

High-Dimensional Analysis of Evolutionary Autonomous Agents

Lior Segev¹, Ranit Aharonov², Isaac Meilijson³, Eytan Ruppin^{1,4,†}

¹School of Computer Sciences, Tel-Aviv University, Tel-Aviv, Israel
lior@cns.tau.ac.il, ruppin@post.tau.ac.il (tel: +972-3-6406528)

²Center for Neural Computation, The Hebrew University, Jerusalem, Israel
ranit@alice.nc.huji.ac.il (tel: +972-3-6407864)

³School of Mathematical Sciences, Tel-Aviv University, Tel-Aviv, Israel
isaco@post.tau.ac.il (tel: +972-3-6408826)

⁴School of Medicine, Tel-Aviv University, Tel-Aviv, Israel

[†]To whom correspondence should be addressed.

Abstract

This paper presents a new approach to address the important challenge of localizing function in a neurocontroller. The approach is based on the basic Functional Contribution Analysis (FCA) presented earlier, which assigns contribution values to the elements of the network, such that the ability to predict the network's performance in response to multi-unit lesions is maximized. These contribution values quantify the importance of each element to the tasks the agent performs. Here we present a generalization of the basic FCA to high-dimensional analysis, using high-order compound elements. Such elements are composed of conjunctions of simple elements. Their usage enables the explicit expression of sets of neurons or synapses whose contributions are interdependent, a prerequisite for localizing the function of complex neurocontrollers. High-dimensional FCA is shown to significantly improve on the accuracy of the basic analysis, to provide new insights concerning the main subsets of simple elements

in the network that interact in a complex non-linear manner, and to systematically reveal the types of interactions that characterize the evolved neurocontroller.

Keywords: neurocontroller analysis, localization of function, performance prediction, lesioning.

1 Introduction

A major effort in the field of Artificial Life is dedicated to the development of intelligent agents. Considerably less effort has been given to understanding their inner workings. In this paper we present a functional contribution analysis (FCA) approach that quantitatively estimates the contribution of each element in the agent’s brain to its functioning, and hence may serve to localize different functional tasks performed by the agent’s neurocontroller.

Several studies have attempted to analyze neurocontrollers of evolutionary autonomous agents (EAAs). In [5, 4], a rigorous, quantitative analysis of the dynamics of central pattern generator (CPG) networks evolved for locomotion, has been developed. The networks evolved are of very small size, composed of 3,4 or 5 neurons. A high-level description of the dynamics of these CPG networks was developed, based on the concept of a dynamical module: a set of neurons that have a common temporal behavior, making a transition from one quasi-stable state of firing to another together. Dynamical modules give new insights to CPG operation, describing them in terms of a finite state machine, and enabling a rigorous analysis of their robustness to parameter variations. In [10], the activity of internal neurocontroller neurons as a function of a robot’s location and orientation was charted by a simple form of receptive-field measurement. Neuronal functioning was generally highly distributed, but a specific interneuron that had an important role in path planning was also identified. Other researchers have studied the effects of clamping neuronal activity on the robot’s behavior (for example, inducing rotation, straight-line motion, or more complex behaviors such as smooth tracking of moving targets [11]). The “command” neurons described in [3] were discovered by studying the agent’s behavior following single lesions and by receptive field analysis. Finally, a more “procedural” kind of ablation, in which different processes (and not just units or links) are systematically

canceled out was recently employed in [23]. Overall, these studies have provided only glimpses of the neural processing that takes place in EAAs’ neurocontrollers.

In neuroscience, the localization of specific tasks in neural systems is conventionally done either by recording the activity of system elements during behavior, or by measuring the deficit in performance after lesioning specific elements. The term lesioning here refers to a disruption of the activity of the lesioned element. This can be done by killing the element (e.g. cutting out a brain region), deactivating it by cooling, or disrupting its activity using magnetic stimulation (TMS). Obviously inferring significance from recording unit activity during behavior is problematic because correlation of neuronal activity with performance does not necessarily identify causality. The method of lesioning single elements (e.g., [22, 9]) has the disadvantage that it cannot unravel the significance of units that interact in non-linear ways, and hence it cannot in general predict performance under multi-unit lesions. In particular, single lesion experiments cannot unravel the importance of units that have a high degree of redundancy. The limited nature of such experiments has been widely discussed in the literature [21, 20, 8].

The basic 1-dimensional FCA, presented and developed in [2], addresses the challenges posed by the lesioning approach. By utilizing multiple *multi-lesion* experiments, FCA assigns specific “*contribution values*” to the basic elements of the neurocontroller. These contribution values quantify the importance of each element to the task(s) the agent performs. They enable the accurate prediction of the agent’s performance in any new, unseen lesioned state. The current paper examines in more depth how the FCA may be utilized to study function localization in EAA neurocontrollers. To this end, the basic FCA is generalized to high-dimensional analysis, using high-order compound elements. Such elements are composed of conjunctions of simple elements, and enable the explicit expression of sets of neurons or synapses whose contributions are interdependent, i.e., the contribution of each of the simple elements

depends on the state of the other elements in the set. This high-dimensional description is important, indeed essential, for an accurate analysis of EAA neurocontrollers, in which the interactions between elements may be high-dimensional and complex. The introduction of compound elements requires a re-thinking of the concept of the *contributions* of simple elements defined originally in the basic FCA. While the contributions of the simple elements can be reconstructed regardless of the dimension of the FCA, one needs to think in terms of the contributions of compound elements to capture the interactions forming functional groups in the network.

The FCA can be applied to several different tasks performed by the agent. For each function (task), the FCA computes a separate contribution vector describing the contribution of the system elements to that specific function. Thus, it is a method for function localization, detailing which elements participate in which functions. Using the contribution vectors for the different tasks, we have defined quantitative measures of function localization and element specialization in the network. For a description and discussion of these measures see [2, 19]. In this paper we present results of the High-D FCA in the general task of maximizing fitness. However, since the contributions of the simple elements can be reconstructed, it also supplies the basis for measuring function localization.

High-dimensional FCA is carried out via an efficient algorithm that selects the most important compound elements out of a possibly large set of candidates. A detailed high-dimensional analysis of an agent’s neurocontroller is performed, and the structure of the emerging higher-order compound elements is explored. Capitalizing on our ability to perform multi-lesion experiments in EAAs in a computationally tractable manner, high-dimensional FCA is shown to be a new method enabling a systematic and rigorous analysis of function localization in EAA neurocontrollers.

This paper is organized as follows: section 2 presents the EAA model used and the experimental protocol. Section 3 reviews the basic, 1-dimensional FCA. Section 4

extends the basic FCA to higher dimensions. Section 5 describes an in-depth high-dimensional analysis of an EAA neurocontroller. Our results and their implications to the analysis of EAAs are discussed in section 6.

2 The EAA Environment

The EAAs analyzed here live in a discrete 2D grid “world” surrounded by walls (after [3]). The grid is 30 by 30 cells, with a 10 by 10 “food zone” located in a corner. 250 poison items are randomly scattered in the world and 30 food items are randomly scattered within the food zone. The agent’s goal is to find and eat as many food items as possible during its life, while ignoring the poison items. The final score (fitness) of the agent is the number of food items minus the number of poison items it consumes, divided by 30, so that the maximum possible fitness is 1. Since the agent’s lifetime is short, this is practically an unattainable score. The agent is equipped with a set of sensors, motors, and a fully recurrent synchronous neurocontroller of binary neurons. The neurocontroller is coded in the genome and evolved; the sensors and motors are given and kept constant.

The agent’s neurocontroller consists of 10 internal and output, recurrently connected neurons. In addition, the agent has five input neurons connected to the sensors (see Figure 1, after [3]). Four of the sensors encode the presence of a resource (food or poison, without distinction between the two), a wall, or an empty cell in the cell the agent occupies and in the three cells directly in front of it. The fifth sensor is a “smell” sensor which can differentiate between food and poison underneath the agent, but gives a random reading if the agent is in an empty cell. The four motor neurons initiate movement forward, a turn left or right, and control the state of the mouth (open or closed). Importantly, eating only takes place if the agent is neither moving nor turning. Thus, eating is a costly process requiring a time step with no

other movement, in a lifetime of limited time steps. The task is difficult because only partial sensory information about their environment is available to agents.

