Ca' Foscari University of Venice
Department of Environmental Sciences, Informatics and Statistics

Master Degree programme in Computer Science
Second Cycle (D.M. 270/2004)

Final thesis

Revealing Hierarchical Object Processing in Rat's Visual System Using Cluster Analysis

Graduand:
Ylenia Parin
Matriculation Number 848661

Supervisor:
Ch. Prof. Marcello Pelillo

Assistant supervisors:
Ch. Prof. Davide Zoccolan
Sebastiano Vascon, PhD
Eis Annavini, PhD

Academic Year 2017-2018

Contents

Abstract iii

1 Introduction 1

2 Neurobiological context 5

2.1 The Core Object Recognition Problem . . . 5

2.2 The primate visual system . . . 7

2.3 The rat visual system . . . 7

3 Experimental design and single cell analysis 9

3.1 The experimental design . . . 9

3.1.1 Stimulus set . . . 9

3.1.2 Response set . . . 10

3.1.3 Feature description . . . 11

3.2 Single cell analysis . . . 11

3.2.1 Information theory measures . . . 11

3.2.2 Feature and response bins . . . 14

4 Population analysis and clustering algorithms 17

4.1 Population vector space . . . 17

4.2 Distance measures and RDMs . . . 18

4.3 Dimensionality reduction with t-SNE . . . 19

4.4 Cluster analysis . . . 19

4.4.1 Feature selection . . . 21

4.4.2 Clustering algorithm . . . 21

4.4.3 Cluster validity assessment . . . 25

4.4.3.1 Internal criteria . . . 26

4.4.3.2 External criteria . . . 29

4.4.4 Interpretation of results . . . 31

4.5 Dominant sets theory . . . 31

4.5.1 Definitions . . . 31
4.5.2 A game-theoretic perspective of dominant sets . . . 32
4.5.3 On evolutionary dynamics for dominant set extraction . . . 34
4.6 Kmeans theory . . . 37

5 Results **41**
5.1 Information theoretic results . . . 41
5.2 Representational Dissimilarity Analysis . . . 45
5.3 Data represented using t-SNE . . . 49
5.4 Dominant sets on V1 . . . 52
5.4.1 Results of Gaussian kernel on Euclidean distance . . . 52
5.4.2 Results of Gaussian kernel on correlation distance . . . 58
5.5 Kmeans on V1 . . . 62
5.5.1 Results of Euclidean distance . . . 62
5.5.2 Results of 1-Pearson correlation distance . . . 67
5.6 Comparison of clustering results on V1-LM-LI-LL . . . 71

6 Discussion **75**

A Dominant sets on LM, LI and LL **77**
A.1 DS res. on LM with sim. based on Euclidean dist. . . . 77
A.2 DS res. on LM with sim. based on 1-Pearson corr. dist. . . . 78
A.3 DS res. on LI with sim. based on Euclidean dist. . . . 78
A.4 DS res. on LI with sim. based on 1-Pearson corr. dist. . . . 88
A.5 DS res. on LL with sim. based on Euclidean dist. . . . 88
A.6 DS res. on LL with sim. based on 1-Pearson corr. dist. . . . 88

B Kmeans on LM, LI and LL **93**
B.1 Kmeans results on LM with Euclidean distance . . . 93
B.2 Kmeans results on LM with 1-Pearson correlation distance . . . 96
B.3 Kmeans results on LI with Euclidean distance . . . 96
B.4 Kmeans results on LI with 1-Pearson correlation distance . . . 102
B.5 Kmeans results on LL with Euclidean distance . . . 102
B.6 Kmeans results on LL with 1-Pearson correlation distance . . . 102

Acknowledgments **109**

Bibliography **115**

Abstract

Being able of recognizing objects despite changes in their appearance is a necessary skill for the vast majority of living beings. Visual systems, such the one of rats, can solve this task without an apparent effort and with surprising speed. In order to do that, the visual cortex must be able to build a representation of a specific visual stimulus that is invariant to identity preserving transformations. In particular, the rat lateral extrastriate areas are thought to be organized as a hierarchy, each one representing features of a certain complexity. Several studies have shown that, along the V1-LM-LI-LL progression of rat visual areas, information about low-level features is gradually pruned in favor of information about high-level features.

In this scenario, artificial intelligene techniques became extremely important. Information theory measures, dimensionality reduction techniques and unsupervised learning are used for respectively measuring the amount of information in neuronal responses, reducing the high dimensionality of neuronal representation and extracting its hidden structures.

This thesis aims to expand the analysis of object representation in rat visual cortex, and its evolution across the extrastriate visual areas, especially using the dominant sets clustering (Pelillo et al.). The purpose of the latter is to extract maximally coherent groups satisfying internal homogeneity and external inhomogeneity properties by a sequential search of structures in the data adhering to this cluster notion.

Keywords

invariant object recognition, cluster analysis, dominant sets

Chapter 1

Introduction

In our daily life, we are often asked to identify objects in our visual field. Think about identifying a glass on a table, recognizing an obstacle while driving a car, or words while reading a magazine. We perform these activities so easily that we cannot realize the tremendous computational task they entail. Our visual system makes us able to surprisingly quickly detect and classify objects among plenty of possible variations in their appearance, without any evident effort. This set of abilities is called the Core Object Recognition [9]. Among all reasons making the problem so complex, being able of invariantly recognize objects is the most complicated. Indeed, each object is not presented to us, in the same way, more than once. In fact, it is subject of the so-called identity preserving transformations, or better changes in appearance that let object identity be the same (i.e., translations, lighting variations, scaling).

In recent years, many studies have investigated the presence of such abilities on mammals. An example is monkeys, which have demonstrated to accomplish this task thanks to a dedicated area of the visual cortex, which process visual information along the so-called ventral stream [10]. This area is organized hierarchically and is thought to perform object recognition by sequentially building more and more complex representational spaces. By analyzing neural responses, neurobiologists have shown that, in such spaces, objects become progressively more discernible, encoding low-level information in bottom levels and high-level information in the top ones.

However, studying neuronal responses could be a mess. In that scenario, data mining and machine-learning techniques have become invaluable to investigate how the brain and, particularly, the visual system work. Kiani et al. [21] used clustering techniques to examine responses of monkeys inferior temporal (IT) cortex to assess whether the overall population responses represent a categorical structure

of objects (in particular for animate and inanimate). Instead, Zoccolan et al. [2] used again unsupervised techniques to perform a multivariate analysis of monkey IT responses: they found that objects were discernible according to their shape better than how semantic membership does. Furthermore, DiCarlo et al. [46] proposed a high-performing convolutional neural network that matched human performance on many recognition tasks by predicting quite well IT neuronal responses. However, these are only a few examples of the vast amount of studies investigating how monkey brain encodes visual stimuli by using artificial intelligence. Probably, unsupervised techniques are the most used. Indeed, they aim to find underlying data structures without considering any pre-specified information about data. This is exactly the case of neurobiology, where the so-called ground-truth is missing. However, other tools are used too. For example, information theoretic measures have been massively used to study responses of single neurons. Instead, dimensionality reduction techniques have become fundamental to make visualization of neuronal responses possible: they transform the enormous neuronal representation space into two- or three-dimensional maps, extremely useful to perform a first visual analysis on how data is structured.

For long times, these and other techniques have been used almost only to study monkey visual system because of its strong similarity to that of humans. However, other mammals have been becoming subjects of interest to the scientific community. Indeed, new experimental methodologies able to accurately collect neuronal responses have been proposed. In the last few years, rodents have been increasingly considered as models of mammalians visual system. In particular, rats object-processing capabilities are supposed to be located in an area called extrastriate lateral stream [15, 38]. The International School for Advanced Studies (SISSA), located in Trieste, is one of the most prestigious neurobiology research center that has studied such area of rat's cortex [1, 40]. In particular, SISSA scientists perform the following experiment. They prepared a stimulus set of 1440 images of 40 objects posed into 36 different combinations of identity-preserving transformations. Then, they presented these stimuli to rats and recorded neural responses for each of the four levels of the V1-LM-LI-LL progression of which the area is composed. Moreover, scientists collected a set of features for each stimulus image, which were analyzed to verify whether they are encoded in responses. After that, researchers extensively analyzed such datasets, through a wide range of tools like Principal Component Analysis, hierarchical clustering, etc., demonstrating the progressive information reduction about low-level features along the V1-LM-LI-LL progression again, and suggesting that higher levels encode for a combination of other features, to be determined yet [1].

Born as a collaboration between Ca' Foscari University and SISSA, the present study aims to assess and investigate further rats object recognition abilities on the same dataset collection, kindly granted to us. In particular, datasets were analyzed again with both single cell and population analysis. Single cell analysis was performed using information theoretic measures. Instead, population analysis regarded the building of two-dimensional maps using t-SNE (a dimensionality reduction technique), representational dissimilarity matrices (which allow us to have a first overview over data structures), and then the core of the thesis, namely the cluster analysis.

Many clustering algorithms have been proposed in the last years, making hard the decision of which to use. Each one makes some assumptions on data, meaning that it imposes a certain notion of what a cluster is, which could be appropriate or not for a specific dataset. More important, a clustering algorithm usually needs also parameter tuning, a costly task that, if not correctly performed, might bring the examiner to wrong conclusions on data. Our work concentrates on the applications of two different algorithms, kmeans and dominant sets. The first is probably the most famous partitional algorithm ever proposed. It partitions data into a set of a pre-defined number of clusters. In this study, we used both its basic version, using Euclidean distance, and another one based on 1-Pearson correlation distance, a frequently used measure to compare neural responses [22]. The second algorithm is a more recent framework proposed by Pelillo et al., which takes in input similarities between objects and extract maximally coherent groups by exploiting notions of game theory. Again, both Euclidean and correlation distance were used, this time as the input of a Gaussian kernel used for generating similarities. Kmeans asks for tuning the number of clusters, while dominant sets algorithm does not require parameter tuning. However, the similarity function the algorithm uses is strictly related to a parameter, again to be tuned. Cluster validity assessment was used to properly tune such parameters, using MST Dunn's and Silhouette as internal indexes. Then, external criteria indexes were exploited to investigate clusters composition. In particular, Adjusted Mutual Information and purity were extremely useful to verify whether some clusters encode for specific visual stimuli features. Clustering schemas of all levels of the V1-LM-LI-LL progression were then compared to investigate how structures change across the hierarchy.

We discovered that both algorithms, when based on Euclidean distances, produced bunches strongly related to stimulus luminosity, with brightest stimuli particularly discernable. Instead, correlation distances let better emerge clusters oriented to the position of stimuli. This behavior disappears along the hierarchy becoming less emphasized. However, surprisingly the last level LL has shown an

opposite behavior, being luminosity more encoded with correlation distance with respect to Euclidean ones. Furthermore, the analysis reported that specific objects were clearly discriminated from all the others. However, this happened because they were the brightest of the entire dataset, suggesting that their luminosity characterized them much more than their identity.

The thesis is composed of several chapters: a) Chapter 2 briefly discusses the Object Core Recognition problem and provides additional details about the area of rat's visual system the clustering analysis is applied to; b) Chapter 3 firstly recaps the stimuli, features and responses sets, then it explains all methodologies used for single cell analysis; c) Chapter 4 longly discusses all population analysis tools, specially cluster analysis and all theoretic details about dominant sets and kmeans algorithms; d) Chapter 5 reports information theoretic results, representational dissimilarity analysis, and t-SNE two-dimensional maps for all levels of rat's visual system; then reports the complete cluster analysis for the combination of clustering algorithms and distance measures only for the first level V1; as the last point, results of the complete hierarchy are discussed; e) Chapter 6 briefly recaps the entire study and suggests possible future developments.

Chapter 2

Neurobiological context

The present thesis aims to apply clustering and other analysis to neural response datasets. However, to be able to interpret results, it is a must to understand which type of data the algorithm is applied to. The following sections explain the neurobiological context of the present work and, mainly, which are the most important questions the thesis tries to answer to.

2.1 The Core Object Recognition Problem

Being able to recognize words while reading an article on a web page, a pen in your pencil case or a person in a group photo are activities that we perform quickly and effortlessly in our daily lives. Object tracking, segmentation, obstacle avoidance, etc., is a small and also incomplete set of skills that allow us to carry out those activities. Despite their great and fundamental importance, these abilities seem obvious and easy to us. We usually cannot realize the amazing computational challenge they entail. Recognizing objects is an expensive task for many reasons, but mainly because any object can produce an infinite set of different images on the retina, depending on its position, pose, illumination, presence of visual clutter, etc. Although we typically see an object many and many times, we effectively never see the same exact image of that object twice. Thus, the real crux of object recognition is the so-called *invariance problem* [10], or better that an object has to be recognized in spite of the transformations it might be subject to.

Such abilities and, mainly, their high computational performances and robustness, have been massively studied in recent years. In particular, it has been shown that humans and other nonhuman primates can robustly and quickly recognize objects despite changes in object appearance, lighting conditions, position, dimension, rotations, background context and so on [10].

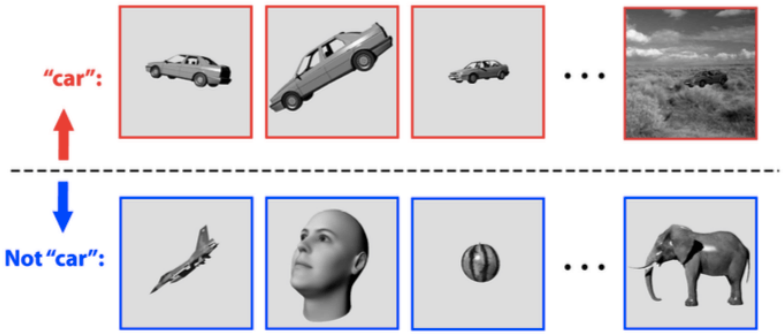
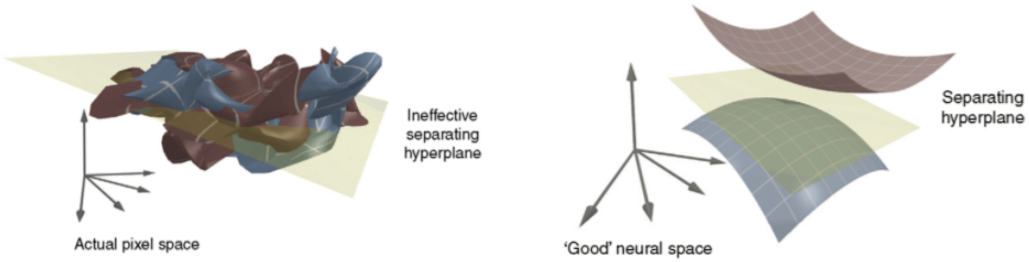


Figure 2.1: Example of discrimination of a visual object (a car) subject to identity-preserving transformations. Adapted from [10]

This family of abilities is known as *Core Object Recognition* that is the ability to exceptionally rapidly and accurately discriminate a given visual object (*identification*) or set of objects (*categorization*) from all other possible objects, materials, textures, over a range of identity-preserving transformations [9].



(a) Initial pixel space

(b) Untangled space

Figure 2.2: Visual representation of the same object on different spaial complexity. Adapted from [9]

To perform such a recognition task, an individual has to produce a specific neuronal representation of the visual scene. Then the brain has to find a decision boundary, or better a hyperplane, that separates the representation of an object from those of all the other. Hence, finding that hyperplane is the heart of the recognition problem. This hyperplane is computed by performing a weighted sum of the neuronal responses after receiving the image of the scene [9]. However, this step could be difficult because of the high complexity and dimensionality of the neuronal representation. One way to solve the problem is to progressively transform the retinal representation into new ones more linearly separable. That is thought to happen in the visual system of primates [9, 10]. At the retinal level, object manifolds, i.e. the set of all the responses elicited by one object, are highly curved and tangled with each other (Figure 2.2a), but they are progressively reformatted

to make it possible and easier to find the decision hyperplane that separates them (Figure 2.2b). Such process is thought to be performed by the area of the visual system described in the following sections.

2.2 The primate visual system

The primate visual cortex is a hierarchical system composed of many areas, organized in two distinct processing streams: the *dorsal* and the *ventral stream*, both starting from the primary visual cortex V1 (see Figure 2.3).

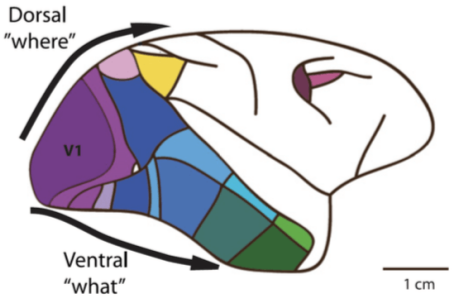


Figure 2.3: Dorsal and ventral pathways in visual system of macaque brain. Adapted from [28]

The *dorsal stream* processes visual input to obtain position and direction of motion of the objects in the visual field, while the *ventral stream* processes visual input to define shape and identity of each object in the visual environment. The first is also called *the where stream*, the second *the what stream* [1]. The ventral stream is thus a prime candidate as the area where the problem of invariant visual object recognition is solved, a fact confirmed by the impairment in object recognition performance arising after lesions. However, this description regards the primate visual system. From now on, we will focus on the ventral stream homologous in the rat visual system, the area of interest of the present thesis.

2.3 The rat visual system

Because of the strong similarity between theirs and our visual system, monkeys, especially macaque, has been being the most used model in order to understand how object vision works. Generally rodents have not been considered interesting by the visual neuroscience community because of their different anatomy. Moreover, they are crepuscular animals; hence they rely also on other senses to perceive the world. For such reasons, the rat has not been considered as a good model until

recently. In the last few years, however, there has been a rising interest in the rat as a model of mammalian vision. In particular, it was discovered that the visual system of rats is composed of 13 different areas, the biggest starting in V1. This region is adjoined by many extrastriate areas in both *lateral* and *anterior* directions [38], which are organized in a hierarchical fashion [15] (see Figure 2.4). Moreover, evidence suggests that rodents anterior and lateral areas are analogous to monkey dorsal and ventral streams [28]. For such reason, we use the terms rat's lateral stream and rat's putative ventral stream interchangeably in this thesis.

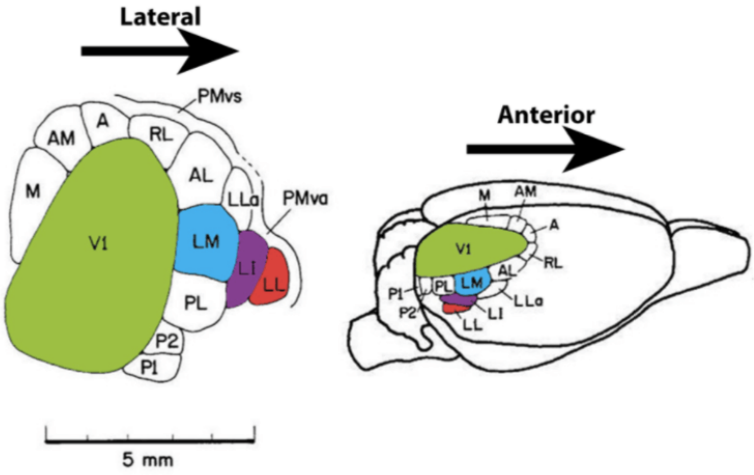


Figure 2.4: Lateral and anterior pathways in visual system of mouse brain. Adapted from [38]

The putative ventral stream, which is composed of the V1-LM-LI-LL progression, is the subject of the present work. Many studies were conducted to assess the capabilities of rat putative ventral stream, showing that rats own the ability to recognize objects in spite of the aforementioned identity-preserving transformations. Such process is believed to be performed in the putative ventral stream, where object manifolds are gradually pruned from low- and mid-level information. [1, 40].

This thesis purpose is to study that area of rat's visual system by applying the same analysis on all levels V1, LM, LI and LL of the putative ventral stream to assess recognition abilities with other machine-learning techniques.

Chapter 3

Experimental design and single cell analysis

The chapter will discuss the experimental design followed by SISSA scientists and the first step of the analysis, namely the single cell analysis.

3.1 The experimental design

In the following sections, description about stimulus, features and responses sets are presented. The International School for Advanced Studies (SISSA), located in Trieste, kindly granted such data. To get more details, especially those related to surgical aspects, electrophysiology and spike sorting, refer to [1].

3.1.1 Stimulus set

In their experiments, SISSA researchers built a large stimulus set using the 3D models of 40 real-world objects. Figure 3.1 illustrates the tree structure of such objects.

The tree was built following a semantic criterion. However, that choice was not made because semantic abilities were supposed to be present in the analyzed area of the rat's putative ventral stream. Classes are only used to identify visual similarity among objects (like the shape, textures, etc.) [1]. The stimulus set is composed of both natural and artificial objects. In particular, natural objects are divided into animals (in turn divided into four legs animals, insects, and birds) and plants (composed of trees, fruits, and vegetables). On the contrary, artificial objects are divided into vehicles and furniture. Again do not consider a class as a semantic one. For example, the four legs animal class is made of objects that should share the same shape. Conversely, the fact that two objects are not in the same class

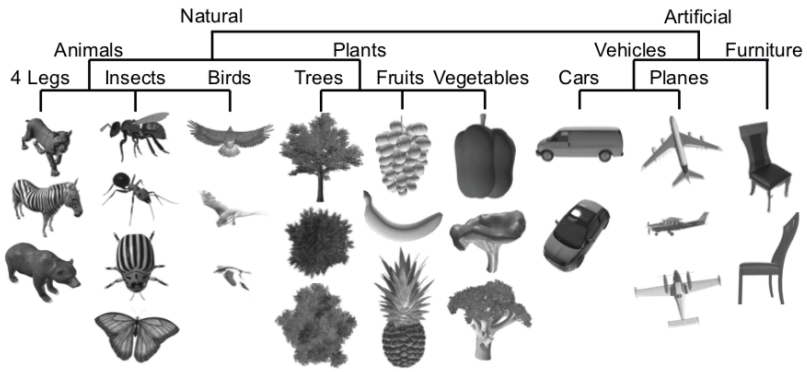


Figure 3.1: Tree structure of object models used to build the stimulus set. Adapted from [1]

does not imply that they do not share any visual feature. Indeed, the eagle and the plane, even if belonging to two different classes, have the same shape.

Each object was rendered in 1920×1080 grayscale images with 36 different random combinations of transformations. Especially each combination was composed by the choice of a pose (frontal, lateral, top and 45° in azimuth and elevation), position (−15°, 0° and 15°), size (30°, 35° and 40°) and rotations (−45°, 0°, 45° and 90°). Then random noise was applied to increase stimulus variability. For further details refer to [1].

3.1.2 Response set

For each stimulus, neural responses were collected at each level of the V1-LM-LI-LL progression. Furthermore, responses are classified into two classes: single and multi-units. The term single unit refers to the response of a single neuron that was correctly isolated. The contrary happens for multi-unit, representing a mixture of responses of many neurons. Table 3.1 illustrates the number of single-units (SU), multi-units (MU) and the total number (SU-MU) of neural responses recorded for each level.

Level	SU units	MU units	SU-MU units
V1	44	133	177
LM	11	27	38
LI	28	55	83
LL	19	29	48

Table 3.1: Number of recorded neural responses for each level and unit type

3.1.3 Feature description

Once stimulus images were generated, SISSA researchers analyzed each one and extrapolated its visual characteristics. They extracted the bounding ellipse, the perimeter and the pixel intensity. Thenceforth, they defined for each stimulus a set of low- and mid-level features. Particularly, they collected stimulus position on the screen, luminosity, area, contrast, aspect ratio and orientation. Luminosity was an extensively studied feature, such that two different definitions were proposed:

$$L_{Total} = \sum_{x=1}^{1920} \sum_{y=1}^{1080} I(x, y), \tag{3.1}$$

$$L_{RF} = \sum_{x=1}^{1920} \sum_{y=1}^{1080} I(x, y) \cdot RF(x, y), \tag{3.2}$$

where $I(x, y)$ and $RF(x, y)$ are, respectively, the intensity and the receptive field normalized intensity of pixel (x, y). Rather, $\overline{I}$ is the mean intensity of all pixels.

Figure 3.2 shows histograms of each feature. Both position and orientation show peaks in the distribution, three and four peaks respectively, reflecting how stimuli were generated. Instead, total luminosity, contrast, and area show a single peak distribution.

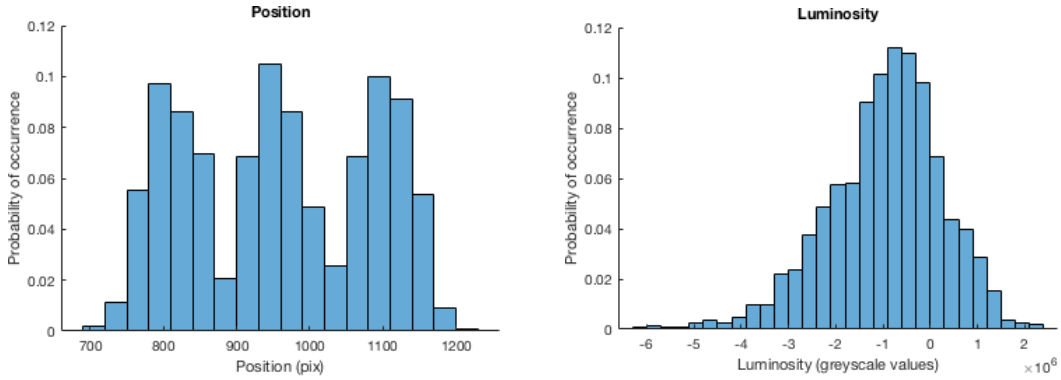
3.2 Single cell analysis

The first step of the analysis is to consider each neuron separately and find which is the relationship between the input image and its output. Since such relation is thought to be highly nonlinear, it is a must to use proper tools that can capture such relationship.

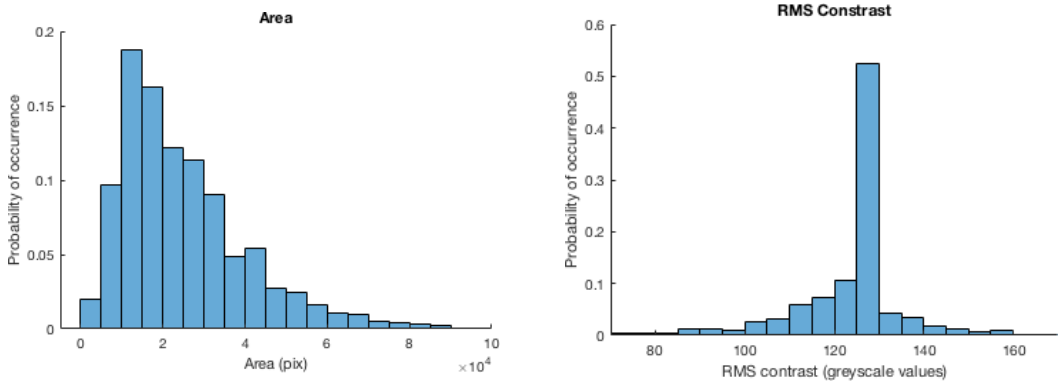
3.2.1 Information theory measures

Information theory is an increasingly used framework that comes from the mathematical theory of communication. Information measures are often applied to study how the brain encodes visual and other sensory information [6]. Indeed they have important attributes which make them high-quality tools to determine if and how the response modulates the stimulus. The most important is their non-linear nature, making them very well suited to capture whatever type of relation, including the one between neural responses and features.

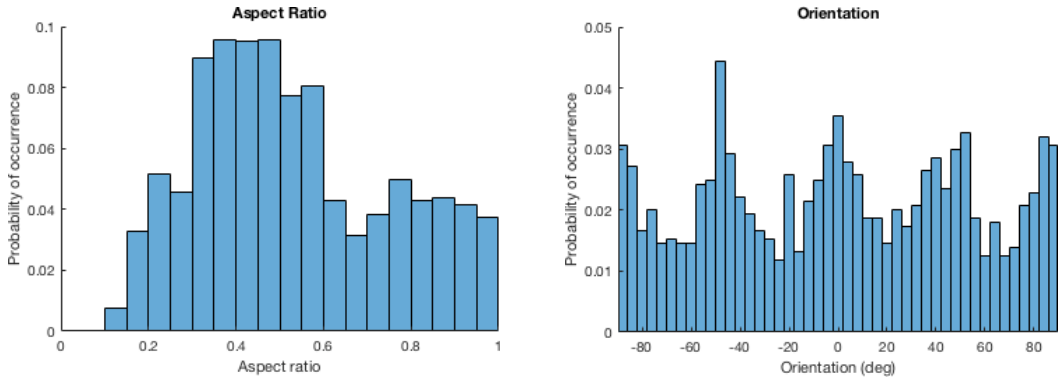
Using Shannon's mutual information [8, 39] is one way to measure whether



(a) Position **(b)** Total luminosity



(c) Area **(d)** RMS Constrast



(e) Aspect ratio **(f)** Orientation

Figure 3.2: Histograms of stimuli features

neural responses code for stimuli parameters. It measures the amount of conveyed information about a certain variable given that another one is known. It could also be seen as the quantification of uncertainty reduction about one variable after the observation of another [19]. Let the former be the feature of a stimulus F and the latter the corresponding neuronal response R. The mutual information formula is the following:

$$I(R;F) = H(F) - H(F|R). \tag{3.3}$$

The first term $H(F)$ is the feature entropy and quantifies the overall variability of the stimulus parameter. It is expressed as:

$$H(F) = -\sum_{f} P(f) log_2 P(f), \tag{3.4}$$

$P(f)$ being the unconditional probability of observing the stimulus parameter f. The second term $H(F|R)$ is the noise entropy which quantifies the stimulus variability given fixed a certain response. It is defined as:

$$H(F|R) = -\sum_{r} P(r) \sum_{f} P(f|r) log_2 P(f|r), \tag{3.5}$$

$P(r)$ and $P(f|r)$ respectively being the unconditional probability of observing the response r at any condition and the conditional probability of observing feature f given that a specific response r has been recorded. If information assumes a positive value, then stimulus feature and response are dependent. Viceversa, no relation occurs between response and feature when $I(R;F) \approx 0$.

All these considerations are made considering only one feature. However, suppose that multiple features define the stimulus set. Let F_i be the i-th feature. Then the mutual information between neural response and such feature is denoted by $I(R;F_i)$. In that case, two features might be correlated. Let's suppose that features F_i and F_j are dependent, or better $I(F_i;F_j) > 0$. The fact that $I(R;F_j) > 0$ does not imply that responses code for F_j, since it is related to F_i and thus it might happen that responses code for F_i. In such cases, it is essential to be able to distinguish the true relations. Conditional mutual information [19] could be used to solve this problem:

$$I(R;F_j|F_i) = I(R;F_j\&F_i) - I(R;F_i), \tag{3.6}$$

where F_i and F_j are the i-th and the j-th features. This measure expresses the amount of information conveyed by the response R about the feature F_j at

fixed values of F_i. Indeed, since the value of F_i is fixed, any dependence between the response and F_j represents genuine information that cannot be explained by the relation between F_i and F_j. The feature F_j has a genuine influence on the response beyond the effect of its relation with F_i if and only if $I(R; F_j|F_i) > 0$. Indeed by equation 3.6, this happens when $I(R; F_j\&F_i) > I(R; F_i)$, thus when joint information is bigger than the conveyed by only F_i, meaning that also F_j carries information.

The direct computation of such measures could result in a systematic estimation bias due to the finite sampling. The Panzeri-Treves correction [26, 31] was applied on unconditional and conditional mutual information to remove such bias.

The analysis presented in Chapter 5 consists on computing mutual information between features and responses of all the levels of the V1-LM-LI-LL progression. However, each level is composed of a set of neurons. To be able of quantifying the information conveyed about the feature by the complete set of neural responses of a level, the median and the standard error of the information between the feature and all neurons are considered. Moreover, a separate analysis was performed for single and multi-units neural responses. When a dependence between two features was suspected, conditional mutual information was used to assess it. The 1-tailed Mann-Whitney U test [14], Bonferroni-Holm corrected [18], was applied to check whether differences among the levels were significant. All analysis regarding information theory were performed using the *Information Breakdown ToolBox for MatLab, IbTB* [26].

3.2.2 Feature and response bins

Notice that all measures in the previous section refer to discrete stochastic variables. Moreover, *IbTB* requires discrete variables to make it easier to quantify the necessary probabilities for the computation of the measures [26]. For such reasons, features and responses were discretized into a specific set of meaningful bins. SISSA researchers proposed reasonable binnings according to feature and response distributions. For example, the feature position was divided into three bins, representing left, center and right positions. The same happened for orientation, whose distribution (see Figure 3.2f) suggested to use of four bins. For the other features, whose distribution did not present more than one peak, ten is number of bins in which the variable was discretized. Instead, for object identity the number of selected bins was trivially forty, each one representing a model used to generate the stimuli. Moreover, responses of each neuron were binned into three bins. All binnings were selected with the purpose to take at least five trials for each bin to

avoid limited sample bias, as suggested in [30, 31]. We used the same binnings in cluster analysis to investigate clusters composition.

Chapter 4

Population analysis and clustering algorithms

Population analysis deals with the general behavior of neuronal responses belonging to the same area of the visual cortex. This chapter will first describe the population vector space. Then it discusses representational dissimilarity analysis and the t-SNE technique. Moreover, it will longly describe the cluster analysis process and theory about dominant sets and kmeans algorithms.

Single cell analysis aims to discover the relation between features and responses of single neurons, without dealing with the general behavior of neurons belonging to the same area of the visual cortex. Population analysis deals with the latter aspect, whose tools will described in this chapter.

4.1 Population vector space

As the first step, let's define the *population vector space*. Consider a population of P neurons. The state of such population on a specific time consists of the set of instantaneous firing rates $\boldsymbol{\nu} = (\nu_1, \nu_2, ..., \nu_P) \in \mathbb{V} \subseteq \mathbb{R}^P$. Thus, the population evolution can be described by a trajectory on $\mathbb{V}$, while the responses to a stimulus is a single point on such trajectory. Consider a set of N stimuli, then the i-th stimulus is defined by the vector $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{iP}) \in \mathbb{V}$, with $i = 1, ..., N$. We normalized population state vectors by subtracting mean and dividing by standard deviation as suggested in [21, 22].

The space over defined could be a good representation of the one described in Section 2.1. On that space, all points corresponding to an object lie on the same manifold. As already explained, being able of finding a hyperplane that separates manifolds of two different objects is the core problem of invariant recognition

abilities. Moreover, such manifolds should be tangled in the first levels, where only low-level features are thought to be encoded, and becomes more and more linearly separable across the V1-LM-LI-LL progression.

4.2 Distance measures and RDMs

After having defined the population vector space, it is a must to determine also the distance over its points. There are many metrics proposed in the literature [22]. The 1-Pearson correlation is one of the most popular, which focuses on patterns of the population response and discards effects that nonspecifically modifies neurons firing rate [2, 21, 46]. The Euclidean distance, the Mahalanobis distance or t value distance are other very often used metrics. Among these distances, the 1-Pearson correlation and the Euclidean distances were selected, the former because of its vast usage, the latter for the sake of comparison.

The 1-Pearson correlation distance between stimulus $\mathbf{x}_i$ and $\mathbf{x}_j$, belonging to $\mathbb{V}$, is defined as:

$$C(\mathbf{x}_i, \mathbf{x}_j) = 1 - \rho(\mathbf{x}_i, \mathbf{x}_j), \tag{4.1}$$

with $\rho(\mathbf{x}_i, \mathbf{x}_j)$ defined as the correlation between the two vectors $\mathbf{x}_i = (\mathbf{x}_{i1}, ..., \mathbf{x}_{,P})$ and $\mathbf{x}_j = (\mathbf{x}_{j1}, ..., \mathbf{x}_{jP})$. Furthermore, the Euclidean distance between the two vectors is given by the norm of the difference:

$$E(\mathbf{x}_i, \mathbf{x}_j) = ||\mathbf{x}_i - \mathbf{x}_j||_2. \tag{4.2}$$

These distances were used along the whole analysis of the V1-LM-LI-LL progression. In particular, they can be visualized with the so-called Representational Dissimilarity Matrices (RMDs) [22], often used to compare representational geometries between brain areas. The matrix shows as much rows and columns as the number of presented stimuli. Each cell (i, j) of the matrix represents the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. Such matrix could be encoded as an image, in which each cell becomes a pixel whose color depends on the distance the cell contains. The matrix might not report any interesting pattern at a first sight. The trick is to use a specific sorting criterion for rows and columns. For example, stimuli can be ordered according to their position on the screen. If responses of stimuli with a similar position are close together in the population vector space (according to a specific distance metric), the corresponding RDM should present diagonal blocks, meaning that groups of similar stimuli are present.

In Section 5.2, RDMs are extensively used both to visually assess how features

are encoded in the various level of the V1-LM-LI-LL progression and also to check clusters goodness, by ordering rows and columns by cluster index first and then by stimulus identity.

4.3 Dimensionality reduction with t-SNE

Today datasets are characterized by high data dimensionality. Because of this, it could be difficult to visualize data, which is often very useful. A way to address this problem is to use specific dimensionality reduction techniques, which concentrate as much information as possible in few variables. One of these techniques is t-SNE [25], a non-linear dimensionality reduction tool which creates a two- or three-dimensional map. The map is built such that each data point has its own location and could be easily displayed on a scatter plot. The mapping reveals local and global data structures at many different scales. Indeed, when high-dimensional data points lie on a low-dimensional and non-linear manifold, it keeps such data points as close as possible in the low-dimensional representation. In other words, similar objects are modeled as nearby points, while dissimilar objects are mapped into two distant points. This fact makes this reduction technique particularly interesting for our analysis because, as described in 2.1, stimuli of the same object should lie on a precise manifold. The algorithm basically consists of two steps. In the first step, t-SNE builds a probability distribution over data points pairs in such a way that similar objects have a high probability of being picked, while pairs of different objects have low probability. After that, the same probability distribution is defined over the lower-dimensional map such that the Kullback-Leibler divergence between the two distributions is minimized. Obviously, a measure of similarity, or better of dissimilarity, should be provided for the construction of the probability distribution. We obviously proposed again both the Euclidean and the 1-Pearson correlation distances, defined in Section 4.2.

