

Representations of Space and Time in the Maximization of Information Flow in the Perception-Action Loop

Alexander S. Klyubin

A.Klyubin@herts.ac.uk

*Adaptive Systems Research Group, School of Computer Science,
University of Hertfordshire, Hatfield Herts AL10 9AB, U.K.*

Daniel Polani

D.Polani@herts.ac.uk

*Adaptive Systems and Algorithms Research Groups, School of Computer Science,
University of Hertfordshire, Hatfield Herts AL10 9AB, U.K.*

Chrystopher L. Nehaniv

C.L.Nehaniv@herts.ac.uk

*Adaptive Systems and Algorithms Research Groups, School of Computer Science,
University of Hertfordshire, Hatfield Herts AL10 9AB, U.K.*

Sensor evolution in nature aims at improving the acquisition of information from the environment and is intimately related with selection pressure toward adaptivity and robustness. Our work in the area indicates that information theory can be applied to the perception-action loop.

This letter studies the perception-action loop of agents, which is modeled as a causal Bayesian network. Finite state automata are evolved as agent controllers in a simple virtual world to maximize information flow through the perception-action loop. The information flow maximization organizes the agent's behavior as well as its information processing. To gain more insight into the results, the evolved implicit representations of space and time are analyzed in an information-theoretic manner, which paves the way toward a principled and general understanding of the mechanisms guiding the evolution of sensors in nature and provides insights into the design of mechanisms for artificial sensor evolution.

1 Introduction ---

1.1 Information in the Perception-Action Loop. In nature, sensors and actuators of living agents are constantly engaged in complicated interaction. Despite the sensors and the actuators being specific to agents' niches and needs, a general and at the same time a minimal model of sensorimotor interaction is desired.

Although a large body of research identifies information as useful for understanding sensors, less work has been done on actuators. The pressure to capture information relevant to an agent's survival may push sensors to the limits. For example, vision sensors can operate at the limits set by physics (Bouman, 1959; Bialek, 1987). Capturing and processing sensory information have their metabolic costs (Laughlin, de Ruyter van Stevenick, & Anderson, 1998). Neural pathways have limited information transmission rates. Optimization of information-related quantities in artificial neural networks results in phenomena similar to ones found in biological sensory pathways (Barlow, 1959; Linsker, 1988; Nadal & Parga, 1995; Bell & Sejnowski, 1995) and information-theoretic tools provide versatile ways to analyze biological and artificial neural systems (Avraham, Chechik, & Ruppín, 2003; Schneidman, Bialek, & Berry, 2003).

Actions have also been identified as important from an informational perspective; however, to the best of our knowledge, less research has treated actions information-theoretically on par with sensors. Actions may facilitate active perception to uncover hidden information or simplify information processing in sensory pathways (Kirsh & Maglio, 1994; Nolfi & Marocco, 2002). Actions are also crucial for offloading and later reacquisition of information (Hutchins, 1995; Klyubin, Polani, & Nehaniv, 2004c).

In this letter, the use of causal Bayesian networks is introduced to model the perception-action loop. This formalism allows the treatment of perception action in a consistent information-theoretic manner: from the capture of environmental information through the sensors, to the "imprinting" of information onto the environment through the actuators. The causal structure of the networks allows the formulation of the concept of information flow (Ay & Wennekers, 2003; Klyubin, Polani, & Nehaniv, 2004a; Ay & Krakauer, 2007). Information flows are a signature of the causal structure, and they allow one to track the flow of pieces of information through the system analogous to how the flow of radioisotope tracers is tracked through a living organism.

Quantitative information-theoretic treatment of sensors and actions as parts of the perception-action loop was implicitly done in the early work of Ashby (1956) and recently revived by Touchette and Lloyd (2000). This letter proposes an information-theoretic method with a minimal set of assumptions for studying and constructing perception-action loops, which may span an arbitrary number of time steps as opposed to just a single time step as in the models of Ashby, and Touchette and Lloyd. The use of information theory reduces the bias of the processing architecture to a minimum, creating a "coordinate-free"¹ view of what happens with information processing.

¹ We are indebted to Peter Dauscher for suggesting this analogy.

1.2 Contributions of the Letter. The main contributions of this letter are:

1. Formulating an information-theoretic picture of the perception-action loop with a minimal set of assumptions. The interactions of an agent's sensors, actuators, memory, and the environment are modeled using causal Bayesian networks.
2. Maximizing information flow through the full perception-action loop of an agent, replacing structural assumptions such as those of infomax of Linsker (1988; e.g., receptive fields, layers) or the compositionality of temporal infomax of Ay and Wennekers (Ay & Wennekers, 2003; Wennekers & Ay, 2005; i.e., components of a finely grained computational architecture, which are identifiable through time) by agent-environment interaction over many time steps and minimal assumptions about the agent's information processing.
3. Using information-theoretic tools to analyze representations² of space and time resulting from the perception-action loop infomax. Since some of the representations are hard to interpret due to the lack of architectural assumptions, the judicious use of information-theoretic tools is essential.

1.3 Outline of the Letter. This letter is structured as follows. First, related work is discussed in relation to this letter in section 2. In section 3, a formalism for treating the perception-action loop using information-theoretic methods is laid out.

In section 4, the formalism is used to maximize an information flow through a simulated agent and its environment through time. Evolutionary search is employed in the experiment to find agent controllers that capture the maximum amount of information about the state of the world.

The evolved controllers are analyzed in the second part of the letter by looking at their mechanisms and their behavior through information-theoretic tools. The following questions are addressed. What information about the initial position do the evolved automata capture? How do the controllers capture the information? How is the information represented in the state of the controller?

In section 5 the representation of the agent's initial position in the state of its controller at the end of the run is analyzed, extending the results of Klyubin et al. (2004a) by studying how the representations change with small variations in the agent's "embodiment" (what the sensors capture from the environment and what the actuators do in the environment).

² *Representation* is a loaded term (cf. Millikan, 2002; Usher, 2001). In this letter, the "representation" of one variable by another means the probabilistic mapping or correspondence between the two variables.

In section 6 the representation of time inside the agent and by the agent-environment system as a whole is studied. A method to quantify the time dependence of representations is proposed in section 7. Since some of the representations appear to be factorizable (decomposable into independent components), in section 8 the representations of space (initial position) and time are factorized. Finally, overall discussion and conclusions are presented in section 9.

In appendix A, the information-theoretic quantities used in the letter are briefly defined. Appendix B remarks on modeling an agent's controller using causal Bayesian networks. Appendices C and D explain how information flows are measured and how the controllers are evolved in this letter. Appendix E proves the assertion for the grid-world model used in this letter that when an agent follows the gradient and reaches the source, most of the information about the agent's initial position relative to the source passes through its sensors. Appendix F continues the discussion of time dependence of representation by considering two examples. Appendix G contains the details of the factorization method used in section 8 and discusses its relation to synergy and the parallel bottleneck case of the multivariate information bottleneck method.

2 Background

2.1 Related Work

2.1.1 Perception-Action Loop and Information Theory. Gibson (1979) proposed that animals and humans do not normally view the world in terms of a geometrical space, independent arrow of time, and Newtonian mechanics and that the natural description is rather in terms of what an agent can perceive and do. The interplay between sensors and actuators, the perception-action loop, is also central to the notion of homeostasis, introduced by Cannon (1939) and treated in more detail by Ashby (1952, 1956). The main assumption is that agents act so as to keep certain internal variables (essential variables) within desirable ranges. Considering that the essential variables are the result of perception of external or internal states, homeostasis creates a purely internal goal, where action is for the purpose of controlling perception (Powers, 1974). Homeokinesis (Der, Steinmetz, & Pasemann, 1999) is a dynamic version of homeostasis where the agent acts so as to minimize the prediction error of its past and future sensoric input.

If the sensors can be seen as taking information in from the environment, the actuators can then also be thought of as having the capacity to modify the environment in terms of information. Interestingly, Gibson (1979) argued against applying information theory to the agent's perceptions, his main argument being that the environment does not intentionally "speak" to the observer and that "the assumption that information can be transmitted and

the assumption that it can be stored" are not appropriate for the theory of perception (p. 242).

However, work in perceptual neural networks (see section 2.1.2), recent work in control (Touchette & Lloyd, 2004), and implicitly also Ashby's work (1956) show that information theory can be productively applied to perception-action without the need to assume any intentionality. Touchette and Lloyd (2000, 2004) provided bounds on the efficiency of any sensor for one-step control (entropy reduction). Their result for closed-loop control can be interpreted as a generalized statement of Ashby's law of requisite variety, which states that only variety (entropy) in actions can reduce the variety (entropy) in the environment.

2.1.2 Neural Networks and Information Theory. Attneave and Barlow in the mid-1950s recognized the importance for the brain not just to capture information but also to represent it in a convenient way. Attneave (1954) observed that sensoric input of living organisms is very much redundant. He hypothesized that sensory pathways take advantage of the redundancy and also abstract away unnecessary or overwhelming details to represent the remaining sensoric information in an economical way. Barlow (1959, 1963) proposed the principle of redundancy reduction according to which sensory information in the brain is incrementally recoded to reduce the redundancy of its representation. Recently (Barlow, 2001) relaxed the claim by noting that (1) further up the processing chain, redundancy of representations can actually increase rather than decrease, and (2) highly compressed representations may be harder to use than more redundant ones. The emphasis was thus shifted from redundancy reduction to redundancy exploitation.

In a layered or hierarchical system it is unclear what information each of the layers ought to preserve without designing the knowledge of ultimate goals of the system into each layer. This is addressed by Linsker's principle of maximum information preservation, or infomax (Linsker, 1988). The principle requires each layer to maximize information transmission from its inputs to its outputs without bias as to what information to preserve and without the need for a layer to know anything about other layers. Infomax usually assumes spatial structure. For example, applied to a simulated retina-like layered feedforward neural network with localized receptive fields, the principle made the network self-organize and develop various feature detectors found also in the retinas of living beings.

In his later work Barlow extended what he meant by redundancy to include not only the unused capacity of a channel but also what he called "hidden redundancy"—the statistical interdependencies between components of output (Barlow, 1963, 2001). He proposed minimal entropy coding, which reduces "hidden" redundancy by decorrelating the components and leads to factorial codes where probabilities of individual components are independent of each other (Barlow, 1989). Nadal and Parga (1995) demonstrate that this particular type of Barlow's redundancy reduction and Linsker's

infomax are related and sometimes equivalent. A similar result is obtained in the context of source separation by Bell and Sejnowski (1995). A related method for arriving at factorial codes and feature detectors is the principle of predictability minimization by Schmidhuber (1992; Schmidhuber, Eldracher, & Foltin, 1996). The parallel information bottleneck principle proposed by Friedman, Mosenzon, Slonim, and Tishby (2001) as a variety of multivariate information bottleneck can also create factorial codes, the advantage over other methods above being the capability for lossy relevance-based compression.

Ay (2002) noted that although infomax may explain the functioning of early sensory stages, at later stages information from various pathways is integrated. He proposed the hypothesis of invariant maximization of interaction according to which "learning... is driven by mechanisms that maximize the stochastic interaction" between components of systems. The maximization of global stochastic interaction results in local infomax consistent with Linsker's idea. By incorporating the dynamical aspect Ay and Wennekers (Ay & Wennekers, 2003; Wennekers & Ay, 2005) proposed the principle of temporal infomax according to which the spatiotemporal stochastic interaction between various pathways or components is maximized. The approach is compositional as it assumes that an information processing system consists of components (e.g., neurons) that retain their identity through time. The complexity of its information processing is quantified information-theoretically by comparing components at two adjacent time steps. Temporal infomax consists of the maximization of this complexity.

2.2 Relation of Previous Work to This Letter. The prior work already noted is related to this work as follows:

1. The minimalistic model of an agent employed in this letter (see section 3.1) is defined in terms of what the agent's controller can access: sensors, actuators, and memory, conforming to Gibson's philosophy.
2. In this letter, the treatment of the perception-action loop by Ashby and by Touchette and Lloyd is generalized by modeling an arbitrary number of time steps.
3. The conceptual difference between this work and Linsker's infomax is that internal spatial structure in the processing architecture (i.e., receptive fields and layers of neurons) is replaced by natural constraints coming only from the agent-environment interaction ("embodiment") extended over many time steps.
4. Although information flow is maximized through time in this letter, the perception-action loop infomax is conceptually different from the temporal infomax of Ay and Wennekers. First of all, compositionality, the existence of separate finely grained components, is central to their

quantification method, whereas the split into sensors, actuators, and memory in this letter is not crucial for determining information flow quantities. Secondly, the temporal aspect of temporal infomax spans only two adjacent time steps, whereas this letter deals with information flows spanning an arbitrary numbers of time steps. The formalism presented here can be used to quantify and maximize information flows occurring between different parts of a system unconstrained by any temporal restrictions (e.g., information flows occurring across components but not through time), whereas the temporal aspect is central to the temporal infomax.

5. The factorization of the representations resulting from perception-action infomax (see section 8) is intimately related to but different from the creation of factorial codes. Here not only the output but also the mapping between the input and the output is required to be factorized.

3 Bayesian Model of the Perception-Action Loop

3.1 The Model. Central to the methodology is to model the perception-action loop of an agent as a causal Bayesian network. This can be a general and minimal model of an agent with a controller, possibly with memory. The following assumptions are made in the model:

- The agent is part of a larger agent-environment system.
- The system has discrete states.
- The system is discrete in time.
- Consecutive states of the system form a Markov chain. The momentary state of the system makes the past of the system statistically independent from its future.
- The agent's controller selects actions and modifies its own memory having access only to the momentary state of the sensors and own memory.

There are many ways to partition such a system into subsystems. Here the partitioning is done from the perspective of the agent's controller. The controller has direct access only to the agent's sensor,³ the actuator, and its own memory. Everything else in the system that is not encompassed by these three components is the rest of the system.

