

March 29, 2019

Bursting with Curiosity: Combined Analysis of Behavioral and Neural Imaging Data in Mice Performing the Object Space Task.

Sebastian Tiesmeyer, s4373162

Supervisors:

Francesco Battaglia, Ph.D.

Katja Seeliger, M.Sc.

Luca Ambrogioni, M.Sc.

sebastian.tiesmeyer@student.ru.nl

Radboud University Nijmegen



Donders Institute
for Brain, Cognition and Behaviour



Donders Institute
Centre for Cognitive Neuroimaging
Kapittelweg 29, Nijmegen
NL-6525 EN
The Netherlands
<http://www.ru.nl/donders>

I Abstract

The object space task investigates memory consolidation in rodents. The task involves a series of training sessions during which animals are made to explore an arena that contains a number of objects placed according to a number of repositioning rules. Presuming an innate curiosity in the rodents ('novelty bias'), one can use the exploratory behavior towards the objects as a measure of internalization of the object repositioning rules.

This paper presents an integrated pipeline for the combined analysis of behavioral videos and synchronized Ca^{2+} -imaging footage of animals performing the object space task. It includes a streamlined version of the CNMF-E Ca^{2+} -signal source extraction algorithm by [Pnevmatikakis et al. \(2016\)](#), a deep learning-based behavioral video classifier and a number of supervised and unsupervised statistical tests to investigate the neural correlates of memory formation.

The basic functioning of the pipeline could be confirmed using a number of random permutation linear regression fits of the neural and behavioral data points. Moreover, statistical tests could be used for a first exploratory investigation of the information content of the neural signal. Cells with correlated activity could be shown to encode certain types of abstract behavior like roaming, hiding or object exploration. Also, the neural signal could be shown to carry some intentional information when comparing the predictive power of the current signal for future and past behavioral states. Unfortunately, the amount of data analyzed so far is not sufficient to draw any final conclusions on neural ensemble formation during the object space task itself. However, large dataset of Ca^{2+} -imaging recordings during the objects space task is currently available at the group and ready to be analyzed, with the potential to produce valuable insights on the neural correlate of memory consolidation.

II Declaration

I hereby declare that I am the sole author of this MSc thesis and that I have not used any sources other than those listed in the bibliography and identified as references. I further declare that I have not submitted this thesis at any other institution in order to obtain a degree.



March 29, 2019

signature

date

III Acknowledgements

First of all, I want to thank my parents and family for the wonderful and continuous long-distance support they gave me during my five years of study in the Netherlands - ranging from frequent telephonic counsel to an eagerly awaited annual supply of homemade Christmas cookies.

Further thanks are due to the heads of the Donders Memory Dynamics group, Francesco Battaglia and Lisa Genzel, who developed the idea for this project and were a continuous source of help and inspiration throughout.

I want to thank the Donders Department for Neuroinformatics and the Memory Dynamics group that created a wonderful academic environment and provided great technical, academic, emotional and culinary support during the entire project. Special thanks go to Evelien Schut for establishing the Ca^{2+} -imaging infrastructure at the Group and assembling the high-quality dataset used for this study and to Ronny Eichler for setting up and maintaining the computational hardware and for his frequent advice on code design.

Last but not least, I want to thank my internal supervisor Katja Seeliger second reader Luca Ambrogioni.

Contents

I	Abstract	i
II	Declaration	ii
III	Acknowledgements	iii
1	Introduction	1
2	Ca^{2+}-Imaging Pipeline	5
1	Introduction	5
2	Preprocessing	6
3	Motion Correction	7
3.1	Rigid Transform	8
3.2	Non-Rigid Transform	9
4	Trial Alignments	10
4.1	Preprocessing	10
4.2	PCA-ICA source extraction	10
4.3	Alignment Algorithm	11
5	Source Extraction:	12
5.1	The Factorization Model	12
5.2	Variable Initialization	14
5.3	Factorization	14
6	Results	15
3	Behavioral Analysis	19
1	Introduction	19
2	Pose Estimation	20
3	Behavioral Classification	22
4	Data Integration	25
5	Results	27
6	Conclusion	28

4 Statistical Analysis	31
1 Introduction	31
2 Supervised methods	31
2.1 Regression Analysis	31
3 Unsupervised Methods	34
3.1 Introduction	34
3.2 Methods	35
3.3 Variable NMF: A Simulation	40
4 Tensor Decomposition	41
5 The Object Space Task	45
1 Introduction	45
2 Methods	46
3 Results	48
3.1 Ca^{2+} -imaging analysis	48
3.2 Behavior	49
3.3 Supervised Statistics	52
3.4 Unsupervised Statistics	55
3.5 Tensor Decomposition	58
4 Conclusion	59
6 Conclusion	64

Chapter 1

Introduction

Ca^{2+} -imaging is a powerful technique to investigate local neural networks in the living mammalian brain. Its is now in wide use with established protocols for the physiological and functional analysis of both single neurons and neural ensembles (Grienberger, Konnerth, 2012). Technically, Ca^{2+} -imaging is a fluorescence recording technique that uses the photon re-emission of calcium-sensitive molecules to determine the Ca^{2+} -concentration inside the cell (figure 1.1). As such, it allows visual tracking of individual neurons over a period of several weeks and can be used to trace the intracellular Ca^{2+} concentration as a rich indicator for a wide range of Ca^{2+} -dependent neural processes Grienberger, Konnerth (2012): Most importantly, depolarization at the synaptic terminal induces calcium influx, which, in excitatory neurons, initiates the release of neurotransmitters into the synaptic cleft. In the post synapse, an increased Ca^{2+} concentration can induce *activity-dependent synaptic plasticity* (Zucker, 1999), which is known to play a role in memory formation. Next to that, Ca^{2+} -ions regulate gene transcription in virtually every cell type in the body.

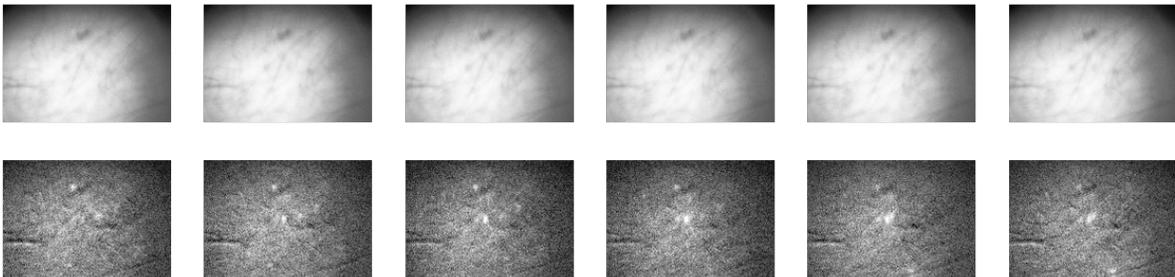


Fig. 1.1: Six consecutive frames of Ca^{2+} -imaging footage recorded at $10Hz$. The top row shows the raw footage, the bottom row shows the same frames as derivation from the movie's median frame.

This project aims to explore the potential of Ca^{2+} -imaging for a newly developed memory consolidation experiment for rodents called *the object space task* (Genzel et al., 2017). The OST makes use of the rodents' natural curiosity and tendency to explore novel objects and situations in their environment. During the task, animals are repeatedly exposed to an arena containing two objects that are repositioned across trials. The exploration behaviour is used as an indicator for short-term 'episodic' memory or long-term 'semantic' memory formation. One (week-long) session consisted of 21 five-minute exploration trials during which both the neural Ca^{2+} signal and the animal behavior were recorded. After each trial, another 5 minutes of Ca^{2+} signal was recorded while the animal was in a resting cage.

Single-photon neural Ca^{2+} -imaging suggest itself as an investigative method for this task since it allows the tracking of single cells over entire, week-long periods of time while the animal can behave freely in a given environment. Also, as mentioned above, intracellular Ca^{2+} has been linked to activity-dependant plasticity and memory formation. Accordingly, this project interprets the raw calcium signal as a simple indicator for momentary cell activity with a potential effect on memory consolidation.

The OST experiments produced an extensive dataset of Ca^{2+} -imaging footage and a body of synchronized behavioral videos captured from a camera above the arena. The main aim of this pilot project was therefore to construct an integrated automated analysis pipeline (1.2) for the collected neural and behavioral data. Initially, both datasets were processed in their proper analysis pipelines to retrieve the relevant information: The Ca^{2+} -imaging footage was motion corrected, aligned and passed to a factorization algorithm to extract neural activity traces. The behavioral video data was processed using a deep learning ensemble to determine the animal's position and exploratory behavior.

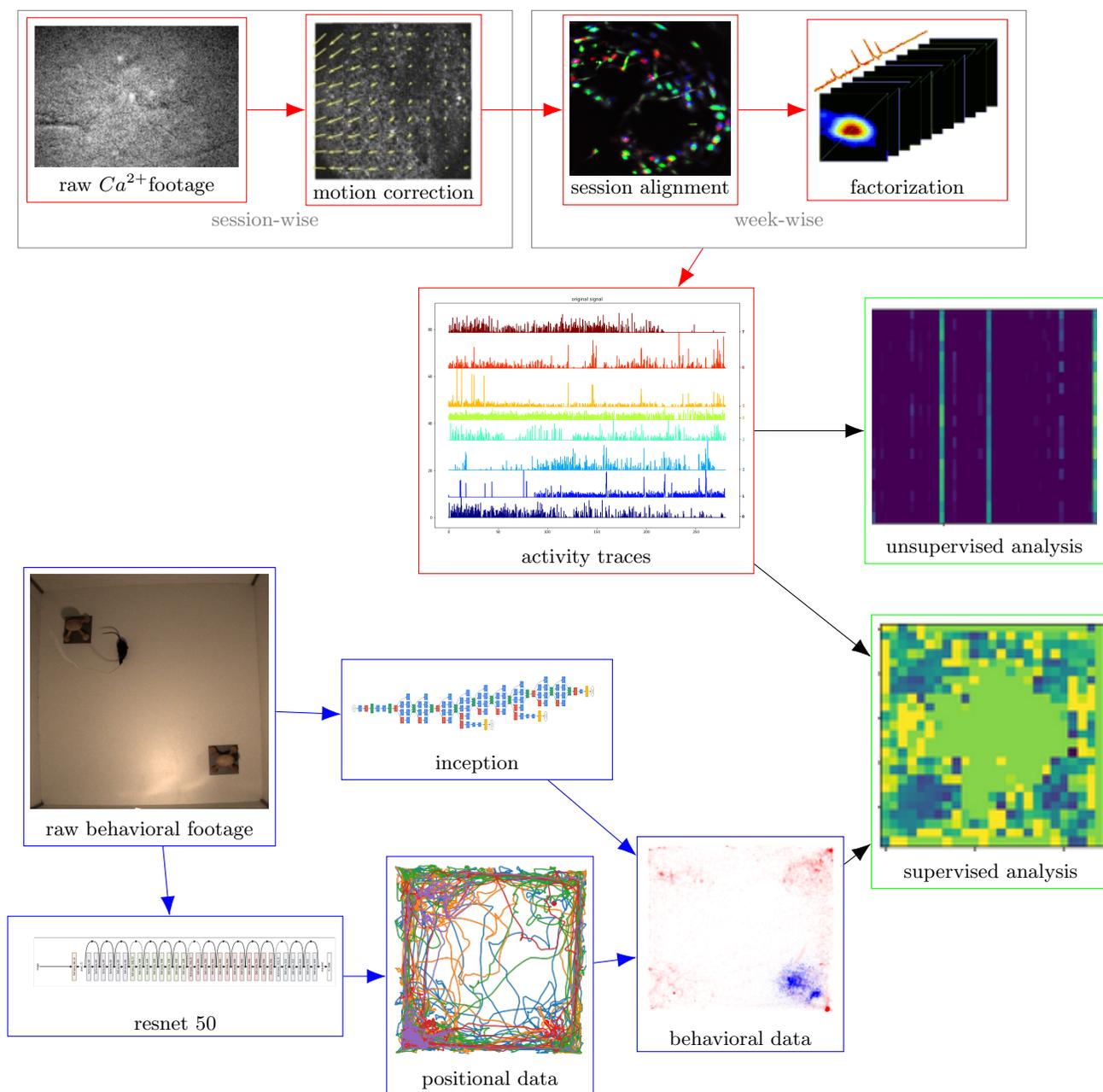


Fig. 1.2: A general overview of the project: The main segments are a Ca^{2+} -imaging pipeline, a behavioral pipeline and a sequence of statistical tests. The Ca^{2+} -imaging pipeline extracts neural activity traces from raw Ca^{2+} -imaging footage. The behavioral pipeline extracts behavioral variables from raw behavioral videos. The output of both pipelines is then passed to a number of supervised and unsupervised statistical tests.

Next to the behavioral and neural analysis, the last major part of the project was the creation of a range of supervised and unsupervised statistical tests to assess the quality of the extraction pipelines and to preliminary explore the information content of the extracted signals.

In the remaining part of the thesis, the different pipelines and their individual components will be discussed in detail. Chapter 2 will discuss the Ca^{2+} -imaging pipeline. Chapter 3 will treat the deep learning ensemble of the behavioral pipeline. Chapter 4 treats the different evaluating and exploratory statistical tests implemented as part of this project. The final chapter 5 will present the results of this pipeline when actually applied to a subset of the OST dataset.

Chapter 2

Ca^{2+} -Imaging Pipeline

1 Introduction

The construction of an extended Ca^{2+} -imaging analysis pipeline was the core objective of this project. The pipeline was to be used to analyze Ca^{2+} -imaging videos recorded in mice performing the OST.

There already exists a considerable body of research on computational Ca^{2+} -imaging analysis. State-of-the-art algorithms typically apply a form of dimensionality reduction on the video frames to retrieve sources with high pixel intensity correlations. Often, a combination of principal component analysis and independent component analysis (PCA/ICA) is used since there exist very efficient and well-proven implementations that easily scale up to large data sets. However, PCA/ICA is a very self-contained method that does not incorporate additional, problem-specific constraints very well, which is why we decided to use the more computationally intensive method of *constrained non-negative matrix factorization (CNMF)* for this project. CNMF is a very versatile method of dimensionality reduction that can easily be customized by including problem-specific constraints. Also, the output of the CNMF algorithm is easier to interpret in terms of neurophysiological data since it is naturally sparse and strictly non-negative.

A single execution of the OST protocol produced a total of 42 five-minute videos captured over five consecutive days. To be able to do an effective neurophysiological analysis, it was crucial for the pipeline to integrate all 42 recordings into one conjoined data vector. Therefore, the first component of the pipeline was a pre-processing step that reformatted the data and reduced the size of the dataset for more efficient computation. Since the natural movement of the camera interferes

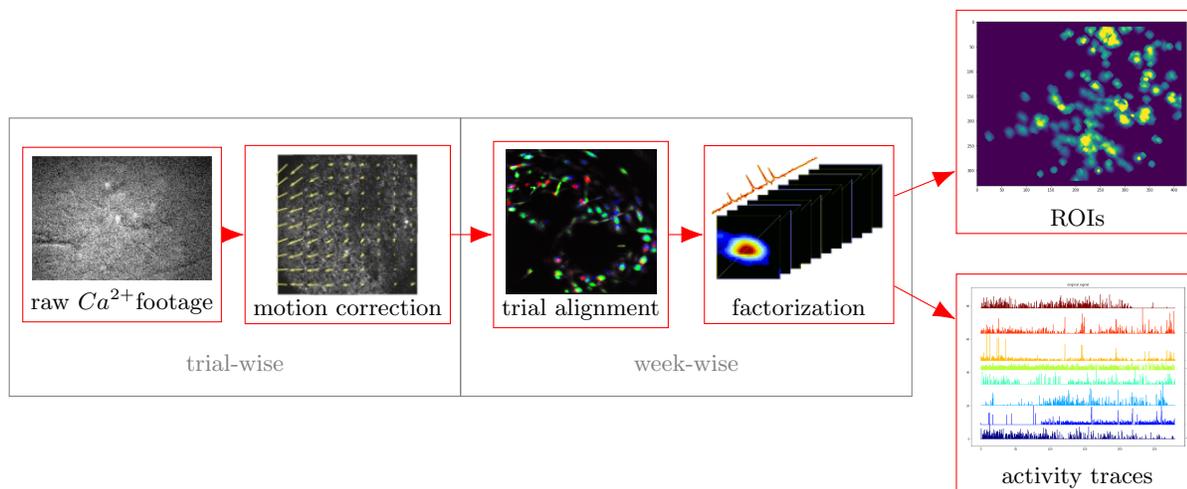


Fig. 2.3: The different components of the Ca^{2+} -imaging analysis pipeline: The raw footage had to be motion corrected trial-wise, the different trials had to be aligned over a whole week and the aligned material had to be factorized to extract the relevant neurophysiological data.

with the pixel-value based factorization algorithm, a motion correction algorithm was applied to the pre-processed trial videos. The trial videos were then joined together using a custom alignment algorithm. The aligned, concatenated video data of all 42 trial recordings was then used as input to the CNMF algorithm. The output of the pipeline was a matrix of recovered source ROIs and a matrix of cell activity vectors. In the following, each step of the pipeline will be treated in more detail.

2 Preprocessing

Main goal of the pre-processing step was to reduce the data size to accelerate further computations. This was achieved by down-sampling the recording in all three spatio-temporal dimensions. The down-sampling algorithm was applied during the conversion from the raw, proprietary recording format to the uncompressed `.tif` format that was used as pipeline input. The input was reduced by a factor of 2

in each dimension (using linear interpolation as an anti-aliasing method), which reduced the video frame-rate to $10Hz$. Only the frames 200 through 3000 of each recording were used to exclude detrimental image movements that occur during the transport of the animals between the resting and trial cages. For spatial cropping, ideal cropping points were determined manually for each animal and week through visual inspection of the Ca^{2+} -imaging footage. Goal was to further reduce the size of the dataset and to enhance the motion correction algorithm by removing the edges of the camera objective, which were visible as stable structures on the edges of the frame. The resulting videos had a size of $2800 \times 600 \times 420$ pixels and a field of view of $\sim 800\mu m \times 600\mu m$. They were saved as '.tif' files and fed as such to the further pipeline.

