که می‌دانیم مستقل از اسپایک زدن هستند(تمام داده‌ها) نیز اعمال کنیم. با توجه به حجم بالای داده‌ها N مساوی تعداد اسپایک‌ها را در نظر می‌گیریم و ۵ بار فرایند تولید control correlation matrix را انجام می‌دهیم و میانگین مقدارویژه‌ها را محاسبه می‌کنیم. با توجه به متن مقاله حاصل این میانگین‌گیری $\pm 5/2SD$ بازه‌ی اطمینان ما با $4 < p < 10$ خواهد بود. و بنابراین اگر مقدارویژه‌ای خارج از این بازه‌ی اطمینان باشد؛ آن مقدارویژه و بردارویژه دارای تفاوت آشکار خواهد بود.(در واقع آن بردار ویژه راستاییست که تحريك‌های موجب اسپایک در آن واریانس و در نتیجه توزیع احتمال کاملاً متفاوتی با توزیع اولیه دارند)

۵.۱ نتایج پژوهش

همانطور که در بخش‌های قبل اشاره کردیم هدف پیدا کردن eigenvalue‌هایی بود که خارج از بازه‌ی اطمینان باشند که آن‌ها به عنوان ویژگی‌های اصلی موجود در receptive-field گزارش شود. نتایج مقاله نشان می‌دهد که برای تعداد زیادی از سلول‌های پیچیده دو بردارویژه پیدا شده‌است که این بردار ویژه‌ها ناحیه‌ی ON-OFF مشخص و مجزا از همی دارند و همچنین correlation بسیار پایینی دارند. همچنین می‌بینیم که این راستاهای به خوبی سبب تحريك نورون‌ها می‌شوند با این حال برای eigenvalue‌هایی که در بازه‌ی اطمینان مذکور قرار دارند تاثیر بسیار کمتری در تحريك نورون‌ها مشاهده شده است.

۲ آشنایی با دیتاست

۱.۲ بخش اول

با استفاده از تابع تعبیه شده‌ی `fget_hdr.m` یکی از فایل‌های `sa0`. را باز می‌کنیم و به محتویات آن توجه می‌کنیم.

The screenshot shows the MATLAB workspace with two tables side-by-side:

hdr.DataInfo	
Field	Value
ID	'a01bmsq1D'
DataFrom	'000412.a01bmsq1D.ral'
Channel	0
SampleRate	10000
Gain	10000
DAQMode	0
DAQResolution	0
DataType	0
DataUnit	0
TimeMode	0
TimeOffset	0
ThreshPeakHigh	6542
ThreshPeakLow	53250
ThreshValleyHigh	-66541
ThreshValleyLow	-28176
ThreshWidthMax	-19
ThreshWidthMin	506

hdr.FileInfo	
Field	Value
Type	'DAN_SPK'
Version	'1.0'
Fname	'000412.a01bmsq1D.s...
Creator	'f21mlv.vi'
Time	'2000,4,12,20,10,16'

شکل ۲: محتویات فایل hdr

می‌توانیم بینیم که در این فایل اطلاعاتی مربوط به خود فایل از جمله زمان ایجاد دیتاست و سازنده‌ی آن و همچنین اطلاعاتی در مورد داده‌ها از جمله نرخ نمونه‌گیری وجود دارد. همچنین در فایل `10g`. نیز اطلاعاتی مربوط به نورونی که از آن نمونه‌گیری می‌شود وجود دارد.

نوع اطلاعات	برخی از اطلاعات موجود
اطلاعات مربوط به فایل	زمان ایجاد فایل، سازنده‌ی فایل، ورژن فایل
اطلاعات مربوط به تحریک	نوع تحریک، نرخ بروزرسانی مانیتور، نرخ تغییر فریم‌ها
اطلاعات مربوط به tuning curve	زاویه‌ی ترجیحی نورون، طول و عرض ترجیحی نورون
اطلاعات مربوط به نمونه‌گیری	نرخ نمونه‌گیری، نوع کارگذاری الکترودها، تعداد چنل‌ها

و اطلاعات دیگری از این دست می‌باشد.

۲.۰ دوم بخش

تابع `Func_ReadData` را به صورتی که خواسته شده پیاده‌سازی می‌کنیم. ایده‌ی کلی در پیاده‌سازی این تابع این بود که ابتدا با استفاده از تابع `dir` فایل‌های موجود در دایرکتوری نورون مورد نظر را می‌خوانیم و فایل‌های با `mds1d.sa0` را پیدا می‌کنیم و اطلاعات آن را با استفاده از `fget_spk.m` می‌خوانیم.

همچنین با توجه به اینکه در بخش بعد به Spike-count rate احتیاج پیدا می‌کنیم یکی از خروجی‌های Optional را برابر با آن تعریف می‌کنیم، برای محاسبه‌ی این مقدار نیز تنها کافیست توجه کنیم که:

$$\langle r \rangle = \left\langle \frac{\#spikes}{time} \right\rangle_{trials}$$

نهایتاً تابع مورد نظر به صورت زیر خواهد بود:

قطعه کد: تابع Func_ReadData ۱.۲.۲

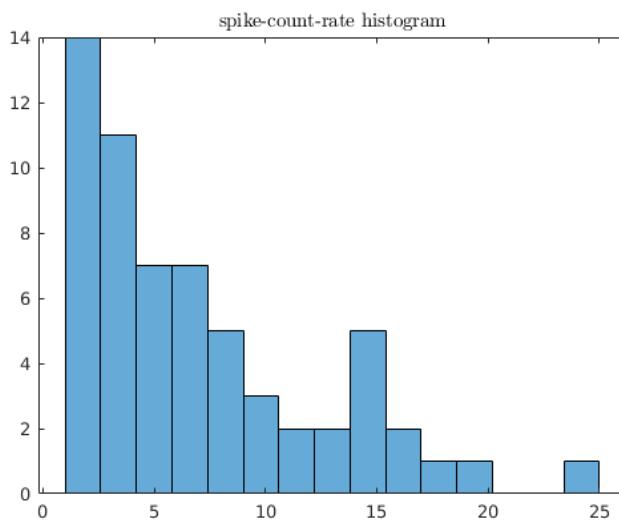
```

1 % Recomend: select Data/ and MatlabFucntions/
2 % directories and add them to matlab PATH
3 function [Output, spike_count_rate] = Func_ReadData(NeuronCode)
4     Path = [pwd, '/Data', '/Spike_and_Log_Files/', NeuronCode];
5     Listing = dir(Path);
6     Index_Counter = 0;
7     events = {};
8     hdrs = {};
9     frame_rate = 59.72; %Hz
10    frame_number = 32767;
11    spike_count = [];
12    for i = 1:length(Listing)
13        if (contains(Listing(i).name, 'msq1d.sa0', 'IgnoreCase', true)
14            ) && ...
15            ~contains(Listing(i).name, 'sa0.sub', 'IgnoreCase', true)
16            && ...
17            ~contains(Listing(i).name, 'sa0.', 'IgnoreCase', true))
18        Index_Counter = Index_Counter + 1;
19        [events{Index_Counter}, hdrs{Index_Counter}] = fget_spk(
20            Listing(i).name, 'get');
21        spike_count(Index_Counter) = length(events{Index_Counter});
22    end
23    Output = struct('events', events, 'hdr', hdrs);
24    spike_count_rate = mean(spike_count) * frame_rate/frame_number;
25 end

```

۳۰۲ بخش سوم

در این بخش ابتدا با استفاده از تابع `dir` در پوشه‌ی مربوط به اطلاعات اسپایک‌ها نام تمام نورون‌ها را استخراج می‌کنیم. سپس با پیمایش بر روی آن‌ها و با استفاده از تابعی که در بخش قبل نوشتم نرخ اسپایک هر نورون را نگهداری می‌کنیم. نهايتا هیستوگرام نرخ اسپایک‌ها به صورت زير خواهد بود:



شکل ۳: هیستوگرام نرخ اسپایک‌های نورون‌ها

سپس به سادگی می‌توانیم نورون‌هایی که نرخ اسپایک کمتر از ۲ دارند را با استفاده از قطعه کد زیر از لیست اولیه‌ای که برای کد نورون‌ها تهیه کردیم حذف کنیم.

۱۰۳۰۲ قطعه کد: حذف نورون‌ها با نرخ اسپایک پایین

```

1 index_counter = 0;
2 for i = 1:61
3   if SCRA(i) < 2
4     index_counter = index_counter + 1;
5     index(index_counter) = i;
6   end
7 end
8 fprintf('Excluded Neurons:\n')
9 disp(neuron_codes(index))
10 neuron_codes(index) = [];
```

خروجي اين قطعه کد شماره‌ی نورون‌های حذف شده را نمایش می‌دهد که به صورت زير است:

Excluded Neurons:

"000413.b03"
 "000413.b04"
 "000413.b05"
 "000418.a01"
 "000420.b02"
 "000524.c01"
 "000907.f07"

٤٠٢ بخش چهارم

در این قسمت تابع Func_StimuliExtraction به گونه‌ای پیاده می‌کنیم که با دریافت دنباله‌ای از زمان اسپایک‌ها تحریک‌هایی با ابعاد 16^*16 که سبب این اسپایک‌ها شدند را خروجی بدهد. برای اینکار باید توجه کنیم که با توجه به متن مقاله نرخ تغییر فریم‌های Hz ۶۰ و به طور دقیق با توجه به فایل‌های لاغ Hz $59/7213$ می‌باشد. همچنین از آنجایی که نرخ نمونه برداری $1ms$ باشد می‌توان دید که با استفاده از کد زیر می‌تواند اندیس‌هایی از فریم‌ها که در آن‌ها اسپایک رخ داده است را ببینیم. (از آنجایی که تمایل داریم ابعاد ماتریس 16^*16 باشد اندیس‌های کمتر از ۱۶ را نادیده در نظر می‌گیریم)

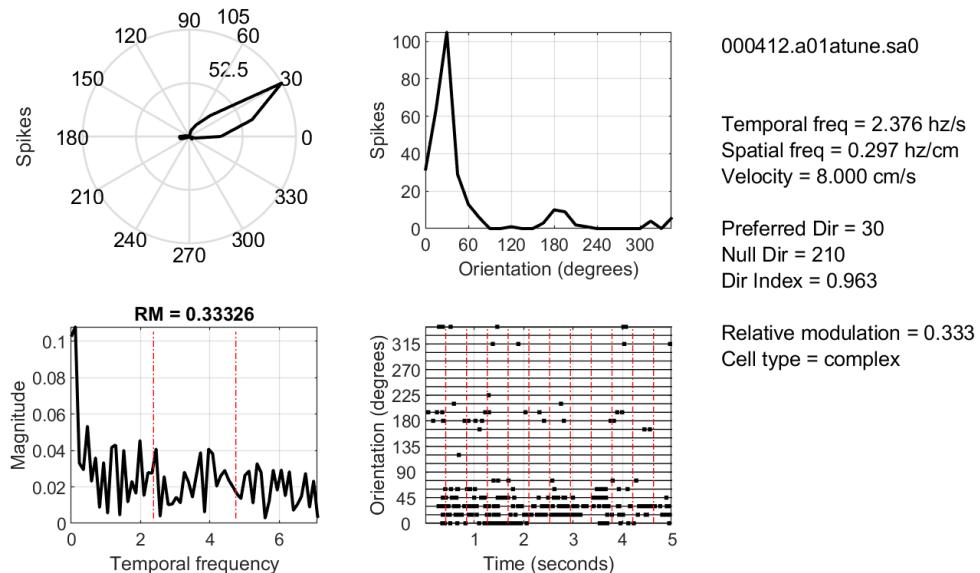
١٠٤٠٢ قطعه کد: یافتن اندیس‌های زمان اسپایک و نادیده گرفتن اندیس‌های کمتر از ۱۵

```
1 indices = ceil((events * frame_rate) / 10000);
2 indices = indices(indices > 15);
```

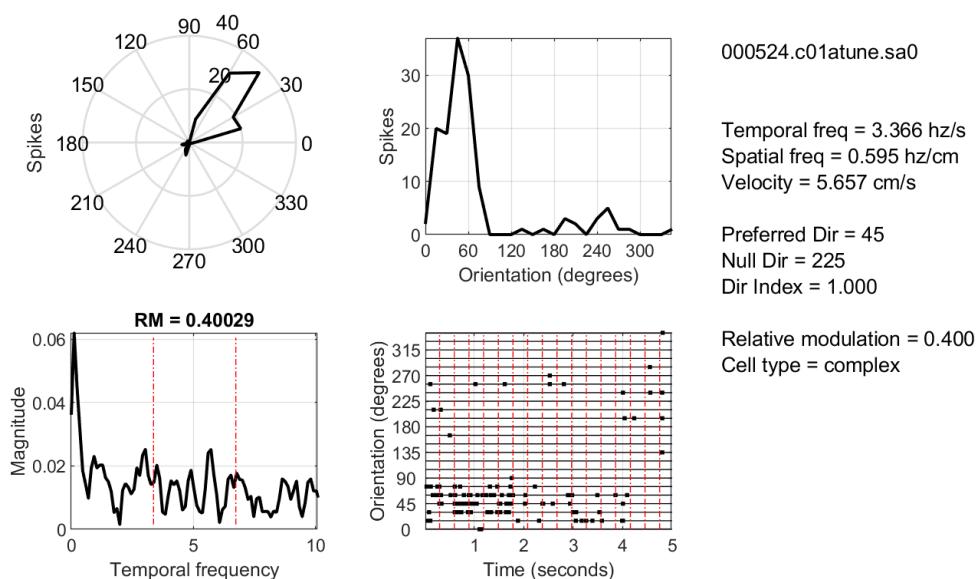
سپس می‌توانیم ماتریس‌های 16^*16 که سبب اسپایک شدند را تشکیل دهیم. برای راحت‌تر شدن کار در مراحل بعد اولاً ورودی دوم ذکر شده در صورت سوال یعنی msq1D را اختیاری در نظر گرفتیم که مقدار دیفالت آن همان فایلی خواهد بود که در پوششی Data قرار دارد. همچنین در مراحل بعد از آنجایی که می‌خواهیم دنباله‌ای تصادفی از اسپایک‌ها تولید کنیم راحت‌تر خواهد بود که فقط اندیس‌ها را به عنوان ورودی به این تابع بدهیم به این منظور یک ورودی option تعريف کردیم که اگر مقدار آن 'random' باشد مقدار events در ورودی را به عنوان اندیس‌ها در نظر می‌گیرد و در غیر این صورت همان منطق قبلی را اجرا می‌کند.

٥.٢ بخش پنجم

برای تشخیص مناسب ترین جهت نشان دادن تصاویر، باید آزمایشی ترتیب داده شود که یک ورودی مشخص، در زوایای مختلف به فرد مورد آزمایش نشان داده شود، و نرخ اسپاکیک زدن متناظر با هر راستا اندازه گیری شود. با شناسایی بیشینه‌ی این نرخ، جهت مناسب پیدا می‌شود.



شکل ۴: اطلاعات فایل تیونینگ نورون 000412.a01



شکل ۵: اطلاعات فایل تیونینگ نورون 000524.c01

۳ برسی با روش کلاسیک STA

در سکشن دوم کد نهایی که برای این بخش ارائه شده است بر روی تمام ۵۴ نورونی که در بخش قبل حذف نشده‌اند عملیات‌ها صورت می‌گیرد. و حاصل تنها ذخیره می‌شود و تصاویر نمودارها می‌شوند. بنابراین برای تست کردن کد این بخش توصیه می‌شود که ابتدا سکشن اول که مربوط به بارگذاری استراکچرهای مورد نیاز است را اجرا کرده و سپس از سکشن سوم به بعد که مخصوص تست طراحی شده‌اند را اجرا کنید.

۱.۳ بخش اول

برای محاسبه‌ی این بخش تابع Func_FindSTA پیاده سازی شده است که نحوه‌ی عملکرد آن به صورت زیر است.

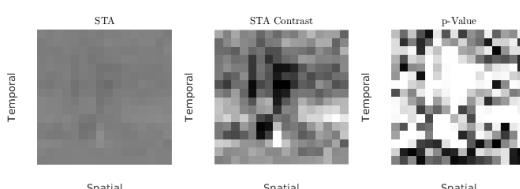
۱.۱.۳ قطعه کد: تابع محاسبه‌ی STA

```

1 function [sta, spike_trigerred] = Func_FindSTA(neuron)
2     outs = neuron.outs;
3     spike_trigerred = [];
4     for i = 1:length(outs)
5         spike_trigerred_trial = Func_StimuliExtraction(outs(i).events);
6         spike_trigerred = cat(3, spike_trigerred, spike_trigerred_trial);
7     end
8     N = length(spike_trigerred);
9     sta = (sum(spike_trigerred, 3)/ N);
10    end

```

که می‌توان دید بنابر تعریف STA پیاده‌سازی شده است. این تابع تمام مقادیر spike_trigerred را نیز خروجی می‌دهد چرا که در مراحل بعد به آن نیاز خواهیم داشت. برای همان مثالی که در صورت پروژه‌آمده است خروجی زیر را می‌گیریم:



شکل ۶: خروجی برای نورون 000601.c05

٢٠٣ بخش دوم

خروجی این بخش در بخش اول نمایش داده شد با این حال ذکر می‌کنیم که برای محاسبه‌ی p-value به دست آمده از t-test از تابع آماده ttest در متلب استفاده کردیم.

١٠٢٠٣ قطعه کد: محاسبه‌ی t-test

```
1 [~, p] = ttest(permute(spike_triggered,[3 1 2]));
```

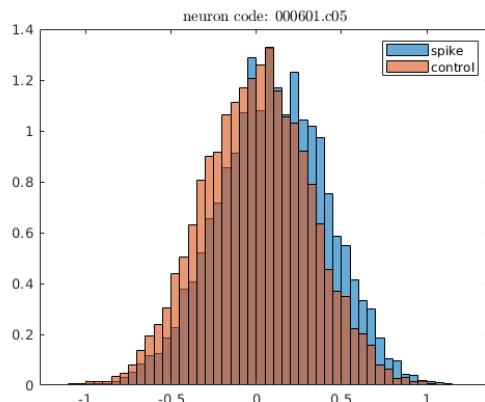
٣٠٣ بخش سوم

برای محاسبه‌ی correlation از آنجایی که یکی از ماتریس‌هایی که در اختیار ماست ۳ بعدی است محاسبه‌ی correlation را به صورت دستی انجام دادیم و تابع Func_Correlation را پیاده‌سازی کردیم، که اساس کار آن ضرب درایه به درایه موجود در متلب می‌باشد. همچنین برای تولید دنباله‌ی تصادفی از اسپایک‌ها یا همان ماتریس کنترل به صورت زیر عمل کردیم و در اینجا اضافه کردن پارامتر ورودی اضافه به تابع کارمان را ساده‌تر می‌کند. و تنها کافیست دنباله‌ای تصادفی از شماره‌ی فریم‌ها ایجاد کنیم.

١٠٣٠٣ قطعه کد: ایجاد دنباله‌ی تصادفی از تحریک‌ها

```
1 spike_sample = randi([16 32767], 1, N);
2 stim_control = Func_StimuliExtraction(spike_sample, 'random');
```

خروجی هریک از ماتریس‌های کنترل و spike-triggered در فضای جدید برای این نورون به صورت زیر است.



شکل ٧: هیستوگرام correlation ماتریس‌های کنترل و STA با spike-triggered

٤٠٣ بخش چهارم

برای محاسبه p-value در اینجا نیز از تابع آماده `ttest2` موجود در متلب استفاده می‌کنیم.
خروجی آن به صورت زیر است.

`p-value = 0.000000, null-hypothesis rejected, means are diffrent.`

نتیجه‌ی آزمون t-test این است که توزیع $p(s|r)$ و $p(r|s)$ در فضای مربوط به STA دارای میانگین‌های متفاوتی می‌باشند. با این حال با توجه به شکل ۸ برایمان روشن است که این دو توزیع علارغم اینکه میانگین‌های متفاوتی دارند اما به دلیل اینکه سطح مشترک زیر نمودار آنها بسیار زیاد است نمی‌توان تفاوت معناداری بین آنها قائل شد.

به عبارت دیگر نتیجه‌ی آزمون می‌گوید که توزیع $p(s|r)$ و $p(r|s)$ تفاوت میانگین قابل توجهی دارند که این بر خلاف انتظار ما نمی‌باشد چرا که $p(s|r)$ از تصویر بردارهای منجر به اسپایک بر روی STA حاصل می‌شود که واضح است که باید میانگین مثبتی داشته باشد و از طرفی روشن است که با انتخاب تحریک‌های رندم و تصویر کردن روی STA میانگینی نزدیک به صفر خواهیم داشت. با این حال همانطور که از شکل مشخص است تفاوت معنادار در میانگین نمی‌تواند تفاوت معنادار در توزیع‌ها را نشان بدهد و همانطور که از متن مقاله به یاد داریم در آنالیزهایی که بر پایه‌ی correlation طراحی می‌شوند تغییر در توزیع‌ها را با تغییر در گشتاور درجه‌ی دوم آنها یا همان واریانس می‌سنجدیم.

٥٠٣ بخش پنجم

به طور کلی میدانیم که خطای تخمین ما از رابطه‌ی زیر محاسبه می‌شود:

$$P(\text{Error}) = P(\text{Error}|s = 1)P(r = 1) + P(\text{Error}|r = 0)P(r = 0)$$

از طرفی می‌دانیم که با توجه به اینکه تحریک‌های اولیه کاملاً تصادفی بودند ($P(s = 1) = \text{نرخ اسپایک}$ زدن هر نورون در همین آزمایش می‌باشد. ما قبل نرخ spike-count rate را محاسبه کرده بودیم و از توضیحات درس به یاد داریم که:

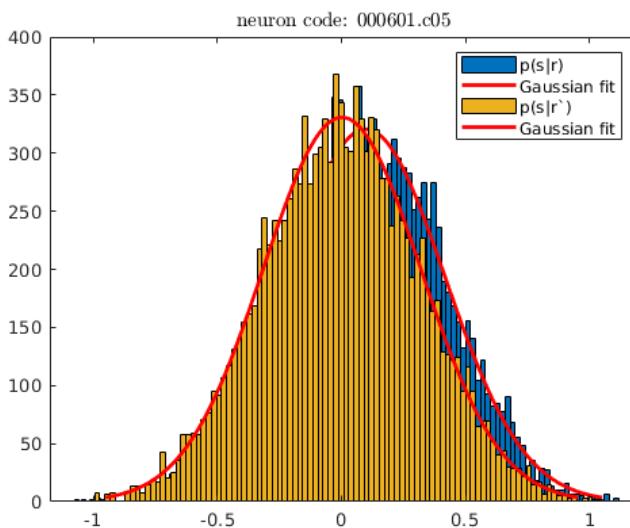
$$P(r = 1) = \frac{\langle \text{count} \rangle}{\Delta t}$$

که در اینجا Δt همان نرخ تغییر فریم‌ها می‌باشد. حال با توجه به نوع تخمینمان که تنها به یک پارامتر θ وابسته می‌باشد مقدار خطای را باز نویسی می‌کنیم:

$$P(\text{Error}) = P(X > \theta | r = 0)P(r = 0) + P(X < \theta | r = 1)P(r = 1)$$

توزیع $P(X|r=1)$ همان توزیع تصویر spike-triggered STA می‌باشد که طبق فرض مسئله آن را نرمال در نظر می‌گیریم. از طرف دیگر با توجه به sparse بودن نورون‌ها و احتمال پایین اسپایک زدن آنها (برای اکثر نورون‌ها با توجه به شکل ۳ در حدود 10^{-2} می‌باشند) می‌توانیم توزیعی که برای تصویر control محاسبه کردیم را توزیع $P(X|r=0)$ در نظر بگیریم و بنابراین می‌توانیم $P(\text{Error})$ را محاسبه کنیم و مشتق آن نسبت θ را برابر با صفر قرار دهیم تا بهترین ترشهولد را محاسبه کنیم.

برای محاسبه این مقادیر در مطلب ابتدا یک فیت گاوی بر روی هیستوگرام‌ها ایجاد می‌کنیم و سپس μ و σ را محاسبه می‌کنیم و آن‌ها را به همراه احتمال اسپایک زدن نورون به تابع نوشته شده‌ی Func_Accuracy می‌دهیم که ترشهولد و دقیق تخمین ما را بیان می‌کند.

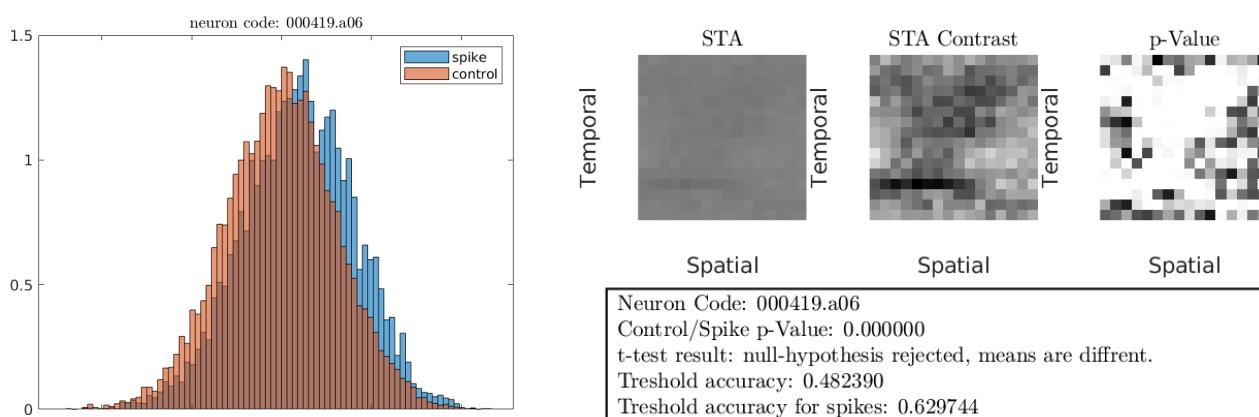
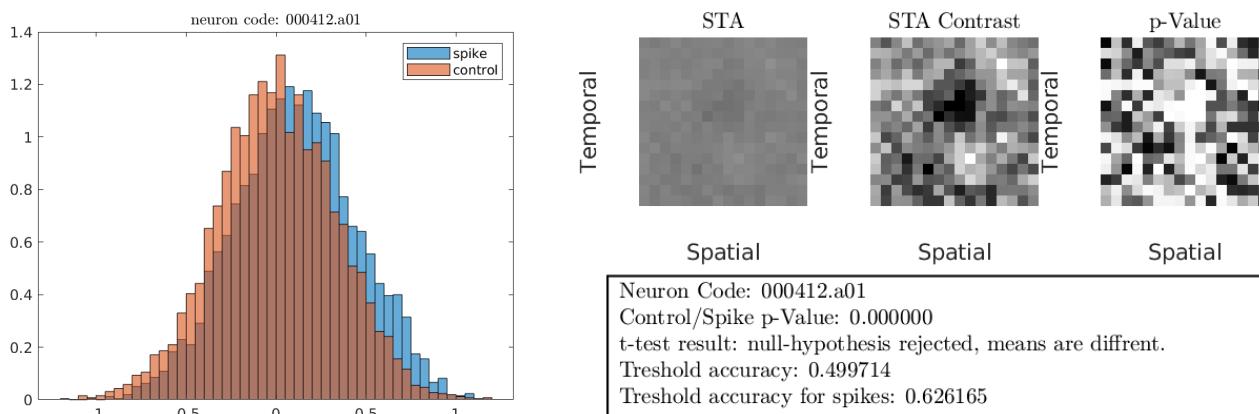


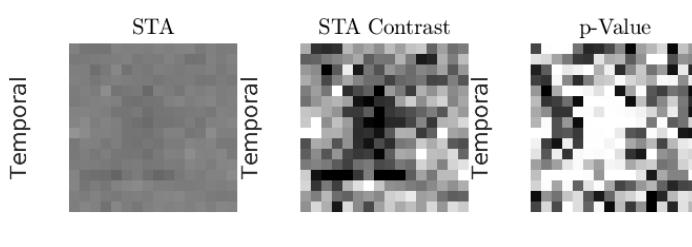
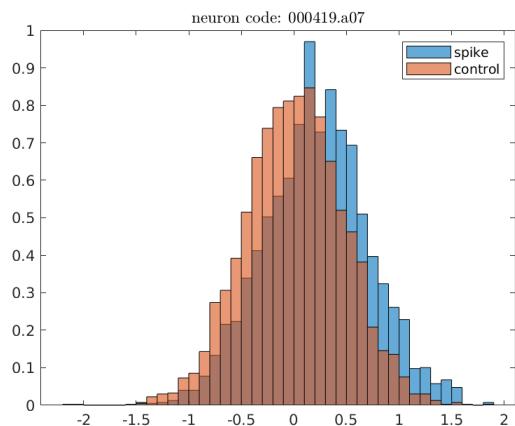
شکل ۸: هیستوگرام‌ها به همراه فیت گاوی آن‌ها

همچنین توجه می‌کنیم که برای محاسبه‌ی Func_Accuracy از سایت‌های محاسبه‌کننده استفاده کردیم و خروجی آنها را در تابع مطلب قرار دادیم.

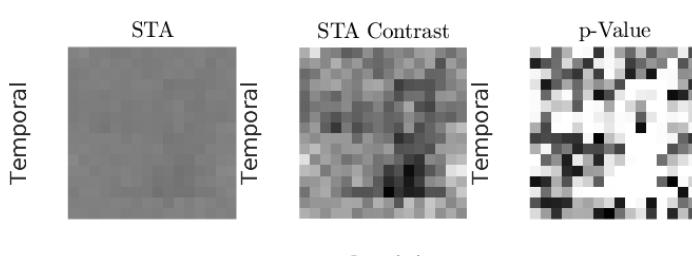
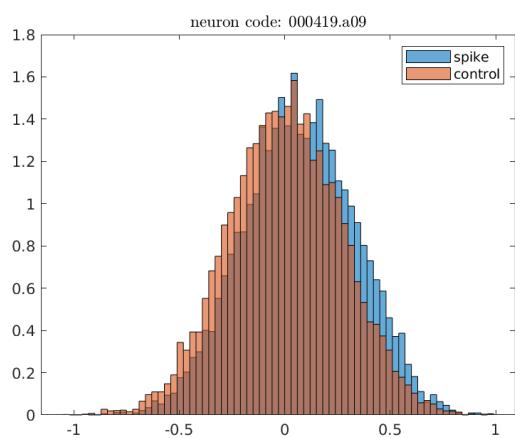
٦٠٣ بخش ششم

با اجرای مراحل بالا برای تمامی ۵۴ نورون باقی مانده به خروجی‌های زیر دست پیدا می‌کنیم لازم است توجه کنیم که داده‌های کنترل به صورت رندم تولید می‌شوند و بنابراین با اجرای این کدها ممکن است نتایج اندکی متفاوت بگیریم هر چند در کلیت پاسخ تاثیری نخواهند داشت.