³ Although the agent can have more than one sensor and actuator, from now on they will be referred to in the singular. Multiple sensors can of course be treated as one composite sensor. Similar reasoning applies to actuators.

All the constituents of the perception-action loop are modeled as random variables:

- S —the state of the sensor
- A —the action performed by the actuator
- M —the state of the controller’s memory
- R —the state of the rest of the system

R formally accounts for the effects of actuation, the agent’s environment, and morphology on the sensors.

Sensors and actuators enable information flows between the agent and its environment. Time is introduced to model the temporal aspect of the flows. The states of the sensor, the actuator, the controller’s memory, and the rest of the system at discrete time t are denoted by random variables S_t , A_t , M_t , and R_t , respectively.

Modeling relations between the variables at different time steps unrolls the perception-action loop in time. The approach is not new (Touchette & Lloyd, 2000). However, as opposed to Touchette and Lloyd (2000), where only one time step is modeled, an arbitrary number of time steps is modeled here.

The relationships between the variables are modeled as a causal⁴ Bayesian network (Pearl, 2001), which is a directed acyclic graph where any node, given its parents, is conditionally independent from any other node that is not its parent or successor (any node directly or indirectly reachable from the node).

The pattern of relations between variables at consecutive time steps is shown in Figure 1. Assume that the pattern of relations is time invariant and thus holds for any t . Thus, the graph in Figure 1 is just a section of the network.

The agent’s controller is modeled with minimum assumptions about its architecture: the controller is discrete state, discrete time, and obeys the laws of classical probabilities. In the most general case, the controller performs a probabilistic mapping $(S_t, M_t) \rightarrow (A_t, M_{t+1})$.⁵ See appendix B for a more detailed discussion of the controller’s model as a causal Bayesian network.

⁴ A Bayesian network captures statistical relationships between observed variables. More than one Bayesian network may fit observed data. In a causal Bayesian network, the relationships between variables are not just statistical but also causal, corresponding to the underlying mechanisms that generate the data. This allows one to calculate effects of interventions—changes in some of the mechanisms (consult Pearl, 2001, for an in-depth treatment). Examples of interventions include fixing a variable to a particular value, or “injecting” information into the system (see below). The latter is used in this letter. In general, interventions can be defined only for causal Bayesian networks.

⁵ Throughout the letter the probabilistic mapping from a random variable X to a random variable Y is denoted using the shorthand notation $X \rightarrow Y$, which stands for the Markov kernel $\mathcal{X} \times \mathcal{Y} \rightarrow [0, 1]$, $(x, y) \mapsto \Pr(Y = y | X = x)$.

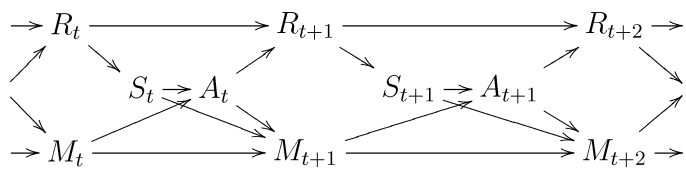


Figure 1: Perception-action loop as a Bayesian network. *S*—state of the sensor; *A*—action performed by the actuator; *M*—state of the memory of the controller; *R*—state of the rest of the agent-environment system. The diagram can be read as follows: action A_t is picked given sensor state S_t and memory state M_t , sensor state S_t is obtained from the state of the rest of the agent-environment system R_t , and R_{t+1} is obtained from R_t and A_t .

The causal Bayesian network model of the perception-action loop is useful for several reasons. First, the relations between the components of the perception-action loop are visualized. Second, various paths of the information flow (see section 3.2) and constraints on its bandwidth are visualized. For example, it is clear that information flows between the memory *M* and the rest of the system *R* are mediated by the sensors and actuators, and, hence, the bandwidth of the sensors and actuators sets a limit on the rate of information flows between the agent’s memory and the rest of the system. Third, the model, being a causal Bayesian network, makes it possible to analytically calculate various probability distributions as well as the results of interventions (see Pearl, 2001).

3.2 Information Flow. In the material world, the concept of matter flow is natural and widespread owing to the conservation laws and additivity of matter. For example, in nuclear medicine, radioisotope tracers are injected into a patient, and their flow through the organism is tracked. Once one tries to formulate an analogous concept of information flow, one encounters a number of differences, one being the nonconservative nature of classical information unlike matter. Classical information can be duplicated or destroyed. Furthermore, information can be “encrypted,” making it disappear unless one knows another “piece” of information that serves as the “decryption” key.⁶

As an illustration, consider an organization where confidential information is leaked to the outside and there are several suspects who have access to the information. To identify which one is leaking the information, one would provide each suspect with a different and deliberately made-up piece of information and see what information gets out. The presence of the

⁶ Here *encryption* is used in the broad sense of the word. The one-time pad encryption scheme (Schneier, 1995) is one example. Another example is the use of an address as a key to access information on a storage medium.

particular piece of injected information outside the organization will have to be solely due to the information flowing from the confidential source via one of the suspects.

Analogously, in this letter, information flow is defined for an arbitrary system modeled as a causal Bayesian network. Given a causal Bayesian network information flow from a random variable X to a random variable Y is defined as the amount of information about X causally transmitted from X to Y . Analogous to the information leakage example, if X is an input node of the network, then any information about X in a node Y is necessarily due to the causal effect (Pearl, 2001) of X on Y . In that case, the amount of information flow from X to Y is $I(X; Y)$, the information-theoretic mutual information between X and Y . Information flow is conceptually different from mutual information. Mutual information is a correlational and thus symmetric quantity: $I(X; Y) = I(Y; X)$ for any X and Y . Information flow is a causal quantity and is therefore directional and asymmetric.⁷

It may be possible to quantify the information flow in the general case, where X is an arbitrary node of the network. One needs to distinguish between information shared between X and Y due to the causal effect of X on Y and the information arriving from common predecessors. One could “block” unwanted paths of information flow to Y , leaving only those passing via X (see Klyubin, Polani, & Nehaniv, 2004b, section 8.2, for an example and Ay & Krakauer, 2007, for a more general treatment). A causal Bayesian network model helps identify the suitable sets of blocking variables.

4 Information Capture Experiment

4.1 Motivation. Linsker’s infomax principle for neural networks is concerned with the maximization of the information flow from the inputs of a perceptual layer of to its outputs (Linsker, 1988). As a consequence of preserving information about inputs, layers of center-surround cells, orientation-sensitive cells, and orientation columns appeared. An extended and generalized version of the infomax experiment can be performed using the causal Bayesian model of the perception-action loop. Instead of maximizing the information flow across just one layer of neurons, one can maximize the information flow through the perception-action loop of an agent through an arbitrary number of time steps. Moreover, since the model is more general, one need not assume linearity or a particular information processing substrate such as locally connected layers of neurons.

Chemotaxis, phototaxis, or gradient following in general is a good scenario because information about the position of the gradient’s point source must flow through the agent’s perception-action loop. In the following

⁷ In fact, information flow is nonzero only in the direction of the causal future (descendants) of X .

experiment, an agent controller is evolved to capture the agent's initial position relative to the source of the gradient. The goal may also be interpreted as the agent reconstructing the position of the gradient source at the beginning of the experiment.

4.2 Outline of the Experiment. An agent moves in an infinite two-dimensional grid world.⁸ The position of the agent corresponds to R , the rest of the system, in the Bayesian model of the perception-action loop. The agent's controller has no direct access to the agent's position. The controller has access to a gradient sensor that provides only a small amount of information about the position.

The agent's controller is evolved to maximize the information flow from the initial position R_0 into M_{15} , the state of the controller's memory at time step 15, under the condition that the controller is always started in the same initial state. Similar to the infomax experiment, the features of the initial position captured by the memory of the evolved controllers are analyzed. It turns out that most of the features have strong spatial continuity properties due to the way the agent perceives the world and acts in it. Therefore, small changes in the sensorimotor apparatus of the agent are made, and the impact on the captured features is studied.

4.3 Test Bed. The environment is a two-dimensional grid of infinite size. A source is located at the origin of the grid. The source emits a signal, the strength P of which in any cell of the grid is $P(r) = r^{-2}$ ($P(0) = 2$), where r is the distance to the source. The exact relation is not important for the experiments. It is only important that the decrease is strictly monotonic with distance.

An agent is situated in a single cell at a time. The agent has a gradient sensor. The gradient points to the cell with the highest signal strength among the four adjacent cells (north, east, south, west). If there are several cells with highest signal strength, the gradient randomly points to one of these with equal probability (see Figure 2). The agent also has an actuator: at each time step the agent performs one of the four available actions: move north, east, south, or west. The agent has a controller, which takes sensoric input and then moves the agent.

The global state of the system consists of just the agent's position on the grid and the state of the agent's controller. Following the perception-action loop model from section 3, denote the sensoric input (the gradient) with the random variable S taking values from the set $\mathcal{S} = \{s_N, s_E, s_S, s_W\}$; the action with the random variable A taking values from the set $\mathcal{A} = \{a_N, a_E, a_S, a_W\}$;

⁸ The grid is large enough so that the agent never experiences the boundary conditions. Without loss of generality, the grid can be considered infinite; however, only a finite number of cells is ever reached.

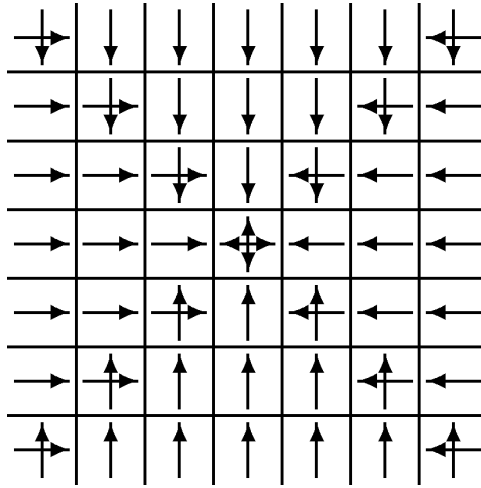


Figure 2: Sensoric input S depending on the position R . In cells with only one arrow the sensor input is constant. In cells with multiple arrows it is randomly chosen from the one of the arrows shown in the cells. The source is located at the center of the grid.

the internal state or the memory of the agent's controller with the random variable M taking values from the set $\mathcal{M} = \{1, 2, \dots, N\}$, where N is the number of states; and the two-dimensional position of the agent with the random variable R taking values from the set \mathbb{Z}^2 . S , A , M , and R completely describe the state of the whole system at any instant.

A_t and M_{t+1} depend directly only on (S_t, M_t) for any t (see Figure 1). The agent's controller, regardless of its implementation, is performing a probabilistic mapping $(S_t, M_t) \rightarrow (A_t, M_{t+1})$. The mapping is assumed to be time invariant.

The probabilistic mapping can be implemented by a nondeterministic Mealy finite state automaton operating with input set \mathcal{S} , output set \mathcal{A} , and state set \mathcal{M} . Preliminary experiments showed that the evolved mappings tend to be deterministic; hence, the experiments are constrained to deterministic automata, allowing only deterministic mappings. However, without any loss of generality, the approach can be used with stochastic mappings. Determinism of the controller is an experimental choice, not a limitation of the model.

4.4 Initial Position Capture. For this experimental setup it can be shown theoretically that, when an agent follows the gradient and ends up near the source, most of the information about the agent's initial position passes through the gradient sensor (see appendix E for a proof). Can this flow be directed into the agent's memory to capture information about

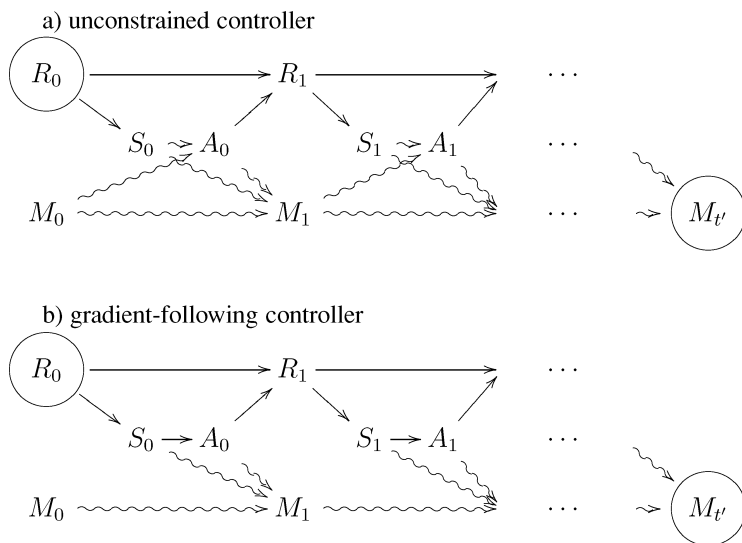


Figure 3: Initial position capture as optimization of the communication channel between R_0 and $M_{t'}$. Only the mappings represented by wavy lines can be modified.

the initial position of the agent, using a controller with limited memory and given limited time?

In finding a controller that maximizes the information flow⁹ from R_0 into $M_{t'}$, where t' is the duration of the run (see Figure 3), the controller can be seen as constructing a temporally extended communication channel between these two temporally separated parts of the system, passing not only through the sensor but also via the memory, the actuator, and the environment.

Two types of controllers are evolved:¹⁰ (1) unconstrained, which are not constrained in terms of the implemented mapping (see Figure 3a) and (2) gradient following (see Figure 3b), which are constrained to follow the gradient. Clearly, gradient-following controllers are a subset of unconstrained controllers where the information flow from M to A is blocked.

