Axiomatic Scalable Neurocontroller Analysis via the Shapley Value

Alon Keinan
Ben Sandbank
School of Computer Science
Tel-Aviv University
Tel-Aviv, Israel
keinanak@post.tau.ac.il
sandban@post.tau.ac.il

Claus C. Hilgetag
School of Engineering and Science
International University Bremen
Bremen, Germany
c.hilgetag@iu-bremen.de

Isaac Meilijson
School of Mathematical Sciences
Tel-Aviv University
Tel-Aviv, Israel
isaco@post.tau.ac.il

Eytan Ruppin
School of Computer Science
Tel-Aviv University
Tel-Aviv, Israel
and
School of Medicine
Tel-Aviv University
Tel-Aviv, Israel
ruppin@post.tau.ac.il

Abstract One of the major challenges in the field of neurally driven evolved autonomous agents is deciphering the neural mechanisms underlying their behavior. Aiming at this goal, we have developed the multi-perturbation Shapley value analysis (MSA)—the first axiomatic and rigorous method for deducing causal function localization from multiple-perturbation data, substantially improving on earlier approaches. Based on fundamental concepts from game theory, the MSA provides a formal way of defining and quantifying the contributions of network elements, as well as the functional interactions between them. The previously presented versions of the MSA require full knowledge (or at least an approximation) of the network's performance under all possible multiple perturbations, limiting their applicability to systems with a small number of elements. This article focuses on presenting new scalable MSA variants, allowing for the analysis of large complex networks in an efficient manner, including large-scale neurocontrollers. The successful operation of the MSA along with the new variants is demonstrated in the analysis of several neurocontrollers solving a food foraging task, consisting of up to 100 neural elements.

Keywords
Neurocontroller analysis, localization of function, multiple perturbations, Shapley value, contributions, interactions

1 Introduction

Neurally driven evolved autonomous agents (EAAs) are computer programs embedded in a simulated virtual environment, typically performing tasks such as navigating, gathering food, evading predators, seeking prey, and pursuing mating partners. An EAA is controlled by an artificial neural network "brain." This neurocontroller receives and processes sensory inputs from the surrounding environment and governs the agent's behavior via the activation of motors controlling its actions. In

recent years, much progress has been made in developing methods to evolve EAAs that successfully cope with diverse tasks [8, 11, 25, 19, 20, 3, 33, 32, 21], employing neurocontrollers composed of several dozen neurons and hundreds of synapses (see [19, 34, 12] for reviews).

One of the major challenges in the field of EAAs is deciphering the mechanisms underlying agents' behavior by understanding the inner workings of the evolved neurocontrollers. The challenge in this field is greater than in other neural computation fields, since the neurocontrollers are evolved, rather than trained, and typically consist of recurrent synaptic connections. EAAs are a very promising model for studying neural processing [23]. Mainly, they are less biased than conventional neural networks used in neuroscience modeling, as their architecture is typically emergent, rather than pre-designed. Hence, the analysis of their inner workings, while also standing on its own, may serve as a test bed for analysis methods to be employed in biology.

In neuroscience, a fundamental challenge is to identify the individual roles of the network's elements, be they single neurons, neuronal assemblies, or cortical regions, depending on the scale at which the system is analyzed (single synapses, neurons, or groups of neurons, in the context of EAAs). Such localization of specific functions is conventionally done by recording neural activity during cognition and behavior, mainly using electrical recordings and functional neuroimaging techniques, and correlating the activations of the neural elements with different behavioral and functional observables (similar correlative approaches have also been employed in the context of EAAs, e.g., in [7]). However, such correlations do not necessarily identify causality. To allow the correct identification of the elements that are causally responsible for a given function, the deficit in performance is measured after perturbing specific elements, mainly by killing the elements, deactivating them, or disrupting their activity using transcranial magnetic stimulation (TMS). Most of the perturbation investigations in neuroscience have been *single-lesion* studies, in which only one element is disabled at a time (similar studies have also been performed in the context of EAAs, e.g., in [2]). Such studies are very limited in their ability to reveal the significance of interacting elements [1, 18], such as elements that exhibit functional overlap.

Acknowledging that single perturbations are insufficient for localizing functions in neural systems, Aharonov et al. [1] have presented the functional contribution analysis (FCA) method and applied it to the analysis of EAAs. The FCA analyzes a data set composed of numerous multiple perturbations that are inflicted upon a neurocontroller, along with the performance scores in a given set of functions. In each multi-perturbation experiment, several elements are perturbed, by concurrently disrupting their operation, and the agent's performance in each function is measured. The FCA is a machine learning algorithm that uses these data to yield a prediction of the agent's performance when new, untested multiple perturbations are imposed on it. It further yields a quantification of the elements' contributions to each function, as a set of values minimizing that prediction error. This definition of the contributions within the FCA is an operative one, based on minimizing the prediction error within a predefined prediction model. In particular, the FCA arrives at different error minima in different runs, yielding diverse contributions [18]. Moreover, since there is no inherent notion of correctness of the FCA contributions, they might converge to unreasonable values [18].

Addressing the same challenge of defining and calculating the contributions of system elements from a data set of multiple perturbations and their corresponding performance scores, while overcoming the shortcomings of the single-perturbation approaches and of the FCA, we have previously presented multi-perturbation Shapley value analysis (MSA) [18]. In this framework, we view a set of multi-perturbation experiments as a *coalitional game*, borrowing relevant concepts from the field of game theory. Specifically, the desired set of contributions is defined to be the *Shapley value* [27], which stands for the unique fair division of the game's worth (the system's performance score when all elements are intact) among the different players (the system's elements). Hence, in this framework, a *contribution* of an element to a function measures its importance, that is, the part it causally plays in the successful performance of that function.

In [18] we have utilized the MSA for the analysis of a neurophysiological model of the lamprey swimming controller [5], providing new insights to the workings of this classical model. Specifically,

the MSA uncovered the neuronal mechanism underlying the controller's oscillatory activity, as well as the redundancies inherent in it. Additionally, the contributions of specific synapses to different characteristics of the oscillation, such as frequency and amplitude, were inferred and analyzed. To test the applicability of the MSA to the analysis of real biological behavioral data, we have also applied it to reversible cooling deactivation experiments in cats [18]. The MSA correctly identified the contributions of different sites to the brain function of spatial attention to auditory stimuli, as well as a *paradoxical* type of interaction [31, 13]. This analysis was later extended to testing spatial attention to visual stimuli [16].

The MSA is not limited to the analysis of neural systems. Specifically, the MSA has been recently applied to the analysis of gene multi-knockout studies [14]. This analysis quantified the importance of genes in the Rad6 DNA repair pathway of yeast, providing a new functional description of the pathway. Incorporating the results with additional biological knowledge has given rise to new hypotheses regarding the functional roles of the genes involved.

The Shapley value as a game theoretical tool requires for its calculation the full knowledge of the behavior of the game at all possible coalitions (all multi-perturbation experiments). Thus, the number of computations needed to calculate the Shapley value grows exponentially with the number of elements in the analyzed system. This is reasonable for the applications just presented, as they involve networks of a limited size (at most nine elements). However, for larger systems containing many elements, such as EAAs' neurocontrollers analyzed on the level of single neurons or single synapses, these computations become infeasible. For this reason, this article presents methods to compute an approximation of the Shapley value with high accuracy and efficiency from a relatively small set of multi-perturbation experiments. These MSA variants are scalable in the number of elements, allowing us to perform multi-perturbation analysis of more complex systems, both artificial and biological, in a computationally tractable manner. The operation of the variants is tested and demonstrated in the analysis of several fully recurrent neurocontrollers.

