# Cognitive systems evolve complex representations for adaptive behavior

**Lars Marstaller**[1,⋆], **Arend Hintze**[2,3,4,⋆] **& Christoph Adami**[2,3]

[1]Macquarie Centre for Cognitive Science, Macquarie University, Sydney, Australia

[2]Microbiology & Molecular Genetics, Michigan State University, East Lansing, MI

[3]BEACON Center for the Study of Evolution in Action, Michigan State University, East Lansing, MI

[4]Computer Science & Engineering, Michigan State University, East Lansing, MI *

**Keywords:** Information, Representation, Cognition, Evolution



**Abstract**

Representations are internal models of the world that provide context to a sensory stream and are formed over evolutionary time as well as learned. We argue here that representations are the expected consequence of an adaptive process, give a formal definition of representation based on information theory, and quantify it using our new measure $R$. To measure how $R$ changes over time, we evolve two types of networks—a recurrent 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 representations build up during the lifetime of these agents. We examine the concepts that are being represented, how they are logically encoded in the networks, and how they form as an agent behaves to solve a task. We conclude that any successful cognitive system that represents its environment within internal states should have a positive $R$.

## 1 Introduction

The notion of representation is as old as cognitive science itself (see, e.g., Chomsky, 1965; Newell and Simon, 1972; Fodor, 1975; Johnson-Laird and Wason, 1977; Marr,

*These authors contributed equally.

1982; Pinker, 1989; Pitt, 2008), but its usefulness for Artificial Intelligence (AI) research has been doubted (Brooks, 1991). Brooks, in his widely cited article "Intelligence without representation", argued instead for a reactive architecture where the components of the cognitive system directly interface with the world through perception and action, rather than with a central executive unit operating on a model of the world. He further argued that a path to AI could be forged by incrementally increasing the complexity of such a reactive architecture. In particular, inspired by the biological path to intelligence, Brooks argued that AI research needs to be rooted in robotics. 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).

However, 20 years after advocating such a radical departure from the common approach to AI, the representation-less 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 et al.,1995; Kawato,1999) are key to complex adaptive behavior. Indeed, while representation-free robotics has made some important strides (Nolfi, 2002), we argue here it can only take you so far. One of the main reasons for abandoning representations, in the forms of models of the world that are provided to the machines by their engineers, was that building accurate and appropriate models of the world simply proved to be impossible. An alternative approach is evolutionary robotics (Nolfi and Floreano, 2000), because Darwinian evolution is able to create appropriate sensory-action loops–and hence can give rise to complex representations of the world–that are not susceptible to intelligent design. As robots evolve to behave appropriately (and survive) in a dynamic and noisy world, representations of the world in the form of predictive models emerge within the cognitive apparatus, and are integrated with the perceived sensory data to make informed decisions–using not only the current state of the world but taking into account historical data as well. While it is in principle possible to operate successfully in a purely reactive manner to a complex and changing world, to do so would require an enormous input-bandwidth, as the *context* of every sensory state would have to be recorded too. Contextual guidance (via a contextual field) has been identified as one of the key elements of local cortical

processors (Phillipps et al., 1994; Phillips and Singer, 1997; Kay and Phillips, 2011) making neural systems reliable but also flexible. Evolved internal models provide this context, are flexible, and can be readjusted given new stimuli that can contradict the current model. They can handle even new sensory input elegantly, at least as long as there is sufficient overlap between the *anticipated* sensory signature (anticipated from the context) and the experienced one (Bongard et al., 2006).

Here, we propose a new way to *measure* how much an embodied agent represents about the world within its internal states by defining representation in information-theoretical terms, and show how the capacity to represent emerges in thousands of generations of simulated evolution. For our purposes, representations can be symbolic or sub-symbolic (e.g., neural states) as long as they have a physical basis, that is, as long as they are encoded in states and thus measurable. However, we distinguish representations from sensorial input: while sensor inputs are information about the world, they are not internal models: they cannot influence decisions in the absence of sensorial input or be based on counterfactual reasoning (as internal models can). We thus explicitly define representations as that part of information that an organism can use that goes beyond the information presented in its sensors (Haugeland, 1991; Clark, 1997). In particular, this implies that representation can, at time, *misrepresent* (Haugeland, 1991)–unlike information present in sensors, who always faithfully represent, albeit trivially. We show further how it is in principle possible to measure what this representation is *about*, and how these representations form during the lifetime of an agent. However, we also observe that often enough the evolved internal models are "epistemically opaque", that is, they are not easily understood by human engineers, and cannot easily be reduced into their components, that is, they may be irreducible. It is probably this quality of the evolved models that make them both successful and very difficult to engineer.

# 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 et al., 1994; Phillips and Singer, 1997; Kay

and Phillips, 2011). Here, we present an information-theoretic construction that explicitly takes the entropy of world states (environment) 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):

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

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 $H(X, Y)$:

$$I(X : Y) = H(X) + H(Y) - H(X, Y) = \sum_{xy} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \ . \tag{2}$$

In Eq. (2), $p(x)$ and $p(y)$ are the probability distributions for the random variables $X$ and $Y$ respectively [that is, $p(x) = P(X = x)$], 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

$$I(X : Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \ . \tag{3}$$

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 Eq. (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 to me, and is calculated as

$$H(X|Y) = -\sum_{xy} p(x, y) \log p(x|y) \ , \tag{4}$$

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 a representing and represented, and use mutual information to measure the correlation between a network's internal states and its environment [see also Marstaller et al. (2010)]. So, for example, we could imagine that $X$ stands for the states of a world or environment, whereas $Y$ is a variable that *represents* those states of the world. We need to be careful, however, to exclude from possible representational variables those that are mere images of the environment, 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, for example, understands representation as something that "stands in" for something in the environment, but that is no longer reflected in the perceptual system of the agent (Haugeland, 1991). Indeed, representation should be different from a mere translation of one structure into another, and requires instead a change in the structure's representational properties. 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 behavior. It simply stores what it received through its inputs without extracting information from its input. That is, the camera's internal states are fully determined by its sensor inputs.

To rule out such trivial representations, we define representation as the shared entropy between environment states and internal states, but *given* the sensor states, that is 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 Fig. 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 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):

$$R = H(E : M|S) = I(E : M) - I(E : M : S) = H_{\text{corr}} - I(S : E) - I(S : M) \ , \quad (5)$$

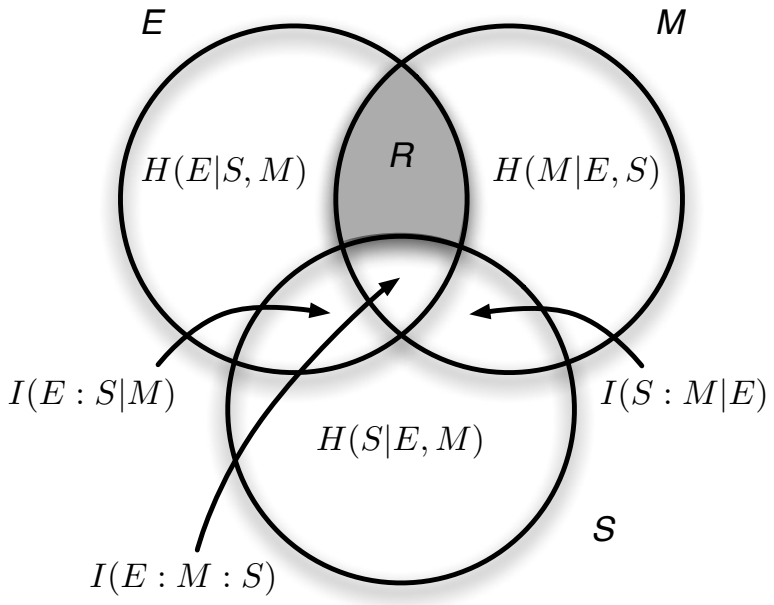


Figure 1: Venn diagram of entropies and informations for the three random variable $E$, $S$, and $M$, describing the states of the environment, sensors, and agent internal degrees of freedom. The representation $R = I(E:M|S)$.