4.4 Cluster analysis

In recent years, various reasons have made data availability dramatically increasing. Consider sensing and storage technologies. They have become more advanced and cheaper, making growth both amount of data and demand for access and analyze it. These factors provide vast potential for the development of systems that automatically perform data analysis, classification, etc. Think about digital and video cameras, become inexpensive and more capable in storage. That technology

could temp a business to provide itself with a surveillance system that automatically recognizes its employees and warns when a stranger enter offices. Nevertheless, this is only one of the possible scenarios. Indeed, to exhaustively list all scientific fields and applications which have to deal with these high-volume, high-dimensionality and often unstructured datasets is challenging. Computer vision, cybersecurity, neurobiology, bioinformatics, etc., are only a few examples.

Many data analysis techniques which can process and, majorly, understand data have been developed to meet these needs. In particular, the process of understanding data patterns to predict the behavior of unseen test data is called *learning*. Usually, learning problems are divided into three reference families [7, 11]:

- *supervised:* knowledge about the dataset is complete, and each training sample is labeled; techniques aim to provide data classifiers which map unseen data into correct class;
- *unsupervised:* knowledge about training samples is absent; that techniques aim to find which patterns rule data;
- *semi-supervised:* it is a halfway between supervised and unsupervised learning in which the algorithm is provided with unlabeled data and other information.

Among the unsupervised problems, clustering emerges as the one of the most difficult but also useful and fascinating. Clustering is an exploratory process used to identify interesting patterns in the underlying data without providing any information regarding grouping properties [16]. Thus, the aim is to extract sensible groupings which allow to discover similarities and differences in data. Recent studies have extensively worked on giving a definition on what “sensible groupings" stands for [20]. In the scientific community, it is defined as a partition in which each group satisfies both internal homogeneity and external inhomogeneity, also named compactness and separation [3]. Compactness means that elements inside a cluster should be as close as possible to each other. Moreover, separation means that members of a cluster should be dissimilar to the ones of other clusters. Hence, clustering refers to the process of extracting maximally coherent groups from the datasets. The partition and how such process is conducted depend on the algorithm used. Recently many different procedures have been defined, each one focused on specific criterions, assumptions and parameters which strongly influence the type of clusters that can be found [4, 11]. For example, Kmeans [4, 20] depends on the number of clusters K and tends to generate spherical clusters. Spectral clustering [24] is another widely used algorithm, which depends again on the number

of clusters but also on a similarity function, which could depend, in turn, on other parameters. As one can understand, the range of possible options is huge. Once having selected the clustering algorithm, it is fundamental to understand which are its inherited advantages and disadvantages and, mainly, which is/are the parameter value/s that produce the best clustering partition. Indeed, improper values could result in bad partitioning scheme which is not optimal for the specific case and bring us to wrong decisions. Moreover, such alleged best partition should be studied and evaluated to understand how much it is good according to specific characteristics, like again compactness and separation. The whole of this phases composed a formal study also referred as *Cluster analysis* [13]. The process is summarized by the following steps: feature selection, clustering algorithm, cluster validity assessment and interpretation of results. In the below sections, these phases are discussed, both in the general and the specific case of the present thesis.

4.4.1 Feature selection

A set of common features define each member of the dataset, someone more fundamental than others. Being able of selecting the proper subset of that features is the first and, probably, the most important step of the cluster analysis. Such set of features strongly influences clustering results. Indeed, they define the hidden structures which let clusters emerge. Missing a fundamental feature for describing dataset members might bring the clustering algorithm to find a false "good" partition, causing wrong interpretation of data.

In this thesis, the set of features is given and corresponds to the set of neuronal responses of each stimulus presented to rats. Moreover, clustering algorithms are run over both single and multi-units responses to capture the entire collected information. Refer to Section 3.1.2 to find the number of neural responses for each level of the V1-LM-LI-LL progression.

4.4.2 Clustering algorithm

Choosing the clustering algorithm is the second step of cluster analysis. Each algorithm is different from the others and its characteristcs are defined according to the following aspects [16]:

- *type of input data:* the algorithm can behave differently according to data features, like geometries, etc.;
- *proximity measure:* is the way to quantify how much two objects are close to each other, possibly considering all available data features;

- *clustering criterion:* is expressed as a set of rules (like the cost function) that guides the algorithm during the cluster extraction process;
- *underlying theory:* the set of fundamental concepts on which the clustering technique is based.

Considering that properties, it is possible to create several classifications over the different algorithms. According to how the algorithm defines what a cluster is, the following four families of clustering procedures are defined:

- *Partitional clustering:* algorithms find all clusters simultaneously as a partition of data, without imposing any hierarchy; an iterative procedure that optimizes a specific criterion or cost function is necessary to perform clustering;
- *Hierarchical clustering:* algorithms do not provide a single partition of data, rather a hierarchy of bunches that merge with each other at certain distances; the recursive procedure that finds nested clusters could be either agglomerative or divisive; the produced hierarchy consists on a tree of clusters, called dendrogram, which helps the user to visually assess which are the clusters, their members and at which height the union of two clusters happens; cutting the dendrogram at a specific height produces a disjoint set of bunches;
- *Density-based clustering:* algorithms aim to group neighboring objects into clusters basing on some density conditions; a cluster coincides with an area of higher density than the rest of the other objects;
- *Grid-based clustering:* specific of spatial data mining, these algorithms quantize the space into a finite set of cells and do all computations on that space.

Furthermore, whether the algorithm creates overlapping partitions or not is another important classification criterion:

- *Fuzzy clustering:* overlapping clusters are allowed, meaning that an object can belong to more than one cluster; usually, a measure of participation is associated to each element; these partitions are considered more compatible with many real applications;
- *Crisp clustering:* overlaps among clusters are not allowed, meaning that each object can belong to only one cluster and not.

Among the multitude of proposed approaches, we selected kmeans and dominant sets for the cluster analysis of this thesis. We used kmeans to expand analysis

in [1] and compare results with dominant sets algorithms, used to try a different clustering approach on the dataset.

Kmeans [4, 20] is a partitional clustering algorithm that takes in input a numerical dataset and the number of clusters K. Then, the partition is found by minimizing the squared error among cluster centroids and the corresponding cluster members. However, with its simplicity, this approach and, in general, partitional algorithms have some drawbacks. First of all, the number of clusters should be either known or tuned, often implying high computational costs. Moreover, every object is hardly assigned to a single one cluster because a partition is produced. This fact involves that: a) outliers are attached too, while it could be interesting to leave them unassigned; b) an object is assigned to only one cluster, while, in many real applications, overlapping partitions better fit the problem. Furthermore, clustering algorithms usually make some strong assumptions over the distance (or similarity) function used to compare objects, like symmetricity, non-negativity, etc.). Specifically, kmeans clustering tends to produce spherical clusters, meaning that different shapes cannot be detected. Furthermore, kmeans cannot work with categorical data. Indeed, as its name suggests, it is a centroid-based clustering, where the prototype coincides with the mean of the cluster members. Consider that such observations regard the basic version of kmeans. Nevertheless, many optimized versions of the algorithms have been presented, like Fuzzy c-means that allow overlapping clusters, and many others.

Dominant sets algorithm [35] follows a different approach to cluster data and overcomes some of the preceding problems. In particular, it provides a new definition of what a cluster is. A cluster is defined as a maximally coherent group, or better a bunch of elements which shows both compactness or internal homogeneity (members of a cluster should be highly similar) and separation or maximality (no other element should be introduced in the cluster). This notion is intriguingly linked with various scientific fields. From a game-theoretic point of view, clustering corresponds to a non-cooperative game, the so-called *clustering game*, whose Evolutionary Stable Strategies (ESSs) correspond to clusters. In graph theory, it corresponds to the notion of maximal clique on weighted graphs. Moreover it is related to solutions of simplex-constrained quadratic optimization problems. Being able of understanding how such fields are connected is fundamental to get entirely how dominant sets algorithm works. Practically, the algorithm takes in input similarities among pairs of objects and sequentially extracts maximally coherent groups until at least one is present. Thus, if a similarity function is defined over data, then there is no restriction on the input type. It makes no assumption on both the underlying data representation and the structure of the affinity matrix, which could be both

symmetric and asymmetric. This implies that the number of clusters should not be known a priori and whatever cluster shape could be found. Furthermore, another significant advantage is that it leaves clutter elements unassigned, marking them as "noisy" members. Again these considerations regards only the basic version of the framework. Its authors have proposed many extensions. For example, dominant sets, if correctly used, can find overlapping clusters [41]. Moreover, it generalizes naturally to hypergraph clustering problems [34]. This is only a brief overview of the two algorithms. Refer to Sections 4.5 and 4.6 to find all technical details.

Let's now concentrate on practical details, specific for the present work which aims to cluster stimuli hardly. As already explained in Section 4.2, Euclidean and correlation distances have been individuated as two good measures to underline stimuli differences. Consequently, it is interesting to run the two algorithms with both. For both distance measures, kmeans was run ten times, each of which run the algorithm again 1000 times with different centroid initialization and returned the best partition. The best over these 10 final partitions was analyzed. Then, dominant sets does not take in input distances but similarities. Thus proper similarity functions must be defined, one for each distance measure. Both were created using a Gaussian kernel that takes in input the metric. Consider N the number of members of the dataset and $\mathbf{x}_i$ the i-th element, with $i = 1, .., N$. Then the first similarity function, which considers Euclidean distances, is defined as:

$$s_E(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{ - \frac{||\mathbf{x}_i - \mathbf{x}_j||_2^2}{2\sigma^2} \right\}.$$

The second similarity function, defined over correlation distances, is:

$$s_C(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{ - \frac{(1 - \rho(\mathbf{x}_i, \mathbf{x}_j))^2}{2\sigma^2} \right\},$$

where $\rho(\mathbf{x}_i, \mathbf{x}_j)$ is the correlation of vectors $\mathbf{x}_i$ and $\mathbf{x}_j$. Notice that, both algorithms request the tuning of a parameter. Specifically, kmeans needs the tuning of the number of clusters K, which in the analysis is considered into the set $\{1, ..., 15\}$. Indeed, a higher number of clusters might be difficult to be interpreted. Instead, dominant sets does not require to tune specific parameter for the algorithm, but the similarity matrix. Indeed, the parameter $\sigma > 0$, linked to the Gaussian kernel, must be properly tuned too. It was considered into a specific interval according to the distance measure used for creating the similarity kernel. Indeed, as Chapter 5 shows, σ acts as a scattering factor on data points: a smaller value of the parameter implies very low similarities among data, and thus a high number of final clusters; instead, a bigger value of the parameter implies high similarities and

thus lower number of clusters. We wanted to study this behavior starting from a small value of σ, approximately zero, to its smallest value for which the partition consisted of only one cluster. The velocity with which this happens is different for each metric. Thus the two intervals are different. Using the Euclidean distance, considered values of σ was in the set $\{1, 2, ..., 850\}$; instead, considered values for correlation distance was in the set $\{0.1, 0.2, ..., 15\}$. Such intervals were used for each area of the V1-LM-LI-LL progression.

4.4.3 Cluster validity assessment

As stated in the previous section, clustering algorithms strongly depend on specific parameters. Different parameters generate different partitions, each one unique and fitting underlying data in a certain way. Among the entire set of possible values the parameter can assume, to pick the one generating the best partition could be a difficult task. A way to address the problem is to verify the validity of results visually. Think about clustering used for segmenting an image. A simple criterion to check validity is to visually assess whether objects in the image are correctly identified or not. However, this procedure is possible only because the so-called "ground-truth" is present and the dataset was defined only on two dimensions. When dataset dimensionality is higher, this task becomes more challenging. Therefore, some statistical tools are needed. As already said, clusters should exhibit two different aspects: compactness and separation. Hence, quantifying both is a way to address the problem. The quantitative evaluation of clustering results is also called *Cluster validity*, an analysis that makes use of various indexes to choose parameter values and then to relate partition with others defined on the same dataset. In cluster validity there are two different approaches to quantify the partition goodness: *internal* and *external criteria* [16]. *Internal criteria* aims to evaluate results only considering that similar objects should belong to the same cluster, while dissimilar objects should belong to different clusters. In other words, it again considers compactness and separation. *External criteria* aims to compare two different partitions by checking how much they correspond. Specifically, the first partition is usually the one produced by the clustering algorithm. Instead, the second partition is often an external benchmark, also called *ground-truth* labels. They are useful to check if produced clusters reflect some specific criteria or grouping properties data might have.

4.4.3.1 Internal criteria

Consider first *internal criteria* indexes. The indexes are used to identify algorithm parameters correctly. Usually, the procedure used to pick them is the following [23]:

1. Step 1: list the set of considered clustering algorithms;
2. Step 2: execute each algorithm with a specific set of parameters and collect results;
3. Step 3: for each clustering schema obtained in the previous point, calculate internal validation index;
4. Step 4: choose the best clustering partition according to such criteria.

In recent years, a various list of internal indexes has been proposed, someone considering both compactness and separation, other considering only one of the two aspects. However, these indexes are often affected by various factors. First of all, some indexes represent the same objective function the algorithm optimizes. Think about one of the most common internal indexes used with kmeans, the so-called *SSE (Sum of Squared Error)*. A simplistic way to raise its value is to increase the number of clusters data is partitioned. Hence, it could be challenging to select the proper cluster number. With indexes that strongly depends on that factor, called *monotonicity*, the optimal cluster number is reached on the elbow of the curve representing the index. However, this is only one of the influencing factors. For example, *outliers presence* strongly influence distances among data points, and thus both compactness and separation. Indexes that does not address this problem could report wrong optimal values. Other factors could be the *density of clusters*, *subclusters presence*, *skewed distributions* or *clusters shapes*, etc. In addition to that aspects, we put another requirement. Two different distance measures were used in clustering algorithms. Thus indexes that do not depend on the used metric were discarded. Indeed, to be congruent, indexes like *RS*, which is the ratio of cluster between and total variances, cannot be used if clusters were found using correlation distances. Indeed, these indexes are computed using only Euclidean distances. Thus, indexes that depend on the distance metric should be used. After having analyzed and studied the majority of proposed indexes, two were selected as the ones that mainly satisfies the above specifications: *Silhouette* and *MST Dunn's indexes*.

The *Silhouette* [37] index combines both compacteness and separation. Consider K number of clusters, c_i the i-th cluster, $c_{\mathbf{x}_i}$ cluster of data point $\mathbf{x}_i$ and $d(\mathbf{x}, \mathbf{y})$ the

distance (Euclidean or 1-Pearson) between the two data points $\mathbf{x}$ and $\mathbf{y}$. Moreover, consider $d(\mathbf{x}, c)$ the average distance of the data point $\mathbf{x}$ to the cluster c:

$$d(\mathbf{x}, c) = \frac{1}{|K|} \sum_{\mathbf{y} \in c} d(\mathbf{x}, \mathbf{y}).$$

Then for each data point $\mathbf{x}_i$, $a_i = d(\mathbf{x}_i, c_{\mathbf{x}_i})$ represents the average distance of the point from the other members of the cluster it belongs to. Instead, $b_i = \min_{j=1,..,K, c_j \neq c_{\mathbf{x}_i}} d(\mathbf{x}_i, c_j)$ represents the average distance of the point $\mathbf{x}_i$ to its clostest cluster, obviously without considering the one it belongs to. Having defined these two measures, Silhouette index for the single data point $\mathbf{x}_i$ is the following:

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)},$$

which assumes values in the interval $[-1, 1]$. Situations in which $s_i \approx -1$ are not desirable since this happens when $a_i > b_i$, or better when the point is more similar to another cluster than the one it is assigned. It is preferable that all data points have higher values of the silhouette index. After having computed the index for every single element, the overall index of the partition defined by the K clusters is simply the mean:

$$S_K = \frac{1}{N} \sum_{i=1}^{N} \frac{b_i - a_i}{\max(a_i, b_i)}.$$

The index does not show any trend over the number of clusters; thus optimal parameter value is determined by the maximum of the index. Moreover, it does not particularly suffer from noise, cluster density, and skewed distributions. However, the index is not able to advise the presence of subclusters, indicating wrong parameter value in that case [23]. Even so, it remains one of the best and most used indexes, and the advantage of using it with whatever distance measure makes it fundamental for the analysis.

To not just trust on what a single index suggested, we used also *MST Dunn index* [29], a variation of the original one proposed by Dunn [12] that tries to reduce the influence of outliers exploiting the concept of minimum spanning tree. Consider first the initial version. It formulates intra-cluster compactness as the maximum diameter among all clusters and inter-cluster separation as the minimum pairwise distance between clusters. It assumes positive values and corresponds to the ratio between separation and compactness measures:

$$D_K = \min_{i=1,\dots,K} \left\{ \min_{j=i+1,\dots,C} \left(\frac{d(c_i, c_j)}{\max_{k=1,..,K} diam(c_k)} \right) \right\},$$

where:

$$d(c_i, c_j) = \min_{\mathbf{x} \in c_i, \mathbf{y} \in c_j} d(\mathbf{x}, \mathbf{y})$$

is the dissimilarity function between clusters c_i and c_j and:

$$diam(c_k) = \max_{\mathbf{x}, \mathbf{y} \in c_k} d(\mathbf{x}, \mathbf{y})$$

corresponds to the diameter of a cluster which measures its dispersion. If clusters are compact, the diameter is expected to be small. Furthermore, if clusters are well-separated, inter-distances are expected to be high. Thus, higher values of the index indicate better partitions. However, such index is particularly sensitive to outliers since their presence makes the diameter increase [23]. For that reason, a modified version of the index which uses MST was proposed. Consider a cluster c_k and the complete subgraph it induces. Each vertex represents a cluster member, while each edge represents the relation between two elements, whose weight corresponds to their distance. Let's define $diam^{MST}(c_k)$ as the weight of the maximum edge in the minimum spanning tree found on that graph. Then MST Dunn's index is defined as follow:

$$D_K^{MST} = \min_{i=1,\dots,K} \left\{ \min_{j=i+1,\dots,C} \left(\frac{d(c_i, c_j)}{\max_{k=1,..,K} diam^{MST}(c_k)} \right) \right\}.$$

In such way outliers influence is considerably reduced. As the original Dunn's index, the parameter that maximizes it is the optimum one. Similarly to the Silhouette index, this measure was selected because it depends on the chosen distance metric, without imposing one a priori.

We considered both indexes when we tuned parameters. The tuning was done separately for each combination of algorithm and distance metric. With kmeans we evaluated the index for each tested value of K and then selected the best one. With dominant sets algorithm we did the same, considering again partitions produced for each σ and computing the two indexes. Consider that similarities were not accounted for the index computation, which only use distances. In this way, σ did not effect indexes computation, making comparisons among partitions produced by different σ meaningful.

4.4.3.2 External criteria

Let's now move to *external criteria*. As already said, these indexes aim to measure how close the partition is to predetermined benchmark classes. Among the long list of indexes we used *purity* and *AMI (Adjusted Mutual Information)*, the latter again chosen for being able of comparing our results with the ones in [1].

Purity [27] measures to what extent clusters contain a single class of the predefined benchmark. Consider F the set of classes and f_i the i-th class; then purity index is defined by the following formula:

$$P(K, F) = \frac{1}{N} \sum_{i=1}^{|K|} \max_{j=1,\dots,|F|} |c_i \cap f_j|.$$

In practice, for each cluster, it computes the number of elements belonging to the most prominent intersection with the benchmark classes. Then it takes the sum over all clusters and divides for the number of data points. Thus it computes the fraction of elements that are correctly grouped according to the predetermined classes. The index assumes values in the interval $[0, 1]$ and does not penalize having a higher number of clusters.

The second external index is *Adjsuted Mutual Information* [43, 44]. This is an information theoretic measure used to quantify how much two partitions are in accordance with each other by computing their mutual information. Consider now two labelings K and F. The first corresponds to the cluster partition produced by the algorithm, while the second is related to specific features predefined on data. Then mutual information between the two partitions is again defined as:

$$I(K; F) = \sum_{i=1}^{|K|} \sum_{j=1}^{|F|} P(i, j) log_2\Big(\frac{P(i, j)}{P(i)P(j)}\Big), \tag{4.3}$$

which can be normalized into the interval $[0, 1]$ generating the cross-entropy, also called *NMI (Normalized Mutual Information)*:

$$NMI(K; F) = \frac{I(K; F)}{\sqrt{H(K)H(F)}}, \tag{4.4}$$

where $H(K)$ and $H(F)$ are, respectively, the entropy of clustering and feature labelings, as defined in the equation 3.4. However, there is no guarantee that NMI converges to 0. Moreover, it would be biased in favor of bigger cluster number. To address this problem, AMI was proposed:

$$AMI(K; F) = \frac{I(K; F) - \mathbb{E}(I(K; F))}{\max(H(K), H(F)) - \mathbb{E}(I(K; F)}, \tag{4.5}$$

which assumes value into the interval $[0, 1]$ too. In this work, purity and AMI were used to verify whether specific features guided cluster building. Thus K corresponds to partition produced by the algorithm, while F corresponds to feature binnings.

The two indexes are different and, for that reason, can be used as complementary information to understand clusters composition. However, they strongly depend on the feature binning. Both the number of bins and how such bins are defined influence the measures. Purity index is the most sensitive. Consider a clustering schema defined on N data points and composed of K clusters. Moreover, consider a feature such that each bin is equally populated. As a consequence, fewer bins implies that each bunch contains more stimuli and thus becomes broader. As bins become more extensive, the most significant intersection between the cluster and all bins probably becomes bigger. Thus purity value is higher. Such behavior is evident when partition consists of a single cluster. In that case, the most significant intersection between the unique cluster and each of the b feature bins is $\frac{N}{b}$ because bins are equally populated. Thus the index becomes $\frac{1}{b}$, which assumes higher values when b is small. Hence, feature with less bins will have higher purity values. Consider now many clustering schemas in which the number of clusters varies, and a single feature is present. When there are many clusters, each bunch is smaller, and thus probability stimuli of a single feature bin compose it is higher. Consider the extreme case in which each stimulus makes a cluster by itself. In that case, the most significant intersection between the cluster and all bins is always 1. Thus the index is 1 too: many clusters implies higher values.

Consequently, purity index should not be compared against features with different bin number. Moreover, it inevitably presents decreasing trend as the number of clusters decreases. However, purity index of the same feature but on different clustering schemas is handy. Indeed, it is useful to compare purity values of the partitions with $K > 1$ with purity index of the whole dataset. If the two schemas present similar values, it means that partitioning data into many clusters do not bring benefit on discriminating such feature. Indeed, if values are similar, it means that stimuli of all bins equally populate each cluster. Higher values mean that some clusters represent, forcefully or not, specific feature bins. Even if AMI is related to bins too, it does not happen as strongly as purity index. Moreover, whatever type of comparison is meaningful.

From now on the symbols $AMI^d_C(l, f, p)$ and $P^d_C(l, f, p)$ stand, respectively, for AMI and purity value of feature f for the clustering algorithm C computed using distance mesaure d and parameter p on level l. For example $AMI^e_D(V1, P, \sigma = 10)$ stands for the adjusted mutual information of position compared with clusters pro-

duced with dominant sets algorithm (D) and similarities computed using euclidean distances (e) with $\sigma = 10$ on level V1.

4.4.4 Interpretation of results

The last phase of cluster analysis consists of results interpretation, generally done by experts of the application area from which the dataset comes. Clustering schema should be further investigated and analyzed to be completely understood and to draw the right conclusions. In particular, the present work extensively studied how clusters are composed and which are the more encoded features.

4.5 Dominant sets theory

The following sections will introduce all theoretic details about the dominant sets framework [35], explaining links with game-theory, graph-theory and optimization theory.

4.5.1 Definitions

Let's now formalize the notion of dominant set. Many approaches are available for data representation, among which graphs emerge as one of the most effective. Let be $G = (V, E, \omega)$ a directed and weighted graph where $V = \{1, ..., N\}$ is the set of vertices, $E \subseteq V \times V$ the set of directed edges, and $\omega : E \to \mathbb{R}$ the weighting function of arcs. The set of vertices corresponds to dataset objects, while edges and weights express, respectively, relationship presence and magnitude among objects. Consider a similarity matrix A such that $A_{ij} = \omega(i, j)$, with $i, j \in V$. Notice that previous definitions do not impose a specific structure over the similarity matrix A, which could be even symmetric or asymmetric. Let $C \subseteq V$ be a subset of vertices such that $C \neq \emptyset$. Then for each vertex $i \in V$ the *average weighted in-degree* with respect to C is:

$$\text{awindeg}_C(i) = \frac{1}{|C|} \sum_{j \in C} A_{ij}, \tag{4.6}$$

where $|C|$ is the cardinality of C. Then, the *relative similarity* between objects $i \in V$ and $j \in C$, with respect to the average similarity of j with members of C, is given by the following formula:

$$\phi_C(i, j) = A_{ij} - \text{awindeg}_C(j). \tag{4.7}$$

Notice that $\phi_C(i,j) > 0$ if the similarity between i and j is higher than the average similarity of j with members of C. Vice-versa when $\phi_C(i,j) < 0$. Let's now define which is the *weight* of i with respect to the whole set C:

$$W_C(i) = \begin{cases} 1 & \text{if } |C| = 1 \\ \sum_{j \in C \setminus \{i\}} \phi_{C \setminus \{i\}}(i,j) W_{C \setminus \{i\}}(j) & \text{otherwise.} \end{cases} \tag{4.8}$$

In few words, $W_C(i)$ is a measure of coupling between i and objects in $C \setminus \{i\}$ considering the total mutual similarity among members of $C \setminus \{i\}$. If $W_C(i) > 0$ then i is highly similar to $C \setminus \{i\}$, thus should be a member of C. The *total weight* of C is then given by:

$$W(C) = \sum_{i \in C} W_C(i). \tag{4.9}$$

Eventually, a dominant set is a non-empty subset of objects $C \subseteq V$ such that $W(T) > 0$ for all $T \subseteq C$, $T \neq \emptyset$, and: a) $W_C(i) > 0$, for all $i \in C$; b) $W_{C \cup \{i\}}(i) < 0$ for all $i \notin C$. This definition coincides with the one of cluster. Internal homogeneity is ensured by point a). Indeed, dominant set members must have positive weight with respect to C. Point b) ensures maximality because no other object should belong to C. Notably, in the case of binary symmetric affinities, or better when the graph $G = (V, E)$ is both undirected and unweighted, a dominant set exactly corresponds with the notion of maximal clique. This suggests that dominant set generalizes the notion of maximal clique in weighted graphs (see [35] for more in-depth details).

4.5.2 A game-theoretic perspective of dominant sets

As previously mentioned, dominant sets are intrinsically liked with game theory, and specifically with evolutionary game theory [17, 45]. Indeed, clustering could be formulated as an evolutionary non-cooperative game, the so-called *clustering game* [35].

Evolutionary game theory was born in the 1970s to model behavior evolution of animal conflicts by proposing an idealized biological context. Consider an ideally infinite population of individuals, belonging to the same species, which compete for a limited resource. The conflict between two individuals is modeled as a symmetric, non-cooperative, two-player game. Individuals do not have complete knowledge about the game and act in a pre-programmed way, or better they follow a specific pure strategy that is transmitted among generations. The set of pure strategies corresponds with the set $V = \{1, ..., N\}$ of objects to be clustered. The game

consists of two players, each one simultaneously picking one pure strategy. When both have chosen the strategy, the player receives a revenue according to how much its object is similar to the one selected by the adversary. Let the similarity matrix A be the payoff matrix describing such revenues. If the two players have picked, respectively, strategies i and j, then the former receives A_{ij}, while the latter A_{ji}. Obviously, each player aims to select an object as much similar as possible to the one selected by the other. Keep in mind that self similarities are intuitively high, thus it could happen that all players coordinate to always pick the same element. To simply overcome this problem, the diagonal of the matrix A should be set to 0, or a sufficiently low value.

Suppose now that the game is played many and many times, generation after generation, by two randomly selected individuals of the population. At a specific instant, supposing x_i the fraction of population playing the pure strategy i, overall population state is $\mathbf{x} = (x_1, ..., x_N) \in \Delta$, being:

$$\Delta = \left\{ \mathbf{x} \in \mathbb{R}^N : \sum_{j \in V} x_j = 1 \text{ and } x_j \geq 0, \text{ for all } j \in V \right\}. \tag{4.10}$$

This state reflects the general and shared behavior of the population. Practically, it is a probability distribution over the set of pure strategies which models the identical stochastic playing strategy of all players, formally called mixed-strategy. The set of mixed-strategies of the entire population is called mixed-strategy profile (in that case composed by identical mixed-strategies).

The population of individuals that select weakly supported strategies will gradually disappear. Vice-versa, individuals selecting strongly supported strategies will survive. Conflicts are repeated again and again until an equilibrium state is reached. Ideally, such equilibrium should be such that all individuals receive the same average payoff and no player has the convenience to change its strategy. The first definition of equilibrium in game theory is the so-called Nash equilibrium. It is a mixed-strategy profile such that it is the best reply to itself. Since in the evolutionary game described above all individuals play the same strategy, let's define what a symmetric Nash equilibrium is. Let be $u(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T A \mathbf{y}$ the expected payoff of strategy $\mathbf{x}$ against $\mathbf{y}$. Then, $\mathbf{x} \in \Delta$ is a symmetric Nash equilibrium if every strategy $\mathbf{y} \in \Delta$ gets a lower average payoff then the population expected payoff, namely, $u(\mathbf{y}, \mathbf{x}) \leq u(\mathbf{x}, \mathbf{x})$ for all $\mathbf{y} \in \Delta$. In the clustering game, this happens when both players agree on the same partition, and there is no reason to change something. The good news is that every finite normal-form symmetric game present at least one symmetric Nash equilibrium, guaranteeing that a solution exists. However,

this condition only ensures internal homogeneity and not maximality. Indeed, if there exists $\mathbf{y} \in \Delta$ such that $u(\mathbf{y}, \mathbf{x}) = u(\mathbf{x}, \mathbf{x})$ then also $\mathbf{y}$ could be chosen as the equilibrium determining the clustering partition. For such reason, it is necessary to introduce a refinement of the Nash Equilibrium: the so-called Evolutionary Stable Strategy. In few words, a strategy is called evolutionary stable if it is resistant to invasion by new strategies. Formally, a mixed strategy $\mathbf{x} \in \Delta$ is a ESS if it is a symmetric Nash Equilibrium such that $u(\mathbf{y}, \mathbf{x}) = u(\mathbf{x}, \mathbf{x}) \implies u(\mathbf{y}, \mathbf{y}) < u(\mathbf{x}, \mathbf{y})$ for all $\mathbf{y} \in \Delta \setminus \{\mathbf{x}\}$. Thus, playing $\mathbf{x}$ always remains the best choice. This second condition introduces maximality. As a consequence, dominant sets are in one-to-one correspondence with ESSs of a clustering game with affinity matrix A [42, 32]. Once the ESS has been found, it is translated to a dominant set. The probability assigned to pure strategy i is the probability the object i belongs to the dominant set. Thus, the support of $\mathbf{x}$, being $\sigma(\mathbf{x}) = \{i \in V : x_i > 0\}$, corresponds to a dominant set of the graph the clustering game is based. See [35] for theorems, demonstrations and other technical details.

4.5.3 On evolutionary dynamics for dominant set extraction

The previous section only discusses theoretic details about dominant sets, without showing how clustering is effectively performed. The algorithm consists of the sequential extraction of dominant sets. Once a dominant set C is extrapolated, the relative graph is made smaller by removing all vertices of C and arcs which have a node of the dominant set as extreme. Then another dominant set is extracted and so on, until no vertex remains and a hard partition is produced. Thus, the core of the algorithm is how a single dominant set is extracted. One possible solution is to enumerate all subsets of V and verify if they respect dominant set definition. Obviously, this solution is not feasible for big datasets. A second solution is to find ESSs of the corresponding clustering game, being each ESS in one-to-one correspondence with a dominant set. Many evolutionary dynamics were proposed to overcome the problem of finding ESS. *Replicator Dynamics (RDs)* are one of the most famous family of such dynamics. RDs aim to replicate population evolution by using ordinary differential equations. Consider again the mixed-strategy $\mathbf{x} \in \Delta$ representing the fractions of the population playing a certain strategy at the current time. Consider then $u(\mathbf{e}^i, \mathbf{x}) = (\mathbf{e}^i)^T A\mathbf{x} = (A\mathbf{x})_i$ the expected average payoff of playing the pure strategy i against the whole population playing $\mathbf{x}$. Furthermore, let $u(\mathbf{x}, \mathbf{x}) = \mathbf{x}^T A\mathbf{x}$ be the population expected payoff. The continuous-time replicator equations proposed in [45] was:

$$\frac{\dot{\mathbf{x}}_i}{\mathbf{x}_i} = u(\mathbf{e}^i, \mathbf{x}) - u(\mathbf{x}, \mathbf{x}) = (A\mathbf{x})_i - \mathbf{x}^T A\mathbf{x}, \tag{4.11}$$

for all $i = 1, .., N$. The average rate of increase $\dot{\mathbf{x}}_i/\mathbf{x}_i$ is proportional to how much it is convenient to play the pure strategy i against the mixed strategy $\mathbf{x}$. It equals the difference between the expected payoff of pure strategy i against the whole population strategy $\mathbf{x}$ and the average payoff of $\mathbf{x}$ against itself. Basically, the fraction of population playing pure strategy i should increase if the latter is convenient considering the whole population behavior, otherwise, it should decrease. The equation 4.11 can be discretized in the following way [45]:

$$x_i^{(t+1)} = x_i^{(t)} \frac{(A\mathbf{x}^{(t)})_i}{(\mathbf{x}^{(t)})^T A\mathbf{x}^{(t)}}. \tag{4.12}$$

Continuous-time replicator dynamic is always simplex-invariant, while the discrete formulation only when A is non-negative. Thus, by starting from the center of the standard-simplex, the equation is used to replicate population evolution. Ideally, the replication should continue until no individual has the convenience of changing its strategy, which happens when for all $i = 1, ..., N$ $u(\mathbf{e}^i, \mathbf{x}^{(t)}) = u(\mathbf{x}^{(t)}, \mathbf{x}^{(t)})$. This situation reflects stability properties of the above described dynamic. Indeed, let $\mathbf{x} \in \Delta$ be an ESS, then it is an asymptotically stable equilibrium point for the replicator dynamics [17, 45]. Unfortunately, the opposite does not hold. It might happen that an asymptotically stable point does not correspond to an ESS. However, if the game is doubly-symmetric ($A = A^T$), then the average population payoff $u(\mathbf{x}) = u(\mathbf{x}, \mathbf{x}) = \mathbf{x}^T A\mathbf{x}$ is strictly increasing along any non-constant trajectory of replicator dynamics. Practically, $du(\mathbf{x}^{(t)})/dt \geq 0$ for all $t \geq 0$, with equality only when $\mathbf{x}^{(t)}$ is a stationary point [17].

This explains dominant sets relation with optimization theory. Indeed, again when $A = A^T$, the previous statement could be formalized as a maximization problem over $u(\mathbf{x}, \mathbf{x})$, also called Standard Quadratic Problem, StQP [5]:

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & u(\mathbf{x}) = \mathbf{x}^T A\mathbf{x} \\ \text{subject to} \quad & \mathbf{x} \in \Delta \subset \mathbb{R}^N. \end{aligned} \tag{4.13}$$

Thus, ESSs of a doubly-symmetric game are in one-to-one correspondence to strict local solutions of the StQP.

Let's try to summarize. Dominant sets are in one-to-one correspondence ESSs of a clustering game with generic payoff matrix A. However, when the game is also doubly-symmetric, ESSs are in one-to-one correspondence with strict local maximizers of the StQP 4.13 and with asymptotically stable points under the

replicator dynamics [17, 45]. As a consequence, dominant sets of the graph with similarity matrix $A = A^T$ are in one-to-one correspondence with asymptotically stable points of RDs. This explains the complex but elegant relations among dominant sets, evolutionary game theory and their dynamics, and optimization theory.

Even if RDs look like the best way to find ESSs, keep in mind that a single step of the dynamic has both quadratic spatial and time complexity. For that reason, such dynamics might get again unfeasible for huge datasets with billions and billions of objects. *Infection and Immunization Dynamics (nImDyn)* [33] were proposed to overcome this problem. They are discrete-time dynamics in which population update is performed linearly, both in terms of space and time. Let be $\mathbf{x}$ the incumbent population state, while $\mathbf{y}$ the mutant population invading $\mathbf{x}$. The strategy $\mathbf{y}$ is said to be infective for $\mathbf{x}$ if the expected payoff of it against $\mathbf{x}$ is greater than the average payoff of the incumbent strategy. Let $\Upsilon(\mathbf{x}) = \{\mathbf{y} \in \Delta : \mathbf{y}^T A\mathbf{x} > \mathbf{x}^T A\mathbf{x}\}$ be the set of infective strategy of $\mathbf{x}$. Suppose that $\mathbf{y}$-strategists, with $\mathbf{y} \in \Upsilon(\mathbf{x})$, invade the incumbent population $\mathbf{x}$. Then, let $\delta_{\mathbf{y}}(\mathbf{x})$ be the fraction of the post-entry population playing $\mathbf{y}$, which is defined as:

$$\delta_{\mathbf{y}}(\mathbf{x}) = \begin{cases} \min\left\{\frac{(\mathbf{x}-\mathbf{y})^T A\mathbf{x}}{(\mathbf{y}-\mathbf{x})^T A(\mathbf{y}-\mathbf{x})}, 1\right\} & \text{if } (\mathbf{y}-\mathbf{x})^T A(\mathbf{y}-\mathbf{x}) < 0 \\ 1, & \text{otherwise.} \end{cases} \tag{4.14}$$

The following formula defines the discrete replicator equation:

$$\mathbf{x}^{(t+1)} = \delta_{\mathcal{S}(\mathbf{x}^{(t)})}(\mathbf{x}^{(t)})[\mathcal{S}(\mathbf{x}^{(t)}) - \mathbf{x}^{(t)}] + \mathbf{x}^{(t)}, \tag{4.15}$$

where the function $\mathcal{S} : \Delta \to \Delta$ is a generic *strategy selection* function which returns an infective strategy for $\mathbf{x}$, if it exists:

$$\mathcal{S}(\mathbf{x}) = \begin{cases} \mathbf{y} & \text{for some } \mathbf{y} \in \Upsilon(\mathbf{x}) \text{ if } \Upsilon(\mathbf{x}) \neq \emptyset \\ \mathbf{x} & \text{otherwise.} \end{cases} \tag{4.16}$$

These dynamics are run until there exists an infective strategy $\mathbf{y} \in \Upsilon(\mathbf{x})$ which make $\mathbf{x}$ violating the Nash condition [33]. As in the case of RDs, if the game is doubly-symmetric, the average payoff is strictly increasing along any non-constant trajectory of InImDyn [36]. This holds independently to how $\mathcal{S}(\mathbf{x})$ is defined, but its formulation may strongly influence the algorithm efficiency. A possible well-defined strategy selection function consists in allowing only pure strategies to be selected as infective [35]. Let be $u \in \arg\max_{j \in V}(A\mathbf{x})_j$ one of the best responses to $\mathbf{x}$, while $v \in \arg\max_{j \in \sigma(\mathbf{x})}(A\mathbf{x})_j$ one of the worst response to $\mathbf{x}$ in its support. The infective

strategy for $\mathbf{x}$ might be either the best response $\mathbf{z}^+ = \mathbf{e}^u$, or a new population state in which the worst response is removed $\mathbf{z}^- = \frac{\mathbf{x}_v}{1-\mathbf{x}_v}(\mathbf{x} - \mathbf{e}^v) + \mathbf{x}$, or $\mathbf{x}$ itself:

$$\mathcal{S}_{\text{Pure}}(\mathbf{x}) = \begin{cases} \mathbf{z}^+ & \text{if } (\mathbf{z}^+)^T A\mathbf{x} > (\mathbf{z}^-)^T A\mathbf{x} \\ \mathbf{z}^- & \text{if } (\mathbf{z}^+)^T A\mathbf{x} < (\mathbf{z}^-)^T A\mathbf{x} \\ \mathbf{x} & \text{otherwise.} \end{cases} \tag{4.17}$$

Again, dominant sets are in one-to-one correspondence with ESSs of the doubly-symmetric clustering game, which are, in turn, in one-to-one correspondence with asymptotically stable points of InImDyn [36].