The rest of the article is organized as follows: We start by presenting the evolutionary environment and the different agents analyzed throughout the article (Section 2). Section 3 introduces the relevant background from the field of game theory and revisits the basic MSA [18], while presenting applications for the analysis of neurocontrollers, both on neuronal and on synaptic levels. Section 4 presents the predicted Shapley value variant, and Section 5 the estimation variants, all with verifications of their accuracy. The predicted Shapley value is based on a predictor, trained on a small set of multi-perturbation experiments, to predict the outcome of other experiments. The estimated Shapley value is an unbiased estimator of the Shapley value, allowing us to approximate the contributions of elements while quantifying their accuracy. In Section 5 an application of the MSA for pruning the synapses of a neurocontroller is also presented and compared with pruning according to the FCA. In Section 6 a two-phase scalable MSA variant is introduced, which, after identifying the important elements of a system, focuses on accurately quantifying only their contributions. Section 7 deals with the scenario in which it is not desired or not possible to concomitantly perturb any number of the system's elements; it examines the neurocontroller analysis under varying perturbation levels. The MSA framework also quantifies the interactions between groups of elements [18], allowing for a higher-order description of the system. The theory of the interactions and its applicability to the analysis of neurocontrollers is detailed in Section 8. Our results and further applications of the MSA are discussed in Section 9. A MATLAB implementation of the MSA can be found at http://www.cns.tau.ac.il/msa.

2 The EAA Environment

The EAA environment used to study and develop the MSA is described in detail in [2]. A brief overview is provided here. The EAAs live in a virtual discrete 2D grid *world* surrounded by walls. Poison items are scattered all around the world, while food items are scattered only in a *food zone* located in one corner. An agent's goal is to find and eat as many food items as possible

during its life, while avoiding the poison items. The performance score of an agent is proportional to the number of food items minus the number of poison items it consumes. The agents are equipped with a set of sensors, motors, and a fully recurrent neurocontroller, which is evolved using a genetic algorithm [2].

Four sensors encode the presence of a wall, a resource (food or poison, without distinction between the two) or a vacancy in the cell the agent occupies and in the three cells directly in front of it (Figure 1). A fifth sensor is a "smell" sensor, which can differentiate between food and poison directly underneath the agent, but gives a random reading if the agent is in an empty cell. The four motor neurons dictate movement forward (neuron 1) or a turn left (neuron 2) or right (neuron 3), and control the state of the mouth (open or closed, neuron 4). In each step a sensory reading occurs, network activity is then synchronously updated, and a motor action is taken according to the resulting activity in the motor neurons.

Previous analysis [2] revealed that successful agents possess one or more *command neurons* that determine the agent's behavioral strategy. Artificially clamping these command neurons either to constant firing activity or to complete quiescence causes the agent to constantly maintain one of the two behavioral modes it exhibits, regardless of its sensory input. These two behavioral modes are *exploration* and *grazing*. Exploration, which normally takes place when the agent is outside the food zone, consists of moving in straight lines, ignoring resources in the sensory field that are not directly under the agent, and turning at walls. Grazing, which usually takes place when the agent is in the food zone, consists of turning towards resources to examine them, turning at walls, and maintaining the agent's location on the grid in a small region.

Throughout this article, we focus on the analysis of four agents, which have been successfully evolved in this environment. All four agents are equipped with the above sensors and motors, and their neurocontrollers are all fully recurrent, consisting of 10 internal neurons, including the motor neurons (not including the sensors). The differences between the agents follow:

1. The neurocontroller of S10 [2] is composed of binary McCulloch-Pitts neurons, whose synaptic strengths were evolved. This agent has been previously analyzed using the FCA [1, 26].

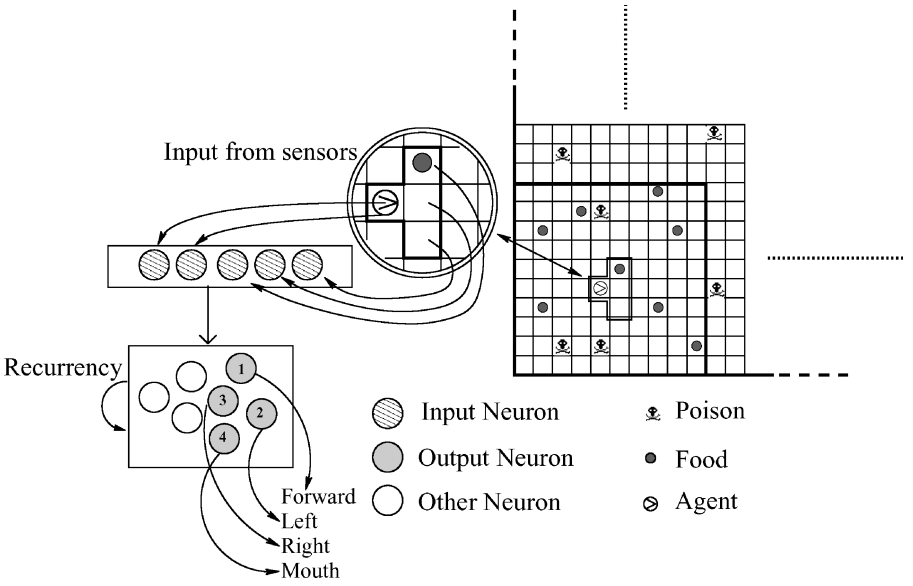

Figure 1. The EAA environment. An outline of the grid world and the agent's neurocontroller. The agent is marked by a small arrow on the grid, whose direction indicates its orientation. The curved lines indicate where in the arena each of the sensory inputs comes from.

2. P10 was obtained by a process in which, after the evolution of a successful agent, its synapses are pruned using an evolutionary network minimization algorithm [9] that deletes synapses and modifies the weights of the remaining ones so as to produce a similar agent with smaller neurocontroller. Like S10, P10 is equipped with a neurocontroller composed of binary McCulloch-Pitts neurons, but only 14 recurrent synapses out of the 100 original ones are left after applying the minimization algorithm.
3. In order to encourage the creation of a fault-tolerant neurocontroller, F10 was evolved using the same network architecture of S10, but while introducing phenotypic faults to the neurocontroller, resulting in a more robust agent [15].
4. Last, W10 copes with a more difficult version of the task, which also involves counting, as the agent has to wait and remain still in a grid cell containing food for five steps without moving or turning in order to eat [24]. Eating takes place in this version only if the agent closes its mouth in the last waiting step. The neurocontroller of W10 is composed of discrete-time integrate-and-fire neurons, whose membrane time constants were also evolved, in addition to the synaptic strengths.

3 The Shapley Value as a Contribution Measure

The starting point of the MSA is a data set of a series of *multi-perturbation* experiments studying a system's (neurocontroller's) performance in a certain function. In each such experiment, a different subset of the system's elements is perturbed concomitantly (constituting a *perturbation configuration*), and the system's performance in the studied function following the perturbation is measured. Given this data set, the main goal of the MSA is to ascribe to each element its contribution (importance) in carrying out the studied function.

The basic observation underlying the solution suggested by the MSA to meet this goal is that the multi-perturbation setup is essentially equivalent to a coalitional game. That is, the system elements can be viewed as players in a game. The set of all elements that are left intact in a perturbation configuration can be viewed as a coalition of players. The performance of the system following the perturbation can then be viewed as the worth of that coalition of players in the game. Within such a framework, an intuitive notion of a player's importance (or contribution) should capture the worth of coalitions containing it (i.e., the system's performance when the corresponding element is intact), relative to the worth of coalitions which do not (i.e., relative to the system's performance when this element, among others, is perturbed). This intuitive equivalence, presented formally below, enables us to harness the pertinent game theoretical tools to solve the problem of function localization in neurocontrollers.

We start by introducing some relevant background from the field of game theory: A *coalitional game* is defined as a pair (N, v), where $N = \{1, \ldots, n\}$ is the set of all *players* and $v(S)$, for every $S \subseteq N$, is a real number associating a worth with the *coalition* S, such that $v(\varnothing) = 0$.[1] In the context of multiple perturbations, N denotes the set of all the system's elements, and for each $S \subseteq N$, $v(S)$ denotes the performance measured under the perturbation configuration in which all the elements in S are intact and the rest are perturbed.

A *payoff profile* of a coalitional game is the assignment of a payoff to each of the players. A *value* is a function that assigns a unique payoff profile to a coalitional game. It is *efficient* if the sum of the components of the payoff profile assigned is $v(N)$. That is, an efficient value divides the overall game's worth (the system's performance when all elements are intact) among the players (the system elements). An efficient value that captures the importance of the different players may serve as a basis for quantifying, in the context of multiple perturbations, the contributions of the system's elements.

