Differentiating Task-Based Functional Ultrasound Signals via Data-Driven Decompositions

Maarten Enthoven

Differentiating Task-Based Functional Ultrasound Signals via Data-Driven Decompositions

Thesis

submitted in partial fulfillment of the
requirements for the degree of

Master of Science

in

Electrical Engineering

by

M.M.F.C. Enthoven
born in Utrecht, The Netherlands

This work was performed in:

Circuits and Systems Group
Department of Microelectronics & Computer Engineering
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Delft University of Technology

Delft University of Technology
Department of
Microelectronics & Computer Engineering

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled **"Differentiating Task-Based Functional Ultrasound Signals via Data-Driven Decompositions"** by **M.M.F.C. Enthoven** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 28/05/2021

Chairman: ______________________________
prof.dr.ir. G.J.T. Leus

Advisors: ______________________________
prof.dr.ir. G.J.T. Leus

____
dr. B. Hunyadi

Committee Members: ______________________________
dr. ir. P. Kruizinga

____

Acknowledgements

I want to express my sincere gratitude to my advisors Geert Leus and Bori Hunyadi. Their help and guidance has been of enormous value to me. Our meetings were fruitful and steered me in the right direction. I have learned a lot from them in terms of precision and quality of work. Their feedback was accurate and immensely useful.

Furthermore, I would like to thank all the people from the CUBE group at Erasmus MC. Without their work, this research would not have been possible. In particular, I want to thank Chagajeg Soloukey and Aybüke Erol. They created the experimental paradigms and performed the data acquisition of the dataset on which I performed my analyses. Both have been enormously kind to me for giving me feedback on my work and I wish them the best of luck in their further academic endeavors.

Lastly, I want to thank my family. My family has given me the best childhood I could have wished for. They have always supported me in my decision-making which has led to where I stand today. And I am grateful for that. Throughout this thesis and study career, I've always had their support and I can not express enough how important they are to me.

M.M.F.C. Enthoven
Delft, The Netherlands
28/05/2021

Abstract

At the Center for Ultrasound and Brain imaging at Erasmus MC in Rotterdam, a mouse's visual cortex had been imaged using the functional ultrasound (fUS) technique. The mouse had been exposed to different visual stimuli. The stimuli varied in position, size, and shape. We investigate how the measured task-based fUS signals differ depending on the visual stimuli presented to the mouse. For that purpose, we decompose the fUS data with four different methods giving different levels of sparsity. This thesis compares the performance of the four methods, and provides neurological insights obtained with these methods.

For modeling the data, we consider four data-driven decomposition methods: independent component analysis (ICA) and three sparse dictionary learning (sDL) variants. The methods decompose the data into better interpretable spatial maps and time courses. Every decomposition is further examined by training ℓ_1-regularized prediction models that optimize for sparsity. The goal is to predict the presented stimulus based on the decomposed data. Furthermore, the potential of group lasso regularization in prediction models is illustrated.

The decompositions extract spatial maps anatomically linked to the visual cortex, superior colliculus and the hippocampus. All decompositions achieve considerable performance in the position prediction but have low success in the size and shape prediction. ICA outperformed the three sDL variants in all prediction tasks. Furthermore, group lasso regularization is found to be a useful tool to obtain discriminatory information in the time dynamics.

Contents

Acronyms

BOLD	blood oxygen level dependent
CBV	cerebral blood volume
DNN	deep neural network
EEG	electroencephalography
fMRI	functional magnetic resonance imaging
fUS	functional ultrasound
GLM	general linear model
HC	hippocampus
HRF	hemodynamic response function
ICA	independent component analysis
MEG	magnetoencephalography
SC	superior colliculus
sDL	sparse dictionary learning
sICA	spatial ICA
SM	spatial map
TC	time course
VC	visual cortex

Introduction 1

Understanding the mind is one of the greatest desires in human history. The field of neuroscience ultimately desires understanding the organization of memories, origin of thoughts, basis of consciousness and others. To answer these questions, neuroscientists have developed and used neuroimaging tools to aid in their research. Neuro- or brain imaging techniques can be used to reveal the structure and function of the brain.

The first neuroimaging techniques had serious drawbacks. These techniques would require the subjects to undergo surgery, be injected with artificial contrast agents or be exposed to ionizing radiation. Due to the inherent risks in these imaging methods, primarily patients with disorders could be analyzed limiting the "variety" of the dataset.

Since its introduction in the 1990s, functional magnetic resonance imaging (fMRI) largely "solved" these issues and has consequently become the gold standard for deep brain imaging. fMRI enabled studying the brain non-invasively by measuring a naturally occurring contrast agent, the oxygenation of blood. This enabled frequent and recurrent testing on healthy subjects because there was no fear of harming them [1].

However, fMRI is not perfect. The scanners are expensive and not portable. Furthermore, it requires the subject to be stationary in the machine.

Recently, a new imaging technique called functional ultrasound (fUS) was developed which measures the cerebral blood volume (CBV). The imaging machine is smaller and, using ultralight probes, could potentially allow the subject to move freely during the experiment. This technique matches the spatial resolution of fMRI and shows an excellent temporal resolution [1].

1.1 Problem statement

At the Center for Ultrasound and Brain imaging at Erasmus MC in Rotterdam (CUBE), a mouse's visual cortex has been imaged with fUS. The mouse had been exposed to visual stimuli. The stimuli varied in position, size, and shape. Are the responses to these visual stimuli present in the measured fUS signal?

The answer to this question lies in the fUS's ability to reveal and explain hemodynamic patterns. The main objective of this thesis is to search for and understand hemodynamic changes observed in the fUS data that are associated with diverse visual stimuli.

The research questions will be split in two categories: RQ 1 tackles questions from a signal processing domain and RQ 2 will deal with the neuroscientific questions.

RQ 1. What type of models can be used to accurately describe the data?

To answer this question: we will consider two modeling approaches: independent component analysis (ICA) and sparse dictionary learning (sDL).

RQ 2. What is the vascular response of the mouse brain to a visual stimulus?

To answer this question, we will consider how the response changes depending on the type of stimulus presented, wherefore the following subquestions will be answered:

2.1 How does the response differ depending on the position of the stimulus in the field of view?

2.2 How does the response differ depending on the size of the stimulus?

2.3 How does the response differ depending on the shape of the stimulus?

1.2 Outline

We will briefly outline how the two research questions will be answered. First we introduce the most important neuroscientific concepts relevant for our research in Chapter 2. Then we describe the type of data and the acquisition methodology in Chapter 3. Subsequently, we start by decomposing the data using the sDL and ICA method. This can be seen as a dimensionality reduction step as we decompose the dataset with 35200 voxels to a dataset of 80 spatial maps, also called features. This will be described in Chapter 4. The question arises: how well does a decomposition describe the data? For that purpose we set up a prediction experiment in which we predict what stimulus was presented to the mouse, using the decomposed data. This will be described in Chapter 5. The goal of the prediction experiments (Chapter 6) is twofold:

- Firstly, if the prediction model can predict the presented stimuli well, the dataset should have useful discriminatory features. In that case we decide that the decomposition describes the data well. This answers RQ1.
- Secondly, we can use the prediction model to obtain neurological insights. By analyzing the decision function we can learn why models behave as they do. We base our neurological insights upon to discriminatory information obtained from prediction models. This answers RQ2.

The results of these experiments can be found in Chapter 7. Finally, the conclusion and recommendations for future research are given in Chapter 8.

Neuroscience 2

In this chapter, a brief introduction to neuroscience will be given, primarily focusing on concepts related to our data. In Section 2.1, it is explained why we are interested in measuring the blood. Subsequently, the method for measuring the blood, fUS, is highlighted in Section 2.2. What type of response is expected and the main decomposition idea is explained in Section 2.3.

2.1 Measurements

The brain is arguably the most complex organ of all animals. It is responsible for acting on almost all external tasks or stimuli. The data stream from the senses, such as a smell or a touch, is processed in various brain networks leading to perception. Also, in the seeming absence of any tasks, there is a resting state network active.

The brain's processing occurs in the neurons with tiny electrical signals. Therefore, it makes sense to measure electrical activity to learn more about the functioning of the brain. This can be done directly with electroencephalography (EEG) or indirectly, using the magnetic fields produced by electric currents, by magnetoencephalography (MEG). However, accurately measuring the electrical activity of the brain is an almost impossible task as there are millions of neurons. For example, there are about 10 million neurons in a mouse brain. The spatial resolution of these techniques is not high enough for measuring individual neurons.

Accompanying the neuronal activity is an increase in blood flow. This is because the energy deficit in the neurons has to be resolved by transporting glucose and oxygen towards them, using the bloodstream. This is done by increasing the blood flow rate or an expansion of blood vessels. Therefore, neuronal activity can be inferred from measurements of the blood flow in the brain.

This blood flow can be measured and quantified using particles in the blood called hemodynamic contrasts. The main difference between imaging techniques lies in what contrast is measured. For example, blood oxygen level dependent (BOLD) fMRI aptly measures the oxygenation of hemoglobin and CBV-weighted fMRI the injected iron oxide particles in the blood.

2.2 Ultrasound imaging

In the novel fUS imaging technique, the hemodynamic contrast is the red blood cells. An ultrasonic pulse is sent to the blood vessels and part of the pulse will be backscattered by red blood cells. The energy of the echoes is proportional to the CBV [1], [2]. The measured signal can be converted into a power Doppler image by computing the average energy of each voxel, for a period of time, typically around 200 ms [2].

Since the start of the 1990s research has been performed on fMRI data using an enormous variety of advanced signal analysis methods. Conversely, most studies on fUS data perform univariate voxel correlations on the task regressor. One study used ICA on fUS data to find neurological insights but found that its results conflicted with previous results obtained from fMRI experiments [3]. The differences were explained by "the enhanced neurofunctional imaging capabilities of fUS as compared to fMRI". The question is if assumptions in fMRI models are equally valid in fUS models.

Whether these assumptions are valid, depends partly on the extent BOLD and CBV signals differ. Research has been conducted in to what extent BOLD and CBV are alike in the brain. In these studies, the CBV was obtained in an fMRI scan by monitoring an exogenous contrasting agent and, under some biophysical assumptions, converted to CBV [4]. It is shown that significant BOLD signal changes can occur in the absence of a corresponding CBV change. Besides that, the BOLD signal showed the post-stimulus undershoot pattern while CBV did not show this effect [5]. Another study found that CBV-weighted fMRI had better localization than BOLD fMRI because there was less activation in draining veins. These differences could be explained by differences in neurovascular coupling [6]. To what extent these different dynamics are of influence to the data model selection is unknown.

2.3 Hemodynamic response

As mentioned before, we want to understand the brain's reaction to stimuli from measurements of blood. The brain's hemodynamic response function (HRF) to a stimulus is dependent on at least three variables: 1. the type of stimulus, 2. the location in the brain, and 3. the timing of the measurement. Even though we can describe all three variables very well, we do not know the interaction between the variables.

The responses measured at different locations can have similar time courses. In that case, we can group the different location in single spatial structure. This grouping action gives a spatial network as output with a corresponding hemodynamic time course. The combination of such a spatial network and its temporal response is what we will call a component.

Because the underlying hemodynamic behavior is poorly understood and the concept of a network is not clearly defined, we need to rely on assumptions in either the temporal or spatial structure for the grouping action. Common assumptions are independence or sparsity in space [7]. It has been argued that sparsity is the superior assumption [8], [9]. The reason for that is that sparsity allows for overlapping networks while independence does not. In Sections 4.4 and 4.5, it is explained how these assumptions can be used to extract spatial networks with a similar time course.

Methodology 3

In this chapter, the type of data and the acquisition methodology are detailed. We start by exploring the dataset and give the imaging parameters in Section 3.1. Subsequently in Section 3.2, the acquisition method will be explained. The preprocessing steps will be laid out in Section 3.3. Finally in Section 3.4, we will briefly discuss the generalizability of possible results.

3.1 Data structure

One coronal slice of a single mouse brain was imaged in one run with an ultrasound scanner. The left and right visual cortex are present in the slice. The output is a grid of N time courses $I(x, z, t)$, each corresponding to the power Doppler value of a voxel. A mean image ($I_\mu(x, z) = \frac{1}{T}\sum_t I(x, z, t)$) is shown in Figure 3.1. In Figure 3.2 the anatomical map corresponding to the coronal location is shown.

The slice was continuously imaged for approximately 42 min with a sampling time of $dt \approx 0.22\,\mathrm{s}$ resulting in $T = 11780$ time samples for each voxel. The PDI images were cropped to slices with a grid of $N_z \times N_x = 160 \times 220$ voxels, giving 35200 voxels in total.

The measurements can intuitively be organized as a 3-dimensional matrix $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{N_z \times N_x \times T}$ representing the number of samples in the depth, length, and time dimension respectively. However, for analysis, it is usually necessary to reshape this into a 2-dimensional matrix $\boldsymbol{Y} \in \mathbb{R}^{T \times N}$. Here $N = N_z \times N_x$ is the total number of voxels and every column represents a voxel's time course.

3.2 Experimental protocol

The mouse is repeatedly shown a visual stimulus, for approximately 3 s followed by a rest period of 5 to 9 seconds. This is what we will call the stimulus pattern. It gives a stimulus frequency $f_{\text{stimulus}} \geq \frac{1}{12}\,\mathrm{Hz}$. Therefore, the number of time samples for which the stimulus was shown is $T_s \approx 12$ and there are at least 37 time samples in each stimulus.

The visual stimuli varied in three ways as illustrated in Figure 3.3:

- Position: the stimulus was either left or right. It is expected that a left (right) stimulus causes a greater response on the right (left) side of the brain due to the lateralization of brain function.

- Size: the stimulus was shown in 5 different sizes. It is expected that a larger size causes a greater response than a smaller size.

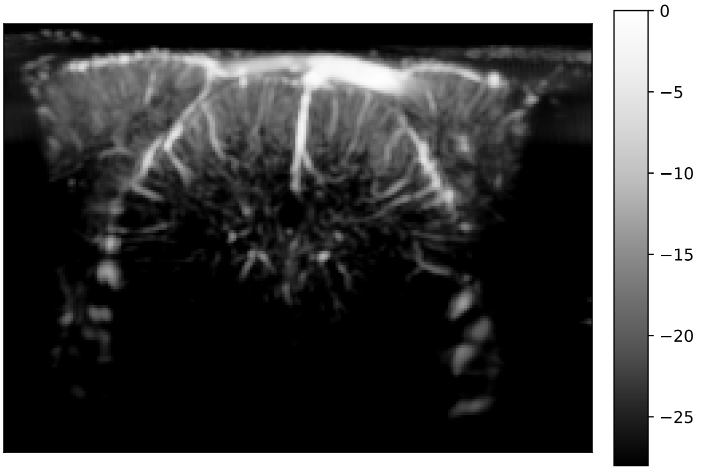


Figure 3.1: Mean image. Vascular structures are revealed by the power Doppler imaging. Taking $10\log_{10}$ of the 99.9^{th} percentile for a decibel scale increases contrast for visualization purposes.

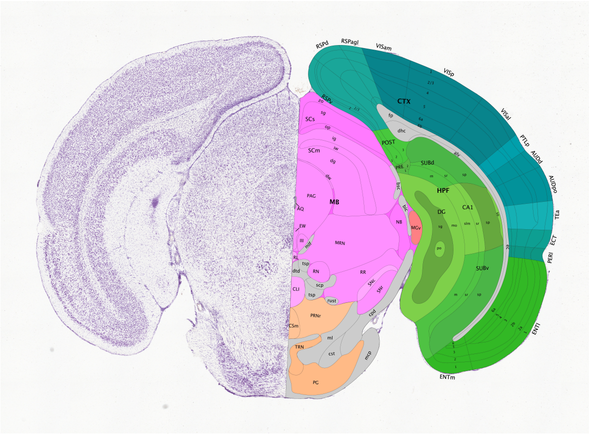


Figure 3.2: Anatomical map of approximately the same slice as Figure 3.1. The image is taken from the online Allen Mouse Brain Atlas. The visual cortex is in the upper left and right corner in the darker turquoise color.

- Shape: the stimulus was either a circle or a square. In advance, it is unclear how the shape influences the hemodynamic response.

These three categories result in a total of $2 \times 2 \times 5 = 20$ different input combinations. Every combination was presented 10 times giving in a total of $S = 200$ stimuli. The experiment was split into two blocks. In the first block, only squares were presented while the size and location varied. Subsequently, there was a short rest period after which the second block started where the circles were shown. The first block was preceded by a short acclimatization period.

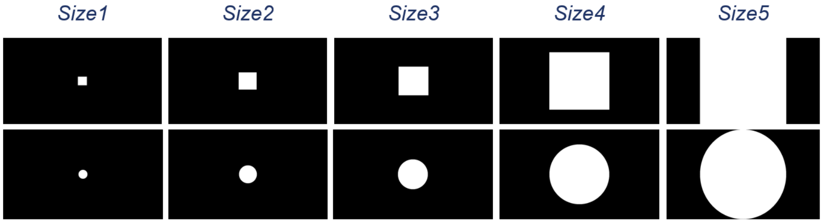


Figure 3.3: Square and circle stimuli shown to the mouse in different sizes.

Group- and stimulus-level

The observations $\boldsymbol{Y}$ can be viewed on two different levels: the group- and stimulus-level. The stimulus-level matrix $\boldsymbol{Y}_i \in \mathbb{R}^{37 \times N}$ where $1 \leq i \leq S$, represents the observations related to stimulus i and the directly following rest period.

A group-level observation matrix $\overline{\boldsymbol{Y}} \in \mathbb{R}^{T \times N}$ (indicated by the overline) can be formed by a temporal concatenation of all stimulus-level matrices $\boldsymbol{Y}_i$, so

$$\overline{\boldsymbol{Y}} = \begin{pmatrix} \boldsymbol{Y}_1 \\ \vdots \\ \boldsymbol{Y}_S \end{pmatrix} .$$

Conversely, this operation can be reversed by the splitting or “ungrouping” the group-level matrix into S stimulus-level matrices.

3.3 Preprocessing

A following operations were performed on the dataset:

- Thresholding: all voxels with a mean power Doppler value below $-25\,\text{dB}$ of the maximum ($10 \log_{10} \frac{I_\mu(x,z)}{\max I_\mu(x,z)} \leq -25$), were regarded as noise and therefore discarded.
- Temporal filtering: all voxels were filtered with a 4th order Butterworth high-pass filter at $f_c = \frac{1}{16}\,\text{Hz} < f_{\text{stimulus}}$.
- Temporal normalization: each voxel for each stimulus was normalized to zero mean and unit variance.

3.4 Generalizability

This dataset was created from one continuous recording of a single mouse. This means that the dataset has limited variation. The scope of the results are limited to only this mouse at this particular recording time. This influences the generalizability of the research.

Besides that, not the whole brain was imaged. Only one coronal slice was investigated. Therefore one should be careful in extrapolating the results to slices in the near vicinity of this slice. For example, in this report three-dimensional anatomical regions such as the visual cortex (VC) will be described. One should keep in mind that only a two-dimensional slice of the three-dimensional structure is imaged.

Decompositions 4

This chapter will concern the decomposition techniques applied on the fUS data. First, we discuss what characteristics a suitable model should exhibit in Section 4.1. In Sections 4.2 and 4.3 we start from a simple linear model and build up to the general linear model (GLM). Sections 4.4 and 4.5 describe the two decomposition techniques (ICA and sDL) that will be applied on the dataset. Section 4.6 describes how time courses can be back-reconstructed after the spatial maps are obtained.

4.1 A brief note on model selection

Little is known about the statistical methods and signal processing for fUS. Most studies on fUS data perform univariate voxel correlations on the task regressor. However, such univariate methods leave some questions unanswered. From three decades of fMRI research, we have learned that more advanced models can give answers to otherwise inconclusive questions.

A major challenge is developing advanced models that not only perform well, but also give some insight into how they work. This conflict is illustrated by modern deep neural networks (DNNs) that deliver astonishing results but require estimating many parameters, therby complicating the interpretability. The black-box nature of DNNs has impeded the adoption in clinical use, because its results are hardly explainable and therefore untrusted by clinicians.

Sparsity

To obtain explainable results, Occam's razor should be used; choose the simplest model that describes the data well [10]. Simple models are typically more interpretable and require less training data to make them work. However, simple models can underfit the data and obstruct the discovery of more complex patterns.

We can stimulate a model's interpretability by actively selecting for sparsity. This can be done with regularization. A regularization term imposes a cost on the optimization function. Common regularization functions are the ℓ_1- and ℓ_2-norm on the weight vector. Both penalties restrain the coefficients to be small, but for parsimonious models, the ℓ_1-norm is recommended. The reason for this is that ℓ_1-regularization encourages coefficients to shrink to zero. Only the most explanatory variables are used and the variables with less explanatory power are not selected. The ℓ_0-norm also stimulates sparse models but is not convex and therefore more difficult to solve.

Model types

The selected model and the posed questions are strongly connected. Hardly ever, different models explain the observed data similarly [11]. To obtain meaningful insights, the chosen model should have dependable assumptions and output types related to the questions asked. Therefore, it is important to have well-justified criteria for choosing among different models.

There are two main modeling approaches that one can choose to use:

- Hypothesis-driven
- Data-driven

The hypothesis-driven approach directly assumes prior knowledge of the model. Using prior knowledge, the model can be steered into the desired direction which can be specifically useful for a particular research question. The output of the model is related to the prior information fed into the model. An example of a hypothesis-driven method is the general linear model (GLM) which is detailed in Section 4.3.

Contrarily, the data-driven approach does not use any prior information about the model. No prior knowledge of the experimental task is necessary to detect hidden patterns. Available information in the data is extracted using possibly constraint-based assumptions. Because the model is not steered in any desired direction (except for the constraints), the outcome is less narrow than in hypothesis-driven models. Examples of data-driven methods are ICA and sDL which are detailed in Sections 4.4 and 4.5.

4.2 Linear model

Without choosing a hypothesis- or data-driven approach, we can start modeling the voxels with a linear model. A single voxel's time course can be modeled as

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} \tag{4.1}$$

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_K \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_K \end{pmatrix}$$

where $\boldsymbol{x}_k$ are column vectors of length T called the regressors and $\boldsymbol{\beta}$ collects the weights of their respective regressors. The idea behind this model is that a single voxel's time course $\boldsymbol{y}$ can be modeled as the sum of K time courses $\boldsymbol{x}_k$ multiplied by their corresponding weights in β_k.

Multiple voxels can be modeled by extending Equation (4.1) with extra columns to the $\boldsymbol{y}$ and $\boldsymbol{\beta}$ for each new voxel. The new model can be written as

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{B} \tag{4.2}$$

where $\boldsymbol{Y} = [\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)} \dots, \boldsymbol{y}^{(N)}]$ and $\boldsymbol{B} = [\boldsymbol{\beta}^{(1)}, \boldsymbol{\beta}^{(2)}, \dots, \boldsymbol{\beta}^{(N)}]$ such that each column contains a different voxel's time course indexed by the superscript. It will be useful to index the rows of $\boldsymbol{B}$ with a subscript, so $\boldsymbol{B}^T = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \dots, \boldsymbol{\beta}_K]$. For totality

- $\boldsymbol{Y}$ is the $T \times N$ observation matrix.
- $\boldsymbol{X}$ is the $T \times K$ design matrix where K is the number of regressors.
- $\boldsymbol{B}$ is the $K \times N$ weighting matrix.

Equation (4.2) can be rewritten as

$$\boldsymbol{Y} = \sum_{k}^{K} \boldsymbol{x}_k \boldsymbol{\beta}_k^T \tag{4.3}$$

highlighting that the observations $\boldsymbol{Y}$ are a summation of K rank-1 components consisting of a column vector of length T multiplied by a row vector of length N. These vectors have a physical meaning:

- The columns of $\boldsymbol{X}$, $\boldsymbol{x}_k$ represent K time courses (TCs).
- The rows of $\boldsymbol{B}$, $\boldsymbol{\beta}_k$ represent K spatial maps (SMs).

