A Radically New Theory of how the Brain Represents and Computes with Probabilities

Gerard (Rod) Rinkus[0000-0003-1725-910X]

[1] Neurithmic Systems, Newton, MA 02465 USA
rod@neurithmicsystems.com

Abstract. Many believe that the brain implements probabilistic reasoning and that it represents information via some form of population (distributed) code. Most prior probabilistic population coding (PPC) theories share basic properties: 1) continuous-valued units; 2) fully/densely distributed codes; 3) graded synapses; 4) rate coding; 5) units have innate low-complexity, usually unimodal, tuning functions (TFs); and 6) units are intrinsically noisy and noise is generally considered harmful. I describe a radically different theory that assumes: 1) binary units; 2) sparse distributed codes (SDC); 3) *functionally* binary synapses; 4) a novel, *atemporal*, combinatorial spike code; 5) units initially have flat TFs (all weights zero); and 6) noise is a resource generated/used, normatively, to cause similar inputs to map to similar codes. The theory, Sparsey, was introduced 25+ years ago as: a) an explanation of the physical/computational relationship of episodic and semantic memory for the spatiotemporal (sequential) pattern domain; and b) a canonical, mesoscale cortical probabilistic circuit/algorithm possessing fixed-time, unsupervised, single-trial, non-optimization-based, unsupervised learning and fixed-time best-match (approximate) retrieval; but was not described as an alternative to PPC-type theories. Here, we show that: a) the active SDC in a Sparsey coding field (CF) simultaneously represents not only the likelihood of the single most likely input but the likelihoods of all hypotheses stored in the CF; and b) the whole explicit distribution can be sent, e.g., to a downstream CF, via a set of simultaneous single spikes from the neurons comprising the active SDC.

Keywords: Sparse distributed representations, probabilistic population coding, cell assemblies, canonical cortical circuit/algorithm.

1 Introduction

It is widely believed that the brain implements some form of probabilistic reasoning to deal with uncertainty in the world [1], but exactly how the brain represents probabilities/likelihoods remains unknown [2, 3]. It is also widely agreed that the brain represents information with some form of distributed—a.k.a. population, cell-assembly, ensemble—code [see [4] for relevant review]. Several population-based probabilistic coding theories (PPC) have been put forth in recent decades, including those in which the state of all neurons comprising the population, i.e., the *population code*, is viewed as representing: a) the single most likely/probable input value/feature [5]; or b) the

entire probability/likelihood distribution over features [6-11]. Despite their differences, these approaches share fundamental properties, a notable exception being the spike-based model of [11]. (1) Neural activation is continuous (graded). (2) *All* neurons in the coding field (CF) formally participate in the active code whether it represents a single hypothesis or a distribution over all hypotheses. Such a representation is referred to as a *fully distributed r*epresentation. (3) Synapse strength is continuous. (4) They are typically formulated in terms of rate-coding [12]. (5) They assume *a priori* that *tuning functions* (TFs) of the neurons are unimodal, e.g., bell-shaped, over any one dimension, and consequently do not explain how such TFs might naturally emerge, e.g., through a learning process. (6) Individual neurons are assumed to be intrinsically noisy, e.g., firing with Poisson variability, and noise is viewed primarily as a problem that to be dealt with, e.g., reducing noise correlation by averaging.

At a deeper level, it is clear that despite being framed as population models, they are really based on an underlying localist interpretation, specifically, that an individual neuron's firing rate can be taken as a perhaps noisy estimate of the probability that a single preferred feature (or preferred value of a feature) is present in its receptive field [13], i.e., consistent with the "Neuron Doctrine". While these models entail some method of combining the outputs of individual neurons, e.g., averaging, each neuron is viewed as providing its own individual, i.e., localist, estimate of the input feature. For example, this can be seen quite clearly in Fig. 1 of [9] wherein the first layer cells (sensory neurons) are unimodal and therefore can be viewed as detectors of the value at their modes (preferred stimulus) and the pooling cells are also in 1-to-1 correspondence with directions. This localist view is present in the other PPC models referenced above as well.

However, there are compelling arguments against such localistically rooted conceptions. From an experimental standpoint, a growing body of research suggests that individual cell TFs are far more heterogeneous than classically conceived [14-21], also described as having "mixed selectivity" [22], and more generally, that sets (populations, ensembles) of cells, i.e., "cells assemblies" [23], constitute the fundamental representational units in the brain [24, 25]. And, the greater the fidelity with which the heterogeneity of TFs is modeled, the less neuronal response variation that needs to be attributed to noise, leading some to question the appropriateness of the traditional concept of a single neuron I/O function as an invariant TF plus noise [26]. From a computational standpoint, a clear limitation is that the maximum number of features/concepts, e.g., oriented edges, directions of movement, that can be stored in a localist coding field of N units is N. More importantly, as explained here, the efficiency, in terms of time and energy, with which features/concepts can be stored (learned) and retrieved/transmitted is far greater if items of information (memories, hypotheses) are represented with *sparse distributed codes* (SDCs) rather than localistically [27-29].

The theory described herein, Sparsey [27-29], constitutes a radically new way of representing and computing with probabilities, diverging from most existing PPC theories in many fundamental ways, including: (1) The representational units (principal cells) comprising a CF need only be *binary*. (2) Individual items (hypotheses) are represented by *fixed-size*, sparsely chosen subsets of the CF's units, referred to as *modular sparse distributed codes* (MSDCs), or simply "codes" if unambiguous. (3) Decoding (read-out) not only of the most likely hypothesis but of the whole distribution, i.e., the

likelihoods of *all* hypotheses stored in a CF, by downstream computations, requires only binary synapses. (4) The whole distribution, is sent via a wave of effectively simultaneous (i.e., occurring within some small window, e.g., at some phase of a local gamma cycle [30-33]) single spikes from the units comprising an active code to a downstream (possibly recurrently to the source) CF. (5) The initial weights of all afferent synapses to a CF are zero, i.e., the TFs are completely flat. The classical, roughly unimodal TFs [as would be revealed by low-complexity probes, e.g., oriented bars spanning a cell's receptive field (RF), cf. [34]] emerge as a side-effect of the model's single/few-trial learning process of storing MSDCs in superposition [35]. (6) Neurons are not assumed to be intrinsically noisy. However, the canonical, mesoscale (i.e., the cell assembly scale) circuit normatively uses noise as a resource during learning. Specifically, noise, presumably mediated by neuromodulators, e.g., ACh [36], NE [37], is explicitly injected into the code selection process to achieve the specific goal of (statistically, approximately) mapping more similar inputs to more similar MSDCs, where the similarity measure for MSDCs is intersection size. In this approach, patterns of correlation amongst principal cells are simply artifacts of this learning process.