3 Motion Correction

The next step in the pipeline was a refined image alignment/ motion correction algorithm according to [Pnevmatikakis, Giovannucci \(2017\)](#), which was applied to each trial video individually. It used a state-of-the-art method for single-photon Ca^{2+} -image motion correction: A frame-wise rigid transform followed by a patch-wise correction (see [2.4](#) for an overview). Both transformation functions are based on the same image registration algorithm which is applied to the entire frame for the rigid transform and to sub-parts of the frame in the non-rigid transform. The algorithm itself is well established and described in e.g. [Guizar-Sicairos et al. \(2008\)](#). It uses frequency space representations of the two images and tries to isolate the phase deviation caused by the spatial shift of pixel values. The following example illustrates the basic functioning of the algorithm in a 1-d example.

The discrete Fourier transform F of an array of data points f is defined as:

$$F(x) = \sum_{n=0}^{N-1} f(n) \cdot e^{-\frac{i2\pi}{N}xn} \quad (3.1)$$

it is apparent that a second array with a shift in pixel values $g(n) = f(n + n_0)$ will preserve a frequency spectrum with the same magnitudes but a shift in phases:

$$G(x) = \sum_{n=0}^{N-1} g(n) \cdot e^{-\frac{i2\pi}{N}xn} \quad (3.2)$$

$$= \sum_{n=0}^{N-1} f(n + n_0) \cdot e^{-\frac{i2\pi}{N}x(n+n_0)} \quad (3.3)$$

The extent of the shift, n_0 , can be factored out:

$$F(x) = G(x)e^{\frac{i2\pi}{N}xn_0} \quad (3.4)$$

and resolved by calculating the normalized cross-correlation of the power spectra (using the complex conjugate of G , G^*):

$$R(x) = \frac{F \cdot G^*}{|F \cdot G^*|} = e^{\frac{i2\pi}{N}xn_0} \quad (3.5)$$

The peak location of this residual term (after re-transformed to pixel space domain) indicates the extend of the original shift, n_0 . This approach translates to the multi-variate space of two-dimensional images without loss of generalization and is used both in the rigid and non-rigid motion correction steps.

3.1 Rigid Transform

As a first step, the above-described algorithm is applied to the whole frames. The video's central frame is used to create an initial reference frame by removing the noise through a low-pass filter (convolution with a Gaussian of $\sigma = 3px$). The adjacent frames are iteratively fitted to this reference frame using the above-described cross-correlation maximization. During each step, the transform is applied and a new template image is generated taking the median values of all corrected frames (see 2.4 for a visual display of the pipeline; rigid transformation means that the patch size is equal to the frame size in this context).

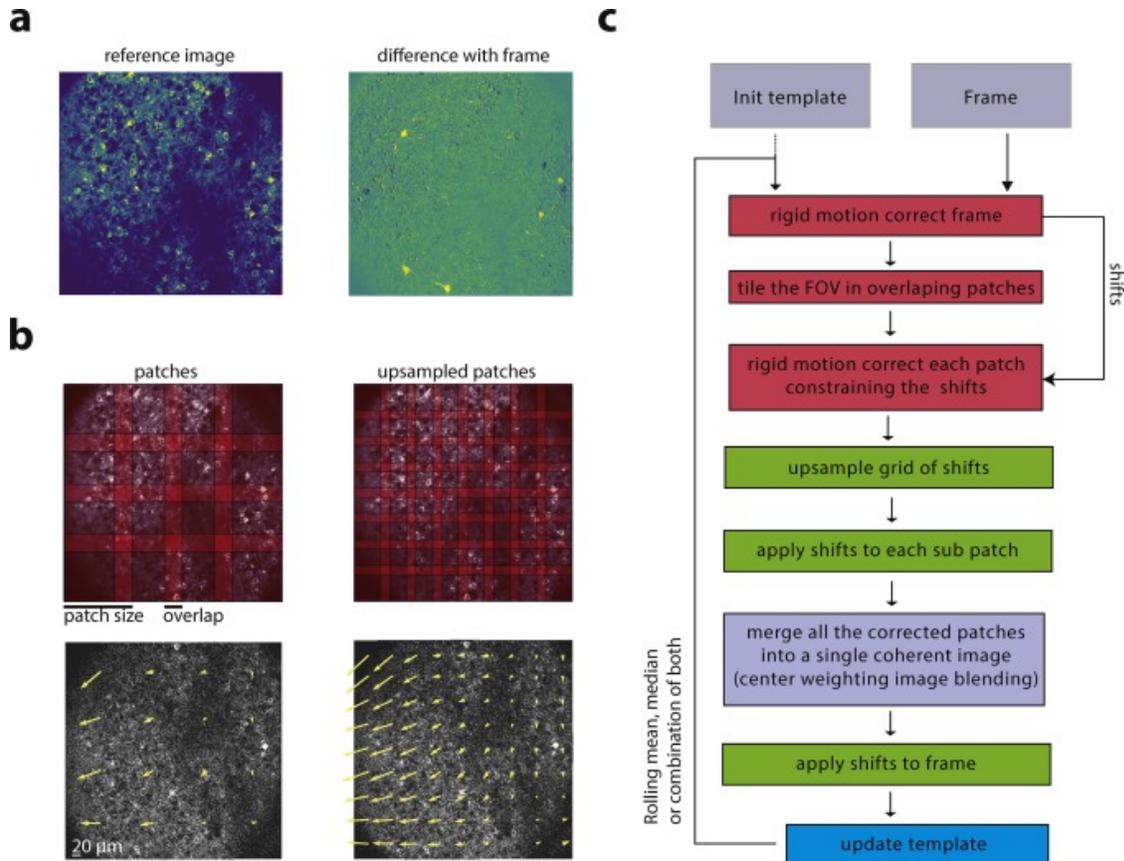


Fig. 2.4: The computational graph of the NoRMCorre motion correction algorithm (Pnevmatikakis, Giovannucci, 2017)

3.2 Non-Rigid Transform

The non-rigid, patch-wise transform has substantially more degrees of freedom and can correct for organic, non-linear movements of the brain tissue between frames. It is build on top of the rigid motion correction. It divides the image into smaller square patches of 80 pixels edge length with a 40 pixel overlap between patches. The individual patches are then fitted to a reference frame through a rigid transform fit of their own using the same algorithm as the rigid motion correction. Overlapping patches are then fused together using the linear interpolation heuristic described in (Pnevmatikakis, Giovannucci, 2017), 2.3.2.

4 Trial Alignments

The motion correction algorithm above used a frame-wise image transformation to maximize pixel similarity between frames. This method is suitable to align cells in a continuous video, as the physical reference structures in frame do not change significantly. To align frames from different trials over a period of different days, a dimensionality reduction method based on PCA-ICA was used to extract relevant structural features from each trial as reference points and determine the individual rigid transforms that align the different videos. The algorithm was based on a Movie Analysis pipeline by Benjamin Ehret of ETHZ (Ehret, 2018).

4.1 Preprocessing

For this PCA/ICA alignment, the first step was to do another pre-processing step. Each frame was divided by a low-pass-filtered version of itself to eliminate wide field fluctuations and global intensity gradients. A Gaussian filter was optimized by hand for each animal and week by visually determining the lowest σ^2 that equalizes intensity over the entire frame after application.

For each (x,y)-pixel location in the video, a baseline value was approximated by taking the intensity mean over time. Every frame in the movie was then divided by this approximated frame of baseline pixels (a procedure named $\frac{\Delta F}{F}$ in the literature). This way, any stationary background structures are divided out of the movie, which leaves the intensity changes of the calcium traces the single most prominent detectable feature.

4.2 PCA-ICA source extraction

This preprocessed $\frac{\Delta F}{F}$ video was then subjected to a consecutive PCA-ICA analysis to extract the location of neurons (as sources of Ca^{2+} -gradient) in the image. The expected number of components needed to be parameterized beforehand and was set to 500 cells(PCA) and 375 cells(ICA), following a thumb rule advised by the

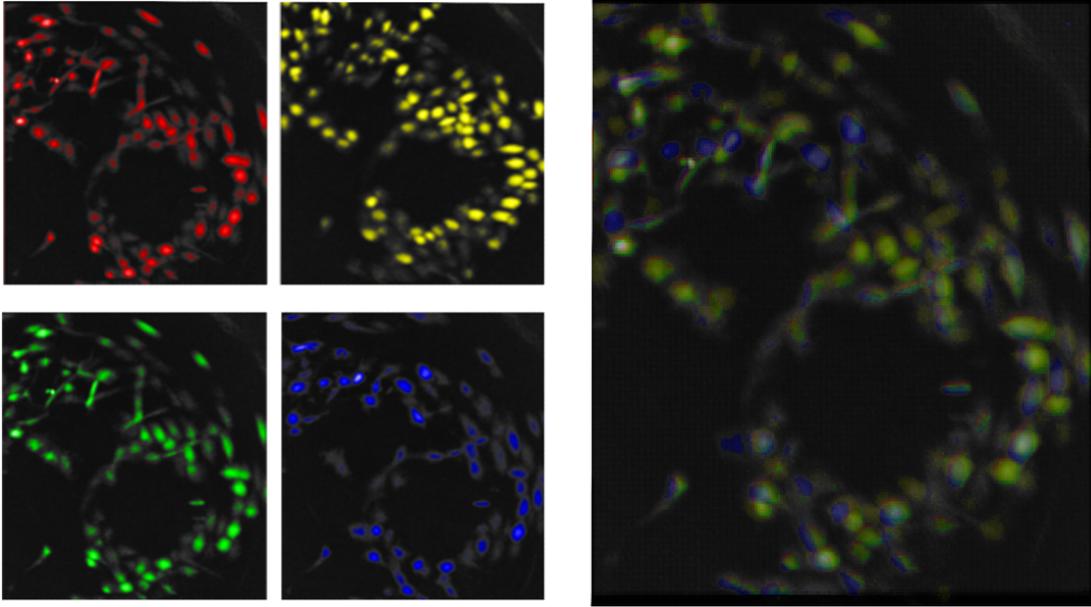


Fig. 2.5: A depiction of the alignment process: The left shows the extracted landmarks from four different trials, which are aligned and superimposed on the right.

creators of the algorithm. The factorization step was implemented using the generic MATLAB functions `pca()` and `rica()`.

4.3 Alignment Algorithm

2.5 shows the outputs of the PCA/ICA source extraction algorithm for four consecutive trials. The relevant independent signal sources are registered individually and used as landmarks in the subsequent alignment algorithm. All 350 landmarks (cells) had to be superimposed to create an image like the ones in 2.5. These images were then aligned using MATLAB's genuine `imregister()` function, which rendered a single affine transform matrix per trial.

The resulting image transformations were then applied to the superimposed landmark images and the results were inspected visually and, if necessary, adjusted. The determined transforms were then applied to the trial videos. Afterwards, all 42 videos of one week were concatenated. The entire video was cropped to remove the blank areas created by the transforms at the edges of the frames. The so created

single video file containing the motion corrected and aligned Ca^{2+} -imaging footage of all 42 trials in a row was then saved as a memory-mapped file and as such delivered to the factorization algorithm.

5 Source Extraction:

5.1 The Factorization Model

The *constrained non-negative matrix factorization* algorithm according to (Pnevmatikakis et al., 2016, 2014; Zhou et al., 2016; Friedrich, Paninski, 2016) called CNMF-E is at the center of the analysis pipeline. As described in the introduction, it uses a classic NMF implementation coupled with a few problem-specific constraints.

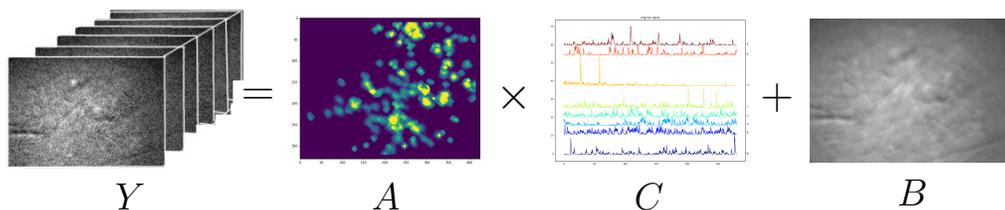


Fig. 2.6: A graphical representation of the factorization algorithm: The movie footage Y is modeled as the product of an ROI matrix A and a Calcium trace matrix C added to a stable background image.

The video is treated as a large three-dimensional matrix (of one temporal and two spatial dimensions, henceforth called 'Y'). Y is then used as a ground truth to be approximated by a model of confined-area, spike-dependant calcium traces before a background in a constrained non-negative matrix factorization.

The model contains several variables of interest:

- A representation of the image area covered by the single cell bodies ('A')
- the light intensity of the area over time ('C', 'calcium trace')

- the respective spiking activity of the individual cells that produce the calcium trace ($'S'$)
- additional background components ($'B'$) and
- and a random gaussian background noise ($'E'$).

These variables are combined to the characteristic equation of the CNMF-E model:

$$Y = A \cdot C + B(+E) \quad (5.1)$$

Discarding the noise term, the objective minimization function of the algorithm is the residual sum of squares (RSS) of the matrix $Y - (AC + B)$:

$$\operatorname{argmin}_{A,C,B} \|Y - (AC + B)\|_F^2 \quad (5.2)$$

while subjecting the variables A , C and B to certain constraints:

- The spatial footprint \mathbf{A} : Each cell region ($'ROI'$) should be confined to a small, spatially consistent region. (Pnevmatikakis et al., 2016)
- The temporal traces \mathbf{C} are modeled as a first-order autoregressive system (of natural Ca^{2+} diffusion) interferred by sparse, non-negative trains of action potentials according to Vogelstein et al. (2010); Pnevmatikakis et al. (2016).
- The activity signal \mathbf{S} is the inferred magnitude of the action potentials that perturb C . It includes all changes in \mathbf{C} that are not explained by the autoregressive model. S is used as the actual measure of neural activity and the pipeline's main output.
- The background components B are split up into a stationary (b^c) and a fluctuating (b^f) component such that $B = B^f + B^c$.
- The stationary background is modeled as a spatial matrix that is constant over time: $B^c = b_0 \cdot T$, with T being an array of ones of the length equal to the total number of frames.

- The fluctuating background is supposed to model local signal noise that typically occurs when out-of-focus neurons are active. These out-of-focus signals usually have a much larger diameter than the in-focus neurons of interest. B^f is modeled using the so-called 'ring model': $B^f = W \cdot Y$, where $W_{i,j} = 0$ for all entries except for when (i,j) is the pixel count of two pixels with a fixed euclidean distance l_n to each other in the frame. This way, B^f for a certain pixel is modeled as the weighted activity of the surrounding pixels at a distance of l_n .
- The normal statistical noise E is considered negligible in magnitude.

5.2 Variable Initialization

The matrices A , C , and B needed to be initialized. For this project, an iterative greedy algorithm was used. A Gaussian filter that approximated the shape of a signal source was applied spatially. In the filtered footage, the spatial pixel location with the maximal temporal variance was chosen as the center of the approximated neuron. A 20 pixel square window around this peak location was isolated and subjected to a rank-1 NMF, the results of which were appended to the matrices A and C . The current product $A \times C$ was then subtracted from Y to produce the residual that was used as input for the next iteration round (See [Pnevmatikakis et al. \(2016\)](#) for a more detailed description of the stopping and acceptance criteria). The frame-wise mean pixel of the eventual residual was used as an estimate of B .

5.3 Factorization

After initialization of the relevant variables, the characteristic equation $Y = A \cdot C + B + E$ needed to be solved. Due to the above-mentioned non-linear constraints, the problem cannot be solved in closed form. It can however be broken down into a number of convex subproblems that can be solved iteratively in a multiplicative update algorithm ([Zhou et al., 2016](#)):

– Estimating A, b_o given \hat{C}, \hat{B}^f :

$$\begin{aligned} & \underset{A, b_0}{\operatorname{argmin}} \|Y - A \cdot \hat{C} - b_0 \cdot 1^T - \hat{B}^f\|_F^2 & (5.3) \\ & \text{subject to:} \\ & A \leq 0 \\ & A \text{ is sparse and local} \end{aligned}$$

– Estimating C, b_o given \hat{A}, \hat{B}^f :

$$\begin{aligned} & \underset{C, S, b_0}{\operatorname{argmin}} \|Y - A \cdot \hat{C} - b_0 \cdot 1^T - \hat{B}^f\|_F^2 & (5.4) \\ & \text{subject to:} \\ & c_i \leq 0 \\ & s_i \leq 0 \\ & G^{(i)} c_i = s_i \\ & s_i \text{ is sparse } i = 1 \dots K \end{aligned}$$

– Estimating W, b_o given \hat{A}, \hat{C}^f :

$$\begin{aligned} & \underset{W, b_0}{\operatorname{argmin}} \|Y - A \cdot \hat{C} - b_0 \cdot 1^T - \hat{B}^f\|_F^2 & (5.5) \\ & \text{subject to:} \\ & B^f \cdot 1 = 0 \\ & B^f = W \cdot (Y - \hat{A}) \cdot \hat{C} - b_0 \cdot 1^T \\ & W_{i,j} = 0 \text{ if } \operatorname{dist}(x_i, x_j) \notin [l_n l_n + 1] \end{aligned}$$

6 Results

After a protracted phase of parameter tuning and code optimization, the motion correction and CNMF-pipelines could process most of the data without exceeding hardware constraints or causing program errors. However, there still exists at least one week of Ca^{2+} -imaging in our database that did not factorize with the current parameter settings, revealing a further need for parameter optimization and debugging of the code base. Typically, improving quality of the motion correction and

alignments can elude computational errors in the factorization algorithm (such as division-by-zero). Due to time constraints, the ultimate cause of these computational errors need to stay unsolved for now.

The quality of the motion correction algorithm was assessed using visual inspection of the results and a peak-to-noise / local pixel correlation indicator. Both indicators are improved after application of the motion correction algorithm, and the quality of the motion correction was deemed sufficient after visual inspection of the corrected footage (Figure 2.7).

For a large majority of the Ca^{2+} -imaging data the pipeline could produce very reasonable results, which are described in detail in chapter 5. A typical number of 200-500 cells could be traced reliably over a whole week (table 1). The recovered cell bodies (*regions of interest/ROIs*) can be evaluated visually by plotting them on top of a local correlation map (table 1). Respective plots show that the pipeline with the current settings can retrieve a reasonable amount of signal sources from the Ca^{2+} -imaging footage.

