LETTER **Communicated by Inman Harvey**

The Evolution of Representation in Simple Cognitive Networks

Lars Marstaller
lars.marstaller@mq.edu.au
Department of Cognitive Science, Macquarie University, Sydney NSW 2101, Australia

Arend Hintze
hintze@msu.edu
Microbiology and Molecular Genetics, BEACON Center, and Computer Science and Engineering, Michigan State University, East Lansing, MI 48244, U.S.A.

Christoph Adami
adami@msu.edu
Microbiology and Molecular Genetics and BEACON Center, Michigan State University, East Lansing, MI 48244, U.S.A.

Representations are internal models of the environment that can provide guidance to a behaving agent, even in the absence of sensory information. It is not clear how representations are developed and whether they are necessary or even essential for intelligent behavior. We argue here that the ability to represent relevant features of the environment is the expected consequence of an adaptive process, give a formal definition of representation based on information theory, and quantify it with a measure *R*. To measure how *R* changes over time, we evolve two types of networks—an artificial neural network and a network of hidden Markov gates—to solve a categorization task using a genetic algorithm. We find that the capacity to represent increases during evolutionary adaptation and that agents form representations of their environment during their lifetime. This ability allows the agents to act on sensorial inputs in the context of their acquired representations and enables complex and context-dependent behavior. We examine which concepts (features of the environment) our networks are representing, how the representations are logically encoded in the networks, and how they form as an agent behaves to solve a task. We conclude that *R* should be able to quantify the representations within any cognitive system and should be predictive of an agent's long-term adaptive success.

L. Marstaller and A. Hintze contributed equally.

Neural Computation **25, 2079–2107** (2013) © 2013 Massachusetts Institute of Technology
doi:10.1162/NECO_a_00475

1 Introduction

The notion of representation is as old as cognitive science itself (Chomsky, 1965; Newell & Simon, 1972; Fodor, 1975; Johnson-Laird & Wason, 1977; Marr, 1982; Pinker, 1989; Pitt, 2008), but its usefulness for artificial intelligence (AI) research has been doubted (Brooks, 1991). In his widely cited article, "Intelligence Without Representation," Brooks argued instead for a subsumption architecture where the autonomous behavior-producing components (or layers) of the cognitive system directly interface with the world and with each other rather than with a central symbol processor dealing in explicit representations of the environment. In particular, inspired by the biological path to intelligence, Brooks argued that AI research needs to be rooted in mobile autonomous robotics and a direct interaction between action and perception. Echoing Moravec (1984), he asserted that the necessary elements for the development of intelligence are mobility, acute vision, and the ability to behave appropriately in a dynamic environment (Brooks, 1991). This architecture achieved insect-level intelligence, and Brooks argued that a path to higher-level AI could be forged by incrementally increasing the complexity of subsumption architecture.

However, 20 years after his advocating such a radical departure from the classical approach to AI, the subsumption approach seems to have stalled as well. We believe that the reason for the lack of progress does not lie in the attempt to base AI research in mobile autonomous robots, but that instead representations—also sometimes called internal models (Craik, 1943; Wolpert, Ghahramani, & Jordan, 1995; Kawato, 1999)—are key to complex adaptive behavior. Indeed, while representation-free robotics has made some important strides (Nolfi, 2002), they are limited to problems that are not "representation hungry" (Clark, 1997), that is, problems that do not require past information or additional (external) knowledge about the current context. In addition, the technical difficulty of developing a subsumption architecture increases with the number of layers or subsystems. This problem of subsumption architecture mirrors the difficulties of classic representational AI approaches to build accurate and appropriate models of the world.

An alternative approach to engineering cognitive architectures and internal models is evolutionary robotics (Nolfi & Floreano, 2000). Instead of designing the structure or functions of a control architecture, principles of Darwinian evolution are used to create complex networks that interface perception and action in nonobvious and often surprising ways. Such structures can give rise to complex representations of the environment that are hard to engineer and equally hard to analyze (Floreano & Mondada, 1996). Evolved representations provide context, are flexible, and can be readjusted given new stimuli that contradict the current assumptions. Representations can be updated during the lifetime or over the course of evolution and thus are able to handle even new sensory input (Bongard, Zykov, & Lipson,

2006). We argue that as robots evolve to behave appropriately (and survive) in a dynamic and noisy world, representations of the environment emerge within the cognitive apparatus and are integrated with the perceived sensory data to create intelligent behavior, using not only the current state of the environment but, crucially, taking into account historical data (memory) as well.

To test this hypothesis and make internal representations of evolved systems accessible to analysis, we propose a new information-theoretic measure of the degree to which an embodied agent represents its environment within its internal states and show how the capacity to represent environmental features emerges over thousands of generations of simulated evolution. The main idea is that representations encode environmental features because of their relevance for the cognitive system in question (Clark & Toribio, 1994). Hence, for our purposes, representations can be symbolic or subsymbolic (e.g., neural states) as long as they have a physical basis, that is, as long as they are encoded in measurable internal states. However, we distinguish representations from sensorial input because sensor inputs cannot provide the same past or external context as internal states. We thus explicitly define representations as that information about relevant features of the environment encoded in the internal states of an organism and goes beyond the information present in its sensors (Haugeland, 1991; Clark, 1997). In particular, this implies that representations can at times misrepresent (Haugeland, 1991)—unlike information present in sensors, which always truthfully correlates with the environment. To illuminate the functioning of evolved cognitive systems, we show how it is in principle possible to determine what a representation is about and how representations form during the lifetime of an agent. We argue that our measure provides a valuable tool to investigate the organization of evolved cognitive systems, especially in cases where internal representations are epistemically opaque.

2 Methods

2.1 Information-Theoretic Measure of Representation. Information theory has been used previously to quantify how context can modulate decisions based on sensory input (Phillipps, Kay, & Smyth, 1994; Phillips & Singer, 1997; Kay & Phillips, 2011). Here, we present an information-theoretic construction that explicitly takes the entropy of environmental states into account. To quantify representation, we first define the relationship between the representing system and the represented environment in terms of information (shared, or mutual, entropy). Information measures the correlation between two random variables, while the entropy H is a measure of the uncertainty we have about a random variable in the absence of information (uncertainty is therefore potential information). For a random variable X that can take on the states x_i with probabilities $p(x_i) = P(X = x_i)$,

the entropy is given by (Shannon, 1948)

$$H(X) = -\sum_{i=1}^{N} p(x_i) \log p(x_i) \ , \tag{2.1}$$

where N is the number of possible states that X can take on.

The information between two random variables characterizes how much the degree of order in one of the variables is predictive of the regularity in the other variable. It can be defined using entropy as the difference between the sum of the entropies of two random variables X and Y (written as $H(X)$ and $H(Y)$) and the joint entropy of X and Y, written as $H(X,Y)$:

$$I(X:Y) = H(X) + H(Y) - H(X,Y) = \sum_{xy} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \ . \tag{2.2}$$

In equation 2.2, $p(x)$ and $p(y)$ are the probability distributions for the random variables X and Y, respectively (i.e., $p(x) = P(X = x)$), while $p(x,y)$ is the joint probability distribution of the (joint) random variable XY. The shared entropy $I(X:Y)$ can also be written in terms of a difference between unconditional and conditional entropies as

$$I(X:Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \ . \tag{2.3}$$

This definition reminds us that information is that which reduces our uncertainty about a system. In other words, it is that which allows us to make predictions about a system with an accuracy that is higher than when we did not have that information. In equation 2.3, we introduced the concept of a conditional entropy (Shannon, 1948). For example, $H(X|Y)$ (read as "H of X given Y") is the entropy of X when the state of the variable Y is known and is calculated as

$$H(X|Y) = -\sum_{xy} p(x,y) \log p(x|y) \ , \tag{2.4}$$

using the conditional probability $p(x|y) = p(x,y)/p(y)$.

