Optimizing Associative Information Transfer within Content-addressable Memory

Mikhail Prokopenko[1], Daniel Polani[2], and Peter Wang[1]

[1] CSIRO Information and Communication Technology Centre
Locked bag 17, North Ryde, NSW 1670, Australia
[2] Department of Computer Science, University of Hertfordshire
Hatfield AL10 9AB, United Kingdom
mikhail.prokopenko@csiro.au

Abstract. This paper[1] investigates an information-theoretic design principle, intended to support an evolution of a memory structure fitting a specific selection pressure: associative information transfer through the structure. The proposed criteria measure how much does associativity in memory add to the information transfer in terms of precision, recall and effectiveness. The study also introduces a conjectural analogy between memory retrieval and self-replication, with DNA as a partially-associative memory containing relevant information. DNA decoding by a complicated protein machinery ("cues" or "keys") may correspond to an associative recall: i.e., a replicated offspring is an associatively-recalled prototype. The proposed information-theoretic criteria intend to formalize the notion of information transfer involved in self-replication, and enable bio-inspired design of more effective memory structures.

1 Introduction

Bio-inspired models have been suggested and used in many areas of Unconventional Computing: parallel processing such as Cellular Automata (CA) and DNA computation; distributed storage and transmission: e.g., neural networks and associative memory; search and optimization: e.g., genetic algorithms and ant colony optimization (ACO). New metaphors are discovered and applied at an increasing pace, improving computational models in terms of robustness, adaptivity and scalability. However, there is a certain lack of a unifying methodology, or at least a set of guiding principles, underlying many recent developments. This is unsatisfactory not only from a methodological, but also from a pragmatic point of view: if some generic principles are not utilized then specific solutions are likely to be suboptimal.

Existence of such core principles may be supported by an observation that most of the bio-inspired models listed above do not fit into a particular category of conventional computing (memory, communication, processing), but cope with multiple aspects. For instance, CA were shown by Langton [23] to support, under certain conditions (*the edge of chaos*), three basic operations of information storage, transmission, and modification, through static, propagating and interacting structures (*blinkers*, *gliders*, *collisions*). ACO algorithms also combine distributed memory, distributed transmission

[1] This paper extends preliminary studies and results reported earlier [34].

and distributed search, employing stigmergy — the process by which multiple ant-like agents indirectly interact through changes in their environment caused by pheromone deposits [7, 8] — and resulting in emergence of optimal solutions. In other words, these fundamental aspects of dealing with information are fused together within these bio-inspired approaches, making them less brittle and more scalable than conventional systems. One compelling explanation is that the motivating biological systems (ranging from cellular tissues to ant colonies) co-evolved the computing components rather than assemble the overall architecture out of separately designed parts [15, 26].

The main question then becomes what are the core principles that inter-relate memory, communication, and processing in evolvable computational systems? Answering this question from an information-theoretic viewpoint may also improve comparability of different bio-inspired approaches. In this paper, we propose an information-theoretic design principle, intended to support an evolution of a memory structure fitting a specific selection pressure: associative information transfer through the structure. In doing so we minimize architectural assumptions about memory or processor structures, hoping instead that such dependencies emerge as a result of the optimization of the information processing dynamics. Our preliminary studies, reported here, indicate that the proposed principle is capable of clearly identifying the range and information dynamics of possible memory structures in a very general sense, enabling design of optimal memory.

The following Section points out some relevant background material on unconventional memory organization, as well as intrinsic information-theoretic fitness criteria used in evolvable computational systems. Section 3 describes the proposed measure, followed by experimental results (Section 4) and conclusions (Section 5).

2 Background and Motivation

Moskowitz and Jousselin [27] have shown that, in a general algebraic sense, the nature of the operations carried out by a computer processor actually determine the structure of the computer memory. In particular, they highlighted the hidden group structure of the address space, and pointed out that “when the integer addition law is used to manipulate addresses, this space is a cyclic group, and memory is seen as a linear array”. When another composition law is used (e.g., a non-commutative address composition), a hypercubic memory structure fits more, greatly reducing complexity of computations.

Another related concept is associative or content-addressed memory: a memory organization in which the memory is accessed by its content rather than an explicit address. Reference clues or keys are “associated” with actual memory contents until a desirable match (or set of matches) is found. A well-known example is a self-organizing map (SOM or Kohonen network). It can be interpreted as an associative memory which encodes the input patterns within the nodes of the network (the neural layer), in the form of weight (codebook) vectors of the same dimension and nature as the input patterns [22]. When a partial or corrupted pattern of data (a sensory cue) is presented in the form of a key input-vector, the rest of the pattern (memory) is associated with it. A characteristic of SOM-based associative memory is its self-organizing ordering: neighboring nodes encode similar codebook vectors, preserving topology: neurons that are

closer in the neural layer tend to respond to inputs that are closer in the input space. A related approach is advocated by Kanerva [16, 17]: a Sparse Distributed Memory (SDM) which is a content addressable, associative memory technique relying on close memory items clustered together: while perceived data sparsely distribute themselves over multiple storage locations, the outcome is a fusion of this distribution. In the auto-associative version of SDM the memory contents and their addresses are from the same space and may be used alternatively. Another well-known example of auto-associative memory reproducing its input pattern as output is the Hopfield neural network [14].

Importance of memory access is discussed by Goertzel [10], who pursues "not a model of how memories are physically stored in the brain or anywhere else, but rather a model of how memory access must work, of how the time required to access different memories in different situations must vary". This pursuit led towards a *structurally associative memory* (STRAM), based on the idea that "if x is more easily accessible than y, those things which are similar to x should in general be more easily accessible than those things which are similar to y" [10]. Goertzel sketched a way of mapping a weighted graph describing STRAM to a physical memory M, by assigning to each pair of elements (x, y) stored by M a distance $D_M(x, y)$ measuring the difficulty of locating x in memory given that y has very recently been located. It was suggested that the distance $D_M(x, y)$ is approximated as a number of links along the shortest path between the graph nodes corresponding to x and y.

It is worth pointing out that our approach does not intend to present just a new measure of associativity or information transfer involved in memory operations, but rather identify an information-theoretic principle contributing to a general methodology. Such a methodology may go beyond computational aspects, including sensing, actuation, and networking in distributed systems, co-evolving under multiple design/selection pressures.

Typically, evolutionary design may employ genetic algorithms in evolving optimal strategies that satisfy given fitness functions, by exploring large and sophisticated search-space landscapes [26]. In general, however, we may approach evolutionary design in two ways: via task-specific objectives or via generic intrinsic selection criteria. The latter approach can be exemplified by *information-driven evolutionary design* which suggested to set intrinsic fitness functions according to information-theoretic criteria [32, 33, 19, 20, 21]. This essentially focuses on optimizing information transfer within specific channels. An example of an intrinsic selection pressure is the acquisition of information from the environment: there is some evidence that pushing the information flow to the information-theoretic limit (i.e., maximization of information transfer in perception-action loops) can give rise to intricate behaviour, induce a necessary structure in the system, and ultimately be responsible for adaptively reshaping the system [18, 19, 20]. Other important selection pressures applicable to distributed systems include stability of self-organizing hierarchies [29, 9]; efficiency of multicellular communication topologies [30]; efficiency of locomotion and distributed actuation [32, 33, 37]. The identification of possible intrinsic fitness criteria is also related to the work of Der *et al.* on self-organization of agent behaviors from domain-invariant principles, e.g., homeokinesis [6].

In summary, our main objective is to identify a selection pressure on associative information transfer involved in memory recall, contributing to the general methodology of information-driven evolutionary design.

3 Information Transfer: Precision and Recall

Since our task is to identify a very generic principle, we choose to abstract away from implementation details and consider instead an unconstrained deterministic function f from two equally distributed random variables K and X to a random variable Y. The variable K is intended to serve as a "key" or "cue" in accessing the memory X, retrieving, as a result of the mapping f, the outcome or "readout" Y, i.e., $Y = f(K, X)$. It is important to realize that while we interpret K, X and Y as key, memory and readout, we do not structurally constrain the variables and the mapping: e.g., there is no requirement that any location x in memory X is accessible by a unique key $k \in K$, etc.

