Overview of different HD Computing/VSA models*

Denis Kleyko

*Lots of images from Internet were used to prepare this presentation

A beloved child has many names

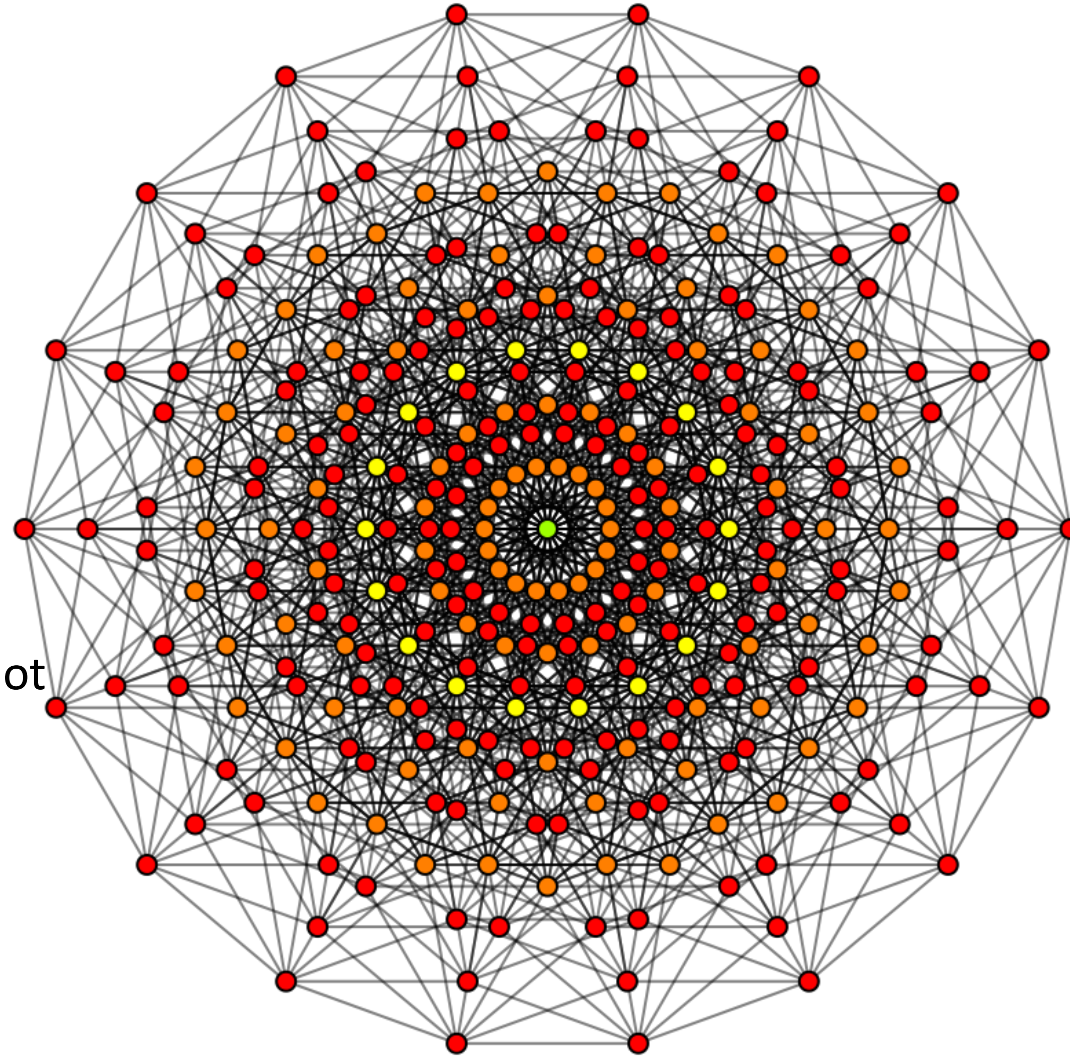
- Umbrella terms:
  - Vector Symbolic Architectures (VSAs) – R. Gayler (2003)
  - Hyperdimensional Computing (HDC/HD computing) – P. Kanerva (2009)

- Concrete models:
  - **Multiply Add Permute, MAP – R. Gayler**
  - Tensor Product Variable Binding, TPR – P. Smolensky
  - (Frequency) Holographic Reduced Representations, (F)HRR – T. Plate
    - Semantic Pointer Architecture, SPA – C. Eliasmith
  - Binary Spatter Codes, BSC – P. Kanerva
  - Matrix Binding of Additive Terms, MBAT – S. Gallant
  - Sparse Binary Distributed Representations, SBDR – E. Kussul, D. Rachkovskij, et al.
  - Sparse Block-Codes, SBC – M. Laiho, et al.
  - Geometric Analogue of Holographic Reduced Representations, GAHRR – D. Aerts, et al.

Recap: basic components of VSAs

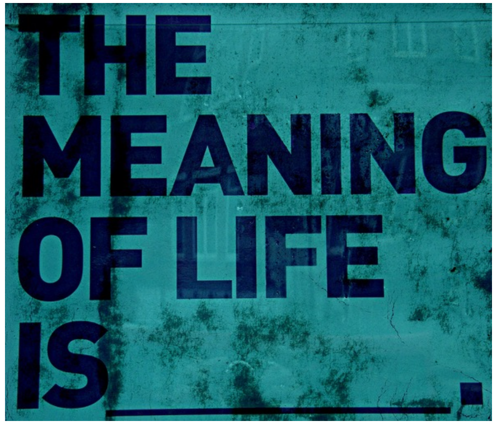
The basic ingredients of any VSA model are:

- Representational space, e.g., binary/bipolar
- High-dimensionality
  - e.g., 10^3 dimensions
  - HD vectors
- Randomness
- Similarity metric, e.g., dot (inner) product: sim_{dot}
- Item memory
- Operations on representations



Associative memory: Meaning

- Meaning is a fundamental component of nearly all aspects of human cognition



- A semantic memory is necessary for humans to construct meaning from otherwise meaningless words