where the correlation entropy $H_{\rm corr}$ of the three variables $E$, $S$, and $M$ [also called "total correlation" (Watanabe, 1960) or "multi-information" (McGill, 1954; Schneidman et al., 2003)] is the amount of information they all three share:

$$H_{\rm corr} = H(E) + H(M) + H(S) - H(E,M,S) \; . \tag{6}$$

In Eq. (5), we introduced the shared conditional entropy between three variables that is defined as the difference between an information that is unshared and one that is shared (with a third system), just as $H(X|Y) = H(X) - I(X:Y)$, from Eq. (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 Fig. 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 (see, e.g., Cover and Thomas,1991).

$R$ 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 world (and task) is complex enough, agents react to this challenge by evolving representations of that world.

## 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 et al. (2005) (see also van Dartel M.F.,2005). Categorization is thought to be one of the key elements in cognition (see Harnad 1987; Cohen and 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 is intimately linked with action. Whether or not 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 and 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 non-sensing units) in 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 Fig. 2A).

We evolve active categorical perception in two experiments, using two very different control architectures. One is a standard recurrent neural network (ANN), and one is a Markov network, specifically, a network of Hidden Markov Gates (HMGs) (Edlund

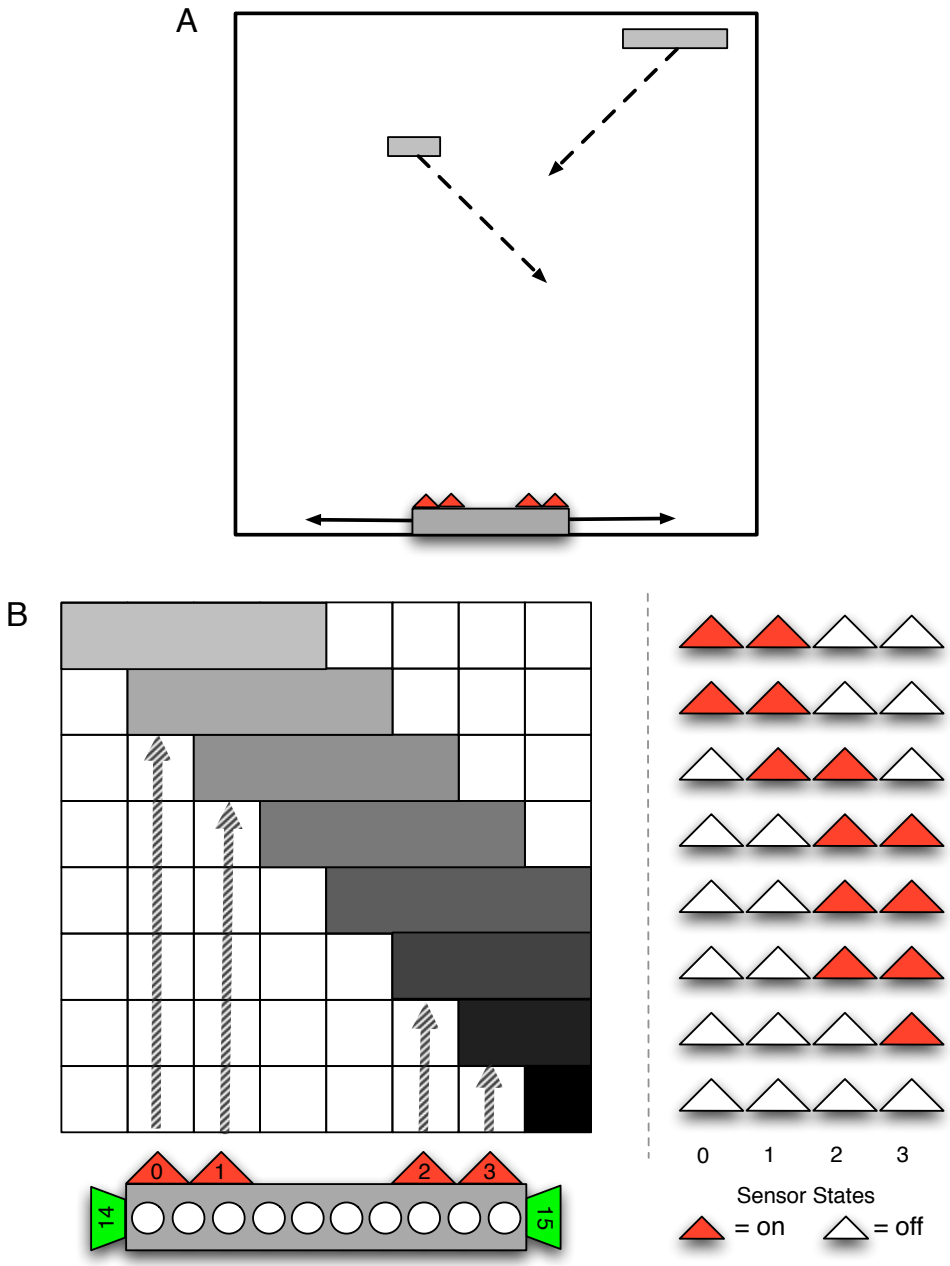


Figure 2: A: The world scene in which large or small blocks fall diagonally towards the bottom row of a $20 \times 20$ world, with the agent on the bottom row. For the purpose of illustrating the task, a large brick (to be avoided) is falling to the left, while a small brick (to be caught) is falling to the right. In simulations, only one block is falling at the time, and both small and large bricks can fall either to the left or to the right. B: A depiction of the agent's neurons (bottom left: triangles depict sensors, circles illustrate brain neurons, trapezoids denote actuators) and the sequence of activity patterns on the agent's 4-bit retina (right), as a large brick falls to the right.

et al., 2011), described in more detail in section 2.3. The agent is located at the bottom row of a $20 \times 20$ world with periodic boundary conditions, and is able to move horizontally one unit per time step (see Fig. 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 physical structure of the agent 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 sensor are in state $s_1 = s_2 = 1$ while the remaining two units of the block are positioned above the two non-sensing units. The other two patterns identify similarly positioned small blocks. All other five input states can be created by either a small block, a big block, or no block at all (see Fig. 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 downwards and sideways one unit per time step. Blocks either always move to the right or to the left. An object is caught if the position of the block's units and of 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 1,600 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 $E = E_0 E_1 E_2 E_3$ to take on 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 certainly underestimates the amount of "discoverable" entropy. However, in

| 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 |

Table 1: Coarse-graining of world states into the four bits $E_0, E_1, E_2, E_3$. Note that $E_1$ 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.

