bioRxiv preprint doi: https://doi.org/10.1101/2020.03.02.967489; this version posted January 9, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

A framework to identify structured behavioral patterns within rodent spatial trajectories

Francesco Donnarumma[1], Roberto Prevete[2], Domenico Maisto[3], Simone Fuscone[4], Emily M. Irvine[5], Matthijs A. A. van der Meer[5], Caleb Kemere[6], and Giovanni Pezzulo[1,*]

[1]Institute of Cognitive Sciences and Technologies (ISTC), National Research Council (CNR), Via San Martino della Battaglia, 44 - 00185 Rome, Italy
[2]Department of Electric Engineering and Information Technologies (DIETI), Università degli Studi di Napoli Federico II, Naples, Italy
[3]Institute for High Performance Computing and Networking (ICAR), National Research Council (CNR), Via Pietro Castellino 111, 80131 Naples, Italy
[4]Aix Marseille Université, France
[5]Department of Psychological and Brain Sciences, Dartmouth College, Hanover, NH, USA
[6]Electrical and Computer Engineering, Rice University, Houston, Texas
[*]giovanni.pezzulo@istc.cnr.it

ABSTRACT

Animal behavior is highly structured. Yet, structured behavioral patterns – or "statistical ethograms" – are not immediately apparent from the full spatiotemporal data that behavioral scientists usually collect. Here, we introduce a framework to quantitatively characterize rodent behavior during spatial (e.g., maze) navigation, in terms of movement building blocks or *motor primitives*. The hypothesis that we pursue is that rodent behavior is characterized by a small number of motor primitives, which are combined over time to produce open-ended movements. We assume motor primitives to be organized in terms of two sparsity principles: each movement is controlled using a limited subset of motor primitives (sparse superposition) and each primitive is active only for time-limited, time-contiguous portions of movements (sparse activity). We formalize this hypothesis using a sparse dictionary learning method, which we use to extract motor primitives from rodent position and velocity data collected during spatial navigation, and successively to reconstruct past trajectories and predict novel ones. Three main results validate our approach. First, rodent behavioral trajectories are robustly reconstructed from incomplete data, performing better than approaches based on standard dimensionality reduction methods, such as principal component analysis, or single sparsity. Second, the motor primitives extracted during one experimental session generalize and afford the accurate reconstruction of rodent behavior across successive experimental sessions in the same or in modified mazes. Third, in our approach the number of motor primitives associated with each maze correlates with independent measures of maze complexity, hence showing that our formalism is sensitive to essential aspects of task structure. The framework introduced here can be used by behavioral scientists and neuroscientists as an aid for behavioral and neural data analysis. Indeed, the extracted motor primitives enable the quantitative characterization of the complexity and similarity between different mazes and behavioral patterns across multiple trials (i.e., habit formation). We provide example uses of this computational framework, showing how it can be used to identify behavioural effects of maze complexity, analyze stereotyped behavior, classify behavioral choices and predict place and grid cell displacement in novel environments.

Introduction

During cognitive tasks, such as spatial navigation, animals exhibit highly structured behavior. These behavioral patterns or motifs can indicate important latent dimensions, such as task complexity (e.g., movement variability) or the amount of training that the animal has received (e.g., movement stereotypy). Yet, a standard methodology to extract structured behavioral patterns from the set of data (e.g., position and velocity) that behavioral scientists usually collect during spatial navigation is still lacking.

Here we present a computational framework that characterizes the latent structure – or a "statistical ethogram" – of rodent movement dynamics during spatial navigation, in terms of movement building blocks or motor primitives. The idea that motor control uses a combination of motor primitives is widely accepted in computational neuroscience[1–8] but its application for the study of rodent spatial navigation is less common. A line of research focusing on rat behavior used a descriptive approach to the study of rodent exploration, which revealed its organization into movement subsystems that can be readily quantified[9–12]. This body of work provides evidence for an organization of behavior in terms of motor primitives, also providing a developmental perspective on their acquisition[13,14].

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.02.967489; this version posted January 9, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Using motor primitives to study rodent movements during spatial navigation

Motor primitives can be characterized as spatiotemporal patterns that recur across an animal's behavior and thus permit reconstructing it in a parsimonious way – in the sense that a small number of motor primitives permit reconstructing a large dataset of movements. Figure 1 illustrates schematically the basic elements of the motor primitives approach. Spatial (x,y) and velocity (x,y) data are collected from animal trajectories during spatial navigation (Figure 1a) and used to derive a set (or dictionary) of motor primitives, such as the ones shown in Figure 1b. Later, these motor primitives can be used to "reconstruct" actual animal trajectories (see Figure 1c for an example of reconstruction of spatial components of the trajectories). Figure 1d shows an example of motor primitives superimposed on one of the mazes that we used in our study (see later for explanation).

There are two fundamental (biological and computational) motivations for using a motor primitive approach to study complex behavior. From a biological perspective, the hypothesis that the brain may adapt motor primitives as fundamental units to organize complex (spatial or other) movements in time has received increased attention[15]. In this perspective, complex movements can be composed by combining over time a limited basis of dynamic motor primitives – such that, for each time frame, only one or a few motor primitives are active. Therefore, inferring the animal's motor primitives during movements can help understanding how the brain organizes complex behavioral patterns.

There is also a second, more practical reason to use the motor primitives approach to study animal movements (even if ultimately the brain uses a different organization). Characterizing animal behavior in terms of its underlying motor primitives can be more practical than using the full spatiotemporal [x(t),y(t)] data and more indicative of underlying regularities (e.g., behavioral stereotypy), as demonstrated by the analysis of behavioral motifs in various animal, species such as Caenorhabditis elegans[16], fruit fly[17], mice[18] and rat[19]. Although previous work has analyzed behaviors such as path stereotypy in terms of path similarity (e.g.[20]), such descriptive approaches cannot generate novel data. More recent unsupervised learning methods[21] or related approaches[22–26] are more similar to our approach, but have not yet been applied to structured maze data.

Double sparsity in the organization of motor primitives and in dictionary learning methods

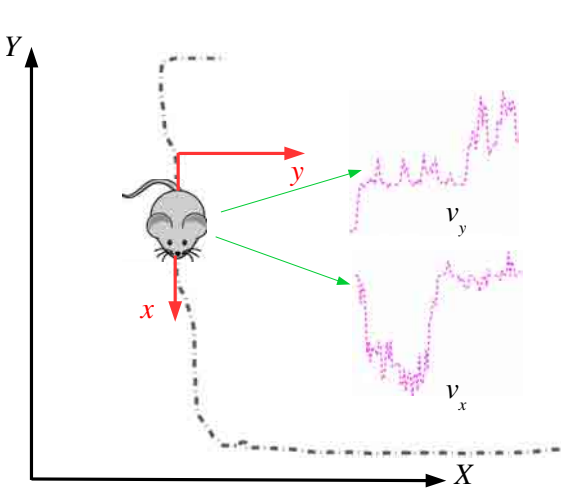
There are various ways to construct motor primitives at neural and computational level (see Figure 2). The approach pursued here is motivated by a *double sparsity* hypothesis, according to which the organization of motor primitives shows simultaneously two kinds of sparsity, which we call *sparse superposition* and *sparse activity*. *Sparse superposition* implies that at each time step, a movement can be reconstructed by superimposing at most a few primitives – and possibly just one (see Figure 2b). *Sparse activity* entails that the same motor primitive is only active for a time-limited, time-contiguous portion of the movement (see Figure 2c).

Figure 2 compares four hypotheses on how motor primitives are organized. The first hypothesis assumes that there is no sparsity in the organization of motor primitives: motor actions are represented as combinations of all the available primitives, for the entire duration of the motor action (Figure 2a). The second hypothesis assumes sparse superposition: given a large enough set of primitives, different actions can be represented by a combination of small subsets of primitives, which can overlap[27,28] (Figure 2b). The third hypothesis assumes sparse activity: each primitive is active only in a limited time interval, rather than for the entire duration of a motor action[29,30] (Figure 2c). Finally, the fourth hypothesis combines the second and third hypotheses and assumes dual sparsity: motor actions are represented by combinations of small subsets of primitives, which are only active in a limited time interval (Figure 2d).

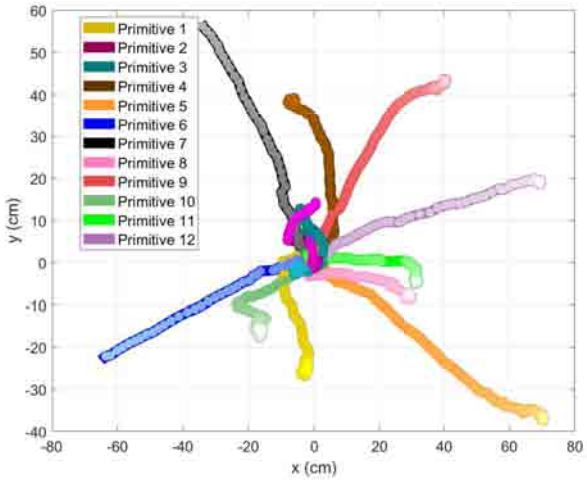
The four hypotheses listed above correspond to four computational approaches to identify motor primitives from data. The first approach corresponds directly to Principal Component Analysis (PCA), which does not assume any sparsity. Interestingly, the other three approaches correspond directly to three sparse dictionary learning methods in machine learning[31–33], which assume different kinds of sparsity: namely, the $\ell 1$-regularization method[34], which assumes recombination of atoms with *sparse coefficients*, and corresponds to the sparse superposition hypothesis; Structured Sparse Principal Component Analysis (SSPCA)[35], which assumes *atom structured sparsity* and thus corresponds to the sparse activity hypothesis; and Sparse Representation of Structured Sparse Dictionaries (SRSSD), which assumes double sparsity. In other words, there is a close relation between the concept of sparse motor primitives in biological motor control and methods for sparse dictionary learning in machine learning. In the latter, the equivalent of motor primitives are called *atoms* and they are part of a *dictionary*, which corresponds to the animal's behavioral repertoire.

In the rest of the article, we introduce a framework for the analysis of rodent movements during spatial navigation that is based on the SRSSD approach – which directly implements a dual sparsity hypothesis. We then compare the SRSSD approach with the three alternative methods, showing that it entails better and more parsimonious reconstruction of data. The possibility to model, predict and reconstruct spatial trajectories in terms of their underlying motor primitives opens the doors to a number of potential applications of the framework, including the possibility to discover sequential structure, repeating patterns and stereotypy in behavior, and predict animal choices – which we discuss and exemplify in the final section.

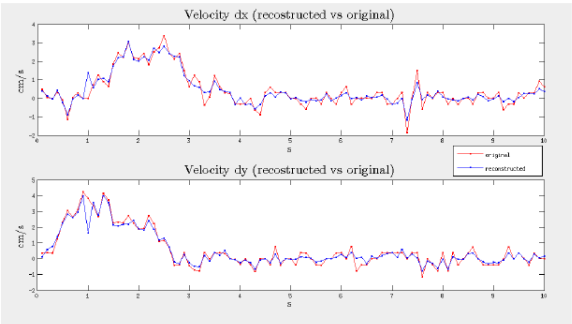
bioRxiv preprint doi: https://doi.org/10.1101/2020.03.02.967489; this version posted January 9, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.