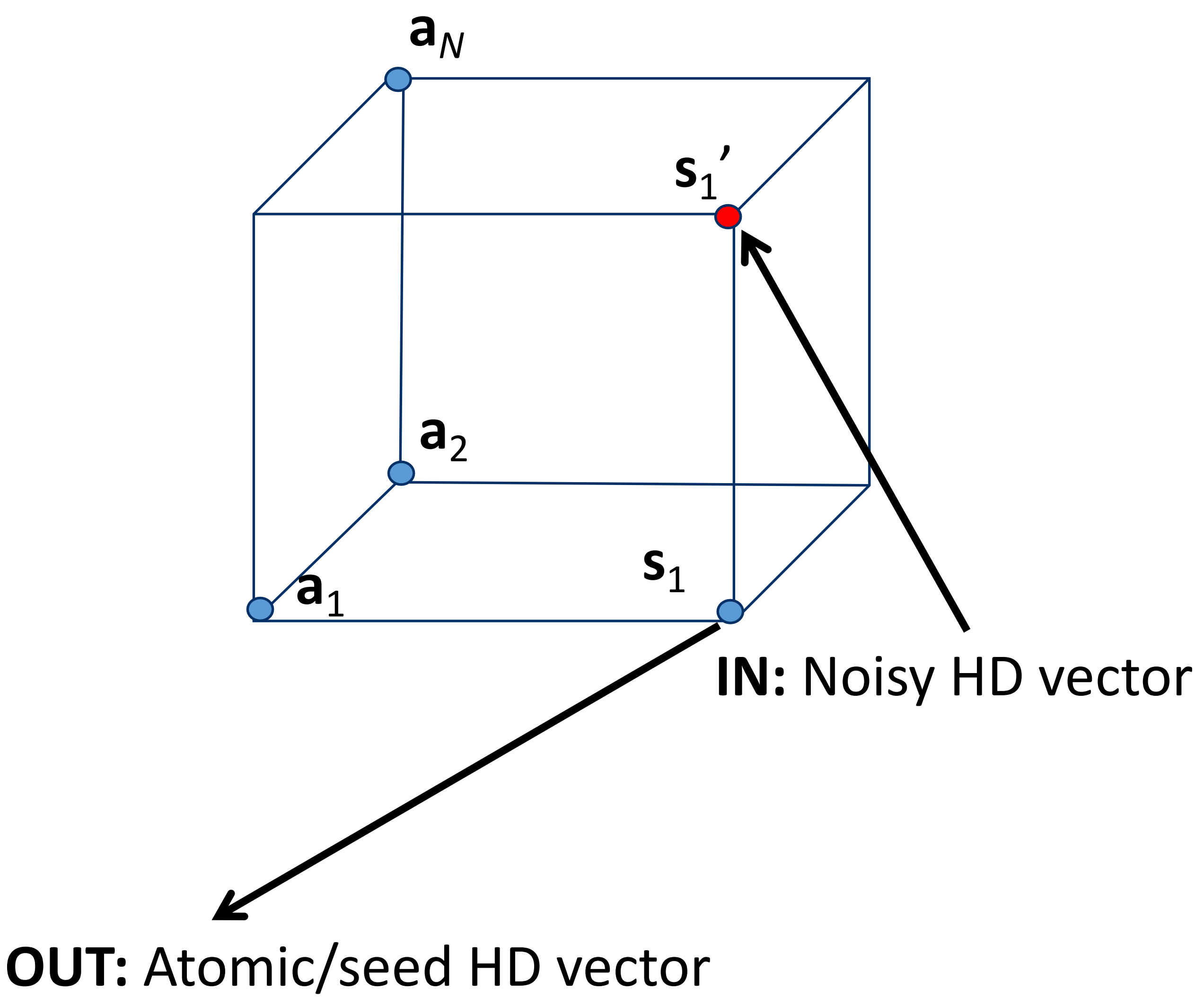
Associative memory: a simple experiment

- corkscrew
- korkskruv
- dugóhúzó
- штопор

Item memory

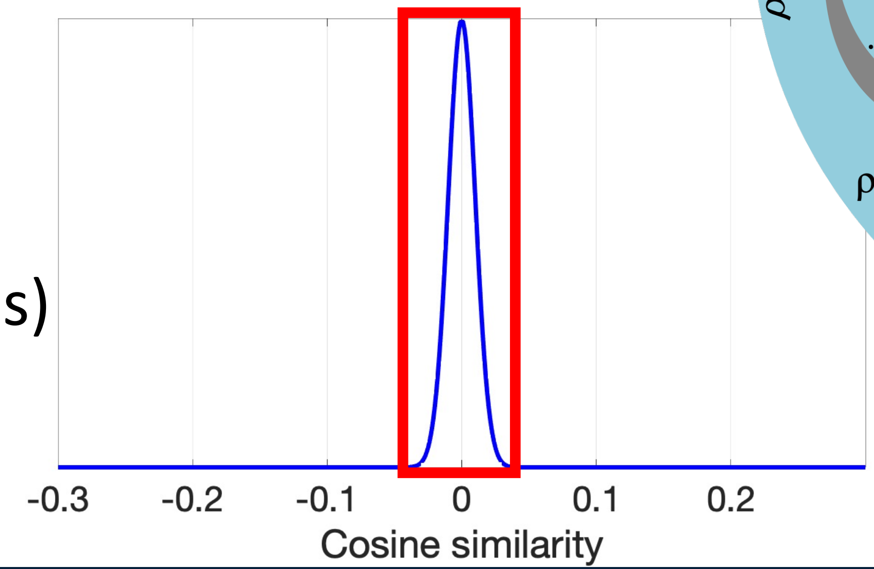
- A codebook of (random) HD vectors with assigned meanings
  - autoassociative
- Also called clean-up memory
- Uses the similarity metric (e.g., sim_{dot})
- Nearest-neighbor search among the set of stored meaningful HD vectors

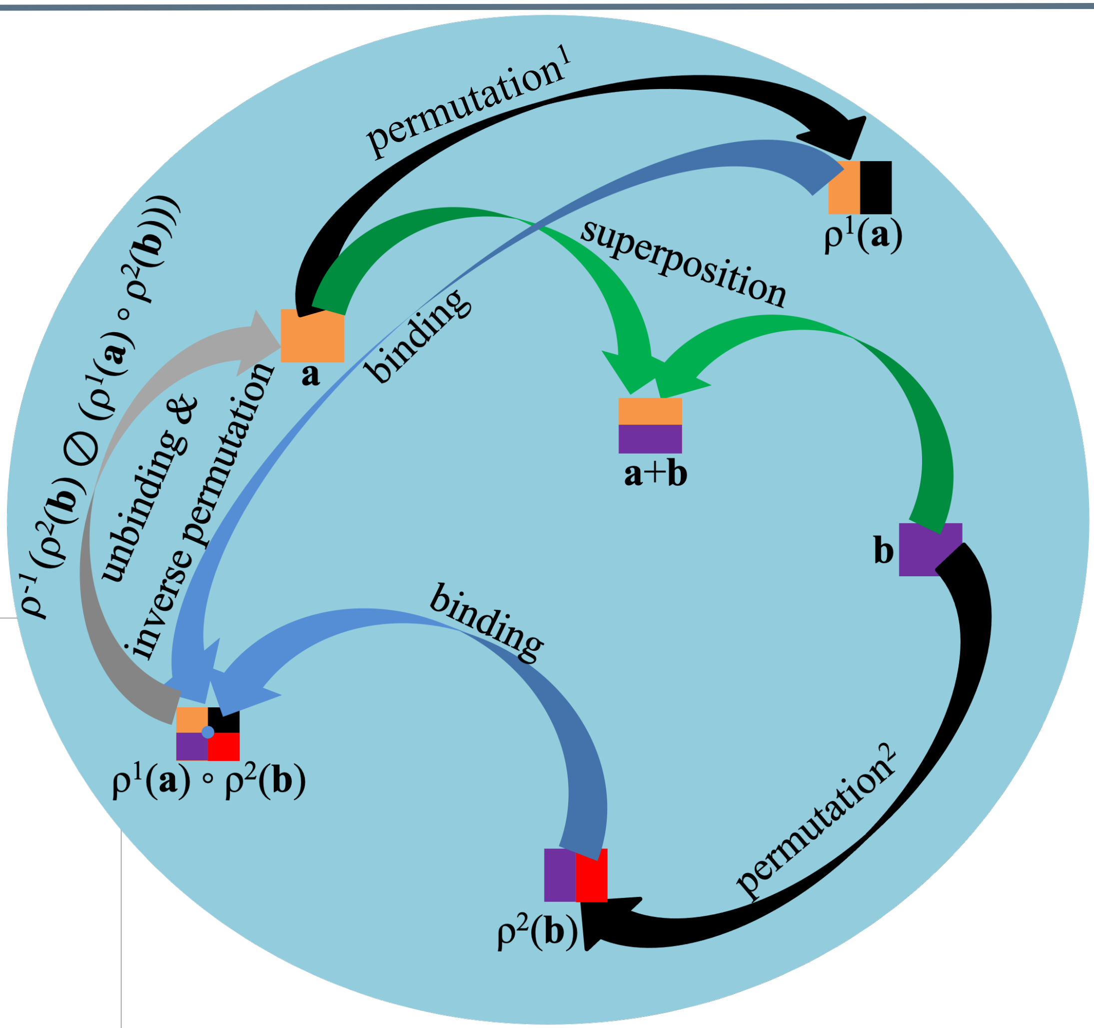
HD-vector	Meaning
-1111-1-11-11-11 ($\mathbf{a}_1$)	Object 1
1-11-11-11-1-111 ($\mathbf{a}_2$)	Object 2
…	
1-1111-111-111 ($\mathbf{a}_N$)	Object *D*-1
-11-11-11-1111-1 ($\mathbf{s}_1$)	Object *D*



HD Computing/VSA operations

- Bundling/superposition (sets)
  - Example: component-wise addition
  - $\mathbf{z}=\mathbf{a}+\mathbf{b}+\mathbf{c}$
  - $\text{sim}_{\text{dot}}(\mathbf{a}, \mathbf{z})>0$
- Binding (variable binding)
  - Example: component-wise multiplication
  - denoted as $\mathbf{a} \circ \mathbf{b}$
  - $\text{sim}_{\text{dot}}(\mathbf{a}, \mathbf{a} \circ \mathbf{b})\approx 0$
- Unbinding (release)
  - denoted as $\mathbf{b} \oslash (\mathbf{a} \circ \mathbf{b})$
  - $\text{sim}_{\text{dot}}(\mathbf{a}, \mathbf{b} \oslash (\mathbf{a} \circ \mathbf{b}))\approx n$
- Permutation (sequences)
  - Example: cyclic shift
  - denoted as $\rho^1(\mathbf{a})$
  - $\text{sim}_{\text{dot}}(\mathbf{a}, \rho^1(\mathbf{a}))\approx 0$





Operations: **superposition** properties

- Superposition can be inverted with subtraction:
  - $\mathbf{z}=\mathbf{a}+\mathbf{b}+\mathbf{c}$
  - $\mathbf{z}-\mathbf{c}=\mathbf{a}+\mathbf{b}$
- The result of superposition is similar to its arguments ($\mathbf{z}=\mathbf{a}+\mathbf{b}+\mathbf{c}$):
  - $\text{sim}_{\text{dot}}(\mathbf{a}, \mathbf{z}) \approx \text{sim}_{\text{dot}}(\mathbf{b}, \mathbf{z}) \approx \text{sim}_{\text{dot}}(\mathbf{c}, \mathbf{z}) > 0$
- Binding arguments can be recovered (approx.) from the superposition:
  - $\mathbf{b} \oslash (\mathbf{a} \circ \mathbf{b} + \mathbf{c} \circ \mathbf{d}) \approx \mathbf{a} + \mathbf{b} \oslash \mathbf{c} \circ \mathbf{d} = \mathbf{a} + \text{noise} \approx \mathbf{a}$
- Superposition is commutative
  - $\mathbf{a}+\mathbf{b} = \mathbf{b}+\mathbf{a}$
- Normalized superposition is approximately associative
  - $g()$ – normalization function
  - $g(g(\mathbf{a}+\mathbf{b})+\mathbf{c}) \approx g(\mathbf{a}+ g(\mathbf{b}+\mathbf{c}))$