2 The Model

Fig. 1a shows a small Sparsey model instance with an 8x8 binary units (pixel) input field that is fully connected, via binary weights (blue lines), all initially zero, to a *modular sparse distributed coding* (MSDC) coding field (CF). The CF consists of Q winner-take-all (WTA) *competitive modules* (CMs), each consisting of K binary neurons. Here, Q=7 and K=7. Thus, all codes have exactly Q active neurons and there are K^Q possible codes. We refer to the input field as the CF's receptive field (RF). Fig. 1b shows a particular input A, which has been associated with a particular code, ϕ(A) (black units); here, blue lines indicate the bundle [cf. "Synapsemble", [30]] of weights that would be increased from 0 to 1 to store this association (memory trace).

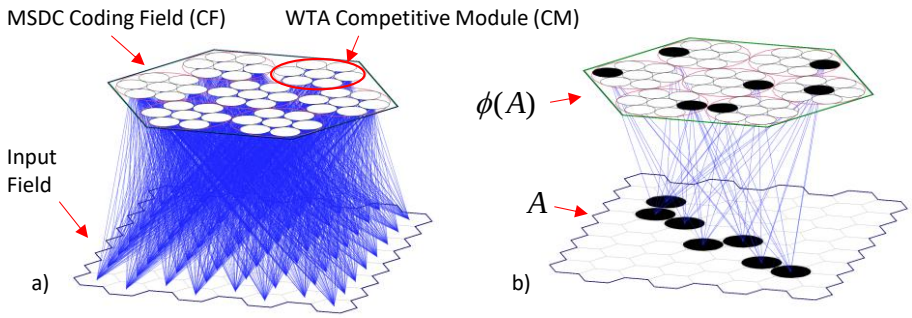

Fig. 1. The *modular sparse distributed code* (MSDC) coding field (CF). See text.

Fig. 2 illustrates MSDC's key property that: ***whenever any <u>one</u> code is fully active in a CF, i.e., all Q of its units are active, <u>all</u> codes stored in the CF will simultaneously be active (in superposition) in proportion to the sizes of their intersections with the***

single maximally active code. Fig. 2 shows five hypothetical inputs, A-E, which have been learned, i.e., associated with codes, ϕ(A) - ϕ(E). These codes were manually chosen to illustrate the principle that *similar inputs* should map to *similar codes* ("SISC"). That is, inputs B to E have progressively smaller overlaps with A and therefore codes ϕ(B) to ϕ(E) have progressively smaller intersections with ϕ(A). Although these codes were manually chosen, Sparsey's *Code Selection Algorithm* (CSA), described shortly, has been shown to statistically enforce SISC for both the spatial and spatiotemporal (sequential) input domains [27-29, 38, 39]: a simulation-backed example for the spatial domain is given in the Results section.

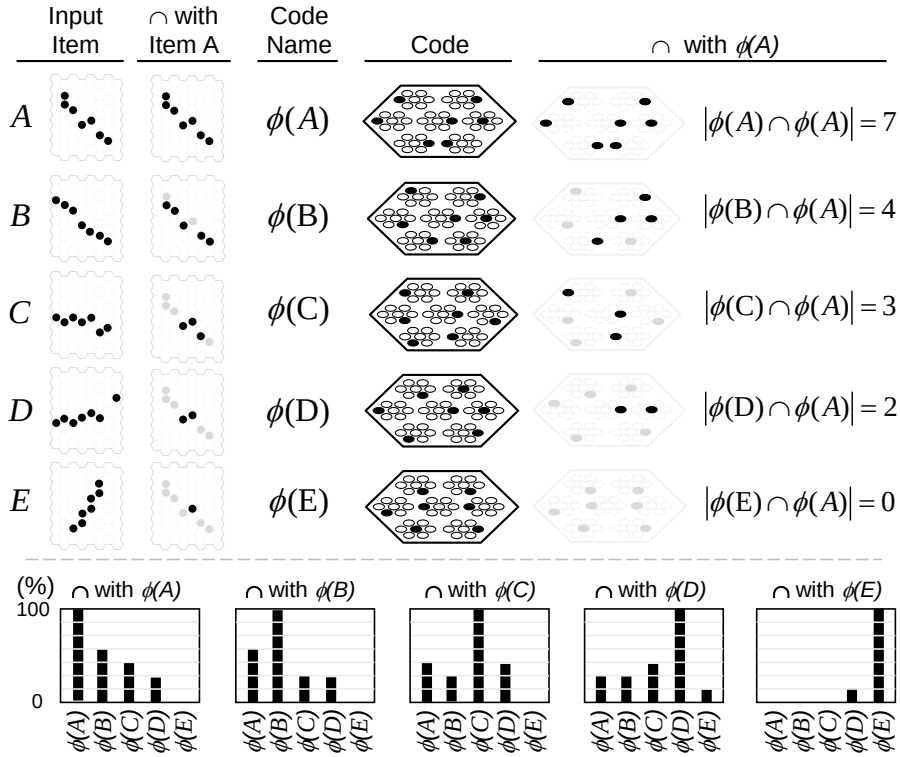

Fig. 2. The probability/likelihood of a feature can be represented by the fraction of its code that is active. When ϕ(A) is fully active, the hypothesis that feature A is present can be considered maximally probable. Because the similarities of the other features to the most probable feature, A, correlate with their codes' overlaps with ϕ(A), their probabilities/likelihoods are represented by the fractions of their codes that are active. In "∩" columns, black units are those intersecting with the input A and with its code, ϕ(A); gray indicates non-intersecting units.

For input spaces for which it is plausible to assume that input similarity correlates with probability/likelihood, i.e., for vast regions of natural input spaces, the single active code can therefore also be viewed as a probability/likelihood distribution over all stored codes. This is shown in the lower part of Fig. 2. The leftmost panel at the bottom of Fig. 2 shows that when ϕ(A) is 100% active, the other codes are partially active in

proportions that reflect the similarities of their corresponding inputs to A, and thus the probabilities/likelihoods of the inputs they represent. The remaining four panels show input similarity (probability/likelihood) approximately correlating with code overlap when each of the four other stored codes is maximally active.