The first constraint that we impose is the criterion:

$$\text{maximization of} \quad \mathcal{P} = I(X; K|Y) \;, \tag{1}$$

where $I(X; K|Y)$ is the conditional mutual information between X and K given Y. Before defining conditional mutual information, let us define the mutual information $I(A; B)$ between A and B:

$$I(A; B) = \sum_{a \in A} \sum_{b \in B} P(a, b) \; \log \frac{P(a, b)}{P(a)P(b)} \;, \tag{2}$$

where $P(a)$ is the probability that A is in the state a, and $P(a, b)$ is the joint probability. Mutual information $I(A; B)$ can be expressed in terms of entropies $H(\cdot)$, joint entropies $H(\cdot, \cdot)$, and conditional entropies $H(\cdot|\cdot)$:

$$I(A; B) = H(A) + H(B) - H(A, B) = H(A) - H(A|B) \;, \tag{3}$$

where the entropies are defined as follows:

$$H(A) = -\sum_{a \in A} P(a) \; \log P(a) \;, \tag{4}$$

$$H(A, B) = -\sum_{a \in A} \sum_{b \in B} P(a, b) \; \log P(a, b) \;, \tag{5}$$

$$H(A|B) = H(A, B) - H(B) \tag{6}$$

When dealing with three-term entropies [25], one typically defines the joint entropy

$$H(A, B, C) = -\sum_{a \in A} \sum_{b \in B} \sum_{c \in C} P(a, b, c) \; \log P(a, b, c) \tag{7}$$

and uses relationships such as

$$H(B|A, C) = H(A, B, C) - H(A, C) \tag{8}$$

A relationship like this is helpful in defining the conditional mutual information:

$$I(X;K|Y) = H(X|Y) - H(X|K,Y) \tag{9}$$

where both conditional entropies on the right-hand side can be obtained via equations (4) – (8).

The criterion (1) maximizes the conditional mutual information between key and memory, given the readout. First of all, we need to clarify that, although K and X are independent and, therefore, mutual information $I(X;K)$ is zero, the conditional mutual information $I(X;K|Y)$ may well be positive. This is analogous to the example of a binary symmetric channel with input X, noise K, and output Y, described by MacKay [25] (we altered the variables names here to avoid confusion): mutual information $I(X;K) = 0$ since input and noise are independent, but $I(X;K|Y) > 0$, because "once you see the output, the unknown input and the unknown noise are intimately related!" [25]. Similarly, the criterion (1) is applied once the readout is obtained, which means that a possible association between memory and key has been made.

Secondly, we draw an analogy with well-known information retrieval metrics: precision and recall. Precision is a measure of usefulness or *soundness* of the readout retrieved in response to a query, and is measured as a fraction of the relevant and retrieved items within the retrieved items (aiming at "nothing but the truth"). Recall is a measure of relevance or *completeness* of the readout, and is measured as a ratio of the relevant and retrieved items over the relevant items (aiming at "the whole truth"). A probabilistic interpretation is possible as well [11]: precision may be defined as the conditional probability that an object is relevant given that it is returned by the system, while the recall is the conditional probability that a relevant object is returned: precision $= P(\text{relevant}|\text{returned})$, and recall $= P(\text{returned}|\text{relevant})$.

Intuitively, the criterion (1) captures the potential $\mathcal{P}$ of precision-driven information transfer. To formalize this intuition, let us apply the chain rule for the mutual information [25]:

$$I(X;Y,K) = I(X;Y) + I(X;K|Y) \tag{10}$$

where the left-hand side contains the mutual information between X and jointly Y and K. This chain rule produces

$$\mathcal{P} = I(X;K|Y) = I(X;Y,K) - I(X;Y)\,. \tag{11}$$

The alternative representation (11) can be interpreted as follows: how much does a key *add to precision of the readout by associating with memory*. The equation (11) contrasts two information transfers: one, $I(X;Y)$, does not use associativity, while the other, $I(X;Y,K)$, incorporates it. The difference between the two transfers captures, we believe, the potential information gain in precision. Another useful representation of the criterion (1) can be obtained in terms of entropies. Applying the relationships (9), (8) and then (6) to the right-hand side of the criterion (1) yields

$$\mathcal{P} = I(X;K|Y) = H(X|Y) - H(X|K,Y) = H(X|Y) - [H(X,Y,K) - H(K,Y)] =$$

$$[H(X,Y) - H(Y)] - H(X,Y,K) + H(K,Y)$$

A further reduction is possible for deterministic functions, where $H(X,Y,K)$ is a constant, making the criterion (1) equivalent to

$$\text{maximization of} \quad \tilde{\mathcal{P}} = H(X,Y) - H(Y) + H(K,Y) \,. \tag{12}$$

The measure $\tilde{\mathcal{P}}$ may, of course, be rewritten as follows:

$$\tilde{\mathcal{P}} = H(X|Y) + H(K,Y) = H(X,Y) + H(K|Y) \,. \tag{13}$$

At this stage we would like to introduce another criterion. We consider

$$\text{maximization of} \quad \mathcal{R} = I(Y;K|X) = I(Y;X,K) - I(Y,X) \,. \tag{14}$$

Intuitively, $\mathcal{R}$ measures how much a key is necessary *to identify the output of the mapping, given the memory*. The criterion (14) captures the potential $\mathcal{R}$ of information transfer involved in the memory recall, and aims to maximize the difference between associative and non-associative information transfer. Using a relationship like (9), we obtain

$$\mathcal{R} = I(Y;K|X) = H(Y|X) - H(Y|K,X) = [H(X,Y) - H(X)] - H(Y|K,X)$$

For deterministic functions, the entropy $H(Y|K,X)$ is zero, and the entropy $H(X)$ is a constant. Hence, maximization of $\mathcal{R}$ is equivalent to

$$\text{maximization of} \quad \tilde{\mathcal{R}} = H(X,Y) \,. \tag{15}$$

It should be noted that since $Y = f(K,X)$, the expression for $\tilde{R}$ is dependent on K.

The overall effectiveness of information retrieval is typically defined as the harmonic mean (the reciprocal of the arithmetic mean of the reciprocals) of recall and precision — hence, we suggest the criterion:

$$\text{maximization of} \quad \mathcal{E} = \frac{2}{\frac{1}{\mathcal{P}} + \frac{1}{\mathcal{R}}} = \frac{2\mathcal{P}\mathcal{R}}{\mathcal{P} + \mathcal{R}} \,, \tag{16}$$

fusing together the potential information gains in both precision and recall.

In order to highlight different roles played by K and X, we consider here scenarios with varying sizes $|K|$ and $|X|$, interpreted in the context of several examples: (a) catalog/book indexing and search; (b) pattern association using a neural network; (c) decoding of genotype (DNA) by proteins. The scenarios are as follows:

(S_1) $|K| \gg |X|$ and $|X| \approx |Y|$, $\qquad$ (S_2) $|K| \approx |X| \approx |Y|$,
(S_3) $|K| \ll |X|$ and $|X| \approx |Y|$

where $|\circ|$ is the cardinal number of the set $\circ$ (in our case, simply the number of its elements). In the example (a), a library catalogue is a database containing records indexed by the authors, titles, subjects, etc. The explicit "cue" is the key, using which a set of catalogue items $Y' \subseteq Y$ can be found as a result of a query. Typically, $|K| \gg |X|$, while $|Y| \approx |X|$: this is our first scenario (S_1). Similarly, a book can be indexed by associating its content (e.g., pages) with keywords. In this case, $|K| \gg |X|$ as well, since

there may be more keywords than pages, while $|Y| \approx |X|$ as the number of retrieved pages may approach their total number. However, the scenario (S_2) pushes the scenario (S_1) to the extreme by restricting the number of possible keys (e.g., a limit on queries), while the memory size is unchanged: $|K| \approx |X|$. This represents a more challenging case with respect to the precision as the relevant items are harder to find.

The example (b) involves an artificial neural network, e.g., a self-organizing map (SOM) implementing associative memory, briefly discussed in section 2. Each neuron in memory (a network node) encodes a retrievable pattern, hence $|Y| \approx |X|$. Of course, memory updates would lead to an increase in the overall number of returned patterns, highlighting the distinction between cumulative memory capacity and memory size. The SOM handles multiple cues/keys as partial or corrupted patterns of data, associating them with the memory, implying $|K| \gg |X|$. This also concurs with the first scenario (S_1). Again, restricting the number of possible keys while keeping the memory size is unchanged (the scenario (S_2)) would challenge the system in terms of the precision.

The third scenario (S_3) may correspond to an auto-associative neural network such as the Hopfield network [14] or a Sparse Distributed Memory [16]. A key is interpreted simultaneously by all neurons which interact by updating their weights until a stable network state is reached: this attractor then represents the network output associated with the key. In this case, $|X| \gg |Y|$ since there is only a limited number of attractor states supported by the network, while $|K| \ll |X|$ due to high-dimensionality of memory. Interestingly, restricting the memory (reducing $|X|$) would challenge precision again, approaching the scenario (S_2) from another direction.

