Compositional Factorization of Visual Scenes with Convolutional Sparse Coding and Resonator Networks

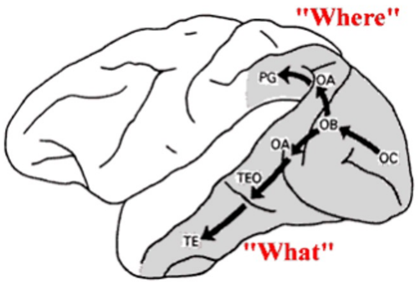
Christopher J. Kymn, Sonia Mazelet, Annabel Ng, Denis Kleyko, Bruno A. Olshausen

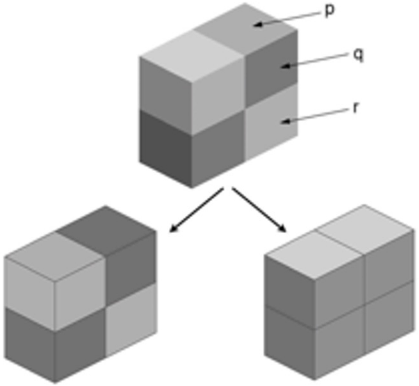
04.24.2024

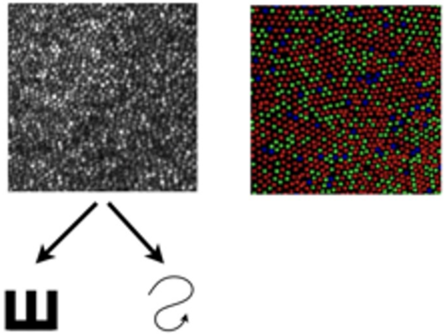
REDWOOD CENTER for Theoretical Neuroscience

Berkeley UNIVERSITY OF CALIFORNIA

Factorization is a problem for visual perception







What & Where

Ungerleider & Mishkin (1982)

Reflectance & Shading

Adelson (2000)

Shape & Motion

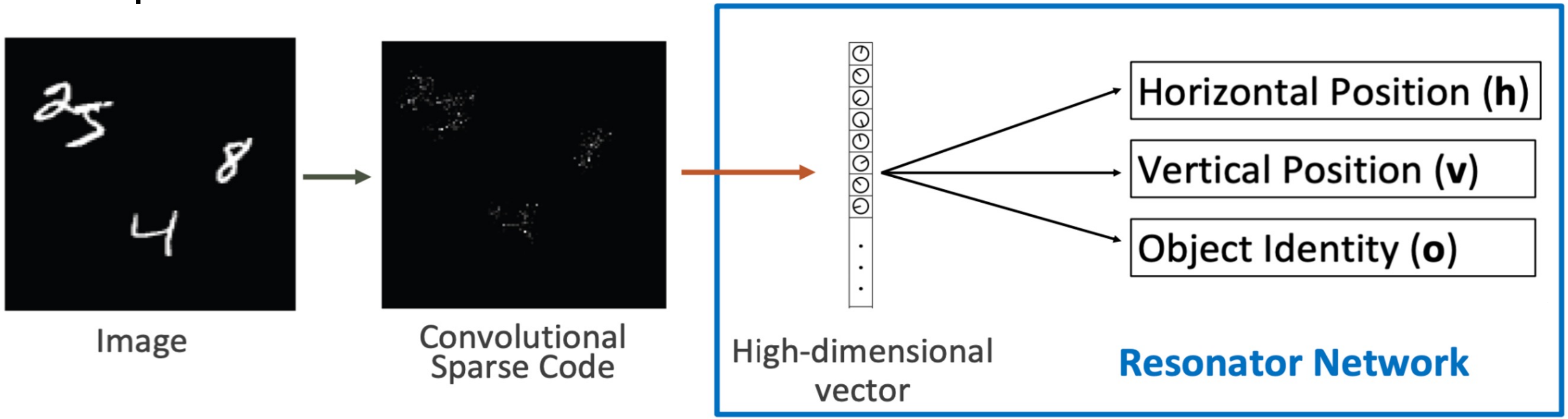
Anderson, Ratnam, Roorda, Olshausen (2020)

Problem statement

Given an image containing one or more objects, can we return each object's identity and position in the image, in a **neuromorphic and efficient way**?