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 brain, by defining representations about different aspects $i$ of the world as the representation $R_i = H(E_i : M|S)$. Thus, we will study the four representations

$$R_{\rm hit} = H(E_0 : M|S) \quad (7)$$

$$R_{\rm LR} = H(E_1 : M|S) \quad (8)$$

$$R_{4/2} = H(E_2 : M|S) \quad (9)$$

$$R_{+/-} = H(E_3 : M|S) \quad (10)$$

that represent whether the sensor has been activated [Eq. (7)], whether the block is to the left or the right of there agent [Eq. (8)], if the block is large (size 4) or small (size 2) [Eq. (9)], or whether the block is moving to the left or right [Eq. (10)]. We can also measure how much (measured in bits) of each binary concept is represented in any particular variable. For example, $R_{\rm LR}(\text{node } 12) = H(E_1 : M_{12}|S)$ 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 onto 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 of the output layer. We further define sensor neurons as those neurons that directly process the input (the input layer) and we define output neurons as those units that do not map to other units in the network or to themselves (the output layer).

**Recurrent Artificial Neural Network (ANN).** In our first experiment, the agent's movements are controlled by an artificial neural network that consists of four input units (one for each sensor), two output units, six hidden units, and four recurrent units (see Fig. 3). We include recurrent neurons in the otherwise feed-forward architecture because we believe that memory is necessary to solve the task perfectly, and recurrent neurons can provide short-term memory for the network (Russell and Norvig, 2003). The states of the input and output units are discrete with values $[0, 1]$ specifying whether an object is detected or not (for the input units), or encoding one of three possible actions: move one unit to the right or left, or do not move. We chose $A_1A_2 = 00$ and $A_1A_2 = 11$ to encode 'stand still', $A_1A_2 = 01$ : 'move right one step', and $A_1A_2 = 10$ : 'move left one step'. Note that the original implementation of van Dartel et al. (2005) allowed for larger jumps to the left and right.

The states of the hidden and recurrent units $m_i$ are continuous with values $[-1, 1]$. All neurons (except for the input units) are updated through multiplication by the connection weights $w_{ij}$ and fed into a transfer function

$$m_i = \tanh\left(\sum_{j=1} w_{ij} m_j\right) \ . \tag{11}$$

In Eq. (11), the sum runs over all those units that feed into it, e.g., over the input and recurrent units for the hidden nodes, and the hidden nodes for the output neurons. The network is evolved in a standard manner, by changing the 84 weights $w_{ij}$ via a Genetic Algorithm, as described in more detail below. Each weight is encoded as floating point numbers, with initial values $w_{ij}^{(0)}$ drawn from the continuous uniform distribution $\mathcal{U}(-1, 1)$, and mutated with a probability of $\mu = 0.02$ per locus, which implies that on

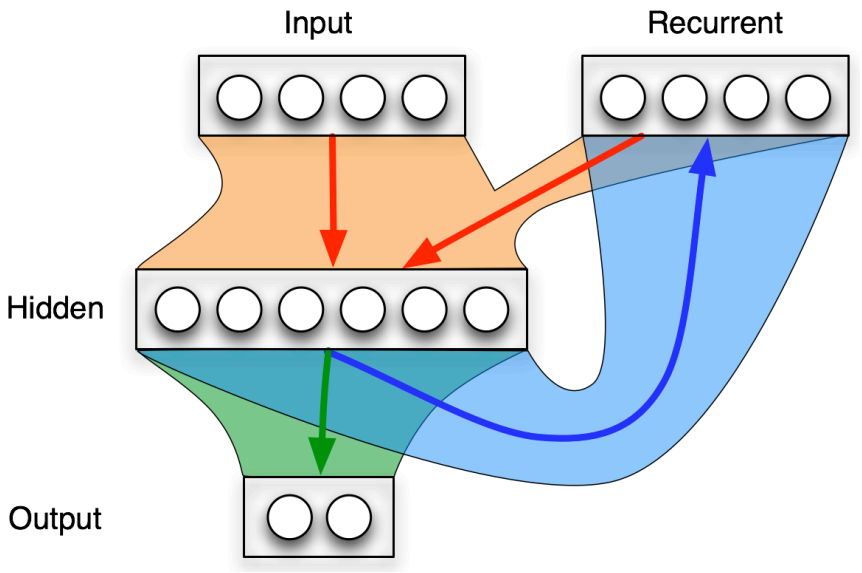


Figure 3: Recurrent Artificial Neural Network, with an input layer consisting of four neurons, a hidden layer of six neurons, an output layer of two neurons, and a recurrent layer of four neurons.

average 1.68 weights are changed per generation. Each mutation changes the weight by adding a number drawn from the continuous uniform distribution $\mathcal{U}(-0.2, 0.2)$. All outputs (values of the neurons in the output layer) are mapped from the continuous interval $(-1, 1)$ to the discrete values $0$ or $1$.

**Markov Brain (MB).** In our second experiment, the agent is controlled by a network of 16 nodes (4 input, 2 output, 10 internal nodes, that is, with the same input/output structure as the ANNs) that are interconnected via Hidden Markov Gates (HMGs, see Edlund et al. 2011). Networks of HMGs (Markov brains or MBs for short) are a type of stochastic Markov network (see, e.g., Koller and Friedman 2009). Markov brains are related to the hierarchical temporal memory (HTM) model of neocortical function (Hawkins and Blakeslee, 2004; George and Hawkins, 2005, 2009) and the HMAX algorithm (Riesenhuber and 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.

Each HMG can be understood as a finite-state machine that is defined by its input/output structure (Fig. 4A) and a state transition table (Fig. 4B). All neurons in

Markov brains are binary, and in principle the gates are stochastic, that is, the output nodes fire (that is, are set to state '1') with a probability determined by the state-to-state transition matrix. Here, each gate can receive up to four inputs, and distribute signals to up to 4 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* gates (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. For the purpose of evolution, each HMG (and its connection to other HMGs) is encoded in a (circular) 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 neurons being written to.

For example, for the gate depicted in Fig. 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 Fig. S1 in Edlund et al. 2011 for more details.) For the example given in Fig. 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 (42,114). Because this combination only occurs by chance once every 65,536 pairs of loci (making start codons rare), we insert four start codons at arbitrary positions into a 5,000 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.

