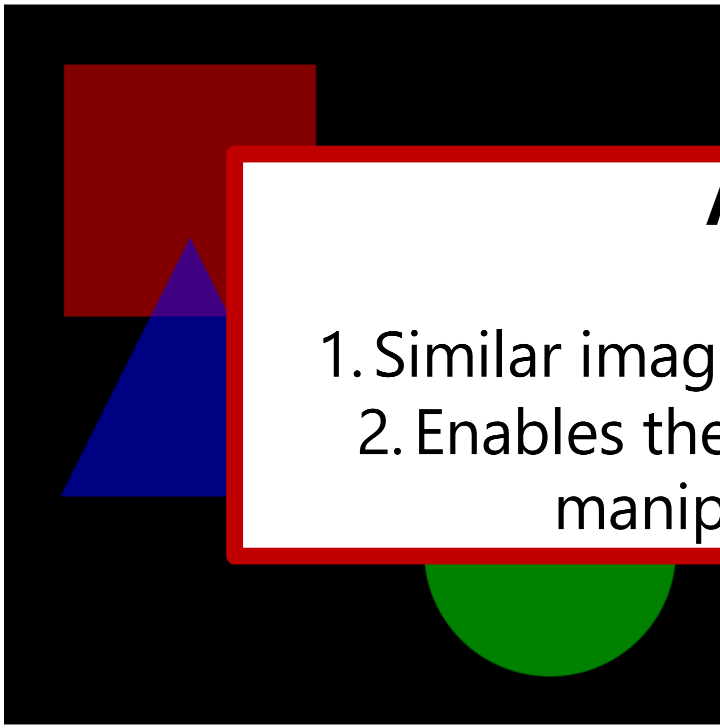
Going from Symbols to Scalars



VSA Formulation

$$I = \mathbf{c}_{red} \odot \mathbf{s}_{square} \odot \mathbf{v}_{top} \odot \mathbf{h}_{left}$$

$$\odot\, \mathbf{h}_{left}$$

$$\odot\, \mathbf{h}_{right}$$

$$\mathbf{h}^{51}$$

$$+\, \mathbf{c}^{465} \odot \mathbf{s}_{triangle} \odot \mathbf{v}^{198} \odot \mathbf{h}^{48.24}$$

$$+\, \mathbf{c}^{532} \odot \mathbf{s}_{circle} \odot \mathbf{v}^{352.1} \odot \mathbf{h}^{344}$$

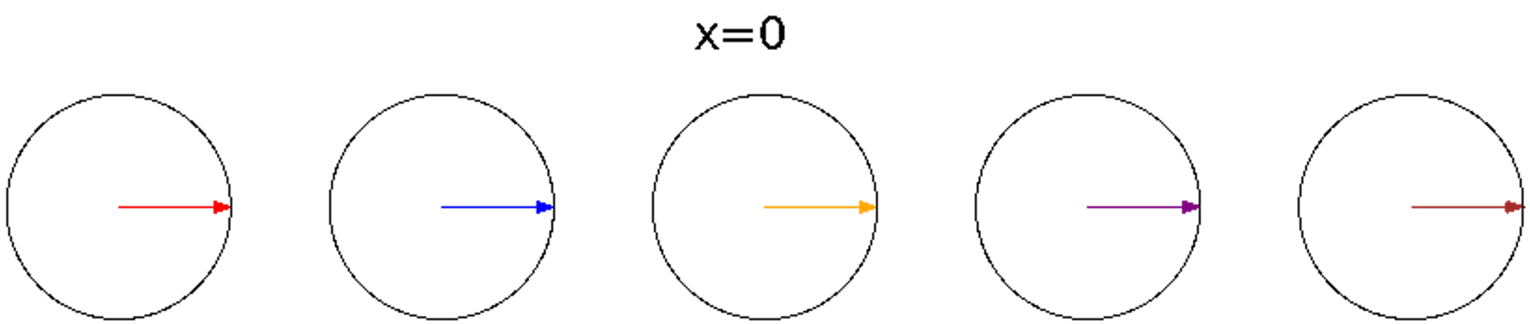
Advantages:

1. Similar images have high inner product
2. Enables the definition, and algebraic manipulation, of functions

Encoding real numbers via **fractional binding**

Key idea 1: Represent any number x, by binding **z** *x* times with itself:

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



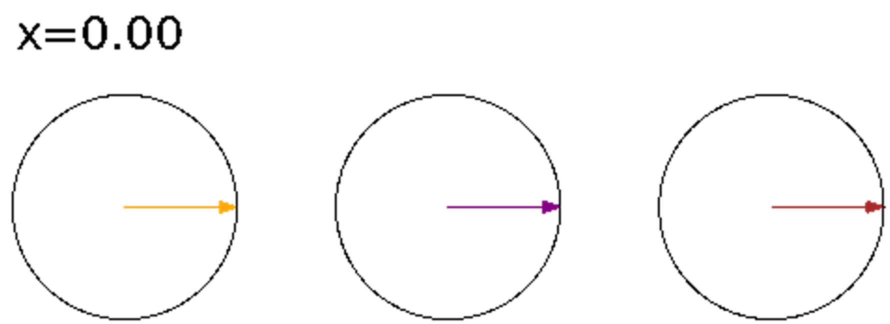
T.A. Plate, “Holographic Recurrent Networks,” Advances in Neural Information Processing Systems (NIPS), pp. 34-41, 1992.
T.A. Plate, “Distributed Representations and Nested Compositional Structure,” University of Toronto, PhD Thesis, 1994.

Encoding real numbers via **fractional binding**

Key idea 1: Represent any number x, by binding **z** *x* times with itself:

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

Key idea 2: Extend this definition to support encoding of non-integer *x* values



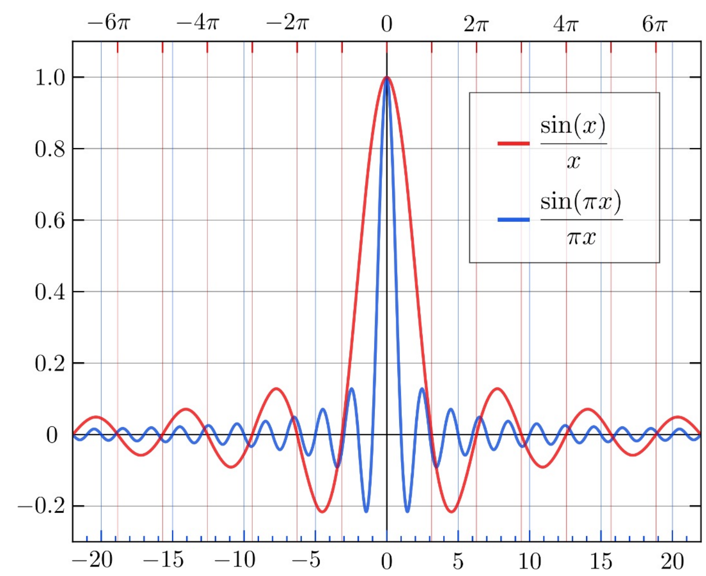
T.A. Plate, “Holographic Recurrent Networks,” Advances in Neural Information Processing Systems (NIPS), pp. 34-41, 1992.
T.A. Plate, “Distributed Representations and Nested Compositional Structure,” University of Toronto, PhD Thesis, 1994.

Fractional binding creates a **similarity-preserving encoding (kernel)**

$$z(x)^T \overline{z(x + \Delta x)} = K(\Delta x) \approx \frac{sin(\pi x)}{\pi x}$$

Inner Product **Kernel** **sinc(x)**

K(Δx)



Representing functions with support points

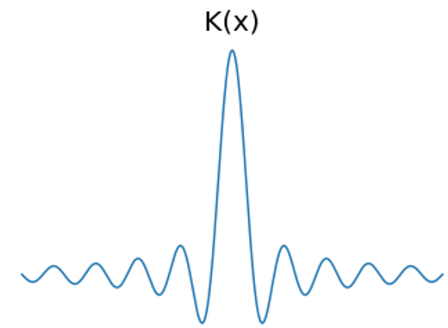
A function:

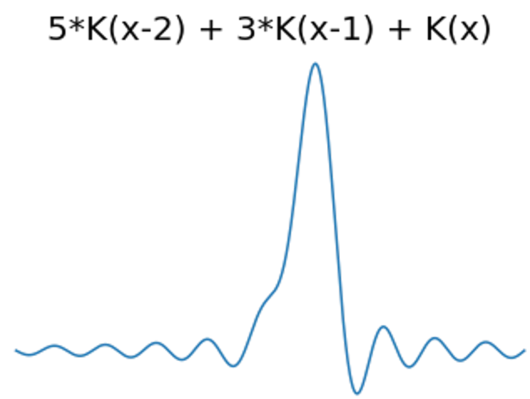
$$f(r) = \sum_k \alpha_k K(r - r_k)$$

is represented as a vector:

$$\mathbf{y}_f \quad = \quad \sum_k \alpha_k \mathbf{z}(r_k)$$

$$f(r) = \mathbf{y}_f^\top \overline{\mathbf{z}(r)}$$





Representing functions with support points

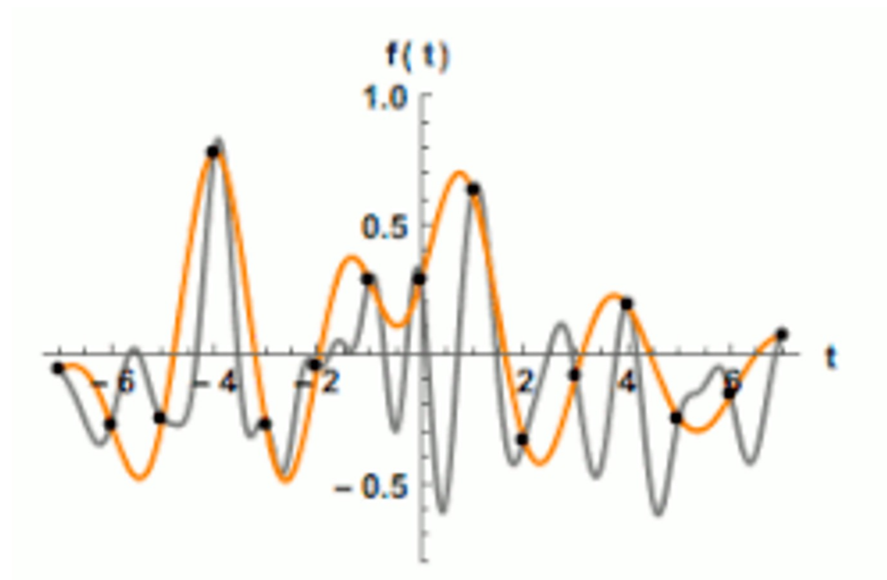
A function:

$$f(r) = \sum_k \alpha_k K(r - r_k)$$

is represented as a vector:

$$\mathbf{y}_f \quad = \quad \sum_k \alpha_k \mathbf{z}(r_k)$$

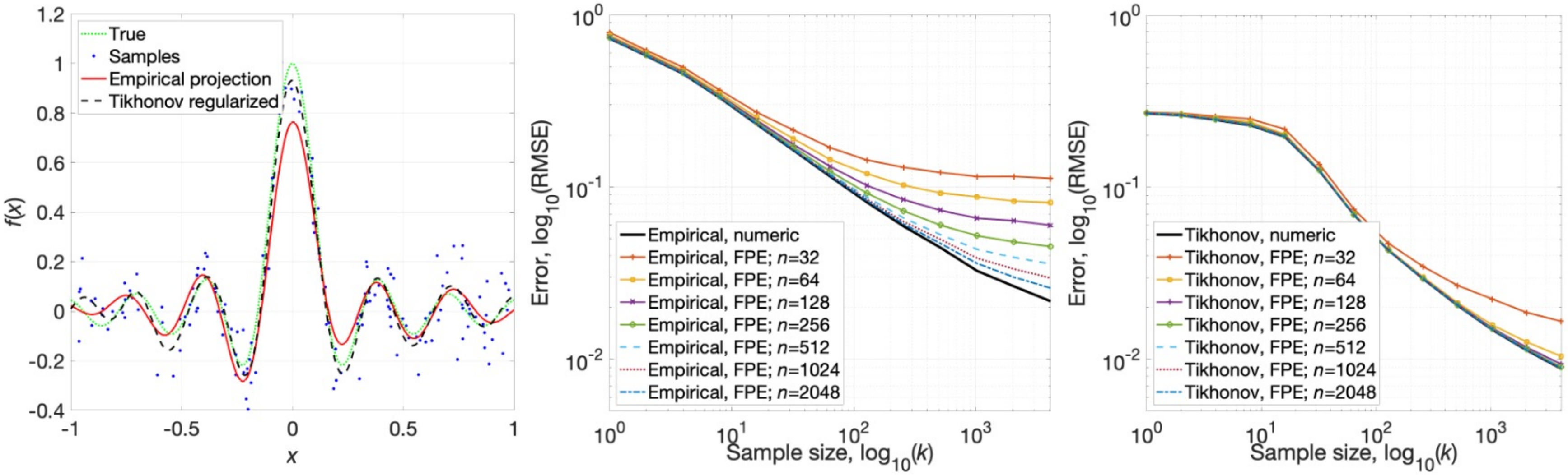
$$f(r) = \mathbf{y}_f^\top \overline{\mathbf{z}(r)}$$



https://en.wikipedia.org/wiki/Whittaker%E2%80%93Shannon_interpolation_formula

Application: regression with sinc kernels

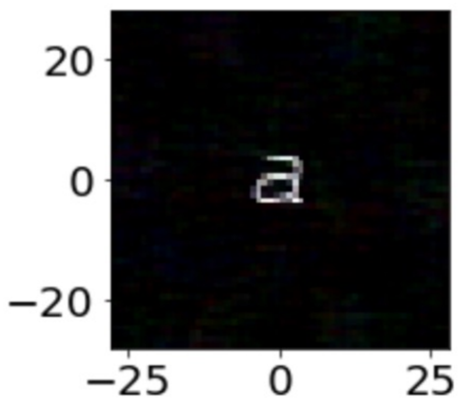
$$\hat{f}_{c,k,n}(x) = \frac{2c}{k\pi}\sum_{i=1}^{k} Y_i \mathrm{sinc}\left(\frac{c}{\pi}(X_i - x)\right) = \frac{2c}{kn\pi}\sum_{i=1}^{k} Y_i \mathbf{z}(\frac{c}{\pi}X_i)^\top \overline{\mathbf{z}(\frac{c}{\pi}x)} = (\mathbf{y}^X)^\top \overline{\mathbf{z}(\frac{c}{\pi}x)},$$