Output

Input

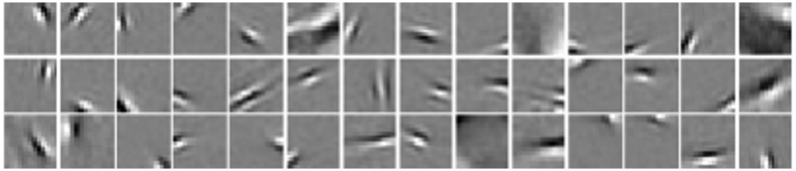


Sparse coding: a **compact** and **efficient** way of encoding images

Images are decomposed as a (small) linear superposition of basis functions, and remaining additive Gaussian noise

$$\mathbf{I} = \sum_j a_j \cdot \phi_j + \mathcal{N}$$

$$E = \underbrace{\frac{1}{2}||\mathbf{I} - \sum_j a_j \cdot \phi_j||_2^2}_{\text{Image Reconstruction}} + \underbrace{\sum_j |a_j|_1}_{\text{Sparsity}}$$



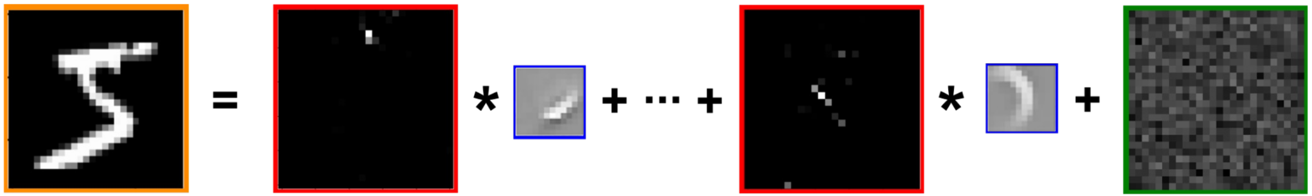
Basis functions learned on natural images

Olshausen, B. A., & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature, 381(6583), 607-609.

Convolutional sparse coding: an **equivariant** version of sparse coding

$$\mathbf{I} = \sum_j [\mathbf{A}_j * \phi_j] + \mathcal{N}$$

$$E = \frac{1}{2}||\mathbf{I} - \sum_j \mathbf{A_j} * \phi_j||_2^2 + \sum_j |\mathbf{A_j}|_1$$



Zeiler, M. D., Krishnan, D., Taylor, G. W., & Fergus, R. (2010, June). Deconvolutional networks. In *2010 IEEE Computer Society Conference on computer vision and pattern recognition* (pp. 2528-2535). IEEE.
Wohlberg, B. (2017). SPORCO: A Python package for standard and convolutional sparse representations. In *SciPy* (pp. 1-8).

Hyperdimensional computing (aka Vector Symbolic Architectures) provides a **compositional grammar** implemented via **distributed representations**

Primitive symbols:

- Assign random, high-dimensional vectors (**"Alice", "Bob", …**)

Rules for composing more complicated symbols:

- Adding, a.k.a. "Bundling" (+, "plus")
  - Alice and Bob = ("Alice" + "Bob")
- Association, a.k.a. "Binding" ($\odot$, "times")
  - Alice saw Bob =
    ("Alice"$\odot$"subject" + "saw"$\odot$"verb" + "Bob" $\odot$"object")

Rule for comparing vectors:

- Similarity metric for vectors (e.g., inner product)

Choice of HD/VSA vectors: Fourier Holographic Reduced Representations (FHRR)

Vectors are complex phasors:

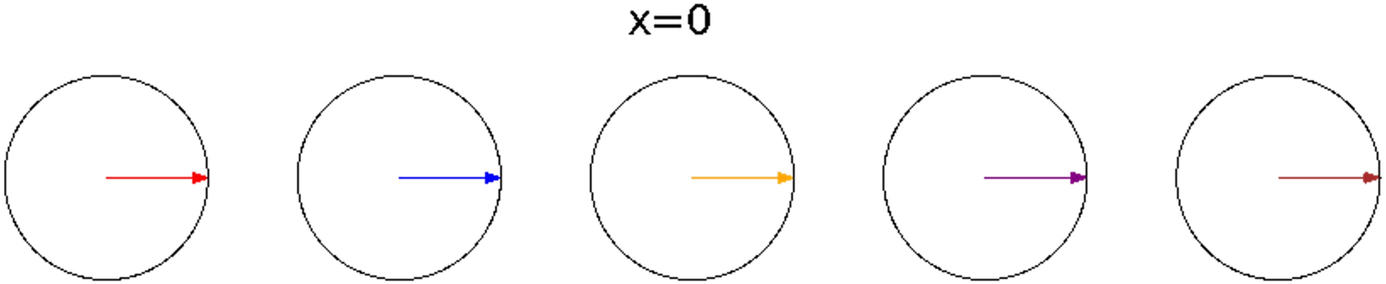
$$\mathbf{z} = \left[e^{i\phi_1}, \ldots, e^{i\phi_D}\right]$$

- Bundling (+) by element-wise addition
- Binding ($\odot$) by element-wise multiplication
- Similarity by normalized inner product

Encoding numbers via **power encoding**

Key idea : Represent any number x, by binding **z** *x* times with itself:

$$\mathbf{z}(x) = \underbrace{\mathbf{z} \odot \cdots \odot \mathbf{z}}_{\text{x times}} = \mathbf{z}^{x}$$



Plate, T. A. (1992). Holographic recurrent networks. Advances in neural information processing systems.
Plate, T. A. (1994). Distributed representations and nested compositional structure. University of Toronto, Department of Computer Science.