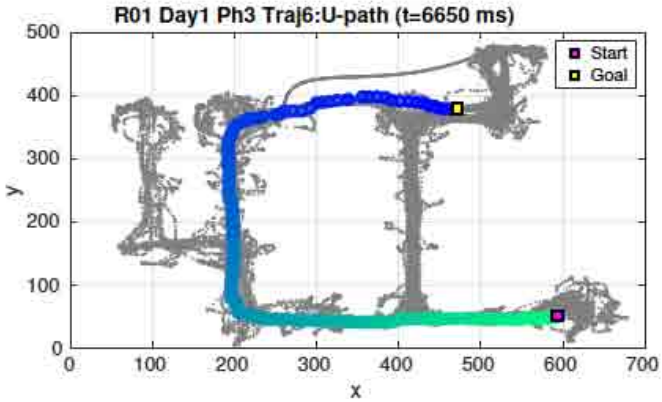
(a) Reference system



(b) Twelve example motor primitives



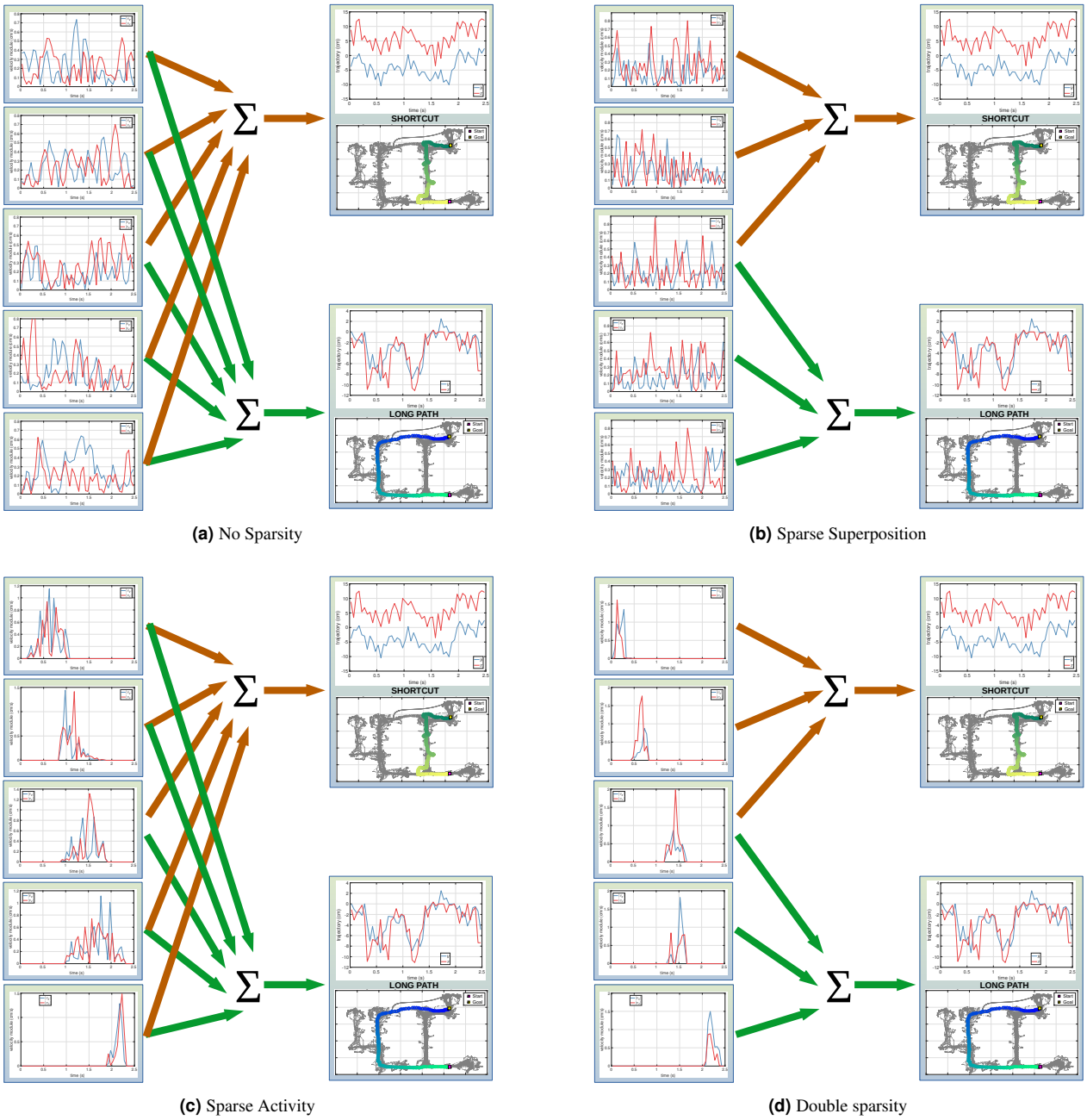
(c) Velocity profiles of real data vs. reconstructed



(d) Motor primitives superimposed on a maze

Figure 1. Schematic illustration of the approach. Panel (a): the rodent-centered coordinate system used in input for the dictionary learning model. The dotted black line is the actual animal trajectory. For each time, we extracted from data a 2-D (x, y) spatial position. Note that our reference system is not anchored to the maze. Rather, at each time point, x represents the last direction of the animal. The resulting, animal-centered reference system permits deriving different primitives for different orientations of the animal. Furthermore, we extracted animal velocity in x and y dimensions. Panel (b): Twelve sample motor primitives extracted from navigation data. For illustration purposes we simulated the primitives starting at point (0,0) at time 0. The gradient of lighter colors for circle markers indicates increase in time. Panel (c): example reconstruction of real rodent data using motor primitives (here, only the velocity in x and y axes is reconstructed). Blue is the actual velocity profile over time, red is the reconstructed profile. Panel (d): example superimposition of motor primitives on one of the mazes used in this study. See the main text for explanation.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.02.967489; this version posted January 9, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.



(a) No Sparsity

(b) Sparse Superposition

(c) Sparse Activity

(d) Double sparsity

Figure 2. *Four alternative approaches and hypotheses on the organization of motor primitives.* Panel (a) illustrates the hypothesis of no sparsity in motor primitives organization, implemented using Principal Component Analysis (PCA). Panel (b) illustrates the sparse superposition hypothesis, implemented using $\ell 1$-regularization. In this perspective, different actions are represented by a combination of small subsets of primitives, which can overlap. The Figure shows an example with five motor primitives, where the first three are jointly used to represent one motor action, and the last three are jointly used to represent another motor action. Panel (c) illustrates the sparse activity hypothesis, implemented using SSPCA. In this perspective, actions are represented using all available motor primitives, but these are active only in a restricted time interval, rather than for the entire duration of the action. The Figure shows an example with five motor primitives, all of which are used to represent both motor actions, but are time-localized. Panel (d) illustrates the double sparsity hypothesis, which combines the previous two hypotheses and is implemented using SRSSD. See the main text for details.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.02.967489; this version posted January 9, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Results

The computational framework that we used in this article is illustrated in Figure 3. First, we divided the experimental (rodent trajectory) data in three sets: a training set, a validation set and a test set. We then used the training set to develop several candidate *dictionaries* of motor primitives from data, using *sparse dictionary learning*[31–33]. Furthermore, we evaluated candidate dictionaries on different reconstructions performed on a validation set, in order to select which combination of parameters (*dictionary size*, *coefficient sparsity* and *atom structured sparsity*) afford the best reconstruction of trajectory data. Finally, we used the selected dictionary to reconstruct data of the test set and to simulate real rodent motor data – hence validating the approach and comparing it with alternatives. Below we discuss in more detail all the elements of the framework.

Dataset

In order to validate our approach, we conducted a series of experiments using a dataset of rodent movements collected in the van der Meer lab (Dartmouth College). This dataset includes data from 3 rats (R01, R02, R03). Each rat experienced a sequence of 8 unique mazes, with novel routes introduced to a familiar environment every session / day. The rats navigated the maze environment between two goal (reward) locations, placed at the ends of a familiar U-shaped track. For each session, maze data was collected in three distinct phases. In the first and second phases, the rats were only able to navigate along the familiar U-shaped track. In the third phase, two novel paths became accessible – one of which was a shortcut between the two reward locations, and the other a dead end. Figure 4 shows the behavioral trajectories from rat R01 for each of the eight mazes in the three phases: phase 1 (green), phase 2 (yellow) and phase 3 (grey). Table 1 shows the overall duration of each of the three phases for rat R01 (across multiple trials, see below) from each of the eight sessions.

R01 (min)	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8
Phase1	10.08	10.05	8.02	8.13	9.88	8.48	8.41	8.10
Phase2	20.33	20.37	15.02	20.55	20.33	21.12	20.24	20.04
Phase3	37.01	46.10	45.05	42.68	52.21	45.82	40.43	55.04

Table 1. Duration in minutes of the different phases, rat R01

Experimental results, animal R01

In all the analyses below, we focus on trajectory data of animal R01 for illustrative purposes; see the Supplementary Material for data on animals R02 and R03, which show the same pattern of results as animal R01.

Building the dictionary of Primitives using sparse dictionary learning methods

The methodology to build and select the dictionary of primitives that we used in our subsequent analyses includes three main steps.

The first (preparation) step of our novel approach consists in dividing the database of rat trajectory data into three sets: a *training set*, which includes data from phases 1, 2 and 3 of day 1; a *validation set* and a *test set*, both including (non-overlapping) data from phase 3 of days 1 to 8. The actual data that we consider in the training, validation and test sets consist in "patches" of rat trajectories (i.e., portions of trajectories in x,y), lasting a maximum of 2.5 seconds each. Each of the three sets consists in 10000, non-overlapping patches, which are randomly selected. Patches with missing data (e.g., cases in which the animal goes out of the video camera frame) were excluded. Thus, the training data consist of an input matrix of dimension 10000x100, corresponding to 10000 patches of size 100 each composed of juxtaposed x,y position variables - $(x(1), y(1), \ldots, x(50), y(50))$. Note that the time bin between two samples $x(k+1)$ and $x(k)$ is fixed and is 0.05 s.

The complexity of these trajectory data can be appreciated by considering that a principal component analysis (PCA) shows that 73 principal components are necessary to explain about 95% of their variance, see Figure 5.

The second (learning) step of our novel approach consists in deriving candidate dictionaries of motor primitives which reconstruct accurately the *training set* data. For this, we use the Sparse Representation of Structured Sparse Dictionaries (SRSSD) approach, which directly implement our double sparsity hypothesis. SRSSD has two parameters of sparsity: coefficient and atom structured sparsity, which correspond to the two hypotheses of sparse superposition and sparse activity, respectively. The method therefore outputs not just dictionaries of motor primitives (V) but also their coefficients U (see Algorithm 1 and Methods Section for more details).

In this learning step, we select the best 600 (6x10x10) candidate dictionaries, obtained by varying 6 different dictionary sizes (with 25, 50, 75, 100, 125, and 150 motor primitives, respectively), 10 levels of atom sparsity and 10 levels of coefficient sparsity.

To measure how well the 600 candidate SRSSD dictionaries reconstructed the *training set* data, we adopted a methodology that is analogous to the *missing pixel method*: we canceled out some parts (from 10% to 90%) of the trajectories that compose

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.02.967489; this version posted January 9, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

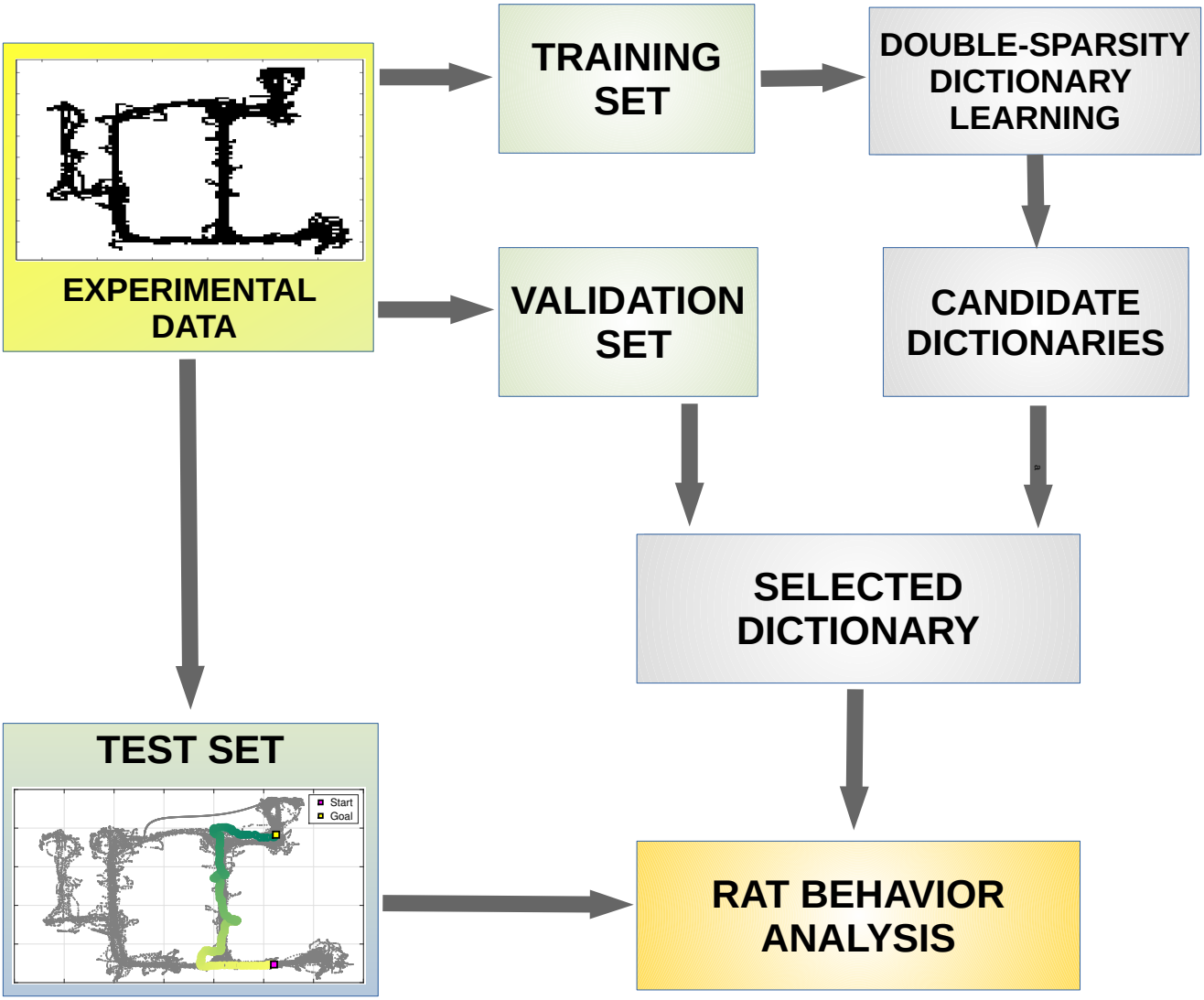


Figure 3. A framework for the analysis of rat behavior. The figure shows the workflow of the framework proposed in our paper. After collecting the experimental data, these data are split into three sets: a training set, a validation set and a test set. The training set is processed by a double-sparse dictionary learning performed with the SRSDD algorithm (see Material and methods Alg. 1), that represents data in a set of atoms - called *dictionary* - which are interpreted as rat motor primitives. Different candidate dictionaries are found during this process with different parameters of atom structured sparsity. Candidate dictionaries are evaluated on different sparse reconstructions performed by means of Alg. 2 on a validation set. At the end of this phase, the selected dictionary can be used on different tests to analyse rat behavior during different motor actions in the maze.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.02.967489; this version posted January 9, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