The extracted activity traces appear plausible upon visual inspection. Their shape is similar to Ca^{2+} signals reported in other neural Ca^{2+} -imaging studies (Friedrich, Paninski, 2016; Pnevmatikakis et al., 2016). A striking feature of many cell signals is an activity bias towards either resting or testing phases as shown in figure 2.8.

The quality of the recovered cell ROIs and the shapes of the corresponding traces were deemed sufficient for to be passed to the remaining pipeline and subjected to a number of supervised and unsupervised statistical tests described in chapter 4. The results and findings from these statistical tests that further help estimate the quality of the factorization are discussed in chapter 5.

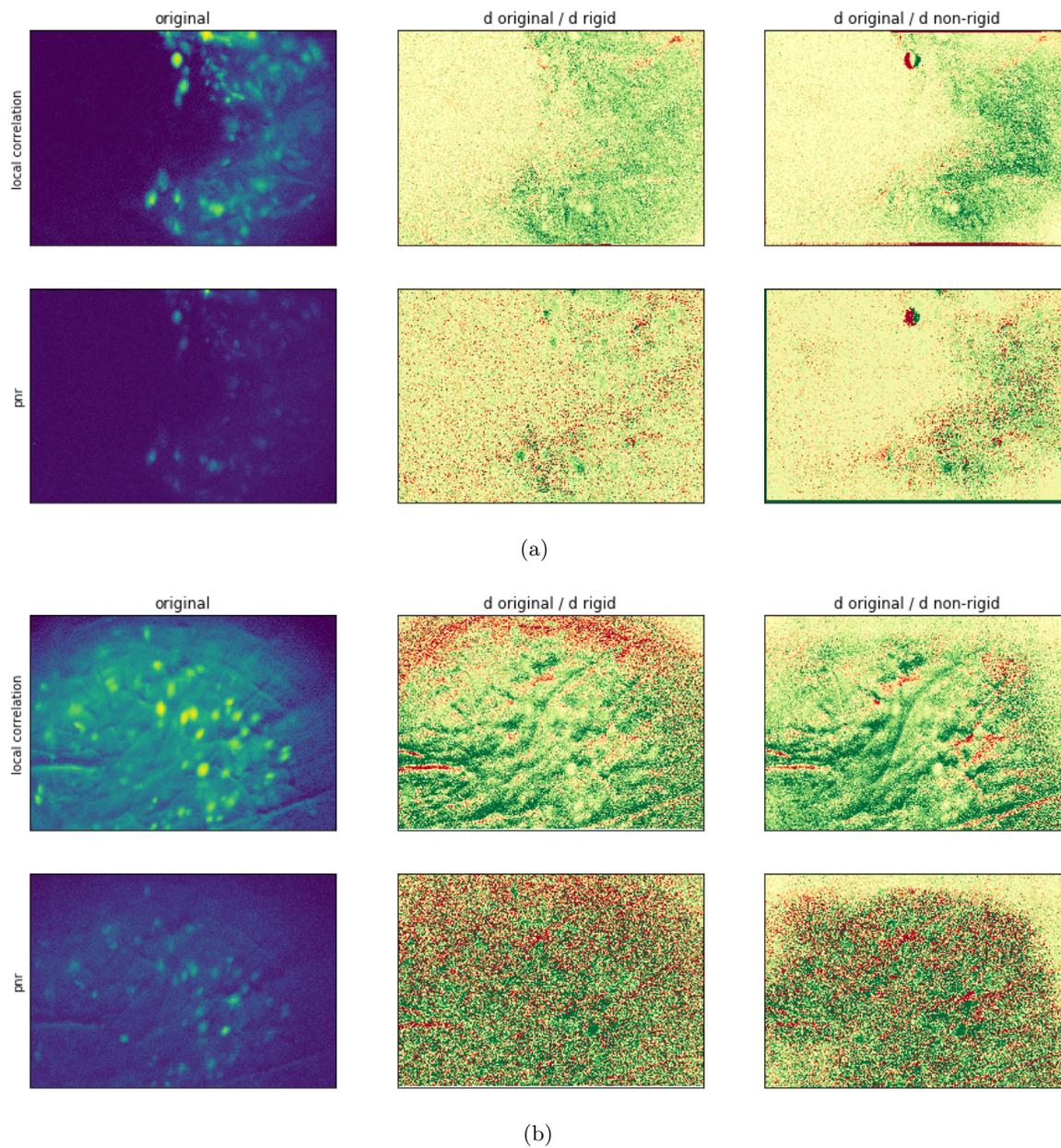


Fig. 2.7: Motion correction quality measures for two different animals (a,b). Rows (1,3) show a local correlation measure, rows (2,4) show the peak-to-noise ratio over 2800 frames. The first column shows the values before motion correction. The second column shows the change in value after rigid motion correction (green→improve, red→worsen). The third row shows the change in value after the non-rigid motion correction. Both measures indicate an effective improvement, where the non-rigid algorithm handles edge distortions especially well.

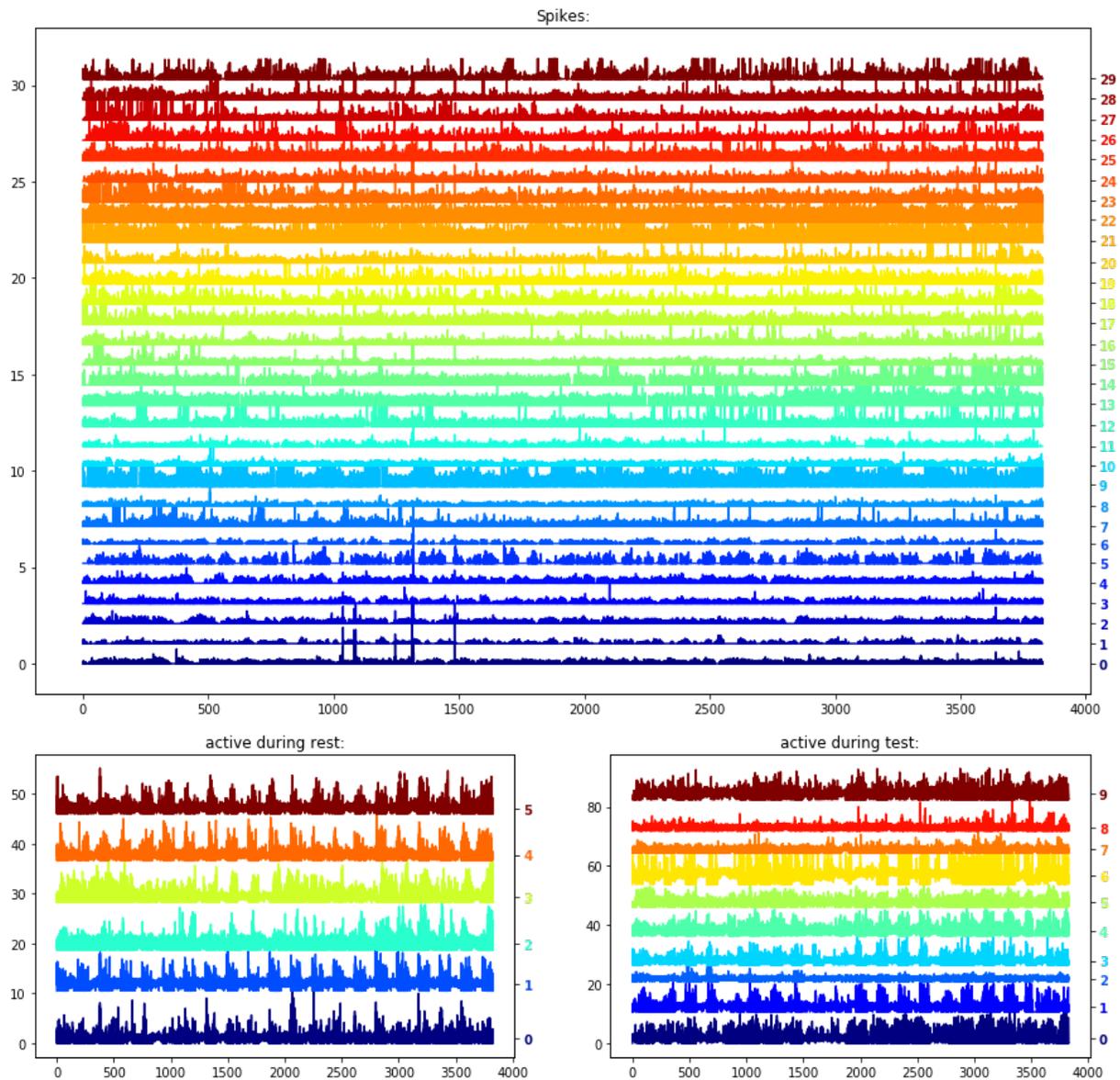


Fig. 2.8: A selection of recovered activity traces of a single animal from video footage collected over a single week. The top figure shows traces from 29 randomly selected cells, the bottom left figure shows cells that are preferably active in the resting phases, the bottom right figure shows cells that are active during test phases. The preferences can clearly be seen as periodically alternating patterns of 21 active and passive segments in the activity traces.

Chapter 3

Automatized Behavioral Analysis

1 Introduction

Next to the Ca^{2+} -imaging footage, the experiment protocol also produced a synchronized behavioral video captured from a camera mounted above the arena (figure 3.9). A behavioral analysis pipeline was created to track the animals' behavior over time and create a number of behavioral variables to correlate to the extracted Ca^{2+} -imaging traces. As relevant behavioral variables were chosen:

- 'Pose': The position of the animal in the arena
- 'Exploration': A Boolean variable that indicates whether the animals explore an object in the arena at a given time.

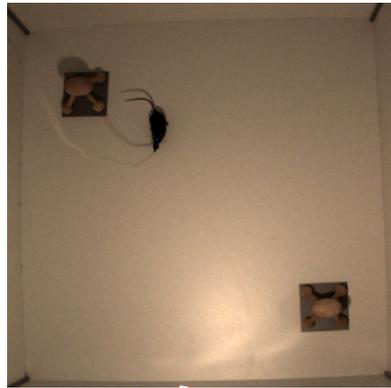


Fig. 3.9: The raw input to the behavioral classifiers: A simple image from a camera mounted above the arena. The position of the animal and its exploratory behavior towards the two objects need to be extracted.

The two 'pose' and 'exploration' variables were extracted using separate deep learning algorithms. The first algorithm, used purely to extract the pose of the

animals, was an adaptation of the *DeepLabCut* pipeline (Insafutdinov et al., 2016, 2017; Mathis et al., 2018). The second one, which was used to determine exploratory behavior in the video frames, was an adaptation of the pre-trained *inception-3d* model published by (Carreira, Zisserman, 2017).

2 DeepLabCut: Retraining ResNet101 for Pose Estimation

DeepLabCut is an established pipeline that uses a transfer learning in a fully-convolutional ResNet model to identify user-defined animal body parts in a given video frame (Insafutdinov et al., 2016, 2017; Mathis et al., 2018).

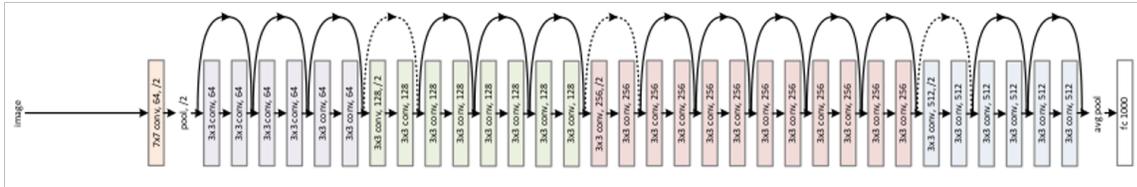


Fig. 3.10: The computational graph of ResNet. The output of each convolutional sub-element is added to its input, which makes elements approximate residuals rather than absolute values. The different colors denote the convolutional blocks (He et al., 2016).

Pre-trained ResNet models in tensorflow’s *slim* format are readily available online (Silberman, Guadarrama, 2017). This project’s version of DeepLabCut used a ResNet-101 architecture (He et al. (2016);figure 3.10) that had been pre-trained on the *ImageNet* dataset (Deng et al., 2009) until convergence. DeepLabCut slightly modifies the generic architecture of ResNet101 to be applicable for pose estimation.

Most importantly, the final dense layers are removed to make the model fully convolutional and preserve the spatial information throughout the entire network.

The convolutional section of ResNet101 is composed out of four functional blocks, each of which shrinks the $[x, y]$ -dimensions of the signal by a factor of two through pooling or stride operations (figure 3.10). This step is necessary to meet hardware restrictions, but discards a lot of spatial information. To diminish some of the effect, part of the signal was by-passed around the last block to counteract information loss during the last down-sampling step. For this purpose, a custom block was added to the graph that performed a factor-2 up-sampling of the final ('block 4') output signal (which had been reduced to $1/16$ and is restored to $1/8$ network input size this way). This up-sampled signal is then stacked with the output of the second-to-last block 3 (which also has $1/8$ network input size). On top of this concatenated layer, a convolutional layer with a sigmoid activation function was added as the network's new output layer. The spatial dimension of the convolutional kernel of this new output layer was $[1, 1]$ and the number of output channels was set to the number of desired output categories (two in our case - 'nose' and 'tail base'). Each of the channels of the output layers was to produce a heat-map for one of the categories ['nose', 'tail base'].

A total of 200 random frames were extracted from the entire body of behavioral videos to retrain the model in a fine-tuning session. The location of the base of the animal's tail and its nose within each frame was manually labeled. The dataset was augmented by mirroring frames randomly across the x- and y-axes and by randomly resizing the input images by a factor between 0.5-1.5 on-line during training. The frames were fed to the network in batches of RGB images of the format $[batches, width, height, RGB_channels]$. The target data was a 3d-array of the size $[batches, \frac{width}{8}, \frac{height}{8}, categories]$. The target array was filled with zeros except for the position $[\frac{x}{8}, \frac{y}{8}, c]$, which denotes the x and y pixel position for each label category c in the input image, which was set to 1.

The network was trained on the dataset for a total of 50000 epochs until the quality of the predictions was considered sufficient for the purpose of this study.

The location of the animals was then obtained through a prediction session of videos of the animals behaving in the environment. From the network output, the (x,y)-coordinates of the peak of each category channel and the p-value of this peak were extracted for the further analysis. This data was then combined with next section's behavioral results in a statistical pipeline described in section 4 of this chapter.

3 AutoScore: Training Inception as a Behavioral Classifier

The second behavioral analysis algorithm was created to detect so-called *exploratory behavior* of the animal in the frame. A single boolean variable was to indicate whether the animal directed its attention towards any of the two objects within the frame (figure 3.11). This classification task was tackled using another deep learning algorithm, albeit with a different basic approach. Other than for the positional tracking algorithm of the previous chapter, a proper training set was available for the behavioral classification task, allowing for a higher dimensional input and an complete training run until convergence.

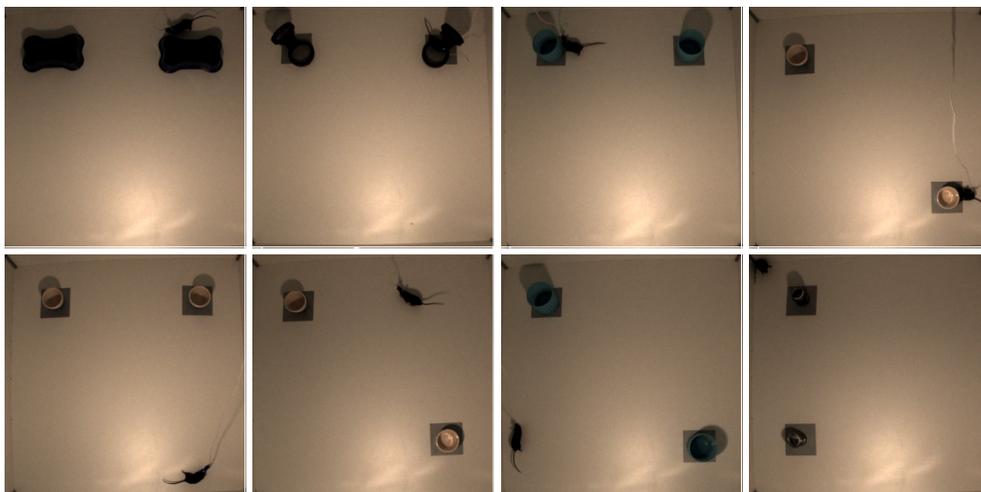


Fig. 3.11: A number of frames to evaluated as *true* (top) and *false* (bottom).

Inception3d is a neural network architecture based on Google’s Inception model for image classification (Szegedy et al., 2015). Inception models are readily available online in tensorflow’s *slim* format and pre-trained on the ImageNet dataset (Silberman, Guadarrama, 2017). The characteristic of the **Inception3d** architecture is an additional temporal dimension obtained in a process called ‘inflation’ first described by Carreira, Zisserman (2017). In their method, all convolution filters are expanded to process 3d $[time, x, y]$ video input instead of only 2d $[x, y]$ image input. To this end, every 2d convolution filter is copied n times and stacked along a new ‘time’ axis. The value of the weights is divided by the factor n , which means that a ‘boring’ video consisting only of a repeated, stacked ImageNet image would generate the same activation patterns in the network as the single image would have in the original 2d Inception. The inflated network was then pre-trained a second time on the *kinetics* video dataset (which is smaller than ImageNet and has an additional time dimension).

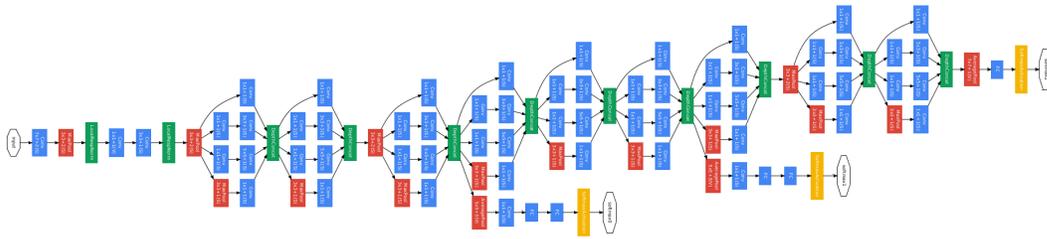


Fig. 3.12: The computational graph of Inception V1: Each inception block consists of a number of parallel convolutional sub-graphs that can represent features of different sizes (Szegedy et al., 2015).

