Weaving together statistical mechanics, machine learning, and neuroscience

Surya Ganguli

Dept. of Applied Physics,
Neurobiology,
and Electrical Engineering

Stanford University

Funding:

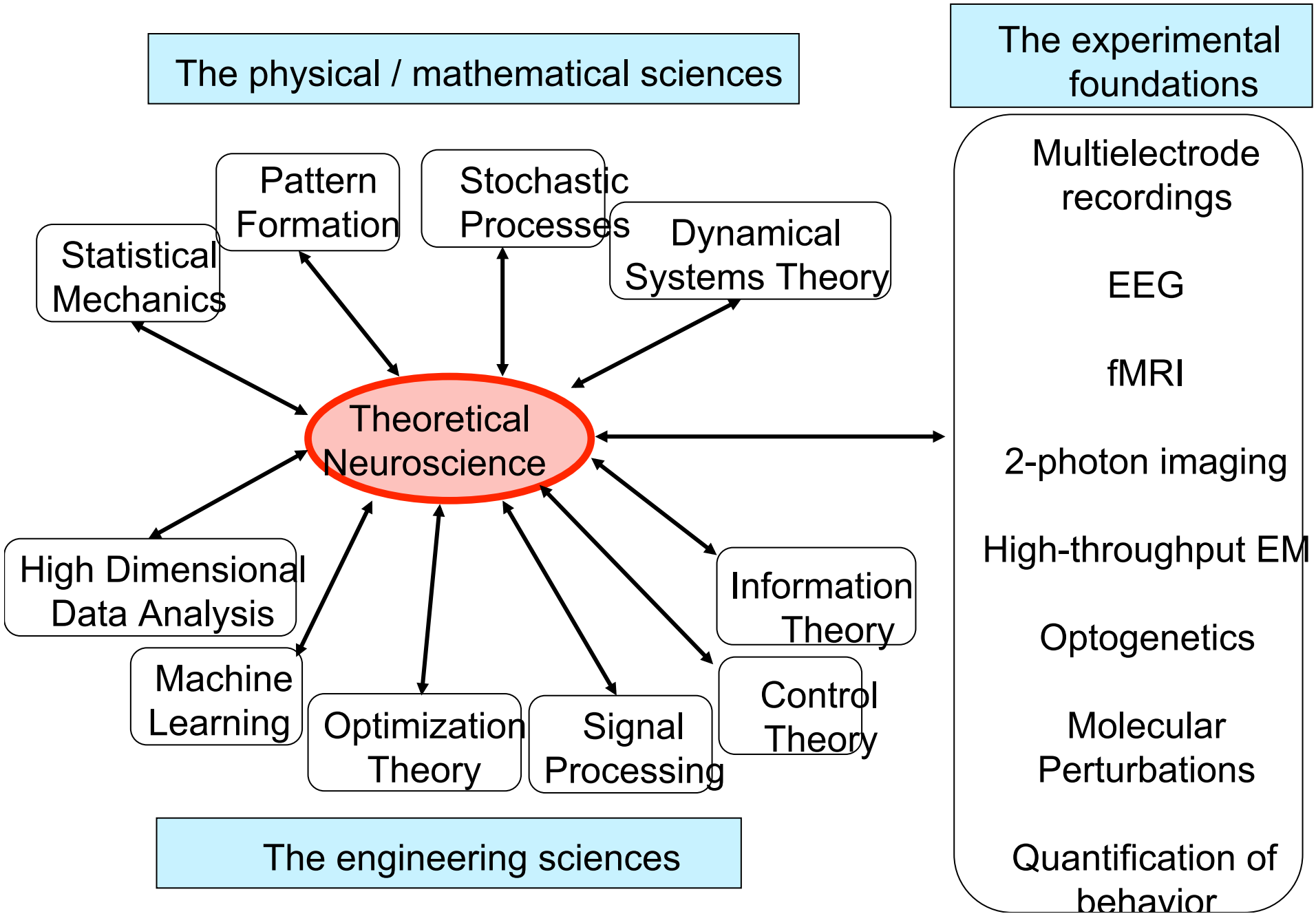
Bio-X Neuroventures
Burroughs Wellcome
Genentech Foundation
James S. McDonnell Foundation
McKnight Foundation
National Science Foundation

NIH
Office of Naval Research
Simons Foundation
Sloan Foundation
Swartz Foundation
Stanford Terman Award

http://ganguli-gang.stanford.edu

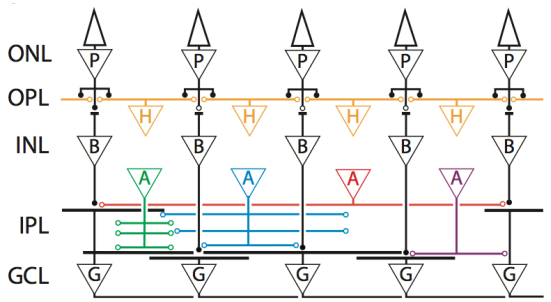
Twitter: @SuryaGanguli

Theoretical neuroscience in the disciplinary landscape

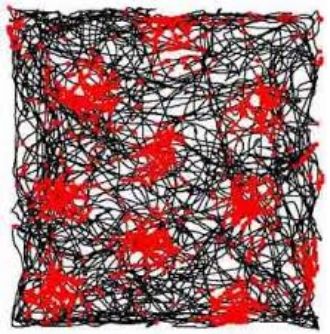


Neural circuits and behavior: theory, computation and experiment

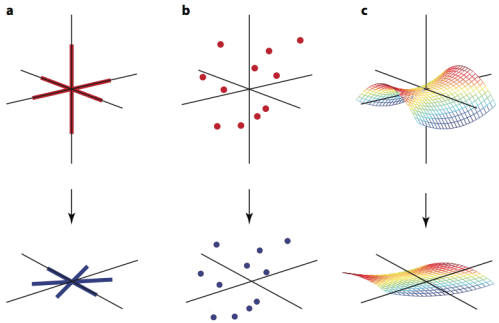
with Baccus lab: inferring hidden circuits in the retina
w/ Niru Maheswaranathan and Lane McIntosh



with Clandinin lab: unraveling the computations underlying fly motion vision from whole brain optical imaging
w/ Jonathan Leong, Ben Poole and Jennifer Esch

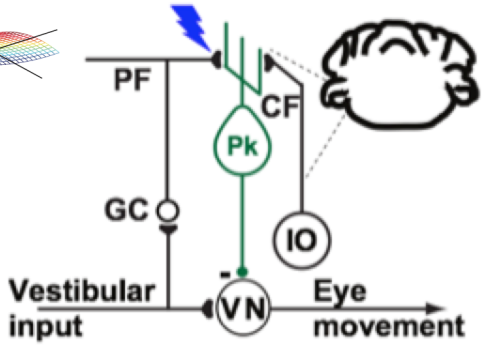


with the Giocomo lab: understanding the internal representations of space in the mouse entorhinal cortex
w/ Kiah Hardcastle and Sam Ocko



with the Shenoy lab: a theory of neural dimensionality, dynamics and measureme
w/ Peiran Gao, Eric Trautmann

with the Raymond lab: theories of how enhanced plasticity can either enhance or impair learning depending on experience
w/ Subhaniel Lahiri, Barbara Vu, Grace Zhao



A palpable technology driven excitement in neuroscience

Measuring Dynamics	Measuring Connectivity	Circuit Perturbations
Multielectrode Recordings	High throughput EM	Optogenetics
Calcium imaging	Viral tracing	TMS
Novel voltage sensors	RNA barcodes	Genetic Knockouts
EEG / fMRI		Crispr / CAS
Quantification of Behavior		

Question: How do we go from this explosion of data to a conceptual understanding?

Issue 1: We will not soon record from the entire brain at single cell single spike-time resolution during any behavior.

Issue 2: Even if we could, what would we do with all the data?

Broader theoretical challenges in neuroscience

Issue 1: Discovering structure with limited data.

High dimensional statistics

How can we understand neural circuits such a high subsampled measurement regime?

Deep Learning

Even if we had all the data, what would we do with it? Lets look to our colleagues in computer science for interesting challenges.

Issue 2: Examples of complete neural circuit models

At the core of every data analysis algorithm lies an implied hypothesis about underlying simplicity in data

Assumed simplicity	Associated algorithm
Low dimensional manifolds	Dimensionality reduction
Independence	ICA
Clusters	Various clustering algorithms
No strong loops	Message passing
Sparsity	Compressed sensing
Low rank matrix structure	Nuclear norm minimization

If any of these simplicities exist in a system, then we can often accurately characterize the structure/function of that system using many fewer measurements than the total dimensionality of the system!

Motivations for an alliance between theoretical neuroscience and theoretical machine learning

- What does it mean to understand the brain (or a neural circuit?)
- We understand how the connectivity and dynamics of a neural circuit gives rise to behavior.
- And also how neural activity and synaptic learning rules conspire to self-organize useful connectivity that subserves behavior.
- The field of machine learning has generated a plethora of learned neural networks that accomplish interesting functions.
- We know their connectivity, dynamics, learning rule, and developmental experience, *yet*, we do not have a meaningful understanding of how they learn and work!

On simplicity and complexity in the brave new world of large scale neuroscience, Peiran Gao and S. Ganguli, Curr. Op. in Neurobiology, 2015.

Outline of talks I, II and III

1. Oldies but goodies

    1. Models of single neurons: Hodgkin Huxley to Hopfield
    2. The Hopfield model
    3. The perceptron learning algorithm: memorization and generalization
    4. Unsupervised learning: PCA, ICA, Sparse Coding

2. High dimensional statistics: theory and experiment

    1. The best way to do regression in high dimensions (Replica theory)
    2. Recovering neural state space dynamics (Rand proj / Matrices / Free prob)
    3. Figuring out how neural circuits learn (Tensor decompositions)

3. Deep learning: theory and practice

    1. Speeding up deep learning (Dynamic criticality)
    2. Error landscape of deep networks (Stat mech of random Gaussian fields)
    3. Deep generative models (Non-equilbrium thermodynamics)
    4. Expressive power of deep networks (Riemannian geometry and chaos theory)
    5. Application: deep models of the retina: the first step in seeing

Statistical mechanics of complex neural systems and high dimensional data

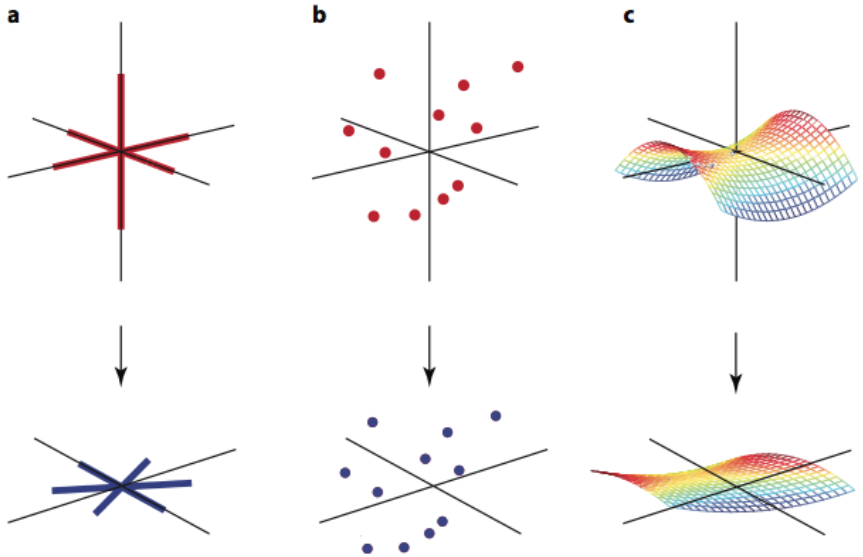
Madhu Advani, Subhaneil Lahiri and Surya Ganguli

Hide affiliations

msadvani@stanford.edu sulahiri@stanford.edu sganguli@stanford.edu

Department of Applied Physics, Stanford University, Stanford, CA, USA

Madhu Advani *et al J. Stat. Mech.* (2013) P03014. doi:10.1088/1742-5468/2013/03/P03014
Received 9 October 2012, accepted for publication 14 January 2013. Published 12 March 20



Canonical models in theoretical neuroscience

Model Neurons: From Hodgkin Huxley to Hopfield, Abbott and Kepler 1990

Neural networks and physical systems with emergent collective computational Abilities, Hopfield, PNAS 1984.

Statistical mechanics of neural networks near saturation, Amit, Gutfreund, Sompolinsky, Annals of Physics, 1987.

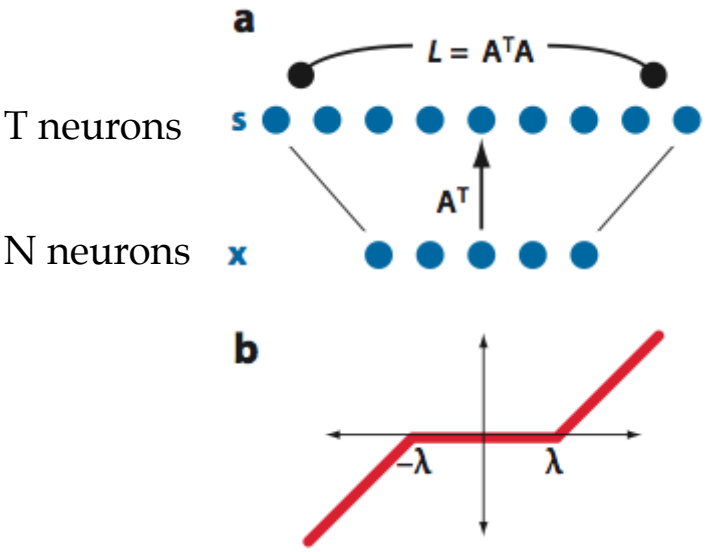
The space of interactions in neural network models,
J. Phys. A: Math. Gen, Gardner E 1988

Expansion: Neural implementation of L1 Minimization

Goal: solve optimization problem:

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} \left\{ \|\mathbf{x} - \mathbf{A}\mathbf{s}\|^2 + \lambda \sum_{i=1}^{T} V(s_i) \right\}$$

Neural circuit solution:



$$\tau \frac{dv_i}{dt} = -v_i + \mathbf{a}^i \cdot \mathbf{x} - \sum_{j=1}^{T} L_{ij} s_j$$

$\mathbf{a}^i$ = i'th column of N by T matrix A
= i'th dictionary element of rep of x
= approximate RF of i'th layer 2 neuron

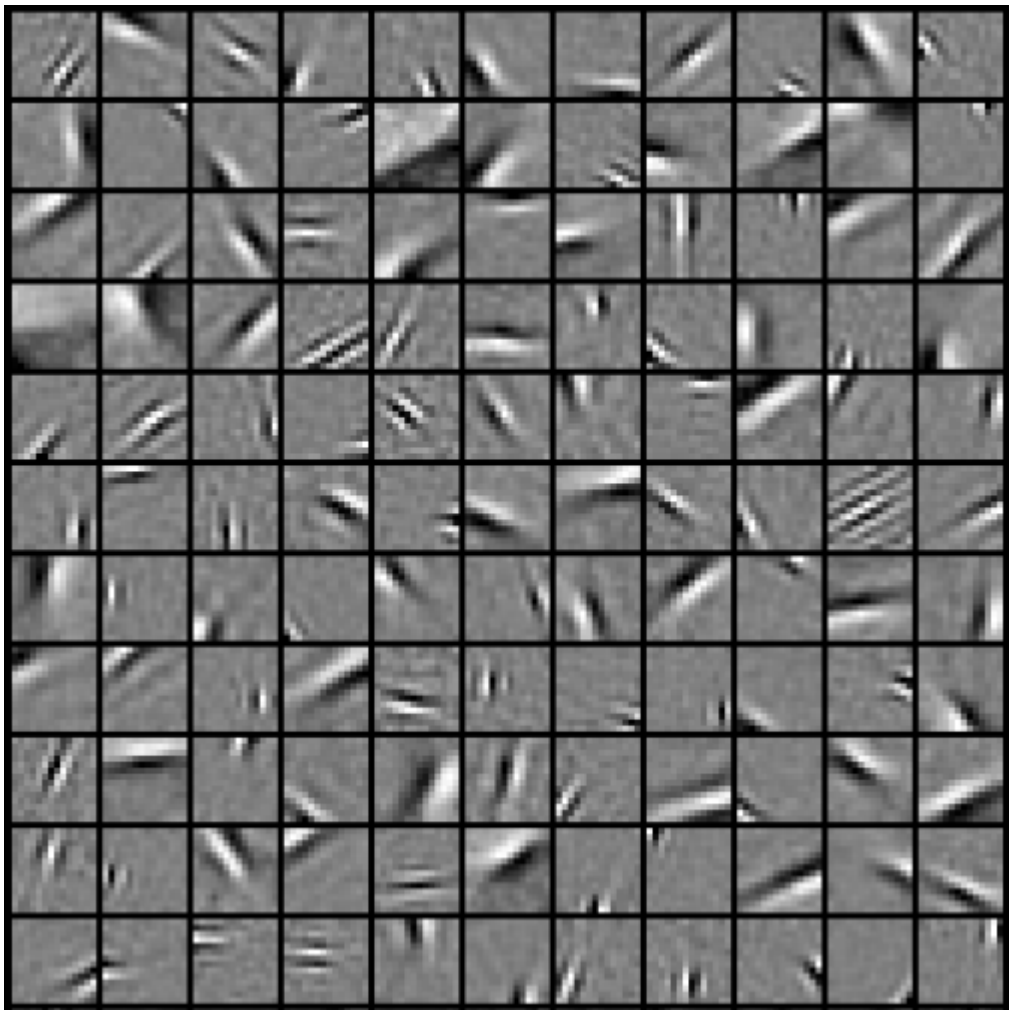
$L_{ij} = \mathbf{a}^i \cdot \mathbf{a}^j$ = lateral inhibition

$s_i = F(v_i)$ = single neuron nonlinearity

For each choice of sparsity penalty V, there is a neuronal nonlinearity F such that neural circuit dynamics ~ gradient descent solution of optimization

Rozell C, et. al. 2008. Sparse coding via thresholding and local competitionin neural circuits. Neural Comp 2010

Expansion: Neural implementation of L1 Minimization



Olshausen and Field, Nature 1996

Outline of talks I, II and III

1. Oldies but goodies

    1. Models of single neurons: Hodgkin Huxley to Hopfield
    2. The Hopfield model
    3. The perceptron learning algorithm: memorization and generalization
    4. Unsupervised learning: PCA, ICA, Sparse Coding

2. High dimensional statistics: theory and experiment

    1. The best way to do regression in high dimensions (Replica theory)
    2. Recovering neural state space dynamics (Rand proj / Matrices / Free prob)
    3. Figuring out how neural circuits learn (Tensor decompositions)

3. Deep learning: theory and practice

    1. Speeding up deep learning (Dynamic criticality)
    2. Error landscape of deep networks (Stat mech of random Gaussian fields)
    3. Deep generative models (Non-equilbrium thermodynamics)
    4. Expressive power of deep networks (Riemannian geometry and chaos theory)
    5. Application: deep models of the retina: the first step in seeing

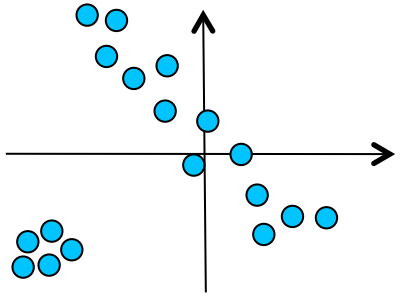
Finding optimal algorithms for nonlinear regressions in high dimensional data analysis

Madhu Advani: Stanford -> Harvard

A revolution in the way we collect and analyze data: The need to think in high dimensions

P = dimensionality of data N = number of data points α = N / P

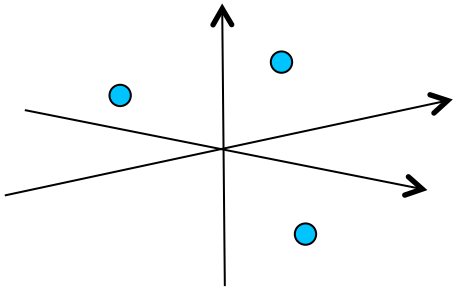
Classical Data



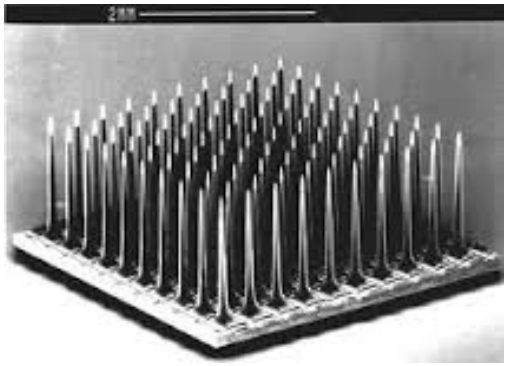
P ~ O(1)
N -> ∞
α -> 0

Modern Data

P -> ∞
N -> ∞
α ~ 0(1)



N = 2 (current and voltage)
M = O(100)

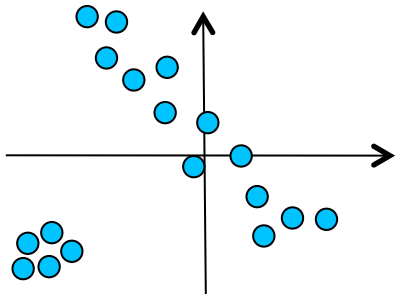


N=O(100) spike rates
M=O(100) trials

Statistical mechanics of high dimensional data analysis

P = dimensionality of data N = number of data points $\alpha = N / P$

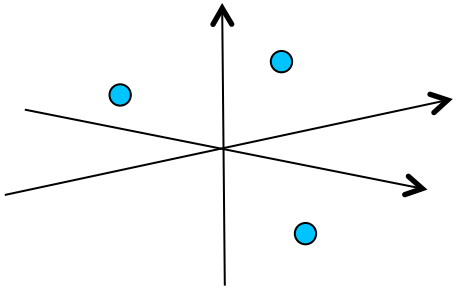
Classical Statistics



$P \sim O(1)$
$N \to \infty$
$\alpha \to 0$

Modern Statistics

$P \to \infty$
$N \to \infty$
$\alpha \sim 0(1)$



Machine Learning and Data Analysis
Learn statistical parameters by maximizing log likelihood of data given parameters.

Statistical Physics of Quenched Disorder
Energy = - log Prob (data | parameters)
Data = quenched disorder
Parameters = thermal degrees of freedom

Statistical mechanics of compressed sensing, S. Ganguli and H. Sompolinsky, PRL 2010.

Short-term memory in neuronal networks through dynamical compressed sensing, NIPS 2010.

Compressed sensing, sparsity and dimensionality in neuronal information processing and data analysis, S. Ganguli and H. Sompolinsky, Annual Reviews of Neuroscience, 2012

Statistical mechanics of optimal convex inference, M. Advani and S. Ganguli, Physical Review X, 2016.