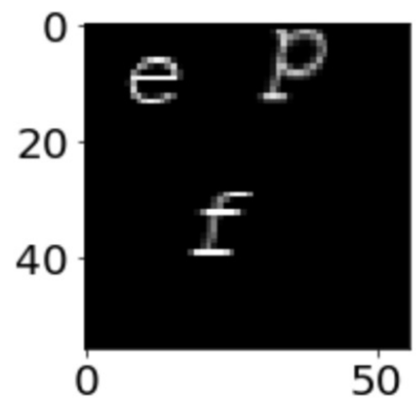
Manipulating functions in VFA

- Point-wise readout of a function $f(s) = \langle f, K_s \rangle = \mathbf{y}_f^\top \overline{\mathbf{z}(s)}$
- Point-wise addition $\mathbf{y}_{f+g} = \mathbf{y}_f + \mathbf{y}_g$
- Function shifting $\mathbf{y}_g = \mathbf{y}_f \circ \mathbf{z}(r)$
  $f(x) \to g(x) = f(x+r)$
- Function convolution $\mathbf{y}_{f*g} = \mathbf{y}_f \circ \mathbf{y}_g$
- Overall similarity between functions $\langle f, g \rangle = \mathbf{y}_f^\top \overline{\mathbf{y}_g}$