Operations: **binding** properties

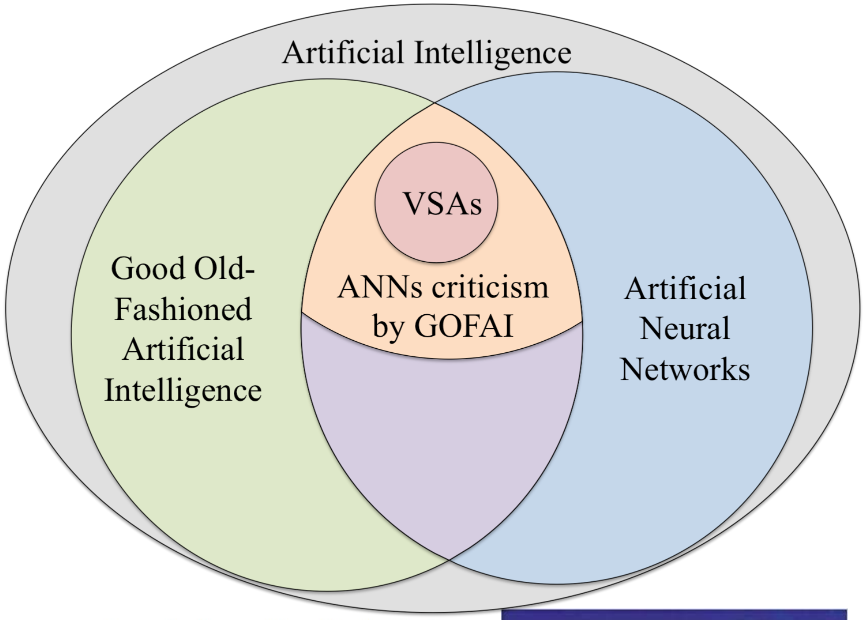
- Commutative: $\mathbf{a} \circ \mathbf{b} = \mathbf{b} \circ \mathbf{a}$
- Associative: $\mathbf{c} \circ (\mathbf{a} \circ \mathbf{b}) = (\mathbf{c} \circ \mathbf{a}) \circ \mathbf{b}$
- Distributes over superpostion: $\mathbf{c} \circ (\mathbf{a}+\mathbf{b}) = (\mathbf{c} \circ \mathbf{a}) + (\mathbf{c} \circ \mathbf{b})$
- Invertible: $\mathbf{b} \oslash (\mathbf{a} \circ \mathbf{b}) = \mathbf{a}$
- The result of binding is dissimilar to its arguments
  - $\text{sim}_{\text{dot}}(\mathbf{a}, \mathbf{a} \circ \mathbf{b}) \approx \text{sim}_{\text{dot}}(\mathbf{b}, \mathbf{a} \circ \mathbf{b}) \approx 0$
- Preserves similarity (for similar **a** & **a'**): $\text{sim}_{\text{dot}}(\mathbf{a}' \circ \mathbf{b}, \mathbf{a} \circ \mathbf{b}) > 0$
- “Randomizing” (since $\text{sim}_{\text{dot}}(\mathbf{a}, \mathbf{a} \circ \mathbf{b}) \approx 0$) but preserves similarity:
  - $\text{sim}_{\text{dot}}(\mathbf{a} \circ \mathbf{b}, \mathbf{c} \circ \mathbf{b}) = \text{sim}_{\text{dot}}(\mathbf{a}, \mathbf{c})$

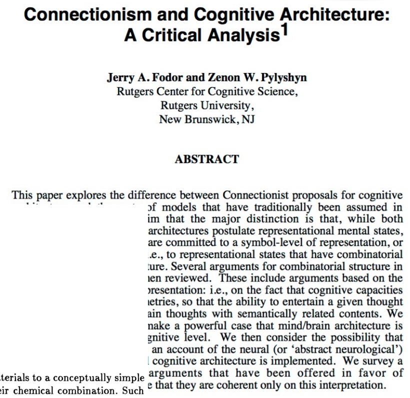
Operations: **permutation** properties

- Invertible for $\mathbf{r} = \rho^{1}(\mathbf{a})$: $\mathbf{a} = \rho^{-1}(\mathbf{r})$
- Distributes over binding: $\rho(\mathbf{a} \circ \mathbf{b}) = \rho(\mathbf{a}) \circ \rho(\mathbf{b})$
- Distributes over superpostion: $\rho(\mathbf{a} + \mathbf{b}) = \rho(\mathbf{a}) + \rho(\mathbf{b})$
- The result of permutation is dissimilar to its argument
  - $sim_{dot}(\mathbf{a}, \rho(\mathbf{a})) \approx 0$
- Preserves similarity (for similar **a** & **a'**): $sim_{dot}(\rho(\mathbf{a'}), \rho(\mathbf{a}))>0$
- “Randomizing” (since $sim_{dot}(\mathbf{a}, \rho(\mathbf{a})) \approx 0$) but preserves similarity:
  - $sim_{dot}(\rho(\mathbf{a}), \rho(\mathbf{b})) = sim_{dot}(\mathbf{a}, \mathbf{b})$

Historical excursus

- Several challenges for connectionist representations
- Superposition catastrophe:
  - red square or blue circle –> no problem
  - red square & blue circle –> issue
- Critics of the connectionism by Fodor & Pylyshyn
  - Composition, decomposition, and manipulation:
    - How are components composed to form a structure
    - How are components extracted from a structure?
    - Can the structures be manipulated using connectionist techniques?
- Jackendoff 's challenges:
  - The problem of two
    - How multiple instances of the same token are instantiated?
    - How “little star” and “big star” are instantiated?
    - Both are stars, yet distinguishable.



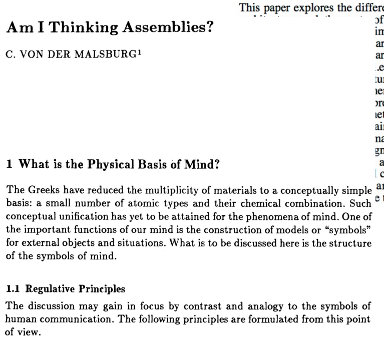


Connectionism and Cognitive Architecture: A Critical Analysis[1]

Jerry A. Fodor and Zenon W. Pylyshyn
Rutgers Center for Cognitive Science,
Rutgers University,
New Brunswick, NJ

ABSTRACT

This paper explores the difference between Connectionist proposals for cognitive … of models that have traditionally been assumed in … im that the major distinction is that, while both … architectures postulate representational mental states, … are committed to a symbol-level of representation, or … .e., to representational states that have combinatorial … ure. Several arguments for combinatorial structure in … en reviewed. These include arguments based on the … resentation: i.e., on the fact that cognitive capacities … etries, so that the ability to entertain a given thought … ain thoughts with semantically related contents. We … nake a powerful case that mind/brain architecture is … gnitive level. We then consider the possibility that … an account of the neural (or ‘abstract neurological’) … cognitive architecture is implemented. We survey a … arguments that have been offered in favor of … e that they are coherent only on this interpretation.



Am I Thinking Assemblies?

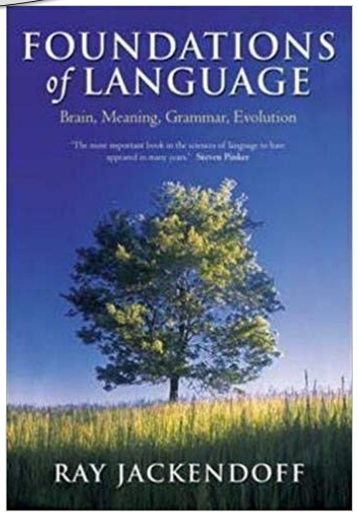
C. VON DER MALSBURG[1]

1 What is the Physical Basis of Mind?

The Greeks have reduced the multiplicity of materials to a conceptually simple basis: a small number of atomic types and their chemical combination. Such conceptual unification has yet to be attained for the phenomena of mind. One of the important functions of our mind is the construction of models or “symbols” for external objects and situations. What is to be discussed here is the structure of the symbols of mind.

1.1 Regulative Principles

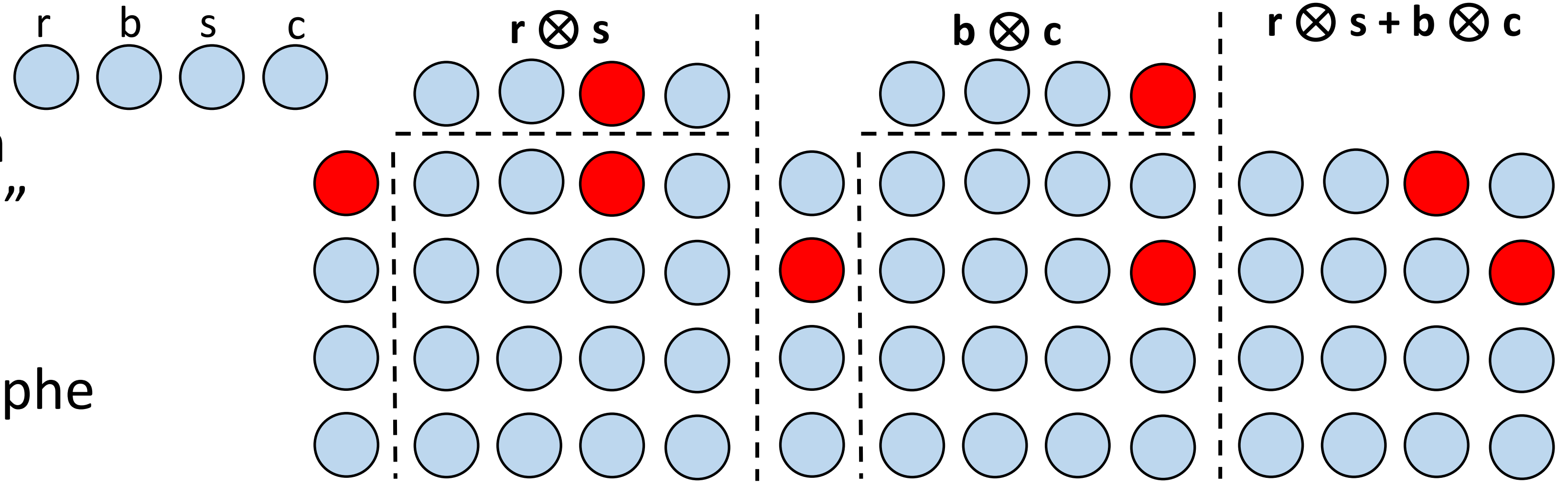
The discussion may gain in focus by contrast and analogy to the symbols of human communication. The following principles are formulated from this point of view.