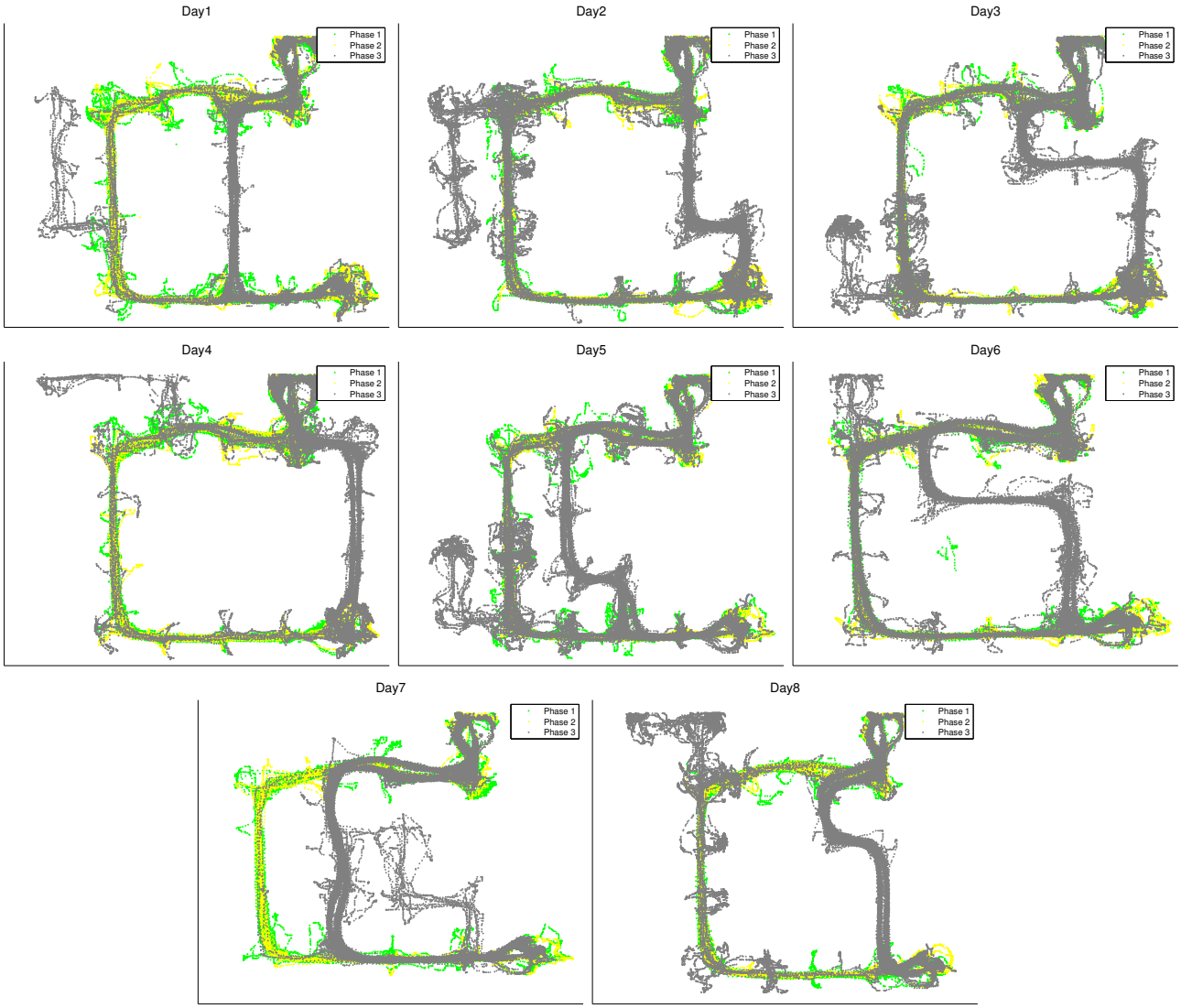


Figure 4. Dataset: trajectories of animal R01 across 8 days, 3 phases every day. See the main text for explanation.

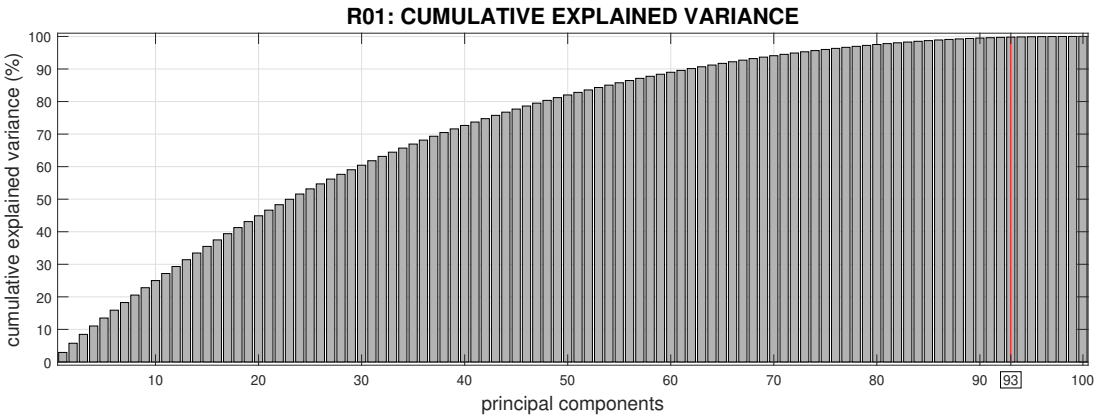


Figure 5. Cumulative explained variance in function of principal components. 93 principal components were selected in the tests, which explain about the 99% of the variance of the training set

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.02.967489; this version posted January 9, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

the training set and evaluated how well the learned dictionaries permit to reconstruct these missing parts (see the Methods section for an explanation of how the reconstruction error RMS is calculated). Figure 6(left) shows coefficient sparsity and atom sparsity of the solutions found by SRSSD for dictionaries of size 150, i.e., the size that gave the best results, during the learning phase. The results show that the solution space is extensively covered, both in coefficient sparsity and in atom sparsity.

In the third (validation) step of the approach we select the best 30 SRSSD dictionaries V amongst the aforementioned 600 candidates. For this, we force a reconstruction with sparse coefficients (see Algortihm 2): we fix V and obtain the couple V and U that better reconstructs the *validation set* data, by varying 5 levels of "noise" and 10 levels of coefficient sparsity.

While we conducted the validation on all the 600 dictionaries considered above, we only show the results of the 100 dictionaries of size 150 (i.e., the size that gave the best results during the learning step). Figure 6(right) shows coefficient and atom sparsity of the solutions in the validation phase of these 100 dictionaries, with 10 levels of coefficient sparsity and 5 levels of "noise" (0.1, 0.3, 0.5, 0.7 and 0.9) – corresponding to the percentage of "missing" trajectory data (10%, 30%, 50%, 70% and 90%) to be reconstructed. The results show the existence of solutions with both high coefficient sparsity and high atom sparsity, hence satisfying simultaneously the sparse superposition and the sparse activity hypotheses – and confirming that the range of the parameter space is well chosen. In the subsequent tests, we verify that successful solutions (those with better reconstruction) do satisfy a reasonably high double sparsity, hence lending support for both the sparse superposition and the sparse activity hypotheses.

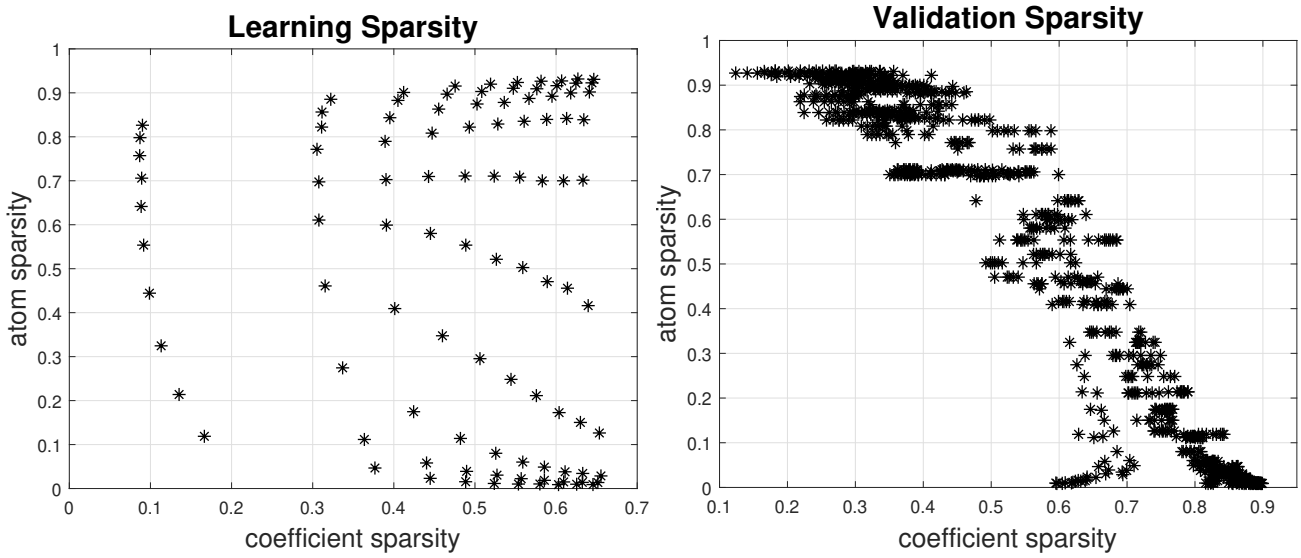


Figure 6. SRSSD method applied to navigation data from animal R01. Left: learning phase. Right: validation phase. Results are shown for dictionaries of size 150 (corresponding to lowest RMS in the test set / validation phase), with 10 values of coefficient sparsity and 5 levels of noise (0.1, 0.3, 0.5, 0.7, 0.9). Notice that the solution space is extensively covered, across both (coefficient and atom sparsity) dimensions.

The results of the validation step are shown in Figure 7 (black asterisks correspond to different dictionaries) and compared with PCA (horizontal line). The PCA results were computed by comparing the reconstruction error with 93 principal components explaining $\approx 99\%$ of the variance of training data. The number of components (from 1 to 100) was selected by taking the best performing set that reconstructs the training set. The comparison illustrates that for each level of noise, various dictionaries have lower reconstruction error (RMS) than the best PCA subset. We also conducted a more systematic evaluation, in which we compared our approach based on double sparsity (SRSSD) with a method that entails no sparsity (PCA) and two other dictionary learning methods, each incorporating only one kind of sparsity: namely, coefficient sparsity $\ell_1 - regularization$[34] and atom structured sparsity SSPCA[35]. We compared the best instances of SRSSD, SSPCA, and ℓ_1, which resulted from considering 10 different values of sparsity parameters in the same range (coefficient sparsity and atom sparsity parameters for SRSSD, coefficients sparsity parameter for ℓ_1 and atom sparsity for SSPCA) and best PCA set of 93 components, as described above. Each method can be seen as the computational counterpart of a different biological hypotheses: no sparsity (PCA), sparse activity (SSPCA), sparse combination (ℓ_1) and double sparsity (SRSSD).

In this comparison, we also considered another method, ℓ_1-RD, which starts from a random dictionary with values in the interval $(-1,1)$ matching the dimensionality of the other sparse methods (SRSSD, SSPCA, ℓ_1). We generated 1000 random dictionaries and selected the best one, which was capable of reconstructing the training set with a sparse reconstruction, varying the sparse coefficient parameter in a similar way as ℓ_1 method, but without optimising the dictionary values. This method

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.02.967489; this version posted January 9, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

allows us to experimentally evaluate the possibility of obtaining a useful dictionary representation by chance.

We compared the five methods with 6 levels of noise (0%, 10%. 30%, 50%, 70%, 90%). For each level of noise, we computed the percent error of the methods in relation to SRSSD, as follows:

$$\% \; error = \frac{RMS(X) - RMS(SRSSD)}{RMS(SRSSD)} \cdot 100$$

where $RMS(X)$ is the best reconstruction error of a selected method X and $RMS(\text{SRSSD})$ is the best reconstruction error of SRSSD.

The results of this comparison are shown in Figure 8 (note that positive values indicate smaller reconstruction error). Our results show that, as expected, higher noise levels decrease reconstruction accuracy across all methods. Moreover, and most importantly, SRSSD outperforms all the other methods across all levels of noise and is more noise-tolerant.

In sum, this first series of tests and comparisons indicates that the choice of a double-sparse approach like SRSSD nicely incorporates the double sparsity hypothesis, which is appealing from a *biological* perspective; and furthermore, it affords a more efficient reconstruction and prediction of motor data of the rat, which is appealing from a *computational* perspective.

Consequently, in the subsequent sections, we shift our focus on a deeper analysis of the double sparsity method by means of the best 30 SRSSD dictionaries learned in these tests and illustrate the potentialities of the proposed approach.

Reconstruction across consecutive days using the best 30 dictionaries

We tested whether the best 30 dictionaries V identified using the above procedure afford an accurate reconstruction of novel data, from the *test set*. More specifically, we were interested in assessing whether a learning procedure that only considered data from day 1 (i.e., *training set*) permitted reconstructing non-overlapping data from the third phases of days 1-8 in our dataset (i.e., *test set*). Note that the reason why the *test set* only includes the third phases of each day is that this is when, in the rodent experiment, novel portions of the maze appear. Data on the third phases of each day are this ideal to test whether our approach generalises to novel and unforeseen situations. Furthermore, we asked whether the motor primitives required to reconstruct animal trajectories in a given day capture essential characteristics of the day's task, such as maze complexity or stereotypy of behavior.

Figure 9 shows the reconstruction results for the best 30 SRSSD dictionaries. There are at least two elements to be appreciated. First, reconstruction error is lower in days 1, 2 and 8. The low reconstruction error in day 8 indicates that the method generalises well across distal days. The fact that reconstruction is better in day 8 than the preceding days could be explained by considering the close spatial similarities between the mazes of days 1 and 8 (see see Figure 4). Second, as shown in Figure 9(b), days 4 and 8 have the highest coefficient sparsity, implying that fewer motor primitives are required for their correct reconstruction. The fact that behavior in different mazes and days can be reconstructed using small subsets of (possibly overlapping) primitives lends support for the biological sparse superposition hypothesis.