In general, information is able to detect arbitrary correlations between signals or sets of events. We assume here that such correlations instantiate semiotic or information relationships between representing and represented and use mutual information to measure the correlation between a network's internal states and its environment (see also Marstaller, Hintze, & Adami, 2010). So, for example, we could imagine that X stands for the states of an environment, whereas Y is a variable that represents those states of the environment. We need to be careful, however, to exclude from possible representational variables those that are mere images of the environment,

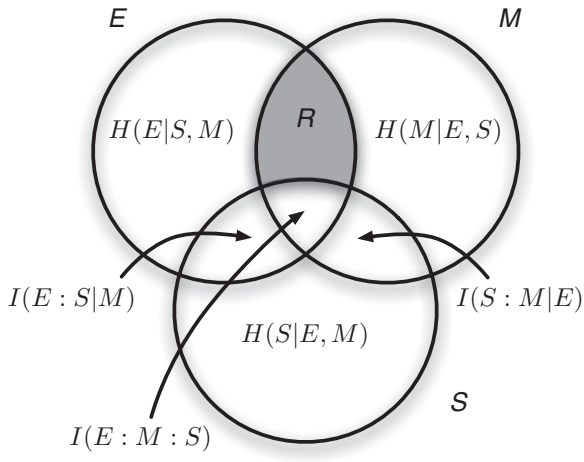

Figure 1: Venn diagram of entropies and informations for the three random variable E, S, and M, describing the states of the environment, sensors, and agent internal degrees of freedom. The representation $R = H(E : M|S)$ is shaded.

such as the trace that the world leaves in an agent's sensors. Indeed, mere correlations between internal states and the environment are not sufficient to be treated as representational because they could be due to behavior that is entirely reactive (Clark, 1997). Haugeland (1991), for example, understands representation as something that "stands in" for something in the environment but is no longer reflected in the perceptual system of the agent. Indeed, representation should be different from a mere translation. Consider a digital camera's relationship with its environment. The photo chip guarantees a one-to-one mapping between the environment structure and the camera's state patterns. But a camera is not able to adapt to its environment. By taking a picture, the camera has not "learned" anything about its environment that will affect its future state. It simply stores what it received through its inputs without extracting information from it; the camera's internal states are fully determined by its sensor inputs. Representation goes beyond mere translation because the content—which feature of the environment is represented—depends on the goals of the system. Not everything is represented in the same way. A camera does not have this functional specification of its internal states.

To rule out trivial representations like a camera's internal states, we define *representation* as the shared entropy between environment states and internal states, but given the sensor states (i.e., conditioned on the sensors). Thus, representation is that part of the shared entropy between environment states and internal states that goes beyond what is seen in the sensors (see Figure 1). For the following, we take E, given by its probability distribution

$p(e_i) = P(E = e_i)$, as the random variable to describe environmental states, while S describes sensor states. If the internal states of the agent (hidden and output states) are characterized by the random variable M with probability distribution $p(m_j) = P(M = m_j)$, then we define the representation R as (for an earlier version, see Marstaller et al., 2010)

$$R = H(E : M|S) = I(E : M) - I(E : M : S) = H_{\text{corr}} - I(S : E) - I(S : M), \tag{2.5}$$

where the correlation entropy H_{corr} of the three variables E, S, and M (also called total correlation; Watanabe, 1960) or multi-information (McGill, 1954; Schneidman, Still, Berry, & Bialek, 2003) is the amount of information that all three share:

$$H_{\text{corr}} = H(E) + H(M) + H(S) - H(E, M, S)\ . \tag{2.6}$$

In equation 2.5, we introduced the shared conditional entropy between three variables that is defined as the difference between information that is unshared and that is shared (with a third system), just as $H(X|Y) = H(X) - I(X : Y)$, from equation 2.4. Thus, the representation R of the world E within internal states M is the total correlation between the three, but without what is reflected in S about E and M, respectively (measured by $I(S : E)$ and $I(S : M)$). The relationship between R and the entropies of the three variables S, E, and M is most conveniently summarized by an entropy Venn diagram, as in Figure 1. In these diagrams, a circle is a quantitative measure of the entropy of the associated variable, and the shared entropy between two variables is represented by the intersection of the variables, and so on (Cover & Thomas, 1991).

Our information-theoretic definition of *representation* carries over from discrete variables to continuous variables unchanged, as can be seen as follows. Let X_E, X_M, and X_S be random variables defined with normalized probability density functions $f_E(e)$, $f_M(m)$, and $f_S(s)$. The differential entropy $h(X)$ (Cover & Thomas, 1991), defined as

$$h(X) = -\int_{\mathcal{S}} f(x) \log f(x) dx, \tag{2.7}$$

where $\mathcal{S}$ is the support of random variable X, is related to the discretized version $H(X^\Delta)$ by noting that

$$H(X^\Delta) + \log \Delta \to h(X) \quad (\Delta \to 0)\ , \tag{2.8}$$

where we introduced the discretization

$$p_i = \int_{i\Delta}^{(i+1)\Delta} f(x)dx \tag{2.9}$$

in order to define the discretized Shannon entropy $H(X^\Delta) = -\sum_i p_i \log p_i$. This implies that an n-bit quantization of a continuous random variable is approximately $H(X) \approx h(X) + n$ (Cover & Thomas, 1991). Let us now assume that the variables X_E, X_M, and X_S are each quantized by n_E, n_M, and n_S bits, respectively. Because $H(E, M, S)$ is then quantized by $n_E + n_M + n_S$ bits, it follows that $H_{\text{corr}} \approx h_{\text{corr}}$, that is, the continuous and discrete variable correlation entropies are (in the limit of sufficiently small Δ) approximately the same because the discretization correction cancels. The same is true for the information $I(S : E)$ and $I(S : M)$, as these are correlation entropies between two variables. Thus, $H(E : M|S) \approx h(X_E : X_M|X_S)$, the differential entropy version of R. We stress that while an exact identity between discrete and continuous variable definitions of R is ensured only in the limit of vanishing discretization, the cancellation of the correction terms $\log \Delta$ implies that the discrete version is not biased with respect to the continuous version.

R defines a relation between a network's activity patterns and its environment as the result of information processing. R yields a positive quantity, measured in bits (if logarithms are taken to base 2). In order to show that this measure of representation reflects functional purpose (Clark, 1997), we evolve cognitive systems (networks) that control the behavior of an embodied agent and show that fitness, a measure for the agent's functional prowess, is correlated with R. In other words, we show that when the environment (and task) is complex enough, agents react to this challenge by evolving representations of that environment.

2.2 Evolution of Active Categorical Perception. We study the evolution of an agent that solves an active categorical perception (ACP) task (Beer, 1996, 2003), but with modifications suggested by van Dartel, Sprinkhuizen-Kuyper, Postma, and van den Herik (2005); (see also van Dartel, 2005). Categorization is thought to be one of the key elements in cognition (see Harnad, 1987; Cohen & Lefebvre, 2005). In categorical perception, an agent has to partition objects in the world into different discrete categories based on their visual appearance. In active categorical perception, the agent takes an active role in which aspects of the object to view, that is, perception, are intimately linked with action. Whether this task requires internal representations may depend on the specific nature of the task, and it is in general not simple to determine whether the agent uses internal states to represent the environment and, in particular, which features of the world are represented by which computational units (Beer, 2003; Ward & Ward, 2009). In the task studied by Beer (1996, 2003), the agent has to discriminate between

circles and diamonds that were falling vertically toward the agent, who could move laterally to change its perception of the object. In the version we study here, the agent has to discriminate between large and small blocks by catching the small blocks while avoiding the large ones. In order to create visual ambiguity, the agent's four upward-looking sensors have a blind spot (two nonsensing units) between them. Furthermore, blocks are falling diagonally (from left to right or right to left), so that the agent not only has to categorize but also to predict (see Figure 2A).

We evolve active categorical perception in two experiments using two different control architectures. One is an artificial neural network (ANN) and one is a Markov network, specifically, a network of hidden Markov gates (HMGs) (Edlund et al., 2011), described in more detail in section 2.3. The agent is located at the bottom row of a 20×20 world with periodic boundary conditions and is able to move horizontally one unit per time step (see Figure 2A). Note that compared to van Dartel et al. (2005), we doubled the vertical size of the arena in order to give the agent more time to assess the direction of the falling block. The agent has four upward-looking sensors. The state of a sensor is $s_i = +1$ when a block is detected and $s_i = 0$ if no block is seen. The arrangement of the agent's sensors does not allow it to unambiguously identify the falling block from three out of eight possible input states (counting the input where no sensors are active). Only the input states $S = (s_0, s_1, s_2, s_3) = (0, 1, 1, 0)$, $(0, 1, 0, 0)$, and $(0, 0, 1, 0)$ are unambiguous. The first classifies a large block that is positioned right above the agent so that the second and the third sensors are in state $s_1 = s_2 = 1$, while the remaining two units of the block are positioned above the two nonsensing units. The other two patterns identify similarly positioned small blocks. All other five input states can be created by a small block, a big block, or no block at all (see Figure 2B). At each trial, a block of either small (two units) or large size (four units) falls from top to bottom in 20 time steps. The blocks move continuously downward and sideways one unit per time step. Blocks always move to either the right or the left. An object is caught if the position of the block's units and the agent's units at time step 20 overlap in at least one unit.