The final, pre-trained video categorization model was then used for a transfer learning session on our own behavioral dataset. To this end, the network’s final, dense layers were removed and replaced by a single dense layer with a sigmoid activation function and two output nodes denoting the categories *exploratory* and

non-exploratory. The binary categorical cross-entropy of the predictions was used as a loss function (also preparing future uses of the model with multiple behavioral categories). The optimizer was a general stochastic gradient descent with an initial learning rate of 0.01 (which decayed by a factor of 10^{-6} after each epoch) and a *Nesterov* momentum of (0.95).

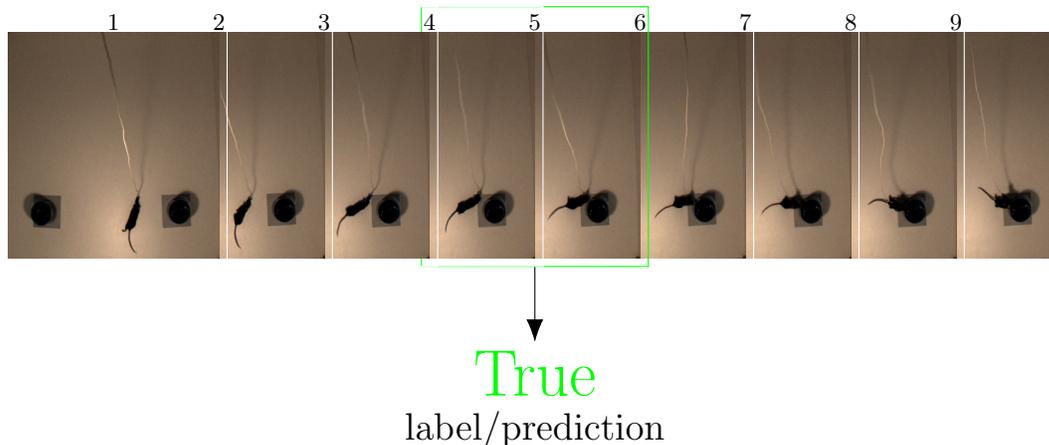


Fig. 3.13: A single data point in the Inception3d training set: The independent variable consists of nine consecutive frames. The behavioral label of the central frame is used as the dependent variable.

The in-house dataset contained a total of 208 behavioral videos of ~ 5 -minutes and 6000 frames length. The videos had been hand-labeled by the experimenters in real time by holding down a button whenever an animal showed exploratory behavior towards one of the two objects in the arena. During re-training, 9 consecutive video frames were used as the independent variable, and the manual label of the central (fifth) frame was used as the dependent output variable (figure 3.13). The network was trained for 100 epochs until the quality of the output was considered sufficient for the purpose of this study.

The network was then used to run predictions on the behavioral videos of our experimental dataset. Before feeding the prediction frames to the network, the slight difference in camera perspective of the training and the prediction set was

corrected for using an affine transformation. The color values of the prediction data sets were adjusted using histogram matching. A representative training histogram was created using two frames from every training video. The histogram of each prediction frame was adjusted to match the histogram of the representative training histogram (figure 3.14).

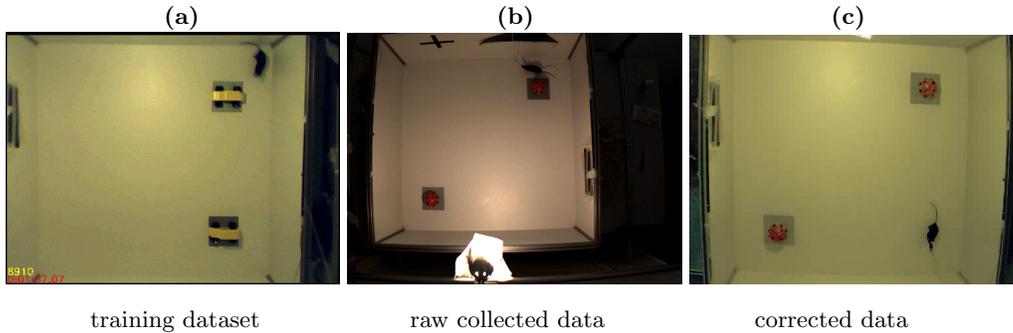


Fig. 3.14: Matching the perspective and the color histograms of the training datasets (a) and the collected behavioral video footage (b) helped to improve model predictions.

This prediction round produced a vector of exploration p-values for each frame of every video. This vector of behavioral predictions was then passed to a post-processing algorithm described in the next section.

4 Post-Processing and Data Integration

The two video analysis models presented above produced two synchronized arrays of behavioral data. One array contained the (x,y) coordinates of the animal alongside a probability estimate of the prediction, the other array contained a frame-wise p-value for the animal's exploratory behavior. Typically, both prediction arrays contained errors or low-confidence predictions that occurred when the animal was covered by the object or showing ambiguous behavior that could be interpreted as

grooming as well as exploring. An augmentation routine was therefore added to the pipeline that tried to eliminate a number of identifiable errors.

For the pose estimation data, low-confidence or wrong predictions could be partially detected both from the provided p-value and from the plausibility of the recovered positions. Both indicators were used to augment the data in a probabilistic model based on a Kalman filter.

A Kalman filter is a data augmentation algorithm that can be used to remove signal noise from a series of measurements using Bayesian reasoning (Kalman, Bucy, 1961). The measurement variables - in our case the location and speed of the animal's head and tail - are modeled as multivariate Gaussian processes, which makes it possible to detect (and correct) improbable data points and to infer an unavailable or 'masked' value in the dataset from the values of the adjacent time-points and dimensions. This way, the position of an animal's nose in a frame can be estimated using the position and velocity of the nose in the previous and subsequent frames, but also using the position and velocity of the tail base in the current frame and the surrounding frames.

For the application of the Kalman filter, all time points with a p-value below 0.2 (which occurred e.g. when a body part was covered by an object or the animal was at an awkward position) were marked as 'masked datapoints'. A matrix containing the positions and velocities of the different body parts at all time points was then smoothed using a python-based Kalman filter algorithm (Duckworth, 2012). The final output of the smoothing algorithm was a plausible array of positional data for the nose and tail base of the animal throughout the entire experiment. Of the two, only the spatial coordinates of the nose was used in subsequent steps in the pipeline.

The behavioral output of the **Inception3D** network was a single p-value between 0 and 1 indicating exploratory behavior in the respective frame. As a first step, the value was transformed to a boolean variable using a threshold of 0.5. Furthermore, **Inception3D** cannot not distinguish between exploratory behavior

towards the two objects in the arena. Therefore, object-specific exploratory behavior had to be restored by combining the behavioral data and spatial positional data of the animal. In the task, the objects had four possible positions in the four corners of the frame. Accordingly, an exploratory event was simply assigned to the object located in the quarter of the frame that also contained the nose of the animal in the given frame.

5 Results

Given the small training set, the sparse and vague nature of the output and the cross-validation training scheme, the eventual performance of **DeepLabCut** is difficult to describe numerically. The trained network output was simply considered sufficient after of the visual inspection of a few prediction results rendered on top of the input videos (see figure 3.15 for an impression of DeepLabChop’s output).

The most powerful version of **Inception3D** trained for our project reached an accuracy value of $0.94-0.95$ correct predictions different, independent testing videos. This is a quite satisfactory performance considering the prevalent human error in the training and validation data and the categorical ambiguity of many situations (the exemplary predictions for a single session are rendered in figure 3.16).

However, there are also a few systematic disadvantages to an automated scoring algorithm over human scoring. Deep learning algorithms often do not adapt well to -seemingly superficial- changes in the prediction task. For instance, the OST protocol requires ‘cues’ on the arena walls for the animals to orientate on. Often, black cut-out shapes are used and glued onto the walls within the field of view of the camera. Both networks could learn to distinguish between the shape of the moving animal and the black orientation cues that are present in the training set video very well. However, this ability to discriminate did not generalize well to a few newer videos that were not included in the training set and featured different looking

orientation cues. For the newer videos, some cues therefore had to be masked and re-stained in the color of the background wall in an additional pre-processing step (Fig 3.17).

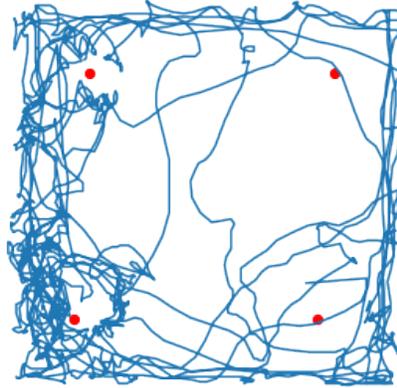


Fig. 3.15: An example of **DeepLabChop** predictions during a single, 5-minute trial. The red dots show the possible object positions in the square arena, where this particular session only included the two objects on the left side. The trace shows a clear preference for positions close to the walls and corners of the arena ('thigmotaxis'). Also, the corners with the objects are frequented more often than the empty corners

6 Conclusion

In conclusion, the behavioral part of the project could show that animal position and exploration analysis can potentially be automatized using a customized deep learning set-up. This automation can possibly save a considerable amount of human scoring hours and remove human bias and error from the scoring data. Also, the algorithm can run in real time with a delay of a few hundred milliseconds (in our case on an Nvidia 1070ti GPU), which allows for an experimental design with

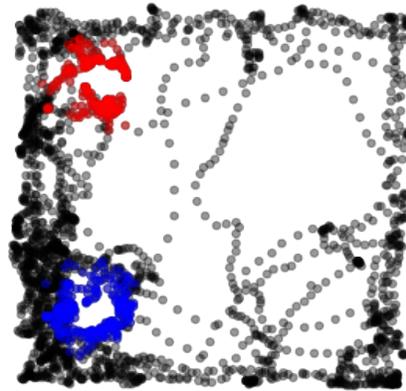


Fig. 3.16: Frame-wise **Inception3D** exploration predictions for the positions from figure 3.15: Blue and red scatter points indicate exploration of object one and two respectively. Black scatter points indicate no exploration.



Fig. 3.17: **DeepLabCut** generalization error: The network correctly identifies the nose and tail of the animal sitting on an object (left). In the next frame, the wall cues are misinterpreted as the animal tail (middle). The error can be corrected for using a color correction of stationary black pixels on the wall (right).

automated animal interaction (e.g. rewards or punishments for a certain observed behavior).

However, as shown in the result section, an algorithmic prediction can produce systematic errors if the input images are not optimal. Seemingly trivial changes in the prediction task can lead to very poor performance and require careful experimental design or additional image-processing steps. Systematic biases can also be introduced if the salience of the frame is not even, which might happen if the camera or the light source are tilted and one object is more visible than the other.

Even though the initial results are promising, further application will need to show whether the **DeepLabCut-Inception3D** ensemble is practicable in long-term use for the behavioral analysis of the object space task. One of the next major steps would probably be the creation of a more diverse dataset to increase the generalization capabilities of the networks. Another interesting project would be the conceptual integration of the two architectures by applying the inflation technique to **DeepLabCut**'s deep learning architecture.

Chapter 4

Statistical Analysis

1 Introduction

The third integral part of this project was the construction of a number of statistical tests to be able to verify the performance of the behavioral and Ca^{2+} -imaging analysis pipelines and to be able to interpret the extracted behavioral and neural signals in the context of the object space task. Both supervised and unsupervised statistical algorithms were implemented as part of this project. A number of supervised tests were based on simple linear regression fits of the extracted neural and behavioral data. The goodness-of-fit of the two data streams was used to verify common information in the two data sets using a random permutation test. Regression fits on systematically shifted data was used to investigate temporal outreach of the mutual information ('Granger causality'). An unsupervised statistical test was implemented to investigate mutual information between individual neurons (such as the formation of neural ensembles during memory formation). A second (mostly) unsupervised test was created based on a tensor decomposition of neural signal snippets of a fixed length around the onsets of exploratory behavior. In the following sections, I want to give a more detailed description of each of the statistical tests.

2 Supervised methods

2.1 Regression Analysis

A statistical method is defined as 'supervised' when it relates the statistical data to a ground truth value. Such methods were available since our dataset contained both

independent neural data and dependent behavioural data. Supervised analysis was used first and foremost as a quality check for both analysis pipelines by determining the amount of mutual information in the extracted data streams. Supposedly, the more mutual information to be found in the pipeline outputs, the higher the quality of the data itself and the more accurate the data processing and signal extraction pipelines. Next to serving as a quality check, supervised analysis can also be used to determine certain properties of the neural signal such as the temporal reference.

All supervised statistical tests used a simple linear regression (as the most basic supervised model), which relates an independent variable X to an independent variable Y through a vector of linear factors β :

$$Y = X\beta \quad (2.1)$$

This simple linear regression problem can be solved in closed form when X and Y are available:

$$\beta = (X^T X)^{-1}(X^T Y) \quad (2.2)$$

with, in our case, X the neural activity trace matrix, Y a matrix of behavioral indicators over time and β the regression coefficients to be inferred. Concretely, Y was prepared as a boolean matrix with behavioral categories as rows and the frame count as columns. Two rows of the matrix encoded exploratory behavior towards the two objects inside the arena. For frames in which the animal explored object 1 or object 2, the respective frame in the respective row was set to 1. The remaining 49 rows represented a grid of square patches in the behavioral camera image. The rows are set to 1 whenever the animal's head was captured inside the patch. The remaining entries of Y were set to 0. The regression was then performed using the equation displayed above in different configurations.

As a first sanity check, the regression was performed on the data of the first four days of the week, whereas the last day's data was used in a cross-validation scheme.

Behavioral predictions for the last day were created using the trained regression model and rendered over the behavioral video for a visual evaluation (figure 4.18). Furthermore, a regression model was trained on each of the first three days of the week and cross-validated on the following day to estimate the across-day continuity of information in the neural signal.

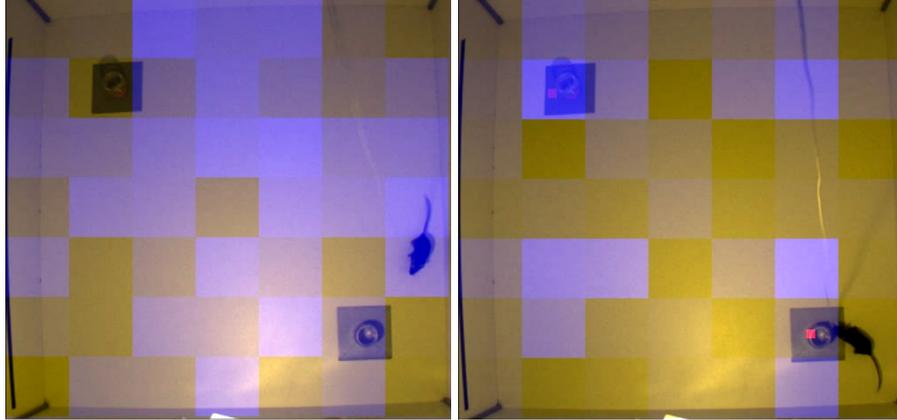


Fig. 4.18: For the supervised analysis, the arena was divided into 49 squares (shown in blue). Furthermore, each of the objects was assigned an exploration probability value (shown as a red dot). In a cross-validation scheme, a linear model was trained on the first 4 days and verified on the last day. The cross-validation predictions are rendered on top of a video of the original behavior for a sanity check.

The overall basic functioning of the analysis pipelines was tested using a simple random permutation/bootstrapping test. To this end, 1000 alternative data sets were produced by randomly shifting the behavioral data along the time axis. A linear regression fit was performed on each of the datasets with the mean squared error as a goodness-of-fit indicator. The mean-squared error of the least-square regressions of the original, unaltered datasets was determined and tested against bootstrapped errors using a simple Student's-T test.

To test the temporal accuracy of the neural signal, a number of linear regression fits were performed on the neural data using a shifted version of the behavioral

target data. The implication is for the neural signal to carry information about 'future' or 'past' behavioral states. To this end, for every animal, 300 alternative behavioral datasets were created by shifting the original dataset in a range from -150 frames to +150 frames. A linear regression fit was then performed using the original neural data and each manipulated behavioral dataset. Mean squared error of the linear regression was used as a goodness-of-fit indicator. A comparison of the different errors could be used to investigate the amount of intentionality and representations of future states in the neural signal ('Granger causality').

The results of the supervised statistical tests applied to the actual Ca^{2+} -imaging dataset are presented in chapter 5.

3 Unsupervised Ensemble detection using non-negative matrix factorization:

3.1 Introduction

Unsupervised statistical methods do not rely on an external depend variable but can instead be used to reveal patterns and structures within a dataset. For this project, unsupervised methods are needed to recover cellular ensembles in the extracted activity traces.

Cellular ensembles are groups of cells with synchronized activity patterns. Such patterns can be extracted from the signal of multiple cells using methods of dimensionality reduction. A well established approach is to apply a PCA/ICA to recombine the activity traces into an equal number of template traces with minimum mutual information (Santos Lopes-dos et al., 2011; Peyrache et al., 2010). Even though this approach is thoroughly worked out statistically and well established, it comes with a few disadvantages. Most importantly, an ICA can produce part-negative components and negative weights, which makes the output difficult to interpret in a neural signaling context.

Therefore, given that the implementation of the Ca^{2+} -extraction pipeline required the respective functionality anyway, I opted for designing a restricted non-negative matrix factorization algorithm to extract symbolic activity traces from the neural signal. This technique has been used for neural ensemble detection before (Mackevicius et al., 2018), even though its application in Ca^{2+} -imaging data is innovative.

In general, the iterative implementation of the NMF algorithm allows for more versatile constraints than a closed-form PCA. Therefore, the NMF-algorithm could be adopted for this project to specifically reveal the dynamics of neural ensembles in Ca^{2+} -imaging data of the object space task. This added functionality to detect not only ensemble traces over time but also estimate the evolution of the ensemble strength over the course of the entire recording week is needed to test the hypothesis of prefrontal ensemble formation during learning.