[Figure 1 about here]

The agents are developed via an evolutionary process. An initial population of 100 agents with random neurocontrollers undergoes selection and crossover for a large number of generations, between 5,000 and 30,000. In each generation all agents are placed in a randomly generated grid world at a random initial position and orientation (each world contains one agent), and allowed to move for 150 steps. In the *last* generation all the agents are tested in 5,000 different random worlds to obtain an accurate estimation of their performance.

Previous analysis [3] revealed that the most successful agents evolved rely on a switch between two behavioral strategies: exploration and grazing. Exploration consists of moving in straight lines, ignoring all resources except food items in the cell the agent is in, and turning at walls. Grazing consists of turning to resources to the left or right in order to examine them, and maintaining the agent's position in a fairly restricted area. In both modes the agent consumes food items that it steps upon. Exploration is mostly observed when the agent is out of the food zone, allowing it to explore the environment and find the food zone. Inside the food zone, however, successful agents almost always display grazing behavior, which results in efficient consumption of food. It is important to emphasize that the switch between these two behavioral modes occurs even though the agent has no knowledge about its position coordinates.

Analysis of successful agents revealed that they possess one or more *command neurons* that determine the agent's behavioral strategy. Artificially clamping these command neurons to either constant firing activity or to complete quiescence causes

the agent to constantly maintain one of the two behavioral modes, regardless of its sensory input. These command neurons emerge even though the agents receive no explicit information about their position. Their activity is not position dependent, but rather tied to the existence of food items. This computational ability is based on a spontaneously emerging stochastic memory of the time elapsed since the last eating event [3]. In these agents feedback from the motor neurons is necessary to supply information about eating events.

Throughout this paper, we focus on the analysis of one successful agent with a neurocontroller consisting of 10 neurons. This agent achieved a fitness score close to 0.4, which is above the performance level obtained by several manually designed algorithms, as well as ones obtained through reinforcement learning [3]. In [19], we have presented a one-dimensional analysis that localizes the two behavioral sub-tasks (grazing and exploration) and the agent’s overall survival task in its neurocontroller. Here we focus on a detailed high-dimensional analysis of the former, i.e., of the localization of its overall survival task.

3 One-dimensional FCA

3.1 The Concept

We measure the performance (fitness) of the agent under different lesioning configurations. Each configuration specifies which of the agent’s neurocontroller units (be they neurons, synapses, or any higher-order module) is lesioned. The FCA algorithm is designed to use this data in order to search for the *contribution vector* $\mathbf{c} = (c_1, \dots, c_N)$, where c_i is the contribution of element i to the task in question and N is the number of elements in the network. These values provide the best prediction of the agent’s performance in terms of Mean Squared Error (MSE), under all possible multi-site

lesions.

More precisely, suppose that a set of elements in the network is lesioned and the agent is tested for its performance level. The lesioning configuration is described by the vector \mathbf{m} where $m_i = 0$ if the element is lesioned, and $m_i = 1$ if it is intact. Within the FCA framework, the prediction of performance in this lesioned state is based on a linear model generalized by a nonlinear transformation. Given a contribution vector \mathbf{c} and a function f , the predicted performance $\tilde{p}_{\mathbf{m}}$ is given by

$$\tilde{p}_{\mathbf{m}} = f(\mathbf{m} \cdot \mathbf{c}) . \tag{1}$$

The function f is a non-decreasing piecewise polynomial. It is non-decreasing to reflect the notion that beneficial elements (those whose lesioning results in performance deterioration) should have positive contribution values, and that negative values indicate elements that hinder performance.

Denoting by $p_{\mathbf{m}}$ the actual performance of the network under lesioning configuration \mathbf{m} , the mean squared prediction error is

$$MSE = \frac{1}{2^N} \sum_{\{\mathbf{m}\}} (\tilde{p}_{\mathbf{m}} - p_{\mathbf{m}})^2 . \tag{2}$$

The vector \mathbf{c} which minimizes this error is the *contribution vector* for the task tested, and the corresponding f is its adjoint *performance prediction function*. Since we can arbitrarily choose the scale of c and maintain the same prediction by modifying f , we normalize \mathbf{c} such that $\sum_{i=1}^N |c_i| = 1$.

3.2 The FCA Algorithm

The goal of the FCA algorithm is to find the contribution vector \mathbf{c} and the performance prediction function f which minimize Eq. (2) given a subset of the full 2^N

configuration set. The optimal \mathbf{c} and f are determined using a training set T of n lesioning configurations \mathbf{m} and the accompanying performance levels $p_{\mathbf{m}}$. The FCA algorithm works as follows:

1. **Initialize** \mathbf{c} by selecting each element c_i randomly in the range $[-1, 1]$. Normalize \mathbf{c} such that $\sum_{i=1}^N |c_i| = 1$. Compute f as in step 3.
2. **Compute** \mathbf{c} by gradient descent to minimize Eq. (2) while keeping f fixed. Normalize \mathbf{c} .
3. **Compute** f to minimize Eq. (2) while keeping \mathbf{c} fixed, by performing an isotone regression on the pairs $\{\mathbf{m} \cdot \mathbf{c}, p_{\mathbf{m}}\}$, and smoothing the result with a cubic spline.

Steps 2 and 3 are repeated a fixed number of times.

The FCA is related to two known modeling and prediction methods, that of Generalized Linear Modeling and that of Projection Pursuit Regression. The Generalized Linear Model (GLM, see [17]) approach can be succinctly defined by the relation $g(\tilde{p}_{\mathbf{m}}) = \mathbf{m} \cdot \mathbf{c}$, where g is the transfer function akin to the performance prediction function f in the FCA formulation (if $g = f^{-1}$ the above equation reduces to Eq. 1). In GLM, the contribution vector \mathbf{c} can be computed via a closed formula, and hence results in a much faster and deterministic computation than that performed within the FCA. However, GLM cannot be used to predict performance since g is not reversible in general; many lesioning configurations can lead to the same performance (e.g., complete failure of the agent, leading to performance value 0). Projection Pursuit Regression (PPR, see [13]) has a formulation very similar to the FCA. PPR approximates the response surface g by a sum of ridge functions $g(\mathbf{x}) \sim \sum_{j=1}^k f_j(\mathbf{c}_j \cdot \mathbf{x})$, which, together with the projection directions \mathbf{c}_j are found in an iterative, greedy manner, minimizing the distance from a target random variable Y_i , $\sum_{j=1}^k (f_j(\mathbf{c}_j \cdot \mathbf{x}) - Y_i)^2$. The FCA is hence a special case of PPR in which $k = 1$ and f is non-decreasing.

Using only one ridge function and a non-decreasing f enables one to preserve the intuitive meaning of \mathbf{c} as the contribution values vector.

3.3 The Experimental Protocol

The FCA algorithm is stochastic, and as such is dependent on both the initial conditions and the random choices made in the gradient descent step. To reduce stochasticity, each of the MSE values reported in this paper is a mean of 10 FCA runs. A single run consists of 10 *trials*, each of which is initialized with a different random contribution vector. The result of the trial with the lowest MSE on the *training* set is chosen for that run. The FCA executes 150 iterations of steps 2 and 3 above. The MSE values are normalized by dividing them by the variance of the agent’s performances $p_{\mathbf{m}}$ in the test set. Thus, an MSE of q means that the FCA explains a fraction $1 - q$ of the variance. That is, if one would predict for any configuration a performance level equal to the mean performance of the agent, then the normalized MSE would equal 1, corresponding to $R^2 = 0$.

In the results presented below, the lesioning of a neuron was performed by making its firing pattern random (*stochastic lesioning*), rather than by completely silencing its output (*biological lesioning*). More precisely, at every time step a lesioned neuron fires with probability equal to its overall mean firing rate under normal conditions, independent of its input field. This ensures that the lesioning does not affect the mean field of other neurons, influencing only the information content received from the lesioned neuron. For a comparison between the lesioning schemes see [2]. When lesioning motor neurons, we do not alter the activity transmitted to the motors themselves. This enables us to isolate the role of the motor units in the computation of the recurrent controller network (i.e., their contribution to other neurons), without immobilizing the agent.

3.4 One-dimensional Agent Analysis

The basic FCA was studied in [1, 19, 2] on several agents and a variety of tasks. For the agent considered in this paper, we used the basic, 1D-FCA both on the neuronal level, i.e., when the simple elements are the 10 neurons of the network, and on the synaptic level. To demonstrate the workings of the basic FCA, we briefly show the results of this neuronal analysis here. For simplicity of this presentation, here the full 2^{10} configuration set was used to train the FCA. In section 3.5, we present an algorithm for the selection of the training set, and demonstrate that a small training set suffices.