In this thesis, we used such InImDyns for sequentially extracting dominant sets from the graph generated from data points similarities and creating hard partitions.

4.6 Kmeans theory

K-means [4, 20] is probably the most famous algorithm used to identify clusters of data points in a multidimensional space. The section briefely recap its well-known theory. Let's consider a data set $V = \{1, ..., N\}$, where N is the number of observations and $\mathbf{x}_i$ is the i-th objects belonging to a P-dimensional space. K-means algorithm aims to partition these points into a fixed number K clusters. A cluster will contain points whose inter-point distances are small and distances to points outside are high, if compared with the former. Each of these K clusters is represented by a P-dimensional vector being the prototype/centroid. Let $\boldsymbol{\mu}_k$ be the prototype of the k-th cluster, with $k = 1, ..., K$. Once centroids are found, the algorithm assigns each point to its closest cluster. Let $r_{ik} \in \{0, 1\}$, for $k = 1, ..., K$, be a binary variable describing object assignment to cluster k. Consider that $r_{ik} = 1$ if and only if $\mathbf{x}_i$ is assigned to cluster k. The goal of K-means is to find $\{\boldsymbol{\mu}_k\}$ and $\{r_{ik}\}$ such that the sum of the squared distances of each data point to its closest vector $\boldsymbol{\mu}_k$ is minimized. This objective function is called *distortion measure*, defined as:

$$J = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} ||\mathbf{x}_i - \boldsymbol{\mu}_k||^2. \tag{4.18}$$

The minimization of J is perfomed as an iterative process. During the initialization step, centroids $\boldsymbol{\mu}_k$ are randomly chosen on data points space. Then each iteration consists two phases:

1. **Minimize J according to r_{ik} keeping $\boldsymbol{\mu}_k$ fixed:** J is a linear function of r_{ik}, thus it is easy to get a closer solution; since all terms involving one data

point are independent among them, each internal summatory is independently minimized; for all $i \in V$ simply choose the cluster k such that $||\mathbf{x}_i - \boldsymbol{\mu}_k||^2$ is minimized and then set $r_{ik} = 1$:

$$r_{ik} = \begin{cases} 1 & \text{if } k = \arg\min_j ||\mathbf{x}_i - \boldsymbol{\mu}_j||^2 \\ 0 & \text{otherwise,} \end{cases} \tag{4.19}$$

in few words, i-th element is assigned to the closest cluster $\boldsymbol{\mu}_k$.

2. **Minimize J according to $\boldsymbol{\mu}_k$ keeping r_{ik} fixed:** this time J is a quadratic function of $\boldsymbol{\mu}_k$; thus, minimize it by computing its first derivative with respect to $\boldsymbol{\mu}_k$ and put it to zero; then the following formula allows to compute the new centroid:

$$\boldsymbol{\mu}_k = \frac{\sum_i r_{ik}\mathbf{x}_i}{\sum_i r_{ik}}. \tag{4.20}$$

Substantially the new centroid $\boldsymbol{\mu}_k$ is the mean of all data points assigned to the k-th cluster.

K-means is a limit case of the EM algorithm, in which the first phase corresponds to the expectation step E and the second to the maximization step M. E and M steps are repeated until convergence or a maximum number of iterations is reached. Since each step reduces the value of J, convergence is ensured, but J might converge to a local minimum instead of a global one, according to the initial centroids.

K-means algorithm was originally based on the Euclidean distance as dissimilarity measure, bringing not only problems related to robustness against outliers but also limits related to type of data points we can use. For this reason the algorithm is generalized to use various dissimilarity measure $\mathcal{V}(\mathbf{x}, \mathbf{x}')$, obtaining this new equation for J:

$$\tilde{J} = \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik}\mathcal{V}(\mathbf{x}_i, \boldsymbol{\mu}_k). \tag{4.21}$$

Again E step assigns elements to their closest centroid according to measure $\mathcal{V}$. Instead, M step could be complicated, according to how $\mathcal{V}$ is defined. Since also correlation distance was used to cluster data in the present work, it is sufficient to define cluster prototype for such measure. Matlab implementation of kmeans[1], we used to perform the algorithm, define the correlation centroid as the component-wise

[1] https://it.mathworks.com/help/stats/kmeans.html

mean of the cluster points, after centering and normalizing those points to zero mean and unit standard deviation.

Chapter 5

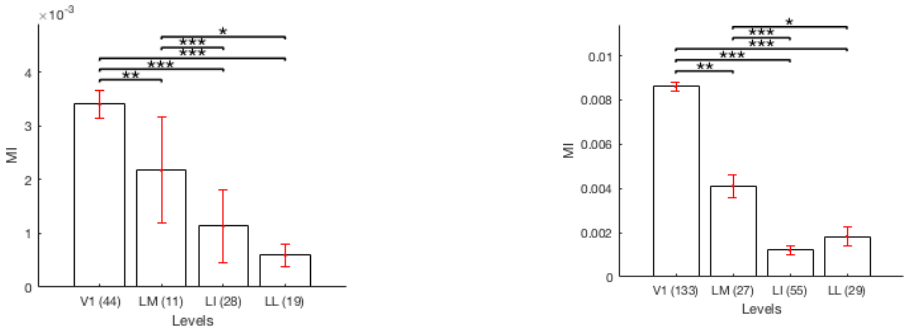
Results

This chapter discusses results obtained by the analysis on the four datasets corresponding to the entire hierarchy of rat's putative ventral stream. The first step consists of the single cell analysis, in which information theory measures are used to perform a first check on which features are encoded. Then the second step considers the whole neuronal response, by using both visual tools (like RDMs and t-SNE scatter plots) and two clustering approaches. Notice that all cluster analysis steps are reported only for the first area of the V1-LM-LI-LL progression. We skip the complete and exhaustive analysis of the other areas to avoid unuseful and repetitive commentary about parameters tuning. The analysis of the other levels is reported in Appendix A, related to dominant sets algorithm, and B, related to kmeans algorithm. Nevertheless, we report results of produced clusters for all area and, more usefully, discuss which is the behavior across the V1-LM-LI-LL hierarchy.

5.1 Information theoretic results

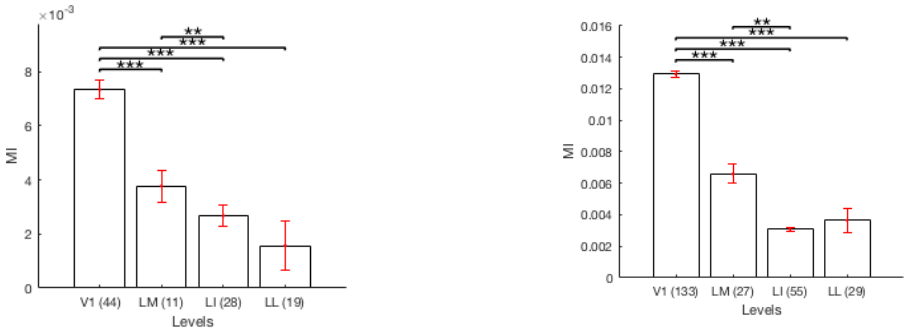
Considering the relation between neural responses and features is the first phase of the analysis. This step is useful to get a general view of data and to assess which features are majorly encoded. In that way, one could understand which and why some of them are subject to further analysis and the others not. Refer to [1] for more in-depth details.

Firstly concentrate on low-level features. Figure 5.1 shows the amount of conveyed information about the stimulus position by the neural responses. In particular Figures 5.1a and 5.1b respectively report mutual information conveyed by single and multi-units responses. Each bar represents the median of mutual information of the level neurons, while the red error bar represents the interval in which the median lies considering the standard error. Moreover, horizontal lines



(a) Single units **(b)** Multi units

Figure 5.1: Mutual information between neural responses and stimulus position

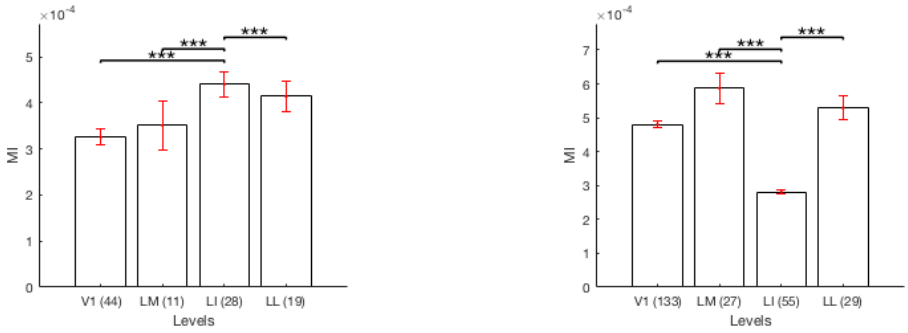


(a) Single units **(b)** Multi units

Figure 5.2: Mutual information between neural responses and stimulus total luminosity

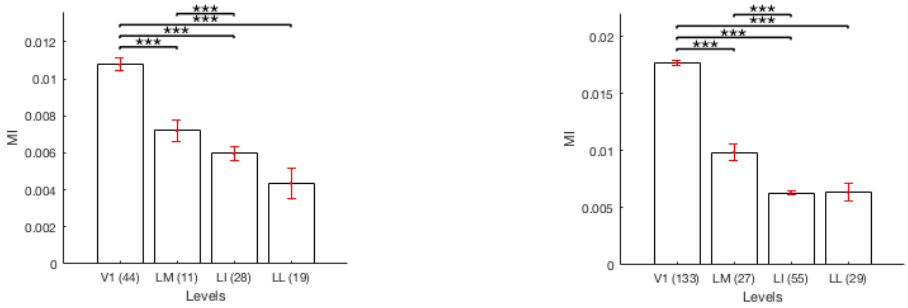
above the bars represent difference significance between two levels, as referred in Section 3.2. In particular, the number of asterisks indicates the p-value of the 1-tailed Mann-Whitney U test, Bonferroni-Holm corrected: * when $p < 0.05$, ** when $p < 0.01$ and *** when $p < 0.001$. A decreasing trend across the V1-LM-LI-LL progression is present on both unit types, meaning that position information is progressively discarded from lower to higher levels. Figure 5.2 shows mutual information related to total luminosity feature. Also, in that case, a sharp decreasing trend is reported, reflecting the same behavior of position stimulus and being more represented in the first levels of the V1-LM-LI-LL progression.

Consider now higher level features, in particular orientation (Figure 5.3) as mid-level feature and object identity (Figure 5.4) as high-level one [1]. Orientation does not present any trend across the hierarchy, both in case of well and not well-isolated units. Here it is omitted, but the same happened for features area, contrast, receptive field luminosity and aspect ratio. Surprisingly, object identity shows a decreasing trend across the V1-LM-LI-LL progression. Mutual information with that feature is computed considering all possible object views which have been



(a) Single units **(b)** Multi units

Figure 5.3: Mutual information between neural responses and stimulus orientation



(a) Single units **(b)** Multi units

Figure 5.4: Mutual information between neural responses and object identity. The computation considered all view transformations of the same object.

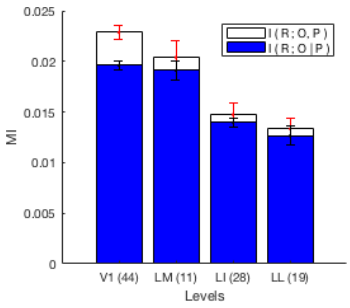
presented to rats. Thus it is a measure of how well single neurons can invariantly recognize objects. However, the reader has to consider that:

1. in feed-forward hierarchies the amount of information can never increase, but the V1-LM-LI-LL hierarchy is not strictly feed-forward, and thus the former statement does not hold [1];

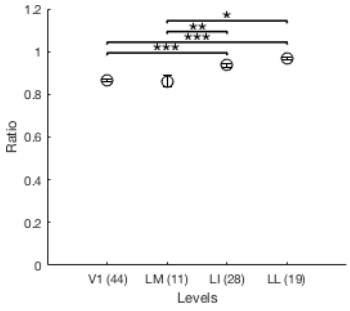
2. the number of recorded neurons decreases along the progression; thus the total amount of information might become smaller while a bigger portion of it is related to object identity [1];

3. some objects might be characterized by specific feature values which act as a confounding factor [1]; for example, one of the 40 presented objects is the plane (Figure 3.1), which is a white object; this implies very high values of luminosity; such high luminosity is present regardless of any other transformation.

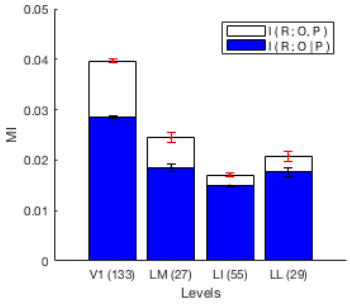
With the purpose of removing this last influence, conditional mutual information measure could be used to compute the information conveyed about the object



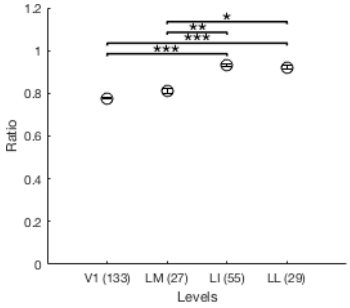
(a) Single units



(b) Ratio, single units

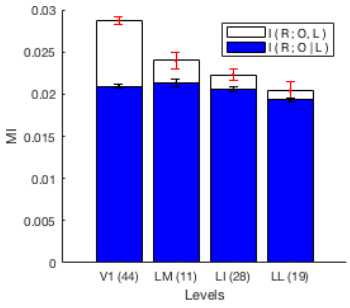


(c) Multi units

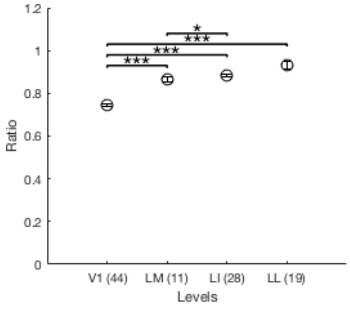


(d) Ratio, multi units

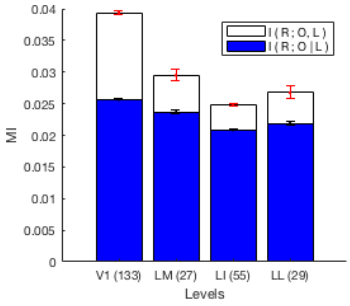
Figure 5.5: Conditional mutual information between neural responses and object identity fixing position. The computation considered all view transformations of the same object.



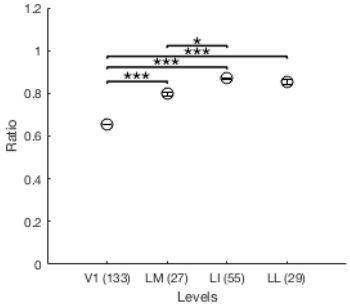
(a) Single units



(b) Ratio, single units



(c) Multi units



(d) Ratio, multi units

Figure 5.6: Conditional mutual information between neural responses and object identity fixing total luminosity. The computation considered all view transformations of the same object.

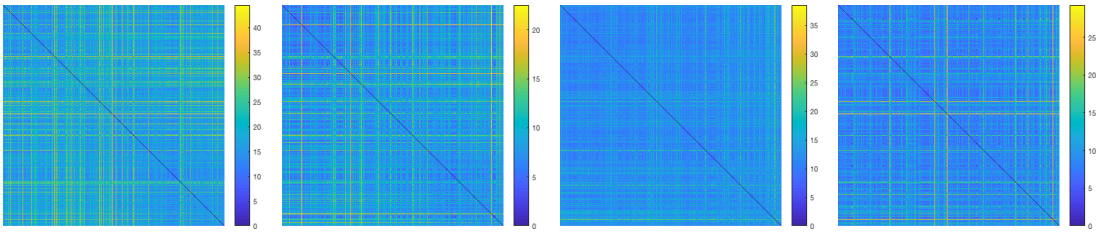
identity fixing other variables, the ones that could co-occur with it. Figures 5.5 and 5.6 illustrate the case in which position and total luminosity are the fixed variables, for both single and multi-units cases. In particular, blue bars on the left figures represent conditional mutual information, while white bars represent joint mutual information of object identity and the fixed feature. The right figures represent, for each level, the ratio between conditional and joint mutual information. The ratio is generally used because the amount of total information decreases along the progression. Thus it is useful to consider which is the fraction of conveyed information only related to object identity. Look at Figure 5.5a, related to the fixed feature position and single units. Conditional information decreases across the progression, but also the joint one decreases. Therefore information about the position is discarded until, in the last level, object identity is largely represented. The same happens for multi-units, Figures 5.5c and 5.5d. This means that, in the last level, object identity information is less polluted by position influence and thus becomes more explicit. The same happens in the case of total luminosity features.

5.2 Representational Dissimilarity Analysis

Representational Dissimilarity Matrices (RDMs) are useful tools to study data points distances visually. If appropriately ordered, then RDM could show block structures. In figures related to RMDs, colors vary from blue to yellow shades, representing small and significant distances among objects. In this thesis, both Euclidean and correlation distance metrics are considered. Thus it is a must to visualize RDMs for both measures. From now on the terms *euclidean RDM* and *correlation RDM* refer, respectively, to RDM of Euclidean and correlation distances.

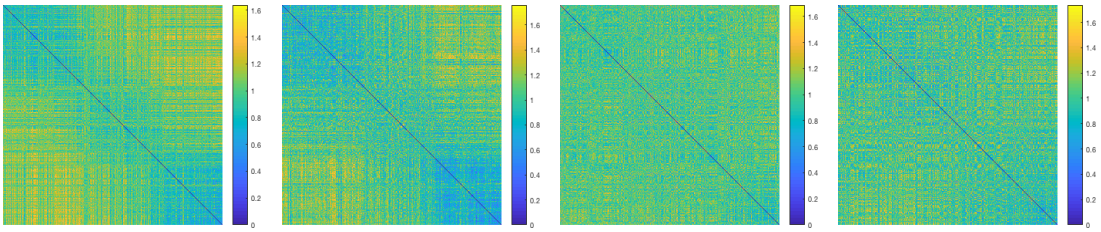
Figures 5.7 and 5.8 show RDMs for each level of the V1-LM-LI-LL progression in which rows and columns are ordered according to position feature. Specifically, Figure 5.7 regards euclidean distances, while Figure 5.8 regards correlation distances. Euclidean RDMs do not show any block structure, suggesting that no information about position is represented. The situation is different in the case of correlation RDMs. Figures 5.8a and 5.8b illustrate, respectively, RDMs related to V1 and LM. A three-block structure appears, more marked in the case of V1. Along the V1-LM-LI-LL progression the structure is lost, confirming that across the hierarchy the amount of information about position decreases (refer to Figure 5.1).

Total luminosity is the other feature that more weakly shows a specific block structure. Figures 5.9 and 5.10 show RDMs whose rows and columns are ordered according to total luminosity feature. Again RDMs for both Euclidean and correla-



(a) V1 (b) LM (c) LI (d) LL

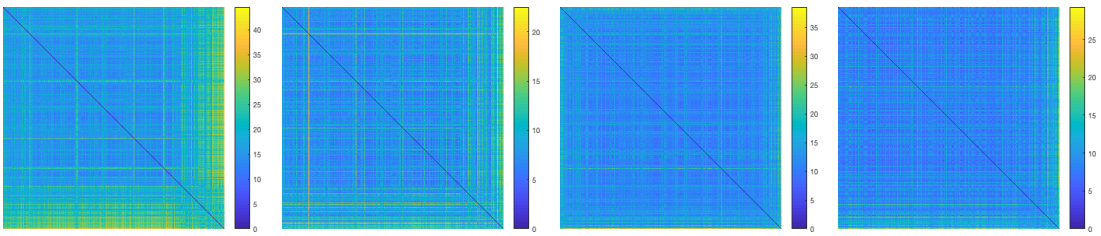
Figure 5.7: Euclidean RDMs ordered by position



(a) V1 (b) LM (c) LI (d) LL

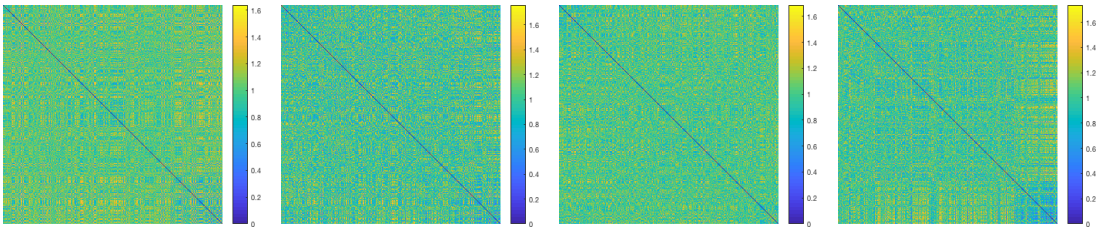
Figure 5.8: Correlation RDMs ordered by position

tion distances are presented, respectively in Figures 5.9 and 5.10. This time a better block structure emerges in euclidean RDMs. Figure 5.9a, related to V1, shows a particularly evident structure, with a small group composed of the brightest stimuli and another one made of stimuli with low and middle values of total luminosity. A similar analysis, with weaker emphasis, for level LM and LL. Unfortunately, no structure appears on level LI. Such structure does not appear with correlation distances, except for the last level of the V1-LM-LI-LL progression, where again the small group of the lightest stimuli becomes evident. These observations suggest that stimuli with high total luminosity are discriminated from all the others. This could be explained by the fact that rats are crepuscular animals, thus probably the high luminosity is a danger warning.



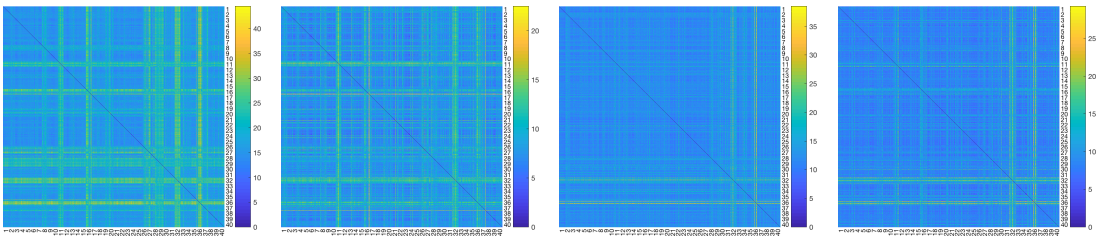
(a) V1 (b) LM (c) LI (d) LL

Figure 5.9: Euclidean RDMs ordered by total luminosity



(a) V1 **(b)** LM **(c)** LI **(d)** LL

Figure 5.10: Correlation RDMs ordered by total luminosity

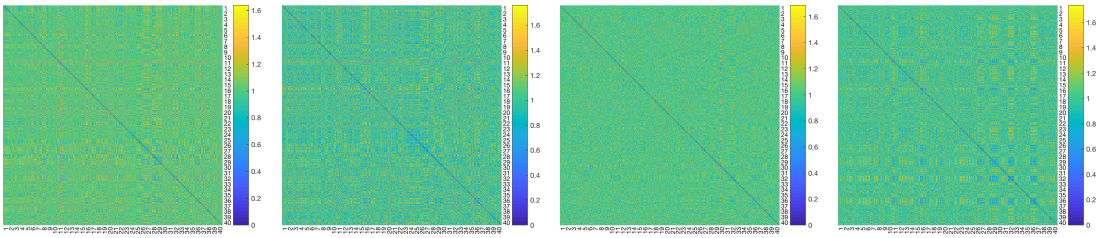


(a) V1 **(b)** LM **(c)** LI **(d)** LL

Figure 5.11: Euclidean RDMs ordered by object identity. The 36 different representations of the same object are grouped. We put labels of the corresponding object identifier on the left and on the bottom of the matrix

Consider now RDMs ordered according to object identifier, in Figures 5.11 and 5.12, respectively regarding euclidean and correlation distances. On the right and at the bottom of the matrices, we put the object identifier to better distinguish stimuli. In the correlation matrices some rows (and the respective columns) are distinctly colored with strong yellow shades, meaning that the corresponding objects have specific characteristics different from all the others. Some objects particularly show this trend and are those with identifiers 11, 16, 20, 27, 28, 29, 32 and 36. Figure 5.13 shows a representative view of each object. We immediately noticed that four of the eight objects remembered birds shape, while three could be seen as food by rats (banana, grapes, pumpkin).

However, they all share a common feature: high values of mean total luminosity over the 36 different views. Figure 5.14 reports mean total luminosity for each object identifier. Notice that objects of Figure 5.13 are exactly the ones marked with red labels. Such behavior is particularly evident in the first level V1 of the progression and becomes less marked in the others. Only two objects are strongly discriminated in each level: grapes and pumpkin, respectively objects 32 and 36. In Euclidean RMDs the structure appears absent, except for the last level LL. As in the correlation matrix, objects 11, 16, 32 and 36 assume lighter shades if compared with the overall matrix, with blue squared blocks on their intersections.



(a) V1 **(b)** LM **(c)** LI **(d)** LL

Figure 5.12: Correlation RDMs ordered by object identity. The 36 different representations of the same object are grouped. We put labels of the corresponding object identifier on the left and on the bottom of the matrix

(a) ID 11 **(b)** ID 16 **(c)** ID 20 **(d)** ID 27

(e) ID 28 **(f)** ID 29 **(g)** ID 32 **(h)** ID 36

Figure 5.13: Recognized objects from RDM ordered by object identifier

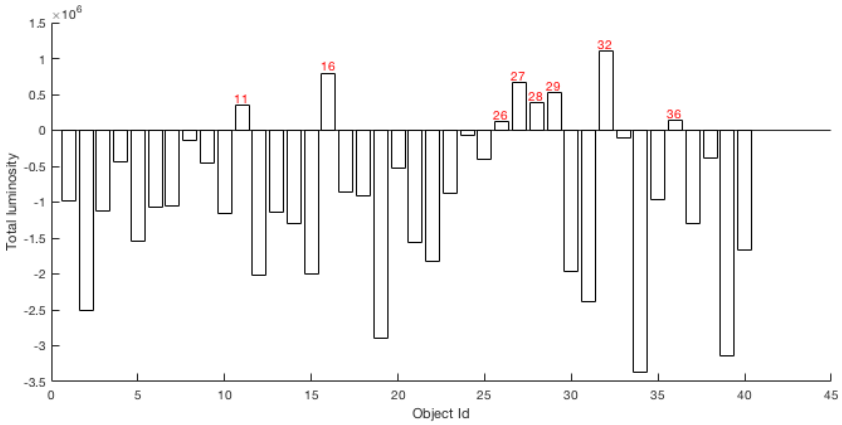


Figure 5.14: Mean object total luminosity

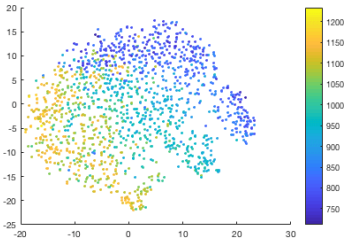
5.3 Data represented using t-SNE

Due to the high number of recorded neural responses, the datasets are characterized by a very high dimensionality. When dimensionality of data is huge, getting visual feedback of data could be difficult. In such cases dimensionality reduction techniques are necessary: we used t-SNE in that case. It allows visualizing high-dimensional data by building a two-dimensional map created using a certain distance metric.

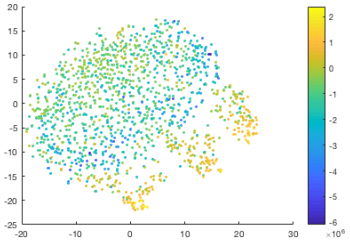
Figure 5.15 illustrates data points of levels V1, LM, LI and LL into the two-dimensional maps created using Euclidean distances. Figure 5.16 reports the same map but created using correlation distances. Especially, left pictures in Figures 5.15 and 5.16 show data points whose color shades represent the position of the stimulus, while right pictures in Figures 5.15 and 5.16 represent, in the same way, the stimulus total luminosity. Plots related to object identifier are difficult to understand since each stimulus is colored according to its object id. Indeed, there are 40 different colors whose shades do not represent a certain sorting criterion. For such reasons, here these figures are omitted.

Consider level V1, the one that has shown the higher relation with position and total luminosity up to now. Consider in particular Figures 5.15a and 5.16a, both regarding position but, respectively, related to euclidean and correlation distances. In the case of correlation distance three cohesively groups are present, one for each bin of position. This group structure is not so evident in the case of Euclidean distances, where boundaries are noiser. The same happens in a weaker way for level LM, where this time correlation distance map creates a structure of two groups, related to position left and position right. As expected, the same behavior is absent for levels LI and LL, in which information about position does not evidently and strongly emerge.

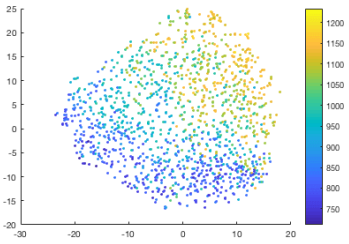
The contrary happens for total luminosity. Figure 5.15b, related to V1, shows three small groups composed of the brightest stimuli. Figure 5.16b is related to correlation distances and shows that each group mentioned above contains a portion of very light stimuli. However, a well-defined group of such stimuli is not present. The same happens for level LL, even if not with the same emphasis. Referring to information theory and RDMs results, LM is thought to report the same behavior. Unfortunately, this does not happen. Moreover, as expected, no exciting considerations should be pointed out about LI.



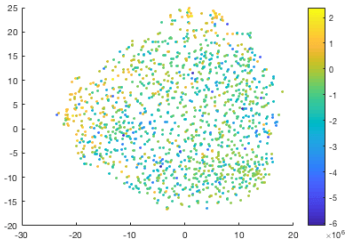
(a) V1, Position



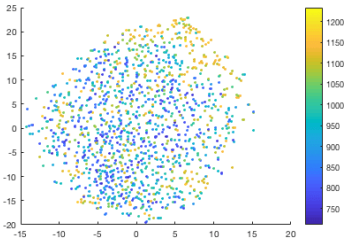
(b) V1, Tot. lum.



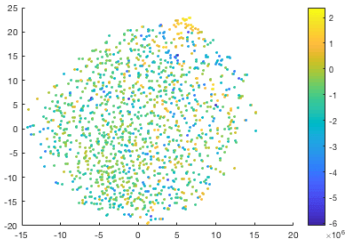
(c) LM, Position



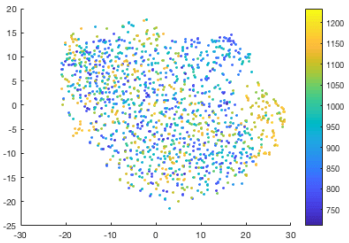
(d) LM, Tot. lum.



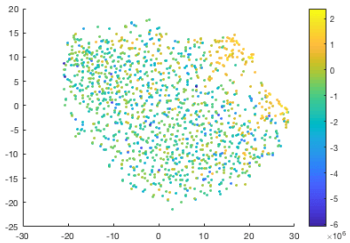
(e) LI, Position



(f) LI, Tot. lum.

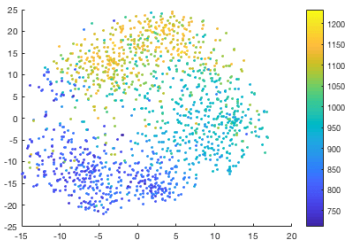


(g) LL, Position

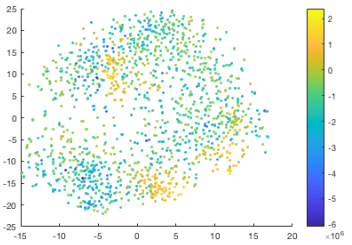


(h) LL, Tot. lum.

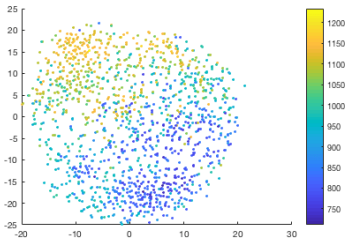
Figure 5.15: t-SNE two-dimensional maps created on Euclidean distances



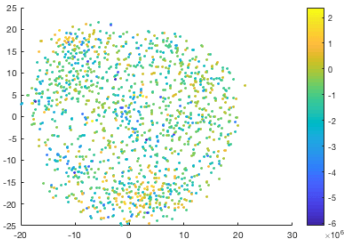
(a) V1, Position



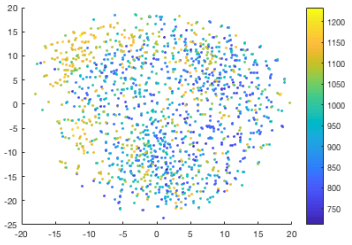
(b) V1, Tot. lum.



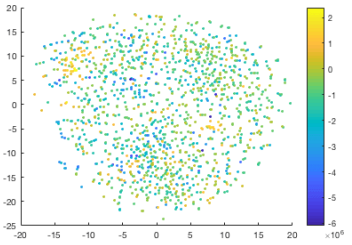
(c) LM, Position



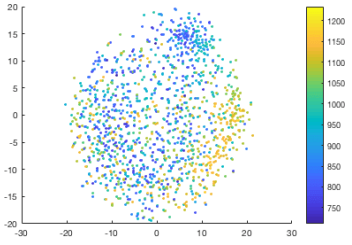
(d) LM, Tot. lum.



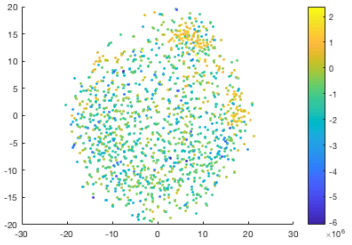
(e) LI, Position



(f) LI, Tot. lum.



(g) LL, Position



(h) LL, Tot. lum.

Figure 5.16: t-SNE two-dimensional maps created on 1-Pearson correlation distances

5.4 Dominant sets on V1

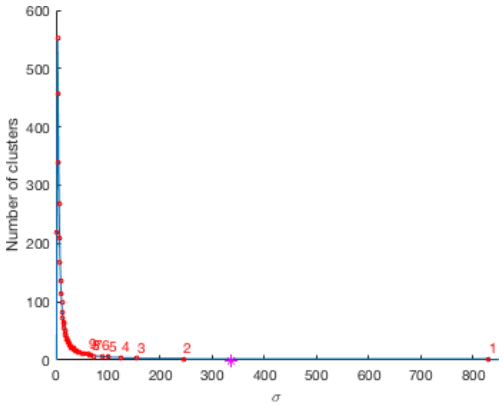
Dominant sets is a clustering algorithm that takes in input a similarity matrix and finds maximally coherent groups, the so-called clusters. Thus, what makes the difference is just the similarity matrix, which is used to determine both number of clusters and the partition. As reported in Section 4.4.2, the similarity matrix was built using a Gaussian kernel that takes in input distances among data points. In the analysis, the algorithm was run with two different kernels, one built using Euclidean distance and the other using correlation distance. Such kernel strongly depends on the parameter σ, that must be appropriately tuned to find the similarity matrix generating the best partition.

To tune that parameter, it is recommended to execute dominant sets for a specific range of σ and then select the correct value according to specific indicators, like the Silhouette and MST Dunn's indices. After that, external indexes (like the adjusted mutual information and purity) could be used to find which is the contribution of a particular feature on defining clusters. Once we selected the most appropriate value of σ, we analyzed clusters composition.

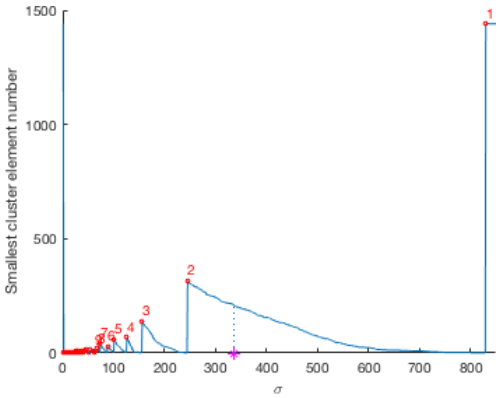
Mainly, the section refers to the first level of the V1-LM-LI-LL progression. We found that dominant sets extracted using the similarity matrix built with Euclidean distances express difference on stimulus total luminosity. Notably, one small cluster represents the brightest objects, while another one, bigger and noisy, represents middle and low luminosity values. Moreover, similarities built on correlation distance express stimulus position, suggesting one cluster for each position bin.