To evaluate a controller, the agent's initial position R_0 is randomly and uniformly distributed over a square with side d centered at the source. At the beginning of the run, the controller is always in state 1. The agent moves for t' time steps, after which its fitness $I(R_0; M_{t'})$ is evaluated.

$I(R_0; M_{t'})$ is the amount of information about the agent's initial position captured by the state of controller at time step t' . It quantifies how well one

⁹ See appendix C for a brief description of how the flow is measured.

¹⁰ The details of the evolutionary algorithm employed can be found in appendix D.

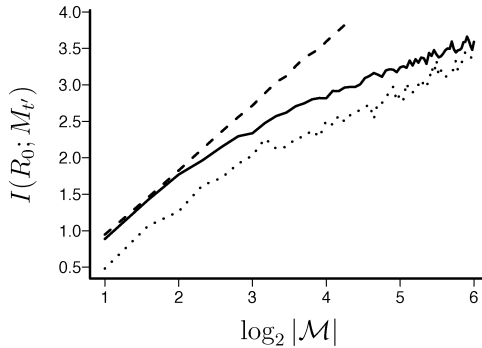


Figure 4: Information flow for best controllers. The amount of information flow from R_0 into $M_{t'}$ (ordinate) is plotted against the size of the controller's memory ($|M|$, abscissa). Both quantities are measured in bits. Dotted line: gradient-following controllers; solid line: unconstrained; dashed line: upper bound dictated by the IB method for gradient-following controllers. Each point is the maximum of five evolutionary runs. $d = 5$, $t' = 5$, 1000 generations.

can predict the state of the controller at time step t' from the initial position of the agent; or, alternatively, how well one can deduce the initial position of the agent knowing only the state of its controller at time step t' .

Evolutionary search is not guaranteed to find optimal solutions. However, an upper bound for achievable performance is the amount of information about R_0 in principle extractable from the cumulative sensoric input $S_{t'-1}^c = (S_0, S_1, \dots, S_{t'-1})$ into $M_{t'}$. Since R_0 , $S_{t'-1}^c$, and $M_{t'}$ form a data processing chain, it follows from the data processing inequality that no controller can extract more information into $M_{t'}$ about R_0 than it obtains via $S_{t'-1}^c$. However, since the cumulative sensoric input contains in the (temporal) limit all the information about the initial position, the main constraint derives from the agent's controller having access only to the momentary sensoric input, one at a time, and having limited memory.

The bound can be estimated for gradient-following controllers. Since all gradient-following controllers by definition employ exactly the same actuation strategy, that is, follow the gradient, the probability distribution of the random variable $S_{t'-1}^c$ is exactly the same for all the controllers. To calculate the bound, the information bottleneck (IB) principle (Tishby, Pereira, & Bielek, 1999) can be used to find a mapping $S_{t'-1}^c \rightarrow M_{t'}$ that maximizes the mutual information $I(R_0; M_{t'})$. The latter is an upper bound for the amount of information about R_0 any gradient-following controller can capture in $M_{t'}$.

In Figure 4, the performance of best-evolved controllers is compared to the size of their memory in bits. The controllers manage to extract information that is temporally "smeared" in sensoric input. Unconstrained

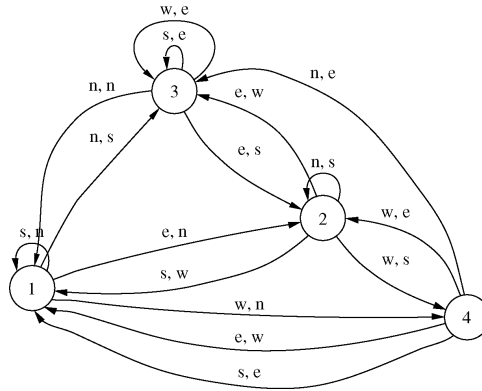


Figure 5: Best evolved four-state unconstrained controller as a finite state automaton. Circles are states. Arrows are transitions between states. Every transaction is annotated with the sensoric input from $\{n, e, s, w\}$, which triggers the transition, and this is followed by the performed action from $\{n, e, s, w\}$.

controllers clearly outperform gradient-following ones. However, as the performance of unconstrained controllers never exceeded the upper bound for the gradient-following controllers, nothing can be concluded about the nonoptimality of gradient-following controllers. However, the results suggest that this upper bound obtained using the IB principle may also apply to unconstrained controllers.

5 Representation of Positional Information

5.1 Introduction. In section 4.4 controllers were evolved to maximize the amount of information they capture about the agent's initial position. What information about the initial position is captured? How do the controllers capture the information? How is it represented in the state of the controllers?

The evolved controllers are finite state automata. One such automaton is shown in Figure 5 as an example. It is possible to analyze the static structure of the automata, for example, through hierarchical decomposition (Dömösi & Nehaniv, 2005). However, since the controllers were evolved to maximize information flow, it is natural to analyze them from the perspective of Shannon information, demonstrating that information-theoretic tools can be used not only for evolving systems but also for analyzing them.

The following sections compare and supplement the insights obtained using information-theoretic tools with prior knowledge about the analyzed system. A later goal, however, is to analyze systems by relying solely on the information-theoretic tools.

In this section in particular, the representation of the agent's initial position by the state of the controller at the end of the run is studied. The analysis extends the results of Klyubin et al. (2004a) by also studying how the representations change with small changes in the agent's embodiment (what the sensors capture about the environment and what the actuators do in the environment).

5.2 Representations as Mappings. Representations are treated as probabilistic mappings. To study the representation of the agent's initial position R_0 in the state of its controller at the end of the run $M_{t'}$, the mapping of $M_{t'}$ onto R_0 is visualized. The evolutionary experiment is rerun with $d = 11$ and $t' = 15$ to obtain spatially larger maps.

Mappings implemented by some of the best-evolved controllers are shown in Figure 6. The controllers perform lossy compression of the agent's initial position: some of the information about the agent's initial state R_0 is contained in the controller's state M_{15} at the end of the run. Unconstrained controllers successfully retain nearly the maximum amount of information that fits in their memory. Gradient-following controllers retain less information.

In some cases factorizable codes are produced: it is possible to find projections of M that independently code for independent features of position, such as odd and even cell checkerboard pattern or left and right half (see Figure 6, gf ++, $|M| = 8$). This suggests interesting parallels with the results obtained for information flow maximization in neural networks (Nadal & Parga, 1995; Bell & Sejnowski, 1995), where factorial codes are produced as a result of maximizing information transmission by a network of neurons. (Factorization is discussed in more detail in section 8.)

Maximizing the information flow from R_0 to M_{15} tends to create maps that induce near-hard partitions over R_0 . This is not surprising, since information maximization between any two variables will tend to create hard partitions. However, any permutation of a map with respect to the 11×11 set of initial position will preserve the mutual information between R_0 and M_{15} . Also, the form of the partitions is not important for capturing information. Theoretically, there is no preferred grouping of initial positions. The employed method of evolving the controller mappings is not biased in this respect either. One could thus expect that the form of the partitions of initial position would be completely different between different controllers due to random factors of the evolutionary process.

However, it turns out that the evolved maps are not arbitrary. There are certain repeating motifs in the maps: checkerboard patterns for gradient-following controllers and spatially solid tiles for unconstrained controllers. These regularities arise because not all of the maps from R_0 to M_{15} are possible. In short, the maps are constrained by and thus reflect the agent's embodiment and its information processing capabilities.

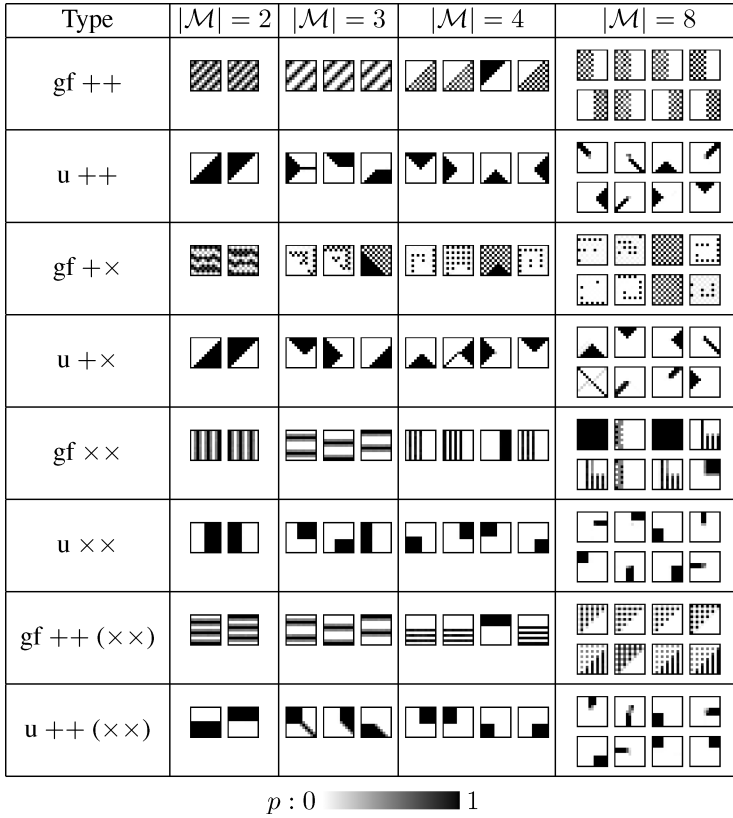


Figure 6: Representations of initial position ($M_{15} \rightarrow R_0$) used by best-evolved controllers with different memory sizes. Types: *gf*—gradient following, *u*—unconstrained, ++—cross-sensor and cross-actuator, +×—cross-sensor and diagonal actuator, ××—diagonal sensor and diagonal actuator, ++ (××)—evolved for ++ but mappings obtained with ××. Each box shows a mapping from a controller’s state at time step 15 onto the agent’s initial position (11×11 cells centered at the source). The intensity of each cell represents the probability of the agent having started in that cell: black—highest, white—zero. For example, state 1 of the unconstrained controller (*u* ++, $|\mathcal{M}| = 2$) at time step 15 means that the agent started in the lower-right half of the world. State 2 means the opposite.

To illustrate where the constraints arise, it is useful to briefly study the map $R_0 \rightarrow M_{15}$, the inverse of $M_{15} \rightarrow R_0$. The map is a product of identical time-invariant maps $(R_t, M_t) \rightarrow (R_{t+1}, M_{t+1})$ (time evolution operator) projected onto the M_t component. Already this fact is a constraint. The form of the time evolution operator is also constrained: the agent can move only into

adjacent cells, the sensor captures only certain features of the world, and, in the case of following the gradient, the agent's controller is constrained to strictly following the gradient.

The evolved maps $R_0 \rightarrow M_{15}$, and also $M_{15} \rightarrow R_0$, are thus not arbitrary. They reflect the structure of the agent's sensorimotor apparatus, the correlations between the actuator and the sensor—in other words, the agent's embodiment. The maps also reflect constraints on the agent's information processing capabilities. This is illustrated by the qualitative differences in maps of gradient-following (constrained) and unconstrained controllers (see Figure 6), as well as by quantitative differences between maps created by controllers with different number of states.

5.3 Different Embodiment. In the previous experiment the agent's body/embodiment was kept constant, and it was the controller that was allowed to change. To illustrate the idea that the evolved maps $R_0 \rightarrow M_{15}$ reflect both the embodiment and the constraints on the controllers, an evolutionary experiment was performed with different sensors and actuators but with the same constraints on the controller.

The simplest change to sensors and actuators is to permute them. This can be seen as changing the meaning of the sensoric states and actions. For instance, the action “go west” can be changed to move the agent one cell north. Correspondingly, the former “go north” action can be changed to move the agent west. The evolutionary search is not biased toward a particular meaning of sensoric states and actions. It treats sensoric inputs and actions as two unordered sets. Similarly, the fitness function, information flow, being an information-theoretic quantity, is not influenced by the permutations either. As a result, the evolutionary search is completely immune to permutations of meaning externally assigned to elements of the sensor and action sets.

What would happen if, instead of permuting sensoric states and actions, a new meaning were assigned to them in terms of what features of the position the sensors capture and how the actions change the position? In the previous experiment, the sensor was measuring the gradient by comparing the signal strength in four cells, located north, east, south, and west from the agent. Due to the layout, the sensor is called a cross sensor. In addition to the cross sensor, a diagonal sensor, which compares signal strength in four diagonally adjacent cells (northeast, southeast, southwest, northwest), will be used in the following experiments. Similar to the two types of sensors, the agent can have a cross (original) or a diagonal actuator.

Two more evolutionary experiments were run optimizing the information flow from R_0 to M_{15} for the case of cross sensor and diagonal actuator, and for the case of diagonal sensor and diagonal actuator. The resulting maps implemented by the best-evolved controllers are visualized in Figure 6, type $+\times$ and $\times\times$, respectively.

The second case, diagonal sensor and diagonal actuator, is interesting because the experiment is almost isomorphic to the original one rotated by 45 degrees. The difference comes from the fact that the initial position of the agent is still distributed around the source in a square exactly as in the original experiment. The best-evolved mappings $M_{15} \rightarrow R_0$ for the diagonal sensor and diagonal actuator case (see Figure 6, $\times \times$) are indeed quite similar to the best-evolved mappings rotated by 45 degrees obtained in the original experiment (see Figure 6, $++$) where cross sensor and cross actuator were used.

Due to this observation, it was interesting to see how the controllers evolved for the original scenario (cross sensor, cross actuator) behaved in the agent with diagonal sensor and diagonal actuator. It turns out that for a small number of states ($|\mathcal{M}| \leq 6$) they produce original maps rotated by 45 degrees (see Figure 6, $++ (\times \times)$) and actually capture a little bit more information about R_0 than in the original scenario. For larger number of states, the performance of gradient-following controllers drops compared to the original embodiment, and the maps no longer represent original features in a clean way (see Figure 5.2, $|\mathcal{M}| = 8$ gf $++ \times \times$).