Encoding an image as a HD/VSA vector using convolutional sparse coding

- Convolutional sparse code features encoding:

convolutional sparse code feature

HD vector for basis function j

$$\mathbf{z}(\mathbf{A}) = \sum_{x,y,j} \mathbf{A}_j(x,y) \cdot \mathbf{h}(x) \odot \mathbf{v}(y) \odot \mathbf{b}(j)$$

HD vectors for position (x,y)

Encoding an image as a HD/VSA vector using convolutional sparse coding

convolutional sparse code feature

HD vector for basis function j

- Convolutional sparse code features encoding:

$$\mathbf{z}(\mathbf{A}) = \sum_{x,y,j} \mathbf{A}_j(x,y) \cdot \mathbf{h}(x) \odot \mathbf{v}(y) \odot \mathbf{b}(j)$$

HD vectors for position (x,y)

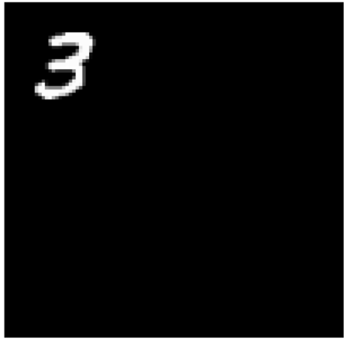
- Pixel encoding*:

$$\mathbf{z}_{\text{pix}}(\mathbf{I}) = \sum_{x,y} \mathbf{I}(x,y) \cdot \mathbf{h}(x) \odot \mathbf{v}(y)$$

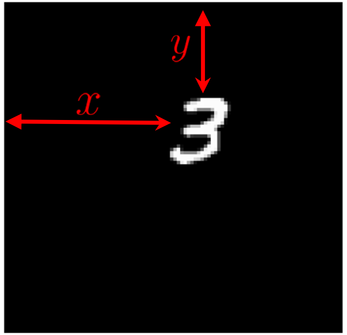
image pixel value

Renner, A., Supic, L., Danielescu, A., Indiveri, G., Olshausen, B. A., Sandamirskaya, Y., ... & Frady, E. P. (2022). Neuromorphic visual scene understanding with resonator networks. arXiv preprint arXiv:2208.12880.

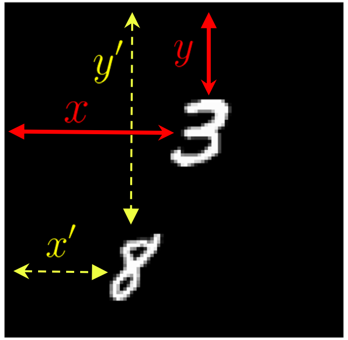
Recovering the objects/positions from the HD/VSA encoding is a factorization problem



$$\mathbf{z} = \mathbf{o}^{(3)}$$



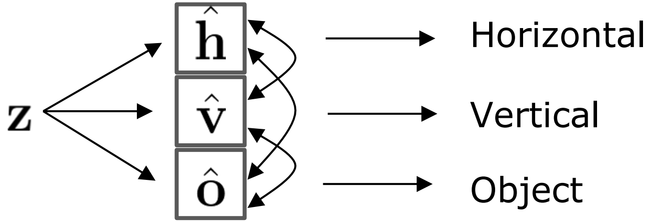
$$\mathbf{z} = \mathbf{h}(x) \odot \mathbf{v}(y) \odot \mathbf{o}^{(3)}$$



$$\mathbf{z} = \mathbf{h}(x) \odot \mathbf{v}(y) \odot \mathbf{o}^{(3)} + \mathbf{h}(x') \odot \mathbf{v}(y') \odot \mathbf{o}^{(8)}$$

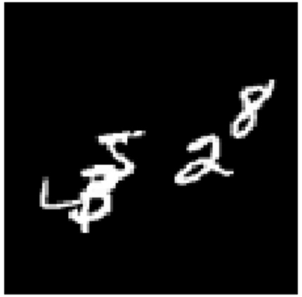
The **resonator network** is an algorithm for solving factorization