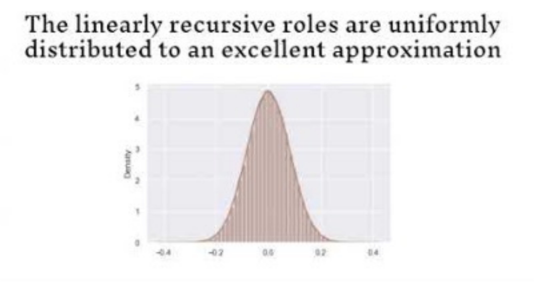


C. von der Malsburg, “Am I thinking assemblies?”, Brain Theory, 1986.
J. A. Fodor and Z. W. Pylyshyn, “Connectionism and Cognitive Architecture: A Critical Analysis,” Cognition, 1988.
R. Jackendoff, “Foundations of Language: Brain, Meaning, Grammar, Evolution”, Oxford University Press, 2002.

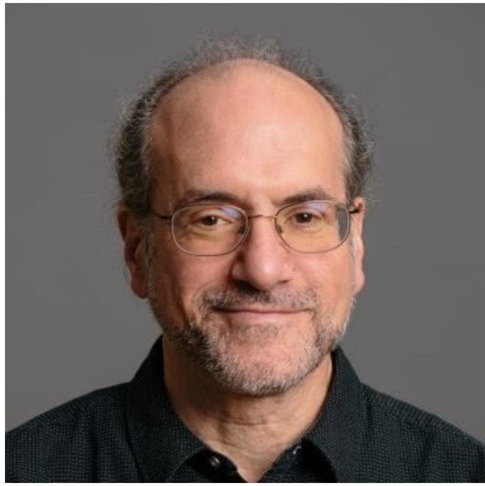
Tensor Product Representations

- Due to P. Smolensky
  - "... work reported here began as a response to this attack ..."
- Binding: tensor product, ⊗
  - Solves superposition catastrophe
  - Unbinding: inner product
- Dimensionality of bound HVs
  - Grows exponentially
  - 2 -> n^2; 3 -> n^3 ; etc.
  - Recursive application of the binding is challenging
  - Binding between different levels is ill-defined
- Underlying "hardware" is changing constantly





Learning and compressing Tensor Product Representations for Large-scale AI problems



Paul Smolensky

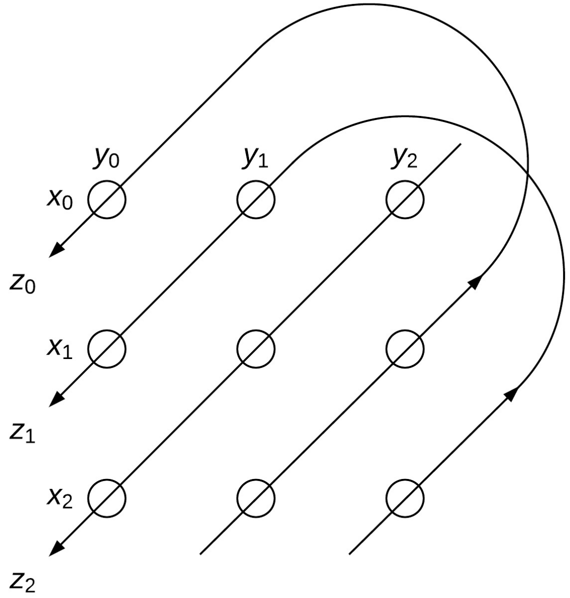
P. Smolensky, "Tensor Product Variable Binding and the Representation of Symbolic Structures in Connectionist Systems," Artificial Intelligence, 1990.
E. Mizraji, "Context-Dependent Associations in Linear Distributed Memories," Bulletin of Mathematical Biology, 1989.

Holographic Reduced Representations

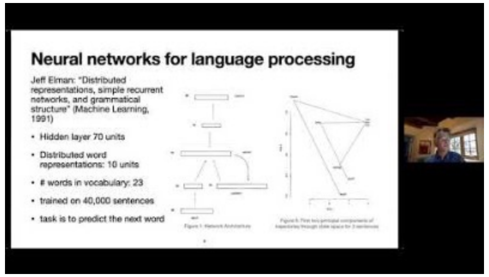
- Due to T. Plate
- Seed HD vectors: ~ $N(0,1/n)$
  - unit L2 norm
- Binding: circular convolution, $\circledast$
  - Compression of the tensor product
  - Unbinding: circular correlation

$$\mathbf{z}_0 = \mathbf{x}_0\mathbf{y}_0 + \mathbf{x}_2\mathbf{y}_1 + \mathbf{x}_1\mathbf{y}_2;$$
$$\mathbf{z}_1 = \mathbf{x}_1\mathbf{y}_0 + \mathbf{x}_0\mathbf{y}_1 + \mathbf{x}_2\mathbf{y}_2;$$
$$\mathbf{z}_2 = \mathbf{x}_2\mathbf{y}_0 + \mathbf{x}_1\mathbf{y}_1 + \mathbf{x}_0\mathbf{y}_2.$$

$$\mathbf{z}_j = \sum_{k=0}^{n-1} \mathbf{y}_k \mathbf{x}_{j-k} \mod n$$



- Similarity: dot product
- Dimensionality is fixed to *n*
- Superposition:
  - Component-wise addition
  - Normalizes HD vectors to preserve norm

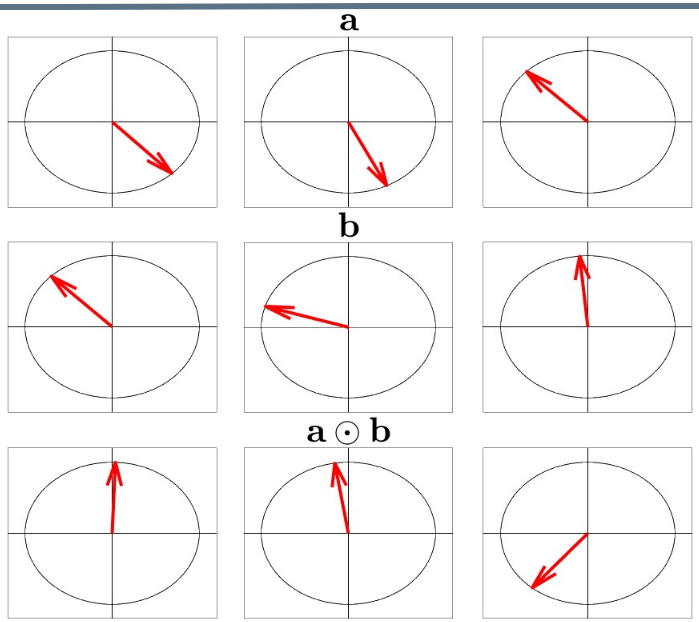


Vector Representations + Addition + Multiplication = Conceptual Reasoning

Tony Plate

G. E. Hinton, "Mapping Part-whole Hierarchies into Connectionist Networks" Artificial Intelligence, 1990.
T. A. Plate, "Holographic Reduced Representations: Convolution Algebra for Compositional Distributed Representations," International Joint Conference on Artificial Intelligence (IJCAI), 1991.

Fourier Holographic Reduced Representations



- Due to T. Plate
- Comes from the observation that:

$$\mathbf{x} \circledast \mathbf{y} = \mathbf{f}'(\mathbf{f}(\mathbf{x}) \odot \mathbf{f}(\mathbf{y}))$$

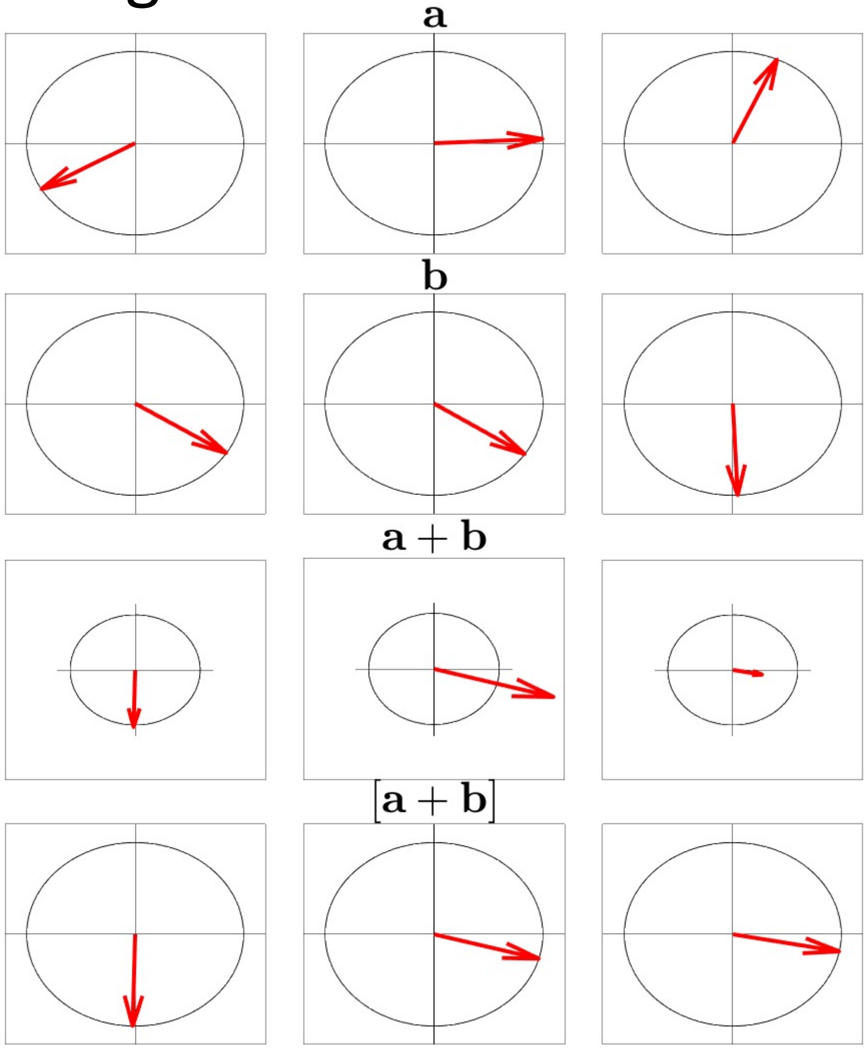
- Seed HD vectors: ~ phasors $e^{i\varphi}$ $\varphi \sim U(-\pi,+\pi)$
- Binding: component-wise multiplication, $\odot$
  - Avoids convolution and Fourier transforms
  - Unbinding: component-wise multiplication with conjugate
- Similarity: mean of sum of cosines of angle differences
  - $\mathrm{Re}(\mathbf{a}^{\mathrm{T}}\mathbf{b}^{*})/n$
- Superposition:
  - Component-wise addition
  - Normalizes each component to have unit magnitude