5.4 Discussion

5.4.1 Experiments. The mappings from the controller's state at the end of the run M_{15} to the agent's initial position R_0 show what information about the agent's initial position gets captured by different controllers. Gradient-following and unconstrained controllers as a class capture different features of the initial position (see Figure 6). The main qualitative difference is that unconstrained controllers divide the initial position into spatially solid chunks. Neighboring initial positions are generally coded by the same state of the controller at the end of the run. Representations facilitated by gradient-following controllers lack the solidity of the chunks. These representations tend to have stripe or checkerboard patterns. The differences in representations derive from the differences in information processing capabilities (unconstrained versus gradient-following) and embodiment.

The results may thus be viewed as having extended and generalized Linsker's work on the infomax principle (Linsker, 1988) (see section 4.1) but with fewer assumptions: this letter does not assume linearity, it does not assume a particular information processing substrate for the agent's controller, and it allows for more than just one time step for information processing. The unconstrained controllers were even allowed to perform actions based on the past inputs. The results, however, are similar to Linsker's: without being told what is important about the agent's initial position, evolution found controllers that extract various features of the initial position as a result of maximizing information preservation. Despite the agent having no concept of geometry, most of the features, especially the ones created

by unconstrained controllers, have strong spatial continuity properties due to the way the agent perceives the world and acts in it.

5.4.2 Scalability. The main aim here was to find basic principles—hence the minimal bias and high inefficiency of the method. A natural question is how to scale the method up. More efficient methods of approximating information flow could be devised for particular environments or computational structures. A promising way of improving efficiency could be to factorize (see section 8) the information flows, thus creating small and manageable building blocks from which more complex hierarchical systems could be composed.

5.4.3 Biological Relevance. It may be argued that natural evolution is influenced by many factors and constraints that have no relation to information processing. However, assuming that the constraints take precedence, Linsker's infomax (Linsker, 1988) is a useful guide because more information is likely to be advantageous given the constraints: information may provide selective advantage (Howard, 1966; Polani, Nehaniv, Martinetz, & Kim, 2006). Thus, if nothing is known (assumed) about the system in advance, infomax is a maximally unbiased approach: transmit as much information as possible, since it is not known what information will be important in later stages. If, however, there is bias in terms of what information is important, then that information should be transmitted, and hence evolution will adapt accordingly by removing or reusing the unnecessary capacity. The information capture experiment based on the perception-action loop model demonstrates how under very limited constraints (limited sensors and actuators), the infomax principle results in self-organization of the information processing in an unstructured system. Additional constraints are expected not to reduce but rather modify the effect (cf. unconstrained versus gradient-following controllers). The minimum-assumption perception-action loop model here, combined with infomax, provides an organizing principle for information processing. However, infomax may be evolutionarily more important, resulting in trade-offs with other constraints. In fact, information may be a primary resource on par with energy: in the resting state the human brain is responsible for around 20% of the total oxygen consumption of the body (Kandel, Schwartz, & Jessell, 1991), and the fly photoreceptors consume around 10% (Laughlin et al., 1998) of the total energy consumed. Whether information is a primary resource or whether it is superseded by other constraints, the above indicates that infomax-like principles may help in understanding the organization of information processing in evolved natural and also artificial systems.

6 Representation of Time

Assume that during the experiment the state of the controller was observed without knowing the time. Would the state of the controller contain

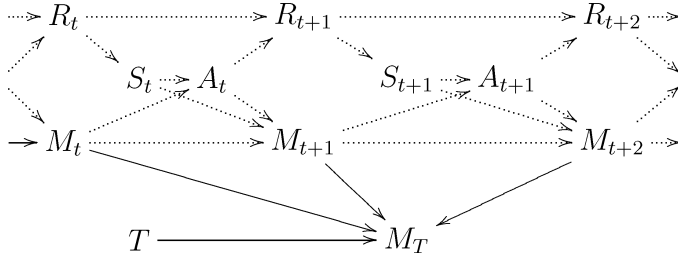


Figure 7: Relation of M_T , T , and the Bayesian model of the perception-action loop. M_T is a readout of the process $(M_t)_{t=0,1,\dots,15}$ at an unknown time step T . Neither T nor M_T has an effect on the dynamics of the loop.

information about the time step? It turns out that it would, provided the probability distribution of the controller's state depends on time.

Time is modeled as a random variable T with values t in $\mathcal{T} = \{0, 1, \dots, 15\}$. A special case is a peaked distribution where $p(t) = 1$ only for a single t and is 0 for all others, resulting in the observed state of memory M_t . However, in general, the distribution of T may be arbitrary. Denote the readout of the process $(M_t)_{t=0,1,\dots,15}$ observed at an unknown time step T by a random variable M_T (see Figure 7) with values m_T . Here $p(m_T)$ is given by

$$p(m_T) = \Pr(M_T = m_T) = \sum_{t \in \mathcal{T}} \Pr(T = t) \Pr(M_T = m_T | T = t), \quad (6.1)$$

where the probability distribution of the state of the readout for a particular time step t is inherited from the memory variable at the time step

$$\Pr(M_T = m_T | T = t) = \Pr(M_t = m_T). \quad (6.2)$$

The relation between M_T , T , and the process $(M_t)_{t=0,1,\dots,15}$ trivially follows from equations 6.1 and 6.2:

$$\Pr(M_T = m_T) = \sum_{t \in \mathcal{T}} \Pr(T = t) \Pr(M_t = m_T). \quad (6.3)$$

The amount of information captured by M_T about the time step T is $I(M_T; T)$. Sensoric input can also be included, in which case the amount of information about time available to the controller is $I((S, M)_T; T) \geq I(M_T; T)$, where $(S, M)_T$ is defined analogous to M_T . Informally, these quantities can be interpreted as the amount of information the controller has on the average about the time step it is at.

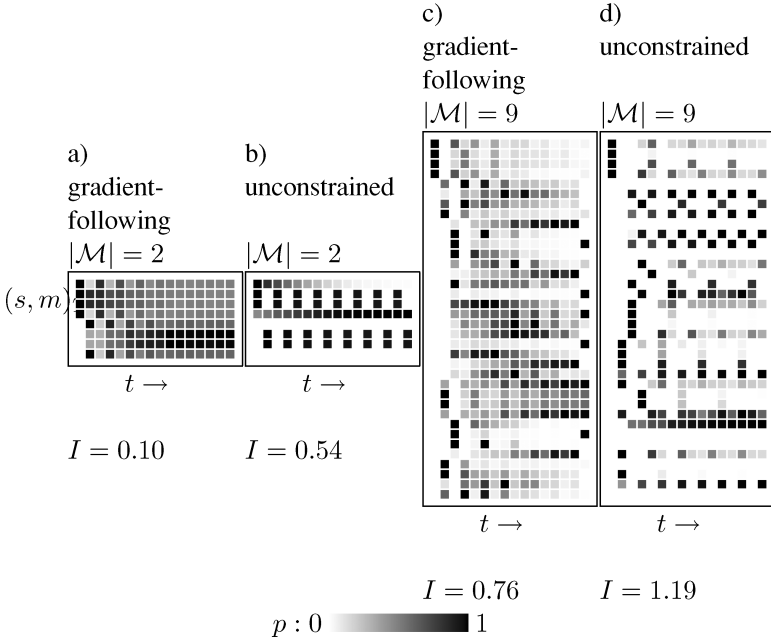


Figure 8: $p(t|(s, m)_T)$ —representation of time by the best two- and nine-state controllers. Vertical axis— $(s, m)_T$ — 2×4 states; horizontal axis— t — $\{0, 1, \dots, 15\}$. The darker the cell, the higher the conditional probability $p(t|(s, m)_T)$. $I = I((S, M)_T; T)$ measured in bits.

Assuming a uniformly distributed T , the controllers evolved in the initial position capture experiment capture from 0.1 to 0.84 bits of information about time in their sensors and memory. For comparison, the global state of the system $(R, M)_T$ captures between 0.93 and 2.00 bits of information about time. The small amounts of the former are not surprising considering that the controllers were not evolved for capturing information about time.¹¹

To investigate how time is represented by the controllers, the mapping $(S, M)_T \rightarrow T$ will now be studied.¹² For every tuple of sensoric input and state of the controller, the mapping tells the probability of being at a particular time step. To illustrate the idea, Figure 8 shows the conditional probability distribution $p(t|(s, m)_T)$ for some of the controllers.

Different states $(s, m)_T$ capture different features of time. One type of state captures local features of time. In Figure 8b, rows 2, 3, 6 and 7 (starting

¹¹ If the controllers were evolved to capture as much information about time as possible, they would evolve into clocks. The simplest solution is to shut out the environment and permute the controller's state.

¹² Strictly speaking, the reconstructed time is not metric time but an ordinal sequence.

from the top) are such examples. These states differentiate between odd and even time steps. In Figure 8d there are states that capture time modulo 2, 3, and 4.

Another type of states captures global properties of time, such as being close to the beginning, the middle, or the end of the experiment. Rows 1 and 4 in Figure 8 are an example.

There are also hybrid states, which capture both local and global properties of time. Row 1 and middle rows in Figure 8 are such states. They locate the system in time in terms of being at a particular modulo 4 time step and also in terms of being at the beginning or the end of the experiment.

The controllers were evolved to capture information about the agent's initial position at the end of 15 time steps. Why do the controllers capture information about time? Is capturing information about time an evolutionary neutral side effect, or is it crucial to capturing information about the agent's initial position? For example, it may be that the processing and integration of information by the controller depend on time, or that different methods of obtaining the information about the agent's position are used at different time steps, or that the controller represents the information in a time-dependent way. The issue of time-dependence of representation is addressed in the next section.

7 Time Dependence of Representation

Section 5 showed how M_{15} , the state of controller at the end of the experiment, represents R_0 , the agent's position at the beginning of the experiment. It turns out that the representation of the initial position in the controller's memory throughout the experiment is time dependent. For illustration purposes the representation used by the best eight-state unconstrained controller is shown in Figure 9.

The degree of time dependence of the representation of the initial position R_0 in the process $(M_t)_{t=0,1,\dots,15}$ is quantified as the probabilistic dependence of the mapping $M_T \rightarrow R_0$ on the time step T , where M_T is the readout of the process at an unknown time T . (See section 6 for the definition of T and M_T .)

For every observed state m_T of the controller, the mapping $M_T \rightarrow R_0$, characterized by the conditional probability distribution $p(r_0|m_T)$, describes where and with what probability the agent could have started given that the controller was found in that state at an unknown time step.

If, in addition to knowing m_T , the time step t is known, the mapping

$$p(r_0|m_T, t) = Pr(R_0 = r_0|M_T = m_T, T = t) = Pr(R_0 = r_0|M_t = m_T) \quad (7.1)$$

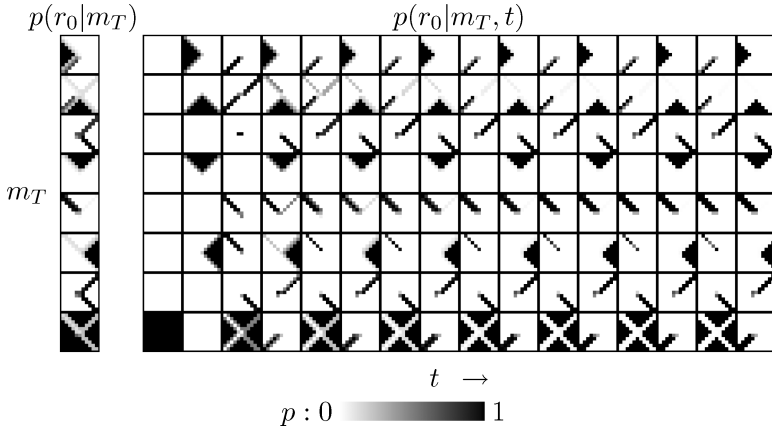


Figure 9: Time dependence of the representation of the initial position R_0 by the process $(M_t)_{t=0,1,\dots,15}$ of the best eight-state unconstrained controller. M_T is the readout of the process at an unknown time step T . Vertical axis: eight states of the controller; horizontal axis: time step (0 to 15). Darker cells code for higher probability of having started there. If the observation time step T is uniformly distributed, then knowing whether the time step is odd or even greatly helps determine what the controller's states mean in terms of initial position.

can be used to find out where the agent could have started. For example, in the case of the best evolved eight-state unconstrained controller (see Figure 9) knowing whether the time step is odd or even helps to significantly reduce the uncertainty in the agent's initial position given the state of the controller. The difference in how the two conditional distributions (time-less and time-dependent) describe the initial position can be used to quantify the dependence of the mapping $M_T \rightarrow R_0$ on the time step T .

Here the degree of time dependence of the representation is quantified as the Kullback-Leibler distance from the conditional distribution $p(r_0|m_T, t)$ to the conditional distribution $p(r_0|m_T)$:¹³

$$\begin{aligned} D_{p(m_T, t)}(p(r_0|m_T, t) || p(r_0|m_T)) &= H(R_0|M_T) - H(R_0|M_T, T) \\ &= I(R_0; T|M_T). \end{aligned}$$

¹³ The method for measuring time dependence of a mapping between two random variables is quite general. In fact, equation 7.2 applies to the general case of having three arbitrary random variables X, Y, Z and measuring the dependency of the mapping $X \rightarrow Y$ on Z :

$$\begin{aligned} D_{p(x, z)}(p(y|x, z) || p(y|x)) &= I(Y; Z|X) \\ I(Y; Z|X) + I(X; Z) &= I(X; Z|Y) + I(Y; Z). \end{aligned}$$

The distance can be interpreted as the average increase in the information about the initial position R_0 due to knowledge of the time step T when the controller's state M_T is known. For example, for the eight-state controller shown in Figure 9, the time dependence is 0.57 bits, assuming a uniform distribution for T .