So the observations are modeled as a sum of SM and TC pairs. This is a latent space model; the observations $\boldsymbol{Y}$ are explained through K latent factors encoded in $\boldsymbol{X}$ and $\boldsymbol{B}$. Generally $K < N$ which results in a reduction of dimensionality and compression of the data. Unfortunately, determining the hyper-parameter K is mostly an open question.

These latent factors or components offer a global, rather than local, analysis. In a local analysis, every voxel is analyzed individually. That is computationally expensive depending on the spatial resolution. Additionally, it is ignorant of any possible spatial correlation that might exist between the voxels. So to simplify, it is useful to parcellate the brain's voxels into several global regions of interest. This stems from the idea of a spatio-temporal organization of the brain: a group of voxels can show similar TCs or at some time samples can show similar activations in a particular SM. For this reason, the model decomposes the input into separate sets of spatial and temporal components. Such component analysis is a way to extract neurologically better interpretable and thus more relevant information.

Limitations

Without additional constraints, Equation (4.2) has inherent limitations. Let $\boldsymbol{R}$ be an invertible matrix, then this model can be written as

$$\begin{aligned} \boldsymbol{Y} &= \boldsymbol{X}\boldsymbol{B} \\ &= (\boldsymbol{X}\boldsymbol{R}) * (\boldsymbol{R}^{-1}\boldsymbol{B}) \\ &= \tilde{\boldsymbol{X}}\tilde{\boldsymbol{B}} \end{aligned} \tag{4.4}$$

thus rendering the decomposition non-unique. However, most decomposition methods impose some form of constraint on $\boldsymbol{X}$ or $\boldsymbol{B}$ make the solutions $\boldsymbol{X}$ and $\boldsymbol{B}$ better defined. For example ICA imposes independence constraints on $\boldsymbol{B}$ and sDL imposes sparsity constraints on $\boldsymbol{B}$. Another common constraint is to make the columns of $\boldsymbol{X}$ unit ℓ_2-norm. Note that in this case, the signs are still ambiguous because we can multiply a component pair by -1 without affecting the model. This is known as the sign ambiguity.

4.3 General linear model

In a general linear model (GLM) the design matrix $\boldsymbol{X}$ from Equation (4.2) is assumed known; thus it is a hypothesis-driven analysis method. In the design matrix, the K columns of $\boldsymbol{X}$ are each constructed to reflect a time signal regressor, the independent variable, thought to influence the observations $\boldsymbol{Y}$, the dependent variables. The list of regressors could include, but are not limited to, the task pattern, head motion, or signal drift. These experimental variables should be recorded as they are not obtained from the data.

With $\boldsymbol{Y}$ and $\boldsymbol{X}$ assumed known, the K columns linearly independent ($\boldsymbol{X}$ is full rank), and $K \leq T$ (overdetermined system), the GLM model is uniquely solved in a least-squares sense. Because GLM is a compact way of simultaneously writing several multiple linear regression models, all individual models can be solved by themselves. The loss function for voxel n is

$$l(\boldsymbol{\beta}^{(n)}) = \frac{1}{2}\|\boldsymbol{y}^{(n)} - \boldsymbol{X}\boldsymbol{\beta}^{(n)}\|^2 \,. \tag{4.5}$$

Then the average loss function over all voxels must be N distinct estimation problems

$$l(\boldsymbol{B}) = \frac{1}{2N}\sum_{n=1}^{N} l(\boldsymbol{\beta}^{(n)}) \,. \tag{4.6}$$

Minimizing the loss in Equation (4.6) gives

$$\begin{aligned} \hat{\boldsymbol{B}} &= \arg\min_{\boldsymbol{B}} l(\boldsymbol{B}) & (4.7)\\ &= \arg\min_{\boldsymbol{B}} \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}\|_F^2 & (4.8)\\ &= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y} \,. & (4.9) \end{aligned}$$

There is a drawback to the GLM modeling method. Dependence in the columns of $\boldsymbol{X}$ would cause the linear system to be ill-conditioned. This usually leads to unreliable estimates of the regression coefficients [12].

Lasso

To solve ill-conditioned GLM problem a regularization function Ω can be added to Equation (4.6) on the $\boldsymbol{\beta}$ vectors:

$$\hat{\boldsymbol{B}} = \arg\min_{\boldsymbol{B}} \frac{1}{2N}\sum_{n=1}^{N} l(\boldsymbol{\beta}^{(n)}) + \alpha\Omega(\boldsymbol{\beta}^{(n)}) \,. \tag{4.10}$$

where α is a positive constant. For example, the ridge regression [13] was proposed to solve the problem of collinearity in linear regression and uses the ℓ_2-norm as regularizer. A drawback of the ridge regression is that it does not set any coefficients to zero and therefore does not give an easily interpretable model.

Another regularization function is the ℓ_1-norm $\Omega(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_1$. This regularization function is known as the lasso [14]. Unlike the ridge regression, lasso does shrink coefficient values to zero and is therefore more interpretable. In lasso, Equation (4.10) becomes

$$\hat{\boldsymbol{B}} = \arg\min_{\boldsymbol{B}} \frac{1}{2N} \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}\|_F^2 + \alpha \|\boldsymbol{B}\|_{1,1} \,. \tag{4.11}$$

The regularization parameter α controls the trade-off between sparsity and minimization error. Large values for α cause the SMs of $\boldsymbol{B}$ to be sparse. Small values for α put more importance to minimizing the loss function.

4.4 Sparse dictionary learning

In sDL the design matrix and mixing matrix are both learned from the data with the constraint that the columns of the mixing matrix $\boldsymbol{B}$ are sparse. The design matrix is now called a dictionary with K columns $\boldsymbol{x}_k \in \mathbb{R}^T$. Two common methods to enforce sparsity in $\boldsymbol{B}$ are the ℓ_1- and ℓ_0-constraint on $\boldsymbol{\beta}^{(n)}$. This means that every voxel is a combination of a few TCs. All voxels with a nonzero value related to a time course are part of the same SM.

In a similar style as Equation (4.11), the ℓ_1-norm regularized sDL problem can be solved as an optimization problem:

$$\hat{\boldsymbol{X}}, \hat{\boldsymbol{B}} = \arg\min_{\boldsymbol{X},\boldsymbol{B}} \frac{1}{2N} \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}\|_F^2 + \alpha \|\boldsymbol{B}\|_{1,1} \,. \tag{4.12}$$

Because both the design and mixing matrix are both learned from the data, sDL is called a data-driven model. As in Section 4.3, the parameter α controls the trade-off between sparsity and the minimization error.

One method for solving Equation (4.12) is the online dictionary learning algorithm [15]. In this algorithm the problem is solved by a two-stage alternating optimization. In the first stage, also called the sparse coding stage, the lasso solution is computed for $\boldsymbol{B}$ assuming $\boldsymbol{X}$ is fixed. In the second stage, also called the dictionary updating stage, $\boldsymbol{X}$ is solved assuming $\boldsymbol{B}$ is fixed.

4.5 Independent component analysis

The goal of ICA is to decompose the observed data $\boldsymbol{Y}$ into $\boldsymbol{X}$ and $\boldsymbol{B}$, like Equation (4.2),

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{B} \tag{4.13}$$

where the rows $\boldsymbol{\beta}_k$ are optimized to be maximally statically independent. Because the rows of $\boldsymbol{B}$ represent SMs, we call this spatial ICA (sICA) where the components are assumed independent in space. Because the SMs are independent, this means that SMs have little overlap in space. This is unlike sDL where overlap is allowed.

The independence can be measured using some function $F(\boldsymbol{B}) = F(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K)$ of independence. There are no assumptions on $\boldsymbol{X}$. Thus, if $\boldsymbol{W}$ is the invertible linear transformation matrix that maximizes $F(\boldsymbol{B})$, then

$$\begin{aligned} \boldsymbol{W}\boldsymbol{Y} &= \boldsymbol{B} \\ \boldsymbol{Y} &= \boldsymbol{W}^{+}\boldsymbol{B} \end{aligned} \tag{4.14}$$

where $^+$ denotes the pseudo-inverse. Comparing Equations (4.13) and (4.14) we can conclude that $\boldsymbol{X}$ must be the pseudo-inverse of $\boldsymbol{W}$. Because both $\boldsymbol{X}$ and $\boldsymbol{B}$ are learned from the data, this is a data-driven model.

Different ICA algorithms use different measures of independence in $F(B)$. We use the fastICA algorithm for computing the solution. FastICA uses non-Gaussianity, measured by an approximation of negentropy, as a measure of independence [16].

4.6 Back-reconstruction

If the SMs $\boldsymbol{B}$ are known, they can be used to derive the TCs in $\boldsymbol{X}$. In that case, it can be considered as a GLM, only now the $\boldsymbol{B}^T$ collects the regressors

$$\boldsymbol{Y}^T = \boldsymbol{B}^T\boldsymbol{X}^T\,. \tag{4.15}$$

and can be solved for $\boldsymbol{X}$ as in Section 4.3. This might seem as a trivial step because Equations (4.12) and (4.13) already compute the $\boldsymbol{B}$ matrix. However, this could be useful when we have already learned a set of SMs and want to use those to extract TC from a new dataset. It is also useful when the last sparse coding step in sDL uses a different algorithm then in the previous sparse coding steps. In that case the time courses do not truly reflect the newly computed $\boldsymbol{B}$ and have to be computed again.

Prediction models 5

In Chapter 4, we assumed no knowledge of the underlying spatial maps. Consequently, we resorted to unsupervised techniques for obtaining those maps. Contrarily, now it is known what stimulus was presented for each experiment. We can use that knowledge with supervised learning techniques to obtain insights. The idea is to create a model that can predict the stimulus that was shown. Once we have trained the model, it is possible to analyze its decision function. The decision function is used to make a prediction. Understanding the decision made by the model gives insight.

In this section, we will introduce metrics and methods to obtain predictive results. Section 5.1 introduces a Pearson correlation method to compare the time courses with the known stimulus pattern. These values are fed into prediction models. A theoretical framework for developing these models is explained in Section 5.2. In Section 5.3 we explain the performance evaluation and cross-validation strategy is detailed in Section 5.4. Finally we discuss methods for determining feature importance in Section 5.5.

5.1 Pearson correlation

The $\boldsymbol{X}$ matrix contains time courses. We can find the correlation of the time courses with the known stimulus pattern. The Pearson correlation coefficient r is a proxy for how well the stimulus pattern is resembled by the time course.

The stimulus pattern $\boldsymbol{a}$ can be modelled as a box-car function $\boldsymbol{a}$ where

$$a_t = \begin{cases} 1, & \text{if } t \leq T_s \\ 0, & \text{if } t > T_s \end{cases}$$

and T_s is the stimulus duration in samples. Additionally, time delaying $\boldsymbol{a}$ with D samples can be more useful as the hemodynamic response is also delayed.

The Pearson correlation r for TC $\boldsymbol{x}$ with stimulus pattern $\boldsymbol{a}$ is computed with

$$r = \frac{\sum_{i=1}^{T}(x_i - \overline{x})(a_i - \overline{a})}{\sqrt{\sum_{i=1}^{T}(x_i - \overline{x})^2(a_i - \overline{a})^2}} \tag{5.1}$$

where the overline designates the mean. The maximum value $r = 1$ indicates a perfect positive correlation. Conversely $r = -1$ indicates a perfect negative correlation. The absence of correlation is indicated by $r = 0$.

The Pearson correlation is the standard method for determining the correlation between two time series. For this method to work we have to make strong assumptions on the shape of the stimulus pattern $\boldsymbol{a}$. Ideally, the shape of $\boldsymbol{a}$ would be exactly the hemodynamic response function (HRF). However, the shape of the HRF is not easy to determine and might be dependent on the anatomical location.

We assume the HRF has an active region with a response and an inactive region without a response. The box-car function gives equal weight to all time samples in the active region. Still, we have no method for determining delay D except by using trial and error. Besides that, we have set the duration of the box-car arbitrarily to the length of duration T_s. As we will see later, group lasso regularization, introduced in Section 5.2, will be useful in determining these two parameters analytically.

5.2 Supervised learning

By correlating a TC $\boldsymbol{x}$ with the box-car function $\boldsymbol{a}$ we obtain the value r (see Equation (5.1)). This can be done for all time courses in a stimulus i to obtain the correlation vector $\boldsymbol{r}_i \in \mathbb{R}^K$. This correlation vector can be used to make predictions. More details on how $\boldsymbol{r}_i$ is obtained is given in Section 6.1.

The goal of the predictions is to predict the stimulus i based on the computed correlations $\boldsymbol{r}_i$. To check the validity of the predictions, we need to know the ground truth to compare the predictions with. We construct a ground truth label vector $\mathbf{y} \in \mathbb{R}^S$. The target to predict is categorical in the position and shape experiments: left or right and square or circle. There are two categories and therefore we use binary label $\mathrm{y}_i \in \{-1, +1\}$. For these categories, predicting the correct label is a classification problem.

In the size experiment the stimulus was in five different sizes so we use $\mathrm{y}_i \in \{1, 2, 3, 4, 5\}$. Predicting to correct size is a regression problem. As we will explain later, this a modeling inadequacy. The target to predict is in fact ordinal because there is an intrinsic order in the sizes (small to large). More appropriate is a qualitative description such as $\mathrm{y}_i \in \{\text{smallest}, \text{small}, \text{medium}, \text{large}, \text{largest}\}$ where we make no assumption on the quantitative size, but where the order of the labels is still present. We will discuss this issue later when we explain the prediction model for the regression problem.

Regularized risk minimization

For making predictions, we need a model that takes the correlation vector r as input and, after a computation with the model's weight vector $\boldsymbol{w}$, gives an output or prediction $\hat{\mathrm{y}}$. Ideally, the output $\hat{\mathrm{y}}$ is the same as the ground truth y. As that is not always the case, we need a loss function l to measure how different the prediction is from the ground truth. The average of all loss function in the training dataset is known as the the empirical risk:

$$\begin{aligned} J(\boldsymbol{w}) &= R_{\text{emp}}(\boldsymbol{w}) \\ &= \frac{1}{S}\sum_{i=1}^{S} l(\mathrm{y}_i, \boldsymbol{r}_i, \boldsymbol{w}) \end{aligned} \tag{5.2}$$

where S is the number of samples in the dataset. The goal is to minimize the empirical risk by tweaking $\boldsymbol{w}$ appropriately. This is called training the model. However, if $\boldsymbol{w}$ is not constrained, the model can suffer from overfitting. In that case, the model is able to predict the training dataset well, but fails on unseen data. This issue is further discussed in Section 5.4.

To prevent overfitting, we want to regularize the weights of the model. A penalty on the weight vector $\boldsymbol{w}$ can be added to Equation (5.2). This is known as the regularized risk:

$$J(\boldsymbol{w}) = \frac{1}{S}\sum_{i=1}^{S} l(\mathrm{y}_i, \boldsymbol{r}_i, \boldsymbol{w}) + \lambda\Omega(\boldsymbol{w}) \tag{5.3}$$

where λ is a constant that controls the importance of the regularization, and Ω is the regularization function [17]. The goal is to choose $\boldsymbol{w}$ such that the cost function in Equation (5.3) is minimized. The chosen $\boldsymbol{w}$ that minimizes the cost function provides the optimal model weights and can be analyzed in more detail.

Adding a regularization term has the added benefit that we can steer the model into a desirable direction. Commonly, a sparsity constraint is enforced on $\boldsymbol{w}$ as it leads to simpler and more interpretable models. Therefore we will primarily use ℓ_1-regularized models.

When the loss function l is constrained to

$$l(\mathrm{y}_i, \boldsymbol{r}_i, \boldsymbol{w}) = l(\mathrm{y}_i, \boldsymbol{r}_i^T\boldsymbol{w})$$

then this is a linear predictor. This is advantageous because every element of the weight vector $\boldsymbol{w}$ is directly linked to an element of the feature vector $\boldsymbol{r}_i$. In that case, if $\boldsymbol{r}_i$ is interpretable then $\boldsymbol{w}$ must be too. However, it does assume that the target labels can be modeled as a linear combination of features. Kernel predictors could be more flexible as they can model non-linear combinations of features [18]. However, we will stick to linear predictors because kernel predictors are generally more difficult to interpret.

Regression

For the regression problems we use lasso [14] so

$$\Omega(\boldsymbol{w}) = \|\boldsymbol{w}\|_1 \tag{5.4}$$

because the ℓ_1-norm forces nonzero coefficients in $\boldsymbol{w}$. This makes the regression model interpretable. The loss function l is the squared loss so

$$l(\mathrm{y}_i, \boldsymbol{r}_i, \boldsymbol{w}) = \frac{1}{2}(\mathrm{y}_i - \boldsymbol{r}_i^T\boldsymbol{w})^2\,. \tag{5.5}$$

Substituting Equations (5.4) and (5.5) into Equation (5.3), gives the cost function

$$J(\boldsymbol{w}) = \frac{1}{2S}\|\mathbf{y} - \boldsymbol{R}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_1\,. \tag{5.6}$$

Training this regression model is equivalent to minimizing the cost function. Once the model is trained, its prediction is

$$\hat{\mathrm{y}}_i = \boldsymbol{r}_i^T\boldsymbol{w}\,. \tag{5.7}$$

As mentioned before, using continuous labels for the size experiment is a modeling mistake. The linear model regards the size labels as a continuous variable while the labels are ordinal. Ordinal regression techniques are appropriate in this case [19]. However, while the model is not correct, it can be good enough for our purposes. For example, continuous regression models have been applied on fMRI size prediction tasks with high accuracy [20]. Also, because we are interested in the relative performance between different decompositions, and all models suffer from the same inadequacy, they can still be compared. However, we have to be careful in interpreting the model's weights, as it might attempt to model relations which are only in the data due to the artificial labeling.

Classification

For binary classification problems (position and shape) we use the ℓ_1-constrained logistic regression [21]. That is, l is the log-loss of a logistic model

$$\begin{aligned} l(\mathrm{y}_i, \boldsymbol{r}_i, \boldsymbol{w}) &= -\log \frac{1}{1+e^{-\mathrm{y}_i \boldsymbol{r}_i^T \boldsymbol{w}}} \\ &= \log\left(1+\exp\left(-\mathrm{y}_i \boldsymbol{r}_i^T \boldsymbol{w}\right)\right). \end{aligned} \tag{5.8}$$

Substituting the ℓ_1-norm from Equation (5.4) and Equation (5.8) into Equation (5.3), the cost function becomes

$$J(\boldsymbol{w}) = \frac{1}{S}\sum_{i=1}^{S} \log\left(1+\exp\left(-\mathrm{y}_i \boldsymbol{r}_i^T \boldsymbol{w}\right)\right) + \lambda \|\boldsymbol{w}\|_1 . \tag{5.9}$$

Training this regression model is equivalent to minimizing the cost function. Once the model is trained, its prediction is

$$\hat{\mathrm{y}}_i = \mathrm{sign}(\boldsymbol{r}_i^T \boldsymbol{w}) . \tag{5.10}$$

Group regularization

We can also leave out the Pearson correlation step and directly unravel the time courses of stimulus i into $\boldsymbol{r}_i \in \mathbb{R}^{37K}$. More details on this can be found in Section 6.6. We should not use the ℓ_1-norm for regularization anymore because we have $37 \times 80 = 2960$ features. A trained model with ℓ_1-regularization would probably be hardly interpretable because it selects features sporadically. Therefore we need a different regularization function.

As before, we can use the regularization to steer the model into a desirable direction. It is desirable to have predefined groups of variables in $\boldsymbol{r}$ be included or excluded together. In our case, a group is defined as all 37 time features related to one SM.

We can use group lasso for this purpose [22]. In group lasso, the sparsity in the group is forced by the regularization function

$$\Omega(\boldsymbol{w}) = \sum_{g=1}^{G} \|\boldsymbol{w}^{(g)}\|_2 \tag{5.11}$$

where G is the number of groups and $\boldsymbol{w}^{(g)}$ is the coefficient vector of group g. This regularization function can be viewed as an ℓ_2-norm over members of each group and an ℓ_1-norm over all groups. If the size of each group is 1, this gives us the regular lasso solution.

The group lasso regression uses the loss from Equation (5.5) so the cost function is

$$J(\boldsymbol{w}) = \frac{1}{2S}\|\boldsymbol{y} - \sum_{g=1}^{G} \boldsymbol{R}^{(g)}\boldsymbol{w}^{(g)}\|_2^2 + \lambda \sum_{g=1}^{G} \|\boldsymbol{w}^{(g)}\|_2 \tag{5.12}$$

where $\boldsymbol{R}^{(g)}$ is the submatrix of $\boldsymbol{R}$ with columns corresponding to the predictors in group g.

The group penalty can also be used in combination with the logistic regression [23] which gives

$$J(\boldsymbol{w}) = \frac{1}{S}\sum_{i=1}^{S} \log(1 + \exp(-\mathrm{y}_i(\boldsymbol{r}_i^T\boldsymbol{w}))) + \lambda \sum_{g=1}^{G} \|\boldsymbol{w}^{(g)}\|_2 \,. \tag{5.13}$$

5.3 Performance evaluation

To evaluate the performance of the trained models, we need to introduce performance metrics. The accuracy of the classifier, denoted κ, is defined as

$$\kappa(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{S}\sum_{i=1}^{S} \delta(\hat{\mathrm{y}}_i, \mathrm{y}_i) \tag{5.14}$$

where δ is Kronecker's delta and $\hat{\mathrm{y}}_i$ is calculated with Equation (5.10). In other words, it is the number of correct predictions over the total number of predictions. This is a fair score because the dataset is class-balanced; there is an even number of positive and negative labels. A score of $\kappa = 1$ means that all samples are classified correctly ($\mathbf{y} = \hat{\mathbf{y}}$) and a score of $\kappa = 0.5$ would be expected when the samples are randomly assigned a label.

For regression analysis, the performance of the model is evaluated using the coefficient of determination, denoted R^2, and is calculated with

$$R^2(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{\sum_{i=1}^{S}(\mathrm{y}_i - \hat{\mathrm{y}}_i)^2}{\sum_{i=1}^{S}(\mathrm{y}_i - \bar{\mathrm{y}})^2} \,. \tag{5.15}$$

where $\hat{\mathrm{y}}_i$ is calculated with Equation (5.7). It represents the proportion of the variance in $\mathbf{y}$ that can be explained by the independent variables. Alternative metrics would be the mean squared error or the mean absolute error. However, since the range 1 to 5 only indicates the order and not a measurable length, they are not selected. The R^2 metric is selected because we are interested in what extent of the variance of $\mathbf{y}$ can be explained.

When the prediction is always correct, $\hat{\mathbf{y}} = \mathbf{y}$, then $R^2 = 1$. A score of $R^2 = 0$ could be obtained by always predicting the mean value $\hat{\mathbf{y}} = \bar{\mathbf{y}}$. For illustration, if the model correctly predicts the smallest and largest size and fails on the remaining three sizes (that is, it predicts the mean size), then $R^2 = 0.8$.

5.4 Cross-validation

In training any model derived from Equation (5.3), we have to choose the regularization constant λ. We select λ such that the model's performance, measured with Equations (5.14) and (5.15), are maximized on data it has not seen before. This is called the model's generalization performance. To obtain an estimate of the generalization performance we have to test the model's performance on the data it had been trained with. Ideally we would split that data in three sets: a training set, a validation set, and a test set. The model is trained with the training set, the λ is validated with the validation set, and the generalization performance is calculated on the test set. However, we do not have the luxury of splitting the dataset into three parts because there are a small number of samples $S = 200$. We resort to cross-validation techniques to obtain a good estimate of the generalization performance.

Because the distribution of the target classes is balanced, we can resort to the basic k-fold cross-validation strategies. The classes are balanced because every stimulus combination is presented an equal amount. Therefore it can be assumed that the relative class frequencies are approximately preserved in each train and test set.