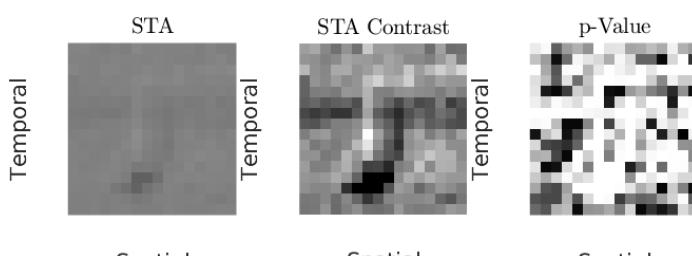
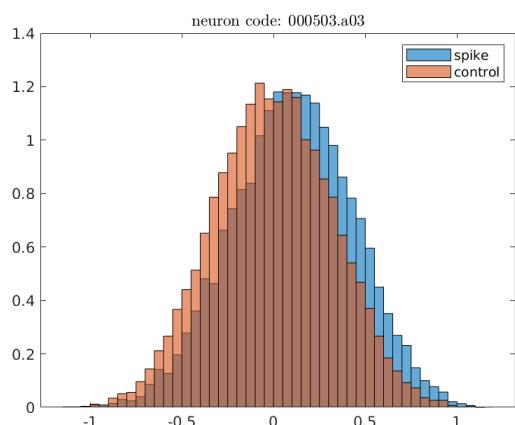




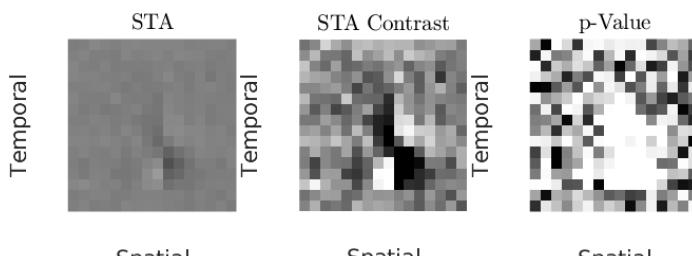
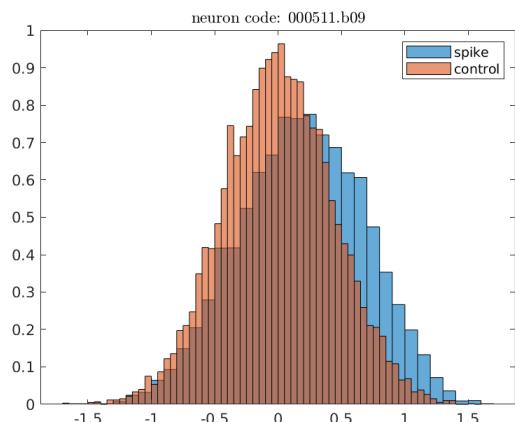
Neuron Code: 000419.a07
Control/Spike p-Value: 0.000000
t-test result: null-hypothesis rejected, means are different.
Treshold accuracy: 0.499489
Treshold accuracy for spikes: 0.681407



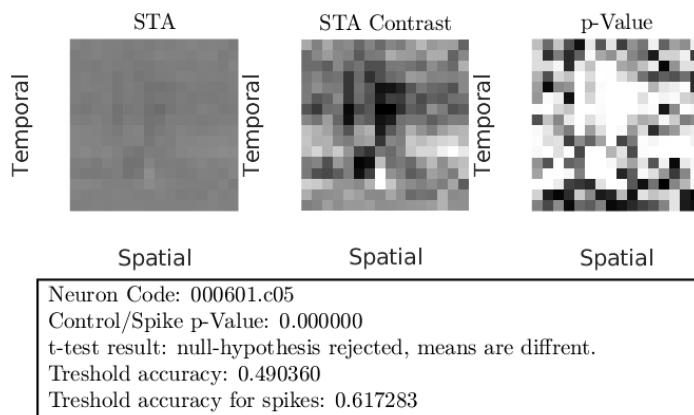
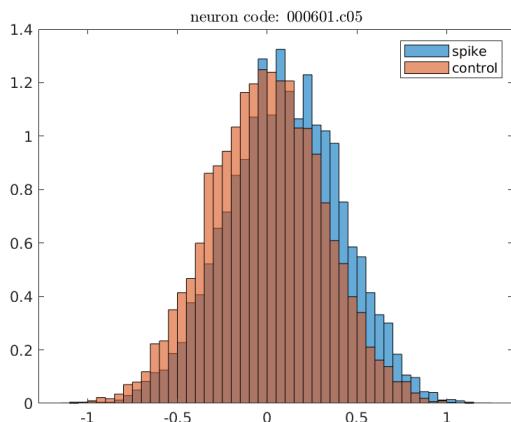
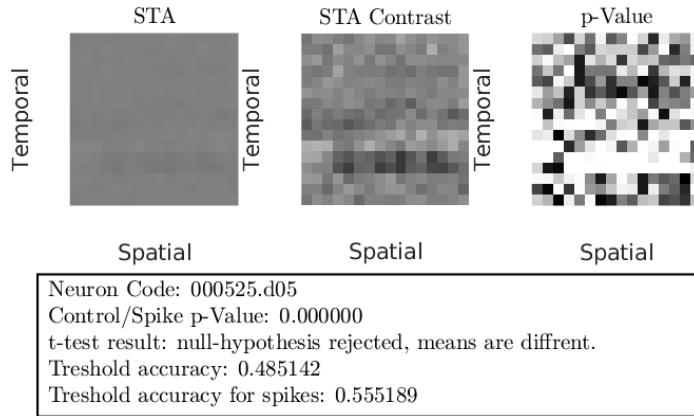
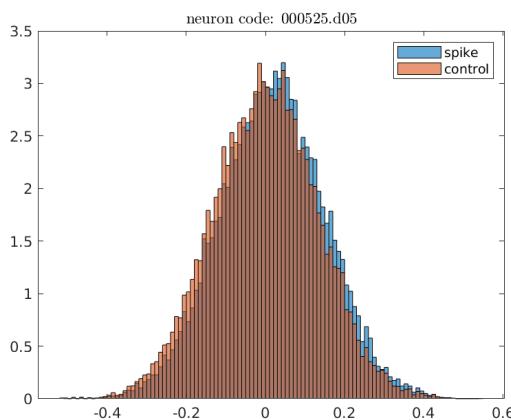
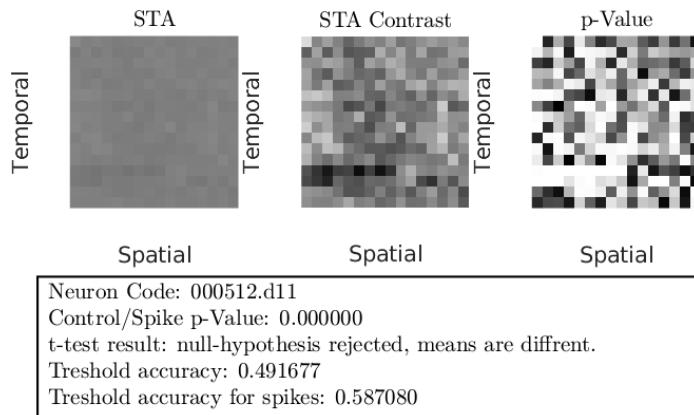
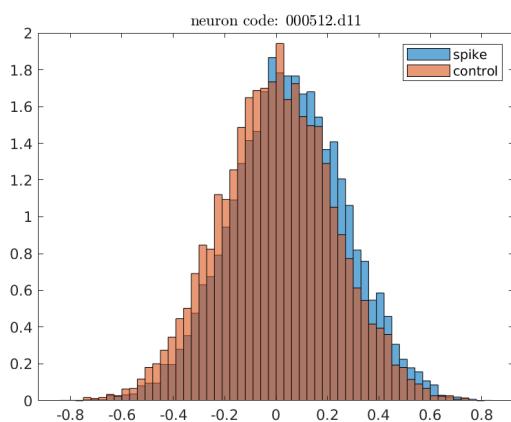
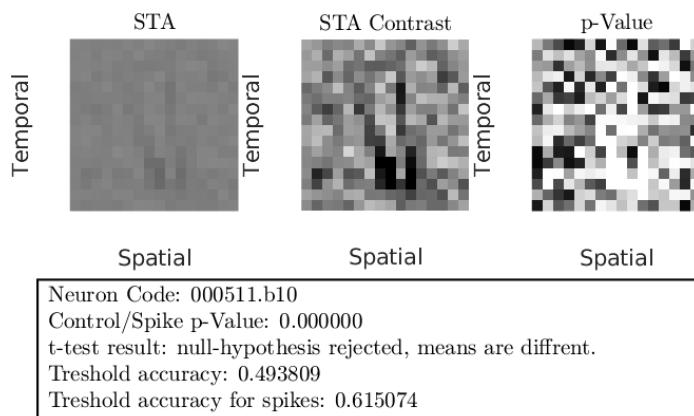
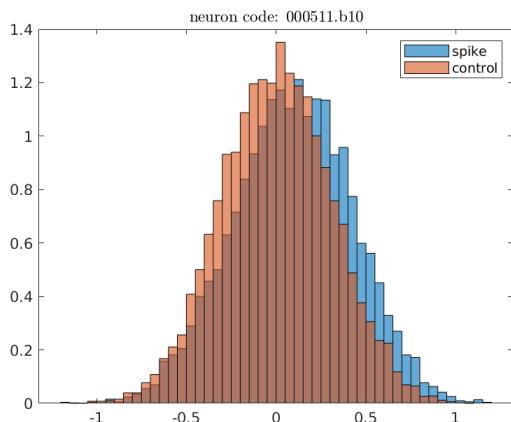
Neuron Code: 000419.a09
Control/Spike p-Value: 0.000000
t-test result: null-hypothesis rejected, means are different.
Treshold accuracy: 0.493368
Treshold accuracy for spikes: 0.598505

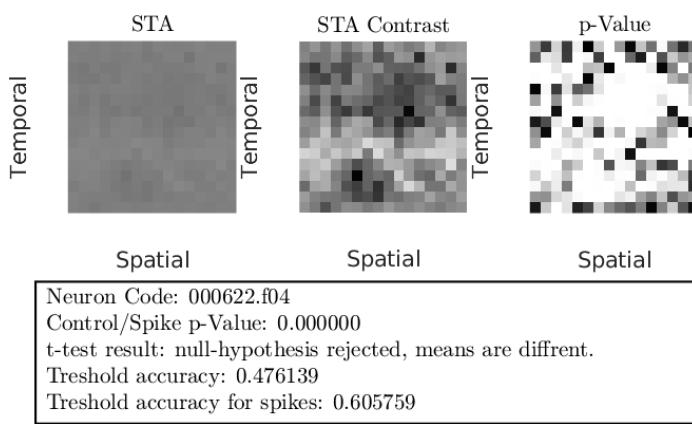
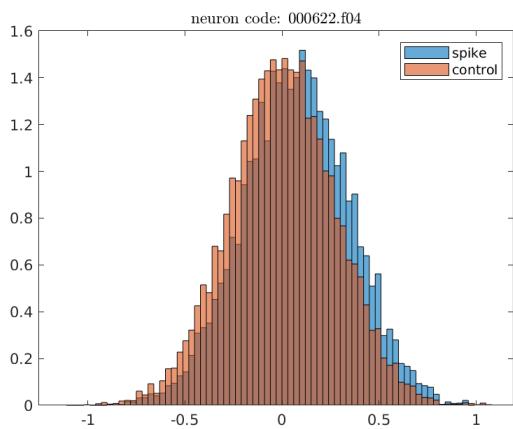
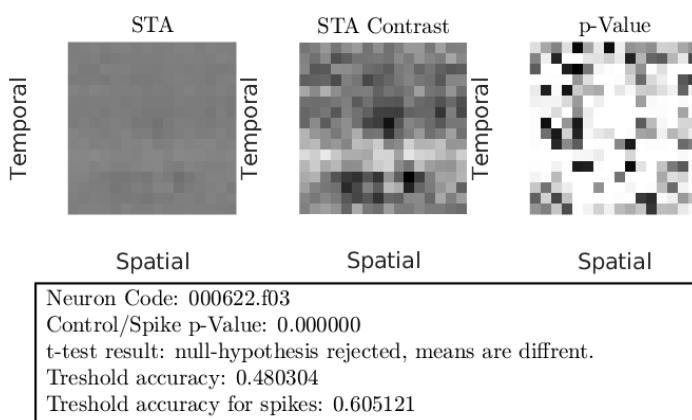
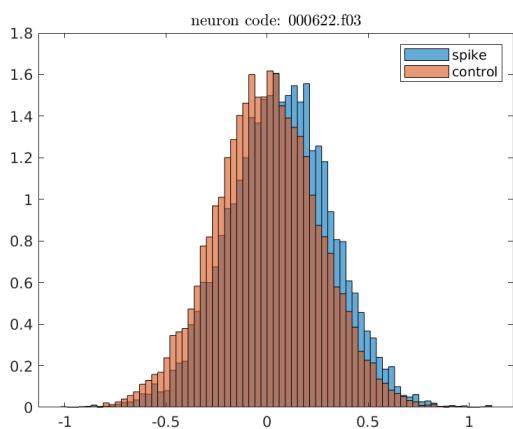
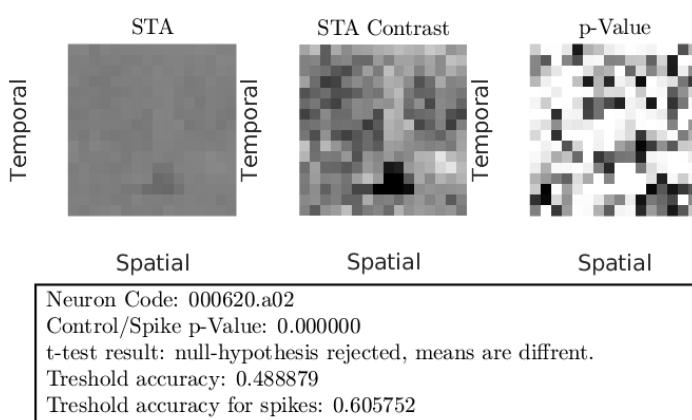
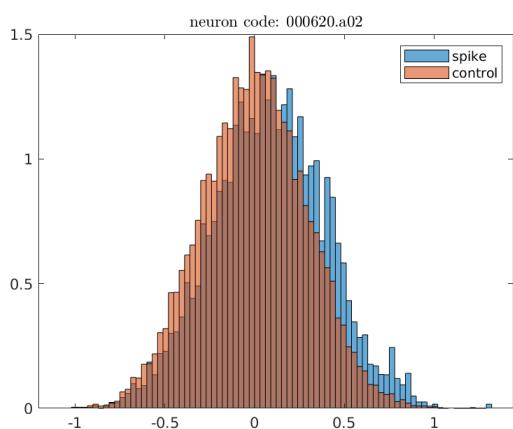
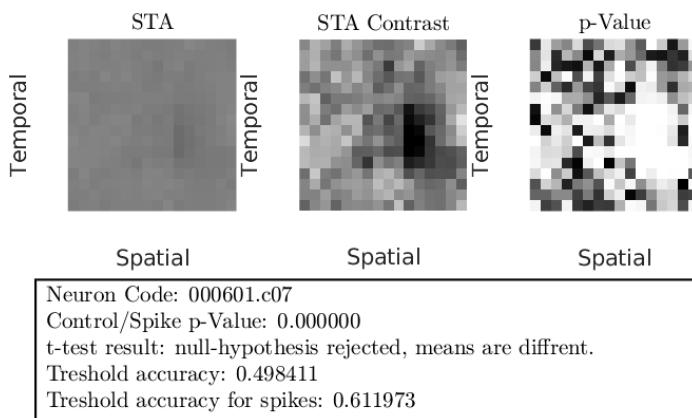
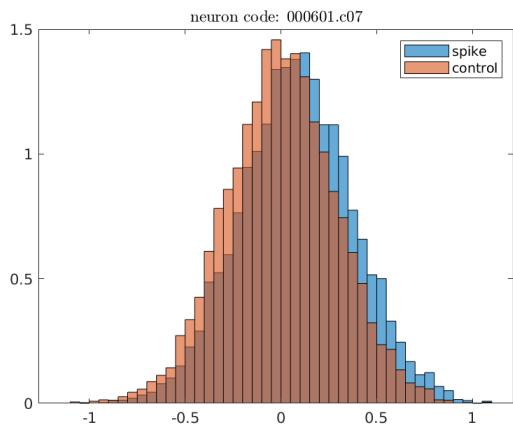


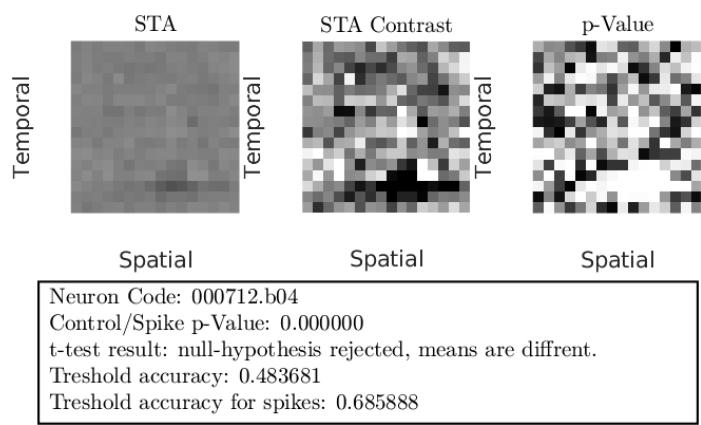
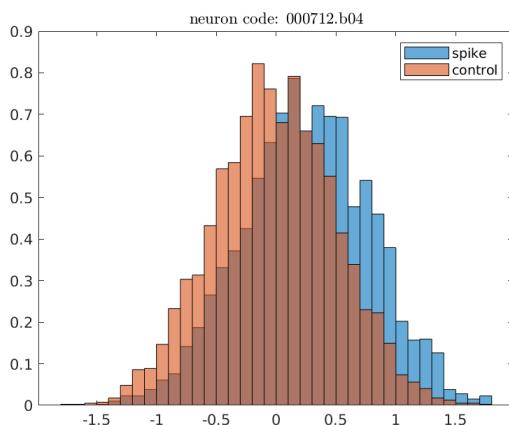
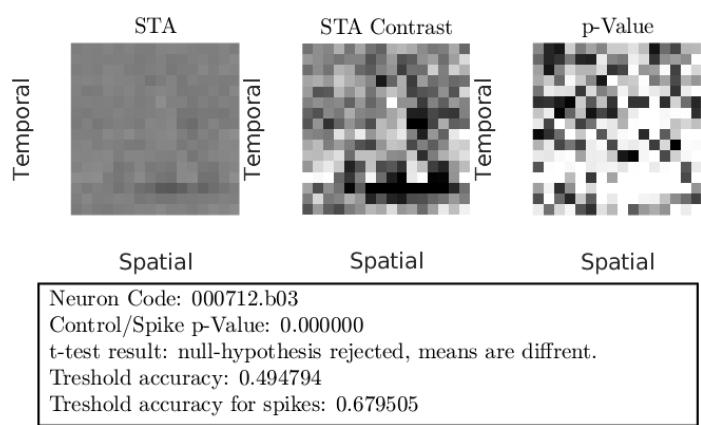
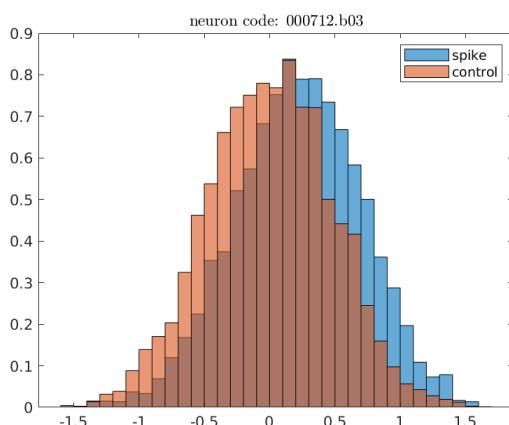
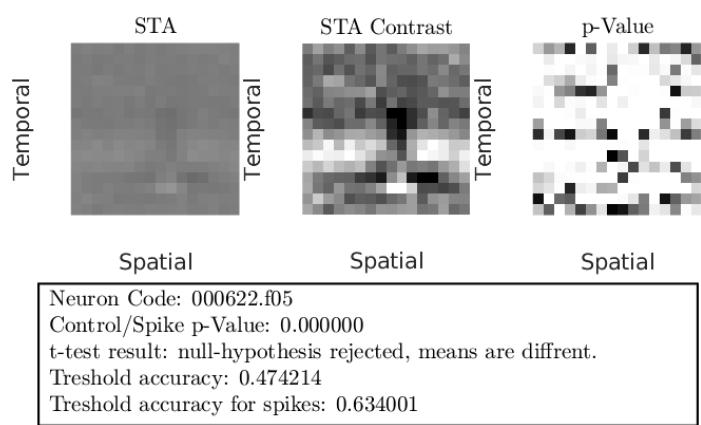
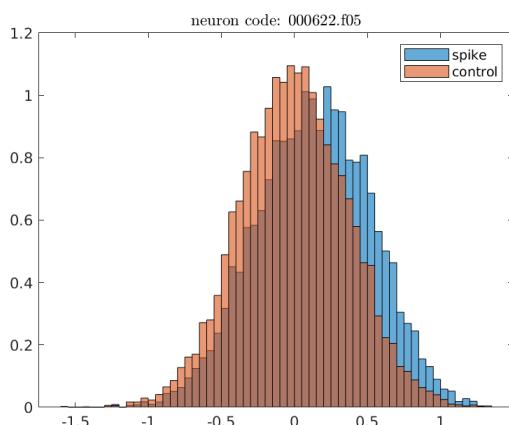
Neuron Code: 000503.a03
Control/Spike p-Value: 0.000000
t-test result: null-hypothesis rejected, means are different.
Treshold accuracy: 0.479796
Treshold accuracy for spikes: 0.627188

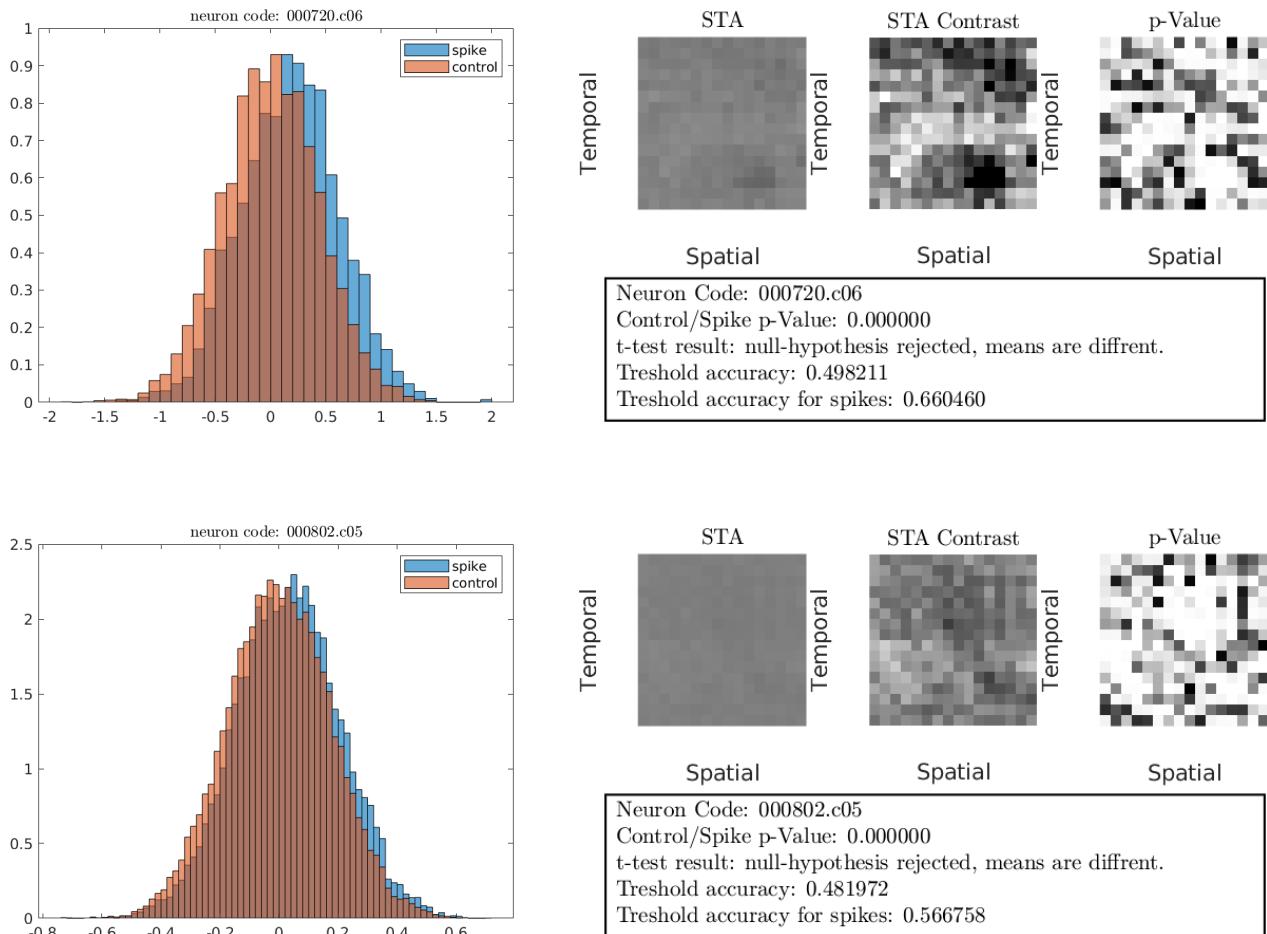


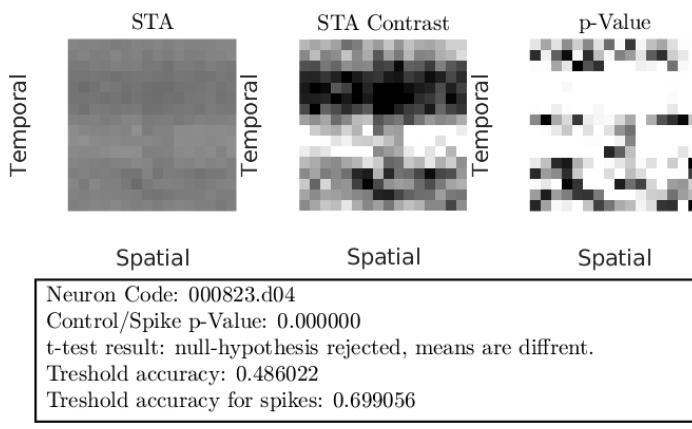
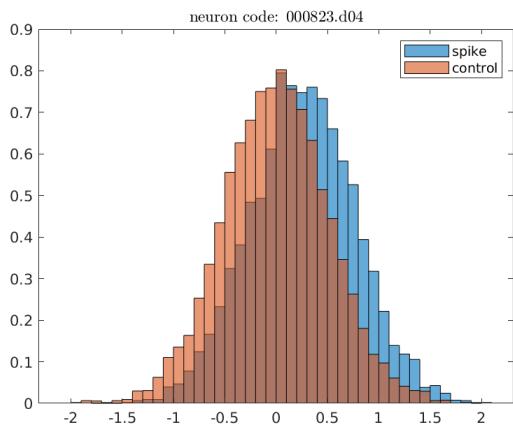
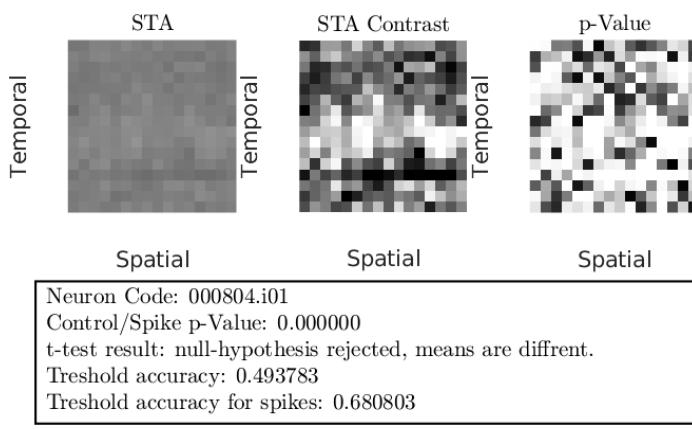
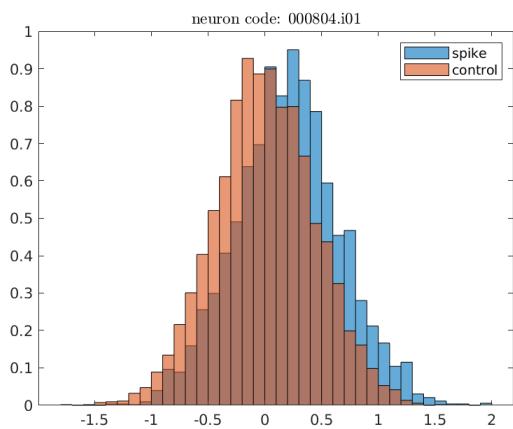
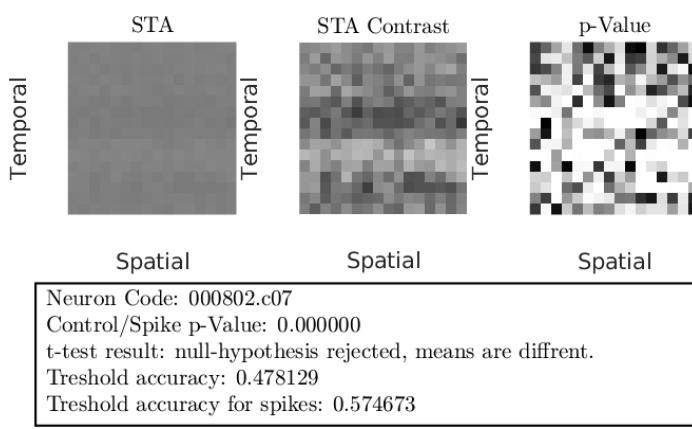
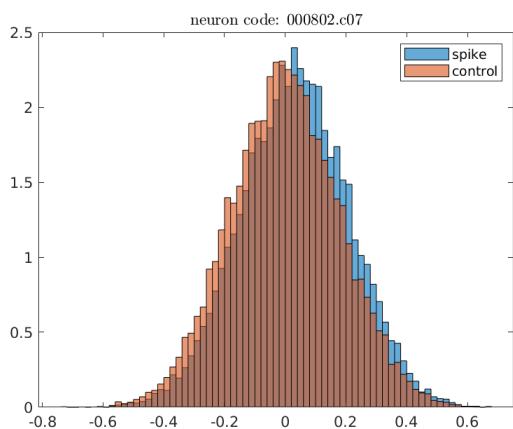
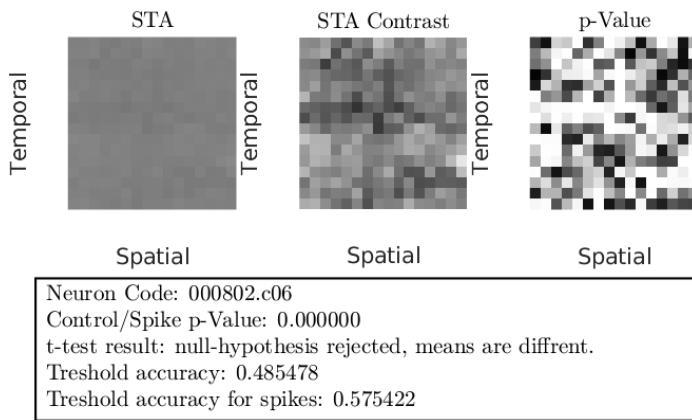
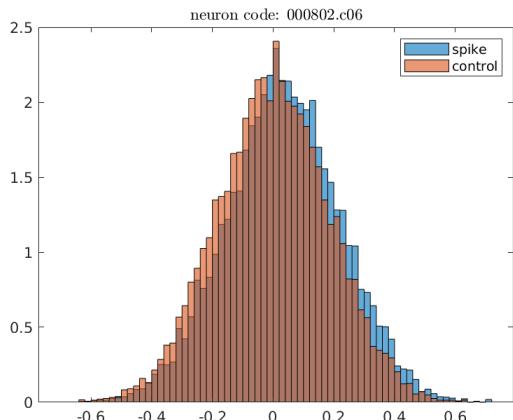
Neuron Code: 000511.b09
Control/Spike p-Value: 0.000000
t-test result: null-hypothesis rejected, means are different.
Treshold accuracy: 0.480744
Treshold accuracy for spikes: 0.646897