Algorithm



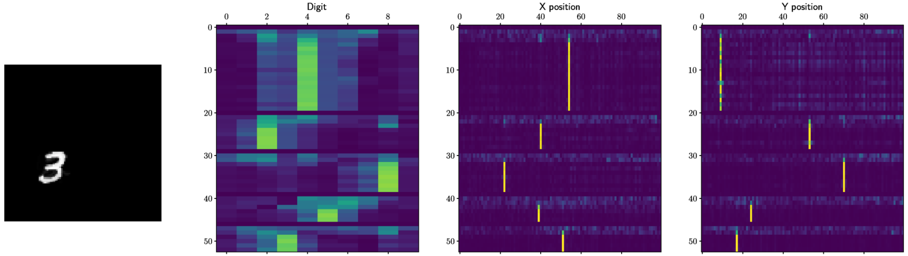
$$\hat{\mathbf{h}}_{t+1} = g(\mathbf{H}\mathbf{H}^{\dagger}(\mathbf{z} \odot \hat{\mathbf{v}}_t^{\dagger} \odot \hat{\mathbf{o}}_t^{\dagger}))$$
$$\hat{\mathbf{v}}_{t+1} = g(\mathbf{V}\mathbf{V}^{\dagger}(\mathbf{z} \odot \hat{\mathbf{h}}_t^{\dagger} \odot \hat{\mathbf{o}}_t^{\dagger}))$$
$$\hat{\mathbf{o}}_{t+1} = g(\mathbf{O}\mathbf{O}^{\dagger}(\mathbf{z} \odot \hat{\mathbf{h}}_t^{\dagger} \odot \hat{\mathbf{v}}_t^{\dagger}))$$