1 This type of game is most commonly referred to as a *coalitional game with transferable payoff*.

The definite value in game theory and economics for this type of coalitional game is the *Shapley value* [27], defined as follows: Let the *marginal importance* of player i to a coalition S, with $i \notin S$, be

$$\Delta_i(S) = v(S \cup \{i\}) - v(S). \tag{1}$$

Then the Shapley value is defined by the payoff

$$\gamma_i(N, v) = \frac{1}{n!} \sum_{R \in \mathcal{R}} \Delta_i(S_i(R)) \tag{2}$$

assigned to player i, for all $i \in N$, where $\mathcal{R}$ is the set of all $n!$ orderings of N, and $S_i(R)$ is the set of players preceding i in the ordering R. The Shapley value can be interpreted as follows: Suppose that all the players are arranged in some order, all orders being equally likely. Then $\gamma_i(N, v)$ is the expected marginal importance of player i to the set of players who precede him. The Shapley value is efficient, since the sum of the marginal importance of all players is $v(N)$ in any ordering. Alternative views of the Shapley value in the context of multiple perturbations, including its axiomatic foundation, have been presented in [18].

The Shapley value can also be expressed in terms of a sum over all possible coalitions instead of a sum over all possible orderings, namely

$$\gamma_i(N, v) = \frac{1}{n!} \sum_{S \subseteq N \setminus \{i\}} \Delta_i(S) \cdot |S|! \cdot (n - |S| - 1)!. \tag{3}$$

Substituting according to Eq. (1) results in

$$\begin{aligned} \gamma_i(N, v) = & \tfrac{1}{n!} \sum_{S \subseteq N, i \in S} v(S) \cdot (|S| - 1)! \cdot (n - |S|)! \\ & - \tfrac{1}{n!} \sum_{S \subseteq N, i \notin S} v(S) \cdot (|S|)! \cdot (n - |S| - 1)!, \end{aligned} \tag{4}$$

where each coalition S contributes a summand to one of the two sums, depending on whether player i belongs to it. Thus, the Shapley value calculation consists of going through all coalitions and calculating for each element the two sums in the above equation. This formulation highlights the fact that taking the average in Eq. (2) over all *orderings* is equivalent to taking a weighted average over all *coalitions*, where the weight of each coalition is determined by its size; the farther away the size is from $n/2$, the larger is the weight. Due to combinatorial considerations, there are considerably more coalitions of size close to $n/2$ than coalitions of size close to 1 or n. Taking an unweighted average of the marginal importance of a player over the set of all coalitions would have therefore resulted in a value primarily determined by mid-size coalitions. The choice of weights used in Eq. (4), which is derived from the Shapley value definition, entails that the overall part played by all coalitions of size k is independent of k.

In the fifty years since its construction, the Shapley value as a unique fair solution has been successfully used in many fields. Probably the most important application is in cost allocation, where the cost of providing a service is to be shared among the different receivers of that service. This application was first suggested in [29], and the theory was later developed by many authors (e.g., 22, 4). This use of the Shapley value has received recent attention in the context of sharing the cost of multicast routing [6]. In epidemiology, the Shapley value has been utilized as a mean to quantify the

population impact of exposure factors on a disease load [10]. Other fields where the Shapley value has been used include, among others, politics (starting from the strategic voting framework introduced in [28]), international environmental problems, and economic theory (see [30] for discussion and references).

The MSA, given a data set of multiple perturbations, uses the Shapley value as the fair division of the system's performance among the different elements, assigning to each element its contribution as its average importance to the function in question.[2] The higher an element's contribution according to the Shapley value, the larger is the part it plays in the successful performance of the function.[3] This set of contributions is a unique solution, in contrast to the multiplicity of possible contribution assignments that may plague an error minimization approach like the FCA [1].

Once a game is defined, its Shapley value is uniquely determined. However, different analyses may utilize different perturbation methods, with each method perturbing a different aspect of the element's function. Obviously, each perturbation method may result in different values of v and, as a consequence, different Shapley values. The perturbation methods employed throughout this article for the analysis of EAAs' neurocontrollers are *stochastic lesioning* [1] and *informational lesioning* [17]. Stochastic lesioning is performed by randomizing the activity of a perturbed element (a neuron or a synapse) while keeping its mean activity unchanged [1]. Informational lesioning employs more minute perturbations, enabling one to vary the level of perturbations and to capture the long-term contributions of elements [17]. In Section 9 we return to discuss the effects of different perturbation methods on the contributions found.

The rest of this section deals with the ideal situation in which the full set of all 2^n possible perturbation configurations are given, along with the performance measurement for each. Hence, the Shapley value may be straightforwardly calculated using Eq. (4), where the summation runs over all 2^n configurations of n elements. We apply this *full-information* analysis to agent P10 in order to determine the contributions of each of its 14 synapses to its performance (defined in Section 2). To this end, we measure the agent's performance score under the entire set of 2^{14} synaptic perturbation configurations, using stochastic lesioning, where each synaptic perturbation configuration indicates for each of the 14 recurrent synapses in the neurocontroller whether it is perturbed or left intact. Figure 2 plots the Shapley value, calculated in a straightforward manner using the full information. The four most important synapses are, in order of importance, the synapse from the right motor to the left one, the synapse from the forward motor to the command neuron (neuron 8), the recurrent synapse from the command neuron to itself, and the synapse from the command neuron to the right motor. The analysis uncovers the main mechanism underlying this minimized neurocontroller's operation, while quantifying the part played by each of the mechanism's constituents. The rest of the synapses exhibit minor contributions, with two synapses [(7,1) and (7,3)] having slightly negative contributions, testimony to the fact that, on the average, they slightly hinder the performance. For comparison with the previous (FCA) method, Figure 2 also presents the FCA contributions yielded when training the FCA 10 times with the training set consisting of all 2^{14} configurations. Only for 2 out of the 14 synapses is the FCA contribution within 1 standard deviation of the corresponding contribution defined by the Shapley value, which captures the fair axiomatic division of the contributions. Moreover, the FCA assigns a near-vanishing contribution to the self-synapse of the command neuron, which has been shown to facilitate the short-term memory of the agent and to be essential to its neural underpinnings [2].

For W10, the counting agent, the performance score under the entire set of 2^{10} neuronal perturbation configurations was measured using stochastic lesioning, and the neurons' contributions were calculated (Figure 3). Previous analysis [24] revealed neuron 10 to be the command neuron, neurons 4 and 9 to participate in the temporal counting required for the precise timing of food

2 Since $v(\emptyset) = 0$ does not necessarily hold in practice (because it depends on the performance measure definition and the perturbation method), the sum of the contributions assigned to all the elements equals $v(N) - v(\emptyset)$.

3 Since no limitations are enforced on the shape of v, a negative contribution is possible, indicating that the element hinders the performance of the function studied on the average.

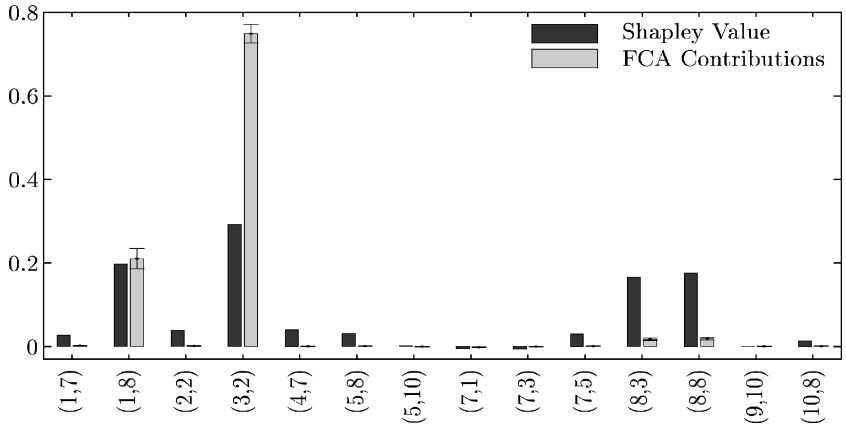

Figure 2. Shapley value and FCA normalized contributions of the synapses of P10. The Shapley values (black bars) of the synapses of P10 are plotted along with FCA contributions (gray bars; mean and standard deviation across 10 FCA runs). The *x* axis presents the synapses in the form (*presynaptic neuron*, *postsynaptic neuron*). Both sets of contributions are normalized, as in all the results to follow (except where stated otherwise), so that the sum over all synapses equals 1.

consumption, and neuron 1, the forward motor, to count the last two steps before moving forward. The MSA accurately reveals those neurons to be the most significant ones.