For the information-theoretic characterization of correlations, we have to assign probabilities to the possible states of the world. Theoretically, a falling block can be in any of 20 different starting positions, large or small, and falling left or right, giving rise to 80 possible experimental initial conditions. While the agent can be in any of 20 initial positions, the periodic boundary conditions ensure that each of them is equivalent, given the 20 initial positions of the falling block. Because there are 20 time steps before the block reaches the bottom row, there are in total 1600 possible different states the world can be in. We do not expect that all of these states will be discriminated by the agent, so instead we introduce a coarse graining of the world by introducing four bits that we believe capture salient aspects of the world. We define the environmental (joint) variable $E = E_0E_1E_2E_3$ to take on

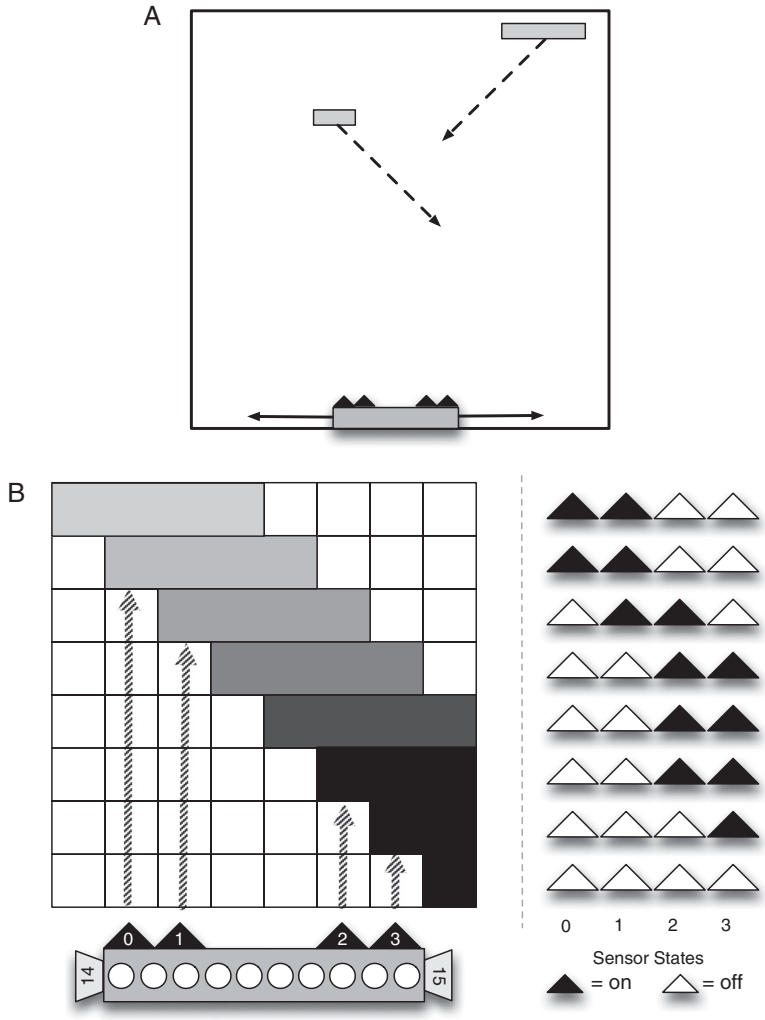

Figure 2: (A) In the simulation, large or small blocks fall diagonally toward the bottom row of a 20 × 20 world, with the agent on the bottom row. For the purpose of illustrating the task, a large brick (to be avoided) is falling to the left, while a small brick (to be caught) is falling to the right. In simulations, only one block is falling at the time, and both small and large bricks can fall to either the left or the right. (B) A depiction of the agent's neurons (bottom left: triangles depict sensors, circles illustrate brain neurons, and trapezoids denote actuators) and the sequence of activity patterns on the agent's four-bit retina (right) as a large brick falls to the right.

Table 1: Coarse Graining of World States into the Four bits E_0, E_1, E_2, E_3.

World State	World Character
$E_0 = 0$	No sensor activated
$E_0 = 1$	At least one sensor activated
$E_1 = 0$	Block is to the left of agent
$E_1 = 1$	Block is to the right of agent
$E_2 = 0$	Block is two units (small)
$E_2 = 1$	Block is four units (large)
$E_3 = 0$	Block is moving left
$E_3 = 1$	Block is moving right

Notes: E_1 could be ambiguous in case the block is centered over the agent or exactly 10 units away. We resolve this ambiguity by setting $E_1 = 0$ when the block is centered over the agent, and $E_1 = 1$ when it is exactly 10 units away.

states as defined in Table 1. Of course, this encoding reveals a bias in what we, the experimenters, believe are salient states of the world, and it certainly underestimates the amount of "discoverable" entropy. However, in hindsight, this coarse graining appears to be sufficient to capture the essential variations in the world and furthermore lends itself to study which aspects of the world are being represented within the agent's network controller by defining representations about different aspects i of the world as the representation $R_i = H(E_i : M|S)$. Thus, we will study four representations,

$$R_{\text{hit}} = H(E_0 : M|S), \tag{2.10}$$

$$R_{\text{LR}} = H(E_1 : M|S), \tag{2.11}$$

$$R_{4/2} = H(E_2 : M|S), \tag{2.12}$$

$$R_{+/-} = H(E_3 : M|S), \tag{2.13}$$

that represent whether the sensor has been activated (see equation 2.10), whether the block is to the left or the right of the agent (see equation 2.11), if the block is large (size 4) or small (size 2) (see equation 2.12), or whether the block is moving to the left or right (see equation 2.13). We can also measure how much (measured in bits) of each binary concept is represented in any particular variable. For example, $R_{\text{LR}}(\text{node } 12) = H(E_1 : M_{12}|S)$ measures how much of the "block is to my left or to my right" concept is encoded in variable 12.

2.3 Two Architectures for Cognitive Systems. The agent is controlled by a cognitive system, composed of computational units (loosely referred

to as neurons from here on) that map sensor inputs into motor outputs. The cognitive system also has neurons that are internal (a hidden layer), which are those neurons that are not part of the input or the output layer. We further define sensor neurons as the neurons that directly process the input (the input layer), and we define output neurons as the units that do not map to other units in the network or to themselves (the output layer).

2.3.1 Artificial Neural Networks with Evolvable Topology. In our first experiment, the robot's movements are controlled by an artificial neural network (ANN) that consists of 16 nodes: 4 input units (one for each sensor), 2 output units, and 10 hidden units. The states of the input units are discrete with values $[+1, -1]$ specifying whether an object is detected. The states of the output units (or actuators) are discrete with integer values $A = a \in [0, 1]$ encoding one of three possible actions: move one unit to the right or left, or do not move ($A_1A_2 = 00$: stand still; $A_1A_2 = 01$: move right; $A_1A_2 = 10$: move left; $A_1A_2 = 11$: stand still. While the hidden units' states m_i are continuous with values $[-1.0, 1.0]$, when evaluating these states to calculate R, we discretize them to binary (values below 0.0 become 0; every other value becomes 1). As discussed earlier, this discretization does not introduce a bias in the value of R.

Usually classic ANNs have a fixed topology—one or more layers and their connections are defined and associated with a weight. In a previous experiment, we found that such fixed topologies lead to approximately constant R even as the fitness of the agent increases (data not shown). One way to increase the complexity of the network and the information it represents is to evolve a network's topology as well as the connection weights. We make the network topology evolvable beyond searching the connection weights by using neuronal gates (NG). An NG can arbitrarily connect nodes of any type (input, hidden, and output nodes) without the fixed layered topology of classic ANNs. Each connection is associated with a certain weight. An NG calculates the sum of the values from a set of incoming nodes $m_j^{(k)}$ via gate k, multiplied by the associated weight $w_j^{(k)}$, and applies a sigmoid function to calculate its output m,

$$m(t+1) = \tanh\left(\sum_{k=1}^{n}\sum_{j=1} w_j^{(k)} m_j^{(k)}(t)\right), \tag{2.14}$$

where the sum over j runs over all the neurons that feed into gate k. This value m is then propagated to every node this NG is connected to.

To apply a genetic algorithm to this system, each ANN is encoded in a genome as follows. A start codon of two loci marks the beginning of an NG, the subsequent two loci encode the NG's number of inputs and outputs, and two further loci specify the origin of the inputs (which

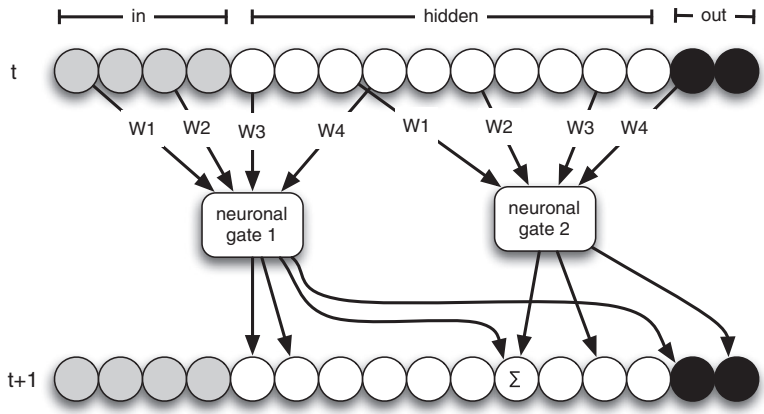

Figure 3: Open-topology artificial neural network, with 4 input neurons (light gray), 10 hidden neurons (white), and 2 motor neurons (dark gray). The nodes are connected via two neuronal gates (NG). Each NG connects 4 arbitrary input nodes with weight W to 4 output nodes. This figure illustrates how the nodes become updated from time point t to $t + 1$. When two NGs write into the same node, their outputs are added (indicated by $\sum$) before the sigmoid function is applied.

neurons feed into the gate) and the outputs (where the NG writes into). This information is then followed by an encoding of the n weights of an n-input NG (see Figure 3). The number of gates in the network can change as it evolves and is determined only by the number of start codons in the genome. The genomes encoding these ANNs can undergo the same mutational changes as described later in the MB section. In this respect, evolving open-topology ANNs is similar to using evolutionary algorithms (such as NEAT; see Stanley & Miikkulainen, 2002) to evolve neural networks with augmented topology.