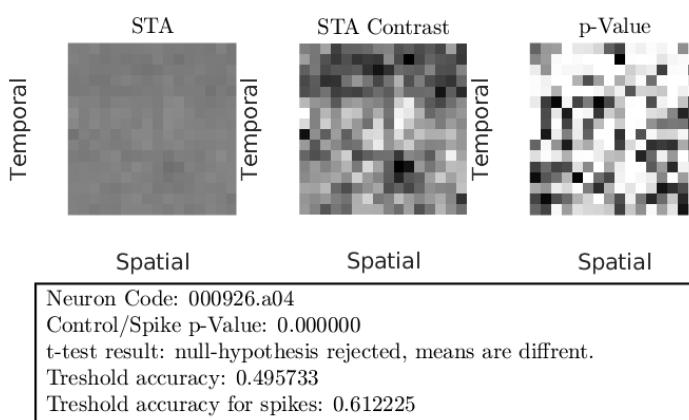
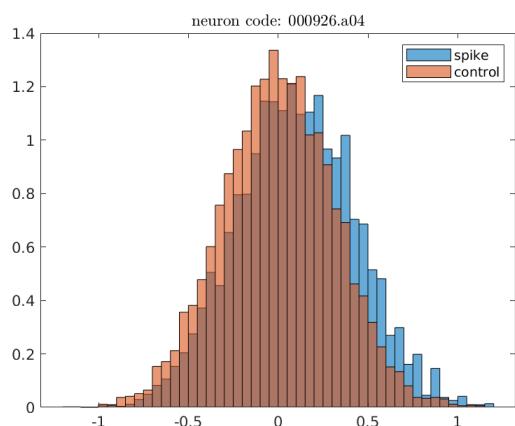
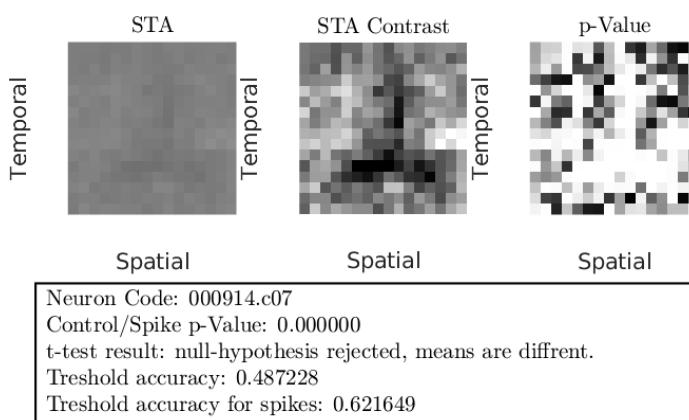
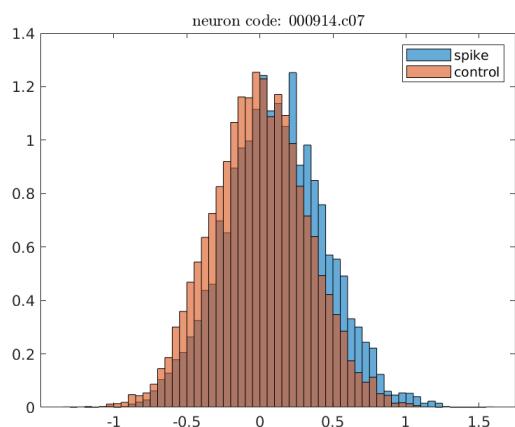
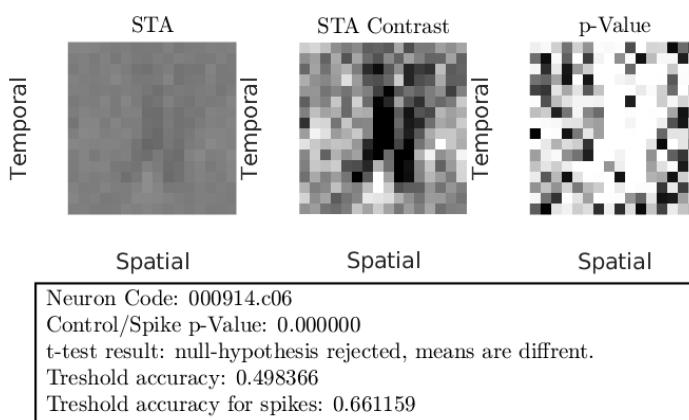
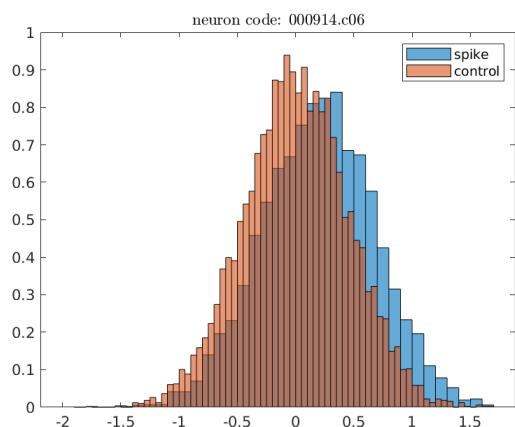
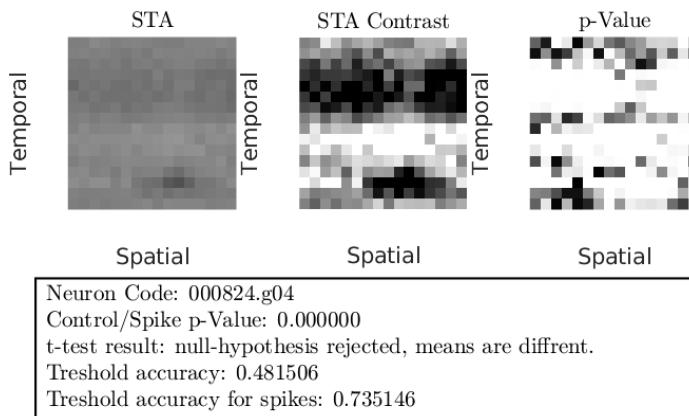
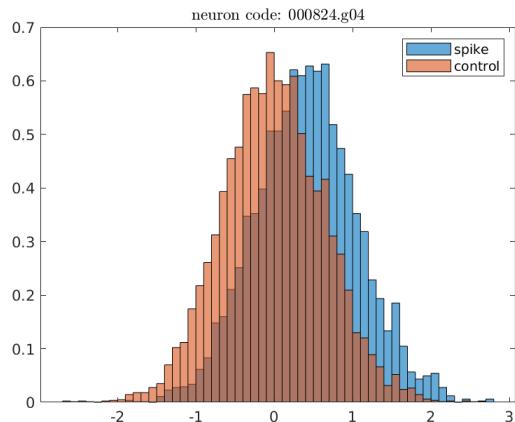


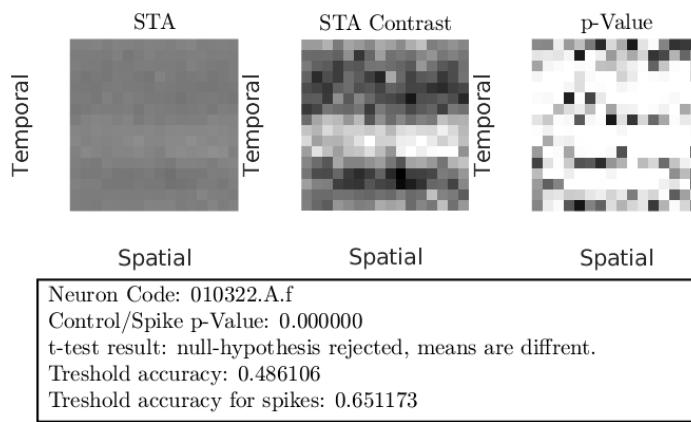
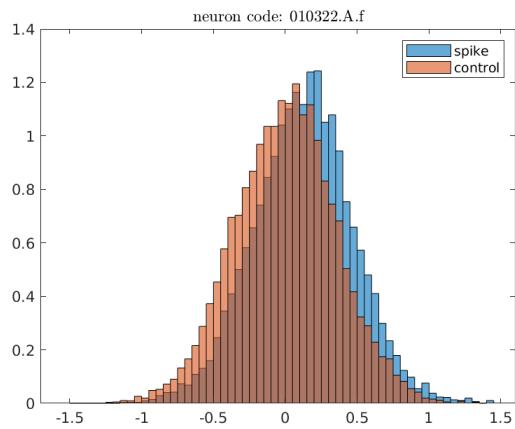
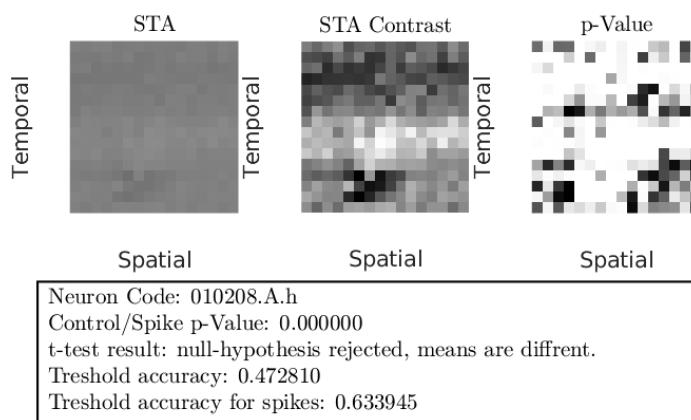
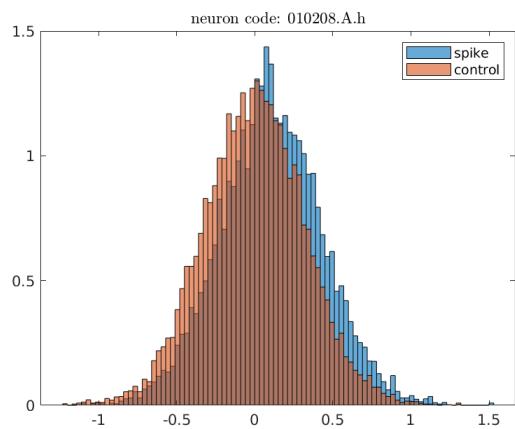
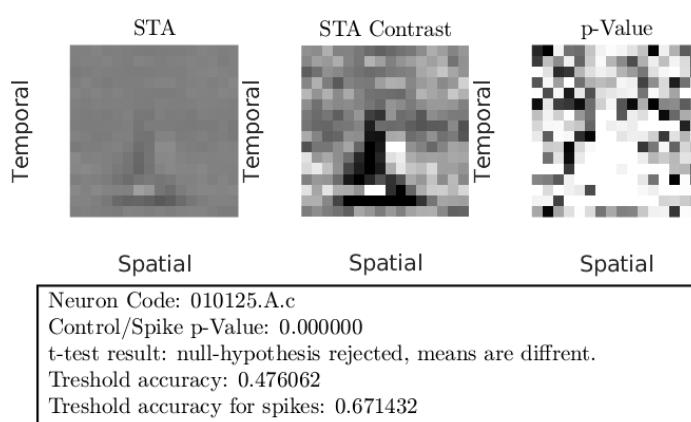
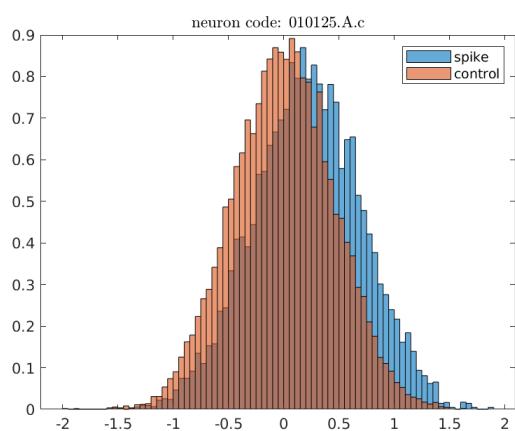
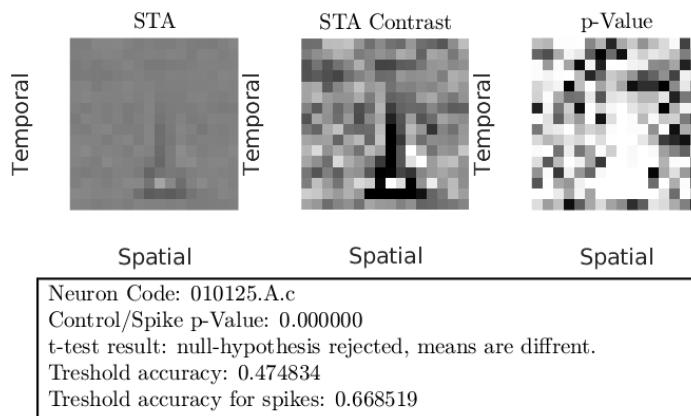
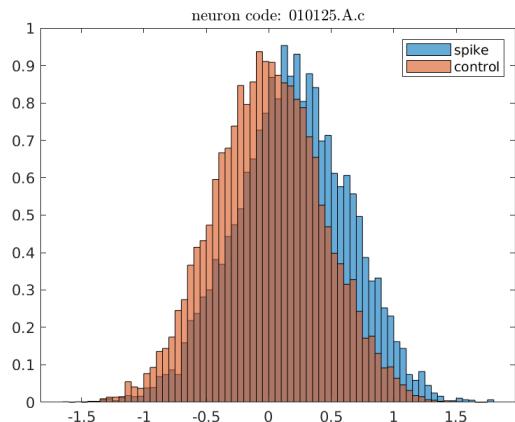


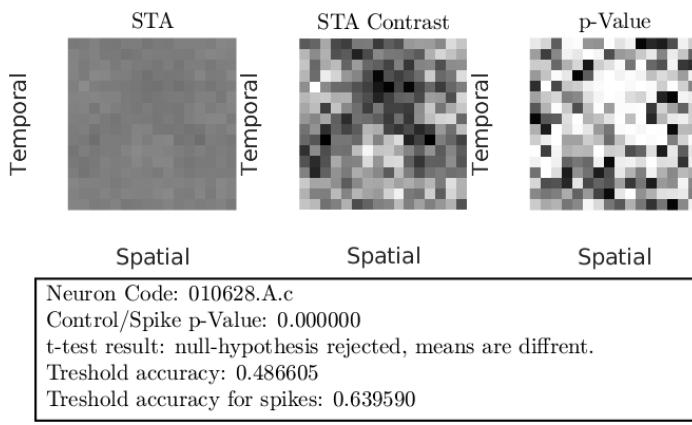
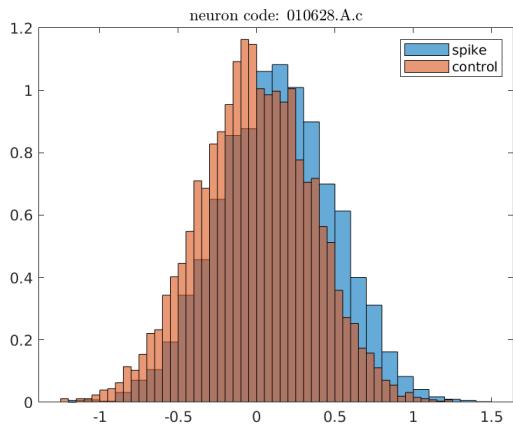
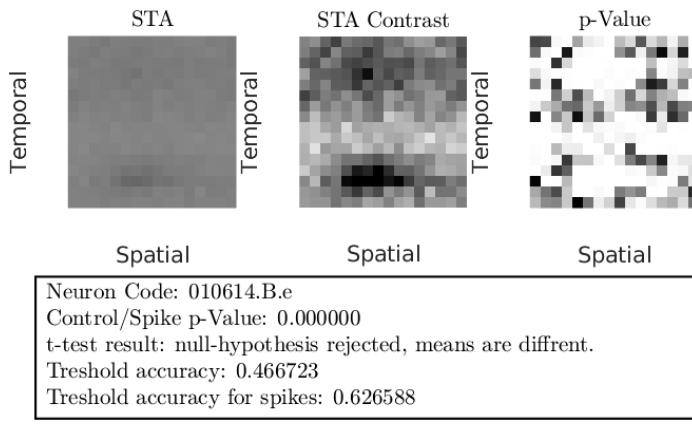
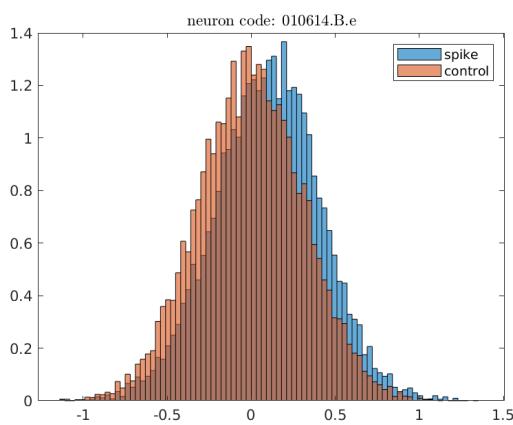
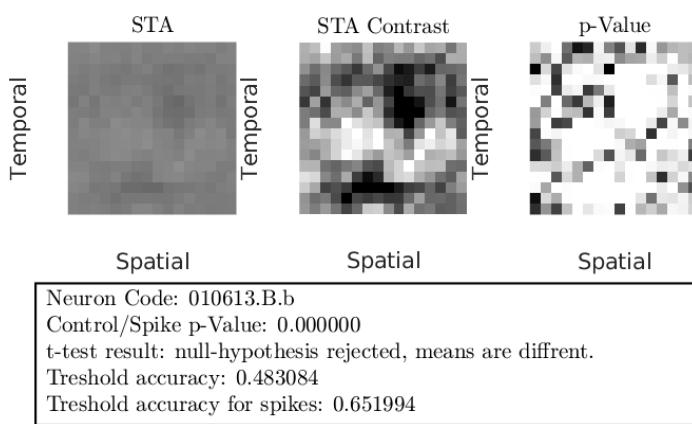
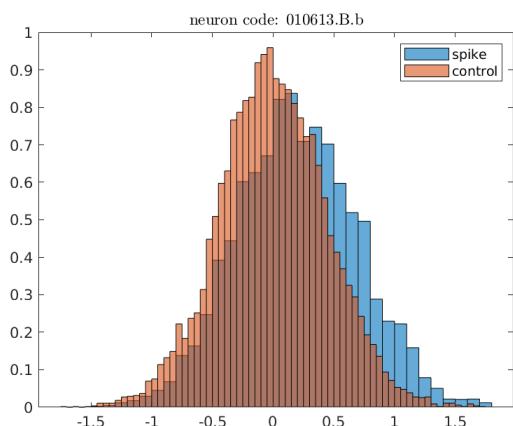
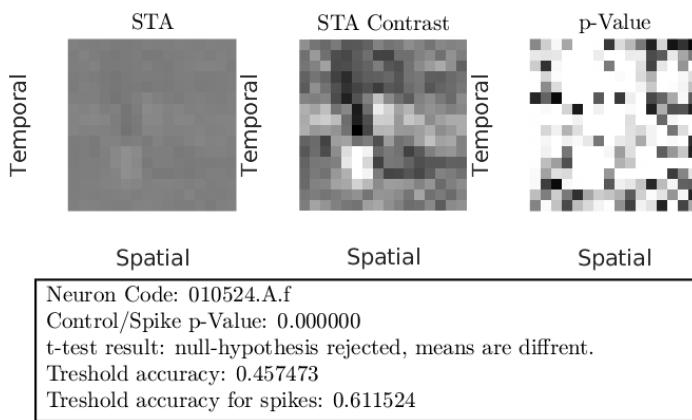
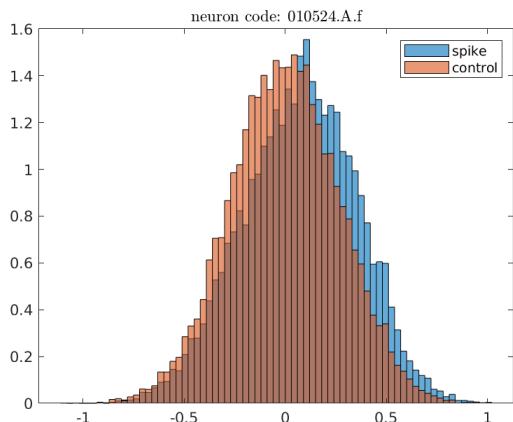


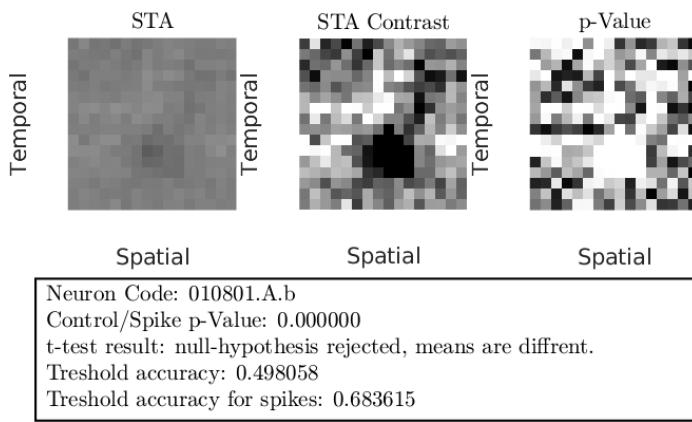
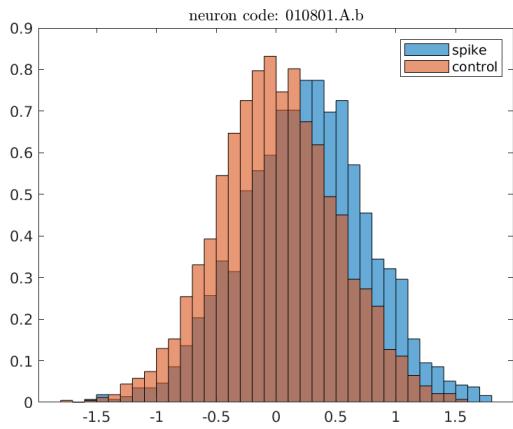
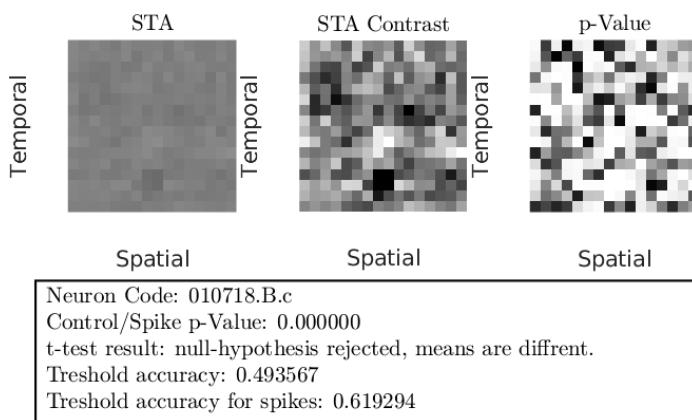
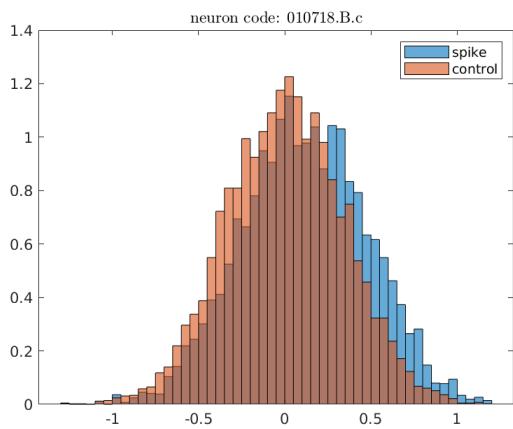
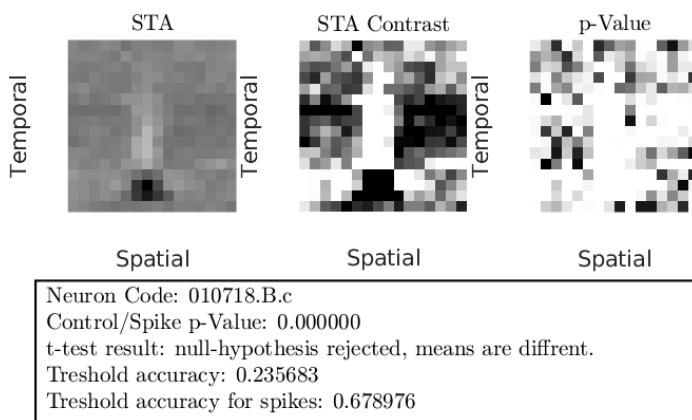
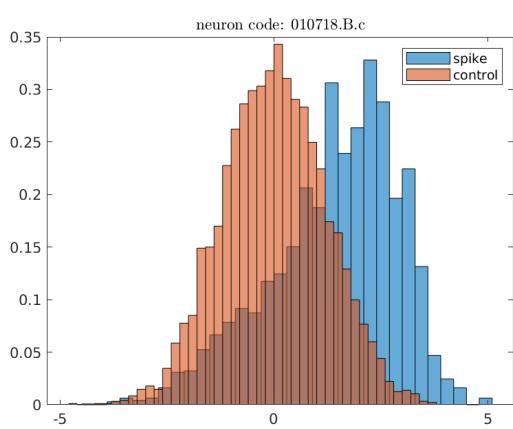
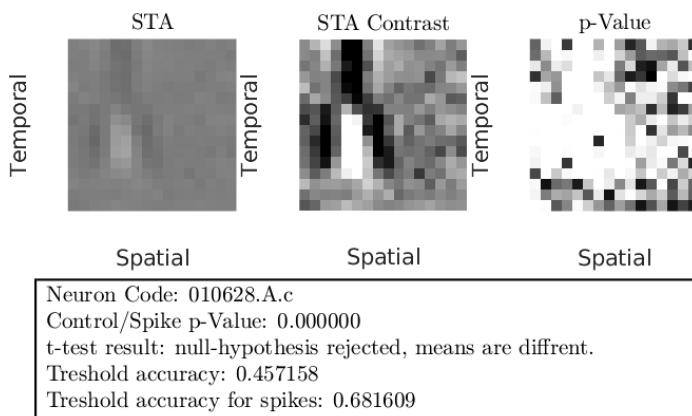
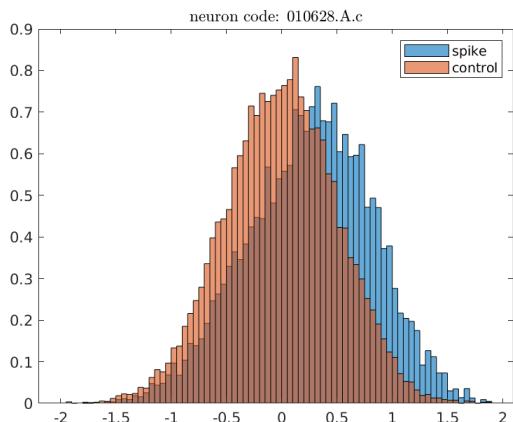


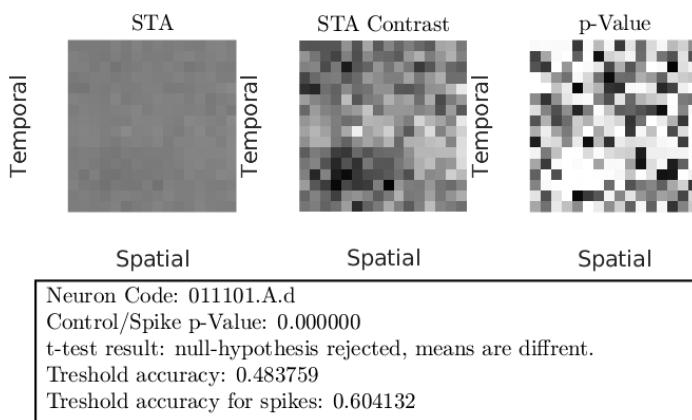
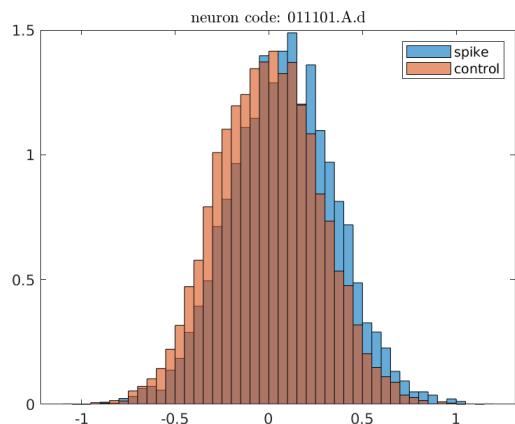
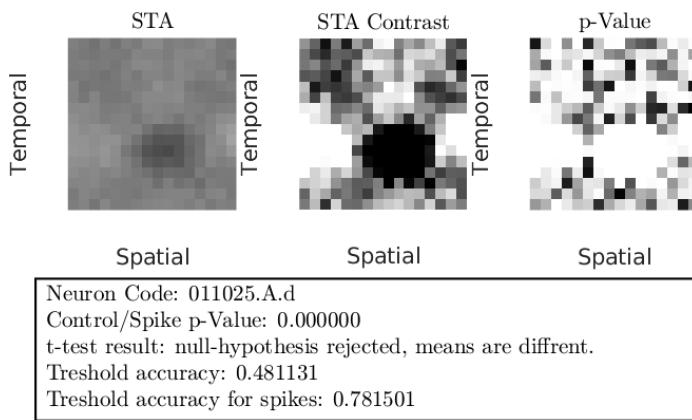
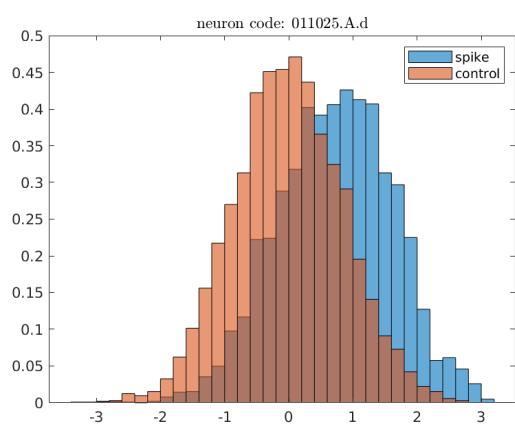
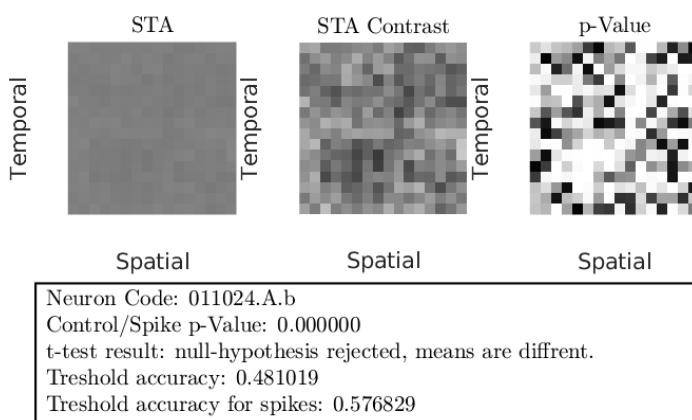
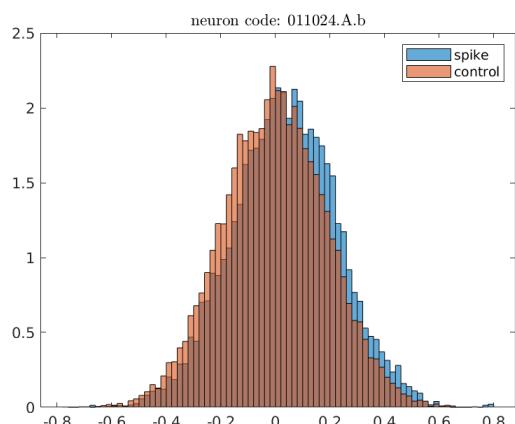
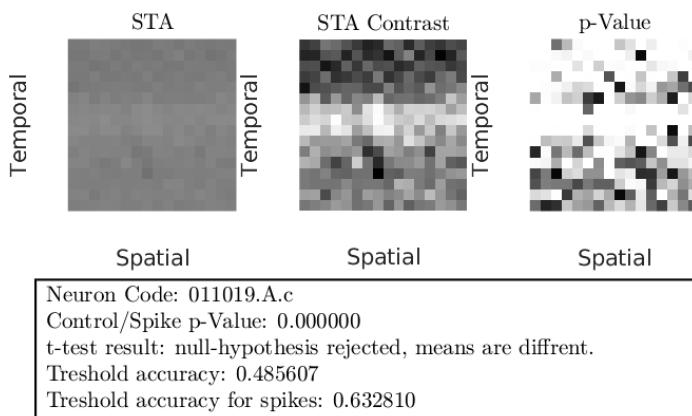
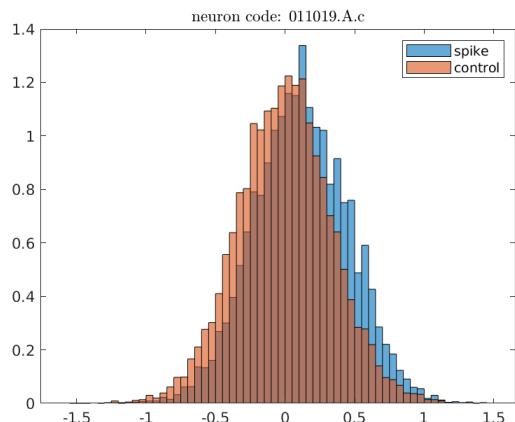