Furthermore, the stimuli were presented in two blocks with a test period in between. The first block contained only square stimuli while the second had only circles. Therefore, we will need to shuffle the stimuli beforehand to ensure that each training and validation set contains an approximately equal number of squares and circles.

In k-fold cross-validation, the data is split into k "folds". We choose $k = 10$ as that is suggested as good choice in general [24]. In each iteration of the algorithm, the model is trained on the $k - 1$ folds as input and tested on the remaining fold. The next iteration trains the model on a different subset of folds and tests on another fold. This way we evaluate the performance of model k times on unseen data. The performance metrics (Section 5.3) can be averaged over all folds giving the mean and the standard deviation of the R^2 or κ value. When cross-validation algorithm is run for several values of λ, we choose the λ that gives the largest mean performance over all folds.

The problem with this method is that the estimation of the generalization performance has an optimistic bias because the performance is directly optimized in choosing the λ parameter [25]. An alternative is nested cross-validation where the λ parameter is chosen independently of the generalization performance. Therefore the predictions will be unbiased. So if we want to compare the performance of the decompositions we have to use nested cross-validation. It is nested because the λ parameter is cross-validated in an inner loop. The inner loop is run inside the outer loop. The outer loop measures the unbiased performance. A more detailed overview of the intricacies of cross-validation is given by Raschka [24].

Note that nested cross-validation does not give a single best value λ. Because the λ is chosen independently of the performance, it can have a different value in each iteration. So if we want to find the best λ, we still have to use the normal or "flat" cross validation to determine that value [25]. When that λ is found, we can use it to train the final model on the whole dataset.

5.5 Feature importance

After choosing the optimal λ with flat cross-validation strategy, the model can be trained on the whole dataset once more, but this time with the selected λ as hyperparameter. The resulting model can be analysed. This is part of a research direction called explainable artificial intelligence where it is investigated why models behave as they do. The most popular technique for this purpose is feature importance [26].

Feature importance can be defined as a measure of the contribution of a feature to a particular predictor [26]. From this definition we can see that the feature importance is inherently linked to a predictive model. A feature important for one model can be unimportant for another model. Therefore the importance score always has be put in context of the model it was obtained with.

A common method to measure the importance is the permutation feature importance score. In this method the single feature's values are randomly shuffled and the decrease in model performance is measured. However such scores could be misleading in the presence of correlated features [27]. That is because in case of shuffling a correlated feature, the algorithm still has access to the other feature to make prediction. This will result in low scores for both features. In general, most measures of feature importance are biased in the presence of correlated features [28].

As shown in Section 5.2, we opted to use ℓ_1-regularized models for feature selection such that redundant features are eliminated. This can be seen as a first step in determining the feature importance, because actively selecting for sparsity pushes weights to zero, and therefore of lowest importance. However, it can not be concluded that the eliminated features are always unimportant because a different subset of features could give similar predictive performances [29]. That is because the algorithm is selective in the presence of highly correlated features. Therefore, we have to remember in interpreting the results that ℓ_1-regularized models do not necessarily give a comprehensive set of important features and predictive features could be ignored.

When we have obtained a subset of features and their weights, we can order the weights by absolute value. This is usually advantageous because larger weights are generally more important in the model's prediction. However, it is a crude method because the weight is not necessarily a proper measure of importance [29]. For example, multiplying a feature with a scalar and dividing the weight by the same scalar, the resulting model does not change but the conjectured importance does. Nevertheless, the features are all in range of $[-1, 1]$ due to the Pearson correlation, and can therefore for our purposes be compared.

So although there are large limitations to a feature importance measure, it is desirable to investigate it. We will consider the feature importance as given by the model weights $\boldsymbol{w}$. Still, we have to be careful interpreting the weights and take the above mentioned limitations into account.

Experiments 6

In this chapter, we describe the experiments for answering the research questions. In Section 6.1 we explain how the dataset is decomposed with four different methods. Sections 6.2 to 6.4 explain what prediction models are used for each task and how the models are set up. In Section 6.5, we explain the sensitivity study of the prediction models to their regularization parameter. Finally, we explain the experiment with the group lasso regularization in Section 6.6.

6.1 Extraction of SM with ICA and sDL

Four sets of SMs are extracted; one extracted with ICA and the others extracted with sDL variations. In all decompositions we will use the group-level observations $\overline{\boldsymbol{Y}}$ as input and set the number of components to $K = 80$. Additionally, the TCs are constrained to unit ℓ_2-norm, so: $\|\overline{\boldsymbol{x}}_k\|_2 = 1$ to alleviate the scaling ambiguity. Three sDL variations are used. They differ in which algorithm is run in the sparse coding step. The variations are:

(1) ℓ_1-constrained with $\alpha = 1$

(2) ℓ_1-constrained with $\alpha = 10$

(3) ℓ_1-constrained with $\alpha = 1$ and in the last sparse coding step ℓ_0 constrained where $\|\boldsymbol{\beta}\|_0 \leq 1$.

From now on these methods will be referred to as sDL1, sDL2, and sDL3 method respectively.

In sDL, the LARS algorithm was used in the sparse coding step. In sDL3 the last sparse coding step was performed with the OMP algorithm. The dictionary was initialized by randomly shuffling the input data.

Note that the SMs of sDL1 and sDL3 will be alike because the underlying algorithm is the same, except the last step. However, because the last sparse coding step differs, the sDL3 method will be sparser than the sDL1.

The fastICA algorithm is used for the ICA decomposition. The parallel method is used and the algorithm's mixing matrix at initialization is normally distributed. The decompositions are performed with Python via the scikit-learn open-source package [30].

Following the extraction of the spatial maps, the time courses are back-reconstructed as described in Section 4.6. This gives in total four separate sets of SMs $\overline{\boldsymbol{B}}$ and their TCs $\overline{\boldsymbol{X}}$. We can "ungroup" $\overline{\boldsymbol{X}}$ into S stimulus-level matrices $\boldsymbol{X}_i$. For every column in stimulus-level $\boldsymbol{X}_i$, Equation (5.1) can be computed giving $\boldsymbol{r}_i = [r_1, \ldots, r_K]$. We choose a delay $D = 7$. Applying this for all S stimuli and concatenating the results row-wise, gives the feature matrix $\boldsymbol{R} \in \mathbb{R}^{S \times K}$. To sum up, we have obtained four different features matrices, one for every decomposition method.

6.2 Experiment about position

In this experiment, we investigate how the response differs depending on the position of the stimulus in the field of view (RQ 2.1). The goal is to predict if the stimulus was presented left or right by only using the feature matrix $\boldsymbol{R}$. We know the ground truth of the position of the stimulus. For stimulus i, we assign the label $\mathrm{y}_i = -1$ if the stimulus was left, and $\mathrm{y}_i = +1$ if the stimulus was right. So ground truth label vector is

$$\mathbf{y} \in \{-1, 1\}^S .$$

We trained an ℓ_1-regularized logistic regression on $\boldsymbol{R}$ and $\mathbf{y}$ by minimizing Equation (5.9) using the SAGA solver. The model included a bias term that was not regularized.

For all four feature matrices, we compute the generalization performance using the nested cross validation with 10 folds in the outer loop and 3 folds in the inner loop. Furthermore, we choose λ giving the largest mean accuracy using the flat cross-validation with 10 folds. The λ parameter is varied in the range 10^{-2} to 10^0 in 20 steps evenly spaced on the $\log_{10}$ scale. When the best λ is found, we use it to train the final model on the whole dataset. We used Equation (5.14) as the scoring function. More details on the cross-validation are given in Section 5.4.

The final model can be analysed. By analyzing the weights of the model, we can infer which components are mostly related to the position of the stimuli. A more detailed discussion on the importance of features is given in Section 5.5.

6.3 Experiment about size

In this experiment, we investigate how the response differs depending on the size of the stimulus (RQ 2.2). The goal is to predict the size of the stimulus only using the feature matrix $\boldsymbol{R}$. We know the ground truth of the size of the stimulus. Because the stimulus was in five different sizes, we use

$$\mathbf{y} \in \{1, 2, 3, 4, 5\}^S$$

for the ground truth label. We trained an ℓ_1-regularized linear model on $\boldsymbol{R}$ and $\mathbf{y}$ by minimizing Equation (5.6) with the coordinate descent algorithm. The model included a bias term that was not regularized.

The same flat and nested cross-validation procedure as described in Section 6.2 is used. The only difference is that Equation (5.15) is used as the scoring function, and the λ parameter is varied in the range 10^{-3} to 10^0 in 20 steps evenly spaced on the $\log_{10}$ scale.

6.4 Experiment about shape

In this experiment, we investigate how the response differs depending on the shape of the stimulus (RQ 2.3). The goal is to predict if the presented stimulus was a square or circle by only using the feature matrix $\boldsymbol{R}$. We know the ground truth of the shape of the stimulus. For stimulus i, we assign the label $\mathrm{y}_i = -1$ if the stimulus was a square and $\mathrm{y}_i = +1$ if the stimulus was a circle. So ground truth label vector is

$$\mathbf{y} \in \{-1, 1\}^S .$$

An ℓ_1-regularized logistic regression model is trained by minimizing Equation (5.9) with the SAGA solver. The model included a bias term that was not regularized.

The same flat and nested cross-validation procedure as described in Section 6.2 is used. The only difference is that the λ parameter is varied in the range $10^{-1.5}$ to $10^{0.5}$ in 20 steps evenly spaced on the $\log_{10}$ scale.

6.5 Sensitivity analyses

We performed two kinds of sensitivity analyses. The first sensitivity study aims to assess the sensitivity of the final cross-validated model with regard to the regularization parameter λ. We investigate how the model's performance and complexity changes as a function of λ. The model complexity is measured as the number of nonzero coefficients in $\boldsymbol{w}$. The same range of λ as in Sections 6.2 to 6.4 is chosen. Therefore, we can use these results to validate the ranges of λ in the cross-validation stages in the previous experiments. Besides that, the results of this experiment are used to investigate how the models suffer from overfitting and underfitting by comparing the model's complexity with the model's performance. This sensitivity analysis is performed for all four decomposition methods.

The second sensitivity study investigates the sensitivity of the final cross-validated model with regard to the number of components K. We investigate how the model's prediction accuracy and the model's weights change as a function of K. Due to the computational complexity of the sDL decompositions, we could only perform this experiment for the ICA method. For this analysis we use $K \in \{2, 5, 10, 40, 80, 120\}$. We chose this range for K to obtain a clear picture while remaining computationally feasible.

6.6 Experiment with group lasso

In this experiment, the effectiveness of group lasso for position, size, and shape experiments is examined. Unlike before, we skip the Pearson correlation and directly unravel the time courses into feature matrix $\boldsymbol{R} \in \mathbb{R}^{S \times 37K}$ such that one row in $\boldsymbol{R}$ is $[\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_K^T]$. Therefore, $\boldsymbol{R} = \left[\boldsymbol{R}^{(1)}, \ldots, \boldsymbol{R}^{(G)}\right]$, highlighting the group structure of the feature matrix. The submatrix $\boldsymbol{R}^{(k)} \in \mathbb{R}^{S \times 37}$ is one group, namely the group belonging to the time courses of component k. The number of groups is equal to the number of components so $G = K$.

In the position and shape tasks, we minimize Equation (5.13). In the size task, we minimize Equation (5.12). Both cost function were minimized with the FISTA algorithm. The models included a bias term that was not regularized.

We only consider ICA and sDL3 as decomposition methods due to the extensive computational complexity. ICA is used for the position and size experiment, and sDL3

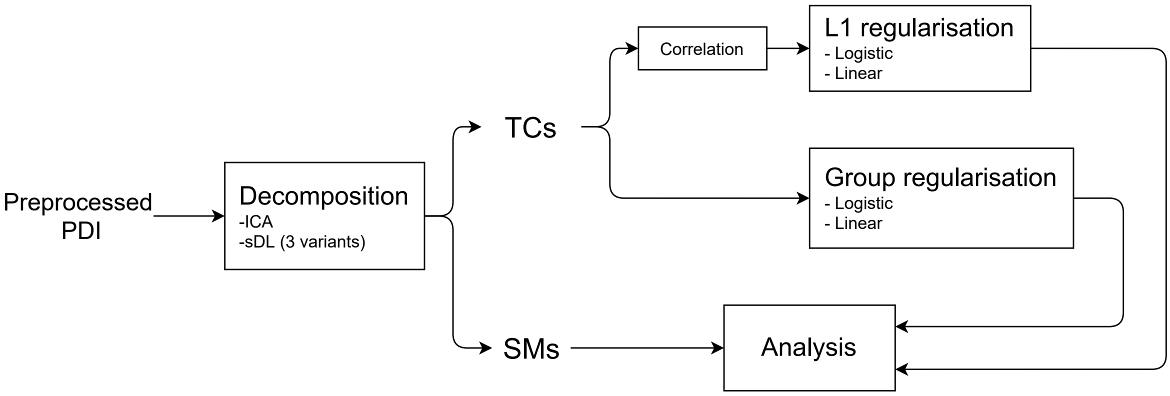


Figure 6.1: Pipeline - After preprocessing the input power Doppler images (PDIs), the images are decomposed into TCs and SMs. The TCs are further analyzed with linear and logistic models depending on the input task. The regularization function is either on the group or on the set of correlations as with ℓ_1-regularization. The output of the models is linked with the SMs from before and analyzed further.

for the shape experiment. Cross-validating λ was computationally expensive, so we hand-picked models with different λ values that provided good insight.

A pipeline of all steps is shown in Figure 6.1. In all experiments the preprocessed data is first decomposed with 4 different methods. The TC are either directly used for training group regularized models, or first correlated for the ℓ_1-regularized models. The weights of both ℓ_1- and group regularized models are analyzed together with the previously obtained SMs.

7 Results

In this section, the research questions will be answered. We start by showing the results of the decompositions in Section 7.1. Subsequently, we show the model's sensitivity on the regularization parameter and sensitivity on the number of components in Sections 7.2 and 7.3 respectively. In Section 7.4, we compare the predictive performance of the decompositions. Sections 7.5 to 7.7 answer neuroscientific RQs 2.1 to 2.3 respectively. Section 7.8 reveals additional insights obtained with the group lasso method.

7.1 Decomposition with ICA and sDL

The SMs obtained with ICA and the three variations of sDL are displayed in Appendix A. As seen in the color bar, red colors represent negative values and blue colors positive values. These sign of the values only hold compared to other voxels in the same SM because of the sign ambiguity. Only by multiplying the SM with its TC we can find the meaning of the component to be positive or negative about its baseline.

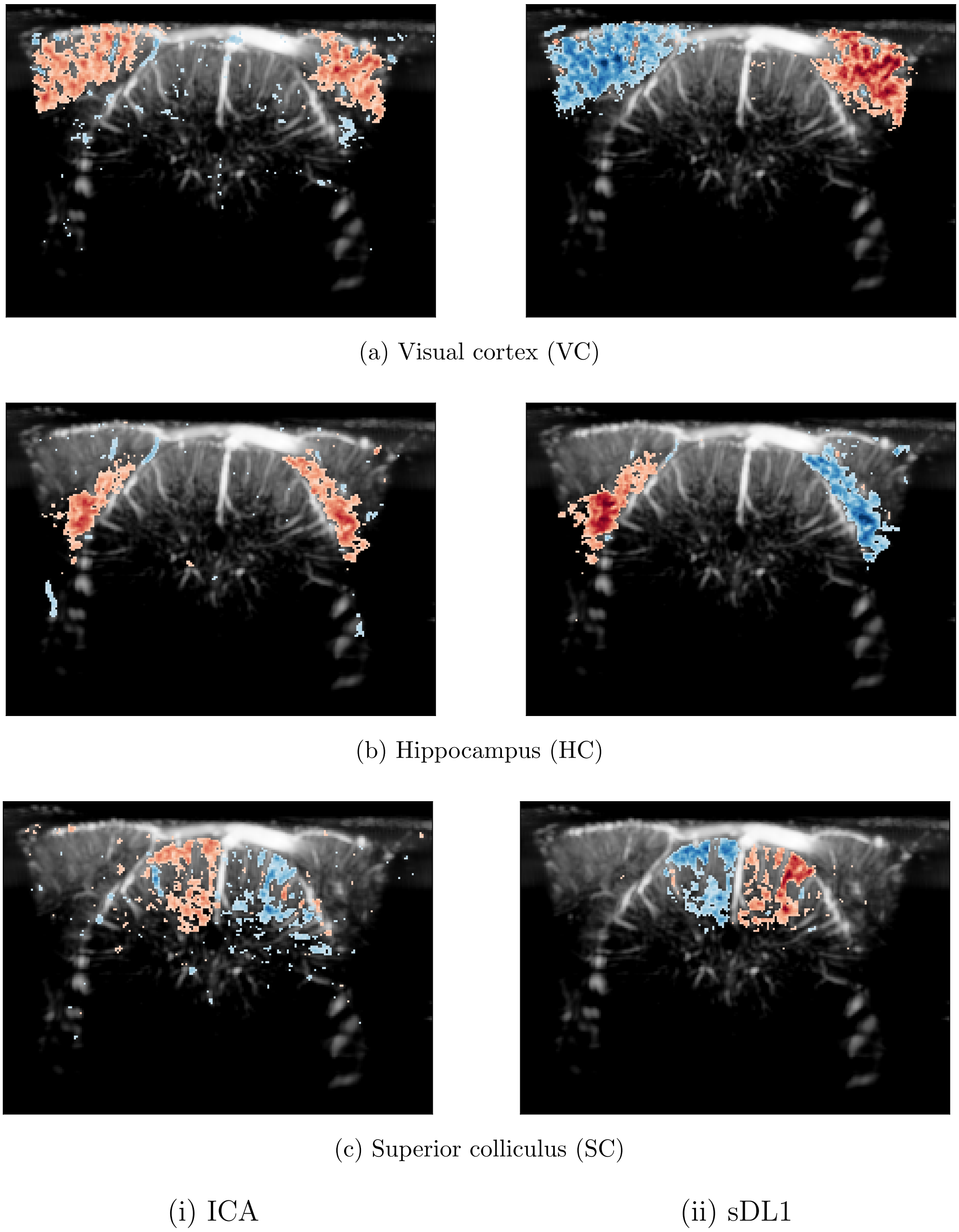
Even though the SMs are quantitatively different, we can qualitatively organize the important anatomical regions. In Table 7.1 it is laid out what indices from Figures A.1 to A.4 correspond to similar-looking spatial maps. Most decompositions extract SMs positioned in one of two hemispheres. The exception is sDL2, where the superior colliculus (SC) from both hemispheres get merged into one.

Two sets of SMs are displayed in Figure 7.1. Only the methods ICA and sDL1 are shown, but comparable SMs are found in the remaining decompositions. Similarities between the two hemispheres and the two decompositions are clear. We found the anatomical names by comparing the images with the Allen Mouse Brain Atlas [31]. Figure 7.1(a) corresponds to the visual cortex (VC), Figure 7.1(b) corresponds to the hippocampus (HC), and Figure 7.1(c) corresponds to the SC[1].

[1] Confirmed in personal communication by neuroscientists at the CUBE group

Table 7.1: SMs extracted by the decomposition methods are similar. Values enclosed in parentheses denote a map that also encompasses another region.

Region	Location	ICA	sDL1	sDL2	sDL3
Visual cortex (VC)	Upper right	69	25	8	25
	Upper left	58	7	7	7
Hippocampus (HC)	Lower right	54	1	(8)	1
	Lower left	34	8	(7)	8
Superior colliculus (SC)	Middle right	21	15	61	15
	Middle left	2	22	(61)	22



(a) Visual cortex (VC)

(b) Hippocampus (HC)

(c) Superior colliculus (SC)

(i) ICA

(ii) sDL1

Figure 7.1: Extraction of mirrored components from both hemispheres. For comparison purposes, we merged the lateral components into one and overlayed that image onto the mean image from Figure 3.1. The left column (i) contains SMs extracted with the ICA method and the right column (ii) from the sDL1 method.

Table 7.2: Average number of nonzero coefficients in every column of $\boldsymbol{B}$ for each decomposition method.

Method	$\frac{1}{N}\sum_{n=1}^{N}\|\boldsymbol{\beta}^{(n)}\|_0$
ICA	80
sDL1	29
sDL2	17
sDL3	1

The average number of nonzero coefficients in every column of $\boldsymbol{B}$ is given in Table 7.2. The value represents the average number of components that influence a single voxel. Large values represent contributions from many components per voxel while small values represent few contributions. As expected, the ICA method does not enforce any sparsity. Thus every voxel is explained by K components. Contrarily, sDL3 enforces the strongest sparsity where every voxel is explained by only one component. The remaining methods offer mediocre parsimony.

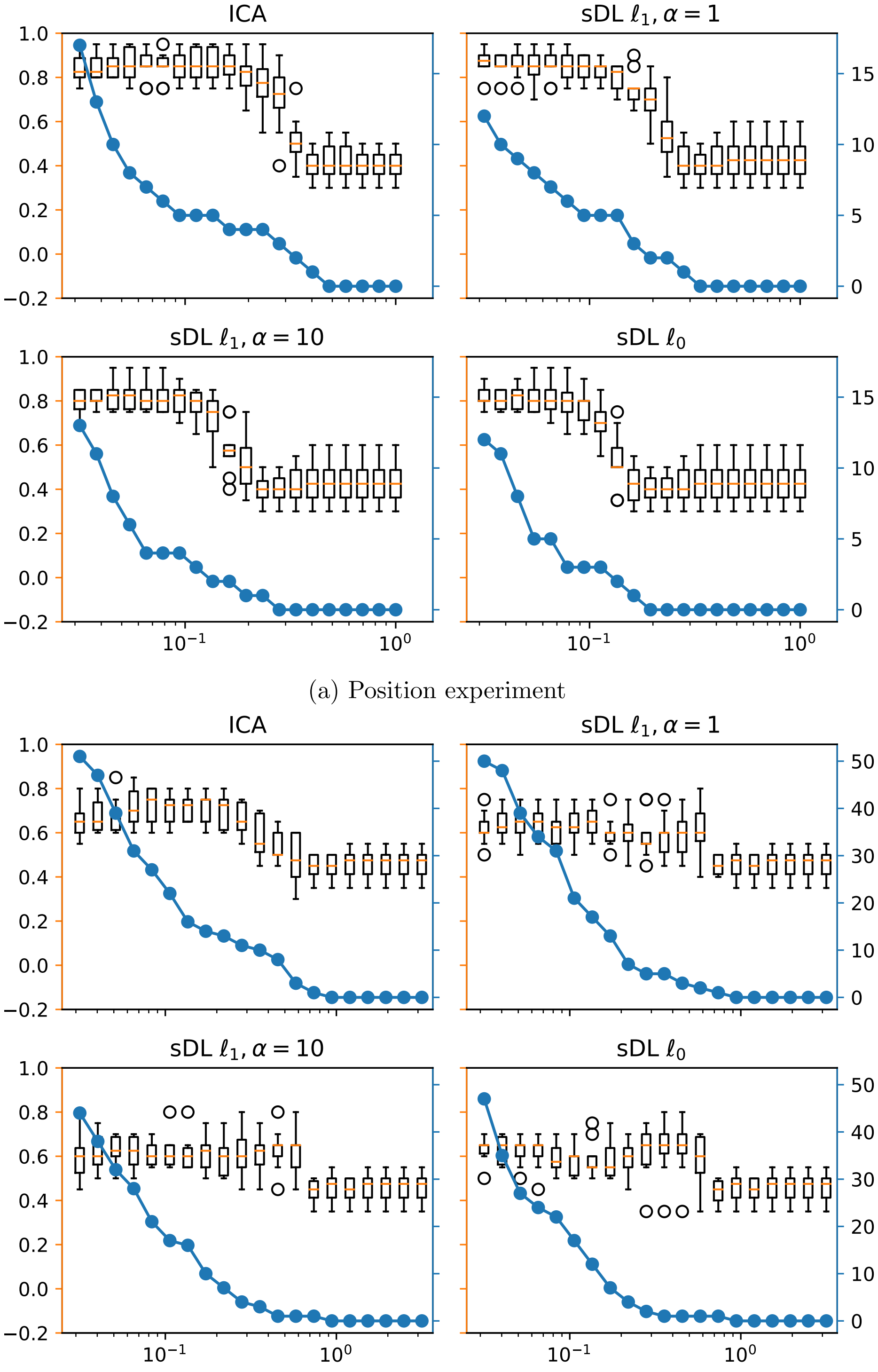
Even though ICA has the largest number of nonzero components, it is unlikely that all components have equal influence on a single voxel. Because of the independence constraint it is likely that only a small number of components have considerable influence on an individual voxel and the remaining components are close to zero.