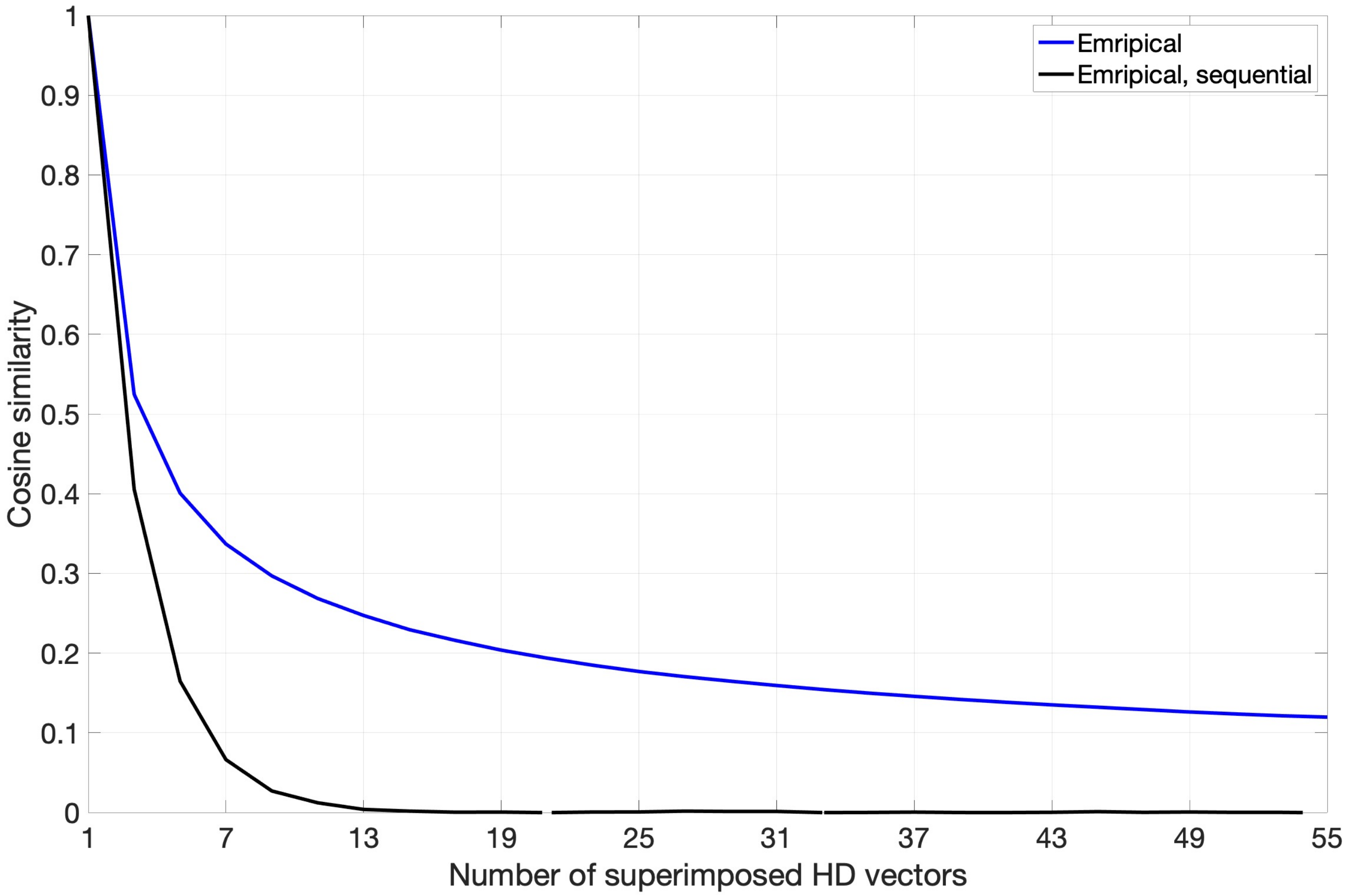
Tony Plate

T. A. Plate, "Distributed Representations and Nested Compositional Structure," University of Toronto, PhD Thesis, 1994.

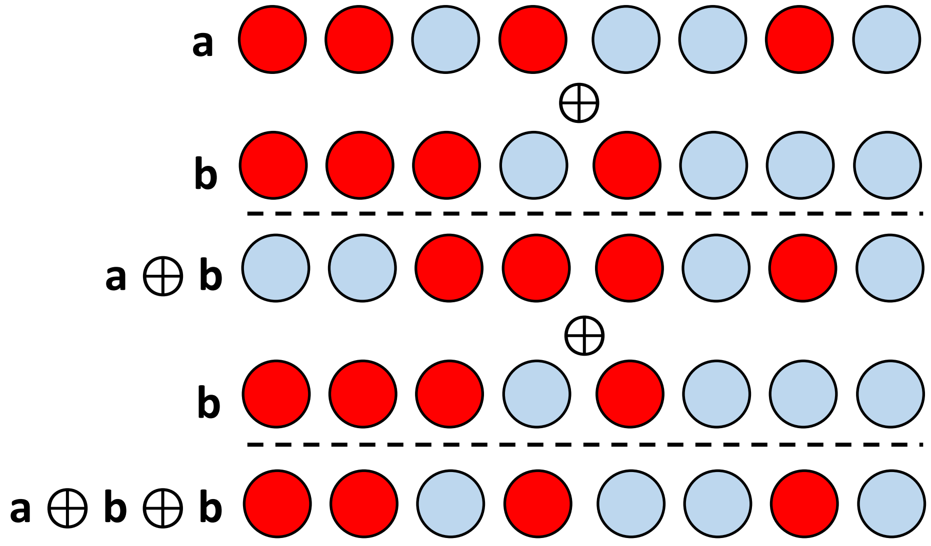
FHRR: Effect of normalization

- Unit magnitude normalization





Binary Spatter Codes



- Due to P. Kanerva
  - Comes from Sparse Distributed Memory
- Seed HD vectors: dense random binary vectors $\{0,1\}^n$
- Binding: component-wise XOR, $\oplus$
  - Result of the binding in also a binary vector
  - Unbinding: component-wise XOR, $\oplus$
- Can be thought as a special case of FHRR
  - phasors $e^{i\varphi}$ $\varphi \sim U(\{0,+\pi\})$
- Similarity: normalized Hamming distance, dist_{Ham}
- Superposition:
  - Component-wise addition
  - Normalize each component to binary value

Pentti Kanerva

P. Kanerva, "The Spatter Code for Encoding Concepts at Many Levels," International Conference on Artificial Neural Networks (ICANN), 1994.
P. Kanerva, "A Family of Binary Spatter Codes," International Conference on Artificial Neural Networks (ICANN), 1995.