Finally, we consider the case (c) when a genotype (DNA) is decoded by proteins. An individual DNA can be interpreted as associative memory in the sense that it contains *potential information* relevant to the niche occupied by the individual's species. As pointed out by Adami [1], "If you do not know which system your sequence refers to, then whatever is on it cannot be considered information. Instead, it is potential information (a.k.a. entropy)". Decoding a DNA involves a complicated protein machinery (the key), and may correspond to an associative recall. In this model, a replicated offspring is an associatively-recalled prototype. In the next section we shall interpret all three scenarios within this analogy.

4 Results

The experimental setup is very simple: we intend to satisfy our criteria (1), (14), and (16) by varying possible deterministic functions $Y = f(K, X)$ over finite size domains K, X and Y, for the scenarios (S_1), (S_2) and (S_3). In particular, we consider three sets of integers $\{1, \ldots, |K|\}$, $\{1, \ldots, |X|\}$ and $\{1, \ldots, |Y|\}$, and vary their sizes $|K|$, $|X|$ and $|Y|$ between experiments. For each experiment, we search for deterministic mappings $Y = f(K, X)$ which maximize $\mathcal{P}$, or $\mathcal{R}$, or $\mathcal{E}$ — repeating the search for each of these criteria. We used a simple genetic algorithm (GA) to evolve solutions to the maximization problems. The initial population is generated by random mappings $Y = f_i(K, X)$, for a sufficiently large number of individual mappings, e.g. $1 \leq i \leq 1000$. At each generation, the mappings are evaluated in terms of the criterion in point (either $\mathcal{P}$, or $\mathcal{R}$, or $\mathcal{E}$). We have chosen a generation gap replacement strategy (the entire

old population is sorted according to the fitness, and the best 10% are chosen for direct replication in the next generation, employing an elitist selection mechanism), and the multiple-point crossover. We also ensure that the mutation results in a unique individual by re-applying this operator if necessary. The GA typically converged to theoretical maxima for the criteria within 8000 generations.

4.1 Grid Contours

Visualizing evolved mappings f is not revealing, as can be observed from Figure 1. We plot instead an analogue of a 2-dimensional contour, but rather than simply using contours, we connect, for a given height $y \in Y$, all points $(k, x) \in K \times X$ which agree either on k or on x, producing a partial grid. For example, if there are entries $7 = f(1, 4)$, $7 = f(3, 4)$, and $7 = f(1, 6)$, we connect points $(1, 4)$ and $(3, 4)$ as they represent the same memory $x = 4$, as well as points $(1, 4)$ and $(1, 6)$ sharing the same key $k = 1$. Such a *grid-contour* combines grids for all values of $y \in Y$ by "overlaying" the grids for all values y.

A random mapping (the zero hypothesis) has no discernable structure for all scenarios (e.g., Figure 2). Let us focus initially on the scenario (S_1). A $\mathcal{P}$-maximizing mapping for this scenario is a structure with dominant horizontal lines (Figure 3). Each horizontal reflects the fact that in the evolved mapping, the same memory is recalled if multiple different keys are associated with it. This, in the context of DNA decoding, corresponds to conservation of DNA (memory) and its robustness to possible decoding errors (multiple keys), ensuring high precision. A $\mathcal{R}$-maximizing mapping maintains the horizontal lines but introduces some vertical lines (Figure 4). Each vertical line means that a key recalls the same content even if associated with different memories. In the context of DNA decoding, this may correspond to pseudo-genes within a DNA (characterised by a lack of protein-coding functionality): redundant code which does not differentiate between offsprings and ensures high recall. Importantly, the effectiveness criterion $\mathcal{E}$ maintains the horizontal lines (robust DNA) but eliminates the vertical lines (no pseudo-genes), as shown in Figure 3. On the other hand, minimization of $\mathcal{E}$ does the opposite, producing a grid-like structure, i.e., for every association (k_1, x_1) there exists an association (k_2, x_2) such that either $k_1 = k_2$ or $x_1 = x_2$ (more precisely, the mapping minimizing the criteria is given by a constant f).

The scenario (S_2) pushes the observed tendencies to their limits. A $\mathcal{P}$-maximizing mapping for this scenario is a structure with no lines (Figure 5). There are no entries which share either a key or memory — in other words, both key and memory are necessary. Such an outcome illustrates the full precision of associative memory (a perfectly succinct DNA). An $\mathcal{R}$-maximizing mapping has some vertical lines (Figure 6), suggesting that some pseudo-genes are possible even in the highest recall case. This can be interpreted as a tendency towards the dominance of precision over recall, i.e., robustness of DNA at the expense of redundancy. However, the effectiveness criterion $\mathcal{E}$ eliminates redundancy and results in a fully associative memory structure (Figure 5), where for every pair of a key and memory, fixing a key k and varying memory x (or vice versa) results in a different readout $y = f(k, x)$.

The results for the scenario (S_3) are not surprising: mappings maximizing $\mathcal{P}$, $\mathcal{R}$ and $\mathcal{E}$ produce structures with only vertical lines. In the context of DNA decoding,

this would correspond to highly redundant and error-prone DNA structures. This model would work for reproduction if different arrays collectively store information (as in an SDM or Hopfield network), "retrieving" offspring as a composite result of data fusion, e.g. genetic cross-over.

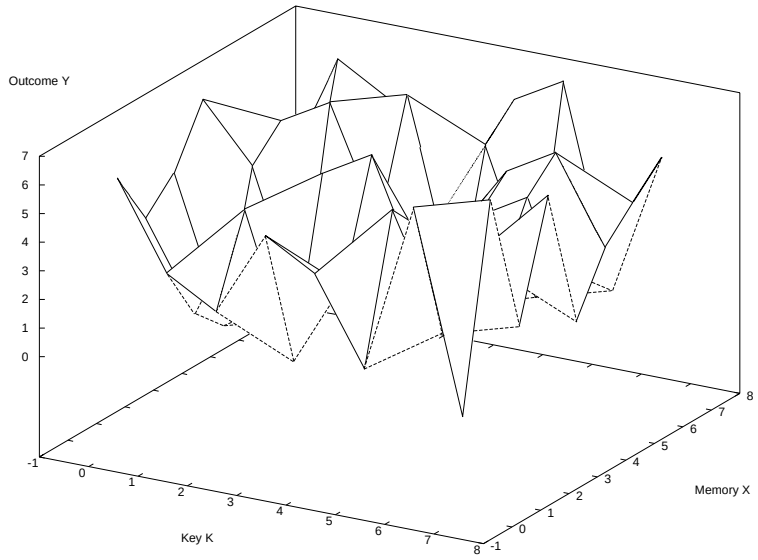

Fig. 1. An evolved mapping: scenario (S_2).

4.2 Conjecture: a Model within The Model

A mapping $Y = f(K, X)$ implementing fully associative memory in the scenario (S_2) (Figure 5) can be interpreted in weak self-referential terms. Self-referentiality has many interpretations, ranging from programming data structures (a self-referential structure contains a pointer to a structure of the same type) to cognitive neuroscience: the self is a cognitive structure with special mnemonic abilities, leading to "the enhanced memorability of material processed in relation to self" [12, 35], suggesting that a self-referential memory — a memory about the self — is not ordinary. According to the well-known interpretation of Hofstadter [13], a self-referential system can be characterised by emergent behaviour and tangled hierarchies exhibiting Strange Loops: "an interaction between levels in which the top level reaches back down towards the bottom level and influences it, while at the same time being itself determined by the bottom level". We shall adopt a weaker interpretation of self-referential memory: the memory using a

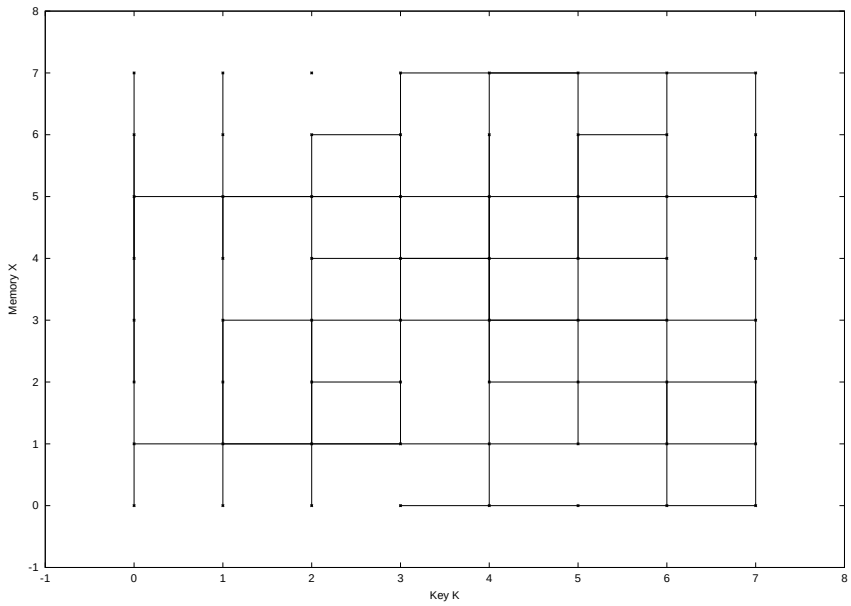

Fig. 2. Grid-contour of a random mapping: scenario (S_2).