5.4.1 Results of Gaussian kernel on Euclidean distance

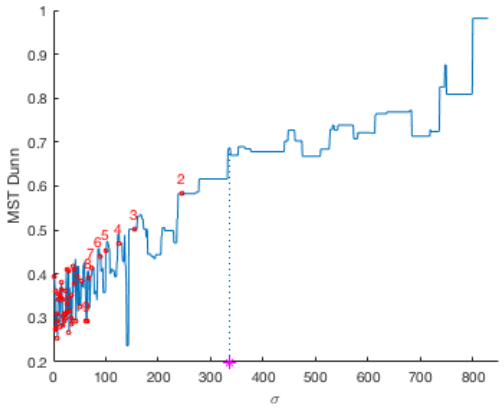
The present section discusses dominant sets results on similarity matrix created using the Gaussian kernel and the Euclidean distance. First of all, analyze what happens when σ varies. Figure 5.17a reports the number of clusters depending on σ. We immediately noticed that smaller values of σ act as a scattering factor on data points. In fact, the number of recognized clusters is very high when $\sigma \approx 0$, approximately 550. As σ increases, points are less scattered. Thus the number of clusters becomes smaller and smaller until all data points are clustered into the same set, which happens at $\sigma = 836$. Figure 5.17b shows the size of the smallest cluster as a function of σ. Each peak is marked with a red circle, which represents the smallest value of σ for which data is partitioned into the number of clusters the circle is labeled with. To facilitate reading, such markers are also reported in the other plots. After the peak, as σ increases, one of the clusters, generally the least



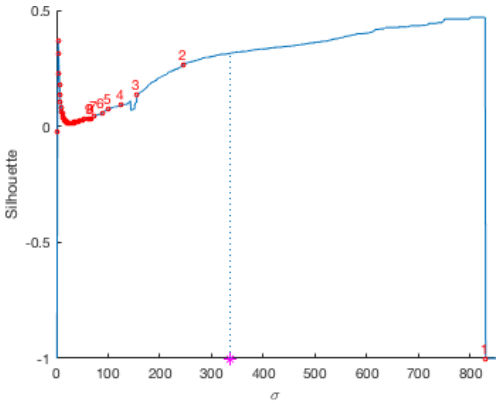
(a) Cluster number



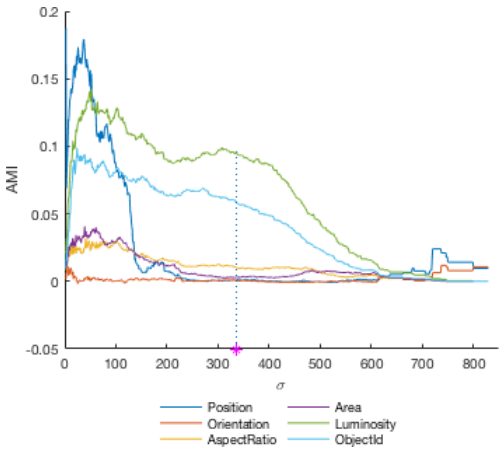
(b) Smallest cluster size



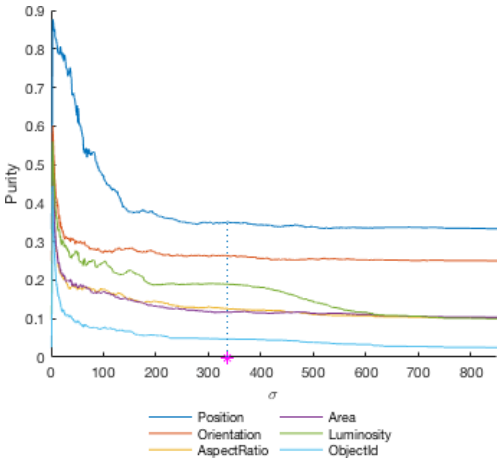
(c) MST Dunn's index



(d) Silhouette index



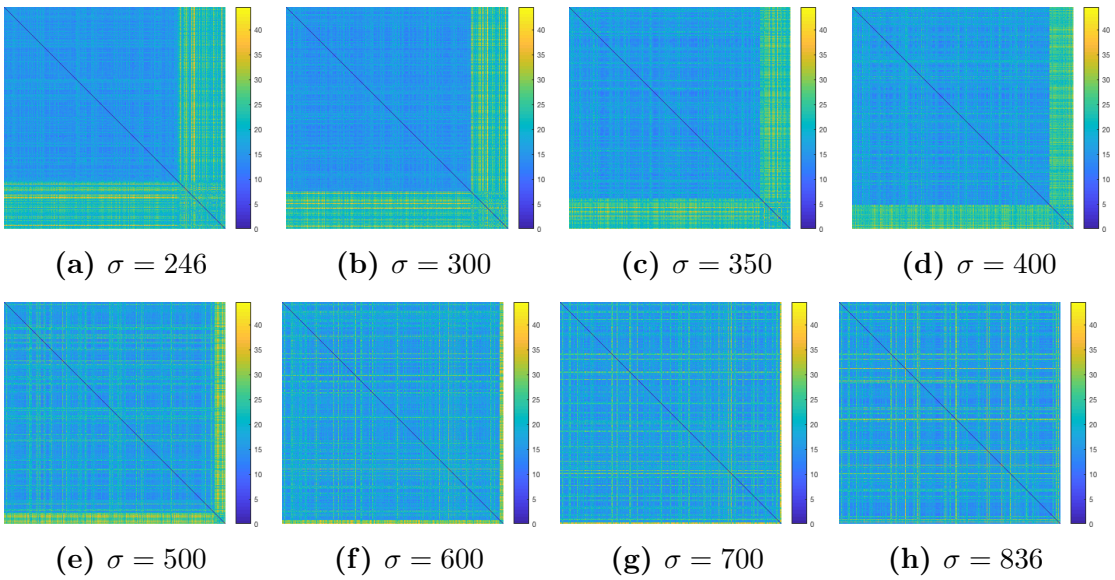
(e) AMI



(f) Purity

Figure 5.17: Statistics on dominant sets algorithm applied on V1 considering Gaussian kernel built using Euclidean distances

cohesive, is divided and incorporated in the other clusters until it disappears and a new peak is reached.



(a) $\sigma = 246$ **(b)** $\sigma = 300$ **(c)** $\sigma = 350$ **(d)** $\sigma = 400$

(e) $\sigma = 500$ **(f)** $\sigma = 600$ **(g)** $\sigma = 700$ **(h)** $\sigma = 836$

Figure 5.18: Euclidean RDMs of level V1 ordered according to dominant sets partitions computed on Gaussian kernel built using Euclidean distances and $246 \leq \sigma \leq 836$

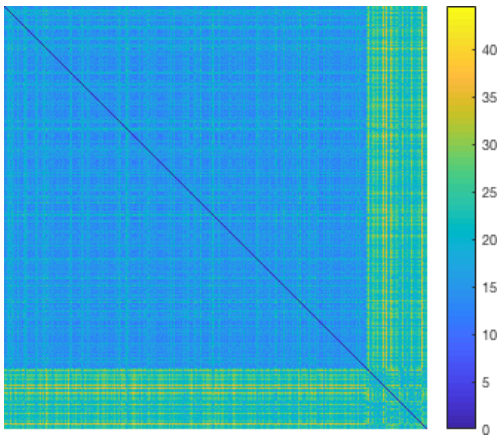
Let's now investigate the internal structure of clusters. Figures 5.17c and 5.17d respectively show the MST Dunn's and Silhouette indices. As explained in Section 4.4.3, the parameter has to be chosen in the proximity of the maximum value of both indexes. Notice that they assume higher values when there are two clusters, thus when $246 \leq \sigma \leq 836$, both reaching the maximum at $\sigma = 806$. However, such behavior looks strange: it is clear that the partition into two bunches of data is preferable, but indices suggest to group all elements into one cluster and leave a single stimulus separated. Other tools are necessary to understand that behavior properly and to choose σ finally.

Up to now only numerical analysis have been considered, thus let's change point of view and try to use visual tools like dissimilarity matrices. RDMs can be ordered by cluster index, produced by specific σ, and then by stimulus number. The evolution of RDMs varying on σ is useful to visually assess what happens to cluster intra and inter distances and to get a tip on choosing σ. Clustering schemas in which darker squared blocks are present on the diagonal, while the rest of the matrix is composed by brighter blocks, are the best. A dark block not on the diagonal suggests that extra cluster distances of the two corresponding clusters are low, meaning they are similar. Instead a light block on the diagonal means that intra clusters distances tend to be high and thus the cluster is not so cohesive.

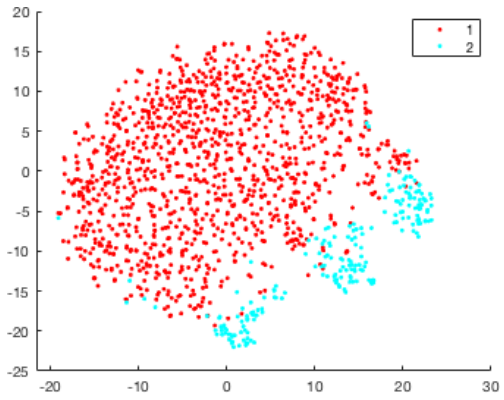
Consider the RDM related to the whole dataset in Figure 5.18h. Many light yellow rows and columns appear on the blue background. Each line corresponds to a stimulus that is particularly distant from all the others. This suggest that two different families of stimuli are present. The first family is composed by stimuli that are near to each other and distant from the ones of the other family. The second family is composed by stimuli that are distant from stimuli of both families. This consideration suggests that stimuli should be partitioned into two clusters, as already indicated by the two internal indexes. Consider now the evolution of RDM on $246 \leq \sigma \leq 836$ (see Figure 5.18). Matrices show the presence of two clusters: not surprisingly, the first is bigger, cohesive and well separated; the second is smaller, not particularly cohesive and not well separated from the first. As σ increase, the bigger cluster incorporates stimuli of the smaller, becoming less and less cohesive as each of the "different" stimulus is added. Hence, let's choose a value of σ such that these particular stimuli are grouped, which discreetly happens when $\sigma = 336$ (see Figure 5.19a), marked in Figure 5.17 with a pink asterisk.

Now external indices are considered to verify how such clusters are composed. Figures 5.17e and 5.17f respectively show AMI and purity. Many curves are present in the two plots, each one representing a specific feature (position, orientation, aspect ratio, area, total luminosity and object id). For small values of σ, AMI increases until it reaches a maximum, then starts to decrease. As suspected and explained in Section 4.4.3, purity values decreases as σ increases, because higher values of the parameter imply few clusters. When the clustering schema consists of a single cluster representing the whole dataset ($\sigma \geq 836$), AMI index cannot be computed and thus is not defined. Instead purity index becomes fixed reaching a threshold ($\approx \frac{1}{b}$ when the feature is binned into b bins, see Section 4.4.3). As already explained, values of both indices strongly depend on the number of bins a feature is partitioned to. In particular, purity index for feature with fewer bins has higher values, vice-versa for which has more bins. Look at $\sigma \geq 836$, in which data partition consists of a single cluster. Position, which is binned into three bins, is the feature with the highest purity index, followed by orientation, whose binnings consist of four bins. Then total luminosity, area, and aspect ratio purity indices assume a similar value because all are partitioned into ten bins. Eventually, the feature with the lowest value of purity index is object id, composed of forty bins.

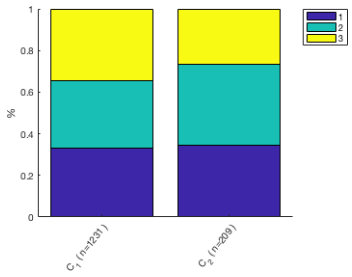
Consider now the current case. Total luminosity AMI values are higher if compared with other features (like position), reaching a local maximum approximately near $\sigma = 336$, with $AMI^e_D(V1, Tl, \sigma = 336) = 0.096$. As expected by previous analysis, object identity appears to follow the total luminosity curve, with $AMI^e_D(V1, Obj, \sigma = 336) = 0.0582$. Position AMI assumes higher values if com-



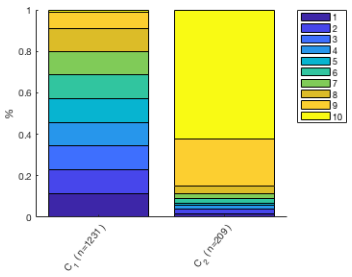
(a) RDM



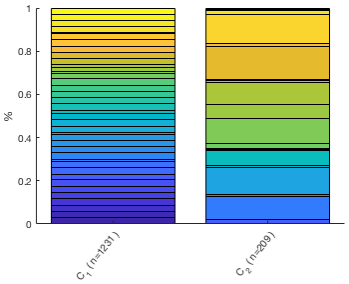
(b) t-SNE



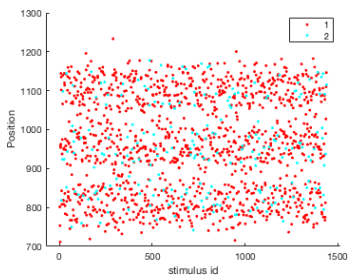
(c) Cluster vs position



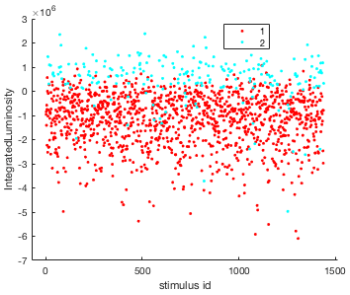
(d) Cluster vs total. lum.



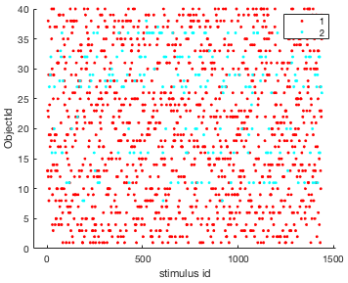
(e) Cluster vs object id



(f) Position vs cluster



(g) Total. lum. vs cluster



(h) Object id vs cluster

Figure 5.19: Clusters of V1 found by dominant sets on Gaussian kernel built using Euclidean distances and $\sigma = 336$

pared with total luminosity when $\sigma \leq 58$. However, these values of the parameter do not produce a good partition according to internal indices, thus should not be used. In order to compare results, position adjusted mutual information is $AMI^e_D(V1, P, \sigma = 336) = 0.0011$. Consider now the purity index. Position purity is always the highest. When $\sigma \geq 300$, it reaches a stable threshold that remains fixed until and after a single cluster is built, at $\sigma = 836$. This suggests that having a single or many clusters do not influence position purity, meaning that stimuli belonging to the three bins equally populate clusters. Total luminosity shows a different behavior. Before reaching the threshold related to the whole dataset, the index assumes higher values suggesting that the partition in more clusters better represents total luminosity bins then the entire dataset. For the sake of comparisons, consider that $P^e_D(V1, P, \sigma = 336) = 0.3507$, $P^e_D(V1, Tl, \sigma = 336) = 0.1903$ and $P^e_D(V1, Obj, \sigma = 336) = 0.0472$.

Let's deepen the analysis of clusters produced with $\sigma = 336$. Figure 5.19 illustrates different pictures that help understanding how clusters are composed. Figure 5.19a reports the euclidean RDM of data ordered according to clusters produced with $\sigma = 336$. The first cluster is bigger, more cohesive and better separated if compared with the second. Figure 5.19b shows the 2-dimensional t-SNE mapping produced with Euclidean distances. Red points represent the first cluster, while light blue points represent the second cluster. By comparing that picture with Figure 5.15b, it appears that light blue stimuli are exactly the ones with high total luminosity.

Let's investigate that statements by going deeper into cluster composition. Figure 5.19c illustrates cluster composition according to feature position. Each bar represents a cluster, while each colored piece of a bar represents the percentage of cluster elements belonging to the feature bin the color refers to. Since both clusters are composed by the same fraction of left, center and right elements, the position was not supposed to be a discriminator factor for cluster building. This confirms previous observations done on position purity index. The situation is different in the case of total luminosity, see Figure 5.19d. As expected, the second cluster is mainly composed of the brightest stimuli of the data set, while the first cluster contains stimuli with middle and low luminosity values. Figure 5.19e represents cluster composition according to the object identity. It shows that the second cluster is majorly composed of specific objects, described below.

Let's look at the other side of the coin and see which is the composition of features according to cluster membership. Figures 5.19f, 5.19g and 5.19h report the value of, respectively, position, total luminosity and object identifier for each stimulus. X-axis refers to the stimulus identifiers and y-axis to the feature value.

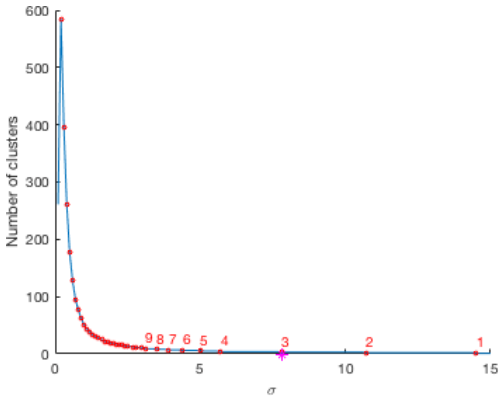
Each point is colored according to the cluster it belongs to. The position is not coded into that clusters, while the produced partition put a clear boundary between the brightest and medium/low luminosity stimuli. As Figure 5.19h regards discrete values, many rows appear, one for each object id. Light blue dotted lines are present, meaning that the second cluster is composed of specific objects. They correspond to the ones of Figure 5.13, which were already recognized as the eight brightest objects of the stimulus set. Hence the second cluster represents the brightest stimuli and, thus, the objects.

5.4.2 Results of Gaussian kernel on correlation distance

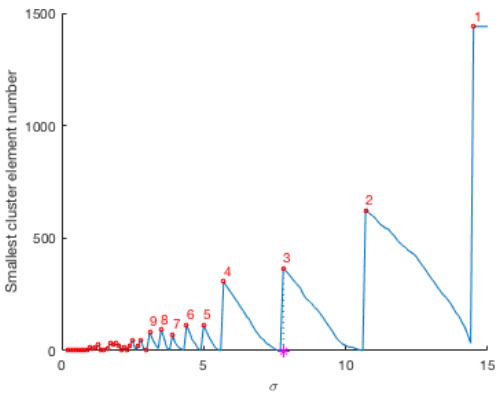
The present section discusses dominant sets results on similarity matrix created using the Gaussian kernel and the correlation distance. As in the case of Euclidean distance, smaller values of σ implies more scattered data points. Again the number of recognized clusters is very high when $\sigma \approx 0$, approximately 600. Figure 5.20b shows the size of the smallest cluster as a function of σ. As σ increases, points become less scattered until all data points are clustered into the same cluster, which happens at $\sigma = 14.5$. Again when σ increases, in general, the most cohesive cluster is divided and incorporated in the others until it disappears.

As already done for the previous analysis, consider the internal structure of clusters to choose the most appropriate value of σ. Figure 5.20c shows the MST Dunn's index, while Figure 5.20d reports the Silhouette index. The two indices show a different behavior with respect to the case of similarity matrix computed using Euclidean distances, reporting many peaks. For example, the silhouette index clearly shows three peaks corresponding to phases in which a cluster disappears being incorporated by the others. In particular at $\sigma = 5.7$ the number of clusters becomes $K = 4$, in $\sigma = 7.8$ it becomes $K = 3$ and in $\sigma = 10.7$ it becomes $K = 2$. Unfortunately, such results are not in line with MST Dunn's index, which also has many peaks, but on different values of σ. However, consider that MST Dunn's index is more sensitive to outliers. Thus Silhouette one is preferable and more reliable. Again, representational dissimilarity matrix should be used. Obviously is not possible to report the entire collection of RDMs varying on σ, but let's say that the ones associated with the presence of three clusters show a good block structure. By considering all tools, clusters produced with $\sigma = 7.8$ were chosen to be analyzed, represented with a pink asterisk in Figure 5.20.

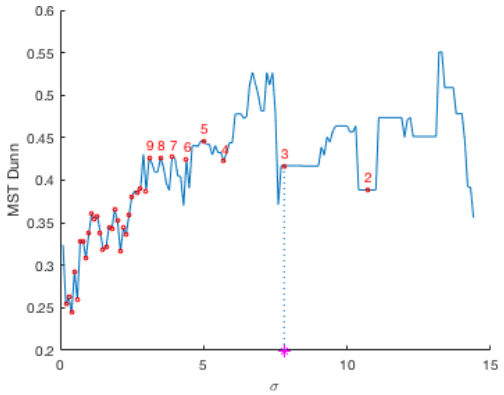
Consider now external indeces to analyze clusters relationship with features. Figure 5.20e shows AMI indeces. Position appears as the most expressed feature for each tested value of σ, reaching values near 0.4. In particular, at $\sigma = 7.8$ the



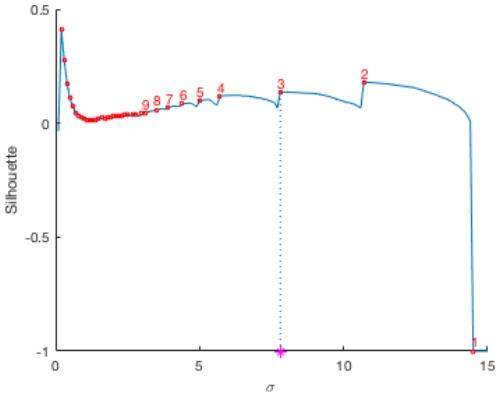
(a) Cluster number



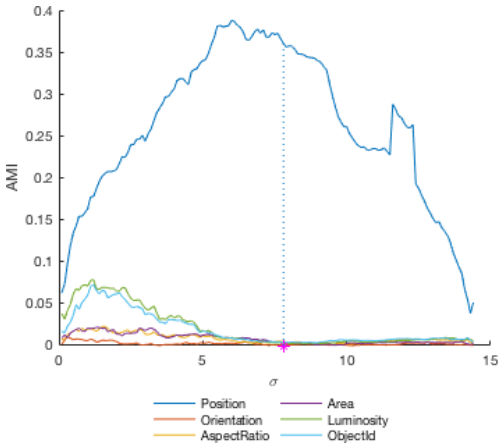
(b) Smallest cluster size



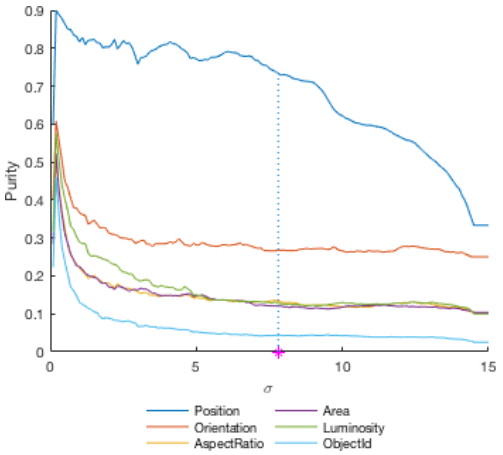
(c) MST Dunn's index



(d) Silhouette index

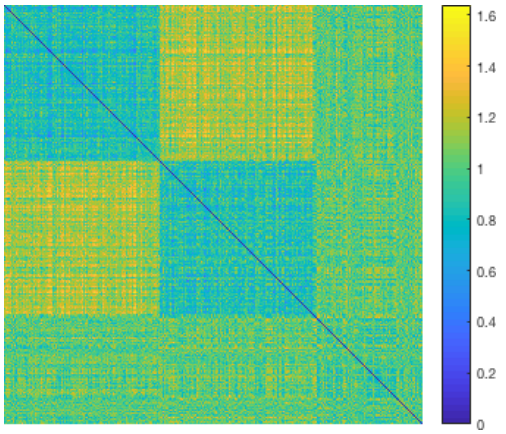


(e) AMI

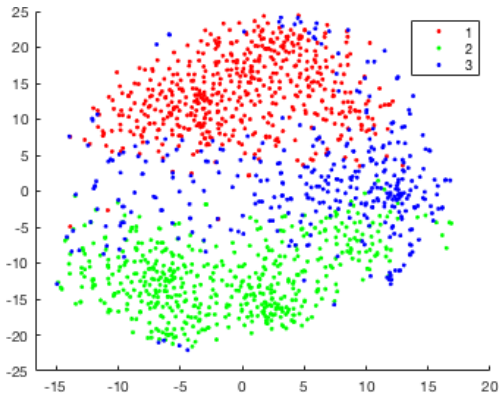


(f) Purity

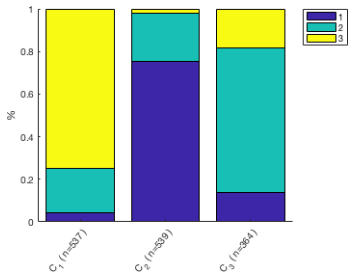
Figure 5.20: Statistics on dominant sets algorithm applied on V1 considering Gaussian kernel built using 1-Pearson correlation distances



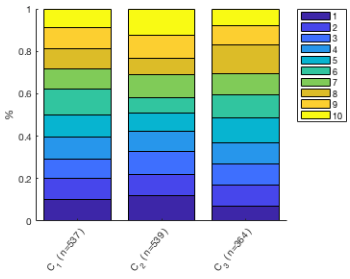
(a) RDM



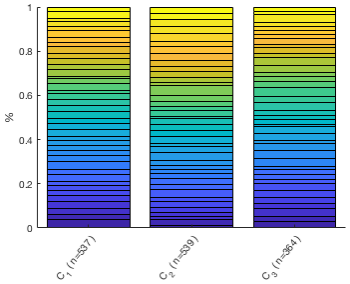
(b) t-SNE



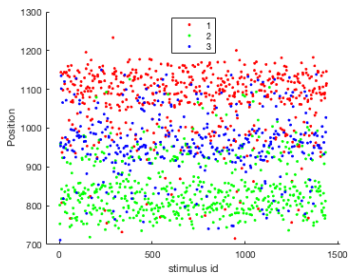
(c) Cluster vs position



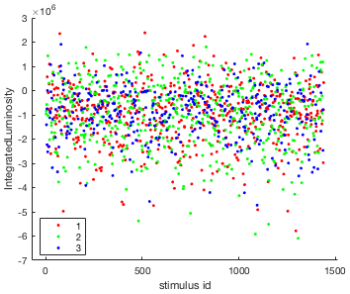
(d) Cluster vs total. lum.



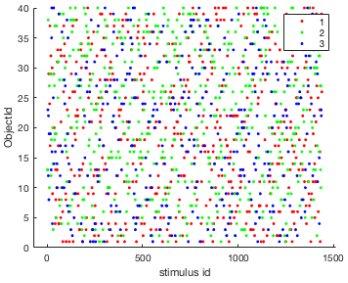
(e) Cluster vs object id



(f) Position vs cluster



(g) Total. lum. vs cluster



(h) Object id vs cluster

Figure 5.21: Clusters of V1 found by dominant sets on Gaussian kernel built using 1-Pearson correlation distances and $\sigma = 7.8$

AMI for position feature is $AMI^c_D(V1, P, \sigma = 7.8) = 0.3603$. All the other features present values smaller than 0.005 for that value of σ, e.g., total luminosity and object identity AMIs are $AMI^c_D(V1, Tl, \sigma = 7.8) = 0.002$ and $AMI^c_D(V1, Obj, \sigma = 7.8) = 0.0032$. Regarding purity index, position purity assumes extremely high values if compared with the other features. But again, pay attention on comparing purity with other features, since the number of bins is different. Anyway, each feature, except position, assumes higher values when $\sigma \approx 0$ and then decreases until, after a while, it reaches a flatted point, the threshold corresponding to the whole dataset. This does not happen in the case of position, which shows a slow decrease while reaching $\sigma = 14.5$. This suggest that position is encoded in generated clusters. In particular purity of position, total luminosity and objects identity are, respectively, $P^c_D(V1, P, \sigma = 7.8) = 0.7333$, $P^c_D(V1, Tl, \sigma = 7.8) = 0.1278$ and $P^c_D(V1, Obj, \sigma = 7.8) = 0.0424$.

Let's now deepen the analysis of cluster composition. Figure 5.21 illustrates results about the composition of clusters produced when $\sigma = 7.8$, which produces three clusters. Figure 5.21a shows the correlation RDM, ordered as already explained. The first two clusters are cohesive and well separated from each other, while the last is neither highly cohesive nor separated. Consider now the composition of clusters according to specific features.

Figure 5.21c shows composition according to feature position. Each cluster represents one bin of the feature: the first cluster represents right stimuli, the second left stimuli, instead, the third center stimuli. Even if each cluster is representative of a bin, it is not only composed by stimuli of a single bin. The first cluster, for example, is also composed of few stimuli presented on the left or at the center of the screen. Figure 5.21f report position value for each stimulus of the dataset, in which all data points are colored according to cluster membership. Again each cluster is mainly but not exclusively concentrated on specific values of position. Consider now Figures 5.21b and 5.16a, both representing t-SNE maps created using correlation distances but in which, respectively, data points are colored according to cluster membership and position value. The three groups recognized in Section 5.3 are not exactly identified by the three clusters, making fuzzy the distinction among them. Figures 5.21d and 5.21e show cluster composition according to total luminosity and object id. Differently to what happens in the case of Euclidean distance, here no cluster represents a certain range of luminosity or specific object identities.

5.5 Kmeans on V1

In the following section, kmeans algorithm is applied to V1 and is used to compare results obtained with dominant sets framework. In [1] the algorithm was used to find which was the contribution of a specific feature to create clusters. Hence, for each analyzed feature, kmeans was applied with a value of K equals to the number of feature bins. Then produced clusters were compared with such bins.

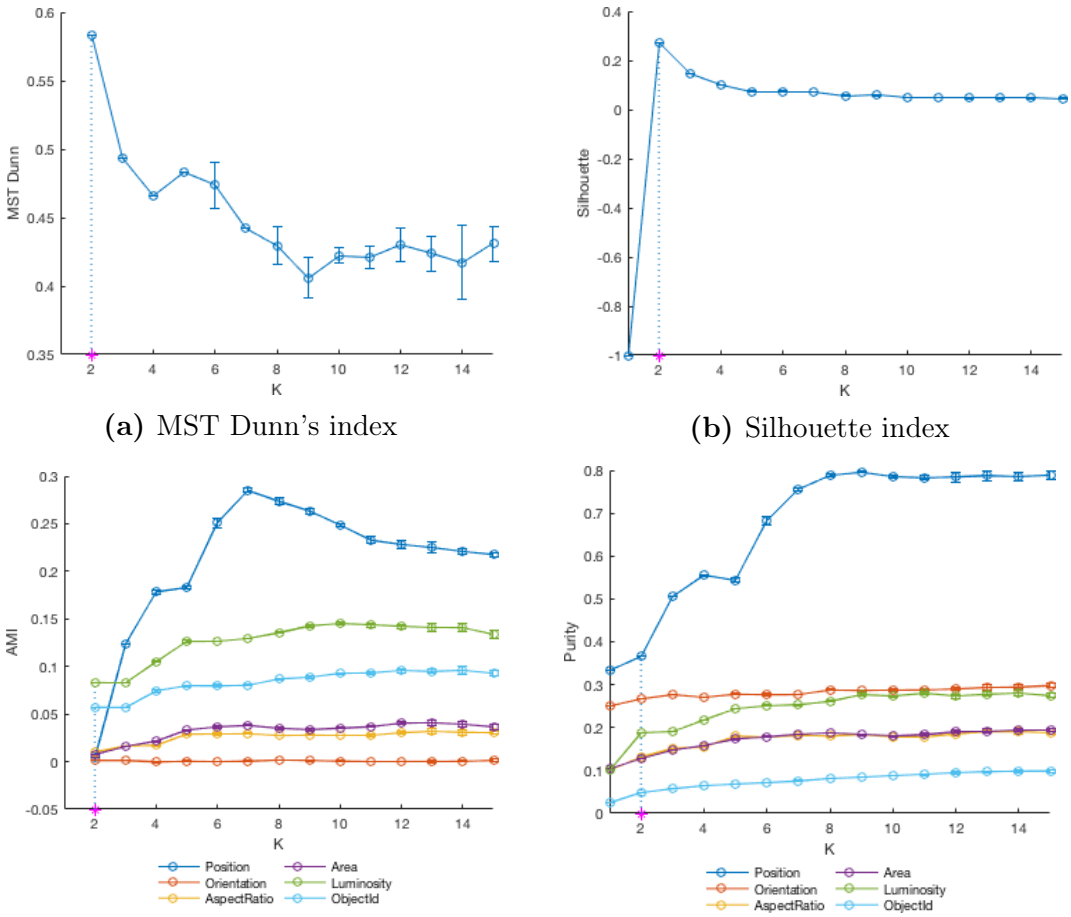
However, as explained in Section 4.6, it is preferable to execute kmeans for a specific range of K, without setting K a priori, to find which is the best underlying data structure. Then the correct value of K should be selected according to specific indicators. In the analysis Silhouette and MST Dunn's indexes are again used for the sake of comparison. After that, external indices are used to find which is the contribution of a particular feature on creating clusters. Once we selected the most appropriate value of K, we analyzed resulting clusters composition. Moreover, contrary to what was done in [1] and to compare dominant sets results, we run kmeans was run using both Euclidean and correlation distances.

We noticed that the used distance measure strongly influences results, pointing out different intrinsic characteristics. In particular, it emerges again that Euclidean distances produce clusters representing the stimulus luminosity, while correlation distances extrapolate clusters highly related to position bins.

5.5.1 Results of Euclidean distance

The present section illustrates results of kmeans run with Euclidean distances. Again, the first step of the analysis consists of considering cluster internal structures. Figures 5.22a and 5.22b show MST Dunn's and Silhouette indexes average values on 10 kmeans runs for $K = 1, ...15$. As indicated in Section 4.4.3, for both indices the best clustering schema (and thus the value of K) is the one that maximizes the index. Therefore, by considering both indices, the most appropriate candidate is $K = 2$.

To visually assess what happens to cluster intra and extra distances, the evolution of Euclidean RDMs for the produced clusters varying on K could be used. Figure 5.23 reports such evolution for the considered values of K. RDMs suggest that the choice of $K = 2$ is the best one. Indeed each value of K produces two different families of clusters. The first family is composed of clusters that are highly cohesive but similar to each other and different from the other family. The second family is composed of clusters with low cohesion and dissimilar from all the other clusters, independently of the original family. This suggests a partition into two



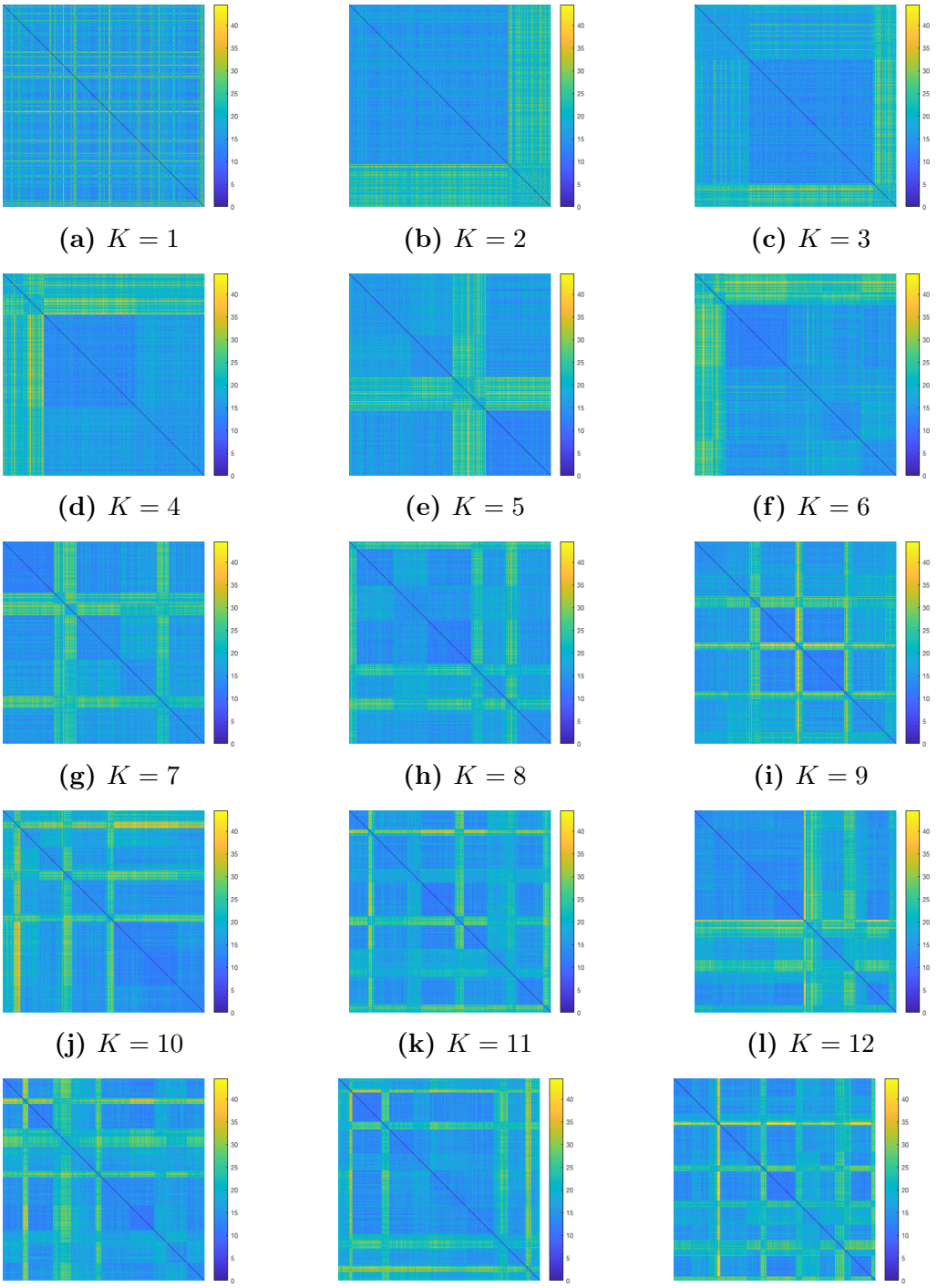
(a) MST Dunn's index

(b) Silhouette index

(c) AMI

(d) Purity

Figure 5.22: Statistics on kmeans algorithm applied considering Euclidean distances on level V1. We run kmeans 10 times for each value of K, reporting average and standard deviation for all indexes.