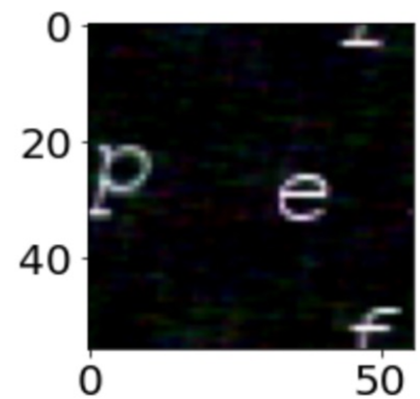
Representing images in VFA



$$\mathbf{v}_a = \sum_{x,y} Im(x, y) \cdot \mathbf{x}^x \odot \mathbf{y}^y$$

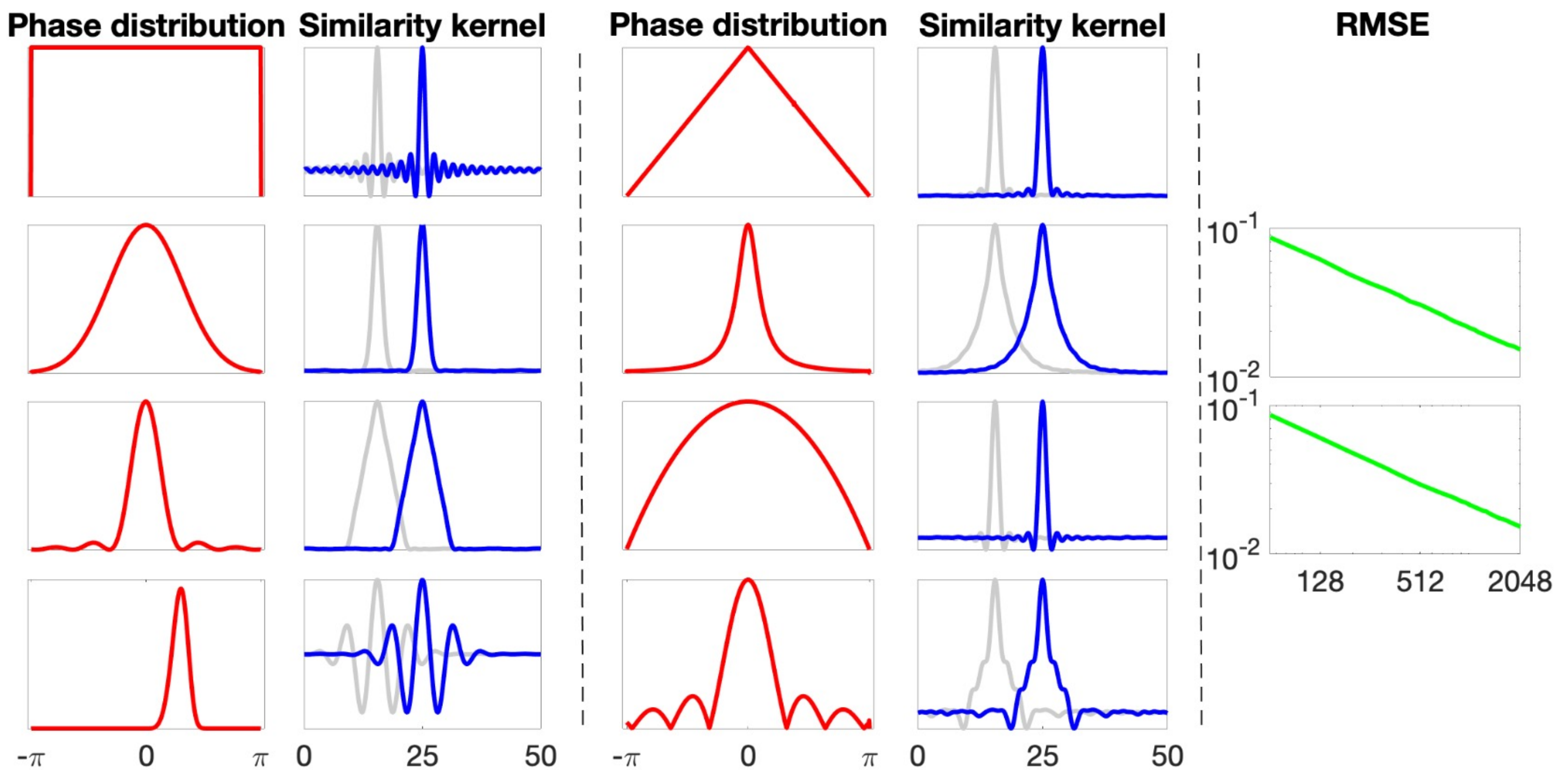


$$\begin{aligned}\mathbf{v}_{scene} &= \mathbf{v}_e \odot \mathbf{x}^{12} \odot \mathbf{y}^{10} \\ &+ \mathbf{v}_p \odot \mathbf{x}^{35.86} \odot \mathbf{y}^{7.22} \\ &+ \mathbf{v}_f \odot \mathbf{x}^{22.7} \odot \mathbf{y}^{35}\end{aligned}$$



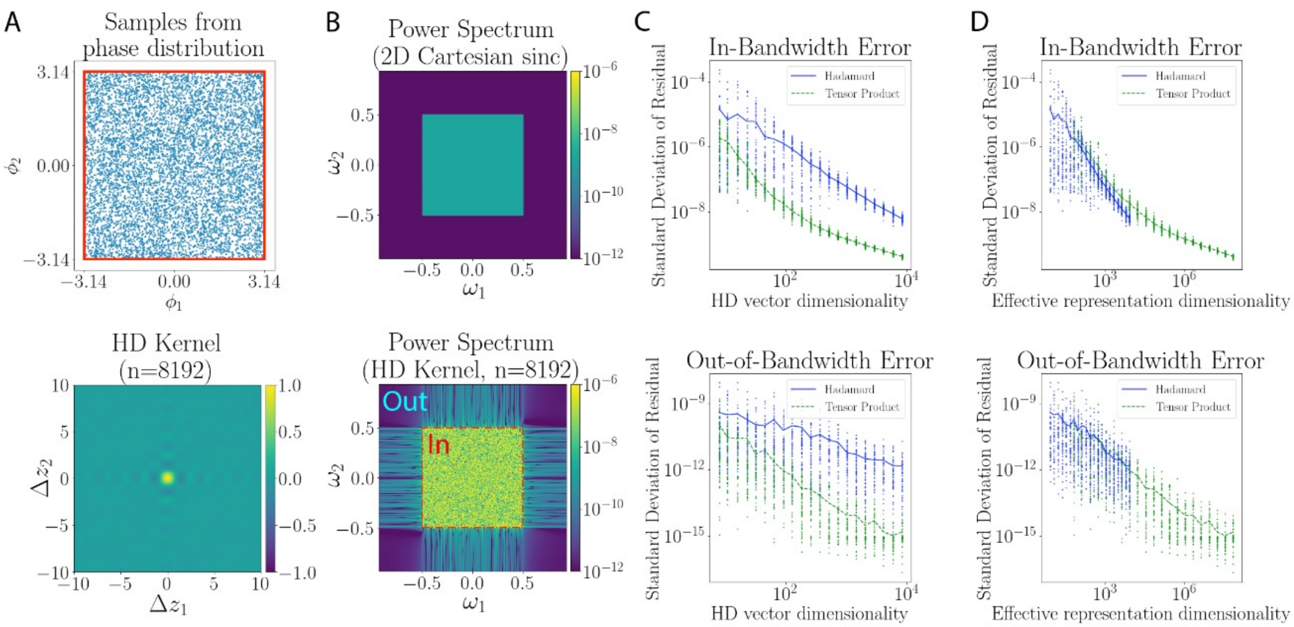
$$\mathbf{v}^{tr}_{scene} = \mathbf{v}_{scene} \odot \mathbf{x}^{25.8} \odot \mathbf{y}^{20.2}$$