2.1 The Learning Algorithm

A simplified version of the CSA, sufficient for this paper's examples involving only purely spatial inputs, is given in Table 1 and we briefly summarize it here. [The full model handles spatiotemporal inputs, multiplicatively combining bottom-up, top-down, and horizontal (i.e., signals from codes active on the prior time step via recurrent synaptic matrices) inputs to a CF.] CSA Step 1 computes the raw input sums (u) for all $Q \times K$ cells comprising the coding field. In Step 2, these sums are normalized to U values, in [0,1]. All inputs are assumed to have the same number of active pixels, thus the normalizer, π_U, can be constant. In Step 3, we find the max U in each CM and in Step 4, a measure of the familiarity of the input, G, is computed as the average max U across the Q CMs. In Steps 5 and 6, G is used to adjust the parameters of the nonlinear I/O transform in the same way for all of the CF's units. In Step 7, each unit applies that "U-to-μ" transform, yielding an intermediate variable, μ, representing an unnormalized probability of the unit being chosen winner in its respective CM. Step 8 renormalizes μ values to total probabilities (ρ) of winning (within each CM) and Step 9 is the final draw from the ρ distribution in each CM resulting in the final code. G's influence on the distributions can be summarized as follows.

a) When high global familiarity is detected ($G \approx 1$), those distributions are exaggerated to bias the choice in favor of cells that have high input summations, and thus, high *local* familiarities (U), which acts to increase correlation.

b) When low global familiarity is detected ($G \approx 0$), those distributions are flattened so as to reduce bias due to local familiarity, which acts to increase the expected Hamming distance between the selected code and previously stored codes, i.e., to decrease correlation.

Since the U values represent *signal*, exaggerating the U distribution in a CM increases signal whereas flattening it increases noise. The above behavior (and its smooth interpolation over the range, G=1 to G=0) is the means by which Sparsey achieves SISC. And, it is the enforcement (statistically) of SISC during learning, which ultimately makes possible the immediate, i.e., *fixed time* (the number of algorithmic steps needed to do the retrieval is independent of the number of stored codes), retrieval of the best-matching (most likely, most relevant) hypothesis.

Table 1 Simplified Code Selection Algorithm (CSA)

	Equation	Short Description
1	$u_i = \sum_{j \in \mathrm{RF_U}} x(j) w(j,i)$	Compute raw input (u) sums.
2	$U_i = u_i / \pi_U w_{\max}$	Compute normalized input sums. π_U

3	$\hat{U}_q = \max_{i \in CM_q} U_i$	Find the max U, $\hat{U}_q$, in each CM, CM_q
4	$G = \sum_{q=1}^{Q} \hat{U}_k \Big/ Q$	Compute the input's *familiarity*, G, as average $\hat{U}$ value over the Q CMs.
5	$\eta = 1 + \left(\left[\frac{G - G_-}{1 - G_-} \right]^+ \right)^{\gamma} \times \chi \times K$	Determine expansivity (η) of U-to-μ sigmoid function. In this paper, γ=2, χ=100, G^-=0.1.
6	$\sigma_1 = \frac{\left((\eta - 1)/0.001\right)^{1/\sigma_4} - 1}{e^{\sigma_2 \sigma_3}}$	Sets σ_1 so that the overall sigmoid shape is preserved over full η range. σ_2= 7, σ_3= 0.4, σ_4= 9.5.
7	$\mu_i = \frac{(\eta - 1)}{\left(1 + \sigma_1 e^{-\sigma_2 (U_i - \sigma_3)}\right)^{\sigma_4}} + 1$	To each cell, apply sigmoid function, which collapses to constant fn, μ_i=1, when $G \le G^-$.
8	$\rho_i = \mu_i \Big/ \sum_{k \in CM} \mu_k$	In each CM, normalize relative (μ) to final (ρ) probabilities of winning.
9	Select a final winner in each CM according to the ρ distribution in that CM.	

3 Results

The simulation-backed example of this section demonstrates that the CSA achieves the property, i.e., statistical (approximate) preservation of similarity from inputs to codes, qualitatively described in Fig. 2. In the experiment, the six inputs, I_1 to I_6, at top of Fig. 3a, were presented once each and assigned to the codes, $\phi(I_1)$ to $\phi(I_6)$ (not shown), via execution of the CSA (Table 1). The six inputs are disjoint only for simplicity of exposition. The input field (receptive field, RF) is a 12x12 binary pixel array and all inputs are of the same size, 12 active pixels. Since all inputs have exactly 12 active pixels, input similarity is simply $sim(I_x,I_y) = |I_x \cap I_y|/12$, shown as decimals under inputs. The CF consists of Q=24 WTA CMs, each having K=8 binary cells. The second row of Fig. 3a shows a novel stimulus, I_7, and its varying overlaps (yellow pixels) with I_1 to I_6. Fig. 3b shows the code, $\phi(I_7)$, activated (by the CSA) in response to presentation of I_7. Black indicates cells that also won for I_1, red indicates active cells that did not win for I_1, and green indicates inactive cells that did win for I_1. Fig. 3c shows (using the same color interpretations) the detailed values of all relevant variables (u, U, μ, and ρ) computed by the CSA when I_7 presents, and the winners drawn from the ρ distribution in each of the Q=24 CMs.

If we consider presentation of I_7 to be a retrieval test, then the desired result is that the code of the most similar stored input, I_1. should be retrieved (reactivated). In this case, the red and green cells in a given CM can be viewed as substitution errors, i.e., the green cell had the max U value in the CM and should have been reactivated, but since the final winner is a draw, occasionally a cell with a (possibly much) lower U value wins (CMs, 0, 12, 17). However, these are sub-symbolic scale errors, not errors at the scale of whole inputs (hypotheses), as a whole input is *collectively* represented by the entire MSDC code (entire *cell assembly*). In this example, appropriate threshold

settings in downstream computations, would allow the model as a whole to return the correct answer given that 18 out of 24 cells of I_1's code, $\phi(I_1)$, are activated, similar to thresholding schemes in other associative memory models [40, 41].