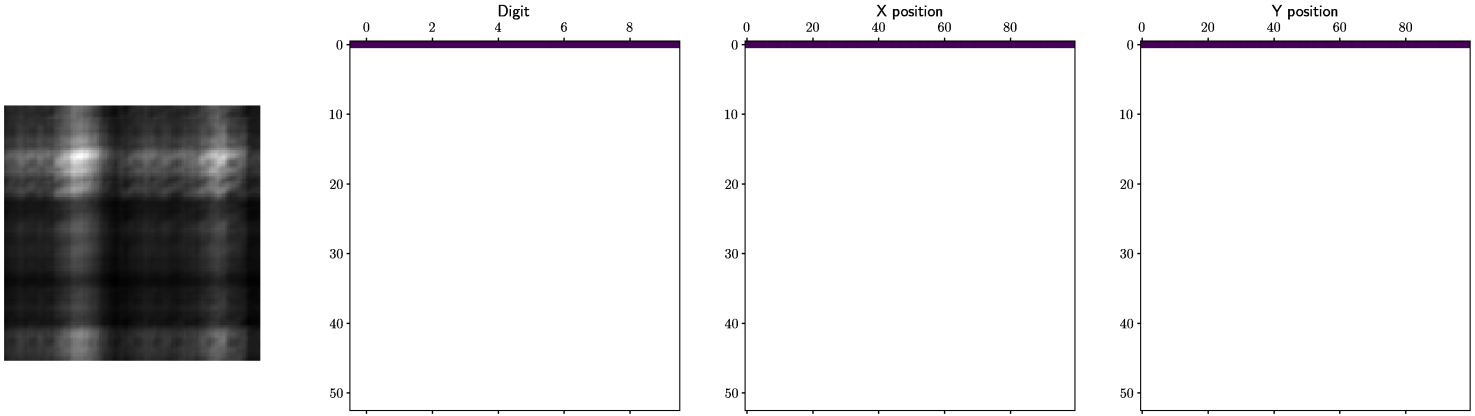
Input



$\longrightarrow \mathbf{z}$

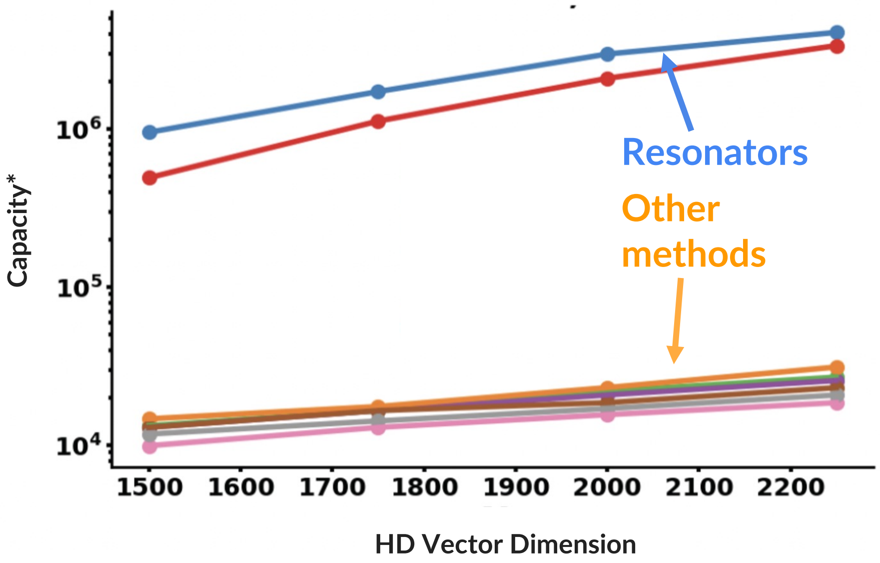
Simulation




Digit
X position
Y position
0
2
4
6
8
10
20
30
40
50
60
80

The **resonator network** is an algorithm for solving factorization

Performance

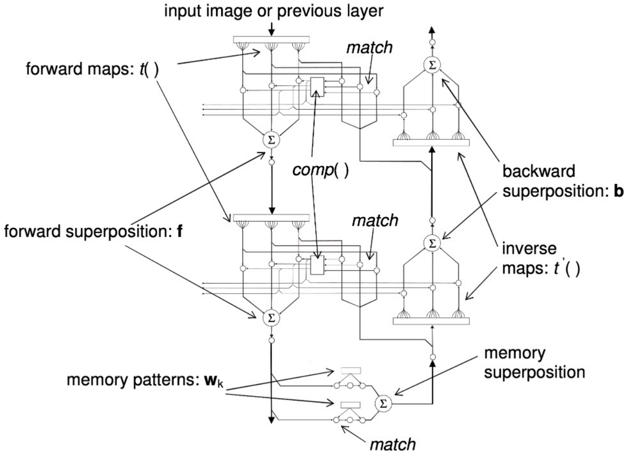


*Capacity = maximum search problem solved at 99% accuracy with fixed number of iterations.

Kent, S. J., Frady, E. P., Sommer, F. T., & Olshausen, B. A. (2020). Resonator networks, 2: Factorization performance and capacity compared to optimization-based methods. *Neural computation*, *32*(12), 2332-2388.

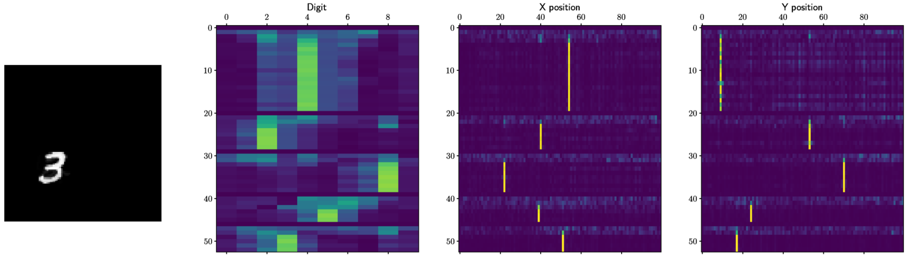
What makes it work: **searching in superposition**

Map Seeking Circuits:



- Search in superposition: check quality of weighted sums of guesses