Phase distribution of the base vector determines similarity kernel

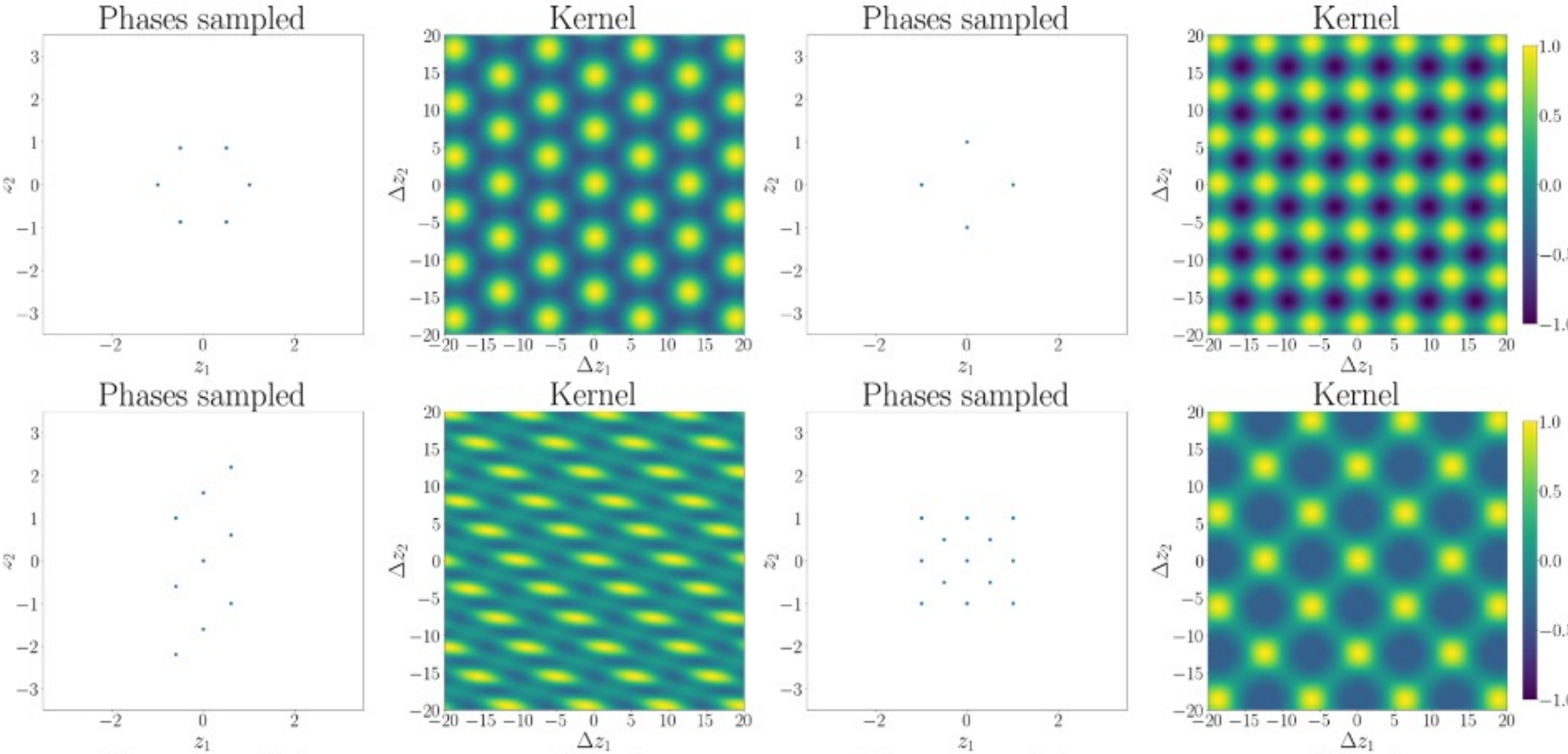


Binding FPEs to produce multi-dimensional kernels

$$\mathbf{z}(\mathbf{r}) = \mathbf{z}_1(r_1) \circ \mathbf{z}_2(r_2) \circ \ldots \circ \mathbf{z}_m(r_m)$$