3.2 Methods

The restricted non-negative matrix factorization has the same basic approach as the PCA/ICA: The signal traces are recombined into a new set of representative traces and a matrix of recombination factors. While PCA minimizes correlation and ICA minimizes mutual information between the traces, NMF minimizes the loss between the factorization product and the input matrix. The implementation of this project is heavily based on python’s *sklearn.NMF* class, which was slightly modified.

In general, ensemble dynamics were modeled using a factor matrix W of size $[n_cells, n_ensembles]$ that contained a participation factor for each cell and each ensemble, and a pattern matrix H of size $[n_ensembles, frames]$ that contained template activity traces for each ensemble. In our implementation, W was a square matrix with equal width and height. As a loss value, the factorization algorithm

used a the Kullback-Leibler divergence between the dot product of the weight matrix W and the trace pattern matrix H and the measured activity traces X :

$$\operatorname{argmin}_{W,H} (KL(X, WH)) \quad (3.1)$$

The fact that the number of components in the weight matrix is equal to the number of traces makes this problem overdetermined: One of many trivial, perfect solutions would be to set $H = X$ and $W = I$, with I the identity matrix. In order to enforce sparsity on the solution, we need to add a number of regularization terms to the minimization:

$$\operatorname{argmin}(KL(X, WH) + \mathcal{L}(H, W)), \quad (3.2)$$

with:

$$\mathcal{L}(H, W) = \lambda_{h1}|H|_1^1 + \lambda_{h2}|H|_2^2 + \lambda_{w1}|W|_1^1 + \lambda_{w2}|W|_2^2 \quad (3.3)$$

Specifically, a weighted sum of the l_1 and l_2 norms of W and H are added to the minimization problem to enforce sparsity. This minimization problem can be solved through an iterative algorithm with two alternating multiplicative update steps (Lee, Seung, 2001; Cichocki, Phan, 2009; Févotte, Idier, 2011):

$|X - HW|$ is non-increasing under the update rule for H :

$$H \leftarrow H \circ (W^T X) \circ (W^T W H)^{-1} \quad (3.4)$$

as well as for the update rule for W :

$$W \leftarrow W \circ (H X^T) \circ (H H W^T)^{-1} \quad (3.5)$$

with \circ the element-wise Hadamard-product (a proof is given in Lee, Seung (2001)). The lambda regularization parameters can be integrated into the update equations as added penalty terms:

$$H \leftarrow H \circ (W^T X) \circ (W^T W H + \lambda_{h1} + H \lambda_{h2})^{-1} \quad (3.6)$$

$$W \leftarrow W \circ (H X^T) \circ (H H W^T + \lambda_{w1} + W \lambda_{w2})^{-1} \quad (3.7)$$

These two update steps are alternated until convergence or for a maximum number of 200 times.

Up to this point, the factorization algorithm renders a number of representative week-long neural activity patterns H and a weight matrix W that, for all recorded cell traces, contains the linear combination factors of H that approximate the original signal X .

The fact that weight matrix W contains only a single indicator for each cell and activity pattern makes it difficult to interpret the progress of the linear combinations over time. In order to recover drifts in the ensemble structures - such as cells entering or leaving the ensembles or entire ensembles emerging or dissolving - one would need to apply external processing steps, such as determining the changes in correlation between cell activity and the ensemble trace in H over time.

A different approach would be to perform multiple factorizations across a smaller time frame - say trial-wise - and compare the cell ensembles that are found throughout the different trials. Such a model would have additional degrees of freedom to represent different ensemble combinations over time. However, such trial-wise factorizations do not contain any continuous, inclusive information to relate the detected trial-wise ensembles to each other. One might try to relate trial-wise ensembles using the proximity of their rows in W (i.e. the similarity of the ensemble

participants). However, a considerable amount of heuristics would be necessary to define ensemble identity over time.

The week-wise and the trial-wise factorization in fact constitute two versions of the same algorithm, but with exclusive flaws and benefits: The week-wise factorization produces week-long ensembles without any dynamic changes and therefore maximal continuity. The trial-wise factorization, on the other hand, accurately displays the ensemble structures for individual trials without a clear concept of continuity between trials.

The general versatility of the NMF algorithm allowed the design of a hybrid algorithm that integrated the above-described approaches. The intuition is to create an optimization algorithm with two competing loss functions: One that rewards global, integrated ensemble structures and one that rewards accurate representation of firing dynamics during N smaller time frames:

$$\operatorname{argmin}_{W,H} \left(KL(X, H_{week} \cdot W_{week}) + \frac{\sum_n^N KL(X_n, W_n \cdot H_n)}{N} \right) \quad (3.8)$$

(without regularization parameters)

This algorithm uses a global pattern matrix H and the familiar weight matrix W_{week} that encodes the ensemble activity over a whole week alongside $N = 42$ matrices W_{trial} encoding the ensemble structures and $N = 42$ matrices H_{trial} encoding the representative activity patterns of the 42 individual trial recordings throughout the week.

During optimization, the week-wise and trial-wise factorizations are computed iteratively as before. The innovation is to integrate the results after each optimization step using a combined matrix H_{hybrid} . To this end, a concatenated version of the lined-up H_{trials} is linearly combined with the matrix H_{week} of the same size:

$$H_{hybrid} = \frac{H_{week} + [H_{trials}]}{2} \quad (3.9)$$

This newly created H_{hybrid} is used in the following update steps for W_{week} and, re-fragmented to 42 pieces of trial length, the update steps for the different instances of W_{trial} . All W -updates use the same formula as described above:

$$W_{trial[i]} \leftarrow W_{trial[i]} \circ (H_{hybrid[i]} X^T) \circ (H_{hybrid[i]} H_{hybrid[i]} W_{trial[i]}^T + \lambda_{w1} + W_{trial[i]} \lambda_{w2})^{-1} \quad (3.10)$$

$$W_{week} \leftarrow W_{week} \circ (H_{hybrid} X^T) \circ (H_{hybrid} H_{hybrid} W_{week}^T + \lambda_{w1} + W_{week} \lambda_{w2})^{-1} \quad (3.11)$$

The matrices are initiated randomly and a first update step of W_{week} and H_{week} is performed, which renders a rough (but quite accurate) global estimate of ensembles and their activity traces. There follows an update step of all W_{trial} using H_{week} that ensures that the different W_{trial} matrices have a similar initial structure. The result is a number of 42 W_{trial} that resemble the initial W_{week} with slight increases or decreases depending on local within-trial ensemble strength.

The algorithm is then iterated until convergence, resulting in a global ensemble indicator W_{week} and a number of 42 ensemble indicators that contain refined versions of W_{week} for each trial.

Interpreting the significance of such an unsupervised statistical result is not trivial. In our case, the outer product matrix of the ensemble participants (row in W) and the pattern trace (column in H_{week}) represented the part of the entire signal that was modeled by the given ensemble/factor. The l_1 -norm of this product matrix was used as an indicator of the information content of the ensemble. To create a significance measure, a number of factorizations was performed on copies of the datasets with randomly shifted activity traces. From these repeated 'bootstrapping' factorizations, the above described content indicator was computed for

each recovered ensemble. The grey bar histograms in figure 4.19 show the distribution of information content for all generated ensembles. Given the observed shape of the histograms and general interpretative plausibility, the permuted factorization results were modeled as a Gamma process. A Gamma distribution $\Gamma(k, \theta)$ was fitted to the information content indicators (fig. 4.19, black plot). The right tail of the distribution that included 0.01 of the area under the curve (fig. 4.19, green line) was used as the significance threshold. Accordingly, all ensembles detected in the NMF run of the actual dataset with an information content indicator greater than the significance threshold were accepted as genuine and suited for further analysis.

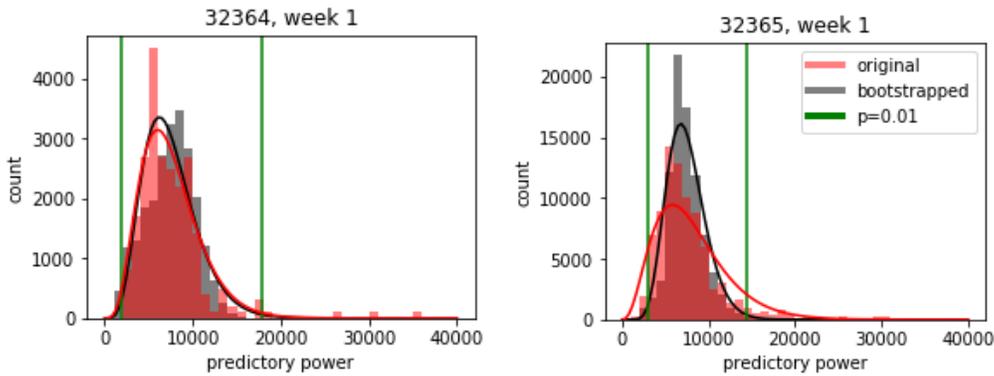


Fig. 4.19: Results from the bootstrapping process of two neural datasets. The grey histograms show the predatory power of all ensembles detected in 100 permuted datasets, the red histograms show the (scaled) predatory power of the ensembles from the actual nmf. Both are approximated using a Gamma distribution. The green lines show the tail points of the Gamma distribution where $\int p = 0.01$. All (red) ensembles on the right of the green line are considered in the further analysis.

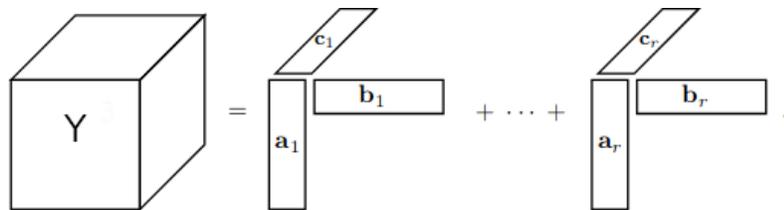
3.3 Variable NMF: A Simulation

To showcase the principle of the hybrid NMF model, a very basic simulation was created using the parameters of the Ca^{2+} recordings at hand.

A dataset of 50 randomly generated neural signals was created. A signal was produced for 6 consecutive trials of 1000 frames each. The covariance matrix of 50 randomly picked recorded activity traces was used as a multivariate Gaussian probability distribution to draw random numbers from and create an artificial data set of 50 activity traces. For two of the traces, the created signal was interpolated between an independent random signal and the above described correlated signal to simulate the entry into/exit from a cellular ensemble over the course of the different trials. Figure 4.20 shows the generated ground truth and the extracted results of the simulation indicating that the implementation of the algorithm is generally functional.

4 Tensor Rank Decomposition of Neural Signals around Exploration Onsets

The Tensor Rank Decomposition algorithms are conceptually related to non-negative matrix factorization, with the difference that the former can process higher dimensional datasets than 2d matrices. In a tensor decomposition, a target tensor is approximated as the linear combination of a number of lower-rank tensor products. [Vannieuwenhoven \(2015\)](#) includes an intuitive (albeit informal) graphical representation of the problem of approximating a target tensor Y with a linear combination of rank-one tensor products $\sum a \otimes b \otimes c$:



There exist multiple algorithms for this optimization problem. A python implementation of the *PARAFAC tensor rank decomposition* ([Bro, 1997](#); [Kossaifi et al., 2016](#)) was used for this project.

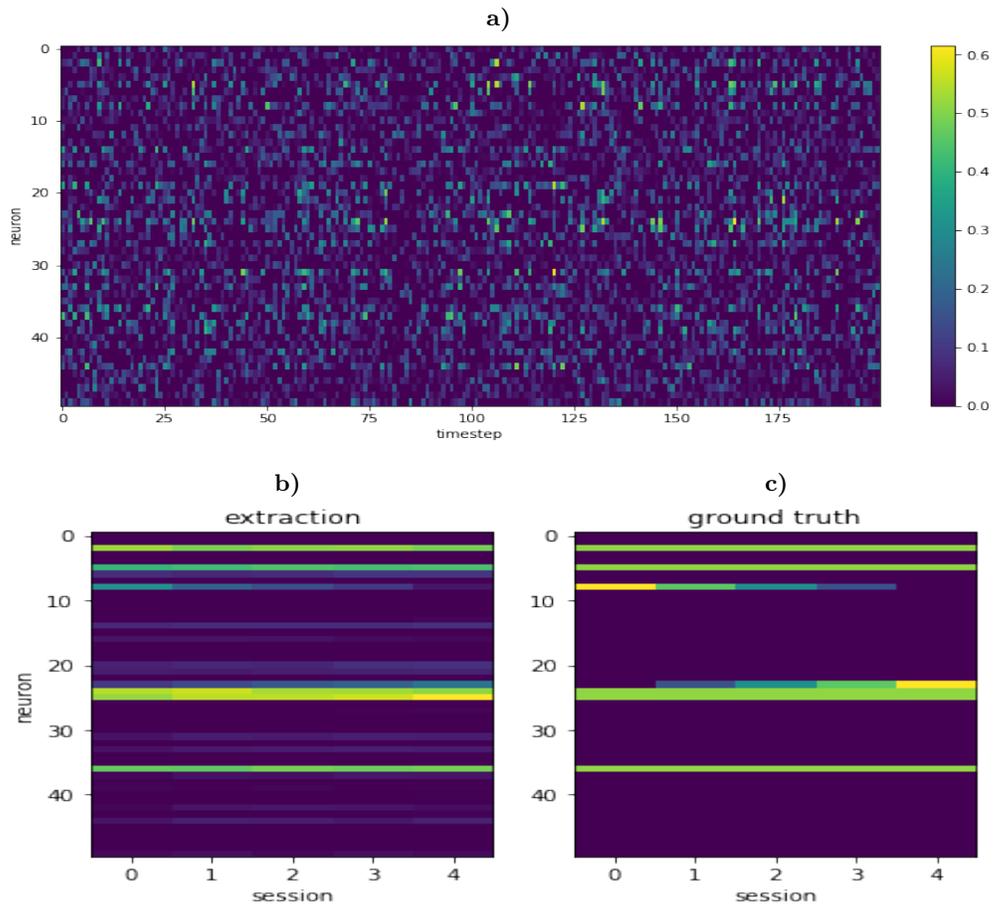


Fig. 4.20: Illustration of an NMF extraction from a simulated neural dataset. (a) shows a cut-out of a simulated Ca^{2+} -imaging dataset using the ground truth (c) with added noise. The variable NMF could extract the trial-wise activity matrix (b).

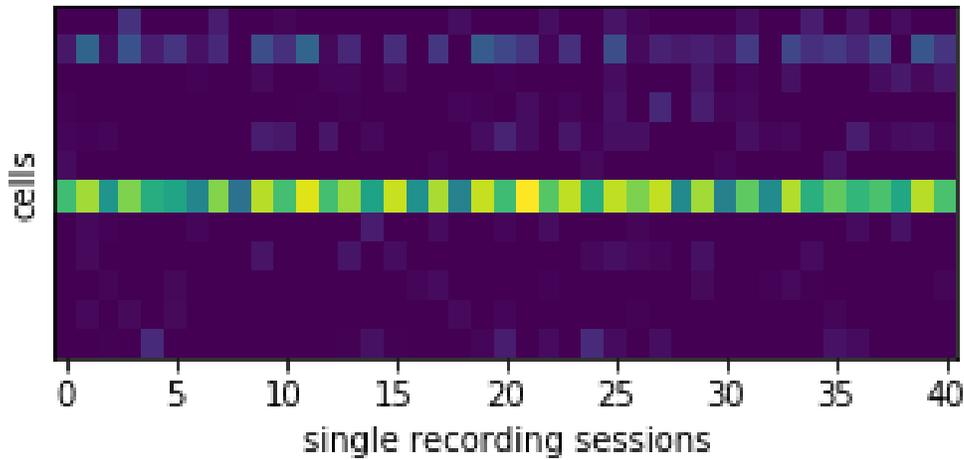


Fig. 4.21: An example of an variable nmf output on actual Ca^{2+} -imaging data: Two cells with partially synchronized firing behavior. The signal synchronization is especially strong during resting recordings (uneven numbers) and at the beginning of each day (every tenth recording).

The target data was created using snippets of the recorded neural activity traces 'around exploration onset'. An 'exploration onset event' happens when the animal starts showing exploratory behavior towards one of the objects. The extracted signal of neurons from a time frame from $600ms$ before to $300ms$ after exploration onsets were extracted from the recordings. These extracted signals were then stacked to form a 3d tensor of the shape [onset_events, n_cells, n_frames]. This tensor was passed to a PARAFAC decomposition function with an arbitrary factor depth of 10. The results of the decomposition will be discussed more in detail in the following chapter on an actual case study. Figure 4.22 gives a brief intuition of the nature of the results.

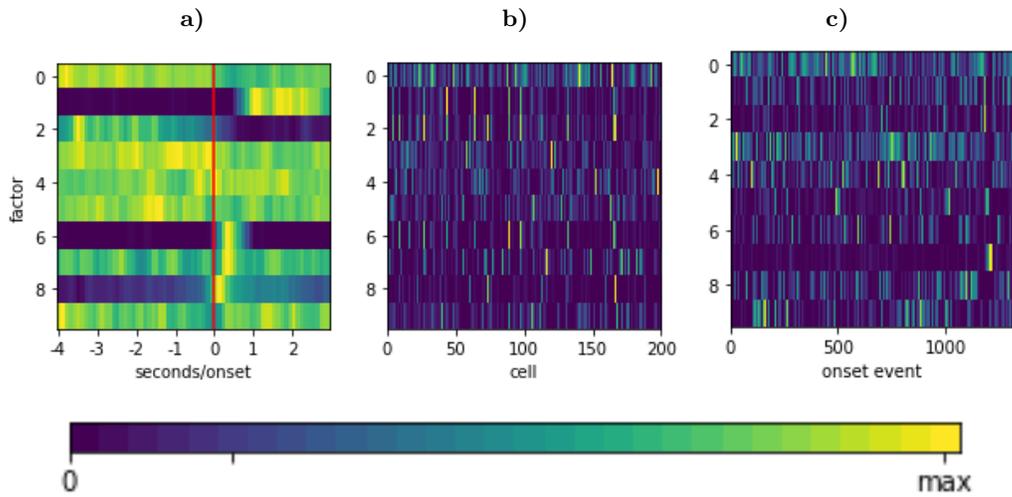


Fig. 4.22: The result of a depth-10 tensor rank decomposition: The three subfigures show the three dimensions: (a) contains the activity traces around exploration onset, (b) contains the exploration onset events and (c) displays participation of the different cells. The columns contain the 10 different rank-one tensor outputs. The result can be read as 'The very salient trace of factor six (i.e. (a), row 6) shows a clear peak around the onset time at 0 seconds. An activity trace of this shape appears noisily in exploration onset events throughout all week without a clear upward or downward trend ((b), row 6). An ensemble of 6-10 neurons mainly contributes to the activity trace ((c), row 6)'.