The time-dependence measure is nonnegative. It is zero if and only if knowing the time step does not improve the knowledge about the initial position, that is, when the state of controller represents the initial position in a time-independent way. The higher the measure, the higher the time dependence of the representation.

The time-dependence measure is not symmetric with respect to R_0 and M_T . However, the dependence of $M_T \rightarrow R_0$ and its inverse, $R_0 \rightarrow M_T$ on T , is related:

$$I(R_0; T|M_T) + I(M_T; T) = I(M_T; T|R_0) + I(R_0; T). \quad (7.2)$$

Since the initial position of the agent is not dependent on time, the last term, $I(R_0; T)$, vanishes. The time dependence of the two representations and the amount of information captured by the controller about time ($I(M_T; T)$) are then related as

$$I(R_0; T|M_T) + I(M_T; T) = I(M_T; T|R_0). \quad (7.3)$$

See appendix F for further discussion of time dependence.

8 Factorization of Representations

8.1 Rationale. In section 5, it was mentioned that some of the representations of the agent's initial position by the state of the controller at the end of the run shown in Figure 6 are factorizable. In this letter, factorization of a representation means that the corresponding probabilistic mapping can be represented as a product of two or more component mappings, such that they capture independent pieces of the information captured by the original mapping, creating an orthogonal coordinate system. Another way of looking at the factorization of the representation of the initial position is to think of it as factorization of the initial position itself. However, only certain features of the initial position are factorized. The mapping between the initial position and the state of the controller at the end of the run defines what features are relevant. This is more in line with the relevance-based compression view adopted in the parallel bottleneck method (Friedman et al., 2001) discussed in more detail in appendix G.

8.2 Problem Statement. Assume two random variables X and Y with a given joint probability distribution $p(x, y)$. Y represents X in some way. The goal is to factorize this representation, that is, factorize Y with respect to X .

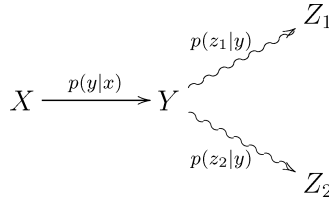


Figure 10: Factorization of Y with respect to X into Z_1 and Z_2 . X and Y are given. The task is to create two new variables, Z_1 and Z_2 , based on Y .

Y is to be split into two new random variables, Z_1 and Z_2 (see Figure 10), such that they meet the following criteria.¹⁴

First, the two new variables should capture as much information about X as possible. Hence, $I(Z_1, Z_2; X)$ should be as large as possible. Second, the two variables should comprise an orthogonal coordinate system for X . Hence, the two variables should be as independent as possible ($I(Z_1; Z_2)$ close to zero), and the conditional probability distribution $p(z_1, z_2|x)$ describing the projection of X onto Z_1 and Z_2 should be closely approximated by the product of partial projections $p(z_1|x)$ and $p(z_2|x)$. The distance between $p(z_1, z_2|x)$ and the product $p(z_1|x)p(z_2|x)$ can be measured as the Kullback-Leibler distance:

$$D_{p(x)}\left(p(z_1, z_2|x) \parallel p(z_1|x)p(z_2|x)\right) = I(Z_1; Z_2|X). \quad (8.1)$$

The distance should be as close to zero as possible. Note that although the distance is for conditional distributions, it still depends on the distribution of X .

Minimizing $I(Z_1; Z_2|X)$ makes Z_1 and Z_2 as conditionally independent given X as possible. This avoids synergetic or XOR-like coding schemes where, taken separately, Z_1 and Z_2 contain less information about X than when they are combined.¹⁵ (See section G.3 for more details.)

To summarize the task, given a joint distribution $p(x, y)$, find two conditional distributions $p(z_1|y)$ and $p(z_2|y)$ satisfying the following three criteria:

1. $I(Z_1, Z_2; X) \leq I(X; Y)$ is maximal.
2. $I(Z_1; Z_2) \geq 0$ is minimal.
3. $I(Z_1; Z_2|X) \geq 0$ is minimal (goal: $p(z_1, z_2|x) = p(z_1|x)p(z_2|x)$).

¹⁴ Factorizing into three or more variables places more constraints on the solution. Two variables are sufficient to start with. Later, each of the new variables can be factorized further if necessary.

¹⁵ *Synergy* colloquially refers to the case when the whole is more than the sum of its parts.

Appendix G discusses the details of the factorization method used in this letter, as well as the relation to synergy and the parallel information bottleneck method.

8.3 Factorization of the Representation of the Initial Position. The mappings $M_{15} \rightarrow R_0$ created by the best controllers evolved for capturing information about the initial position of the agent (see section 5) are factorized here. In terms of the factorization task, R_0 is X , and M_{15} is Y . Thus, the information about R_0 available in M_{15} was factorized into two new variables, Z_1 and Z_2 , by means of mappings $M_{15} \rightarrow Z_1$ and $M_{15} \rightarrow Z_2$. The original mappings, as well as mappings from Z_1 , Z_2 , and Z back to initial position R_0 , are shown in Figure 11.

The factorization of the representation of initial position used by the best-evolved eight-state gradient-following controller (see Figure 11, $|\mathcal{M}| = 8$) is a good example of the previous claims that these representations are factorizable. In the first factorization, the two new variables are constrained to be binary ($|Z_1| = |Z_2| = 2$). The first variable captures the initial position in terms of being left or right from the source. The second variable captures the position roughly in terms of being on either white or black cells of a checkerboard pattern. Note that there is a vertical discontinuity in the pattern in the middle.

The factorization is not at the ideal point ($d = (0.008, 0.000, 0.000)$) (see appendix G) because 0.008 bit of the information about R_0 captured by M are not captured by the new variables. To capture this residual information, one of the new variables was grown to have three states. The resulting best factorization is shown in the middle of Figure 11 ($8 \rightarrow 3 \times 2$). There only 0.001 bit of original information is lost. The representation of the initial position employed by the second variable is exactly the same as used by the first variable in the previous case. However, the first variable uses a proper checkerboard coordinate system, without the vertical fault in the middle.

The controllers were originally evolved to capture at the end of the run as much information about the agent's initial position as possible. Alternatively, this task may be seen as establishing a temporally extended information flow from R_0 to M_{15} . Factorization of the initial position through the state of the controller at the end of the run can be seen as splitting the information flow, albeit only at the end, into two or more independent components. Later, each component can be processed separately.

Such factorization has several advantages. First, the state space of the variables representing information is reduced due to compression. Second, information is split into independent parts or features. The state spaces of features are normally smaller than the state space of the original variables. Compact representation may allow for more efficient processing, requiring less space to store and transmit. Third, the fact that individual components are independent of each other allows for their parallel independent processing. At a later stage, the results of the parallel processing can be again

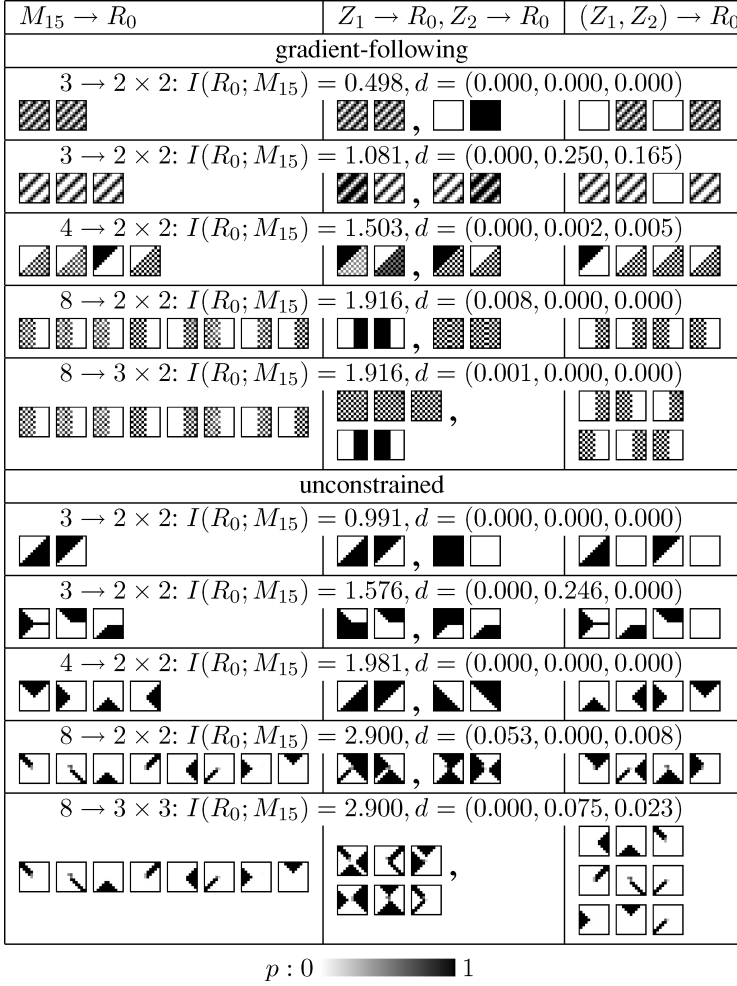


Figure 11: Factorization of information about the initial position captured by the best-evolved two-, three-, four-, and eight-state controllers into pairs of binary (2×2) or ternary (3×3) variables. Mapping $M_{15} \rightarrow R_0$ is factorized into $Z_1 \rightarrow R_0$ and $Z_2 \rightarrow R_0$, and then reconstructed as $(Z_1, Z_2) \rightarrow R_0$. Each box shows the mapping from a state (m or z) onto 11×11 cells of initial position. The darker the cell, the higher the probability.

combined. Moreover, if individual components can be factorized further, they could serve as starting points for hierarchies.

8.4 Factorization of Representations of Time. In this section, the information about time available to the agent is factorized. To be more specific,

the information about T as seen through $(S, M)_T$ is factorized. T is assumed to be uniformly distributed over $\{0, 1, \dots, 15\}$. In terms of the original definition of the factorization task, T is X , and $(S, M)_T$ is Y . $(S, M)_T$ can be thought of as containing information available to the agent's controller.

The amount of information about time contained in internal state $(S, M)_T$ of the best-evolved gradient-following controllers is quite small. The best factorizations into two binary variables are shown in Figure 12a. Most of the resolution of the new variables is concentrated on the first half of the run. Because of the very low level of information about time in $(S, M)_T$, the factorized variables contain even less information. Consequently, it is hard to interpret them.

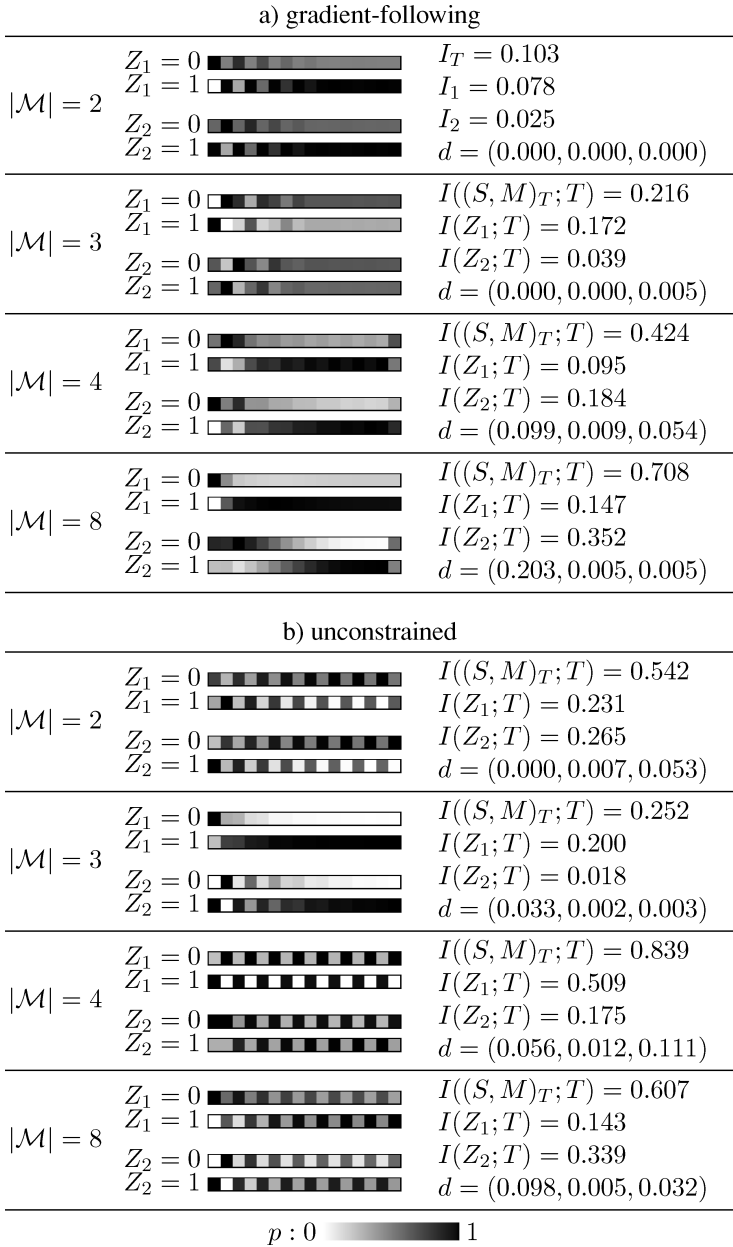
Factorizations of the information about time in $(S, M)_T$ of the best-evolved unconstrained controllers produce features distinguishing mostly between odd and even time steps (see Figure 12b), although in the case of the best three-state controller the first feature distinguishes between the beginning and the rest of the run.