(a) $K = 1$ **(b)** $K = 2$ **(c)** $K = 3$

(d) $K = 4$ **(e)** $K = 5$ **(f)** $K = 6$

(g) $K = 7$ **(h)** $K = 8$ **(i)** $K = 9$

(j) $K = 10$ **(k)** $K = 11$ **(l)** $K = 12$

(m) $K = 13$ **(n)** $K = 14$ **(o)** $K = 15$

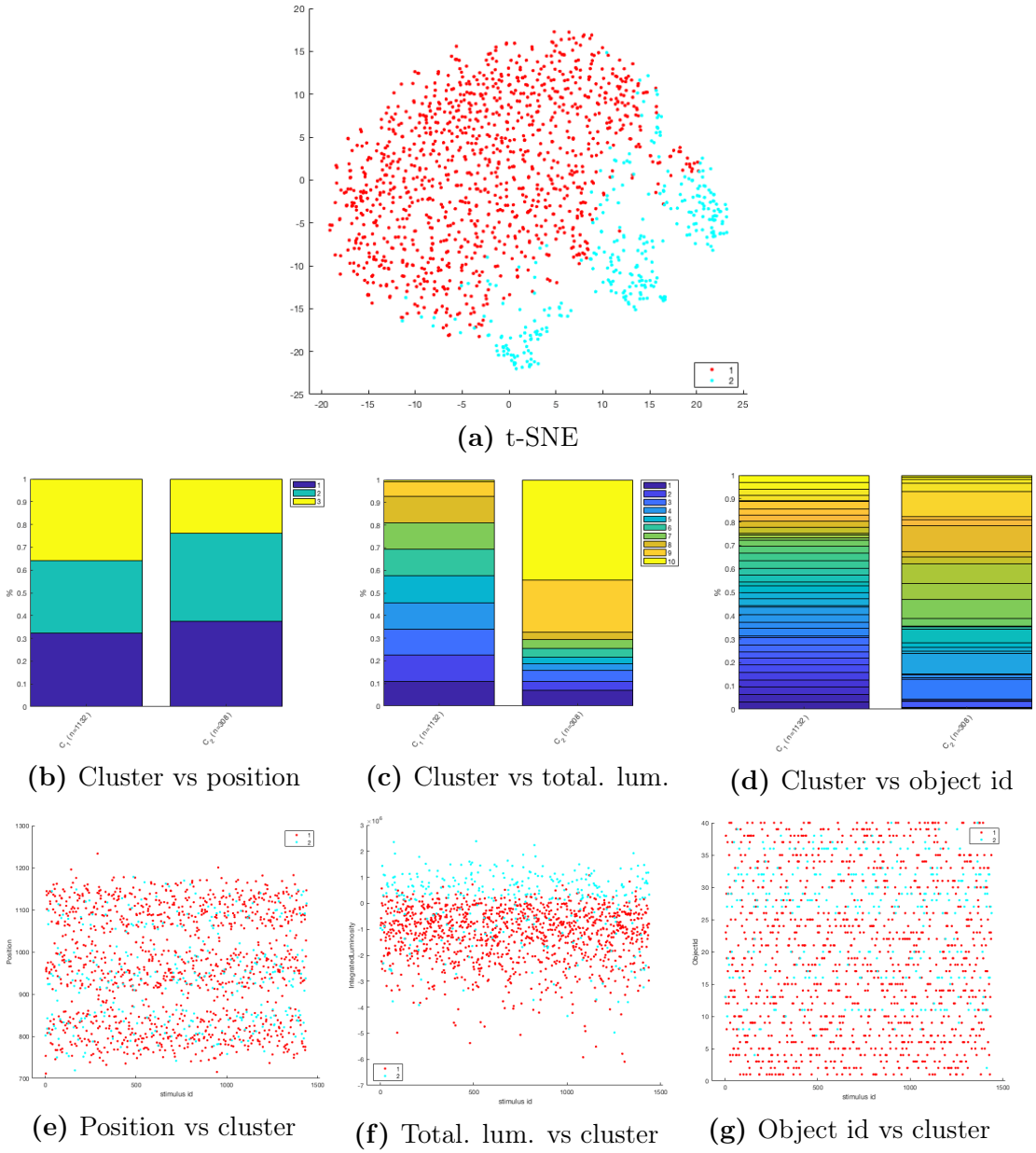
Figure 5.23: Euclidean RDMs of level V1 ordered according to kmeans partitions computed with Euclidean distances on $1 \leq K \leq 15$

sets. Indeed, RDM related to $K = 2$ shows a big, cohesive and well-separated cluster and a small, not much cohesive but well separated too cluster.

Consider external indexes to verify how such clusters are composed. Figure 5.22 shows averages of AMI and purity indices on 10 kmeans runs. Each point in a curve represents the average index value and its standard deviation. Again each feature has a specific color. Consider Figure 5.22c, which shows the average AMI as a function of K. When $K = 2$ the two features with highest AMI values are total luminosity ($AMI^e_K(V1, Tl, K = 2) = 0.083$) and object id ($AMI^e_K(V1, Obj, K = 2) = 0.0574$), while all the other features, like position ($AMI^e_K(V1, P, K = 2) = 0.0048$), show AMI values near to 0. Instead, purity of total luminosity and object identity are $P^e_K(V1, Tl, K = 2) = 0.1882$ and $P^e_K(V1, Obj, K = 2) = 0.0486$, while for position is $P^e_K(V1, P, K = 2) = 0.3653$. With higher values of K, the AMI values of quite almost all features increase.

Consider now in more in-depth details the analysis of clusters produced with $K = 2$. Figure 5.24 illustrates pictures useful for understanding how clusters are composed. Figure 5.24a shows each stimulus mapped into the 2-dimensional map produced by t-SNE with Euclidean distances. The first cluster is the one colored in red; the second is the one colored in light blue. By comparing that picture with the one of Figure 5.15b, it appears that the first/red cluster exactly matches the same area of the map in which total luminosity assumes its highest values. The second cluster represents stimuli with low and middle luminosity. Figures 5.24b and 5.24c confirm such observations, the former regarding the composition of clusters according to the position, the latter regarding the same but according to total luminosity. No trend exists for feature position; indeed both clusters are composed exactly in the same way, without containing position specific stimuli. Instead, the situation is different for total luminosity. As already suspected by t-SNE plot, the first cluster is mainly composed of elements with high luminosity (bins number 9 and 10), while the second cluster is composed of low and middle luminosity stimuli. Figures 5.24e, 5.24f and 5.24g show value of, respectively, position, total luminosity and object identity for each stimulus, whose color reflect cluster membership. The position is not coded into that clusters; indeed colors are scattered on the entire range of position values. Instead, total luminosity highest values are well separated, even if not perfectly. Also, in that case, the eight brightest objects are discriminated from all the others, belonging to the cluster of the brightest stimuli.

This clustering partition is not so different from the one proposed in Section 5.4.1. Rather, it shares more common characteristics than different ones. Both clustering schemas propose two bunches of data: the first cluster is small, not strongly cohesive



(a) t-SNE

(b) Cluster vs position

(c) Cluster vs total. lum.

(d) Cluster vs object id

(e) Position vs cluster

(f) Total. lum. vs cluster

(g) Object id vs cluster

Figure 5.24: Clusters of V1 found with kmeans on Euclidean distances and $K = 2$

but well separated from the other and consists of the brightest stimuli and thus the eight brightest objects; the second is big, cohesive, separated from the latter and composed by middle and low luminosity stimuli. Nevertheless, dominant sets clusters strongly represent the distinction of high luminosity stimuli. Indeed $AMI^e_D(V1, Tl, \sigma = 336) = 0.096$ is higher than in the case of kmeans in which $AMI^e_K(V1, Tl, K = 2) = 0.083$. The same happens in the case of purity index with $P^e_D(V1, Tl, \sigma = 336) = 0.1903$ against $P^e_K(V1, Tl, K = 2) = 0.1882$. Nevertheless, these differences are not substantial.

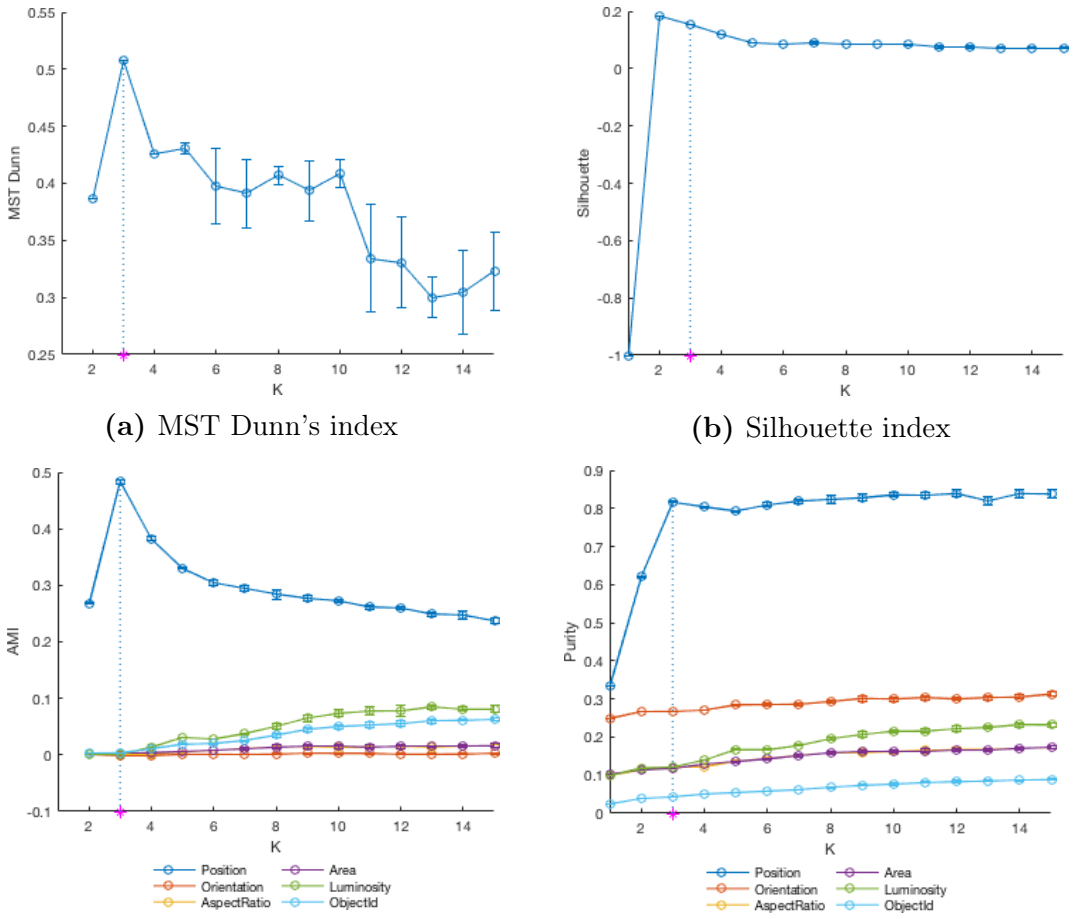
5.5.2 Results of 1-Pearson correlation distance

Consider now the same analysis but with a different distance measure: the correlation distance. Again the first step is to consider internal indices, shown in Figure 5.25. MST Dunn's and Silhouette indexes are not maximized by the same value of K. According to MST Dunn's index, the best value is $K = 3$. Instead, according to Silhouette, $K = 2$ is the best. However, Silhouette index for $K = 3$ is not too much different from $K = 2$, contrary to what happens with Euclidean distance in which the choice was forced to $K = 2$. Thus, since MST Dunn's index strongly indicates $K = 3$ and Silhouette index reports similar values of $K = 2, 3$, $K = 3$ is the chosen value for the parameter, marked again with the pink asterisk.

Consider RDMs regarding correlation distances ordered as previously described, in Figure 5.26. Clusters tend to be of similar size, differently to what happens in the case of Euclidean distance. Notice that diagonal blocks of all matrices appear darker more or less the same way, suggesting that all clusters are cohesive enough. About separation from the other clusters, for some pairs of clusters, the similarity tends to be high, even if not so much as in the case of the Euclidean distance. The two RDMs which mostly report proper clustering schema, considering both internal homogeneity and external inhomogeneity, are $K = 2, 3$, even if the center cluster in $K = 3$ shares similarities with the other two clusters.

Consider AMI and purity values in Figure 5.25. Indices values related to position feature are much higher with respect to kmeans run with Euclidean distances for each value of K. This suggests that correlation distances better underline position feature. Moreover, in $K = 3$ the AMI index reaches a peak, with $AMI^c_K(V1, P, k = 3) = 0.4836$. Also purity index shows a maximum point in $K = 3$, with $P^c_K(V1, P, k = 3) = 0.8164$. The other features appear weakly coded into clusters, reporting values near to 0 for AMI and values near to the threshold of the whole dataset for purity index.

Consider now more in-depth details for the analysis of clusters composition when



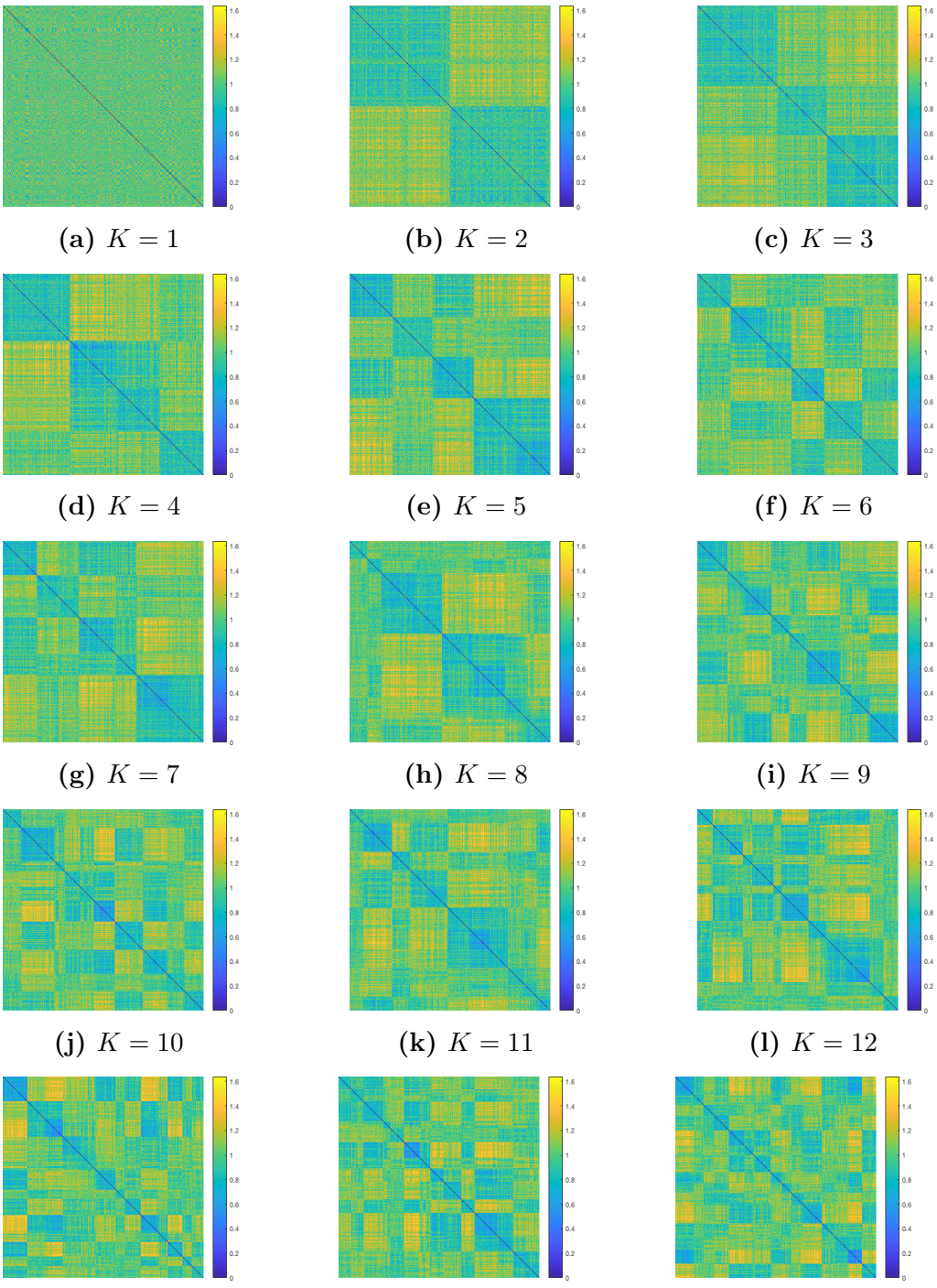
(a) MST Dunn's index

(b) Silhouette index

(c) AMI

(d) Purity

Figure 5.25: Statistics on kmeans algorithm applied considering 1-Pearson correlation distances on level V1. We run kmeans 10 times for each value of K, reporting average and standard deviation for all indexes.



(a) $K = 1$ **(b)** $K = 2$ **(c)** $K = 3$

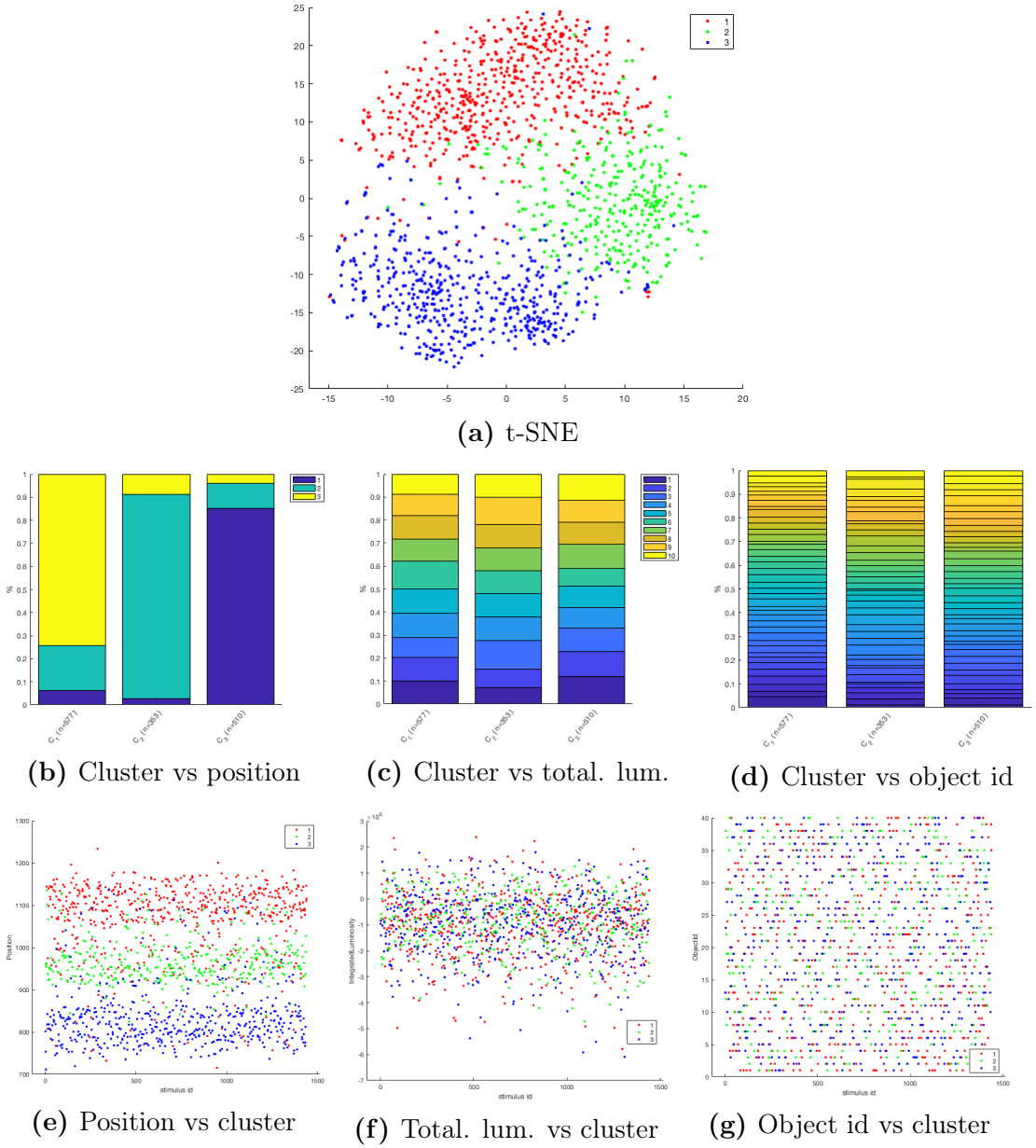
(d) $K = 4$ **(e)** $K = 5$ **(f)** $K = 6$

(g) $K = 7$ **(h)** $K = 8$ **(i)** $K = 9$

(j) $K = 10$ **(k)** $K = 11$ **(l)** $K = 12$

(m) $K = 13$ **(n)** $K = 14$ **(o)** $K = 15$

Figure 5.26: Correlation RDMs of level V1 ordered according to kmeans partitions computed with 1-Pearson correlation distances on $1 \leq K \leq 15$



(a) t-SNE

(b) Cluster vs position

(c) Cluster vs total. lum.

(d) Cluster vs object id

(e) Position vs cluster

(f) Total. lum. vs cluster

(g) Object id vs cluster

Figure 5.27: Clusters of V1 found with kmeans on 1-Pearson correlation distances and $K = 3$

$K = 3$. Figure 5.27 illustrates such composition. Position acts as a discriminator factor for cluster building, while total luminosity and object identity appear to play no role. Figure 5.27b reports cluster composition according to position. The first cluster mainly represents the first bin of position, which corresponds to left position. The second cluster represents elements of the second bin, or better screen centered stimuli. The third cluster represents the last bin, thus stimuli whose object lies on the right of the screen. Differently, to the clustering schema proposed by dominant sets with correlation distances and $\sigma = 7.8$, each cluster strongly represents a bin of position. Refer to Figure 5.16a, which represents t-SNE maps created with correlation distance whose data points are colored according to position value. Each one of the group found in that figure is represented in a more defined way than dominant sets did. In fact, both AMI and purity indices of kmeans clusters assume higher value then dominant sets: from 0.3603 to 0.4836 for AMI and 0.7333 to 0.8164 for purity index. As in the case of dominant sets, total luminosity and object identity did not act as a discriminator factor among clusters.

5.6 Comparison of clustering results on V1-LM-LI-LL

We conducted the same analysis reported in the previous sections for all area of the V1-LM-LI-LL progression. Notably, for both algorithms parameters were tuned as showed in the case of V1. Nevertheless, such analysis is omitted in this thesis since it might be monotonous and, majorly, not useful. However, they are reported in Appendix A and B. Instead, let's examine outcomes among the several areas, by checking how features are encoded along the progression and comparing results of the two clustering algorithms combined with both Euclidean and 1-Pearson correlation distances. Table 5.1 shows internal indexes and external ones for features position, total luminosity, and object identity for each level, algorithm and distance metric. Particularly, notice that each level is colored differently: V1 indexes are colored in red shades, LM in orange, LI in green and, eventually, LL in blue. Moreover, results of Euclidean distance are colored with darker shades if compared with correlation results. We wrote the best value of an index among all areas and algorithms in bold. Furthermore, the best value of an index inside a single level is underlined with a line whose color respects the area it belongs. Firstly, recap results for each level.

Area V1 emerges as the one the highly encodes analyzed features. First of all, kmeans produced majorly cohesive clusters by using correlation distances

according to both MST Dunn's and silhouette indexes. The contrary holds for dominant sets, which produced better partition with similarities based on the Euclidean metric. As already seen, Euclidean distances let luminosity differences emerge. Both algorithms proposed two bunches of data in which one cluster strongly represents the brightest stimuli of the dataset, showing AMI values of one order of magnitude bigger than correlation distances. Dominant sets produced clusters majorly oriented to this feature, with $AMI^e_D(V1, Tl, \sigma = 336) = 0.096$ against the one of kmeans $AMI^e_K(V1, Tl, \sigma = 2) = 0.083$. Purity also clearly indicates that dominant sets clusters are much clearer, with a purity of 0.1903 against kmeans with 0.1882. This also holds for object identity, but we already saw how it is related to luminosity. Indeed the recognized objects are the brightest of the entire collection, making their recognition caused by luminosity. Instead, 1-Pearson correlation better represents position feature. Both algorithms proposed a partition of three classes, each one representing one bin of position. AMI values are about two magnitude orders bigger than Euclidean results, with kmeans and dominant sets values of, respectively, 0.4836 and 0.3603. In that case, kmeans clusters are purer, with $P^c_K(V1, P, K = 3) = 0.8164$.

As regard LM area, dominant sets produced better partition according to both internal indexes. However, again euclidean underlines luminosity feature, while correlation codes for the position, even if with less emphasis. The two algorithms, using Euclidean distances, proposed again two bunches in which one cluster represents the brightest stimuli. Conversely, dominant sets based on correlation distances proposed two clusters representing left and right stimuli, while kmeans proposed a partition again into three class, one for each position bunch. Nevertheless, kmeans produced purer clusters for both features.

The third level LI of the progression is probably the ones less coding for features. As in the first level, dominant sets produced better partition with Euclidean distance, while kmeans with correlation distance. According to correlation results, no feature information is conveyed. Indeed, both algorithms better encode position and luminosity using Euclidean distances. However, no feature particularly emerges. Moreover, both dominant sets and kmeans with Euclidean distances produced two clusters, one coding for some right and bright stimuli.

The last level LL is particularly difficult to be read too. Again dominant sets generated better partition according to internal indexes. Surprisingly, in this level euclidean distances let feature position emerges, while 1-Pearson let luminosity emerges. In particular, all algorithm and distance combinations produced two bunches in which one cluster partially represents bright and right stimuli.

The two algorithms produced similar results, making difficult to say which is

the best. Generally, with V1 and LI areas dominant sets produced better partition with Euclidean distance, while kmeans with correlation distances. Instead, for LM and LL areas dominant sets always generated better clustering schema.

Let's now analyze how information is coded along the hierarchy. Partitions produced with correlation distances better encodes position feature. Both AMI and purity indexes show a decreasing trend across the V1-LM-LI-LL progression, gradually disappearing. The same happens for luminosity, and thus object identity, when Euclidean distances are used. Instead, when clustering schema is produced with Euclidean metric, feature position becomes more encoded along the progression, being more represented in the last level LL. However, the values the external indexes assume in LL with Euclidean measure are much smaller than the case of the first level V1 with correlation distances. In particular, AMI of position for kmeans with euclidean distances is $AMI^e_K(LL, P, k = 2) = 0.0064$, while for correlation distances on V1 is $AMI^c_K(V1, P, k = 3) = 0.4836$. Surprisingly, the same happens for luminosity with correlation distance: LI is the level which less encodes, then V1, LM, and LL. In particular, the last level LL presents very high value of AMI and purity. Indeed, consider that for the first level V1, for example with dominant sets, the adjusted mutual information is $AMI^e_D(V1, Tl, \sigma = 336) = 0.096$, where for LL it is $AMI^c_D(LL, Tl, \sigma = 10) = 0.0264$. Even if the index is smaller for LL, it has the same magnitude order, suggesting that relationship is strong. This could be seen also in Figure 5.9, which shows correlation RDMs ordered according to total luminosity. Matrix related to the last level clearly shows a block structure, with a small block representing brighter objects.

Concluding, no clustering algorithm is better than the other, not significantly. Moreover, algorithms that use Euclidean distances encode luminosity and object identity, which disappears across the progression. Instead, correlation distance let position emerges, again following a decreasing trend. Level LL represents an exception, with luminosity being well represented with correlation distances.

Level	Algorithm			MST Dunn	Silhouette	AMI			Purity		
						Pos.	Tot. lum.	Obj. id	Pos.	Tot. lum.	Obj. id
V1	Kmeans	E	$K = 2$	0.583	0.2696	0.0048	0.083	0.0574	0.3653	0.1882	**0.0486**
		C	$K = 3$	0.5083	0.1549	**0.4836**	0.0005	0.004	**0.8164**	0.1216	0.0435
	Dom. sets	E	$\sigma = 336$	**0.6868**	0.3149	0.0011	**0.096**	**0.0582**	0.3507	**0.1903**	0.0472
		C	$\sigma = 7.8$	0.4171	0.1361	0.3603	0.002	0.0032	0.7333	0.1278	0.0424
LM	Kmeans	E	$K = 2$	0.3692	0.2108	0.006	0.0342	0.0257	0.3785	0.1639	0.0458
		C	$K = 3$	0.2939	0.1632	0.2611	0.0145	0.0084	0.6142	0.1528	0.0431
	Dom. sets	E	$\sigma = 170$	0.5199	0.3246	0.0003	0.0304	0.0293	0.3451	0.1507	0.0438
		C	$\sigma = 10.9$	0.3492	0.1758	0.2372	0.0049	0.0038	0.5958	0.1299	0.0382
LI	Kmeans	E	$K = 2$	0.3882	0.1495	0.0167	0.0094	0.005	0.411	0.1417	0.0382
		C	$K = 2$	0.5275	0.1388	0.001	0.0009	0.0005	0.3576	0.1146	0.034
	Dom. sets	E	$\sigma = 250$	0.5078	0.3099	0.0131	0.0207	0.0112	0.3688	0.1333	0.0368
		C	$\sigma = 9.1$	0.5087	0.1381	0.0025	0.0008	0.0006	0.3646	0.1146	0.0347
LL	Kmeans	E	$K = 2$	0.4019	0.2072	0.0064	0.0114	0.0068	0.3769	0.1444	0.041
		C	$K = 2$	0.3633	0.1458	0.0011	0.0183	0.0109	0.358	0.1503	0.0431
	Dom. sets	E	$\sigma = 250$	0.5433	**0.4191**	0.0026	0.0182	0.0142	0.3472	0.1285	0.0361
		C	$\sigma = 10$	0.4386	0.1416	0.0016	0.0264	0.0153	0.3535	0.159	0.0451

Table 5.1: Summary of clustering indexes. Results of each level is colored differently: V1 in red, LM in orange, LI in green and LL in blue. Results related to euclidean distance are colored with darker shades, vice-versa for results of correlation distance. For each index, the combination which gets the best results is marked, while the best one inside a single area is underlined with a bar of the same color of the level.

Chapter 6

Discussion

In this thesis, we further investigate whether rats' visual system can recognize objects against several identity-preserving transformations, e.g., rotations, scaling, translations, background context, etc. Our analysis were done on neural responses datasets kindly granted by SISSA (Trieste). Particularly, SISSA researches performed an experiment in which 1440 stimulus images were presented to rats, whose neuronal responses were recorded in a specific area of rat's visual system called extrastriate lateral stream, made, in turn, by the V1-LM-LI-LL progression. On this neural responses we first performed single cell analysis, in which mutual information was used to check the relation between stimulus features and responses of single neurons. Position, luminosity and object identity emerge as the most encoded features, with decreasing trend across the hierarchy. For that reasons, these three features were extensively analyzed in population analysis. We use appropriately ordered RDMs to visually assess whether these features generate block structures. We noticed how different distance metric strongly influences blocks structure, with Euclidean distance showing blocks when rows and columns are ordered according to stimulus luminosity; while correlation matrices show better block structure when sorted according to stimulus position, except for the last level LL, in which groups are more evident if ordered by luminosity. Moreover, when euclidean RDMs are ordered according to object identity, eigth objects appear to be sensitively different from all the others. This makes us think that such objects are perfectly discernible. However, they were the brightest objects of the entire collection, suggesting that they were not discriminated because of their identity rather of their luminosity. Furthermore, we also use t-SNE maps in which each point was colored according to specific feature, confirming the analysis of RDMs. Then, we performed cluster analysis using both kmeans and dominant sets algorithm, combined with either Euclidean or 1-Pearson correlation distances. The behavior of the two algorithms

with a specific distance was the same. Using Euclidean distances on V1, both algorithms generate two bunches in which the brightest stimuli were grouped into the same bunch. When correlation distances were used with V1, three clusters were produced by both algorithms, each one related to one bin of position feature. Along the V1-LM-LI-LL progression, the algorithms tend to produce a partition of two bunches on data points. Again, Euclidean distances underline luminosity, while correlation position. However, clusters become noisier and noisier. An exception is again presented in LL, where clusters produced with correlation distances better encode stimulus luminosity than position.

In future developments, we could try to apply clustering algorithms that combine both distance measures. Indeed, since Euclidean and correlation distances respectively underline luminosity and position features, the combination of them might include a more significant amount of information which could bring to exciting results. Moreover, another idea could be to use other different metrics, which might stress differences of stimuli according to other features. Furthermore, since clusters produced in the first level V1 were particularly representative of specific stimuli classes, we could try to build a classifier that, given the set of neuronal responses for a single stimulus, tries to assign it a class (like "light" stimulus, "right" stimulus, etc.). However, these are just a few examples of possible developments in a scientific field so young and unexplored as neurobiology.

Appendix A

Dominant sets on LM, LI and LL

In this section we discuss results of dominant sets algorithm on levels LM, LI and LL, which are omitted in the body of the thesis. The algorithm was executed twice, once for each distance measure used to define data points similarities. The performed analysis was the same already conducted for level V1. We firstly tuned the parameter σ using both the internal indexes, Silhouette and MST Dunn's. Then, we analyzed AMI and purity indexes behavior as function of σ. Once σ was selected, we again investigated how clusters were composed to understand which features were encoded.

A.1 DS results on LM with similarities based on Euclidean distance

Figure A.1 shows some statistics on results of dominant sets algorithm applied to level LM and based on the Gaussian kernel built using the Euclidean distances. The tuning of the parameter strongly remembers the corresponding one performed for level V1. Again two families of stimuli appear, one composed of stimuli similar to each other and dissimilar to the other family, one composed of stimuli highly dissimilar to all the others. Following the same considerations made for V1, $\sigma = 170$ is selected (marked with a pink asterisk). By considering the external indices AMI and purity, total luminosity and object identity are the most encoded, with $AMI^e_D(LM, Tl, \sigma = 170) = 0.0304$ and $AMI^e_D(LM, Obj, \sigma = 170) = 0.0293$, confirming that Euclidean distances particularly encode for such features. Consider Figure A.2, which shows clusters composition. As expected, the RDM reported in Figure A.2a shows that the clustering schema is composed of two clusters, one particularly cohesive and well-separated, and the second neither cohesive nor separated. The second cluster contains light stimuli and, thus, the corresponding

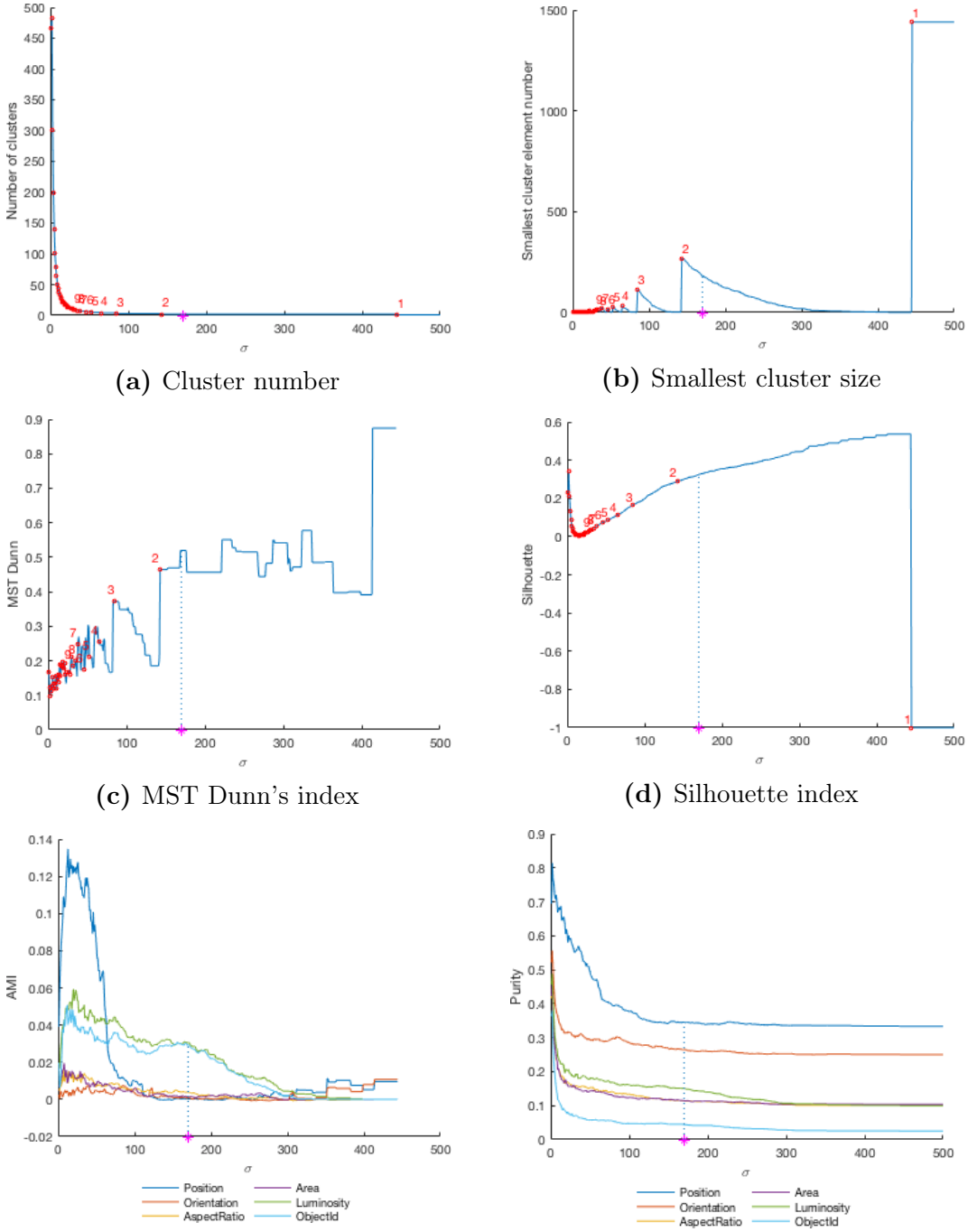
brightest objects. However, boundary between light and mid/high-luminosity stimuli is not so clear as in the case of level V1. Instead, the first cluster is not composed by specific stimuli. Concluding, as happened for V1, Euclidean distance let stimulus luminosity emerges, while position is almost not coded.