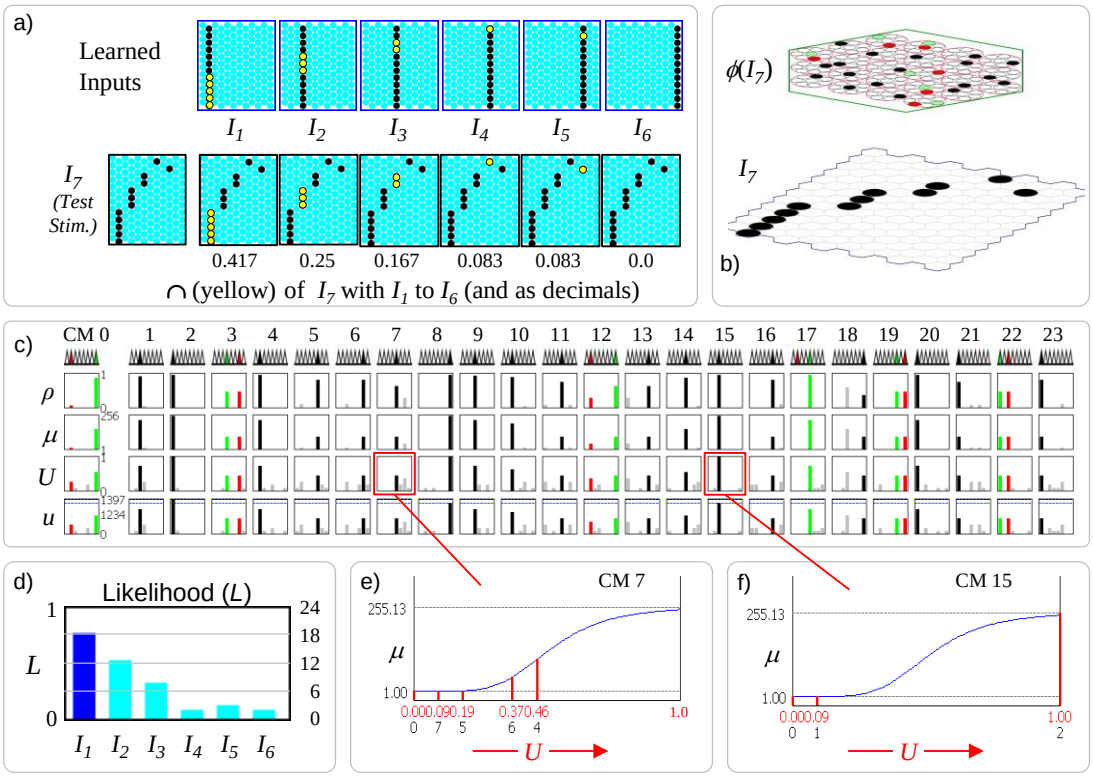

Fig. 3. In response to a novel input, I_7, the codes for the six previously learned (stored) inputs, I_1 to I_6, i.e., hypotheses, are activated with strength approximately correlated with the similarity (pixel overlap) of I_7 input and those stored inputs. Test input I_7 is most similar to learned input, I_1, shown by the intersections (yellow pixels) in panel a. Thus, the code with the largest fraction of active cells is $\phi(I_1)$ (18/24=75%) (blue bar in panel d). The other codes of the other inputs are active in rough proportion to their similarities with I_7 (cyan bars). (c) Raw (u) and normalized (U) input summations to all cells in all CMs. Note: all weights are effectively binary, though "1" is represented with 127 and "0" with 0. Hence, the max u value possible in any cell when I_7 is presented is 12x127=1524. The U values are transformed to un-normalized win probabilities (μ) in each CM via a sigmoid transform whose properties, e.g., max value of 255.13, depend on G and other parameters. μ values are normalized to true probabilities (ρ) and one winner is chosen in each CM (indicated in row of triangles: black: winner for I_7 that also won for I_1; red: winner for I_7 that did not win I_1: green: winner for I_1 that did not win for I_7. (e, f) Details for CMs, 7 and 15. Values in lower row of U axis are indexes of cells (within the CM) having the U values above them (red). Some CMs have a single cell with much higher U (and thus ρ) value than the rest (e.g., CM 15), some CMs have two cells tied for the max (CMs 3, 19, 22).

More generally, when I_7 Is presented, we would like *all* of the stored inputs to be reactivated in proportion to their similarities to the test probe, I_7. Fig. 3d shows that this approximately occurs. The active fractions of the codes, $\phi(I_1)$ to $\phi(I_6)$, are highly rank-correlated with the pixel-wise similarities of the corresponding inputs to I_7. Thus, the blue bar in Fig. 3d represents the fact that the code, $\phi(I_1)$, for the best matching stored input, I_1, has the highest active code fraction, 75% (18 out 24, the black cells in Fig. 3b) cells of $\phi(I_1)$ are active in $\phi(I_7)$. The cyan bar for the next closest matching stored input, I_2, indicates that 12 out of 24 of the cells of $\phi(I_2)$ (code note shown) are active in $\phi(I_7)$. In general, many of these 12 may be common to the 18 cells in $\{\phi(I_7)\cap\phi(I_1)\}$. And so on for the other stored hypotheses. [Note that even the code for I_6 which has zero intersection with I_7 has two cells in common with $\phi(I_1)$. In general, the expected code intersection for the zero input intersection condition is not zero, but chance, since in that case, the winners are chosen from the uniform distribution in each CM, in which case the expected intersection is Q/K.]

If, instead of viewing presentation of I_7 as a retreival test, we view it as a learning trial, we want the sizes of intersection of the code, $\phi(I_7)$, activated in response, with the six previously stored codes, $\phi(I_1)$ to $\phi(I_6)$, to approximately correlate with the similarities of I_7 to inputs, I_1 to I_6. But again, this is what Fig. 3d shows. As noted earlier, we assume that the similarity of a stored input I_x to the current input can be taken as a measure of I_x's probability/likelihood. And, since all codes are of size Q, we can divide code intersection size by Q, yielding a measure normalized to [0,1]: $L(I_1)=|\phi(I_7)\cap\phi(I_1)|/Q$. Thus, this result demonstrates that the CSA, a single-trial, unsupervised, non-optimization-based, and *most importantly, fixed time,* algorithm statistically enforces SISC. In this case, the red cells would not be considered errors: they would just part of a new code, $\phi(I_7)$, being assigned to represent a novel input, I_7, in a way that respects similarity in the input space. Crucially, because all codes are stored in *superposition* and because, when each one is stored, it is stored in a way that respects similarities with all previously stored codes, the patterns of intersection amongst the set of stored codes reflects not simply the *pairwise* similarity structure over the inputs, but, in principle, the similarity structure *of all orders present* in the input set. This is similar in spirit to another neural probabilistic model [2, 42], which proposes that overlaps of distributed codes (and recursively, overlaps of overlaps), encode the domain's latent variables (their identities and valuednessess), cf. "anonymous latent variables", [43].

The likelihoods in Fig. 3d may seem high. After all, I_7 has less than half its pixels in common with I_1, etc. Given these particular input patterns, is it really reasonable to consider I_1 to have such a high likelihood? Bear in mind that our example assumes that the only experience this model has of the world are single instances of the six inputs shown. We assume no prior knowledge of any underlying statistical structure generating the inputs. Thus, it is really only the *relative* values that matter and we could pick other parameters, notably in CSA Steps 5-7, that would result in a much less expansive sigmoid nonlinearity, which would result in lower expected intersections of $\phi(I_7)$ with the learned codes, and thus lower likelihoods. The main point is simply that the expected code intersections correlate with input similarity, and thus, likelihood.