[Figure 2 about here]

Figure 2A depicts the contribution values of the agent’s neurons. The computed contribution values are consistent over different runs of the FCA (see standard deviation bars), testifying to the significance of the results. The FCA correctly identifies the significant neurons known from previous analysis [3], the motor neurons (neurons 1–3) that are important for feedback, the command neuron (neuron 5), and another interneuron (neuron 10). Panel B depicts the shape of the performance prediction function. As can be seen, when all neurons are lesioned ($\mathbf{m} \cdot \mathbf{c} = 0$), the agent’s performance is about 0.3 of the intact network ($\mathbf{m} \cdot \mathbf{c} = 1$) baseline performance. The performance is not 0 because the input neurons are left intact, as well as the activity transmitted from the output neurons to the motors. Thus, the fully lesioned network corresponds to the minimal network of input and output neurons with no internal processing. Lesioning any of the five identified significant neurons results in a quite sharp decrease in performance as is evident from the large slope of f near 1. The function allows one to fairly accurately predict the performance of any lesioned state

using the contribution values depicted in panel A (Eq. (1)). Indeed, the normalized MSE (Eq. (2)) in this case is 0.015 (explaining 98.5% of the variance).

3.5 Fast FCA: Adaptive Lesion Selection

The analysis above was performed using the full set of lesioning configurations. It is crucial, however, that the FCA generalizes well when using only a small subset of the full configuration set for training. As Figure 3 demonstrates, using small training sets yields good prediction capabilities for unseen configurations even when the set is *randomly* selected. Yet, it is important to optimize the choice of the training set, i.e. judiciously select a small subset of lesions that will maximize the FCA accuracy for a given training size. For this purpose, we have developed the *Adaptive Lesioning (AL) algorithm*, which iteratively selects the next lesioning configuration to be evaluated, based on the configuration set used so far:

1. Create a random initial core set of N configurations¹. Compute \mathbf{c} and f using the FCA.
2. From all possible configurations that are not yet in the current set, find the configuration whose *estimated* performance (using Eq. (1)) is farthest from the *known* performance values of the configurations currently in the training set. That is, given the current configuration set T , choose the next lesioning configuration \mathbf{m} such that

$$\mathbf{m} = \arg \max_{\mathbf{m}' \notin T} \left\{ \min_{\mathbf{m}'' \in T} |p_{\mathbf{m}''} - f(\mathbf{m}' \cdot \mathbf{c})| \right\} \quad (3)$$

3. Add \mathbf{m} to T , recompute \mathbf{c} and f using the FCA, and return to step 2.

¹Since \mathbf{c} has dimension N , the training set must consist of at least N configurations. The set always includes the all-intact and the all-lesioned configurations.

Steps 2 and 3 are repeated until either a predefined number of training examples is reached, or the change in the test prediction error falls below a threshold criterion. As can be seen in Figure 3, for any training set size, the AL algorithm results on average in a lower MSE on the test set than randomly choosing the training set. Also, a very small training set (about 40 configurations) selected with the AL algorithm suffices to reach the test error achieved when training on the full configuration set (1024 configurations). Moreover, the AL algorithm is much more consistent in finding good training sets (see error bars in Figure 3). Random sets are prone to ineffective sampling of the full configuration space, and thus more often result in a large MSE on the test set.

[Figure 3 about here]

The Adaptive Lesioning algorithm is closely related to the field of *adaptive learning* [6, 7], which seeks to improve both the generalization and speed of machine learning algorithms. Attempting to uniformly sample the performance values p , the AL algorithm effectively samples the values of $\mathbf{m} \cdot \mathbf{c}$ with density proportional to the slope of f . Thus, the resulting FCA is accurate because it samples more data in the more crucial regions, those in which small perturbations in $\mathbf{m} \cdot \mathbf{c}$ result in large variations in the performance $f(\mathbf{m} \cdot \mathbf{c})$ (see Figure 4).

[Figure 4 about here]

4 High-dimensional FCA

4.1 Motivation and Definition

The results shown briefly above and in more detail in [2, 19] demonstrate that the contributions of the units of simple EAA neurocontrollers can be computed with the basic 1D-FCA. However, to deepen one’s understanding of the agents’ neurocontrollers, it is imperative to look further and examine the nature of the interactions between these units. The units of the controller may interact in such a way that the contribution value of a unit is not a single constant value but rather depends on the state (lesioned or intact) of the other units. Indeed, certain forms of interaction cannot *in principle* be described by the basic FCA. A classical example of such an interaction was given by Sprague in the neuroscience literature [21], showing that deficits in orienting towards a stimulus, resulting from large cortical visual lesions, can be reversed through additional removal of contralateral areas. This has been known as the Sprague effect, or paradoxical lesioning [12, 16]. Such effects have posed an intriguing conceptual challenge: if an area is beneficial for a behavior (lesioning it hinders performance), how can its lesioning under a different condition, i.e., when another structure is already lesioned, improve the behavior? The solution is that the utility of such a unit is *context-dependent* – it depends on the state of the rest of the network. To address these challenges the FCA has now been generalized to include high-dimensional context-dependent interactions. This generalization is based on the usage of *compound elements*. Extending upon the simple, single-unit elements used in the FCA, a compound element $\pi = \{\pi_1, \pi_2, \dots, \pi_k\}$ denotes a specific combination of a few simple elements. The *order* k of a compound element is the number of simple elements composing it ($k = |\pi|$), and this determines the dimensionality of the FCA analysis. In the basic FCA only simple elements of order 1 are used, and hence it is

denoted a 1D-FCA.

In high-dimensional FCA, we concatenate the lesioning state of the simple and compound elements to form the lesioning configuration vector \mathbf{m} , such that its length is now $N + N_c$, where N_c is the number of compound elements. A compound element is considered intact only if all of the simple elements composing it are intact. Analogously, \mathbf{c} is of length $N + N_c$ as well, describing the contributions of both simple and compound elements. Thus, the performance prediction given by Eq. (1) still holds in high-dimensional FCA.

Section 3.5 demonstrates that the 1D-FCA generalizes well when a sample set is used for training instead of the full configuration set. Throughout the paper, the training set used for the High-D analysis is the full configuration set, to maximize the accuracy of the results. However, in Section 5.3 we demonstrate that the High-D FCA also generalizes well and obtains a fairly accurate MSE with a relatively small sample of the full configuration set.

4.2 Selecting Compound Elements

Adding compound elements to the description of the system introduces the problem of selecting the most informative ones. The simplest iterative method is *greedy*: at each iteration one compound element is added to the set of elements until a stopping criterion is reached. To select the element to be added, the FCA is run on a training set including this candidate element and the current set of elements. The compound element which leads to the smallest MSE (on the training set) is selected and added to the element set. The cycle then repeats with the new, extended element set. The greedy algorithm is computationally very expensive, and hence impractical when the potential compound element set to be searched is large. A faster approach, suitable only for 2D-FCA, is to compute the FCA using all order-2 elements, select those which

have the highest contribution values, and then re-run the FCA using the reduced set of elements (the Highest Contributions algorithm). However, this approach requires $O(N^2)$ compound elements, and hence becomes intractable when N is large. We devised a more efficient approximation algorithm, called CERE (Compound Element Error Reduction Estimation) which uses *estimates* of the reduction in the prediction error that each conjunction yields.

CERE estimates the prediction error when a compound element π is added to the current set of elements by,

$$\Delta MSE_{\pi} = \left(\frac{1}{2^N} \frac{N_{\pi}}{n_{\pi}} \right) \sum_{\mathbf{m} \in T_{\pi}} ([f(\mathbf{m} \cdot \mathbf{c}) - p_{\mathbf{m}}]^2 - [f(\mathbf{m} \cdot \mathbf{c} + \tilde{c}_{\pi}) - p_{\mathbf{m}}]^2), \quad (4)$$

where T_{π} is the set of lesioning configurations from the training set that match π . A lesioning configuration \mathbf{m} is said to match an element π if all of the elements of π are intact in \mathbf{m} . The term $\left(\frac{1}{2^N} \frac{N_{\pi}}{n_{\pi}} \right)$ normalizes the error on the training set to the expected error over the complete set of 2^N configurations. N_{π} is the number of possible configurations that match π (equal to 2^{N-k} , where $k = |\pi|$ is the order of π), and n_{π} is the number of such configurations in the training set. \tilde{c}_{π} is the contribution value assigned to π , computed to maximize ΔMSE_{π} by a simple one-dimensional gradient descent. The CERE algorithm is comprised of the following steps:

1. Initialize the set of compound elements to the empty set. Compute $\{\mathbf{c}, f\}$ using the FCA on simple elements.
2. For each candidate compound element π ,
 - (a) Initialize \tilde{c}_{π} to 0.
 - (b) Update \tilde{c}_{π} via gradient descent to maximize ΔMSE_{π} (Eq. (4)):

$$\tilde{c}_{\pi} \leftarrow \tilde{c}_{\pi} + \epsilon \frac{\partial}{\partial \tilde{c}_{\pi}} \Delta MSE_{\pi}. \quad (5)$$

- (c) Iterate step 2b a given number of times.
3. Select the compound element π that maximizes the reduction in the prediction error ΔMSE_π , and add it to the set of compound elements.
 4. Recompute $\{\mathbf{c}, f\}$ using the FCA, using all N simple elements and all compound elements selected so far.