Most importantly, we found the lower coefficient sparsity of days 4 and 8 to reflect well maze characteristics and animal behaviour. This can be appreciated if one consider our next analysis of task demands and complexity, see Figure 10. The first measure of task demands we considered is the mean time the animal needed to complete a trial, see Figure 10a. This analysis shows that time is shorter in days 4 and 8, which are the two days in which our previous analysis revealed higher coefficient sparsity, see Figure 9(b).

The second measure we considered is the spatial complexity of the maze, see Figure 10b. We assessed spatial complexity by dividing the maze into bins, counting how many bins the animal occupies for each trial of phase 3, and then averaging (i.e., dividing this number by the number of trials); see the Supplementary Material for Figures illustrating the results of this process. High spatial complexity (or entropy) implies that the animal has explored and occupied all the available positions, while low spatial complexity means that the animal followed stereotyped trajectories. Note that there are multiple ways to construct spatial complexity[36,37]. In our approach, complexity depends on behavior (similar to[38]) – or better, on a graph constructed on the basis of the actual trajectories of the animal, rather than a fixed graph that only includes spatial dependencies. Our analyses reveals that spatial complexity of days 4 and 8 is lower compared to the other days, revealing stereotyped behavior in these days (see also below on a convergent measure of stereotypy). The same results emerge if we plot coefficient sparsity versus time to complete the trial (Figure 10c) and coefficient sparsity versus complexity (Figure 10d). Note that days are color-coded and days 4 and 8 (orange and grey, respectively) have the highest values in both measures.

Overall, this analysis reveals that in days 4 and 8, coefficient sparsity is higher, the animal is faster and maze complexity is lower. The coherence across these measures supports the idea that the motor primitive formalism we adopted captures essential task components such as its complexity.

Example applications of the method in the generation, description and classification of behavior

Animal behavior is highly structured. Behavioral scientists and neuroscientists often deal with complex datasets of behavioral data – e.g., the full spatiotemporal x(t), y(t) data – that are not convenient for asking questions like: how stereotyped is the

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.02.967489; this version posted January 9, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

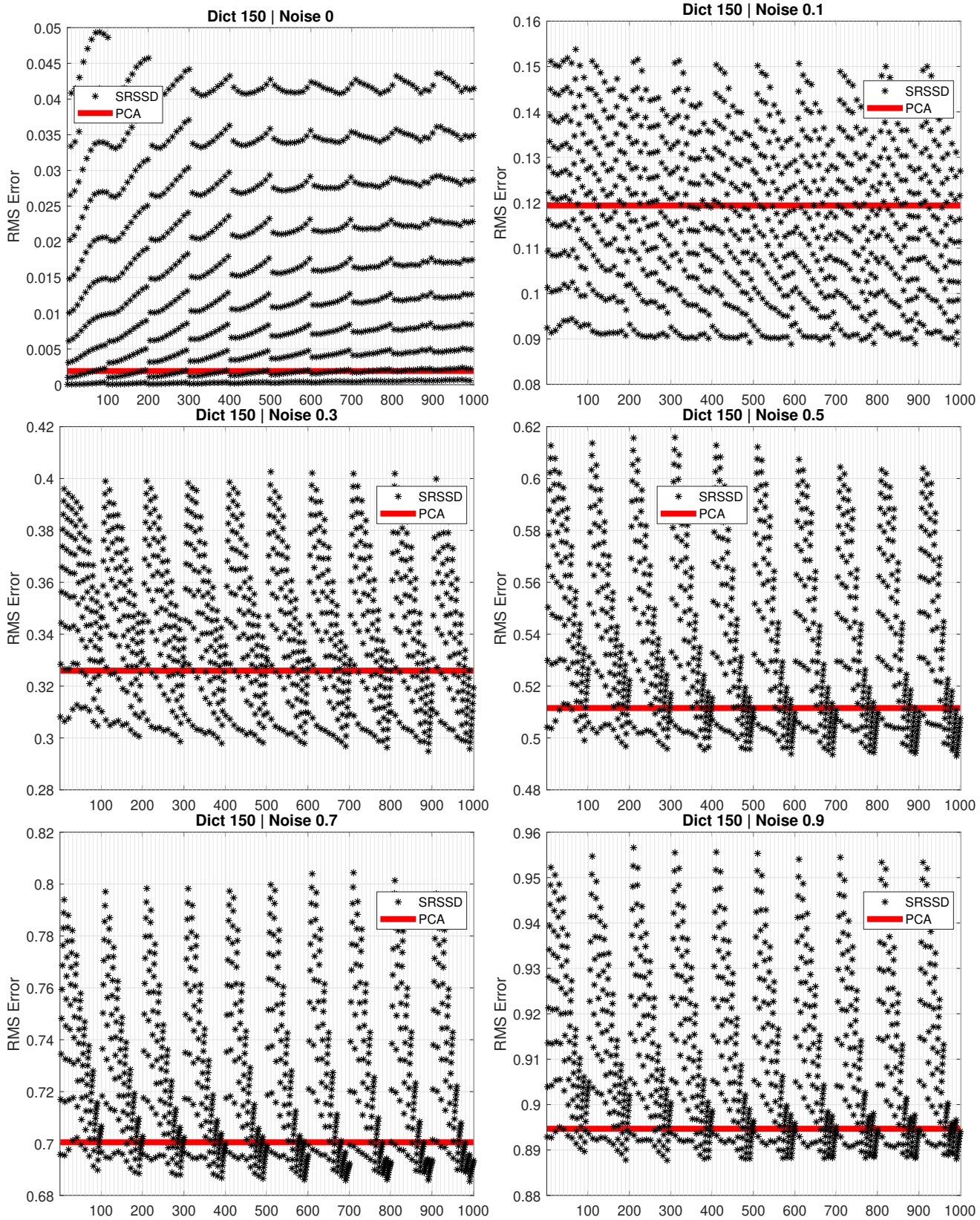


Figure 7. Comparison between the reconstruction (RMS) error of SRSSD (blue dots) and PCA (red line) across various levels of noise. Each figure shows 100 dictionaries found during the learning step. We remind that each reconstruction relies on 2 kinds of sparsity: atom structured sparsity (varied during the learning phase) and coefficient sparsity (which is varied here); see the Methods section. Here, each of the 100 dictionaries (with its fixed atom sparsity) is tested with 10 different values of the coefficient sparsity parameter. Hence, for each figure, each of the 1000 asterisks corresponds to a unique combination of dictionary and coefficient sparsity. The reconstruction error for all the asterisks that appear under the red line is better than PCA.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.02.967489; this version posted January 9, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

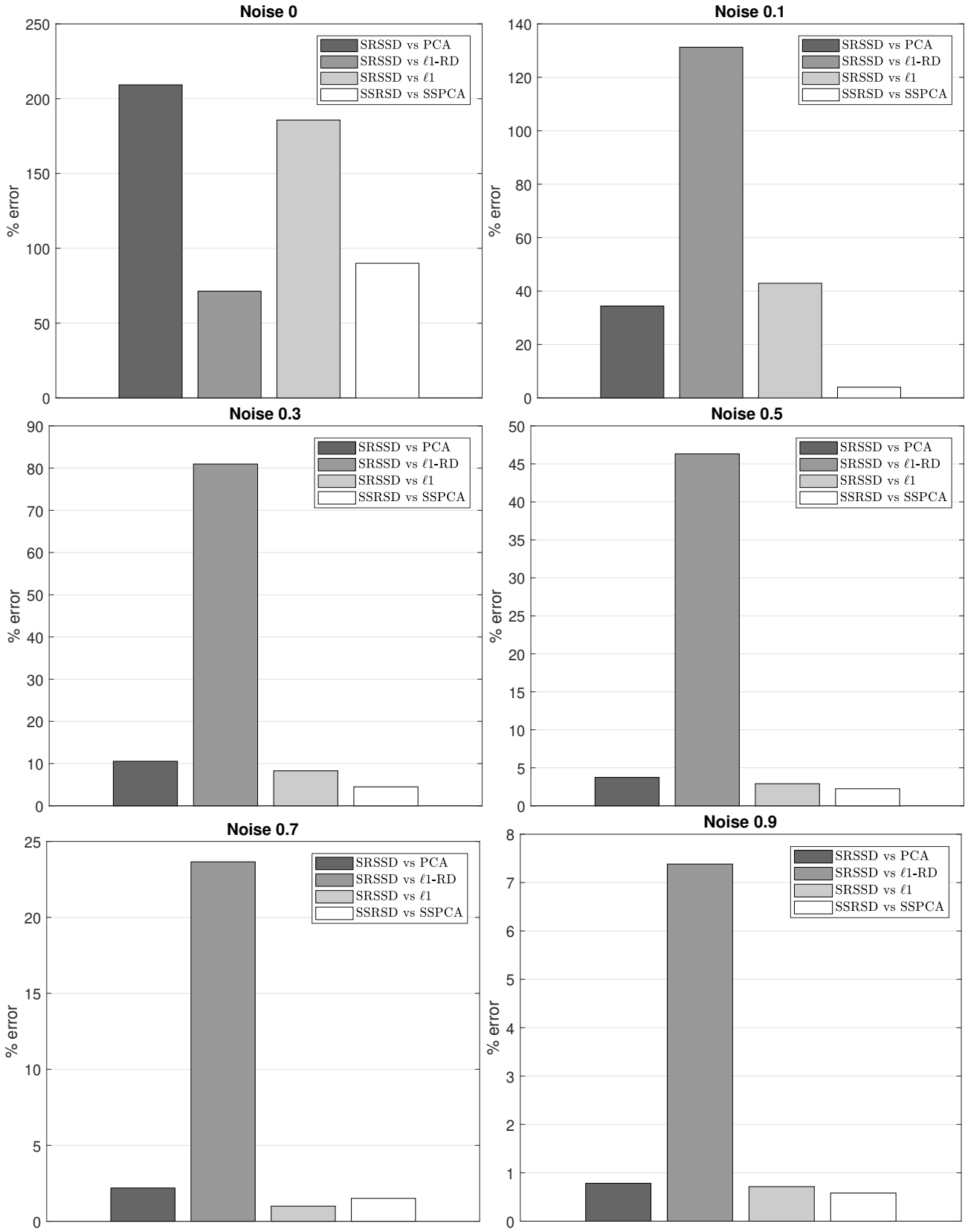
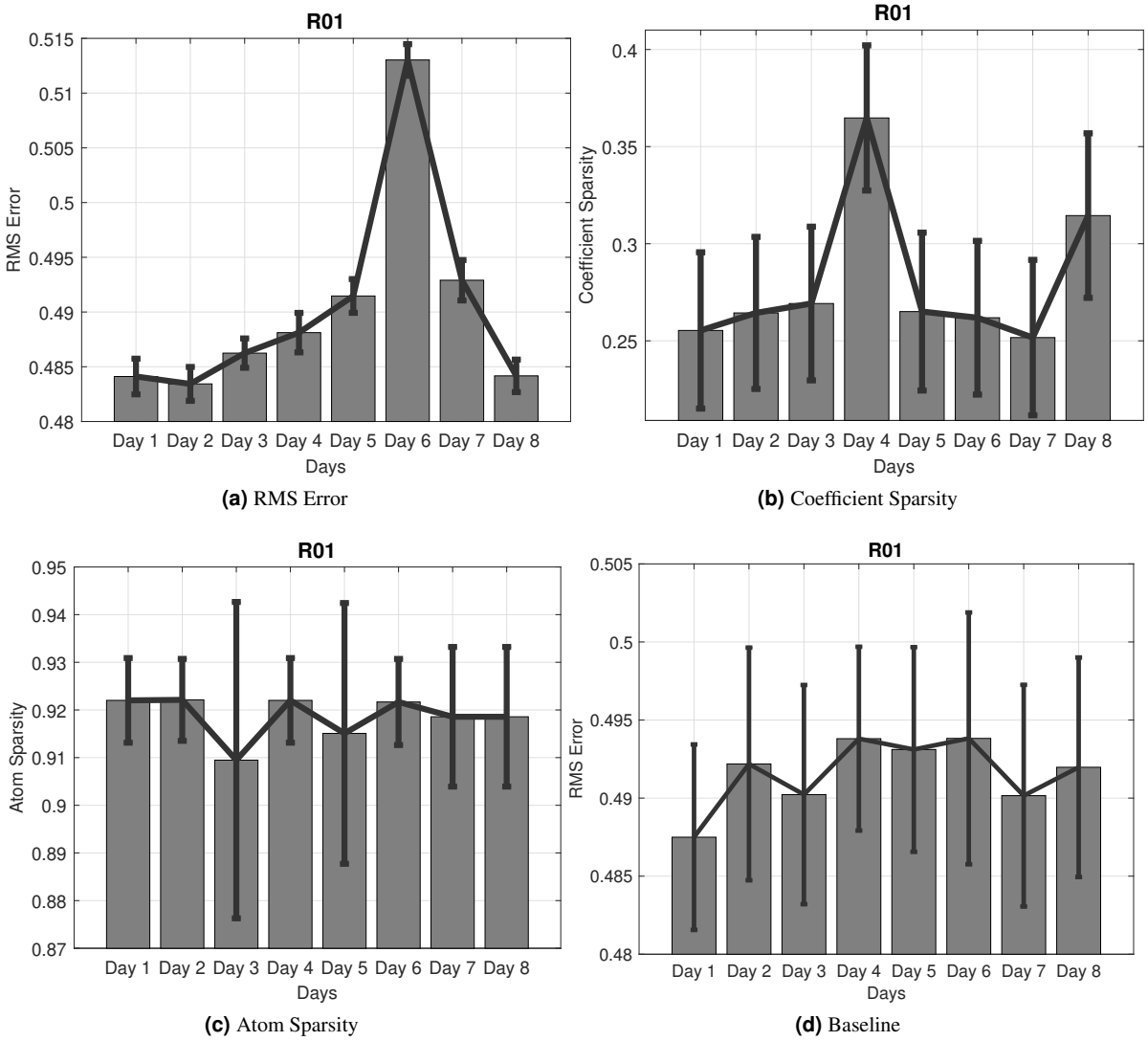