Chapter 5

Ca^{2+} -Imaging for the Object Space Task: A Pilot.

1 Introduction

The Object Space Task (OST, [Genzel et al. \(2017\)](#)) is the behavioral rodent experiment that provides the base for this entire project. The OST was originally created to explore memory consolidation in rodents without having to rely on external motivation. During the task, the rodents are repeatedly made to explore an environment with a total of two objects in it. The rodents' natural curiosity typically lets the animals move towards and explore a novel or repositioned object and ignore other, established properties of the environment. To check whether the animals transfer episodic knowledge about the most recent experienced trials into more abstract semantic knowledge, objects are repositioned across trials according to a rule that the rodents can in principle learn - namely that one object is moved whereas the other one remains stationary at all times. In a final trial, where no object is repositioned, animals that have formed a semantic representation of the repositioning rules are expected to favor the mobile object (that should typically be repositioned according to the rule). If the animals instead are indifferent towards both objects (neither of which changed position in the last trial), it would show that animals evaluate their environment on a shorter, trial-to-trial based time scale and that no semantic representations were formed. Next to this 'overlapping' condition of a stable and a moving object, the experiment also includes two control conditions called 'stable' (two stable objects) and 'random' (two moving objects).

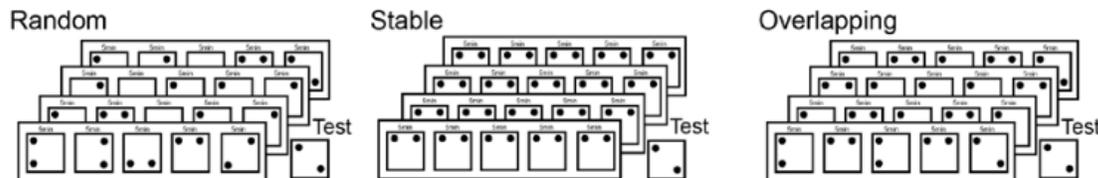


Fig. 5.23: A graphical depiction of the Object Space Task taken from (Genzel et al., 2017) that shows the procedures for each of the categories. Five 5-minute trials are completed over four consecutive days. The object repositioning patterns differ between categories.

2 Methods

The methodological set-up of the behavioral task is described in detail in Genzel et al. (2017). Here, I will render a shortened version of the preparation and execution of the experiment from the paper:

The animals used in the study were male C57BI6/J mice. All animals were habituated to human contact and the experimental environment for a number of weeks before the actual start of the experiments. The environment itself was a square arena of $75\text{cm} \times 75\text{cm}$. Walls and floor of the environment were white to facilitate automated video analysis. As described above, the object space task consists of three conditions: stable, overlapping and random. Conditions and locations were counterbalanced among animals and trials and the experimenter was blinded to the condition. At the beginning of each trial, cues were placed on the walls inside the box and at least one 3D cue was placed above one of the other walls. Cue distribution was intentionally non-symmetric. A camera was placed above the box to record every trial and to allow for online scoring of exploration time. In each condition, animals were allowed to explore two objects for five minutes with an inter-trial interval of 30min. Ca^{2+} -footage was recorded during the entire five minutes of the testing phases and during a five-minute phase of each resting interval. Mice were trained interleaved in groups of 4 with two groups per day (morning/afternoon).

Before the beginning of each sample trial, the box and the objects were thoroughly cleaned with 70% ethanol. Each sample trial consisted of a different pair of matching objects varying in height, width, texture and material (including metal, glass, plastic and wood). Object sizes ranged from 4-26cm in height to 5-18cm in width. Objects were glued onto metal coasters and placed onto the magnets that were fixed on the floor of the arena. Objects were never repeated during the training period of one condition (1 trial). This procedure was repeated over the course of 4 consecutive days in which they were presented with 5 sample trials per day, thus accumulating to 20 total sample trials. The test trial, 24hrs after the last sample trial, consisted of another trial of two objects that animals were allowed to explore for 10 minutes, of which only the initial 5 minutes were used in the analysis. Ca^{2+} -imaging was recorded during five more minutes in a subsequent resting phase.

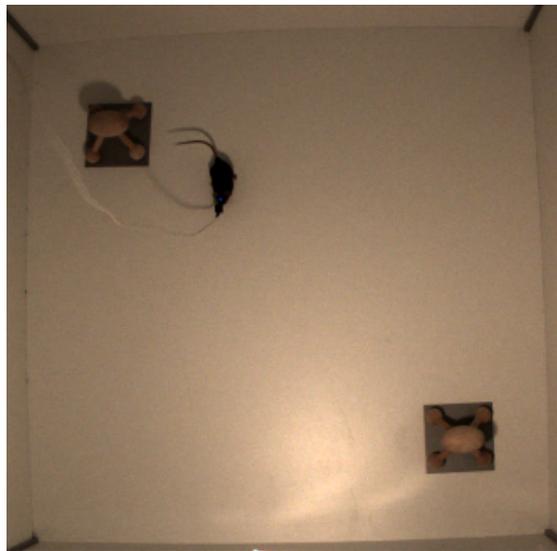


Fig. 5.24: The set-up of the experiment during a test trial. The animal is allowed to freely explore the environment containing two objects while Ca^{2+} -imaging footage is recorded.

In total, 6 recording weeks were used for this pilot analysis, balanced between the three conditions 'overlapping', 'random', and 'stable'. Accordingly, the dataset

consisted of six data points, each of which contained Ca^{2+} -imaging footage of 20 5-minute testing trials and 20 5-minute resting phases and the same amount of synchronized behavioral camera footage.

The individual data points were then processed using the pipelines described in the previous chapters. The Ca^{2+} -imaging footage was aligned and factorized using a CNMF algorithm (chapter 2), the behavioral footage was analyzed using a deep learning ensemble (chapter 3) and the results were analyzed using an unsupervised hybrid-NMF model, a tensor rank decomposition of neural data during exploration onsets and a simple linear regression model on the neural and behavioral data sets (chapter 4).

3 Results

3.1 Ca^{2+} -imaging analysis

After a prolonged period of parameter optimization and code adaptation, the CNMF pipeline could successfully process the footage from six complete recording weeks, with an execution time of one to two days each. Table [1] shows a summary of the results for each pipeline execution. The column 'ROIs' in Table [1] shows a summary of the detected cells before a local correlation background estimate. It shows that most detected sources are plausible independent cells and that other correlated signal sources (such as blood vessels) are successfully excluded. The density of cells in the recorded footage differs from animal to animal, with all recordings of animal 32365 showing a cell count a few magnitudes higher than the other animals 32364 and 56165.

The plot 'correlation over distance' in Table [1], column 4 gives an estimate of the correlation of the distance between cell pairs on one hand and the correlation of their activity traces on the other hand. The fact that proximate cells have a higher activity correlation can indicate a deficit in the segmentation algorithm. This 'upwards tail' on the left side of the clusters is visible for every pipeline execution and

could not be eliminated. It simply needs to be accepted as a systematic pollution of the signals in the further statistical analysis.

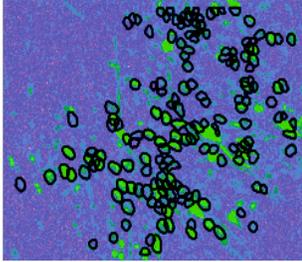
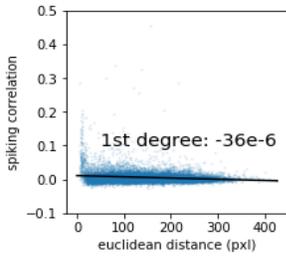
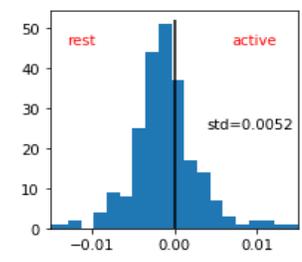
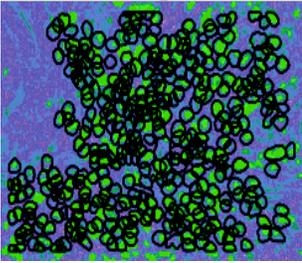
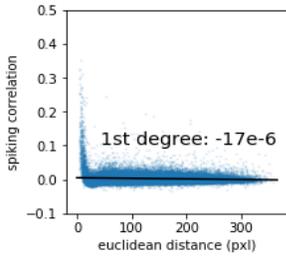
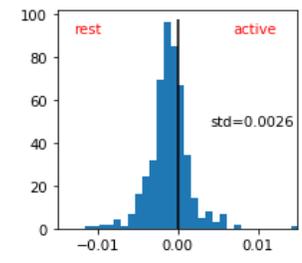
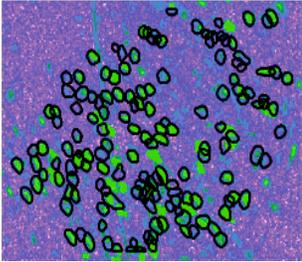
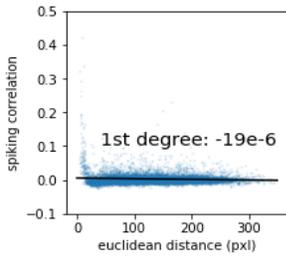
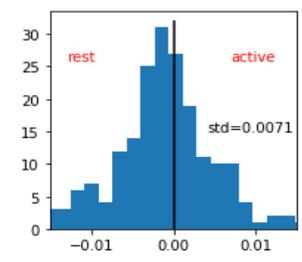
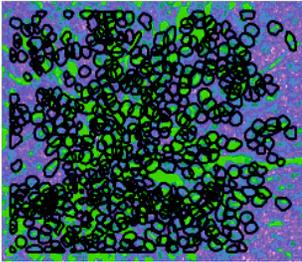
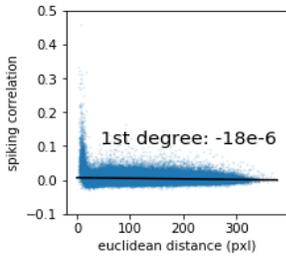
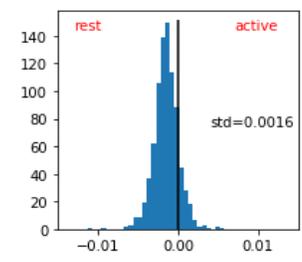
The last column of Table [1] shows a histogram of all cells based on the activity differences between active and resting phases. Cells in the center show no activity preference towards active and passive recording trials. The left tail of the distribution shows cells with a resting phase preference, the right tail shows cells with a test phase preference. The shape of the resulting distribution can be used as a first indicator of information content: A broader distribution denotes a low-entropy signal with many cells carrying information about the current recording state. A narrow distribution however denotes a signal with low information content, be it due to the cells being functionally independent or due to noisy and polluted signal recordings.

3.2 Behavior

The behavioral pipeline could be applied smoothly to all corresponding behavioral videos. Table [2] gives an overview of the extracted behavioral data.

The column '⊗ location' of Table [2] shows a two-dimensional histogram representation of the arena as displayed in Figure 5.24. The color brightness denotes the visiting frequency of a certain location, with exploration time of Object 1 and 2 represented in the red and green color channels respectively. Locations with idle behavior are displayed in white. Before all, the color markers show that the two behavioral and location detection networks produce a plausible, well-synchronized data stream. The exploration data represents the positions of the objects in the arena for all categories. It is apparent that the animals spend an above-average amount of time exploring the objects in the arena. Also, the corners of the arena are frequently visited as a hiding spot that resembles the animal's dense natural habitat.

The columns 'exploratory behavior' of Table [2] display the percentage of total time spent for exploration of the two objects. The column ' $\frac{O_2 - O_1}{O_2 + O_1}$ ' shows the

animal	week	cell count	ROIs	correlation/distance	rest/test preference
Condition: Overlapping					
32364	1	237			
32365	3	481			
Condition: Stable					
32364	2	200			
32365	1	847			

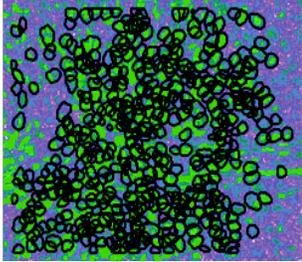
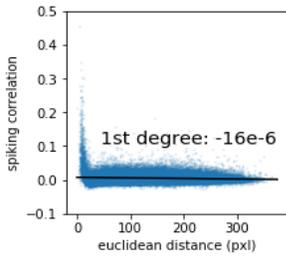
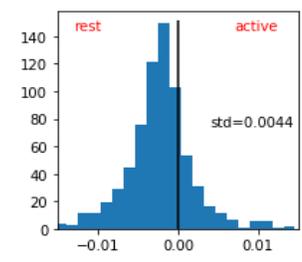
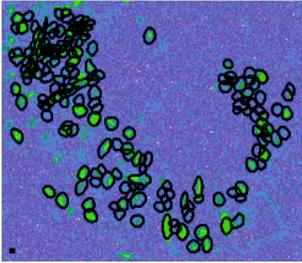
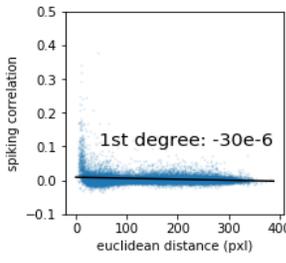
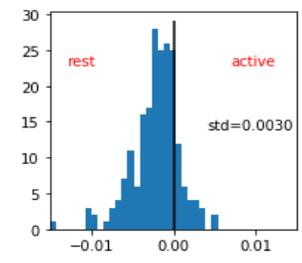
Condition: Random					
animal	week	cell count	ROIs	correlation/distance	rest/text preference
32365	2	713			
56165	4	207			

Table 1: A summary of the CNMF-pipeline outputs. Each row shows the result of one pipeline execution. Column 1 and 2 show the animal id and week of recording. Column 3 shows the total count of signal sources found in the footage. Column 4 shows the outline of all detected signal sources laid over a local pixel value correlation background. Column 5 shows the activity Correlation of each cell pair over the distance as a measure of source separation quality. Column 6 shows a histogram of activity preference for the firing behavior of each cell.

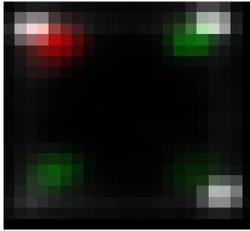
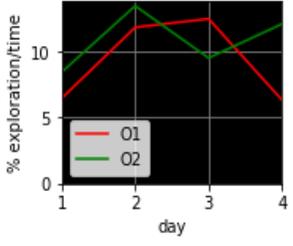
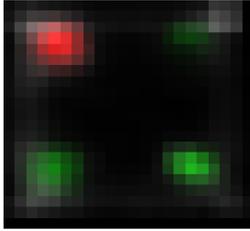
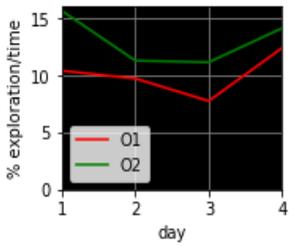
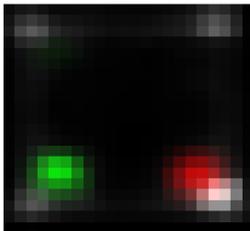
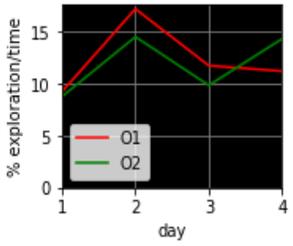
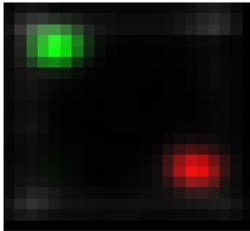
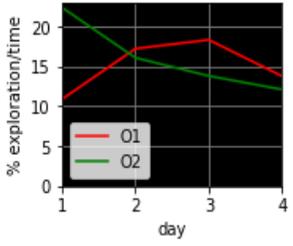
difference of relative object exploration times with a positive number indicating a preference for object '2' and a negative number indicating a preference for object '1'. The column ' \ominus exploration time/day' breaks down the exploration percentages per object for each day throughout the four days of the recording week.

As for the results, for the overlapping condition, the motile object '2' is frequented more often than the stable object '1' in seven out of eight recording days (Table 2, ' \ominus exploration time/day'). Also, the two overlapping recording weeks show the highest preference ratios towards object '2' (Table 2, ' $\frac{O_2-O_1}{O_2+O_1}$ '), with a T-test for independence between the categories "overlapping" and the two other categories rendering a p-value of 0.067. All in all, the behavioral results seem plausible and in line with previous publications on the OST, but not significant given the small sample sizes and the large variance within conditions.

3.3 Supervised Statistics

A random permutation/bootstrapping analysis was performed as a functionality test for the entire pipeline. To this end, a total of 1000 linear least-squares fits were performed on the neural data and a generated set of randomly shifted object exploration data. Figure 5.26 shows the results of these bootstrapping trials as histograms of the mean-squared errors along with the mse of the un-shifted linear regression fit. The sigma values for all random permutation tests are below 0.01, with animal 32365 having a much lower significance values than the other animals.