A.2 DS results on LM with similarities based on 1-Pearson correlation distance

Figure A.3 shows statistics related to results of dominant sets algorithm applied to level LM and based on the Gaussian kernel built using the 1-Pearson correlation distances. This time selection of parameter σ is simpler. Indeed, both Silhouette and MST Dunn's indexes indicate $\sigma = 10.9$ as the most appropriated. As happened for level V1, both AMI and purity suggest that position is the most encoded feature, with $AMI^c_D(LM, P, \sigma = 10.9) = 0.2372$. Figure A.4 shows cluster composition of the two produced bunches. Two clusters are recognized, both cohesive and well-separated from the other (as can be seen in the RDM reported in Figure A.4a). The first represents "right" stimuli, while the second represents "left" stimuli. No cluster strongly consists of specific luminosity values or object identifiers.

A.3 DS results on LI with similarities based on Euclidean distance

Figure A.5 shows statistics on results of dominant sets algorithm applied to level LI and based on the Gaussian kernel built using the Euclidean distances. Again the two families of stimuli appear, precisely as for levels V1 and LM. Let's select $\sigma = 250$, which generates the two clusters whose composition is figured out in Figure A.6. The first cluster is particularly bigger if compared with the second, which contains only 102 of the entire dataset. Moreover, the second cluster is composed by both light and "right" stimuli. Then, obviously, it contains two of the brightest objects, precisely the ones with identifiers 32 and 36. Even if it is majorly composed of light stimuli, these clusters do not strongly code for a specific feature. Let's focus on Figures A.5e and A.5f, which report AMI and purity indexes as σ varies. No feature appears to be particularly coded, with total luminosity being slightly higher if compared with the others. Moreover, notice that AMI values are much smaller than the ones related to level V1 (Figure 5.17e) and LM (Figure A.1e), suggesting that the level does not strongly code for either luminosity or position.



(a) Cluster number

(b) Smallest cluster size

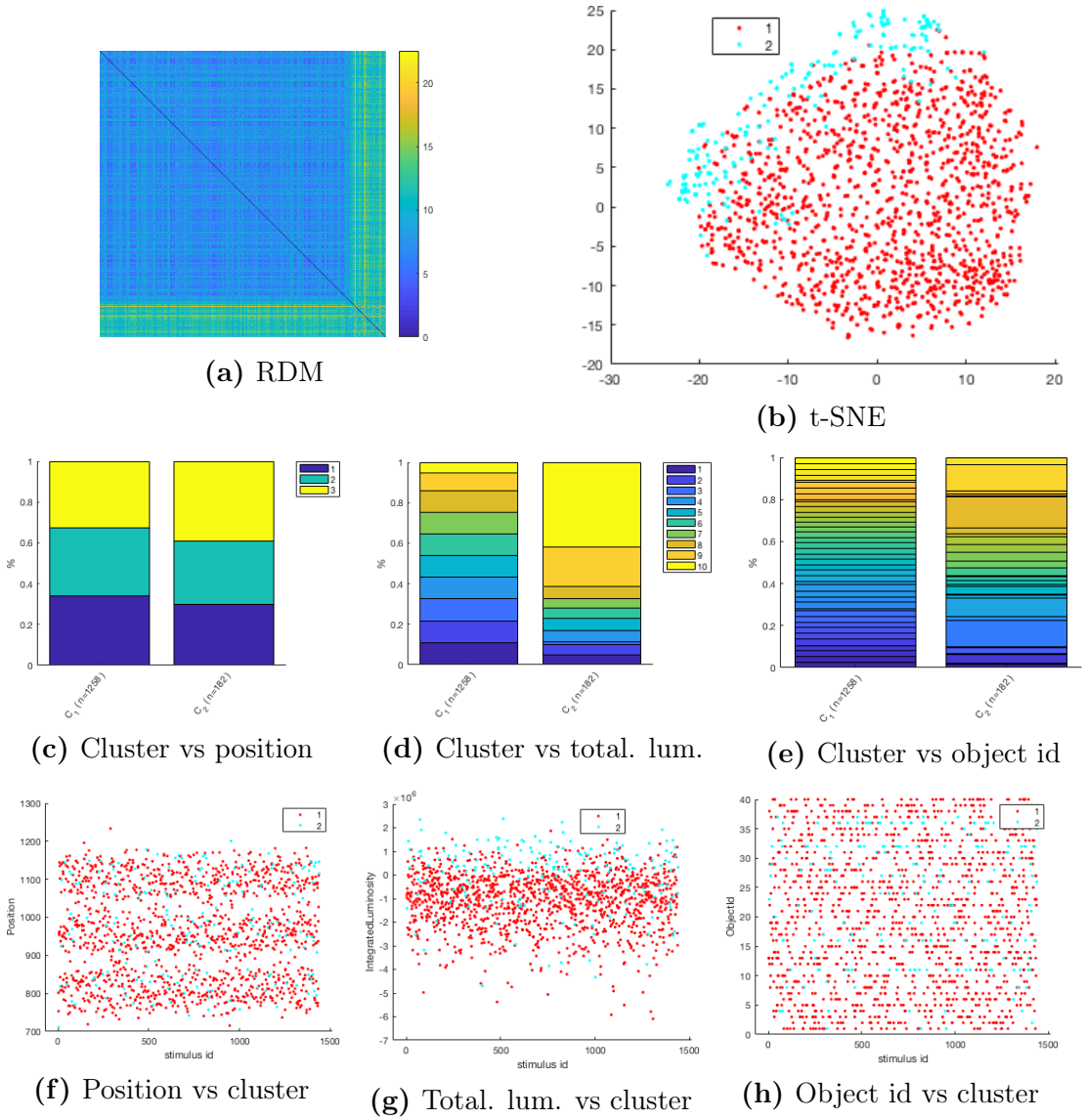
(c) MST Dunn's index

(d) Silhouette index

(e) AMI

(f) Purity

Figure A.1: Statistics on dominant sets algorithm applied on LM considering Gaussian kernel built using Euclidean distances

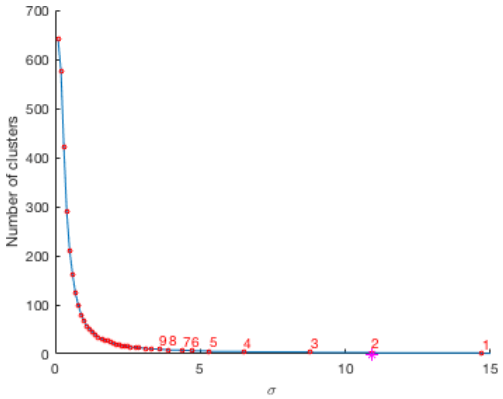


(a) RDM **(b)** t-SNE

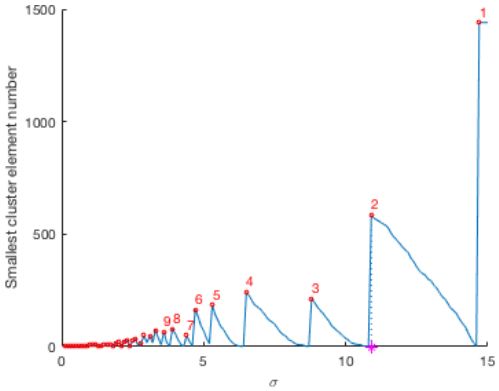
(c) Cluster vs position **(d)** Cluster vs total. lum. **(e)** Cluster vs object id

(f) Position vs cluster **(g)** Total. lum. vs cluster **(h)** Object id vs cluster

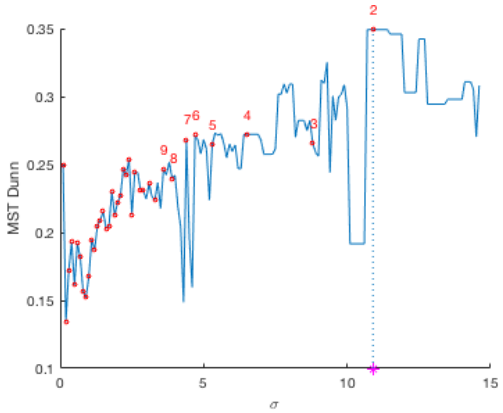
Figure A.2: Clusters of LM found by dominant sets on Gaussian kernel built using Euclidean distances and $\sigma = 170$



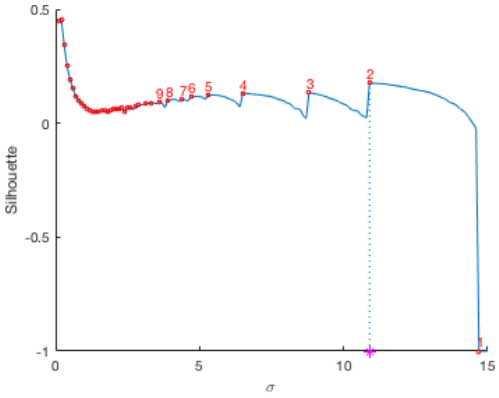
(a) Cluster number



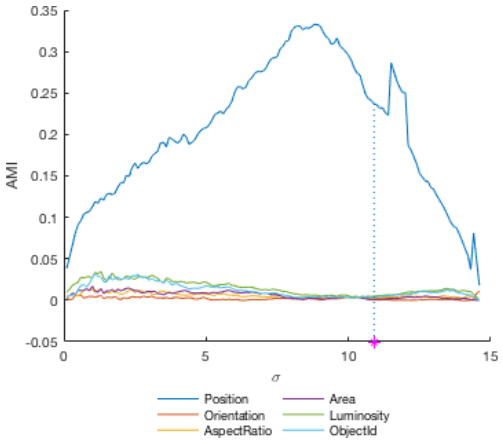
(b) Smallest cluster size



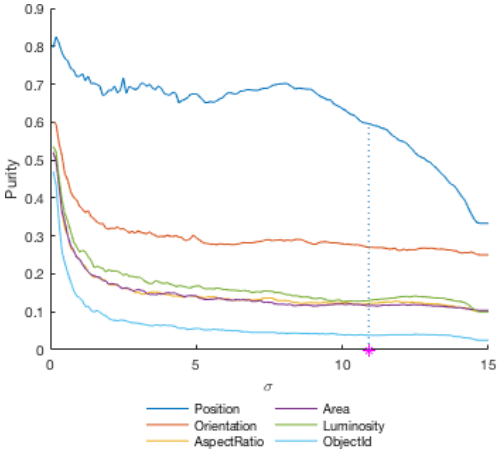
(c) MST Dunn's index



(d) Silhouette index

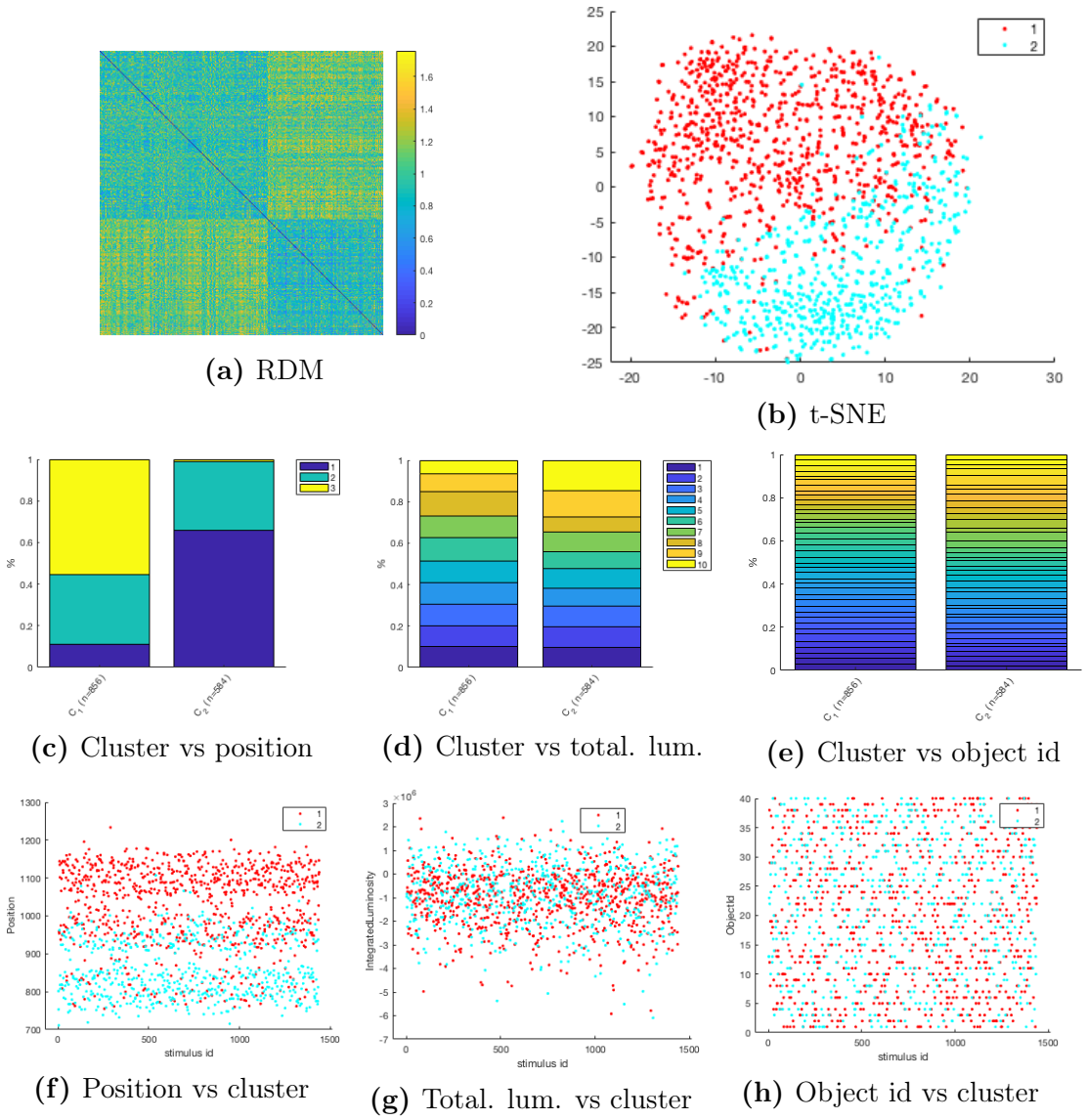


(e) AMI



(f) Purity

Figure A.3: Statistics on dominant sets algorithm applied on LM considering Gaussian kernel built using 1-Pearson correlation distances



(a) RDM

(b) t-SNE

(c) Cluster vs position

(d) Cluster vs total. lum.

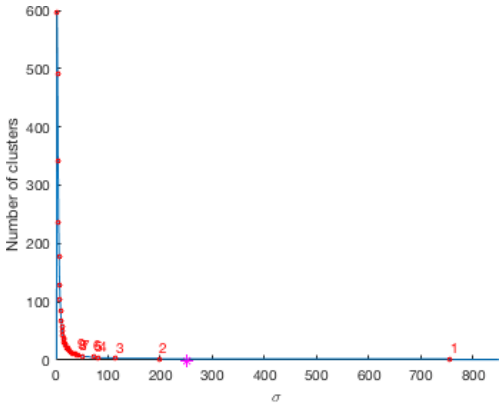
(e) Cluster vs object id

(f) Position vs cluster

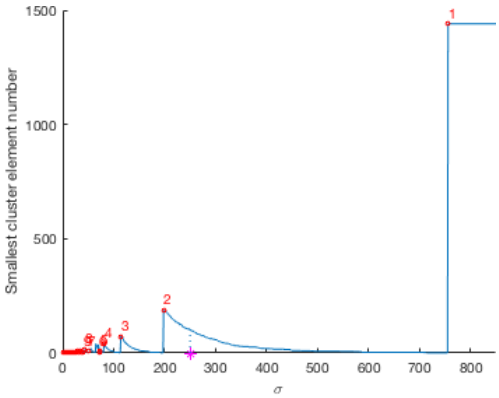
(g) Total. lum. vs cluster

(h) Object id vs cluster

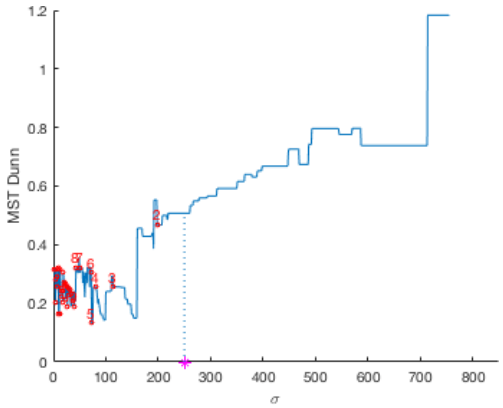
Figure A.4: Clusters of LM found by dominant sets on Gaussian kernel built using 1-Pearson correlation distances and $\sigma = 10.9$



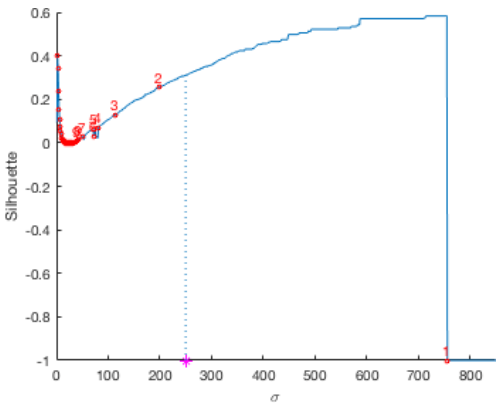
(a) Cluster number



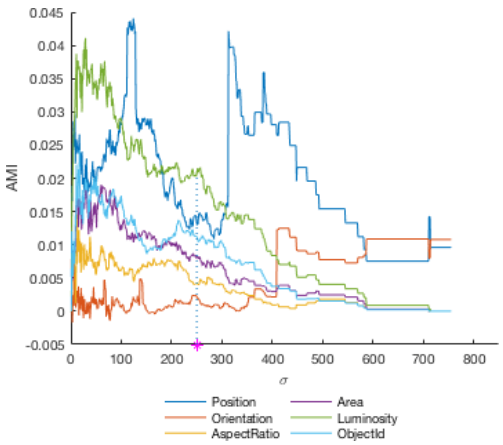
(b) Smallest cluster size



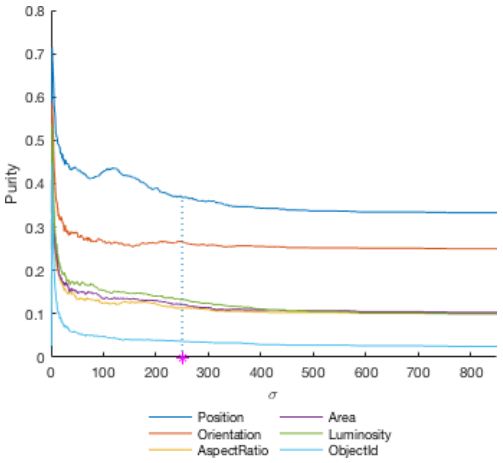
(c) MST Dunn's index



(d) Silhouette index

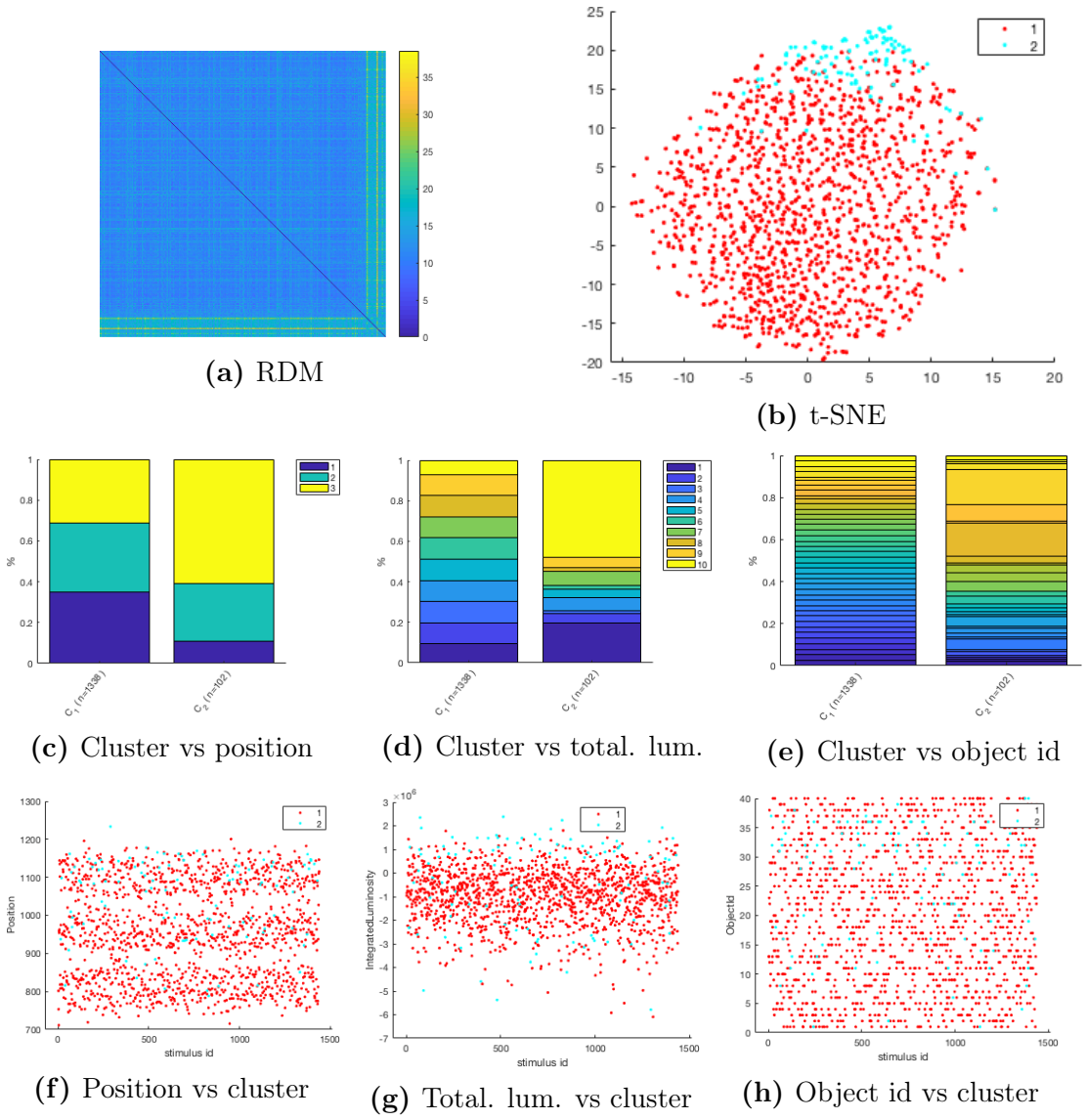


(e) AMI



(f) Purity

Figure A.5: Statistics on dominant sets algorithm applied on LI considering Gaussian kernel built using Euclidean distances

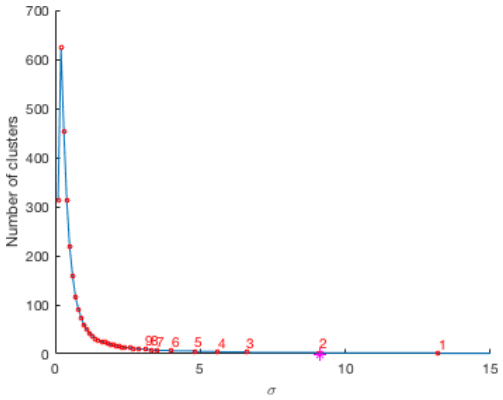


(a) RDM **(b)** t-SNE

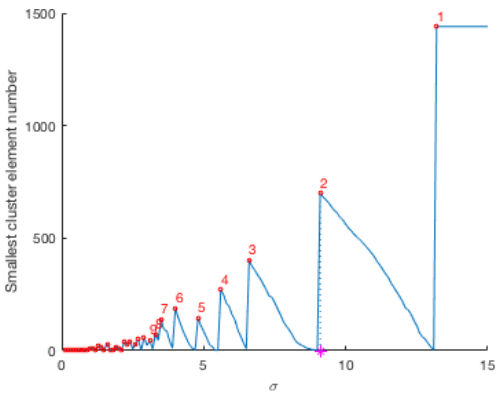
(c) Cluster vs position **(d)** Cluster vs total. lum. **(e)** Cluster vs object id

(f) Position vs cluster **(g)** Total. lum. vs cluster **(h)** Object id vs cluster

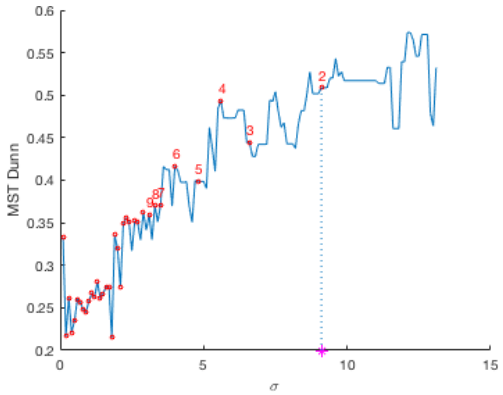
Figure A.6: Clusters of LI found by dominant sets on Gaussian kernel built using Euclidean distances and $\sigma = 250$



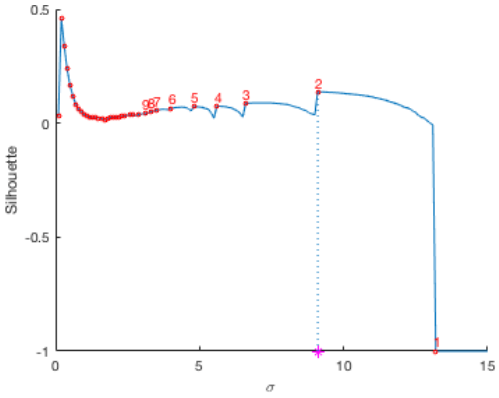
(a) Cluster number



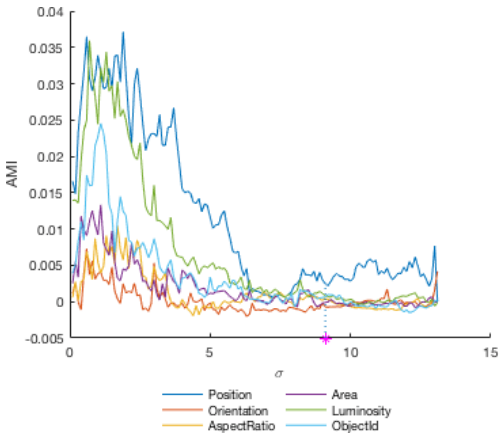
(b) Smallest cluster size



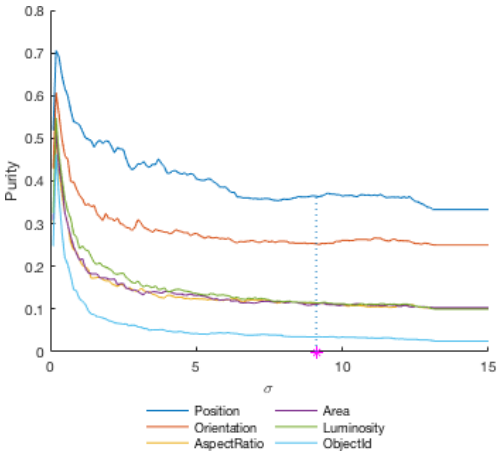
(c) MST Dunn's index



(d) Silhouette index

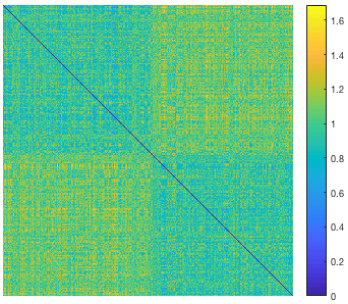


(e) AMI

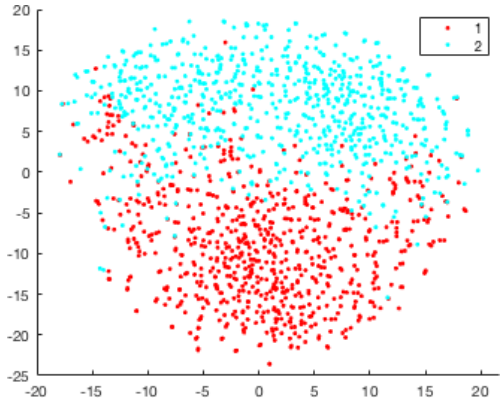


(f) Purity

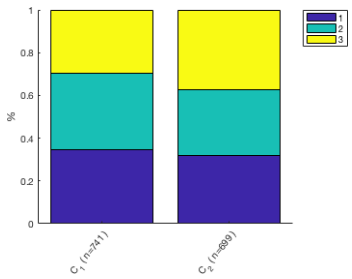
Figure A.7: Statistics on dominant sets algorithm applied on LI considering Gaussian kernel built using 1-Pearson correlation distances



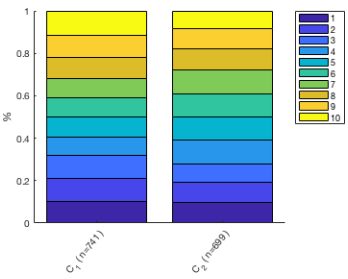
(a) RDM



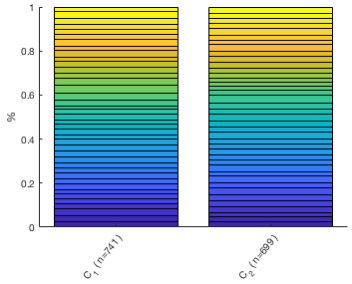
(b) t-SNE



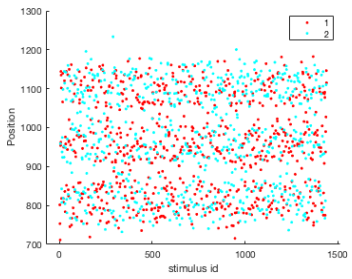
(c) Cluster vs position



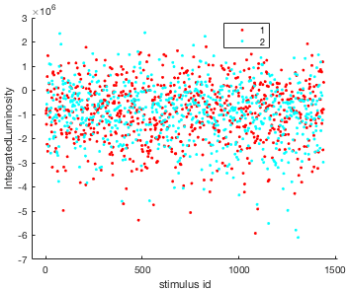
(d) Cluster vs total. lum.



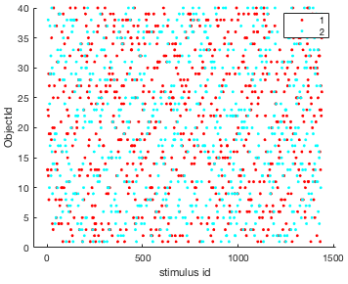
(e) Cluster vs object id



(f) Position vs cluster

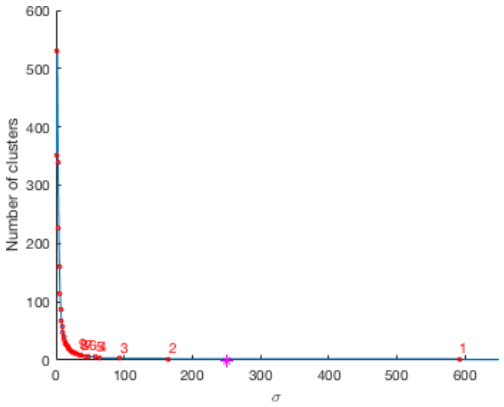


(g) Total. lum. vs cluster

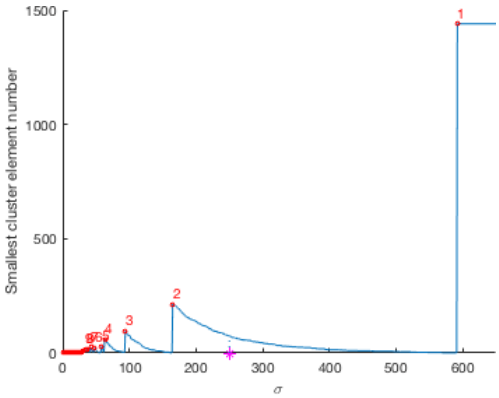


(h) Object id vs cluster

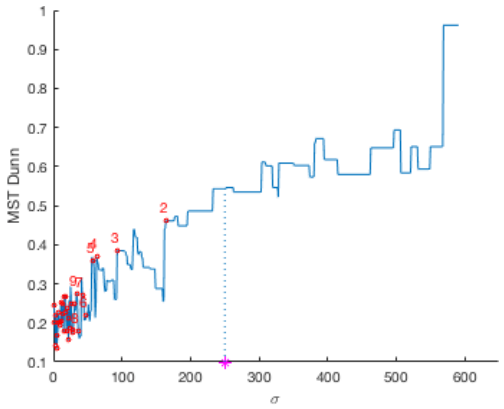
Figure A.8: Clusters of LI found by dominant sets on Gaussian kernel built using 1-Pearson correlation distances and $\sigma = 9.1$



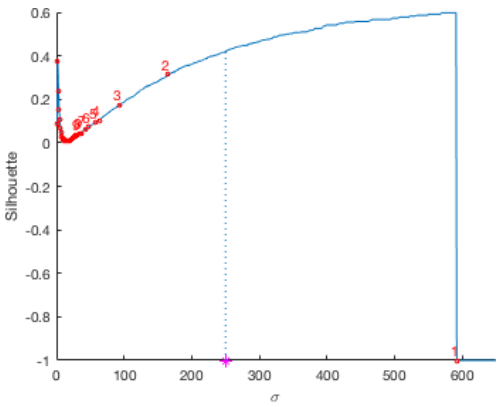
(a) Cluster number



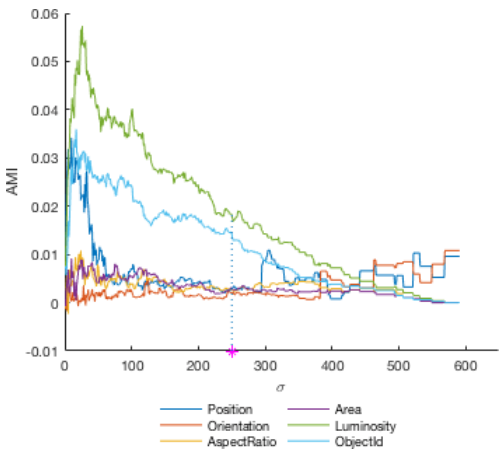
(b) Smallest cluster size



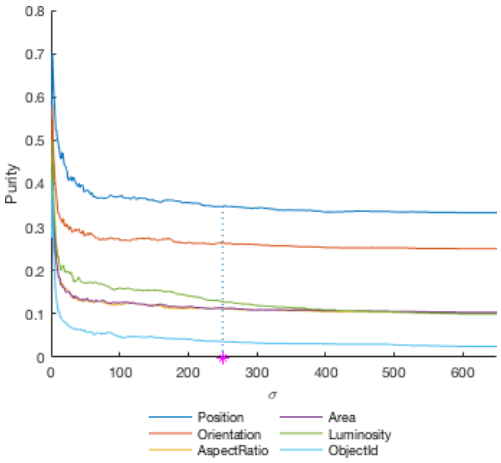
(c) MST Dunn's index



(d) Silhouette index



(e) AMI



(f) Purity

Figure A.9: Statistics on dominant sets algorithm applied on LL considering Gaussian kernel built using Euclidean distances

A.4 DS results on LI with similarities based on 1-Pearson correlation distance

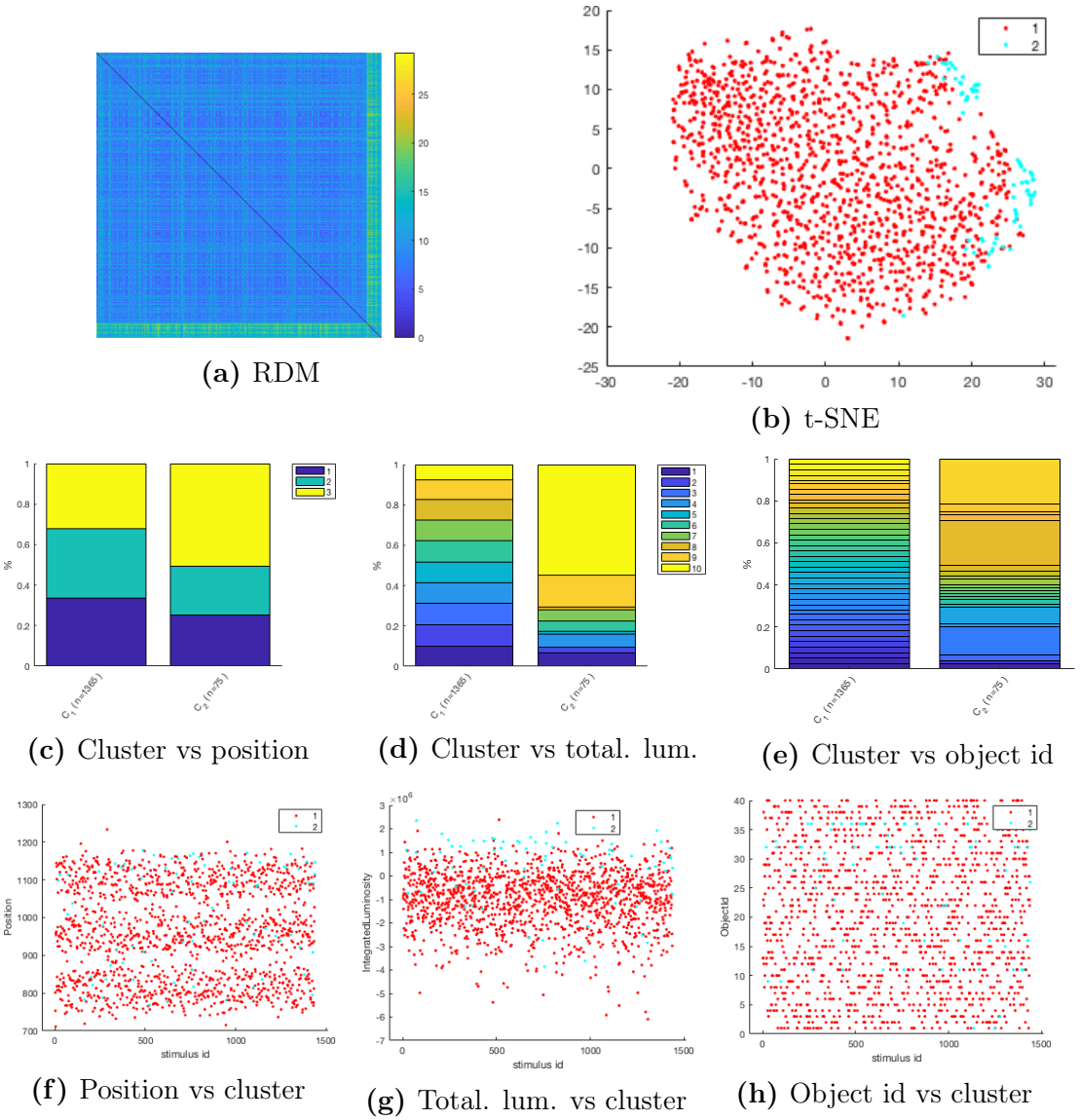
Figure A.7 shows statistics on results of dominant sets algorithm applied to level LI and based on the Gaussian kernel built using the 1-Pearson correlation distances. As happened in the case of level LM (with the same distance metric), the choice of σ is easy considering both internal indexes. Hence, $\sigma = 9.1$ is selected, which generates two bunches over data. However, Figures A.7e and A.7f, related to AMI and purity, show how weakly all features are encoded, with position reporting the most significative value ($AMI^c_D(LI, P, \sigma = 9.1) = 0.0025$). As expected, Figure A.8, which reports clusters composition, shows how specific feature values do not compose the two clusters, suggesting that no information about postition or luminosity is coded.