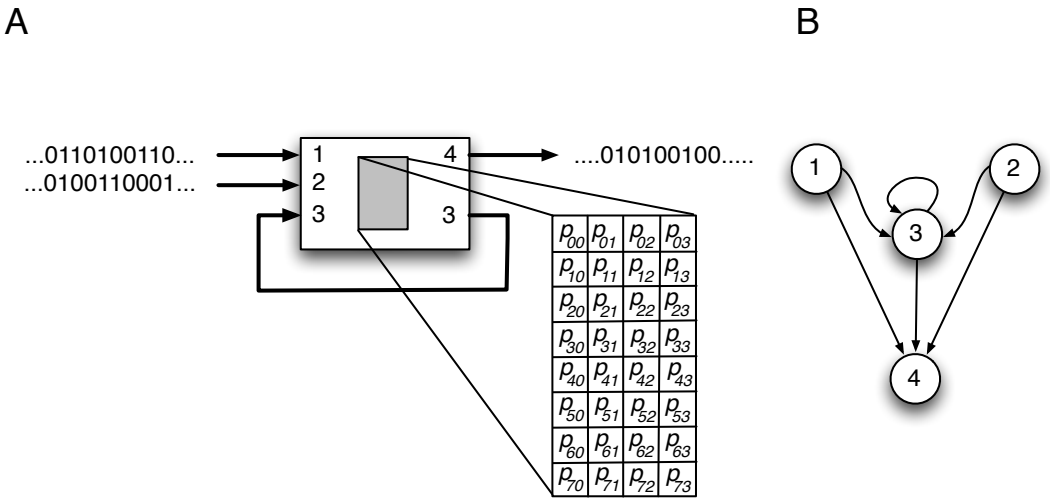


Figure 4: **A**: A single HMG with three inputs and two outputs reads from neurons 1, 2, and 3, and writes into neurons 3 and 4, updating the states of these neurons in the process. The output states are determined by a $2^3$-by-$2^2$ matrix of probabilities, here denoted as $p_{xy}$, where $x$ and $y$ 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, $P(11|111) = p_{73}$. **B**: The "causal" connectivity induced by the gate, obtained by connecting by a directed arrow all those neurons that can directly affect each other via the gate. Note that because neuron three affects its own state, it is shown to map to itself and can potentially be used as memory.

## 2.4 Evolutionary Algorithm

We evolve the two types of networks (ANNs and MBs) using a Genetic Algorithm (GA). A GA finds locally optimal solutions to problems by using evolutionary search [see, e.g., 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 zero and 1). For the purpose of selection, we use an exponential fitness score 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 with a probability of $\mu = 0.005$ per locus. We evolve networks through 10,000 generations, and run 200 replicates of each experiment. Note that the number of loci is different between ANNs (84) and MBs (variable), so the rate of evolution of the two networks cannot be compared directly. Also, the average deleterious effect of a mutation on a site differs between the two approaches, so the optimal mutation rate where evolution proceeds the fastest (Orr, 2000) differs.

At the end of each evolutionary run, we reconstruct the evolutionary *line of descent* (Lenski et al., 2003), by following the lineage of the most successful agent at the end of 10,000 generations backwards all the way to the random ancestor that was used to seed the experiment. This is possible because we do not use cross-over 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 (5).

### 2.5 Extracting probabilities from behavior

The input to our information-theoretic measure of representation $R$ are probabilities to observe a particular state $x$, $p(x)$, as well as joint probabilities $p(x, y)$ to observe a state $x$ when at the same time another variable $Y$ takes on the state $Y = y$ with probability $p(y)$. For the representation $R$ defined by Eq. (5), we distinguish sensor variables, internal variables ("memory") and world variables. For any particular organism (an agent that performs the ACP task with an evolved controller) we can measure $R$ at any point during evolution by placing the organism into the simulated world and recording time series data of the state of all 16 nodes (neurons or units) and the state of the world at the same time, and use it to calculate the frequency of states (how many times a particular state was observed during the lifetime of the agent). For other information-theoretical quantities that take into account the temporal order of events (for example, 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 $y_{t+1}$), we record the relevant frequency of pairs of states, etc. If a particular state (or combination of states) never occurs, we record a probability of zero for that entry (even though in principle the state or combination of states could occur). We usually calculate $R$ (and other information-theoretic quantities introduced below) for organisms on the evolutionary line of descent, which allows us to follow how $R$ evolves from a random ancestor to an adapted behaving agent. For ANNs that have internal neurons with continuous rather than binary states, we map the interval $[-1, 0] \to 0$ and $]0, 1] \to 1$ before calculating probabilities.

## 3 Results

To establish a baseline, we created 100,000 random controllers (100,000 random ANNs and 100,000 random MBs) and obtained the distribution of $R$ values as well as fitness values. This baseline serves two purposes: we want to know how well a randomly generated (unevolved) network performs and how much information about the world is represented in a network by chance. At the same time, we are interested in the distribution of these values. We created random ANNs by choosing weights drawn

from a uniform probability distribution $\in [-1, 1]$ for each of the 84 weights. Random Markov brains were created just as the evolutionary ancestors, by choosing random genomes of 5,000 loci, each locus drawn form a uniform probability distribution on the integers $\in [0, 255]$, sprinkled with 4 start codons at arbitrary positions within the genome.

In Fig. 5A, we show the probability distribution of fitnesses for 100,000 random ANNs, and their representation scores in Fig. 5B. ANNs achieve a wide distribution of fitnesses because random ANNs are dense within the space of dynamical systems (Albers et al., 1998), meaning that a wide variety of dynamical behaviors are available without any fine-tuning of parameters. Surprisingly perhaps, ANNs also represent significant amounts of information about their world by chance (see Fig. 5B). We will discuss this finding in more detail below. In contrast, the distribution of fitness for Markov brains is dominated by a single peak, due to a large majority of random agents that stand still and do not react to the falling blocks. Such agents achieve some fitness because they avoid on average $22$ of the large blocks (out of 40: twenty of them moving left, 20 to the right) and are hit by $14$ of the small blocks for a fitness of $w = 36/80 = 0.45$. The distribution of representation in random Markov brains is highly skewed towards zero, with a small tail. Thus, random Markov brains are born without representing information about the world (except by chance), whereas the ANN structure appears to represent by construction.

We now turn to analyzing the evolutionary trajectories for fitness and representation in our two neural architectures. To visualize the changes in the networks as evolution unfolds, we analyze fitness and representation on networks that lie along 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 coalesces quickly to a single line, so that the LOD effectively recapitulates the genetic changes that led from random networks to proficient ones. ANNs start out being significantly more fit than Markov brains because of the wide distribution of fitness in random networks seen in Fig. 5A. However, adaptation from thereon is slow and incremental (see Fig. 6A). Markov brains start out with random fitness 0.45 (agents that do not move), and increase their fitness steadily, overtaking ANNs after 2,000 generations (note that the development of fitness and representation over evolutionary time in Fig. 6 is averaged over 200 independent

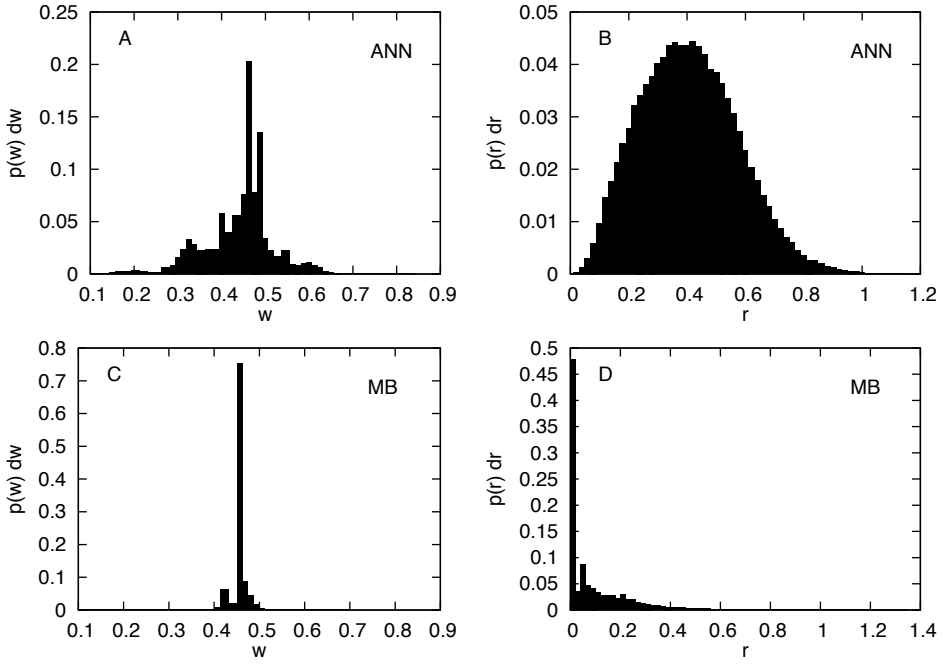


Figure 5: Probability distribution of fitnesses and representation scores of random machines. **A**: Probability distribution of fitnesses (fraction of successful actions) $p(w)dw = P(w < W < w + dw)$ for 100,000 random ANNs ($dw = 1/80$) **B**: Probability distribution of the representation variable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

replicates).