Steps 2-4 repeat until a stopping criterion is met: either a predefined number of compound elements is used, or the prediction error falls below a threshold.

A simpler version of CERE, called Linear CERE, is faster to compute and, as shown below, on the evolved agent analyzed here, outperforms CERE. It is essentially the same as CERE, except that f is ignored, or rather, treated as if it were the identity function. Eq. (4) is replaced by,

$$\Delta MSE_\pi = \left(\frac{1}{2^N} \frac{N_\pi}{n_\pi} \right) \sum_{\mathbf{m} \in T_\pi} ([\mathbf{m} \cdot \mathbf{c} - p_{\mathbf{m}}]^2 - [\mathbf{m} \cdot \mathbf{c} + \tilde{c}_\pi - p_{\mathbf{m}}]^2) . \quad (6)$$

Differentiating Equation 6, one can obtain closed form solutions for \tilde{c}_π and ΔMSE_π .

$$\tilde{c}_\pi = \frac{1}{n_\pi} \sum_{\mathbf{m} \in T_\pi} (p_{\mathbf{m}} - \mathbf{m} \cdot \mathbf{c}) \quad (7)$$

and, by replacing \tilde{c}_π in Equation 6,

$$\Delta MSE_\pi = \frac{N_\pi}{2^N} \tilde{c}_\pi^2. \quad (8)$$

[Figure 5 about here]

The accuracy of all four algorithms in the order-2 analysis of the agent is compared in Figure 5. Evidently, in this example all 2D incremental algorithms significantly outperform the basic 1-dimensional method, and come close to the performance of the full 2D algorithm at fairly moderate numbers of compound elements.

5 High-dimensional Neurocontroller Analysis

5.1 Compound Elements and Interactions

Section 3.4 described agent analysis using 1D-FCA. Here we perform a high order 2-dimensional analysis of the same agent and describe what can be learned about the system from the emerging structure of compound elements. Figure 6A depicts the contribution values of the simple and compound elements when eight order-2 compound elements are used to describe the agent’s performance². The figure demonstrates that the identity of the compound elements chosen by the Linear CERE and their corresponding contribution values are stable. The addition of the eight compound elements (out of the possible 45 pairs) considerably decreases the prediction error, from the 0.0153 reached by the 1-dimensional analysis to 0.0047. Even though more than eight compound elements were chosen over all 10 runs, only eight such elements were given non-negligible contribution values. These pairs are all combinations of significant simple elements, i.e., those with non-vanishing contribution values in the original 1-dimensional analysis (Figure 2). Analogously, Figure 6B depicts the contribution values of the simple and compound elements when eight compound elements of maximum order 3 are used, resulting in an MSE of 0.0037. Again, the selection of compound elements is very stable (there are 165 possible pairs and triplets), as are the values given to the chosen elements. The accompanying performance prediction

²After which the reduction in the prediction error levels off.

functions are depicted in panels C and D.

[Figure 6 about here]

What can be learned from these results? An important observation is that the contributions of the simple elements (Figure 2) change significantly when higher order elements are introduced. This is because the overall significance of an element is no more measured by its single contribution. Rather, compound elements that include this unit must be considered to reveal its overall contribution. As shown in section 5.4, using the contribution values of the compound elements to recompute the simple elements' contributions restores the results of the 1D-FCA. Thus, the higher order analysis singles out the interactions and modulations in the network faithfully, without losing the information about the original, basic units. In the 2D analysis, the command neuron (neuron 5) forms a significant compound element pair with every other significant neuron, two of which remain as 2D elements in the selected compound element set when 3D analysis is employed. Inferring the type of interaction between the elements of a pair from the corresponding contribution values is discussed in the next section. When 3D analysis is employed, all the 2D elements selected in the 2D analysis still appear, either as pairs, or as part of a triplet in which the effect of the pair is modulated by a third element.

[Figure 7 about here]

Figure 7A depicts the network whose nodes are the five significant neurons and whose edges are the eight major 2D interactions. For comparison, the network whose nodes are the same but whose edges are the eight (out of the 20 possible) synapses with the highest contributions (computed using a 1D-FCA on the synapses [2]) is depicted

in panel B. For additional comparison, the network with the eight largest synaptic weights (in absolute values) connecting these five neurons is depicted in panel C. The similarity between the two FCA-derived networks (A and B) is apparent. Both FCA analyses reveal a strong underlying edges’ backbone composed of the neuron pairs $\{1, 2\}$, $\{2, 5\}$ and $\{5, 3\}$, with several weaker edges (two of which appear in both). In both, the importance of the command neuron (neuron 5) is apparent. In contradistinction, the structure revealed by the synaptic weights’ network (C), although similar in some ways to the FCA-computed synaptic contributions, differs in a significant manner. The importance of the command neuron is less apparent, and the strong synapse between neurons 2 and 3 does not appear important from the FCA analyses. Moreover, although the synaptic weight from neuron 1 to 10 is large (C), testifying to the fact that the activity of neuron 1 strongly influences that of neuron 10, this synapse actually receives a small contribution value and no significant interaction between the two neurons is identified (A and B). Thus, neither the significance of synapses, nor the interactions between neurons can be inferred directly by looking at the strength of the synapses in the evolved network.

5.2 Types of 2D Interactions

The introduction of high-order compound elements does not only enrich the system’s description and allow for better prediction, but may also reveal the presence of specific types of high-order interactions. Here we focus on compound elements of order 2, i.e., pairs of simple units. First, note that not all interactions require compound elements for their description. A simple example is a pair of units which exhibit redundancy. Such interaction can be described with simple elements owing to the nonlinearity introduced by the performance prediction function, allowing e.g., for the assignment of similar performance values to configurations with different “lesion” ($\mathbf{m} \cdot \mathbf{c}$) levels.

However, other interactions, e.g., paradoxical lesioning, cannot be described in a 1-dimensional model. Let us now examine the types of interactions which may exist between any two simple units.

Consider a compound element consisting of two simple units. To understand the interaction between its two elements, we compare their contribution values c_1 and c_2 with that of the compound element, $c_{\{1,2\}}$. Recall that when both simple units are intact their overall contribution is $c_1 + c_2 + c_{\{1,2\}}$ (Eq. (1)). When only one simple unit is intact the contribution is c_1 (or c_2) only, and when both are lesioned their contribution to the argument sum of Eq. (1) vanishes. Since the prediction function is non-decreasing, a positive contribution to the sum of contributions in Eq. (1) indicates that this specific combination of units is advantageous (or at least neutral) for performance, and vice versa. Specifically, we look at the values c_1 and $c_1 + c_{\{1,2\}}$ (a symmetrical analysis can be made for c_2 and $c_2 + c_{\{1,2\}}$). Two interesting cases can then be identified:

1. $c_1 < 0$ but $c_1 + c_{\{1,2\}} > 0$. This is the classical case of Paradoxical Lesioning, where element 1 is advantageous only if element 2 is intact because $c_1 + c_2 + c_{\{1,2\}} > c_2$. However, if element 2 is lesioned element 1 becomes harmful, since its contribution alone (c_1) is negative.
2. $c_1 > 0$ but $c_1 + c_{\{1,2\}} < 0$. In contradistinction, in this Inverse Paradoxical Lesioning case, element 1 is advantageous only if element 2 is lesioned.

