The Capacity of VSA representations

Friedrich T. Sommer
NCL Intel Labs, Intel
and
UC Berkeley
Redwood Center for Theoretical Neuroscience
Helen Wills Neuroscience Institute

VSA Lecture, October 13, 2021

The traps of theoretical neuroscience

Problems with reverse engineering the brain:

- "Neuromimicry" – looks brain-like but does not explain brain function
- Normative models often too simple and single minded
- Neural network models are black boxes themselves – limited explanatory value
- Neuroscience data are complicated and spotty – models do not just emerge from data analysis
- Kuhn cycle between experiment and theory still not productive in neuroscience

VSA

Structured computing with distributed representations:

- Can represent data structures by vectors
- Data structures represented by vectors of same dimension – this has to be lossy
- Compute in superposition i.e., search set of items simultaneously
- Binding is also lossy
- Memory-based error correction interspersed with computation

Open questions we faced around 2017:

- Capacity: How many items can be superimposed in VSA vector ?
- How different are different VSA models ?
- Connections between VSA algorithms and neural networks ?

Mapping data to vector spaces

Source coding (remove redundancy in data)

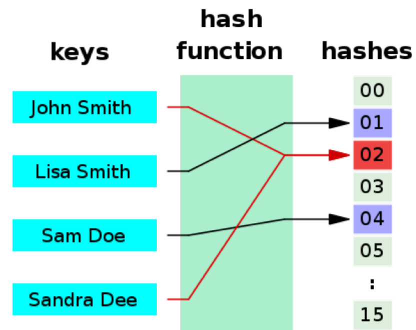
Data lie in subspace (SS)	*Learning method*	*Coordinates in SS*
Linear low-D SS	PCA	Axes of covariance matrix
Nonlinear low-D SS	Manifold learning	location on manifold
Clusters	Cluster analysis	Cluster number (+ loc.)
Union of lin. low-D SS	Sparse coding	Axes of Indep. Comp.
Union of nonlin. Low-D SS	Manifold learning	Manifold number + loc.

Vector encoding of the new coordinates

Feature local: a neuron's activity encodes a coordinate: PCA, ICA,…

Distributed: values of a coordinate are represented by many neurons: VSA

Hashing vs. VSA



Data indexing:

Hash function: data points -> index space

Properties: uniformity

efficiency: computational complexity and collision handling vs. compactness of indices

avalanche criterion: Single bit flip in input -> each output bit changed with p=0.5

In VSA: symbols -> i.i.d. random vectors ~ P(x)

Requires lookup table of assigned vectors in memory

In VFA: LPE: data points -> randomized representations with kernel property

Encoding sequences of vectors in VSA

"write" "read"

$$(a_1, a_2) \rightarrow \mathbf{x} = a_1 \mathbf{\Phi}_1 + a_2 \mathbf{\Phi}_2 \qquad \hat{a}_i = (\mathbf{\Phi}_i)^T \mathbf{x}$$

In high dimension random vectors are almost orthogonal

Forming unique encoding vectors for each time step:

$$(a_1(t), a_2(t)) \rightarrow \mathbf{x}(t) = a_1 \mathbf{W}^t \mathbf{\Phi}_1 + a_2 \mathbf{W}^t \mathbf{\Phi}_2$$

with W orthogonal matrix.

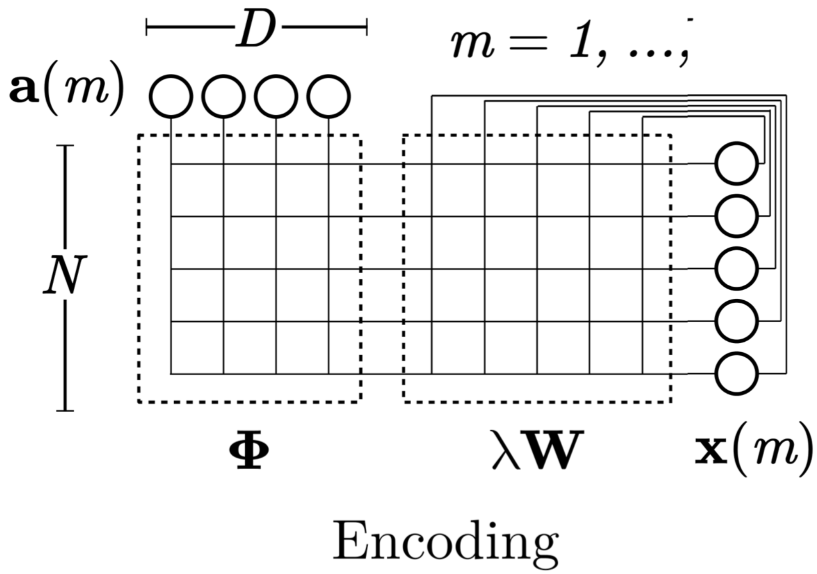
Encoding of entire time series: $\mathbf{x} = \sum_{t=0}^{M} \mathbf{x}(t)$

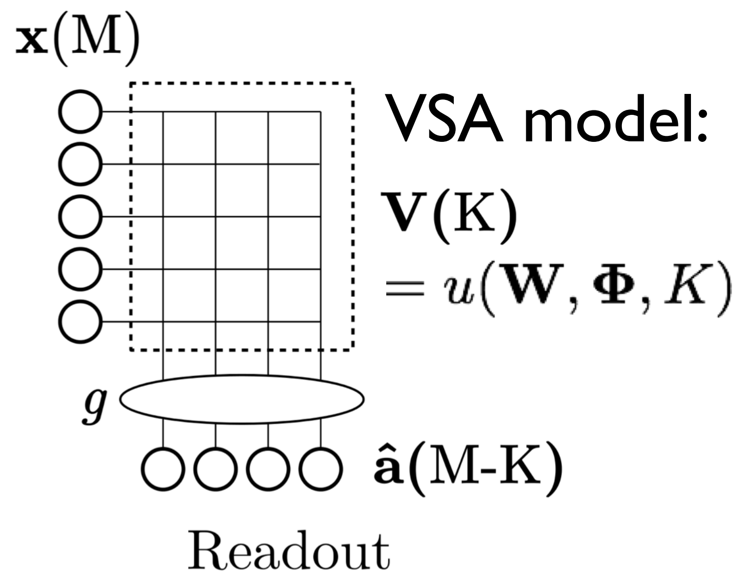
Readout: $\hat{a}_i(t) = (\mathbf{W}^t \mathbf{\Phi}_i)^\top \mathbf{x}$

VSA sequence encoding network model

Network for “write“

Network for “read”





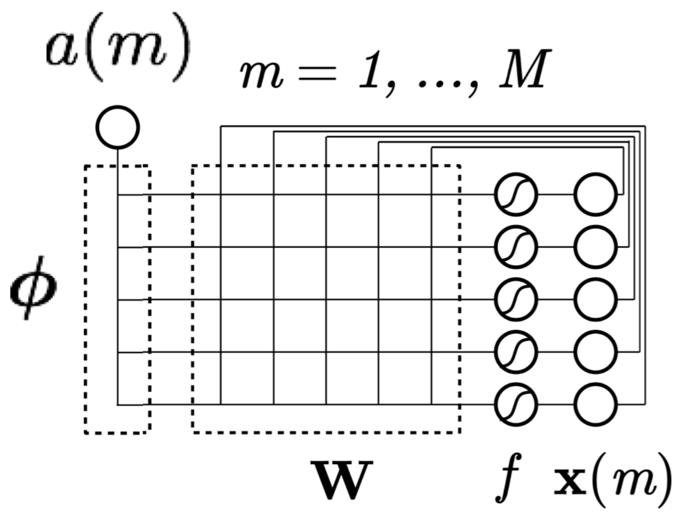
with Φ pseudorandom, and W orthogonal with long cycle length

Cases considered:

Capacity $\left(\frac{bits}{neuron}\right)$	memory type reset	memory type buffer
symbolic analog	?	?