7.2 Sensitivity on regularization parameter

A sensitivity analysis for the classification models with regard to parameter λ is performed. The results of that analysis are shown in Figure 7.2. The box plots show the range of the cross-validated scores. The ℓ_0-norm of $\boldsymbol{w}$, the number of nonzero elements, serves as a proxy for model complexity and is displayed with a line plot.

The standard convention for box-and-whisker plots is used. "The box extends from the lower to upper quartile values of the data, with a line at the median. The whiskers extend from the box to show the range of the data. Flier points are those past the end of the whiskers" [32]. The range of the data is defined as 1.5 times the interquartile range above the upper quartile and 1.5 times the interquartile range below the lower quartile.

In Figure 7.2(a) the sensitivity of the classification accuracy in the position experiment is shown. It explored the sensitivity of the κ score and the model complexity to hyperparameter λ. Each of the four graphs contains a box plot. In each graph, the box plot can be broken up in three regions (from right to left): 1. a baseline region (around 0.5), 2. a transition region and 3. a plateau region (around 0.85). It is clear that for large values of λ, $\boldsymbol{w} = \boldsymbol{0}$ and the model's prediction is a random guess. As we decrease λ, the model enters a transition region where the model's complexity rises and simultaneously improves the prediction accuracy. For smaller λ, the prediction plateaus, while model complexity keeps increasing. These patterns are broadly consistent for all four decomposition methods. After the performance score has plateaued the model complexity keeps increasing. This could be problematic for our purposes. We find that



(a) Position experiment

(b) Shape experiment

Figure 7.2: Cross-validated accuracy κ (box plots, left y-axis) vs regularization parameter λ (x-axis). Besides that, ℓ_0-norm of $\boldsymbol{w}$ (line plot, right y-axis) is given for the same values of λ. Upper left: ICA- Upper right: sDL1 - Lower left: sDL2 - Lower right: sDL3

the plateau is reached with only a few nonzero coefficients. This indicates that only a select group of SMs contain valuable information for the position classification task.

The sensitivity of the classification accuracy in the shape experiment is illustrated in Figure 7.2(b). Similarly to Figure 7.2(a), we can split the box plots into three regions. However, the differences between the regions are more subtle than in the position experiment. The transition region in the ICA decomposition is linear, meaning that model complexity and accuracy increase simultaneously over a range of lambdas. This steady increase is unlike the other three decompositions, where the transition is short and over a much shorter range of lambdas. It is sharp because a slight increase in model complexity from $\|\boldsymbol{w}\|_0 = 0$ to $\|\boldsymbol{w}\|_0 = 1$ gives a (near) optimal change in the shape classification task. This result implies that ICA finds various components slightly related to the shape task while the sparsity inducing methods find only one that is strongly related.

Finally, we performed a similar sensitivity analysis for the size experiment detailed in Figure 7.3. It explored the sensitivity of the R^2 score and the model complexity to hyperparameter λ. The results indicate that the R^2 score is sensitive to the regularization parameter. For large lambdas the model predicts the average size, because the bias term is equal to the mean size $\overline{\mathrm{y}} = 3$ and therefore the $R^2 = 0$. As we decrease λ, the model complexity and R^2 score increase linearly until reaching an optimal value. After that, model complexity keeps increasing while the R^2 score decreases, suggesting over-fitting. The results indicate that a large group of SMs give a similar amount of information for the size regression task. Only sDL3 shows aberrant behavior. Unlike the other models, at its optimal R^2 value there is a large variation in cross-validated scores, as seen in the lengths of the whiskers at the largest median value. This could be a sign of over-fitting. Therefore the cross-validation is possibly ineffective.

In both the position and shape sensitivity analysis we find for sufficiently small λ where the prediction score is insensitive to λ while the model complexity is sensitive to λ. This means that the model complexity keeps increasing while the prediction score remains approximately constant (reached the plateau region). This could be problematic. If the prediction score is insensitive to λ, then the cross validation algorithm does not have a clear “winner” and can be indifferent in the λ that it selects. Any trained model with a sufficiently small λ could be selected as the optimal model. This is not a problem for the predictive performance (as the predictions are stable), but it is a problem for the model's interpretability. If the cross-validation algorithm is indifferent, then it could select a model with high complexity by chance. This slight increase in model performance accompanied by large increase in model complexity is undesirable.

7.3 Sensitivity on number of components

In this section, we investigate the sensitivity of the prediction models on the parameter K, the number of components. The cross-validated prediction scores of the position, size, and shape task are plotted for 6 values of K in Figure 7.4. We observe for a small number of components ($K = 2$ or $K = 5$) that the predictions are unsuccessful in all tasks. For $K = 10$, the position experiment significantly improves. As we see in Figure 7.5(a), this is caused by the extraction of the left and right VC. The other

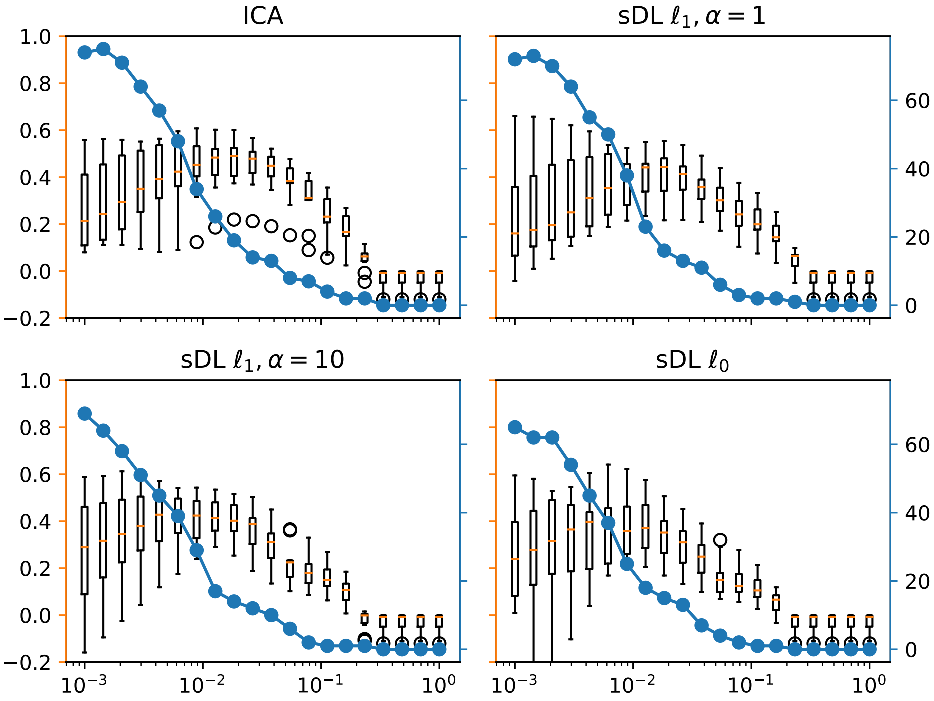


Figure 7.3: Size experiment - Cross-validated coefficient of determination R^2 (box plots, left y-axis) vs regularization parameter λ (x-axis). Besides that, ℓ_0-norm of $\boldsymbol{w}$ (line plot, right y-axis) is given for the same values of λ. Upper left: ICA - Upper right: sDL1 - Lower left: sDL2 - Lower right: sDL3

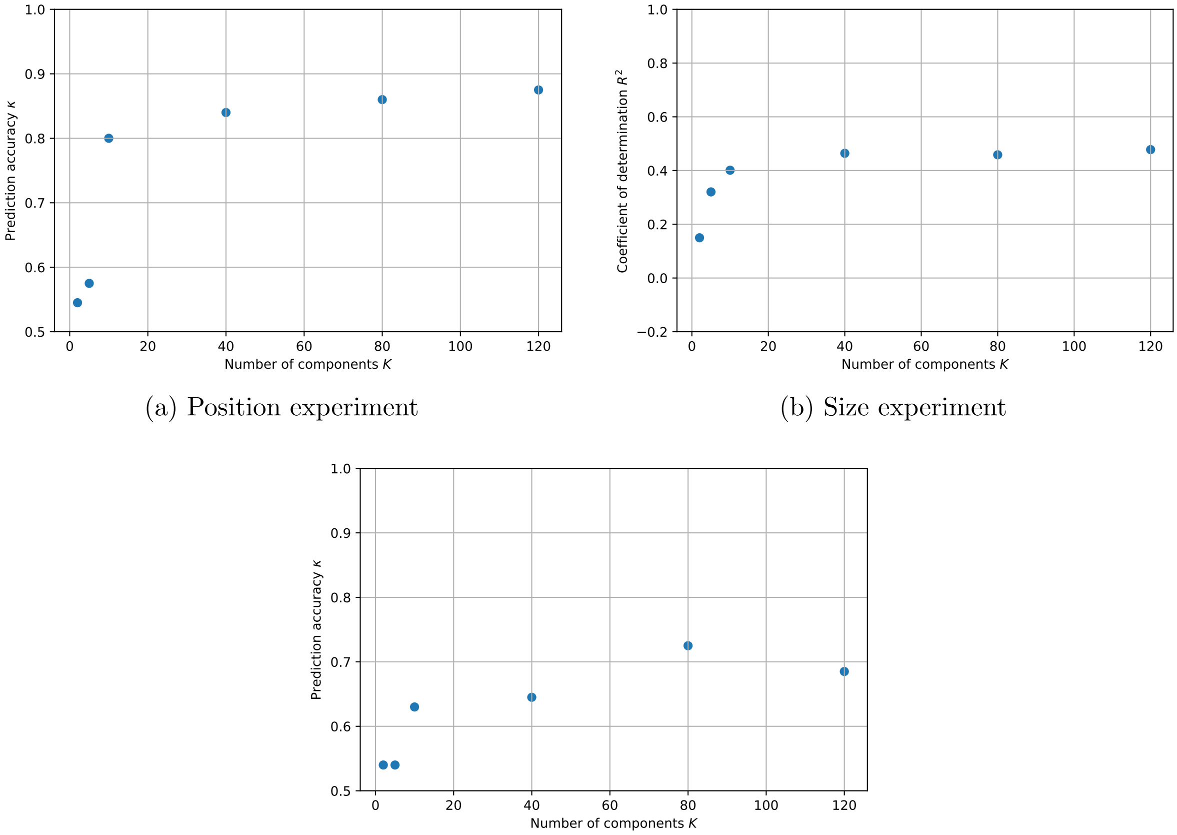
two tasks also improved slightly, as seen in Figures 7.4(b) and 7.4(c). For the three largest values of K, the prediction results are mostly constant. The exception is the shape experiment at $K = 80$. Therefore, we conclude that the prediction accuracies are mostly insensitive to the K parameter when K is sufficiently large.

We also analyze the components of best classifier for the four largest K's in the position experiment. We want to investigate to what extent similar spatial maps are found. We only consider the four largest K's because they all had good classification accuracy. We find that Figures 7.5(b) to 7.5(d) give the HC largest weights. For $K = 10$, seen in Figure 7.5(a), the HC was not separated into both hemispheres. The VC was separated into two components and therefore given the largest weights.

We conclude that for a sufficiently large K, the most important features are stable. However, this is only tested for a small set of K's in the position task and only when using the ICA method. Therefore this finding should not be extrapolated to the other experiments and should function as an illustration.

7.4 Comparison of performance

We start this section by giving the generalization performance of the four methods obtained with nested cross-validation. These scores are unbiased and therefore their performances can be fairly compared. However, the performance scores are derived from



(a) Position experiment

(b) Size experiment

(c) Shape experiment

Figure 7.4: Sensitivity of prediction scores on parameter K, the number of components. We plot the best cross-validated prediction score (κ or R^2) for 6 values of K. Only the ICA method is examined for computational purposes.

models with different λ parameters and different model weights. Therefore we can only use these models to investigate about the generalization performance.

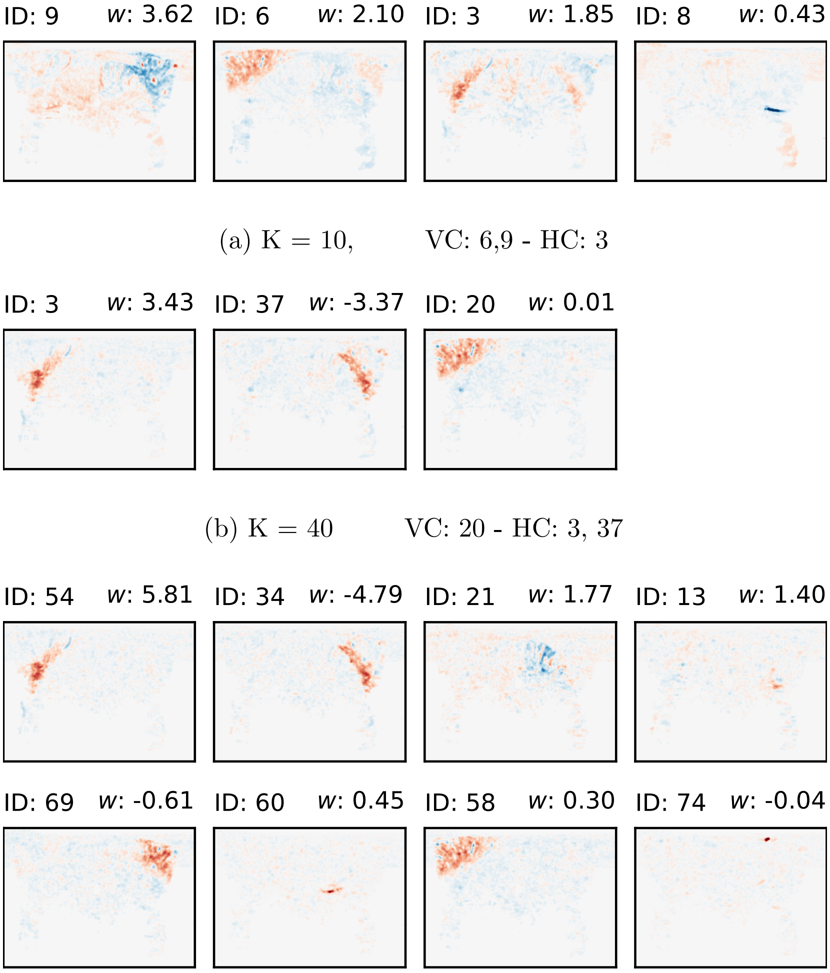
The unbiased estimations of generalization performance are given in Table 7.3. The presented scores are calculated taking the average and standard deviation of the unbiased estimations. We find that all decompositions obtain similar generalization performances in the position experiment. There are large variations in the R^2 score between ICA and sDL3 in the size experiment. The same is true for the shape experiment. Based on these observations we can say that the most predictive information is available in the ICA method and the least predictive information in sDL3.

We will train the models a second time with the flat cross-validation algorithm. The models trained with the standard cross-validation will be analysed in the remainder of the chapter. We should be careful, in comparing the performance scores because they are biased. However, by comparing Table 7.3 with Tables 7.4 to 7.6 we find the performances are slightly better than unbiased estimation, but mostly comparable. The exception is the size experiment where large differences are observed. Therefore the performances in the size experiment should be regarded with care.

Table 7.3: Unbiased estimation of generalization performance in the position, size and shape task.

Method	Position		Size		Shape	
	mean κ	std κ	mean R^2	std R^2	mean κ	std κ
ICA	0.85	0.09	0.43	0.15	0.70	0.09
sDL1	0.83	0.09	0.36	0.14	0.64	0.12
sDL2	0.81	0.11	0.28	0.16	0.65	0.09
sDL3	0.81	0.10	0.24	0.16	0.59	0.09

The results of the position experiment are given in Table 7.4. All decompositions can achieve mostly similar classification performances. The minor variations in cross-validated means and standard deviations give a slight advantage to the sDL1 method.

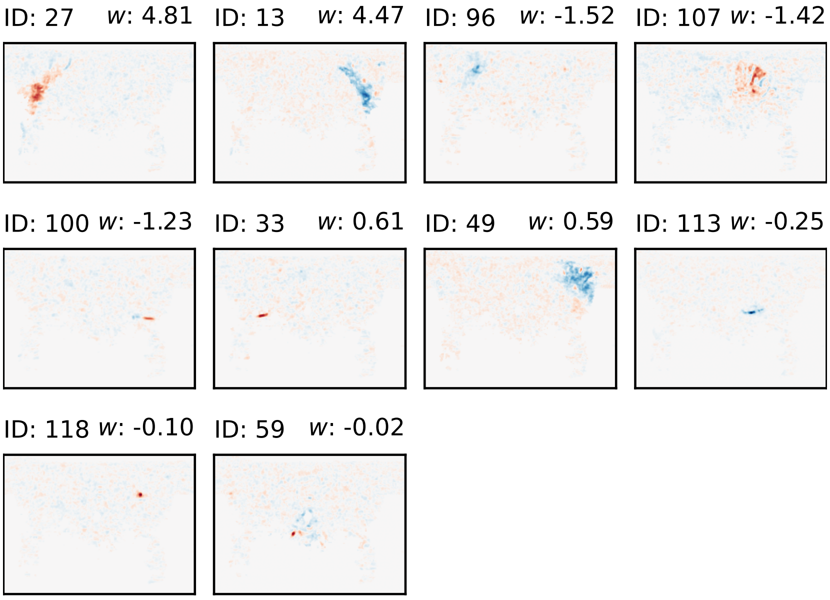


(a) K = 10, VC: 6,9 - HC: 3

(b) K = 40 VC: 20 - HC: 3, 37

(c) K = 80 VC: 58, 69 - HC: 34, 54 - SC: 21

Figure continued on next page



(d) K = 120 VC: 49 - HC: 13,27 - SC: 107

Figure 7.5: Position experiment - Illustration of the stability of the weights for 3 values of K.

Additionally, we prefer solutions with few nonzero components for interpretability. This preference for sparse solutions means that ICA and sDL2 are helpful too. sDL3 is just as sparse as ICA but has poorer accuracy so therefore slightly worse overall.

The bias term requires an explanation. All methods find a positive bias which means that the model classifies the stimulus to be "right" by default. We speculate that is because left stimuli are more noticeable in all decompositions. Right stimuli are relatively less detectable, and therefore the model obtains a positive bias.

The results of the size experiment are given in Table 7.5. Again, the mean and standard deviation of R^2 are mostly comparable, although the ICA and sDL2 methods perform slightly better. ICA excels in the mean score while the sDL2 method has slightly smaller standard deviation and model complexity. For a visual understanding of what an $R^2 = 0.46$ score resembles, we compare the true sizes and predicted sizes visually in Appendix B.

The results of the shape experiment are given in Table 7.6. ICA outperforms the other methods. It achieves considerably better scores, albeit with the largest model complexity. This complexity renders the model tough to interpret. The sDL2 and sDL3 method find an optimal classifier with one nonzero coefficient in the model. These models also change the bias slightly negative such that it predicts a square by default. For the model to make sense, we can deduce that the nonzero coefficient turns the model positive in particular cases and is therefore related to circle stimuli.

To conclude, only the shape experiment had notable differences in performance. In the position and size experiment, the scores and biases were alike.

Table 7.4: Position experiment - Comparison of the different decomposition methods in terms of biased classification performance, model complexity and bias.

Method	mean κ	std κ	$\|\boldsymbol{w}\|_0$	Bias
ICA	0.86	0.07	8	0.50
sDL1	0.87	0.05	12	0.41
sDL2	0.82	0.06	6	0.58
sDL3	0.82	0.05	8	0.27

Table 7.5: Size experiment - Comparison of the different decomposition methods in terms of biased regression performance, model complexity and bias.

Method	mean R^2	std R^2	$\|\boldsymbol{w}\|_0$	Bias
ICA	0.46	0.10	19	1.95
sDL1	0.41	0.10	23	2.01
sDL2	0.42	0.08	17	2.21
sDL3	0.38	0.13	25	2.14

Table 7.6: Shape experiment - Comparison of the different decomposition methods in terms of biased classification performance, model complexity and bias.

Method	mean κ	std κ	$\|\boldsymbol{w}\|_0$	Bias
ICA	0.73	0.08	27	0.32
sDL1	0.65	0.06	17	0.21
sDL2	0.63	0.09	1	-0.04
sDL3	0.64	0.11	1	-0.04

7.5 Results of position experiment

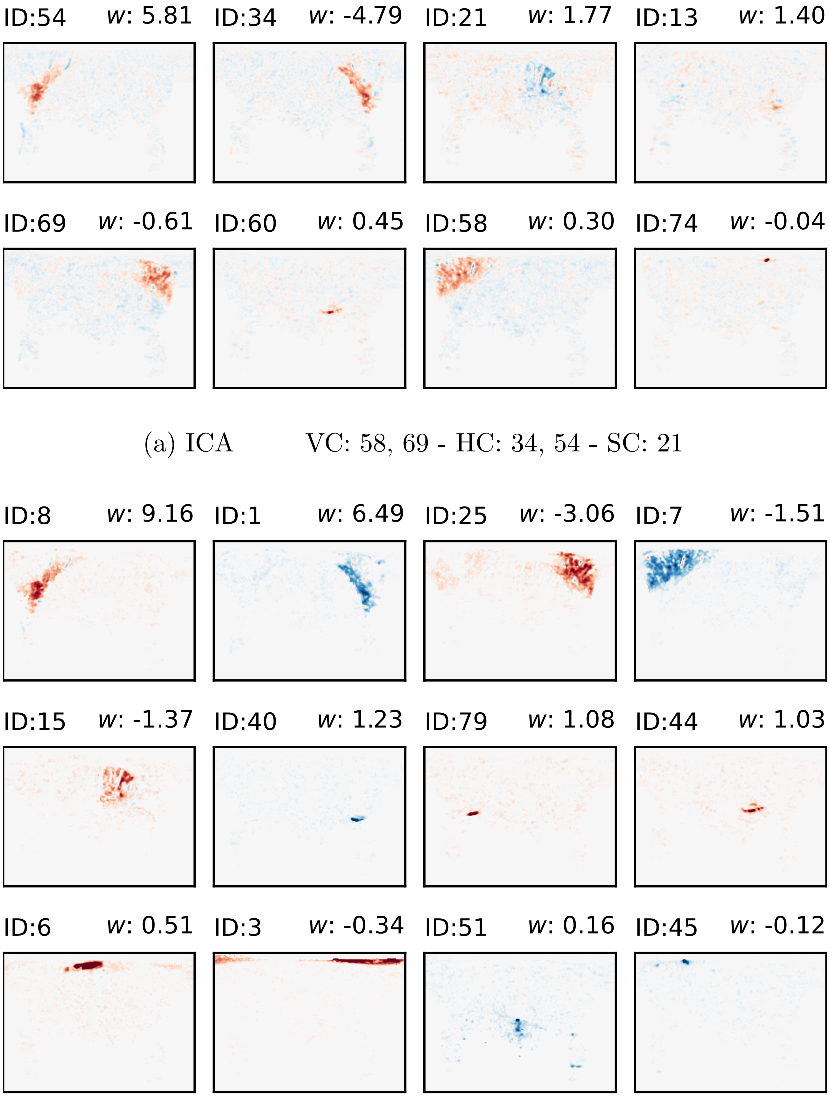
We analyze the optimal classifiers from Table 7.4 in more detail. In Figure 7.6 all extracted SMs are displayed. The weights of the SMs are given on the top right side of each image. The SMs are ordered by the absolute value of their weight.