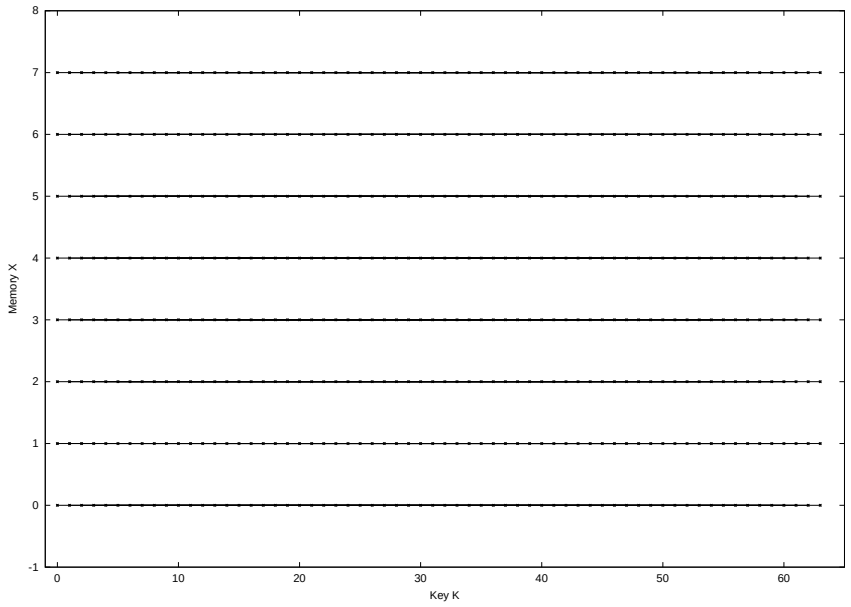

Fig. 3. Grid-contour of a $\mathcal{P}$-maximizing mapping, as well as an $\mathcal{E}$-maximizing mapping: scenario (S_1).

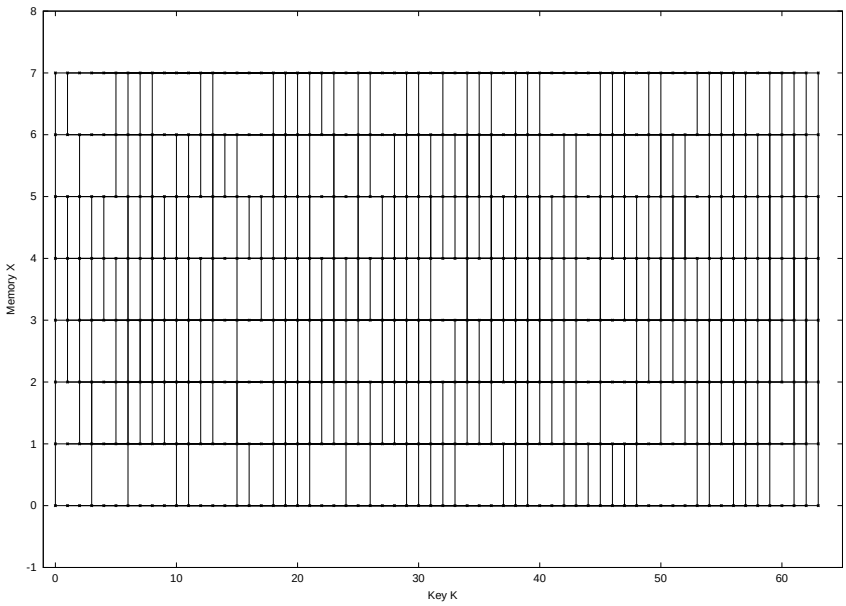

Fig. 4. Grid-contour of a $\mathcal{R}$-maximizing mapping: scenario (S_1).

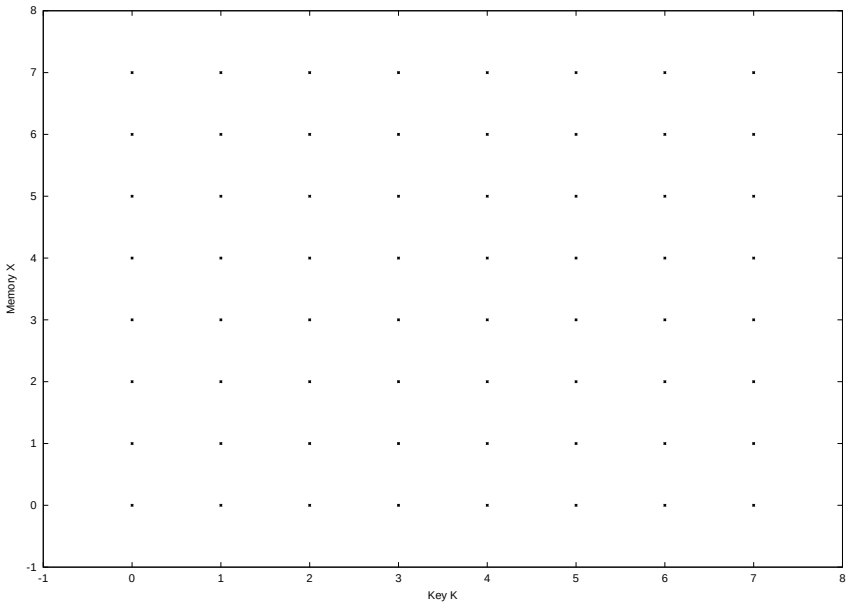

Fig. 5. Grid-contour of a holographic $\mathcal{P}$-maximizing mapping, as well as an $\mathcal{E}$-maximizing mapping: scenario (S_2). Its 3-dimensional counterpart is shown in Figure 1.

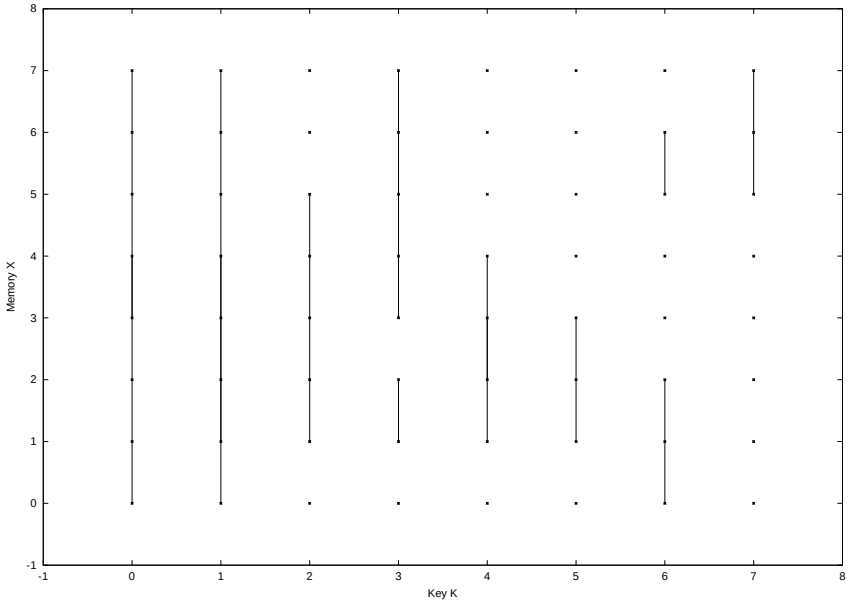

Fig. 6. Grid-contour of a $\mathcal{R}$-maximizing mapping: scenario (S_2).

model of itself. This limited a-model-within-the-model view is not intended to preclude emergence of tangled hierarchies, or references to the cognitive self of the agent using this memory.