A cell's U value represents the total *local evidence* that it should be activated. However, rather than simply picking the max U cell in each CM as winner (i.e., hard max),

which would amount to executing only steps 1-3 of the CSA, the remaining CSA steps, 4-9, are executed, in which the U distributions are transformed as described earlier and winners are chosen as draws (shown in the row of triangles just below CM indexes) from the ρ distributions in each CM. Thus, an extremely cheap-to-compute (CSA Step 4) *global* function of the whole CF, G, is used to influence the *local* decision process in each CM. We repeat for emphasis that no part of the CSA explicitly operates on, i.e., iterates over, stored hypotheses (codes); indeed, there are no explicit (localist) representations of stored hypotheses on which to operate.

Fig. 4 shows that presentation of different novel inputs yields different likelihood distributions that correlate approximately with similarity. Input I_8 (Fig. 4a) has highest intersection with I_2 and a different pattern of intersections with the other learned inputs as well (refer to Fig. 3a). Fig. 4c shows that the codes of the stored inputs become active in approximate proportion to their similarities with I_8, i.e., their likelihoods are simultaneously physically represented by the fractions of their codes which are active. The G value in this case, 0.65, yields, via CSA steps 5-7, the U-to-μ transform shown in Fig. 4b, which is applied in all CMs. Its range is [1,300] and given the particular U distributions shown in Fig. 4d, the cell with the max U in each CM ends up being greatly favored over other lower-U cells. The red box shows the U distribution for CM 9. The second row of the abscissa in Fig. 4b gives the within-CM indexes of the cells having the corresponding (red) values immediately above (shown for only three cells). Thus, cell 3 has U=0.74 which maps to approximately $\mu \approx 250$ whereas its closest competitors, cells 4 and 6 (gray bars in red box) have U=0.19 which maps to $\mu = 1$. Similar statistical conditions exist in most of the other CMs. However, in three of them, CMs 0, 10, and 14, there are two cells tied for max U. In two, CMs 10 and 14, the cell that is not contained in I_2's code, $\phi(I_2)$, wins (red triangle and bars), and in CM 0, the cell that is in $\phi(I_2)$ does win (black triangle and bars). Overall, presentation of I_8 activates a code $\phi(I_8)$ that has 21 out of 24 cells in common with $\phi(I_2)$ manifesting the high likelihood estimate for I_2.

Finally. Fig. 4e shows presentation of a more ambiguous input, I_9, having half its pixels in common with I_3 and the other half with I_6. Fig. 4g shows that the codes for I_3 and I_6 have both become approximately equally (with some statistical variance) active and are both more active than any of the other codes. Thus, the model is representing that these two hypotheses are the most likely and approximately equally likely. The exact bar heights fluctuate somewhat across trials, e.g., sometimes I_3 has higher likelihood than I_6, but the general shape of the distribution is preserved. The remaining hypotheses' likelihoods also approximately correlate with their pixelwise intersections with I_9. The qualitative difference between presenting I_8 and I_9 is readily seen by comparing the U rows of Fig. 4d and 4h and seeing that for the latter, a tied max U condition exists in almost all the CMs, reflecting the equal similarity of I_9 with I_3 and I_6. In approximately half of these CMs the cell that wins intersects with $\phi(I_3)$ and in the other half, the winner intersects with $\phi(I_6)$. In Fig. 4h, the three CMs in which there is a single black bar, CMs 1, 7, and 12, indicates that the codes, $\phi(I_3)$ and $\phi(I_6)$, intersect in these three CMs.

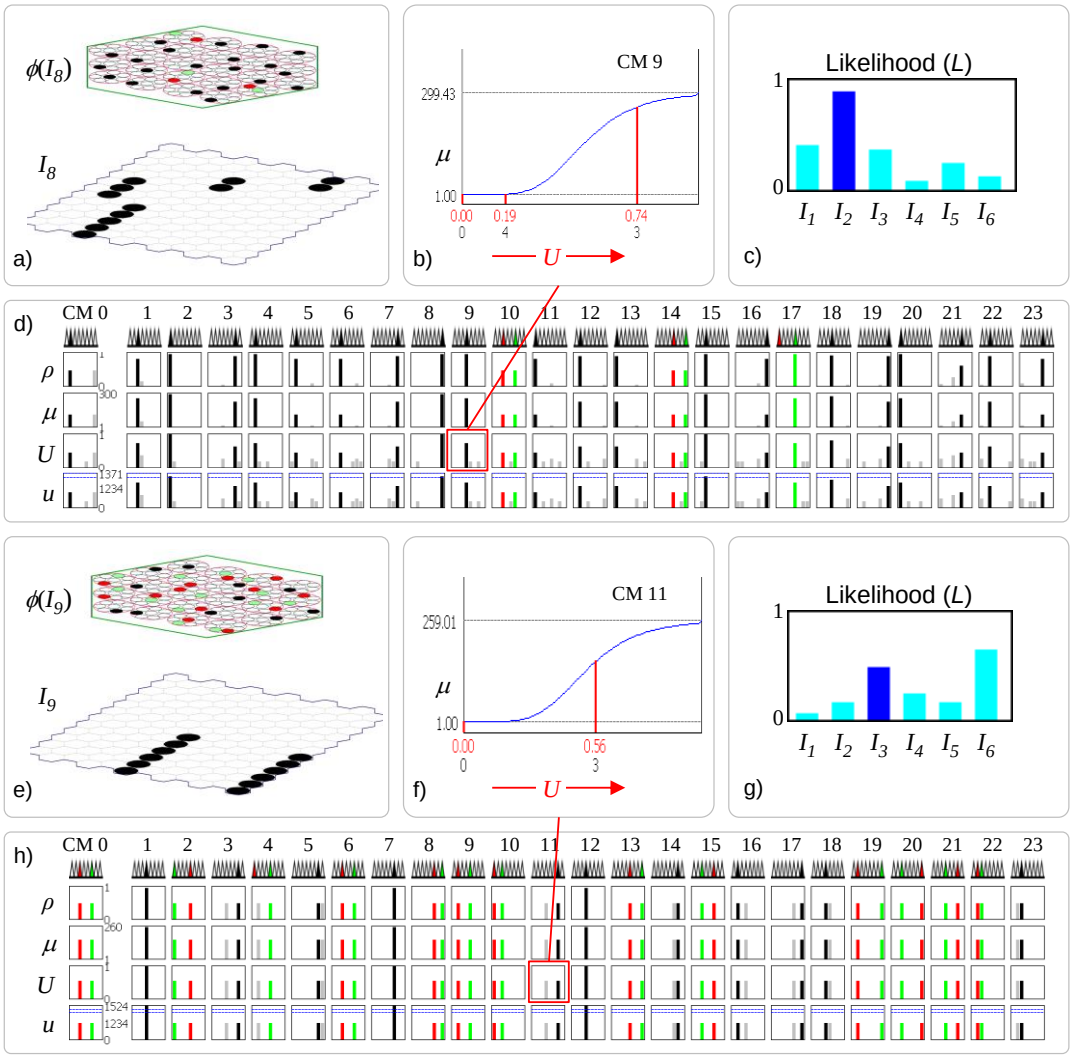

Fig. 4. Details of presenting other novel inputs, I_8 (panels a-d) and I_9 (panels e-h). In both cases, the resulting likelihood distributions (panels c,g) correlate closely with the input overlap patterns. Panels b and f show details of one example CM (red boxes in panels d and h) for each input.