In Figure 7.6 we observe large weights for the left and right HC. The left HC has positive weight and therefore pushes the classifier into predicting a 'right' stimulus. Vice versa for the right HC. The right SC is found in Figures 7.6(a) and 7.6(b) but not in Figures 7.6(c) and 7.6(d). That is odd because sDL3 had extracted that region. The VC is found with varying importance. In Figures 7.6(a) and 7.6(d) it is given very minor weights while in Figure 7.6(b) it is more significant. In general, the HC contains the most explanatory information for the position classification task. The importance of the VC varies for different decompositions, it appears valuable but not as much as the HC. Only the right SC is meagerly important.

The left and right VC, HC, and SC naturally allow themselves to be compared. Figure 7.7 plots the correlation values of the left and right HC, VC, and SC against each other. We observe a trend in the data approximately along the diagonal. This is particularly visible in the HC and SC and to a lesser extent in the VC. The hyperplane separating the two classes is also approximately along the diagonal. Therefore, the discriminatory information is in the difference between the two correlations, i.e., which correlation is larger. In a stimulus from the right, the left hemisphere has a stronger correlation than the right hemisphere and vice versa.

7.6 Results of size experiment

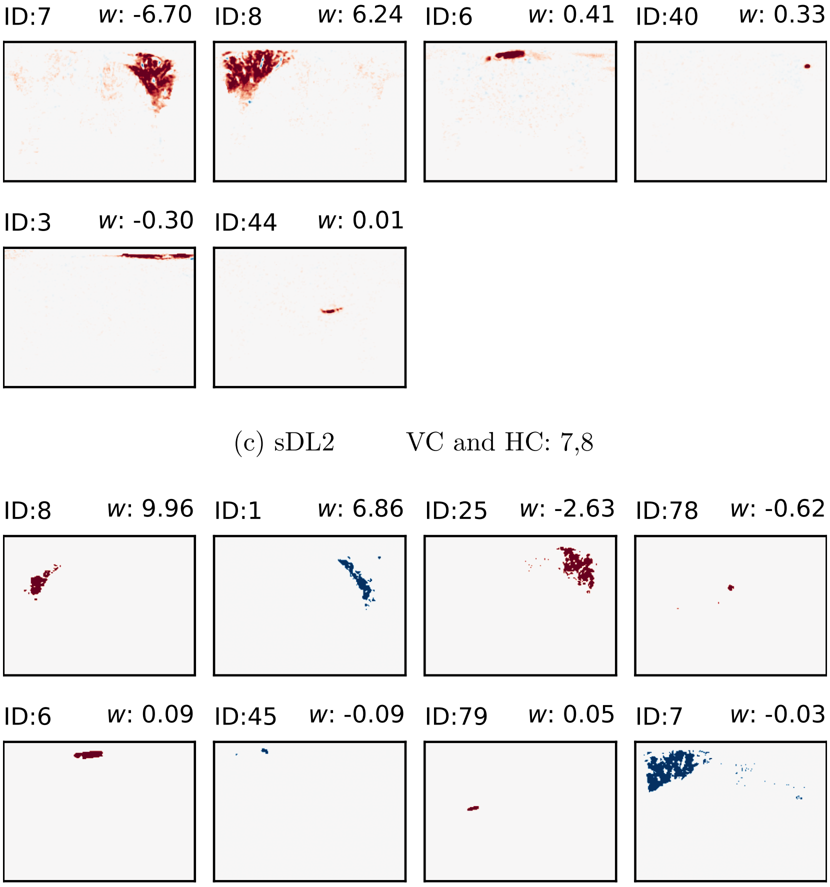
We want to extract neuroscientific understanding using the optimal model, similar to the previous section. However, as given in Table 7.5, all optimal models have many nonzero coefficients. The SMs from the ICA method are shown in Figure 7.8. We only show this decomposition as this was the best performing one and the other decomposition



(a) ICA VC: 58, 69 - HC: 34, 54 - SC: 21

(b) sDL1 VC: 25, 7 - HC: 8, 1 - SC: 15

Figure continued on next page



(c) sDL2 VC and HC: 7,8

(d) sDL3 VC: 7, 25 - HC: 1,8

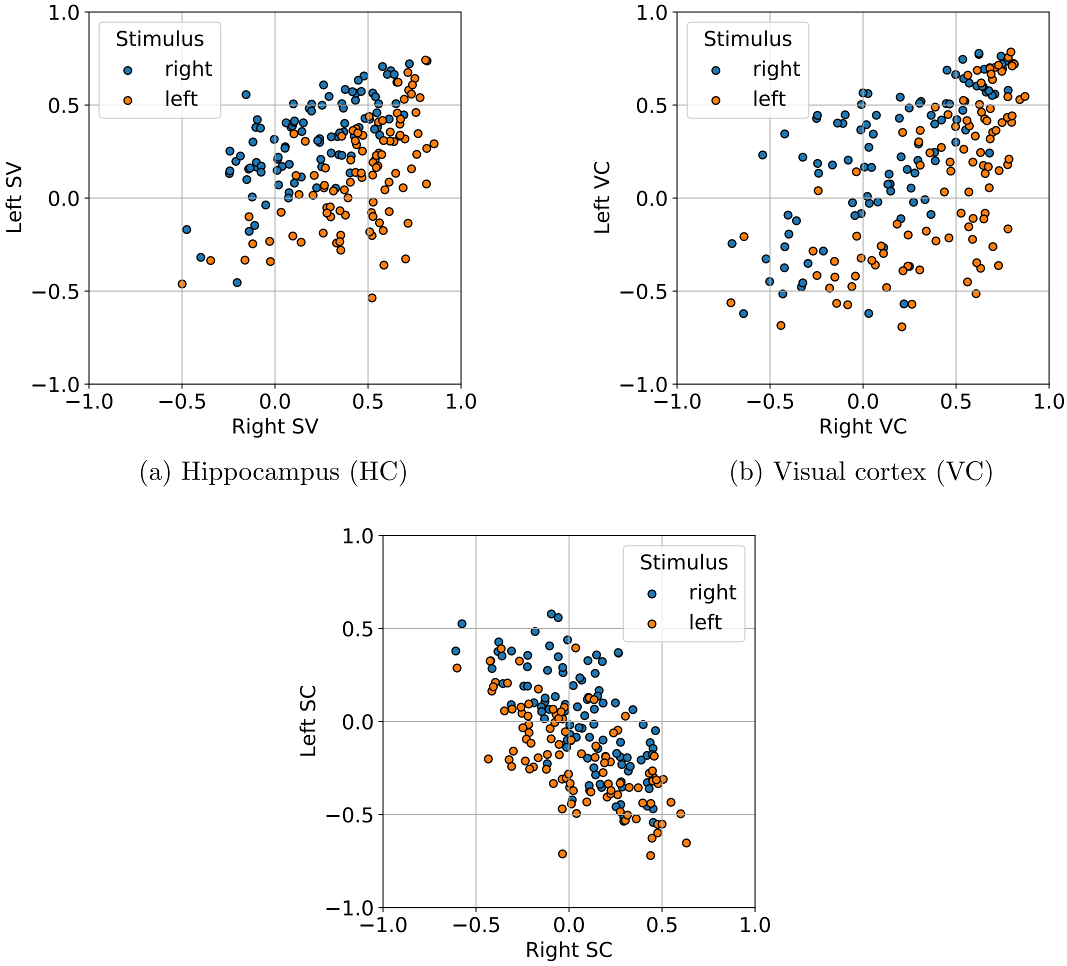
Figure 7.6: Position experiment - Spatial maps found by the best performing models. The weight for each SM is given in the top right corner. SMs are ordered by absolute value of their weight.

methods did not provide new information. The other SMs are shown in Appendix B.

We find that the left and right VC and HC are extracted. The model does not consider SC to be important. Also fifteen other SMs are extracted. These fifteen SMs are difficult to understand anatomical. Therefore, it is not easy to interpret the extracted models from the size experiment.

We can analyze the correlation from the left and right HC in detail. In Figure 7.9 we plot the variations of the correlations for each size. As expected, a larger input size causes a stronger response. The right HC gives stronger and less variation in the responses than the left HC, which explains why the model gave it a larger weight.

Another observation is that our model finds in the HC only minor differences in the correlation values between sizes 4 and 5. Also, between sizes 1 and 2, a slight change in correlation value was observed. These resemblances might explain why the model scored poorly in the coefficient of determination score.



(a) Hippocampus (HC)

(b) Visual cortex (VC)

(c) Superior colliculus (SC)

Figure 7.7: Position experiment - Scatter plot of the correlation values, comparing the left and right hemisphere of (a) the HC, (b) the VC, and (c) the SC. The y-axis displays the correlations on the left hemisphere and the x-axis the correlations on the right hemisphere.

7.7 Results of shape experiment

In the shape experiment we found significant differences among the optimal models. As seen in Table 7.6, the number of nonzero coefficients and model accuracy vary significantly. We investigated all models and found that sDL1 method provided no additional information. Therefore, we only describe the other three models in this section. To be thorough, the components of sDL1 are shown in Appendix C.

sDL2 and sDL3 methods find component 13 and 18 respectively, which relates to the same anatomical region, as shown in Figures 7.10(b) and 7.10(c). The correlation values of this component are detailed in Figure 7.11. The circle and square stimuli have different correlation distributions. Square stimuli are on average negatively correlated while circle stimuli are on average positively correlated. Accordingly, circle stimuli elicit on average an increase in blood flow compared to the baseline. In Figure 7.12 the component is

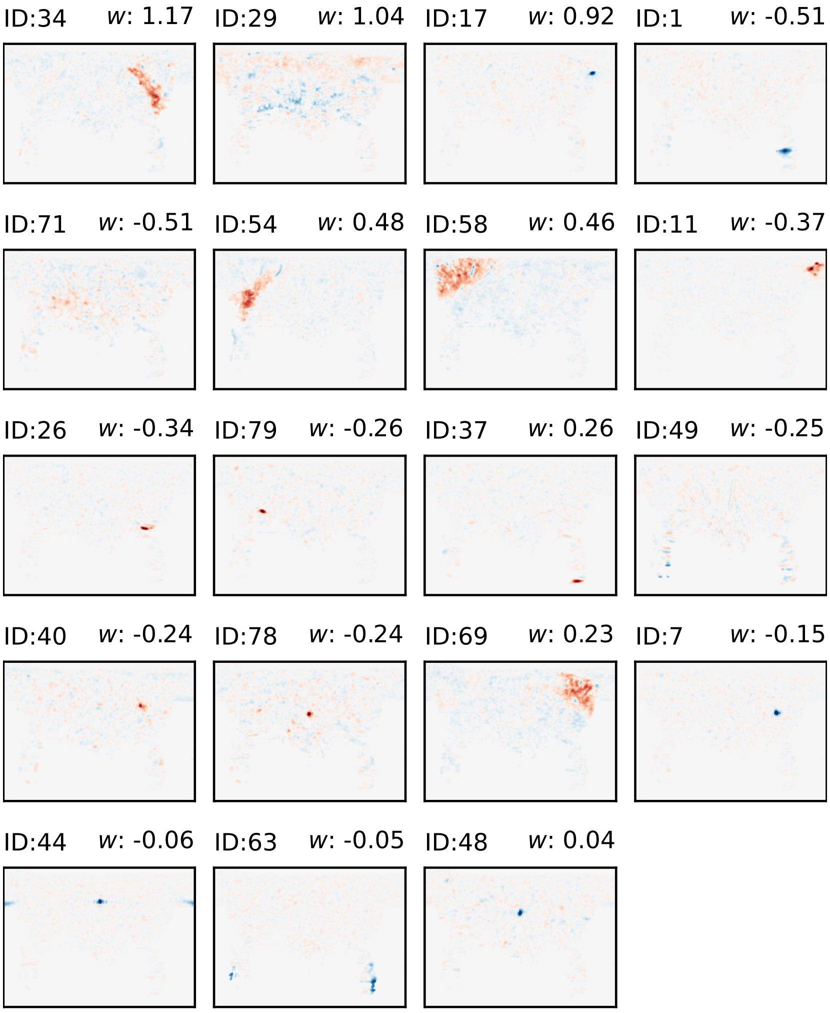


Figure 7.8: Size experiment - Spatial maps found by the best performing model. The weight for each SM is given in the top right corner. SMs are ordered by absolute value of their weight. Only the SMs from the ICA method are shown.

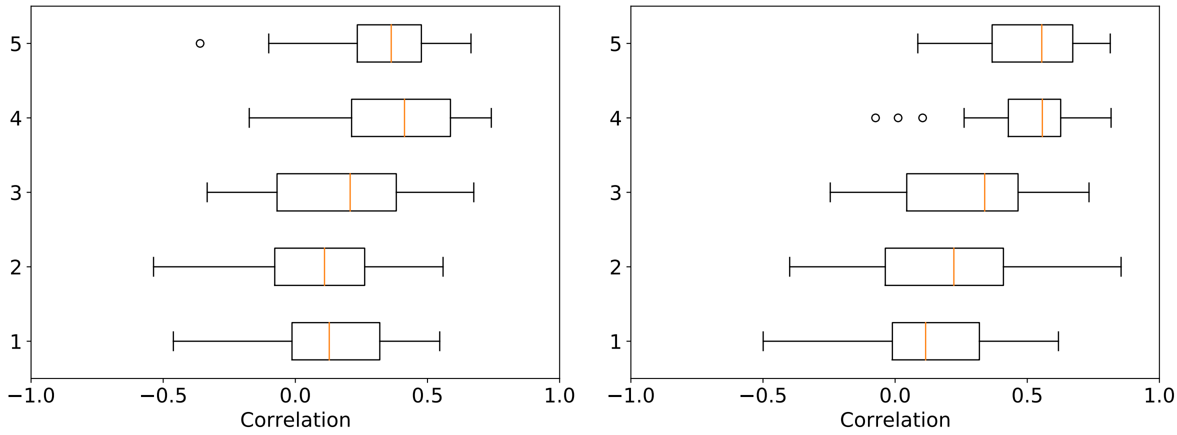
VC: 58, 69 - HC: 34, 54

displayed on the mean image of Figure 3.1. This component is hypothesized to be a vessel that supplies blood to or receives blood from the HC [2].

We give two explanations for this observation. Firstly, there could be a different hemodynamic response depending on the type of shape presented to the mouse. In this case the underlying physiological explanation is not clear. Secondly, we repeat that the the stimuli are presented in two blocks, where the first block consisted of squares and the second of circles. The differences could be explained by the state of excitement or boredom in the mouse. The available data can not disclose this issue.

The best performing classifier was made with ICA. Its SMs are displayed in Figure 7.10(a). Component 19 resembles the same component as in Figure 7.12. However

[2] Confirmed in personal communication by neuroscientists at the CUBE group



(a) Left HC

(b) Right HC

Figure 7.9: Size experiment - Box plots of the left and right HC's correlations for five different stimuli sizes represented on the y-axis. The smallest stimulus size is on label 1 and the largest on label 5.

in this classifier, component 17 received a larger weight. The correlation values of these components are compared in Figure 7.13. It is clear that indeed, component 19 does not have the same discriminatory value as it had in Figure 7.11. Component 17 is positively correlated with square stimuli while it appears neutral for circle stimuli.

7.8 Results obtained with group lasso

This section highlights the new results and insights obtained with a group lasso regularization for classification and regression. As before, we tackle the position, size, and shape problem individually. Each section is started by briefly showing the sensitivity of the performance on the regularization parameter. After that, we explore two models obtained with two different value of λ giving a different model complexity.

Because the Pearson correlation step was omitted we can use the full time scale in the model. The results obtained with this method can be used to validate or reject the assumptions made in the Pearson correlation step.

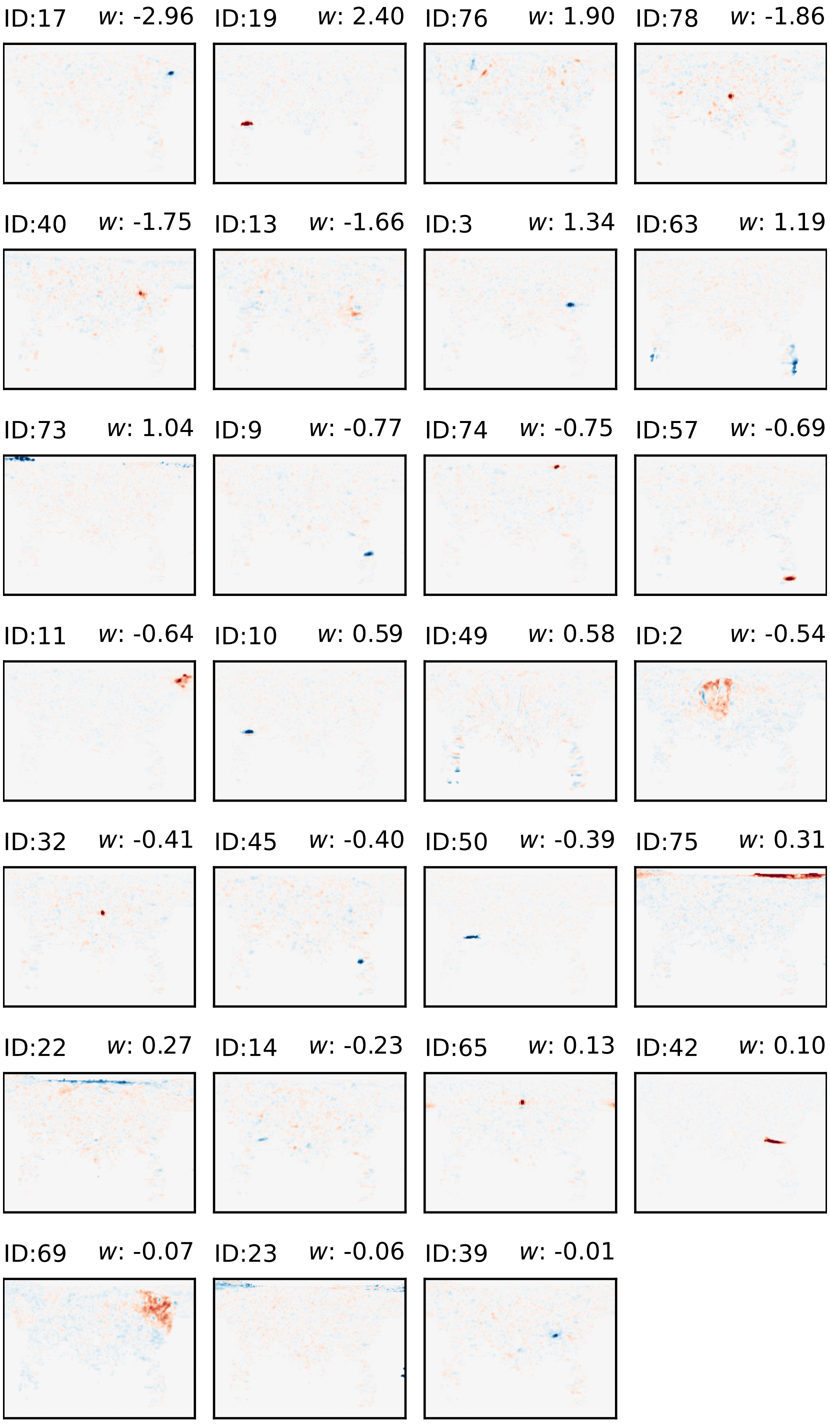
Position experiment

Figure 7.14 provides the sensitivity analysis results for the position classification problem. This figure illustrates that the classification accuracy κ is stable for sufficiently small λ.

At $\lambda = 2 \cdot 10^{-2}$, only two nonzero groups can provide a biased cross-validated accuracy of $\kappa = 0.92$. This is significantly more accurate than any model from Table 7.4. However, it is premature to say that the performance is better because we are comparing biased estimations. Unfortunately a nested cross validation was computationally too expensive.

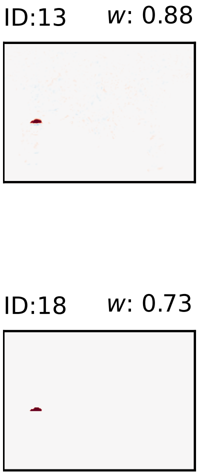
Figure 7.15(a) displays the SMs and the weights for every time sample belonging to these two groups. We find that the left and right VC provide the most useful discriminatory information.

We find nonzero weights for the whole time scale. Therefore, the whole time scale



(a) ICA

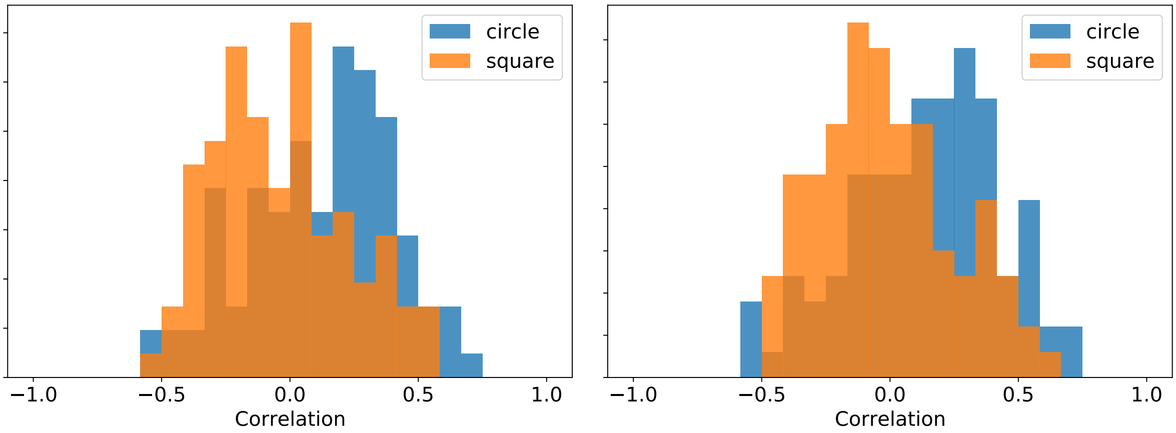
Figure continued on next page



(b) sDL2

(c) sDL3

Figure 7.10: Shape experiment - Spatial maps found by the best performing models. The weight for each SM is given in the top right corner. SMs are ordered by absolute value of their weight.