We begin by observing that, on the one hand, for any k, x_1, x_2 $(x_1 \neq x_2)$, we have $f(k, x_1) \neq f(k, x_2)$, i.e., the readouts differ for the same key and a varying memory. This means that every memory is sufficiently sensitive to its own content/location, and there is no redundant information in the associated key: the difference in the readout is due to different memory. This observation can be formalised as follows. Let us introduce an *array-readout* $\|Y_k\| = [f(k, x_1), \ldots, f(k, x_{|X|})]$. In other words, $\|Y_k\|$ is an array of readouts $f(k, x)$ produced by the mapping f given a fixed key k. Then the observation that every memory is sufficiently sensitive to its own content/location is formally expressed by stating that each array-readout $\|Y_k\|$ is isomorphic to the memory space X (i.e., $\|Y_k\|$ cannot be made less informative than the space X, or any array $[1, \ldots, n]$, where $n = |K| = |X|$).

On the other hand, for any x, k_1, k_2 $(k_1 \neq k_2)$, we have $f(k_1, x) \neq f(k_2, x)$, i.e., the readouts differ for a varying key and identical memory. This means that every key is sufficiently informative to produce different readouts upon association with the same memory. In other words, every memory content is sufficiently sensitive to each key (as well as to its content/location), and therefore, encodes information about all possible keys. Formally, each array-readout $\|Y_x\| = [f(k_1, x), \ldots, f(k_{|K|}, x)]$ for a fixed memory x is isomorphic to the key space K (i.e., $\|Y_x\|$ cannot be made less informative than the space K, or any array $[1, \ldots, n]$, where $n = |K| = |X|$).

Another interpretation is that the evolved mapping implements a *Latin square* — an $n \times n$ table filled with n different symbols in such a way that each symbol occurs exactly once in each row and exactly once in each column [24]. In particular, the evolved mapping can be described by a multiplication table of a cyclic abelian group $\mathbb{Z}_n$ of order $n = |X| = |K| = |Y|$.

Furthermore, let us consider two array-readouts $\|Y_{x_1}\|$ and $\|Y_{x_2}\|$. Each of these array-readouts is as informative as any permuted array $[1, 2, \ldots, n-1, n]$, where $n = |K| = |X|$. Such permuted arrays can be interpreted as *arrays of permuted keys*, given a fixed memory. Without a loss of generality, $\|Y_{x_1}\|$ may be represented by the array $[1, 2, \ldots, n-1, n]$, and the array $[2, 3, \ldots, n, 1]$ may represent $\|Y_{x_2}\|$. Importantly, the circular shift (in the "horizontal" direction across keys) between the arrays ensures that, given a fixed key, the outcomes retrieved from memories x_1 and x_2 would always be different.

Each permuted array (i.e., each array-readout), however, is a memory model *per se*, and we shall use this in closing the loop around our system $Y = f(X, K)$ with a feedback

$$f'(k, x) = \|Y_{D(f(k,x))}\| \tag{17}$$

Here $D(y)$ is a "diagonalization" function from Y to X. It takes the readout $y = f(k, x)$, uniquely maps it to an integer i, and returns a memory $x' = x_i$. When the memory x' is identified, the feedback is set: $f'(k, x)$ is the new associative memory content, filled by the array-readout $\|Y_{x'}\|$ of memory x'. To re-iterate, the initial readout $f(k, x)$ is *interpreted* as the new memory x', producing the array-readout $\|Y_{x'}\|$.

Let us consider a simple example, shown in Table 1. The associative mapping $f(k, x)$ at the first iteration may, in particular, be represented by a Latin square ($n = 3$), i.e. a cyclic abelian group of order 3.

	k_1	k_2	k_3	array-readout
x_1	[1]	[2]	[3]	[1, 2, 3]
x_2	[2]	[3]	[1]	[2, 3, 1]
x_3	[3]	[1]	[2]	[3, 1, 2]

Table 1. Mapping $f(k, x)$ at the first iteration. Latin square ($n = 3$), or a cyclic abelian group of order 3. The right-hand side column shows array-readouts $\|Y_{x_i}\|$.

In order to illustrate the feedback $f'(k, x) = \|Y_{D(f(k,x))}\|$, let us select, as an example, the readout $f(k_2, x_2) = [3]$, and use it in producing the memory $x' = x_3$. Here we capitalize on the fact that the readouts y contain integers: $y = [i]$, i.e. $y \in \{[1], [2], [3]\}$, and use a simple diagonalization $x_i = D(y)$, such that function D returns the first integer of the first element of the array y (the need for such recursion will become clear at the second iteration). When the memory $x' = x_3$ is identified, the array-readout $\|Y_{x_3}\|$

is obtained as $[3, 1, 2]$, and the feedback is set: $f'(k_2, x_2) = [3, 1, 2]$. Closing the loop for all pairs (k, x) results in the system shown in Table 2.

	k_1	k_2	k_3	array-readout
x_1	$[1, 2, 3]$	$[2, 3, 1]$	$[3, 1, 2]$	$[[1, 2, 3], [2, 3, 1], [3, 1, 2]]$
x_2	$[2, 3, 1]$	$[3, 1, 2]$	$[1, 2, 3]$	$[[2, 3, 1], [3, 1, 2], [1, 2, 3]]$
x_3	$[3, 1, 2]$	$[1, 2, 3]$	$[2, 3, 1]$	$[[3, 1, 2], [1, 2, 3], [2, 3, 1]]$

Table 2. Mapping $f('k, x)$ at the second iteration. The right-hand side column shows new array-readouts $\|Y'_{x_i}\|$.

The loop continues with new iterations: the function $D(y)$ always retrieves the first integer in a nested readout $y = f(k, x)$, and uses it in pointing out the memory $x' = D(y)$ and the corresponding $\|Y_{D(y)}\|$. The iterations preserve the cyclic group characteristic of the system. In general, the function D may be quite involved, e.g. it may introduce some noise into the feedback, resulting in more complex scenarios. Importantly, at every iteration of the closed loop, each new memory maintains a possible model of itself. Moreover, the feedback $f'(k, x) = \|Y_{D(f(k,x))}\|$ iteratively "packs" more and more structure into the memory nested at multiple scales.

An analogous arrangement (but in the "vertical" direction across memories) can be obtained with new readouts $\|Y_k\|$ represented by *permuted memory arrays*. It is also possible to interleave iterations of permuted key-arrays with iterations of permuted memory-arrays.

The closed-loop system results in a Latin-square grid contour — the one produced by the evolved fully associative mapping, i.e. both key and memory are necessary for retrieval. We believe that this closed-loop fully-associative memory exhibits self-referentiality and optimizes information transfer in terms of precision and recall. The self-referentiality emerges under the pressures imposed by restricting the number of queries and readouts to the memory size: the scenario (S_2). If one of these pressures is relaxed, self-referentiality is not needed and a memory does not have to encode information about all possible keys: hence, the presence of horizontal lines in the optimal structures for the scenario (S_1), or vertical lines for the scenario (S_3).

4.3 Connectivity within Optimal Associative Memory

The previous subsection presented a conjecture that a self-referential memory (a model within the model) may produce a grid contour identical to the one exhibited by the evolved memory structures. Given that the set-up adopted in this work is intentionally generic, it is not possible to demonstrate an explicit memory architecture for the system X, K, Y, and verify "packing" of information at multiple scales. In other words, the presented results may be related only to a snapshot (a single iteration) of the closed loop that we believe is necessary for a self-referential memory. Nevertheless, we intend

to further analyse optimal mappings f, hoping to discern a multi-scale structure of associative connections in terms of readouts.

In doing so, we performed a multi-objective optimization of f with respect to precision $I(X;K|Y)$ and recall $I(Y;K|X)$ (as usual, X and K were independently equidistributed), finding the Pareto front of the mappings f where both are non-dominated. For this purpose, we used the NSGA II code [5] with the following parameters:

Population size	40
Generations	300
η_c[2]	20
η_m	30
Crossover probability	0.2
Mutation probability	0.002

The size of the chromosomes depends on the problem considered. Via the multi-objective optimization, one could identify potential trade-off surfaces at the Pareto front, but it turns out actually that precision and recall are simultaneously optimized. In particular, the variance of the Pareto front is typically small (deviations of less than 1% of the absolute value the are the norm) for both objectives which means that the objectives undergo little trade-off, if at all.

We consider memory sizes $|X| = |K| = |Y| = 16$ and $|X| = |K| = |Y| = 64$. As our aim is to better understand the evolved memory structure, we study whether different keys k, k' can be interpreted as pointing to overlapping or distinct "memory locations". Since the system X, K, Y has no explicit memory architecture, it is now our task to partially "reverse engineer" the evolved memory structure — for a single iteration of the closed loop.