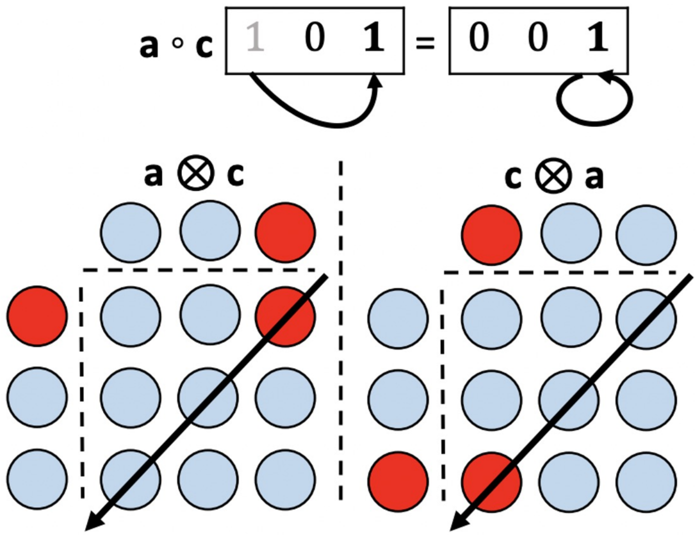


Periodic 2D kernels (Lattices)

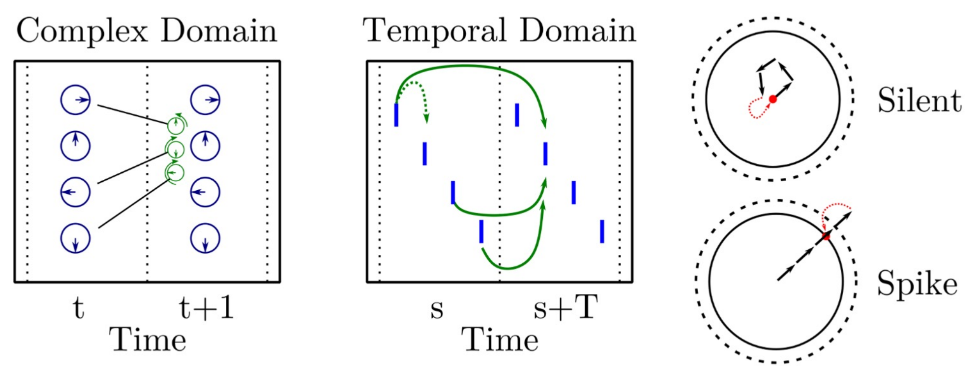


How to map VFA onto neural circuits?

1. Use *sparse* high-dimensional vectors



2. Map amplitudes to firing rate and phases to spike timing



Frady, E.P., Sommer, F.T. (2019) Robust computation with rhythmic spike patterns. PNAS 116(36) 18050-59.
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.