Figure 8. The percent error for PCA, $\ell 1$-RD, $\ell 1$-regularization, and SSPCA compared to SRSSD, across six levels of noise (0%, 10%. 30%, 50%, 70%, 90%). It is possible to appreciate that SRSSD outperforms the other methods, across all noise levels.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.02.967489; this version posted January 9, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

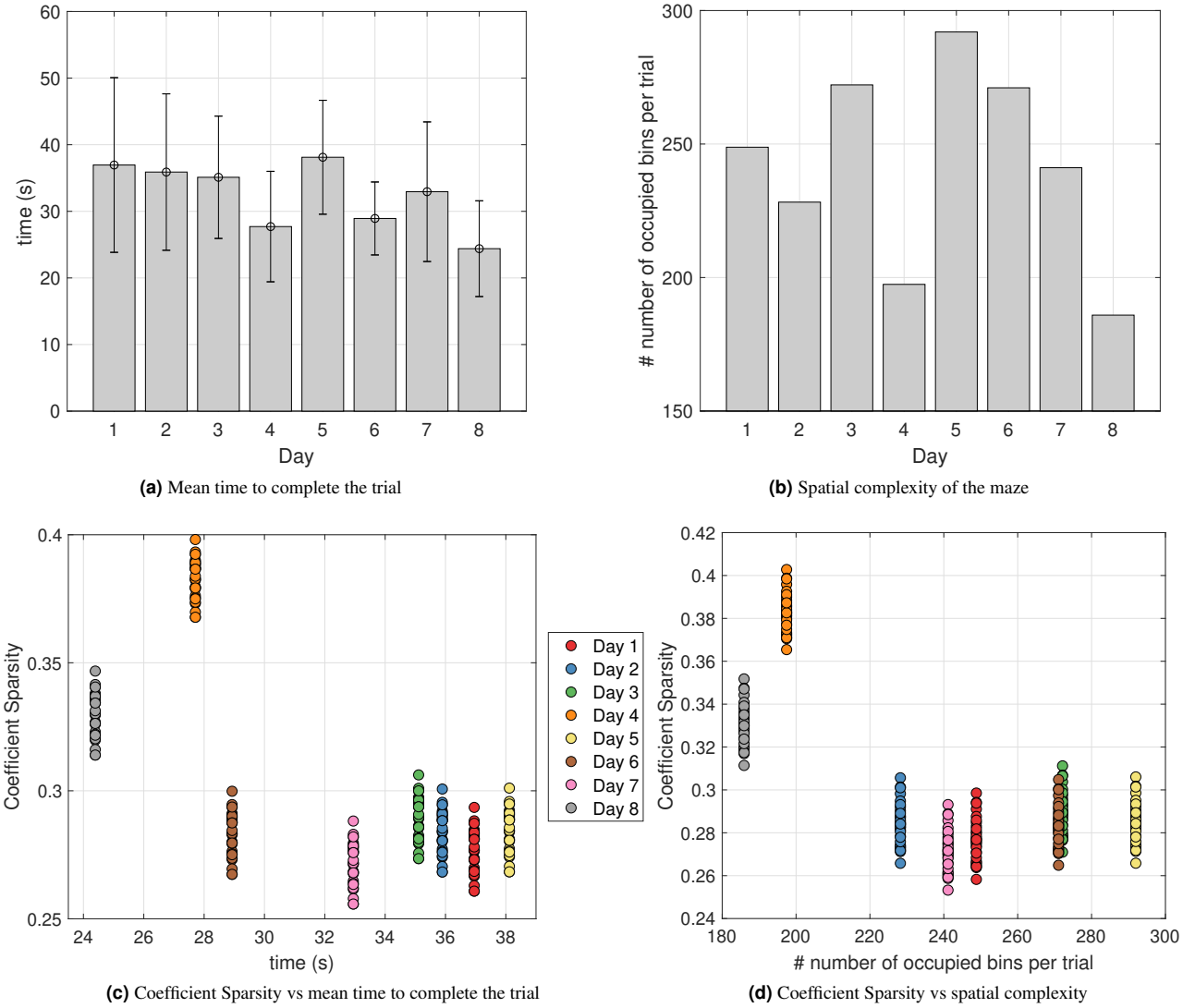


(a) RMS Error **(b)** Coefficient Sparsity

(c) Atom Sparsity **(d)** Baseline

Figure 9. Reconstruction across 8 days (after learning only from data of day 1. Noise is 0.5 and the best 30 SRSSD dictionaries found on day 1 are used. Panel (a): RMS Error. Panel (b): coefficient sparsity. Panel (c) atom sparsity. Panel (d) Baseline performance: reconstruction of Phase 1. Note that Panel (a) shows an increase in the reconstruction error, until day 6 and then lower levels of the error in the next two days. The increase of reconstruction error across successive days is to be expected, given that the dictionary is acquired on day 1 (the high error value of day 6 may be due to specific characteristics of the corresponding maze). The decrease of error in the last two days may be due to stereotypy of behavior, specific characteristics of the corresponding mazes, or a combination of both factors.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.02.967489; this version posted January 9, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.



(a) Mean time to complete the trial

(b) Spatial complexity of the maze

(c) Coefficient Sparsity vs mean time to complete the trial

(d) Coefficient Sparsity vs spatial complexity

Figure 10. Measures of task demands and complexity of the rodent task. (a) Average time in seconds and standard deviation required to complete a trial and secure a reward. (b) Spatial complexity of the maze: number of bins per trial occupied during the task. (c) Scatter Plot of coefficient sparsity versus time to complete the trial. (d) Scatter Plot of coefficient sparsity versus spatial complexity.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.02.967489; this version posted January 9, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

animal's behavior? How well can future behavior be predicted by past behavior? How does behavior evolve with experience on the task, within and across sessions?

One appeal of the motor primitive code is that it provides a description of rat behavior that is more compact than the full spatiotemporal [x(t),y(t)] data yet richer than a simple mono-dimensional descriptor [e.g., speed(t)] – and lends itself to a number of applications. Our method based on motor primitives permits estimating the building blocks of a "computational ethogram" that captures the main behavioral regularities (i.e., repeated spatiotemporal sequences that are highly predictable within-sequence) and the temporal dependencies between them (e.g., transitions or contingencies between sequences). This method can be used in a number of ways.

Generating simulated behavior on novel mazes and associated neuronal responses

One possible application would be generating simulated behavior on various novel mazes. Simulated trajectories can be helpful in estimating the to-be-expected complexity of behavioral data. Furthermore, they can be fed as input to other components to predict hippocampal place cell activity, thus affording power analyses. One example is shown in Figure 11. We used the computational models of[39,40] to simulate the firing of example grid and place cells during uniform exploration (Panels (e) and (f), respectively) versus partial exploration of the environment following a simulated trajectory (Panels (c) and (d), respectively), also inferring the connectivity structure that link the simulated place cells (S1 to S20 in Panel (g)) This example illustrates the possibility to use simulated trajectories to predict neuronal firings in novel environments. Note that in the above examples, the trajectories are generated by chaining motor primitives randomly, mimicking uniform exploration. Future studies may consider extensions of this method that learn (flat, hierarchical or context-dependent) transition probabilities between primitives and can generate more contextually-adequate trajectories.

Assessing behavioral stereotypy after learning.

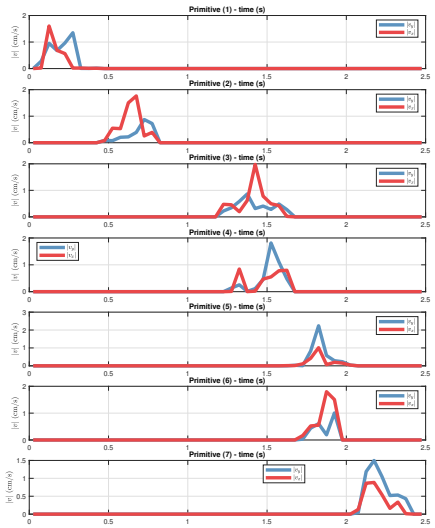
Another application is testing stereotypy of behavior (as a possible index of habitization), by looking at coefficient sparsity during learning. Stereotypy of behavior can be tested by considering whether coefficient sparsity increases during learning, indexing the fact that the animal is using fewer primitives (and hence a more restricted and stereotyped behavioral repertoire). As an example of this analysis, we considered how coefficient sparsity changes while animal R01 learns to navigate the basic U-shaped maze (i.e., during phase 1 of day 1). For this, we have split data of phase 1 of day 1 into 5 intervals of the same length, removed the two extremes (1 and 5) and the middle interval (3), and compared intervals 2 vs. 4, corresponding to initial and final phases of learning, see Figure 12. One-way analysis of variance (ANOVA) indicates that the difference between coefficient sparsity values of intervals 2 and 4 is statistically significant ($F[1,78] = 4.197$, $p < 0.05$), with the latter interval showing greater coefficient sparsity. These results indicate that as the animal R01 learns the task, it uses less primitives and a more stereotyped behavior. Assessing whether this method is potentially more robust than measures of path stereotypy[20,41] or simple descriptions of x(t), y(t) data like Fourier decomposition[42] is beyond the scope of this work. Yet, in general, describing spatiotemporal data in terms of (movement) primitives is considered to be more noise-tolerant in motor control[1–4].

Classifying and predicting behavior.

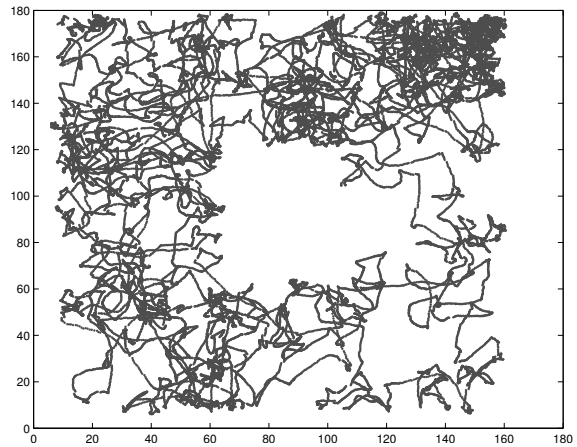
Our approach based on the identification of motor primitives can also be used to classify and predict animal behavior and choices. To exemplify this, we performed an experiment aimed at studying whether motor primitives extracted during different trials of phases 3 of each day permit to classify whether in these trials animal R01 follows the U-path or the shortcut. For each day, we randomly select 1000 trajectories (500 Shortcut and 500 U-path trajectories) and apply the reconstruction method, for each of the 30 best dictionaries, in order to find sparse coefficients. We then use the coefficients to classify U-path versus shortcut trajectories (which we had previously labelled, see examples in the Supplementary Materials). For this, we perform a classification test on the set of 1000 trajectories, using a linear support vector machine (SVM) classifier in a k-fold cross validation (with k=5). We compute mean and standard deviation over the dictionaries of the accuracies. Figure 13 shows that this method affords very accurate classification across all the 8 days. Note that this result does not trivially depend on the fact that the animal occupies different x-y coordinates during U-path versus shortcut trajectories. The motor primitives are agnostic about maze-centered x-y coordinates, as their reference system is centered on the animal, see Figure 1a.

The last bar of Figure 13 (labelled Day1-8) shows the results of a second experiment, in which we used a single dataset of 8000 samples of trajectories (1000 for each day, resulting in 4000 U-path and 4000 shortcut trajectories, combined together). Note that while the U-path remains constant across days, the shortcuts change every day, making this experiment potentially more challenging than the former. Again, we performed a k-fold cross validation test using a linear SVM classifier and we computed accuracies over the 30 dictionaries coefficients. Even in this second experiment, our method affords very accurate classification of U-path versus shortcut trajectories. These results illustrate that motor primitives extracted through our method capture behaviorally relevant regularities that are informative (for example) about animal trajectories and choices.

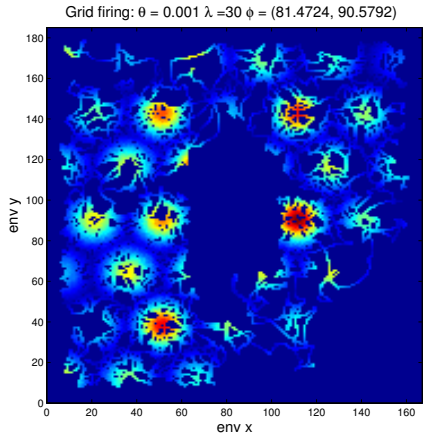
bioRxiv preprint doi: https://doi.org/10.1101/2020.03.02.967489; this version posted January 9, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.