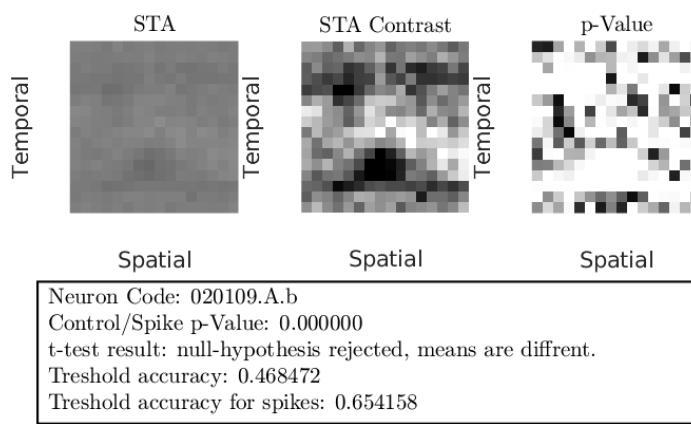
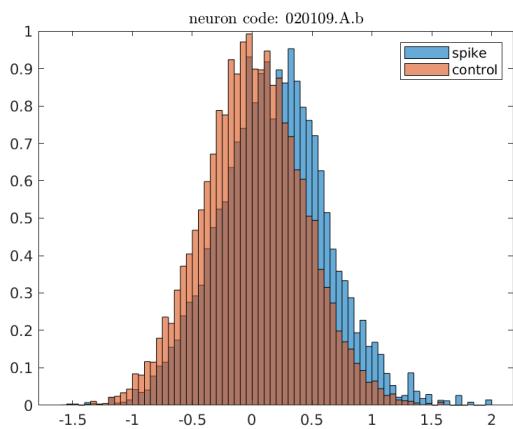
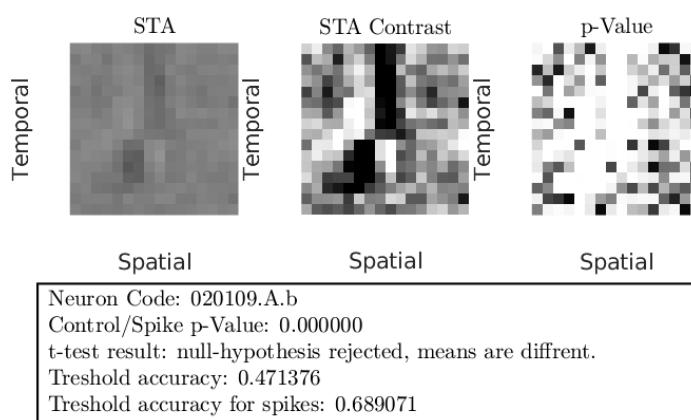
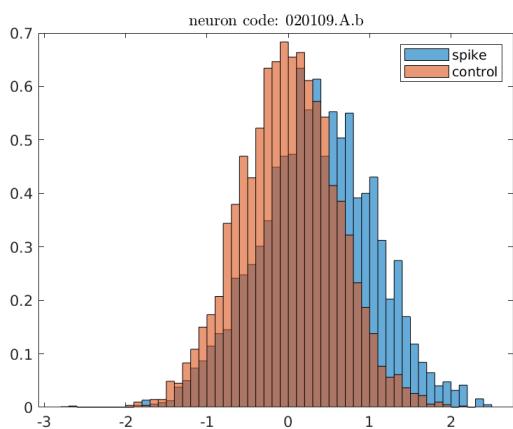
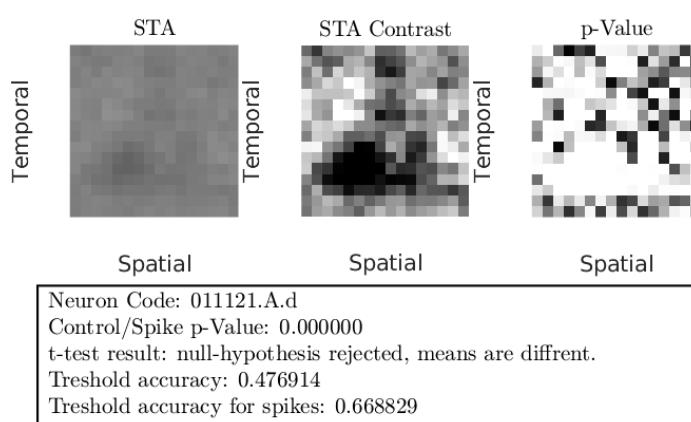
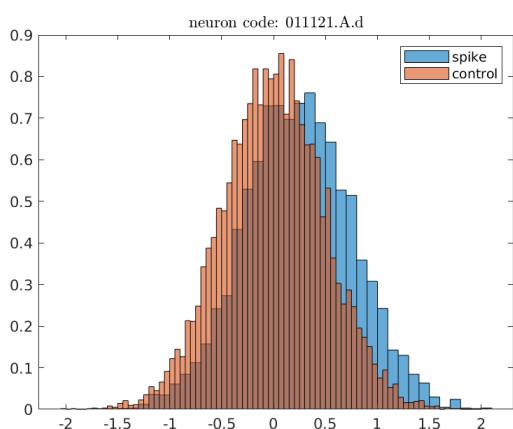
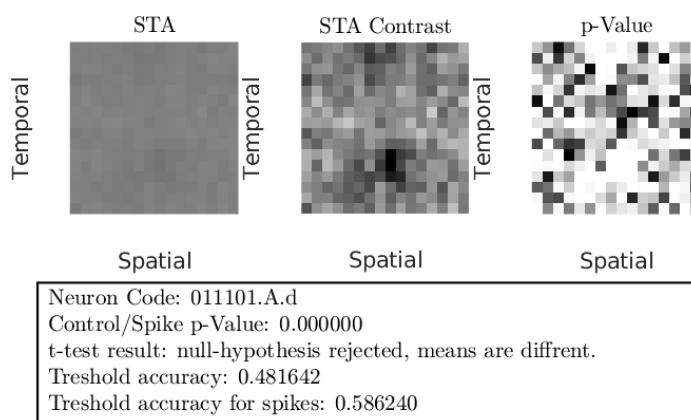
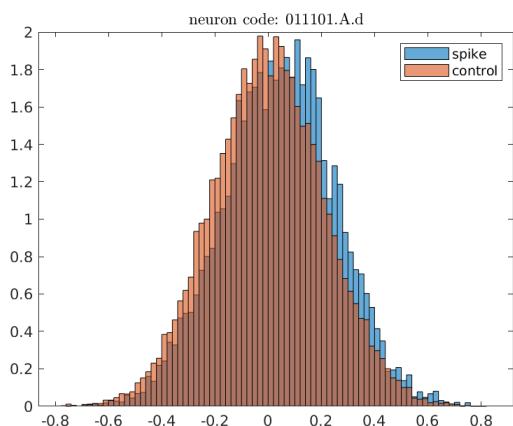


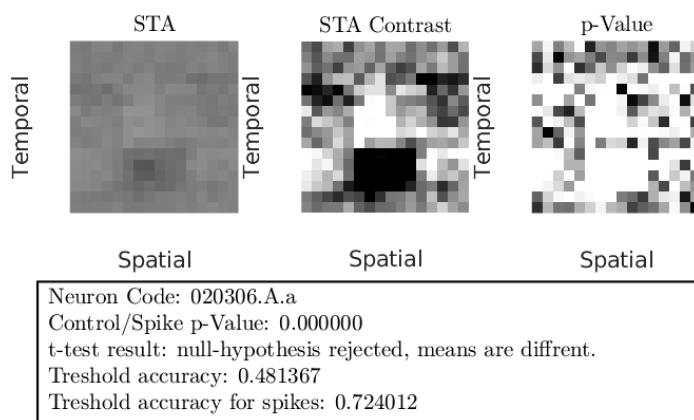
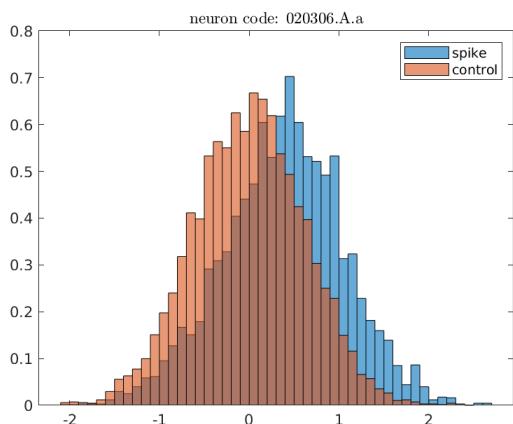
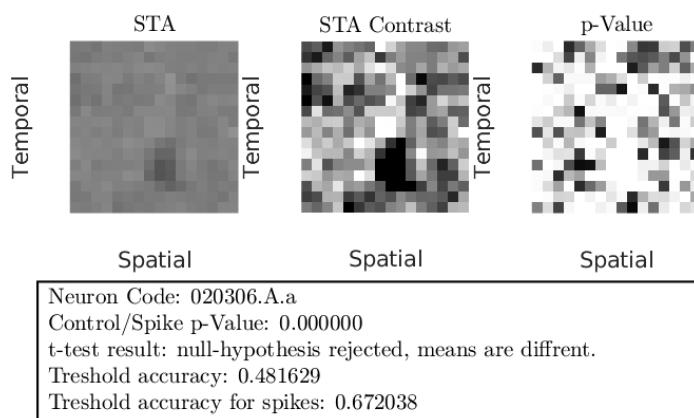
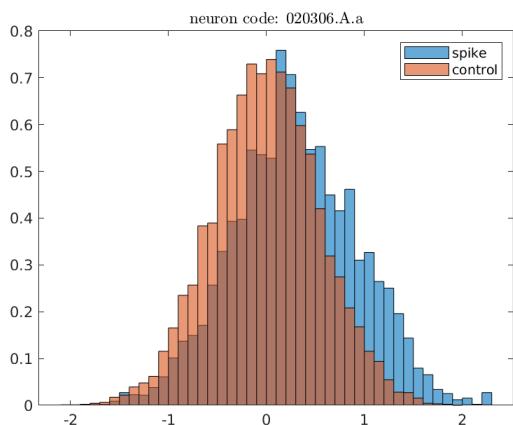
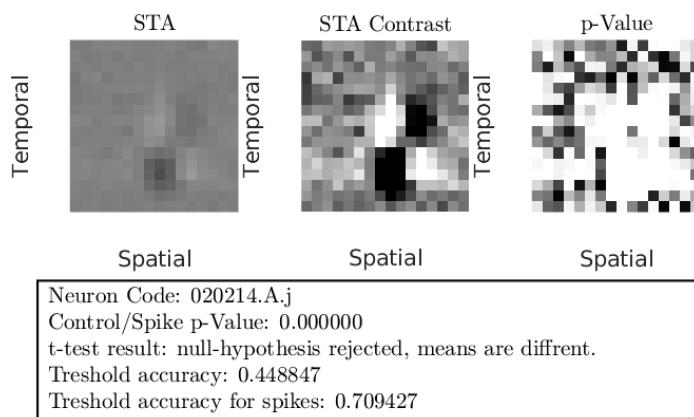
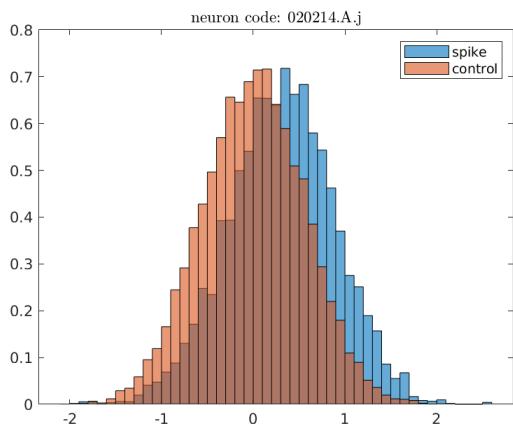
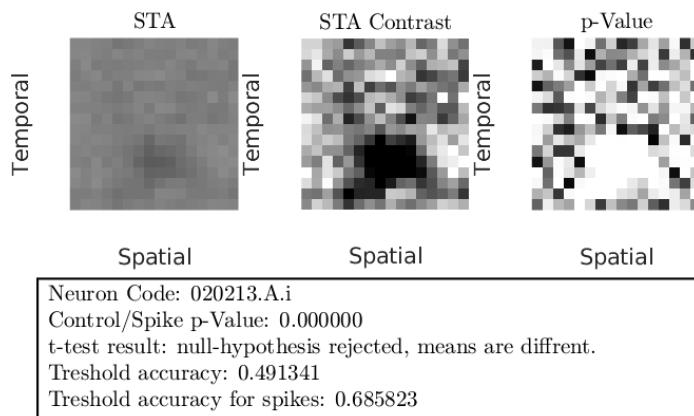
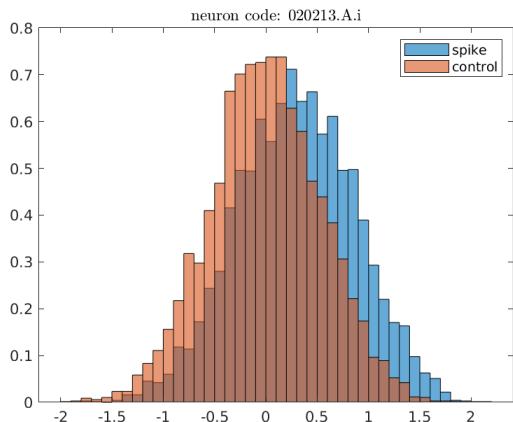


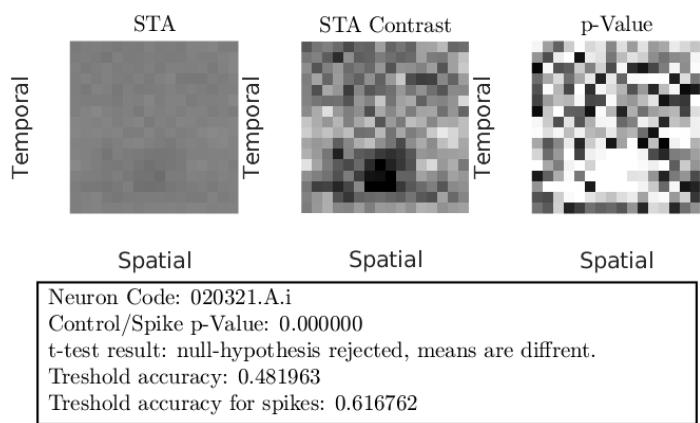
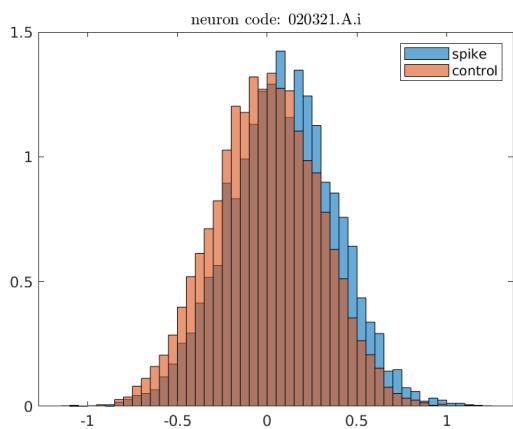
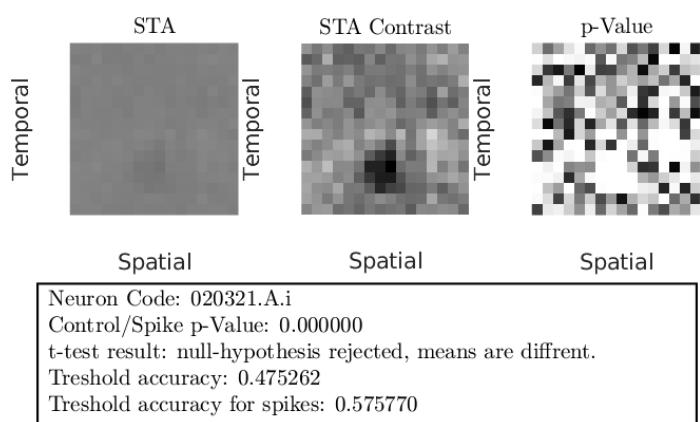
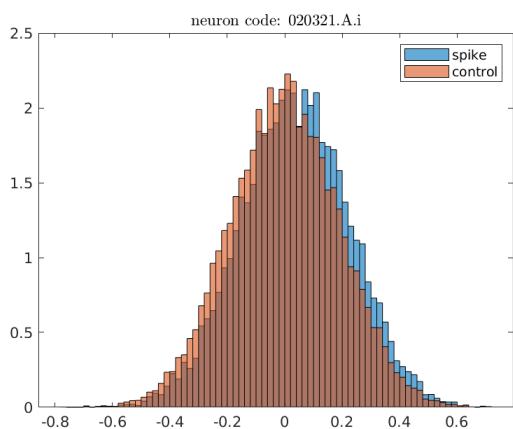
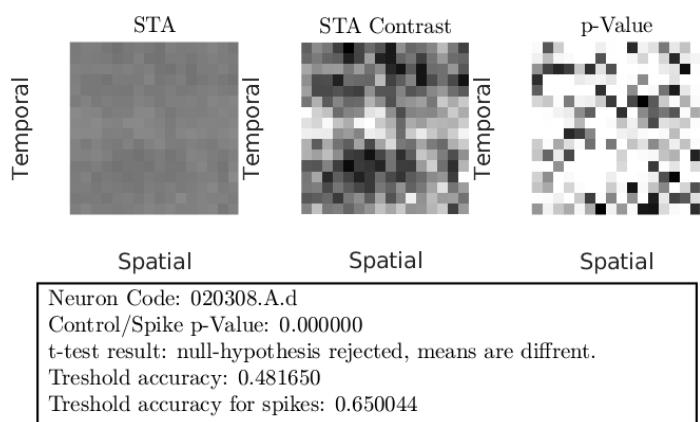
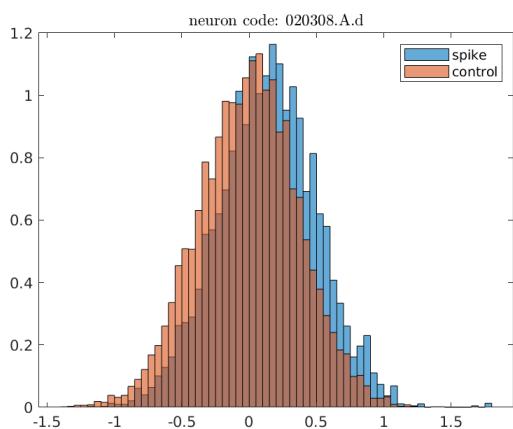






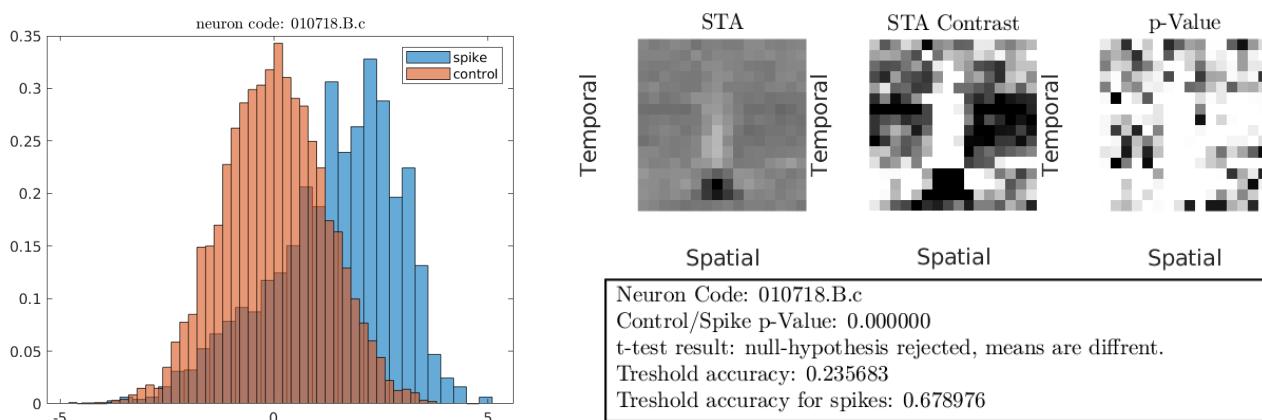




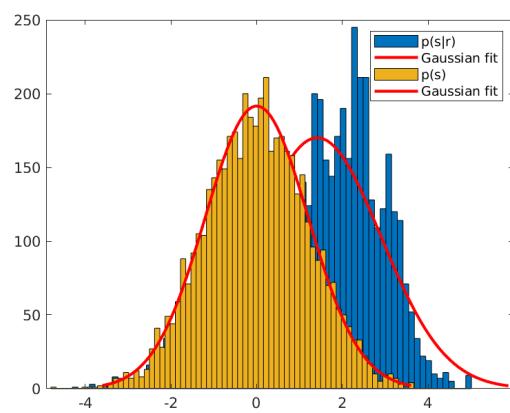


٧.٣ بخش هفتم

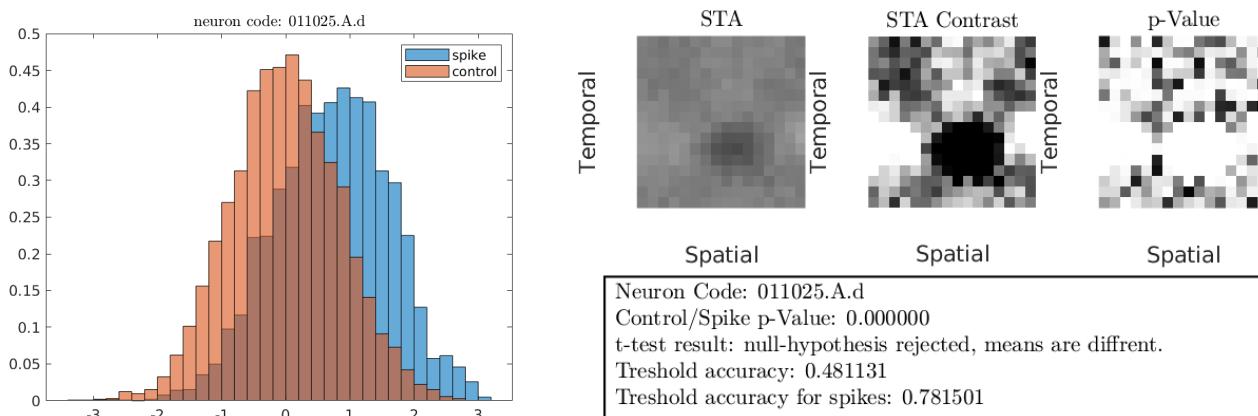
با اینکه میانگین توزیع‌ها با توجه به تست‌های آماری با هم تفاوت دارند از آنجایی که تقریباً تمامی نمودارهای رسم شده نشان می‌دهند که توزیع تصویر تحریک‌ها بر راستای STA برای کلیه تحریک‌ها و تحریک‌هایی که موجب اسپاک شدن تفاوت چشمگیری با هم ندارند و به درستی نمی‌توان ترشهولدی پیدا کرد که با انتخاب آن داده‌ها را با دقت خوبی بتوان دسته‌بندی کرد. از طرف دیگر مولفه‌ی STA در بیشتر نورون‌ها الگوی معناداری ندارد و بیشتر شبیه نویز می‌باشد. به عبارتی تا حد خیلی زیادی می‌توان به اعای مقاله مبنی بر پیچیدگی نورون‌هایی که با آنها سروکار داریم استناد کرد. با این وجود می‌توان دید که برخی از نورون‌های جدولی که در بالا ارائه کردیم نه تنها پترنی با الگوی معنادار دارند بلکه نمودار توزیع آن‌ها به علت تفاوت میانگین که باهم دارند را به خوبی می‌توان تفکیک کرد که نسبت به پیچیده بودن رفتار این دسته از نورون‌ها می‌توان تردید داشت. از بین می‌توان نورون با کد 010718.B.c را مثال زد به نمودار و نتایج به دست آمده از آن توجه می‌کنیم.



می‌توانیم ببینیم که دو توزیع آن نسبتاً جدایی پذیر به نظر می‌آیند و همچنین الگو معناداری در تصویر با کانتراست بالای STA وجود دارد. البته باید توجه کنیم علت اینکه ترشهولد مناسب برای این نورون انتخاب نشده این است که توزیع spike در آن گاووسی به نظر نمی‌رسد و فیت گاووسی ما اطلاعات نادرستی از توزیع به ما می‌دهد. به این فیت و داده‌های حذف شده توسط آن توجه کنید.



در بین نورون‌های بررسی شده اما نورون‌های دیگری هم وجود داشتند که می‌شد الگوی معناداری آنها مشاهده کرد و از این بین می‌شود به نورون 011025.A.d اشاره کرد که توزیع نمودارهای آن هم جدا از هم به نظر می‌رسد.



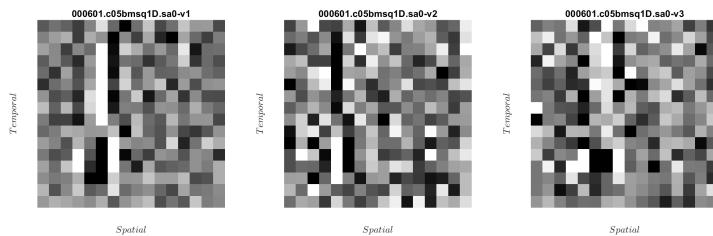
همچنین ترشحولدی که برای این نورون پیدا کردیم درصد قابل توجهی از فریم‌های منجر به اسپایک را شناسایی کرده است.

نهایتاً می‌توان گفت به طور کلی ادعای مقاله درباره پیچیده بودن این نورون‌ها درست به نظر می‌رسد با این حال در مورد تعدادی از آنها باید احتیاط بیشتری به خرج بدھیم.

۴ بررسی با روش STC

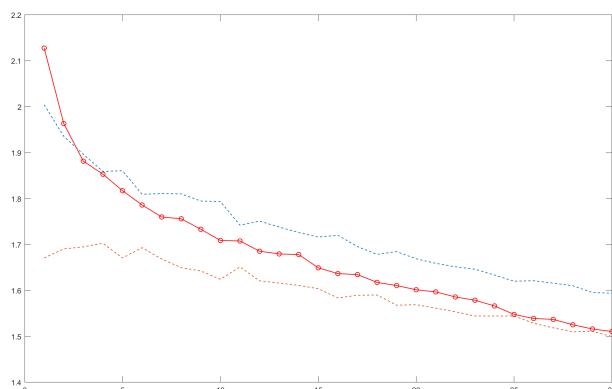
در این بخش، از نورون شماره ۱۰ با نام c05.000601 استفاده شده است.

۱۰.۴ بخش اول



شکل ۹: بردار ویژه‌های اول تا سوم

۲۰.۴ بخش دوم



شکل ۱۰: مقدار ویژه‌ها در کنار بازه‌ی اطمینان

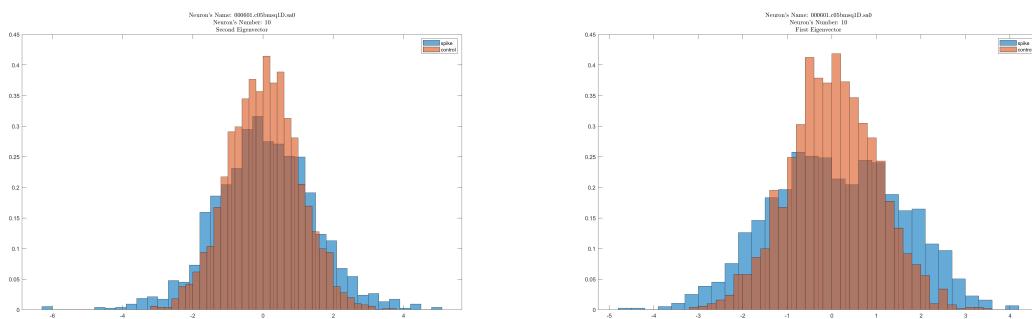
برای به دست آوردن بازه‌های اطمینان مربوط به هر نورون، به روش زیر عمل کردیم:
به تعداد زمان‌های اسپایک زدن نورون اصلی، زمان تصادفی با توزیع یکنواخت تولید کردیم و
این کار را ۲۰ مرتبه انجام دادیم. با استفاده از تابع Func_StimuliExtraction، تحريك‌های
متناظر با هر دنباله زمانی را به دست آوردیم. بنابراین در نهایت ۲۰ دنباله تصادفی اسپایک به دست
آوردیم که از هر کدام، یک ماتریس کوریلیشن تصادفی به دست می‌آید. با محاسبه مقادیر ویژه این
۲۰ ماتریس و مرتب‌سازی آن ها ۲۰ مجموعه ۲۵۶ تایی از مقادیر ویژه به دست آوردیم. سپس
برای محاسبه‌ی بازه‌ی اطمینان مربوط به مقدار ویژه n از ماتریس کوریلیشن اصلی، به سراغ مقادیر

ویژه $n = 20$ ماتریس کنترل رفته و میانگین و انحراف معیار این جامعه (جامعه‌ی $n = 20$ امین مقدار ویژه که 20 نمونه از آن را در اختیار داریم) را حساب کردیم. سپس بازه اطمینان را به صورت $mean \pm 10/4SD$ تعریف کردیم.

۳.۴ بخش سوم

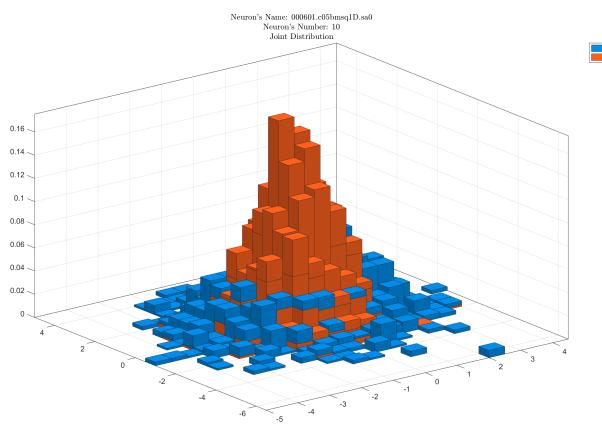
داده‌های به دست آمده از دو سؤال قبلی، این را به ما نشان میدهد که در میان راستاهای پیدا شده در فضای تحریک‌ها، فقط چند راستای معنی‌دار وجود دارند و آن راستاهای بردار ویژه‌هایی با معنی دارند که در شکل مربوط به سؤال ۱، دو نمونه از آن‌ها (v_1 ، v_2) قابل مشاهده هستند. از طرفی، سایر بردار ویژه‌ها ارزش خاصی ندارند و می‌توانیم برای کاهش بُعد مسئله و سادگی، آنها را در نظر نگیریم و می‌دانیم که اطلاعات زیادی را از دست نمی‌دهیم. در واقع، این بردارهای با معنی، متناظر با مقادیر ویژه بزرگ هستند، چرا که بزرگی مقادیر ویژه، نشانگر پراکندگی اطلاعات در راستای بردار ویژه‌های متناظر است و این بدان معنی است که این راستا، اطلاعات زیادی دربر دارد و در مقابل، راستاهایی با مقادیر ویژه کوچک، پراکندگی کمی دارند و اطلاعات زیادی در آنها وجود ندارد. بازه‌های اطمینان بدست آمده در سؤال ۲ نیز همین مسئله را تأیید می‌کنند. یعنی فقط مقادیر ویژه‌ای ارزشمند هستند (و دارای بردار ویژه معنی‌دار می‌باشند) که به شکل قابل قبولی از دیگران بزرگ‌تر باشند و خارج از بازه اطمینان تعیین شده قرار بگیرند.

٤٠٤ بخش چهارم



(ب) تصویر تحریک‌ها بر بردار ویژه‌ی دوم

(آ) تصویر تحریک‌ها بر بردار ویژه‌ی اول



(ج) توزیع مشترک تصویرها

هیستوگرام‌های فوق، از نظر مفهومی مشابه هیستوگرام موجود در بخش ۳-۳ هستند. ابتدا هیستوگرام دوبعدی را در نظر بگیرید. در این شکل، دو توزیع به صورت همزمان رسم شده است که یکی مربوط به داده‌های تصادفی (control) و دیگری مربوط به داده‌های (spike) اصلی هستند. این شکل، نشان می‌دهد که این دو توزیع چه وضعیتی نسبت به هم دارند. محورهای افقی، در واقع راستاهای ۷۱ و ۷۲ هستند که تحریک‌های منجر به اسپایک و تحریک‌های تصادفی روی این دو راستا تصویر شده‌اند و این نمودار را تشکیل داده‌اند. مفهوم رسم این دو نمودار روی یک تصویر و مقایسه‌ی آن‌ها، این است که وضعیت یک تحریک، نسبت به این دو راستا چگونه باشد تا منجر به اسپایک بشود، یا نشود. به عنوان نمونه، می‌بینید که قله‌ی هیستوگرام نارنجی رنگ (control) (control) در نقطه (۰،۰) واقع است. با توجه به این که در حقیقت، مقادیر روی محورها، حاصل ضرب داخلی تحریک در ۷۱ و ۷۲ هستند. نقطه (۰،۰) راستایی را نشان می‌دهد که هم بر ۷۱ و هم بر ۷۲ عمود است. می‌بینیم که در چنین راستایی، اختلاف تحریک‌های منجر به اسپایک و تحریک‌های تصادفی تقریباً به ماکزیمم خود میرسد و این یعنی اگر یک تحریک نامشخص در این نقطه داشته باشیم، با احتمال بسیار زیاد، منجر به اسپایک نیست و احتمال بیشتری دارد که جزء

هیستوگرام نارنجی باشد تا هیستوگرام آبی. به همین ترتیب، در هر نقطه دلخواه از این هیستوگرام سه بعدی، می‌توان وضعیت دو توزیع منجر به اسپایک و غیرمنجر به اسپایک را مشاهده کرد و آن‌ها را مقایسه نمود.

دو هیستوگرام دو بعدی دیگر که در بالای هیستوگرام سه بعدی مشاهده می‌کنید، یکی مربوط به تصویر تحریک‌های منجر، و غیرمنجر به اسپایک روی ۷۱ و دیگری بر روی ۷۲ است. به تعبیری، این دو هیستوگرام، همان شکل سه بعدی هستند که روی یکی از بعدهای خود جمع شده (انتگرال گرفته شده) است. مفهوم این دو هیستوگرام نیز مشابه بخش ۳-۳ است، یعنی مقایسه ارتباط تحریک‌های منجر به اسپایک و تحریک‌های غیرمنجر به اسپایک با راستای ۷۲ و ۷۳، که هر چقدر دو نمودار (که بر روی یک شکل رسم شده‌اند) بیشتر از هم جدا باشند، نشان میدهد که راستای مربوطه، توانایی بیشتری در جدا کردن این دو گونه تحریک دارد. این حرف در مورد شکل سه بعدی نیز برقرار است و هرچقدر دو هیستوگرام سه بعدی اسپایک و کنترل بیشتر از هم جدا باشند، می‌توان گفت که راستاهای ۷۱ و ۷۲، در کنار هم، به شکل مناسب‌تری میتوانند تحریک‌ها را جدا کنند و می‌توانیم با داشتن تحریک و این دو راستا، در مورد اسپایک زدن یا نزدن در اثر آن تحریک صحبت کنیم. بنابراین، دیدیم که مفهوم این کار با بخش ۳-۳ و عملی که برای STA انجام دادیم، کاملاً مشابه است. تفاوت این دو بخش آن جاست که در قسمت ۳-۳، ما یک راستا را به عنوان راستای تصمیم‌گیرنده انتخاب کردیم که همان راستای STA بود؛ اما در این بخش، ادعا، می‌کنیم که یک راستا به تنها یی نمیتواند همه اطلاعات را در بر بگیرد و چند (دو) راستا، میتوانند این عمل را انجام بدهند و بنابراین، داده‌ها را در دو راستا تصویر می‌کنیم و به مقایسه تحریک‌های منجر به اسپایک و تحریک‌های بی‌اثر می‌پردازیم.