(a) sDL2

(b) sDL3

Figure 7.11: Shape experiment - Histogram of the correlations related to component 13 and 18 for sDL2 and sDL3 methods respectively.

contains discriminatory information. We can dissect the full time scale in three regions:

(1) Pre-response undershoot. This region ends at the first intersection of the x-axis

(2) Main response

(3) Post-response undershoot. This region starts at the second intersection of the x-axis

We remark that the pre- and post-response undershoot both have negative weights. The compelling explanation is that the normalization step in the preprocessing causes negative values in $\boldsymbol{X}$. The negative weights are more significant in the pre-response

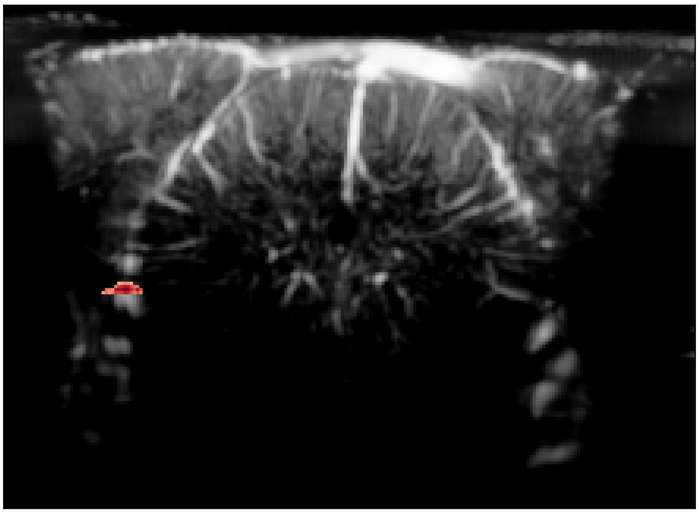
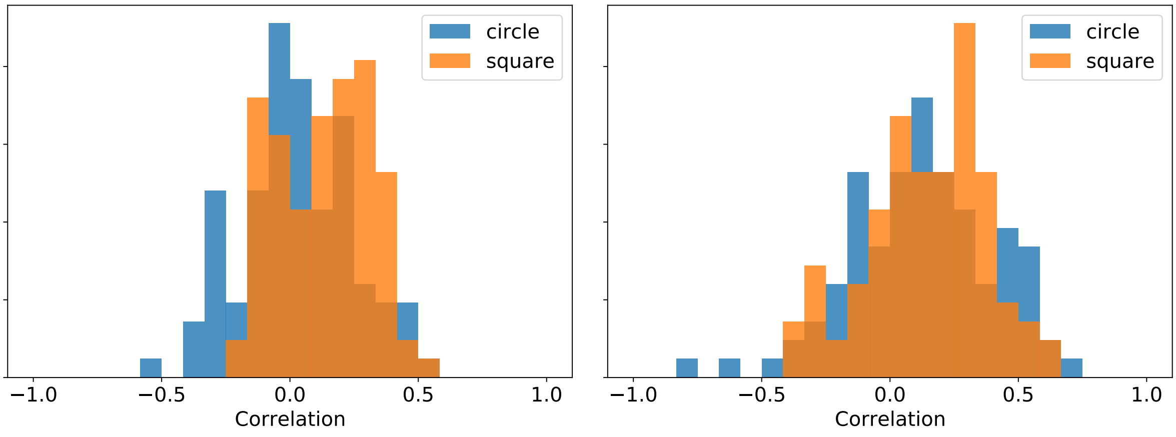


Figure 7.12: Shape experiment - Component 18 from sDL3 method displayed on the mean image of Figure 3.1.



(a) Component 17

(b) Component 19

Figure 7.13: Shape experiment - Histogram of the correlations related to component 17 and 19 for the ICA method

period than in the post-response period, suggesting that the time courses had not settled to a steady baseline before the next stimulus was shown.

Besides that, we observe a double-peak structure in the VC. The weights in the positive response dip at approximately sample 18 but rise a second time shortly after that. In the trough, there is less discriminatory information between left and right. We investigate the average time courses of the VC in Figure 7.16 for (a) the left VC and (b) the right VC. The time courses are averaged for each time sample to obtain a single time course. We observe that for the right VC, there is an approximate equal response around time sample 18 in the left and right stimuli. This means that there is barely any discriminatory information and that explains the trough.

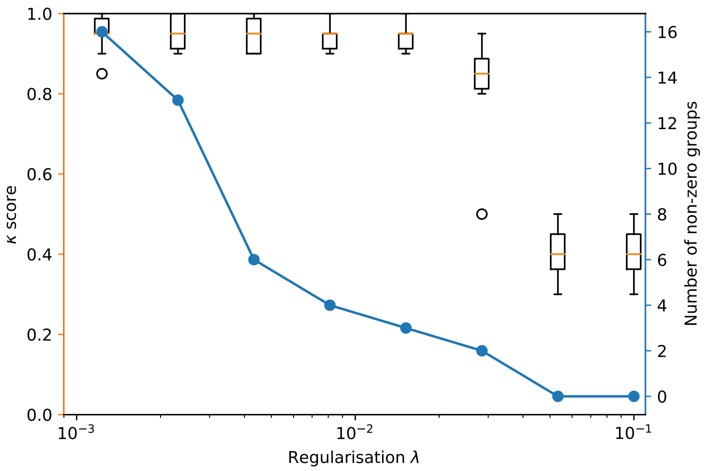


Figure 7.14: Group lasso for position experiment - Cross-validated predicted accuracy κ (box plots, left y-axis) vs regularization parameter λ (x-axis). The line plot shows the number of nonzero groups in $\boldsymbol{w}$ (right y-axis) for the same values of λ. These results are obtained with the ICA method.

We decreased the regularization to $\lambda = 8 \cdot 10^{-3}$ which allows for an increase in model complexity. This results in six nonzero groups, shown in Figure 7.15(b). The component numbers 41 and 74 have negligible weights. The VC, HC, and SC, do have significant weights and are considered the primary explanatory groups.

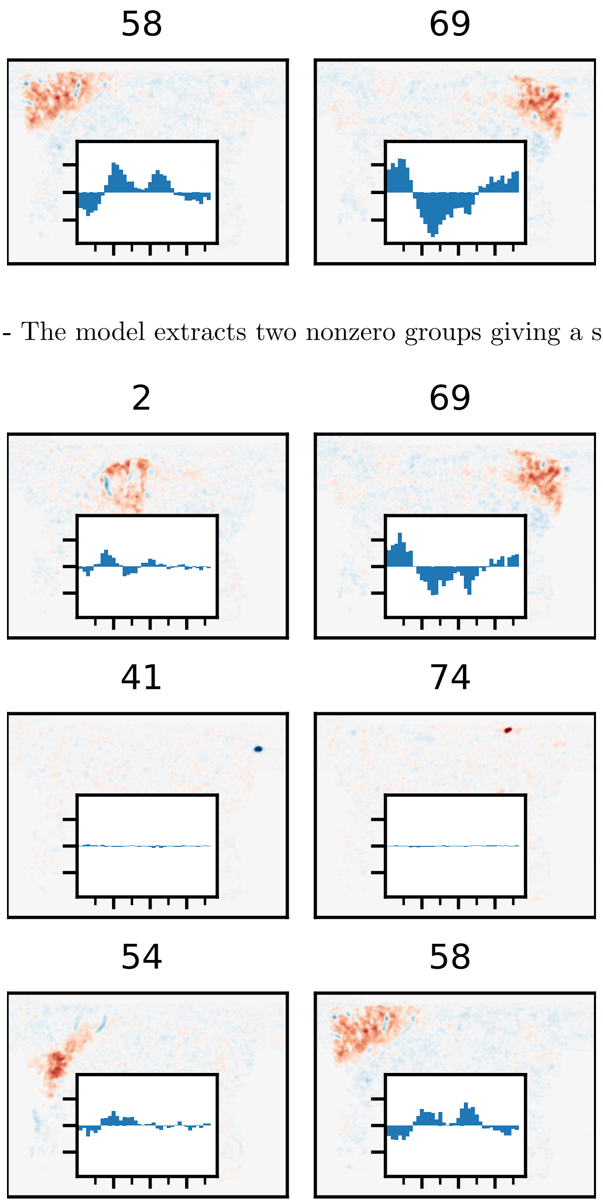
From bar charts in Figure 7.15(b) we observe the same time dynamics in the VC such as the double-peak structure and negative weights. The right HC and SC are not considered important regressors by this model. We speculate this is because the right VC gave enough discriminatory information.

Finally, we observe that the main response region starts between time samples 5 and 10. The main response of the HC and SC slightly precedes the VC's response. Using this knowledge we can make an informed statement about the validity of the Pearson correlation. As mentioned we opted for a delay of 7 samples followed by a box-car function of 12 samples long. By looking at the extracted weights, this region (from sample 7 to sample 19) closely matches the first peak in the VC but not in the HC and SC. In the HC the first peak is of much shorter duration and in the SC there is an undershoot which can effectively nullify the correlation. Therefore we have to be careful and understand the box-car correlation methodology has limitations.

Size experiment

Results of the sensitivity analysis for the size regression problem are provided in Figure 7.17. The R^2 is not stable over this regularization range. In the visible box plots no significant improvement in R^2 score was obtained compared with Table 7.5.

In Figure 7.18 we show the extracted nonzero groups for varying values of λ. Setting $\lambda = 8 \cdot 10^{-2}$, the model finds only the VC as in the group lasso position experiment. However, the shape of the computed weights is different, as shown in Figure 7.18(a). In this size experiment, the shape is characterized by an initial undershoot, followed by a



(a) $\lambda = 2 \cdot 10^{-2}$ - The model extracts two nonzero groups giving a score of $\kappa = 0.92$

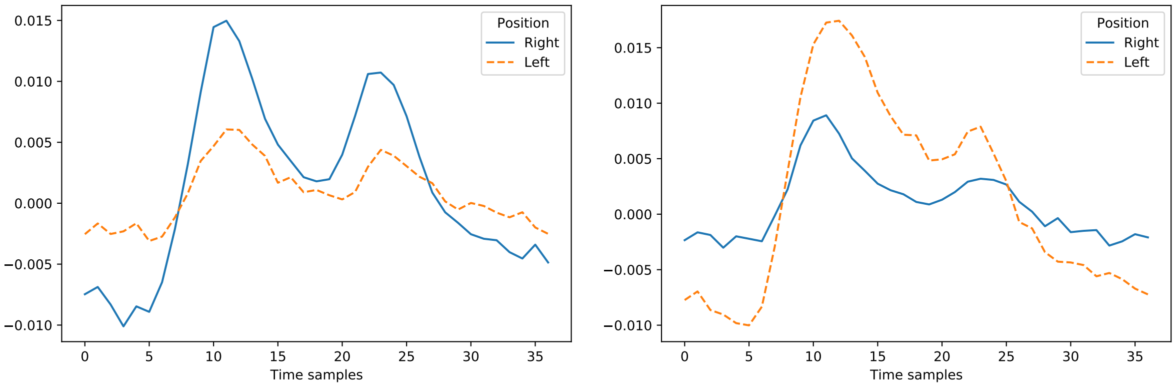
(b) $\lambda = 8 \cdot 10^{-3}$ - The model extracts six nonzero groups.

Figure 7.15: Group lasso for position experiment. Spatial maps and their group weight vector $\boldsymbol{w}_g$. The inset axis contains a barchart for all 37 values in $\boldsymbol{w}_g$. The ticks on the x-axis depicts steps of five samples.

VC: 58, 69 - HC: 54 - SC: 2

single peak and ending in slow decay to the baseline. This shape is unlike Figure 7.15(a), which had a double-peak structure in the middle section.

The reason for the difference in shapes is illustrated in Figure 7.19. The time courses are averaged for all sizes. We find that no matter the size of the stimuli, there is a second activation around sample 23. This means that there is no discriminatory information for



(a) Left VC

(b) Right VC

Figure 7.16: Group lasso for position experiment - Average time courses in left and right VC for left and right stimuli.

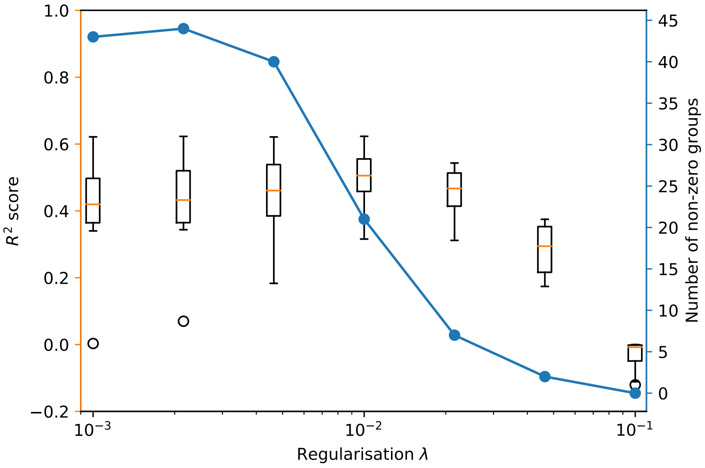
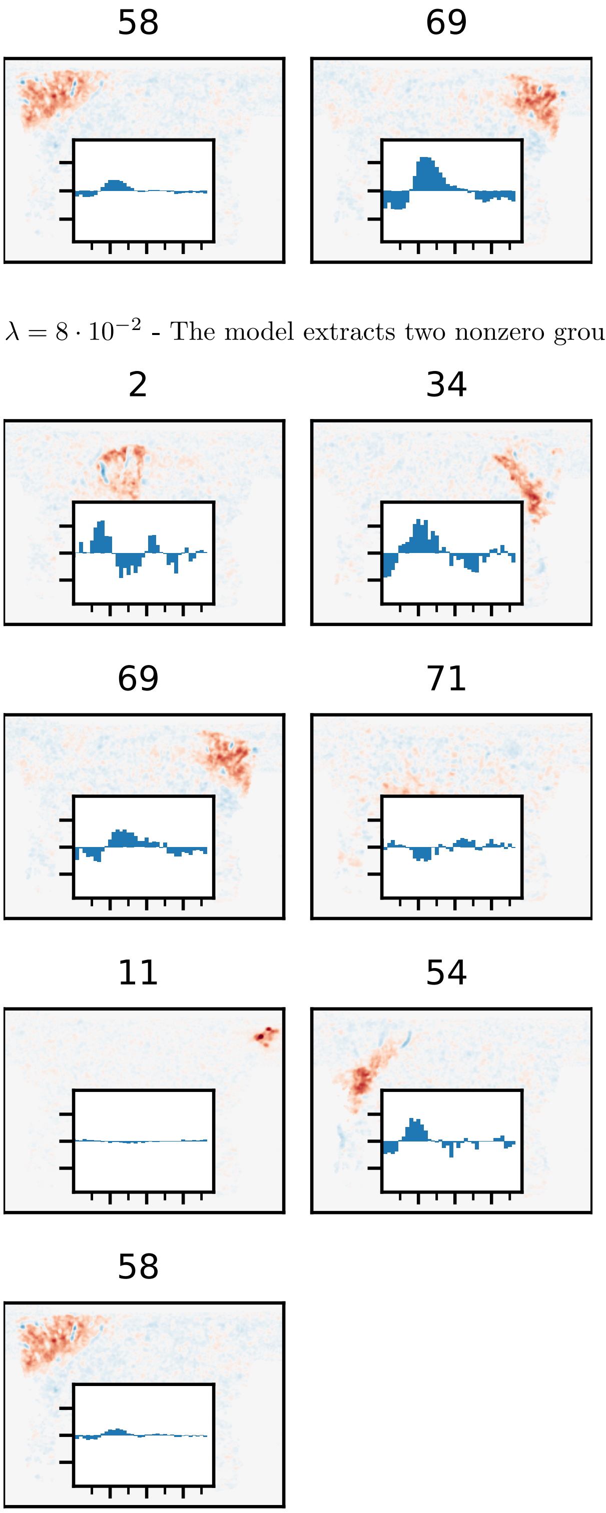


Figure 7.17: Group lasso for size experiment - Cross-validated coefficient of determination R^2 (box plots, left y-axis) vs regularization parameter λ (x-axis). The line plot shows the number of nonzero groups in $\boldsymbol{w}$ (right y-axis) for the same values of λ. These results are obtained with the ICA decomposition.

size task in the second peak. That explains why the second peak is absent in Figure 7.18. The largest variations in the curves can be found in the first 20 samples. The smallest size elicits no activation in the first peak. The largest two sizes have the strongest activation. The sizes 3 and 4 have intermediary responses.

We decreased the regularization parameter to $\lambda = 2 \cdot 10^{-2}$ and obtained seven nonzero groups shown in Figure 7.18(b). The weights of component 11 are near zero, and component 71 is a non-distinct region. Therefore the five remaining components are considered the primary explanatory groups.

Comparing the maps in Figures 7.15 and 7.18, the same components are extracted. The only exception is the right HC, which was not important in the position problem but is in the size experiment. Both problems did not consider the right SC to be important.

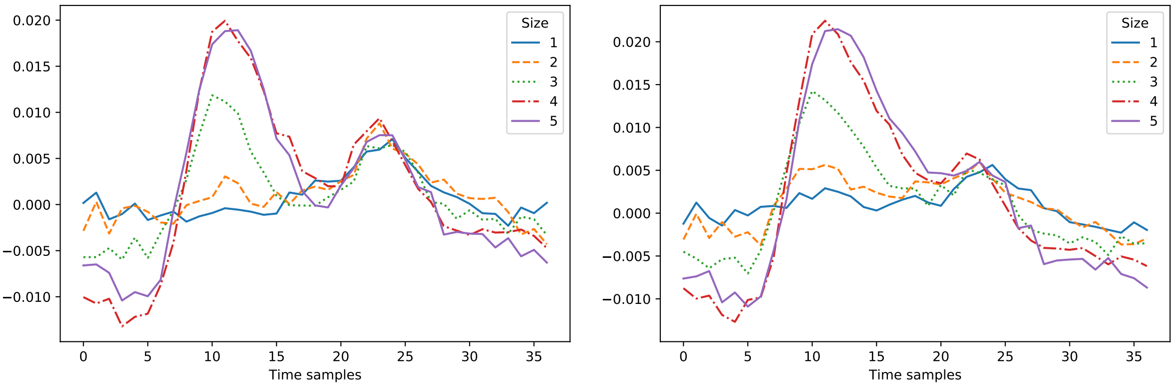


(a) $\lambda = 8 \cdot 10^{-2}$ - The model extracts two nonzero groups

(b) $\lambda = 2 \cdot 10^{-2}$ - The model extracts seven nonzero groups giving a biased score of $R^2 = 0.5$.

Figure 7.18: Group lasso for size experiment - Spatial maps and their group weight vector $\boldsymbol{w}_g$. The inset axis contains a barchart for all 37 values in $\boldsymbol{w}_g$. The ticks on the x-axis depict steps of five samples.

VC: 58, 69 - HC: 34, 54 - SC: 2



(a) Left VC

(b) Right VC

Figure 7.19: Group lasso for size experiment - Average time courses in left and right VC for five different sizes.

Throughout the position and size experiments, the left SC had an oscillating time course. After the initial positive response comes a significant undershoot, and after that, a smaller overshoot. The shape is consistent for both experiments, although the significance of the weights varies. The oscillating pattern illustrates the vulnerability of our Pearson correlation step. After oscillation with the box-car function we are left with a correlation likely around $r = 0$. So even though it does have discriminatory information it will not be considered by the prediction models. This also explains why this component was not found in the earlier experiments.

Shape experiment

In Figure 7.20 we show the sensitivity of the models performance on λ. We find the accuracy unstable. Only a marginal increase over a random prediction is obtained by increasing the number of nonzero groups.

Taking $\lambda = 1.14 \cdot 10^{-2}$, we find one nonzero group shown in Figure 7.21(a). This group belongs to the same SM found by the classifiers of sDL2 and sDL3. The accuracy of this classifier, $\kappa = 0.64$, is comparable to the accuracies found in Table 7.6. Therefore, we note that including more time samples does not increase this SM's discriminatory value.

Decreasing the regularization parameter to $\lambda = 1.135 \cdot 10^{-2}$ reveals two new components as shown in Figure 7.21(b). It is unclear what the groups resemble because it is difficult to link them to an anatomical region.

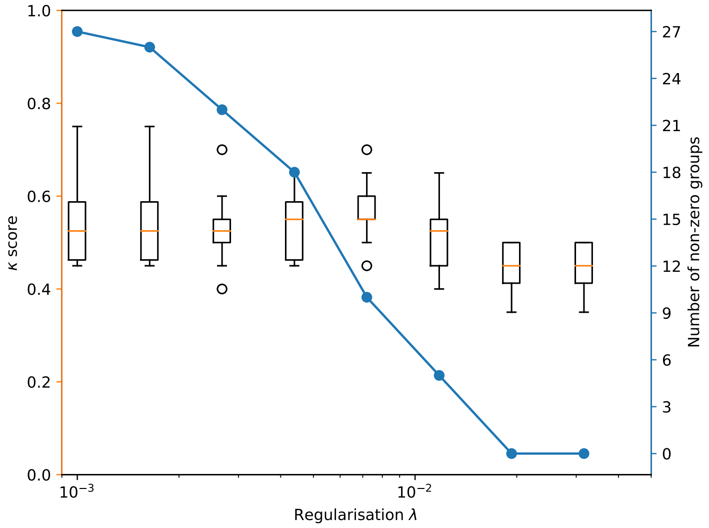
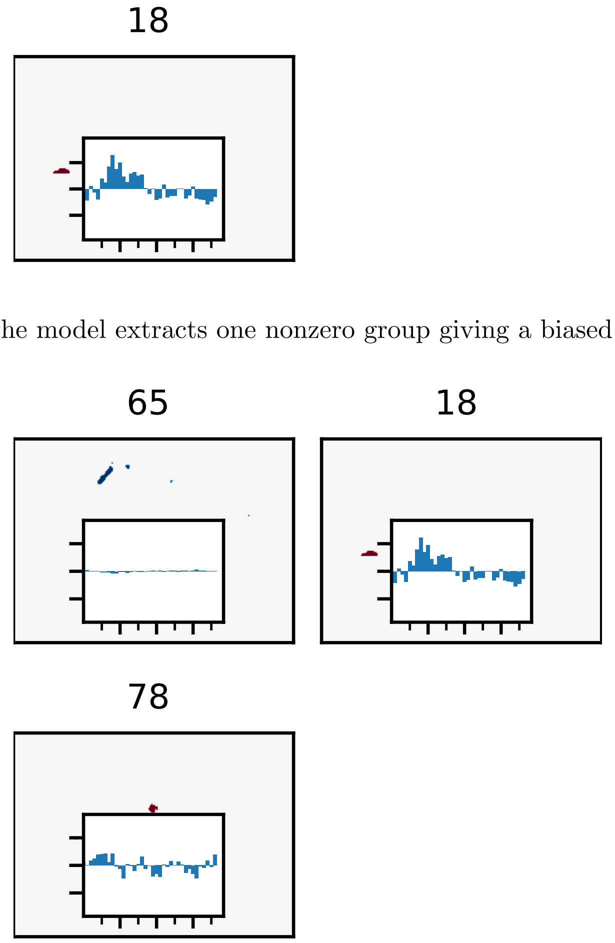


Figure 7.20: Group lasso for shape experiment - Cross-validated predicted accuracy κ (box plots, left y-axis) vs regularization parameter λ (x-axis). The line plot shows the number of nonzero groups in $\boldsymbol{w}$ (right y-axis) for the same values of λ. These results are obtained with the sDL3 decomposition.



(a) $\lambda = 1.14 \cdot 10^{-2}$ - The model extracts one nonzero group giving a biased performance $\kappa = 0.64$

(b) $\lambda = 1.135 \cdot 10^{-2}$ - The model extracts three nonzero groups.

Figure 7.21: Group lasso for shape experiment. Spatial maps and their group weight vector $\boldsymbol{w}_g$. The inset axis contains a barchart for all 37 values in $\boldsymbol{w}_g$. The ticks on the x-axis depict steps of five samples.

Conclusion 8

The main objective of this research was to search for changes in the fUS data of a mouse associated with diverse visual stimuli. The stimuli were grouped into three categories: position, size, and shape. For each category, we wanted to understand the hemodynamic change associated with that category. In particular, we set to find how response differs depending on the position, size, and shape of the stimulus. Furthermore, we examined what decomposition could be used to describe these changes, independent component analysis (ICA) or sparse dictionary learning (sDL).

Four decomposition variations were examined; one with ICA and three variations of sDL. We first decomposed the input data into spatial maps (SMs) and time courses (TCs). That gave four sets of spatial maps with different levels of sparsity in the SMs. All decomposition methods, except sDL2, found maps belonging to the visual cortex (VC), hippocampus (HC), and superior colliculus (SC) and in both hemispheres.

All the decompositions are data-driven; we only constrain the algorithm to find sparse or independent SMs. The advantage of the data-driven nature is that the model has a broad range of outcomes because the model is not steered in a desired direction. The disadvantage is that the components can be non-interpretable or instable over the number of components K. We found that many extracted components required expert knowledge for interpretation or were non-interpretable at all. This complicated the neurological interpretation. Furthermore, determining the optimal setting of K is an open question. We have shown that the predictive performance is approximately stable in range $K > 40$ in the ICA method, but verifying the same for the sDL methods is computationally expensive.