Next to the permutation test, linear regression models were fitted on neural data and a number of systematically shifted behavioral datasets as described in chapter 4, with the mean-squared error as a goodness-of-fit indicator. Figure 5.25 a) shows the relative decrease of the mean-squared error as compared to a baseline value (acquired from randomly shifted datasets in the bootstrapping process). As to be expected, linear regression fits are the most accurate around a shift of 0 ms, when the neural data and the behavioral data are synchronized. Figure 5.25 b) is a depiction of the same data zoomed around the origin, which shows that the

animal	week	exploratory behavior			⊙ Location	⊙ exploration time/day
		object 1	object 2	$\frac{O_2 - O_1}{O_2 + O_1}$		
Condition: Overlapping						
32364	1	9.89%	10.72%	0.0838		
32365	3	11.54%	13.03%	0.1290		
Condition: Stable						
32364	2	12.12%	12.00%	-0.0101		
32365	1	15.17%	15.72%	0.0358		

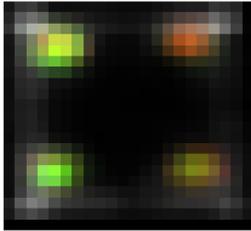
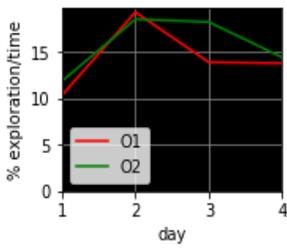
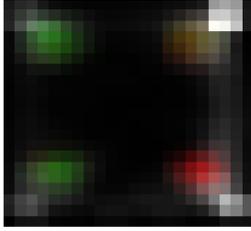
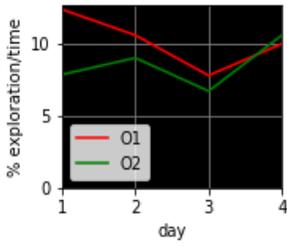
Condition: Random						
animal	week	exploratory behavior			⊙ Location	⊙ exploration time/day
		object 1	object 2	$\frac{O_2 - O_1}{O_2 + O_1}$		
32365	2	15.00%	15.71%	0.0475		
56165	4	9.26%	8.61%	-0.0702		

Table 2: A summary of the behavioral pipeline outputs. Each row shows the result of one pipeline execution. Column 1 and 2 show the animal id and week of recording. The following three columns show the percentage of time spent on exploring objects one and two next to an indicator for preference disparity (positive \rightarrow object two). Column '⊙ Location' depicts a visualization of the arena from above with the visiting frequency for each location indicated by the color brightness. Exploration periods for objects one and two are rendered in the red and green color channels, whereas neutral behavior is depicted as white. The last column shows a day-wise breakdown of the exploratory behavior towards the different objects.

best linear regression fits typically do not occur at the synchronized frame, but for regression models that correlate the current neural signal against the behavior one or two frames (10-20 ms) ahead.

Figure 5.25c) further indicates the intentional, future-oriented nature of the recorded neural signal: It shows the difference between the right half and the left half of the graphs in Figure 5.25a), starting from the origin. Intuitively, the plots in c) show, from the current neural signal, how much easier it is to predict the behavior of x milliseconds in the future than x milliseconds in the past. The plots illustrate that the neural signal has a bias towards future behavioral states in most of the recorded animals, especially in low-noise recordings of animals 32364 and 56165.

In summary, the supervised analysis could show that the neural signal in all animals contains behavioral information. This information is conserved throughout the pipeline and can be retrieved using a simple linear regression model. The information within the neural traces as well as the mutual information between the traces and the behavior differs between the different recorded data points. Interestingly, the statistical parameters of the recorded data points continuously show a low within-animal variance and a comparatively high within-behavioral-category variance. Unfortunately, this characteristic of the dataset impedes small sample statistical analysis that would be necessary for conclusions specific to the object space task.

3.4 Unsupervised Statistics

Most unsupervised ensemble clustering methods described in the literature were developed for the more temporally precise electrophysiology-based recording techniques that arguably produce a much lower entropy signal than the one-photon Ca^{2+} -recordings recorded during this project.

The NMF-based clustering algorithm described in chapter 4 could produce a some sensible output when applied to the Ca^{2+} -imaging data at hand without

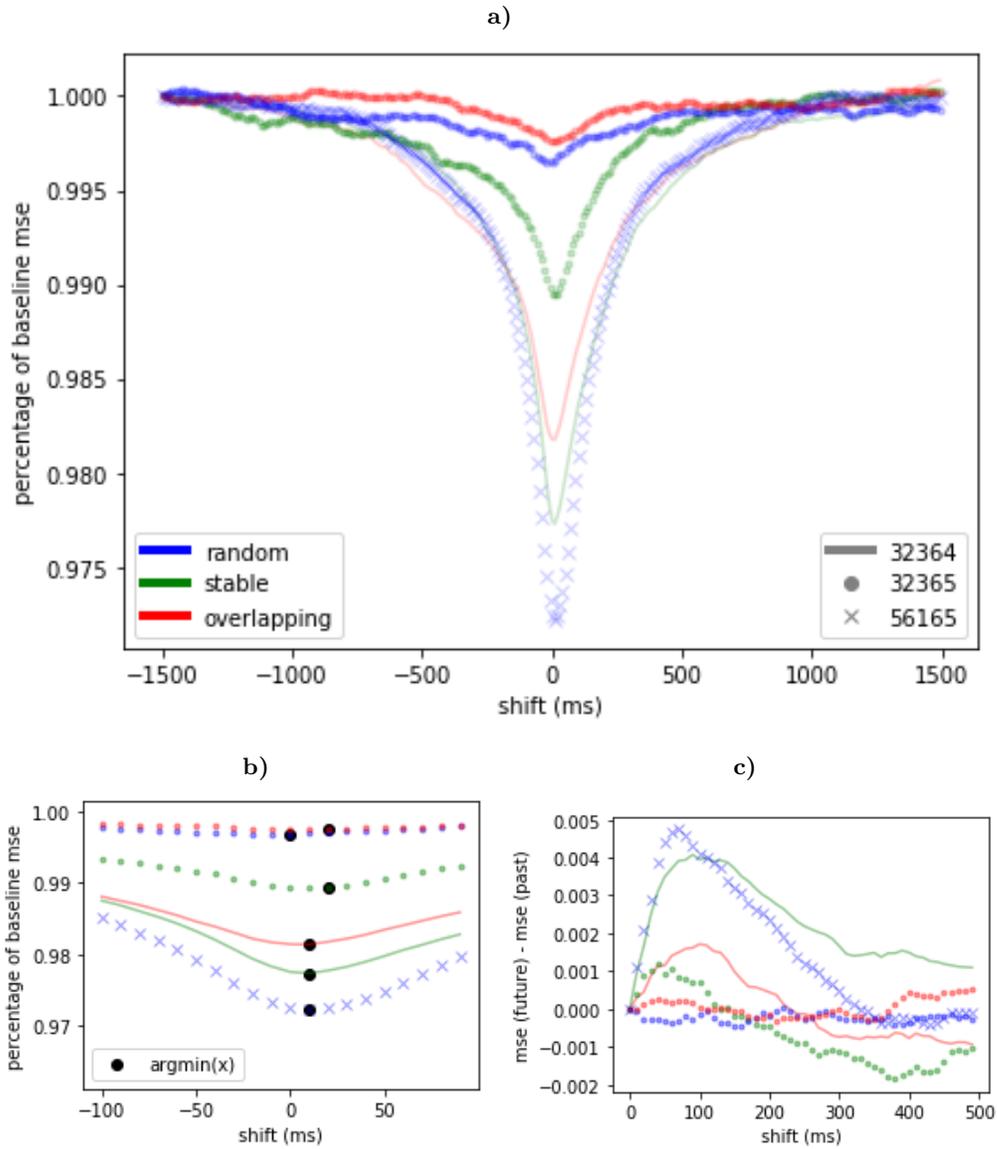


Fig. 5.25: The results from a linear regression analysis of the shifted neural dataset. **a)** shows the decrease of the mean squared error of a linear regression fit compared to a baseline value where the x-axis indicates the shifts of the neural data. As to be expected, the error is the lowest around a shift of 0 ms, when the neural data is correlated to the synchronous behavior. Error increases when the regression is performed on behavioral data from the past/future, just to reach a baseline error value at around +/- 1000ms. **b)** shows that the peak of the curves in **a)** do not match current behavior but rather the behavior of 15-20 ms ahead for most data-points. **c)** further shows that the bell curves from **a)** are tilted towards future behavior: Subtracting the mirrored left half from the right half of the curves in **a)** renders curves with a positive slope at the origin.

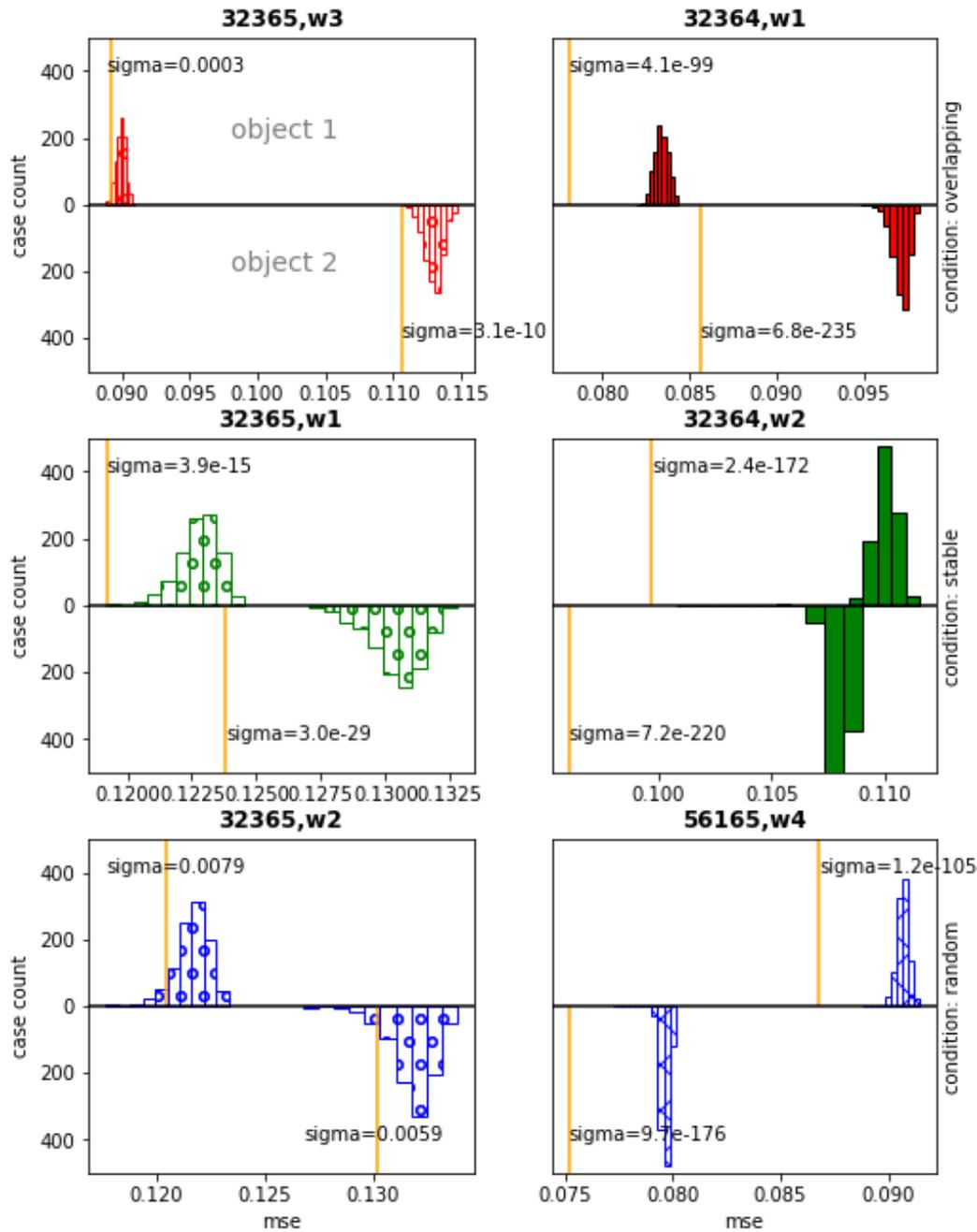


Fig. 5.26: Object-wise back-to-back histograms of the mse-values of 1000 bootstrapped linear regression fits of the neural signal with a randomly shifted exploratory behavior. The yellow lines depict the mse of the linear fit for non-shifted behavioral data. The sigma value is below 0.01 for each of the data points. The sigma values for animal 32365 are significantly higher than for the other animals, again indicating a lower information content in the signal. Color encodes category (red \rightarrow overlapping, green \rightarrow stable, blue \rightarrow random) and patterns encode animals.

rendering a definitive result on ensemble formation during learning in the object space task. The binarized spiking data of each recording week was factorized independently. As described in chapter 4, the NMF produced a global weight matrix W , a trial-wise weight matrix W_s and a trace pattern matrix H . A bootstrapping algorithm was used to determine ensemble significance.

Between 1 and 20 (animal-wise) ensembles could be recovered from the Ca^{2+} -imaging recordings. For all animals and each extracted ensemble, a spatial activity histogram was created by dividing the arena into $(21 \cdot 21)$ squares and by determining the average ensemble activity of all frames during which the animal is located within the square. Upon inspection, many of the so produced spatial histograms featured a number of basic activity patterns. Subjecting the recovered spatial histograms to another round of non-negative matrix factorization could nicely reproduce these observable patterns as the five largest recovered factors (Table 3, column 'spatial histograms'). The trial-wise activity matrices W_s of all ensembles contributing to each the template patterns were subjected to a last round of NMF in order to extract prototypical developments of ensemble strength during the recording weeks. Expected developments would for example be 'stable, increasing, decreasing, test-biased, rest-biased', all of which could be observed in the different factors (Table 3, column 'typical activities/week').

3.5 Tensor Decomposition

A summary of the tensor decomposition results for each recording week are depicted in table 4. It is apparent that each of the recording weeks contains neural signals that can in some way be related to exploration onset behavior. Again, the animals 32364 and 56165 appear to have a more salient signal than animal 32365. Much like the detected spatial histograms in the 'unsupervised statistics' pipeline, the discovered traces from the tensor decomposition (Table 4, column 'activity traces') can be subjected to an unsupervised classification using an NMF algorithm. The extracted activity traces around exploration onset from all animals are depicted in

figure 5.27. Much like the recovered ensemble activities from the previous section, the strongest pattern is a stable baseline pattern with a slight decrease of activity at $t_{onset} = 0$. Other traces contain a sharp activity burst shortly before, during or after exploration onset (Figure 5.27, traces 3,5,8) and traces that show stable activity before exploration onset and fall silent afterwards (traces 2,4) and vice versa (trace 9).

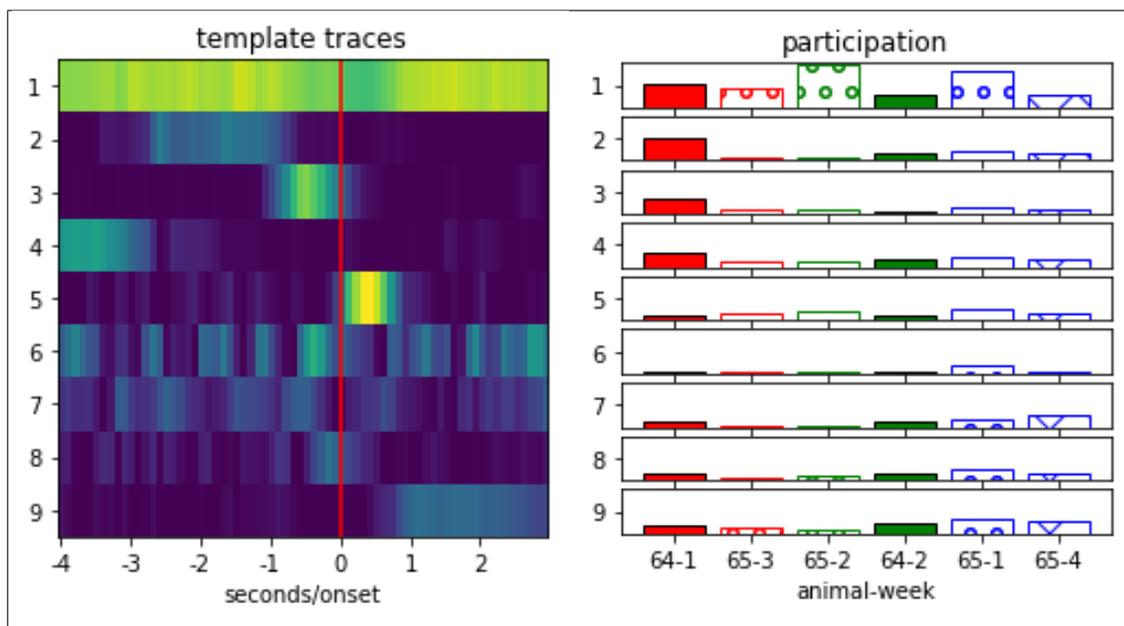


Fig. 5.27: The output from a second round of unsupervised classification on the extracted onset activity traces from all animals (as depicted in Table 4, column 'activity traces'). The Figure 'participation' shows the participation of the different recording weeks in the depicted template traces.

4 Conclusion

This chapter describes the results of the pipeline applied to a total of six data points of neural and behavioral data collected during in mice performing the object space task.