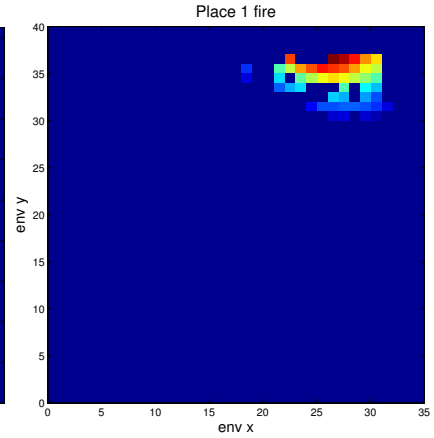
(a) Example motor primitives



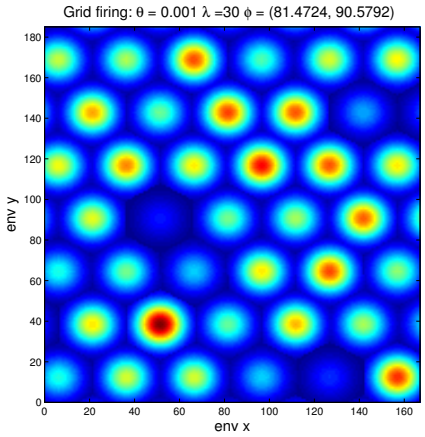
(b) Example trajectory generated by randomly chaining motor primitives, in an open arena



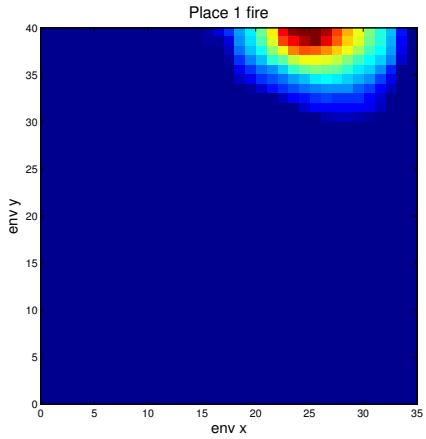
(c) Simulated grid cells, following the trajectory shown in Panel (b)



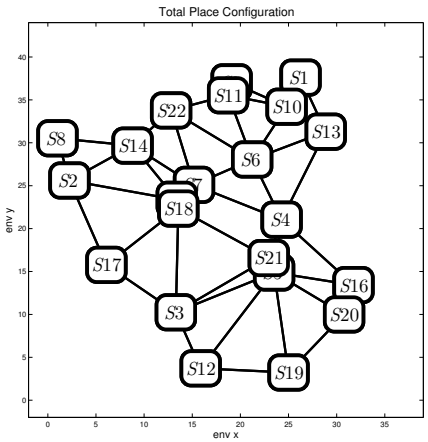
(d) Simulated place cells, following the trajectory shown in Panel (b)



(e) Simulated grid cells, with uniform exploration



(f) Simulated grid cells, with uniform exploration



(g) Higher level planning

Figure 11. Graphical illustration of potential applications of the analysis. Panel (a) provides examples of motor primitives extracted from a maze, which could be used to simulate animal behavior in novel mazes, such as the open arena shown in Panel (b). In turn, the simulated trajectories shown in Panel (b) can be used as inputs to computational models that permit simulating grid cells (Panel (c)) and place cells (Panel (d)); which are different from grid cells (Panel (e)) and place cells (Panel (f)) generated under the hypothesis of uniform exploration. It is also possible to infer the connectivity structure that link the simulated place cells (S1 to S20 in Panel (g)), by considering which ones are linked via motor primitives. In principle, this connectivity structure can be used to design planning algorithms.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.02.967489; this version posted January 9, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

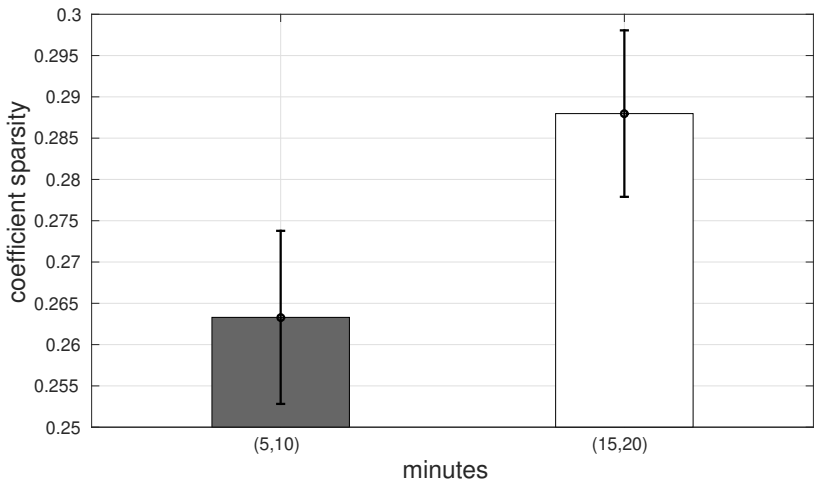


Figure 12. Mean and standard error of coefficient sparsity values for animal R01 behavior in the 2nd and 4th intervals of phase 1 of day 1. The coefficient sparsity values relative to the animal behavior range in $[0.25, 0.30]$. While the animal learns to navigate in the U maze, coefficient sparsity increases significantly ($F[1,78] = 4.197$, $p < 0.05$), revealing an increased stereotypy of behavior.

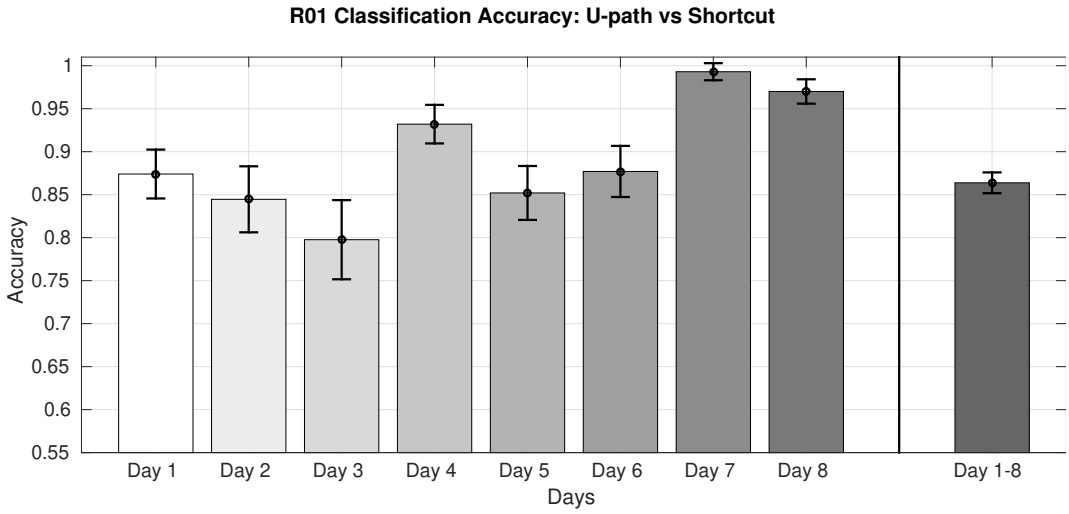


Figure 13. Classification of trajectories (U-path or shortcut) of phase 3, animal R01

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.02.967489; this version posted January 9, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Discussion

We introduced a novel framework based on sparse dictionary learning to quantitatively characterize rodent behavior during spatial navigation. This approach is capable of extracting latent structure – a "statistical ethogram" – from rodent motor data (i.e., position and velocity), in terms of movement building blocks of motor primitives. The approach adopted here (SRSSD) specifically incorporates the hypothesis that motor primitives are organized in terms of two sparsity principles: each movement is controlled using a limited subset of motor primitives (sparse superposition) and each primitive is active only for time-limited, time-contiguous portions of movements (sparse activity).

Our experiments on rodent trajectories in eight mazes show that the double sparse dictionary learning (SRSSD) method outperforms principal component analysis (PCA) and two other sparse dictionary learning methods, which incorporate only sparse coefficients ($\ell_1 - regularization$) or atom structured sparsity (SSPCA), across all the noise levels we tested, with appreciable results over 0.5 noise – thus speaking in favour of the robustness of the methods. Furthermore, our results show good generalization across eight consecutive days. Despite the fact that we only used data from day 1 in the training set, the reconstruction of trajectories was not degraded in the successive seven days, suggesting that the motor primitives approach extracts regularities that are persistent over time. These regularities correspond to spatiotemporal patterns of movement, which can include, for example, rapid forward movements or rotations, which are repeated multiple times during the maze navigation and may be thus revelatory of an underlying modular organisation of behavior.

Our further analysis indicates that our approach based on motor primitives characterizes well task regularities such as spatial complexity and movement variance in terms of (coefficient) sparsity constraints. This makes intuitive sense, as navigation scenarios in which the animals' movements had lesser variance (e.g., for days 4 and 8 from animal R01) require less motor primitives (i.e., allow for more sparsity). It is also reassuring to notice that, although our method allowed the construction of motor primitives of up to 10 second, the primitives that actually populate the best dictionaries are of about 1 second, which is a more realistic time constraint for biological movement.

This is, to the best of our knowledge, the first framework that applies the concept of motor primitives – which is popular in motor control and computational neuroscience – to a data-driven analysis of rodent spatial navigation data. The method introduced here is based on 2D position data, which could be seen as limited compared to more sophisticated (3D camera, accelerometer) approaches[21]. However, functionality with 2D data is more applicable to the vast majority of behavioral scientists and neuroscientists.

The approach we have devised can be used by behavioral scientists and neuroscientists in behavioral and neural analyses. Some example uses that we have shortly illustrated in this paper are: controlling for confounding effects (e.g., of maze complexity on behavior and reward collection), analyzing habitual or stereotyped behavior, classifying and predicting animal choices, and predicting place and grid cell displacement in novel mazes. The results we reported illustrate the feasibility of the method, which can be successfully added to the toolbox of behavioral scientists and neuroscientists to improve their ability to quantitatively characterize and understand animal behavior.

Besides, our approach can be used for other purposes, such as, for example, to support the analysis of neuronal data or to design experiments that test the neuronal underpinning of motor primitives for spatial navigation. In decoding analyses, tuning curves (aka, the encoding model[43]) are typically estimated from those epochs during which the animal is running, which in turn is estimated using a running speed cutoff. However, the cutoff cannot clearly distinguish between different situations, e.g., grooming, directed running towards a goal or performing exploratory moves. Distinguishing these different behaviors is important as they modulate neural firings, above and beyond spatial position. The approach presented here can distinguish different movement primitives (or even behavioral patterns or modes) and has thus the potential to improve our ability to relate neural activity to behavior, as done e.g. in decoding analyses. Furthermore, the motor primitives extracted from behavior may be characterized neurally. Studying the potential neuronal correlates of (spatial) motor primitives remains an important objective for future research.

Finally, our approach can be extended to reveal and study more complex patterns of behavior than those considered here. Note that the method we have described does not try to estimate sequential dependencies between primitives. However, it would be trivial to compute those dependencies (e.g., a probability distribution $P(pi|pj)$ that a primitive pi follows a primitive pj) from training data, in order to estimate the most likely (or most surprising) transitions. It would be then possible to derive measures of path novelty, which would be related to (relatively) unpredictable transitions; or to study sequential (possibly, planning-related) patterns of behavior. These and other potential applications of the method remain to be tested in future studies.

Materials and methods

To characterize and extract motor primitives from trajectory data (i.e., position and velocity) during spatial navigation we adopt *sparse dictionary learning* techniques [31–33,44–46]. Sparse dictionary learning techniques consist of unsupervised learning of a set of atoms that can be linearly superimposed to represent the elements of the dataset (in our case rodent spatial trajectories).

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.02.967489; this version posted January 9, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

In these approaches, either the atoms or the coefficients of the linear combinations (or both) are imposed to have a large number of components equal to zero. These two cases correspond to *atom structured sparsity* and *coefficient sparsity*, respectively. In dictionary learning literature a reconstruction that forces sparse coefficients is also referred to as *sparse coding*.

These two kinds of sparsity distinguish dictionary learning methods from more traditional approaches like PCA, which decompose data using orthogonal basis functions. Sparse representations are instead based on the construction of a set of nonorthogonal – *over-complete* – representations, usually called *dictionaries*; see[47] for a thorough review.

Under an over-complete basis, the decomposition of a signal is not unique, but this can offer clear advantages. Firstly, it results in a greater flexibility in capturing structure in the data: instead of a small set of general basis functions, there is a larger set of more specialized basis functions. In this way, relatively few elements are required to represent parts of the learning signals. This eventually results in more compact representations, because each basis function can describe a significant amount of structure in the data. In pattern recognition literature, these over-complete dictionaries are able to achieve more meaningful and stable representations of the source data[48]. For this reason, sparse methods generally outperform the traditional orthogonal basis functions[49]. Another advantage of dictionary learning methods over PCA is that the length of the primitives does not need to be predefined or fixed, but it is automatically determined by the algorithm, allowing for more flexibility depending on the particular application case.

Dictionary learning and the SRSSD approach

Recent machine learning approaches[44–46,50–53] represent signals as linear combinations of a large number of *atoms*, collected in sets called *dictionaries*. The dictionaries are computed using prior information encoded in penalization terms defining a particular kind of a minimization problem. These approaches are informatively grouped into two classes:

(i) ***Sparse atoms.*** This is the case when each atom involves just a small number of the original variables (see, for example, Sparse-PCA[50], sPCA-rSVD[54] and Structured-Sparse-PCA[35]);

(ii) ***Sparse coding.*** In this case, an *overcomplete* set of atoms is learned from the data, but the approximation of each signal involves only a restricted number of atoms. Hence, signals are represented by sparse linear combinations of the atoms (see, for example, MOD[55], *K-SVD*[53], and ℓ_1-regularized[34]).

Recently, a new algorithm named Structured Sparse Dictionaries for Sparse Representation (*SRSSD*) was presented in[33] and applied to a cognitive science study[32]; its fundamental characteristics is to search for sparse atoms and sparse coding simultaneously, reconciling, in this way, the two different dictionary-learning approaches.