4 Predicted Shapley Value

Obviously, measuring the performance levels of all perturbation configurations required for the calculation of the Shapley value is often intractable. In such cases, one may measure the system's performance under only a partial subset of the possible configurations, and use the results to train a predictor that assesses the performance levels of unseen configurations. Given a predictor, the predicted outcomes of all multi-perturbation experiments may be extracted, and a predicted Shapley value [18] can be calculated as the Shapley value based on these predictions. Unlike the FCA, where the predictor component and the calculation of the contributions are intertwined, the MSA enjoys an uncoupling between the two and hence may utilize any predictor relevant for the data, without any

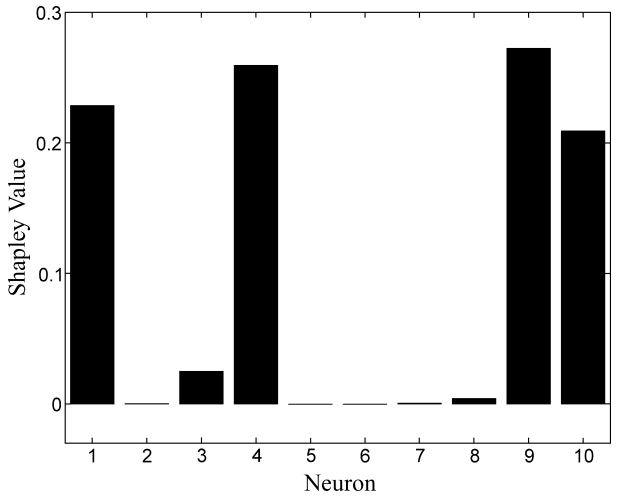

Figure 3. Shapley values of the neurons of W10.

constraints. Such a predictor can be based on standard statistical and machine learning methods. In the results to follow, the FCA itself is used as the predictor within the MSA framework. It is merely used as a black box for predicting the performance of perturbation configurations, without considering the contribution values it yields. Other predictors have also been used within the framework of the MSA [18, 14].

To test the accuracy of the predicted Shapley value, predictors were trained with training sets of randomly chosen synaptic perturbation configurations of P10 of sizes 100, 200, . . . , 1000 (out of 2^{14} = 16,384 configurations). Figure 4 plots the predicted Shapley value contribution of the most important synapse of P10, against the number of configurations in the training set, along with the real Shapley value. The predicted Shapley value is very close to the real one, even for very small numbers of perturbation configurations used for training, and exhibits stability across the different runs, as shown by the small standard deviations. Remarkably, this is true even though the prediction is not very accurate (the average test MSE corresponds to explaining less than 60% of the variance when 100 perturbation configurations are used for training; the prediction improves as the training set size increases). This might be explained by the fact that the Shapley value is obtained via an averaging of a large number of predictions. Assuming that the prediction is unbiased, prediction errors cancel each other out, resulting in a predicted Shapley value that is very similar to the real one. As further evident from Figure 4, the contribution yielded by the FCA approach differs significantly from the Shapley value and exhibits a large standard deviation across different runs, much larger than that of the predicted Shapley value contribution, even though both are based on the same data, the FCA's predictions.

5 Estimated Shapley Value and Estimated Predicted Shapley Value

The predicted Shapley value (Section 4) relieves the MSA of the need for the full set of 2^n perturbation configurations. Nevertheless, $n \cdot 2^n$ computations are still required for its calculation (summing over all predicted configurations for each element). When the number of elements is too large for such a

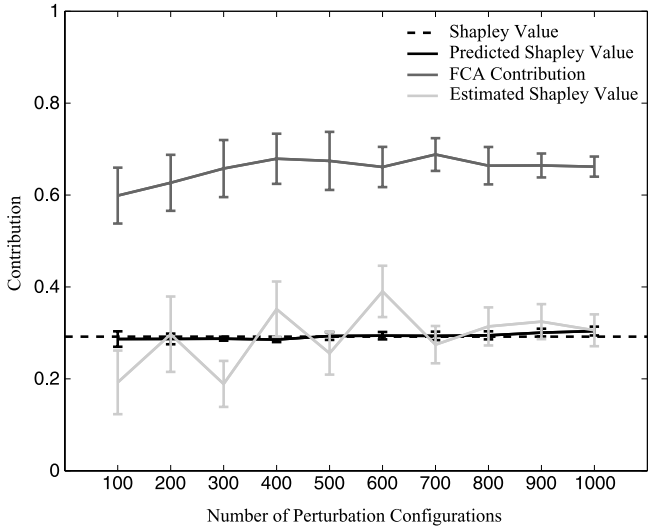

Figure 4. Predicted Shapley value, FCA contribution, and estimated Shapley value using small perturbation sets. These values for the most important synapse of P10, (3, 2), are plotted against the number of perturbation configurations (100, 200, . . . , 1000). The black line plots the mean and standard deviation of the predicted Shapley value, based on 10 different FCA predictions. The dark gray line plots the mean and standard deviation of the FCA contribution across the 10 runs. The light gray line plots the estimated Shapley value and the standard deviation estimates. The dashed black line plots the real Shapley value, calculated using the full set of 2^{14} configurations.

method to be tractable, sampling may be used, facilitating the MSA's scalability. Rather than sampling single configurations, the estimated MSA variant samples whole permutations of the n elements. Let $\hat{\mathcal{R}}$ be a randomly sampled set of permutations (with replacement). Then, based on Eq. (2),

$$\hat{\gamma}_i(N, v) = \frac{1}{|\hat{\mathcal{R}}|} \sum_{R \in \hat{\mathcal{R}}} \Delta_i(S_i(R)) \tag{5}$$

is an unbiased estimator of the Shapley value $\gamma_i(N, v)$. In order to calculate these estimates for every i, for each permutation $a_1, \ldots, a_n$ in $\hat{\mathcal{R}}$, the performance measurements $v(\varnothing)$, $v(\{a_1\})$, $v(\{a_1, a_2\})$, $\ldots$, $v(\{a_1, a_2, \ldots, a_{n-1}\})$, $v(N)$ are needed. Notice, however, that a perturbation configuration should be calculated only once, though it may appear in different permutations. Thus, the number of new multi-perturbation experiments to be performed for each sampled permutation tends to decrease as more permutations are sampled. The resulting *estimated Shapley value* is an efficient value, since the sum of the marginal importance of all elements in any permutation is $v(N) - v(\varnothing)$.

The empirical standard deviation of the marginal importance of element i,

$$s_i(N, v) = \sqrt{\frac{1}{|\hat{\mathcal{R}}|} \sum_{R \in \hat{\mathcal{R}}} [\Delta_i(S_i(R)) - \hat{\gamma}_i(N, v)]^2}, \tag{6}$$

yields an estimator of the standard deviation of the Shapley value estimator $\hat{\gamma}_i(N, v)$:

$$\hat{\sigma}(\hat{\gamma}_i(N, v)) = \frac{s_i(N, v)}{\sqrt{|\hat{\mathcal{R}}|}}. \tag{7}$$

The standard deviation measures how close the estimated Shapley value is to the true value. Specifically, using the Shapley value estimator and the standard deviation estimator, confidence intervals for the contribution of each of the elements can be constructed. It is further possible to test statistical hypotheses on whether the contribution of a certain element equals a given value (e.g., zero or 1/n). Both the confidence intervals and the hypothesis tests are based on the t distribution. Sampling permutations for constructing the set $\hat{\mathcal{R}}$ can be done with a given sample size or sequentially—for instance, stopping on reaching a fixed maximal limit for the number of performance calculations (the number of perturbation configurations) or when all standard deviation estimates are small enough.

The multi-perturbation experiments that should be performed in the estimated Shapley value method are dictated by the sampled permutations. At times, however, one is given an existing data set of performance measures for some set of perturbation configurations, which does not necessarily match a random permutation sample. In this case, the MSA offers an additional estimation variant: A performance predictor is trained using the given set of perturbation configurations and serves as an oracle supplying performance predictions for any perturbation configuration as dictated by the sampled permutations, resulting in an *estimated predicted Shapley value*.