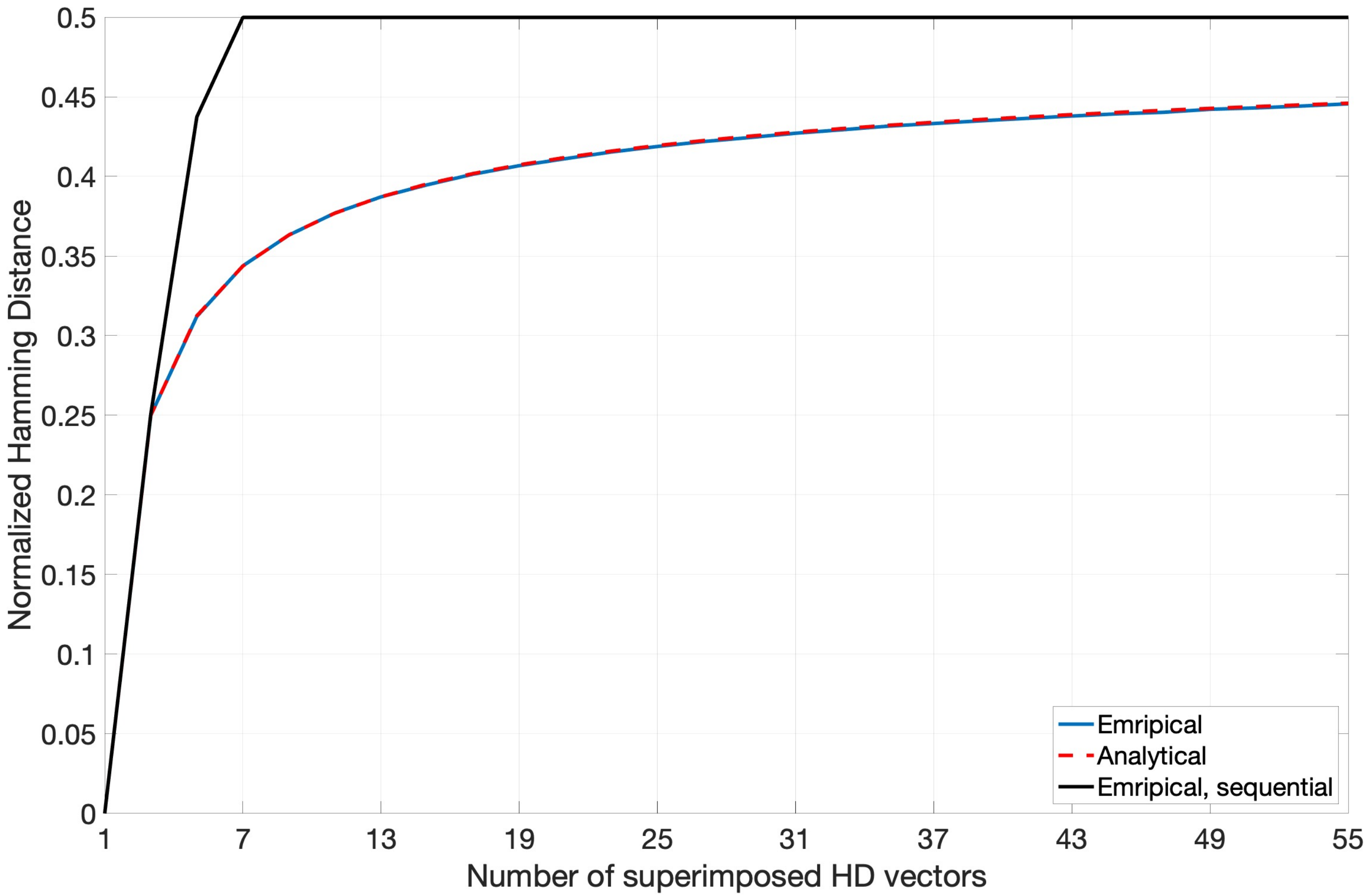
Binary Spatter Codes: majority rule

- Majority rules implements bundling/superposition:
  - $\mathbf{z}=[\mathbf{x}^{(1)}+\mathbf{x}^{(2)}+\ldots+\mathbf{x}^{(m)}]$

$$z_i = \begin{cases} 1, & \text{if } \sum_{j=1}^{m} \mathbf{x}_i^{(j)} > m/2 \\ 0, & \text{otherwise} \end{cases}$$

$\mathbf{x}^{(1)}$	1	0	0	1	1	0	1
$\mathbf{x}^{(2)}$	1	1	0	0	1	0	1
$\mathbf{x}^{(3)}$	0	0	0	1	1	1	0
$\Sigma\mathbf{x}^{(j)}$	**2**	**1**	**0**	**2**	**3**	**1**	**2**
$\mathbf{z}$	**1**	**0**	**0**	**1**	**1**	**0**	**1**

$$\text{dist}_{\text{Ham}} = \frac{1}{2} - \frac{1}{2^m}\binom{m-1}{\frac{m-1}{2}}$$



P. Kanerva, "Fully Distributed Representation," Real World Computing Symposium (RWC), 1997.

Multiply Add Permute

- Due to R. Gayler
- Seed HD vectors: bipolar {-1,+1} or real-valued *U*(-1,+1)
  - Several variants
    - Real-valued
    - Integer
    - Bipolar only
- Binding: component-wise multiplication, $\odot$
  - Unbinding: component-wise multiplication, $\odot$
- Similarity: dot product/cosine similarity
- Superposition:
  - Component-wise addition
  - Depending on the variant may be normalized

Ross Gayler

R. W. Gayler, "Multiplicative Binding, Representation Operators & Analogy," Advances in Analogy Research, 1998.

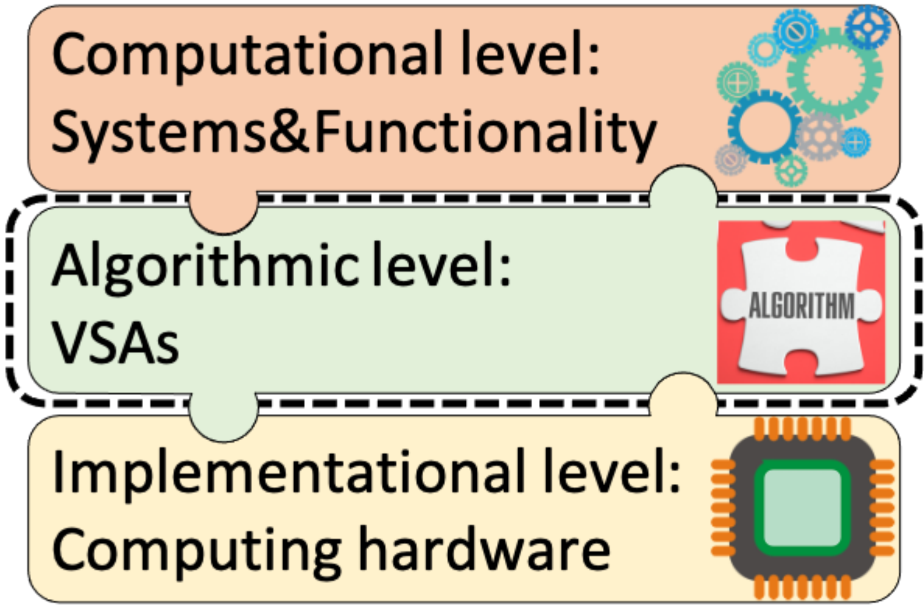
Historically: Why do we **have** all these (and more!) models?

- Several reasons (opinion)
  - Evolutionary development
  - Different initial assumptions
  - Variations in mathematical background of the originators

- Historically:
  - Tensor Product Representations -> Holographic Reduced Representations
    - Fixed dimensionality of representations to *n*
  - Holographic Reduced Representations -> Fourier Holographic Reduced Representations
    - Simplfied binding operation
  - (Fourier) Holographic Reduced Representations -> Binary Spatter codes
    - Binary representations
  - Binary Spatter codes -> Multiply Add Permute
    - Popularized permutation operation
    - Simple binding operation in real-valued domain

Currently: Why do we **need** all these models?

- Marr's three levels for information-processing devices:
  - Computational theory
  - Representation and algorithm
  - Hardware implementation



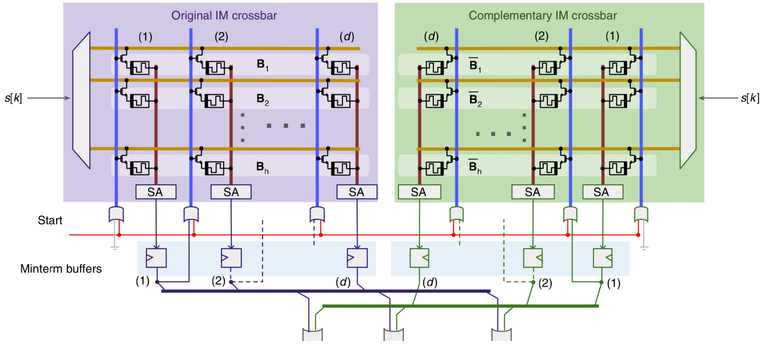
- Novel Computing Hardware:
  - Imprecise computational elements
    - Prone to errors but
    - Increases energy efficiency
  - A computing paradigm to abstract and simplify the functionality implementation
  - Lot of focus on implementing AI/ML capabilities

D. Marr, "Vision: A Computational Investigation into the Human Representation and Processing of Visual Information," W. H. Freeman and Company, 1982.
D. Kleyko, et al., "Vector Symbolic Architectures as a Computing Framework for Nanoscale Hardware," arXiv, 2021.

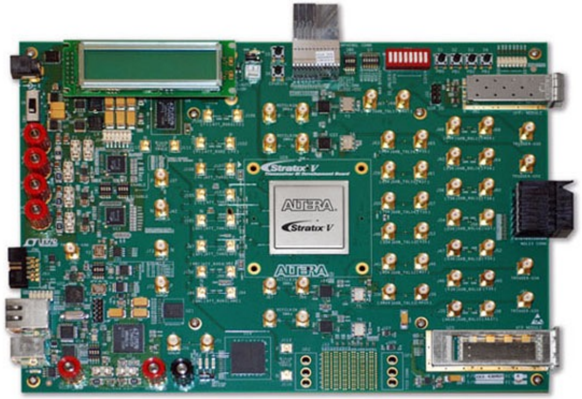
Model <-> Hardware examples

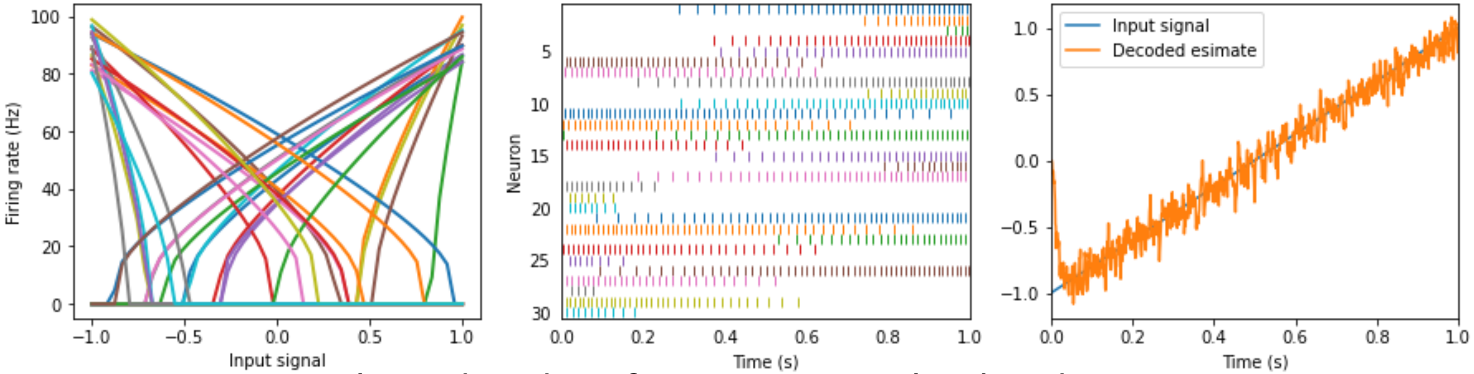
- Promising computing paradigm
  - intrinsic error resistance
  - high level of parallelization
  - simple operations