An equivalence between high dimensional Bayes optimal inference and M-estimation, NIPS 2016.

Random projections of random manifolds, S. Lahiri, P. Gao, S. Ganguli, http://arxiv.org/abs/1607.04331, under review at JMLR.

Optimal inference in high dimensions

$$\mathbf{s}^0 \longrightarrow \underbrace{\mathbf{x}_\mu \longrightarrow y_\mu = \mathbf{x}_\mu \cdot \mathbf{s}^0 + \epsilon_\mu}_{\downarrow} \quad \hat{\mathbf{s}}$$

Generative model and measurements

P dim signal $s^0 \sim P_s$
N measurements with noise $\varepsilon \sim P_\varepsilon$
α = N/P = measurement density

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} \sum_\mu \rho(\mathbf{y}_\mu - \mathbf{x}_\mu \cdot \mathbf{s}) + \sum_j \sigma(s_j)$$

Estimation algorithm

ρ = loss function
σ = regularizer
q_s = L_2 estimation error

$$\frac{1}{P}\sum_j (\hat{s}_j - s_j^0)^2 = q_s(\alpha, \rho, \sigma, P_\epsilon, P_s)$$

Example algorithms

Least squares:	$\rho(\epsilon) = \epsilon^2$	$\sigma(s) = 0$
Maximum likelihood:	$\rho(\epsilon) = -\log P_\epsilon(\epsilon)$	$\sigma(s) = 0$
Ridge regression:	$\rho(\epsilon) = \epsilon^2$	$\sigma(s) = s^2$
LASSO:	$\rho(\epsilon) = \epsilon^2$	$\sigma(s) = \lambda_1 \lvert s \rvert$
Elastic Net:	$\rho(\epsilon) = \epsilon^2$	$\sigma(s) = \lambda_1 \lvert s \rvert + \lambda_2 s^2$
MAP:	$\rho(\epsilon) = -\log P_\epsilon(\epsilon)$	$\sigma(s) = -\log P_s(s)$

Optimal inference in high dimensions

$$\mathbf{s}^0 \longrightarrow \underbrace{\mathbf{x}_\mu \longrightarrow y_\mu = \mathbf{x}_\mu \cdot \mathbf{s}^0 + \epsilon_\mu}_{\hat{\mathbf{s}}}$$

Generative model and measurements

P dim signal $s^0 \sim P_s$
N measurements with noise $\varepsilon \sim P_\varepsilon$
α = N/P = measurement density

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} \sum_\mu \rho(\mathbf{y}_\mu - \mathbf{x}_\mu \cdot \mathbf{s}) + \sum_j \sigma(s_j)$$

Estimation algorithm

ρ = loss function
σ = regularizer
q_s = L_2 estimation error

$$\frac{1}{P}\sum_j (\hat{s}_j - s_j^0)^2 = q_s(\alpha, \rho, \sigma, P_\epsilon, P_s)$$

Question:

For a given signal distribution P_s , noise distribution P_ε , and measurement density α,

what is the best loss function ρ and regularizer σ?

Optimal inference in high dimensions

$$E(\mathbf{s}) = \sum_{\mu} \rho\left(y_\mu - \mathbf{x}_\mu \cdot \mathbf{s}\right) + \sum_{j} \sigma(s_j)$$

Statistical physics analogy

$\mathbf{s}$ (parameters)	Thermal degrees of freedom (e.g spins)
$\mathbf{X}, \mathbf{y}$ (data)	Quenched disorder (e.g. connectivity J_{ij})
$\hat{\mathbf{s}}$ (estimates)	Ground state (zero temp) Gibbs distribution: $P_G = \frac{e^{-\beta E(\mathbf{s})}}{Z}$

Fundamental limits on convex inference in high dimensions

Classical inference bound

The Cramer Rao bound for any unbiased estimator

$$q_s \geq \frac{1}{\alpha J[\epsilon]} \qquad J[\epsilon] = \left\langle \left(\log' P_\epsilon(\epsilon)\right)^2 \right\rangle_\epsilon$$

High dim. without prior information

For any ρ:

$$q_s \geq \frac{1}{\alpha J[\epsilon_{q_s}]} \geq \frac{1}{(\alpha-1)J[\epsilon]}$$

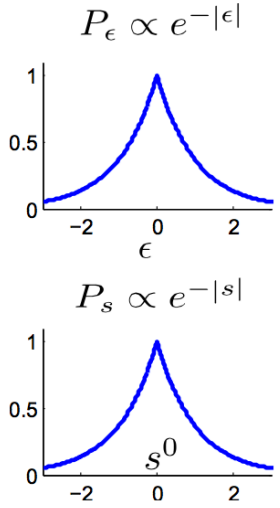
High dim. with prior information

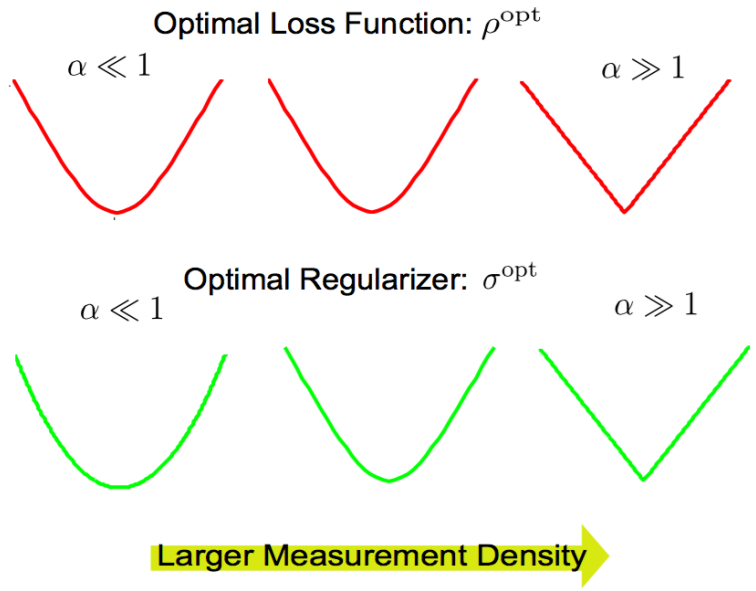
For any convex ρ, σ:

$$q_s \geq \frac{1}{\alpha J[\epsilon_{q_s}] + J[s^0]}$$

Optimal inference in high dimensions

Question: For a given signal distribution P_s , noise distribution P_ε , and measurement density α, what is the best loss function ρ and regularizer σ?





For log-concave signal and noise: the optimal loss and regularizer are nonlinearly smoothed versions of MAP where the smoothing increases as the measurement density decreases.

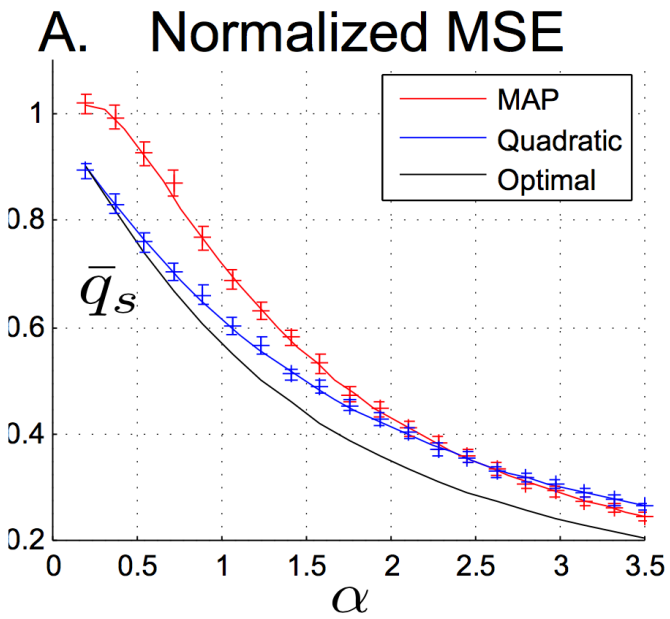
MAP is optimal at high measurement density.

Ridge regression is optimal at low measurement density independent of signal and noise!

No inference algorithm can out-perform our optimal algorithm!

Optimal inference in high dimensions

Question: For a given signal distribution P_s , noise distribution P_ε , and measurement density α, what is the best loss function ρ and regularizer σ?



M. Advani and S. Ganguli, An equivalence between high dimensional Bayes optimal inference and M-estimation, NIPS 2016.

M. Advani and S. Ganguli, Statistical mechanics of optimal convex inference in high dimensions, Physical Review X, 6, 031034, 2016.

Related work by El Karoui, and Montanari.

For log-concave signal and noise: the optimal loss and regularizer are nonlinearly smoothed versions of MAP where the smoothing increases as the measurement density decreases.

MAP is optimal at high measurement density.

Ridge regression is optimal at low measurement density independent of signal and noise!

No inference algorithm can out-perform our optimal algorithm!

Outline of talks I, II and III

1. Oldies but goodies

    1. Models of single neurons: Hodgkin Huxley to Hopfield
    2. The Hopfield model
    3. The perceptron learning algorithm: memorization and generalization
    4. Unsupervised learning: PCA, ICA, Sparse Coding

2. High dimensional statistics: theory and experiment

    1. The best way to do regression in high dimensions (Replica theory)
    2. Recovering neural state space dynamics (Rand proj / Matrices / Free prob)
    3. Figuring out how neural circuits learn (Tensor decompositions)

3. Deep learning: theory and practice

    1. Speeding up deep learning (Dynamic criticality)
    2. Error landscape of deep networks (Stat mech of random Gaussian fields)
    3. Deep generative models (Non-equilbrium thermodynamics)
    4. Expressive power of deep networks (Riemannian geometry and chaos theory)
    5. Application: deep models of the retina: the first step in seeing

A theory of recovering neural state space dynamics

Surya Ganguli

Dept of Applied Physics

And, by courtesy,

Neurobiology
Electrical Engineering

collaboration with Shenoy Lab

Stanford University

Peiran Gao

Eric Trautmann

A major conceptual elephant

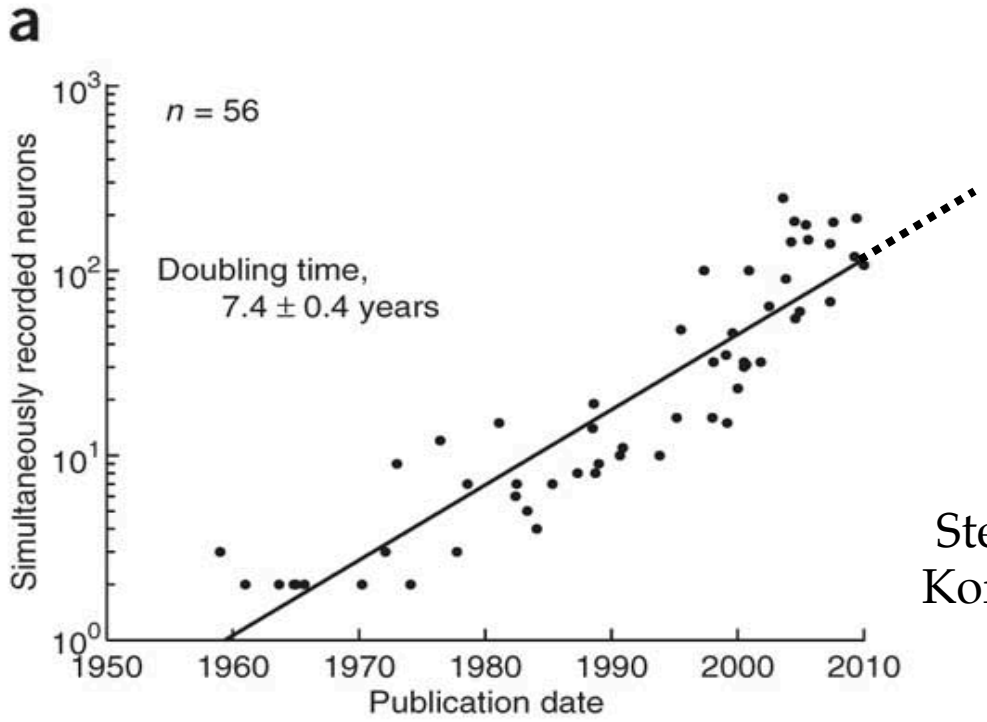
How can we record O(100) neurons in regions deep within the brain and obtain scientifically interpretable results that relate neural activity to behavior and cognition?

This is remarkable, considering these brain regions can contain $O(10^6\text{-}10^9)$ neurons – 5 orders of magnitude more than we record!

How has systems neuroscience been as successful as it has in such an undersampled measurement regime?

Or are we completely misleading ourselves?

An exponential Moore's Law for the number of recorded neurons



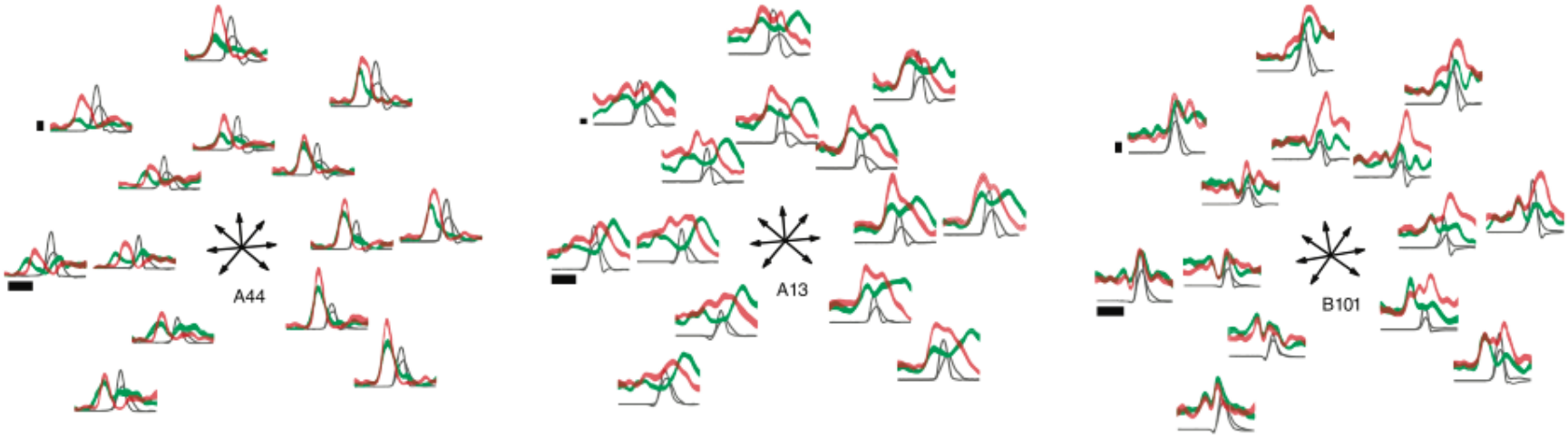
Stevenson & Kording, 2011

Multielectrode recordings allow us to record from 10^2 to 10^3 neurons.
Mammalian circuits controlling complex behaviors contain > 10^6 to 10^9 neurons.

122 years to get 5 orders of magnitude more neurons

We need a theory of neural data analysis that tells us how and when statistical analyses applied to a small subset of neurons reflect the collective dynamics of the much larger, unobserved circuit they are embedded in.

An example dataset: the single neuron view



Churchland and Shenoy, J. Neurophys. 2007

Trial averaged firing rates from 3 neurons while a monkey is reaching to targets
at 7 directions, two lengths and two speeds (red/green)

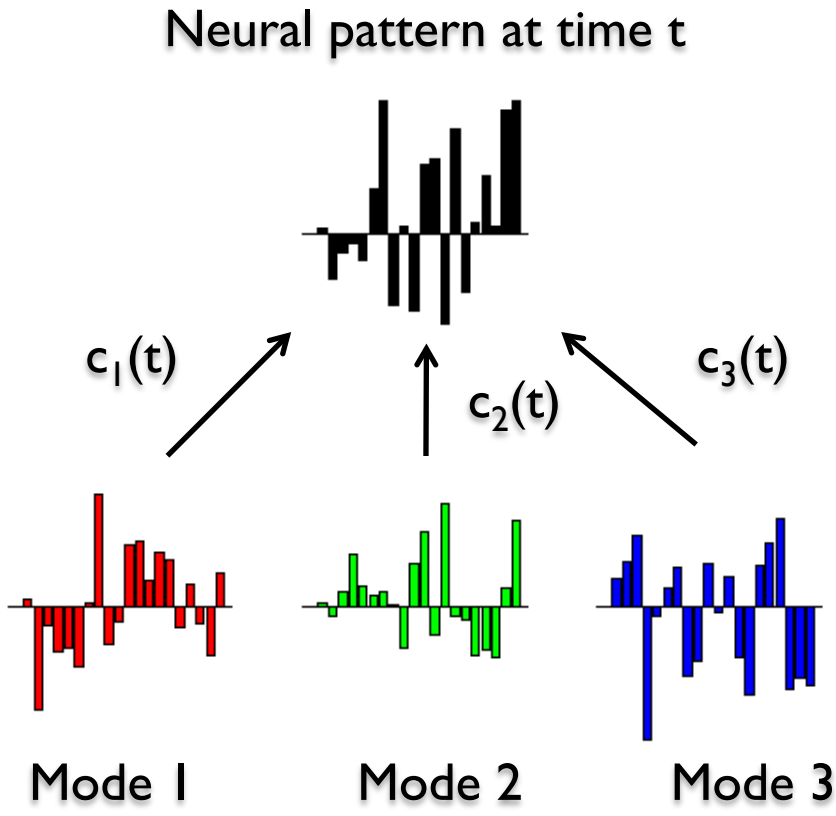
There are about 100 more neurons like these.

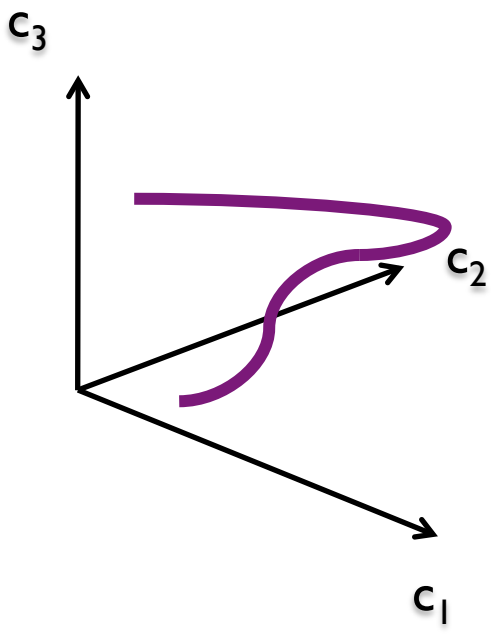
How are such datasets analyzed?

Analyzing neural data with dimensionality reduction

A widespread practice: simultaneously measure the dynamics of N neurons during a task (N ~ 100 to 200)

We often find that all neural activity patterns found during the task can be obtained from a from a small number of basis patterns, or modes

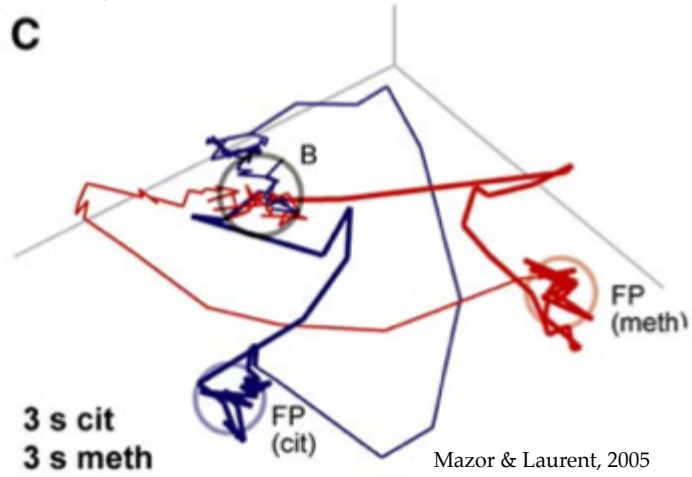




Low dimensional neural trajectory

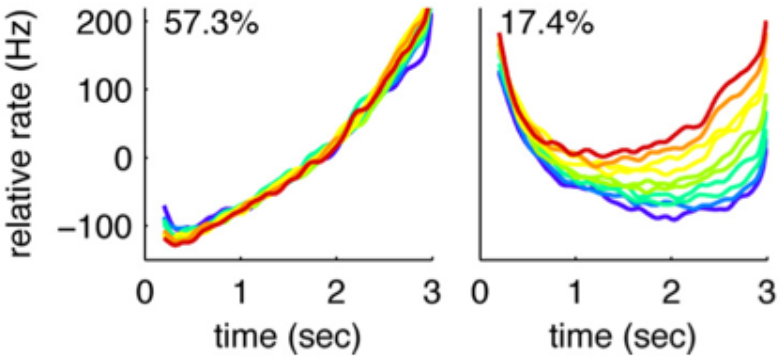
Dynamical portraits of circuit computation via dim reduction

locust, antenna lobe



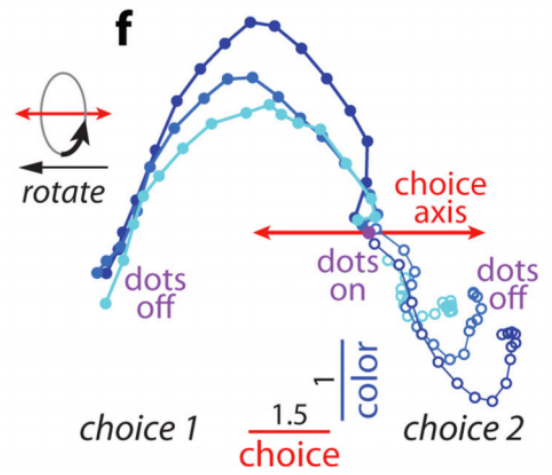
Mazor & Laurent, 2005

monkey, PFC



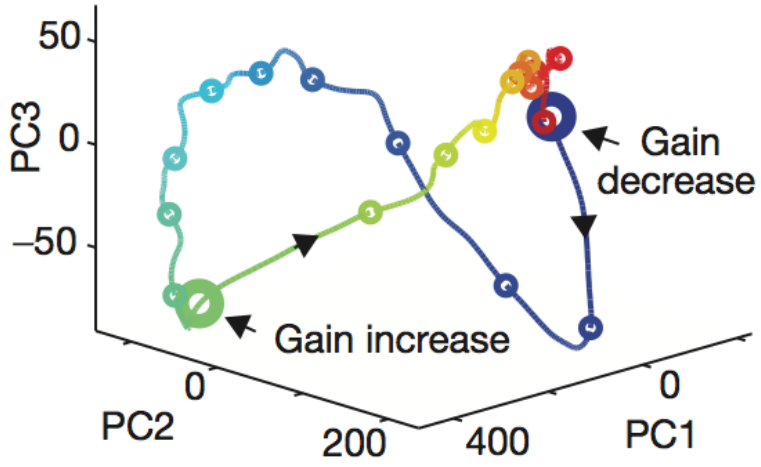
Machens et al., 2010

Monkey pre-frontal cortex



Mante al., 2013

a *zebra fish, whole brain*



Ahrens et al., 2012

Fundamental conceptual questions

Can we trust such dynamical portraits of circuit computation despite recording so few neurons?