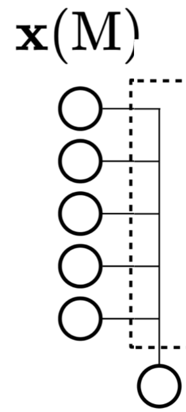
Reservoir network

Network for "write"



Encoding

Network for "read"



Optimal readout:

Linear regression

between a and x:

$\mathbf{v}(K)$

$\hat{a}(M-K)$

Readout

Echostate networks (Jaeger), Liquid state networks (Maass),

State-dependent Networks (Buonomano)

Questions: How to Dissect dynamics into computational operations?

How accurately can memory items be accessed?

How much bits/neuron can be stored?

What are nonlinear neurons good for?

Predicting readout in Reservoir network

Encoding: $$\mathbf{x}(m) = f(\lambda \mathbf{W}\mathbf{x}(m-1) + \mathbf{\Phi}\mathbf{a}(m) + \boldsymbol{\eta}(m)) \tag{1}$$

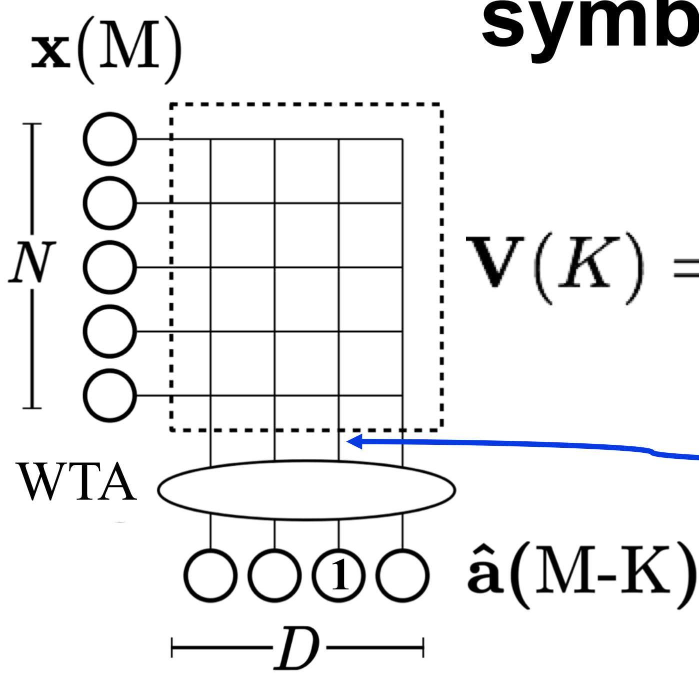
Readout: $$\hat{\mathbf{a}}(M-K) = g(\mathbf{V}(K)^\top \mathbf{x}(M)) \tag{2}$$

The effect of one iteration of equation (1) on the probability distribution of the network state $\mathbf{x}(m)$ is a Markov chain stochastic process, governed by the Chapman-Kolmogorov equation (Papoulis, 1984):

$$p(\mathbf{x}(m+1)|\mathbf{a}(m)) = \int p(\mathbf{x}(m+1)|\mathbf{x}(m), \mathbf{a}(m))\ p(\mathbf{x}(m))\ d\mathbf{x}(m) \tag{3}$$

with a transition kernel $p(\mathbf{x}(m+1)|\mathbf{x}(m), \mathbf{a}(m))$, which depends on all parameters and functions in (1). Thus, to analyze the memory performance in general, one has to iterate equation (3) to obtain the distribution of the network state.

Analysing decoding performance for symbolic sequences



$$\mathbf{V}(K) = c^{-1}\mathbf{W}^K\mathbf{\Phi}$$

$$h_d(K) := \mathbf{V}_d(K)^\top \mathbf{x}(M) \quad (8)$$

$$h_d(K) = \sum_{i=1}^{N} \left(\mathbf{V}_d(K)^\top \mathbf{x}(M)\right)_i = c^{-1}\sum_{i=1}^{N}(\mathbf{\Phi}_d)_i(\mathbf{W}^{-K}\mathbf{x}(M))_i = c^{-1}\sum_{i=1}^{N} z_{d,i} \quad (30)$$

$\mu(h_d) = c^{-1}N\mu(z_{d,i})$ and $\sigma^2(h_d) = c^{-1}N\sigma^2(z_{d,i})$.

Concentration of measure phenomenon (Ledoux, 2001):

$$h_d(K) \rightarrow c^{-1}N\mu(z_{d,i})$$

Convergence fast in N – Hoeffding's inequality (Plate 1993, Thomas et al, 2020)

But what happens at some fixed finite N ?

Detection theory

Accuracy (d' is index of correct component):

Linear readout: $h_d(K) = a_d(M - K) + n_d$

with n describing network and crosstalk noise

$$
\begin{aligned}
p_{corr}(K) &= p\left(h_{d'}(K) > h_d(K)\ \forall d \neq d'\right) \\
&= \int_{-\infty}^{\infty} p(h_{d'}(K) = h)\left[p(h_d(K) < h)\right]^{D-1} dh \\
&= \int_{-\infty}^{\infty} \mathcal{N}(h'; \mu(h_{d'}), \sigma^2(h_{d'}))\left[\int_{-\infty}^{h'} \mathcal{N}(h; \mu(h_d), \sigma^2(h_d))\, dh\right]^{D-1} dh' \\
&= \int_{-\infty}^{\infty} \mathcal{N}(h'; a_{d'}, \sigma^2(n_{d'}))\left[\int_{-\infty}^{h'} \mathcal{N}(h; a_d, \sigma^2(n_d)) dh\right]^{D-1} dh'
\end{aligned}
\tag{10}
$$

The Gaussian variables h and h' in (10) can be shifted and rescaled to yield:

$$p_{corr}(K) = \int_{-\infty}^{\infty} \frac{dh}{\sqrt{2\pi}}\, e^{-\frac{1}{2}h^2} \left[\Phi\left(\frac{\sigma(h_d)}{\sigma(h_{d'})}h - \frac{\mu(h_d) - \mu(h_{d'})}{\sigma(h_{d'})}\right)\right]^{D-1}$$
$$= \int_{-\infty}^{\infty} \frac{dh}{\sqrt{2\pi}}\, e^{-\frac{1}{2}h^2} \left[\Phi\left(\frac{\sigma(n_d)}{\sigma(n_{d'})}h - \frac{a_d - a_{d'}}{\sigma(n_{d'})}\right)\right]^{D-1} \quad (11)$$

where Φ is the Normal cumulative density function.