The evolutionary trajectory for representation $R$ differs significantly between the two architectures. The mean representation in ANNs begins high, as the first few evolutionary steps immediately take advantage of the distribution of random fitnesses, but hardly changes at all. Thus, ANNs represent by construction, but this representation does not increase as the fitness of the network increases (see Fig. 6B). Markov brains, on the other hand, start out with vanishing representation by definition, because unreactive networks cannot show any correlation with the world. But representations are built up quickly in Markov brains, and exceed the "chance-representation" of ANNs within the first 1,000 generations. In fact, when correlating fitness and representation at the end of a run for the 200 replicates of MBs, they are significantly correlated (Spearman's $r = 0.55$, $P = 0$), while ANNs are not ($r = 0.1$, $P = 0.15$).

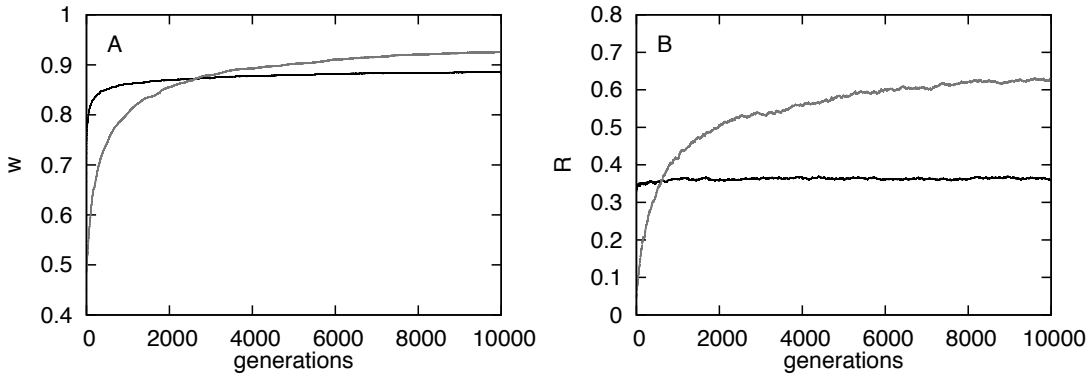


Figure 6: A: Fitness $w$ and B: representation $R$ (in bits) along the line of descent as a function of evolutionary generations, averaged over 200 evolutionary lines, for evolved networks (ANNs: black, MBs: grey)

To close this section, we speculate about the origin of the significant difference in representation $R$ in random MBs as compared to random ANNs. We believe that this difference is due to the network structure that is present *ab initio* in ANNs but not MBs. In ANNs, only the connection weights can be varied while in MBs not only the type of connection, but also the structure of connections between nodes is encoded in the genome and subject to evolutionary change. For example, in the Markov brain ancestral (random) genome, neurons are connected by only four Hidden Markov gates on average, implying that a significant number of neurons are simply unconnected to anything.

As a consequence, over 3/4 of random Markov brains are completely unresponsive, and thus represent zero bits by the definition of $R$. Random ANNs, on the other hand, are much more likely to contain high fitness individuals by chance and therefore begin the evolutionary trajectory with a marked fitness advantage as compared to MBs. Further, because these networks are fully connected, they will be responsive in some manner, and therefore can have a non-vanishing $R$ by chance.

Markov brains ultimately reach significantly higher fitness levels than ANNs do, possibly because of the flexibility in wiring, which is not present in ANN. As previously mentioned, the rate at which fitness is achieved in evolutionary time cannot be compared across architectures because the network is encoded in genomes of different size. But the fitness trajectories suggest that ANNs of the type we have investigated here will not be able to achieve the fitness levels that MBs reach, even though the number of nodes and the input/output architecture of both networks are the same. While we can tentatively explain the difference in performance between ANNs and MBs, we anticipate that the different network architectures also solve the categorization task very differently. To understand the information dynamics and the strategies employed in more detail, we calculated a number of other information-theoretic measures (besides $R$) and compared the networks.

## 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 that represents the category, the prediction (to move away or to stay) could be directly coupled to the attractor. In the task we investigate here, categorical perception requires the agent to compare sensory inputs from at least two different time points (which also allows a prediction of where the object is going to land). This task can only be achieved with memory. In ANNs, we investigated the evolved performance of the agent as a function of the size of the recurrent layer and found that the agent cannot evolve behavior better than chance in the absence of a recurrent layer, supporting our assumption that memory is necessary.

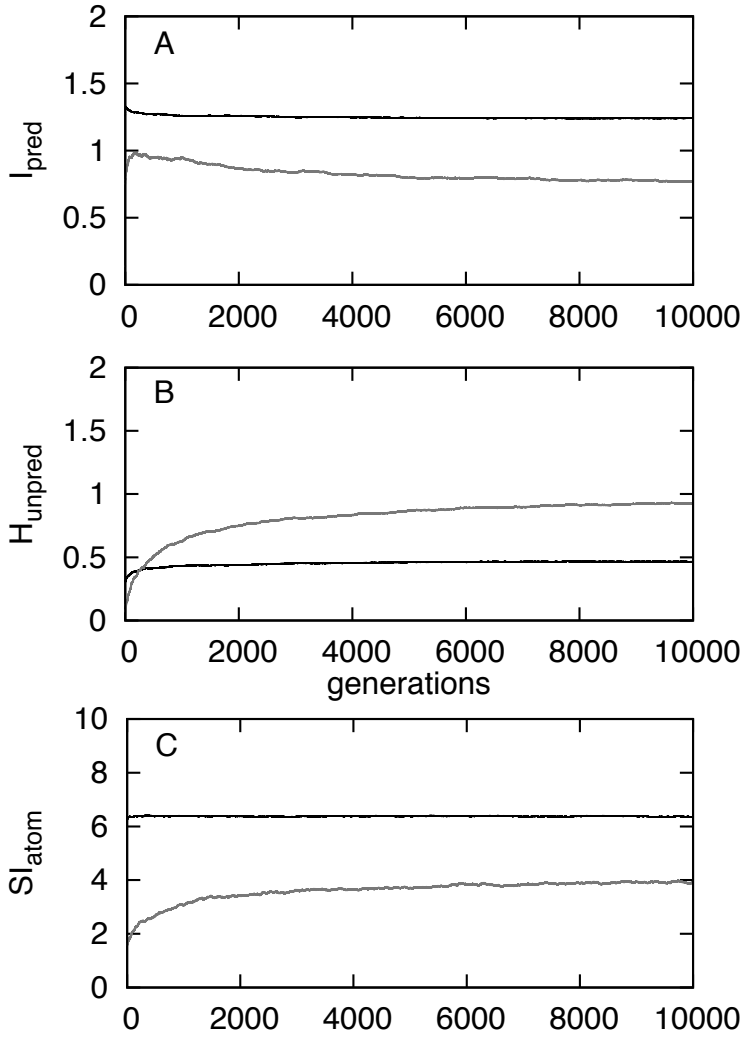


Figure 7: Different measures of information processing and integration along the LOD for both types of network architectures: ANNs (black) and Markov brains (grey). A: Predictive information, Eq. (12). B: Unpredicted entropy [Eq. (13)] of the network's motor variables. C: Information integration $SI_{\rm atom}$ based on Eq. (14).