However, it may be the case that c_1 and $c_1 + c_{\{1,2\}}$ have the same sign, testifying to the fact that element 1 is beneficial (or harmful) irrespective of whether element 2 is intact or lesioned. In such a case, the state of element 2, although not switching the sign of the effect of element 1, still modulates the extent of its beneficial (harmful) effect in a way that cannot be covered by the non-linear prediction function and requires the

inclusion of the non-zero element $c_{\{1,2\}}$. This is more likely to occur when the system is large and the prediction function must cope with many such interactions.

In the evolved agent neurocontroller we find that all but two pairs have either paradoxical or inverse paradoxical effects (the two pairs that do not show these effects, $\{1, 3\}$ and $\{1, 5\}$, have a relatively low compound element contribution). For example, neurons 2 and 5 have a paradoxical lesioning interaction (Figure 6A), namely although lesioning neuron 5 alone damages performance, lesioning it is beneficial if neuron 2 is lesioned (case 1 above). This means that under normal circumstances the command neuron (5) is beneficial, but if the left turn motor neuron (2) is damaged (i.e., its feedback to the network is scrambled), this command neuron is actually harmful. *Importantly, since the two neurons interact with other units in other compound elements, the nature of the interaction cannot be seen directly from the agent’s corresponding lesioned performance levels.* Only by modeling the system and finding the contribution values is it possible to understand the nature of such 2D interactions. An analogous analysis can in principle be carried out for 3D compound elements. However, the space of possible interactions becomes much larger.

5.3 Generalization

As demonstrated in section 3.5, the 1D-FCA generalizes well when trained on a small sample of configurations. In this section we demonstrate the generalization capabilities of the High-D FCA. We train the High-D FCA under the same conditions as presented in Figure 6A, except that the training set is a randomly chosen set of 200 configurations. Like the 1D-FCA, the High-D FCA also generalizes well to the test set consisting of the full configuration set. This is evident from the low test MSE, 0.0074, which is 1.5 times the MSE achieved when *trained* on the full configuration set (this should be compared with the ratio of MSEs in the 1D-FCA, which is 1.3). Figure 8

depicts the resulting compound elements that are selected and their corresponding contribution values (as before, when computing the mean and standard deviation, if a compound element was not chosen in a specific run, it’s contribution is taken to be 0). The resulting contributions are compared with those obtained by using the full set, as appear in Figure 6A. As can be seen, the pairs selected as important are consistent, as well as the contribution values assigned to them.

[Figure 8 about here]

The above results were obtained by randomly selecting the training set. They can be further improved upon by an adaptive algorithm similar to the one presented in section 3.5, which smartly selects lesion configurations to more accurately generalize from a small training sample.

5.4 Corrected Contributions

The introduction of high-order elements to the analysis blurs the meaning of the contribution values of the simple elements, as discussed above. To infer the contribution values of the simple elements from high-dimensional FCA, the contribution values of the high-dimensional elements should be “credited” back to the simple elements. The *corrected* contribution value c'_i of the simple element i is defined as

$$c'_i = \sum_{\pi, i \in \pi} c_\pi \cdot 2^{1-k}, \tag{9}$$

where $i \in \pi$ denotes that the simple element i is included in the element π , and k denotes the order (number of elements) of π . The weighting by 2^{1-k} reflects the observation that a compound element of order k is matched by 2^{1-k} of the configurations that match its simple elements. Clearly, $c'_i = c_i$ when no compound elements

are used. Note that in general $\sum_{i=1}^N |c'_i| \neq 1$, so \mathbf{c}' is re-normalized.

[Figure 9 about here]

Figure 9 compares the corrected contribution values of different high-order analyses. The figure compares 1D-FCA, 2D-FCA using all order-2 compound elements, and sets of 100 high-order compound elements selected by the Linear CERE algorithm with maximum orders 3 and 4. The similarity of all four contribution vectors is evident, testifying to the stability of the FCA: The contribution values of the simple elements can be inferred from higher-order analysis, with very similar results. Thus, as desired, high order FCA retains the contribution values of the simple elements, while singling out important interactions between those elements.

6 Discussion

This paper extends the basic 1D-FCA approach to study high-dimensional FCA in EAAs. As shown, high-dimensional FCA becomes necessary if one strives to obtain a full and accurate description of function localization in evolutionary neurocontrollers. Efficient algorithms for the selection of the relevant subset of compound elements are presented. High-D FCA describes the system in terms of relevant functional groups – the compound elements. Each compound element is a set of elements which modulate each other’s contribution value. Moreover, when the set of selected compound elements achieves low MSE, the newly defined elements are such that now each element has a constant contribution value which is independent of the context, i.e., of the state of the rest of the system. Hence, the High-D FCA finds a new functional description of the system, which is not simply the given single elements. In this functional description the contributions of the elements are constant.

The application of high-dimensional FCA to the analysis of evolutionary neurocontrollers leads to several novel results: 1. A rigorous, quantitative description of localization of function in the neurocontroller, in terms of the contributions of simple and compound elements composing it. 2. Accurate prediction of the effects of possible lesions on the agent’s functioning (i.e., performance). 3. Insights concerning the main subsets of simple elements in the network that interact to modulate each other’s contribution to the system, and the types of interactions that predominate the network’s processing.

The neuroanalysis performed in this paper was primarily focused on a 2D description. However it is not by any means limited to two dimensions. High-D FCA also has the potential to estimate the inherent dimension of function localization in the neurocontroller. The dimensionality can be measured by the dimension needed to accurately describe the system, i.e., by identifying the lowest dimension after which the prediction MSE is no longer significantly decreased when the dimension of the analysis is further increased. This touches upon the delicate issue of the functioning of the network as a “whole”, and upon the “complexity” of its processing. Essentially, a lesion-based approach (like that used traditionally in neuroscience, or in FCA) implicitly assumes that the network’s operation can be decomposed, i.e., viewed and understood, on its parts/units level. But if the network’s operation is irreducible, i.e., it cannot be decomposed on the level of its building blocks units, can such a lesioning approach still make sense? High-dimensional FCA addresses this question by deriving the specific conjunction sets of elementary units that are needed to localize the network functioning, and by providing an upper bound characterizing its dimensionality. From this perspective, high-dimensional FCA searches for an intermediate level decomposition of the network which best reflects the localization of function within it.

FCA also reveals another “dimensionality” that is of immediate, practical impor-

tance. By eliminating those neurons with vanishing contributions and considering only the significant ones, it provides information about the true size of the neurocontroller that is really needed to solve the task. Thus, attempting to find successful solutions to a given problem in the EAA framework, one may use a two-tier approach: first, run a few initial trials starting from relatively large networks, and employ an FCA analysis to assess the number of significant neurons in successful runs, and then employ additional runs starting with the reduced network size that is needed to solve the problem in hand. This reduction of the search space to lower dimensions is likely to increase the chances of finding successful solutions.

An important aspect of the FCA is that it enables one to freely define the elements studied, and the way a lesion effects each element (the “whom” and the “what”). Re. the “whom”, it is possible to study the contributions of synapses, neurons, clusters of neurons, as well as specific parameters of the neuronal activities (e.g. the decay constant of a neuron or the parameters of a learning rule). Re. the “what”, the researcher is free to choose the type of the lesioning, which can be of varying degrees [14]. This is made possible in the FCA because the configuration vector is defined as a general mask, which simply indicates which elements are lesioned. Currently an element can be either lesioned in a certain way or left intact. But this is no inherent limitation, and future studies will extend the configuration vector to a continuous one such that a continuum of lesion possibilities is considered.

The ability to work in an EAA environment is, to our minds, essential to the development of neuroanalysis algorithms like FCA and high-dimensional FCA. Simply, one needs access to the full information characterizing such embodied neurocontrollers in order to develop these methods (and see [18] for a detailed discussion). However, it should be noted that the usage of analysis algorithms like FCA may be extended beyond the animat scope of Alife to the analysis of animate neurocontrollers. One such immediate possible application is the analysis of reversible inactivation exper-

iments, combining reversible neural cooling deactivation with behavioral testing of animals [15]. Other possible applications include the analysis of transcranial magnetic stimulation studies which aim to induce multiple transient lesions and study their cognitive effects (see [24] for a review). Such applications should prove useful in obtaining insights to the organization of natural nervous systems; settling the long-lasting debate about local versus distributed computation in animate systems, and measuring the dimensionality of function localization in these networks. Perhaps not less important, even when these analysis attempts are applied to animate systems solely, they raise interesting new questions and give rise to new insights concerning the basic concepts which guide conventional thinking in neuroscience about animate neural processing.