Further simplification can be made when $\sigma(n_{d'}) \approx \sigma(n_d)$.

The *accuracy* then becomes:

$$p_{corr}(s(K)) = \int_{-\infty}^{\infty} \frac{dh}{\sqrt{2\pi}}\, e^{-\frac{1}{2}h^2} \left[\Phi\left(h + s(K)\right)\right]^{D-1} \quad (12)$$

where the *sensitivity* for detecting the hot component d' from $\mathbf{h}(K)$ is defined:

$$s(K) := \frac{\mu(h_{d'}) - \mu(h_d)}{\sigma(h_d)} = \frac{a_{d'} - a_d}{\sigma(n_d)} = \frac{1}{\sigma(n_d)} \quad (13)$$

Computing accuracy for a particular VSA model

$$h_d(K) = \sum_{i=1}^{N} \left(\mathbf{V}_d(K)^\top \mathbf{x}(M)\right)_i = c^{-1} \sum_{i=1}^{N} (\mathbf{\Phi}_d)_i (\mathbf{W}^{-K} \mathbf{x}(M))_i = c^{-1} \sum_{i=1}^{N} z_{d,i} \quad (30)$$

The quantity $z_{d,i}$ in (30) can be written:

$$\begin{aligned} z_{d,i} &= (\mathbf{\Phi}_d)_i (\mathbf{W}^{-K} \mathbf{x}(M))_i \\ &= \begin{cases} (\mathbf{\Phi}_{d'})_i (\mathbf{\Phi}_{d'})_i + \sum_{m \neq (M-K)}^{M} (\mathbf{\Phi}_{d'})_i (\mathbf{W}^{M-K-m} \mathbf{\Phi}_{d'})_i & \text{if } d = d' \\ \sum_{m}^{M} (\mathbf{\Phi}_d)_i (\mathbf{W}^{M-K-m} \mathbf{\Phi}_{d'})_i & \text{otherwise} \end{cases} \end{aligned} \quad (31)$$

Given the conditions (4)-(7), the moments of $z_{d,i}$ can be computed:

depend on VSA model

$$\mu(z_{d,i}) = \begin{cases} E_{\mathbf{\Phi}}(x^2) + (M-1) E_{\mathbf{\Phi}}(x)^2 & \text{if } d = d' \\ M E_{\mathbf{\Phi}}(x)^2 & \text{otherwise} \end{cases} \quad (32)$$

$$\sigma^2(z_{d,i}) = \begin{cases} V_{\mathbf{\Phi}}(x^2) + (M-1) V_{\mathbf{\Phi}}(x)^2 & \text{if } d = d' \\ M V_{\mathbf{\Phi}}(x)^2 & \text{otherwise} \end{cases} \quad (33)$$

For networks with N large enough, $p(h_d(K)) \sim \mathcal{N}(c^{-1}N\mu(z_{d,i}), c^{-1}N\sigma^2(z_{d,i}))$. By inserting $\mu(h_d)$ and $\sigma(h_d)$ into (11), the accuracy then becomes:

$$p_{corr} = \int_{-\infty}^{\infty} \frac{dh}{\sqrt{2\pi}} e^{-\frac{1}{2}h^2} \times \\ \left[\Phi\left(\sqrt{\frac{M}{M-1+V_{\mathbf{\Phi}}(x^2)/V_{\mathbf{\Phi}}(x)^2}}\, h + \sqrt{\frac{N}{M-1+V_{\mathbf{\Phi}}(x^2)/V_{\mathbf{\Phi}}(x)^2}}\right)\right]^{D-1} \tag{34}$$

Analogous to (12), for large M the expression simplifies further to:

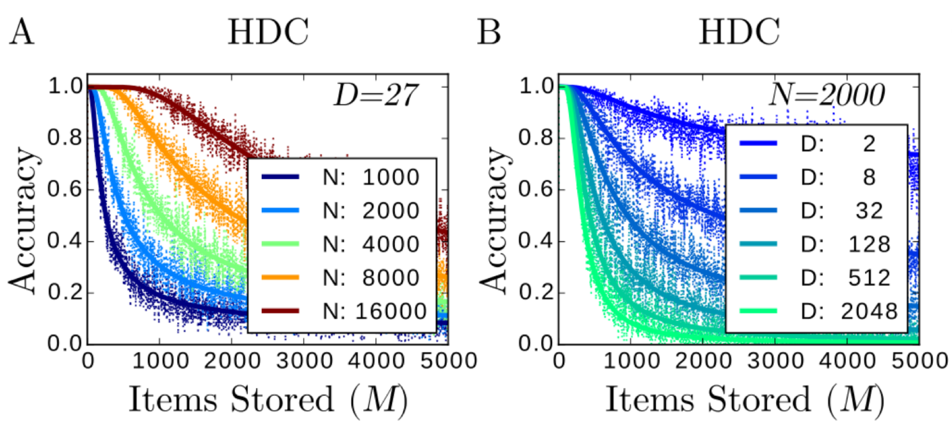
$$\boxed{p_{corr}(s) = \int_{-\infty}^{\infty} \frac{dh}{\sqrt{2\pi}} e^{-\frac{1}{2}h^2} \left[\Phi\left(h+s\right)\right]^{D-1} \quad \text{with } s = \sqrt{\frac{N}{M}}} \tag{35}$$

Capacity for existing VSA models

HDC: Hyperdimensional Computing – binary/bipolar (Kanerva)

HRR: Holographic reduced representations - real-valued (Plate)

FHRR: Fourier HRR - complex-valued (Plate)



Universality:

For all models

$$s(K) = \sqrt{\frac{N}{M}}$$

Same performance!

Relation to previous VSA theory (Plate 1993)