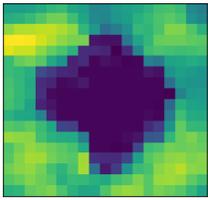
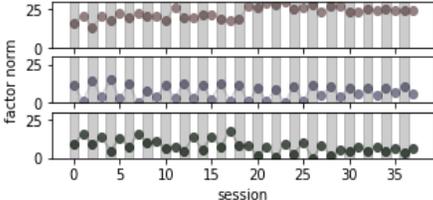
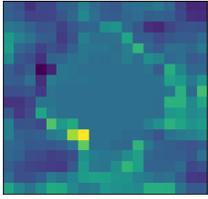
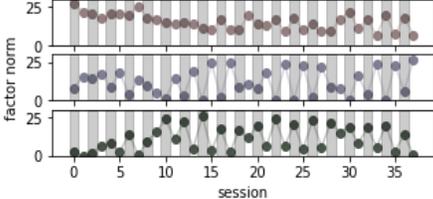
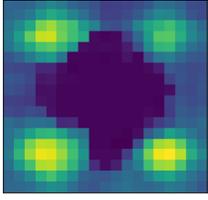
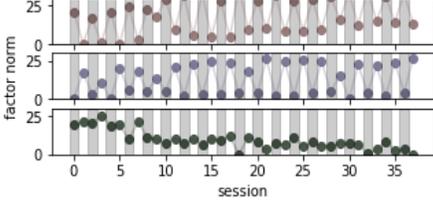
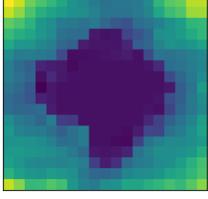
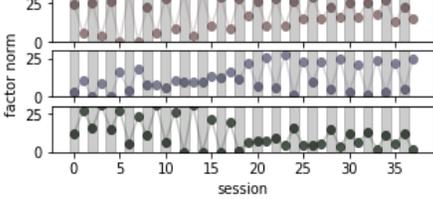
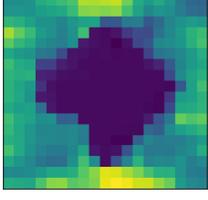
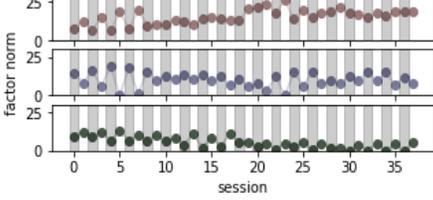
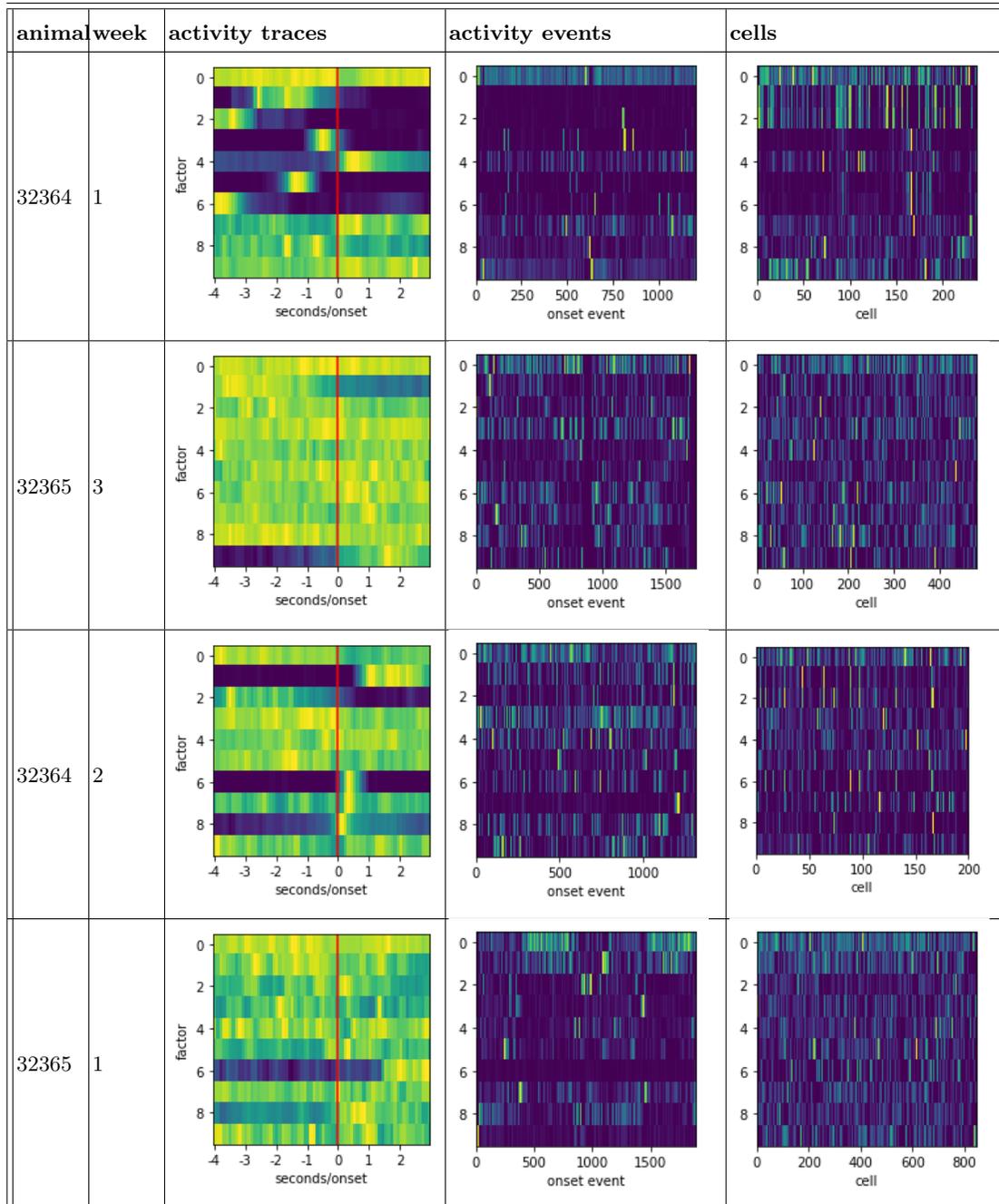
Tag	spatial activity	Typical activities / week	description
'baseline'			A baseline cell activity that is relatively stable throughout all types of behavior and through active and resting phases.
'roaming'			Ensembles encoding active roaming inside the cage.
'exploring'			Encoding object exploration. Interestingly, there exists a type of 'exploratory ensembles' with clear resting-phase preference (2 nd factor norm plot).
'hiding'			Encoding hiding in the cage corners.
'sneaking'			Encoding thigmotaxis/sneaking along the edges of the arena.

Table 3: A number of typical ensembles to be found in the spike traces of all animals. Neural activity histograms over spatial location from the detected ensembles from all recordings were subjected to a round of unsupervised classification. The first five extracted template histograms are shown in the column 'activity map'. The trial-wise activity factors (obtained through the original NMF) of all ensembles belonging to a certain category are then reduced to 3 dimensions to create a few template temporal activity sequences, which are depicted in the column 'Typical activities/week'.



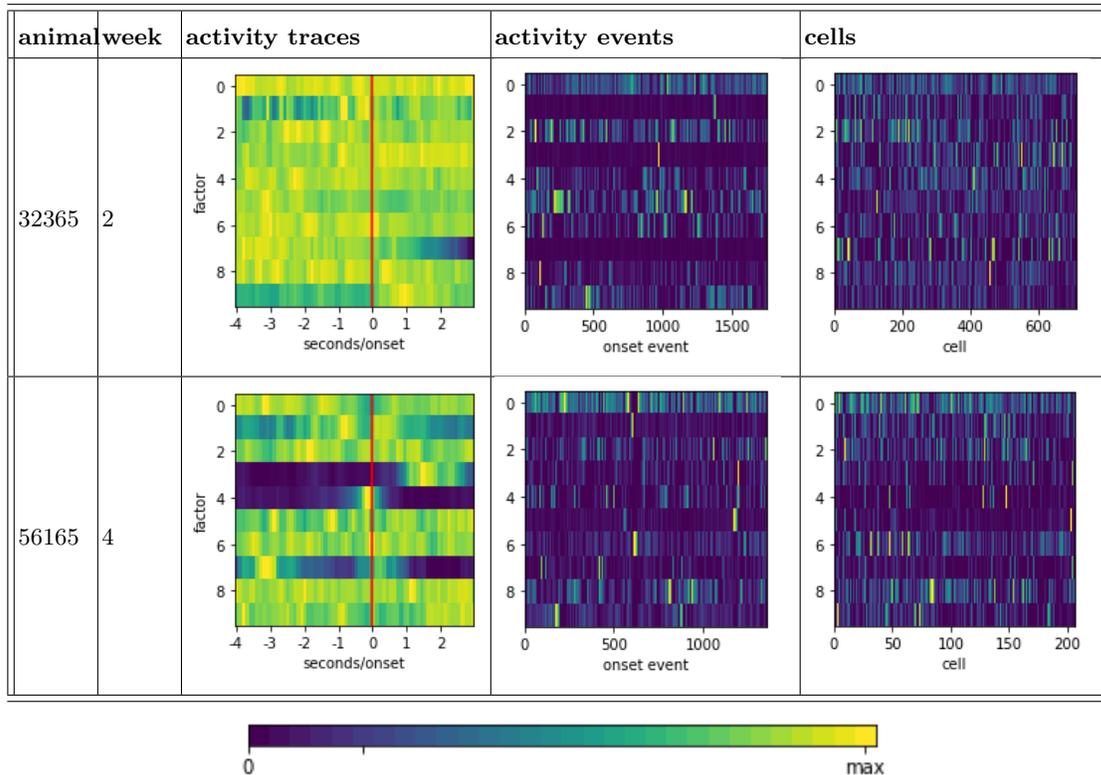


Table 4: An overview of the tensor decompositions from the different analyzed animals. The figures in the column 'activity traces' show the extracted template traces around exploration onset (red line). For the interpretation of the figures, consider Figure 4.22. It is apparent that, throughout all animals and weeks, there exist neural ensembles that are associated to exploration onset. Figure 5.27 shows the result of a round of unsupervised classification on the activity traces of all animals.

Using supervised statistical analysis it could be shown that the recorded , prefrontal neural signal contains a significant amount of information about observable animal behavior. Moreover, it could be shown that the neural signal correlates more with future behavioral states than past behavioral ones ('Granger causality'), which is in line with the involvement of the prefrontal cortex in scheduling and planning.

Using methods of dimensionality reduction for ensemble detection, it could be shown that there exist ensembles that encode certain types of typical, semi-abstract behavior (such as object exploration, roaming or hiding). Interestingly, the analysis produced a hint of an ensemble class that encodes object exploration and is mainly active during resting phases, which could be an indicator of neural replay.

A tensor decomposition analysis could show that there exist a number of archetypal activity traces that encode exploration onset throughout the different animals and trials, again without a clear interpretable result concerning the object space task.

Chapter 6

Conclusion

The goal of this project was to establish a Ca^{2+} -imaging analysis pipeline that can be used to investigate the neural correlate of memory consolidation in rodents during the object space task. In the end, a fully-automated pipeline could be created to process a data set of neural Ca^{2+} -imaging footage and synchronised behavioral videos.

As a functioning test and for a first exploratory investigation of the dataset, a number of statistical tests were also created and performed on a total of six collected data points.

For all data points, a complete, week-long neural signal could be extracted successfully. A few measures of information content of the signal differed substantially between animals, possibly due to flaws in the factorization algorithm in separating spatially overlapping signal sources in the recordings of animal 32365.

Behavioral data could also be extracted successfully using a deep learning set-up. The results produced from the behavioral pipeline were conclusive with former behavioral findings from the object space task.

Supervised statistical analysis could confirm the general functionality of the pipeline by showing a significant correlation between the outputs from the Ca^{2+} -imaging signal extraction pipeline and the behavioral analysis pipeline in a statistical random perturbation test.

A new unsupervised statistical method of dimensionality reduction was also developed and applied to the dataset. The algorithm can model the test/rest period preference that is characteristic of the neural signal during the object space task. Moreover, the method can be used to investigate formation of cellular ensembles

over the course of an entire week, which is a main hypothesis for a neural correlate of memory formation during the object space task.

The Ca^{2+} signal source is temporally coarse-grained by nature and the extent of the pipeline produces a relatively abstract and interpretable signal. Therefore, any insight on neural dynamics to be gained from this pipeline will most likely be stochastic rather than observational-phenomenological in nature. On the upside, this exploratory study could show that there exist neural correlates of relevant behavior in the recoverable Ca^{2+} signal that, with a suitably large set of data points, is ready for a between-category statistical analysis. Unsupervised analysis of ensemble formation could also produce significant and reasonable results. But, in order to statistically analyse the subtle behavioral dynamics of the object space task and relate it to a neural correlate, it presumably is necessary to expand the analysis to a larger dataset beyond only two data points per category, especially given the large observable between-animal variance of the dataset. Such a dataset is available at the institute ready to be analyzed to produce further insight on the neural correlates of memory consolidation during the object space task.

Bibliography

- Bro Rasmus*. PARAFAC. Tutorial and applications // Chemometrics and intelligent laboratory systems. 1997. 38, 2. 149–171.
- Carreira Joao, Zisserman Andrew*. Quo vadis, action recognition? a new model and the kinetics dataset // Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on. 2017. 4724–4733.
- Cichocki Andrzej, Phan Anh-Huy*. Fast local algorithms for large scale nonnegative matrix and tensor factorizations // IEICE transactions on fundamentals of electronics, communications and computer sciences. 2009. 92, 3. 708–721.
- Imagenet: A large-scale hierarchical image database. // . 2009.
- Duckworth Daniel*. PyKalman. 2012.
- Ehret Benjamin*. Movie Analysis Pipeline. 2018. url: <https://git.ee.ethz.ch/behret/MovieAnalysis>, accessed on 2019-01-10.
- Févotte Cédric, Idier Jérôme*. Algorithms for nonnegative matrix factorization with the β -divergence // Neural computation. 2011. 23, 9. 2421–2456.
- Friedrich Johannes, Paninski Liam*. Fast active set methods for online spike inference from calcium imaging // Advances In Neural Information Processing Systems. 2016. 1984–1992.
- Genzel Lisa, Schut Evelien, Schroeder Tim, Eichler Ronny, Bayraktar Gulberk, Cornelisse Nikkie, Gareth Hussein, Giuliani Federico, Gomez Angela, Hulzebos Sidney, others* . The Object Space Task for mice and rats // bioRxiv. 2017. 198382.
- Grienberger Christine, Konnerth Arthur*. Imaging calcium in neurons // Neuron. 2012. 73, 5. 862–885.
- Guizar-Sicairos Manuel, Thurman Samuel T, Fienup James R*. Efficient subpixel image registration algorithms // Optics letters. 2008. 33, 2. 156–158.

- He Kaiming, Zhang Xiangyu, Ren Shaoqing, Sun Jian.* Deep residual learning for image recognition // Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. 770–778.
- Insafutdinov Eldar, Andriluka Mykhaylo, Pishchulin Leonid, Tang Siyu, Levinkov Evgeny, Andres Bjoern, Schiele Bernt.* Arttrack: Articulated multi-person tracking in the wild // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 4327. 2017.
- Insafutdinov Eldar, Pishchulin Leonid, Andres Bjoern, Andriluka Mykhaylo, Schiele Bernt.* Deepercut: A deeper, stronger, and faster multi-person pose estimation model // European Conference on Computer Vision. 2016. 34–50.
- Kalman Rudolph E, Bucy Richard S.* New results in linear filtering and prediction theory // Journal of basic engineering. 1961. 83, 1. 95–108.
- Kossaiji Jean, Panagakis Yannis, Pantic Maja.* Tensorly: Tensor learning in python // arXiv preprint arXiv:1610.09555. 2016.
- Lee Daniel D, Seung H Sebastian.* Algorithms for non-negative matrix factorization // Advances in neural information processing systems. 2001. 556–562.
- Mackevicius Emily L, Bahle Andrew H, Williams Alex H, Gu Shijie, Denissenko Natalia I, Goldman Mark S, Fee Michale S.* Unsupervised discovery of temporal sequences in high-dimensional datasets, with applications to neuroscience // bioRxiv. 2018. 273128.
- Mathis Alexander, Mamidanna Pranav, Cury Kevin M., Abe Taiga, Murthy Venkatesh N., Mathis Mackenzie W., Bethge Matthias.* DeepLabCut: markerless pose estimation of user-defined body parts with deep learning // Nature Neuroscience. 2018.
- Peyrache Adrien, Benchenane Karim, Khamassi Mehdi, Wiener Sidney I, Battaglia Francesco P.* Principal component analysis of ensemble recordings reveals cell assemblies at high temporal resolution // Journal of computational neuroscience. 2010. 29, 1-2. 309–325.

- Pneumatikakis Eftychios A, Gao Yuanjun, Soudry Daniel, Pfau David, Lacefield Clay, Poskanzer Kira, Bruno Randy, Yuste Rafael, Paninski Liam.* A structured matrix factorization framework for large scale calcium imaging data analysis // arXiv preprint arXiv:1409.2903. 2014.
- Pneumatikakis Eftychios A, Giovannucci Andrea.* NoRMCorre: An online algorithm for piecewise rigid motion correction of calcium imaging data // Journal of neuroscience methods. 2017. 291. 83–94.
- Pneumatikakis Eftychios A, Soudry Daniel, Gao Yuanjun, Machado Timothy A, Merel Josh, Pfau David, Reardon Thomas, Mu Yu, Lacefield Clay, Yang Weijian, others .* Simultaneous denoising, deconvolution, and demixing of calcium imaging data // Neuron. 2016. 89, 2. 285–299.
- Santos Vitor Lopes-dos, Conde-Ocazonez Sergio, Nicoletis Miguel AL, Ribeiro Sidarta T, Tort Adriano BL.* Neuronal assembly detection and cell membership specification by principal component analysis // PloS one. 2011. 6, 6. e20996.
- Silberman Nathan, Guadarrama Sergio.* Tensorflowslim image classification model library. 2017.
- Szegedy Christian, Liu Wei, Jia Yangqing, Sermanet Pierre, Reed Scott, Anguelov Dragomir, Erhan Dumitru, Vanhoucke Vincent, Rabinovich Andrew.* Going deeper with convolutions // Proceedings of the IEEE conference on computer vision and pattern recognition. 2015. 1–9.
- The tensor rank decomposition. // . 2015.
- Vogelstein Joshua T, Packer Adam M, Machado Timothy A, Sippy Tanya, Babadi Baktash, Yuste Rafael, Paninski Liam.* Fast nonnegative deconvolution for spike train inference from population calcium imaging // Journal of neurophysiology. 2010. 104, 6. 3691–3704.
- Zhou Pengcheng, Resendez Shanna L, Rodriguez-Romaguera Jose, Jimenez Jessica C, Neufeld Shay Q, Stuber Garret D, Hen Rene, Kheirbek Mazen A, Sabatini Bernardo L, Kass Robert E, others .* Efficient and accurate extraction of

in vivo calcium signals from microendoscopic video data // arXiv preprint arXiv:1605.07266. 2016.

Zucker Robert S. Calcium-and activity-dependent synaptic plasticity // Current opinion in neurobiology. 1999. 9, 3. 305–313.