Rigorous localization of functioning in neurocontrollers is an important step in advancing our understanding of the neurocontrollers that we evolve. In previous papers [19, 2] we have shown how the basic FCA may also provide information about the synaptic architecture underlying the network’s functioning, and about the localization of various subtasks that the agent performs in the network. Obviously, these are merely the first steps in our quest for “really understanding” the operation of emerging neurocontrollers. Other current tools, primarily ones borrowed from information theory, and newly developed tools, should be applied towards this end. This task is very difficult and challenging even in the relatively small and seemingly simple neurocontrollers that currently drive EAAs. We are just at the beginning.

Acknowledgments

We acknowledge the valuable contributions and suggestions made by Alon Keinan, Hanoch Gutfreund, Tuvik Beker and Matan Ninio, and the technical help provided by Oran Singer. This research has been supported in part by the FIRST grant of

the Science Foundation, by the Adams Super Center for Brain Studies in Tel Aviv University, and by the Israel Science Foundation founded by the Israel Academy of Sciences and Humanities. Ranit Aharonov is supported by the Horowitz foundation.

References

- [1] R. Aharonov, I. Meilijson, and E. Ruppin. Understanding the agent's brain: A quantitative approach. In *The Sixth European Conference in Artificial Life (ECAL-2001)*, Prague, pages 216–225, 2001.
- [2] R. Aharonov, L. Segev, I. Meilijson, and E. Ruppin. Localization of function via lesion analysis. *Neural Computation*, In press.
- [3] R. Aharonov-Barki, T. Beker, and E. Ruppin. Emergence of memory-driven command neurons in evolved artificial agents. *Neural Computation*, (13):691–716, 2001.
- [4] R.D. Beer, H.J Chiel, and J.C. Gallagher. Evolution and analysis of model CPGs for walking II. general principles and individual variability. *Journal of Computational Neuroscience*, 7:119–147, 1999.
- [5] H.J. Chiel, R.D. Beer, and J.C. Gallagher. Evolution and analysis of model CPGs for walking I. Dynamical modules. *Journal of Computational Neuroscience*, 7:99–118, 1999.
- [6] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. In G. Tesauero, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 705–712. The MIT Press, 1995.
- [7] A. Engelbrecht and I. Cloete. Incremental learning using sensitivity analysis. In *IEEE IJCNN*, Washington DC, 1999.

- [8] M. J. Farah. *Visual agnosia*. MIT Press, Cambridge, MA, 1990.
- [9] M. J. Farah. Is face recognition ‘special’? evidence from neuropsychology. *Behavioral Brain Research*, (76):181–189, 1996.
- [10] D. Floreano and F. Mondada. Evolution of homing navigation in a real mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, 26(3):396–407, 1996.
- [11] I. Harvey, P. Husbands, and D. Cliff. Seeing the light: Artificial evolution, real vision. In D. Cliff, P. Husbands, J. A. Meyer, and S. Wilson, editors, *From Animals to Animats 3, Proc. of 3rd Intl. Conf. on Simulation of Adaptive Behavior, SAB94*, pages 392–401. MIT Press/Bradford Books, 1994.
- [12] C. C. Hilgetag, S. G. Lomber, and B. R. Payne. Neural mechanisms of spatial attention in the cat. *Neurocomputing*, 38:1281–1287, 2000.
- [13] P. J. Huber. Projection pursuit. *The Annals of Statistics*, 13(2):435–475, June 1985.
- [14] A. Keinan, I. Meilijson, and E. Ruppin. Controlled analysis of neurocontrollers with informational lesioning. *Submitted to Phil. Trans. R. Soc. Lond. A*, 2002.
- [15] S. G. Lomber. The advantages and limitations of permanent or reversible deactivation techniques in the assesment of neural function. *J. of Neuroscience Methods*, 86:109–117, 1999.
- [16] S. G. Lomber and B. R. Payne. Task-specific reversal of visual hemineglect following bilateral reversible deactivation of posterior parietal cortex: A comparison with deactivation of the superior colliculus. *Visual Neuroscience*, 18(3):487–499, 2001.

- [17] P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Chapman and Hall, 2nd edition, 1989.
- [18] E. Ruppin. Evolutionary autonomous agents: A neuroscience perspective. *Nature Reviews Neuroscience*, (3):132–141, 2002.
- [19] L. Segev, R. Aharonov, I. Meilijson, and E. Ruppin. Localization of function in neurocontrollers. In *Proceedings of the International Conference on the Simulation of Adaptive Behavior (SAB2002)*, Edinburgh, 2002.
- [20] M. Sitton, M. Mozer, and M. J. Farah. Superadditive effects of multiple lesions in a connectionist architecture: Implications for the neuropsychology of Optic aphasia. *Psychological Review*, (107):709–734, 2000.
- [21] J. M. Sprague. Interaction of cortex and Superior Colliculus in mediation of visually guided behavior in the cat. *Science*, (153):1544–1547, 1966.
- [22] L. R. Squire. Memory and the Hippocampus: A synthesis of findings with rats, monkeys, and humans. *Psychological Review*, (99):195–231, 1992.
- [23] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. Technical Report TR-AI-01-290, Department of Computer Sciences, The University of Texas at Austin, 2001.
- [24] V. Walsh and A. Cowey. Transcranial magnetic stimulation and cognitive neuroscience. *Nature Reviews Neuroscience*, (1):73–79, 2000.

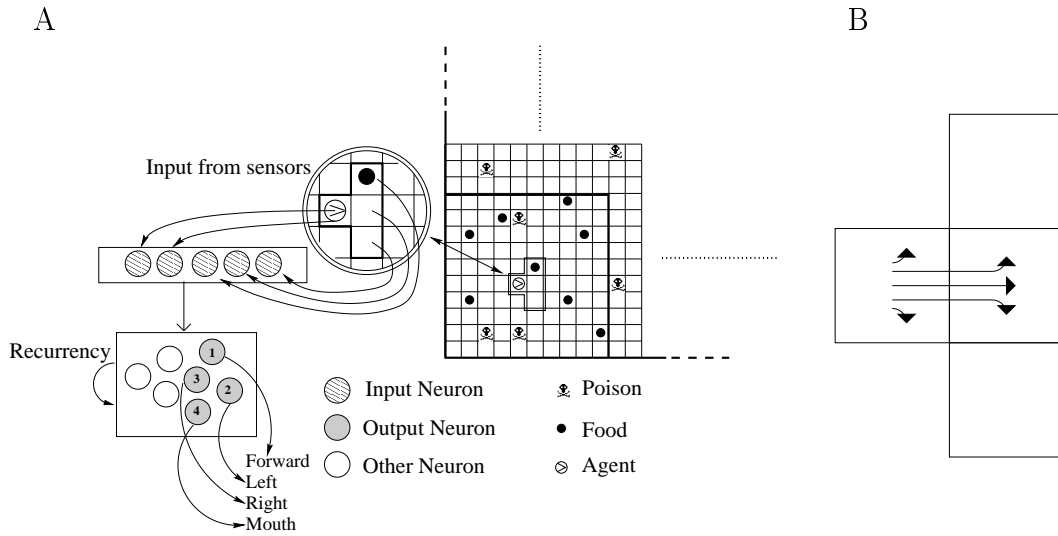


Figure 1: *The EAA environment*. A. An outline of the grid world and the agent's neurocontroller. B. The agent's movements. The agent can turn left, turn right, move forward, move forward and then turn left, or move forward and turn right. The combined forward plus turn is important, as it enables the agent to maintain "eye contact" with an item it has seen to its left or right. The arrowheads denote the orientation of the agent after moving. Here, the agent is initially in the left cell, facing right.