3.1 A MSDC simultaneously transmits the full likelihood distribution via an atemporal combinatorial spike code

The use of MSDC allows the likelihoods of *all* hypotheses stored in the distribution, to be transmitted via a set of simultaneous single spikes from the neurons comprising the active MSDC. This is shown in the example given in Fig. 5e, which, at the same time, compares this fundamentally new *atemporal, combinatorial spike code*, with

temporal spike codes and one prior (in principle) atemporal code. For a single source neuron, two types of spike code are possible, rate (frequency) (Fig. 5b), and latency (e.g., of spike(s) relative to an event, e.g., phase of gamma) (Fig. 5c). Both are fundamentally temporal and have the crucial limitation that only one value (item) represented by the source neuron can be sent at a time. Most prior population-based codes also remain fundamentally temporal: the signal depends on spike *rates* of the afferent axons, e.g., [1-4] (not shown in Fig. 5).

Fig. 5d illustrates an (effectively) atemporal population code [5] in which the *fraction* of active neurons in a source field carries the message, coded as the number of simultaneously arriving spikes to a target neuron (shown next to the target neuron for each of the four signals values). This *variable-size* population (a.k.a. "thermometer") code has the benefit that all signals are sent in the same, short time, but it is not combinatorial in nature, and has limitations, including: a) the max number of representable values (items/concepts) is the number (N) of units comprising the source CF; and b) as for the temporal codes defined with respect to a single source neuron, any single message sent can represent *only one* item, e.g., a single value of a scalar variable, i.e., implying that any one message carries only $\log_2 N$ bits.

In contrast, consider the *fixed-size* MSDC code of Fig. 5e. The source CF consists of Q=5 CMs, each with K=4 binary units. Thus, all codes, are of the same fixed size, Q=5. As done in Fig. 2, the codes for this example were manually chosen to reflect the similarity structure of scalar values (Col. a) (the prior section has already demonstrated that the CSA statistically preserves similarity). As suggested by charts at right of Fig. 5, any single MSDC, ϕ_1, represents (encodes) the similarity distribution over *all* items (values) stored in the field. Note: blue denotes active units *not* in the intersection with ϕ_1. We're assuming that input (e.g., scalar value) similarity correlates with likelihood, which again, is reasonable for vast portions of input spaces having natural statistics.

Since any one MSDC, ϕ_1, encodes the full likelihood distribution, the set of single spikes sent from it simultaneously transmits that full distribution, encoded as the instantaneous sums at the target cells. Note: when any MSDC, ϕ_1, is active, 20 wts (axons) will be active (black), thus, all four target cells will have Q=5 active inputs. Thus, due to the *combinatorial* nature of the MSDC code, the specific values of the binary weights are essential to describing the code (unlike the other codes where we can assume all wts are 1). Thus, for the example of Fig. 5e, we assume: a) all wts are initially 0; b) the four associations, $\phi_1 \rightarrow$ target cell 1, $\phi_2 \rightarrow$ target cell 2, etc., were previously stored (learned) with single trials; and c) on those learning trials, coactive pre-post synapses were increased to w=1. Thus, if ϕ_1 is reactivated, target cell 1's input sum will be 5 and other cells' sums will be as shown (to left of target cells). If ϕ_2 is reactivated, target cell 2's input sum will be 5, etc. [Black line: active w=1; dotted line: active w=0; gray line: w=0.] As described in Fig. 3 of [7], the four target cells could be embedded in a recurrent field with inhibitory infrastructure allowing sequential read out in descending input sum order, implying that the full similarity (likelihood) order information over all four stored items is sent in each of the four cases. As there are 4! orderings of the four items, each such message, each a set of 20 simultaneous spikes sent from five active CF units, sends $\log_2(4!)$=4.58 bits. We suggest this marriage of fixed-size MSDCs and an atemporal spike code is a crucial advance beyond prior population-based models,

i.e., the "distributional encoding" models (see [8, 9] for reviews), and may be key to explaining the speed and efficiency of probabilistic computation in the brain.

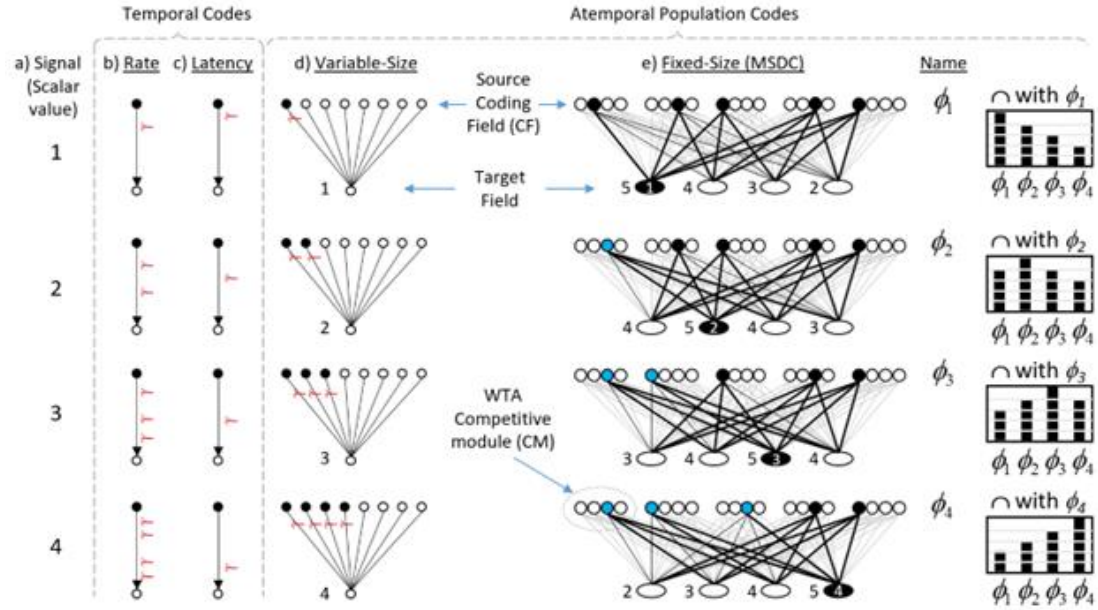

Fig. 5. Temporal vs. atemporal spike coding concepts. The fixed-size MSDC code has the advantage of being able to send the entire distribution, i.e., the likelihoods of *all* codes (hypotheses) stored in the source CF, with a set of simultaneous single spikes from Q=5 units comprising an active MSDC code. See text for details.