To analyze how information is processed, we calculated the predictive information (Bialek et al., 2001) of the evolved networks, given by the mutual Shannon information between the network's inputs at time $t$, and outputs at time $t+1$. Predictive information, defined this way (Ay et al., 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. Using the variable $S$ for sensor states and $A$ for actuator states as before, 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

$$I_{\rm pred} = I(S_t : A_{t+1}) = -\sum_{s_t, a_{t+1}} p(s_t, a_{t+1}) \log \frac{p(s_t, a_{t+1})}{p(s_t)p(a_{t+1})} \ , \tag{12}$$

where $p(s_t) = P(S_t = s_t)$ is the probability to observe variable $S_t$ in state $s_t$, $p(a_{t+1}) = P(A_{t+1} = a_{t+1})$ is the probability to observe variable $A_{t+1}$ in state $a_{t+1}$, etc. Note that $S_t$ and $A_{t+1}$ are joint random variables built from the variables of each neuron, so $S_t$ can take on 16 different states while $A_{t+1}$ takes 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 $I_{\rm pred}$ decreases for both types of networks (even though only marginally for ANNs), after an initial increase for Markov brains. This effect is much stronger in MBs than in ANNs, suggesting that the former increase their use of internal representations.

To test whether it is indeed internal states that increasingly guide the agent, we subtracted the predictive information 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 firings are uncorrelated to signals from the input:

$$H_{\rm unpred} = H(A_{t+1}) - I_{\rm pred} = H(A_{t+1}|S_t) \ . \tag{13}$$

Figure 7B shows that $H_{\rm unpred}$ increases over the course of evolution, suggesting that indeed signals other than the sensor readings are guiding the motors. In principle, this 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 via 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 $I_{\rm pred}$ coupled with an increasing $H_{\rm unpred}$ to be the same as for the one-step predictive information, because the sensorial signal stream is itself correlated.

Finally, we calculated a measure of information integration called "synergistic information" to quantify the synergy of the network. 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 neuron, that is, it measures the extent to which the whole network is–informationally–more than the sum of its parts (Edlund et al., 2011):

$$SI_{\rm atom} = I(X_t : X_{t+1}) - \sum_{i=1}^{n} I(X_t^i : X_{t+1}^i) \ . \tag{14}$$

In Eq. (14), $I(X_t : X_{t+1})$ 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 Eq. (14) has been used before, to quantify the *redundancy* of information processing in a neural network (Atick, 1992; Nadal and Parga, 1994; Schneidman et al., 2003). $SI_{\rm atom}$ is a special case of the information integration measure $\Phi$ (Tononi, 2008; Balduzzi and Tononi, 2008), which is computationally far more complex than $SI_{\rm atom}$ because it relies on computing information integration across all possible partitions of a network. $SI_{\rm atom}$, instead, calculates information integration across the "atomic" partition only, that is, the partition where each node is its own part.

Figure 7C shows that $SI_{\rm atom}$ increases for Markov brains as the network adapts to perform the task. For ANNs, $SI_{\rm atom}$ is constant (as the structure of the ANN does not change during adaptation) and high. Neural networks appear to perform tasks synergistically by construction, as neurons have to cooperate to create attractors. However, this synergy is not adaptive: it is independent of the actual attractors of the network.

Taken together, Figures 6 and 7 suggest that Markov networks perform better than ANNs in this task because of the ability to represent, and to choose between different

actions based on the information encoded in these representations. In terms of the language of Bayesian inference, the network's representations can be understood as priors that reflect the relevant statistical distribution of the environment (Kording and Wolpert, 2004; Fiser et al., 2010; Thornton, 2009). We thus 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? Which features of the world are represented and form the successful task-solving strategies?

### 3.1 Epistemically Opaque Strategies

In order to analyze a Markov brain's function, we have recourse to a number of different tools. First, we can generate a *causal diagram* by drawing an edge between any two neurons that are connected via an HMG. The edges are directed, but note that each edge can in principal perform a different computation. When creating the causal diagram, we remove nodes that are never written into by any other nodes, as they are computationally inert (they remain in their default 'off' state). Such nodes can also be identified via a "knock-out" 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. Fig. 8 shows the causal diagram of an evolved Markov brain that solves the classification task perfectly. We can see that this brain uses inputs from the sensors, motors, and memory simultaneously for decisions, fusing these 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 present 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, for example by feeding the state-to-state transition table into a logical analyzer (Logic Friday, http://sontrak.com). 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 Fig. 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

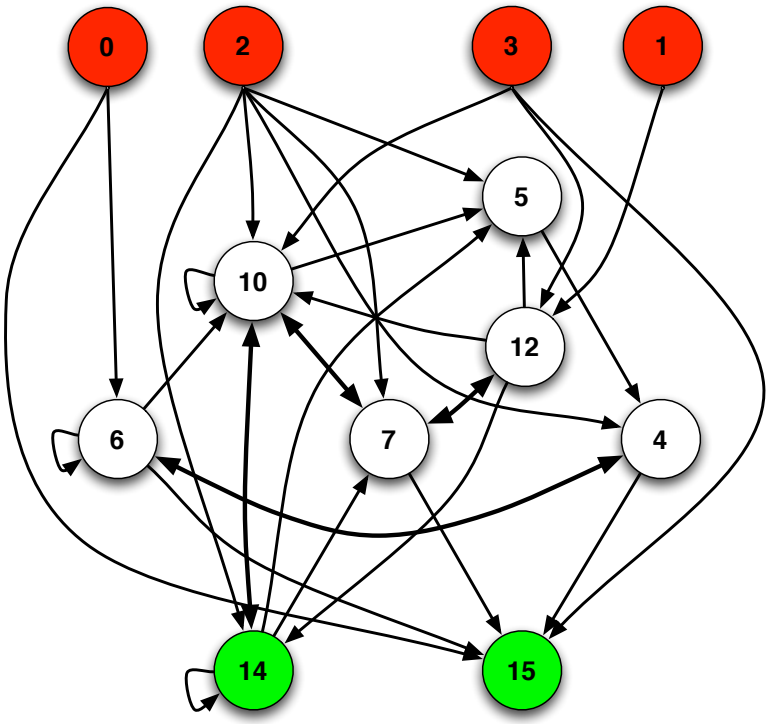


Figure 8: Causal diagram (see Fig. 4B) of a Markov brain with perfect fitness, correctly catching all small blocks and avoiding all large ones. Nodes colored in red are sensors, while motor variables are green. Double arrows represent two connections, one each way. Nodes with arrows that point to themselves write their output back into their input, and may work as memory (all nodes return to a default "0" after firing, so state information can only be maintained 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.