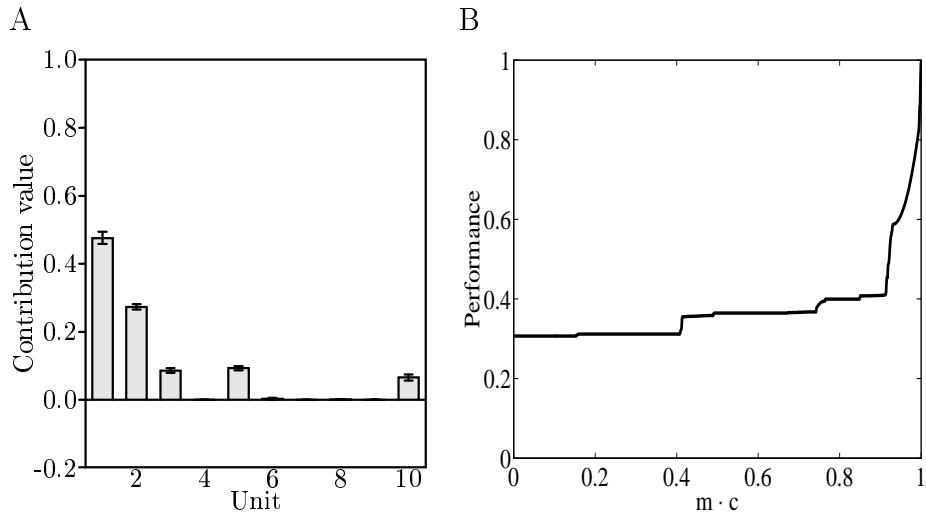


Figure 2: *Basic FCA analysis of the evolved agent.* A. Contribution values of the agent's neurons. Bars denote standard deviation over 10 FCA runs. B. The performance prediction function.

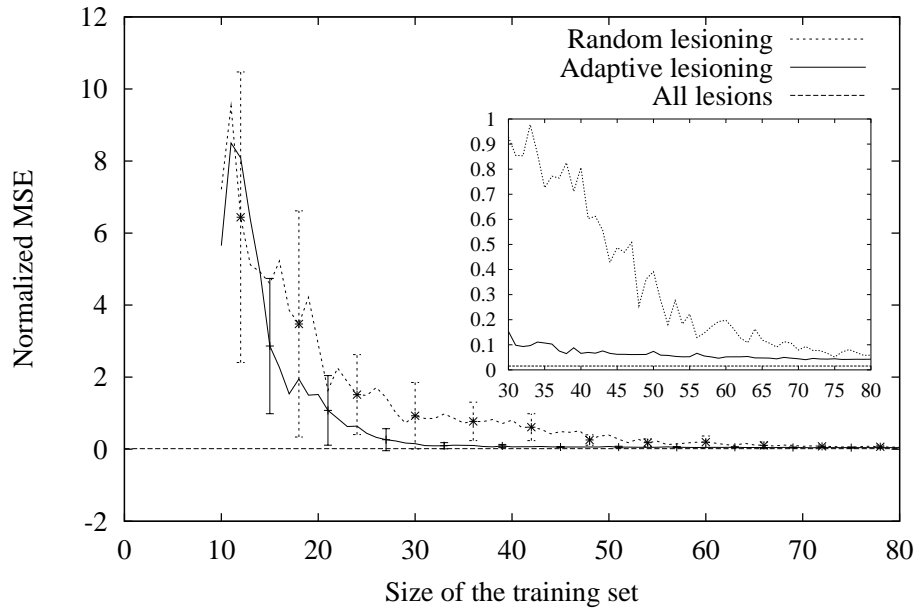


Figure 3: *Adaptive Lesioning vs. random configuration selection.* Mean and standard deviation of the normalized test MSE vs. number of training configurations of the agent. Test set is the full 2^{10} configuration set. The “All lesions” dashed line denotes the test error when training on the full configuration set. The inset focuses on a subset of the same data, portraying the number of training configurations leading to significant, low test MSE values.

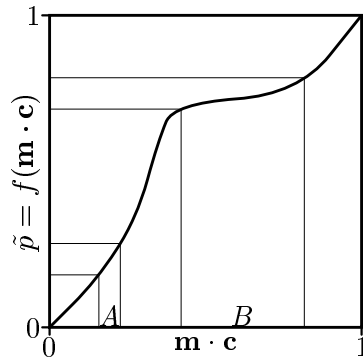


Figure 4: *Visualization of the Adaptive Lesioning algorithm.* The slope of f in domain A is much larger than its slope in domain B . As a result, a small perturbation in $\mathbf{m} \cdot \mathbf{c}$ where $\mathbf{m} \cdot \mathbf{c} \in A$ changes the performance estimation $\tilde{p} = f(\mathbf{m} \cdot \mathbf{c})$ more than if $\mathbf{m} \cdot \mathbf{c}$ were in B . The AL algorithm attempts to sample the p -axis uniformly, inducing the density of sampling along the $\mathbf{m} \cdot \mathbf{c}$ -axis to be proportional to the derivative of f .

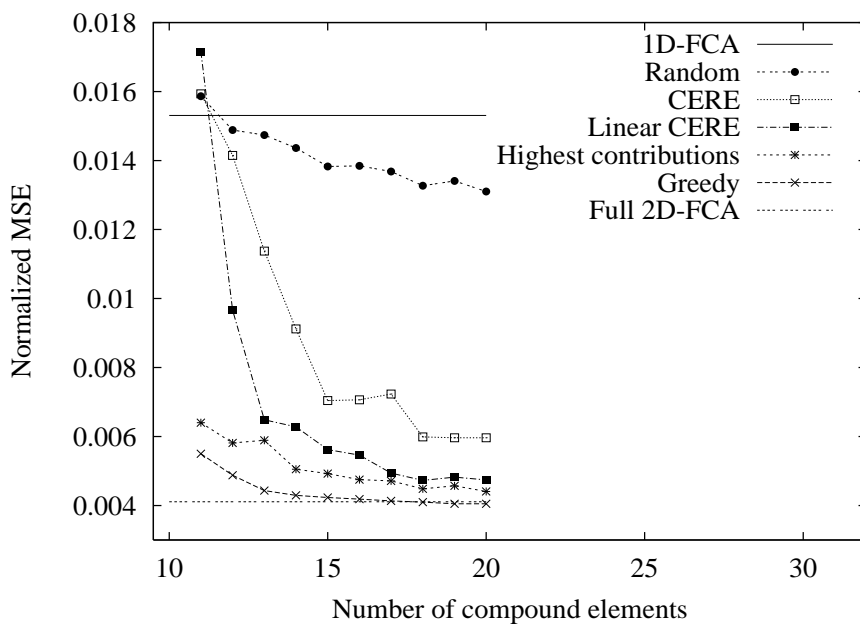


Figure 5: *Comparison of compound element selection algorithms.* Normalized test MSE vs. the number of order-1 and order-2 compound elements, for each of the algorithms (see text). “Random” denotes random selection of order-2 compound elements. The training and test set is the complete 2^{10} configuration set. 1D-FCA and full 2D-FCA (using all order-2 compound elements) are depicted as horizontal lines across the figure. All results are average of 10 runs.