۵.۴ بخش پنجم

بخش عمده‌ی جواب این سؤال را میتوانید در توضیحات مربوط به سؤال قبل جستجو کنید. در واقع، مطابق با توضیحاتی که در بخش قبل دادیم، می‌خواهیم روشی انتخاب کنیم که با داشتن یک تحریک نامشخص، تصمیم بگیریم که آیا این تحریک منجر به اسپایک می‌شود یا نه. برای این کار به هیستوگرام دو بعدی بخش قبل مراجعه می‌کنیم. تحریک داده شده را روی دو (یا چند) راستای معنی‌دار تصویر می‌کنیم. فرض کنید دو راستای معنی‌دار داشته باشیم. در این صورت تصویر آن روی ۷۱ و ۷۲، دو تایی مرتبی مثل (۱۲، ۲۱) به ما می‌دهد. باید بینیم که در نقطه متناظر (۱۲، ۲۱) در هیستوگرام بخش قبل، کدام یک از دو دسته‌ی spike یا control مقدار بیشتری داشته‌اند. فرض کنید که تحریک‌های منجر به اسپایک در این نقطه مقدار بیشتری دارند. در این صورت می‌توانیم بگوییم احتمال آن که چنین تحریکی منجر به اسپایک شود، بیشتر از آن است که منجر به اسپایک نشود. بنابراین تصمیم گیری خود را بر مبنای این احتمال گذاشته و به عنوان جواب نهایی، اعلام

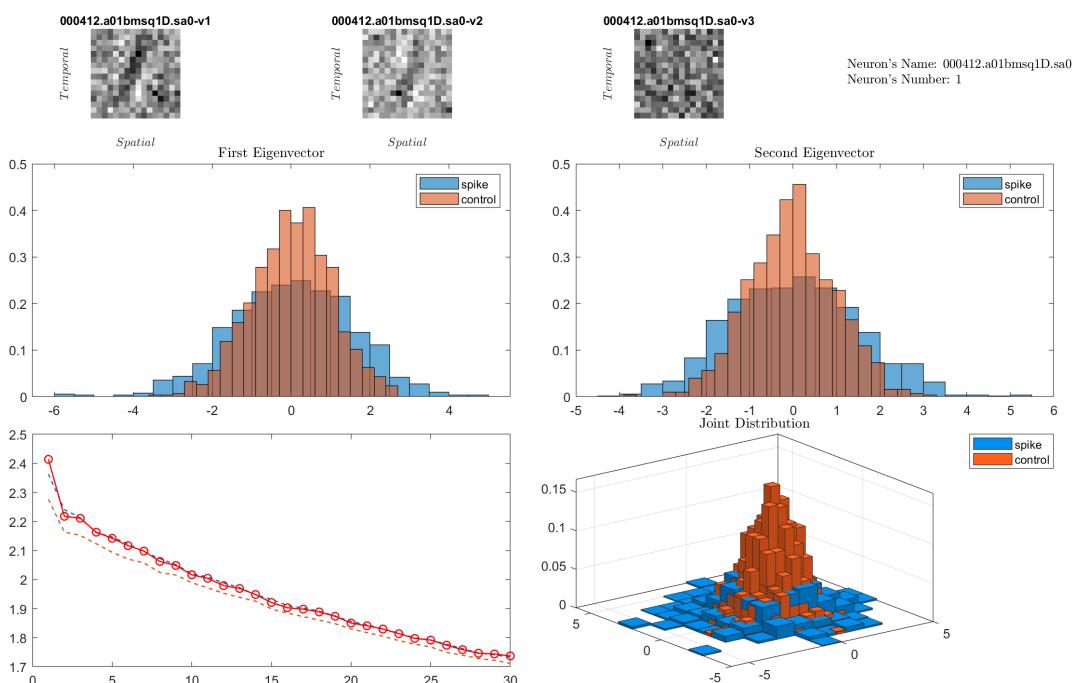
میکنیم که این تحریک منجر به اسپایک میشود. حال، با توجه به این توضیحات، واضح است که برای آن که یک ملاک کلی برای تشخیص اسپایک بیابیم، باید تلاقي دو منحنی را حساب کنیم که با فرض گوسی بودن هر دو، میتوانیم با داشتن میانگین و واریانس و ضریب همبستگی خطی مربوط به هر دو توزیع، محل تلاقي آنها را حساب کنیم و شرط اسپایک زدن را اعلام نماییم. برای آن که ببینیم این روش چقدر موفق عمل کرده، باید بینیم چند درصد از تحریک هایی که منجر به اسپایک شده‌اند را به درستی تشخیص میدهد. برای این کار، میتوان همان‌طور که گفته شد، محل تلاقي دو توزیع را پیدا کرد. برای نورون در نظر گرفته شده، با توجه به روش فوق، نتیجه به شکل زیر است.

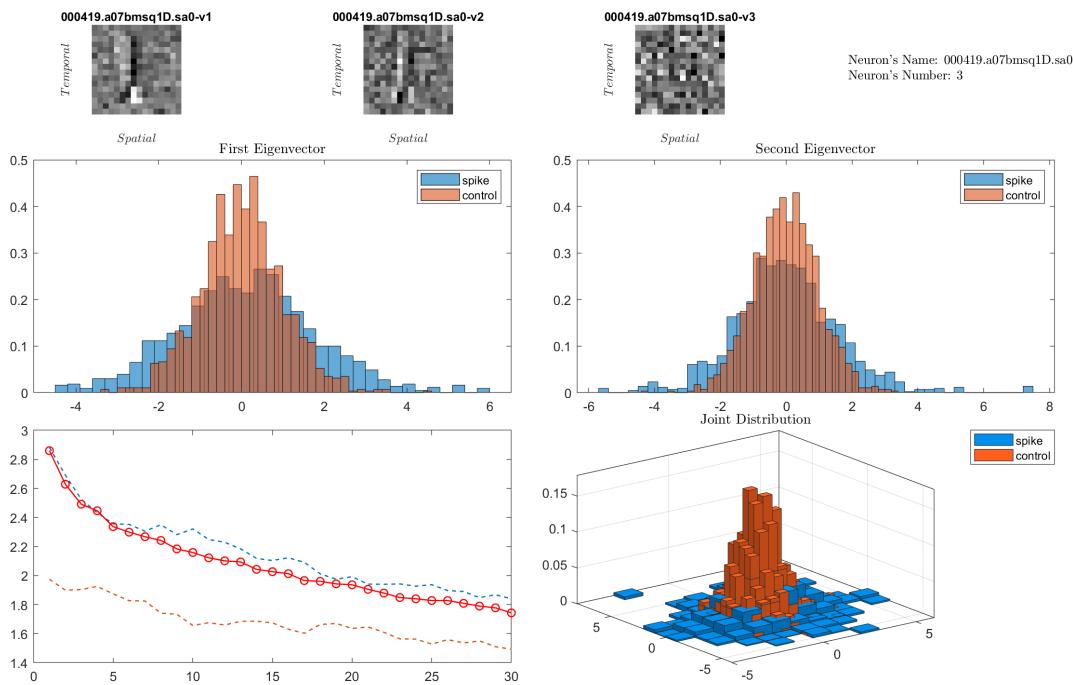
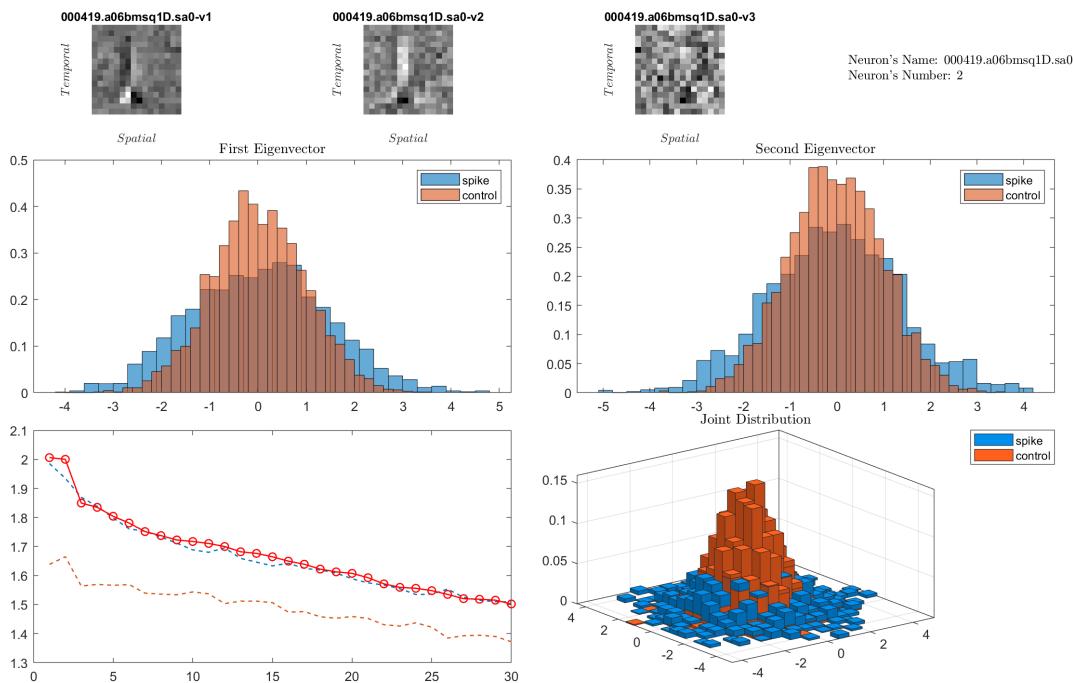
accepted_percentage =

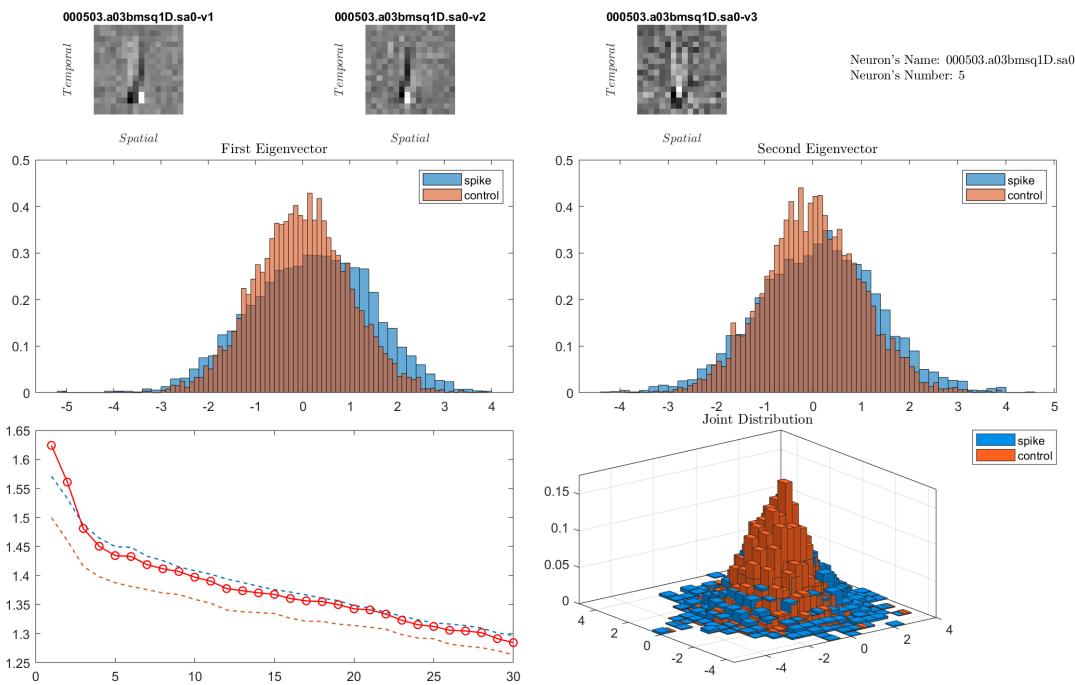
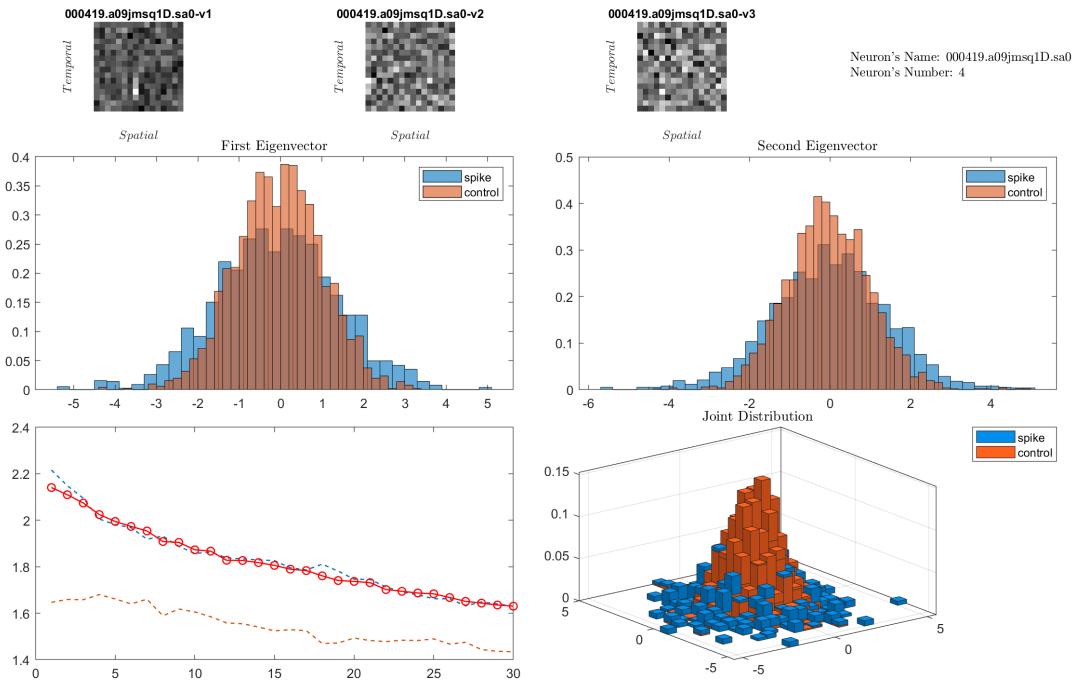
51.4741

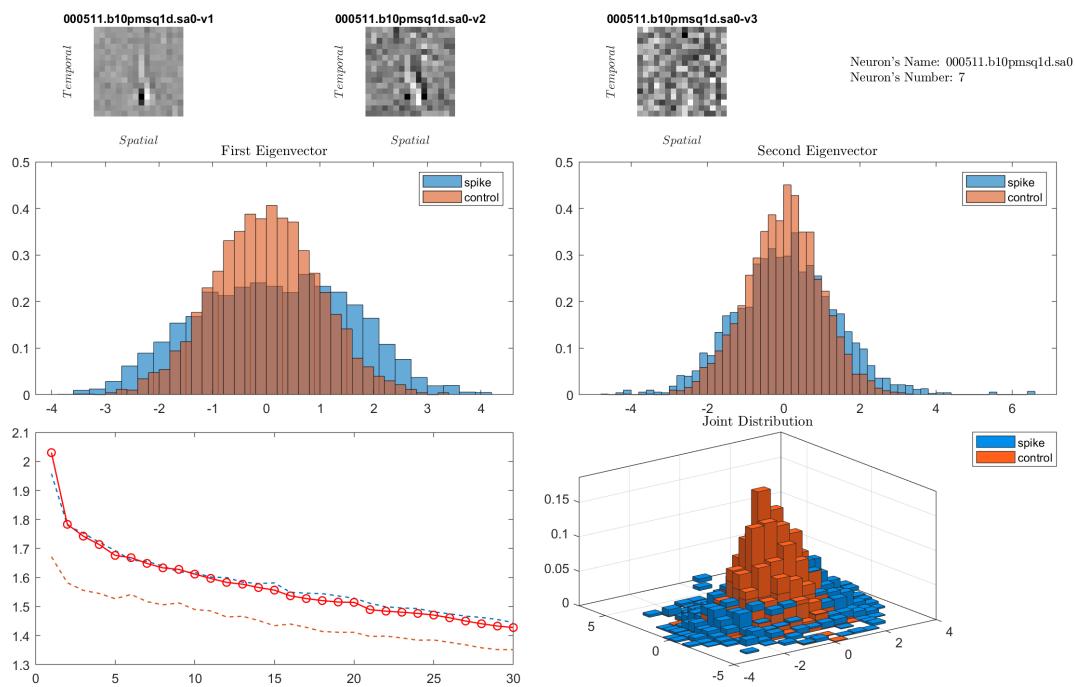
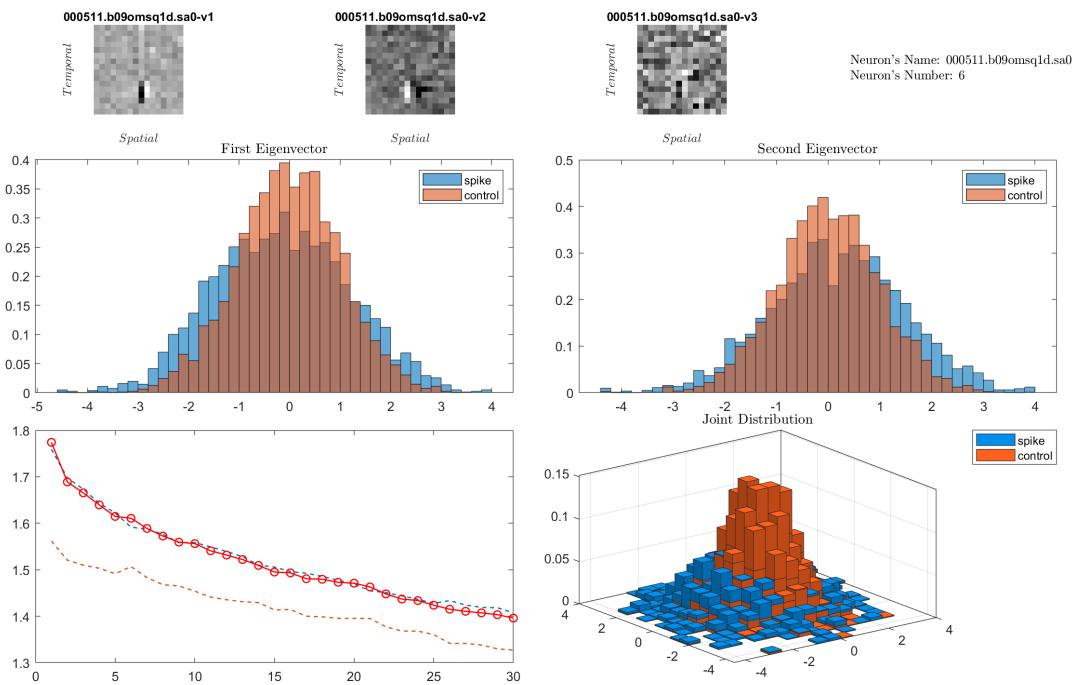
٦٠٤ بخش ششم

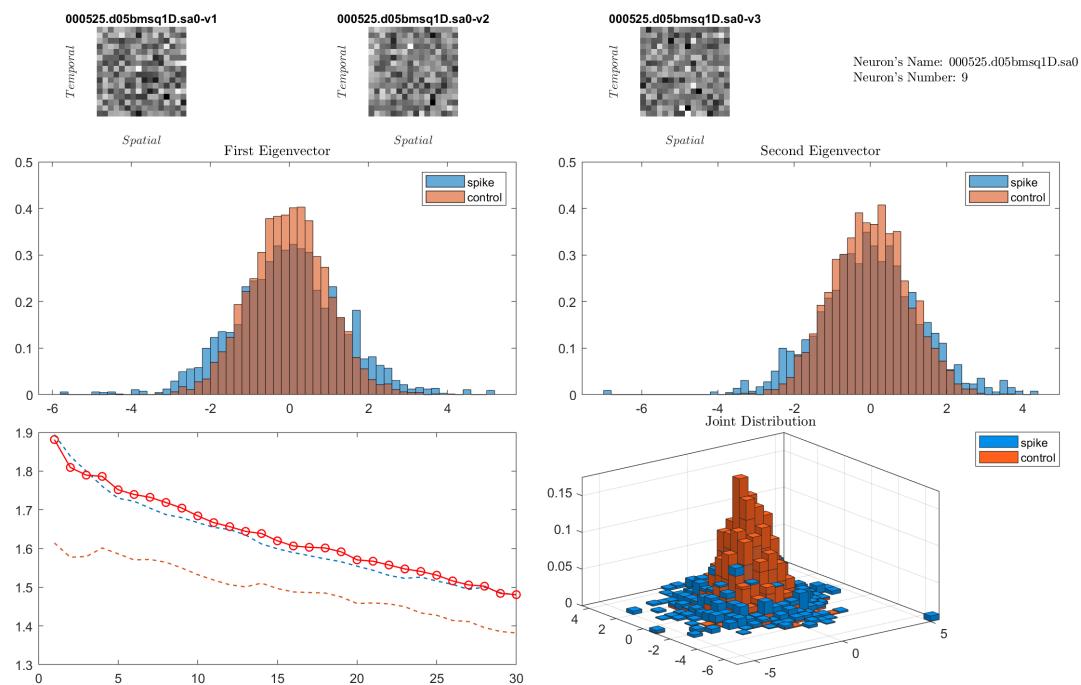
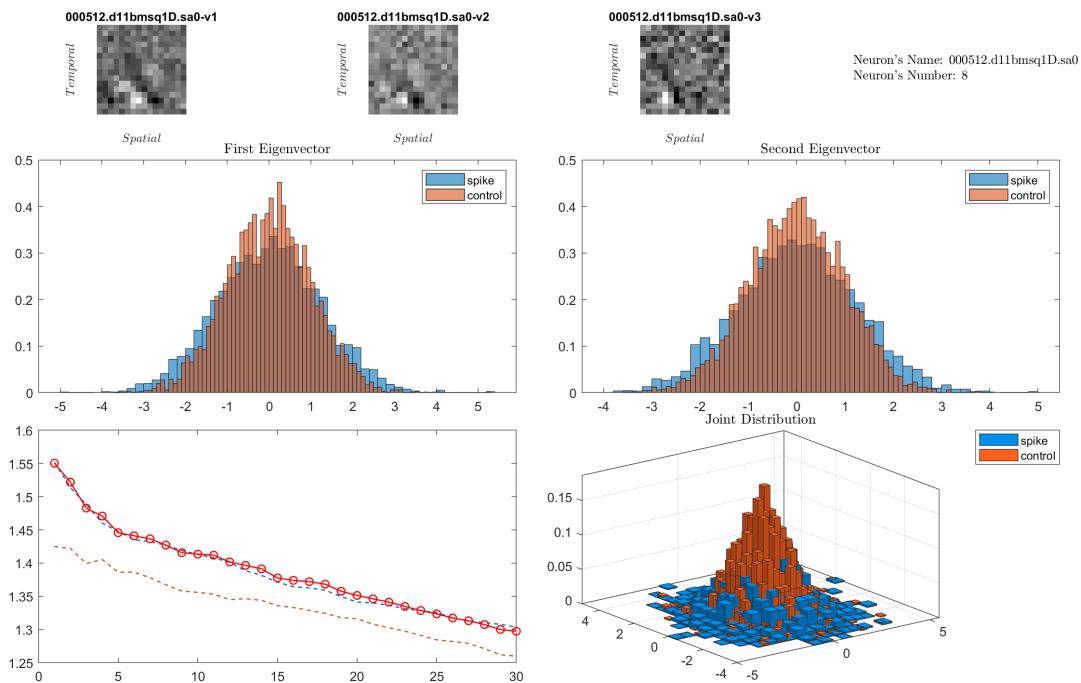
در این بخش با تکرار مراحل ۱ تا ۵ خروجی‌ها را بدست آورده و ذخیره کردیم.

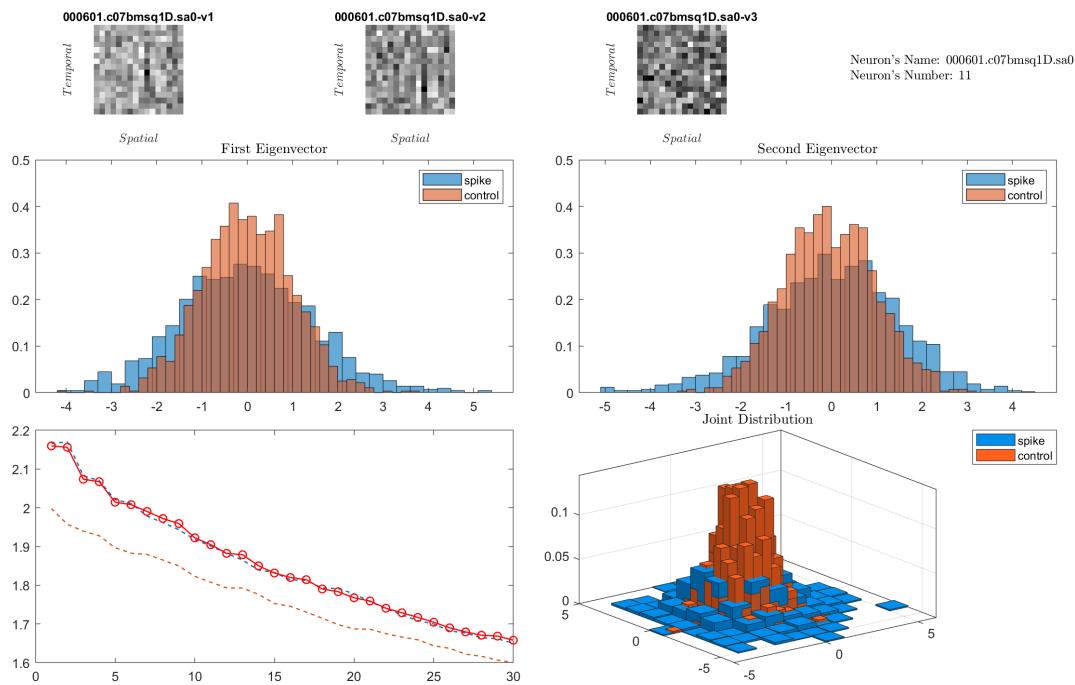
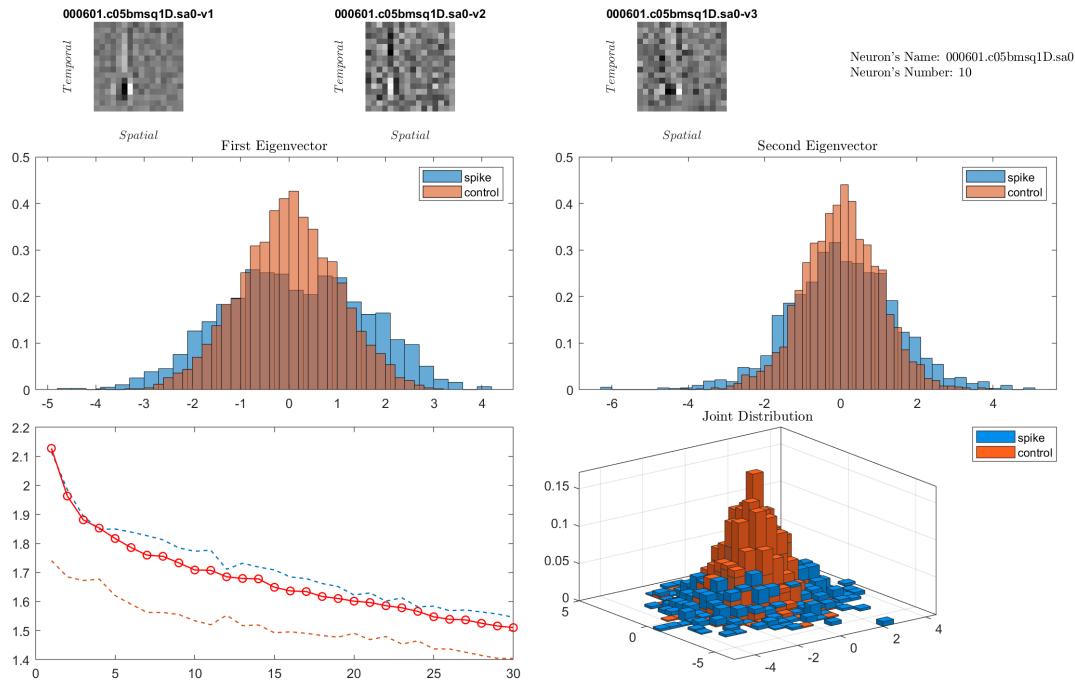


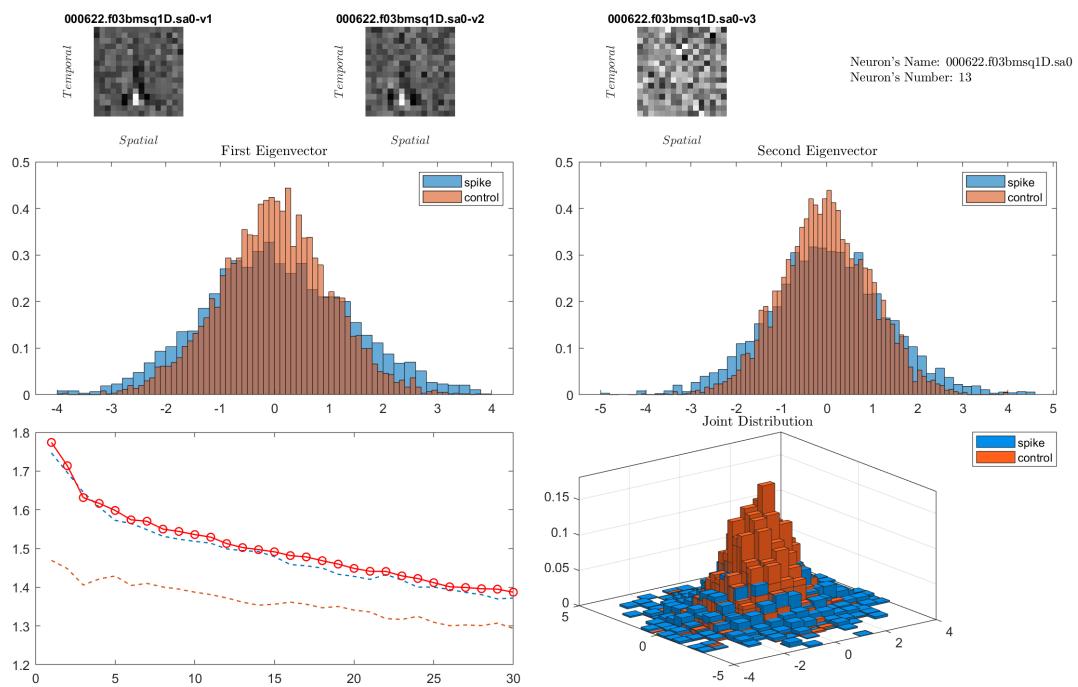
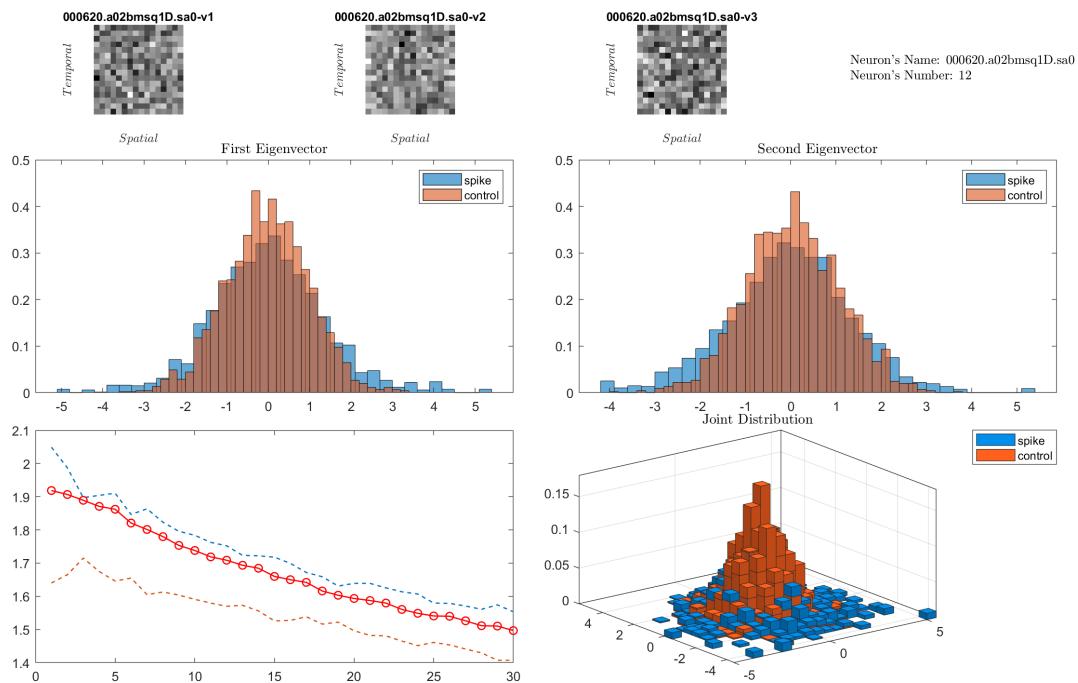