For this purpose, consider for a moment the family of random variables Y_k for different (fixed!) k. Here, k and k' are fixed values from the domain of K, and the joint distribution of the variables Y_k and $Y_{k'}$ is to be interpreted according to the following Bayesian network:

$$Y_k \leftarrow X \rightarrow Y_{k'} \ ,$$

i.e. their distribution is given by

$$p(y_k, y_{k'}) = \sum_x p(y_k|x,k)\, p(x)\, p(y_{k'}|x,k') \tag{18}$$

where $p(y_k|x,k) = \delta_{f(x,k)}(y_k)$ with δ being the Kronecker function, and likewise for $y_{k'}$.

Now we can consider quantities such as $I(Y_k;Y_{k'})$, the mutual information between outputs Y induced by switching on particular keys k. If this mutual information is nonvanishing only for $k = k'$, it means that the keys are "incommensurable", i.e. they point to distinct memory locations which have no structure with respect to each other.

[2] η_c and η_m are parameters used in the SBX crossover implementation for binary strings. See [4] for details.

In the optimization runs, this turns out to be the generic case for general sizes of X, K and Y.

However, for sizes $|X| = |K| = |Y|$, i.e. the scenario (S_2), we do find some overlapping between the readouts Y_k for different fixed keys k. To allow for a more geometrical interpretation of the memory structure, instead of the mutual information $I(Y_k; Y_{k'})$, we consider the information-theoretic Crutchfield distance ("information metric") [3] to measure the relation between two keys k and k':

$$d(Y_k, Y_{k'}) = H(Y_k|Y_{k'}) + H(Y_{k'}|Y_k) \ . \tag{19}$$

This allows us to create the distance matrix between all k. The histogram of all assumed distances gives a good indication of whether there is some relation between different k. If the values are mainly concentrated at two values (for instance, one of them 0), that indicates incommensurable keys.

As Fig. 7 shows, the case where the sizes of X, K, Y are equal provides a much more interesting structure.

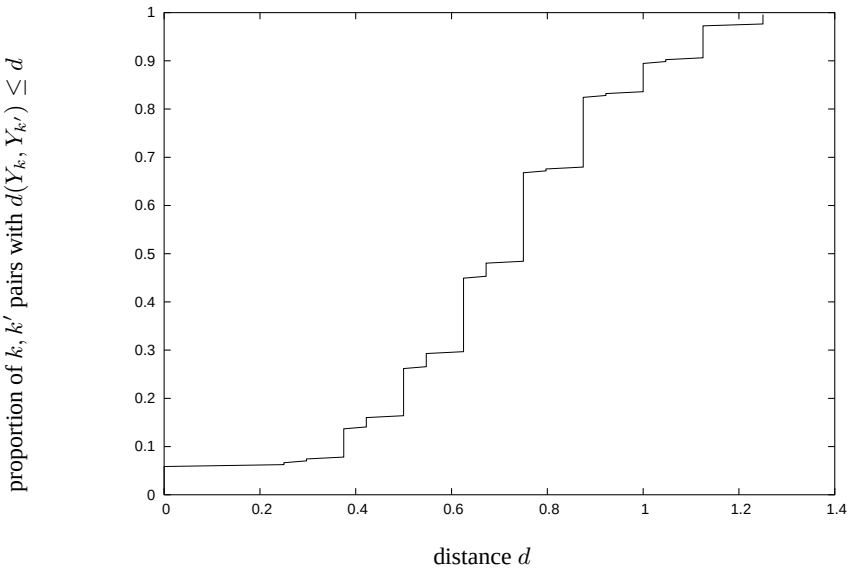

Fig. 7. Distance histogram for $|X| = |K| = |Y| = 16$. The graph shown is the cumulative distribution for distances $d(Y_k, Y_{k'})$, i.e. the integral of the probability of finding a particular distance $d = d(Y_k, Y_{k'})$. A histogram for incommensurable key structures (not shown) would essentially find just one steep growth close to $d = 0$ and one close to the maximal attained value of $d = d_{\max}$, corresponding to a probability distribution with two peaks, one at 0 and one at $d_{\max}$. The current figure thus shows a case with more structure (see text).

This can be further investigated by projecting (embedding) the distance matrix into a Euclidean space, e.g. by finding those points in 2-dimensional space whose distance

matrix best matches the Crutchfield distance matrix of the pairs $(Y_k, Y_{k'})$. Such an *embedding* is shown for size 16 in Fig. 8, and for size 64 in Fig. 9. Figure 10 shows the embedding (size 64) for a random mapping.

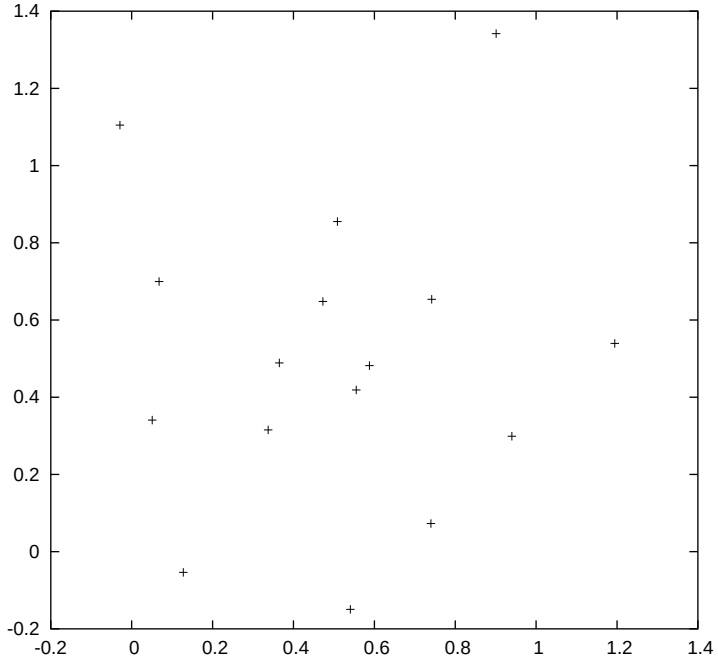

Fig. 8. Projected Crutchfield distances of $(Y_k, Y_{k'})$ pairs (system size 16; evolved mapping). The distances of the points are an approximation of the distances of the Y_k.

The obvious difference between embeddings produced for the evolved and random mappings is that the former is much more compact than the latter, e.g., the embedding in Fig. 9 (evolved mapping) occupies half the area of the embedding in Fig. 10 (random mapping). Besides the approximation effort (measured via the sum of square distances between the original distance matrix and the embedded distance matrix) indicates that the embedding for the evolved mapping maintains distances twice as well as the embedding for the random mapping. These observations support the expectation that a degree of commensurability, and hence associativity, in the evolved system is higher than such a degree in a random mapping: the higher is the associativity the easier it is to represent the distances in 2D, while low associativity would require more dimensions to maintain the distances.

The feedback from readouts of a closed-loop system into the new memory, given by equation (17), would inject and preserve this associativity at one scale, while creating associations on the new scale. We may conjecture that tangled hierarchies of a self-referential memory require iterations of multiple memory levels, and more precisely

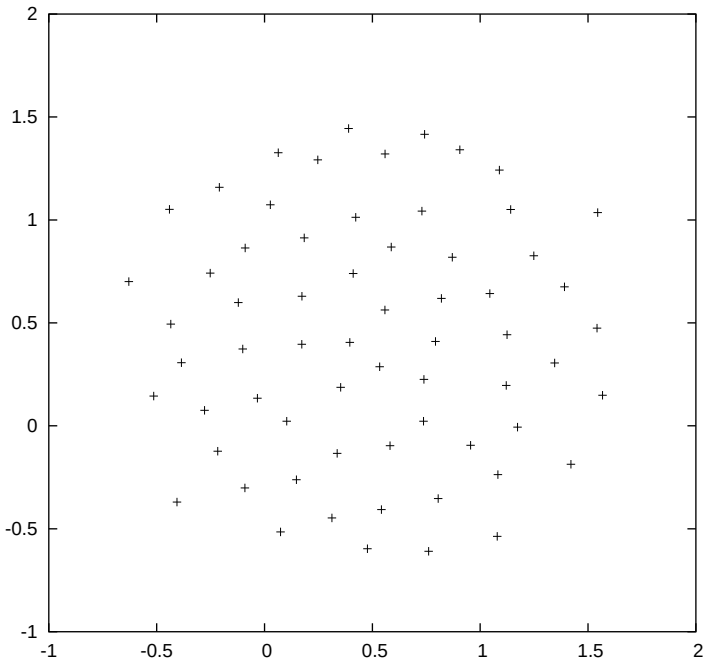

Fig. 9. Projected Crutchfield distances of $(Y_k, Y_{k'})$ pairs (system size 64; evolved mapping). The distances of the points are an approximation of the distances of the Y_k.

iterations of self-similar scales (nested within more and more refined scales). However, an investigation of the relationship between scale-invariance, tangled hierarchies, and the associative memory is outside the scope of this work.