stands for NOT)

$$
\begin{aligned}
4_{t1} &= (\overline{2}_{t0} \wedge \overline{5}_{t0}) \vee (2_{t0} \wedge 6_{t0}) \\
5_{t1} &= (\overline{2}_{t0} \wedge 10_{t0} \wedge \overline{12}_{t0}) \vee (\overline{10}_{t0} \wedge \overline{12}_{t0} \wedge \overline{14}_{t0}) \vee (\overline{2}_{t0} \wedge 10_{t0} \wedge 12_{t0} \wedge \overline{14}_{t0}) \\
6_{t1} &= \overline{4}_{t0} \wedge (0_{t0} \vee 6_{t0}) \\
7_{t1} &= (2_{t0} \wedge \overline{12}_{t0}) \vee (\overline{10}_{t0} \wedge 12_{t0} \wedge 14_{t0}) \\
10_{t1} &= (2_{t0} \wedge 6_{t0}) \vee (\overline{2}_{t0} \wedge 10_{t0} \wedge \overline{14}_{t0}) \vee (10_{t0} \wedge 12_{t0} \wedge \overline{14}_{t0}) \vee (2_{t0} \wedge 14_{t0}) \\
&\quad \vee (3_{t0} \wedge \overline{7}_{t0} \wedge \overline{10}_{t0} \wedge 12_{t0} \wedge 14_{t0}) \\
12_{t1} &= (\overline{1}_{t0} \wedge \overline{7}_{t0}) \vee (\overline{3}_{t0} \wedge 7_{t0}) \vee (3_{t0} \wedge \overline{7}_{t0}) \\
14_{t1} &= (2_{t0} \wedge 10_{t0} \wedge 14_{t0}) \vee (2_{t0} \wedge \overline{10}_{t0} \wedge \overline{12}_{t0}) \vee (\overline{2}_{t0} \wedge 10_{t0} \wedge 12_{t0}) \\
15_{t1} &= (3_{t0} \wedge \overline{7}_{t0}) \vee (\overline{0}_{t0} \wedge 6_{t0}) \vee (4_{t0} \wedge 6_{t0}) \vee (0_{t0} \wedge \overline{4}_{t0} \wedge \overline{6}_{t0}) \ .
\end{aligned}
$$

Note that while this 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 and 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.

### 3.2 Concepts and Memory

To understand what representations are acquired (representations about which concepts), we calculated $R$ for each property of the environment defined in Eq. (7-10), within each of the key nodes in our example network. Figure 9 shows that some nodes prefer to represent certain features 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, only tells part of the story as it is clear that representations are "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, that is, variables can represent synergistically. In order to discover which combination of nodes represents which

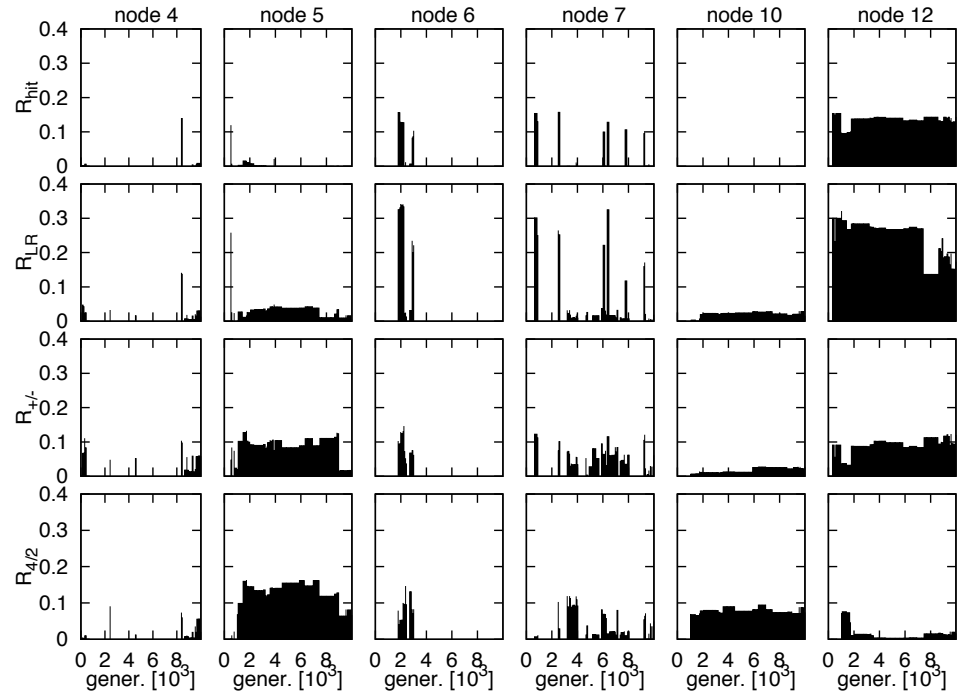


Figure 9: Representation of each of the 4 environmental properties (*concepts*) defined in Eqs. (7-10), within each of the nodes of a brain that evolved to become the one depicted in Fig. 8. Representation is measured in bits, along the line of descent, that is, over time (measured in generations).

feature most accurately, a search over all partitions of the network would have to be performed, much like in the search for the partition with minimum information processing (MIP) in the calculation of a network's synergistic information processing (Balduzzi and Tononi, 2008).

We 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. Because representations form in the memory of the brain, we are asking whether it represents more distant or more proximal events. To do this, we define temporal representations by including the temporal index of the Markov variables. For example, representation at the same time point $t$ is defined [as implicit in Eq. (5)] as

$$R_t = H(E_t : M_t|S_t) \ , \tag{15}$$

while we can define representation about events one update prior as

$$R_{t-1} = H(E_{t-1} : M_t|S_t) \ , \tag{16}$$

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, we can define temporal representations about more distant events in the same manner.

We calculated $R_t, R_{t-1}$, and $R_{t-2}$ averaged over all 80 experiments in each generation (for both ANNs and MBs) over the course of evolution. Figure 10A shows that while representation in ANNs is fairly constant (as noted before), $R_{t-1}$ is larger than $R_t$, and $R_{t-2}$ is larger still (but note that $R_{t-3}$ is smaller, data not shown). The same trend is visible in Markov brains, except that in these brains representation increases over evolutionary time.

We suggest that this peak in representation at a time difference of two updates 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 evolved, and furthermore that they *build up during an agent's lifetime* as memory of past events shape the agent's decisions.