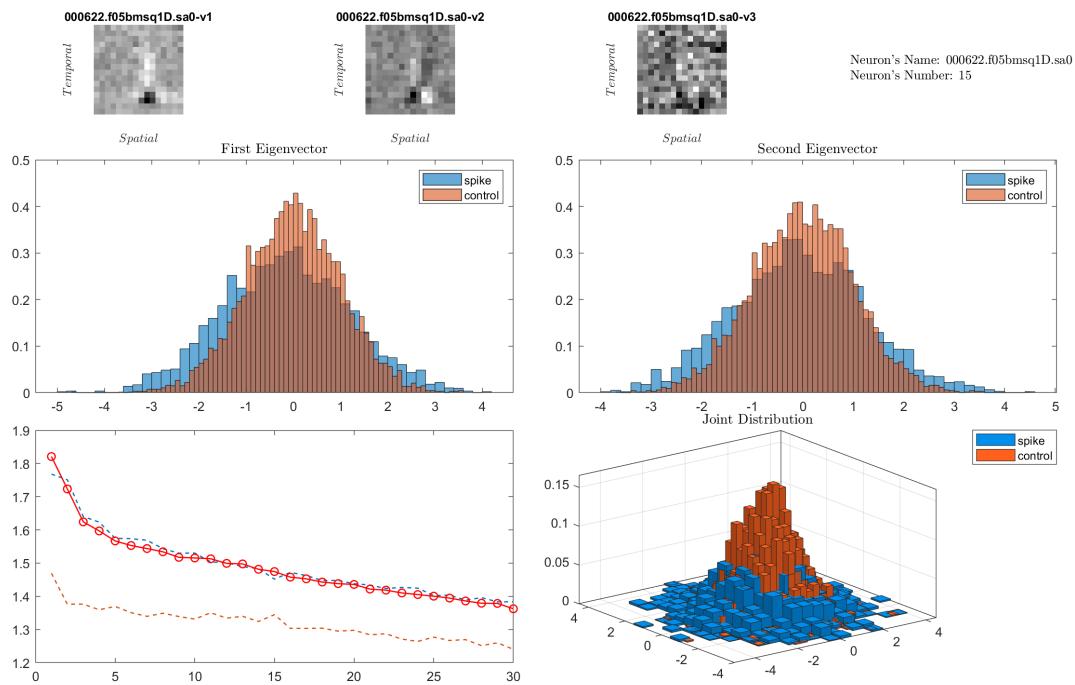
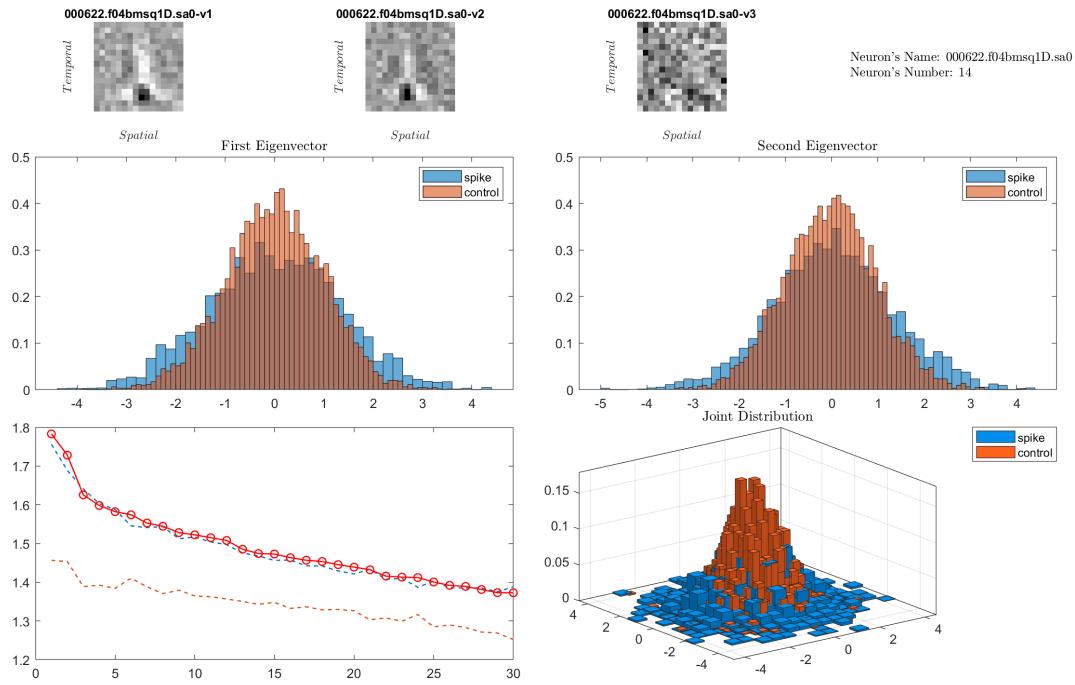


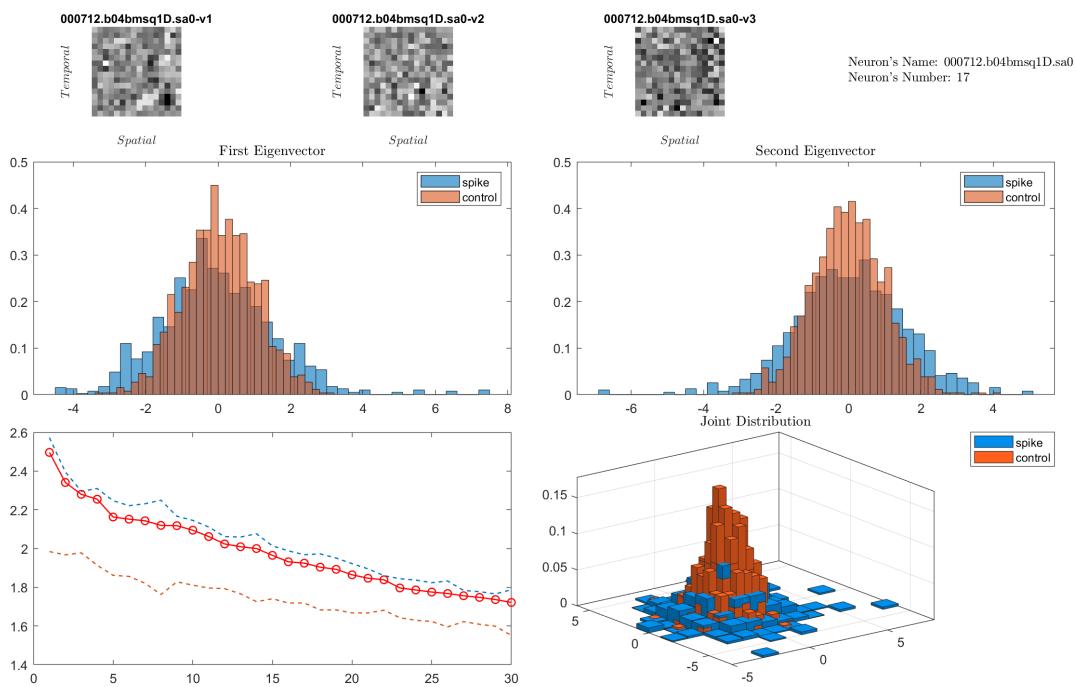
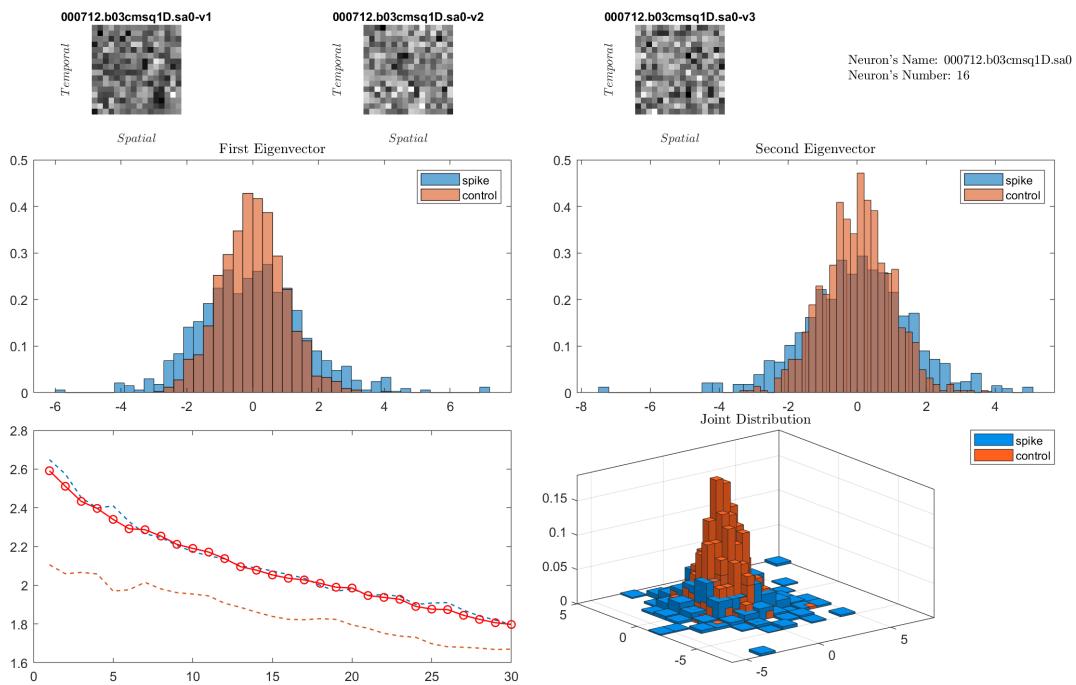


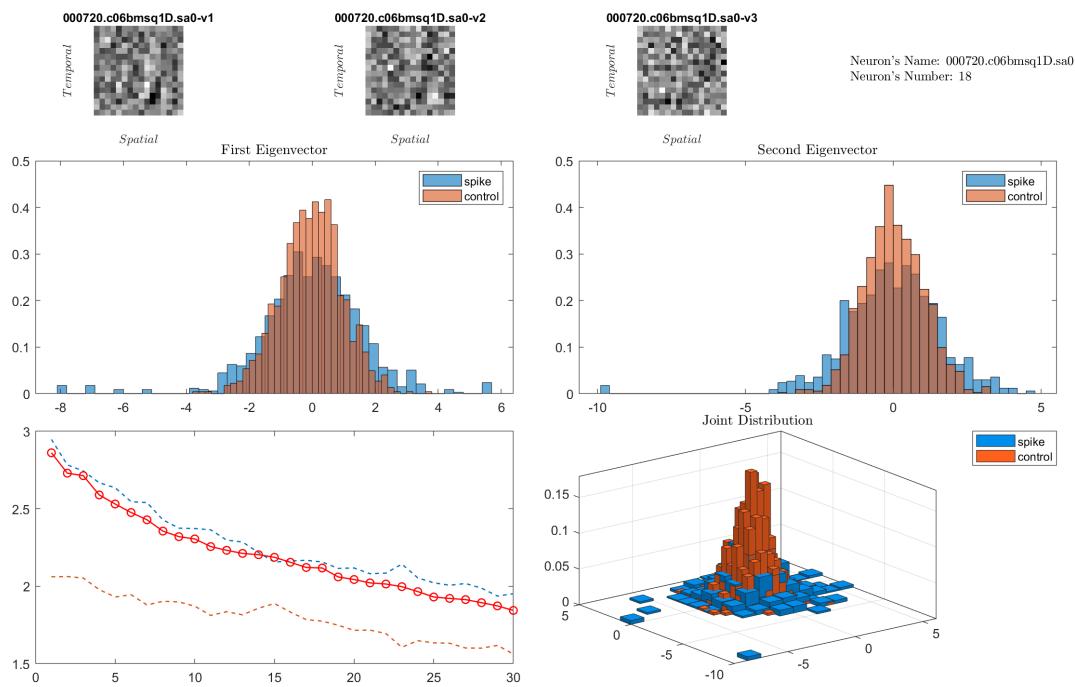


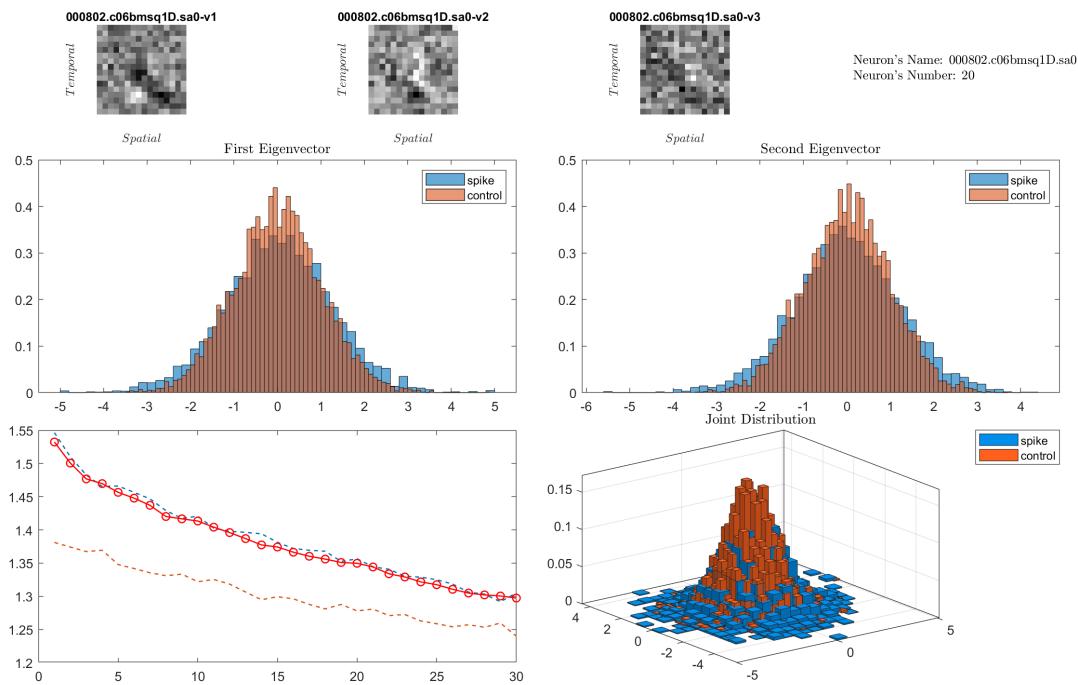
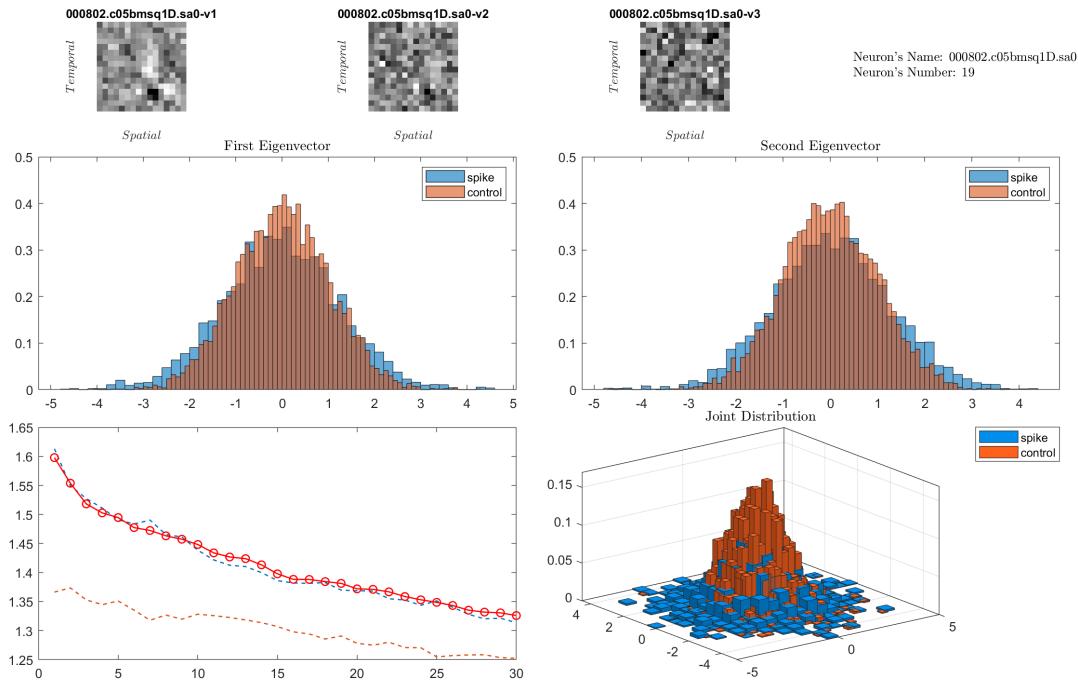


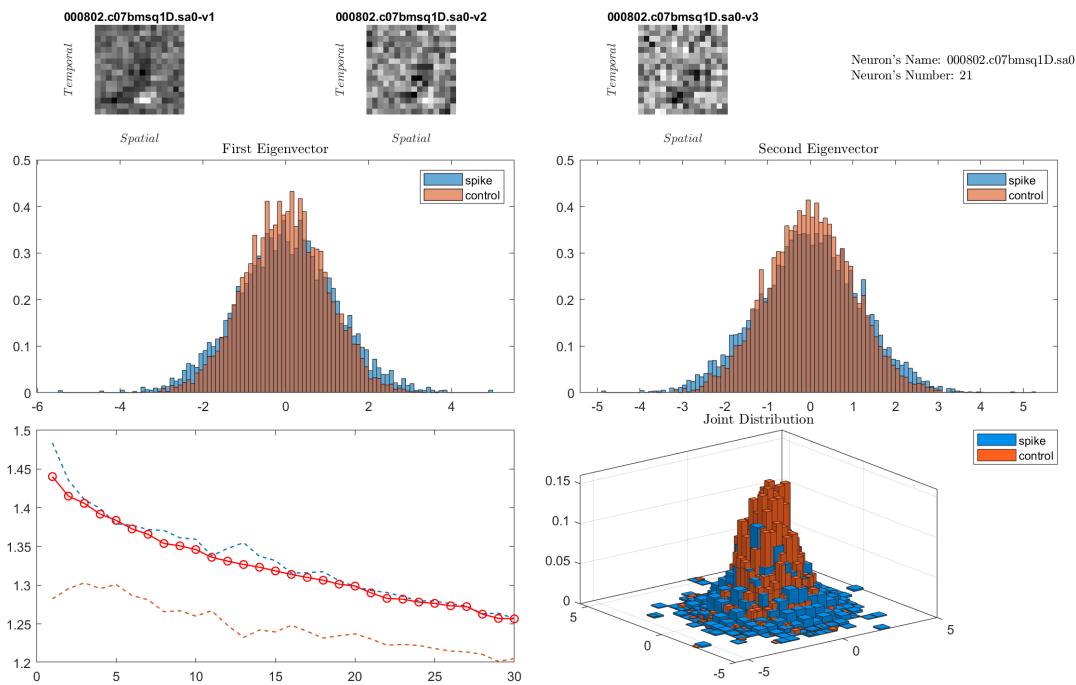


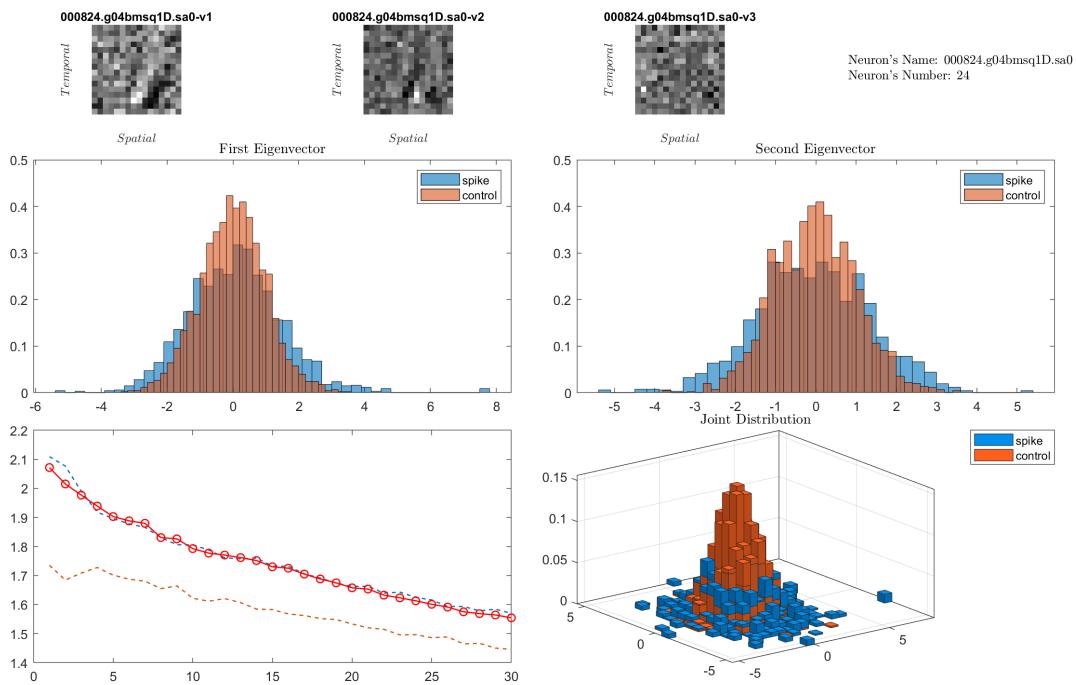
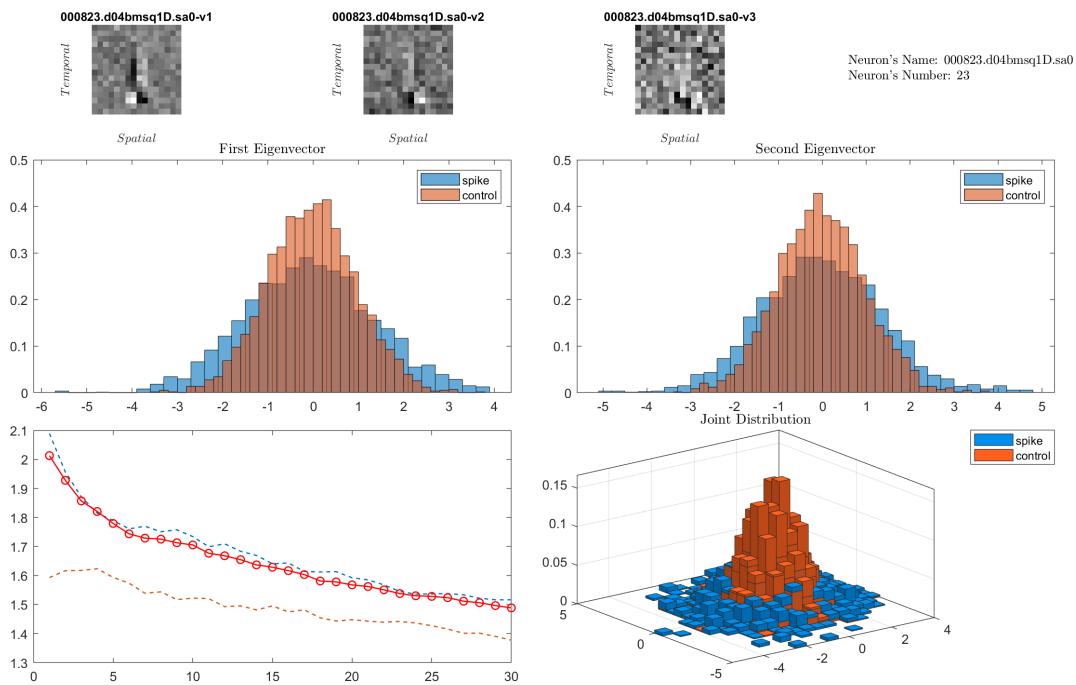


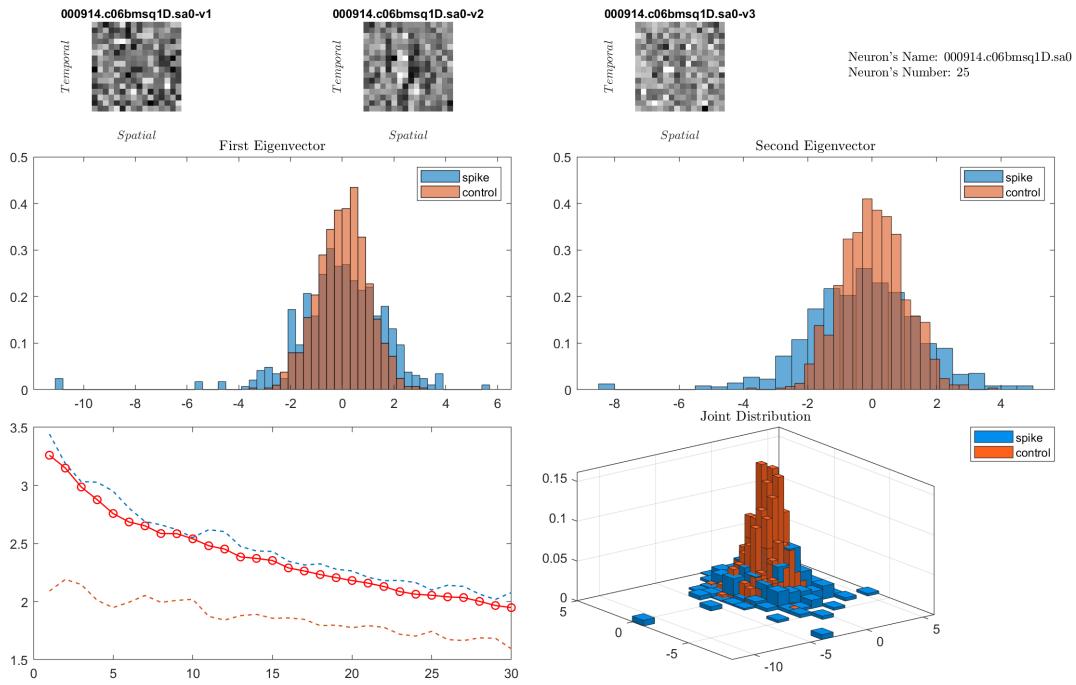


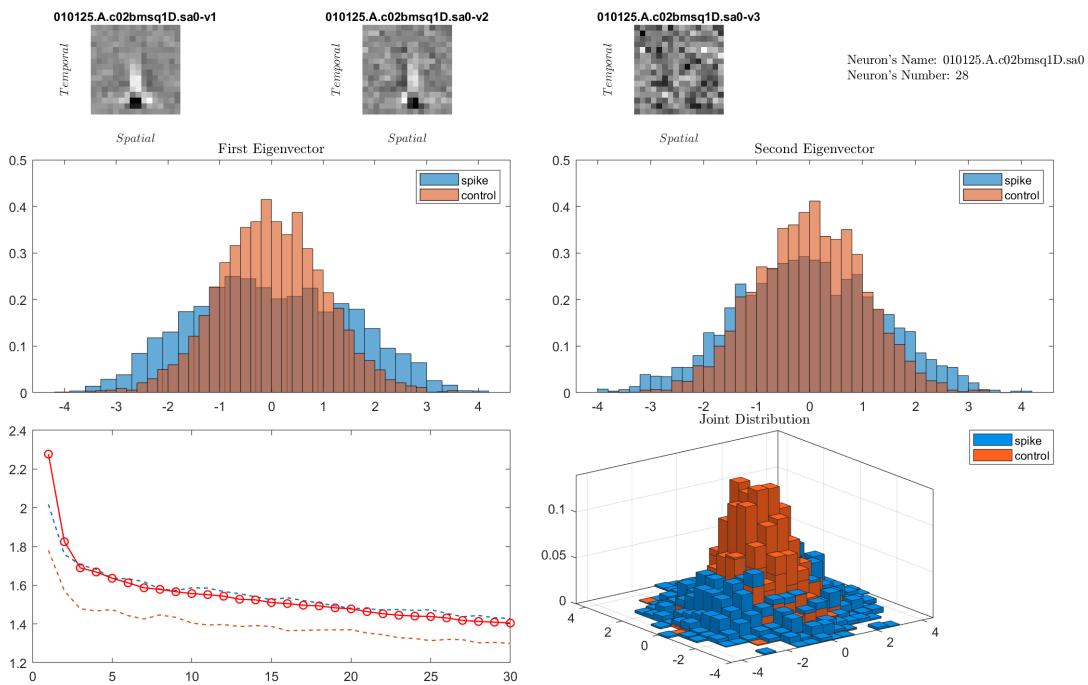
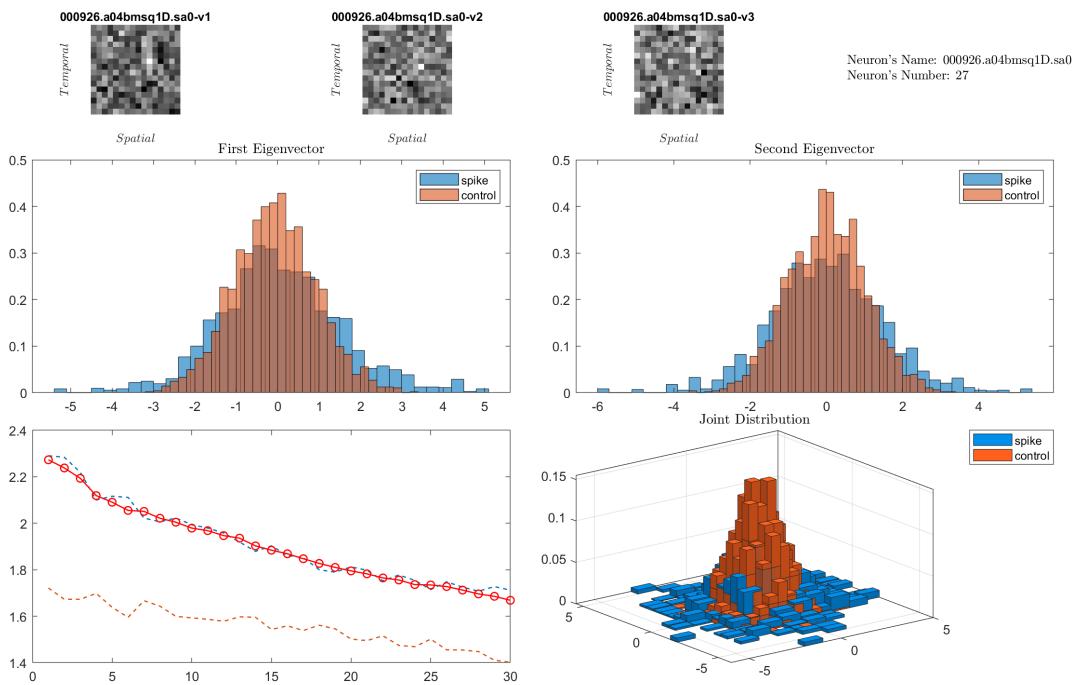


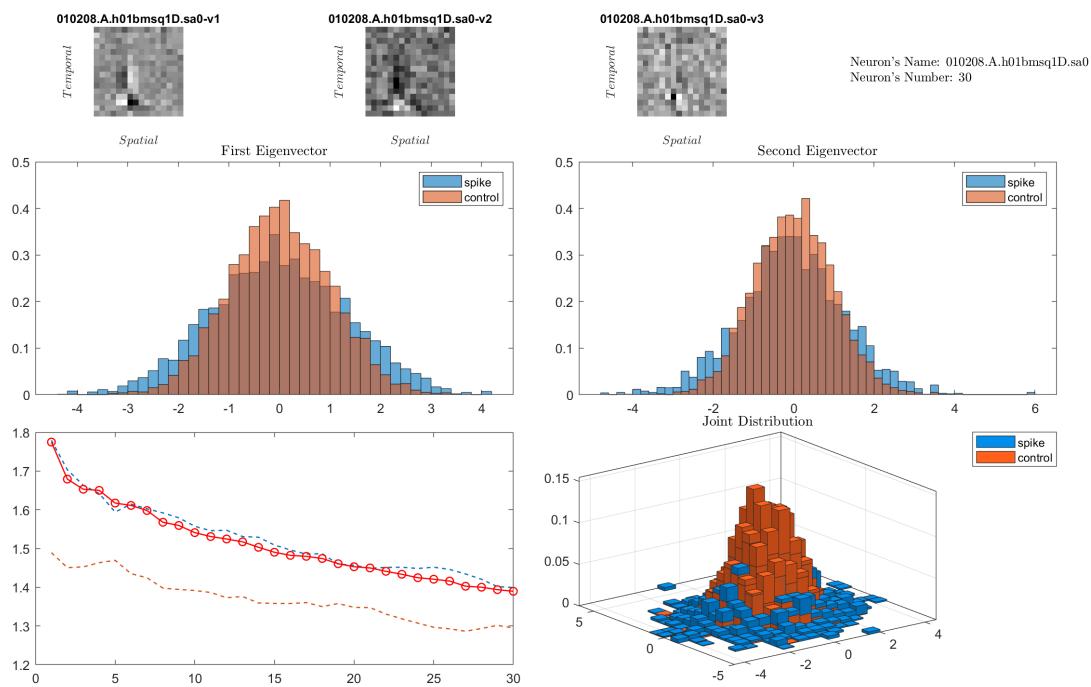
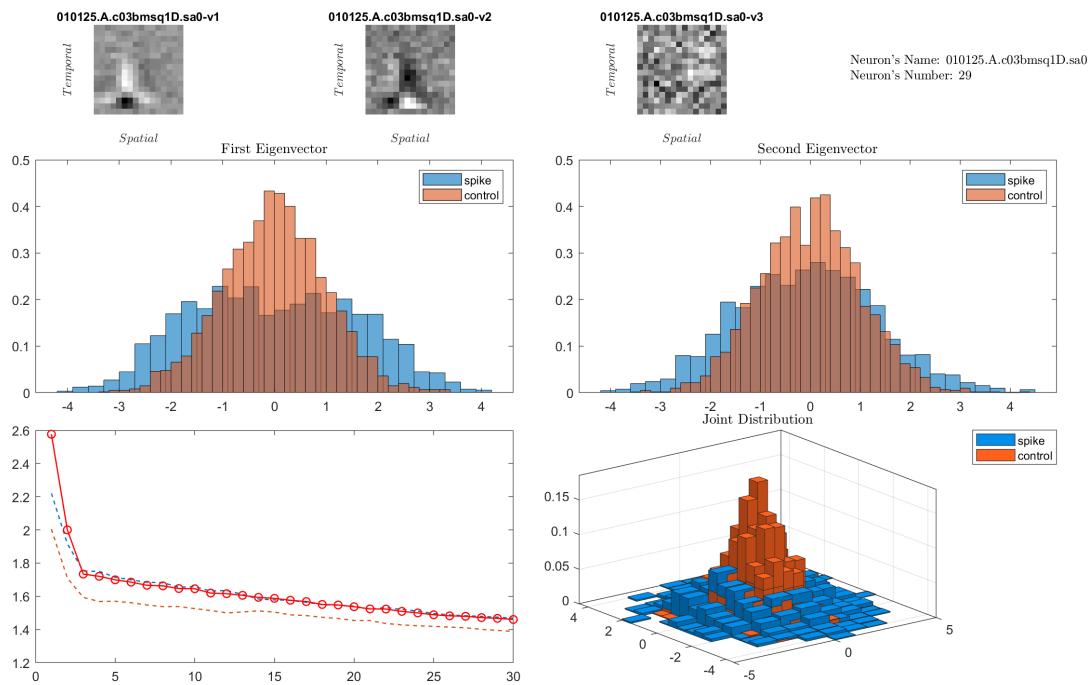