Figure 4 plots the estimated Shapley value, along with its standard deviation estimate, for the most important synapse of P10, against the number of perturbation configurations used.[4] As expected from the theory, the estimated Shapley value appears to be an unbiased estimator for the real Shapley value,

4 Since whole permutations are sampled, the actual number of configurations used for a defined size s is between s and $s + n - 2$, where $n = 14$ in this case.

and its standard deviation generally decreases with increasing sample size. Notably, the standard deviation of the estimated Shapley value is much larger than that of the predicted Shapley value.

As a demonstration of a case where the entire set of predictions is computationally intractable, we turn to analyze the full recurrent synaptic neurocontroller of S10, consisting of 100 synapses (and hence 2^{100} possible configurations). An estimated Shapley value is calculated based on a sample of 100 permutations (dictating 9,833 perturbation configurations), where informational lesioning is employed in order to capture the long-term contributions (lesioning level of 0.5 [17]). Training a predictor with the same sample, the estimated predicted Shapley value is computed by sampling configurations from the predictor using sequential sampling, stopping when the standard deviation estimates of all 100 estimated predicted contributions are below 0.005. Arbitrarily defining an *important* synapse as one with a normalized contribution above 0.03 (3% of the total performance of the neurocontroller), the same nine synapses are yielded as important by both the estimated Shapley value and the estimated predicted one (Figure 5), with very similar contributions. These conclusions are rather insensitive to the choice of threshold used for defining an important synapse. By finding the important synapses, the MSA reveals the recurrent backbone of the neurocontroller, containing, in this case, only 9 out of the 100 synapses. Focusing on the backbone may simplify the further analysis of such fully recurrent networks [1, 18].

The MSA may be useful for pruning a neurocontroller by removing the synapses according to the magnitude of their contributions in ascending order. In [1] it was shown that pruning by the FCA contributions outperforms pruning by synaptic weight magnitude. To compare the validity of the contributions obtained by the MSA with those obtained by the FCA, we incrementally pruned the full recurrent synaptic neurocontroller of S10 using the two methods. Figure 6 depicts the performance of the agent as a function of the number of pruned synapses, starting from the intact neurocontroller. It compares pruning according to the above-estimated Shapley value with pruning according to the FCA contributions, when the FCA is trained using the same random sample. Evidently, the degradation in the performance using MSA-based pruning is slower than with the FCA, testifying that the MSA better captures the inherent importance of the synapses.

6 A Two-Phase Procedure for Large-Scale Analysis

Within the MSA framework, we suggest another scalable method for handling systems with a large number of elements. This *two-phase MSA procedure* is motivated by the observation that often only a

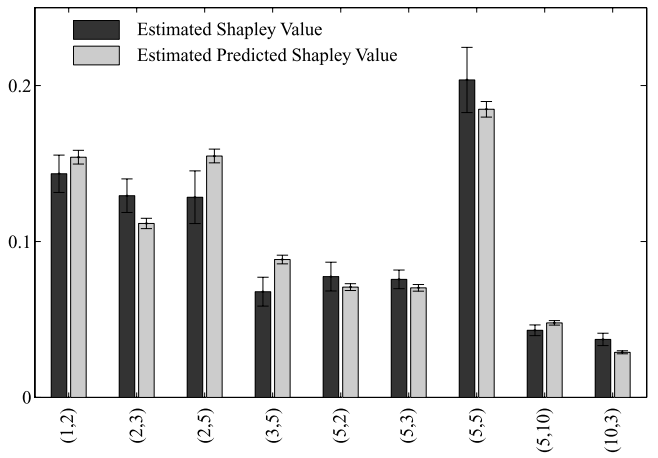

Figure 5. Estimated Shapley value and estimated predicted Shapley value of the important synapses of S10. Error bars of both denote the standard deviation estimates.

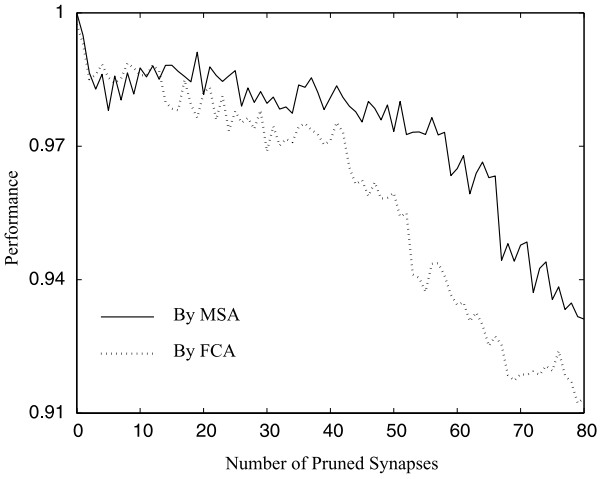

Figure 6. Agent S10's normalized performance as a function of pruning level, by MSA and by FCA. In both methods the synapses are incrementally pruned in ascending order of their contribution. The figure focuses on the first 80 synapses pruned, where the agent still has viable performance, after which its performance drastically decreases with both methods of pruning.

small fraction of the possibly large number of elements significantly contribute to the specific function tested within the analysis. The first phase hence finds those elements with significant contributions. This phase uses a small sample in order to calculate the estimated Shapley value and the standard deviation estimates (Section 5). A two-sided t test is then performed on the contribution of each element, where the null hypothesis indicates that the contribution is zero, thus identifying the significant elements.

The second phase focuses on finding the accurate contributions of the significant elements. This phase may use the same small sample from the first phase, but it focuses on the coalitional game $(N', v^{N'})$, where N' is the set of elements found as significant in the first phase. Given the characteristic function v of the original game consisting of all elements, $v^{N'}$ may be defined such that for each $S \subseteq N'$, $v^{N'}(S)$ equals the average of $v(T)$ over all $T \subseteq N$ satisfying $T \cap N' = S$. Thus, using the original sample, some of the $v^{N'}$ are based on an average over many perturbation configurations, while others might not be evaluated due to lack of data. In the case where the characteristic function $v^{N'}$ cannot be fully calculated, a predictor is trained using the available data (Section 4). The predictor is trained on perturbation configurations consisting of $|N'|$, rather than $|N|$, elements, facilitating faster training and increased scalability. In the case where the number of significant elements is too large for the explicit calculation of a predicted Shapley value, sampling is incorporated (Section 5) also in this second phase, using the much smaller configuration space.

We begin by applying the two-phase MSA procedure to the analysis of a moderate-size system in order to allow for the comparison of the results of the procedure with the real Shapley value. Such a moderate-size system is obtained by focusing on part of S10's synaptic network, analyzed in Section 5, consisting of 14 synapses out of the 100, after pruning the rest of the synapses. These 14 synapses are the ones previously found most important by the FCA [17]. First, we perform a full-information MSA to identify the true contributions. Then, we perform a two-phase analysis. In the first phase of the latter, a small random sample of 20 permutations (dictating 227 perturbation configurations out of the 2^{14}) is used to estimate the Shapley value and the standard deviations. Performing t-tests (two-sided, $\alpha = 0.05$) using the estimates results in the identification of 11 significant synapses (out of the 14 found as most important by the FCA). The second phase focuses on those significant synapses, based on the same sample of 227 configurations. A predictor is trained on the sample,[5] and a *two-phase predicted*

5 The insignificant synapses are ignored in the perturbation configurations, and the performance scores of identical configurations are averaged, resulting in 164 configurations out of the 2^{11} possible ones.

Shapley value is calculated using the predictions for the full set of 2^{11} configurations. Figure 7 presents the results of this analysis. First, the three synapses with near-vanishing contributions according to the real Shapley value are the ones found insignificant in the first phase of the procedure. Second, the final two-phase predicted Shapley value is much closer to the true contributions than the estimated Shapley value calculated in the first phase, and with much smaller standard deviations.

To examine the two-phase MSA procedure on a larger scale (thus losing the ability to calculate and compare with the true contributions), we turn to analyze the full recurrent synaptic network of S10, consisting of all 100 synapses. The first phase, using a very small random sample consisting of 10 permutations, identifies 20 synapses as significant (two-sided t-tests, $\alpha = 0.05$). A predictor is trained on the set induced by this sample, and a two-phase predicted Shapley value is calculated from the predictions for the full set of 2^{20} synaptic perturbation configurations. The mean normalized training MSE corresponds to explaining more than 99.8% of the variance, which is five times more accurate than when training on the original sample consisting of configurations of all 100 synapses. Figure 8 displays the two-phase predicted Shapley value for the 20 significant synapses, illustrating small standard deviations of the contributions, testifying to their consistency. In this two-phase procedure, the nine synapses with largest contributions are the same ones found using the single-phase MSA methods (Section 5).