- Principle: **cull** solutions until at most one remains

Resonator networks:

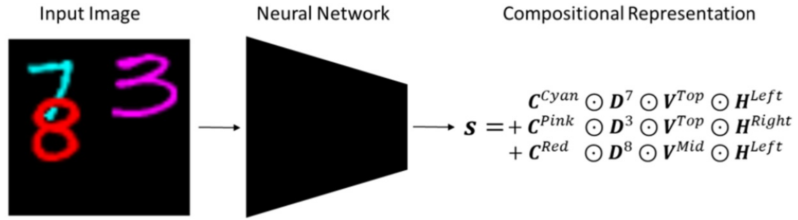


- Also search in superposition, but leverage the "blessing of [high] dimensionality"
- Principle of **self-consistency**: correct explanations are fixed points of the dynamics

Arathorn, D. W. (2002). *Map-seeking circuits in visual cognition: A computational mechanism for biological and machine vision*. Stanford University Press.

Related work on resonator networks for image factorization

- Direct encoding of the pixel values:
  - Renner et al., 2022
- Distributed representation computation using a neural network:
  - Frady et al., 2020
  - Hersche et al., 2022



Experimental setup

Datasets:

1. **MNIST**
2. Random Bars
3. Letters

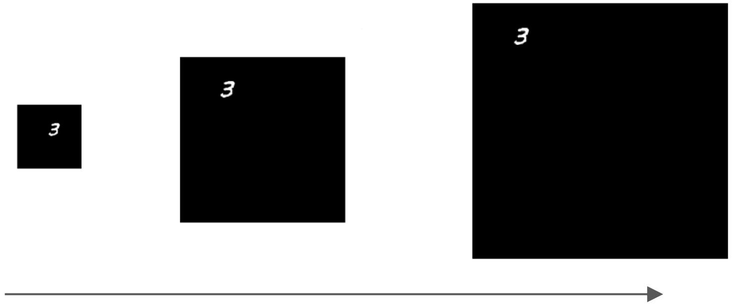
Metrics:

1. Accuracy
2. Convergence time
3. Multi-object scenes
4. Confidence

Resonator network hyperparameters:

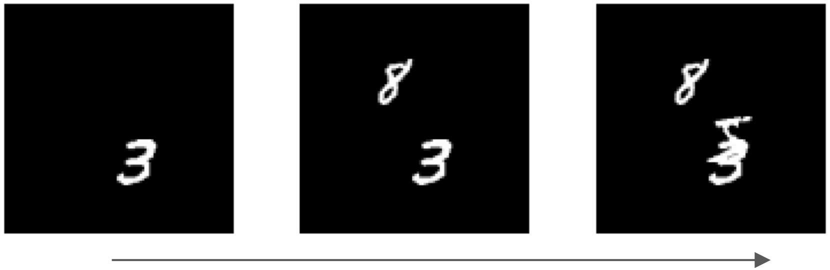
1. HD Vector dimension
2. Maximum number of iterations
3. Convergence criteria (fixed point vs. confidence-based)

Experiments on the translated MNIST dataset



Increasing the number of x,y positions (L)

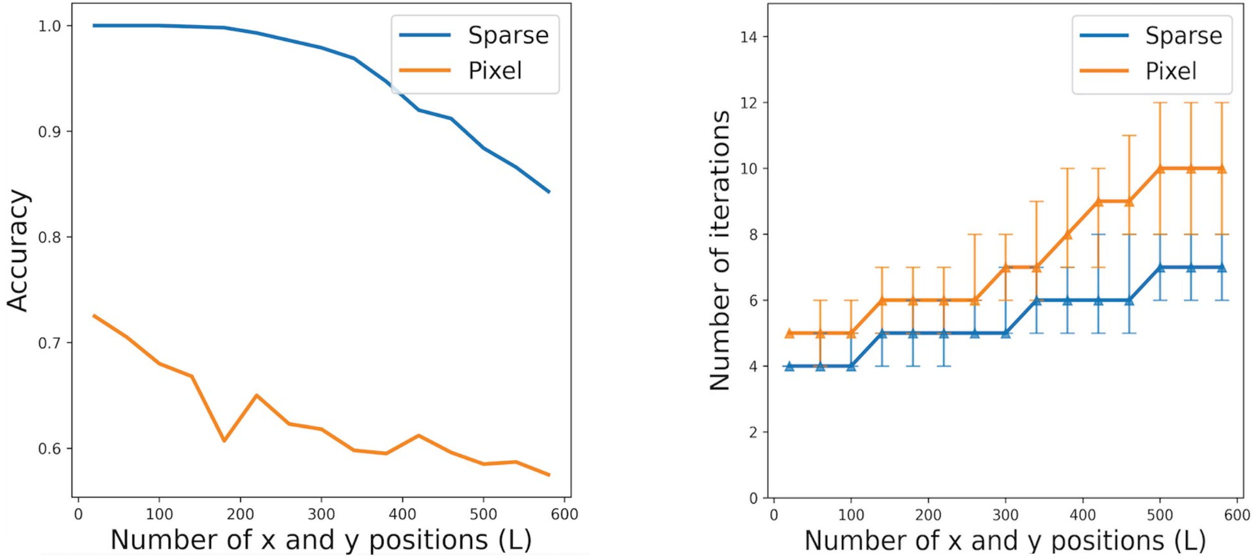
Search space size : $10L^2$



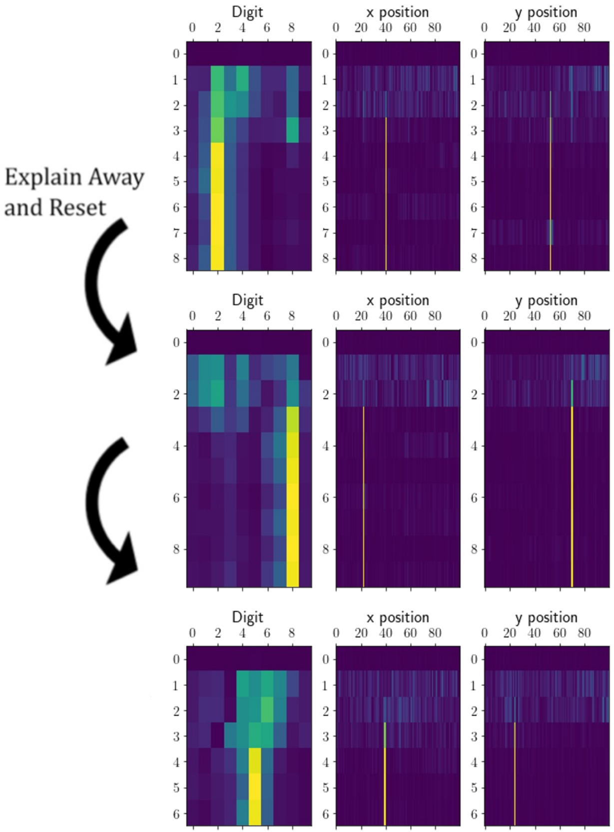
Increasing the number of objects

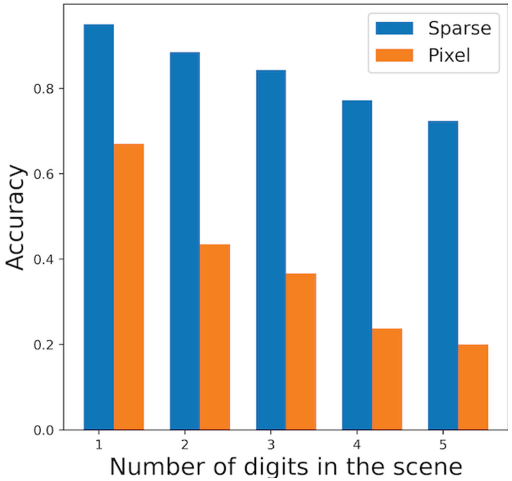
Object superposition causes crosstalk noise and pixel overlap.

Sparse coding improves accuracy and convergence time



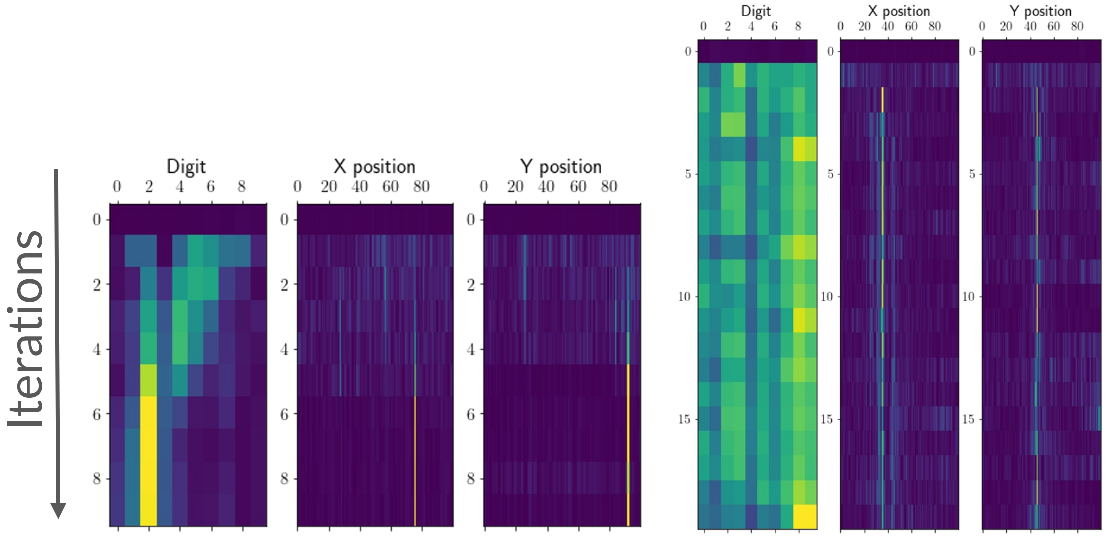
Results on the MNIST dataset : multi-object scenes





Convolutional sparse coding introduces efficient and compact representations, which we then adapt to HD computing.