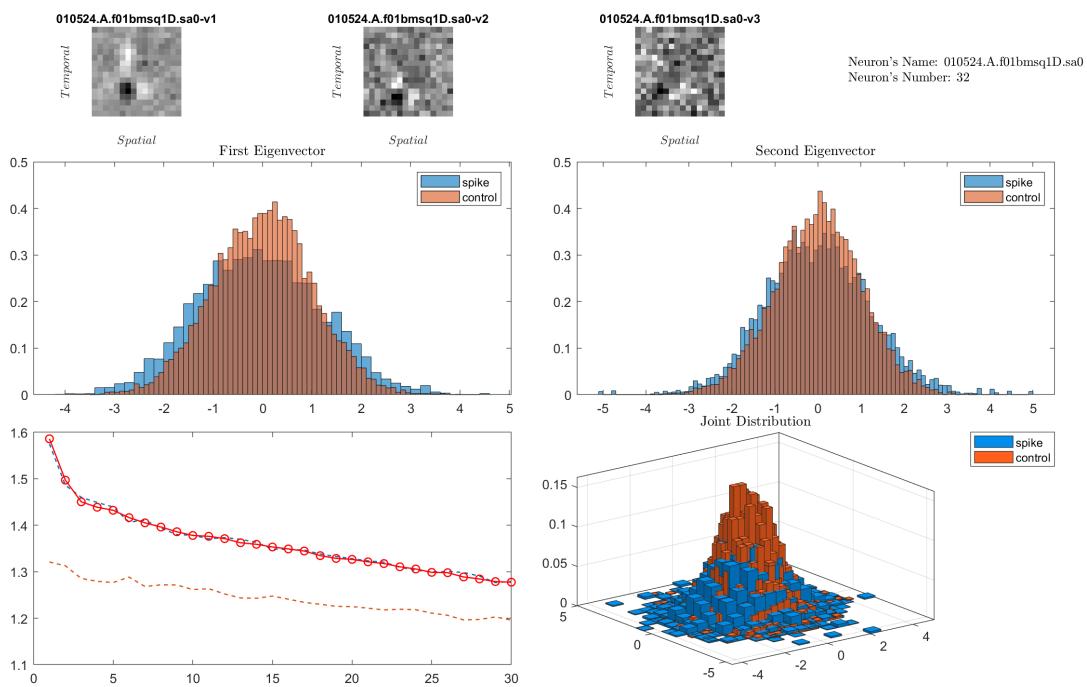
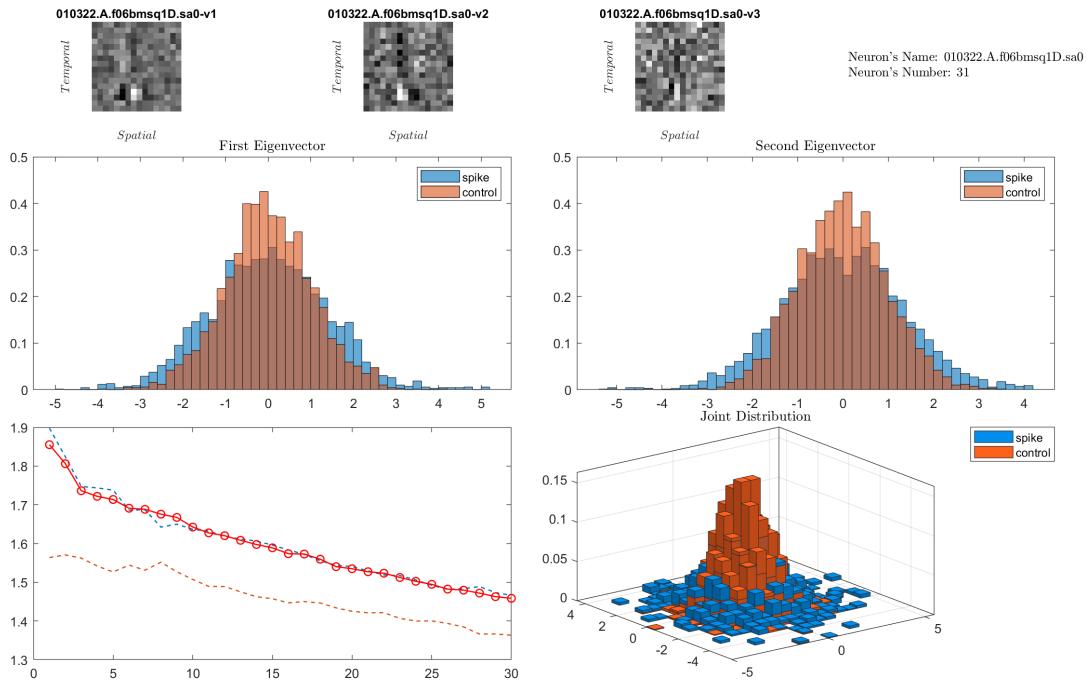


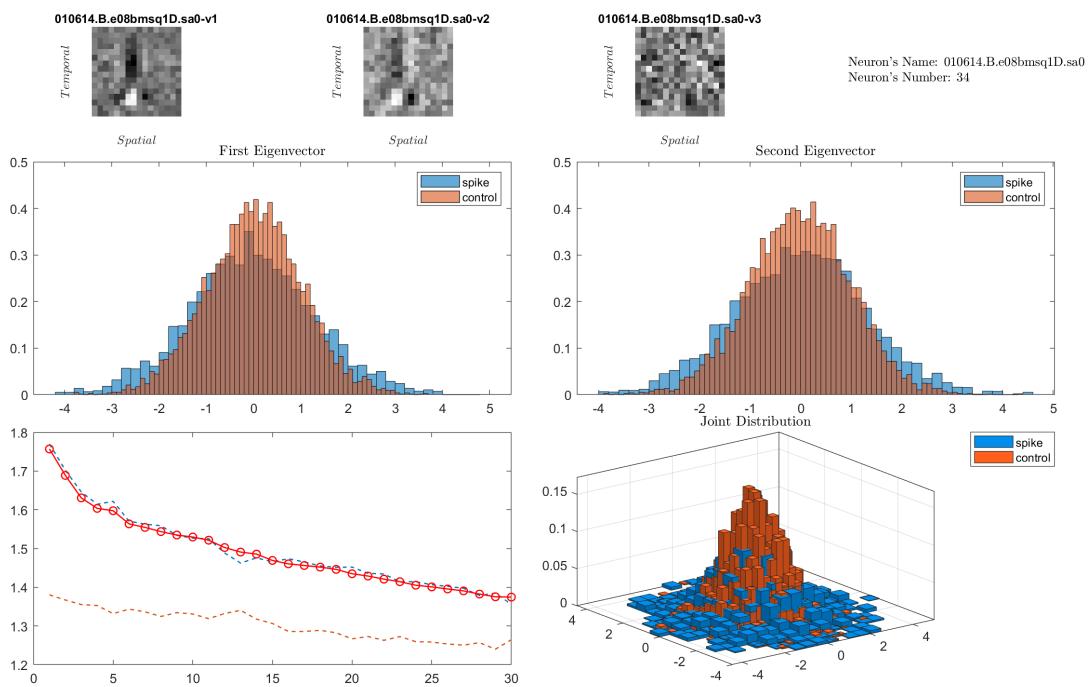
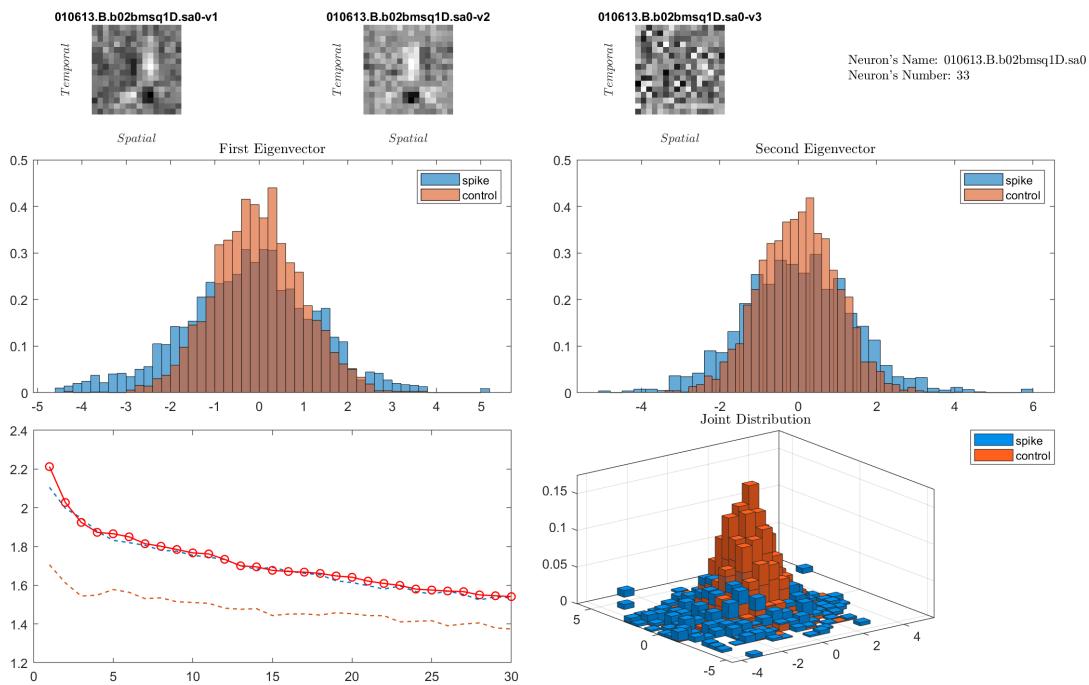


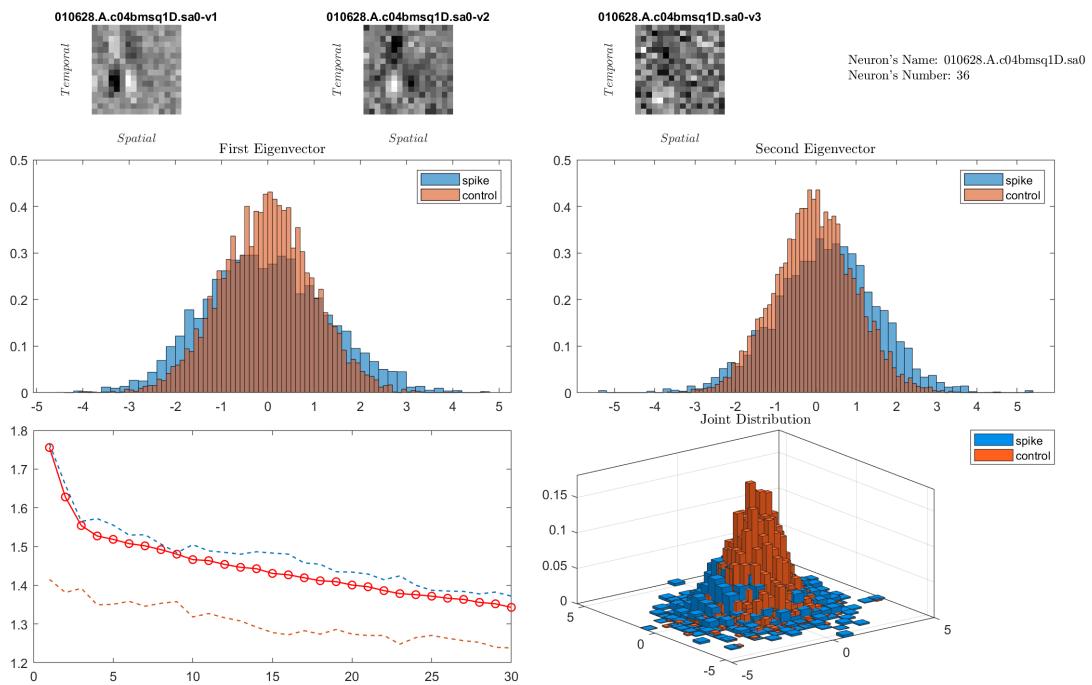
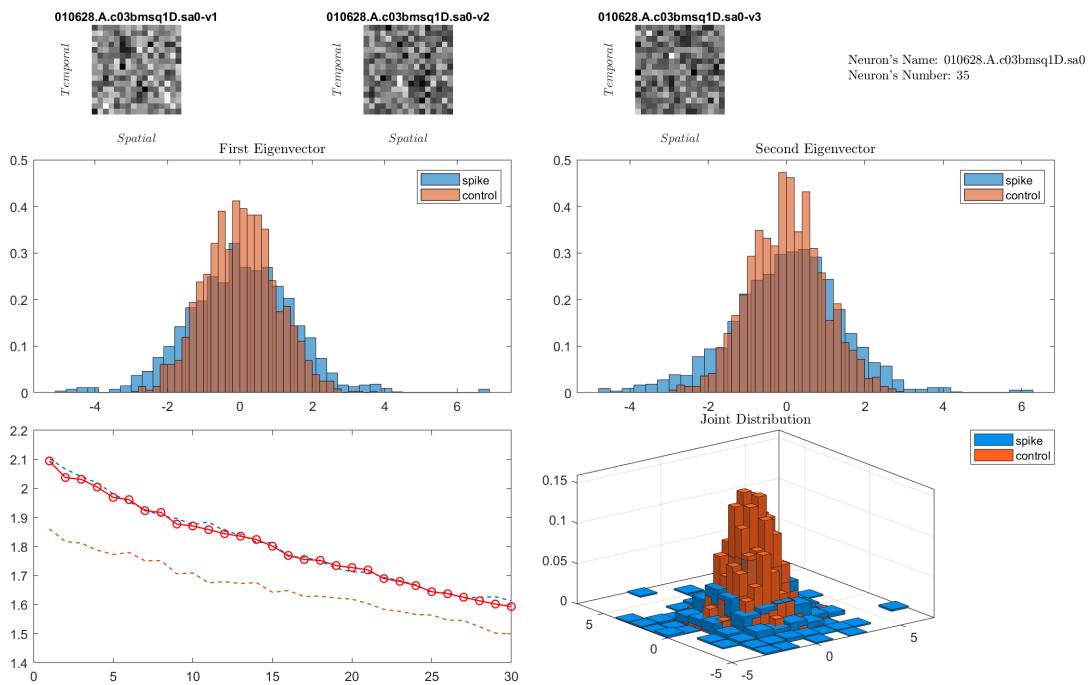


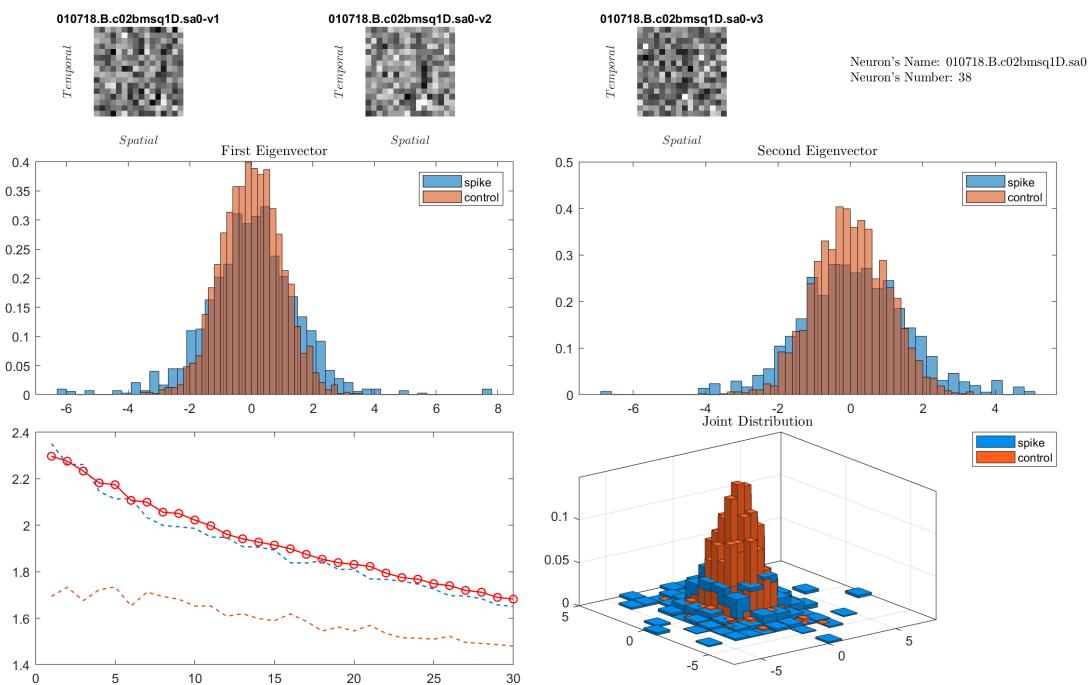
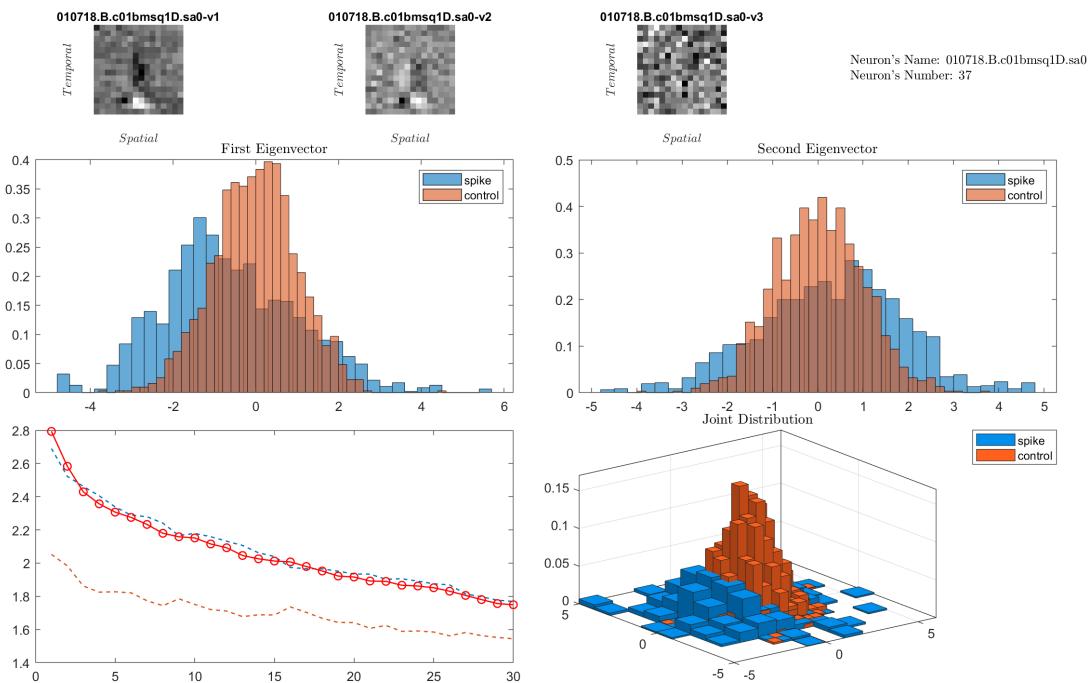


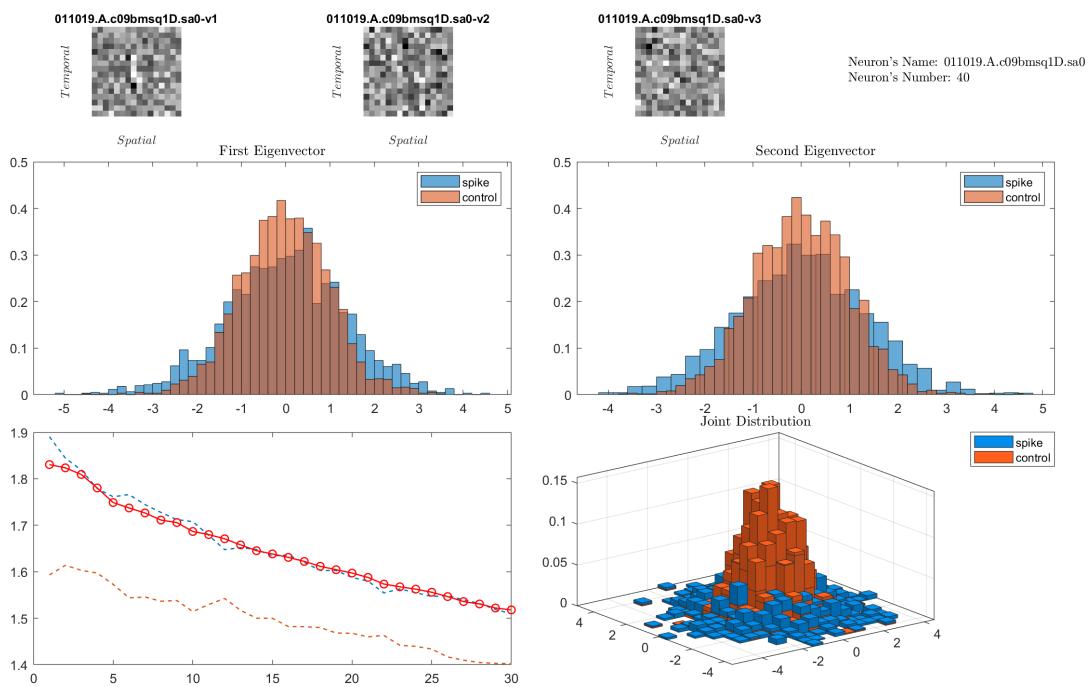
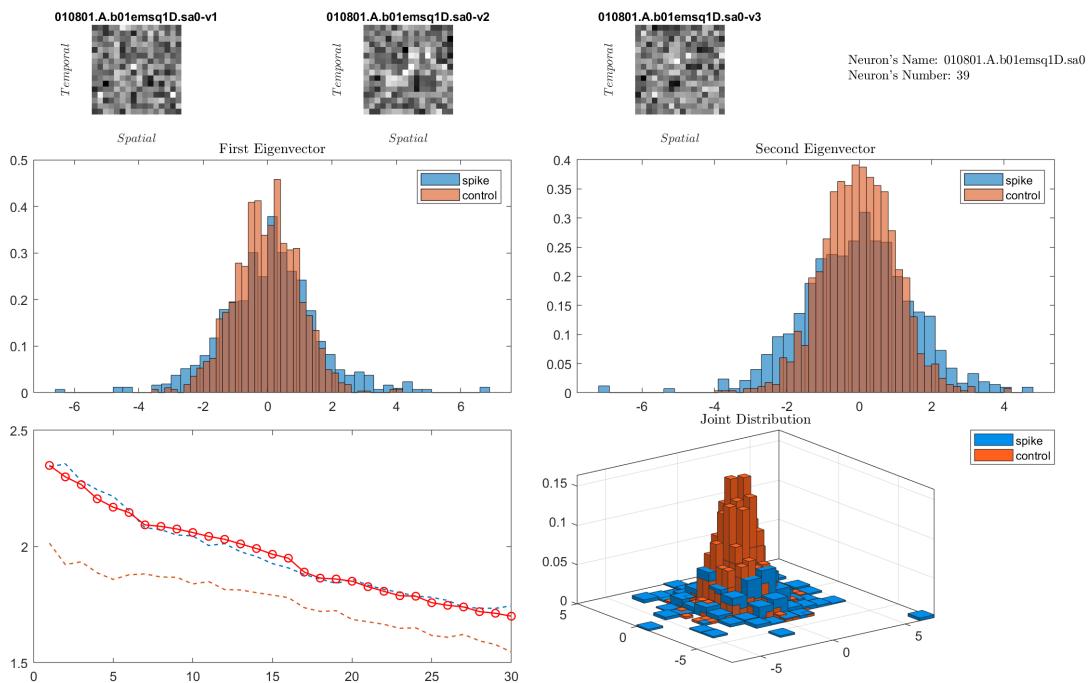


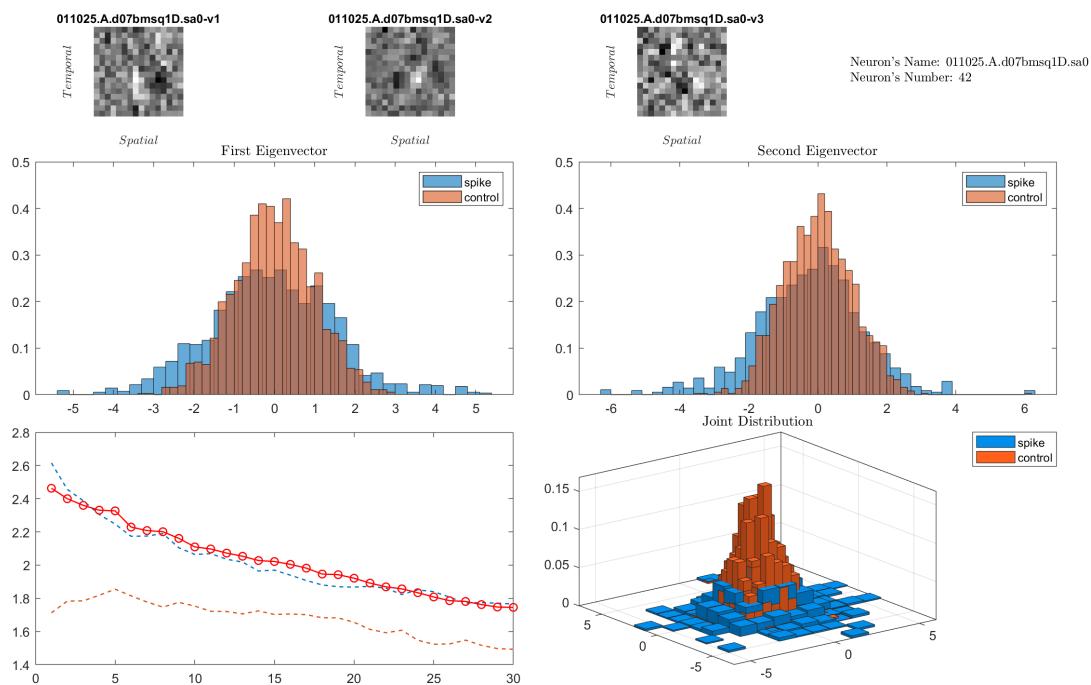
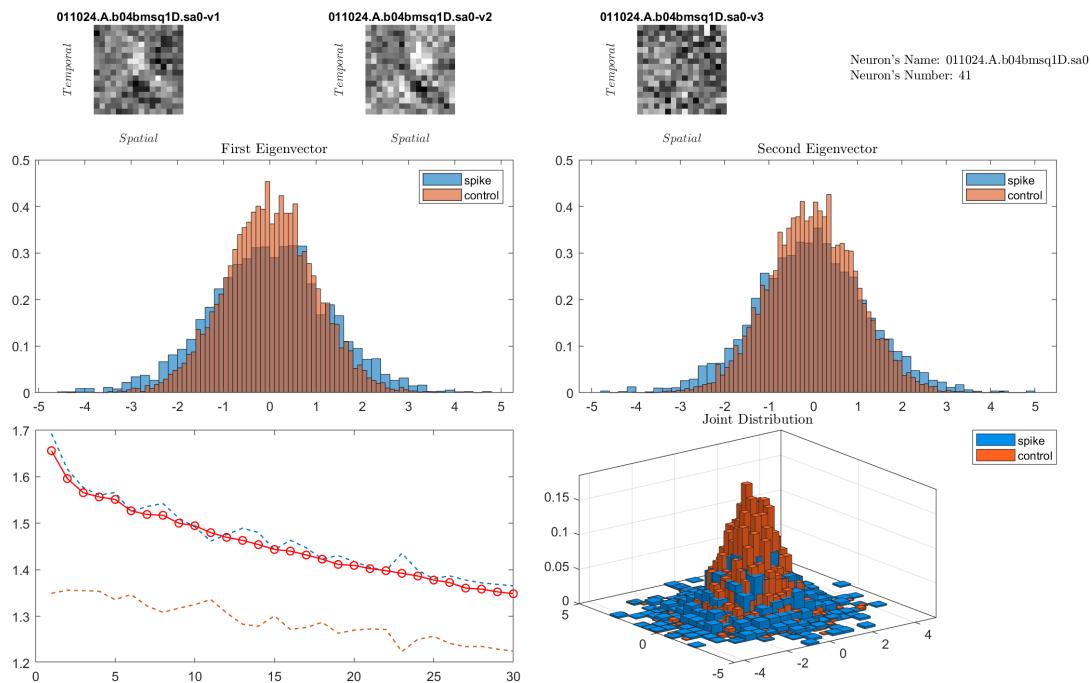


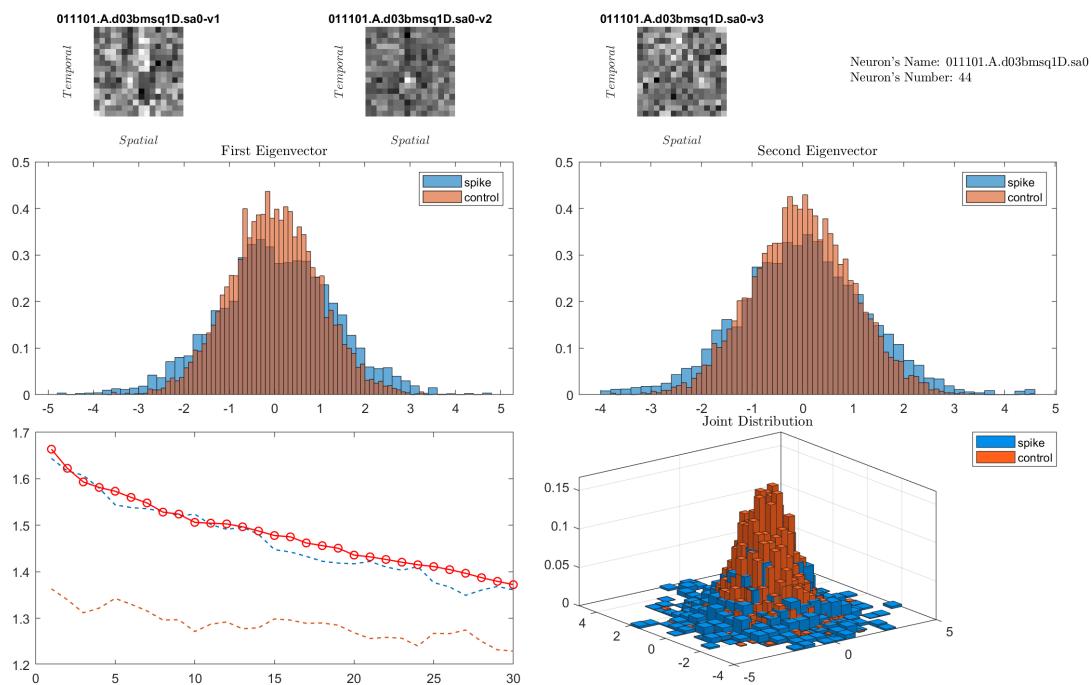
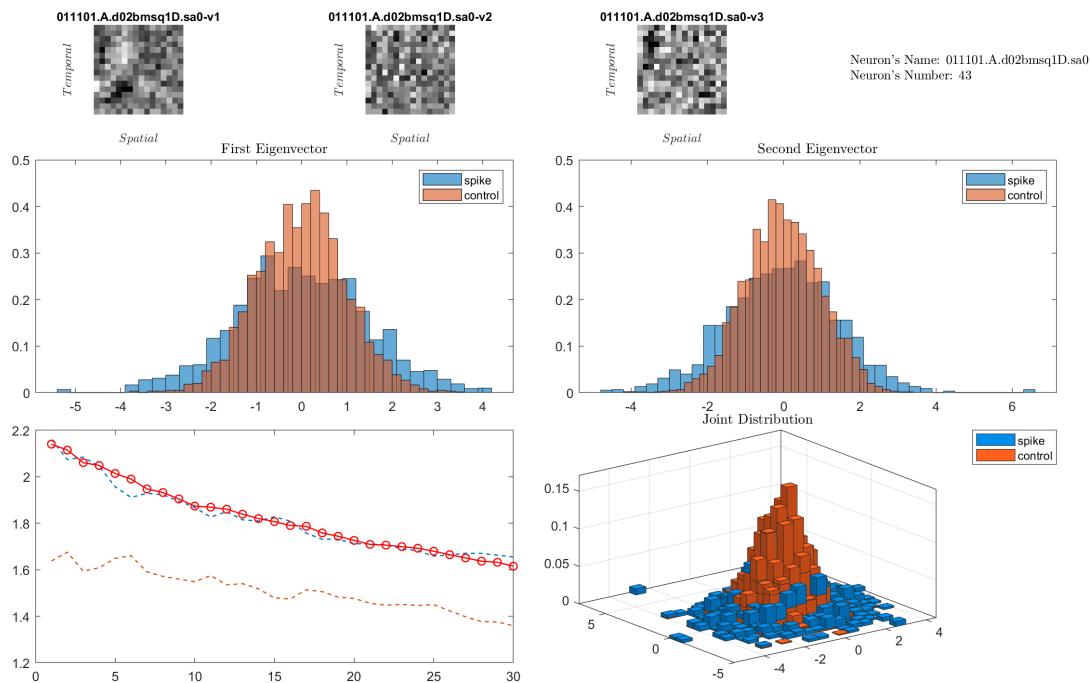