7 Bounded Perturbation Level Analysis

As shown in Section 3, the overall part played by all coalitions of size k in determining an element's contribution is identical for every k. While this is an important aspect of the definition of the Shapley value, within the context of perturbation experiments a different approach may be desired. Specifically, configurations in which most of the system's elements are perturbed may exhibit totally different behavior from that of the original system. Thus, the marginal importance of an element to such a configuration may have no bearing on its true contribution to the system. For this reason, it may be prudent to limit the analysis to concomitantly perturbing only a small number of elements, and to define an element's contribution based on these experiments solely.

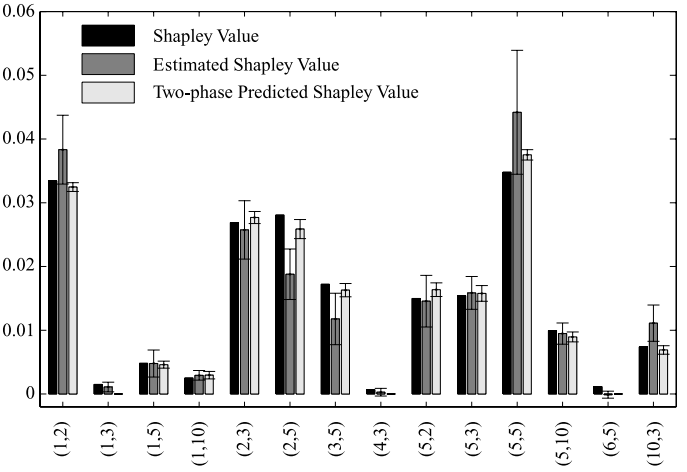

Figure 7. Shapley value, estimated Shapley value (with standard deviation estimates), and two-phase predicted Shapley value (mean and standard deviation across 10 predictors in the second phase) for the 14-synapse network (see main text). Synapses found insignificant in the first phase [(1,3), (4,3), and (6,5)] are assigned a two-phase predicted contribution of zero. In order for the different values to be comparable, they are not normalized, but rather the sum of the Shapley value and the sum of the estimated one equal $v(N) - v(\varnothing)$, where N is the group of all 14 synapses, and the sum of the two-phase predicted ones equals $v^{N'}(N') - v^{N'}(\varnothing)$, where N' is the group of the 11 significant synapses.

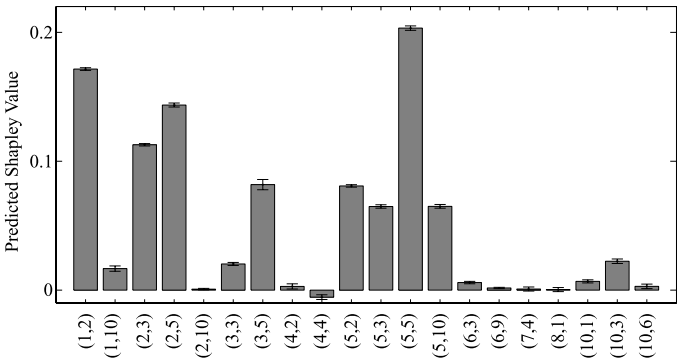

Figure 8. Two-phase predicted Shapley value of the 20 synapses of S10 found significant in the first phase (mean and standard deviation across 10 predictors).

A more mundane reason for limiting the number of concomitantly perturbed elements may be the practical impossibility of perturbing more than a few at a time, as is the case in most types of biological experiments.

In such cases, suppose that, for some perturbation level k, one is given all perturbation configurations in which no more than k elements are perturbed. Let the marginal importance of element i to a coalition S, with $i \notin S$, be

$$\Delta_i^k(S) = \begin{cases} v(S \cup \{i\}) - v(S), & |S| \geq n - k, \\ 0, & |S| < n - k. \end{cases} \tag{8}$$

Then, the contribution of element i, in the spirit of the Shapley value from Eq. (2), can be defined as

$$\gamma_i^k(N, v) = \frac{1}{k(n-1)!} \sum_{R \in \mathcal{R}} \Delta_i^k(S_i(R)). \tag{9}$$

These *MSA k-bounded contributions* coincide with the Shapley value for $k = n$ and with single-perturbation analysis for $k = 1$. Similarly to the full ($k = n$) case, if the calculation of all perturbation configurations with up to k perturbed elements is intractable, a predictor can be trained to predict the performance levels of all those configurations, and *predicted k-bounded contributions* can be calculated. Further, when the set of those configurations is too large to even enumerate them, an unbiased estimator for the k-bounded contributions, *estimated k-bounded contributions*, can be calculated by sampling permutations while ignoring the configurations with more than k perturbed elements in each permutation. Based on these, an estimated predicted variant, as in Section 5, can also be calculated, and in addition a two-phase procedure can be carried out, as in Section 6.

The k-bounded contributions enable one to examine the space between the contributions yielded by single perturbations only (single-lesion analysis) and the contributions yielded by a full MSA. For each $k = 1, 2, \ldots, 10$, we calculated the k-bounded contributions of the neurons of the fault-tolerant agent F10. Figure 9 depicts the distance between the normalized k-bounded contributions vector and the normalized Shapley value vector of contributions, as a function of k. The k-bounded contributions gradually approach the Shapley value, starting from the single-perturbation contributions, which in this case are as far from the Shapley value as random normalized vectors are. A previous analysis has revealed the distance between the k-bounded contributions and the Shapley

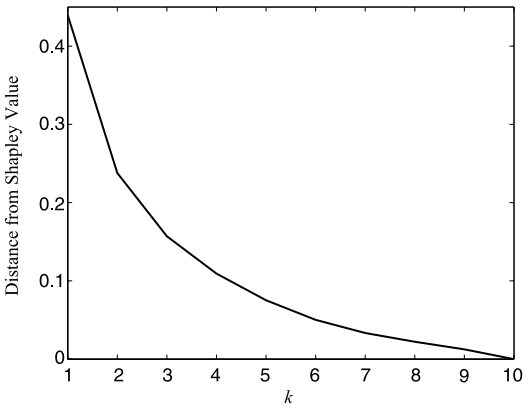

Figure 9. *k*-Bounded contributions versus Shapley value contributions. The Euclidean distance between the normalized MSA *k*-bounded contributions and the normalized full MSA contributions (the Shapley value) as a function of the perturbation level *k*, for agent F10.

value to be larger for fault-tolerant neurocontrollers than for regular ones, due to the functional overlap between the neurons [15].

8 Two-Dimensional MSA

The Shapley value serves as a summary of the game, indicating the average marginal importance of an element over all possible element permutations. For complex systems, where the importance of an element may depend on the state (perturbed or intact) of other elements, a higher-order description is necessary to capture sets of elements with significant interactions. For example, when two elements exhibit a high degree of functional overlap, that is, redundancy, it is desired to capture this interaction, aside from the average importance of each element. We focus here on the description of two-dimensional interactions, elaborated upon in [18], and present its applicability for the analysis of neurocontrollers.

A natural definition of the interaction between a pair of elements is as follows: Let $\gamma_{i,j} = \gamma_i(\mathrm{N}\backslash\{j\}, v^{\mathrm{N}\{j\}})$ be the Shapley value of element i in the subgame of all elements without element j. Intuitively, this is the average marginal importance of element i when element j is perturbed. Let us now define the coalitional game (M, v^M), where $M = N \setminus \{i,j\} \cup \{(i,j)\}$ [(i,j) is a new compound element] and $v^M(S)$, for $S \subseteq M$, is given by

$$v^M(S) = \begin{cases} v(S), & (i,j) \notin S, \\ v(S\backslash\{(i,j)\} \cup \{i,j\}), & (i,j) \in S, \end{cases} \tag{10}$$

where v is the characteristic function of the original game with elements N. Then $\gamma_{i,j} = \gamma_{(i,j)}(M, v^M)$, the Shapley value of element (i,j) in this game, is the average marginal importance of elements i and j when jointly added to a configuration. The two-dimensional interaction between element i and element j, $j \neq i$, is then defined as

$$I_{i,j} = \gamma_{i,j} - \gamma_{i,\bar{j}} - \gamma_{j,\bar{i}}, \tag{11}$$

which quantifies how much the average marginal importance of the two elements together is larger (or smaller) than the sum of the average marginal importances of each of them when the other one is

perturbed. Intuitively, this symmetric definition ($I_{i,j} = I_{j,i}$) states how much "the whole is greater than the sum of its parts," where the whole is the pair of elements. In cases where the whole is indeed larger than the sum of its parts, the interaction is positive. In cases where the whole is smaller than the sum of its parts, that is, when the two elements exhibit functional overlap, the interaction is negative.