2.3.2 Markov Brains. In our second experiment, the agent is controlled by a network of 16 nodes (4 input, 2 output, and 10 internal nodes, with the same types and number of nodes as the ANNs), which are connected via hidden Markov gates (HMGs; see Edlund et al., 2011). Networks of HMGs (Markov brains, MBs) are a type of stochastic Markov network (Koller & Friedman, 2009) and are related to the hierarchical temporal memory model of neocortical function (Hawkins & Blakeslee, 2004; George & Hawkins, 2005, 2009) and the HMAX algorithm (Riesenhuber & Poggio, 1999), except that Markov brains need not be organized in a strictly hierarchical manner as their connectivity is evolved rather than designed top down.

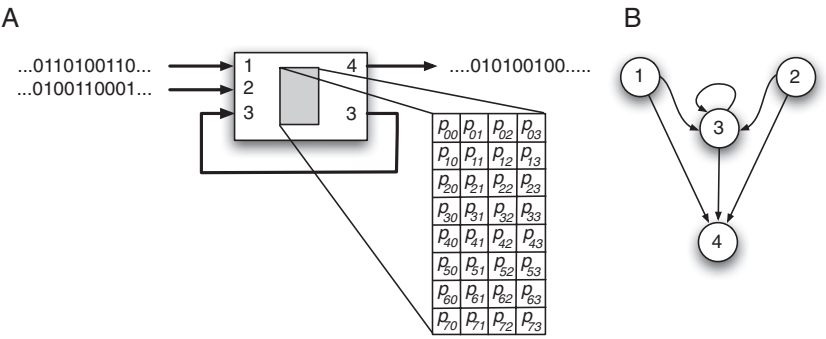

Figure 4: (A) A single HMG with three inputs and two outputs reads from nodes 1, 2, and 3 and writes to nodes 3 and 4, updating the states of these nodes in the process. (B) The output states are determined by a set of 2^{3+2} probabilities, here denoted as p_{xy}, where x and y are the decimal equivalent of the binary pattern of the input and output, respectively. For example, p_{73} is the probability for the pattern 11 to fire if the input was 111, that is, $P(11|111) = p_{73}$.

Each HMG can be understood as a finite state machine that is defined by its input-output structure (see Figure 4A) and a state transition table (see Figure 4B). All nodes in Markov brains are binary, and in principle the HMGs are stochastic, that is, the output nodes fire (i.e., are set to state 1) with a probability determined by the state-to-state transition table. Here, each HMG can receive up to four inputs and distribute signals to up to four nodes, with a minimum of one input and one output node (these settings are configurable). For the evolution of the ACP task, we consider only deterministic HMGs (each row of the transition table contains only one value of 1.0, and all other transitions have a probability of 0.0), turning our HMGs into classical logic gates. In order to apply an evolutionary algorithm, each HMG is encoded in a similar way as the NGs using a genome that specifies the network as a whole. Each locus of the genome is an integer variable $\in [0, 255]$. Following a start codon (marking the beginning of a gene, where each gene encodes a single HMG), the next two loci encode the number of inputs and outputs of the gate, respectively, followed by a specification of the origin of the inputs and the identity of the nodes being written to.

For example, for the HMG depicted in Figure 4, the loci following the start codon would specify "3 inputs", "2 outputs", "read from 1,2,3", "write to 3,4." This information is then followed by an encoding of the 2^{n+m} probabilities of an n-input and m-output state transition table (see supplementary Figure S1 in Edlund et al., 2011, for more details). For the example given in Figure 4, the particular HMG is specified by a circular genome with 39 loci (not counting the start). The start codon is universally (but arbitrarily)

chosen as the consecutive loci (42,213). Because this combination occurs only by chance once every 65,536 pairs of loci (making start codons rare), we insert four start codons at arbitrary positions into a 5000 loci initial genome to jump-start evolution. Thus, the ancestral genomes of all experiments with Markov brains encode at least 4 HMGs. A set of HMGs encoded in this manner uniquely specifies the Markov brain. The encoding is robust in the sense that mutations that change the input-output structure of an HMG leave the probability table intact, while either adding or removing parts of the table. This flexibility also implies that there is considerable neutrality in the genome, as each gene has 256 loci reserved for the probability table even if many fewer loci are used.

MBs and ANNs differ with respect to the gates connecting the nodes in each network. ANNs use weights, sums, and a tanh function, together with continuous variables, to compute their actions. In contrast, MBs use discrete states and Boolean logic to perform their computation. Using a very similar encoding of the topology means that mutations will have a similar effect on the topology of both systems but different effects on the computations each system performs.

2.4 Evolutionary Algorithm. We evolve the two types of networks (ANNs and MBs) using a genetic algorithm (GA). A GA can find solutions to problems by using evolutionary search (see Michalewicz, 1996). The GA operates on the specific genetic encoding of the networks' structure (the genotype) by iterating through a cycle of assessing each network's fitness in a population of 100 candidates, selecting the successful ones for differential replication, and finally mutating the new candidate pool. When testing a network's performance in controlling the agent, each network is faced with all 80 possible initial conditions that the world can take on. The fitness w is calculated as the fraction of successful actions (the number of large blocks avoided plus the number of small blocks caught) out of 80 tests (a number between 0 and 1). For the purpose of selection, we use an exponential fitness measure that multiplies the score by a factor 1.1 for every successful action but divides the score by 1.1 for every unsuccessful action, or $S = 1.1^{80(2w-1)}$. After the fitness assessment, the genotypes are ranked according to S and placed into the next generation with a probability that is proportional to the fitness (roulette wheel selection without elite). After replication, genotypes are mutated. We implemented three different mutational mechanisms that all occur after replication, with different probabilities. A point mutation happens with a probability of $\mu = 0.005$ per locus and causes the value at that locus to be replaced by a uniform random number drawn from the interval $[0, \ldots, 255]$. There is a 2% chance that we delete a sequence of adjacent loci ranging from 256 to 512 in size and a 5% chance that a stretch of 256 to 512 adjacent loci is duplicated (the size of the sequence to be deleted or duplicated is unformly distributed in the range given). The duplicated stretch is randomly inserted between any two loci in

the genome. Duplications and deletions are contrained so that the genome is not allowed to shrink below 1024 sites and genomes cannot grow beyond 20,000 sites. Because insertions are more likely than deletions, there is a tendency for genomes to grow in size during evolution.

We evolve networks through 10,000 generations, and run 200 replicates of each experiment. Note that the type of gates is different between ANNs (neuronal) and MBs (logic), so the rate of evolution of the two networks cannot be compared directly because mutations will have vastly different effects with respect to the function of the gates. Thus, the optimal mutation rate differs among networks (Orr, 2000). At the end of each evolutionary run, we reconstruct the evolutionary line of descent (Lenski, Ofria, Pennock, & Adami, 2003) of the experiment by following the lineage of the most successful agent at the end of 10,000 generations backward all the way to the random ancestor that was used to seed the experiment. This is possible because we do not use crossover between genotypes in our GA. This line of descent, given by a temporally ordered sequence of genotypes, recapitulates the unfolding of the evolutionary process, mutation by mutation, from ancestor to the evolved agent with high fitness, and captures the essence of that particular evolutionary history. For each of the organisms on each of the 200 lines of descent of any particular experiment, we calculate a number of information-theoretic quantities, among which is how much of the world the agent represents in its brain, using equation 2.5.

2.5 Extracting Probabilities from Behavior. The inputs to the information-theoretic measure of representation R are the probabilities of observing a particular state x, $p(x)$, as well as the joint probabilities $p(x,y)$ describing the probability of observing a state x when at the same time another variable Y takes on the state $Y = y$. For the representation R defined by equation 2.5, sensor, internal, and environment variables are distinguished. For any particular organism (an agent that performs the ACP task with an evolved controller), R is measured at any point during the evolution by placing the organism into the simulated world and concurrently recording time series data of the states of all 16 controller nodes and the states of the environment. The recordings are then used to calculate the frequency of states. Based on the frequencies of states, all probabilities relevant for the information-theoretical quantities can be calculated, including those that take into account the temporal order of events (e.g., the probability $p(x_t, y_{t+1})$ that variable X_t takes on state x_t while variable Y_{t+1} takes on the state y_{t+1}). If a particular state (or combination of states) never occurs, a probability of zero is recorded for that entry (even though in principle the state or combination of states could occur). R (and the other information-theoretic quantities introduced in section 3) is calculated for organisms on the evolutionary line of descent, making it possible to follow the evolutionary trajectory of R from random ancestor to adapted agent. For ANNs that have internal nodes with continuous rather than binary states, a

mapping of intervals $[-1, 0] \to 0$ and $]0, 1] \to 1$ is applied before calculation of probabilities.