A.5 DS results on LL with similarities based on Euclidean distance

Figure A.9 reports statistics on results of dominant sets algorithm applied to level LL and based on the Gaussian kernel built using the Euclidean distances. As in the case of level V1, LM and LI, two families of stimuli appear. Following the same principles as in the other levels, $\sigma = 250$ was selected. Again observe how luminosity ($AMI^e_D(LL, Tl, \sigma = 250) = 0.0182$) and object identity ($AMI^e_D(LL, Obj, \sigma = 250) = 0.0142$) are more encoded than the other features (refer to Figures A.9e and A.9f). As happened for the level LI with Euclidean distances, the second cluster is smaller if compared with the first and contains just 75 stimuli. Moreover, it is composed again of the brightest objects (mainly the one with identifiers 32 and 36) and of a bigger fraction of "right" stimuli.

A.6 DS results on LL with similarities based on 1-Pearson correlation distance

Figure A.11 shows some statistics on results of dominant sets algorithm applied to level LL and based on the Gaussian kernel built using the 1-Pearson correlation distances. As happened for levels LM and LI, when dominant sets on correlation distances is used, internal indexes suggest similar values of σ. We precisely selected $\sigma = 10$, which produces again two clusters over data points. Consider Figures A.11e



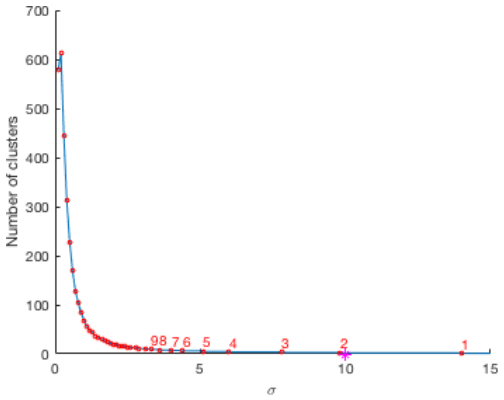
(a) RDM **(b)** t-SNE

(c) Cluster vs position **(d)** Cluster vs total. lum. **(e)** Cluster vs object id

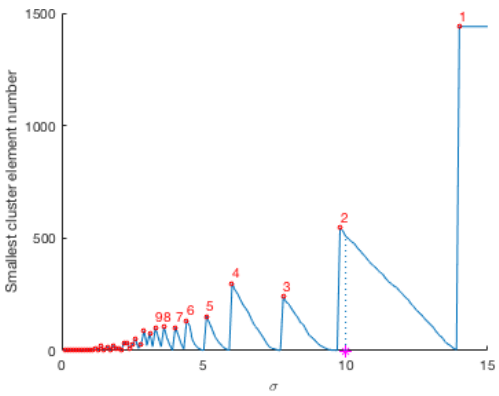
(f) Position vs cluster **(g)** Total. lum. vs cluster **(h)** Object id vs cluster

Figure A.10: Clusters of LL found by dominant sets on Gaussian kernel built using Euclidean distances and $\sigma = 250$

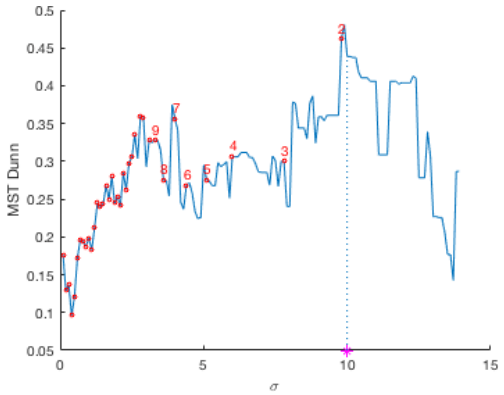
and A.11f, which report AMI and purity indexes. As happened for the previous levels, we expect that clusters reflect position feature. However, surprisingly, the two most encoded features are luminosity and, thus, object identity. As already discussed in Section 5.6, AMI and purity values are in the same order of magnitude of the ones related to clusters produced on level V1 using Euclidean distances. This indicates that correlation distances let stimulus luminosity quite well emerges. Moreover, this also means that not only Euclidean distances code for luminosity, suggesting that a combination of both distances could be useful to extract interesting clusters. Let's consider clusters composition in Figure A.12. Again two clusters were produced, respectively containing 933 and 507 stimuli. As expected, the second cluster contains a discrete quantity of light stimuli, even if it contains also many middle- and high-luminosity ones. Indeed, Figure A.12g shows that the boundary between high and low luminosity values is not particularly sharp.



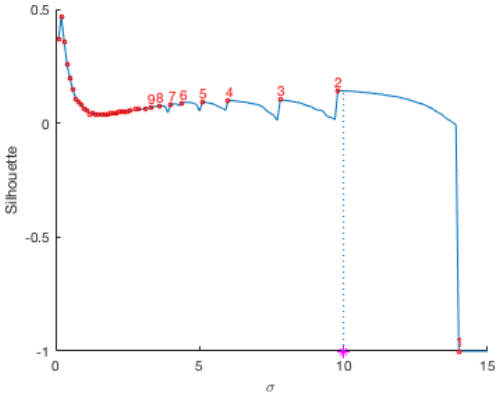
(a) Cluster number



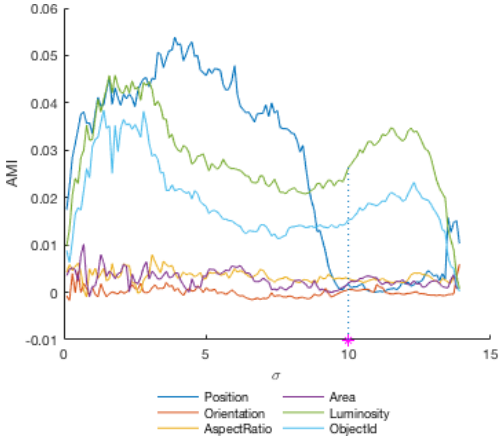
(b) Smallest cluster size



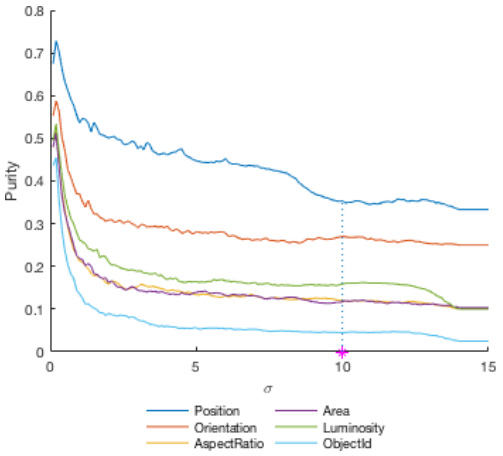
(c) MST Dunn's index



(d) Silhouette index

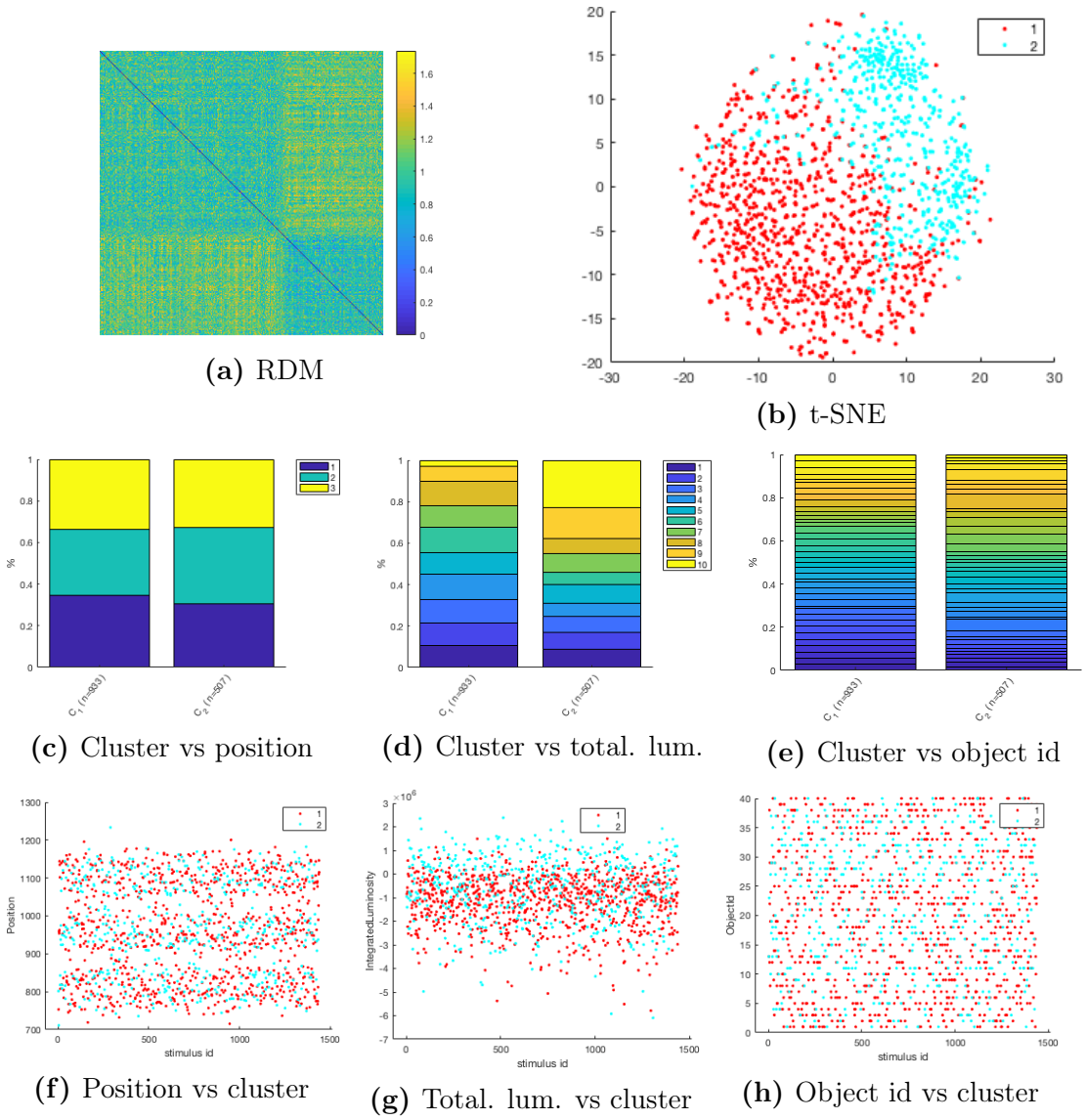


(e) AMI



(f) Purity

Figure A.11: Statistics on dominant sets algorithm applied on LL considering Gaussian kernel built using 1-Pearson correlation distances



(a) RDM **(b)** t-SNE

(c) Cluster vs position **(d)** Cluster vs total. lum. **(e)** Cluster vs object id

(f) Position vs cluster **(g)** Total. lum. vs cluster **(h)** Object id vs cluster

Figure A.12: Clusters of LL found by dominant sets on Gaussian kernel built using 1-Pearson correlation distances and $\sigma = 10$

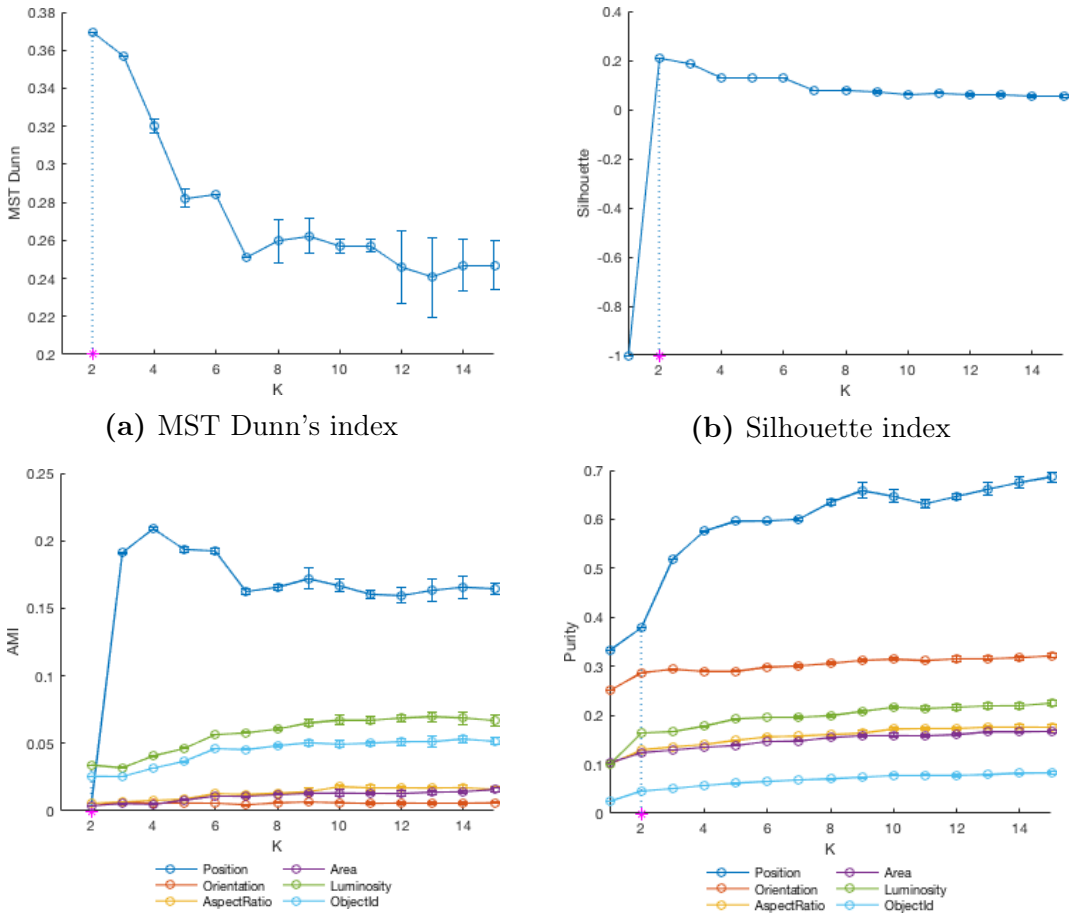
Appendix B

Kmeans on LM, LI and LL

In this section we discuss results of Kmeans algorithm on levels LM, LI and LL, which are omitted in the body of the thesis. The algorithm was executed twice, once for each distance measure. The performed analysis was the same already conducted for level V1. We firstly tuned the parameter K using both the internal indexes, Silhouette and MST Dunn's. Then, we analyzed AMI and purity indexes behavior as function of K. Once K was selected, we again investigated how clusters were composed to understand which features were encoded.

B.1 Kmeans results on LM with Euclidean distance

Figure B.1 shows internal and external indexes for kmeans applied to level LM with Euclidean distances. Both Silhouette and MST Dunn's indexes suggest to choose $K = 2$. The AMI index in Figure B.1c shows that luminosity and, thus, object identity are the most encoded features, with respectively $AMI^e_K(LM, Tl, K = 2) = 0.0342$ and $AMI^e_K(LM, Obj, K = 2) = 0.0257$, much higher than AMI of position ($AMI^e_K(LM, P, K = 2) = 0.006$). Figure B.2 shows clusters composition according to the three analyzed features. The algorithm produces two bunches of data in which one cluster is smaller and composed of stimuli dissimilar to all the others, while the second cluster is bigger and both highly cohesive and well-separated. As Figure B.2d shows, the first cluster is slightly majorly composed of light stimuli. However, it does not contain only such stimuli, making the boundary between high and middle/low-luminosity stimuli fuzzy (see Figure B.2g).



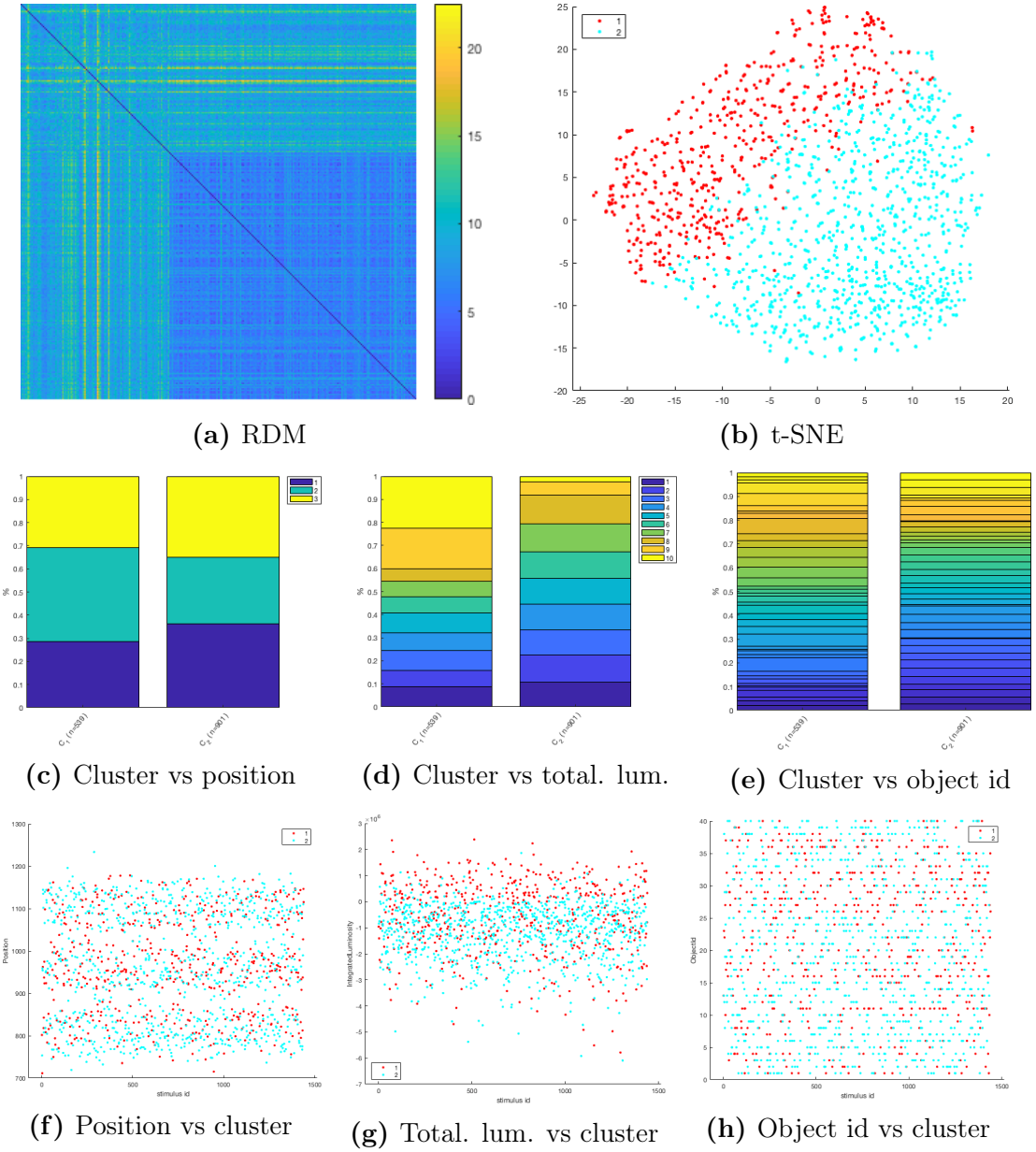
(a) MST Dunn's index

(b) Silhouette index

(c) AMI

(d) Purity

Figure B.1: Statistics on kmeans algorithm applied considering Euclidean distances on level LM. We run kmeans 10^3 times for each value of K, reporting average and standard deviation for all indexes.



(a) RDM **(b)** t-SNE

(c) Cluster vs position **(d)** Cluster vs total. lum. **(e)** Cluster vs object id

(f) Position vs cluster **(g)** Total. lum. vs cluster **(h)** Object id vs cluster

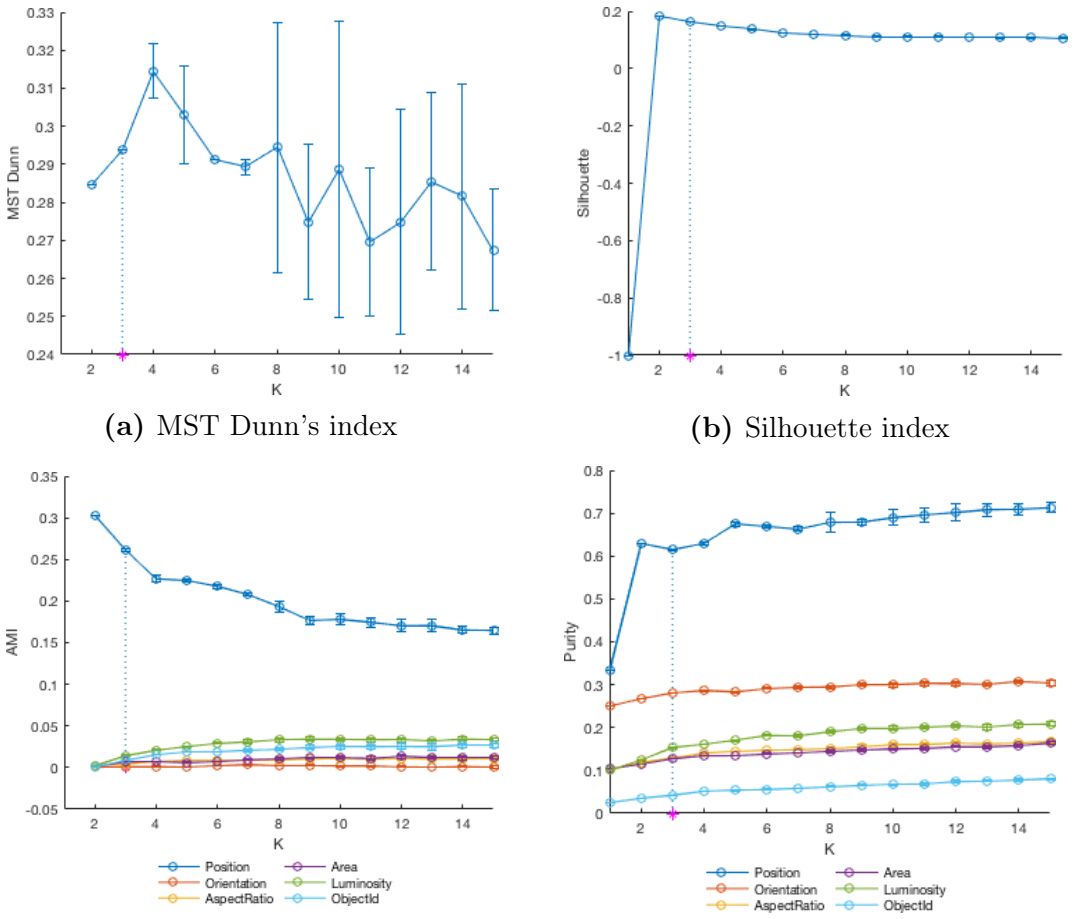
Figure B.2: Clusters of LM found with kmeans on Euclidean distances and $K = 2$

B.2 Kmeans results on LM with 1-Pearson correlation distance

Figure B.3 reports internal and external indexes for kmeans algorithm applied on level LM with 1-Pearson correlation distances. This time the selection of K is much more complicated. Indeed, the two indexes are not in agreement, with Silhouette suggesting $K = 2$ and MST Dunn's indexes suggesting $K = 4$. To overcome this problem we considered both the RDMs of all partitions and that Silhouette index is generally more reliable than MST Dunn's index when possible outliers are present. Finally, we chose $K = 3$, being its Silhouette index similar to the case $K = 2$, but much better on MST Dunn's index. Figure B.4a shows correlation RDM of level LM ordered according to cluster index. It suggests that the three clusters are all cohesive, with the middle one not perfectly separated from the other two. As suggested by Figures B.3c and B.3d, position is the most encoded feature when $K = 3$, reaching $AMI^c_K(LM, P, K = 3) = 0.2611$, marked with the pink asterisk. As expected, Figure B.4c, which shows clusters composition according to feature position, reports that the first cluster is mainly composed of "left" stimuli, while the third is composed of "right" stimuli. However, the second cluster does not clearly represent "center" stimuli, being almost composed of equal fractions of position bins. No cluster particularly represents specific luminosity values.

B.3 Kmeans results on LI with Euclidean distance

Figure B.5 reports external and internal indexes related to kmeans algorithm applied on level LI with the Euclidean distance. Also in that case, both internal indexes suggest clearly to choose $K = 2$. However, as already observed with dominant sets algorithm, this is one of the levels which less encodes feature. Figure B.5c shows AMIs for the analyzed features. It reports that position is the most encoded feature, but the index does not assume particularly higher values if compared with the other features. For example, AMI of position and luminosity are respectively $AMI^e_K(LI, P, K = 2) = 0.0167$ and $AMI^e_K(LI, Tl, K = 2) = 0.0094$. Cluster composition analysis confirms this, as reported in Figure B.6. Notice how both clusters are not composed of specific stimuli. Only the first cluster shows that it contains a bigger fraction of "right" and bright stimuli. However, it is not substantial.



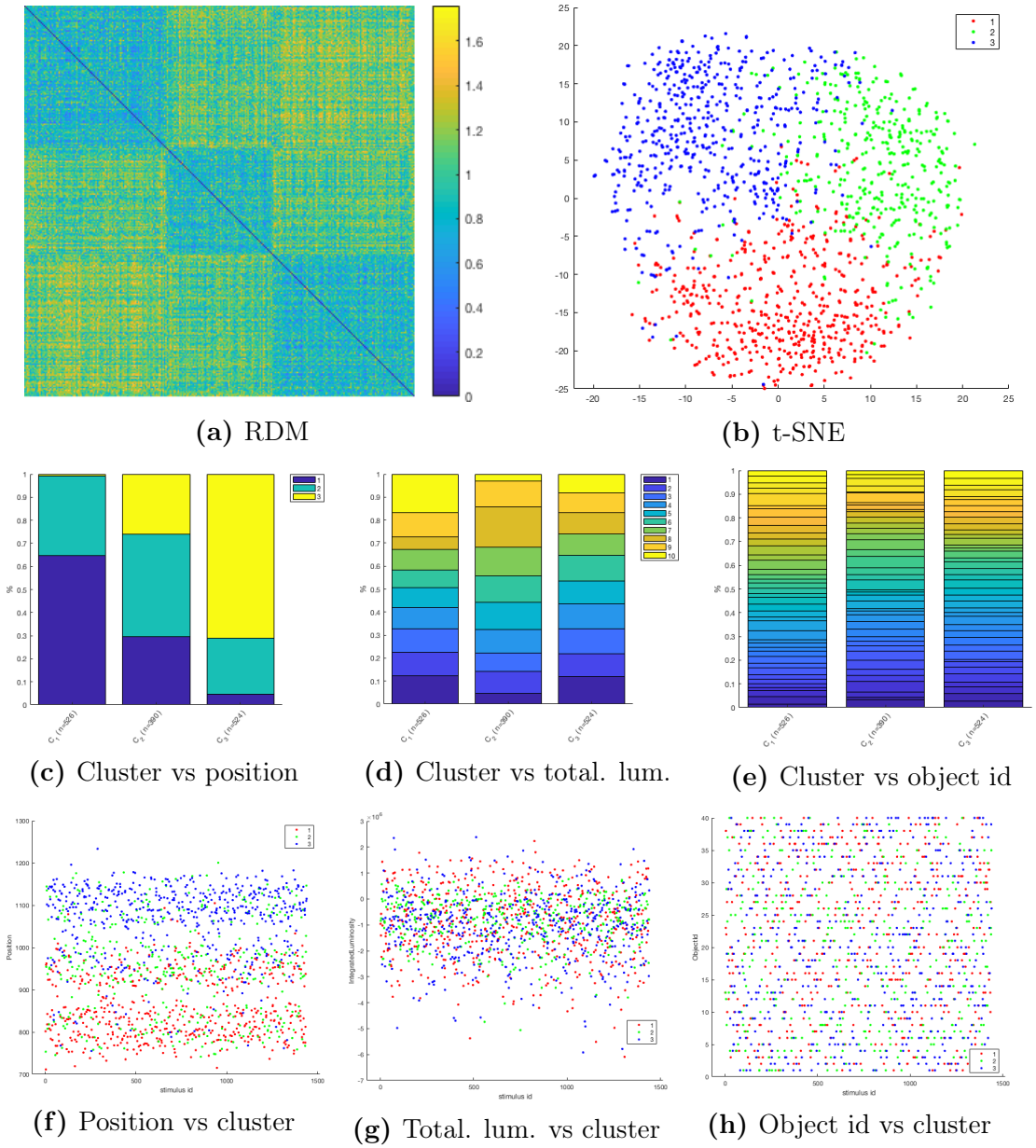
(a) MST Dunn's index

(b) Silhouette index

(c) AMI

(d) Purity

Figure B.3: Statistics on kmeans algorithm applied considering 1-Pearson correlation distances on level LM. We run kmeans 10 times for each value of K, reporting average and standard deviation for all indexes.

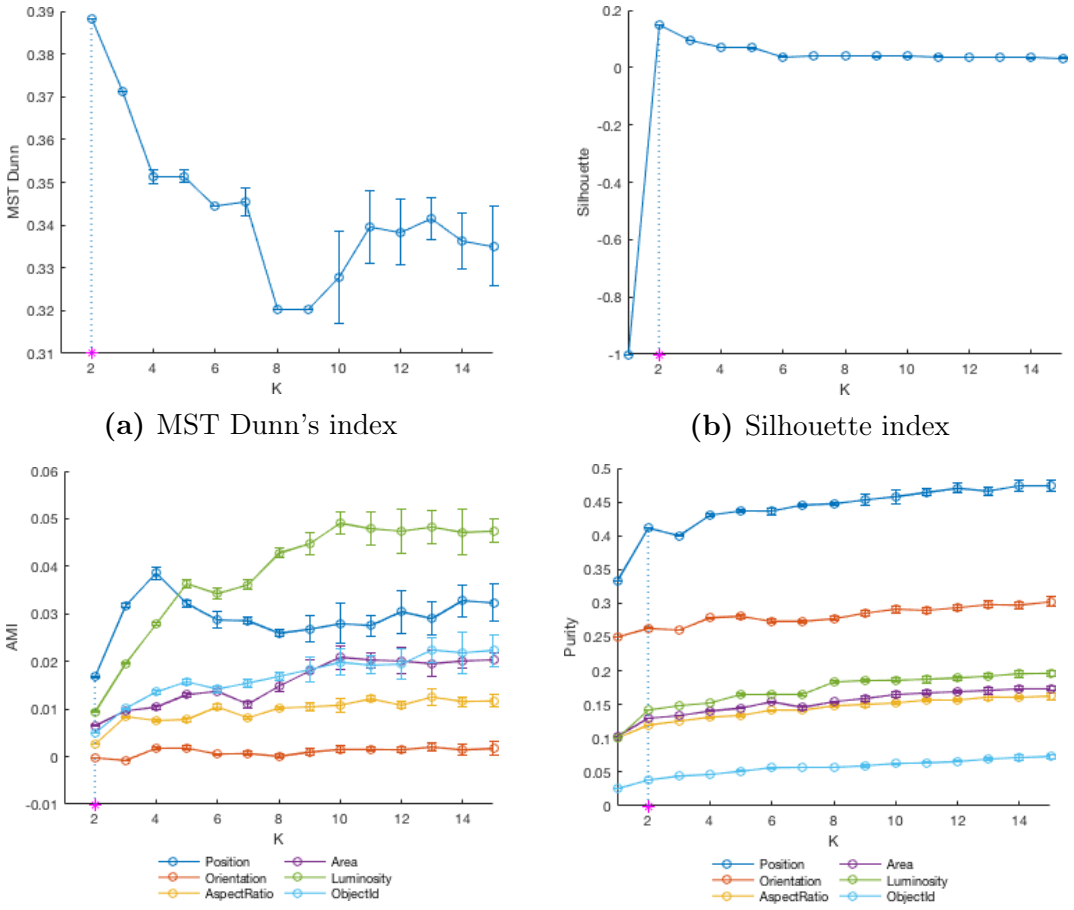


(a) RDM **(b)** t-SNE

(c) Cluster vs position **(d)** Cluster vs total. lum. **(e)** Cluster vs object id

(f) Position vs cluster **(g)** Total. lum. vs cluster **(h)** Object id vs cluster

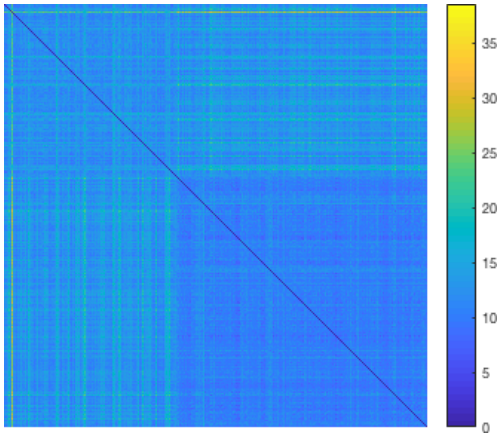
Figure B.4: Clusters of LM found with kmeans on 1-Pearson correlation distances and $K = 3$



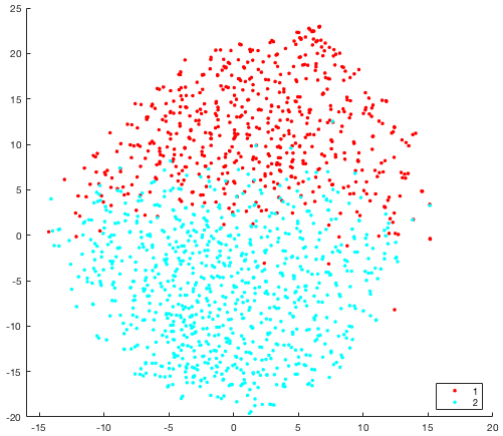
(a) MST Dunn's index **(b)** Silhouette index

(c) AMI **(d)** Purity

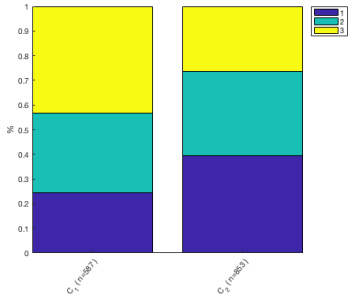
Figure B.5: Statistics on kmeans algorithm applied considering Euclidean distances on level LI. We run kmeans 10 times for each value of K, reporting average and standard deviation for all indexes.



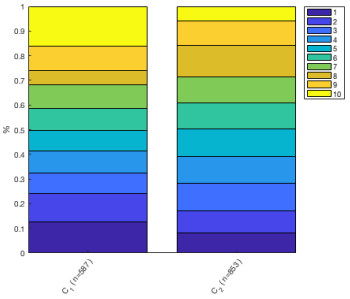
(a) RDM



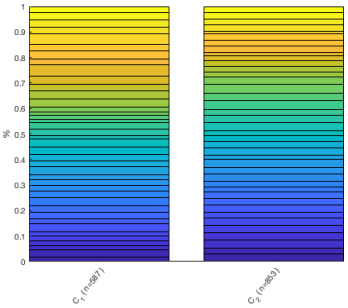
(b) t-SNE



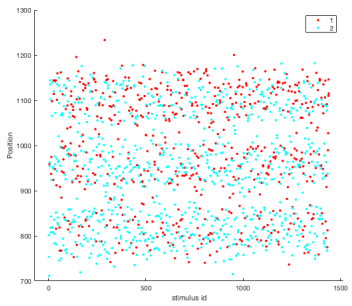
(c) Cluster vs position



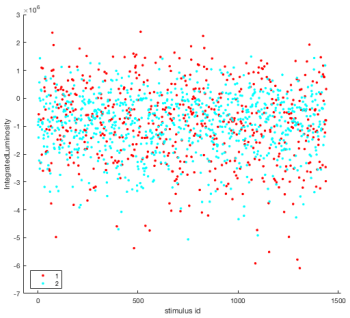
(d) Cluster vs total. lum.