How would the shape of these portraits change if we recorded more neurons?

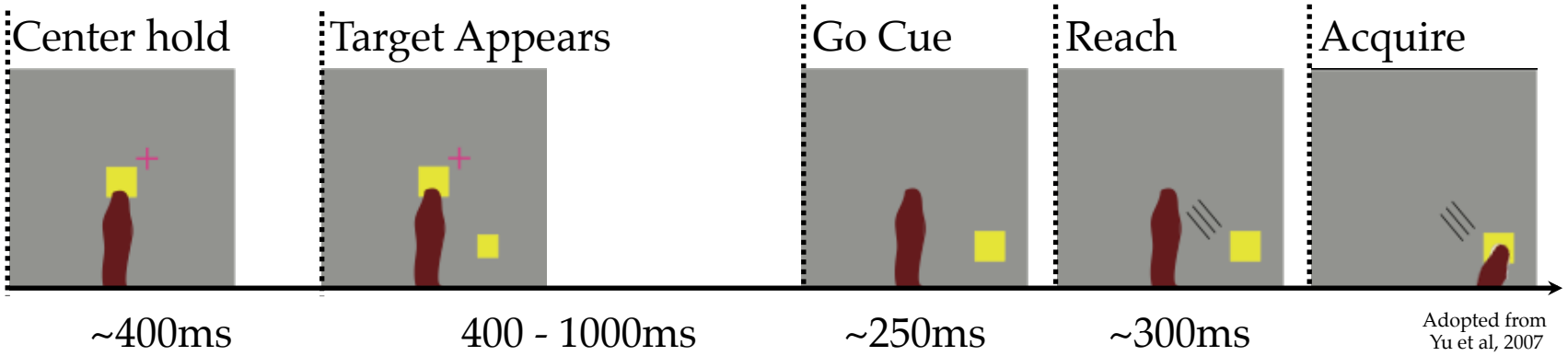
Would their dimensionality increase if we recorded more neurons?

What (if anything) can we learn about large dynamical networks at such an overwhelming level of under sampling?

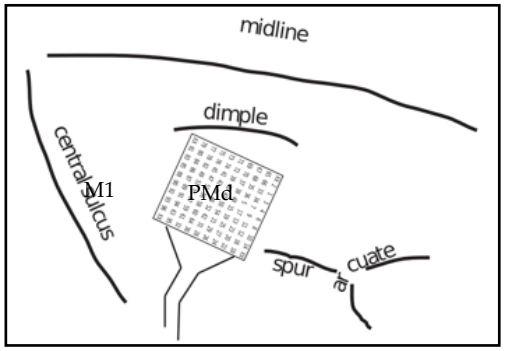
Can we obtain a predictive theory of experimental design that can tell us how many more neurons we should record?

How should this number depend on the properties of neural activity and the behavioral task?

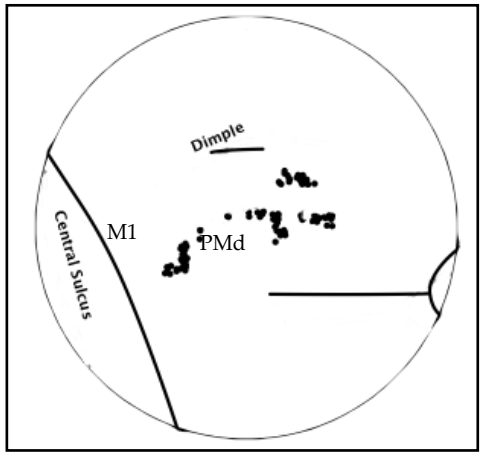
Example dataset:



Extra-cellular recordings from PMd and M1:



Yu et al. 2007



Churchland et al. 2007

Dataset 1 (Monkey H)
8 directions
8 task conditions
multi-electrode array, 109 single units

Dataset 2 (Monkey A)
7 directions, 2 speeds and 2 distances
28 task conditions
single electrode, 64 preparatory recordings

Dimensionality in motor cortex

In primate motor cortex there are
O(100 million) neurons controlling O(650) skeletal muscles.

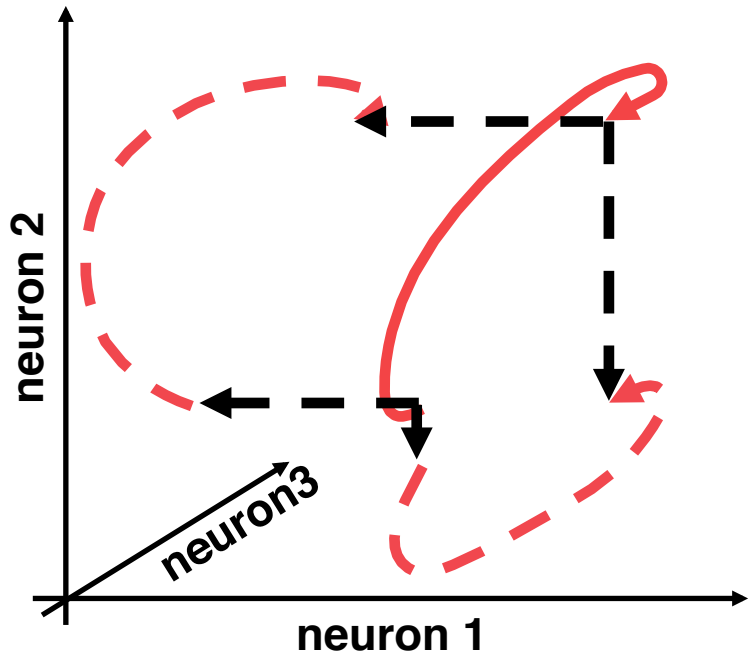
In these experiments, O(100) neurons were recorded.

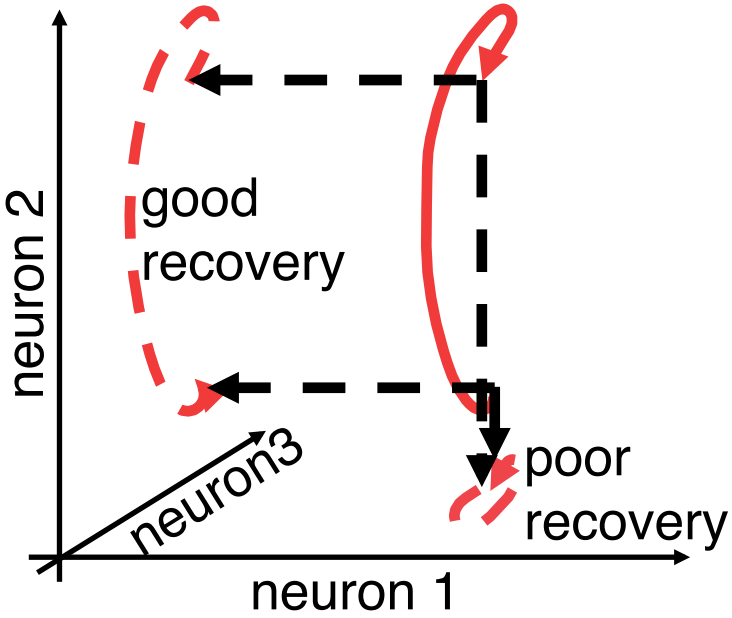
The PCA dimensionality (~ 70% variance explained) across all 8 reaches is 7.

The PCA dimensionality (~ 70% variance explained) for one reach is 3.3.

Measuring the Dynamic Portrait under Sub-sampling

When are portraits from relatively few neurons = those from all neurons?



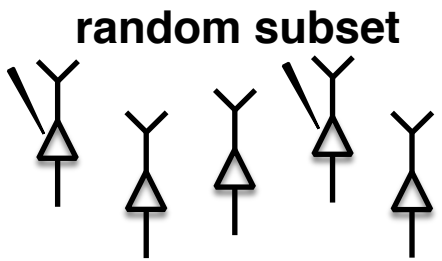


When patterns of neural activity are distributed across neurons, we can accurately recover dynamic portraits despite subsampling

The act of neuronal measurement as a random projection

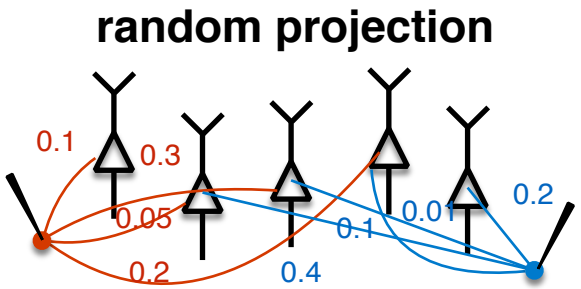
If neural manifold is randomly oriented:

An experiment we can do: measure a random subset of M neurons

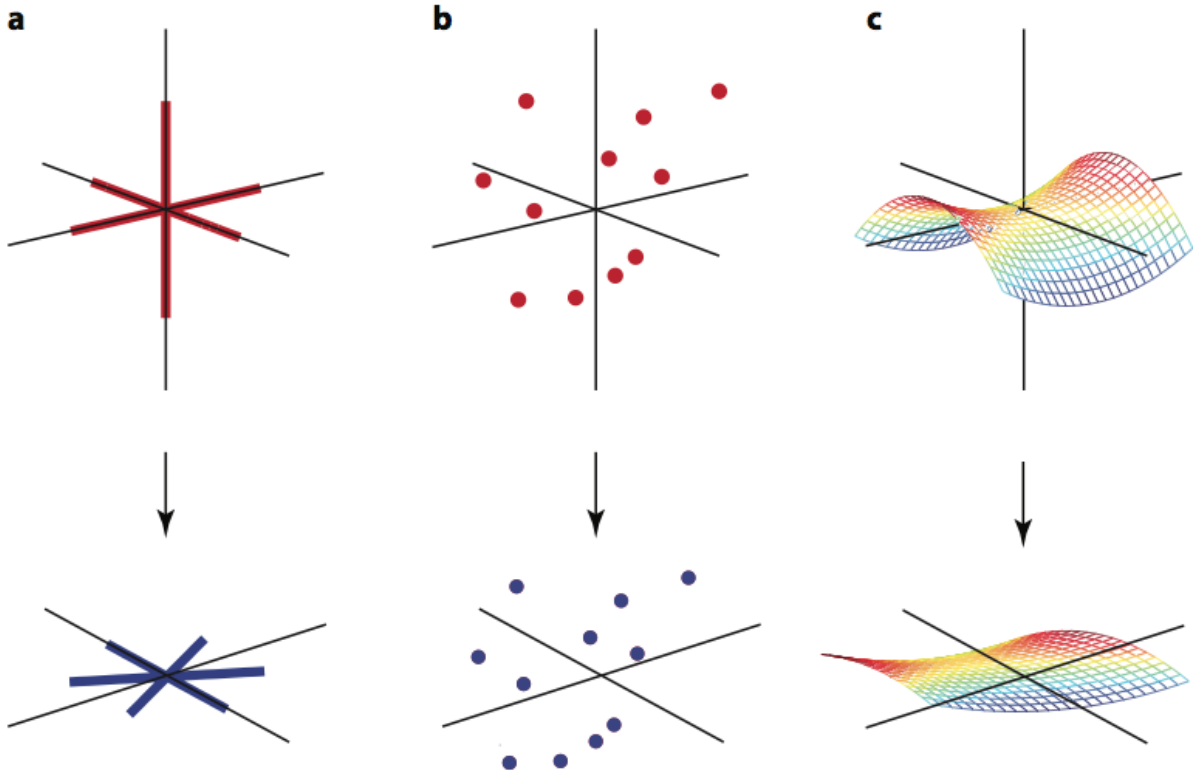


is equivalent to

An experiment we cannot yet do: measure M random linear combinations (i.e. random projections) of *all* neurons



A larger context: random projections



$\mathbf{x} = \mathbf{A}\mathbf{s}$ is a random projection from a N dim space down to an M dim space

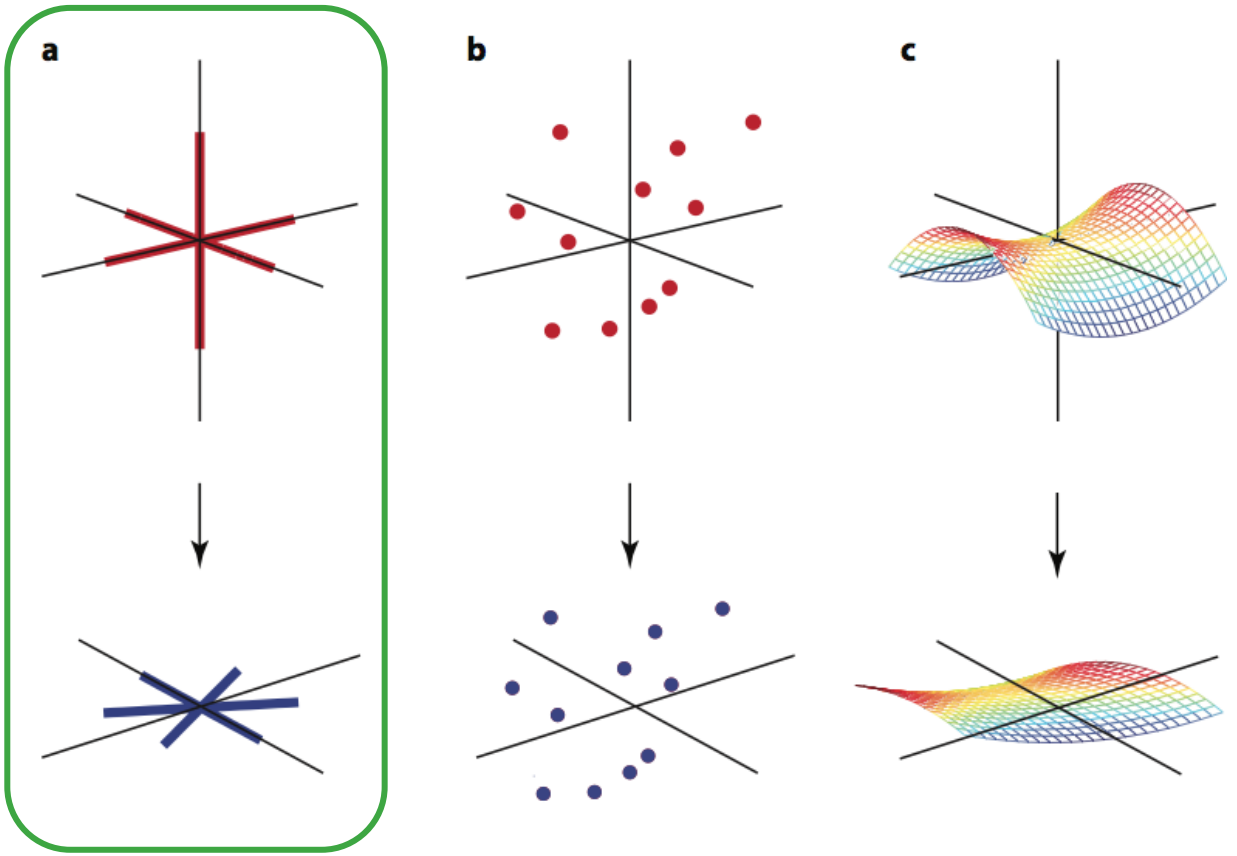
Data / interesting signals live on a K-dim submanifold in N-dim space

When will the geometry of this manifold be preserved under a random proj. ?

Distortion: $D_{ab} = (\| \mathbf{A}\mathbf{s}^a - \mathbf{A}\mathbf{s}^b \|^2 - \| \mathbf{s}^a - \mathbf{s}^b \|^2) / \| \mathbf{s}^a - \mathbf{s}^b \|^2$

A larger context: random projections

K-dim manifold
N-dim space



Random proj
To M-dim space

Manifold of K-sparse signals = Union of N choose K K-dim hyperplanes

As long as $M > O(1/\varepsilon^2 * K \log N/K)$, then $\max_{ab} |D_{ab}| = O(\varepsilon)$ with high prob ov
random choice of projection **A**

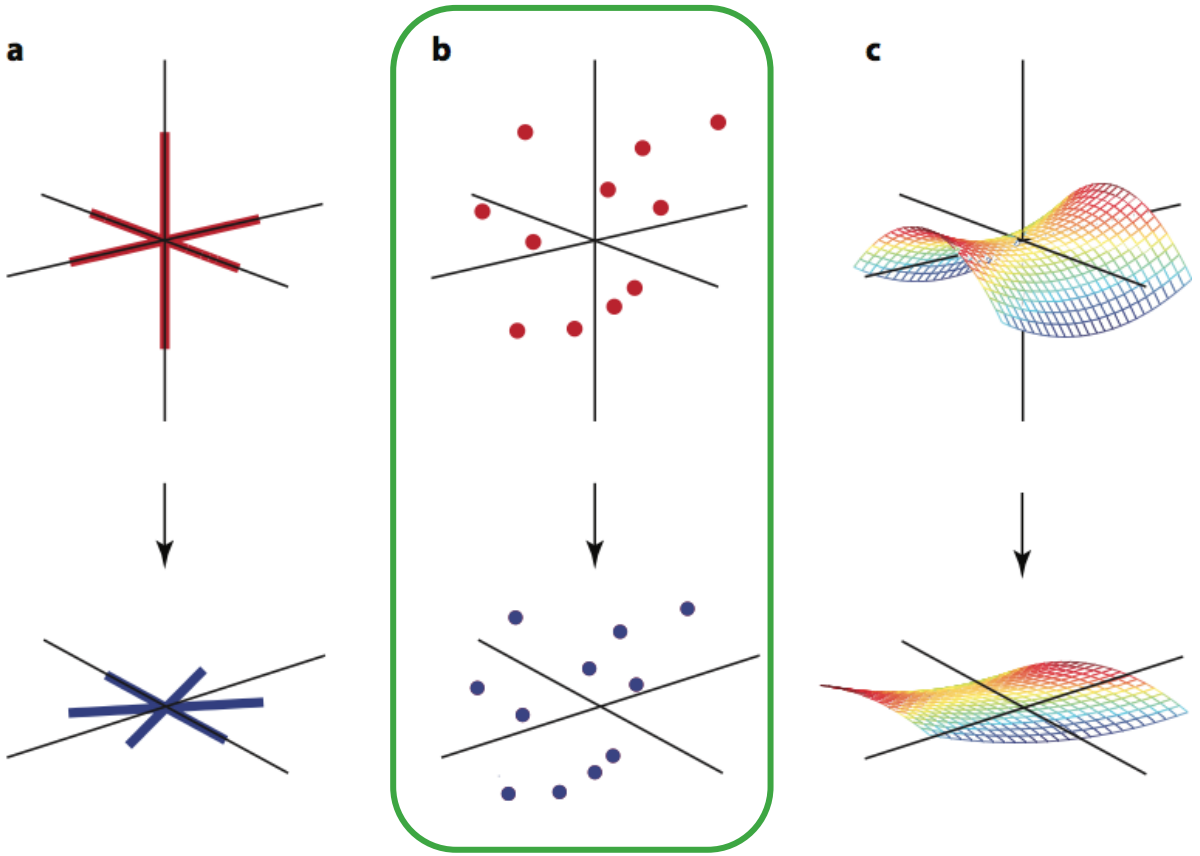
Baraniuk et. al. 2008

Deterministic result: for any projection **A** with small distortion, one can reconstruct sparse signal from its projection (i.e. compute its pre-image)

Tao, Candes

A larger context: random projections

P points in
N-dim space



Random proj
To M-dim space

Point cloud = Union of P points in N-dim space

As long as $M > O(1/\varepsilon^2 * \log P)$, then $\max_{ab} |D_{ab}| = O(\varepsilon)$ with high prob over random choice of projection **A**

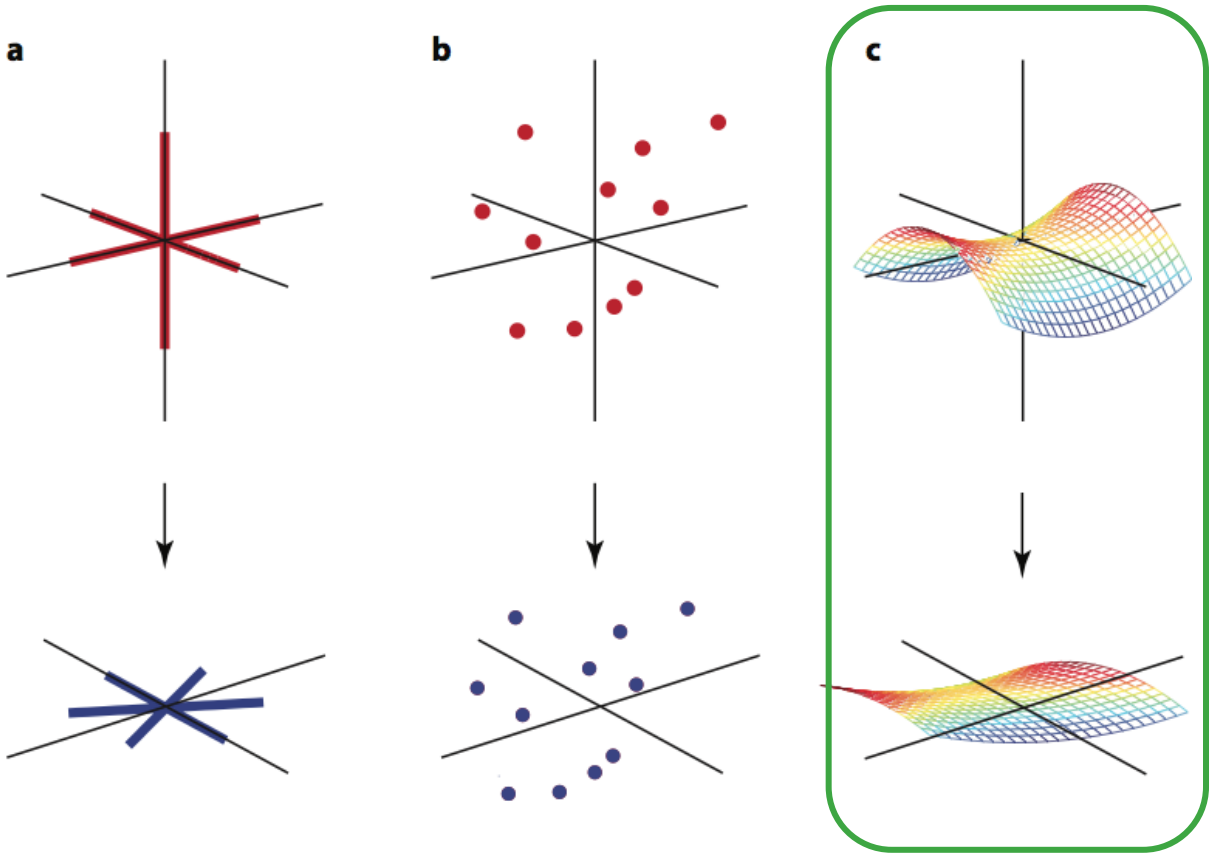
Johnson-Lindenstrauss Lemm

Compressed computation: with so few measurements, one cannot recover high-dim points, but any algorithm which depends on pairwise distances can be applied in low dim space