3 Results

To establish a baseline, 100,000 random controllers for each of the two network types were created, and the distributions of R and fitness values were obtained. This baseline served two purposes: it shows how well randomly generated (unevolved) networks perform and how much information about the world they represent by chance, as well as providing information about the distribution of these values. Random ANNs and MBs were created in the same way by randomly drawing values from a uniform probability distribution of the integers $\in [0, 255]$ for each of the genome's loci. Each genome was then sprinkled with four start codons at arbitrary positions within the genome.

Figure 5A shows the distribution of fitness scores w for 100,000 random ANNs, and Figure 5B shows the distribution of their representation R. The respective distributions for MBs are shown in Figure 5C for fitness scores and in Figure 5D for representation. While both systems use different types of gates, ANNs and MBs do not differ with respect to their initial fitness or representation distributions. This shows that random genomes with a high fitness are—as expected—very rare and need to evolve their functionality in order to perform optimally.

In order to compare the two network architectures, the evolutionary trajectories for fitness and representation were analyzed for the evolutionary line of descent (LOD) as described in section 2.4. The different LODs obtained from following back any other member of the final population quickly coalesce to a single line. Hence, the LOD effectively recapitulates the genetic changes that led from random networks to proficient ones. The development of fitness and representation over evolutionary time in Figure 6 is averaged over 200 independent replicates. While both ANNs and MBs have on average low fitness at the beginning of an evolutionary run (as seen in Figure 5), MBs become significantly more fit than ANNs, and we find that after 10,000 generations, 18 out of 200 runs MBs have evolved to perfect fitness, while none of the ANNs reached this level (the best ANNs correctly make 77 out of 80 decisions). At the same time, we see that the fitness of the ANNs after 10,000 generations is not stagnating, which suggests that more run time will allow further improvement. As previously mentioned, the rate at which fitness is achieved in evolutionary time cannot be compared across architectures because mutations affect the function of the gates differently. While we tentatively explain the difference in performance between ANNs and MBs by their difference in representing the world below, we anticipate that the different network architectures also solve the categorization task very differently. To understand the information dynamics and

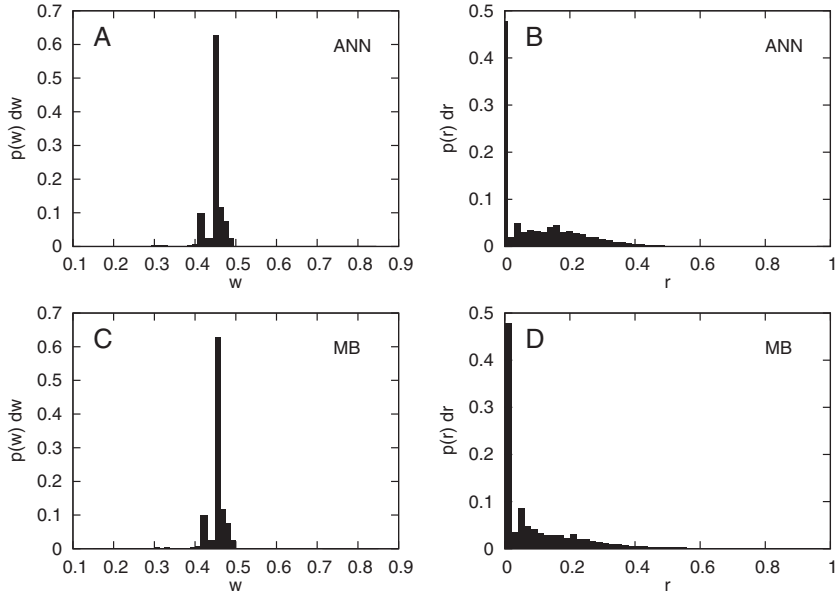

Figure 5: Probability distribution of fitnesses and representation scores of random machines. (A) Probability distribution of fitnesses (fraction of successful actions) $p(w)dw = P(w < W < w + dw)$ for 100,000 random ANNs ($dw = 1/80$) (B) Probability distribution of the representation variable R for the same random ANNs $p(r)dr = P(r < R < r + dr)$, with $dr = 0.02$. (C) Distribution of fitnesses for 100,000 random Markov brains ($dw = 1/80$). (D) Distribution of representation R in the same MB networks ($dr = 0.02$).

the strategies employed in more detail, we measured a number of other information-theoretic measures (besides R).

The evolutionary trajectory for representation R (see Figure 6B) is similar to the evolution of fitness (see Figure 6A), but MBs evolve to a significantly higher value of R. We attribute this difference to the difference in fitness between the two types of networks, as the discretization of the continuous ANN variables cannot introduce a bias in R. Thus, it appears that an increased representation of the world within an agent's network controller correlates with fitness. We can test this correlation between fitness and representation at the end of a run for the 200 replicates of MBs and ANNs and find that fitness and R are significantly correlated (Spearman's $r = 0.55$, $P = 2.5 \times 10^{-17}$) for MBs but not for ANNs ($r = -0.10$, $P = 0.15$). We speculate that because ANNs are forced to compute using a sigmoid function only (effectively implementing a multiple-AND gate) while MBs can use arbitrary logic operations to process data, ANNs struggle to internalize (i.e., represent) environmental states. In other words, it appears that the ease of

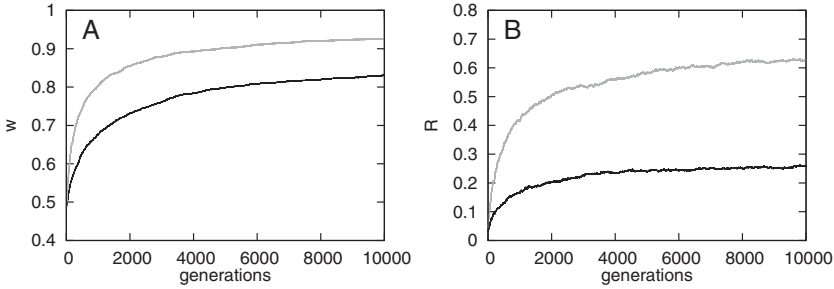

Figure 6: (A) Fitness w and (B) representation R (in bits) along the line of descent as a function of evolutionary generations, averaged over 200 independent evolutionary lines, for evolved networks (ANNs: black; MBs: gray).

memory formation is crucial in forming representations, which are then efficiently transformed into fit decisions in MBs (Edlund et al., 2011).

3.1 Analysis of Network Structures and Strategies. In order to be successful at the task described, an agent has to perform active categorical perception followed by prediction. In the implementation of the ACP task by Beer (1996, 2003), prediction can be achieved without memory, because once the network has entered the attractor representing a category, the prediction (to move away or to stay) can be directly coupled to the attractor. The task used here can be achieved only using memory (data not shown) and requires the agent to perform categorical perception by comparing sensory inputs from at least two different time points (which also allows a prediction of where the object is going to land).

In order to analyze how information is processed, we calculated the predictive information (Bialek, Nemenman, & Tishby, 2001) of the evolved networks, given by the mutual Shannon information between the network's inputs at time t and its outputs at time $t+1$. Predictive information, defined this way (Ay, Bertschinger, Der, Guettler, & Olbrich, 2008), measures how much of the entropy of outputs (the firings of motor neurons that control the agent) can be understood in terms of the signals that have appeared in the agent's sensors just prior to the action. Indirectly, predictive information therefore also indicates the contribution of the hidden nodes of the network. A high predictive information would show that the hidden nodes do not contribute much and that computations are performed mainly by input and output neurons. Using the variable S for sensor states and A for actuator states, the predictive information can be written in terms of the shared entropy between sensor states at time t and motor states at time $t+1$ as

$$I_{\text{pred}} = I(S_t : A_{t+1}) = -\sum_{s_t, a_{t+1}} p(s_t, a_{t+1}) \log \frac{p(s_t, a_{t+1})}{p(s_t)\, p(a_{t+1})}\,, \tag{3.1}$$