Confidence as an early stopping criterion



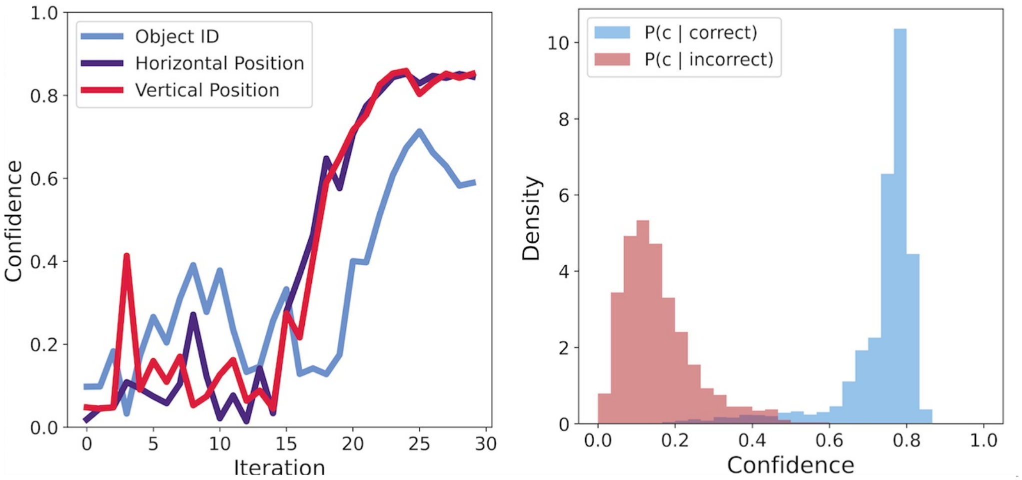
Correct: high confidence

Incorrect: low confidence

Confidence:

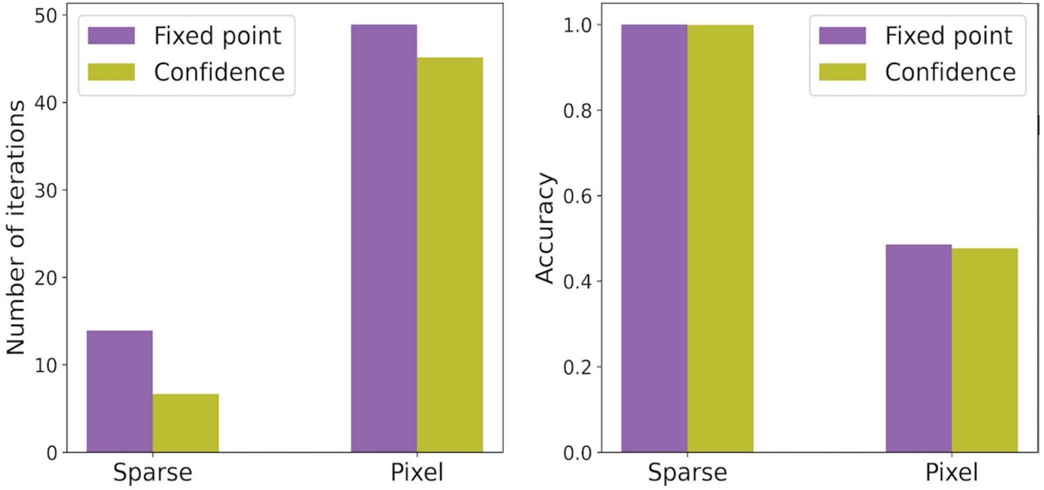
- Calculated for each codebook, for each iteration
- Difference between similarity of best guesses
- Normalized between 0 and 1

Confidence as an early stopping criterion



- Two stages in the resonator dynamics : **exploration** and **confirmation**.
- Final confidence is highly correlated with accuracy

Confidence as an early stopping criterion



- Early stopping : no decrease in accuracy, less iterations.
- Larger benefit for sparse encodings

Future directions

- Neuromorphic implementations
  - Sparse coding and resonators have been implemented in neuromorphic hardware*
- Scaling up to other kinds of scenes & variations
  - Video
  - Color
- Explore other resonator extensions
  - Efficiency gains from residue number systems
  - Nonlinearities in resonator network dynamics
  - Log-polar coordinate transformations to handle cases such as rotation and scaling

*for example,

Chavez Arana, D., Renner, A., & Sornborger, A. (2023, April). Spiking LCA in a Neural Circuit with Dictionary Learning and Synaptic Normalization. In Proceedings of the 2023 Annual Neuro-Inspired Computational Elements Conference (pp. 47-51).

Langenegger, J., Karunaratne, G., Hersche, M., Benini, L., Sebastian, A., & Rahimi, A. (2023). In-memory factorization of holographic perceptual representations. Nature Nanotechnology, 18(5), 479-485.

Wan, Z., Liu, C. K., Ibrahim, M., Yang, H., Spetalnick, S., Krishna, T., & Raychowdhury, A. (2024). H3DFact: Heterogeneous 3D Integrated CIM for Factorization with Holographic Perceptual Representations. arXiv preprint arXiv:2404.04173.

Takeaways

- Improvement in terms of accuracy and convergence time over pixel-based encodings.
- **Explicitly compositional** model: all images constructed from a relatively small number of codebook entries and operations.
- **Transparent** model: sparse coding makes the structure of images explicit.
- **Confidence**: method for faster convergence and resonator explainability.
- Connections to circuit models in computational neuroscience; improvements for neuromorphic computing

Thanks for your attention!

REDWOOD CENTER
for Theoretical Neuroscience