8.5 Discussion. Why is factorization of the agent's initial position as seen through the state of the controller at the end of the run (see section 8.3) much more successful than factorization of time as seen through the agent's internal state and the sensor (see section 8.4)?

To quickly rehash the structure of the factorization problem, random variables X and Y are given, and two new random variables, Z_1 and Z_2 , are individually mapped from Y so that they capture independent pieces of information about X . The factorization method used in this section can be approximately thought of as consisting of two stages. First, similar to the information bottleneck principle, it compresses the information contained in Y about X into new variables, Z_1 and Z_2 , which have fewer states than Y , hence resulting in a compact representation of the information. Second, factorization splits this information into two independent variables.

The initial position of the agent as seen through the state of its controller at the end of the run was factorized first (see section 8.3). The controllers were evolved so that the state of the controller captured as much information about the initial position as possible. Thus, the state of the controller at the end of the run already is a compact, compressed representation of the initial position. Factorization simply represents the captured information in terms of two independent variables. In most cases the results of the factorization algorithm were very close to the ideal point.

The factorization of the information about time as seen through the sensor-controller state tuple (see section 8.4) was much further away from the ideal point. In most cases the algorithm failed to capture a significant share of the information about time that was to be factorized. The explanation may be quite simple. Although the amount of information about time captured by the agent's internal state or the system's global state is not high, the information cannot be squeezed into two binary variables. It is



“smeared” over a large number of states ($|T| \cdot |S| \cdot |M|$) and, being an average quantity, cannot be extracted without noticeable loss into two binary variables. This hypothesis was confirmed by relaxing the factorization constraints and simply trying to extract as much of the information as possible into just four states. The amount of the extracted information is a lower bound for the distance from the ideal point the factorization algorithm can achieve. Thus, the original factorization may have been less successful simply because the controllers were not evolved to capture information about time in a convenient compressed way.

The factorization of the highly compressed information about the initial position offers the advantage of viewing the captured features in terms of two smaller independent features. The case of the best evolved eight-state gradient-following controller in Figure 11 is a good example. There, factorization creates two natural variables: left versus right and black versus white cell of the checkerboard.

When factorizing the relatively scarce and widely “smeared” information about time, the most useful feature of factorization is that relevance-based compression is performed. This projects the information about time from a high number of states to just a few. These are easier to interpret. For example, it becomes obvious that the two most common features of time captured are roughly (1) whether the time step is odd or even and (2) whether roughly half of the run has elapsed.

9 Conclusion

An information-theoretic approach to analysis and construction of perception-action loops was presented. The approach enables measuring information flows between various parts of the loop through time, constructing perception-action loops with desired properties, and analyzing the resulting behavior at the level of information.

To show the approach in action, a two-dimensional grid world with an agent controlled by a finite state automaton and having access only to a gradient sensor was presented. The agent’s sensor captures partial information about the agent’s position. It was shown that it is possible to capture more of the information by integrating sensoric input over time and employing an evolved movement strategy.

Figure 12: Factorization of internal state $(S, M)_T$ of best-evolved controllers with respect to time. Each section shows the pair of mappings $Z_1 \rightarrow T$ and $Z_2 \rightarrow T$ created from $(S, M)_T \rightarrow T$. Four boxes show the probabilistic mappings described by $p(t|z_1)$ and $p(t|z_2)$ from states of Z_1 and Z_2 onto the 16 states of time t (left to right, from 0 to 15). Darker color codes for higher probability. Since new variables Z are binary, $p(t|Z_n = 1) = 1 - p(t|Z_n = 0) \forall t \forall p$.

Controllers with a limited amount of memory were evolved to capture as much information as possible about the agent's initial position under the constraints of the agent's embodiment. To use the memory efficiently, the controllers performed lossy compression, which resulted in near-hard partitioning of the position with respect to the controller's state.

From the information-theoretic perspective, these controllers were evolved to create a temporally extended communication channel of maximum bandwidth between two temporally separate parts of the system: the agent's position at the beginning of the run and the controller's state at the end of it. The channel is created through interaction of the agent with the environment. Compression, partitioning, and the movement strategy employed by the controllers are all induced by maximization of information transfer (infomax) through the perception-action loop through time. This is related to results for neural networks, where information transfer maximization resulted in phenomena such as feature detectors (Linsker, 1988), factorial codes (Nadal & Parga, 1995; Bell & Sejnowski, 1995), and source separation (Bell & Sejnowski, 1995), but is more general, since it allows arbitrary temporally extended channels and removes assumptions about the internal structure of the underlying information processing.

In the second part of this letter, a way to use information-theoretic tools to analyze representations of space (initial position) and time by the best-evolved controllers was presented. The intention was twofold: to provide more insight into what information the controllers capture and to demonstrate the power of several information-theoretic tools.

The representations of the agent's initial position by the state of controllers at the end of the experiment were studied. The representations reflect some properties of the agent's embodiment, the interaction and constraints of the world, and the agent's sensorimotor apparatus. Slight changes to the agent's actuator produce similarly slight changes in the representations. In some cases, even old representations are suitable for a slightly changed actuator. A method for factorizing representations was demonstrated. It turned out that some of the representations factorize nicely into smaller independent features. Factorized features, being independent by definition, can serve as starting points for hierarchies if factorized further.

The amount of information captured by the agent about time was quantified by treating time as a random variable, and the representation of time in the agent's controller was studied. A method for information theoretically measuring the time dependence of representations was demonstrated, linking time dependence with the amount of information captured about time. Last but not least, the representations of time were factorized to understand what is captured. The most common features of time captured are (1) whether the time step is odd or even and (2) whether the first half of the run has elapsed.

The applications of information-theoretic tools to the perception-action loop are quite promising. They allow one to analyze information processing

even when one does not understand the underlying substrate. For example, there is no requirement to assume finely grained components (e.g., neurons) that perform the information processing. The model of computation is thus more abstract and accommodating than that of traditional infomax. The various information-theoretic tools presented here can provide a peek into systems that do information processing in a nonobvious way.

Viewing information as the currency or essence of computation in the perception-action loop provides a plausible mechanism, where a few assumptions and constraints, such as embodiment with various bottlenecks coming from the limited capacity of sensory and motor channels, and limited memory, can give rise to structures in a self-organized way. This paves the way toward a principled and general understanding of the mechanisms guiding the evolution of sensors, actuators, and information processing substrates in nature and provides insights into the design of mechanisms for artificial evolution.

The main contributions of this letter are (1) building up an information-theoretic picture of the perception-action loop with a minimal set of assumptions, (2) maximizing information flow through the full perception-action loop of an agent through time, as opposed to the well-established maximization of the flow through layers in neural networks, and (3) introducing the use of information-theoretic tools to analyze the representations arising from the maximization of information flow through the perception-action loop.

Appendix A: Information Theory

In this section, the central information-theoretic notions used in this letter are briefly introduced. For a detailed introduction to information theory (Shannon, 1948), consult, for example, Cover and Thomas (1991).

A random variable can assume various values with various probabilities. In this letter, exclusively discrete random variables are considered. Denote random variables with uppercase letters (e.g., X), their sets of values with calligraphic letters (e.g., \mathcal{X}), and their values with lowercase letters (e.g., x). Denote composite random variables by listing their elements inside parentheses (e.g., (X, Y)) with values (x, y) from the set $\mathcal{X} \times \mathcal{Y}$. By abuse of notation, denote $Pr(X = x)$, the probability that X assumes the value x , by $p(x)$. Similarly, the joint probability of X and Y is denoted by $p(x, y)$ and the conditional probability of X given Y by $p(x|y)$.

The entropy of X , denoted by $H(X)$, is defined as a measure of the uncertainty of the probability distribution of X :

$$H(X) := - \sum_{x \in \mathcal{X}} p(x) \log_2 p(x).$$

Entropy as well all other information-theoretic measures used in this letter are measured in bits. Note that all of the information-theoretic measures presented in this appendix are nonnegative.

The conditional entropy of X given Y , denoted $H(X|Y)$, is defined as the uncertainty of X knowing Y weighted by the probability of Y :

$$\begin{aligned} H(X|Y) &:= \sum_{y \in \mathcal{Y}} p(y) H(X|Y = y) \\ &= - \sum_{y \in \mathcal{Y}} p(y) \sum_{x \in \mathcal{X}} p(x|y) \log_2 p(x|y). \end{aligned}$$

The mutual information between X and Y , denoted $I(X; Y)$, is defined as the average reduction in the uncertainty of X given Y :

$$I(X; Y) := H(X) - H(X|Y).$$

Kullback-Leibler distance (Kullback & Leibler, 1951) from a distribution $p_1(x)$ to a distribution $p_2(x)$, denoted $D(p_1||p_2)$, is a generalized measure of distance between the two distributions:

$$D(p_1||p_2) = \sum_{x \in \mathcal{X}} p_1(x) \log_2 \frac{p_1(x)}{p_2(x)}.$$

$D(p_1||p_2) = 0$ if and only if p_1 is identical to p_2 . The measure can assume infinity if there exists x for which $p_1(x) \neq 0$ and $p_2(x) = 0$. Kullback-Leibler distance is not actually a distance in the mathematical sense, in particular since it is not symmetric with respect to p_1 and p_2 .

Kullback-Leibler distance from a conditional distribution $p_1(x|y)$ to a conditional distribution $p_2(x|y)$ is defined as the sum of Kullback-Leibler distances from $p_1(x|Y = y)$ to $p_2(x|Y = y)$ weighted by the probability of Y :

$$\begin{aligned} D_{p(y)}(p_1||p_2) &= \sum_{y \in \mathcal{Y}} p(y) D(p_1(x|Y = y)||p_2(x|Y = y)) \\ &= \sum_{y \in \mathcal{Y}} p(y) \sum_{x \in \mathcal{X}} p_1(x|y) \log_2 \frac{p_1(x|y)}{p_2(x|y)}. \end{aligned}$$

Appendix B: Bayesian Network Model of the Agent's Controller _____

The agent's controller selects actions based on sensoric input and its own memory. The assumptions about the agent's controller are minimal: the controller is finite time and finite state. Given the sets of sensoric states \mathcal{S} , actions \mathcal{A} , and the memory states of the controller \mathcal{M} , any such controller

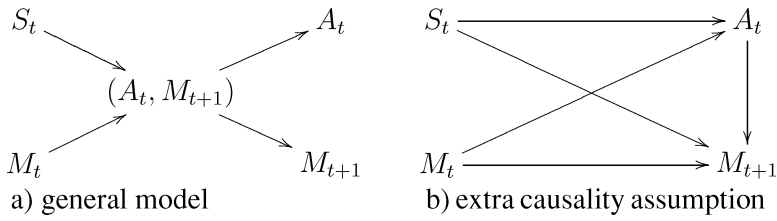


Figure 13: Causal Bayesian network models of an agent's controller. S —state of the sensor; A —action performed by the actuator; M —state of the memory of the controller. (a) General model. (b) Model with an extra causality assumption. The new state of memory is influenced by the choice of action.

can be viewed as performing a probabilistic mapping $S \times \mathcal{M} \rightarrow \mathcal{A} \times \mathcal{M}$. The interpretation is that the controller picks an action and the next state of its memory based on the current sensoric input and the current state of its memory.

The causal Bayesian network model for the general case of any such probabilistic mapping is as shown in Figure 13a. To simplify the diagrams, in the main body of the letter a different model shown in Figure 13b was used. This model, although simpler to understand at first glance, contains an extra assumption that the new state of memory can be influenced by the choice of action. In general, this need not be the case. For example, it is easy to envisage a controller where it is the action that is influenced by the new state of memory. The general model, shown in Figure 13a, on the other hand, does not assume any particular causality between actions and the new state of the controller's memory.

Appendix C: Measuring Information Flows

The goal of the information capture experiment in section 4 is to evolve agent controllers that maximize the information flow from the agent's initial position to the memory of the controller at a later time. To calculate the amount of information flow, the joint probability distribution of the parts of the system that are the source and the destination of information flow, is required. Here the relevant parts are the agent's initial position R_0 and M_{15} , the state of memory of the agent's controller at time step 15.

Although the joint distribution could be obtained from the Bayesian network analytically, here Monte Carlo simulations are used for estimating the probability distributions. The system is started in an initial state drawn from the probability distribution of the initial state (see section 4.4 for an example), then iterated for the required number of time steps after which data of interest are gathered. The process is repeated until the estimates of probability distribution of interest are stable enough.

To spot and avoid undersampling, for each of the possible initial states of the system, 256 or more samples are produced depending on the quantities measured. The number of samples is then increased by a factor of at least 16 to check whether the quantities of interest remain stable.

Appendix D: Method of Evolution

In the information capture experiment in section 4, evolution is used as a search in the space of controllers. The aim is not to develop new evolutionary computation techniques but to search for controllers with the maximum information flow from the agent's initial position to its controller's memory at a fixed time step. To do that in a straightforward, transparent, and unbiased way, a minimal setup is employed. This is to emphasize that nothing in the general approach is specific to the particular model or the search methods employed.

The controller is evolved. The controller is described by a probabilistic time-invariant mapping $(S_t, M_t) \rightarrow (A_t, M_{t+1})$ (see section 4.3). Evolving the controller means evolving the mapping for fixed sizes of state (memory), sensoric input, and action sets.

The search is limited to deterministic mappings only. To any pair of sensoric input and state (s_t, m_t) there corresponds exactly one action and new state (a_t, m_{t+1}) . A mutation is performed picking a pair of (s_t, m_t) at random with uniform probability from the set $\mathcal{S} \times \mathcal{M}$ and setting its image to a pair (a_t, m_{t+1}) picked at random with uniform probability from the set $\mathcal{A} \times \mathcal{M}$. This type of mutation is unbiased in the sense that it is not designed for a particular fitness function. *Mutation is the only evolutionary operator employed to generate variation.*

Fitness is evaluated by letting the controller control the agent. The fitness function is the amount of information flow from the agent's initial position R_0 to the memory of the controller M_t at a later time step. The population is initialized with five randomly generated controllers. After each generation, five best controllers are kept in the population and produce five offspring, each by mutation. Thus, the size of the population is between 5 and 30.