Hi-Fi regime

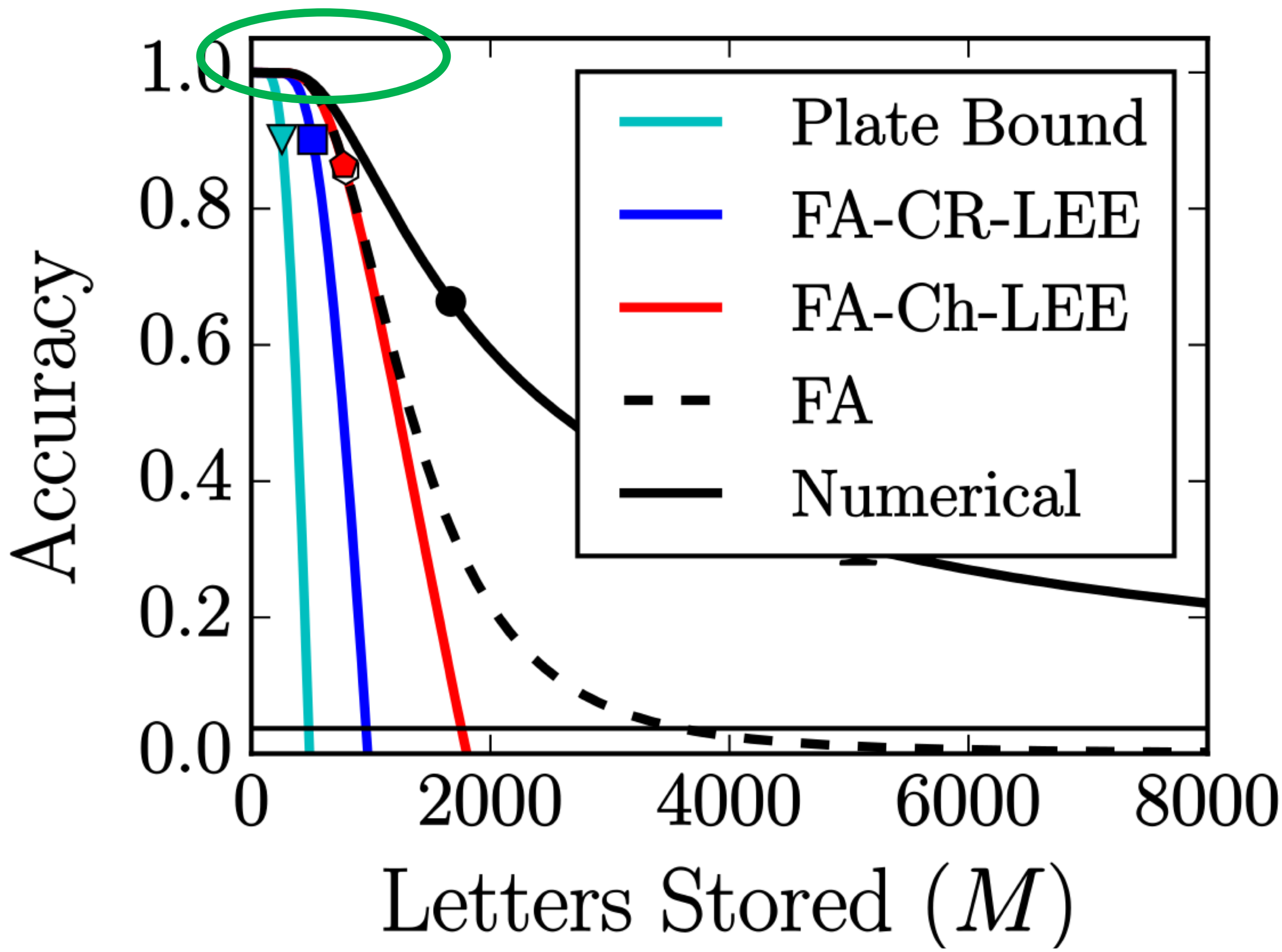
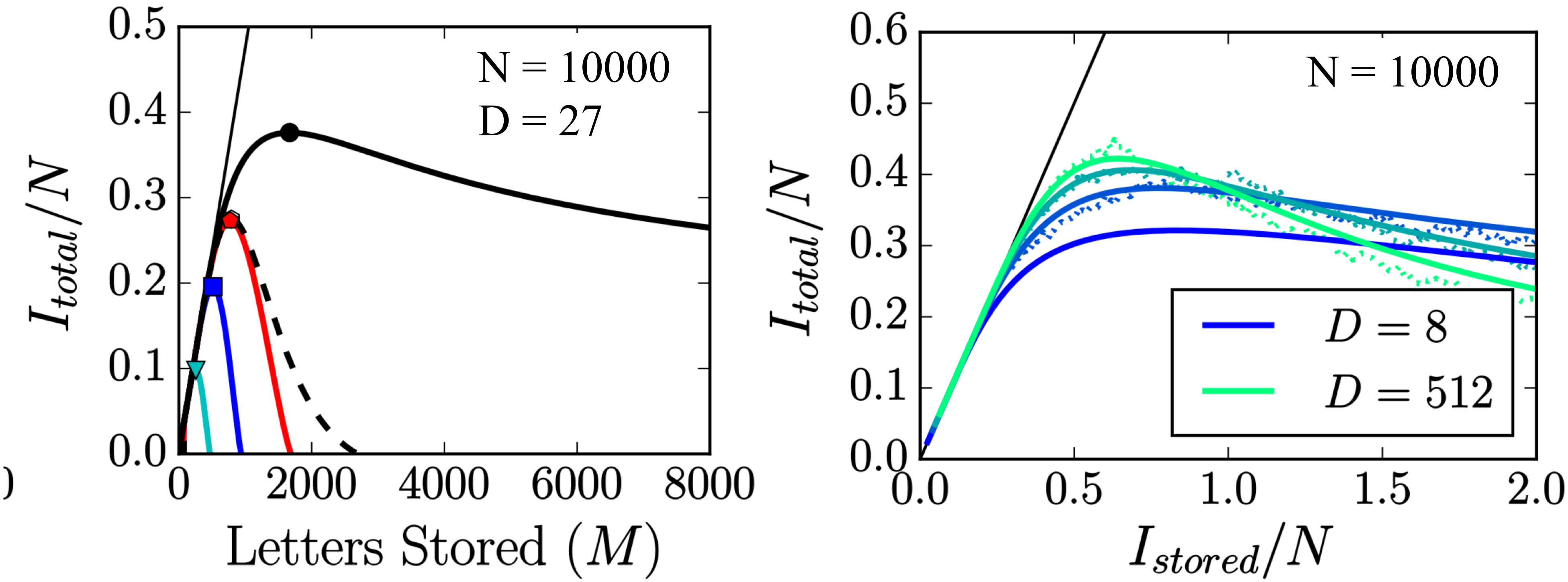


Plate theory (underestimates performance)

Information capacity of reset memories



- High-fidelity regime: 0.3 bits/neuron, not 0.1 bits/neuron
- Total maximum of capacity at lower fidelity
- Higher capacity for larger alphabet sizes

Memory buffer

Always store recent sequence by replacing hard reset by gradual forgetting mechanism

- Implements in VSA volatile data structure in which time stamped data are exchanged continuously
- Working memory in the brain? Recency effect

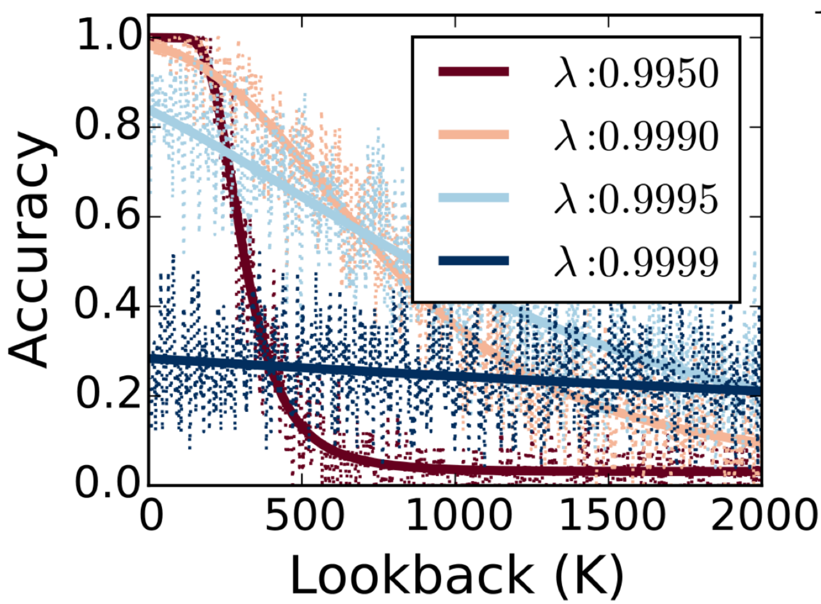
Questions:

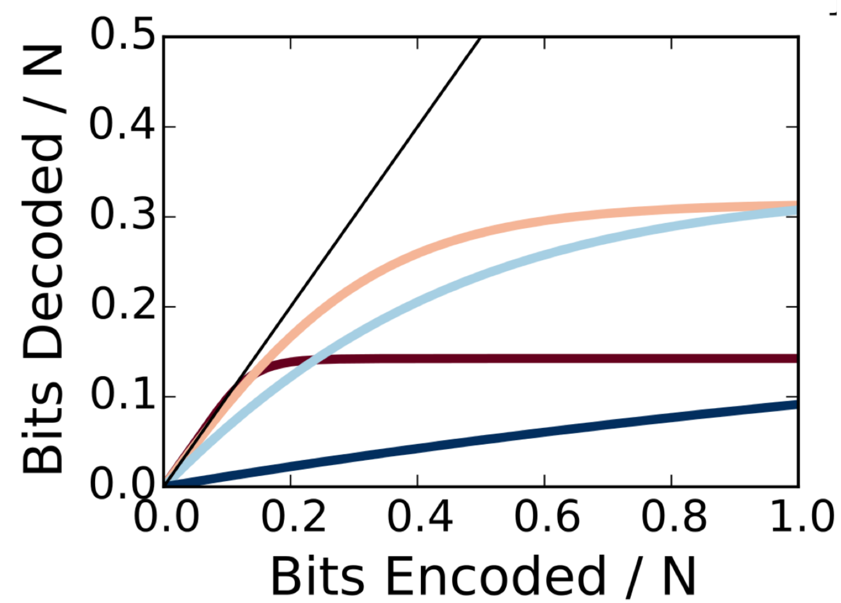
- Performance of different forgetting mechanisms:
  linear contraction, different types of neural nonlinearity ?
- Capacity comparision to static reset network ?
- Does the theory still work ?

Memory buffer with linear contraction

Sensitivity (for reset (λ=1): $s(K) = \sqrt{\frac{N}{M}}$)

for buffer (M=∞, λ<1): $s(K) = \lambda^K \sqrt{N(1-\lambda^2)}$





Theory works!

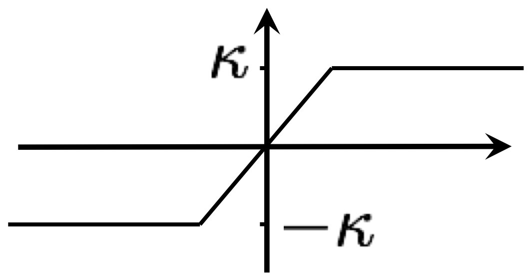
Capacity somewhat lower (70%) than for reset memory

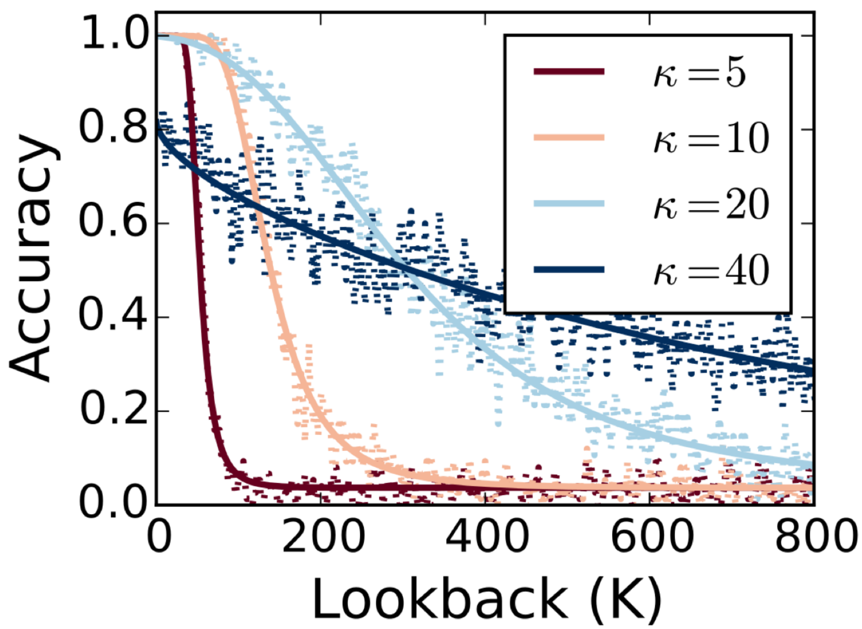
Memory buffers with non-linear neurons

Chapman-Kolmogorov equation:

$$p(\mathbf{x}(m+1)|\mathbf{a}(m)) = \int p(\mathbf{x}(m+1)|\mathbf{x}(m),\mathbf{a}(m))\; p(\mathbf{x}(m))\; d\mathbf{x}(m)$$

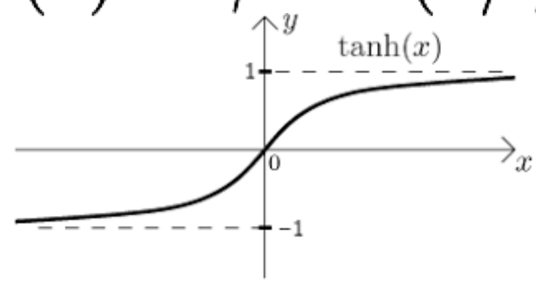
clipped-linear neurons

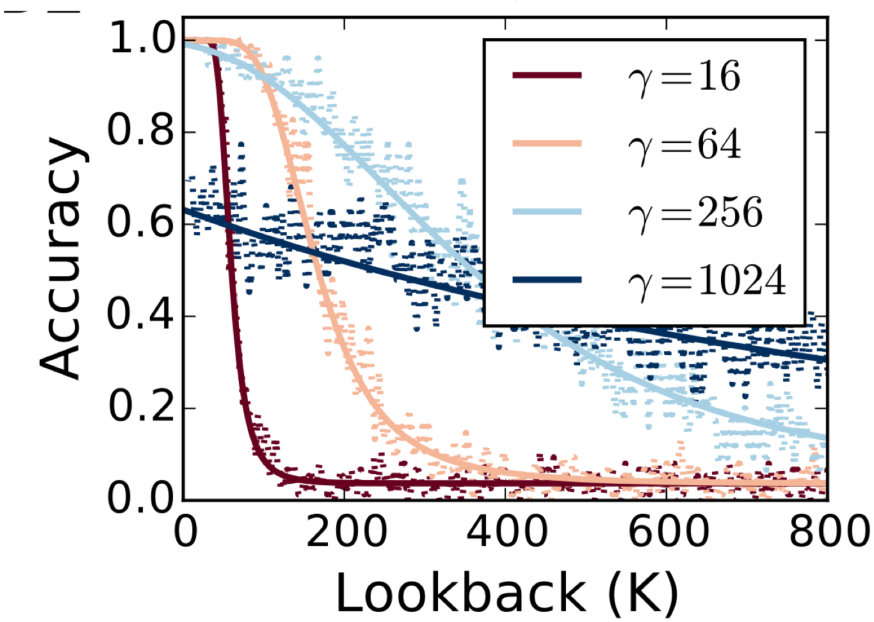




tanh neurons

$$f(x) = \gamma \tanh(x/\gamma)$$





Forgetting time constants

Linear contraction:

$$\tau(\lambda) = -1/\log\lambda$$

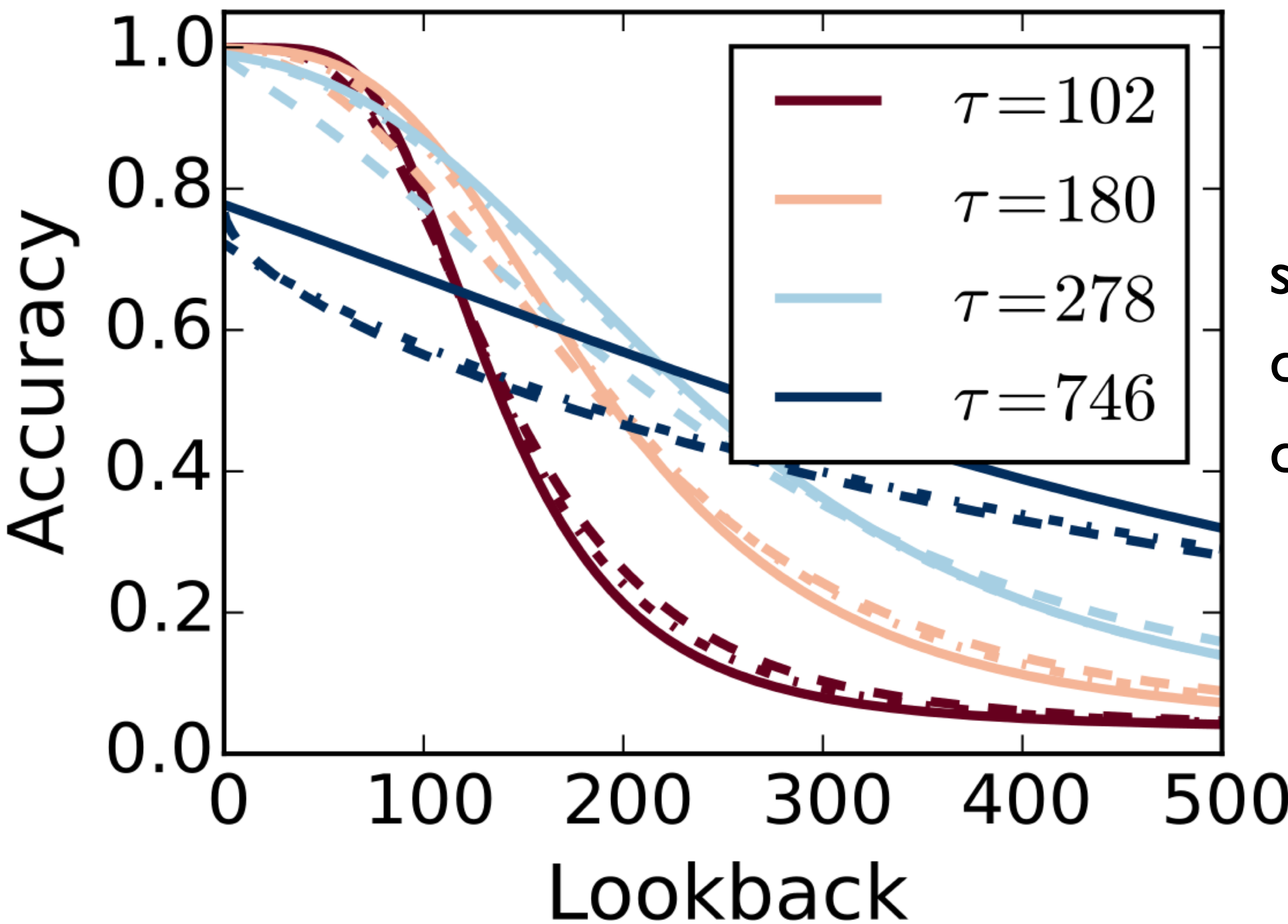
Clipped-linear neurons:

$$\tau(\kappa) = \frac{-2}{\log\left(1 - \frac{3}{\kappa(\kappa+1)}\right)} \approx \frac{2}{3}\kappa^2 \approx \frac{2}{3}\frac{(\kappa^*)^2}{V_{\mathbf{\Phi}}}$$

Tanh neurons:

no analytic expression (numerical estimation)

Comparison of memory buffers



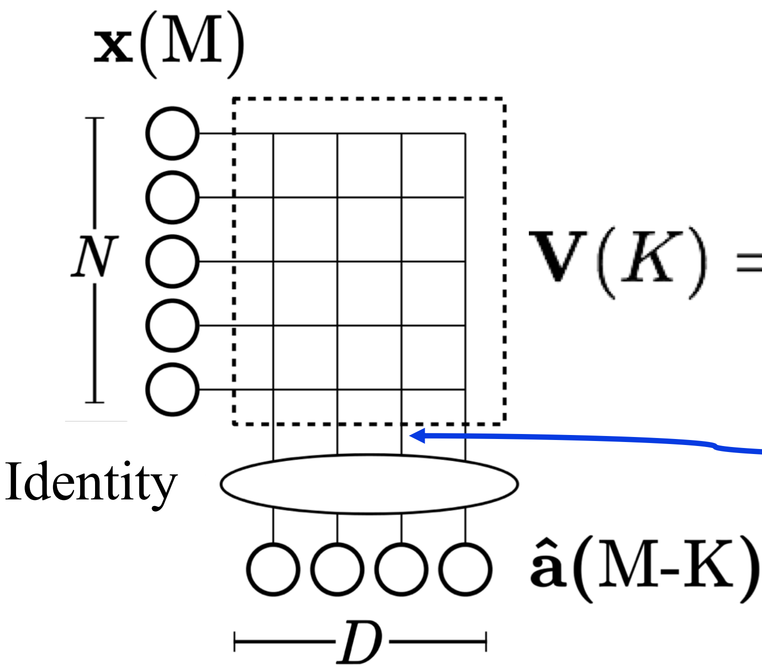
solid: contracting linear

dashes: clipped-linear

dots: tanh

Buffers with different decay mechanisms behave quite similarly!