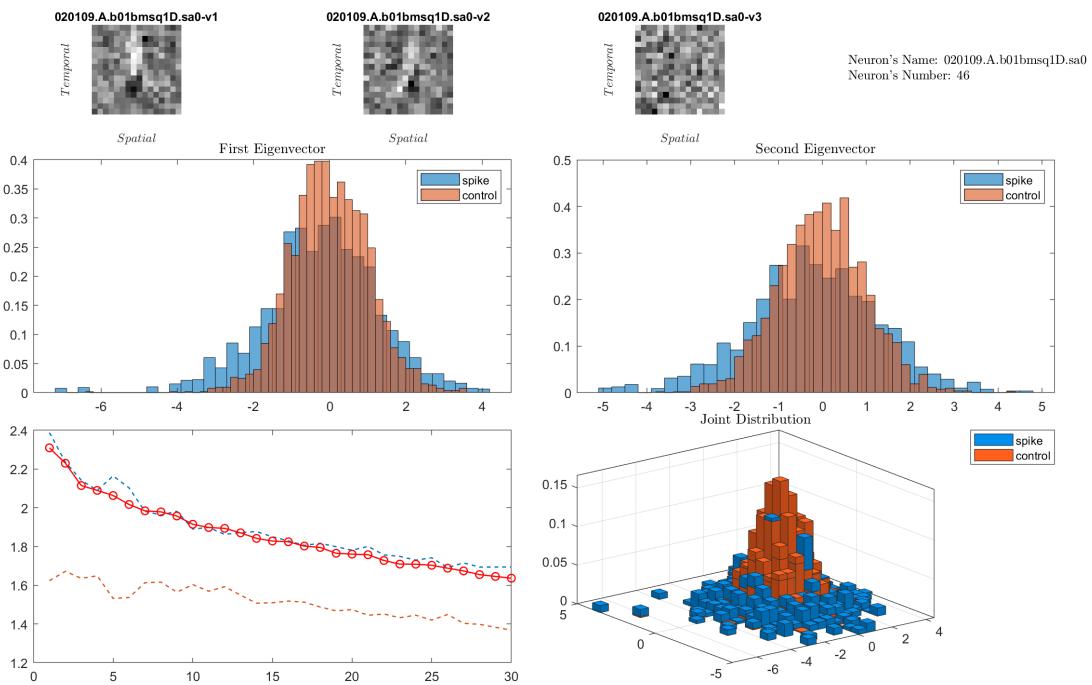
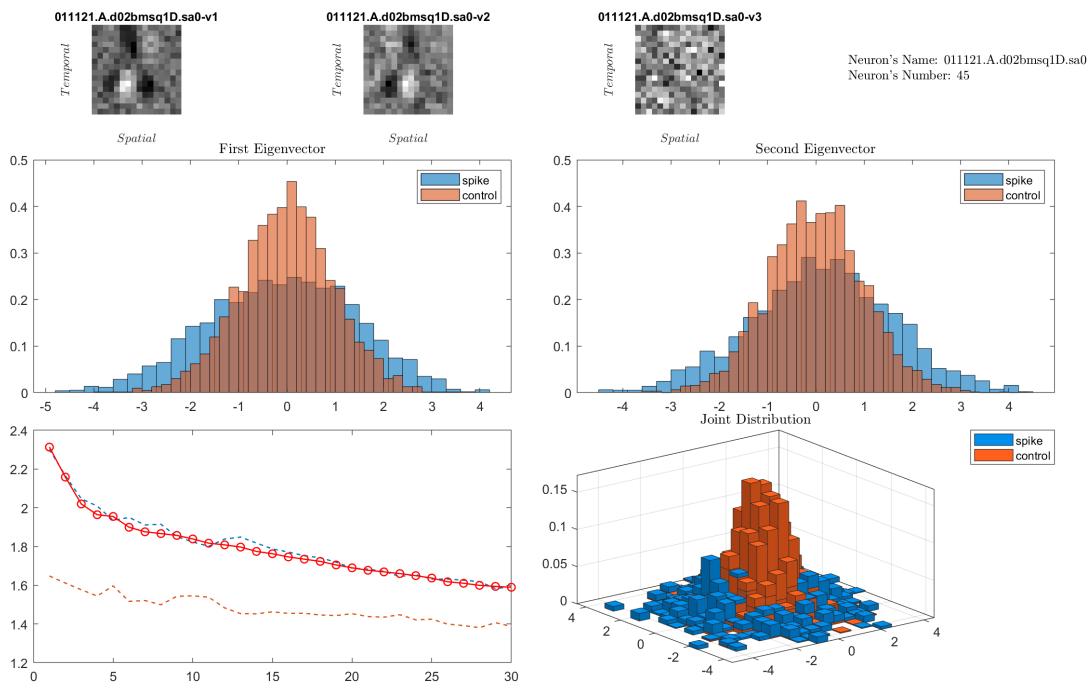


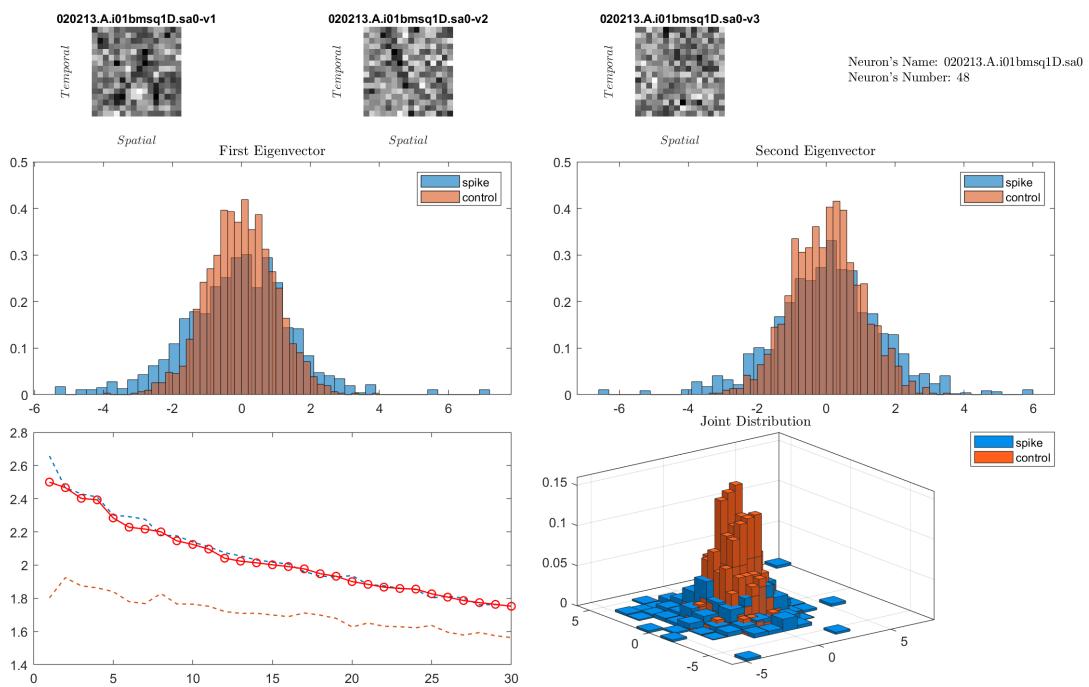
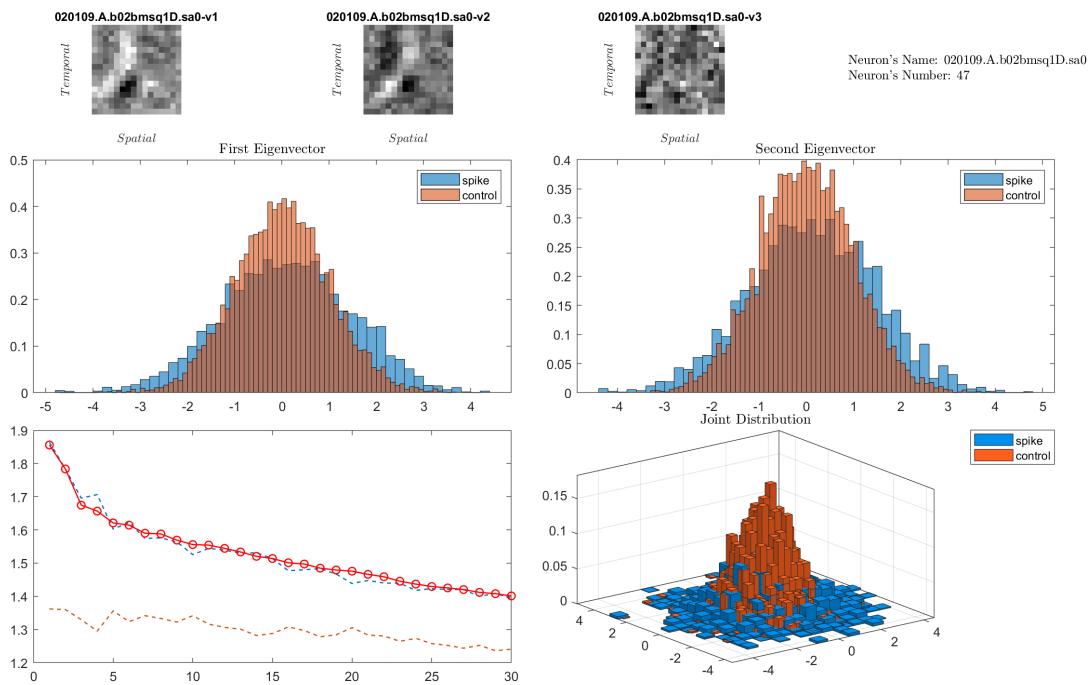


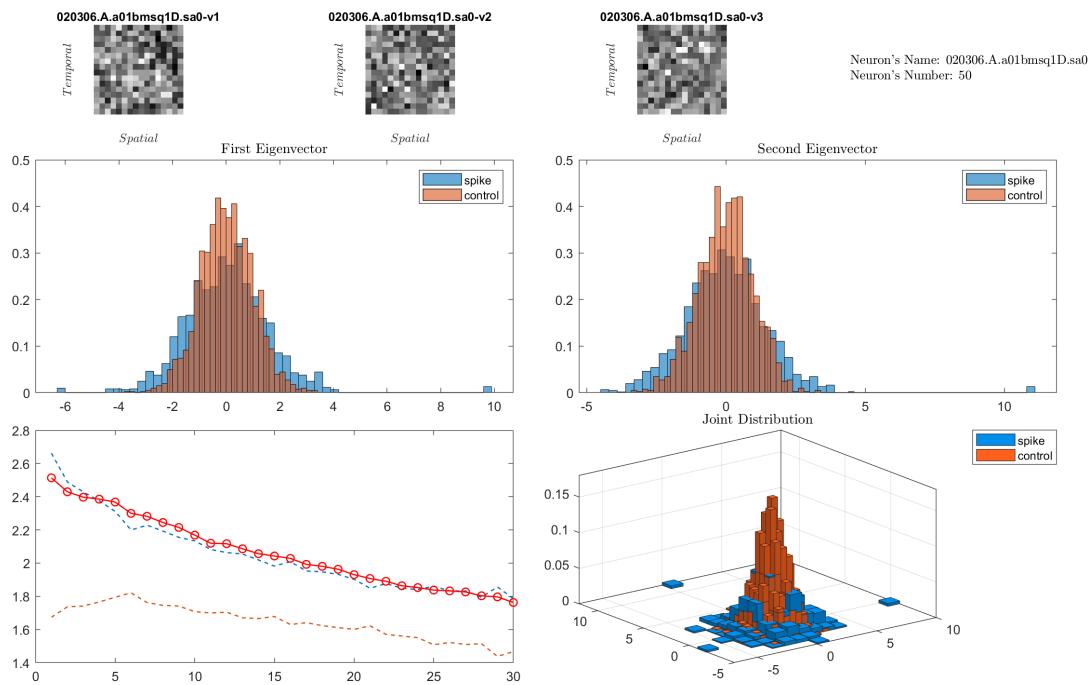
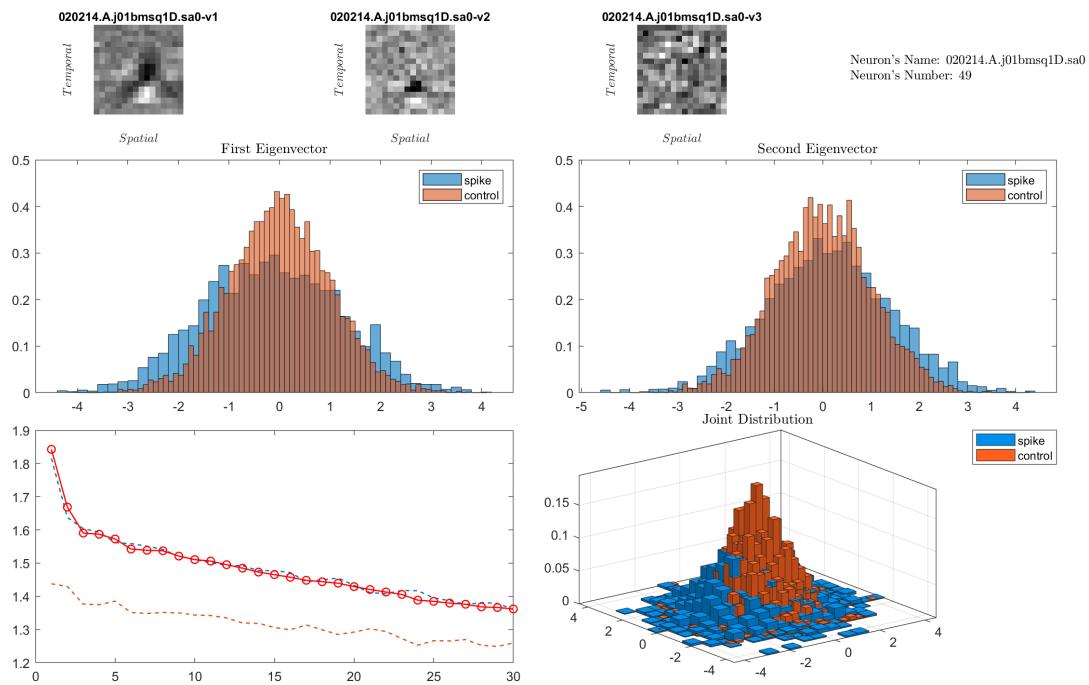


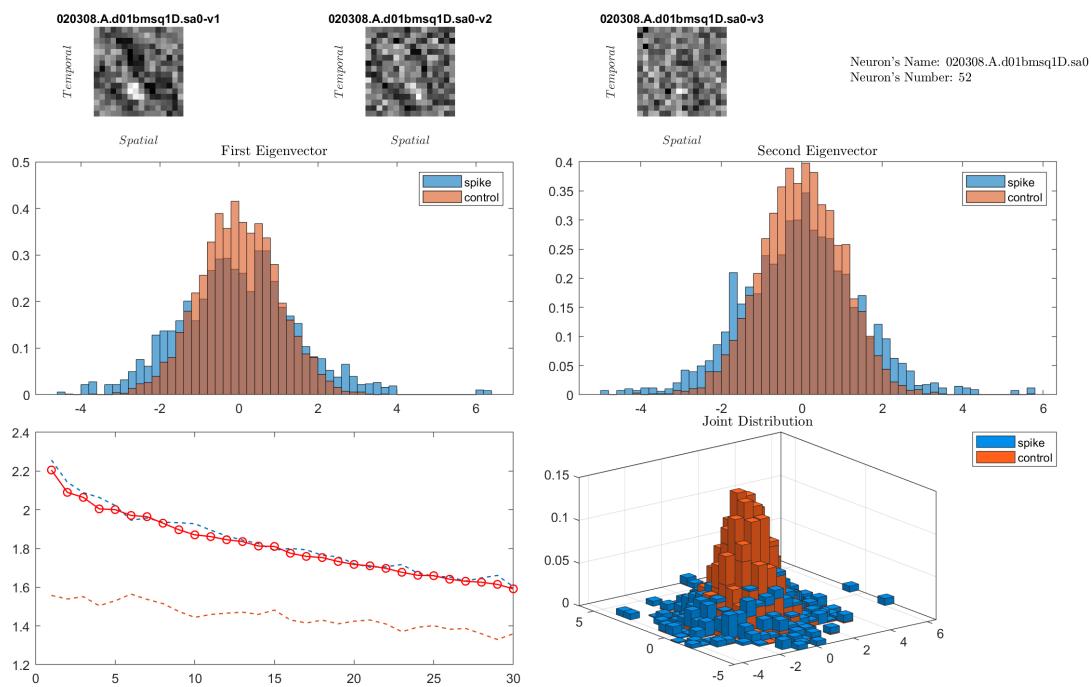
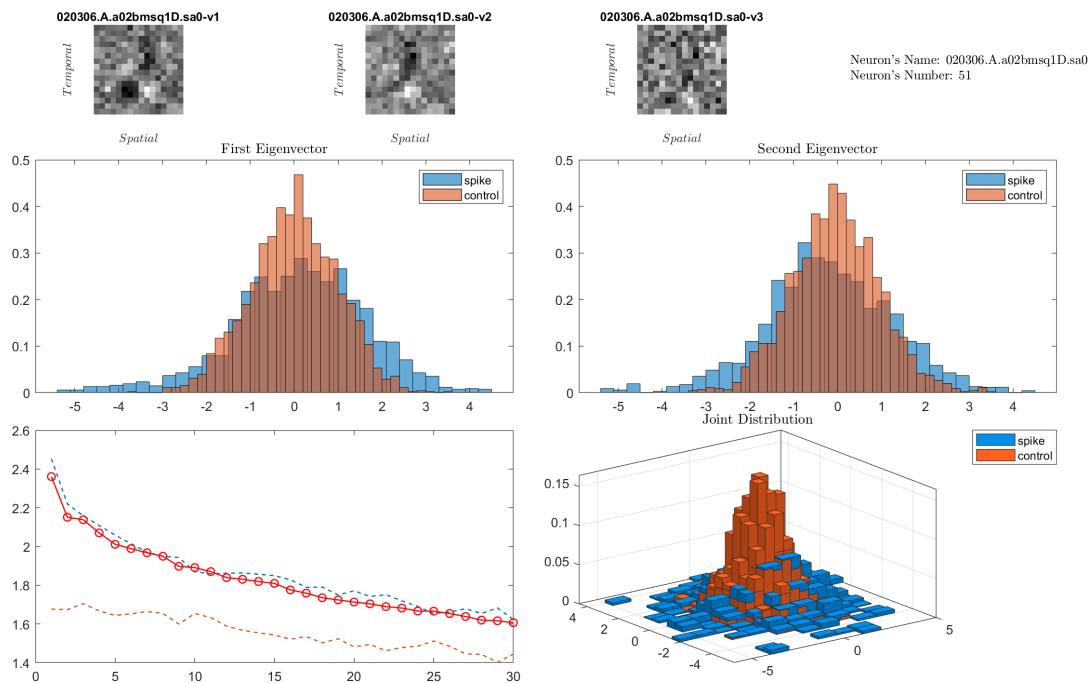


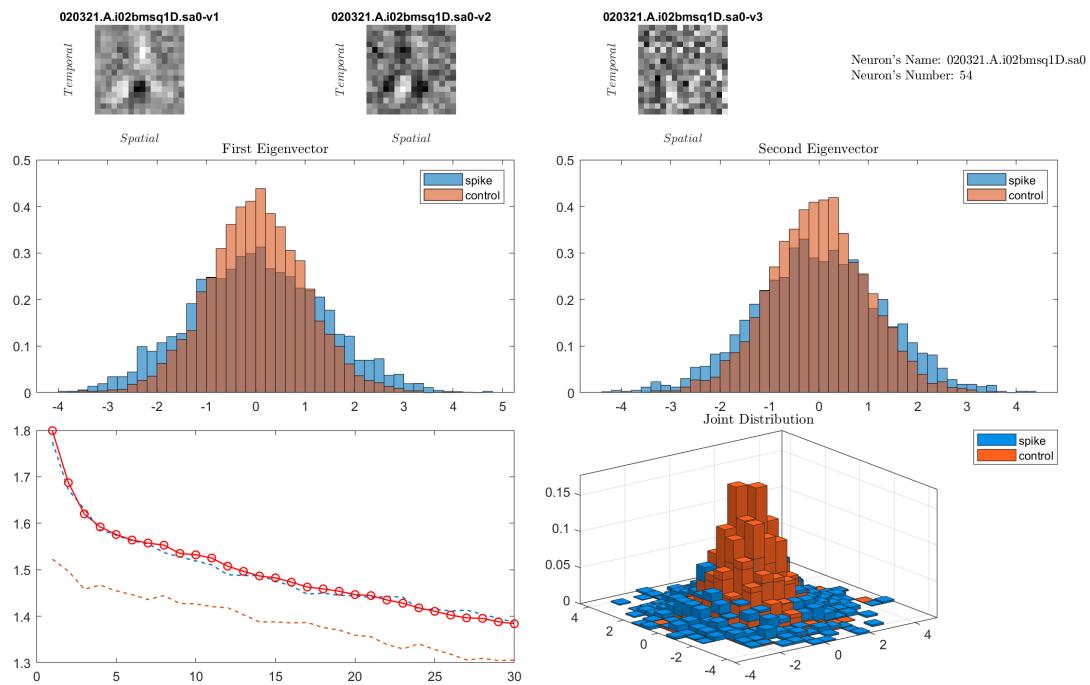
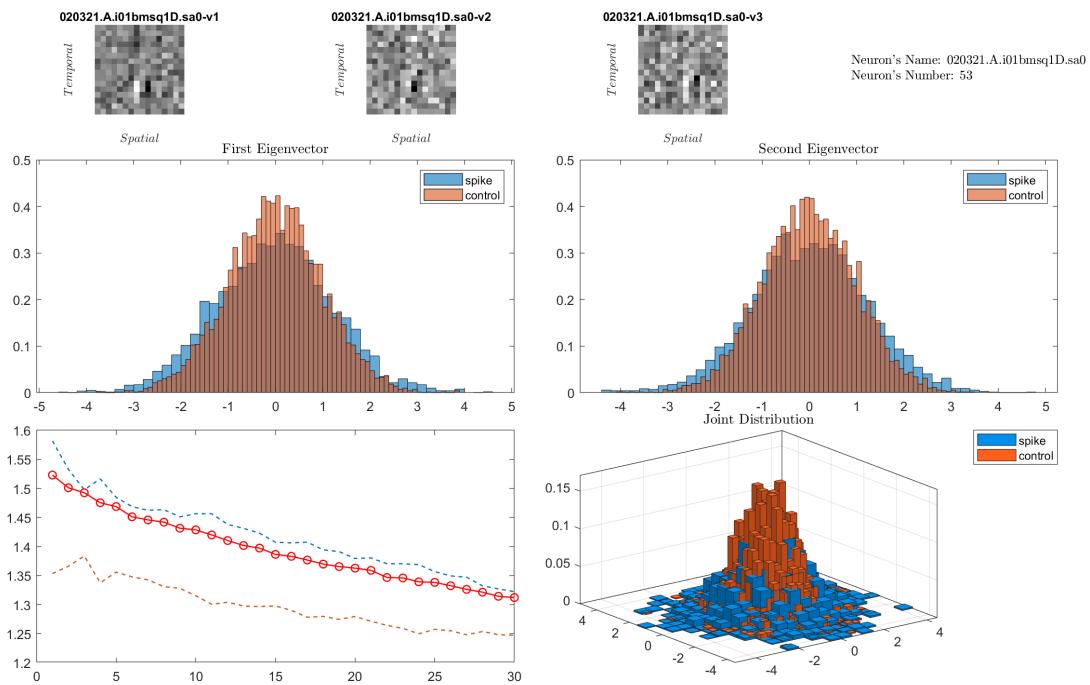












٧٤ بخش هفتم

با بررسی جدول قسمت قبل، می‌بینیم که تقریباً تمامی بردار ویژه‌های اول، عمدۀ بردار ویژه‌های دوم و اندکی از بردار ویژه‌های سوم، دارای پترن‌های معنی‌دار هستند. این می‌تواند در مقایسه با STA یک پیشرفت محسوب شود، چرا که در STA، بسیاری از receptive field های بدست آمده، قادر پترن معنی‌دار بودند. اما سؤال مهم‌تر این است که آیا در تمامی موارد توانسته‌ایم بهتر از روش کلاسیک عمل کنیم یا نه؟ ابتدا به درصد‌های موجود در جدول بخش قبل مراجعه می‌کنیم. آنچه در جدول می‌بینیم کاملاً خلاف انتظار ماست. اولاً که تعداد زیادی درصد کمتر از ۵۰ مشاهده می‌کنیم (که در روش کلاسیک چنین نبود) و این به آن معناست که سنجش ما دارد ضعیفتر از حالت رندوم عمل می‌کند. یعنی اگر تحریک‌ها را به صورت تصادفی به دو دسته موثر و بیاشر تقسیم می‌کردیم، نتیجه‌ی بهتری از این دسته‌بندی می‌گرفتیم. البته در این گونه مسائل دسته‌بندی، نتیجه‌ی بسیار بد، مانند روی دیگری از یک سکه است که در طرف دیگر آن، نتیجه‌ی بسیار خوب قرار دارد. یعنی اگر به عنوان مثال، ما در ۲۰ درصد موارد بتوانیم جواب درست را تشخیص دهیم (با توجه به این که جواب، دو حالت دارد؛ اسپایک زدن یا نزدن) می‌توانیم صرفاً جوابمان را در تمامی حالات عوض کنیم. به این ترتیب، در ۸۰ درصد موارد، جواب درست می‌دهیم. با این وجود، ما انتظار داریم که پس از حجم زیادی از تحلیل و محاسبات، درصد تحریک‌هایی که به درستی تشخیص داده می‌شوند، با اختلاف بالای ۵ باشند که در عمل، مشاهده کردیم که چنین نشد. در واقع، دیدیم که در بسیاری از موارد گویی STA بهتر پاسخ می‌دهد تا STC (Spike-Triggered Correlation).

۵ طرح سوال دلخواه

۱۰.۵ صورت مسئله‌ی طرح شده

با خواندن مقاله متوجه می‌شویم که هدف آن یافتن راستاهاییست که در آن داده‌های موجب اسپایک، واریانس به طور قابل توجه زیادتری نسبت به کل داده‌ها داشته باشد و در نتیجه توزیع داده‌های موجب اسپایک در آن راستا با توزیع کل داده‌ها متفاوت باشد. در طول انجام این پژوهه به این موضوع اشاره نشد.

در اینجا می‌خواهیم این موضوع را مشاهده کنیم یعنی اولاً ببینیم که در راستاهای پیشنهادی مقالی واریانس داده‌ها به طرز قابل توجهی بیشتر از کل داده‌های است و ببینیم که در روش کلاسیک STA این موضوع برقرار نیست. ثانیاً با بررسی بردارویژه‌های مرتب شده نوع تغییر p-value در تست واریانس را ببینیم.

۲۰.۵ تست مورد استفاده

برای بررسی موضوعات گفته شده به یک تست آماری نیاز داریم که بزرگ‌تر بودن واریانس یک نمونه نسبت به یک نمونه‌ی دیگر را بررسی کند. با جستجو برای این هدف به تست f-test دست پیدا کردیم که تابع آماده آن موجود است به طور کلی f-test برای دو جمعیت استفاده می‌شود تا ببینیم آیا این دو جمعیت واریانس یکسانی دارند یا نه. ما در اینجا می‌خواهیم از آزمون یک-طرفه استفاده کنیم و تنها بزرگ‌تر بودن واریانس جمعیت اول (توزیع تحریک‌ها به شرط اسپایک) را بررسی کنیم بنابراین:

- $H_0 : \sigma_1^2 = \sigma_2^2$

- $H_a : \sigma_1^2 > \sigma_2^2$

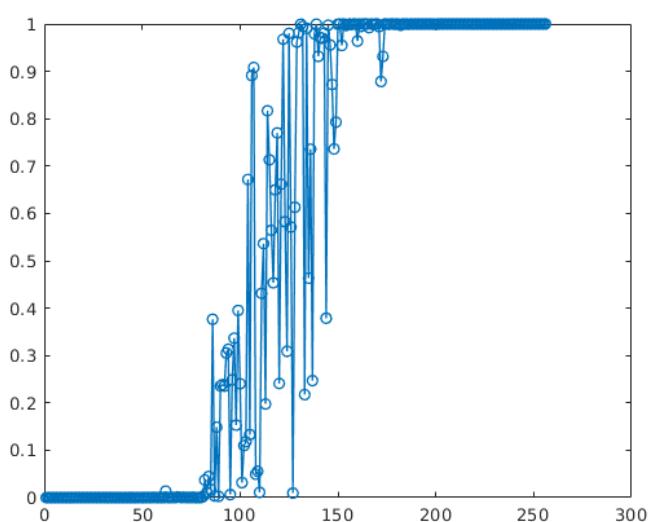
۳۰.۵ رویکرد و نتایج به دست آمده

رویکردی که در پیش می‌گیریم این است که ابتدا تصویر توزیع‌ها در راستای STA و بردارویژه‌ی اصلی بدست آمده در روش STC را ملاک قرار می‌دهیم و برای تمامی نورون‌ها آزمایش f-test را انجام می‌دهیم نهایتاً می‌خواهیم درصد رد شدن فرض صفر به ازای همهٔ نورون‌ها و برای هر رویکرد بررسی کنیم. کلیت کار مشابه قبل خواهد بود و با پیمایش بر روی تمام نورون‌ها correlation تنها تستی که می‌گیریم متفاوت است. خروجی به صورت زیر خواهد بود:

```
mean result for Spike triggered correlation:1
mean result for Spike triggered correlation:6.481481e-01
```

مشاهده می‌کنیم که برای STC با درصد بسیار خوبی ۱۰۰ درصد تمامی فرض صفرها را رد کردیم به عبارت دیگر برای تمامی نورون‌ها توزیع تحریک به شرط وجود اسپایک با توزیع تحریک در راستای مقدارویژه‌ی اصلی که برای ماتریس STC به دست می‌آوریم به لحاظ واریانس اختلاف قابل توجهی دارد که این موضوع برای STA صدق نمی‌کند.

در بخش بعد برای نورون با کد 020321.A.i و برای تمامی بردارویژه‌هایی که از روش STC به دست آوردیم مراحل تست را تکرار می‌کنیم، اینجا پرسش ما این است که آیا p-value-ها ادعای مقاله مبنی بر بی‌اثر بودن تعداد زیاد از ویژگی‌ها را تایید می‌کند یا خیر. خروجی به دست آمده از اجرای کد نمودار زیر خواهد بود.



شکل ۱۱ : p-value ها برای بردارویژه‌های بزرگ‌تر تا کوچک‌تر

همانطور که می‌توان دید برای بسیاری از نورون‌ها p-value مقدار یک را دارد که این نشان می‌دهد آن بردارویژه‌ها با احتمال بسیار بالایی به هیچ‌وجه در تغییر واریانس و در نتیجه تغییر توزیع قبل و پس از دیدن پاسخ نورون ندارند.