To speed up the search, ideas from integer programming (Glover, 1986) are used: (1) random shakeup—the number of mutations applied to each offspring is uniformly distributed between 1 and $1 + (G \bmod 20)$, where G is the generation, and (2) permanent tabu list—offspring controllers that have been evaluated before or are present in the population are discarded. Additionally, at least five separate evolutionary runs are performed to sample different solutions.

Appendix E: Gradient Following as Initial Position Capture

This section proves the assertion in section 4.4 that when the agent follows the gradient, most of the information about the agent's initial position passes through its sensors.

Denote the agent's initial position with a random variable R_0 . The entropy of initial position is $H(R_0)$. This is the maximum amount of information there is to know about the initial position.

Assume that the agent follows the gradient for t time steps. Denote the agent's position after t time steps with a random variable R_t .

Denote the sequence of sensoric input up to but excluding the time step t with a random variable $S_{t-1}^c = (S_0, S_1, \dots, S_{t-1})$.

The goal is to prove that

$$I(S_{t-1}^c; R_0) = H(R_0) - \epsilon, \quad (\text{E.1})$$

where ϵ is a small quantity.

Denote the sequence of actions the agent performed up to but excluding the time step t with a random variable $A_{t-1}^c = (A_0, A_1, \dots, A_{t-1})$. Using standard information-theoretic equalities, the amount of information the sequence of actions A_{t-1}^c contains about the agent's initial position R_0 can be expressed as

$$I(A_{t-1}^c; R_0) = H(R_0) - I(R_0; R_t | A_{t-1}^c) - H(R_0 | R_t, A_{t-1}^c). \quad (\text{E.2})$$

All actions move the agent deterministically over the grid. Given a final position r_t , the sequence of actions performed, a_{t-1}^c , identifies the exact starting position r_0 . This means that the initial position is a deterministic function of the final position and the sequence of actions:

$$H(R_0 | R_t, A_{t-1}^c) = 0. \quad (\text{E.3})$$

Equation E.2 becomes

$$I(A_{t-1}^c; R_0) = H(R_0) - I(R_0; R_t | A_{t-1}^c). \quad (\text{E.4})$$

When the agent follows the gradient, it does not end up exactly at the source. Because there is no "do nothing" action, once the agent is at the source, it cannot stay there and moves into one of the four adjacent cells at the next time step. One time step later, it comes back following the gradient. Thus, after a certain amount of time, the agent cycles through two states: (1) being in the center and moving in one of the four randomly chosen directions and (2) being in one of the four cells adjacent to the central cell and moving back to the central cell.

Due to this fact, given enough time, the tails of sequences of actions will consist of a repeating pattern: a random action, then the action opposite to it—for example, $a_i^c = (\dots, \uparrow, \downarrow, \rightarrow, \leftarrow, \uparrow)$. When the agent is at the central cell, it enters the loop of performing an action followed by its counteraction. When not in the central cell, the agent performs an action followed by its

counteraction only in one situation: when the agent is in a cell adjacent to the center.

The following two sequences are examples of what the agent can perform if it starts in the cell adjacent from the left to the central cell: $(\rightarrow, \leftarrow, \rightarrow, \leftarrow, \rightarrow, \leftarrow)$, $(\rightarrow, \leftarrow, \rightarrow, \uparrow, \downarrow, \uparrow)$. The first sequence consists of the same repeating tuple of action and its counteraction. In principle, the same sequence of actions can be performed by the agent starting from the central cell. The second sequence of action cannot, because the agent must follow the gradient consistently. Thus, only a tuple of action and its counteraction followed by a tuple of a different action and its counteraction signify that the agent has reached the center. Sequences of one and the same tuple of action and its counteraction do not mean that the agent has reached the center. However, the probability of such sequences decreases exponentially with their length.

The above observation means that the higher the t , the more the sequence of actions determines where the agent finished, and, hence, where it started. This means that $H(R_0|A_{t-1}^c)$ tends to zero. Since

$$I(R_0; R_t|A_{t-1}^c) = H(R_0|A_{t-1}^c) - H(R_0|A_{t-1}^c, R_t) \quad (\text{E.5})$$

$$\leq H(R_0|A_{t-1}^c), \quad (\text{E.6})$$

it follows that

$$I(R_0; R_t|A_{t-1}^c) \rightarrow 0 \quad (\text{E.7})$$

and from equation E.4,

$$I(A_{t-1}^c; R_0) \rightarrow H(R_0). \quad (\text{E.8})$$

When the agent follows the gradient, its actions are completely determined by the sensoric input. The sequence of actions is in one-to-one correspondence with the sequence of sensoric input. Hence, the sequence of actions contains the same amount of information about the initial position as the sequence of sensoric input:

$$I(S_{t-1}^c; R_0) = I(A_{t-1}^c; R_0). \quad (\text{E.9})$$

Thus,

$$I(S_{t-1}^c; R_0) \rightarrow H(R_0). \quad (\text{E.10})$$

with increasing t .

Appendix F: Examples of Time Dependence

F.1 Clock Example. It is possible to imagine a case where the mapping $M_T \rightarrow R_0$ is time independent (first term in equation 7.3 is 0), whereas the reverse mapping $R_0 \rightarrow M_T$ is time dependent. Assume a controller with two substates: $M = (M^{pos}, M^{clock})$. Substate M_T^{pos} permanently encodes some information about the agent's initial position (m^{pos} is constant in time), and substate M_T^{clock} captures some information about time, for example, counts modulo some number.

The combined state (M^{pos}, M^{clock}) of the controller will be changing with time because M^{clock} does. Nevertheless, the mapping $(M^{pos}, M^{clock})_T \rightarrow R_0$ will be time independent, with all pairs $(m_T^{pos}, *)$ mapping to the same subset of initial position. The reverse mapping $R_0 \rightarrow (M^{pos}, M^{clock})_T$ will be time dependent because the substate M^{clock} changes with time. Knowing only the initial position predicts only M_T^{pos} . Without knowledge of time, M_T^{clock} remains completely random. In this scenario, the time dependence of $R_0 \rightarrow (M^{pos}, M^{clock})_T$ is exactly the amount of information about time T contained in M_T^{clock} .

$$I((M^{pos}, M^{clock})_T; T | R_0) = I((M^{clock})_T; T). \quad (F.1)$$

F.2 XOR Example. At the other extreme of the time-dependence scale lies the case where both mappings, $M_T \rightarrow R_0$ and $R_0 \rightarrow M_T$, are time dependent, though the state of the controller does not contain any information about time. Consider any two-state controller that is started in state 1 if the agent is above the signal source, and in state 2 otherwise. At each time step, regardless of the sensoric input, the controller permutes its state: $1 \rightarrow 2$, $2 \rightarrow 1$.

In this case, the state of the controller contains no information about time $I(M_T; T) = 0$. For instance, the fact that the state is 1 means that the agent started above the source provided the time step is odd, and that the agent started elsewhere provided the time step is even. Thus, without knowledge of the time step, the state of the controller does not tell anything about the initial position of the agent.

The coarsely grained initial position (above the source or not), time step T modulo 2, and state of the controller M are in an XOR relationship. Knowing only one of the variables does not provide information about any of the other two. Knowing the states of any two variables provides full information about the third. The mapping $M_T \rightarrow R_0$ maps every state m_T uniformly onto initial position \mathbb{Z}^2 . However, with the knowledge of time, the mapping becomes more deterministic, mapping state 1 onto one half of \mathbb{Z}^2 and state 2 onto the other. Similarly, the reverse mapping $R_0 \rightarrow M_T$ is made completely deterministic by knowing whether the current time step is odd or even. Agreeing with

equation 7.3, the time dependence of both mappings is identical since $I(M_T; T) = 0$:

$$I(R_0; T|M_T) = I(M_T; T|R_0) = 1. \quad (\text{F.2})$$

F3 Factoring Time Out. In the light of the two examples above, it is possible to give an interpretation to equation 7.3, linking time dependence of the mappings between R_0 and M_T and the amount of information about time contained in M_T . The state of the controller M_T can be thought of as containing two pieces of information about time. One is available without reference to any other variable; its amount is quantified by $I(M_T; T)$. The other is “encrypted” through R_0 . Knowledge of R_0 provides $I(R_0; T|M_T)$ more bits of information about time. The total amount of information about time extractable from M_T with the help of R_0 is then $I(M_T; T|R_0)$.

One could attempt to factor M_T into two variables: one containing information about time, the other containing information about the rest of M_T . In the clock example above, this type of factorization is easy. The new variables correspond to M_T^{clock} and M_T^{pos} by definition. In the XOR example, though, it is not possible to split M_T due to the fact that all information about time in M_T is “encrypted” through R_0 and is thus not “freely” available without the knowledge of R_0 . The term $I(M_T; T)$ in equation 7.3 is the upper bound to the amount of information about time that can be factored out of M_T .

Factoring out or removing information about time from M_T reduces the time dependence of the $R_0 \rightarrow M_T$ mapping by exactly the amount of information about time removed. However, this still does not guarantee that the mapping will become time independent. At best the time dependence of $R_0 \rightarrow M_T$ can be lowered to the time dependence of its inverse, $M_T \rightarrow R_0$. However, the time dependence of the latter cannot be influenced by removing information about time from M_T .

Appendix G: Factorization: Details

G.1 Ideal Solutions. An ideal situation is when (1) all of the information about X contained in Y is captured by (Z_1, Z_2) , (2) Z_1 and Z_2 are independent, and (3) the mapping $X \rightarrow (Z_1, Z_2)$ is the product of the mappings $X \rightarrow Z_1$ and $X \rightarrow Z_2$. The ideal point in terms of the left-hand side values of the three criteria above is thus $(I(X; Y), 0, 0)$.

It should be noted that in case \mathcal{Z}_1 or \mathcal{Z}_2 is large enough to accommodate all the states of \mathcal{Y} , it is possible to create a trivial factorization, such as where $Y \rightarrow Z_1$ is information preserving and $Y \rightarrow Z_2$ preserves no information about Y . This trivial factorization meets the ideal point criterion but is not interesting. A similarly trivial factorization is known for numbers: any number can be factorized into the product of itself and 1. To avoid trivial factorization, both $|\mathcal{Z}_1|$ and $|\mathcal{Z}_2|$ were made smaller than $|\mathcal{Y}|$.

G.2 The Method. Each solution is a tuple of conditional distributions $p(z_1|y)$ and $p(z_2|y)$. Comparing different solutions requires trading off between the three criteria above. As there is no known precise method for finding an optimal factorization, the following method was used.

When the number of states for the two new random variables Z_1 and Z_2 is given, the goodness or fitness of a tuple is defined in terms of how well it satisfies the three criteria above. The ideal situation is where all of the information about X contained in Y is captured by Z_1 and Z_2 , and the left-hand side of the other two criteria is zero. The fitness is defined as the Manhattan distance in the criteria space from the ideal point,

$$d = (c - I(Z_1, Z_2; X)) + I(Z_1; Z_2) + I(Z_1; Z_2|X), \quad (\text{G.1})$$

where $c = \min(I(X; Y), \log_2 |Z_1||Z_2|)$ is the maximum amount of information that Z_1 and Z_2 could in principle capture about X . c is a constant for all tuples with same sizes of Z_1 and Z_2 . The goal is to find a tuple with minimal distance from the ideal point.

Here hill climbing is employed in the space of the tuples of mappings $(p(z_1|y), p(z_2|y))$ to search for the best tuple. The search is restricted to deterministic mappings. The best solution is chosen from five independent runs of the hill climbing search. Each search run is started with a randomly chosen tuple and run for 50,000 steps. At each step, the tuple is modified $1 + (t \bmod 20)$ times, where t is the step. A modification is performed by randomly choosing one of the two mappings and then deterministically mapping a randomly chosen state y to a randomly chosen state z_1 or z_2 , depending on which of the two mappings is modified. If after the modifications the new tuple is closer to the ideal point than the old one, the tuple is used as a base for the next step; otherwise, the new tuple is discarded, and the old one is used at the next step.

G.3 Relation to Synergy. *Synergy* colloquially refers to the case when the whole is more than the sum of its parts. In neural networks, synergy refers to an ensemble of neurons providing more information about a stimulus than the sum of the information provided by neurons individually (Brenner, Strong, Koberle, Bialek, & de Ruyter van Steveninck, 2000; Schneidman et al., 2003).

Synergy can also be applied to the factorization problem presented in section 8. The total amount of information about X contained in $Z = (Z_1, Z_2)$ can be expressed as

$$I(Z; X) = I(Z_1; X) + I(Z_2; X) + I(Z_1; Z_2|X) - I(Z_1; Z_2).$$

$I(Z_1; X)$ and $I(Z_2; X)$ is the amount of information about X contained in Z_1 and Z_2 , respectively. The extra term $I(Z_1; Z_2|X) - I(Z_1; Z_2)$, called synergy

(Brenner et al., 2000; Schneidman et al., 2003),

$$\begin{aligned} Syn(Z_1, Z_2) &= I(Z_1, Z_2; X) - I(Z_1; X) - I(Z_2; X) \\ &= I(Z_1; Z_2|X) - I(Z_1; Z_2), \end{aligned}$$

can be seen as the amount of information about X that is “jointly” contained in Z_1 and Z_2 in an “encrypted” form and can be extracted only when both variables are known. Synergy can be negative. When synergy is negative, information about X is contained in Z_1 and Z_2 in a redundant way.