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

Block sparse HD vector: n=18

1 0 0	0 0 1	1 0 0	0 1 0	0 1 0	0 0 1

Block: k=3

- 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

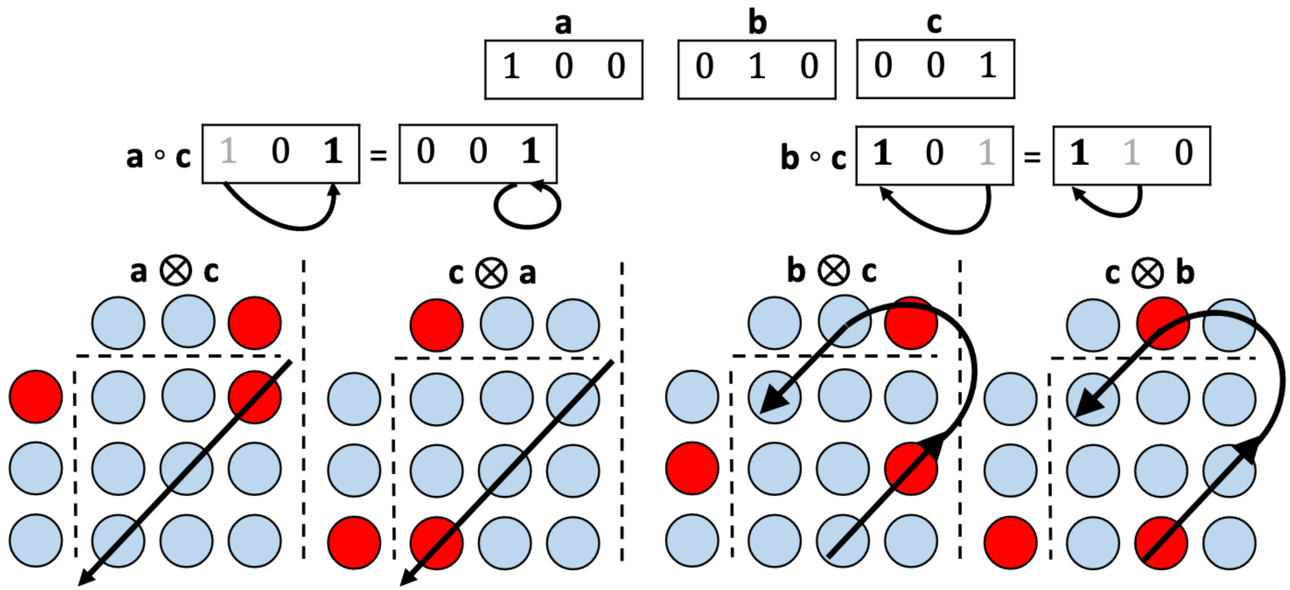
Mika Laiho

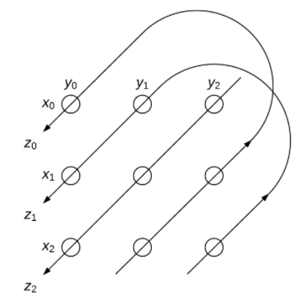
M. Laiho, et al., “High-Dimensional Computing with Sparse Vectors,” IEEE Biomedical Circuits and Systems Conference (BioCAS), 2015.

Slide from Denis Kleyko

Sparse Block-Codes: binding

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

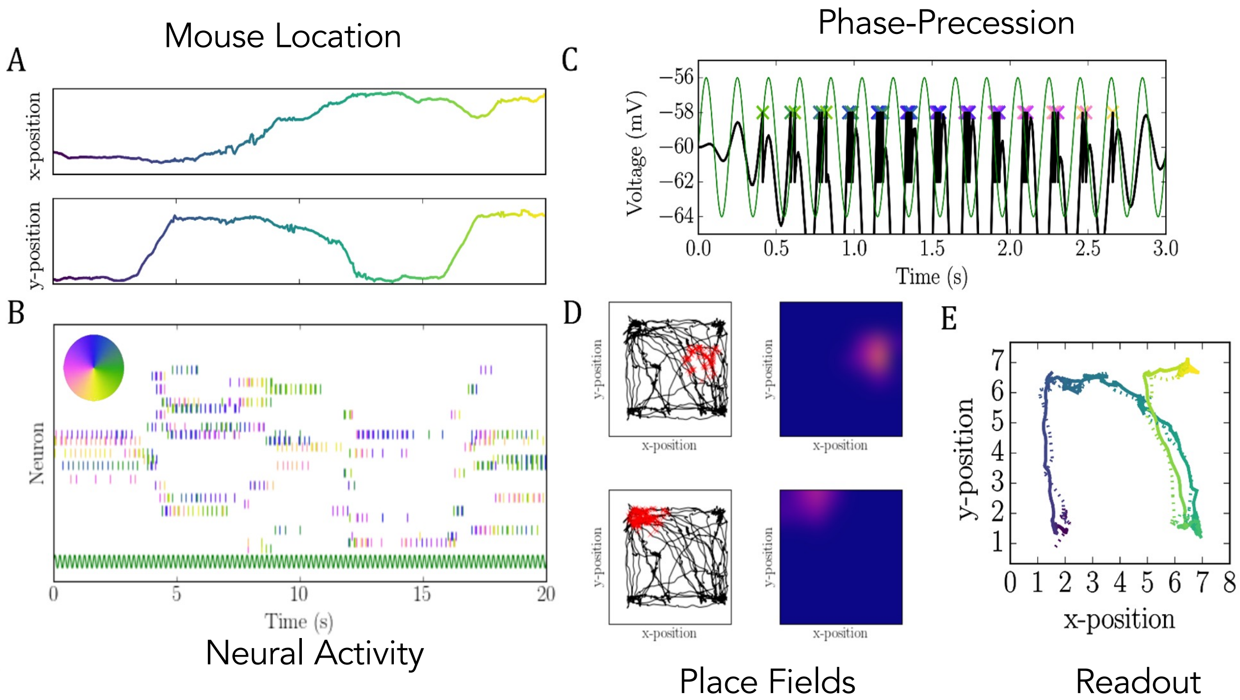




M. Laiho, et al., “High-Dimensional Computing with Sparse Vectors,” IEEE Biomedical Circuits and Systems Conference (BioCAS), 2015.
E. P. Frady, et al., “Variable Binding for Sparse Distributed Representations: Theory and Applications,” IEEE Transactions on Neural Networks and Learning Systems, 2021.

Slide from Denis Kleyko

An encoding model of hippocampus predicts place fields and phase precession



Frady, P., Kanerva, P., & Sommer, F. (2018). A framework for linking computations and rhythm-based timing patterns in neural firing, such as phase precession in hippocampal place cells. In *Proceedings of the Conference on Cognitive Computational Neuroscience*.