A larger context: random projections

(K-dim) manifold
N-dim space

Random proj
To M-dim space



Arbitrary K-dim manifold in N dim space

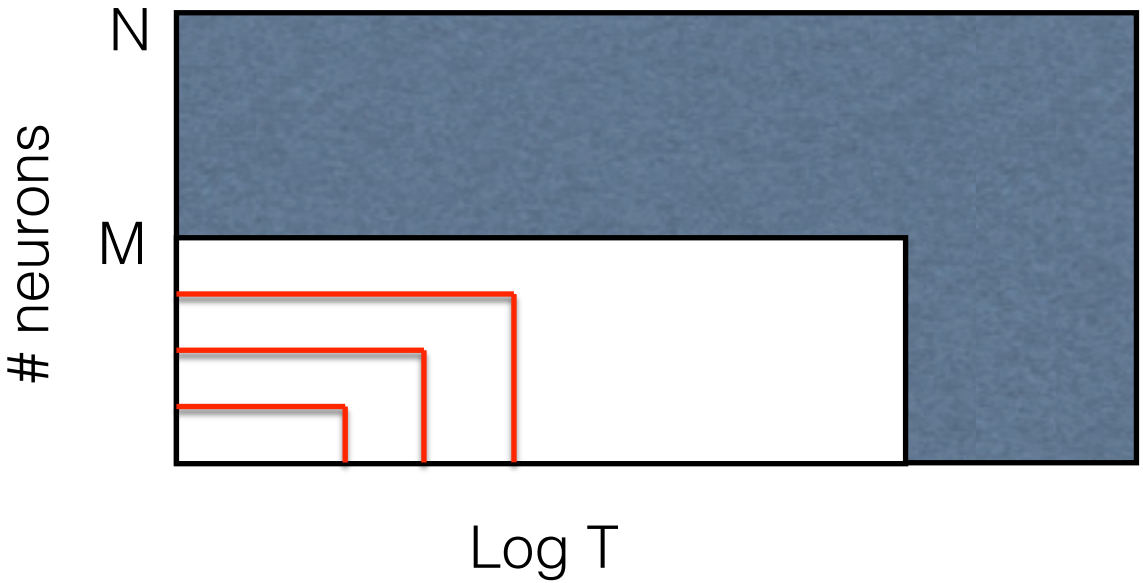
As long as $M > O(1/\varepsilon^2 * K \log [C*Vol])$, where C is related to curvature, then $\max_{ab} |D_{ab}| = O(\varepsilon)$ with high prob over random choice of projection **A**

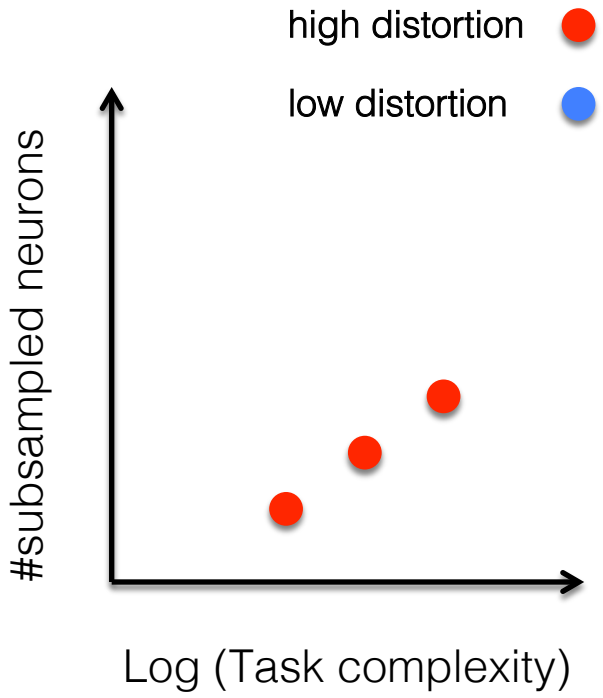
Baraniuk and Wakin 2007

A consequence of neuronal measurement as a random projection

By adapting random projection theory:

$$\text{\# neurons needed} = \frac{1}{\text{distortion}^2}\left(c_1 \log(\text{task complexity}) + c_2\right)$$



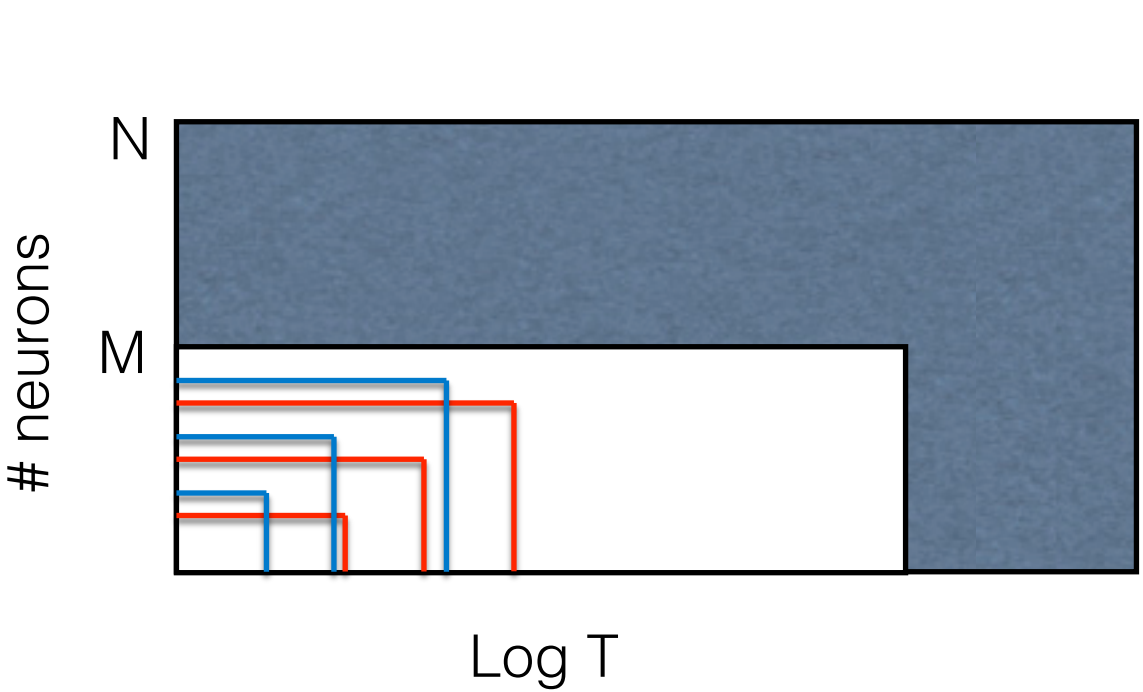


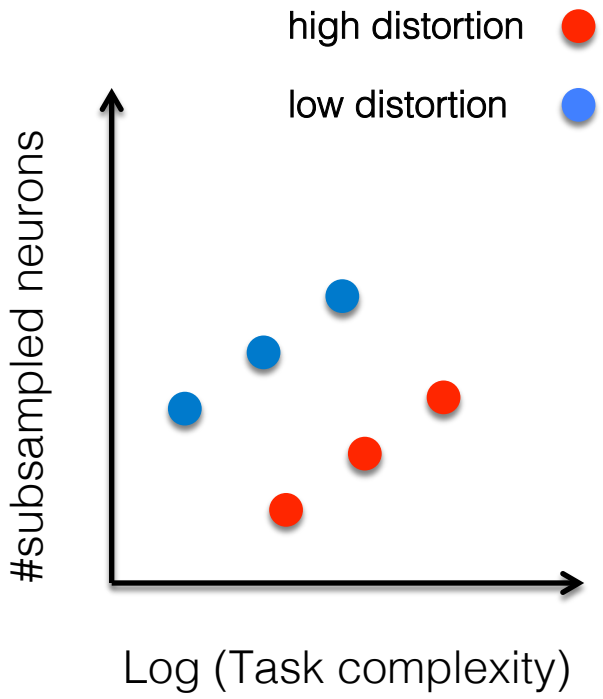
To keep the same level of desired distortion, # of neurons need only scale logarithmically with task complexity (good news!)

A consequence of neuronal measurement as a random projection

By adapting random projection theory:

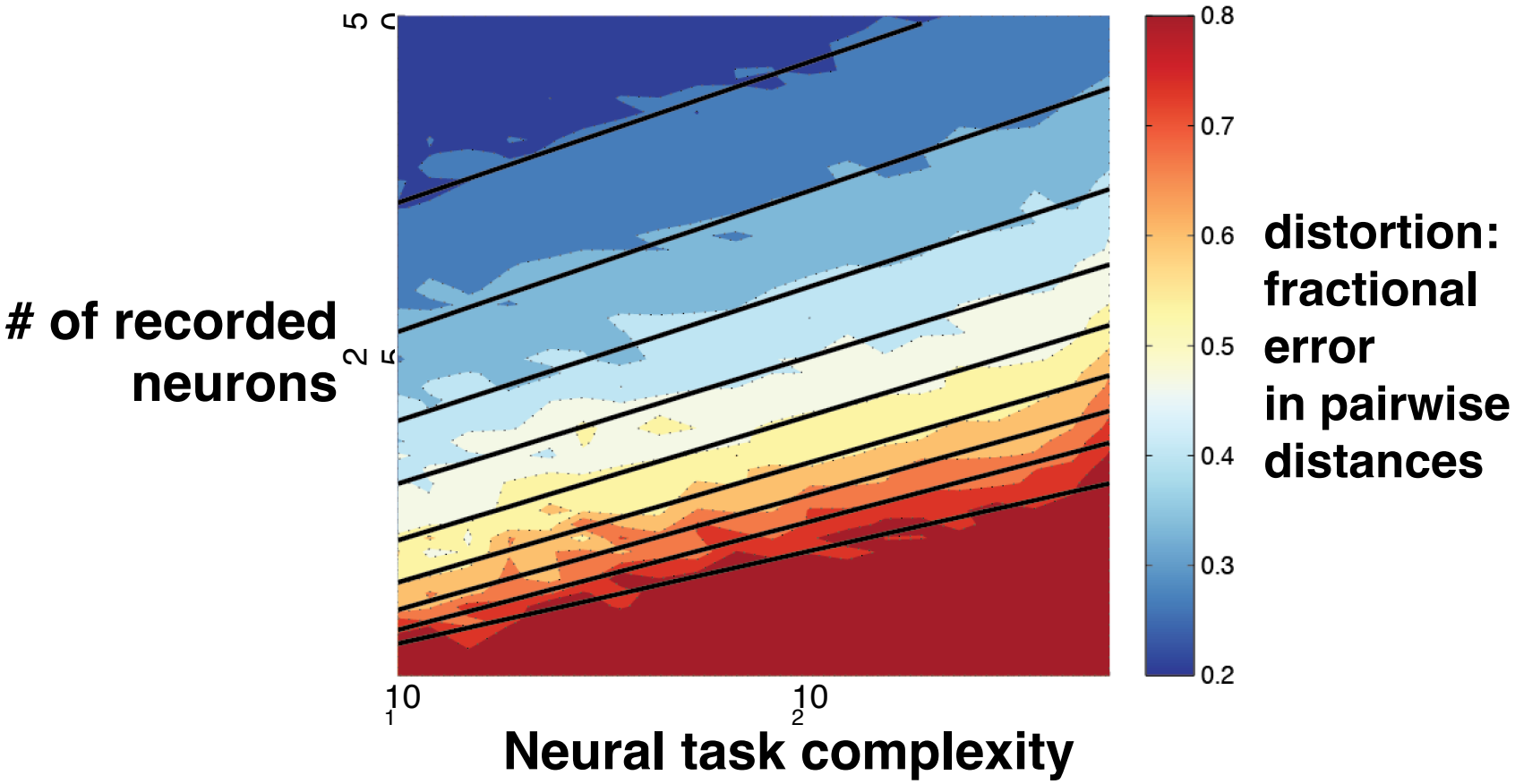
$$\text{\# neurons needed} = \frac{1}{\text{distortion}^2}\,(c_1 \log(\text{task complexity}) + c_2)$$





To keep the same level of desired distortion, # of neurons need only scale logarithmically with task complexity (good news!)

To maintain accuracy of the recovered portraits,
# of neurons required ~ log(task complexity)



distortion contours of motor cortical data

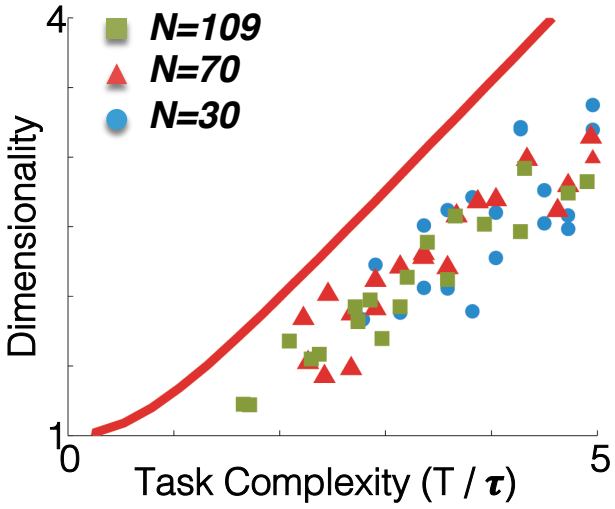
Conclusions

Neural dimensionality:

compare measured dimensionality to our upper bounds

= dimensionality constrained by task complexity and smoothness

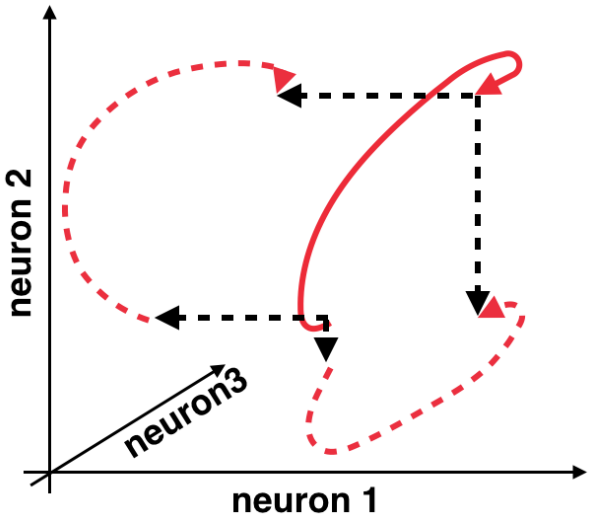
< additional intrinsic dynamic constraint in the data beyond task complexity and smoothness



Neural measurements (optimistic messages):

Subsampling can recover accurate dynamic portraits when neural activities are highly distributed

To recover dynamic portraits from more complex experiments, no need for many more neurons.



Towards a single trial theory

Given limited experimental resources, like

M = number of neurons we can record
P or T = number of training stimuli or amount of time we can record
SNR = signal-to-noise ratio, or trial to trial reproducibility of our data

And a measure of the complexity of our experiment:

K = some measure of the complexity of our
stimuli/behavior/latent variables/manifold of visited neural states

How well can we: Decode behavior on single trials?

Learn the structure of unobserved latent cognitive variables contributing to trial-to-trial variability?

Such a theory should help us design experiments ***before*** they are done!

Towards a single trial theory

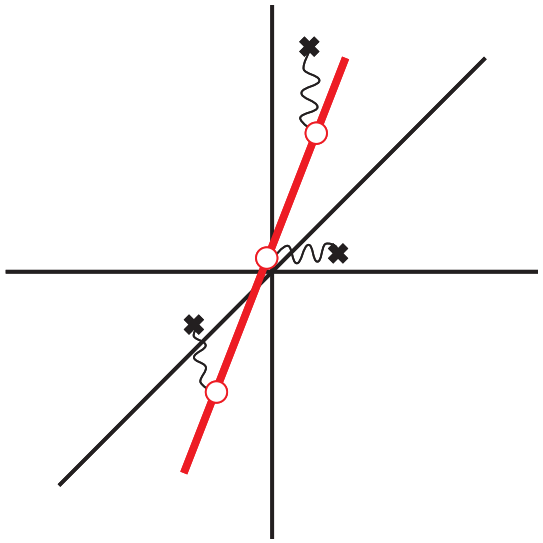
Recovering latent cognitive subspaces: towards a theory of gaussian process factor analysis.

Towards a Rosetta stone between dynamics and statistics: how do statistical latent variable models fit to a subset of neurons, reflect the dynamical properties of a much larger neural circuit?

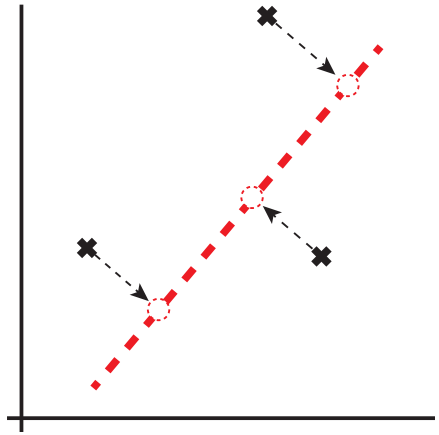
Inferring latent cognitive subspaces

Low K-dimensional stimulus space

Embedding in the N-dimensional space



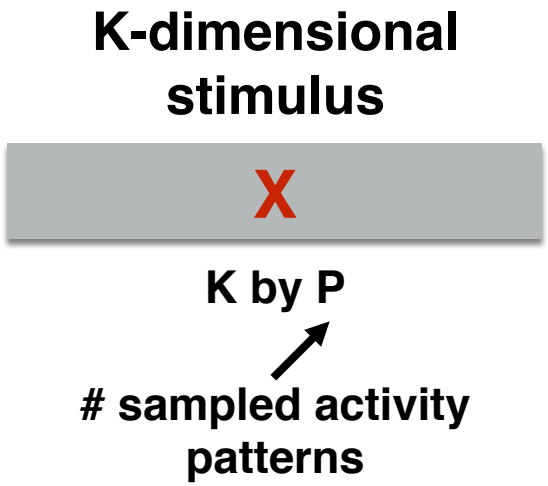
Subsampled in the M-dimensional space



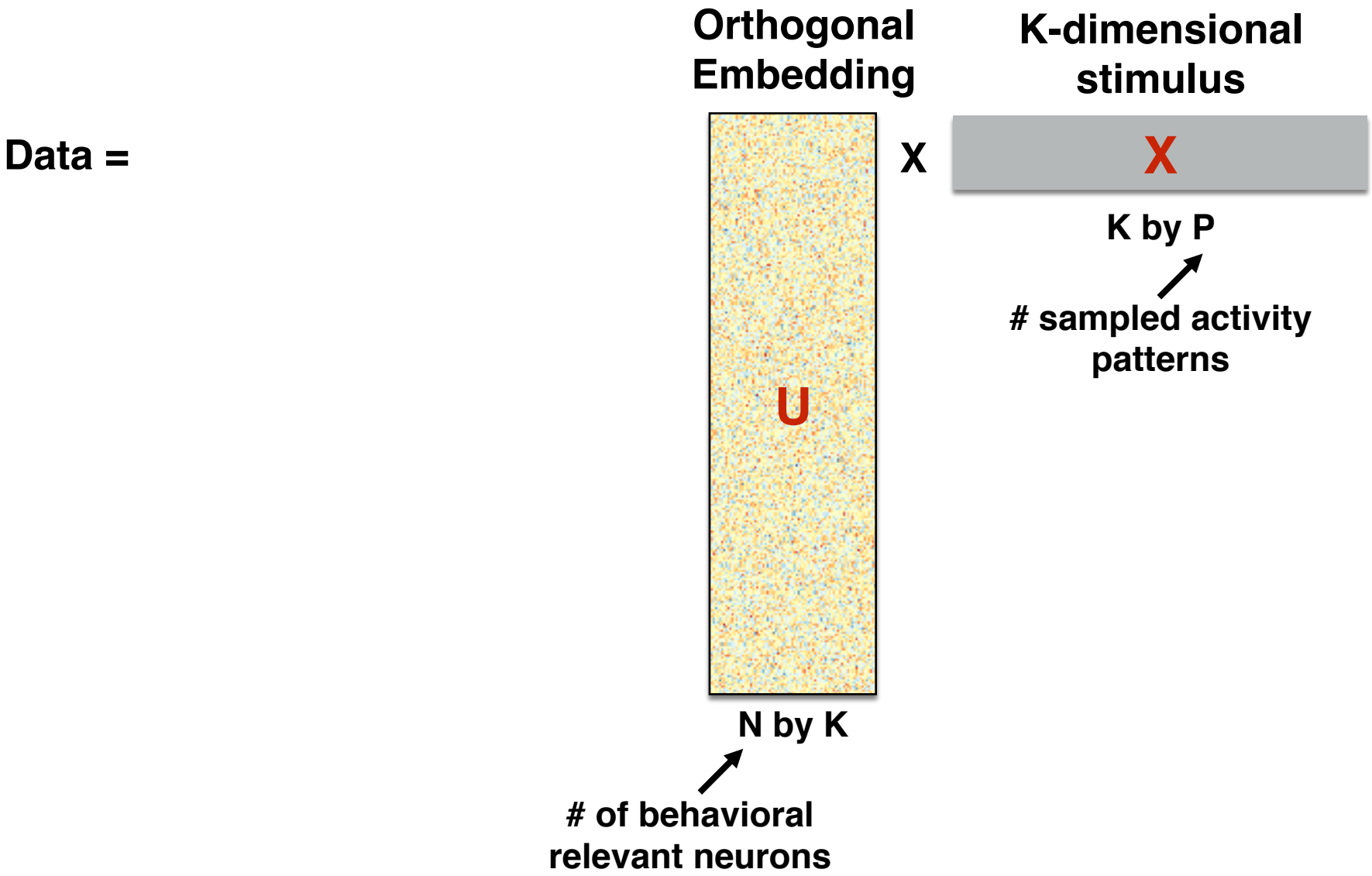
- **Consider a high dimensional neural circuit with N neurons.**
- **We can only record M of them for a finite number of stimuli P.**
- **Suppose the stimuli are encoded simply in a K dimensional subspace (or nonlinear curved manifold in which case K -> NTC)**
- **For what regimes of M, N, P and K and the SNR, can we correctly recover both the subspace and its dimensionality?**