Readout of sequences with analog numbers



$$\mathbf{V}(K) = c^{-1}\mathbf{W}^K\mathbf{\Phi}$$

$$h_d(K) := \mathbf{V}_d(K)^\top \mathbf{x}(M) \quad (8)$$

$$h_d(K) = \sum_{i=1}^{N} \left(\mathbf{V}_d(K)^\top \mathbf{x}(M)\right)_i = c^{-1}\sum_{i=1}^{N}(\mathbf{\Phi}_d)_i(\mathbf{W}^{-K}\mathbf{x}(M))_i = c^{-1}\sum_{i=1}^{N} z_{d,i} \quad (30)$$

Analysis for continuous Gaussian inputs

$$
\begin{aligned}
z_{d',i} &= (\mathbf{\Phi}_{d'})_i (\mathbf{W}^{-K}\mathbf{x}(M))_i \\
&= (\mathbf{\Phi}_{d'})_i \left[(\mathbf{\Phi}_{d'})_i a_{d'}(M-K)\right] \\
&+ (\mathbf{\Phi}_{d'})_i \left[\sum_{d \neq d'}^{D} (\mathbf{\Phi}_d)_i a_d(M-K) + \sum_{m \neq (M-K)}^{M} \left(\mathbf{W}^{M-K-m}\left(\sum_{d}^{D} \mathbf{\Phi}_d a_d(m)\right)\right)_i\right]
\end{aligned}
\tag{52}
$$

The signal and the noise term are split onto two lines. In the expression $c^{-1}z_{d',i}$, the variance of the signal term is unity, and the resulting SNR is:

$$
\begin{aligned}
r = \frac{\sigma^2(a_{d'})}{\sigma^2(n_{d'})} &= \frac{N\sigma^2(a_{d'})}{\left(\sum_{d\neq d'} a_d^2(M-K) + \sum_{m\neq(M-K)} \sum_d a_d^2(m)\right)} \\
&= \frac{N}{(MD-1)} \approx \frac{N}{MD}
\end{aligned}
\tag{53}
$$

When neuronal noise is present, the SNR becomes:

$$
r = \frac{N}{MD}\left(\frac{1}{1+\sigma_\eta^2/(DV_{\mathbf{\Phi}})}\right) \tag{54}
$$

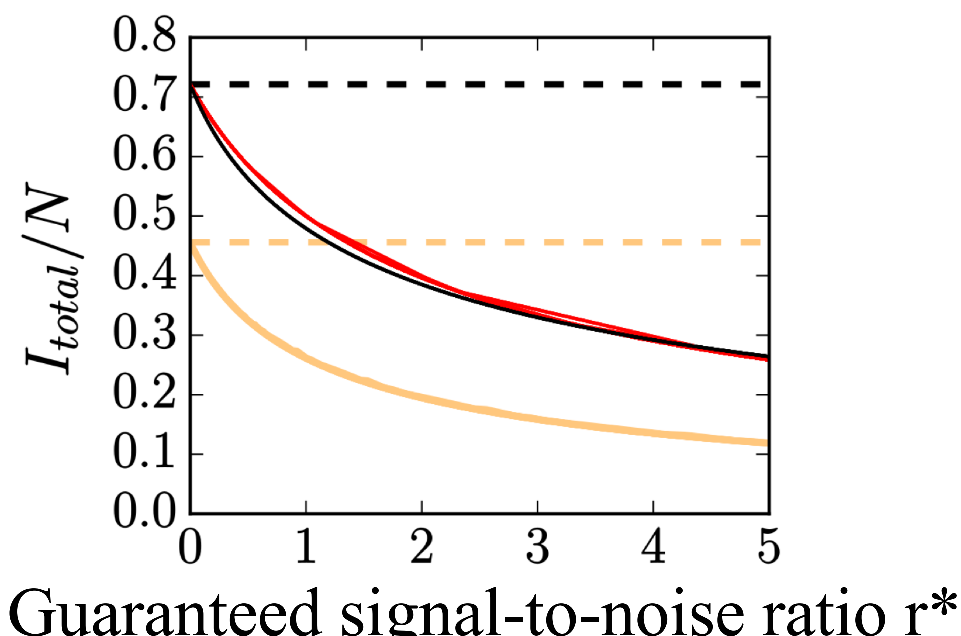
Capacity for continuous input (Gaussian) with standard VSA readout

Signal-to-noise-ratio: $r = \frac{N}{MD}$

Analytic bounds

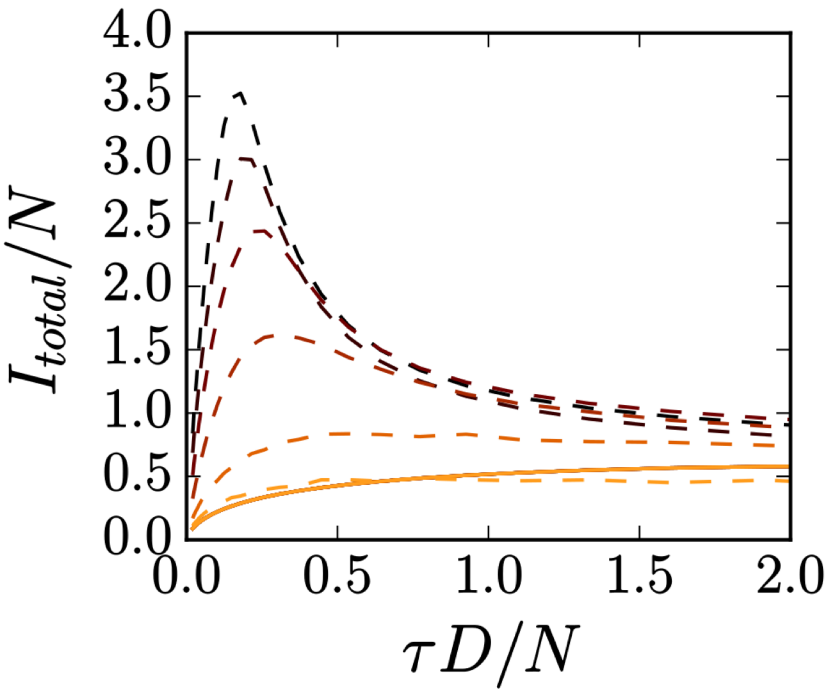
reset memory: $\frac{I_{total}}{N}(r^* \to 0) = \frac{1}{2\ln(2)} = 0.72..$ bits/neuron

buffer (r>r*): $\frac{I_{total}}{N}(r^* \to 0) = \frac{1-e^{-1}}{2\ln(2)} = 0.46..$ bits/neuron