Models for position, size, and shape prediction were trained given the TCs correlated with a delayed stimulus box-car function. Although this a standard procedure for finding the resemblance of a TC with a known stimulus pattern, it is vulnerable to assumptions. Using group lasso, we found an analytical procedure validating the stimulus pattern. By looking at the start and end point of the “main response” we could determine the duration of the stimulus pattern. Further, we could determine the shape of the stimulus pattern.

We found that there is no single shape nor duration of the stimulus pattern that fits all components. For example, the onset of the discriminatory information in the SC and HC is almost simultaneously while the onset of the VC is later. Besides that, the shape of the left SC had a deviating time pattern compared to the other components. The left SC was oscillating in the active region while the other components were not. This explained why it was not considered in any of the created models. We did find VC was well described using our box-car methodology.

We trained an ℓ_1-regularized logistic regression for the position and the shape classification task. A lasso linear regression was trained for the size regression task. Both models select for sparsity, which should lead to interpretable models. The models are interpretable in the sense that they are simple and with few nonzero coefficients. However, the simplicity can be detrimental because being selective might be undesirable in the presence of correlated features.

Nevertheless, we found that the models were not overly selective. In fact, all three experiments resulted in a large set of predictive components which could not be interpreted. The goal of the prediction functions was the opposite. Therefore the sparsity inducing regularization in combination with cross-validation was ineffective. This can be explained by the large sensitivity of the model complexity on λ while the biased performance estimation is insensitive after reaching a plateau. This means that model complexity grows without suffering from overfitting.

By comparing the model's estimates of the generalization performance we could determine what decomposition methods gave the most informative features for the prediction tasks. The reasoning is that if the decomposition gives discriminatory features then it accurately describes the data.

We found significant differences in the estimates of the generalization performance. ICA outperformed the other three decompositions in all prediction tasks. Especially in the size prediction there were large variations in performance between ICA and the other methods. In the position and shape task, the difference in performance between ICA and sDL1 was slight but existent. We therefore conclude that ICA gives the most informative features and thereby describes the data the best of all investigated methods. However, this result is only verified with our predictor models. It is unclear if different predictors would rate other decompositions higher.

The goal of the predictive models was twofold: not only did we want to measure the generalization performance, we also wanted to obtain neurological insights. By basing our neurological insights upon the predictive models we were essentially limited to discriminatory insights. For example, we investigated how the response from a circle stimulus differs from a square stimulus and not how the circle stimulus causes a hemodynamic response in general.

We investigated the differences in response by looking at the model's weights, a proxy for feature importance. However, feature importance is a complex subject that we have to be careful about. The feature importance rankings did not align with the most important features obtained with group lasso. This is most likely caused by additional temporal information included in the group lasso regularized model. This illustrates that the feature rankings have to be placed in relation to the underlying assumptions in the Pearson correlation step.

In the position experiment, the most valuable discriminatory information is in the HC. We illustrated that also the VC and SC provided discriminatory information but were considered less important by our feature importance score. Group lasso regularization also provided an interesting look in the temporal dynamics. Using that method, we found the most valuable features in the VC. There was discriminatory information in strength of the two peaks but less in between.

In the size experiment, all four models found a large group of SMs, making it hard to interpret the results. An explanation for this unclear result is the use of a continuous variable where an ordinal variable is correct. The model might have tried to predict the artificial continuous spectrum by including many unnecessary components.

Group lasso regularization provided a more coherent picture because we manually constrained it to simple models. It also found the left and right VC important in the size prediction task. We showed that the discriminatory information in the VC is primarily in the first peak and the second peak is present for all sizes and thus gives no information for this task. Although group lasso did give a more coherent picture, there was only a minor increase in generalization performance. This means that the biased performances were similar to regular ℓ_1-regularized models.

We found the most considerable model differences in the shape experiment. sDL2 and sDL3 found a single SM possibly related to the shape stimulus. Also group lasso verified its validity. This SM is hypothesized to be a vessel that supplies to or receives blood from the HC. It provided a minor but considerable improvement in classification accuracy. The underlying physiological explanation is unclear.

To sum up, we obtained a clearer picture on the vascular response, related to how different stimuli are processed differently. As expected, we found a clear lateralization of the brain function. We also saw that in general larger sizes give stronger responses. This is only the case for the first peak and not in the second. We found little discriminatory information in the shape prediction. In general, the circles and squares are processed similarly except for one component that appears to be slightly different for circles and squares.

Future directions

A problem with the acquisition methodology is the long duration (42 min). The mouse's sensitivity to the stimulus can decrease over time as the mouse becomes bored. This is a problem because the squares were presented in the first block of the recording and the circles in the second block. Mixing the shapes throughout the recording would already help resolve this issue. Furthermore, longer rest periods could be useful. As shown in the results of group lasso, we found that the VC had not settled to a steady baseline before the next stimulus was shown. It is unclear if this is a problem, but it is worthwhile to investigate if longer rest periods give clearer results.

We limited our decompositions to ICA and sDL algorithms. There are many other data-driven decompositions to consider. For example, one can consider a total-variation constrained sDL that uses a smoothness constraint in addition to the sparsity constraint [33]. This would especially be useful if it would create more robust components. It could potentially allow for a smaller K because the brain networks are constrained to be localized.

Another idea is to use dynamic mode decomposition models [34]. They model the measurements at time t as a linear mapping of the measurements at time $t-1$. When the functional ultrasound (fUS) technique has the ability to measure three-dimensional structures, it could potentially reveal dynamical modes in the blood flow.

Group lasso regularization is an exciting direction for continuing the research, as it gives not only a list of SMs but also gives a full temporal scale. However, due to massive computational overload, we could not test this regularization for all decomposition methods. One can even extend to sparse group lasso [35] that can provide sparsity in the temporal dimension. It is unclear if this desired. A probably more desirable direction is fused group lasso [36]. Fused group lasso does not enforce sparsity in the temporal dimension but smoothness with a total variation regularizer. Smoothness is a desirable property as the hemodynamic response function (HRF) is also smooth. However, the results we obtained with group lasso are already quite smooth, so it might not be beneficial.

Another idea is to use multi-task learning methods. In this architecture multiple tasks are solved at once [37]. In our case, that would mean we train the model for the position, size, and shape tasks simultaneously. Multi-task learning algorithms can exploit the commonalities shared among the tasks. Requiring the model to perform well on its related task can act as a superior regularization method. This is in contrast with regularization by penalizing model complexity, as we did in this thesis.