Inferring latent cognitive subspaces

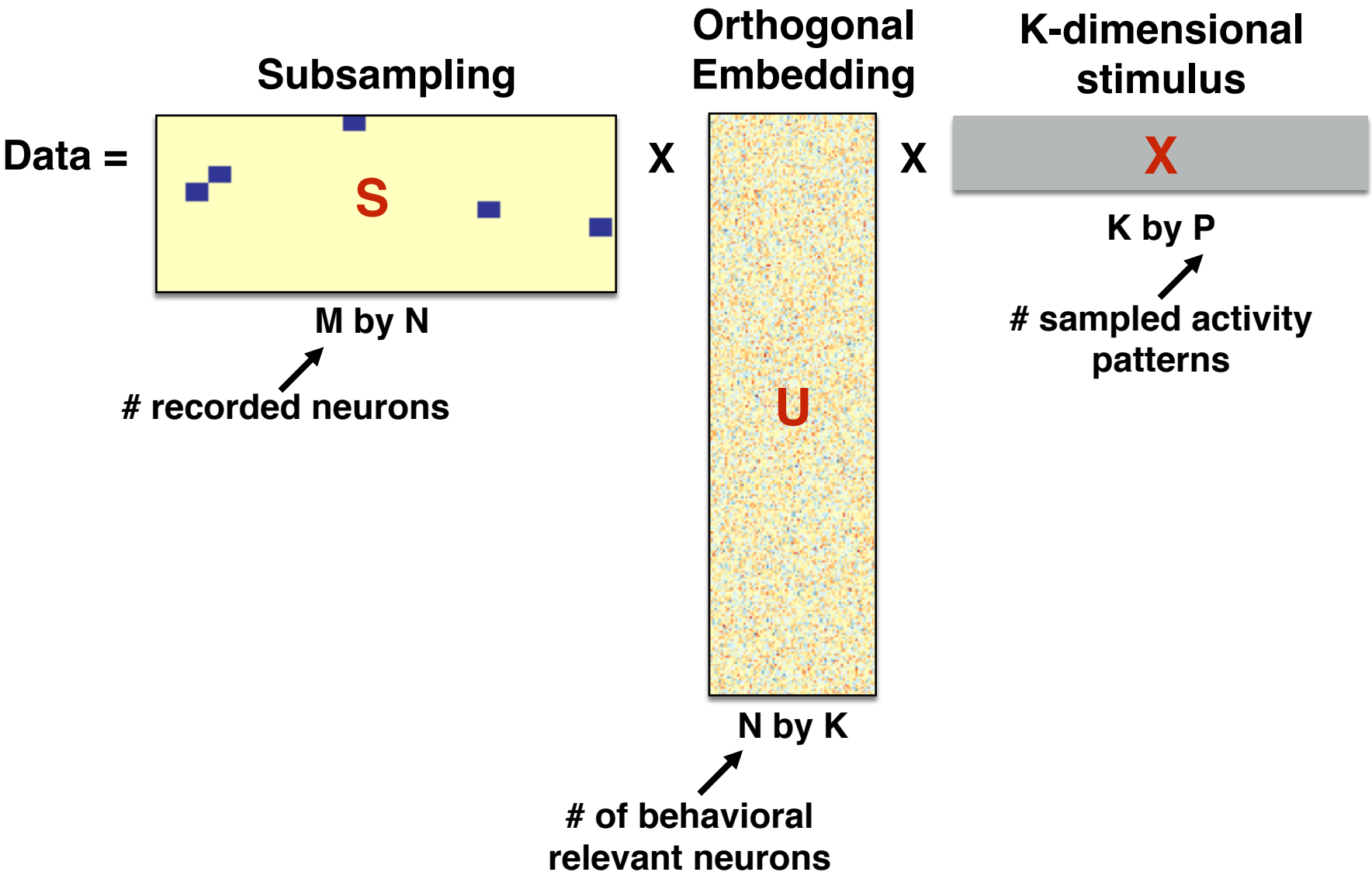
Data =



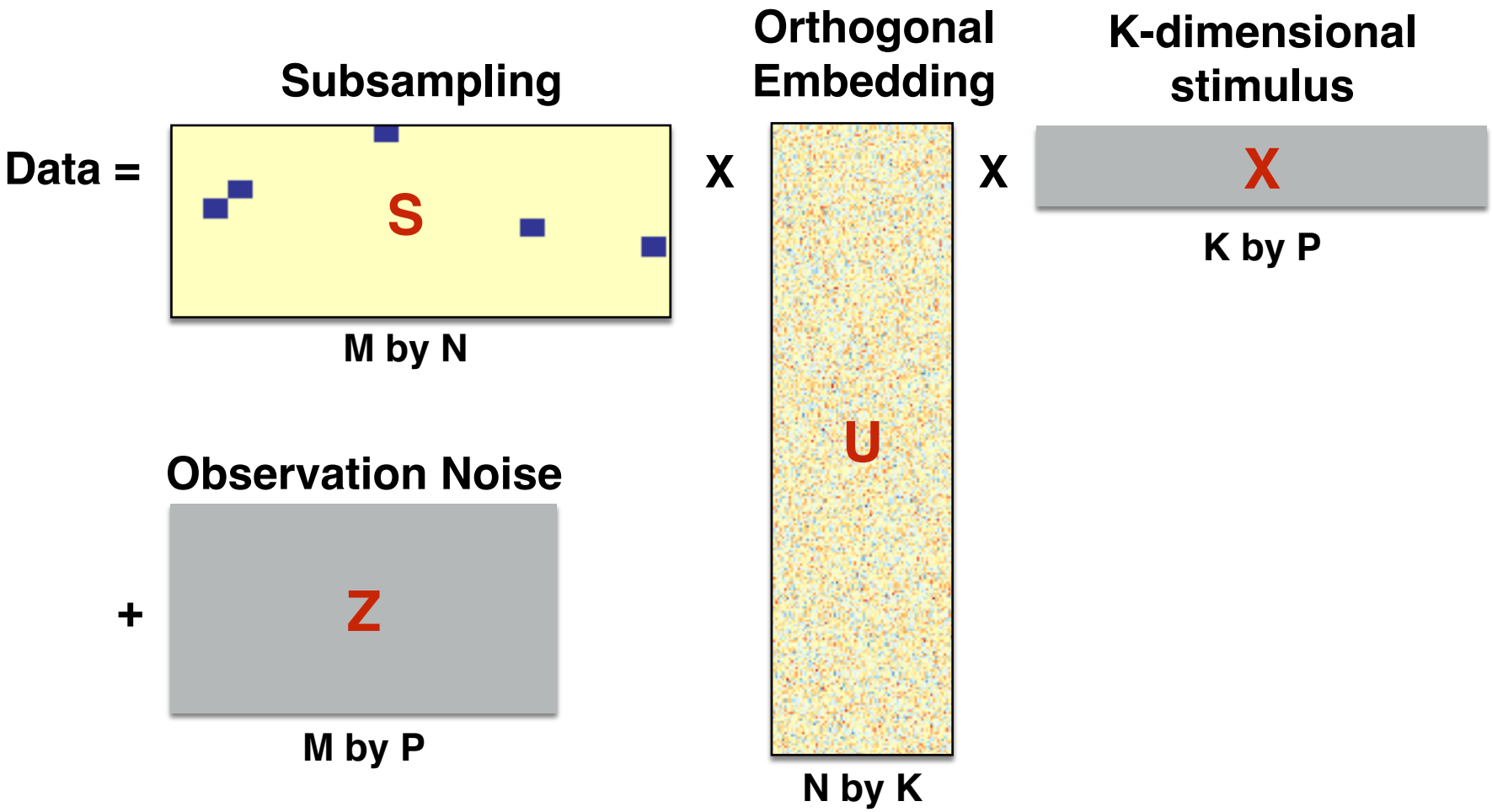
Inferring latent cognitive subspaces



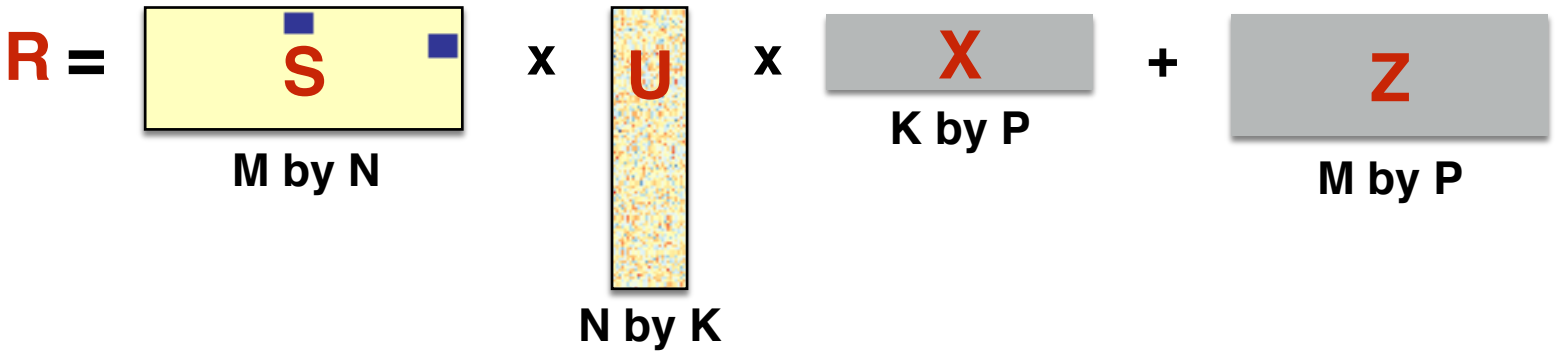
Inferring latent cognitive subspaces



Inferring latent cognitive subspaces



Inferring latent cognitive subspaces



Dimensions:

N (# nrns) > P (# trials) > M (# record nrns) > K (stim dim)

Signal and Noise Models:

$$X_{ij} \sim \mathcal{N}(0, \frac{N}{K}\sigma_s^2) \qquad Z_{ij} \sim \mathcal{N}(0, \sigma_n^2)$$

Neuronal Signal-to-Noise ratio:

$$\mathrm{SNR} = \frac{\sigma_s^2}{\sigma_n^2}$$

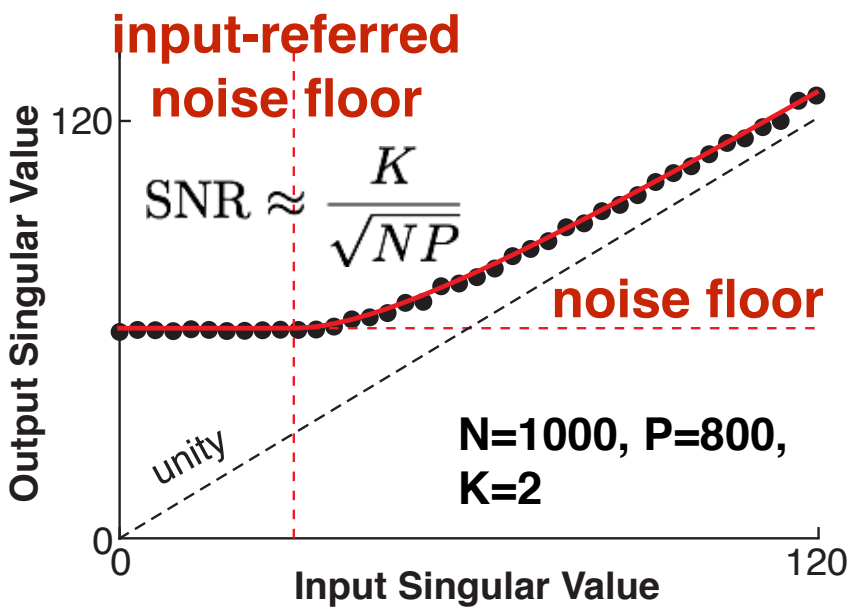
Low-rank Matrix Perturbation Theory

With completion observation (i.e. M = N):

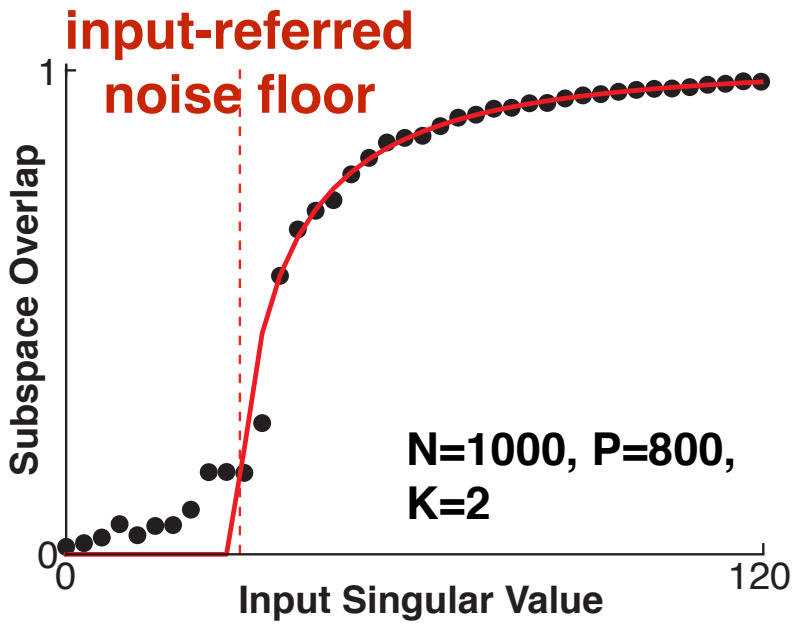
$$R = UX + Z$$

low-rank signal high-dim noise

Singular value transfer function



Subspace overlap



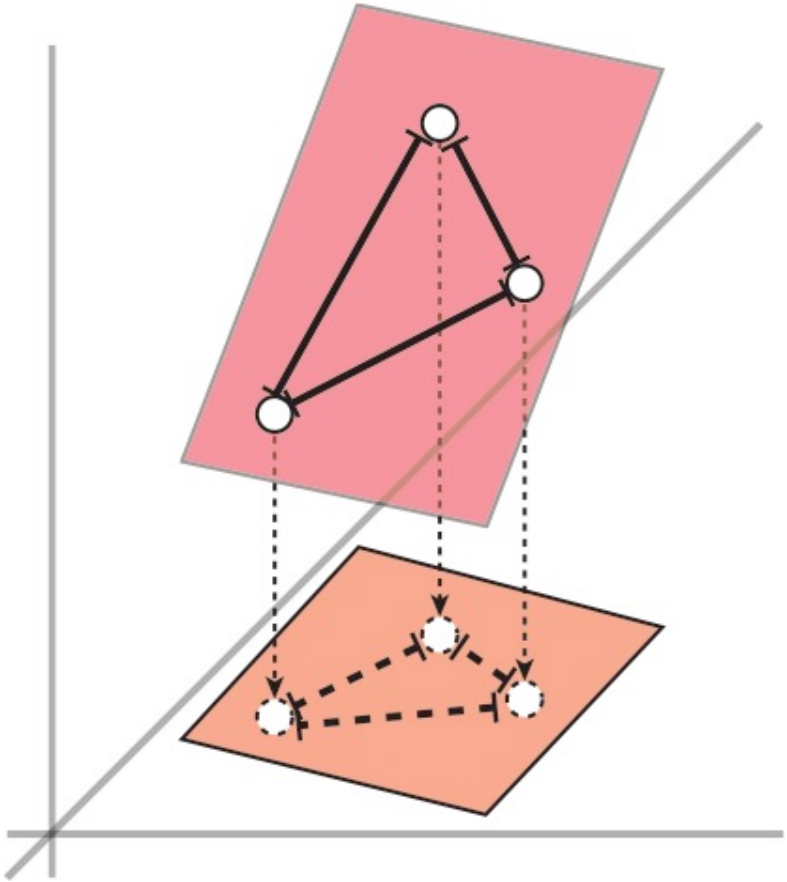
Subsampling

With partial observation (i.e. M < N):

$$R = SUX + Z$$

low-rank signal

high-dim noise



- K(=2)-dimensional stimulus space
- Embedded in N(=3)-dimensional neural space
- Subsampled to M(=2)-dimensional subspace
- Distance between sampled activity patterns are compressed
- Compressions are different depending on orientations
- Compression determined by the K singular values of SU

Singular Value Spectrum of SU

M by N random sampling matrix

x

N by K random orthogonal embedding matrix

Singular value spectrum:

$$\mu_{SU}(\sigma)$$

distribution of all possible signal distortions by the SU matrix

SU's singular value squared are the eigenvalues of S^TSU^TU

$$\sigma_k \left[SU\right]^2 = \mathrm{eig}_k \left[SUU^T S^T\right] = \mathrm{eig}_k \left[S^T SUU^T\right]$$

Free probability theory: S^TS and U^TU are two independent projection matrices, their *S*-transforms obey:

$$S_{S^T SU^T U}(z) = S_{S^T S}(z) S_{U^T U}(z)$$

Nica & Speicher, 1996

$$S_A(z) \doteq \frac{1+z}{z} M_A^{-1}(z)$$

$$M_A(z) \doteq \sum_{m=0}^{\infty} \frac{1}{N} \mathrm{Tr}\left[(AA^T)^m\right] z^m$$

Distribution of distortions:

$$\mu_{SU}(\sigma) = \frac{N}{M} \frac{\sqrt{(\sigma^2 - \sigma_-^2)(\sigma_+^2 - \sigma^2)}}{\pi\sigma(1-\sigma^2)}$$

$$\sigma_\pm = \sqrt{\frac{M}{N}}\sqrt{1-\frac{K}{N}}\left(1 \pm \sqrt{\frac{K}{M}\frac{N-M}{N-K}}\right)$$

upper/lower bounds of distortion

Singular Value Spectrum of SU

M by N random sampling matrix

x

N by K random orthogonal embedding matrix

Singular value spectrum:

$$\mu_{SU}(\sigma)$$

distribution of all possible signal distortions by the SU matrix

Distribution of distortions:

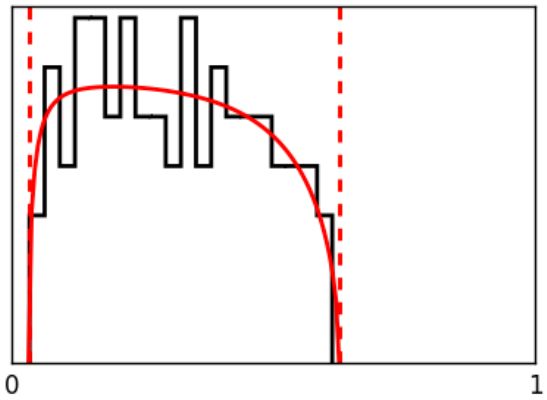
$$\mu_{SU}(\sigma) = \frac{N}{M} \frac{\sqrt{(\sigma^2 - \sigma_-^2)(\sigma_+^2 - \sigma^2)}}{\pi\sigma(1-\sigma^2)}$$

$$\sigma_\pm = \sqrt{\frac{M}{N}}\sqrt{1-\frac{K}{N}}\left(1 \pm \sqrt{\frac{K}{M}\frac{N-M}{N-K}}\right)$$

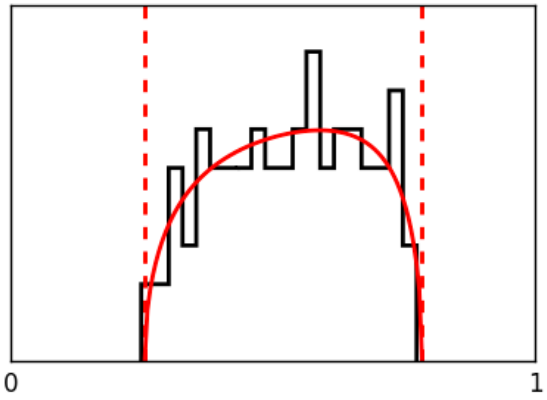
upper/lower bounds of distortion

Simulations (N = 1000, K = 100):