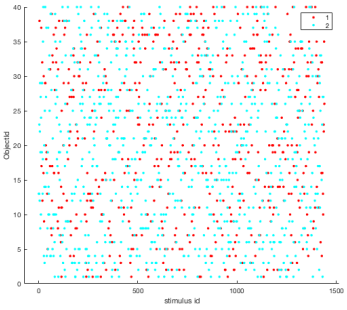
(e) Cluster vs object id



(f) Position vs cluster

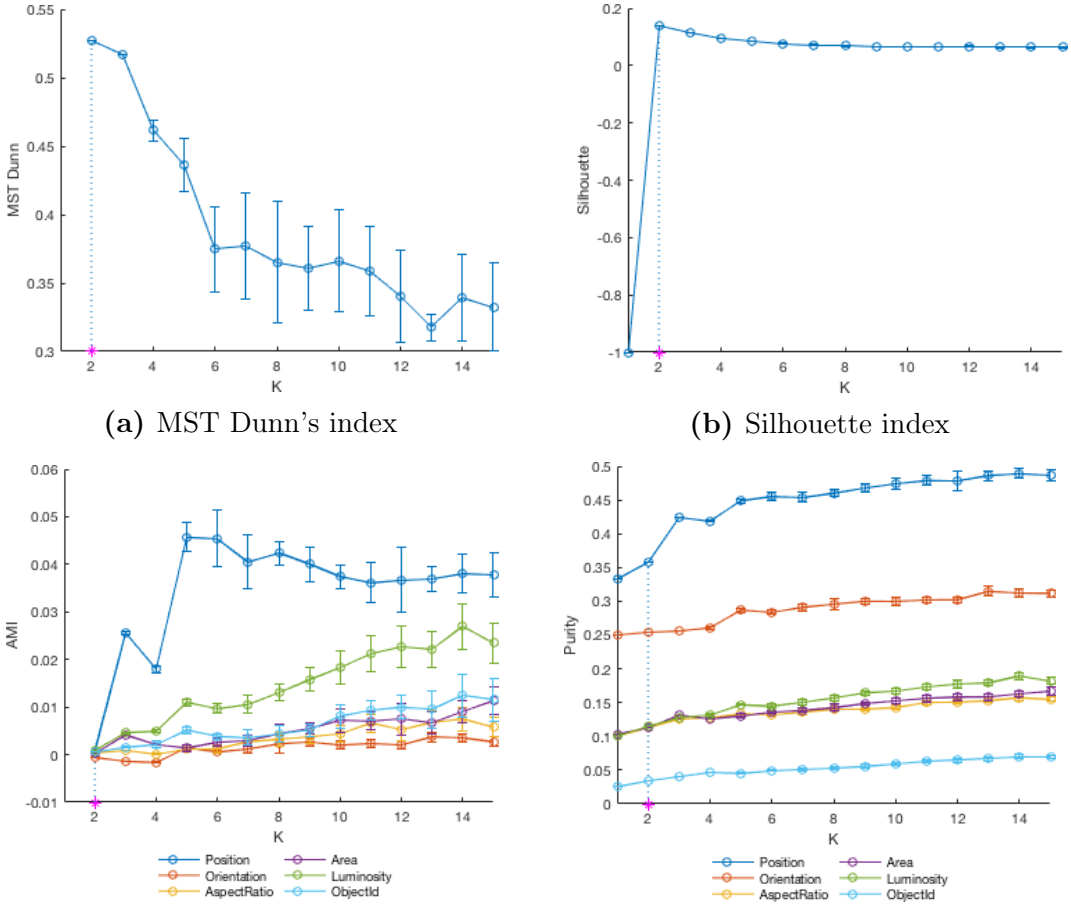


(g) Total. lum. vs cluster



(h) Object id vs cluster

Figure B.6: Clusters of LI found with kmeans on Euclidean distances and $K = 2$



(a) MST Dunn's index **(b)** Silhouette index

(c) AMI **(d)** Purity

Figure B.7: Statistics on kmeans algorithm applied considering 1-Pearson correlation distances on level LI. We run kmeans 10 times for each value of K, reporting average and standard deviation for all indexes.

B.4 Kmeans results on LI with 1-Pearson correlation distance

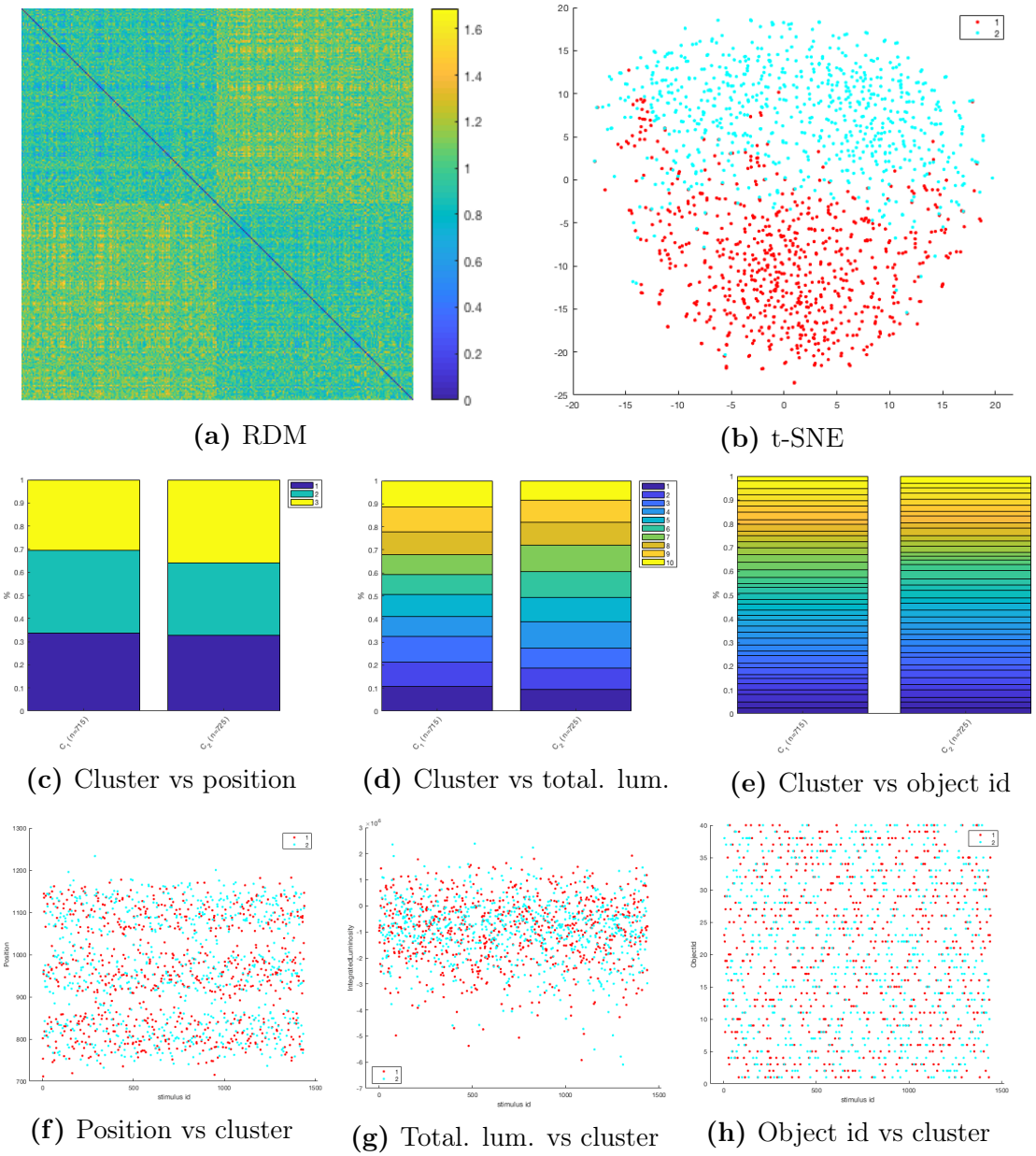
Figure B.7 reports statistics related to kmeans algorithm applied on level LI with the 1-Pearson correlation distance. Both indexes are maximzed on $K = 2$, making obvious its choice. Figure B.7c shows AMI index as K varies. When $K = 2$, marked with the pink asterisk, the AMI values of all features is approximately 0, indicating that no feature is encoded. Figure B.8, which shows clusters composition, report that no cluster represents specific stimuli.

B.5 Kmeans results on LL with Euclidean distance

Figure B.9 reports external and internal indexes related to kmeans algorithm applied on level LL with the Euclidean distance. Both internal indexes suggest to choose $K = 2$. Again, as almost always happened with Euclidean distances, luminosity is the most encoded features, as Figure B.9c reported. Figure B.10 shows clusters composition according to position, luminosity and object identifiers. The first cluster, which contains dissimilar stimuli to all the others, is composed of a bigger fraction of "right" and bright stimuli. However, they also contains many other stimuli, of each position and luminosity bunch. The second cluster, highly cohesive and well-separated from the first, is equally composed of all position and luminosity bins.

B.6 Kmeans results on LL with 1-Pearson correlation distance

Figure B.11 reports statistics related to kmeans algorithm applied on level LL with the 1-Pearson correlation distance. Let's select $K = 2$ because it is the value that maximized the Silhouette index and the second highest of MST Dunn's. Consider now Figure B.11c, which reports AMI as K varies. Surprisingly, as happened with dominant sets applied to the Gaussian kernel built on correlation distances, luminosity is the majorly encoded features, with $AMI^c_K(LL, Tl, K = 2) = 0.0183$. However, there is no cluster which strongly codes for luminosity. Consider Figure B.12, which reports clusters composition. Only a small fraction of the first cluster consists of brightest stimuli. Moreover, as expected no cluster represents particular position bins.

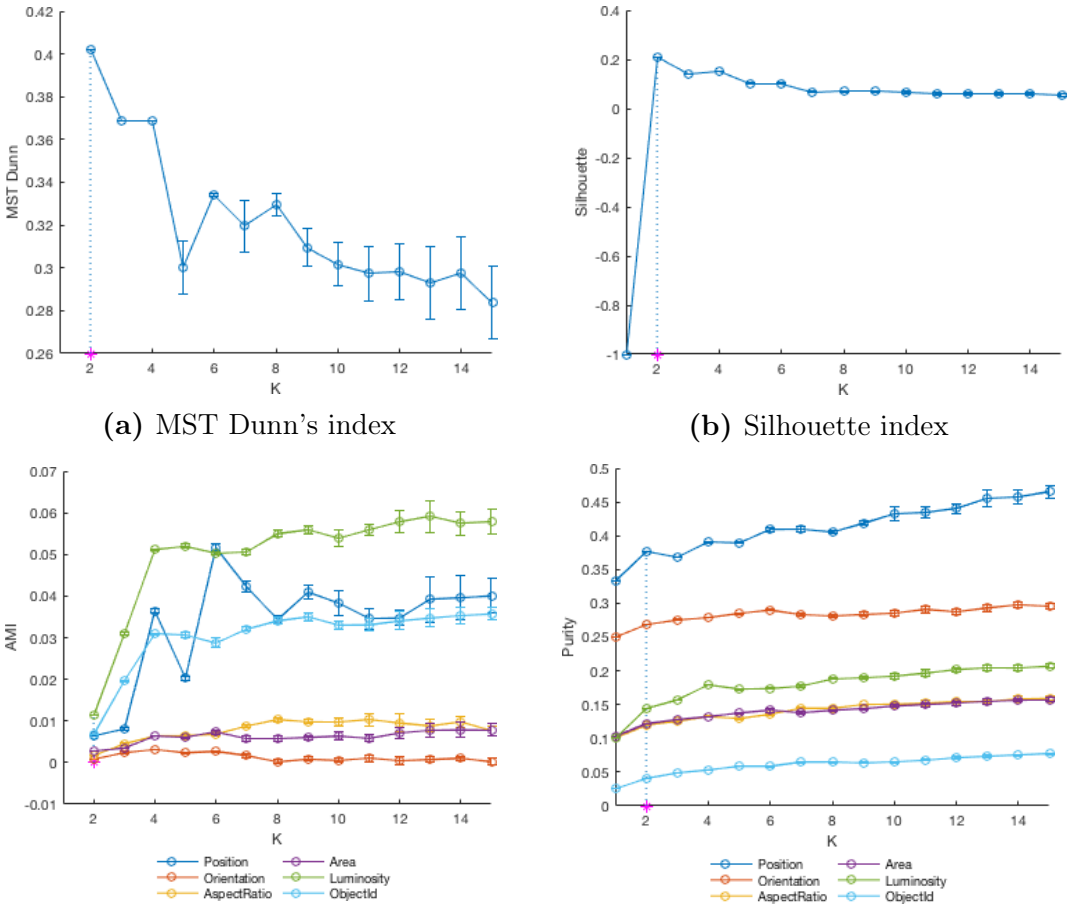


(a) RDM **(b)** t-SNE

(c) Cluster vs position **(d)** Cluster vs total. lum. **(e)** Cluster vs object id

(f) Position vs cluster **(g)** Total. lum. vs cluster **(h)** Object id vs cluster

Figure B.8: Clusters of LI found with kmeans on 1-Pearson correlation distances and $K = 2$



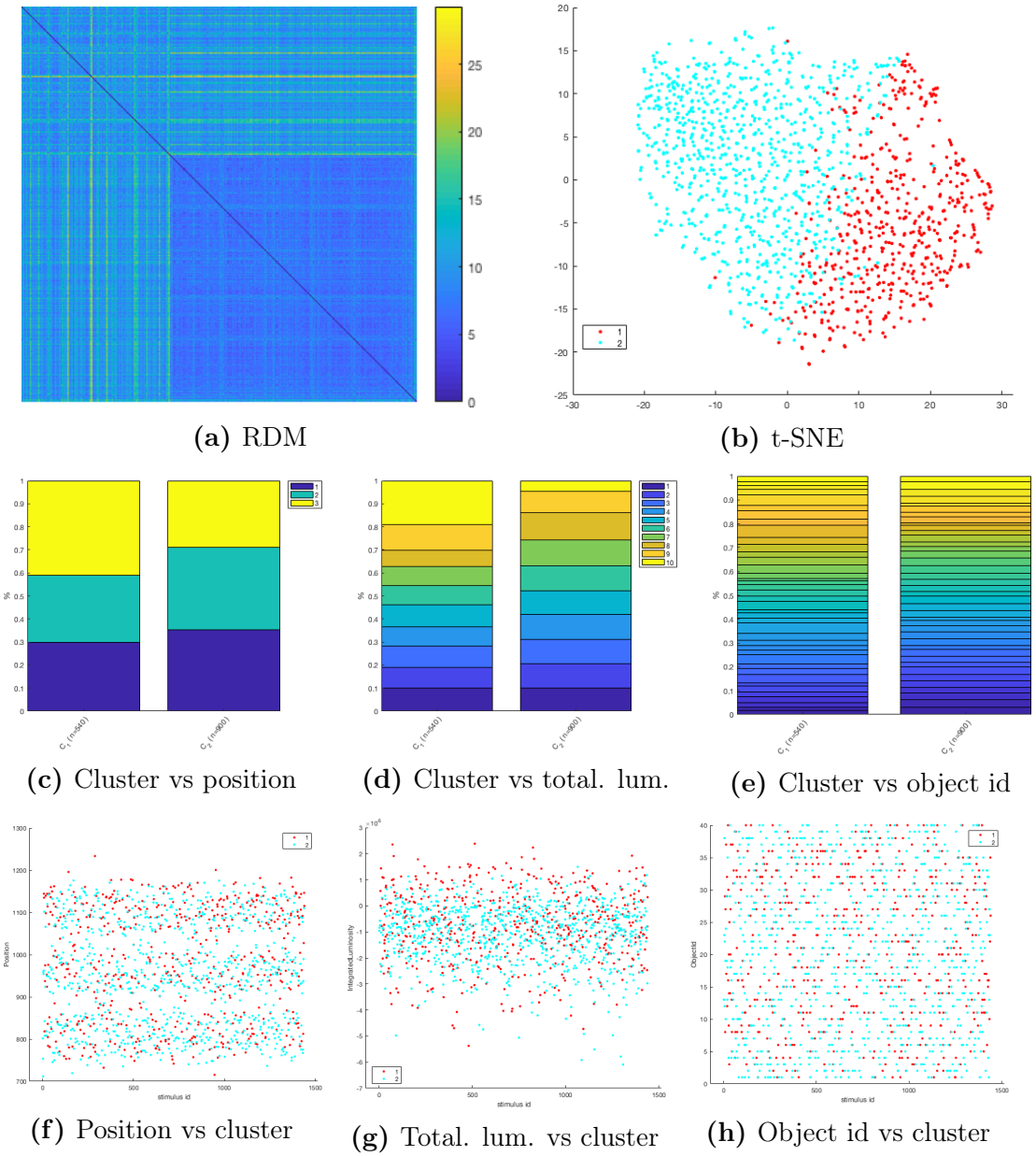
(a) MST Dunn's index

(b) Silhouette index

(c) AMI

(d) Purity

Figure B.9: Statistics on kmeans algorithm applied considering Euclidean distances on level LL. We run kmeans 10 times for each value of K, reporting average and standard deviation for all indexes.

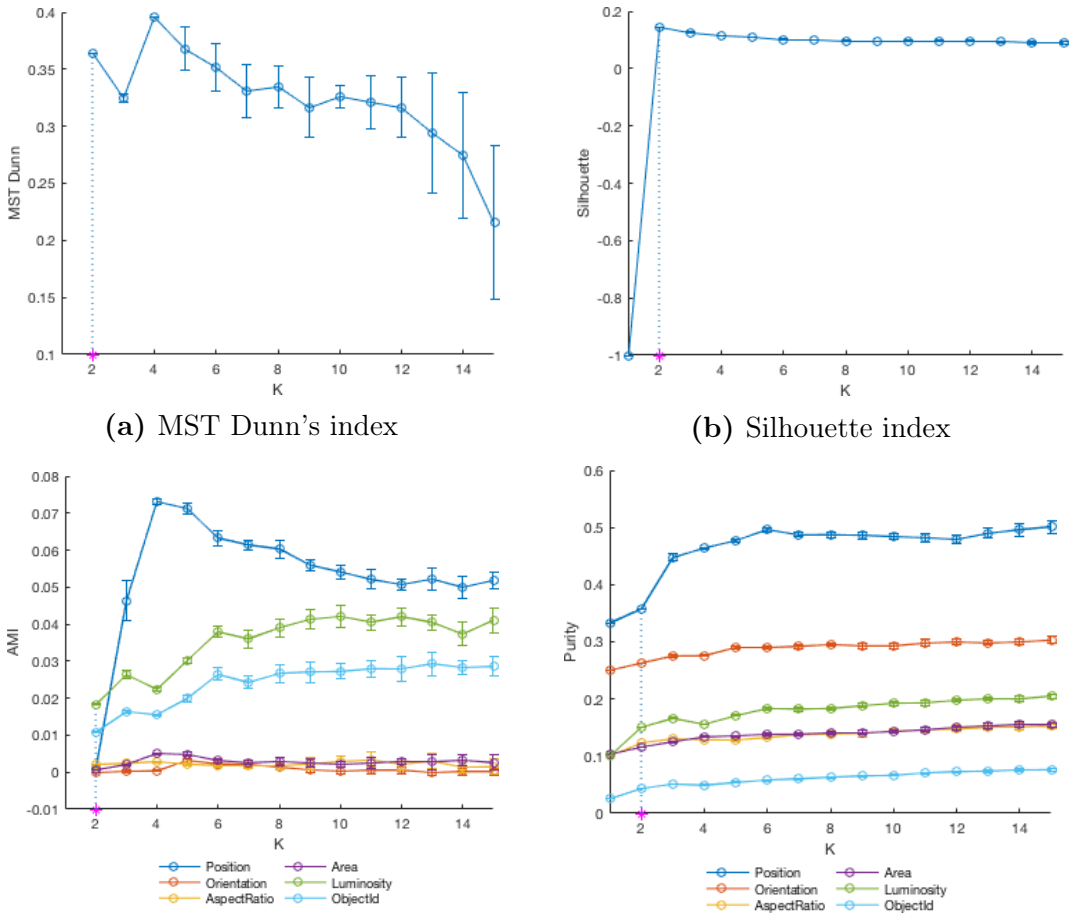


(a) RDM **(b)** t-SNE

(c) Cluster vs position **(d)** Cluster vs total. lum. **(e)** Cluster vs object id

(f) Position vs cluster **(g)** Total. lum. vs cluster **(h)** Object id vs cluster

Figure B.10: Clusters of LL found with kmeans on Euclidean distances and $K = 2$



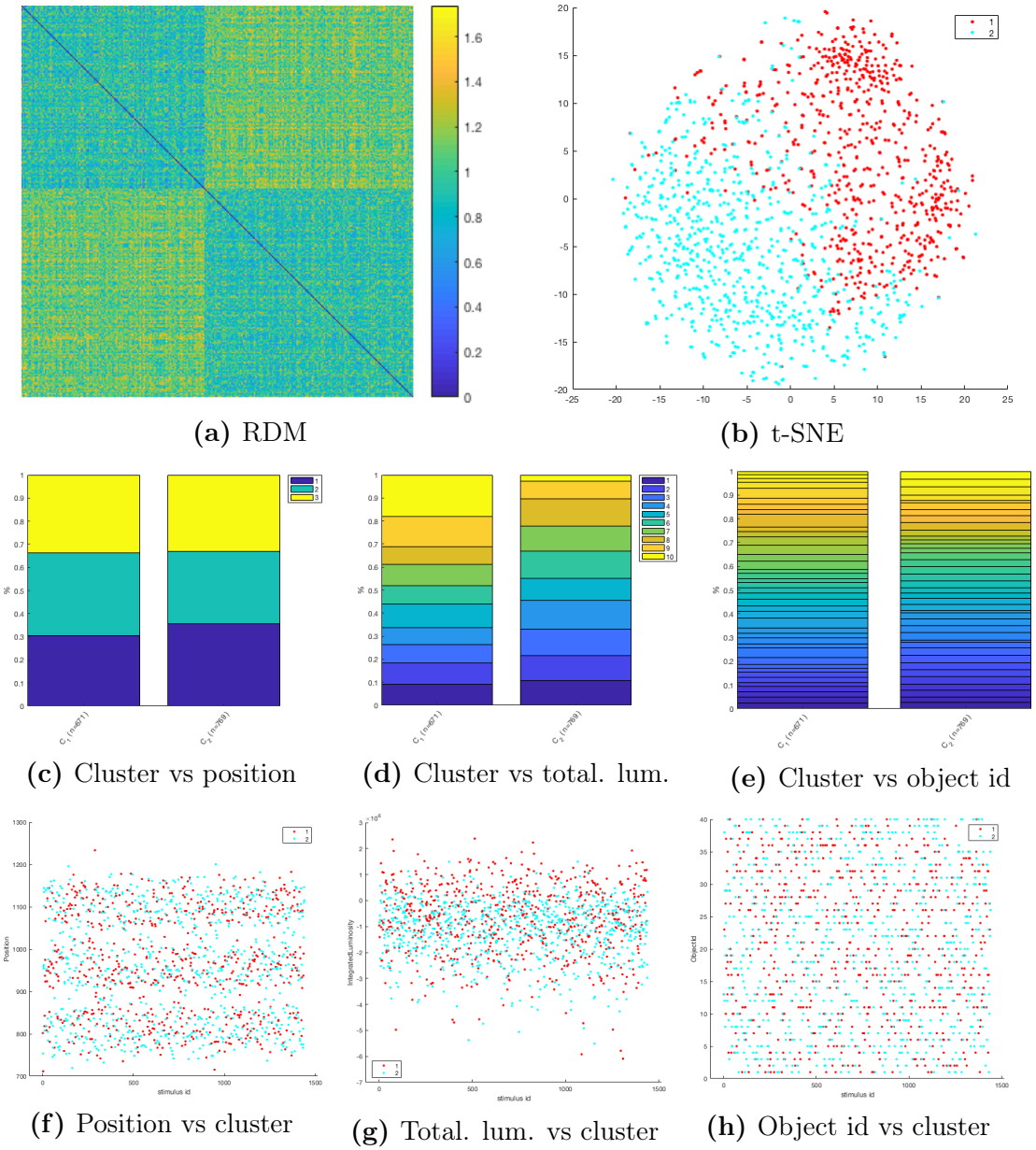
(a) MST Dunn's index

(b) Silhouette index

(c) AMI

(d) Purity

Figure B.11: Statistics on kmeans algorithm applied considering 1-Pearson correlation distances on level LL. We run kmeans 10 times for each value of K, reporting average and standard deviation for all indexes.



(a) RDM **(b)** t-SNE

(c) Cluster vs position **(d)** Cluster vs total. lum. **(e)** Cluster vs object id

(f) Position vs cluster **(g)** Total. lum. vs cluster **(h)** Object id vs cluster

Figure B.12: Clusters of LL found with kmeans on 1-Pearson correlation distances and $K = 2$

Acknowledgments

I would like to thank my supervisor, Professor Marcello Pelillo, and researcher Sebastiano Vascon for their invaluable support throughout this thesis. Moreover, this project would have been impossible without the support of Professor Davide Zoccolan, and researchers Eis Annavini and Mattia D'Andola, who kindly granted the datasets I used and helped me making the neurobiology world a little bit less of a mistery. A huge thank goes to my parents Eros and Nicoletta, and to my sister Alessia. A special thank you goes to Silvia and Giulia: they have always been my support during these difficult and often exhausting months. The most important thank you goes to Davide, who has been my backbone in the last few years and even more in recent months, encouraging me to never give up and enduring me in all those weekends completely dedicated to the study.

List of Figures

2.1 Example of discrimination of a visual object (a car) subject to identity-preserving transformations 6
2.2 Visual representation of the same object on different spatial complexity 6
2.3 Dorsal and ventral pathways in visual system of macaque brain . . 7
2.4 Lateral and anterior pathways in visual system of mouse brain . . . 8

3.1 Tree structure of object models used to build the stimulus set . . . 10
3.2 Histograms of stimuli features 12

5.1 Mutual information between neural responses and stimulus position 42
5.2 Mutual information between neural responses and stimulus total luminosity 42
5.3 Mutual information between neural responses and stimulus orientation 43
5.4 Mutual information between neural responses and object identity . 43
5.5 Conditional mutual information between neural responses and object identity fixing position 44
5.6 Conditional mutual information between neural responses and object identity fixing total luminosity 44
5.7 Euclidean RDMs ordered by position 46
5.8 Correlation RDMs ordered by position 46
5.9 Euclidean RDMs ordered by total luminosity 46
5.10 Correlation RDMs ordered by total luminosity 47
5.11 Euclidean RDMs ordered by object identity 47
5.12 Correlation RDMs ordered by object identity 48
5.13 Recognized objects from RDM ordered by object identifier 48
5.14 Mean object total luminosity 48
5.15 t-SNE two-dimensional maps created on Euclidean distances 50
5.16 t-SNE two-dimensional maps created on 1-Pearson correlation distances 51
5.17 Statistics on dominant sets algorithm applied on V1 considering Gaussian kernel built using Euclidean distances 53

5.18 Euclidean RDMs of level V1 ordered according to dominant sets partitions computed on Gaussian kernel built using Euclidean distances and $246 \leq \sigma \leq 836$ 54
5.19 Clusters of V1 found by dominant sets on Gaussian kernel built using Euclidean distances and $\sigma = 336$ 56
5.20 Statistics on dominant sets algorithm applied on V1 considering Gaussian kernel built using 1-Pearson correlation distances 59
5.21 Clusters of V1 found by dominant sets on Gaussian kernel built using 1-Pearson correlation distances and $\sigma = 7.8$ 60
5.22 Statistics on kmeans algorithm applied considering Euclidean distances on level V1 63
5.23 Euclidean RDMs of level V1 ordered according to kmeans partitions computed with Euclidean distances on $1 \leq K \leq 15$ 64
5.24 Clusters of V1 found with kmeans on Euclidean distances and $K = 2$ 66
5.25 Statistics on kmeans algorithm applied considering 1-Pearson correlation distances on level V1 68
5.26 Correlation RDMs of level V1 ordered according to kmeans partitions computed with 1-Pearson correlation distances on $1 \leq K \leq 15$ . . . 69
5.27 Clusters of V1 found with kmeans on 1-Pearson correlation distances and $K = 3$ 70

A.1 Statistics on dominant sets algorithm applied on LM considering Gaussian kernel built using Euclidean distances 79
A.2 Clusters of LM found by dominant sets on Gaussian kernel built using Euclidean distances and $\sigma = 170$ 80
A.3 Statistics on dominant sets algorithm applied on LM considering Gaussian kernel built using 1-Pearson correlation distances 81
A.4 Clusters of LM found by dominant sets on Gaussian kernel built using 1-Pearson correlation distances and $\sigma = 10.9$ 82
A.5 Statistics on dominant sets algorithm applied on LI considering Gaussian kernel built using Euclidean distances 83
A.6 Clusters of LI found by dominant sets on Gaussian kernel built using Euclidean distances and $\sigma = 250$ 84
A.7 Statistics on dominant sets algorithm applied on LI considering Gaussian kernel built using 1-Pearson correlation distances 85
A.8 Clusters of LI found by dominant sets on Gaussian kernel built using 1-Pearson correlation distances and $\sigma = 9.1$ 86

A.9 Statistics on dominant sets algorithm applied on LL considering Gaussian kernel built using Euclidean distances 87
A.10 Clusters of LL found by dominant sets on Gaussian kernel built using Euclidean distances and $\sigma = 250$ 89
A.11 Statistics on dominant sets algorithm applied on LL considering Gaussian kernel built using 1-Pearson correlation distances 91
A.12 Clusters of LL found by dominant sets on Gaussian kernel built using 1-Pearson correlation distances and $\sigma = 10$ 92

B.1 Statistics on kmeans algorithm applied considering Euclidean distances on level LM 94
B.2 Clusters of LM found with kmeans on Euclidean distances and $K = 2$ 95
B.3 Statistics on kmeans algorithm applied considering 1-Pearson correlation distances on level LM 97
B.4 Clusters of LM found with kmeans on 1-Pearson correlation distances and $K = 3$ 98
B.5 Statistics on kmeans algorithm applied considering Euclidean distances on level LI 99
B.6 Clusters of LI found with kmeans on Euclidean distances and $K = 2$ 100
B.7 Statistics on kmeans algorithm applied considering 1-Pearson correlation distances on level LI 101
B.8 Clusters of LI found with kmeans on 1-Pearson correlation distances and $K = 2$ 103
B.9 Statistics on kmeans algorithm applied considering Euclidean distances on level LL 104
B.10 Clusters of LL found with kmeans on Euclidean distances and $K = 2$ 105
B.11 Statistics on kmeans algorithm applied considering 1-Pearson correlation distances on level LL 106
B.12 Clusters of LL found with kmeans on 1-Pearson correlation distances and $K = 2$ 107

Bibliography

[1] E. Annavini, "Functional evidence of hierarchical object processing in rat lateral extrastriate cortex", PhD thesis, SISSA, Trieste, Italy, 2018.

[2] C. Baldassi, A. Alemi-Neissi, M. Pagan, J. J. DiCarlo, R. Zecchina, and D. Zoccolan, "Shape Similarity, Better than Semantic Membership, Accounts for the Structure of Visual Object Representations in a Population of Monkey Inferotemporal Neurons", *PLOS Computational Biology*, vol. 9, no. 8, pp. 1–21, 2013.

[3] M. J. Berry and G. Linoff, *Data Mining Techniques: For Marketing, Sales, and Customer Support*. New York, NY, USA: John Wiley & Sons, Inc., 1997.

[4] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[5] I. M. Bomze, "On Standard Quadratic Optimization Problems", *Journal of Global Optimization*, vol. 13, no. 4, pp. 369–387, 1998.

[6] A. Borst and F. E. Theunissen, "Information theory and neural coding", *Nature Neuroscience*, vol. 2, no. 11, pp. 947–957, 1999.

[7] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*, 1st. Cambridge, MA: The MIT Press, 2006.

[8] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY, USA: John Wiley & Sons, Inc., 2006.

[9] J. J. DiCarlo and D. D. Cox, "Untangling invariant object recognition", *Trends in Cognitive Sciences*, vol. 11, no. 8, pp. 333–341, 2007.

[10] J. J. DiCarlo, D. Zoccolan, and N. C. Rust, "How does the brain solve visual object recognition?", *Neuron*, vol. 73, no. 3, pp. 415–434, 2012.

[11] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd. New York, NY, USA: John Wiley & Sons, Inc., 2001.

[12] J. C. Dunn, "Well-Separated Clusters and Optimal Fuzzy Partitions", *Journal of Cybernetics*, vol. 4, no. 1, pp. 95–104, 1974.

[13] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, *Advances in Knowledge Discovery and Data Mining*. Menlo Park, CA, USA: American Association for Artificial Intelligence, 1996.

[14] J. D. Gibbons and S. Chakraborti, *Nonparametric Statistical Inference*, 5th ed. Chapman and Hall, CRC, 2010.

[15] L. L. Glickfeld and S. R. Olsen, "Higher-Order Areas of the Mouse Visual Cortex", *Annual Review of Vision Science*, vol. 3, no. 1, pp. 251–273, 2017.

[16] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On Clustering Validation Techniques", *Journal of Intelligent Information Systems*, vol. 17, no. 2-3, pp. 107–145, 2001.

[17] J. Hofbauer and K. Sigmund, *Evolutionary Games and Population Dynamics*. Cambridge: Cambridge University Press, 1998.

[18] S. Holm, "A Simple Sequentially Rejective Multiple Test Procedure", *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65–70, 1979.

[19] R. A. A. Ince, A. Mazzoni, A. Bartels, N. K. Logothetis, and S. Panzeri, "A novel test to determine the significance of neural selectivity to single and multiple potentially correlated stimulus features", *Journal of Neuroscience Methods*, vol. 210, no. 1, pp. 49–65, 2012.

[20] A. K. Jain, "Data Clustering: 50 Years Beyond K-means", *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2009.

[21] R. Kiani, H. Esteky, K. Mirpour, and K. Tanaka, "Object Category Structure in Response Patterns of Neuronal Population in Monkey Inferior Temporal Cortex", *Journal of Neurophysiology*, vol. 97, no. 6, pp. 4296–4309, 2007.

[22] N. Kriegeskorte and R. A. Kievit, "Representational geometry: integrating cognition, computation, and the brain", *Trends in Cognitive Sciences*, vol. 17, no. 8, pp. 401–412, 2013.

[23] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of Internal Clustering Validation Measures", in *Proceedings of the 2010 IEEE International Conference on Data Mining*, ser. ICDM '10, Washington, DC, USA: IEEE Computer Society, 2010, pp. 911–916.

[24] U. von Luxburg, "A Tutorial on Spectral Clustering", *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[25] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE ", *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[26] C. Magri, K. Whittingstall, V. Singh, N. K. Logothetis, and S. Panzeri, "A toolbox for the fast information analysis of multiple-site LFP, EEG and spike train recordings", *BMC Neuroscience*, vol. 10, no. 81, pp. 1–24, 2009.

[27] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[28] C. M. Niell, "Exploring the Next Frontier of Mouse Vision", *Neuron*, vol. 72, no. 6, pp. 889–892, 2011.

[29] N. R. Pal and J. Biswas, "Cluster validation using graph theoretic concepts", *Pattern Recognition*, vol. 30, no. 6, pp. 847–857, 1997.

[30] S. Panzeri, R. Senatore, M. A. Montemurro, and R. S. Petersen, "Correcting for the Sampling Bias Problem in Spike Train Information Measures", *Journal of Neurophysiology*, vol. 98, no. 3, pp. 1064–1072, 2007.

[31] S. Panzeri and A. Treves, "Analytical estimates of limited sampling biases in different information measures", *Network: Computation in Neural Systems*, vol. 7, no. 1, pp. 87–107, 1996.

[32] S. Rota Bulò, "A game-theoretic framework for similarity-based data clustering", PhD thesis, Università Ca' Foscari, Venice, Italy, 2009.

[33] S. Rota Bulò and I. M. Bomze, "Infection and immunization: A new class of evolutionary game dynamics", *Games and Economic Behavior*, vol. 71, pp. 193–211, 2011.

[34] S. Rota Bulò and M. Pelillo, "A game-theoretic approach to hypergraph clustering", *IEEE Transactions On Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1312–1327, 2013.

[35] S. Rota Bulò and M. Pelillo, "Dominant-set clustering: A review", *European Journal of Operational Research*, vol. 262, no. 1, pp. 1–13, 2017.

[36] S. Rota Bulò, M. Pelillo, and I. M. Bomze, "Graph-based Quadratic Optimization: A Fast Evolutionary Approach", *Computer Vision Image Understanding*, vol. 115, no. 7, pp. 984–995, 2011.

[37] P. Rousseeuw, "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis", *J. Comput. Appl. Math.*, vol. 20, no. 1, pp. 53–65, 1987.

[38] M. I. Sereno and J. M. Allmann, in *The Neural Basis of Visual Function*. 1991, ch. Cortical visual areas in mammals, pp. 160–172.

[39] C. E. Shannon, "A Mathematical Theory of Communication", *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[40] S. Tafazoli, H. Safaai, G. De Franceschi, F. B. Rosselli, W. Vanzella, M. Riggi, F. Buffolo, S. Panzeri, and D. Zoccolan, "Emergence of transformation-tolerant representations of visual objects in rat lateral extrastriate cortex", *eLife*, vol. 6, pp. 1–39, 2017.

[41] A. Torsello, S. Rota Bulò, and M. Pelillo, "Beyond Partitions: Allowing Overlapping Groups in Pairwise Clustering", in *19th International Conference on Pattern Recognition*, ser. ICPR 2008, IEEE Computer Society, 2008, pp. 1–4.

[42] A. Torsello, S. Rota Bulò, and M. Pelillo, "Grouping with Asymmetric Affinities: A Game-Theoretic Perspective", in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, ser. CVPR 2006, IEEE Computer Society, 2006, pp. 292–299.

[43] N. X. Vinh, J. Epps, and J. Bailey, "Information Theoretic Measures for Clusterings Comparison: Is a Correction for Chance Necessary?", in *Proceedings of the 26th Annual International Conference on Machine Learning*, New York, NY, USA: ACM, 2009, pp. 1073–1080.

[44] N. X. Vinh, J. Epps, and J. Bailey, "Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance", *Journal of Machine Learning Research*, vol. 11, pp. 2837–2854, 2010.

[45] J. W. Weibull, *Evolutionary game theory*. Cambridge, Massachusetts: The MIT Press, 1997.

[46] D. L. K. Yamins, H. Hong, C. F. Cadieu, E. A. Solomon, D. Seibert, and J. J. DiCarlo, "Performance-optimized hierarchical models predict neural responses in higher visual cortex", *Proceedings of the National Academy of Sciences of the United States of America*, vol. 111, no. 23, pp. 8619–8624, 2014.