Appendices

A Spatial Maps

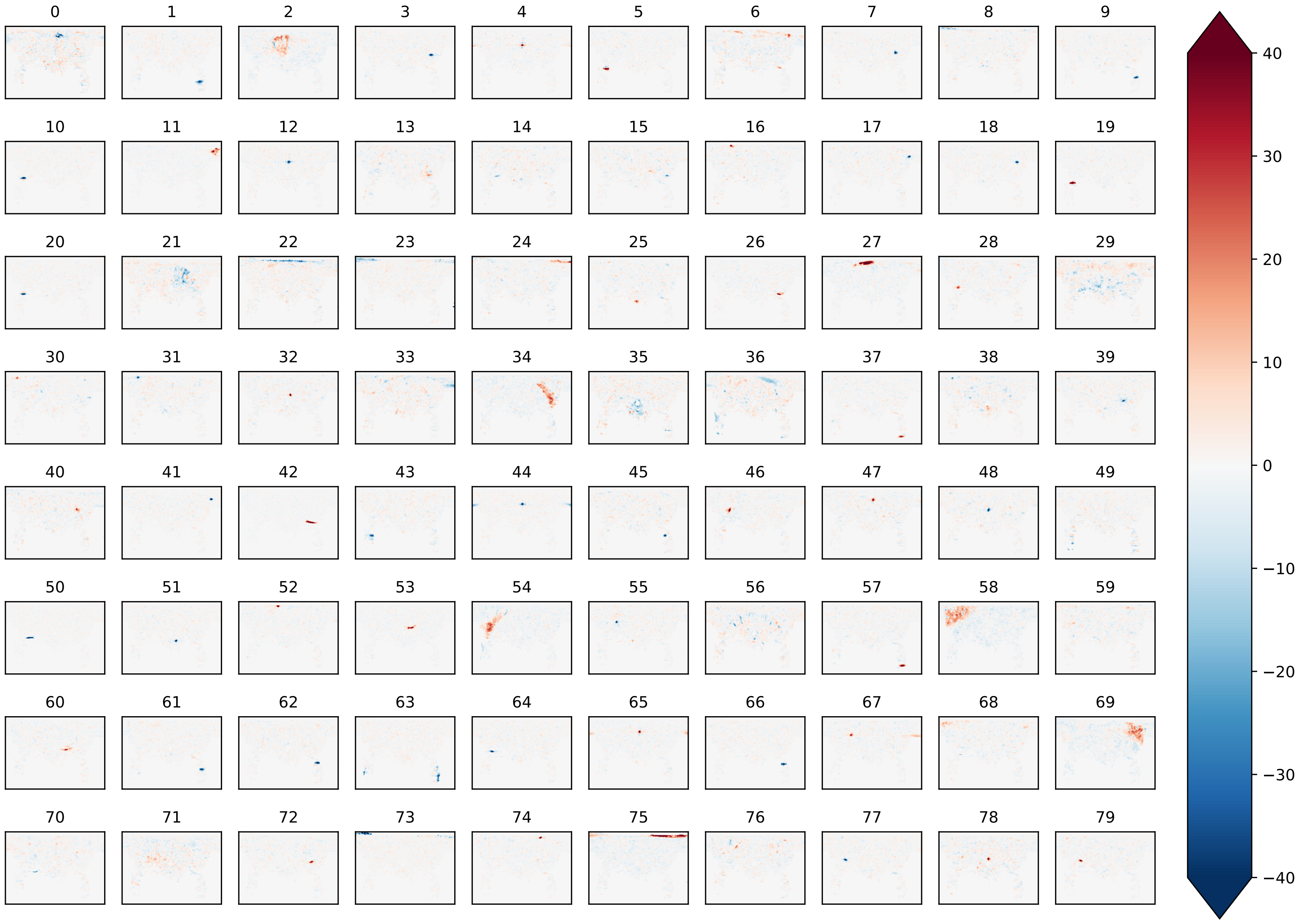


Figure A.1: ICA

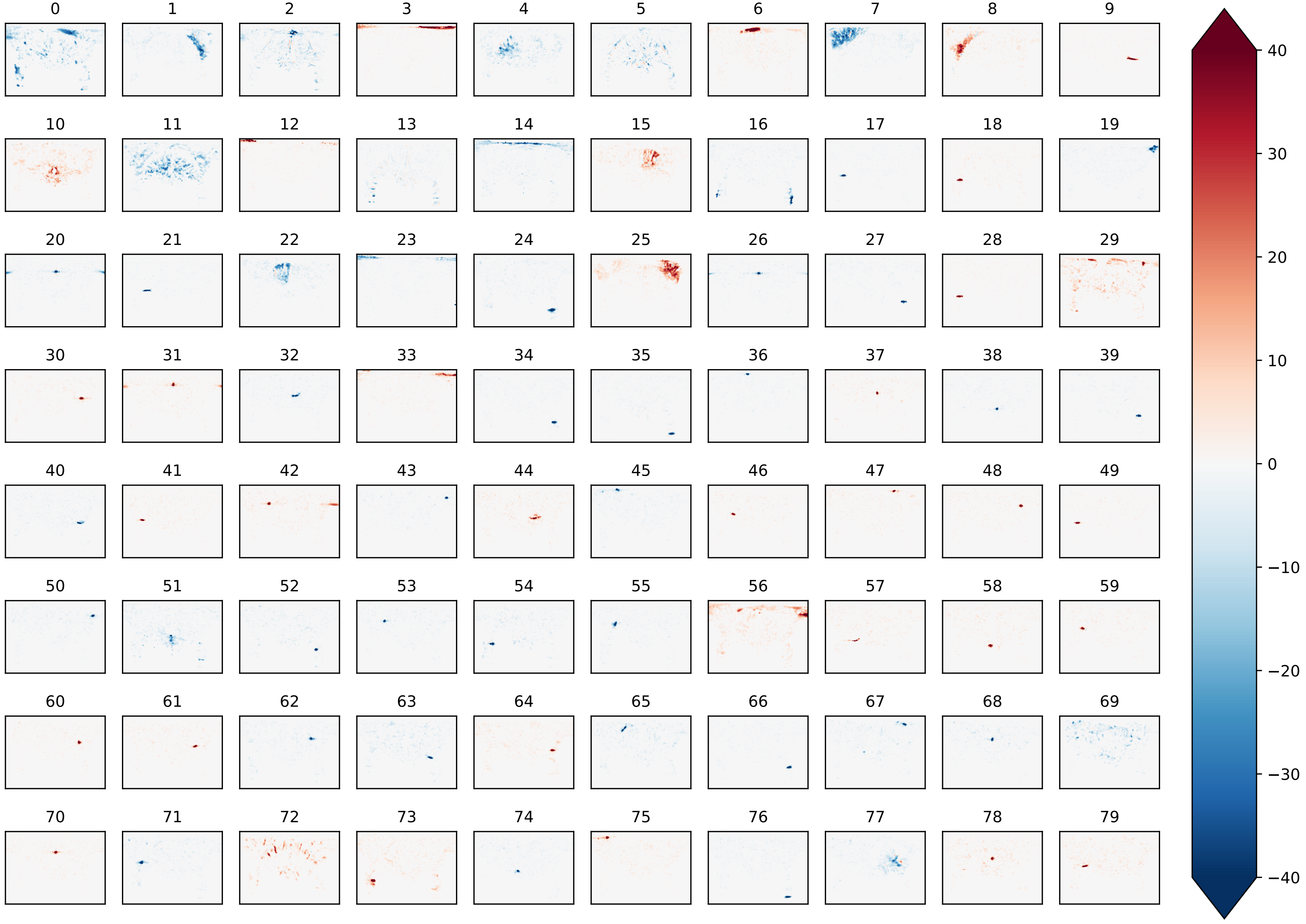


Figure A.2: sDL1

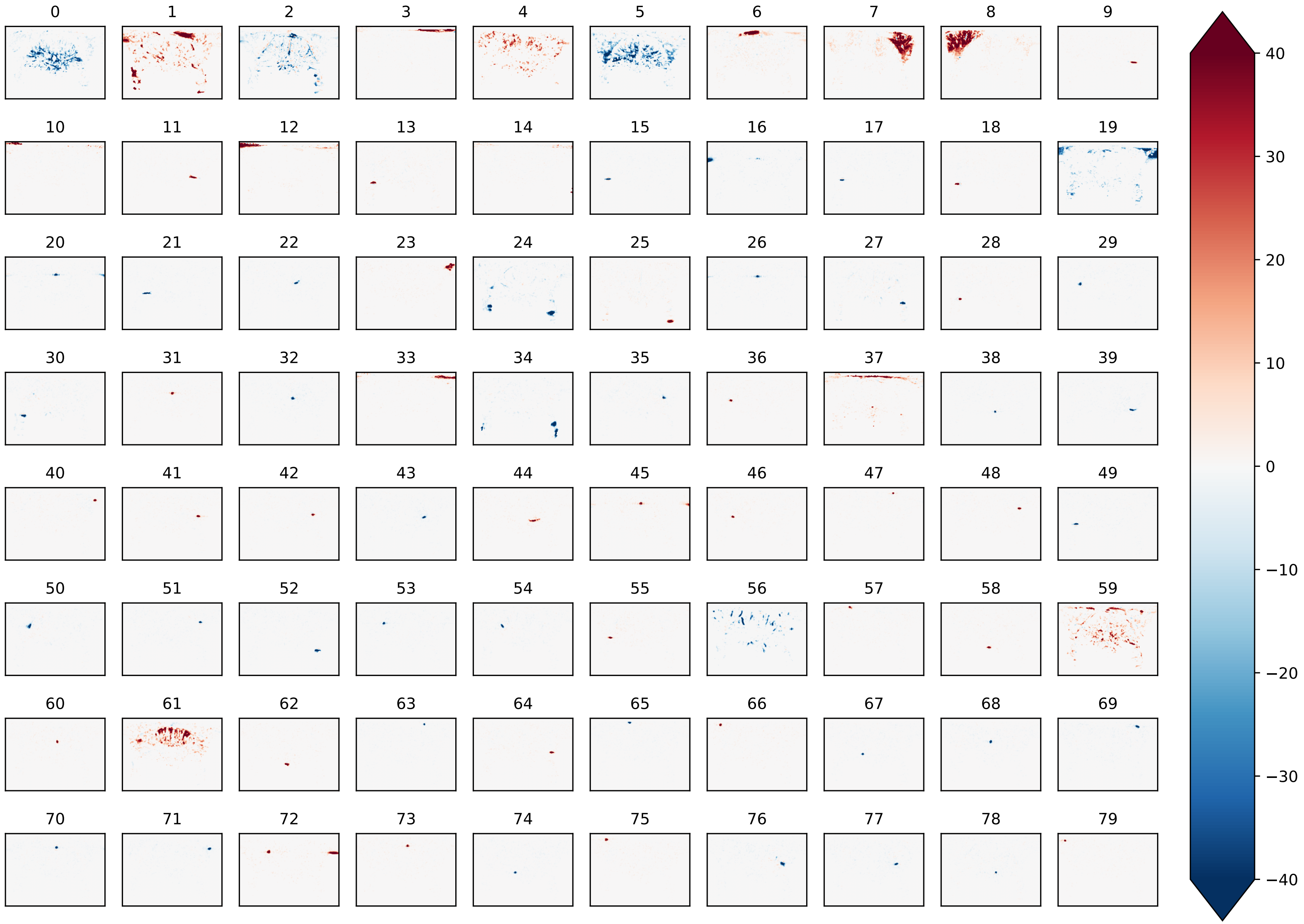


Figure A.3: sDL2

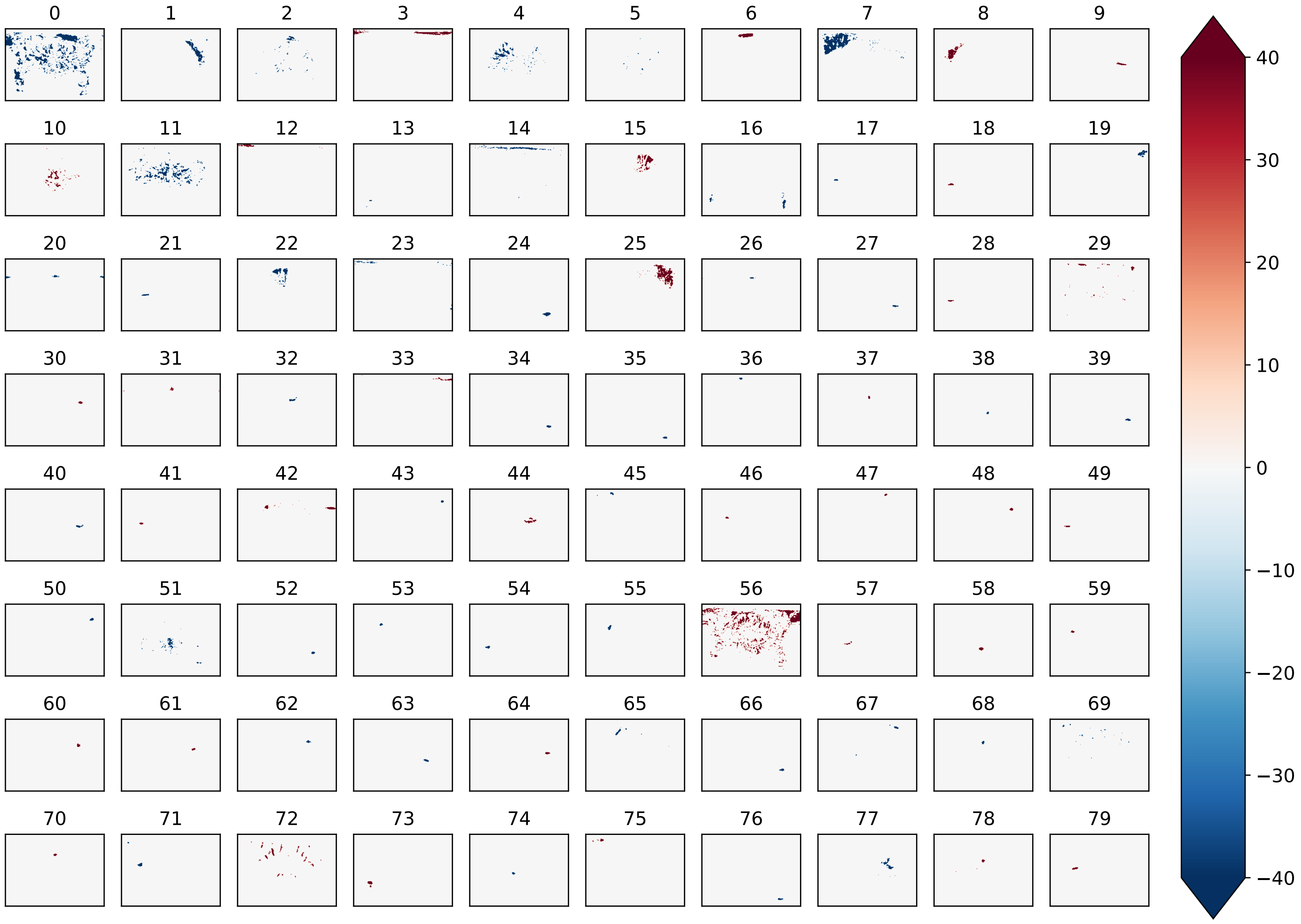


Figure A.4: sDL3

B Additional results size experiment

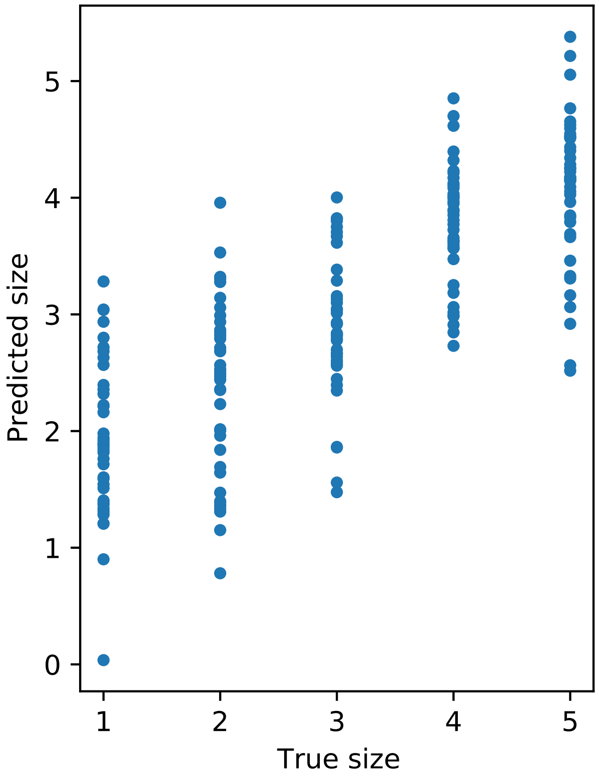
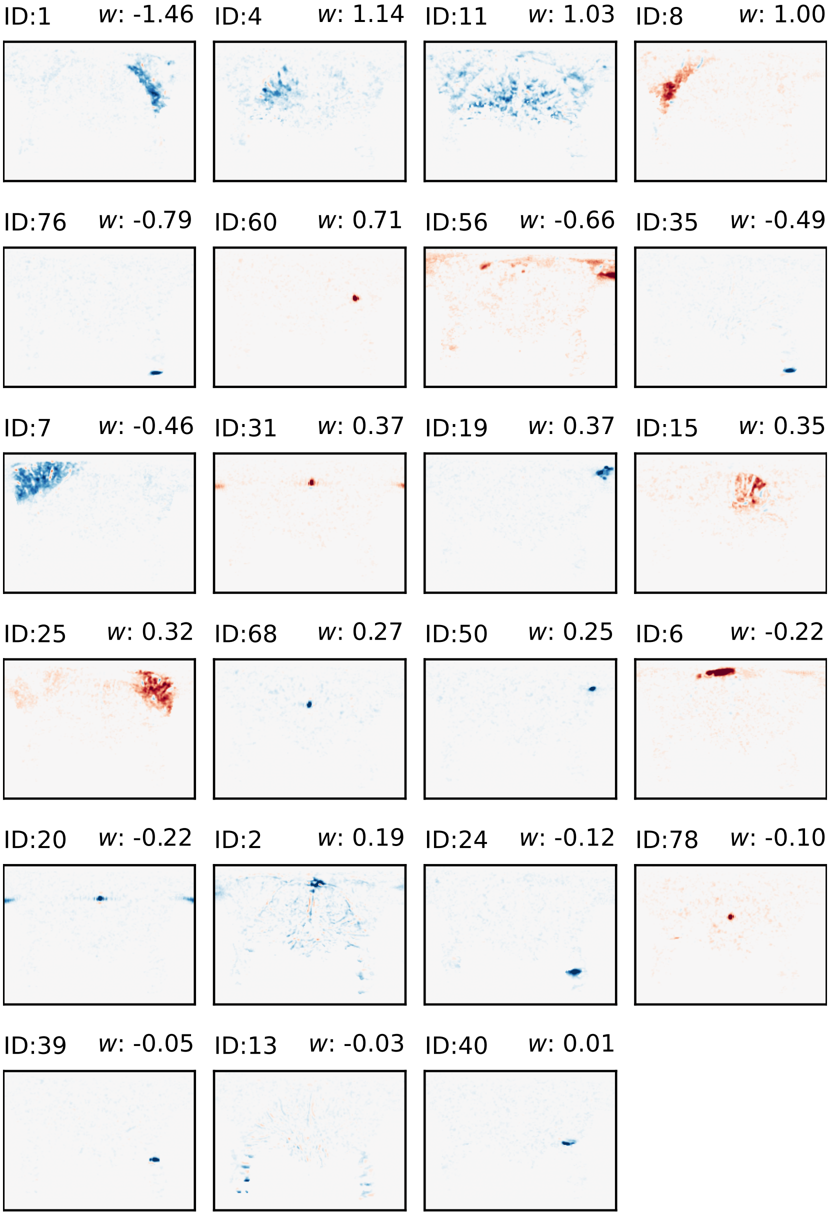
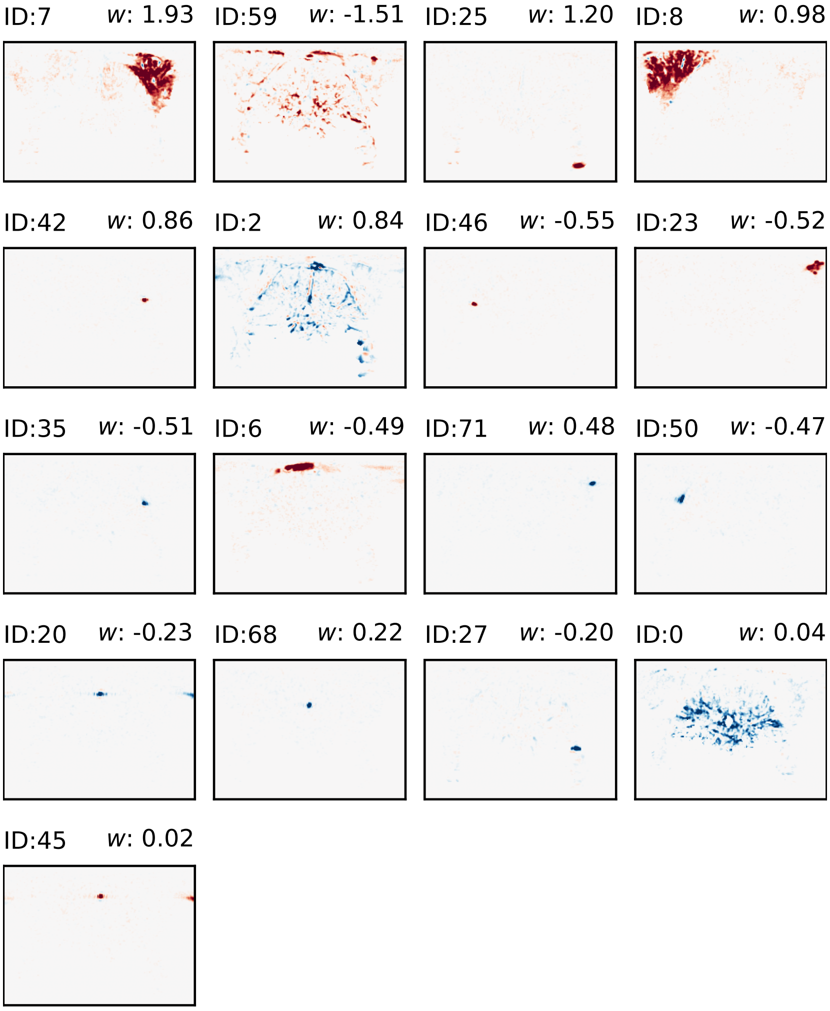


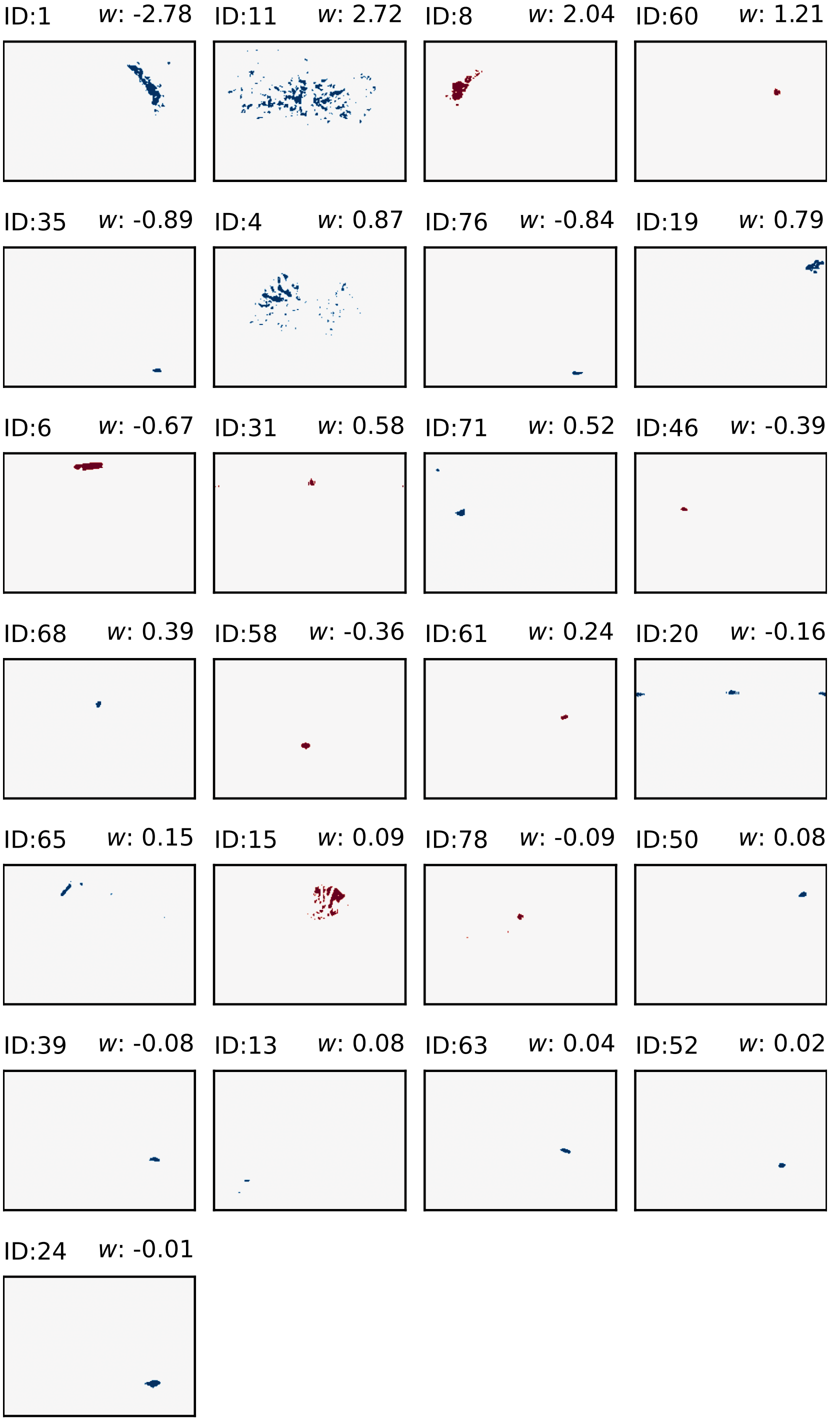
Figure B.1: Size experiment - true size labels vs predicted labels. Every point represents one stimulus. The predictions are made with the ICA decomposition.


ID:1 *w*: -1.46
ID:4 *w*: 1.14
ID:11 *w*: 1.03
ID:8 *w*: 1.00
ID:76 *w*: -0.79
ID:60 *w*: 0.71
ID:56 *w*: -0.66
ID:35 *w*: -0.49
ID:7 *w*: -0.46
ID:31 *w*: 0.37
ID:19 *w*: 0.37
ID:15 *w*: 0.35
ID:25 *w*: 0.32
ID:68 *w*: 0.27
ID:50 *w*: 0.25
ID:6 *w*: -0.22
ID:20 *w*: -0.22
ID:2 *w*: 0.19
ID:24 *w*: -0.12
ID:78 *w*: -0.10
ID:39 *w*: -0.05
ID:13 *w*: -0.03
ID:40 *w*: 0.01

(a) sDL1


ID:7 w: 1.93
ID:59 w: -1.51
ID:25 w: 1.20
ID:8 w: 0.98
ID:42 w: 0.86
ID:2 w: 0.84
ID:46 w: -0.55
ID:23 w: -0.52
ID:35 w: -0.51
ID:6 w: -0.49
ID:71 w: 0.48
ID:50 w: -0.47
ID:20 w: -0.23
ID:68 w: 0.22
ID:27 w: -0.20
ID:0 w: 0.04
ID:45 w: 0.02

(b) sDL2



(c) sDL3

Figure B.2: Remaining SM extracted in the size experiment.

C Additional results shape experiment

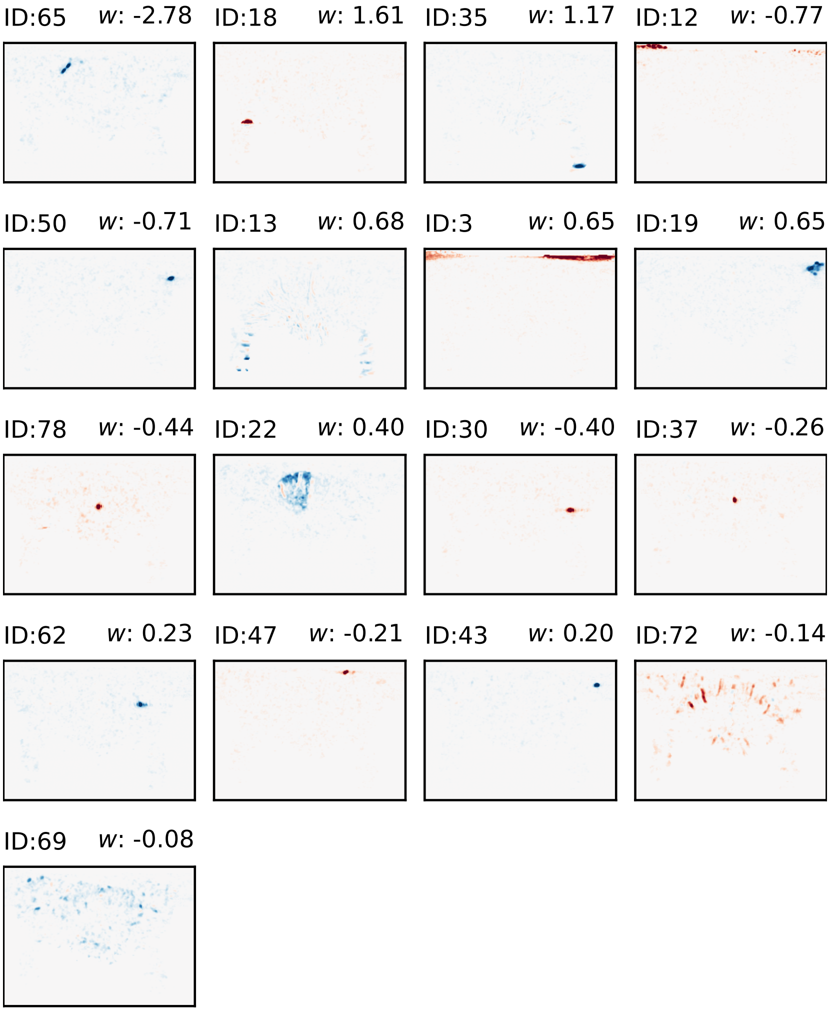


Figure C.1: sDL1- The SM extracted in the shape experiment.

Bibliography

[1] E. MacÉ, G. Montaldo, I. Cohen, M. Baulac, M. Fink, and M. Tanter, "Functional ultrasound imaging of the brain," Nature Methods, vol. 8, no. 8, pp. 662–664, Aug. 2011, ISSN: 15487091. DOI: 10.1038/nmeth.1641. [Online]. Available: https://www.nature.com/articles/nmeth.1641.

[2] L. A. Sieu, A. Bergel, E. Tiran, et al., "EEG and functional ultrasound imaging in mobile rats," Nature Methods, vol. 12, no. 9, pp. 831–834, Sep. 2015, ISSN: 15487105. DOI: 10.1038/nmeth.3506. [Online]. Available: https://www.nature.com/articles/nmeth.3506.

[3] J. Ferrier, E. Tiran, T. Deffieux, M. Tanter, and Z. Lenkei, "Functional imaging evidence for task-induced deactivation and disconnection of a major default mode network hub in the mouse brain," Proceedings of the National Academy of Sciences of the United States of America, vol. 117, no. 26, pp. 15 270–15 280, Jun. 2020, ISSN: 10916490. DOI: 10.1073/pnas.1920475117. [Online]. Available: https://www.pnas.org/content/117/26/15270.

[4] A. J. Kennerley, J. E. Mayhew, P. Redgrave, and J. Berwick, "Vascular Origins of BOLD and CBV fMRI Signals: Statistical Mapping and Histological Sections Compared," The Open Neuroimaging Journal, vol. 4, pp. 1–8, Mar. 2010, ISSN: 1874-4400. DOI: 10.2174/1874440001004010001. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2887650/.

[5] Q. Shen, H. Ren, and T. Q. Duong, "CBF, BOLD, CBV, and CMRO2 fMRI signal temporal dynamics at 500-msec resolution," Journal of Magnetic Resonance Imaging, vol. 27, no. 3, pp. 599–606, Mar. 2008, ISSN: 10531807. DOI: 10.1002/jmri.21203. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2900800/.

[6] M. Magnuson, W. Majeed, and S. D. Keilholz, "Functional connectivity in blood oxygenation level-dependent and cerebral blood volume-weighted resting state functional magnetic resonance imaging in the rat brain," Journal of Magnetic Resonance Imaging, vol. 32, no. 3, pp. 584–592, Sep. 2010, ISSN: 10531807. DOI: 10.1002/jmri.22295. [Online]. Available: /pmc/articles/PMC2936716/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2936716/.

[7] B. Afshin-Pour, C. Grady, and S. Strother, "Evaluation of spatio-temporal decomposition techniques for group analysis of fMRI resting state data sets," NeuroImage, vol. 87, pp. 363–382, Feb. 2014, ISSN: 10959572. DOI: 10.1016/j.neuroimage.2013.10.062.

[8] M. U. Khalid and A. K. Seghouane, "A single SVD sparse dictionary learning algorithm for FMRI data analysis," in IEEE Workshop on Statistical Signal Processing Proceedings, IEEE Computer Society, 2014, pp. 65–68, ISBN: 9781479949755. DOI: 10.1109/SSP.2014.6884576.

[9] I. Daubechies, E. Roussos, S. Takerkart, et al., "Independent component analysis for brain fMRI does not select for independence," Proceedings of the National Academy of Sciences of the United States of America, vol. 106, no. 26, pp. 10 415–10 422, Jun. 2009, ISSN: 00278424. DOI: 10.1073/pnas.0903525106. [Online]. Available: www.pnas.org.

[10] I. J. Myung and M. A. Pitt, "Applying Occam's razor in modeling cognition: A Bayesian approach," Psychonomic Bulletin and Review, vol. 4, no. 1, pp. 79–95, Mar. 1997, ISSN: 10699384. DOI: 10.3758/BF03210778. [Online]. Available: https://doi.org/10.3758/BF03210778.

[11] B. W. Brunton and M. Beyeler, "Data-driven models in human neuroscience and neuroengineering," Current Opinion in Neurobiology, Computational {Neuroscience}, vol. 58, pp. 21–29, Oct. 2019, ISSN: 18736882. DOI: 10.1016/j.conb.2019.06.008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0959438818302502.

[12] N. R. Draper and H. Smith, "Ill-Conditioning in Regression Data," in John Wiley & Sons, Ltd, Apr. 1998, pp. 369–386. DOI: 10.1002/9781118625590.ch16. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1002/9781118625590.ch16%20https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118625590.ch16%20https://onlinelibrary.wiley.com/doi/10.1002/9781118625590.ch16.

[13] N. Problems, A. E. Hoerl, and R. W. Kennard, "American Society for Quality Ridge Regression: Biased Estimation for," Tech. Rep. 1, 1970, pp. 55–67.

[14] R. Tibshirani and R. Tibshirani, "Regression Shrinkage and Selection Via the Lasso," JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B, vol. 58, pp. 267–288, 1994. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.7574.

[15] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," Tech. Rep., 2009, pp. 689–696.

[16] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," Neural Networks, vol. 13, no. 4-5, pp. 411–430, Jun. 2000, ISSN: 08936080. DOI: 10.1016/S0893-6080(00)00026-5. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0893608000000265.

[17] C. H. Teo, Q. Le, A. Smola, and S. V. N. Vishwanathan, A Scalable Modular Convex Solver for Regularized Risk Minimization. 2007, ISBN: 9781595936097.

[18] T. Hofmann, B. Schölkopf, and A. J. Smola, "KERNEL METHODS IN MACHINE LEARNING 1," The Annals of Statistics, vol. 36, no. 3, pp. 1171–1220, 2008. DOI: 10.1214/009053607000000677.

[19] P. A. Gutiérrez, M. Pérez-Ortiz, J. Sánchez-Monedero, F. Fernández-Navarro, and C. Hervás-Martínez, "Ordinal Regression Methods: Survey and Experimental Study," in IEEE Transactions on Knowledge and Data Engineering, vol. 28, IEEE Computer Society, Jan. 2016, pp. 127–146. DOI: 10.1109/TKDE.2015.2457911.

[20] V. Michel, A. Gramfort, G. Varoquaux, E. Eger, and B. Thirion, "Total variation regularization for fMRI-based prediction of behavior," IEEE Transactions on Medical Imaging, vol. 30, no. 7, pp. 1328–1340, Jul. 2011, ISSN: 02780062. DOI: 10.1109/TMI.2011.2113378.

[21] S.-I. Lee, H. Lee, P. Abbeel, and A. Y. Ng, "Efficient L 1 Regularized Logistic Regression," Tech. Rep. [Online]. Available: www.aaai.org.

[22] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 68, no. 1, pp. 49–67, 2006. DOI: https://doi.org/10.1111/j.1467-9868.2005.00532.x. eprint: https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.

1467-9868.2005.00532.x. [Online]. Available: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2005.00532.x.

[23] L. Meier, S. Van De Geer, and P. Bühlmann, "The group lasso for logistic regression," Journal of the Royal Statistical Society. Series B: Statistical Methodology, vol. 70, no. 1, pp. 53–71, Feb. 2008, ISSN: 13697412. DOI: 10.1111/j.1467-9868.2007.00627.x. [Online]. Available: https://rss.onlinelibrary.wiley.com/doi/full/10.1111/j.1467-9868.2007.00627.x%20https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2007.00627.x%20https://rss.onlinelibrary.wiley.com/doi/10.1111/j.1467-9868.2007.00627.x.

[24] S. Raschka, "Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning," Tech. Rep., 2018. arXiv: 1811.12808v3.

[25] J. Wainer and G. Cawley, "Nested cross-validation when selecting classifiers is overzealous for most practical applications," Tech. Rep. arXiv: 1809.09446v1.

[26] M. Saarela and S. Jauhiainen, "Comparison of feature importance measures as explanations for classification models," 123. DOI: 10.1007/s42452-021-04148-9. [Online]. Available: https://doi.org/10.1007/s42452-021-04148-9.

[27] G. Hooker and L. Mentch, "Please Stop Permuting Features An Explanation and Alternatives," Tech. Rep. arXiv: 1905.03151v1.

[28] L. Toloşi and T. Lengauer, "Classification with correlated features: Unreliability of feature ranking and solutions," Bioinformatics, vol. 27, no. 14, pp. 1986–1994, Jul. 2011, ISSN: 13674803. DOI: 10.1093/bioinformatics/btr300. [Online]. Available: http://www.mpi-inf.mpg.de/.

[29] A. Zien, N. Krämer, S. Sonnenburg, and G. Rätsch, "LNAI 5782 - The Feature Importance Ranking Measure," Tech. Rep.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, et al., "Scikit-learn: Machine learning in python," Journal of Machine Learning Research, vol. 12, no. 85, pp. 2825–2830, 2011. [Online]. Available: http://jmlr.org/papers/v12/pedregosa11a.html.

[31] S. M. Sunkin, L. Ng, C. Lau, et al., "Allen Brain Atlas: an integrated spatio-temporal portal for exploring the central nervous system," Nucleic Acids Research, vol. 41, no. D1, pp. D996–D1008, Nov. 2012, ISSN: 0305-1048. DOI: 10.1093/nar/gks1042. eprint: https://academic.oup.com/nar/article-pdf/41/D1/D996/3594088/gks1042.pdf. [Online]. Available: https://doi.org/10.1093/nar/gks1042.

[32] J. D. Hunter, "Matplotlib: A 2d graphics environment," Computing in Science & Engineering, vol. 9, no. 3, pp. 90–95, 2007. DOI: 10.1109/MCSE.2007.55.

[33] A. Abraham, E. Dohmatob, B. Thirion, D. Samaras, G. Varoquaux, and G. Varoquaux, "Extracting brain regions from rest fMRI with Total-Variation constrained dictionary learning," Tech. Rep., 2013. [Online]. Available: https://hal.inria.fr/hal-00853242.

[34] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," Journal of Fluid Mechanics, vol. 656, pp. 5–28, 2010, ISSN: 14697645. DOI: 10.1017/S0022112010001217. [Online]. Available: https://doi.org/10.1017/S0022112010001217.

[35] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "A sparse-group lasso," Journal of Computational and Graphical Statistics, vol. 22, no. 2, pp. 231–245, 2013. DOI: 10.1080/10618600.2012.681250. eprint: https://doi.org/10.1080/10618600.

2012.681250. [Online]. Available: https://doi.org/10.1080/10618600.2012.681250.

[36] C. M. Alaíz, Á. Barbero, and J. R. Dorronsoro, “Group fused lasso,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Le vol. 8131 LNCS, Springer, Berlin, Heidelberg, 2013, pp. 66–73, ISBN: 9783642407277. DOI: 10.1007/978-3-642-40728-4_9. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-40728-4%7B%5C_%7D9.

[37] N. S. Raoy, C. R. Cox, R. D. Nowaky, and T. T. Rogers, “Sparse Overlapping Sets lasso for multitask learning and its application to fMRI analysis,” Tech. Rep., 2013. arXiv: 1311.5422.