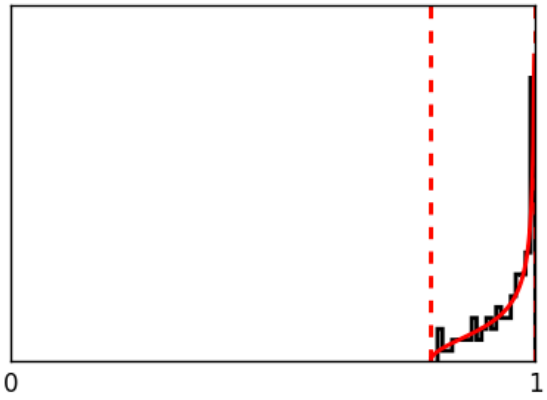
M = 120 ~ K



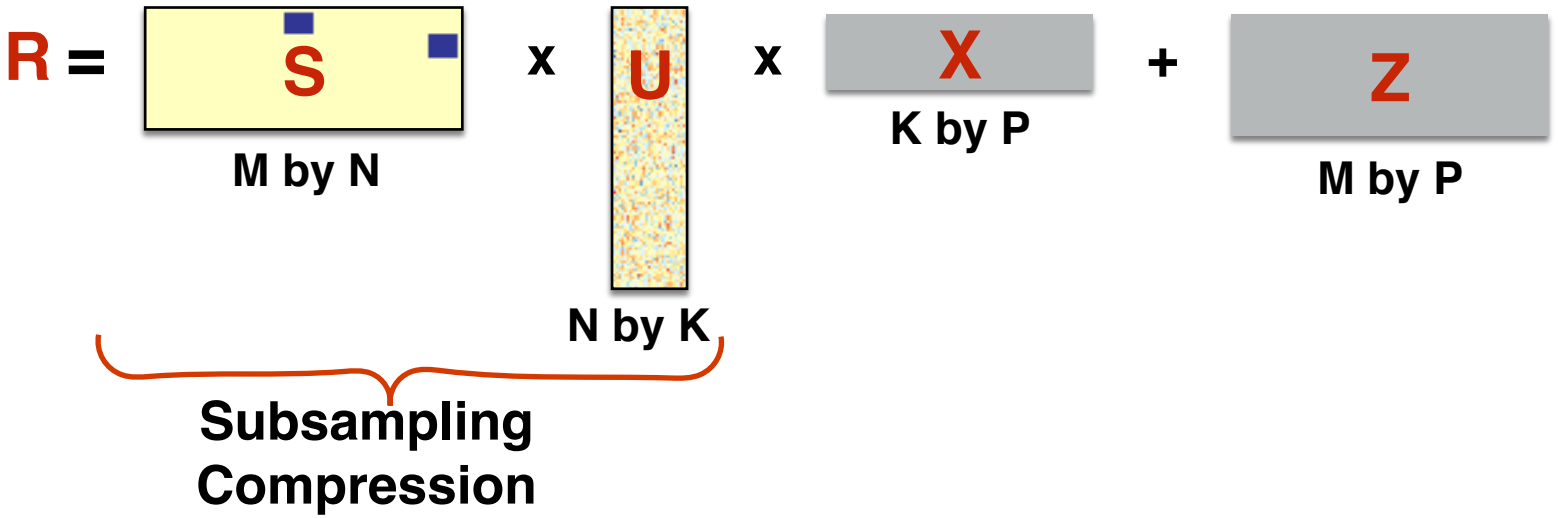
M = 300 > K



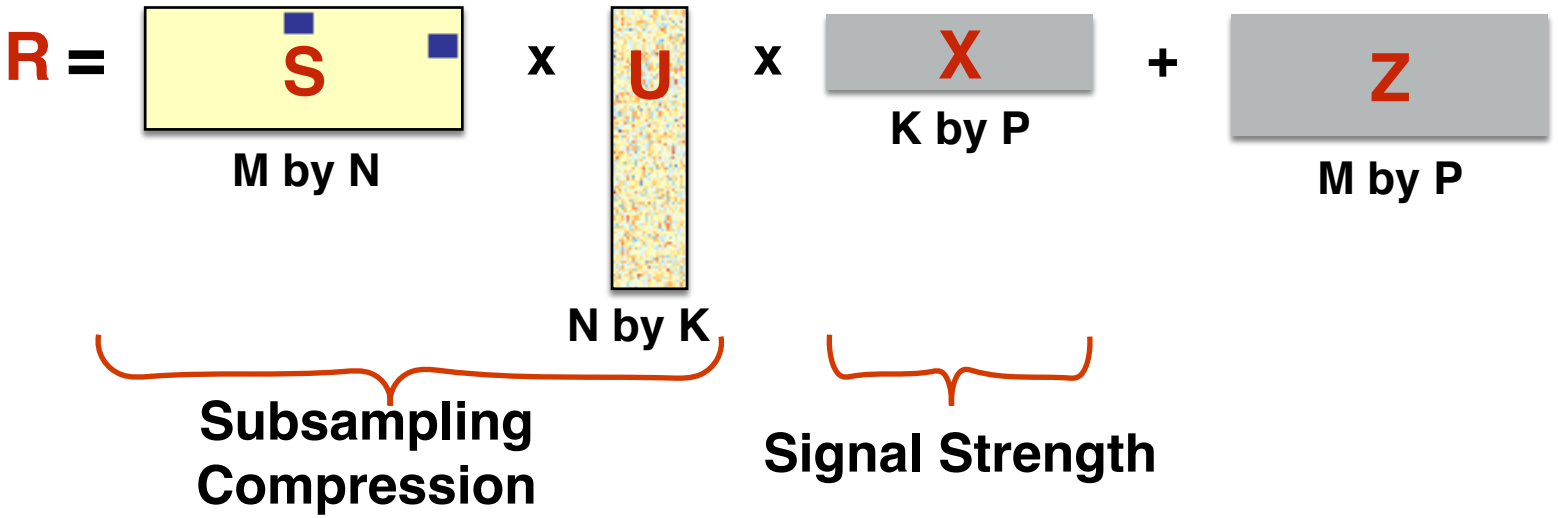
M = 900 >> K



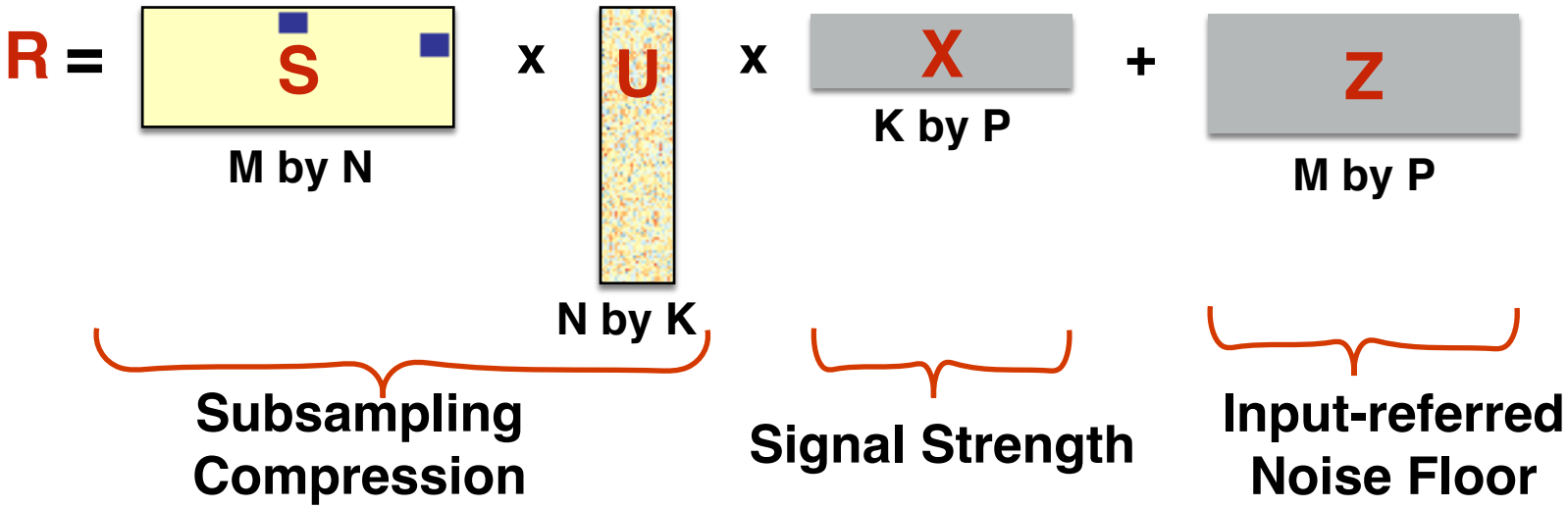
Static Decoding - Recovering Dimensionality



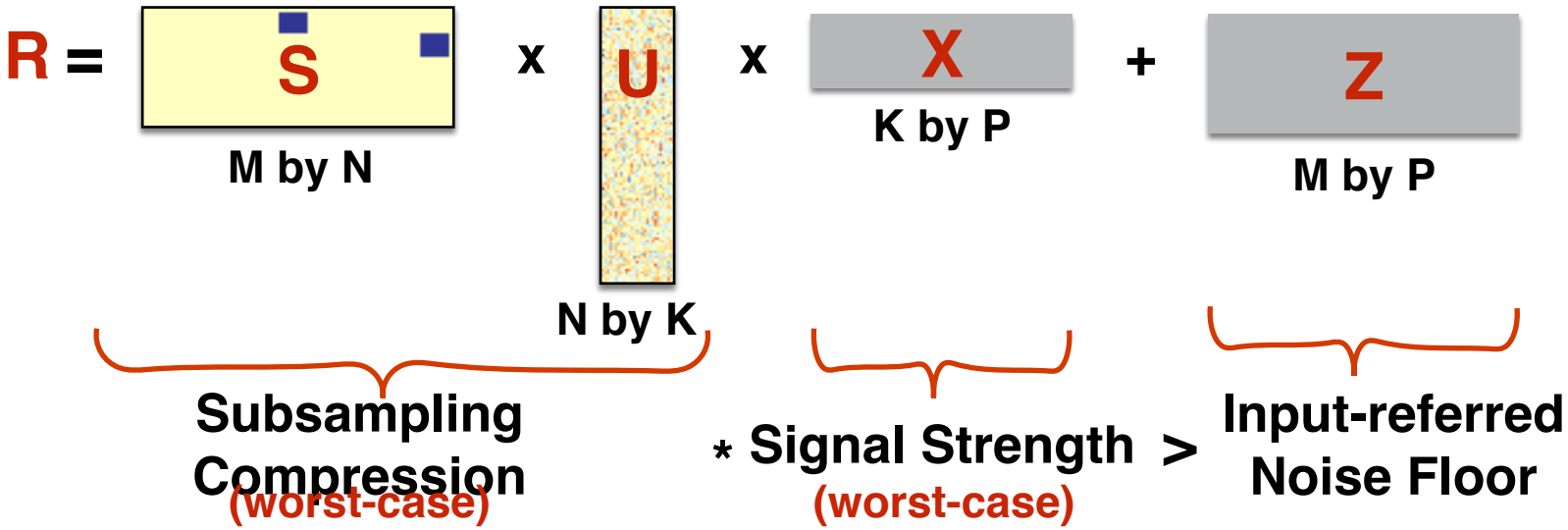
Static Decoding - Recovering Dimensionality



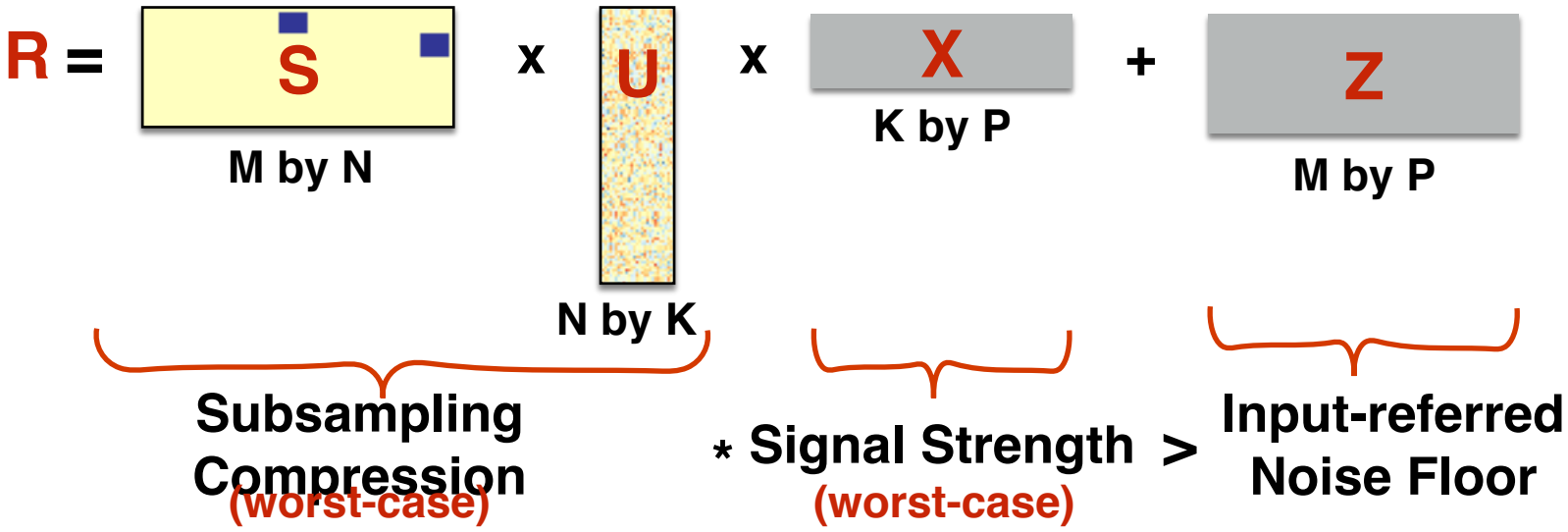
Static Decoding - Recovering Dimensionality



Static Decoding - Recovering Dimensionality



Static Decoding - Recovering Dimensionality

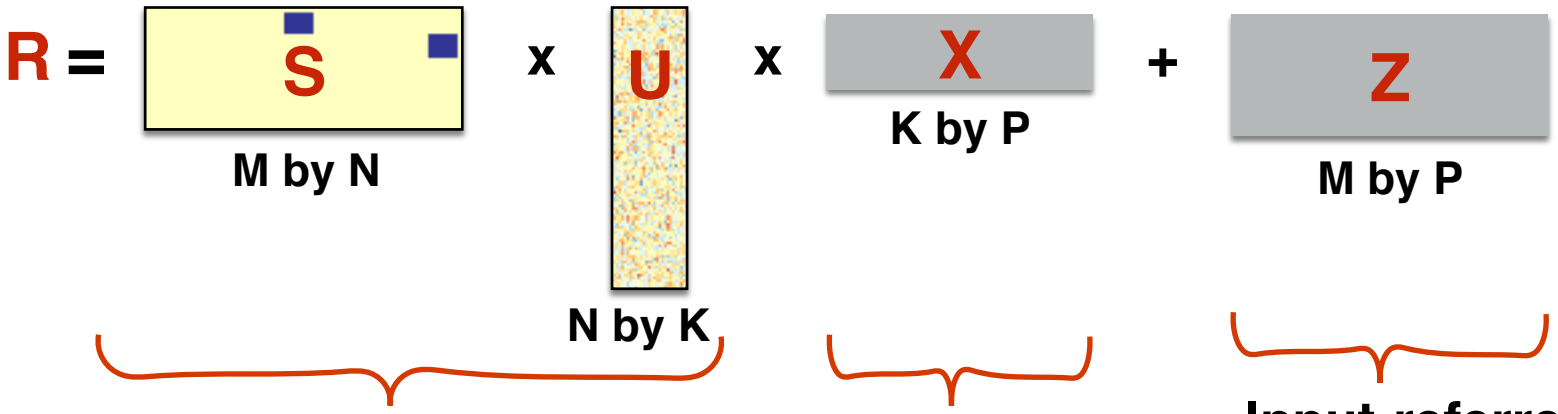


$$\sigma_{\min}[SU]$$

$$\sqrt{\frac{M}{N}}\sqrt{1-\frac{K}{N}}\left(1-\sqrt{\frac{K(N-M)}{M(N-K)}}\right)$$

$$\sqrt{\frac{M}{N}}$$

Static Decoding - Recovering Dimensionality



Subsampling Compression * **Signal Strength** > **Input-referred Noise Floor**

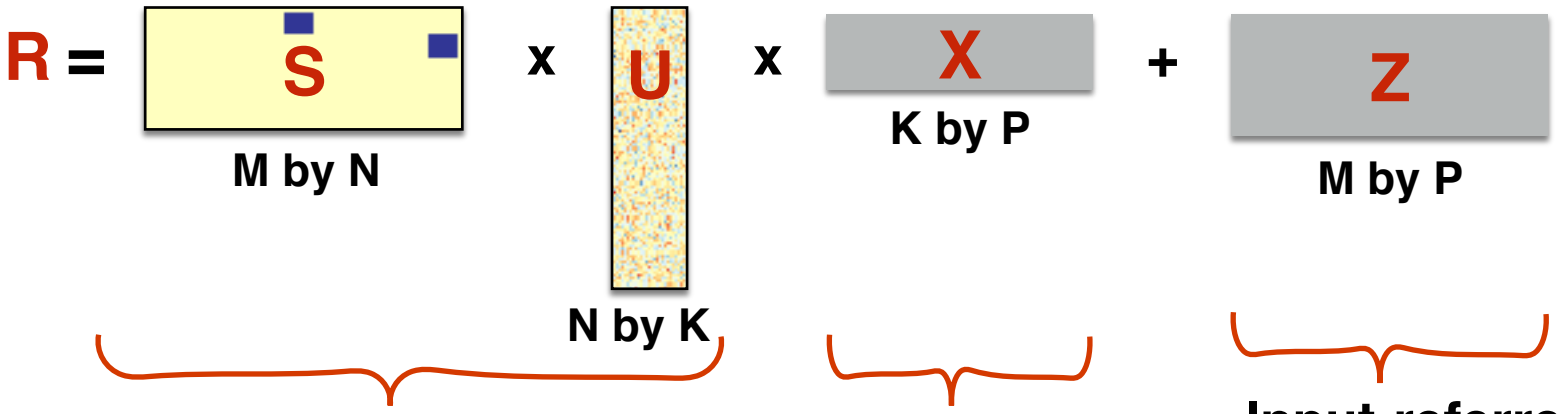
(worst-case) **(worst-case)**

$$\sigma_{\min}[SU] \qquad \sigma_{\min}[X]$$

$$\sqrt{\frac{M}{N}}\sqrt{1-\frac{K}{N}}\left(1-\sqrt{\frac{K(N-M)}{M(N-K)}}\right) \qquad \sigma_s\sqrt{\frac{N}{K}}(\sqrt{K}+\sqrt{P})$$

$$\sqrt{\frac{M}{N}} \qquad \sigma_s\sqrt{\frac{N}{K}}\sqrt{P}$$

Static Decoding - Recovering Dimensionality



Subsampling Compression * **Signal Strength** > **Input-referred Noise Floor**

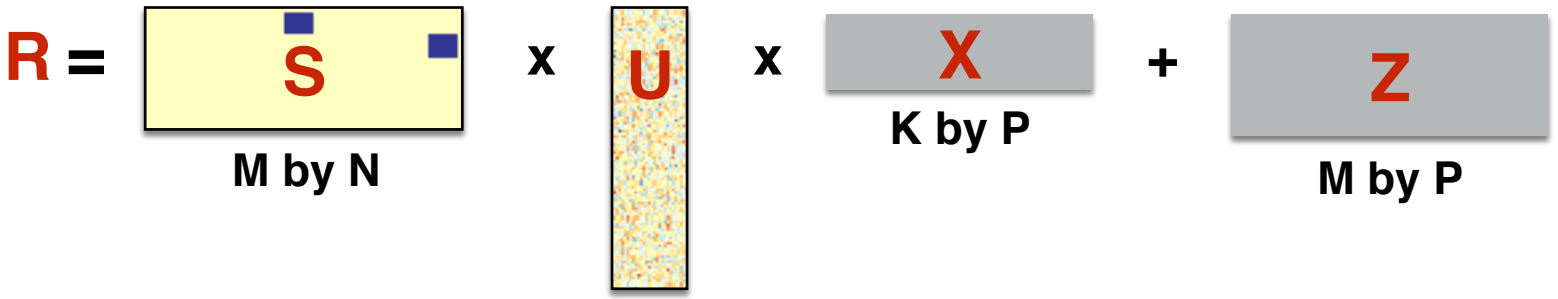
(worst-case) **(worst-case)**

$$\sigma_{\min}[SU] \qquad \sigma_{\min}[X] \qquad F^{-1}(\sigma_{\max}[Z])$$

$$\sqrt{\frac{M}{N}}\sqrt{1-\frac{K}{N}}\left(1-\sqrt{\frac{K(N-M)}{M(N-K)}}\right) \qquad \sigma_s\sqrt{\frac{N}{K}}(\sqrt{K}+\sqrt{P}) \qquad \sigma_n(MP)^{1/4}$$

$$\sqrt{\frac{M}{N}} \qquad \sigma_s\sqrt{\frac{N}{K}}\sqrt{P} \qquad \sigma_n(MP)^{1/4}$$

Static Decoding - Recovering Dimensionality



Subsampling Compression (worst-case) * **Signal Strength** (worst-case) > **Input-referred Noise Floor**

$$\sigma_{\min}[SU] \qquad \sigma_{\min}[X] \qquad F^{-1}(\sigma_{\max}[Z])$$

$$\sqrt{\frac{M}{N}}\sqrt{1-\frac{K}{N}}\left(1-\sqrt{\frac{K(N-M)}{M(N-K)}}\right) \qquad \sigma_s\sqrt{\frac{N}{K}}(\sqrt{K}+\sqrt{P}) \qquad \sigma_n(MP)^{1/4}$$

$$\sqrt{\frac{M}{N}} \qquad \sigma_s\sqrt{\frac{N}{K}}\sqrt{P} \qquad \sigma_n(MP)^{1/4}$$

neuronal gain

recording/trial gains

$$\mathrm{SNR}\sqrt{MP} = \sqrt{MP}\frac{\sigma_s^2}{\sigma_n^2} \gtrsim K$$

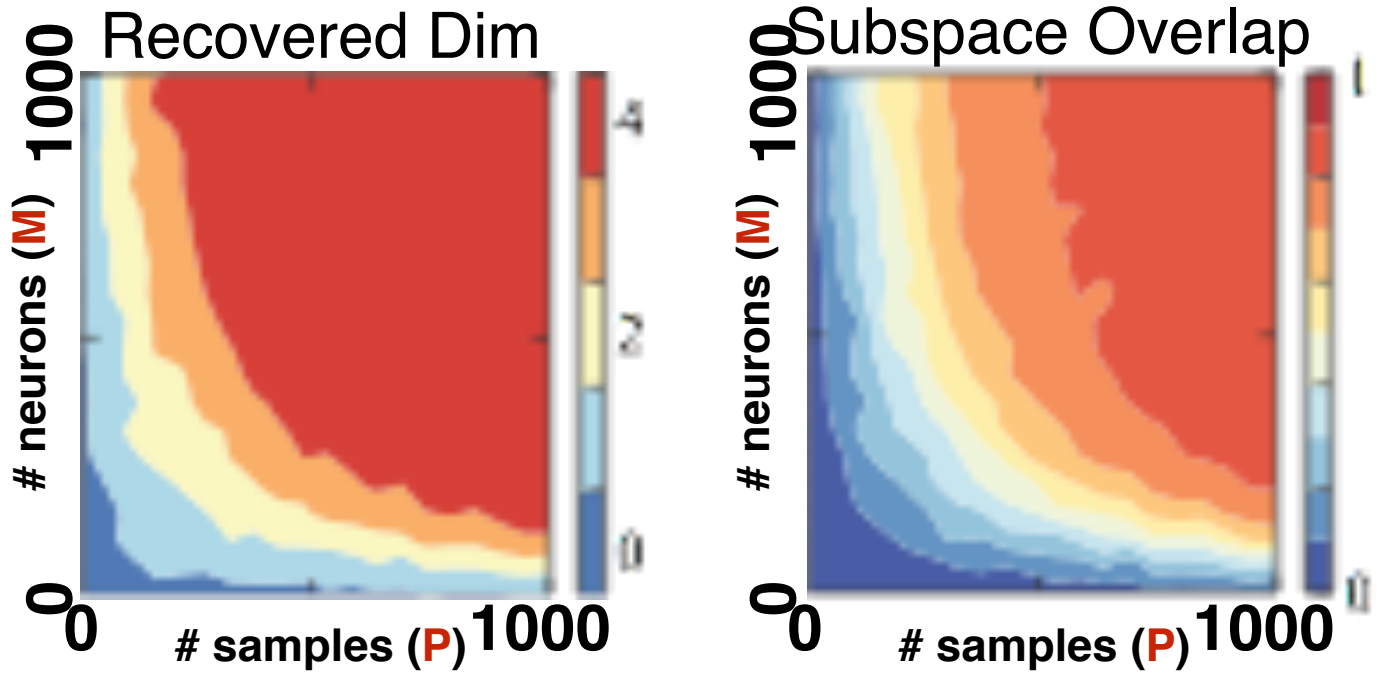
M > K
P > K

Latent State Recovery: Simulations

$$\sqrt{MP} \geq \frac{D}{\mathrm{SNR}} \qquad M > D, P > D$$

Prediction: hyperbolic phase transition in the M-by-P plane:

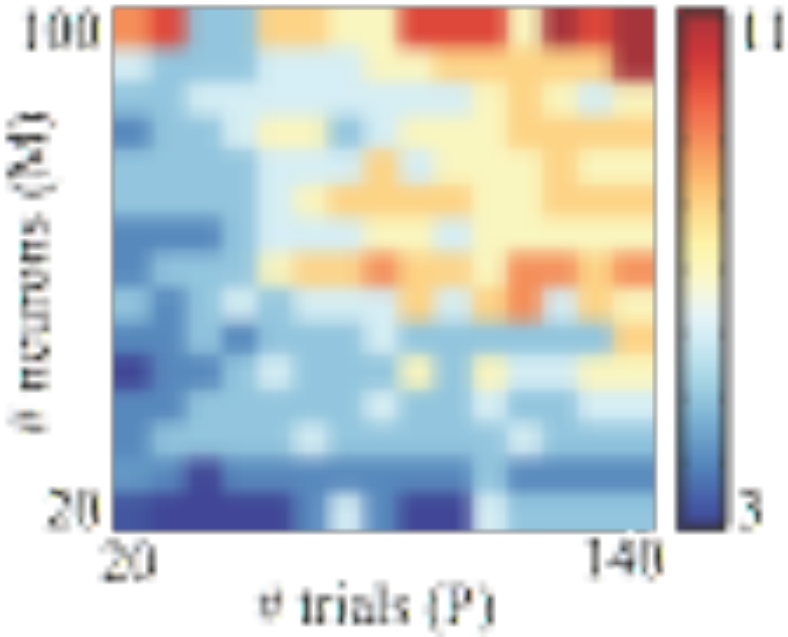
Simulated recovery of random smooth trajectories with **SNR** = 0.02, **D**=4



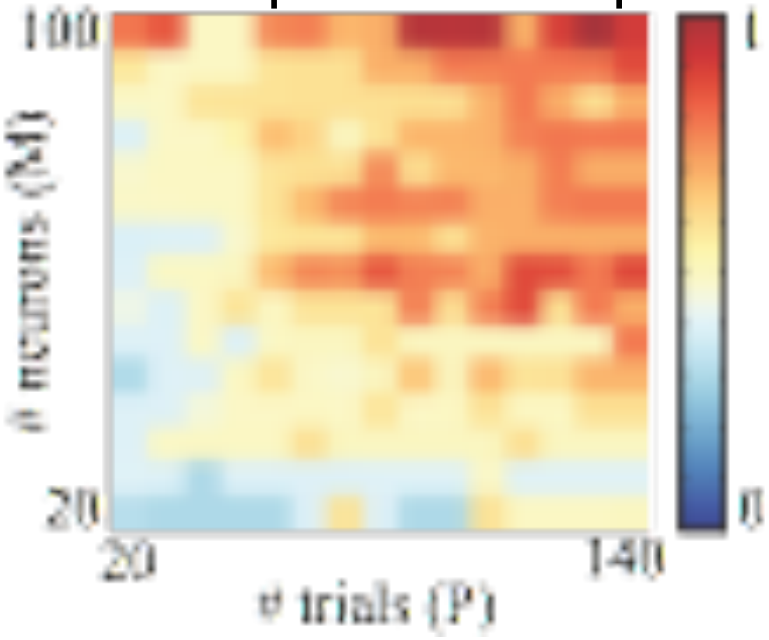
Latent State Recovery: Monkey Data

- Motor cortical data from Shenoy lab, center-out reaching tasks
- GPFA to extract latent trajectories in 147 trials for reaches to a single target *Yu et al. 2009*
- **Different algorithm**: **D** inferred using cross-validated Gaussian Process Factor Analysis
- **Different noise in the data**: pink and temporally correlated

Inferred **D**



Subspace Overlap



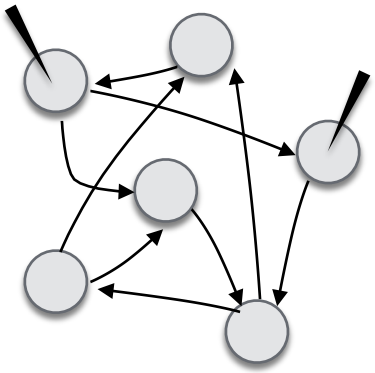
Qualitatively similar tradeoff between M and P

Towards a single trial theory

Recovering latent cognitive subspaces: towards a theory of gaussian process factor analysis.

Towards a Rosetta stone between dynamics and statistics: how do statistical latent variable models fit to a subset of neurons, reflect the dynamical properties of a much larger neural circuit?

Discovering structure in subsampled neural dynamics



- Consider a high dimensional neural circuit with N neurons.
- We can only record M of them for a finite amount of time T.
- What can we correctly infer about the circuit dynamics when M << N and T is not too large?
- In general - nothing!
- However if we might assume an underlying simplicity, for example low dimensional dynamics of dimension K.
- For what regimes of M, N, T and K can we correctly recover dynamical properties of the circuit?

Model Linear Neural Network

N-dimensional state

$$x_{t+1} = e^{-\frac{1}{\tau}} x_t + W x_t + \eta_t$$

M-dimensional observation

$$y_t = B x_t$$

rank-K connectivity

input

random sampling

neuronal property

Data often modeled using latent linear dynamical systems

nonlinear and stochastic transforms of y may be used to model spikes directly

$$z_{t+1} = Az_t + \xi_t, \text{ with } \xi_t \sim \mathcal{N}(0, Q)$$
$$y_t = Cz_t + \eta_t, \text{ with } \eta_t \sim \mathcal{N}(0, R = rI)$$
$$z_1 \sim \mathcal{N}(0, \Pi)$$

SSID used to find the slow mode eigenvalues of the generative model

Hankel Matrix:

$$\mathbf{H} = \begin{bmatrix} \langle y_t y_{t+1}^T \rangle & \langle y_{t-1} y_{t+1}^T \rangle & \langle y_{t-2} y_{t+1}^T \rangle \\ \langle y_t y_{t+2}^T \rangle & \langle y_{t-1} y_{t+2}^T \rangle & \cdots \\ \langle y_t y_{t+3}^T \rangle & \vdots & \ddots \end{bmatrix}$$

approximated using data

$$= \begin{bmatrix} CA\Pi C & CA^2\Pi C & CA^3\Pi C \\ CA^2\Pi C & CA^3\Pi C & \cdots \\ CA^3\Pi C & \vdots & \ddots \end{bmatrix} \qquad \Pi = \langle y_t y_t^T \rangle$$

theoretical values with infinite data

$$= \begin{bmatrix} CA \\ CA^2 \\ \vdots \end{bmatrix} \begin{bmatrix} \Pi C & A\Pi C & \cdots \end{bmatrix}$$

factorization

Linear regression to find A's eigenvalues

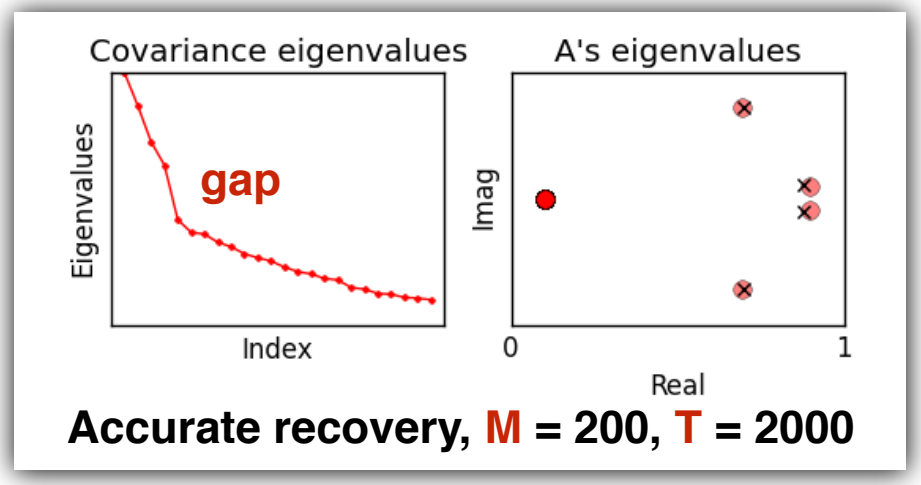
Data often modeled using latent linear dynamical systems

nonlinear and stochastic transforms of y may be used to model spikes directly

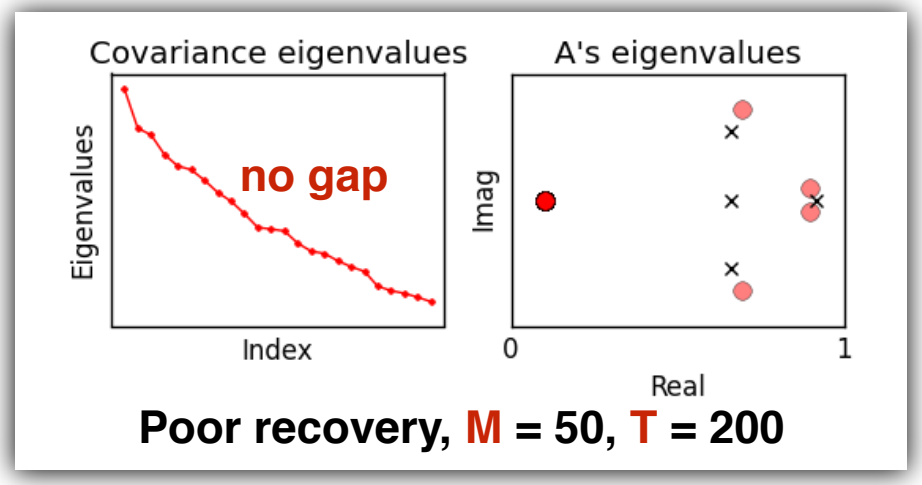
$$z_{t+1} = Az_t + \xi_t, \text{ with } \xi_t \sim \mathcal{N}(0, Q)$$
$$y_t = Cz_t + \eta_t, \text{ with } \eta_t \sim \mathcal{N}(0, R = rI)$$
$$z_1 \sim \mathcal{N}(0, \Pi)$$

When are the eigenvalues of fitted (SSID) latent dynamics *A* close to those of slowest modes of the generative model?

N = 1000, K = 4, τ = 0.4, τ_{slow} = 9.5



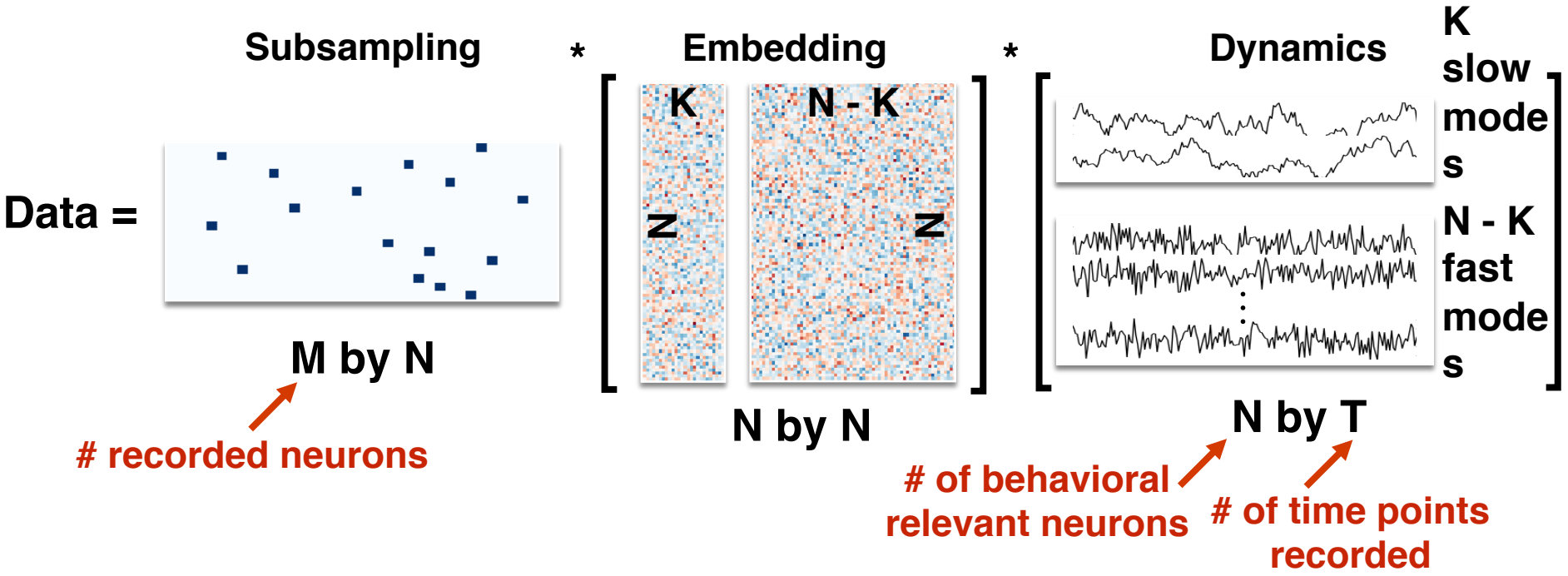
Accurate recovery, M = 200, T = 2000



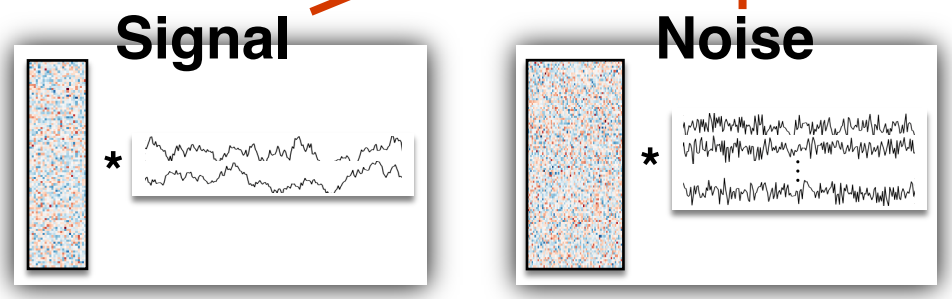
Poor recovery, M = 50, T = 200

A gap is required in the eigenvalue spectrum of the observation Y's covariance matrix.

To understand the spectrum of the covariance matrix, we factorize the data



$$X = \text{sampling} * (X_{slow} + X_{fast})$$



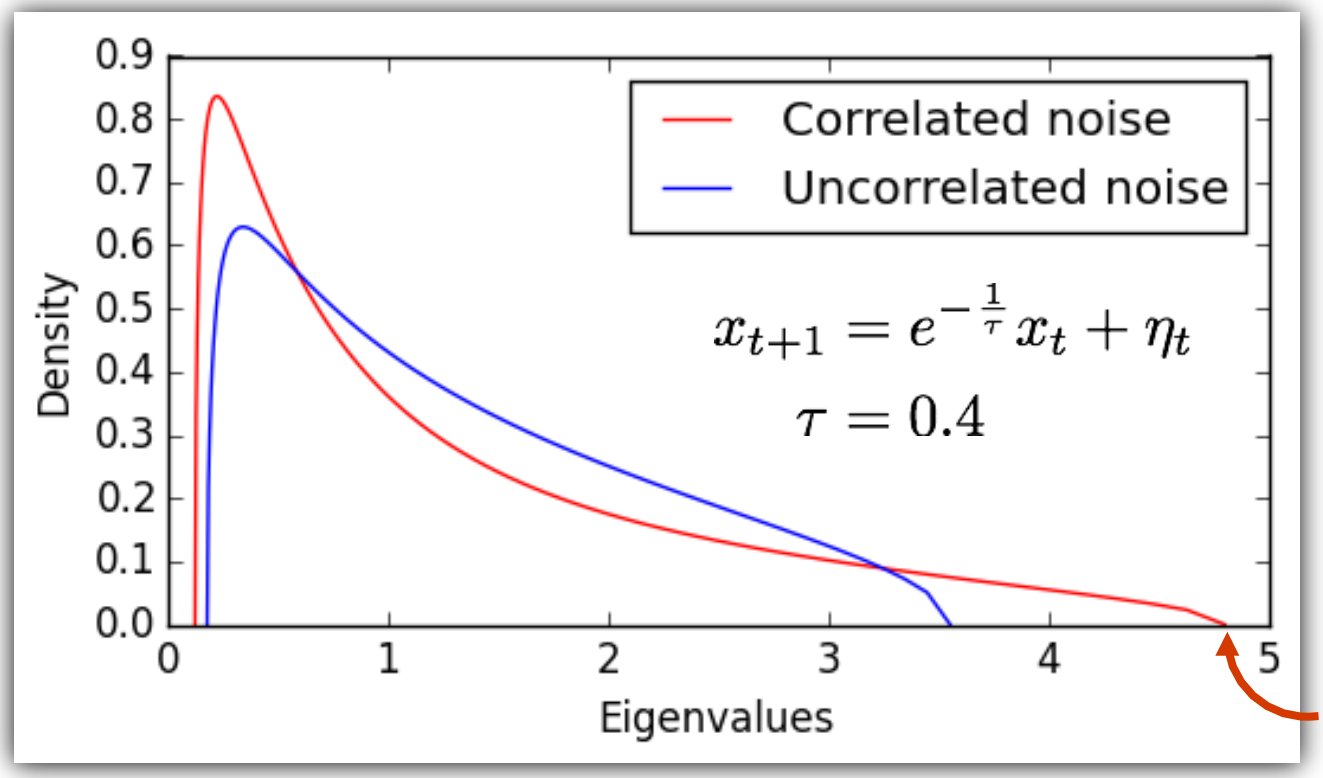
Similar setup to factor analysis, but the noise is correlated across time

Data can be thought of as a low-rank perturbation of a random noise matrix

$X_{slow} + X_{fast}$ *Benaych-Georges & Nadakuditi, 2012*

Eigenvalue spectrum of correlated noise deviates from the Marchenko-Pastur law

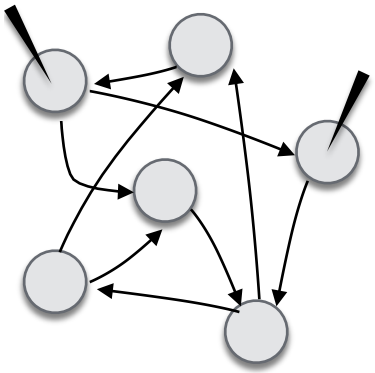
Theoretical eigenvalue spectrum for N = 1000, T = 2000 noise matrix



Marchenko & Pastur, 19
Bai et al., 2008, Yao, 20

b(N, T, τ)
noise floor

Discovering structure in subsampled neural dynamics



Model Linear Neural Network

N-dimensional state

$$x_{t+1} = e^{-\frac{1}{\tau}} x_t + W x_t + \eta_t$$

M-dimensional observation

$$y_t = B x_t$$

rank-K connectivity

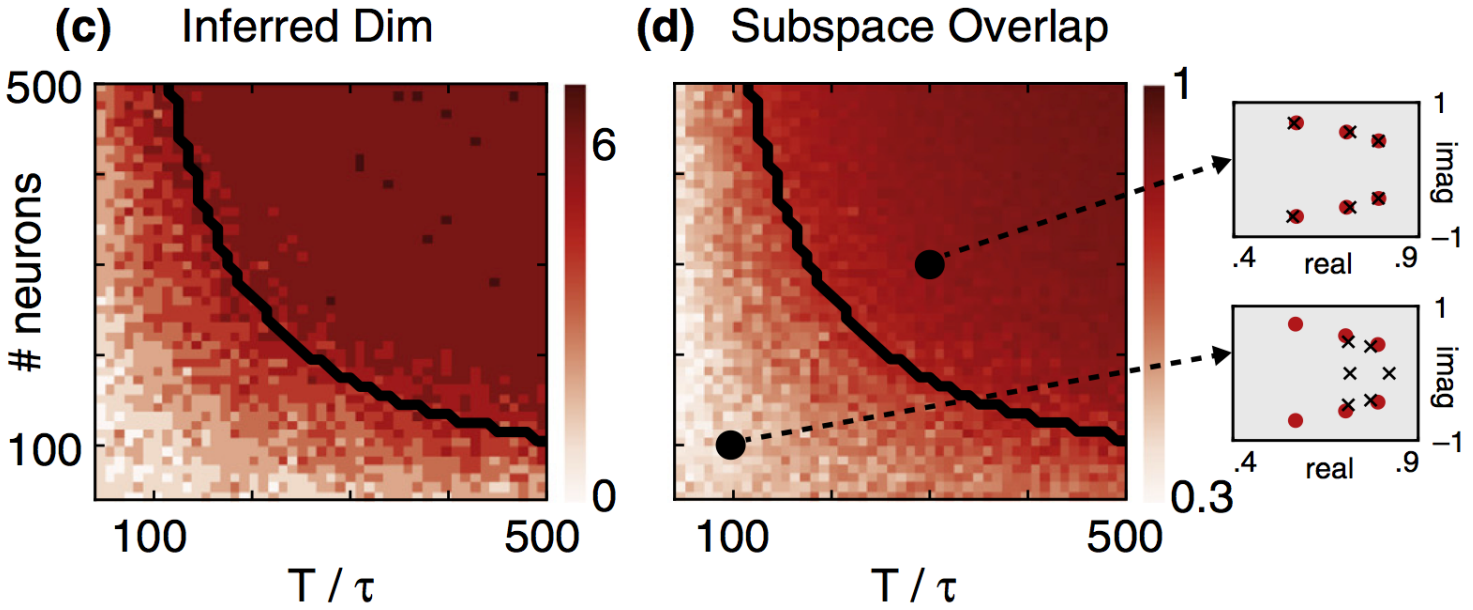
input

random sampling

neuronal property

N = 5000 K = 6

Dynamics Learning



Model Nonlinear Neural Network

compressive nonlinearity

N-dimensional state: $$\mathbf{r}^{t+1} = e^{-1/\tau_\epsilon}\mathbf{r}^t + \mathbf{W}\left[s\tanh\left(\frac{1}{s}\mathbf{r}^t\right)\right] + \boldsymbol{\eta}^t$$

neuronal time constant

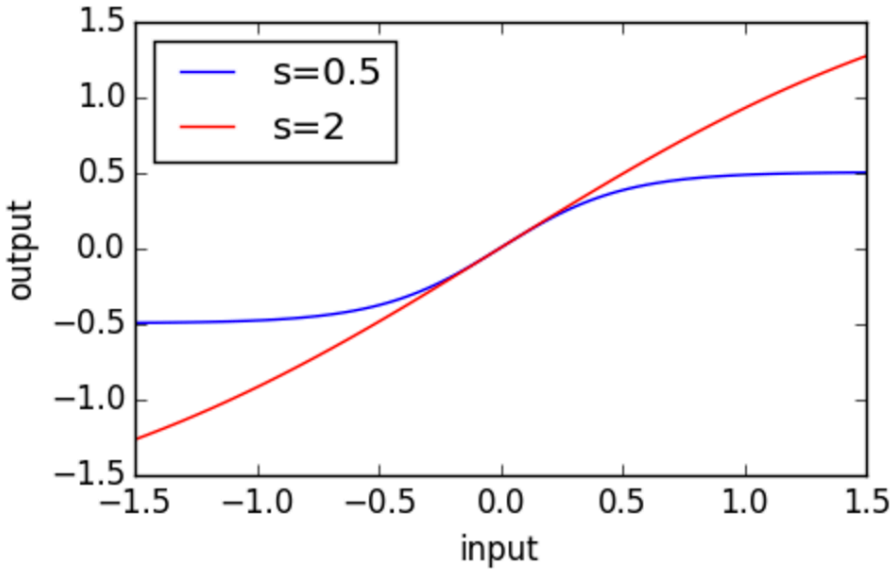
rank-D recurrent connectivity

random input

M-dimensional observation: $$\mathbf{r}_M^t = \mathbf{S}\mathbf{r}^t$$

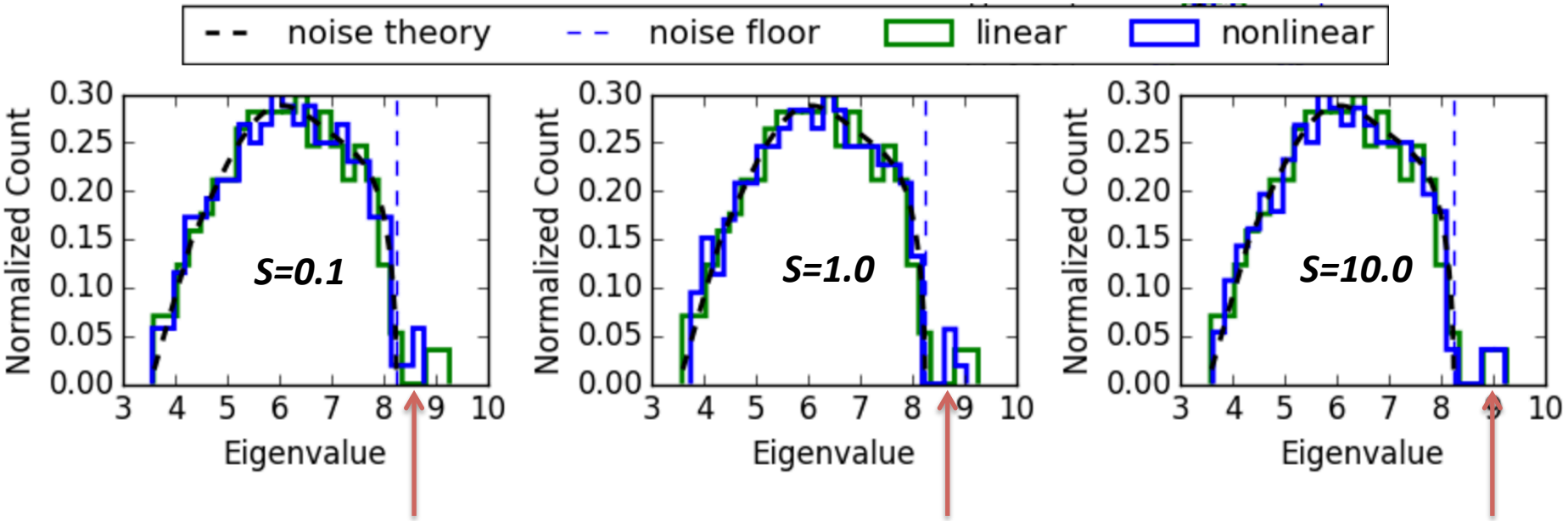
Random sampling operator

Nonlinearity scaling parameter ***s***: smooth transition between linear neuronal responses (large ***s***) and binary responses (small ***s***)