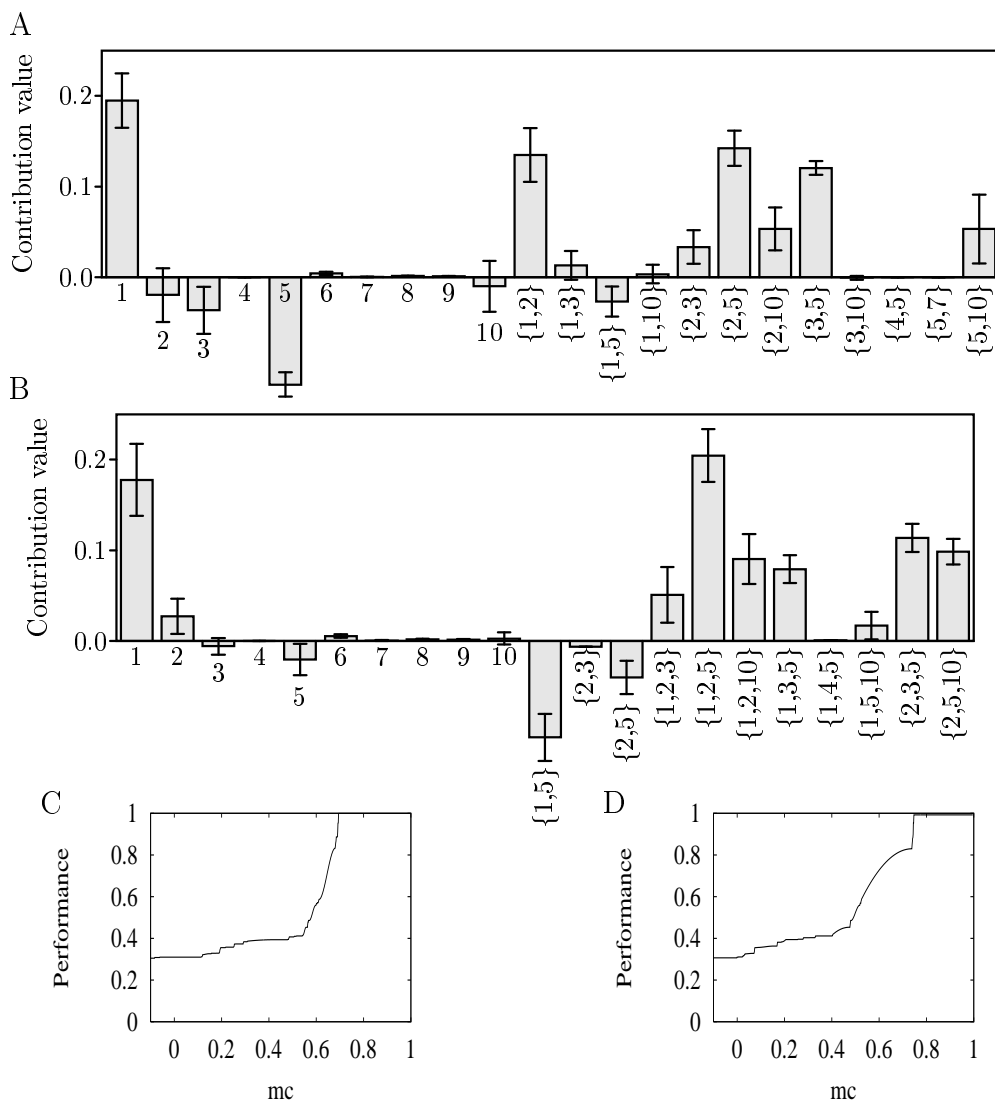
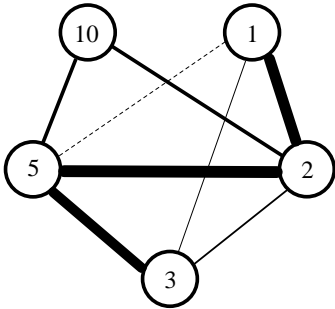
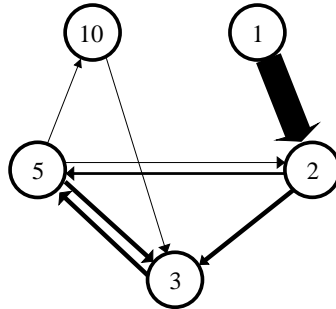


Figure 6: *2D and 3D FCA*. A, B. Mean and standard deviations of the contributions selected by 2D (A) and 3D (B) FCA. Results are from 10 runs of the linear CERE algorithm (limited to 8 compound elements). Both training and test sets are the complete 2^{10} possible configurations. Some elements were selected only in some of the runs. Hence, even though eight compound elements were selected, over all runs more than eight elements are chosen and depicted here (for calculating mean and standard deviation, the contribution values were taken as 0 when not chosen). C, D. Performance prediction function of the 2D-FCA (C) and 3D-FCA (D). All 10 runs result in similar functions. For clarity, the figure depicts one representative function.

A. 2D Contributions



B. Synaptic Contributions



C. Synaptic Weights

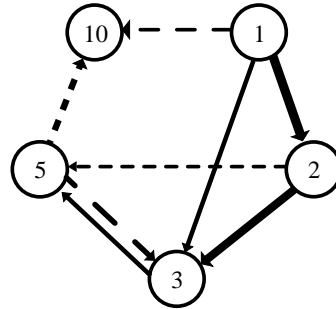


Figure 7: *Major interactions and synapses.* A. The eight largest 2D contributions between the five significant neurons (by absolute value) are depicted as connections. B. The eight synapses with highest contributions (by absolute value) between the five significant neurons. C. Similarly for the largest synaptic weights (by absolute value). Dashed lines denote negative values. The thickness of the lines scale with their absolute value, and are normalized to have the same sum in each panel.

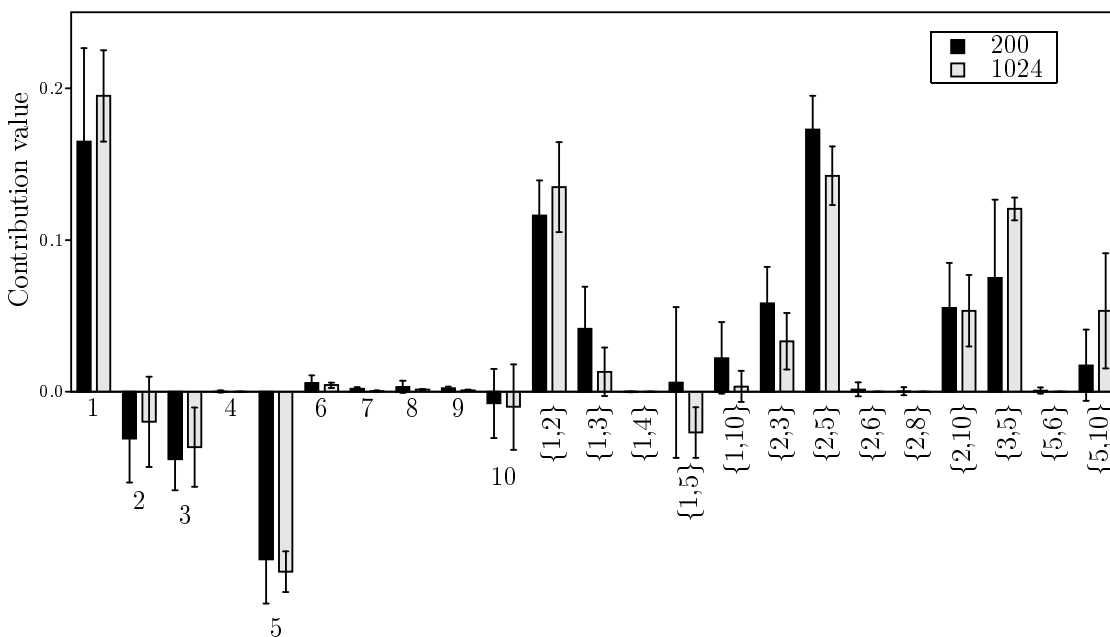


Figure 8: *Generalization in 2D-FCA*. Comparison of mean and standard deviations of the contributions selected by 2D-FCA. The training set is a randomly chosen set of 200 configurations (black bars) or the full 2^{10} configuration set (gray bars). Mean is over 10 runs of the linear CERE algorithm (limited to 8 compound elements). Some elements were selected only in some of the runs. Hence, even though eight compound elements were selected, over all runs more than eight elements are chosen and depicted here (for calculating the mean and standard deviation, the contribution values were taken as 0 when not chosen)

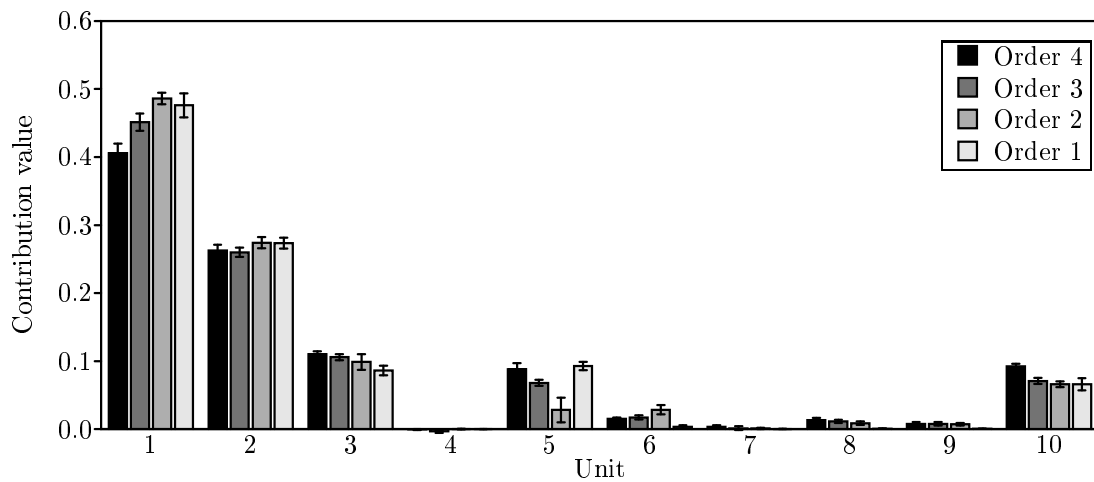


Figure 9: *Comparison of corrected contributions* for different maximum order of compound elements. For maximum order 1 and 2, all compound elements were used. For maximum order 3 and 4, 100 high-order compound elements were selected by the Linear CERE algorithm.