Within the SRSSD method, we denote $\mathbf{X} \in \mathbb{R}^{n\times p}$ as a matrix where rows correspond to experimental observations (e.g., sequences of x and y coordinates that compose a trajectory, e.g., x1, y1, x2, y2, etc.). Note that n is the number of patches and p their length. Furthermore, we denote $\mathbf{V} \in \mathbb{R}^{p\times r}$ as a *dictionary*, whose r columns $\mathbf{V}^k$ represent the atoms learned by the dictionary. Thus, r is the number of atoms and p their maximum length. Note that given its sparsity, the SRSSD method can learn dictionary elements having different length, whose maximum we set to p (e.g., the number of nonzero consecutive elements can be $< p$). Finally, we define $\mathbf{U} \in \mathbb{R}^{n\times r}$ as the coefficient matrix.

The aim of SRSSD is finding out the best approximation of $\mathbf{X}$ in terms of $\mathbf{V}$ and $\mathbf{U}$ (i.e., $\mathbf{X} \approx \mathbf{U}\mathbf{V}^{\mathrm{T}}$).

This problem can be formulated in terms of a minimization problem as follows:

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V}} \frac{1}{2np}\|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_F^2 + \lambda \sum_{k=1}^{r} \Omega_v(\mathbf{V}^k) \\ s.t.\ \ \forall j, \Omega_u(\mathbf{U}_j) \le \eta,\ \forall k\ \|\mathbf{V}^k\|_2 = 1 \end{aligned} \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm for matrices, $\mathbf{U}_j$ are the rows of $\mathbf{U}$, $\Omega_v(\mathbf{V}^k)$ and $\Omega_u(\mathbf{U}_j)$ are norms or quasi-norms constraining (*regularizing*) the solutions of the minimization problem, with the parameters $\lambda \ge 0$ and $\eta \ge 0$ that control to which extension the dictionary and the coefficients are regularized, respectively. If one assumes that both Ω_u and Ω_v are convex, the problem (1) is convex w.r.t. $\mathbf{U}$ for $\mathbf{V}$ fixed, and vice versa.

Following[35], the atom structured sparsity is imposed by setting

$$\Omega_v(\mathbf{V}^k) = \left\{ \sum_{i=1}^{s} \| \mathbf{d}^{G_i} \circ \mathbf{V}^k \|_2^{\alpha} \right\}^{\frac{1}{\alpha}} \tag{2}$$

where $\alpha \in (0,1)$, and each $\mathbf{d}^{G_i}$ is a p-dimensional vector satisfying the condition $d_j^i \ge 0$, with $G_i \in \mathscr{G}$ where $\mathscr{G}$ is a subset of the power set of $\{1,\dots,p\}$, such that $|\mathscr{G}| = s$ and $\bigcup_{i=1}^{s} G_i = \{1,\dots,p\}$. Thus, the vectors $\mathbf{d}^i$ define the structure of the atoms. More specifically, each $\mathbf{d}^{G_i}$ individuates a group of variables such that $d_j^i > 0$ if $j \in G_i$ and $d_j^i = 0$ otherwise. The norm $\| \mathbf{d}^{G_i} \circ \mathbf{V}^k \|_2^{\alpha}$

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.02.967489; this version posted January 9, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

penalizes the variables selected by $d^i_j > 0$, hence, by this norm, each vector $\mathbf{d}^{G_i}$ induces non zero values for the j-th elements of the atom $\mathbf{V}^k$, when $j \in G$.

The resulting set of selected variables depends on the contribution of each $\mathbf{d}^{G_i}$ as described in[56]. For example, if the vectors $\mathbf{d}^{G_i}$ represent a partition on the set $\{1,\ldots,p\}$, then the penalization term (2) favours atoms $\mathbf{V}^k$ composed of non-zero variables belonging to just one part of the partition and so on: for specific choices of $\{G_i\}_{i=1}^s$, Ω_v leads to standard sparsity-inducing norms.

Nevertheless, the norm Ω_v expressed in Equation (2) is not differentiable and, consequently Equation (1) is not convex with respect to $\mathbf{V}$ for $\mathbf{U}$ fixed — although the converse is still true. By using results presented in[35,57], we can write down Ω_v as a quadratic expression and reformulate the Equation (1) as:

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V},\mathbf{H}} \frac{1}{2np}\|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_F^2 + \frac{\lambda}{2}\sum_{k=1}^{r}\left[(\mathbf{V}^k)^T Diag(\mathbf{Z}^k)^{-1}\mathbf{V}^k) + \|\mathbf{H}_k\|_\beta\right] \\ s.t.\ \forall j,\ \Omega_u(\mathbf{U}_j) \le \eta, \quad \forall i\ \|\mathbf{V}^i\|_2 = 1 \end{aligned} \tag{3}$$

where $\mathbf{H} \in \mathbb{R}_+^{r\times s}$ is a matrix satisfying the condition $h_{ki} \ge 0$ with $\beta = \frac{\alpha}{2-\alpha}$ and that minimizes the second term of the previous expression in which $\mathbf{Z} \in \mathbb{R}^{p\times r}$ has got its own elements defined as $z_{jk} = \left\{\sum_{i=1}^s \left(d^i_j\right)^2 (h_{ki})^{-1}\right\}^{-1}$. Notice that, as shown in[35], for both $\mathbf{U}$ and $\mathbf{V}$ fixed, the minimizer of (3) can be given in the closed form $\bar{h}_{ki} = |y^k_i|^{2-\alpha}\|\mathbf{y}^k\|_\alpha^{\alpha-1}$, for $k = 1,2,\ldots,r$ and $i = 1,2,\ldots,s$, where each $\mathbf{y}^k \in \mathbb{R}^{1\times s}$ is the vector $\mathbf{y}^k = (\|\mathbf{d}^1 \circ \mathbf{V}^k\|_2, \|\mathbf{d}^2 \circ \mathbf{V}^k\|_2, \ldots, \|\mathbf{d}^s \circ \mathbf{V}^k\|_2)$. Ultimately, we impose the constraint $\|\mathbf{V}^i\|_2 = 1$ to avoid solutions where $\mathbf{V}$ goes towards $\mathbf{0}$, or $\mathbf{U}$ becomes a sparse matrix, whose non-zero elements have excessively high values.

Since the functional in (3) is separately convex in each variable, we solve the minimization problem following the usual approach of alternating optimizations with respect to the values $\mathbf{H}$, to the coefficients $\mathbf{U}$ and to the dictionary $\mathbf{V}$. Most methods are based on this alternating scheme of optimization[34,58] and its convergence towards a critical point of the functional is guaranteed by Corollary 2 of[59].

Algorithm 1 summarises the procedural steps of *SRSSD*, which comprises three stages:

S1 ***Matrix* H *update*.** In the first stage, both $\mathbf{U}$ and $\mathbf{V}$ are fixed and $\mathbf{H}$ values are updated. As said above, one can update $\mathbf{H}$ by computing $\bar{h}_{ki} = |y^k_i|^{2-\alpha}\|\mathbf{y}^k\|_\alpha^{\alpha-1}$. However, to avoid numerical instabilities near zero, we adopt the following smoothed update: $\mathbf{H}_k \leftarrow max\{\bar{\mathbf{H}}_k, \varepsilon\}$ with $\varepsilon \ll 1$.

S2 ***Sparse Coding*.** In the second stage, both $\mathbf{V}$ and $\mathbf{H}$ are fixed and $\mathbf{U}$ values are updated. Note that equation (3) comprises two terms to be minimized, but the second term does not depend on $\mathbf{U}$. This implies that the optimization problem of equation (3) can be reformulated as: $\min_{\mathbf{U}} \|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_F^2$ $s.t.$ $\forall j, \|\mathbf{U}_j\|_0 \le \eta$. There are several well-known "pursuit algorithms" that find approximate solutions to this kind of problems, such as Basis Pursuit (BP)[60] and Orthogonal Matching Pursuit (OMP)[61]. Here, we approximate the ℓ_0 norm (a non convex problem) with its best convex approximation, i.e., ℓ_1 norm – which allows us to perform sparse coding by applying an Iterative Soft-Thresholding (IST)[62]. Note that this stage is equivalent to the application of Algorithm 2, where we apply only sparse coding with a fixed dictionary.

S3 ***Structured Atoms*.** In the third stage, both $\mathbf{U}$ and $\mathbf{H}$ are fixed and the dictionary $\mathbf{V}$ is updated; furthermore, following[35], an atom structured sparse representation can be found. As both $\mathbf{U}$ and $\mathbf{H}$ are fixed, the problem of equation (3) can be reformulated as:

$$\begin{aligned} \min_V \frac{1}{2}\|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_F^2 + \frac{\lambda np}{2}\sum_{k=1}^{r}(\mathbf{V}^k)^T Diag(\mathbf{Z}^k)^{-1}\mathbf{V}^k) \\ \forall\ i,\ \|\mathbf{V}^i\|_2 = 1 \end{aligned} \tag{4}$$

In this case, as both terms of Equation (4) are convex and differentiable terms with respect to $\mathbf{V}$, a closed-form solution for $\mathbf{V}$ can be found. However, a proximal method is considered to avoid p matrix inversions:

$$\begin{aligned} \mathbf{V}^k \leftarrow \Pi_v(Diag(\mathbf{Z}^k)Diag(\|\mathbf{U}^k\|_2^2\mathbf{Z}^k + np\lambda\mathbf{I})^{-1}\cdot \\ \cdot(\mathbf{X}^T\mathbf{U}^k - \mathbf{V}\mathbf{U}^T\mathbf{U}^k + \|\mathbf{U}^k\|_2^2\mathbf{V}^k)) \end{aligned} \tag{5}$$

where $\Pi_v(w)$ is simply the Euclidean projection of w onto the unit ball, and the argument of Π_v is obtained by composing a forward gradient descent step on the first term with the proximity operator of the second term of (4).

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.02.967489; this version posted January 9, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Method for reconstructing test trajectories and calculating reconstruction error (RMS)

To build the dataset to be reconstructed, we split the trajectories into training segments or patches $X \in \mathbb{R}^{n \times p}$, where n is the number of patches (10000 in our test set) and $p/2$ is the length in time of the patches. Each row of X includes a sequence of x and y coordinates that compose a patch, e.g., x1, y1, x2, y2, etc. We split trajectories lasting more than $p/2$ time steps into two or more rows of X. Then, we use SRSSD to learn a dictionary $V \in \mathbb{R}^{p \times r}$ of r atoms, having maximum length p. For the sake of comparison, we adopt the same procedure using PCA (where the matrix V represents r principal components; but note that PCA requires all element to be the same length, which we fix to p).

In general, once learned the Dictionary V, X can be reconstructed by computing $U \in \mathbb{R}^{n \times r}$, where U represents the coefficients of the atoms computed with sparse coding (see Algorithm 2), or the PCA coefficients $U = XV$. Using the coefficients, a new reconstruction matrix $X_{rec} = UV^T$ (with $X_{rec} \in \mathbb{R}^{n \times p}$) can be obtained, which can be compared with the original X, to calculate a reconstruction error RMS. However, this method would only measure how well the learned dictionary represents the learned trajectories, not how well it generalises.

For this, we use a more compelling methodology, inspired by the *missing pixel method*: we create "holes" in the original trajectories (by removing consecutive columns of X) and measure how well our method permits to reconstruct them. More specifically, we firstly build a *restricted* matrix $X' \in \mathbb{R}^{n \times p'}$, with $p' < p$, by removing $p - p'$ (consecutive) columns from X. Therefore, we build a *restricted* dictionary $V' \in \mathbb{R}^{p' \times r}$ (o r PCA components), by removing $p - p'$ rows from V. We then compute the matrix $U' \in \mathbb{R}^{n \times r}$, using X' and V' (instead of X and V as above). Because U' has the same matrix dimensions as U, it is possible to use the original matrix V to compute $X'_{rec} = U'V^T$, with $X'_{rec} \in \mathbb{R}^{n \times p}$ that has the same dimension of original matrix X, thus resulting in a "full" trajectory with reconstructed holes.

Finally, we calculate the reconstruction error RMS using a *Frobenius* norm $\|\cdot\|_F$, which considers the difference between a target trajectory $X \in \mathbb{R}^{n \times p}$ and X'_{rec}.

$$RMS \equiv \left\|X - X'_{rec}\right\|_F = \sqrt{\sum_i^n \sum_j^p \left(x_{ij} - x'_{rec,ij}\right)^2} \tag{6}$$

We adopt the same (missing pixel) procedure for both SRSSD and PCA, for test and validation. Furthermore, to test the robustness of the method, we use the learned dictionaries to reconstruct different portions of trajectories, from 10% to 90% – the latter coming closer to reconstructing full trajectories.