Based on the two-dimensional interactions presented above, Figure 10a portrays the results of a two-dimensional analysis performed on the counting agent W10, extending the one-dimensional analysis (Section 3). Evidently, all pairs of significant neurons found previously in the one-dimensional analysis (1, 4, 9, and 10) exhibit strong positive interaction, while the pairs involving non-significant neurons exhibit weak interactions. Specifically, neurons 4 and 9, participating in the counting process when waiting in a food cell, exhibit the strongest interaction. Further examining the marginal contributions of each with respect to the other, neuron 9 has a very significant contribution of 0.16 when neuron 4 is intact. When neuron 4 is perturbed, however, neuron 9 has a near-vanishing contribution of 0.005, showing that neuron 9 cannot count by itself, without neuron 4. The opposite is also true, as neuron 4 has a significant contribution of 0.15 when neuron 9 is intact, and a vanishing contribution when it is perturbed. Interestingly, the excess of positive over negative interactions indicates that there is an evolutionary pressure towards the formation of cooperation between neurons.

Observing the interactions between all pairs of neurons of the fault-tolerant agent F10 (Figure 10b) reveals many negative ones, pointing to pairs of neurons that back up each other's function. These results exemplify the multiplicity of negative interactions in agents evolved while faults are introduced into the neurocontrollers. The necessity of fault tolerance induces the emergence of functional overlap between the neurons, at the expense of the formation of cooperation [15].

9 Discussion

Over the last couple of years we have developed and described a new framework for quantitative causal function localization via multi-perturbation experiments. Departing from the original ad hoc error minimization approach [1, 26, 17], we recently presented the MSA—an axiomatic framework based on an analytical definition of all elements' contributions via the Shapley value [18]. In this article we further develop this framework and present estimation and prediction variants, which allow it to efficiently and accurately identify the contributing elements of much larger systems. We demonstrate the MSA's applicability by applying it to a group of agents solving a similar food-foraging task and guided by fully recurrent neurocontrollers consisting of up to a hundred synapses.

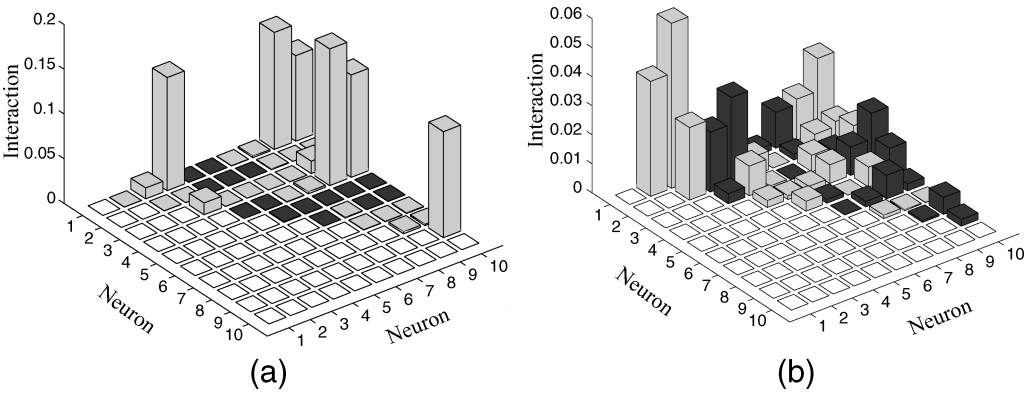

Figure 10. Two-dimensional interactions. The symmetric interaction $I_{i,j}$ between each pair of neurons ($i < j$) is portrayed for (a) agent W10 and (b) agent F10. The graphs present the absolute values of the interactions, with dark bars denoting the negative interactions and light bars denoting the positive ones. Note the different scales of the two figures.

The analysis uncovers the important neuronal and synaptic elements in each agent and quantifies their contributions to its performance, as well as their functional interactions. The MSA was further utilized to prune the synapses of a successful agent's neurocontroller while maintaining a high level of performance.

While this article has focused on the analysis of EAAs' neurocontrollers, the MSA is a general function localization framework applicable to a wide variety of systems. Indeed, there are only two requirements for a system to be eligible for such an analysis—an ability to impose multiple perturbations upon its constituent elements, and an ability to measure its performance in respect to the studied functions. Moving beyond artificial systems, recent technological advances have enabled real biological systems to meet these requirements. In neuroscience, lesioning, reversible deactivation, and transcranial magnetic stimulation (TMS) all allow the collection of multi-perturbation performance data. Outside of neuroscience, the recent development of RNA interference (RNAi) has made multiple concomitant gene knockouts possible, opening the way for a multi-perturbation analysis of genetic networks. As the biological systems under investigation grow in size, the MSA estimation variants presented in this article will be required for a computationally tractable analysis. The k-bounded MSA should be particularly useful for such analyses, as current techniques can only perturb a relatively small number of elements concomitantly.

Due to its generality, the MSA cannot be viewed as a black box to be used, as is, on a given system in order to gain insight into its functioning. Rather, three issues have to be resolved before the MSA can be applied—what constitutes an element in the system; what system function ought to be analyzed; and what perturbation method should be used. While at first sight the need to answer those questions might seem like a drawback of the framework, in fact it is exactly this flexibility that gives the MSA much of its strength. Conducting several analyses of the same system, each with a different choice of element definition, studied function, or perturbation method, may engender an in-depth understanding of all aspects and levels of its inner workings. The following paragraphs discuss each of these three issues in turn.

1. *Choice of elements.* Implicit in the description of the MSA was the assumption that the system to be analyzed is composed of discrete identifiable elements. While in most systems some definition of elements may be more natural than others, it is often possible, perhaps even desirable, to consider other definitions as well. For example, while neurocontrollers are most often seen as composed of neurons, so that their analysis may be expected to uncover the different neuronal contributions to the neurocontroller's performance, we have shown in this article several analyses on the level of single synapses, resulting in a more fine-grained understanding of the network. Conversely, while biological brains are composed of neurons as well, lesion analyses work on the level of whole brain regions, taking each region as a single element. Within the context of biological knockout experiments, genes are most often seen as the basic elements. However, in some contexts it might be preferable to consider single amino acids as the basic elements, or perhaps consider whole genetic subsystems.
2. *Choice of the studied function.* This issue is essential in any functional localization analysis. Applying the MSA for different functions performed by the same system will uncover the contribution of each element to each function separately. A thoughtful specification of these functions may help to illuminate different aspects of the system's behavior. Aharonov et al. [1] used the FCA to analyze agents developed in the environment presented in this article with respect to two performance measures, reflecting the two modes of behavior these agents display (see Section 2). The first measures the agent's capacity to travel long distances, and hence reflects its capacity for exploration. The second, reflecting its capacity for grazing, measures the agent's ability to cover large areas in its search for food. The contributions derived from this analysis were then used to assess the degree of localization of each task, as well as the extent to which each neuron is specialized to a given task. As mentioned in the introduction, we used the full-information MSA on a single segment of a neurophysiological model of a swimming lamprey to uncover the

synaptic backbones underlying its oscillatory activity and various characteristics of these oscillations [18]. This was achieved by defining a different function for each characteristic of interest and running the MSA on the multi-perturbation data gathered for each function. Saggie et al. [24] focused on the study of counting agents such as W10 mentioned in this article. A careful definition of the function to be analyzed allowed the MSA to reveal those synapses specifically in charge of counting, differentiating them from the synapses that contribute to other aspects of the agent's performance.