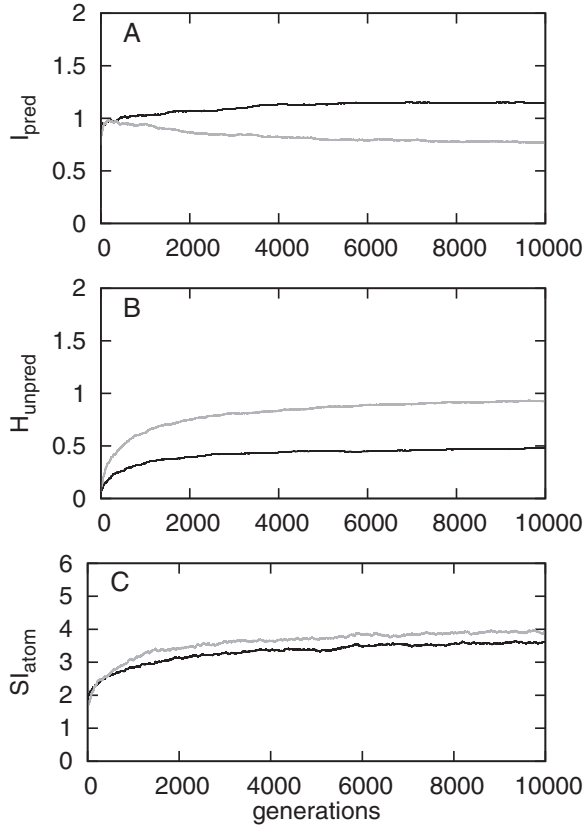

Figure 7: Different measures of information processing and integration along the LOD for both types of network architectures: ANNs (black) and Markov brains (gray). (A) Predictive information, equation 3.1. (B) Unpredicted entropy, equation 3.2 of the network's motor variables. (C) Information integration SI_{atom} based on equation 3.3.

where $p(s_t) = P(S_t = s_t)$ is the probability of observing variable S_t in state s_t, $p(a_{t+1}) = P(A_{t+1} = a_{t+1})$ is the probability observe variable A_{t+1} in state a_{t+1}, and so on. Note that S_t and A_{t+1} are joint random variables created from the variables of each node, implying that S_t can take on 16 different states while A_{t+1} can take on 4 possible states. The probabilities are extracted from time series data as described in section 2.5. Figure 7A shows that over the course of evolution, the predictive information I_{pred} decreases for MBs after an initial increase but increases slightly overall for ANNs. The drop in predictive information for the MBs indicates that their actions become less dependent on sensor inputs and are driven more by the hidden neurons, while the ANNs' actions remain to be predictable by sensor inputs.

To test whether it is the internal states that increasingly guide the agent, the predictive information was subtracted from the entropy of the output states (maximally two bits) to calculate the unpredicted entropy of the outputs, that is, how much of the motor outputs are uncorrelated to signals from the input:

$$H_{\text{unpred}} = H(A_{t+1}) - I_{\text{pred}} = H(A_{t+1}|S_t) \ . \tag{3.2}$$

Figure 7B shows that H_{unpred} increases over the course of evolution, suggesting that indeed signals other than the sensor readings are guiding the motors. In principle, this increase could be due to an increase in the motor neuron entropy; however, as the latter stays fairly constant, we can conclude that the more a network adapts to its environment, the less its outputs are determined by its inputs and the more by its internal states. Again, this effect is stronger for MBs than for ANNs and suggests that it is indeed the internal states that encode representations that drive the network's behavior. It is also possible that the motors evolve to react to sensor signals further back in time. Because sensor neurons cannot store information, such a delayed response also has to be processed by internal states. While the absolute value of the predicted information and unpredicted entropy can depend on this time delay, we expect the overall trend of a decreasing I_{pred} coupled with an increasing H_{unpred} to be the same as for the one-step predictive information, because the sensorial signal stream itself has temporal correlations, that is, it is nonrandom.

To quantify the synergy of the network, we calculated a measure of information integration called *synergistic information*. Roughly speaking, synergistic information measures the amount of information that is processed by the network as a whole that cannot be understood in terms of the information processing of each individual node; that is, it measures the extent to which the whole network is—informationally—more than the sum of its parts (Edlund et al., 2011):

$$SI_{\text{atom}} = I(X_t : X_{t+1}) - \sum_{i=1}^{n} I(X_t^i : X_{t+1}^i) \ . \tag{3.3}$$

In equation 3.3, $I(X_t : X_{t+1})$ measures the amount of information that is processed (across time) by the whole network X (the joint random variable composed of each of the node variables), whereas $I(X_t^i : X_{t+1}^i)$ measures how much is processed by node i. The negative of equation 3.3 has been used before, to quantify the redundancy of information processing in a neural network (Atick, 1992; Nadal & Parga, 1994; Schneidman et al., 2003). SI_{atom} is a special case of the information integration measure Φ (Tononi, 2008; Balduzzi & Tononi, 2008), which is computationally far more complex than the SI_{atom} because it relies on computing information integration across all possible partitions of a network. SI_{atom} instead calculates information

integration across the "atomic" partition only, that is, the partition where each node is its own part. Figure 7C shows that the SI_{atom} increases for Markov brains as well as for ANNs, which indicates that both architectures evolve the ability to integrate information to perform the task at hand. We see only a marginal difference between MBs and ANNs in their ability to integrate information, while at the same time MBs are more dependent on internal states and ultimately perform better. This suggests that measuring integrated information in terms of equation 3.3 does not allow inferences about a system's capacity to memorize. In summary, we observe that MBs evolve to become less dependent on sensorial inputs than ANNs; in addition, the actions of the MBs become more dependent on internal states than in ANNs. Thus, we conclude that the network properties that R measures are indeed representations that the networks create as an adaptive strategy. But how do the networks represent their environments? Which features of the world are represented and form the successful task-solving strategies?

3.2 Epistemically Opaque Strategies. In order to analyze a Markov brain's function, a number of different tools are available. First, a causal diagram can be generated by drawing an edge between any two neurons that are connected via an HMG. The edges are directed, but each edge can in principle perform a different computation. When creating the causal diagram, nodes that are never written into by any other nodes are removed, as they are computationally inert (they remain in their default off state). Such nodes can also be identified by a knockout procedure, where the input of each node is forced to either the 0 or 1 state individually. If such a procedure has no effect on fitness, the node is inert.

Figure 8 shows the causal diagram of an evolved Markov brain that solves the classification task perfectly. This network uses inputs from the sensors, motors, and memory simultaneously for decisions, fusing the different modalities intelligently (Murphy, 1996).

The causal diagram by itself, however, does not reveal how function is achieved in this network. As each HMG in the instantiation represents a deterministic logic gate (generally, they are stochastic), it is possible to determine the logical rules by which the network transitions from state to state by feeding the state-to-state transition table into a logical analyzer (Rickmann, 2011). The analyzer converts the state transition table into the minimal description of functions in Boolean logic using only NOT, AND ($\wedge$), and OR ($\vee$). With these functions we can exactly describe each node's logical influence on other nodes (and possibly itself). For the network depicted in Figure 8, the logic is given by (here, the numeral represents the node, and its index the state at time $t0$ or the subsequent time point $t1$, while an overbar stands for NOT):

$$4_{t1} = (\overline{2}_{t0} \wedge \overline{5}_{t0}) \vee (2_{t0} \wedge 6_{t0}),$$

$$5_{t1} = (\overline{2}_{t0} \wedge 10_{t0} \wedge \overline{12}_{t0}) \vee (\overline{10}_{t0} \wedge \overline{12}_{t0} \wedge \overline{14}_{t0}) \vee (\overline{2}_{t0} \wedge 10_{t0} \wedge 12_{t0} \wedge \overline{14}_{t0}),$$

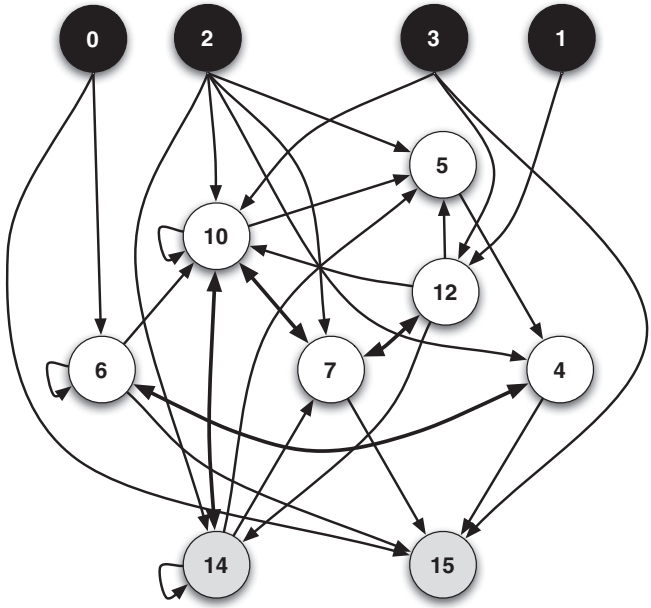

Figure 8: Causal diagram of a Markov brain with perfect fitness, correctly catching all small blocks and avoiding all large ones. Nodes colored in black are sensors; motor variables are gray. Double arrows represent two causal connections—one each way. Nodes with arrows that point to themselves write their output back into their input and may work as memory (all nodes return to a default 0 if not set otherwise by each update), so state information can be maintained only via such self-connections). Internal nodes can read from motors, giving rise to proprioception or, more precisely, kinesthesia: the ability to sense one's own motion. The motors themselves can be used as memory.