4 Discussion

We described a radically different theory, from prevailing probabilistic population coding (PPC) theories, for how the brain represents and computes with probabilities. This theory, Sparsey, avails itself only in the context of *modular sparse distributed coding* (MSDC), as opposed to the fully distributed coding context in which the PPC models have been developed (or a localist context). The theory, Sparsey, was introduced 25+ years ago, as a model of the canonical cortical circuit and a computationally efficient explanation of episodic and semantic memory for sequences, but its interpretation as a way of representing and computing with probabilities was not emphasized. The PPC models [5, 7-10, 12, 42] share several fundamental properties: 1) continuous neurons; 2) full/dense coding; 3) due to 1 and 2, synapses must either be continuous or rate coding must be used to allow decoding; 4) they generally assume rate coding; 5) individual neurons are generally assumed to have unimodal, e.g., bell-shaped, tuning functions (TFs); 6) individual neurons are assumed to be noisy, and noise is generally viewed as degrading computation, thus, needing to be mitigated, e.g., averaged out.

In contrast to these PPC properties/assumptions, Sparsey assumes: 1) binary neurons; 2) items of information are represented by small (relative to whole CF) sets of

neurons (MSDCs) and any such code simultaneously represents not only the likelihood of the single best matching stored hypothesis, but, simultaneously, the likelihoods of *all* stored hypotheses; 3) only effectively binary synapses; 4) signaling via waves of simultaneous single (e.g., first) spikes from a source MSDC; 5) all weights are initially zero, i.e., the TFs are initially completely flat, and emerge via single/few-trial, unsupervised learning to reflect a neuron's specific history of inclusion in MSDCs; 6) rather than being viewed as a problem imposed by externalities (e.g., common input, intrinsically noisy cell firing), noise functions as a resource, controlled usage of which yields the valuable property that similar inputs are mapped to similar codes (SISC).

The CSA's algorithmic efficiency, i.e., both learning (storage) and best-match retrieval are fixed time operations, has not been shown for any other computational method, including hashing methods, either neurally-relevant [44-46], or more generally [reviewed in [47]]. Although time complexity considerations like these have generally not been discussed in the PPC literature, they are essential for evaluating the overall plausibility of models of biological cognition, for while it is uncontentious that the brain computes probabilistically, we also need to explain the extreme speed with which these computations, over potentially quite large hypothesis spaces, occur.

One key to Sparsey's computational speed is its extremely efficient method of computing the *global* familiarity, G, simply as the average of the max U values of the Q CMS. In particular, computing G *does not require* explicitly comparing the new input to every stored input (nor to a log number of the stored inputs as is the case for tree-based methods). G is then used to adjust, in the same way, the transfer functions of all neurons in a CF. This dynamic, and fast timescale (e.g., 10 ms), modulation of the transfer function, based on the *local* (to the CF, thus *mesoscale* circuit) measure, G, is a strongly distinguishing property of Sparsey: in most models, the transfer function is static. While there has been much discussion about the nature, causes, and uses, of correlations and noise in cortical activity; see [48-50] for reviews, the G-based titration of the amount of noise present in the code selection process, to achieve the specific goal of approximately preserving similarity (SISC) is a novel contribution to the discussion.

Enforcing SISC in the context of an MSDC CF realizes a balance between:

a) maximizing the storage capacity of the CF, and
b) embedding the similarity structure of the input space in the set of stored codes, which in turn enables fixed-time best-match retrieval.

In exploring the implications of shifting focus from information theory to coding theory viz. theoretical neuroscience, [51] pointed to this same tradeoff, though their treatment uses error rate (coding accuracy) instead of storage capacity. Understanding how neural correlation ultimately affects things like storage capacity is considered largely unknown and an active area of research [52]. Our approach implies a straightforward answer. Minimizing correlation, i.e., maximizing average Hamming distance over the set of codes stored in an MSDC CF, maximizes storage capacity. Increases of any correlations of pairs, triples, or subsets of any order, of the CF's units increases the strength of embedding of statistical (similarity) relations in the input space.

Sparsey has many more features and capabilities than can be described here, e.g., it has been generalized to the temporal domain and to hierarchies of CFs. Nevertheless, the results shown here will hopefully pique further interest.