3. *Choice of the perturbation method.* This issue was only briefly touched upon in this article. Much as in the choice of function, different perturbation methods may be used to illuminate different aspects of the system's functioning. While the most natural choice for perturbing a neural element would probably be simply removing it from the network, this is by far not the only alternative. Indeed, Aharonov et al. [1] suggested using stochastic lesioning, which effectively negates the information content of the perturbed element's activity, without changing its mean firing rate, and hence without affecting the mean input field of other elements. Keinan et al. [17] generalized this concept in the development of the informational lesioning method, which makes available a whole range of increasingly minute perturbations, once again affecting only the information content of the perturbed element. It was shown that different perturbation magnitudes tend to reveal different aspects of the neural functioning, with the smaller magnitudes accentuating neural elements involved in longer-range dynamics. Lastly, Saggie et al. [24] evolved neurocontrollers composed of discrete-time integrate-and-fire neurons. In order to determine the contributions of the neurons' integration capability to the agent's overall performance (as opposed to the importance of the neuron's information content, as revealed by using stochastic lesioning), a perturbation method was used that simply set a perturbed neuron's membrane time constant to zero, thereby rendering it a standard McCulloch-Pitts neuron.

As evolved neurocontrollers grow more and more complex, there is a clear need for principled methods for their analysis. As pointed out previously, and as evident from effects such as functional overlap, single-lesion approaches do not suffice to portray the correct function localization in a system. Aiming at a causal analysis, multiple perturbations must be imposed upon the neurocontroller. The MSA is the first framework to make sense out of such multi-perturbation experiments in a formal, axiomatic, and rigorous manner. The newly introduced variants now extend its capabilities by increasing its scalability and allowing for the efficient analysis of large-scale systems.

Acknowledgments

We acknowledge the valuable contributions and suggestions made by Ranit Aharonov, Alon Kaufman, and Ehud Lehrer. This research has been supported by the Adams Super Center for Brain Studies in Tel Aviv University, by the Israel Science Foundation founded by the Israel Academy of Sciences and Humanities, and by the Horowitz-Ramot foundation. A. K. is supported by the Dan David Prize scholarship.

References

1. Aharonov, R., Segev, L., Meilijson, I., & Ruppin, E. (2003). Localization of function via lesion analysis. *Neural Computation*, *15*, 885–913.
2. Aharonov-Barki, R., Beker, T., & Ruppin, E. (2001). Emergence of memory-driven command neurons in evolved artificial agents. *Neural Computation*, *13*, 691–716.
3. Beker, T., & Hadany, L. (2002). Noise and elitism in evolutionary computation. *Frontiers in Artificial Intelligence and Applications*, *87*, 193–201.
4. Billera, L. J., Heath, D., & Raanan, J. (1978). Internal telephone billing rates—A novel application of non-atomic game theory. *Operations Research*, *26*, 956–965.
5. Ekeberg, O. (1993). A combined neuronal and mechanical model of fish swimming. *Biological Cybernetics*, *69*, 363–374.

6. Feigenbaum, J., Papadimitriou, C. H., & Shenker, S. (2001). Sharing the cost of multicast transmissions. *Journal of Computer and System Sciences*, *63*, 21–41.

7. Floreano, D., & Mondada, F. (1996). Evolution of homing navigation in a real mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics—Part B*, *26*, 396–407.

8. Floreano, D., & Mondada, F. (1998). Evolutionary neurocontrollers for autonomous mobile robots. *Neural Networks*, *11*, 1461–1478.

9. Ganon, Z., Keinan, A., & Ruppin, E. (2003). Evolutionary network minimization: Adaptive implicit pruning of successful agents. In W. Banzhaf, T. Christaller, P. Dittrich, J. T. Kim, & J. Ziegler (Eds.), *Advances in Artificial Life—Proceedings of the 7th European Conference on Artificial Life (ECAL)* (pp. 319–327). Berlin: Springer-Verlag.

10. Gefeller, O., Land, M., & Eide, G. E. (1998). Averaging attributable fractions in the multifactorial situation: Assumptions and interpretation. *Journal of Clinical Epidemiology*, *51*, 437–441.

11. Gomez, F., & Miikkulainen, R. (1997). Incremental evolution of complex general behavior. *Adaptive Behavior*, *5*, 317–342.

12. Guillot, A., & Meyer, J. A. (2001). The animat contribution to cognitive systems research. *Journal of Cognitive Systems Research*, *2*, 157–165.

13. Kapur, N. (1996). Paradoxical functional facilitation in brain-behavior research. A critical review. *Brain*, *119*, 1775–1790.

14. Kaufman, A., Kupiec, M., & Ruppin, E. (2004). Multi-knockout genetic network analysis: The Rad6 example. In *Proceedings of IEEE Computational Systems Bioinformatics Conference (CSB'04)* (pp. 332–340).

15. Keinan, A (2004). Analyzing evolved fault-tolerant neurocontrollers. In *Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9)* (pp. 557–562).

16. Keinan, A., Kaufman, A., Sachs, N., Hilgetag, C. C., & Ruppin, E. (2004). Fair localization of function via multi-lesion analysis. *Journal of Neuroinformatics*, *2*, 163–168.

17. Keinan, A., Meilijson, I., & Ruppin, E. (2003). Controlled analysis of neurocontrollers with informational lesioning. *Philosophical Transactions of the Royal Society of London: Series A*, *361*, 2123–2144.

18. Keinan, A., Sandbank, B., Hilgetag, C. C., Meilijson, I., & Ruppin, E. (2004). Fair attribution of functional contribution in artificial and biological networks. *Neural Computation*, *16*, 1887–1915.

19. Kodjabachian, J., & Meyer, J. A. (1998). Evolution and development of neural controllers for locomotion, gradient-following and obstacle-avoidance in artificial insects. *IEEE Transactions on Neural Networks*, *9*, 796–812.

20. Marocco, D., & Floreano, D. (2002). Active vision and feature selection in evolutionary behavioral systems. In D. Cliff, P. Husbands, J. A. Meyer, & S. K. Wilson (Eds.), *Proceedings of the Third International Conference on Simulation of Adaptive Behavior (SAB2002)*. Cambridge, MA: MIT Press.

21. Reisinger, J., Stanley, K. O., & Miikkulainen, R. (2004). Evolving reusable neural modules. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2004)*. New York: Springer-Verlag.

22. Roth, A. E. (1979). *Axiomatic models of bargaining*. Berlin: Springer-Verlag.

23. Ruppin, E. (2002). Evolutionary autonomous agents: A neuroscience perspective. *Nature Reviews Neuroscience*, *3*, 132–141.

24. Saggie-Wexler, K., Keinan, A., & Ruppin, E. (in press). Neural processing of counting in evolved spiking and McCulloch-Pitts agents. *Artificial Life*.

25. Scheier, C., Pfeifer, R., & Kunyioshi, Y. (1998). Embedded neural networks: Exploiting constraints. *Neural Networks*, *11*, 1551–1569.

26. Segev, L., Aharonov, R., Meilijson, I., & Ruppin, E. (2003). High-dimensional analysis of evolutionary autonomous agents. *Artificial Life*, *9*, 1–20.

27. Shapley, L. S. (1953). A value for *n*-person games. In H. W. Kuhn & A. W. Tucker (Eds.), *Contributions to the Theory of Games* (pp. 307–317). Princeton, NJ: Princeton University Press.

28. Shapley, L. S., & Shubik, M. (1954). A method for evaluating the distribution of power in a committee system. *The American Political Science Review*, *48*, 787–792.

29. Shubik, M. (1962). Incentives, decentralized control, the assignment of joint costs and internal pricing. *Management Science*, *8*, 325–343.

30. Shubik, M. (1985). *Game theory in the social sciences*. Cambridge, MA: MIT Press.

31. Sprague, J. M. (1966). Interaction of cortex and superior colliculus in mediation of visually guided behavior in the cat. *Science*, *153*, 1544–1547.

32. Stanley, K. O., & Miikkulainen, R. (2004). Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, *21*, 63–100.

33. Stanley, K. O., & Miikkulainen, R. (2004). Evolving a roving eye for Go. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2004)*. New York: Springer-Verlag.

34. Yao, X. (1999). Evolving artificial neural networks. *Proceedings of the IEEE*, *87*, 1423–1447.