Algorithm 1 $SRSDD(\mathbf{X}, r, \eta, \lambda, stop_criterion)$

Require: matrix $\mathbf{X}$, number r of elements in the dictionary,
sparse coding parameter η, sparse atoms parameter λ
a termination condition "*stop_criterion*"

Ensure: Dictionary $\mathbf{V}$, Coefficients $\mathbf{U}$

initialize random $\mathbf{V}$ and $\mathbf{U}$
while (**check** $stop_criterion$ is not satisfied) **do**

stage S1 – *update* H
compute $\mathbf{y}^k = (\|\mathbf{d}^1 \circ \mathbf{V}^k\|_2, \|\mathbf{d}^2 \circ \mathbf{V}^k\|_2, \ldots, \|\mathbf{d}^s \circ \mathbf{V}^k\|_2)$ for $k = 1, \ldots, r$
compute $\bar{h}_{ki} = |y_i^k|$ for $k = 1, \ldots, r$ and $i = 1, \ldots, s$
compute $\mathbf{H}_k \leftarrow max\{\bar{\mathbf{H}}_k, \varepsilon\}$ with $\varepsilon \ll 1$

stage S2 – *sparse coding*
update $\mathbf{U}$ minimizing $\min_{\mathbf{U}} \frac{1}{np}\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 + \eta \sum_{j=1}^n \|\mathbf{U}_j\|_1$ by IST

stage S3 – *structured atoms*
compute $z_{jk} = \left\{\sum_{i=1}^s \left(d_j^i\right)^2 (h_{ki})^{-1}\right\}^{-1}$ for $k = 1, \ldots, r$ and $i = 1, \ldots, s$
compute
$\mathbf{V}^k \leftarrow Diag(\mathbf{Z}^k) Diag(\|\mathbf{U}^k\|_2^2 \mathbf{Z}^k + \lambda \mathbf{I})^{-1} (\mathbf{X}^T \mathbf{U}^k - \mathbf{V}\mathbf{U}^T \mathbf{U}^k + \|\mathbf{U}^k\|_2^2 \mathbf{V}^k)$
normalize the columns of $\mathbf{V}$ to have unit norm

end while

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.02.967489; this version posted January 9, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Algorithm 2 *Sparse Coding*$(\mathbf{X}, \mathbf{V}, \eta, stop_criterion)$

Require: matrix $\mathbf{X}$, Dictionary $\mathbf{V}$,
sparse coding parameter η, a termination condition "*stop_criterion*"
Ensure: Coefficients $\mathbf{U}$
1: **initialize** random $\mathbf{U}$
2: **while** (**check** *stop_criterion* is not satisfied) **do**
3: **update** $\mathbf{U}$ minimizing $\min_{\mathbf{U}} \frac{1}{np}\|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_F^2 + \eta \sum_{j=1}^{n} \|\mathbf{U}_j\|_1$ by IST
4: **end while**

References

1. Iberall, T., Bingham, G. & Arbib, M. A. Opposition space as a structuring concept for the analysis of skilled hand movements. *Exp. Brain Res. Ser.* **15**, 158–173 (1986).
2. Santello, M., Flanders, M. & Soechting, J. F. Postural hand synergies for tool use. *J. Neurosci.* **18**, 10105–10115 (1998).
3. Mussa-Ivaldi, F. A. & Bizzi, E. Motor learning through the combination of primitives. *Philos Trans R Soc Lond B Biol Sci* **355**, 1755–1769 (2000).
4. Mason, C. R., Gomez, J. E. & Ebner, T. J. Hand synergies during reach-to-grasp. *J Neurophysiol* **86**, 2896–910 (2001).
5. D'Avella, A. & Bizzi, E. Shared and specific muscle synergies in natural motor behaviors. *Proc. Natl. Acad. Sci.* **102**, 3076–3081 (2005).
6. d'Avella, A., Portone, A., Fernandez, L. & Lacquaniti, F. Control of fast-reaching movements by muscle synergy combinations. *The J. Neurosci.* **26**, 7791–7810 (2006).
7. Donnarumma, F., Prevete, R. & Trautteur, G. Programming in the brain: a neural network theoretical framework. *Connect. Sci.* **24**, 71–90 (2012).
8. Pezzulo, G., Donnarumma, F., Iodice, P., Maisto, D. & Stoianov, I. Model-based approaches to active perception and control. *Entropy* **19**, 266 (2017).
9. Tchernichovski, O. & Golani, I. A phase plane representation of rat exploratory behavior. *J. neuroscience methods* **62**, 21–27 (1995).
10. Tchernichovski, O., Benjamini, Y. & Golani, I. The dynamics of long-term exploration in the rat. *Biol. cybernetics* **78**, 423–432 (1998).
11. Drai, D., Benjamini, Y. & Golani, I. Statistical discrimination of natural modes of motion in rat exploratory behavior. *J. neuroscience methods* **96**, 119–131 (2000).
12. Dudchenko, P. A. & Wallace, D. Neuroethology of spatial cognition. *Curr. Biol.* **28**, R988–R992 (2018).
13. Golani, I. The developmental dynamics of behavioral growth processes in rodent egocentric and allocentric space. *Behav. Brain Res.* **231**, 309 – 316, DOI: https://doi.org/10.1016/j.bbr.2012.01.039 (2012).
14. Thompson, S. M., Berkowitz, L. E. & Clark, B. J. Behavioral and neural subsystems of rodent exploration. *Learn. Motiv.* **61**, 3 – 15, DOI: https://doi.org/10.1016/j.lmot.2017.03.009 (2018).
15. d'Avella, A. Modularity for motor control and motor learning. In *Progress in Motor Control*, 3–19 (Springer, 2016).
16. Brown, A. E., Yemini, E. I., Grundy, L. J., Jucikas, T. & Schafer, W. R. A dictionary of behavioral motifs reveals clusters of genes affecting caenorhabditis elegans locomotion. *Proc. Natl. Acad. Sci.* **110**, 791–796 (2013).
17. Berman, G. J., Choi, D. M., Bialek, W. & Shaevitz, J. W. Mapping the stereotyped behaviour of freely moving fruit flies. *J. The Royal Soc. Interface* **11**, 20140672 (2014).
18. Geddes, C. E., Li, H. & Jin, X. Optogenetic editing reveals the hierarchical organization of learned action sequences. *Cell* **174**, 32–43 (2018).
19. Gehring, T. V., Luksys, G., Sandi, C. & Vasilaki, E. Detailed classification of swimming paths in the morris water maze: multiple strategies within one trial. *Sci. reports* **5**, 14562 (2015).
20. Schmitzer-Torbert, N. & Redish, A. D. Development of path stereotypy in a single day in rats on a multiple-t maze. *Arch. italiennes de biologie* **140**, 295–301 (2002).

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.02.967489; this version posted January 9, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

21. Wiltschko, A. B. *et al.* Mapping sub-second structure in mouse behavior. *Neuron* **88**, 1121–1135 (2015).

22. Johnson, M. J., Duvenaud, D. K., Wiltschko, A., Adams, R. P. & Datta, S. R. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in neural information processing systems*, 2946–2954 (2016).

23. Matsumoto, J., Nishimaru, H., Ono, T. & Nishijo, H. 3d-video-based computerized behavioral analysis for in vivo neuropharmacology and neurophysiology in rodents. In *In Vivo Neuropharmacology and Neurophysiology*, 89–105 (Springer, 2017).

24. Shan, H. & Mason, P. Multiscale dictionary of rat locomotion. *arXiv preprint arXiv:1707.03360* (2017).

25. Vouros, A. *et al.* A generalised framework for detailed classification of swimming paths inside the morris water maze. *Sci. reports* **8**, 15089 (2018).

26. Datta, S. R. Q&a: Understanding the composition of behavior. *BMC biology* **17**, 1–7 (2019).

27. Tessitore, G., Sinigaglia, C. & Prevete, R. Hierarchical and multiple hand action representation using temporal postural synergies. *Exp. brain research* **225**, 11–36 (2013).

28. Gallego, J. A., Perich, M. G., Miller, L. E. & Solla, S. A. Neural manifolds for the control of movement. *Neuron* **94**, 978–984 (2017).

29. Vinjamuri, R., Lee, H. & Mao, Z. Dimensionality reduction in control and coordination of the human hand. *IEEE Trans Biomed Eng* **57**, 284–295 (2010).

30. Vinjamuri, R. *et al.* Temporal postural synergies of the hand in rapid grasping tasks. *Inf. Technol. Biomed. IEEE Transactions on* **14**, 986–994 (2010).

31. Jenatton, R., Mairal, J., Bach, F. R. & Obozinski, G. R. Proximal methods for sparse hierarchical dictionary learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 487–494 (2010).

32. Prevete, R., Donnarumma, F., D'Avella, A. & Pezzulo, G. Evidence for sparse synergies in grasping actions. *Sci. Reports* **8**, 616 (2018).

33. Tessitore, G. & Prevete, R. Designing structured sparse dictionaries for sparse representation modeling. In *Computer Recognition Systems 4*, 157–166 (Springer, 2011).

34. Lee, H., Battle, A., Raina, R. & Ng, A. Y. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems 19*, 801–808 (Massachusetts Institute of Technology Press, 2007).

35. Jenatton, R., Obozinski, G. & Bach, F. Structured sparse principal component analysis. *Int. Conf. on AISTATS* (2010).

36. Amancio, D., Oliveira, O. & Costa, L. d. F. On the concepts of complex networks to quantify the difficulty in finding the way out of labyrinths. *Phys. A: Stat. Mech. its Appl.* **390**, 4673–4683 (2011).

37. Saar, M. *et al.* The interplay between maze complexity, colony size, learning and memory in ants while solving a maze: A test at the colony level. *PloS one* **12**, e0183753 (2017).

38. Bagnall, A. J. & Zatuchna, Z. V. On the classification of maze problems. In *Foundations of Learning Classifier Systems*, 305–316 (Springer, 2005).

39. Rolls, E. T., Stringer, S. M. & Elliot, T. Entorhinal cortex grid cells can map to hippocampal place cells by competitive learning. *Network: Comput. Neural Syst.* **17**, 447–465 (2006).

40. Si, B. & Treves, A. The role of competitive learning in the generation of dg fields from ec inputs. *Cogn. neurodynamics* **3**, 177–187 (2009).

41. Jackson, J. C., Johnson, A. & Redish, A. D. Hippocampal sharp waves and reactivation during awake states depend on repeated sequential experience. *J. Neurosci.* **26**, 12415–12426 (2006).

42. van der Meer, M. A., Knierim, J. J., Yoganarasimha, D., Wood, E. R. & van Rossum, M. C. Anticipation in the rodent head direction system can be explained by an interaction of head movements and vestibular firing properties. *J. neurophysiology* **98**, 1883–1897 (2007).

43. Johnson, A., Fenton, A., Kentros, C. & Redish, A. D. Looking for cognition in the structure within the noise. *Trends Cogn Sci* **13**, 55–64 (2009).

44. Bruckstein, A., Donoho, D. & Elad, M. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Rev.* **51**, 34–81 (2009).

45. Mairal, J., Bach, F., Ponce, J., Sapiro, G. & Zisserman, A. Supervised dictionary learning. *CoRR* **abs/0809.3083** (2008).

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.02.967489; this version posted January 9, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

46. Mairal, J., Bach, F., Ponce, J. & Sapiro, G. Online Learning for Matrix Factorization and Sparse Coding. *J. Mach. Learn. Res.* **11**, 19–60 (2010).

47. Zhang, Q., Liu, Y., Blum, R. S., Han, J. & Tao, D. Sparse representation based multi-sensor image fusion for multi-focus and multi-modality images: A review. *Inf. Fusion* **40**, 57–75 (2018).

48. Lewicki, M. S. & Sejnowski, T. J. Learning overcomplete representations. *Neural computation* **12**, 337–365 (2000).

49. Zhang, Q., Shi, T., Wang, F., Blum, R. S. & Han, J. Robust sparse representation based multi-focus image fusion with dictionary construction and local spatial consistency. *Pattern Recognit.* **83**, 299–313 (2018).

50. Zou, H., Hastie, T. & Tibshirani, R. Sparse Principal Component Analysis. *J. Comput. Graph. Stat.* **15** (2004).

51. Krishnapuram, B., Carin, L., Figueiredo, M. A. T. & Hartemink, A. J. Sparse multinomial logistic regression: fast algorithms and generalization bounds. *Pattern Analysis Mach. Intell. IEEE Transactions on* **27**, 957–968 (2005).

52. Elad, M. & Aharon, M. Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries. *Image Process. IEEE Transactions on* **15**, 3736–3745 (2006).

53. Aharon, M., Elad, M. & Bruckstein, A. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Process.* **54**, 4311–4322 (2006).

54. Shen, H. & Huang, J. Sparse principal component analysis via regularized low rank matrix approximation. *J. Multivar. Analysis* **99**, 1015–1034 (2008).

55. Engan, K., Aase, S. & Hakon Husoy, J. Method of optimal directions for frame design. In *Proceedings of ICASSP '99*, vol. 5, 2443–2446 (IEEE Computer Society, 1999).

56. Jenatton, R., Audibert, J. & Bach, F. Structured variable selection with sparsity-inducing norms. Tech. Rep., arXiv:0904.3523 (2009).

57. Micchelli, C. & Pontil, M. Learning the kernel function via regularization. *J. Mach. Learn. Res.* **6**, 1099–1125 (2005).

58. Basso, C., Santoro, M., Verri, A. & Villa, S. Paddle: Proximal algorithm for dual dictionaries learning. *CoRR* **abs/1011.3728** (2010).

59. Grippo, L. & Sciandrone, M. On the convergence of the block nonlinear Gauss-Seidel method under convex constraints. *Oper. Res. Lett.* **26**, 127–136 (2000).

60. Chen, S., Donoho, D. & Saunders, M. Atomic decomposition by basis pursuit. *SIAM J. on Sci. Comput.* **20**, 33–61 (1998).

61. Tropp, J. Greed is good: Algorithmic results for sparse approximation. *IEEE Trans. Inform. Theory* **50**, 2231–2242 (2004).

62. Daubechies, I., Defrise, M. & De Mol, C. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. on Pure Appl. Math.* **57**, 1413–1457 (2004).

Acknowledgments

This research has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreements No. 785907 and 945539 (Human Brain Project SGA2 and SGA3 to GP), the European Research Council under the Specific Grant Agreement No. 820213 (ThinkAhead to GP) and the Human Frontier Science Program (grant no. RGY0088/2014 to GP, MVDM and CK). The GEFORCE Titan GPU card used for this research was donated by the NVIDIA Corp. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Author contributions statement

F.D, R.P., M.v.d.M., C.K. and G.P. conceived the experiment(s), F.D., R.P., D.M., S.F., E.I. and G.P. conducted the experiment(s). All authors reviewed and analysed the manuscript.