References

1. Pouget, A., et al., *Probabilistic brains: knowns and unknowns.* Nat Neurosci, 2013. **16**(9)
2. Pitkow, X. and D.E. angelaki, *How the brain might work: Statistics flow in redundant population codes.* (submitted), 2016.
3. Ma, W.J. & M. Jazayeri, *Neural Coding of Uncertainty and Probability.* Ann. Rev. Neuroscience, 2014. **37**(1): p. 205-220.
4. Barth, A.L. and J.F.A. Poulet, *Experimental evidence for sparse firing in the neocortex.* Trends in Neurosciences, 2012. **35**(6): p. 345-355.
5. Georgopoulos, A., et al., *On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex.* The J. of Neuroscience, 1982. **2**(11).
6. Pouget, A., P. Dayan, and R. Zemel, *Info. Proc. with pop. codes.* Nat Rev Neuro, 2000. **1**(2)
7. Pouget, A., P. Dayan, and R.S. Zemel, *Inference and Computation with Population Codes.* Annual Review of Neuroscience, 2003. **26**(1): p. 381-410.
8. Zemel, R., P. Dayan, and A. Pouget, *Probabilistic interpretation of population codes.* Neural Comput., 1998. **10**: p. 403-430.
9. Jazayeri, M. and J.A. Movshon, *Optimal representation of sensory information by neural populations.* Nat Neurosci, 2006. **9**(5): p. 690-696.
10. Ma, W.J., et al., *Bayesian inference with probabilistic pop. codes.* Nat Neuro, 2006. **9**(11)
11. Boerlin, M. & S. Denève, *Spike-Based Pop. Coding and Work. Mem.* PLOS CB, 2011. **7**(2)
12. Sanger, T.D., *Neural population codes.* Current Opin. in Neurobio., 2003. **13**(2)
13. Barlow, H., *Single units and sensation: a neuron doctrine for perceptual psychology?* . Perception, 1972. **1**(4): p. 371 -94.
14. Cox, D.D. and J.J. DiCarlo, *Does Learned Shape Selectivity in Inferior Temporal Cortex Automatically Generalize Across Retinal Position?* J. Neurosci., 2008. **28**(40)
15. Nandy, Anirvan, et al., *The Fine Structure of Shape Tuning in Area V4.* Neuron, 2013. **78**(6)
16. Mante, V., et al., *Context-dependent computation by recurrent dynamics in prefrontal cortex.* Nature, 2013. **503**(7474): p. 78-84.
17. Nandy, Anirvan S., et al., *Neurons in Macaque Area V4 Are Tuned for Complex Spatio-Temporal Patterns.* Neuron, 2016. **91**(4): p. 920-930.
18. Bonin, V., et al., *Local Diversity and Fine-Scale Organization of Receptive Fields in Mouse Visual Cortex.* The Journal of Neuroscience, 2011. **31**(50): p. 18506-18521.
19. Yen, S.-C., J. Baker, and C.M. Gray, *Heterogeneity in the Responses of Adjacent Neurons to Natural Stimuli in Cat Striate Cortex.* Journal of Neurophysiology, 2007. **97**(2)
20. Smith, S.L. and M. Häusser, *Parallel processing of visual space by neighboring neurons in mouse visual cortex.* Nature neuroscience, 2010. **13**(9): p. 1144-1149.
21. Herikstad, R., et al., *Natural Movies Evoke Spike Trains with Low Spike Time Variability in Cat Primary Visual Cortex.* The Journal of Neuroscience, 2011. **31**(44): p. 15844-15860.
22. Fusi, S., E.K. Miller, and M. Rigotti, *Why neurons mix: high dimensionality for higher cognition.* Current Opinion in Neurobiology, 2016. **37**: p. 66-74.
23. Hebb, D.O., *The organization of behavior; a neuropsychological theory*. 1949, NY: Wiley.
24. Yuste, R., *From the neuron doctrine to neural networks.* Nat Rev Neurosci, 2015. **16**(8)
25. Saxena, S. & J.P. Cunningham, *Towards neural pop. doctrine.* Curr Op Neurobio, 2019. **55**.
26. Deneve, S. and M. Chalk, *Efficiency turns the table on neural encoding, decoding and noise.* Current Opinion in Neurobiology, 2016. **37**: p. 141-148.

27. Rinkus, G., *A Combinatorial Neural Network Exhibiting Episodic and Semantic Memory Properties for Spatio-Temporal Patterns*, in *Cognitive & Neural Systems*. 1996, Boston U.
28. Rinkus, G., *A cortical sparse distributed coding model linking mini- and macrocolumn-scale functionality.* Frontiers in Neuroanatomy, 2010. **4**.
29. Rinkus, G.J., *Sparsey^TM: Spatiotemporal Event Recognition via Deep Hierarchical Sparse Distributed Codes.* Frontiers in Computational Neuroscience, 2014. **8**.
30. Buzsáki, G., *Neural Syntax: Cell Assemblies, Synapsembles, ...* Neuron, 2010. **68**(3)
31. Watrous, A.J., et al., *More than spikes: common oscillatory mechanisms for content specific neural representations during perception and memory.* Curr. Opin. in Neurobio., 2015. **31**
32. Igarashi, K.M., et al., *Coordination of entorhinal-hippocampal ensemble activity during associative learning.* Nature, 2014. **510**(7503): p. 143-147.
33. Fries, P., *Neuronal Gamma-Band Synchronization as a Fundamental Process in Cortical Computation.* Annual Review of Neuroscience, 2009. **32**(1): p. 209-224.
34. Hubel, D.H. and T.N. Wiesel, *Receptive fields, binocular interaction and functional architecture in the cat's visual cortex.* J Physiol, 1962. **160**(1): p. 106-154.
35. Rinkus, G., (sub'd CCN) *The Classical Tuning Function is an Artifact of a Neuron's Participations in Multiple Cell Assemblies*. 2023.
36. McCormick, D.A. and D.A. Prince, *Mechanisms of action of acetylcholine in the guinea-pig cerebral cortex in vitro.* J Physiol, 1986. **375**: p. 169-94.
37. Sara, S.J., A. Vankov, and A. Hervé, *Locus coeruleus-evoked responses in behaving rats: A clue to the role of noradrenaline in memory.* Brain Research Bulletin, 1994. **35**(5-6)
38. Rinkus, G., *Population Coding using Familiarity-Contingent Noise (poster)*, in *AREADNE 2008: Research in Encoding And Decoding of Neural Ensembles*. 2008: Santorini, GR.
39. Rinkus, G. *A cortical theory of super-efficient probabilistic inference based on sparse distributed representations*. in *CNS 2013*. 2013. Paris.
40. Willshaw, D.J. et al., *Non Holographic Associative Memory.* Nature, 1969. **222**: p. 960-962
41. Marr, D., *A theory of cerebellar cortex.* J Physiol, 1969. **202**(2): p. 437-470.
42. Rajkumar, V. & X. Pitkow, *Inference by Reparameterization in Neural Pop. Codes*. 2016.
43. Bengio, Y., *Deep Learning of Representations: Looking Forward*, in *Statistical Language and Speech Processing: First International Conference, SLSP 2013, Tarragona, Spain, July 29-31, 2013. Proceedings*, A.-H. Dediu, et al., Editors. 2013, Springer Berlin Heidelberg.
44. Salakhutdinov, R. and G. Hinton. *Semantic Hashing*. in *SIGIR workshop on Information Retrieval and applications of Graphical Models*. 2007.
45. Salakhutdinov, R. & G. Hinton, *Semantic hashing.* Int'l J. Approx. Reasoning, 2009. **50**(7)
46. Grauman, K. and R. Fergus, *Learning Binary Hash Codes for Large-Scale Image Search*, in *Machine Learning for Computer Vision*, R. Cipolla, S. Battiato, and G.M. Farinella, Editors. 2013, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 49-87.
47. Wang, J., et al., *Learning to Hash for Indexing Big Data - A Survey.* Proc IEEE, 2016. **104**(1)
48. Kohn, A., et al., *Correlations and Neuronal Pop. Information.* Ann. Rev. Neuro., 2016. **39**
49. Cohen, M.R. & A. Kohn, *Measuring and interpreting neuronal corr.* Nat Neuro, 2011. **14**(7)
50. Schneidman, E., *Towards design princ. of neural pop. codes.* Curr Op Neurobio., 2016. **37**
51. Curto, C., et al., *Combinatorial Neural Codes from a Mathematical Coding Theory Perspective.* Neural Comp, 2013. **25**(7): p. 1891-1925.
52. Latham, P.E., *Correlations demystified.* Nat Neurosci, 2017. **20**(1): p. 6-8.