At this stage we present another useful tool to study the evolved memory structures: a *distance-graph* — the graph where two nodes Y_k are connected if the distance is below a given threshold. For a threshold of an intermediate value 0.6 (see Fig. 7), one obtains the graph in Fig. 11.

The graph shows quite an intricate structure of the memory with some nodes serving as "hubs", some nodes having quite a few connections and other nodes having only limited similarity to the rest, finally some isolated nodes. It is important to note that there are more "hubs" than nodes with low degrees, i.e. the distribution is opposite to the one of a scale-free graph. There is not enough data, however, to estimate whether there is a power law underlying the observed distribution, and what would be the parameters of such scale-invariance[3].

Nevertheless, we would like to point out that by varying the threshold of the distance-graph, one may zoom into the multiple scales that may or may not be present in the memory structures under investigation. For example, the evolved optimal memory struc-

[3] If confirmed, this scale-invariance would be driven by the tendency to have high associativity (dominating hubs) at multiple scales.

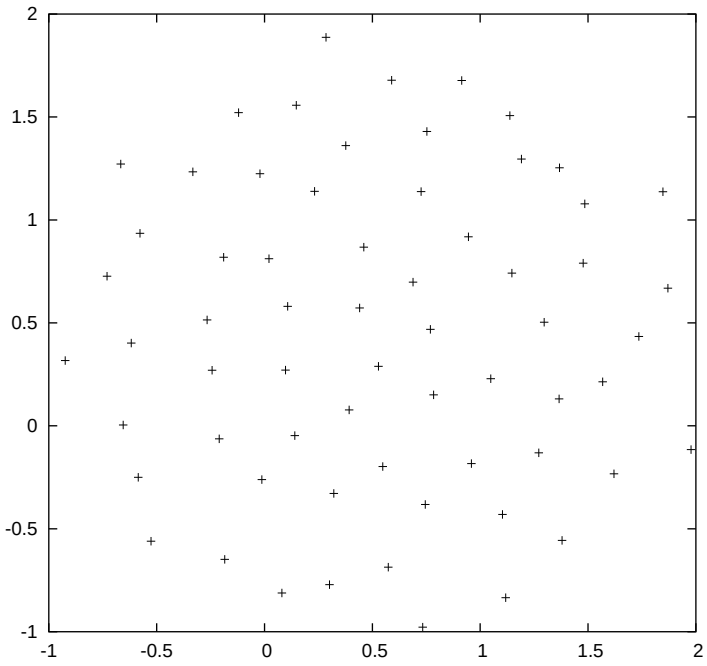

Fig. 10. Projected Crutchfield distances of $(Y_k, Y_{k'})$ pairs (system size 64; random mapping). The distances of the points are an approximation of the distances of the Y_k.

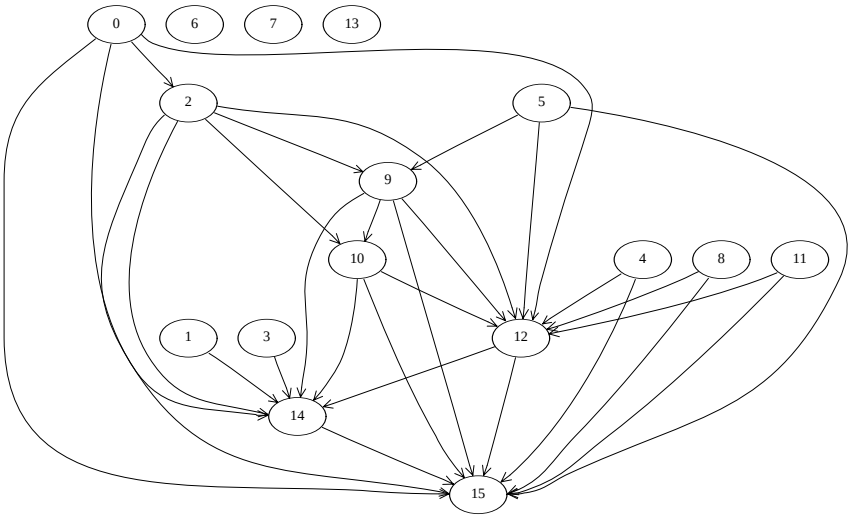

Fig. 11. Threshold-dependent (0.6) distance-graph. $|X| = |K| = |Y| = 16$.

ture allows us to zoom into fine scales (by setting the threshold low and looking at very strong associations), as well as into the coarse scales (by setting the threshold high and looking at very weak associations). If by varying the threshold (and scales) one obtains different distance-graphs, then this would correspond to multiple scales and some hierarchical memory structure. Otherwise (in the case of indistinguishable distance-graphs), one may conclude that the memory structure is flat. The latter case may be observed for either extreme: low connectivity of flat non-associative memories, or high connectivity (e.g., a complete graph) of non-hierarchical associative memories.

In summary, the extent of a revealed memory structure depends on the threshold. At this point we hypothesize that under certain circumstances we may find that optimal memory structures might exhibit different levels of hierarchical or associative memory structures. These preferred structures are *prior* to any particular architecture that is either imposed by design (in engineering) or evolved (in biology), and merely driven by specific information-theoretic requirements. Here, we have collected a set of tools that can help us in this quest.

5 Discussion and Conclusions

In this paper, we have investigated an information-driven evolutionary design of content-addressable memory, and presented a set of tools (the design criteria, grid-contours, embeddings, threshold-dependent distance-graphs) useful for such design. The evolved mappings $Y = f(K, X)$ maximize precision, recall and effectiveness of the potential information transfer throughout associative memory.

It was conjectured (e.g., [28, 31]) that the degree of self-referentiality employed by a self-replicating multi-cellular organism is related to efficiency of its self-inspection and self-repair — and may be quantitatively measured in order to evolve more efficient processes. This conjecture was extended in this work in terms of memory structures and the information transfer. We would like to point out that content-addressable memory model is more generic than a self-referential memory model, and the latter emerges under additional selection pressures. We briefly sketched an example of such a pressure, provided by a closed loop around the system that packs information at multiple scales.

Continuing our analogy with DNA as an associative memory, it is interesting to observe that real-life examples of DNA are not approaching the maximum information transfer, as evidenced by their non-perfect error recovery and significant redundancy (pseudo-genes). Thus, in terms of self-replication, the maximum potential is not realized — it would require higher precision and higher recall, culminating in a perfectly-associative memory. Interestingly, another extreme, lower precision and/or lower recall, can be pointed out already. We believe that a suitable example is the self-replication mechanism exhibited by mineral crystals in the absence of biological enzymes, as advocated by Cairns-Smith [2]: clay crystals can store information as a pattern of inhomogeneities that are propagated from layer to layer, with few errors; they can reproduce by random fragmentation; and they can express a variety of morphological phenotypes. Following this intuition, Schulman and Winfree recently proposed a method of error-correcting self-replication that works by similar growth and fragmentation of algorithmic DNA crystals [36]: “crystal growth extends the layers and copies the se-

quence of orientations, which may be considered its genotype. . . . splitting of a crystal can yield multiple pieces, each containing at least one copy of the entire genotype". Such self-replication can be considered as non-associative memory recall, where a key is not necessary at all, and neither the point of crystal fragmentation nor surrounding environmental conditions are important. In other words, Cairns-Smith model of crystal self-replication is near the low-precision and low-recall extreme, while a self-referential associative memory would implement the highest-effectiveness case.

Adami advocated the view that "evolution increases the amount of information a population harbors about its niche" [1]. The information-theoretic criteria proposed in this work may further formalize the notion of information transfer involved in self-replication, and enable bio-inspired design of more effective memory structures.

Bibliography

[1] Adami, C., 2002, What is complexity?, *Bioessays*, 24(12), 1085–1094.

[2] Cairns-Smith, A.G., 1966, The origin of life and the nature of the primitive gene, *Journal of Theoretical Biology*, 10, 53–88.