$$
\begin{aligned}
6_{t1} &= \overline{4}_{t0} \wedge (0_{t0} \vee 6_{t0}),\\
7_{t1} &= (2_{t0} \wedge \overline{12}_{t0}) \vee (\overline{10}_{t0} \wedge 12_{t0} \wedge 14_{t0}),\\
10_{t1} &= (2_{t0} \wedge 6_{t0}) \vee (\overline{2}_{t0} \wedge 10_{t0} \wedge \overline{14}_{t0}) \vee (10_{t0} \wedge 12_{t0} \wedge \overline{14}_{t0}) \vee (2_{t0} \wedge 14_{t0}),\\
&\quad \vee (3_{t0} \wedge \overline{7}_{t0} \wedge \overline{10}_{t0} \wedge 12_{t0} \wedge 14_{t0}),\\
12_{t1} &= (\overline{1}_{t0} \wedge \overline{7}_{t0}) \vee (\overline{3}_{t0} \wedge 7_{t0}) \vee (3_{t0} \wedge \overline{7}_{t0}),\\
14_{t1} &= (2_{t0} \wedge 10_{t0} \wedge 14_{t0}) \vee (2_{t0} \wedge \overline{10}_{t0} \wedge \overline{12}_{t0}) \vee (\overline{2}_{t0} \wedge 10_{t0} \wedge 12_{t0}),\\
15_{t1} &= (3_{t0} \wedge \overline{7}_{t0}) \vee (\overline{0}_{t0} \wedge 6_{t0}) \vee (4_{t0} \wedge 6_{t0}) \vee (0_{t0} \wedge \overline{4}_{t0} \wedge \overline{6}_{t0}).
\end{aligned}
$$

While this logical representation of the network's dynamics is optimized (and the contribution of inert nodes is removed), it is in general not possible

to determine the minimal logic network based on state-to-state transition information only, as finding the minimal logic is believed to be a computationally intractable problem (Kabanets & Cai, 2000). As a consequence, while it is possible to capture the network's function in terms of a set of logical rules, we should not be surprised that evolution delivers epistemically opaque designs (Humphreys, 2009), that is, designs that we do not understand on a fundamental level. However, the strength of measuring representations with R is that our measure is able to capture representations even if they are highly distributed and take part in complex computations. As such, R provides a valuable tool to analyze evolved neural networks, as can be seen below.

3.3 Concepts and Memory. To understand what representations are acquired (representations about which concepts), we calculated R for each property of the environment defined in equations 2.10 to 2.13, within each of the key nodes in our example network shown in Figure 8. For this network, Figure 9 shows that some nodes prefer to represent given single features or concepts, while others represent several features at the same time. In addition, the degree to which a node represents a certain property changes during the course of evolution. Looking at representation within each individual node, however, tells only part of the story as it is clear that representations are generally "smeared" over several nodes. If this is the case, a pair of nodes (for example) can represent more about a feature than the sum of the representations in each node (i.e., variables can represent synergistically). In order to discover which combination of nodes represents which feature most accurately, a search over all partitions of the network would have to be performed, much as in the search for the partition with minimum information processing in the calculation of a network's synergistic information processing (Balduzzi & Tononi, 2008).

One can also ask whether brain states represent the environment as it is at the time it is being represented, or whether it represents the environment in its past state. In other words, we can ask whether representations are about more distant or more proximal events. To answer this, we define temporal representations by including the temporal index of the Markov variables. For example, a representation at the same time point t is defined (as implicit in equation 2.5) as

$$R_t = H(E_t : M_t | S_t), \tag{3.4}$$

while a representation of events one update prior is defined as

$$R_{t-1} = H(E_{t-1} : M_t | S_t), \tag{3.5}$$

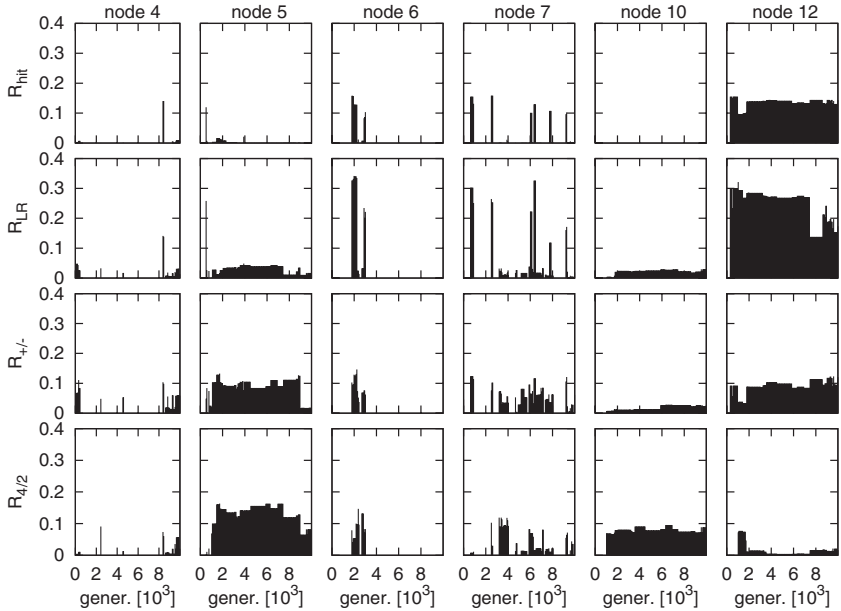

Figure 9: Representation of each of the four environmental properties (concepts) defined in equations 2.10 to 2.13. as a function of time, within each of the nodes of a network that evolved to become the one depicted in Figure 8. Representation is measured in bits, along the temporal (genetic) line of descent (measured in generations).

that is, the shared entropy between the internal variables at time t and the environmental states at time $t-1$, given the sensor's states at time t. Naturally one can define temporal representations about more distant events in the same manner. R_t, R_{t-1}, and R_{t-2} were calculated and averaged over all 80 experiments in each generation (for both ANNs and MBs) over the course of evolution. Figure 10 shows that in both systems, R_{t-1} is larger than R_t, and R_{t-2} is larger still (but note that R_{t-3} is smaller; data not shown), and all values increase over evolutionary time similar to R (see Figure 6B). This suggests that both networks evolve a form of memory that reaches further back than just one update.

We suggest that the peak in representation at a time difference of two updates implied by Figure 10 can be explained by the hierarchical structure of the networks, which have to process the sensorial information through at least two time steps to reach a decision (it takes at least two time steps in order to assess the direction of motion of the block). Decisions have to be made shortly thereafter, however, in order to move the agent to the correct location in time. This further strengthens our view that representations are

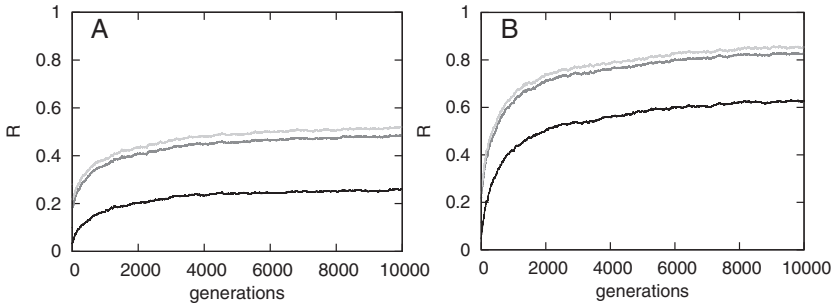

Figure 10: Representation as a function of evolutionary time for three different time intervals. (A) Representation in ANNs (black: R_t; dark gray: R_{t-1}; light gray: R_{t-2}. (B) Representation in Markov brains (shading as in panel A). The black curves are the same as in Figure 6B, and are shown here for the purpose of comparison.

evolved and, furthermore, that they build up during an agent's lifetime as memories of past events shape the agent's decisions.

4 Conclusion

We defined a quantitative measure of representation R in terms of information theory as the shared entropy between the states of the environment and internal "brain" states, given the states of the sensors. While internal states are necessary in order to encode internal models, not all internal states are representations. Indeed, representational information (which is about the environment) is a subset of the information stored in internal states. Information about the state of other past internal states or substates does not count toward R, and neither would information about imagined worlds, for example. Testing which nodes of the system contain representations about what aspect of the environment helps us to distinguish between information present in internal states and information that is specifically used as a representation. We applied this measure to two types of networks that were evolved to control a simulated agent in an active categorical peception task. Our experiments showed that the achieved R increases with fitness during evolution independent of the system used. We also showed that while the (algorithmic) function of both artificial neural networks and Markov networks is difficult to understand, deterministic Markov networks can be reduced to sets of Boolean logic functions. This logic, however, may be epistemically opaque. While representation R increases in networks over evolutionary time, each neuron can represent parts of individual concepts (features of the environment). However, most often concepts are distributed over several neurons and represent synergistically. In addition,

representations also form over the lifetime of the agent, increasing as the agent integrates information about the different concepts to reach a decision. Thus, what evolves in Markov and artificial neural networks via Darwinian processes are not the representations themselves; rather, what evolves is the capacity to represent the environment, while the representations themselves are formed as the agent observes and interacts with the environment. We argued that R can be measured in continuous (artificial neural networks) as well as discrete systems (Markov networks), which suggests that this measure can be used in more complex and more natural systems. We found that in the implementation used here, Markov networks were able to evolve their ability to form representations more easily than artificial neural networks. Future investigations will show what kind of system is more powerful to make intelligent decisions using representations.