Memristive devices

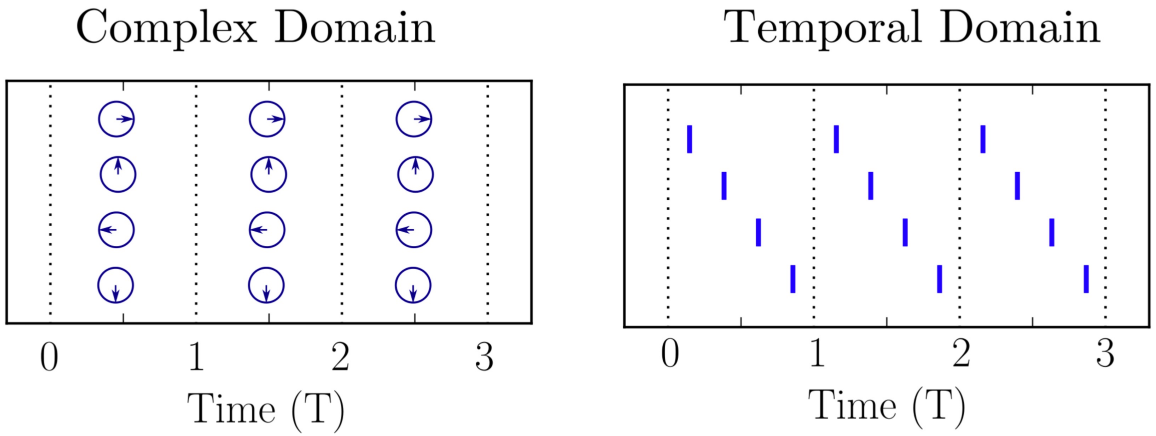


Field-programmable gate arrays





Rate-based coding for neuromorphic hardware



Phase-to-timing mapping

Matrix Binding of Additive Terms

- Due to S. Gallant
  - Developed largely independently

- Seed HD vectors ~ bipolar $\{-1,+1\}^n$ or or real-valued $U(-1,+1)$
- Binding: matrix-vector multiplication
  - Circular convolution can be represented as a matrix
  - Permutation can be represented as a matrix
  - Unbinding: multiplication with matrix inverse
  - Can change the dimensionality of HD vectors

- Similarity: Dot product
- Superposition:
  - Component-wise addition
  - Can be discretized to {-1,+1}


Stephen Gallant

S. I. Gallant and T. W. Okaywe, “Representing Objects, Relations, and Sequences,” Neural Computation, 2013.

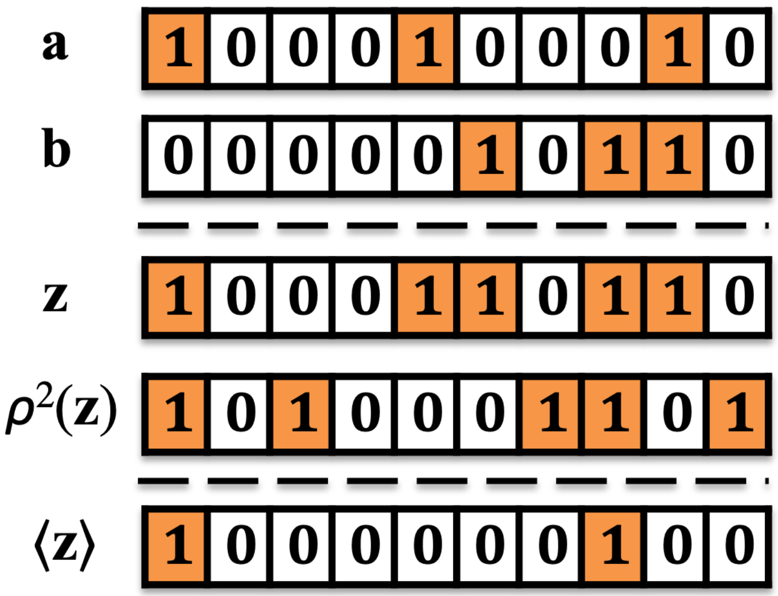
Sparse Binary Distributed Representations

- Due to E. Kussul and D. Rachkovskij
  - Developed independently of other models in 80s-90s
- Seed HD vectors sparse random binary vectors $\{0,1\}^n$
- Binding: context-dependent thinning

$$\mathbf{z} = \vee_{j=1}^{m} \mathbf{x}^{(j)}$$

$$\langle \mathbf{z} \rangle = \vee_{s=1}^{T} (\mathbf{z} \wedge \rho_s(\mathbf{z})) = \mathbf{z} \wedge (\vee_{s=1}^{T} \rho_s(\mathbf{z}))$$

- Similarity: Dot product
- Superposition:
  - Component-wise OR



D. A. Rachkovskij, E. M. Kussul, “Binding and Normalization of Binary Sparse Distributed Representations by Context-Dependent Thinning,” Neural Computation, 2001.

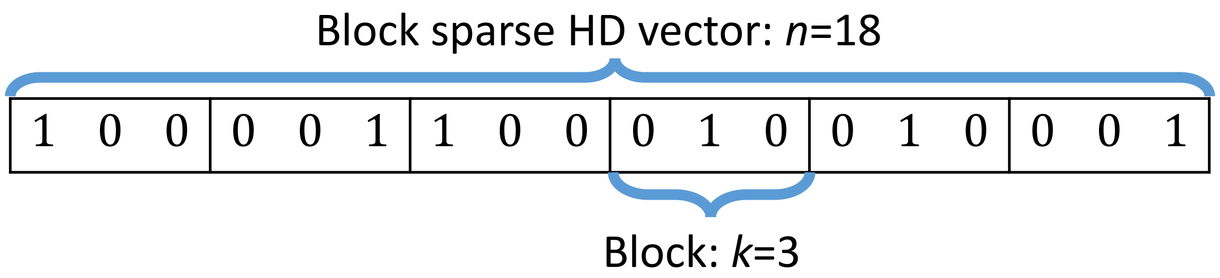
D. A. Rachkovskij, “Representation and Processing of Structures with Binary Sparse Distributed Codes,” IEEE Transactions on Knowledge and Data Engineering, 2001.


Ernst Kussul


Dmitri Rachkovskij

Sparse Block-Codes

- Due to M. Laiho, et al.
- Seed HD vectors sparse random binary vectors $\{0,1\}^n$
  - *n*-dimensional HD vector is treated as being constructed from blocks of size *k*
  - Only one component is active in each block
  - The total number of blocks is n/k
  - Density of HD vector is k/n



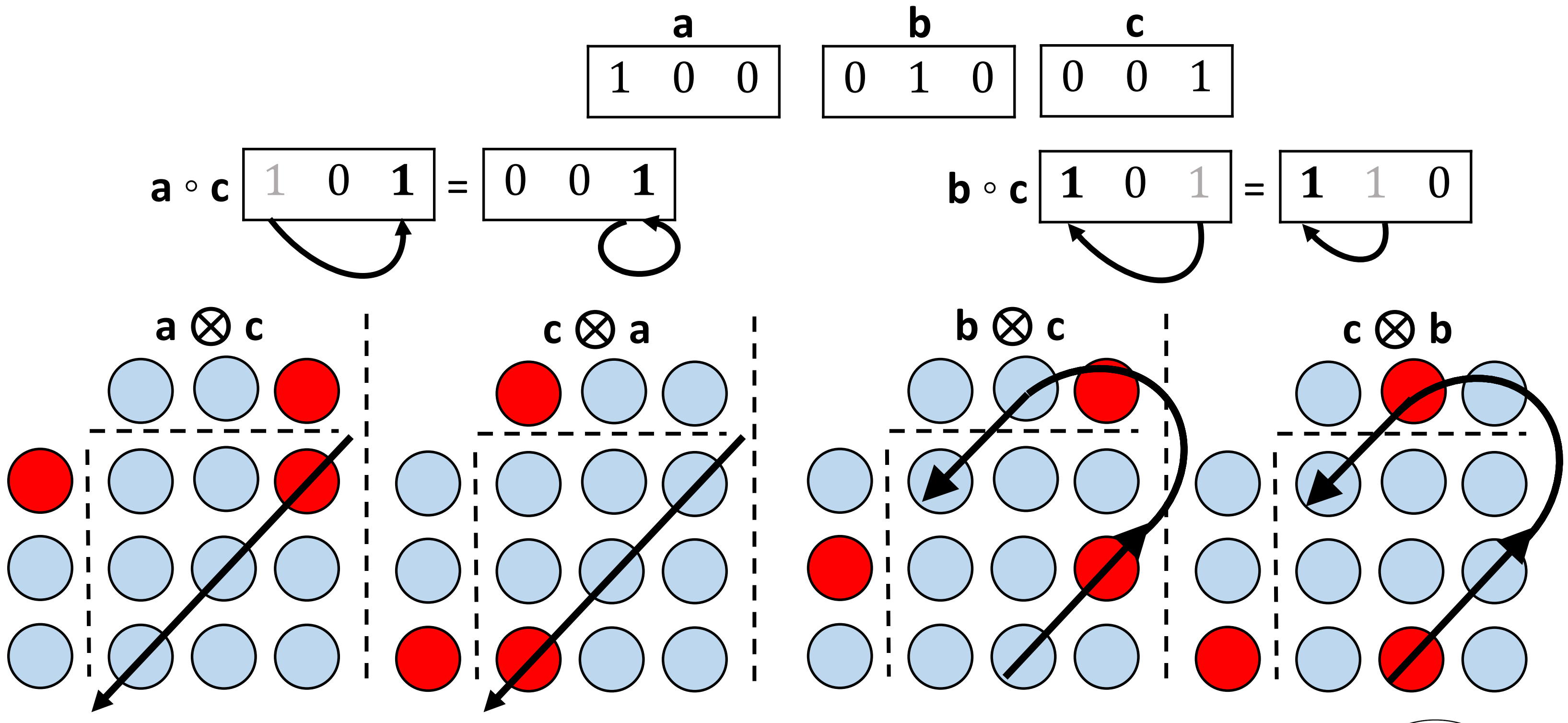
- Binding: defined for block sparse HD vectors
  - Cyclic shift within the blocks (but see next slide)
- Superposition: component-wise addition
  - Increases sparsity
  - WTA within the blocks

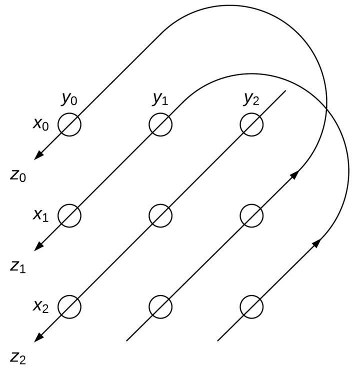
Mika Laiho

M. Laiho, et al., “High-Dimensional Computing with Sparse Vectors,” IEEE Biomedical Circuits and Systems Conference (BioCAS), 2015.