An extreme example of synergetic representation of information is the binary XOR case, when $x = z_1 \oplus z_2$ and Z is uniformly distributed. $I(Z_1; X) = I(Z_2; X) = 0 = I(Z_1; Z_2) = 0$ bit, whereas $I(Z; X) = I(Z_1; Z_2|X) = Syn(Z_1, Z_2) = 1$ bit. Knowing either Z_1 or Z_2 does not tell anything about X , whereas knowing both Z_1 and Z_2 tells everything.

Conditions 2 and 3 of the factorization task minimize $I(Z_1; Z_2)$ and $I(Z_1; Z_2|X)$, respectively. When these two terms reach zero, synergy also becomes zero. In this case, the resulting representation of X by Z_1 and Z_2 is neither synergetic nor redundant. This is exactly what is desired, since Z_1 and Z_2 should be independent components or orthogonal coordinates of X .

G.4 Relation to Parallel Information Bottleneck Method. The factorization task is similar in spirit to the parallel bottleneck case of the multivariate information bottleneck (Friedman et al., 2001). In this section, the similarities and the differences between the two tasks are pointed out briefly. To facilitate easier comparison, this letter’s notation for the parallel bottleneck problem is used instead of the original notation from Friedman et al. (2001). The random variable $Z = (Z_1, Z_2)$ is introduced to save space.

The parallel bottleneck problem is stated as following. Given a joint distribution $p(x, y)$, find two conditional distributions $p(z_1|y)$ and $p(z_2|y)$ such that the two random variables Z_1 and Z_2 compress Y in “parallel,” predicting X as much as possible. The Bayesian network of relations between the random variables X, Y, Z_1 and Z_2 is identical to the Bayesian network from the factorization problem in Figure 10.

The quality of a solution is quantified by the following Lagrangian, which should be minimized in the space of tuples $(p(z_1|y), p(z_2|y))$ to find a good solution,

$$\mathcal{L}^{(1)} = -\beta I(Z; X) + I(Z_1; Z_2) + I(Z; Y), \quad (\text{G.2})$$

where β is a trade-off parameter that specifies the relative importance of preserving information about X .

The parallel bottleneck problem is very much similar to the factorization problem described in this article. The relations between the random variables are identical. The goal is almost identical too. Here the aim is to

factorize the information about X captured by Y into new variables Z_1 and Z_2 . In parallel bottleneck, this requirement is less explicit.

Both methods express the goodness of a solution (a tuple of mappings $p(z_1|y)$ and $p(z_2|y)$) in terms of a single measure. Here it is the Manhattan distance from the ideal point (see equation G.1), and in the parallel bottleneck case, it is the Lagrangian (see equation G.2). Without the constant term c , the distance specified in equation G.1 is

$$d = -I(Z; X) + I(Z_1; Z_2) + I(Z_1; Z_2|X). \quad (\text{G.3})$$

X, Y , and Z form a Markov chain $X \rightarrow Y \rightarrow Z$ (see Figure 10) and therefore $I(Z; Y) = I(Z; X) + I(Z; Y|X)$. Hence, the Lagrangian from equation G.2 can be rewritten as

$$\mathcal{L}^{(1)} = -(\beta - 1)I(Z; X) + I(Z_1; Z_2) + I(Z; Y|X). \quad (\text{G.4})$$

Now two differences between the factorization and the parallel bottleneck are apparent. First, the factorization method in this article can be seen as using a constant parallel bottleneck $\beta = 2$. Second, the last term in the two equations is different. The latter needs closer attention.

$I(Z_1; Z_2|X) = 0$ requires that Z_1 and Z_2 are conditionally independent given X . This avoids synergetic, XOR-like coding schemes where Z_1 and Z_2 individually cannot be used to obtain information about X , where only their combination is helpful. $I(Z; Y|X) = 0$ requires that Z and Y share no information other than information about X . In other words, this requires that Z_1 and Z_2 contain information only about X .

The two requirements are different. However, they are related. Keeping in mind that $Z = (Z_1, Z_2)$ and that Z_1 and Z_2 are obtained from Y using independent mappings,

$$I(Z; Y|X) = I(Z_1; Y|X) + I(Z_2; Y|X) - I(Z_1; Z_2|X). \quad (\text{G.5})$$

Minimizing only $I(Z_1; Z_2|X)$ allows Z_1 and Z_2 to contain “irrelevant” information—information about Y that is not information about X . Parallel bottleneck, on the other hand, attempts to forbid such irrelevant information.

Interestingly, $I(Z; Y|X) \geq I(Z_1; Z_2|X)$.¹⁶ As a result of nonnegativity of mutual information, if $I(Z; Y|X) = 0$, then $I(Z_1; Z_2|X) = 0$. Hence, a

¹⁶ $I(Z; Y|X) = I(Z_1; Y|X) + I(Z_2; Y|X) - I(Z_1; Z_2|X)$. Since $I(Z_1; Z_2|X) \leq I(Z_1; Y|X)$ and $I(Z_1; Z_2|X) \leq I(Z_2; Y|X)$:

$$\begin{aligned} I(Z; Y|X) &\geq I(Z_1; Z_2|X) + I(Z_1; Z_2|X) - I(Z_1; Z_2|X) \\ &\geq I(Z_1; Z_2|X). \end{aligned}$$

minimum of the parallel bottleneck Lagrangian in the case of $I(Z; Y|X) = 0$ is also an optimal solution for the factorization method in this article.

An important advantage of the multivariate bottleneck method, of which parallel bottleneck is just an example, is that it provides an iterative optimization algorithm for finding a good solution. However, the algorithm is designed only for certain types of Lagrangians. The distance d (see equation G.3) used in this article seems to be unsuitable for the algorithm.

Acknowledgments

We thank the Condor Team from the University of Wisconsin, whose high-throughput computing system Condor provided a convenient way to run large numbers of simulations on ordinary workstations.

References

- Ashby, W. R. (1952). *Design for a brain*. New York: Wiley.
- Ashby, W. R. (1956). *An introduction to cybernetics*. London: Chapman & Hall.
- Attneave, F. (1954). Some informational aspects of visual perception. *Psychological Review*, 61(3), 183–193.
- Avraham, H., Chechik, G., & Ruppín, E. (2003). Are there representations in embodied evolved agents? Taking measures. In W. Banzhaf, T. Christaller, P. Dittrich, J. T. Kim, & J. Ziegler (Eds.), *Proceedings of the 7th European Conference on Artificial Life (ECAL)* (pp. 743–752). Berlin: Springer-Verlag.
- Ay, N. (2002). Locality of global stochastic interaction in directed acyclic networks. *Neural Computation*, 14(12), 2959–2980.
- Ay, N., & Krakauer, D. C. (2007). Geometric robustness theory and biological networks. *Theory in Biosciences*, 125(2), 93–121.
- Ay, N., & Wennekers, T. (2003). Dynamical properties of strongly interacting Markov chains. *Neural Networks*, 16(10), 1483–1497.
- Barlow, H. B. (1959). Possible principles underlying the transformations of sensory messages. In W. A. Rosenblith (Ed.), *Sensory communication: Contributions to the Symposium on Principles of Sensory Communication* (pp. 217–234). Cambridge, MA: MIT Press.
- Barlow, H. B. (1963). The coding of sensory messages. In W. H. Thorpe & O. L. Zangwill (Eds.), *Current problems in animal behaviour* (pp. 331–360). Cambridge: Cambridge University Press.
- Barlow, H. B. (1989). Unsupervised learning. *Neural Computation*, 1(3), 295–311.
- Barlow, H. B. (2001). Redundancy reduction revisited. *Network: Computation in Neural Systems*, 12(3), 241–253.
- Bell, A. J., & Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6), 1129–1159.
- Bialek, W. (1987). Physical limits to sensation and perception. *Ann. Rev. Biophys. Biophys. Chem.*, 16, 455–478.

- Bouman, M. A. (1959). History and present status of quantum theory in vision. In W. A. Rosenblith (Ed.), *Sensory communication: Contributions to the Symposium on Principles of Sensory Communication* (pp. 377–401). Cambridge, MA: MIT Press.
- Brenner, N., Strong, S. P., Koberle, R., Bialek, W., & de Ruyter van Steveninck, R. R. (2000). Synergy in a neural code. *Neural Computation*, 12(7), 1531–1552.
- Cannon, W. B. (1939). *The wisdom of the body*. New York: Norton.
- Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. New York: Wiley.
- Der, R., Steinmetz, U., & Pasemann, F. (1999). Homeokinesis—a new principle to back up evolution with learning. In M. Mohammadian (Ed.), *Computational intelligence for modelling, control, and automation* (pp. 43–47). Amsterdam: IOS Press.
- Dömösi, P., & Nehaniv, C. L. (2005). *Algebraic theory of finite automata networks: An introduction*. Philadelphia: Society for Industrial and Applied Mathematics.
- Friedman, N., Mosenzon, O., Slonim, N., & Tishby, N. (2001). Multivariate information bottleneck. In *Uncertainty in Artificial Intelligence: Proceedings of the Seventeenth Conference (UAI-2001)* (pp. 152–161). San Francisco: Morgan Kaufmann.
- Gibson, J. J. (1979). *The ecological approach to visual perception*. Boston: Houghton Mifflin.
- Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13(5), 533–549.
- Howard, R. A. (1966). Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, SSC-2, 22–26.
- Hutchins, E. (1995). How a cockpit remembers its speeds. *Cognitive Science*, 19(3), 265–288.
- Kandel, E. R., Schwartz, J. H., & Jessell, T. M. (1991). *Principles of neural science* (3rd ed.). New York: McGraw-Hill.
- Kirsh, D., & Maglio, P. (1994). On distinguishing epistemic from pragmatic action. *Cognitive Science*, 18(4), 513–549.
- Klyubin, A. S., Polani, D., & Nehaniv, C. L. (2004a). Organization of the information flow in the perception-action loop of evolved agents. In R. S. Zebulum, D. Gwaltney, G. Hornby, D. Keymeulen, J. Lohn, & A. Stoica (Eds.), *Proceedings of 2004 NASA/DoD Conference on Evolvable Hardware* (pp. 177–180). Los Alamitos, CA: IEEE Computer Society.
- Klyubin, A. S., Polani, D., & Nehaniv, C. L. (2004b). *Organization of the information flow in the perception-action loop of evolved agents* (Tech. Rep. 400). Hertfordshire: Department of Computer Science, University of Hertfordshire.
- Klyubin, A. S., Polani, D., & Nehaniv, C. L. (2004c). Tracking information flow through the environment: Simple cases of stigmergy. In J. Pollack, M. Bedau, P. Husbands, T. Ikegami, & R. A. Watson (Eds.), *Artificial Life IX: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems* (pp. 563–568). Cambridge, MA: MIT Press.
- Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 22(1), 79–86.
- Laughlin, S. B., de Ruyter van Steveninck, R. R., & Anderson, J. C. (1998). The metabolic cost of neural information. *Nature Neuroscience*, 1(1), 36–41.
- Linsker, R. (1988). Self-organization in a perceptual network. *IEEE Computer*, 21(3), 105–117.
- Millikan, R. G. (2002). *Varieties of meaning*. Cambridge, MA: MIT Press.

- Nadal, J.-P., & Parga, N. (1995). Information transmission by networks of non linear neurons. In *Proc. of the Third Workshop on Neural Networks: From Biology to High Energy Physics* (1994, Elba, Italy). *International Journal of Neural Systems* (Suppl.), 153–157.
- Nolfi, S., & Marocco, D. (2002). Active perception: A sensorimotor account of object categorization. In B. Hallam, D. Floreano, J. Hallam, G. Hayes, & J.-A. Meyer (Eds.), *From Animals to Animats 7, Proceedings of the VII International Conference on Simulation of Adaptive Behavior* (pp. 266–271). Cambridge, MA: MIT Press.
- Pearl, J. (2001). *Causality: Models, reasoning, and inference*. Cambridge: Cambridge University Press.
- Polani, D., Nehaniv, C. L., Martinetz, T., & Kim, J. T. (2006). Relevant information in optimized persistence vs. progeny strategies. In L. M. Rocha, M. Bedau, D. Floreano, R. Goldstone, A. Vespignani, & L. Yaeger (Eds.), *Proc. Artificial Life X* (pp. 337–343). Cambridge, MA: MIT Press.
- Powers, W. T. (1974). *Behavior: The control of perception*. London: Wildwood House.
- Schmidhuber, J. (1992). Learning factorial codes by predictability minimization. *Neural Computation*, 4(6), 863–879.
- Schmidhuber, J., Eldracher, M., & Foltin, B. (1996). Semilinear predictability minimization produces well-known feature detectors. *Neural Computation*, 8(4), 773–786.
- Schneidman, E., Bialek, W., & Berry II, M. J. (2003). Synergy, redundancy, and independence in population codes. *Journal of Neuroscience*, 23(37), 11539–11553.
- Schneier, B. (1995). *Applied cryptography: Protocols, algorithms, and source code in C* (2nd ed.). New York: Wiley.
- Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27, 379–423.
- Tishby, N., Pereira, F. C., & Bialek, W. (1999). The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing* (pp. 368–377). Urban-Champaign: Department of Electrical and Computer Engineering, University of Illinois.
- Touchette, H., & Lloyd, S. (2000). Information-theoretic limits of control. *Phys. Rev. Lett.*, 84, 1156.
- Touchette, H., & Lloyd, S. (2004). Information-theoretic approach to the study of control systems. *Physica A*, 331(1–2), 140–172.
- Usher, M. (2001). A statistical referential theory of content: Using information theory to account for misrepresentation. *Mind and Language*, 16(3), 311–334.
- Wennekers, T., & Ay, N. (2005). Finite state automata resulting from temporal information maximization. *Neural Computation*, 17(10), 2258–2290.

Copyright of Neural Computation is the property of MIT Press and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.