[3] Crutchfield, J. P., 1990, Information and its Metric, in: Lam, L. and Morris, H. C., eds., *Nonlinear Structures in Physical Systems Pattern Formation, Chaos and Waves*, 119–130, Springer Verlag.

[4] Deb, K. and Agrawal, R. B. (1995). Simulated binary crossover for continuous search space. *Complex Systems*, 9:115–148.

[5] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6:182–197.

[6] Der, R., Steinmetz, U., Pasemann, F., 1999, Homeokinesis — A new principle to back up evolution with learning, *Concurrent Systems Engineering Series*, 55, 43–47.

[7] Dorigo, M., Maniezzo, V., Colorni, A., 1996, The Ant System: Optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics, B*, 26(1), 1–13.

[8] Dorigo, M., and Di Caro, G., 1999, Ant Colony Optimization: A new meta-heuristic, in: *Proceedings of 1999 Congress on Evolutionary Computation, Washington DC*, 1470–1477, IEEE Press, Piscataway, NJ.

[9] Foreman, M., Prokopenko, M., Wang, P., 2003, Phase Transitions in Self-organising Sensor Networks, in: Banzhaf, W. and Christaller, T. and Dittrich, P. and Kim, J.T. and Ziegler, J., eds., *Advances in Artificial Life - Proceedings of the 7th European Conference on Artificial Life (14-17 September, 2003, Dortmund, Germany) (ECAL-03)*, Lecture Notes in Computer Science, vol. 2801, 781–791, Springer-Verlag, Heidelberg, Germany.

[10] Goertzel, B., 1993, *The structure of intelligence: A new mathematical model of mind*, New York: Springer-Verlag.

[11] Goutte, C., and Gaussier, E., 2005, A probabilistic interpretation of precision, recall and F-score, with implication for evaluation, in: *Proceedings of the 27th European Conference on Information Retrieval*, Santiago de Compostela, Spain.

[12] Heatherton, T.F., Macrae, C.N., Kelley, W.M., 2004, What the social brain sciences can tell us about the self, *Current Directions in Psychological Science*, 13(5), 190–193.

[13] Hofstadter, D. R., 1989, *Gödel, Escher, Bach: An eternal golden braid*, New York: Vintage Books.

[14] Hopfield, J.J., 1982, Neural networks and physical systems with emergent collective computation abilities, *Proceedings of the National Academy of Science*, 79, 2554–2558.

[15] Goldsmith, R.S., and Miller, J.F., 2003, Cooperative co-evolution of robot control, sensor relevance and placement, in: *Proccedings of EPSRC Evolvability and Sensor Evolution Symposium*, Birmingham, U.K.

[16] Kanerva, P., 1988, *Sparse Distributed Memory*, Cambridge, Mass.: MIT Press.
[17] Kanerva, P., 1993, Sparse Distributed Memory and related models. In M.H. Hassoun (ed.), *Associative Neural Memories: Theory and Implementation*, New York: Oxford University Press, 50–76.
[18] Klyubin, A. S., Polani, D., Nehaniv, C. L., 2004, Organization of the information flow in the perception-action loop of evolved agents, in: *Proceedings of 2004 NASA/DoD Conference on Evolvable Hardware*, IEEE Computer Society, 177–180.
[19] Klyubin, A. S., Polani, D., and Nehaniv, C. L., 2005. Empowerment: A universal agent-centric measure of control, in: *Proceedings of The 2005 IEEE Congress on Evolutionary Computation*, 128–135, IEEE Press.
[20] Klyubin, A. S., Polani, D., Nehaniv, C. L., 2005, All else being equal be empowered, in: Capcarrère, M. S., Freitas, A. A., Bentley, P. J., Johnson, C. G., Timmis, J., eds., *Advances in Artificial Life, 8th European Conference, ECAL, 2005, Canterbury, UK, September 5-9, 2005, Proceedings*, Lecture Notes in Computer Science, Vol. 3630, 744–753, Springer.
[21] Klyubin, A. S., Polani, D., Nehaniv, C. L., 2007, Representations of Space and Time in the Maximization of Information Flow in the Perception-Action Loop, *Neural Computation*, accepted 17 October 2006, in press.
[22] Kohonen, T., 1984, *Self-organization and associative memory*, Berlin, Springer-Verlag.
[23] Langton, C., 1991, Computation at the Edge of Chaos: Phase transitions and emergent computation, in: S. Forrest, ed., *Emergent Computation*, MIT.
[24] Latin square, 2007, in: Wikipedia, The Free Encyclopedia. Retrieved 07:30, March 17, 2007, from
`http://en.wikipedia.org/wiki/Latin_square`
[25] MacKay, D. J. C., 2003, *Information theory, inference and learning algorithms*, Cambridge University Press.
[26] Miller, J. F., Job, D., and Vassilev, V. K., 2000, Principles in the Evolutionary Design of Digital Circuits - Part I, *Journal of Genetic Programming and Evolvable Machines*, 1(1):8-35.
[27] Moskowitz, J.P., and Jousselin, C., 1989, An algebraic memory model. *ACM SIGARCH Computer Architecture News archive*, 17(1), 55–62, ACM Press, New York, NY, USA.
[28] Prokopenko, M. and Wang, P., 2004, On Self-referential shape replication in robust aerospace vehicles, in Pollack, J., ed., *Artificial Life IX: Proceedings of The 9th International Conference on the Simulation and Synthesis of Living Systems*, Boston, USA, MIT Press, 27–32.
[29] Prokopenko, M., Wang, P., Price, D. C., Valencia, P., Foreman, M., Farmer, A. J., 2005, Self-organizing hierarchies in sensor and communication networks. *Artificial Life*, Special Issue on Dynamic Hierarchies, 11(4), 407–426.
[30] Prokopenko, M., Wang, P., Foreman, M., Valencia, P., Price, D.C., Poulton, G. T., 2005, On connectivity of reconfigurable impact networks in Ageless Aerospace Vehicles. *The Journal of Robotics and Autonomous Systems*, 53(1), 36–58.
[31] Prokopenko, M., Poulton, G., Price, D. C., Wang, P., Valencia, P., Hoschke, N., Farmer, A. J., Hedley, M., Lewis, C., Scott, D. A., 2006, Self-organising impact

sensing networks in robust aerospace vehicles, in: Fulcher, J., ed., *Advances in Applied Artificial Intelligence*, Idea Group, 186–233.

[32] Prokopenko, M., Gerasimov, V., Tanev, I., 2006, Measuring spatiotemporal coordination in a modular robotic system, in: Rocha, L. M., Yaeger, L. S., Bedau, M. A., Floreano, D., Goldstone, R. L., Vespignani, A., eds., *Artificial Life X: Proceedings of The 10th International Conference on the Simulation and Synthesis of Living Systems*, 185–191, Bloomington IN, USA.

[33] Prokopenko, M., Gerasimov, V., Tanev, I., 2006, Evolving spatiotemporal coordination in a modular robotic system, in: Nolfi, S., Baldassarre, G., Calabretta, R., Hallam, J. C. T., Marocco, D., Meyer, J.-A., Miglino, O., and Parisi, D., eds., *From Animals to Animats 9: 9th International Conference on the Simulation of Adaptive Behavior (SAB 2006), Rome, Italy, September 25-29 2006*, Lecture Notes in Computer Science, vol. 4095, 558–569.

[34] Prokopenko, M., Polani, D., Wang P., 2006, Optimizing Potential Information Transfer with Self-referential Memory, in: Calude, C. S., Dinneen, M. J., Paun, G., Rozenberg, G., and Stepney S., eds., *Unconventional Computation : 5th International Conference (UC '06), York, UK*, Springer, Lecture Notes in Computer science, vol. 4135, 228–242.

[35] Rogers, T.B., Kuiper, N.A., Kirker, W.S., 1977, Self-reference and the encoding of personal information, *Journal of Personality and Social Psychology*, 35(9), 677–688.

[36] Schulman, R., and Winfree, E., 2005, Self-replication and evolution of DNA crystals, in: Capcarrère, M. S., Freitas, A. A., Bentley, P. J., Johnson, C. G., Timmis, J., eds., *Advances in Artificial Life, 8th European Conference, ECAL 2005, Canterbury, UK, September 5-9, 2005, Proceedings*, 734–743, Springer.

[37] Tanev, I., Ray, T., Buller, A., 2005, Automated evolutionary design, robustness, and adaptation of sidewinding locomotion of a simulated snake-like robot, *IEEE Transactions On Robotics*, 21, 632–645.