Sparse Block-Codes: binding

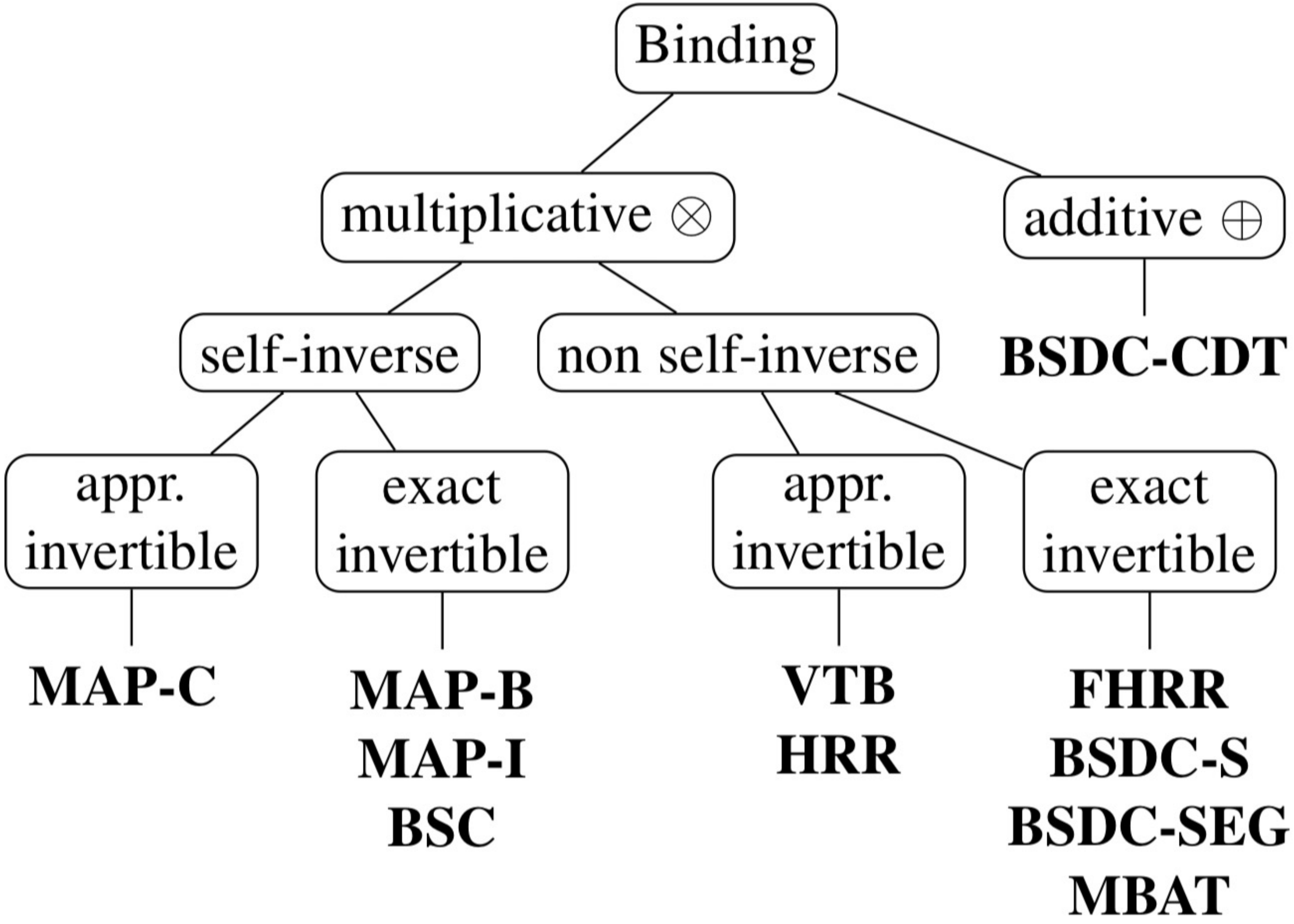
- Binding: defined for block sparse HD vectors
    - Cyclic shift within the blocks
    - Circular convolution on blocks





M. Laiho, et al., “High-Dimensional Computing with Sparse Vectors,” IEEE Biomedical Circuits and Systems Conference (BioCAS), 2015.
E. P. Frady, et al., “Variable Binding for Sparse Distributed Representations: Theory and Applications,” IEEE Transactions on Neural Networks and Learning Systems, 2021.

Taxonomy of binding operations



K. Schlegel, P. Neubert, P. Protzel, “A Comparison of Vector Symbolic Architectures,” arXiv, 2020.

Summary of models

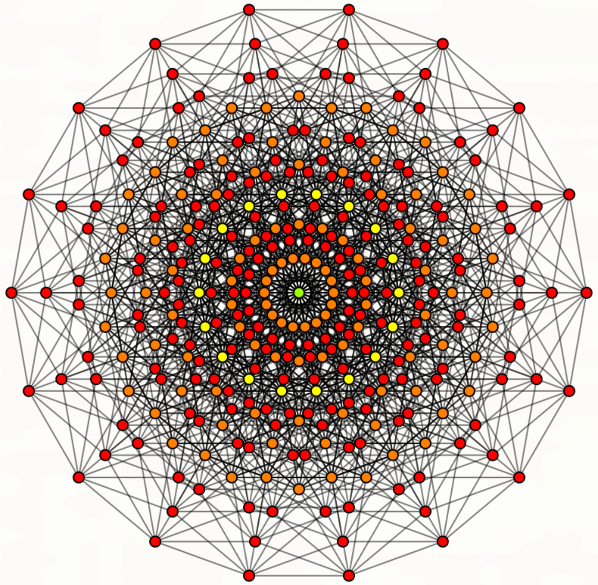
Name	elements X of vector space $\mathbb{V}$	Initialization of x_i	Sim. metric	Bundling	Binding		Unbinding	
					commu-tative	asso-ciative	commu-tative	asso-ciative
MAP-C	$X \in \mathbb{R}^D$	$x_i \sim \mathcal{U}(-1, 1)$	cosine sim.	elem. addition with cutting	elem. multipl. ✓	✓	elem. multipl. ✓	✓
MAP-I	$X \in \mathbb{Z}^D$	$x_i \sim \mathcal{B}(0.5) \cdot 2 - 1$	cosine sim.	elem. addition	elem. multipl. ✓	✓	elem. multipl. ✓	✓
HRR	$X \in \mathbb{R}^D$	$x_i \sim \mathcal{N}(0, \frac{1}{D})$	cosine sim.	elem. addition with normalization	circ. conv. ✓	✓	circ. corr. x	x
MBAT	$X \in \mathbb{R}^D$	$x_i \sim \mathcal{N}(0, \frac{1}{D})$	cosine sim.	elem. addition with normalization	matrix multipl. x	x	inv. matrix multipl. x	x
MAP-B	$X \in \{-1, 1\}^D$	$x_i \sim \mathcal{B}(0.5) \cdot 2 - 1$	cosine sim.	elem. addition with threshold	elem. multipl. ✓	✓	elem. multipl. ✓	✓
BSC	$X \in \{0, 1\}^D$	$x_i \sim \mathcal{B}(0.5)$	hamming dist.	elem. addition with threshold	XOR ✓	✓	XOR ✓	✓
BSDC-CDT	$X \in \{0, 1\}^D$	$x_i \sim \mathcal{B}(1/\sqrt{D})$	overlap	disjunction	CDT ✓	✓	-	
BSDC-SEG	$X \in \{0, 1\}^D$	$x_i \sim \mathcal{B}(1/\sqrt{D})$	overlap	disjunction (opt. thinning)	segment shifting ✓	✓	segment shifting x	x
FHRR	$X \in \mathbb{C}^D$	$x_i = e^{i \cdot \theta}$ $\theta \sim \mathcal{U}(-\pi, \pi)$	angle distance	angles of elem. addition	elem. angle addition ✓	✓	elem. angle subtraction x	x

K. Schlegel, P. Neubert, P. Protzel, “A Comparison of Vector Symbolic Architectures,” arXiv, 2020.

Rhetorical questions

- The more the better?
  - We have (hopefully) seen 8 HD Computing/VSA models
  - Is there a need to develop a new models?
  - Should this be driven by Marr's levels of analysis?

- Are the three operations sufficient?
  - Enough evidence for necessity
  - Would HD Computing/VSA benefit from new/other operations
  - Ideas on new operations can be a discussion point



Overview of different HD Computing/VSA models

Denis Kleyko