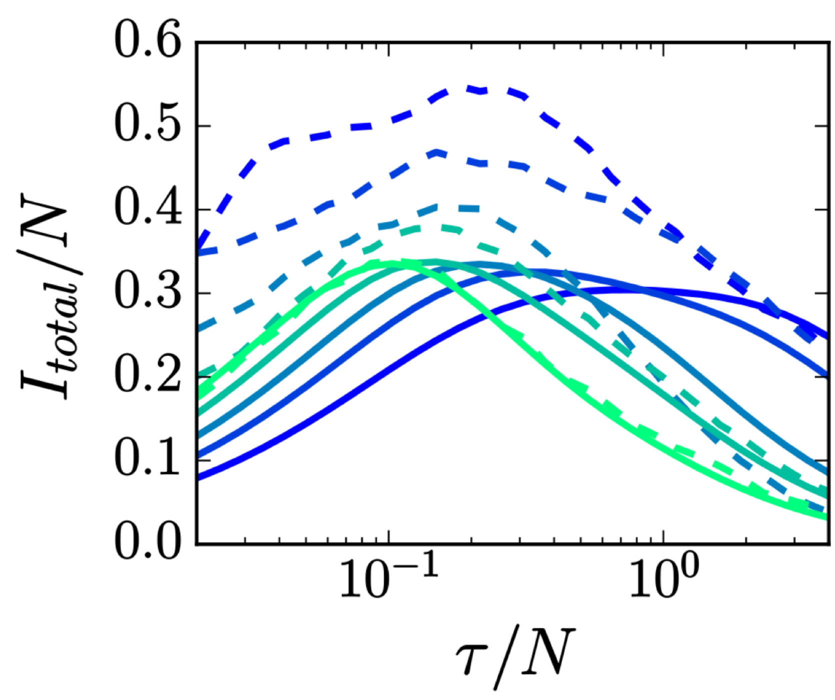


VSA buffer with optimized readout ”a la reservoir computing”

Analog input



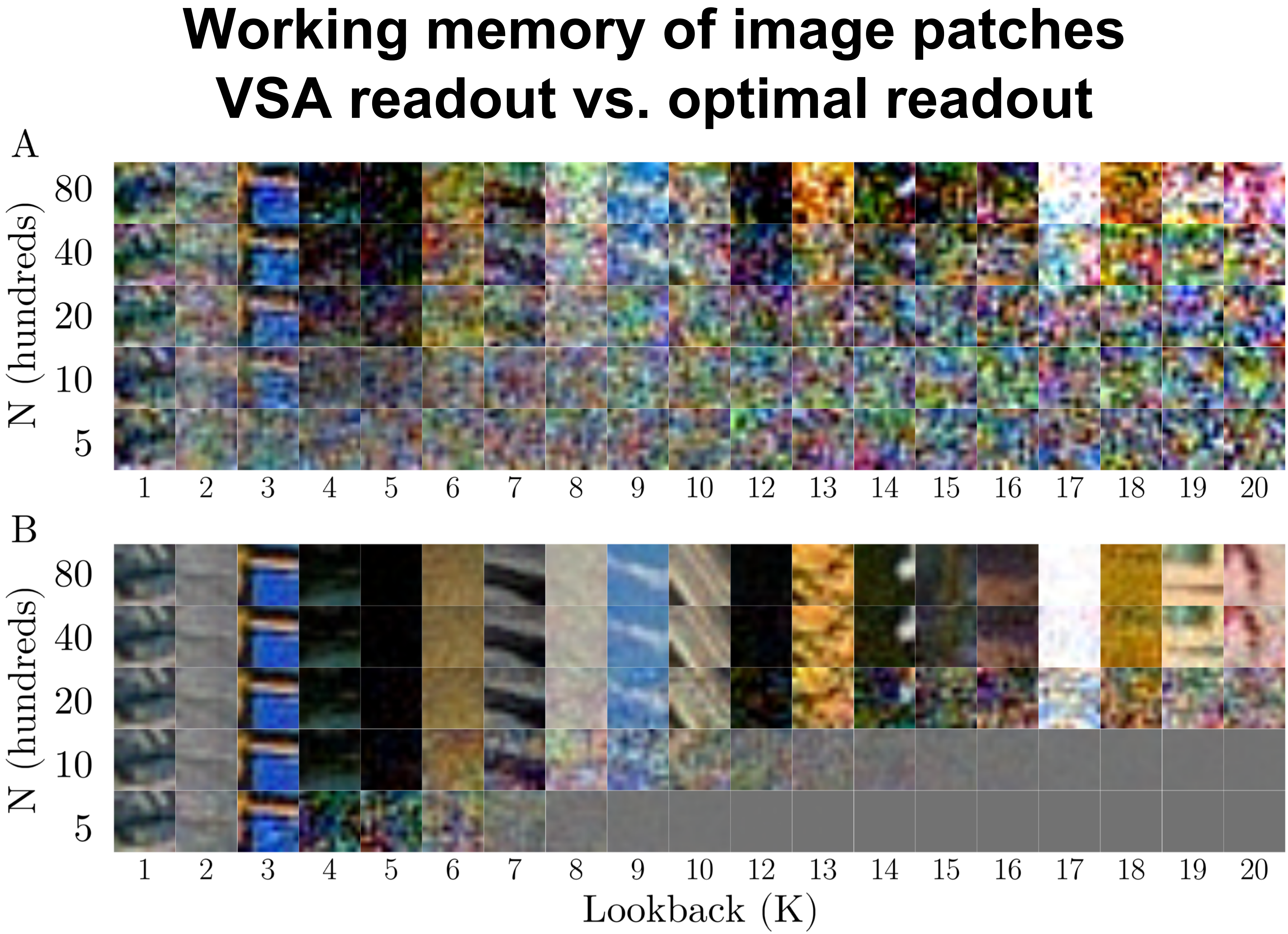
Discrete input



Readout matrix:

$$\tilde{\mathbf{V}}(K) = \tilde{\mathbf{C}}^{-1}\lambda^K \mathbf{W}^K \mathbf{\Phi}$$

$$\tilde{\mathbf{C}} = \frac{\sigma_\eta^2}{1-\lambda^2}\mathbf{I} + \sum_{k=1}^{N} \frac{\lambda^{2k}}{1-\lambda^{2N}} \mathbf{W}^k \mathbf{\Phi}\mathbf{\Phi}^\top \mathbf{W}^{-k}$$



Working memory of image patches
VSA readout vs. optimal readout
A
B
N (hundreds)
80
40
20
10
5
1
2
3
4
5
6
7
8
9
10
12
13
14
15
16
17
18
19
20
Lookback (K)

Lessons for VSAs

- Previous theories underestimate capacity of sequence representations
- Theory valid for VSA representation of data structures other than sequences
- Capacity for superposition is universal across different VSA models: M proportional N
- For sequences of analog vectors, VSA readout is noisy (recoding with VFA principles might help)
- Memory buffers are interesting new concept for VSA, not much explored so far.
- Reservoir computing just a first example how VSA can help dissect opaque neural networks (see new paper on predicting deep nets with VSA)

Lessons for Reservoir Computing

- Reservoir network with pseudo-random input weights and orthogonal W can be dissected into VSA operations: binding with time stamp and superposition

- MMSE readout has higher capacity than VSA method:

Capacity ($\frac{bits}{neuron}$)	VSA		opt. readout	
	reset	buffer	reset	buffer
symbolic	≈ 0.5	≈ 0.3	1	1
analog	0.72	0.46	∞	∞

(Bounds for vanishing intrinsic noise)

- different forgetting mechanisms behave quite similar