Acknowledgments

We thank C. Koch and G. Tononi for extensive discussions about representations, information integration, and qualia. This research was supported in part by the German Federal Ministry of Education and Research, the Paul G. Allen Family Foundation, the National Science Foundation's Frontiers in Integrative Biological Research grant FIBR-0527023, NSF's BEACON Center for the Study of Evolution in Action under contract no. DBI-0939454, and the Agriculture and Food Research Initiative Competitive grant 2010-65205-20361 from the USDA National Institute of Food and Agriculture. We acknowledge the support of the Berlin-Brandenburg Academy of Sciences, the Michigan State University High Performance Computing Center, and the Institute for Cyber Enabled Research.

References

Atick, J. J. (1992). Could information theory provide an ecological theory of sensory processing. *Network*, *3*(2), 213–251.

Ay, N., Bertschinger, N., Der, R., Guettler, F., & Olbrich, E. (2008). Predictive information and explorative behavior of autonomous robots. *Eur. Phys. J. B.*, *63*(3), 329–339.

Balduzzi, D., & Tononi, G. (2008). Integrated information in discrete dynamical systems: Motivation and theoretical framework. *PLoS Comput Biol*, *4*(6), e1000091.

Beer, R. (1996). Toward the evolution of dynamical neural networks for minimally cognitive behavior. In P. Maes, M. Mataric, J. Meyer, J. Pollack, & S. Wilson (Eds.), *From animals to animats: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior* (pp. 421–429). Cambridge, MA: MIT Press.

Beer, R. (2003). The dynamics of active categorical perception in an evolved model agent. *Adaptive Behavior*, *11*, 209–243.

Bialek, W., Nemenman, I., & Tishby, N. (2001). Predictability, complexity and learning. *Neural Computation*, *13*(11), 2409–2463.

Bongard, J., Zykov, V., & Lipson, H. (2006). Resilient machines through continuous self-modeling. *Science*, *314*(5802), 1118–1121.

Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence*, *47*, 139–159.

Chomsky, N. (1965). *Aspects of the theory of syntax*. Cambridge, MA: MIT Press.

Clark, A. (1997). The dynamical challenge. *Cognitive Science*, *21*, 461–481.

Clark, A., & Toribio, J. (1994). Doing without representing? *Synthese*, *101*, 401–431.

Cohen, H., & Lefebvre, C. (Eds.). (2005). *Handbook of categorization in cognitive science*. Amsterdam: Elsevier.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. New York: Wiley.

Craik, K. (1943). *The nature of explanation*. Cambridge: Cambridge University Press.

Edlund, J. A., Chaumont, N., Hintze, A., Koch, C., Tononi, G., & Adami, C. (2011). Integrated information increases with fitness in the simulated evolution of autonomous agents. *PLoS Comput Biol*, *7*, e1002236.

Floreano, D., & Mondada, F. (1996). Evolution of homing navigation in a real mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *26*(3), 396–407.

Fodor, J. (1975). *The language of thought*. New York: Crowell.

George, D., & Hawkins, J. (2005). A hierarchical Bayesian model of invariant pattern recognition in the visual cortex. In D. Prokhorov (Ed.), *Proceedings of the International Joint Conference on Neural Networks* (Vol. 3, pp. 1812–1817). Piscataway, NJ: IEEE.

George, D., & Hawkins, J. (2009). Towards a mathematical theory of cortical microcircuits. *PLoS Comput Biol*, *5*(10), e1000532.

Harnad, S. (Ed.). (1987). *Categorical perception: The groundwork of cognition*. Cambridge: Cambridge University Press.

Haugeland, J. (1991). Representational genera. In W. Ramsey, S. P. Stich, and D. E. Rumelhart (Eds.), *Philosophy and connectionist theory* (pp. 61–89). Hillsdale, NJ: Erlbaum.

Hawkins, J., & Blakeslee, S. (2004). *On intelligence*. New York: H. H.

Humphreys, P. (2009). The philosophical novelty of computer simulation methods. *Synthese*, *169*(3), 615–626.

Johnson-Laird, P., & Wason, P. (1977). *Thinking: Readings in cognitive science*. Cambridge: Cambridge University Press.

Kabanets, V., & Cai, J.-Y. (2000). Circuit minimization problem. In F. Yao, & E. Luks (Eds.), *Proc. 32nd Symposium on Theory of Computing* (pp. 73–79). New York: ACM.

Kawato, M. (1999). Internal models for motor control and trajectory planning. *Current Opinion in Neurobiology*, *9*(6), 718–727.

Kay, J. W., & Phillips, W. A. (2011). Coherent infomax as a computational goal for neural systems. *Bull. Math. Biol.*, *73*(2), 344–372.

Koller, D., & Friedman, N. (2009). *Probabilistic graphical models*. Cambridge, MA: MIT Press.

Lenski, R. E., Ofria, C., Pennock, R. T., & Adami, C. (2003). The evolutionary origin of complex features. *Nature*, *423*(6936), 139–144.

Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. New York: Holt.

Marstaller, L., Hintze, A., & Adami, C. (2010). Measuring representation. In W. Christensen, E. Schier, & J. Sutton (Eds.), *ASCS09: Proceedings of the 9th Conference of the Australasian Society for Cognitive Science* (pp. 232–237). Sydney: Macquarie Centre for Cognitive Science.

McGill, W. J. (1954). Multivariate information transmission. *Psychometrika*, *19*(2), 97–116.

Michalewicz, Z. (1996). *Genetic algorithms + data strucures = Evolution programs*. New York: Springer-Verlag.

Moravec, H. P. (1984). Locomotion, vision & intelligence. In M. Brady and R. Paul (Eds.), *Robotics research 1* (pp. 215–224). Cambridge, MA: MIT Press.

Murphy, R. (1996). Biological and cognitive foundations of intelligent sensor fusion. *IEEE Transactions on Systems Man and Cybernetics Part A—Systems and Humans*, *26*(1), 42–51.

Nadal, J. P., & Parga, N. (1994). Nonlinear neurons in the low noise limit: A factorial code maximizes information transfer. *Network*, *5*(4), 565–581.

Newell, A., & Simon, H. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice Hall.

Nolfi, S. (2002). Power and the limits of reactive agents. *Neurocomputing*, *42*(1–4), 119–145.

Nolfi, S., & Floreano, D. (2000). *Evolutionary robotics*. Cambridge, MA: MIT Press.

Orr, H. (2000). The rate of adaptation in asexuals. *Genetics*, *155*(2), 961–968.

Phillipps, W. A., Kay, J., & Smyth, D. M. (1994). How local cortical processors that maximize coherent variation could lay foundations for representation proper. In L. S. Smith, & P. J. B. Hancock (Eds.), *Neural computation and psychology* (pp. 117–136). New York: Springer-Verlag.

Phillips, W., & Singer, W. (1997). In search of common foundations for cortical computation. *Behavioral and Brain Sciences*, *20*(4), 657–722.

Pinker, S. (1989). *Learnability and cognition*. Cambridge, MA: MIT Press.

Pitt, D. (2008). Mental representation. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy*. Stanford, CA: Metaphysics Research Lab, Stanford University.

Rickmann, S. (2011). Logic friday (version 1.1.3) [computer software]. http://www.sontrak.com/downloads.html

Riesenhuber, M., & Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature Neuroscience*, *2*(11), 1019–1025.

Schneidman, E., Still, S., Berry II, M. J., & Bialek, W. (2003). Network information and connected correlations. *Phys. Rev. Lett.*, *91*(23), 238701.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal, 27*, *379–423*, 623–656.

Stanley, K. O., & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evol. Comput.*, *10*(2), 99–127.

Tononi, G. (2008). Consciousness as integrated information: A provisional manifesto. *Biol. Bull.*, *215*(3), 216–242.

van Dartel, M. F. (2005). *Situated representation*. Unpublished doctoral dissertation, Maastricht University.

van Dartel, M., Sprinkhuizen-Kuyper, I., Postma, E., & van den Herik, J. (2005). Reactive agents and perceptual ambiguity. *Adaptive Behavior*, *13*, 227–242.

Ward, R., & Ward, R. (2009). Representation in dynamical agents. *Neural Networks*, *22*, 258–266.

Watanabe, S. (1960). Information theoretical analysis of multivariate correlation. *IBM Journal of Research and Development*, *4*, 66–82.

Wolpert, D. M., Ghahramani, Z., & Jordan, M. I. (1995). An internal model for sensorimotor integration. *Science*, *269*(5232), 1880–1882.

Received June 27, 2012; accepted February 8, 2013.