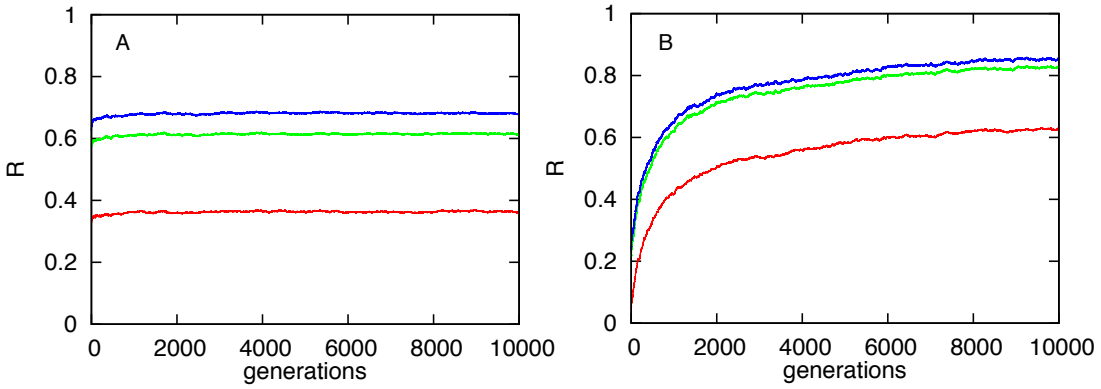


Figure 10: Representation as a function of evolutionary time, for three different time intervals. A: Representation in ANNs (red: $R_t$, green: $R_{t-1}$, blue: $R_{t-2}$. B: Representation in Markov brains (colors as in A). The red curves are the same as in Fig. 6B, and are shown here for the purpose of comparison.

# 4 Conclusions

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. We applied this measure to two types of networks that were evolved to control a simulated agent. Our experiments showed that the achieved $R$ increases with fitness during evolution of Markov brains, but stays approximately constant in the evolution of artificial neural networks. We also showed that while the (algorithmic) function of both neural networks and Markov brains is difficult to understand, deterministic Markov brains can be reduced to Boolean logic functions, which themselves may be epistemically opaque. While representation increases in Markov brains over evolutionary time, each neuron can represent parts of individual concepts, but most often concepts are distributed over several neurons, and represent synergistically, as can be seen by comparing Figures 9 (where the representation within each particular neuron is fairly constant over evolutionary time) and Fig. 10, which shows that the representation in the entire brain increases. The latter figure also shows that representations 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 brains via Darwinian processes are not the representations themselves, but rather what evolves is the

*capacity to represent the world.* The representation themselves are formed as the agent observes and interact with the world. This shows that a biologically realistic account of representation is able to account for context to sensorimotor loops without a central executive. As such, $R$ might prove useful for a wide range of applications in cognitive neuroscience and other related fields.

**Acknowledgements** We thank C. Koch and G. Tononi for extensive discussions about representations, information integration, and qualia. This research was supported in part by the the Paul G. Allen Family Foundation, by the National Science Foundations Frontiers in Integrative Biological Research grant FIBR-0527023, NSFs BEACON Center for the Study of Evolution in Action under contract No. DBI-0939454, as well as by the Agriculture and Food Research Initiative Competitive Grant no. 2010-65205-20361 from the USDA National Institute of Food and Agriculture. We wish to acknowledge the support of the Michigan State University High Performance Computing Center and the Institute for Cyber Enabled Research.

# References

Albers, D., Sprott, J., and Dechert, W. (1998). Routes to chaos in neural networks with random weights. *International Journal of Bifurcation and Chaos*, 8(7):1463–1478.

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., and Olbrich, E. (2008). Predictive information and explorative behavior of autonomous robots. *Eur Phys J B*, 63(3):329–339.

Balduzzi, D. and 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 Maes, P., Mataric, M., Meyer, J., Pollack, J., and Wilson, S., editors, *From Animals to Animats: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, pages 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., and Tishby, N. (2001). Predictability, complexity and learning. *Neural Computation*, 13(11):2409–63.

Bongard, J., Zykov, V., and 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, Mass.: The MIT Press.

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

Cohen, H. and Lefebvre, C., editors (2005). *Handbook of Categorization in Cognitive Science*, Amsterdam, The Netherlands. Elsevier.

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. John Wiley, New York, NY.

Craik, K. (1943). *The Nature of Explanation*. Cambridge University Press, Cambridge (UK).

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

Fiser, J., Berkes, P., Orban, G., and Lengyel, M. (2010). Statistically optimal perception and learning: from behavior to neural representations. *Trends in Cognitive Sciences*, 14(3):119 – 130.

Fodor, J. (1975). *The Language of Thought*. New York: Crowell.

George, D. and Hawkins, J. (2005). A hierarchical Bayesian model of invariant pattern recognition in the visual cortex. In Prokhorov, D., editor, *Proceedings of the Interna-*

*tional Joint Conference on Neural Networks (IJCNN)*, volume 3, pages 1812–1817. IEEE.

George, D. and Hawkins, J. (2009). Towards a mathematical theory of cortical micro-circuits. *PLoS Comput Biol*, 5(10):e1000532.

Harnad, S., editor (1987). *Categorical Perception: The Groundwork of Cognition*, Cambridge, UK. Cambridge University Press.

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

Hawkins, J. and Blakeslee, S. (2004). *On Intelligence*. Henry Holt and Co., New York, NY.

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

Johnson-Laird, P. and Wason, P. (1977). *Thinking: Readings in Cognitive Science*. Cambridge University Press.

Kabanets, V. and Cai, J.-Y. (2000). Circuit minimization problem. In Yao, F. and Luks, E., editors, *Proc. 32nd Symposium on Theory of Computing*, pages 73–79.

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

Kay, J. W. and Phillips, W. A. (2011). Coherent infomax as a computational goal for neural systems. *Bull Math Biol*, 73(2):344–72.

Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models*. MIT Press, Cambridge, MA.

Kording, K. P. and Wolpert, D. M. (2004). Bayesian integration in sensorimotor learning. *Nature*, 427(6971):244–247.

Lenski, R. E., Ofria, C., Pennock, R. T., and 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*. Henry Holt and Co., Inc., New York, NY, USA.

Marstaller, L., Hintze, A., and Adami, C. (2010). Measuring representation. In Christensen, W., Schier, E., and Sutton, J., editors, *ASCS09: Proceedings of the 9th Conference of the Australasian Society for Cognitive Science*, pages 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*. Springer Verlag, New York.

Moravec, H. P. (1984). Locomotion, vision and intelligence. In Brady, M. and Paul, R., editors, *Robotics Research 1*, pages 215–224.

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

Newell, A. and 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. and Floreano, D. (2000). *Evolutionary Robotics*. MIT Press, Cambridge, MA.

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

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

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

Pinker, S. (1989). *Learnability and Cognition*. Cambridge, Mass.: The MIT Press.

Pitt, D. (2008). Mental representation. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, fall 2008 edition.

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

Russell, S. J. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Pearson Education, Upper Saddle River, N.J., 2nd edition.

Schneidman, E., Still, S., Berry II, M. J., and 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.

Thornton, C. (2009). Representation recovers information. *Cognitive Science*, 33(8):1–30.

Tononi, G. (2008). Consciousness as Integrated Information: a Provisional Manifesto. *Biol Bull*, 215(3):216–242.

van Dartel, M., Sprinkhuizen-Kuyper, I., Postma, E., and van den Herik, J. (2005). Reactive agents and perceptual ambiguity. *Adaptive Behavior*, 13:227–42.

van Dartel M.F. (2005). *Situated Representation*. PhD thesis, Maastricht University.

Ward, R. and 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., and Jordan, M. I. (1995). An internal model for sensorimotor integration. *Science*, 269(5232):1880–1882.