Compressive nonlinearity's effect on data's eigenvalue spectrum:

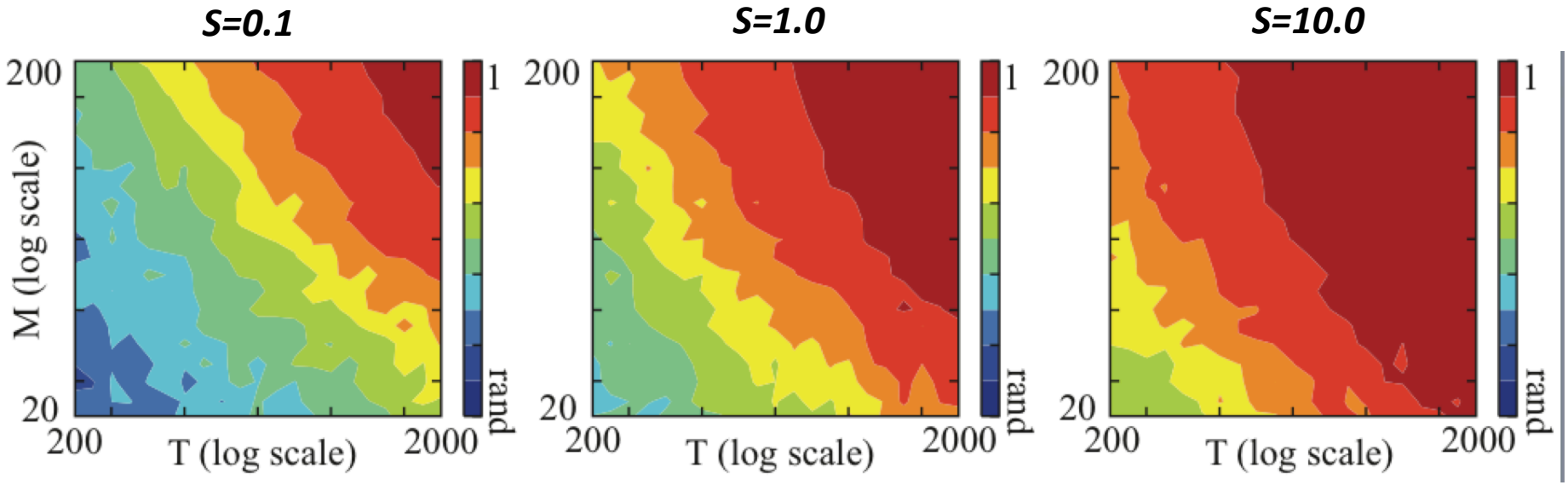
Simulated datasets with N=500, D=4, M=250, T=500 samples,
time constant for neuronal noise = 2.0; for network dynamics = 10.0
Random input's strength (standard deviation): into signal subspace = 3.0, into the rest = 1.0



A preferential squashing of the signal eigenvalues

Compressive nonlinearity's effect on the recovery of latent subspace

True signal subspace obtained empirically after simulating nonlinear networks for 20,000 time steps;
For each M, the true signal subspace is sub-sampled and re-orthogonalized to compute its overlap with the recovered subspace from only T steps of simulation;
The resulting overlaps are rescaled so that 0 => overlap between two random D-dimensional subspaces in M dimension, and 1 => complete overlap



Simulated datasets with N=500, D=4, M varied from 20 to 200, T varied from 200 to 2000
times constant of neuronal noise = 2.0, of network dynamics = 10.0
Random input's standard deviation: into signal subspace = 3.0, into the rest = 1.0

Outline of talks I, II and III

1. Oldies but goodies

    1. Models of single neurons: Hodgkin Huxley to Hopfield
    2. The Hopfield model
    3. The perceptron learning algorithm: memorization and generalization
    4. Unsupervised learning: PCA, ICA, Sparse Coding

2. High dimensional statistics: theory and experiment

    1. The best way to do regression in high dimensions (Replica theory)
    2. Recovering neural state space dynamics (Rand proj / Matrices / Free prob)
    3. Figuring out how neural circuits learn (Tensor decompositions)

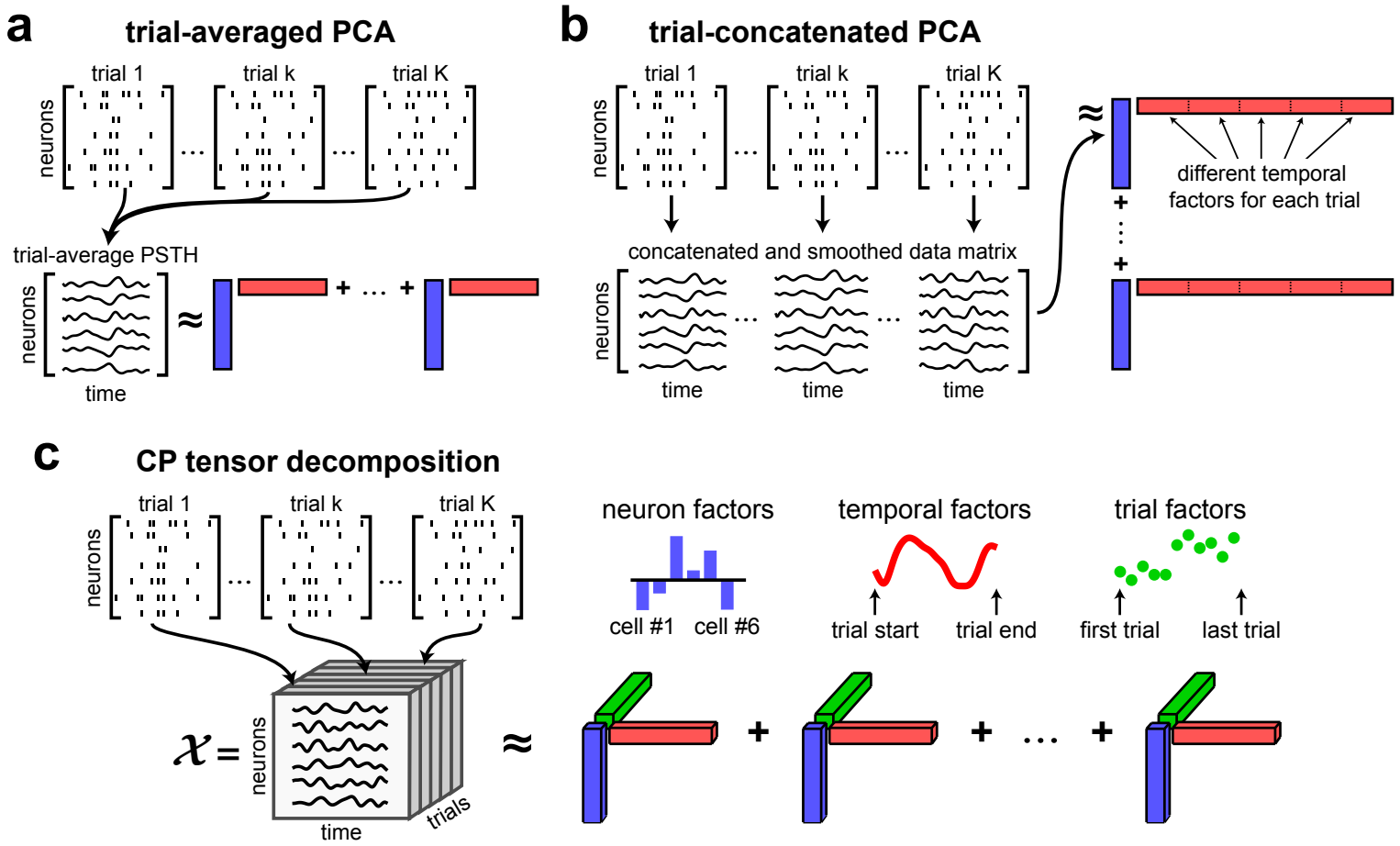
3. Deep learning: theory and practice

    1. Speeding up deep learning (Dynamic criticality)
    2. Error landscape of deep networks (Stat mech of random Gaussian fields)
    3. Deep generative models (Non-equilbrium thermodynamics)
    4. Expressive power of deep networks (Riemannian geometry and chaos theory)
    5. Application: deep models of the retina: the first step in seeing

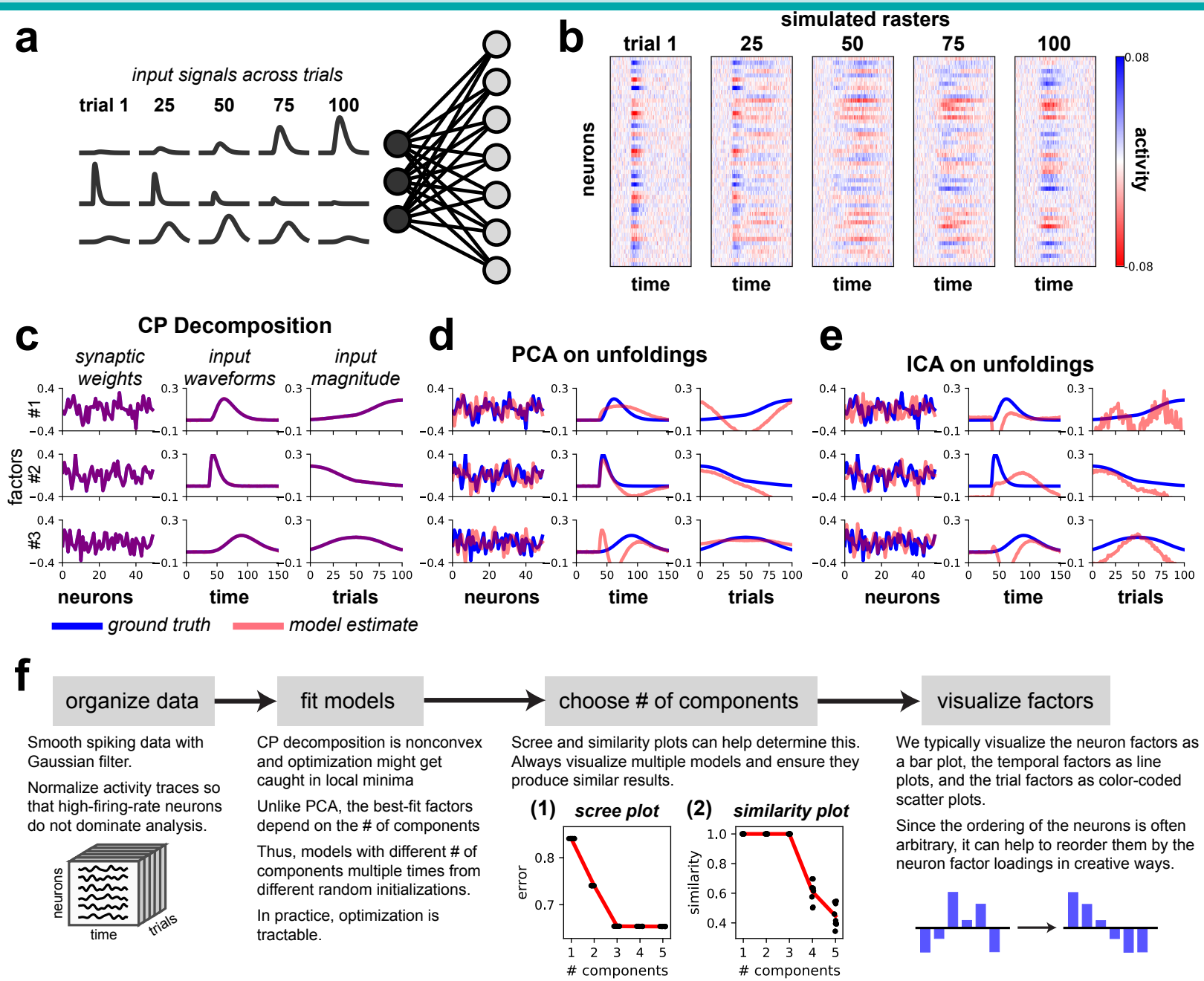
Tensor components analysis

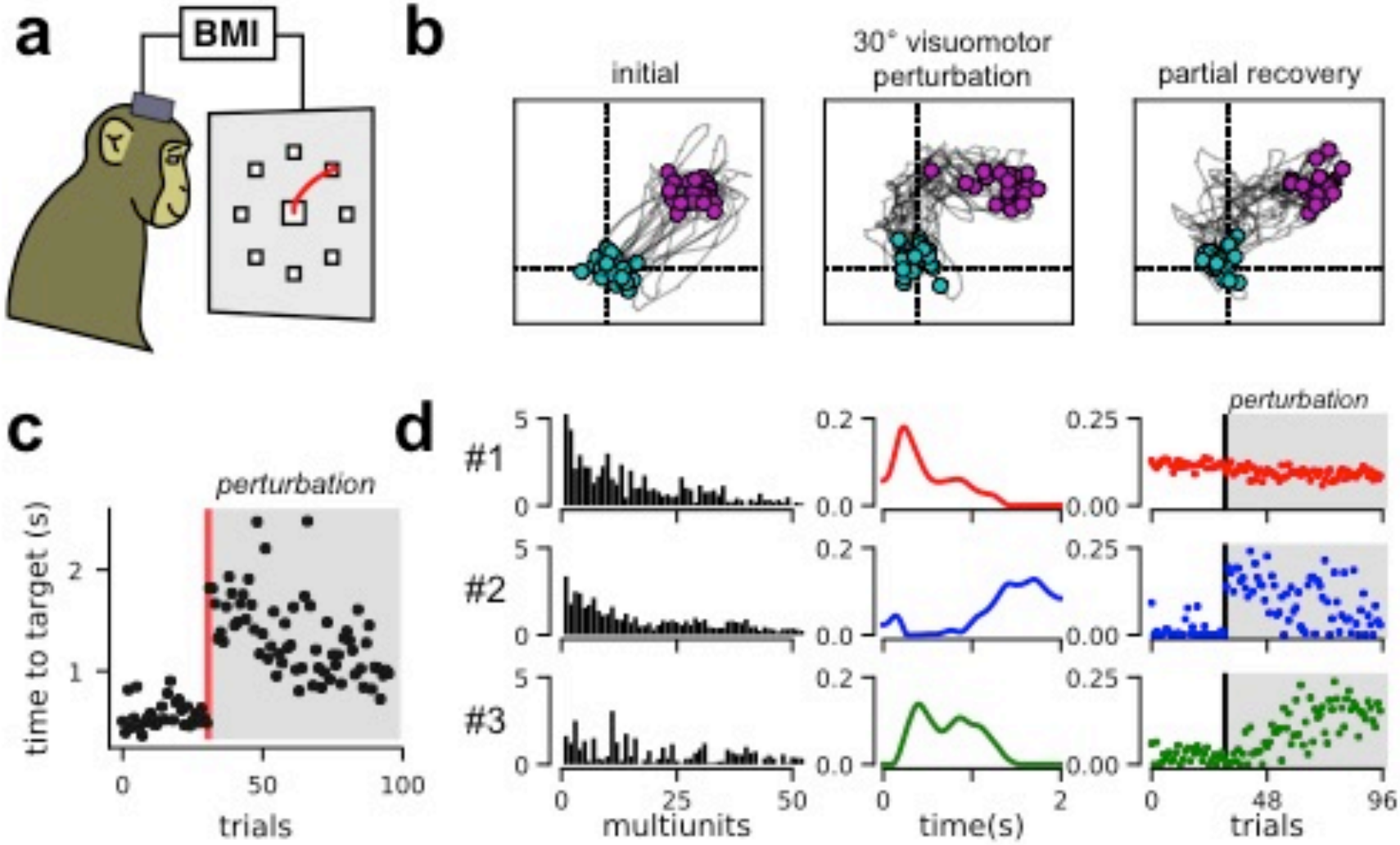
Alex Williams

Tensor components analysis

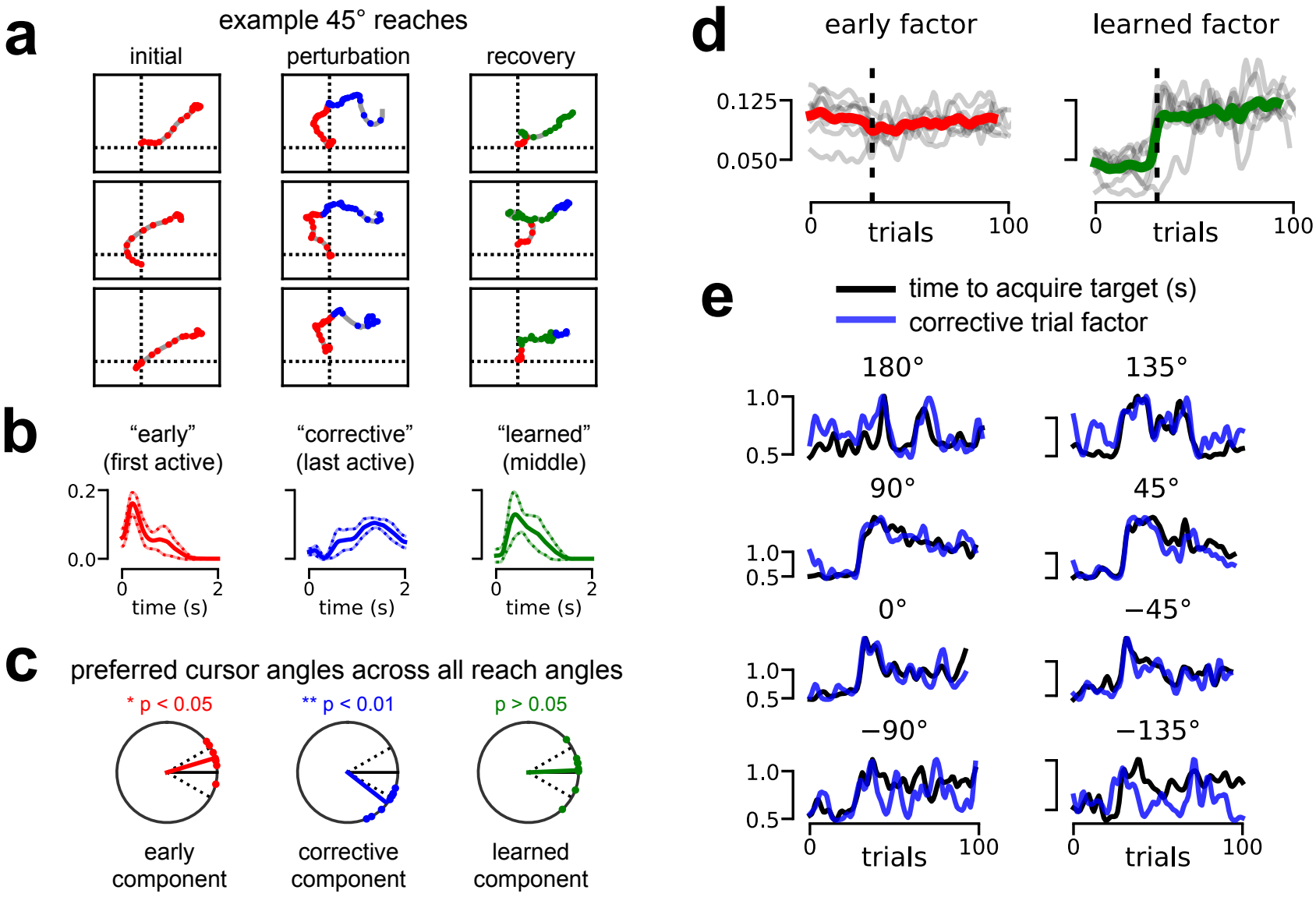


Tensor components analysis




a
BMI
b
initial
30° visuomotor perturbation
partial recovery
c
perturbation
time to target (s)
trials
d
#1
#2
#3
multiunits
time(s)
trials
perturbation

Tensor components analysis



Outline of talks I, II and III

1. Oldies but goodies

    1. Models of single neurons: Hodgkin Huxley to Hopfield
    2. The Hopfield model
    3. The perceptron learning algorithm: memorization and generalization
    4. Unsupervised learning: PCA, ICA, Sparse Coding

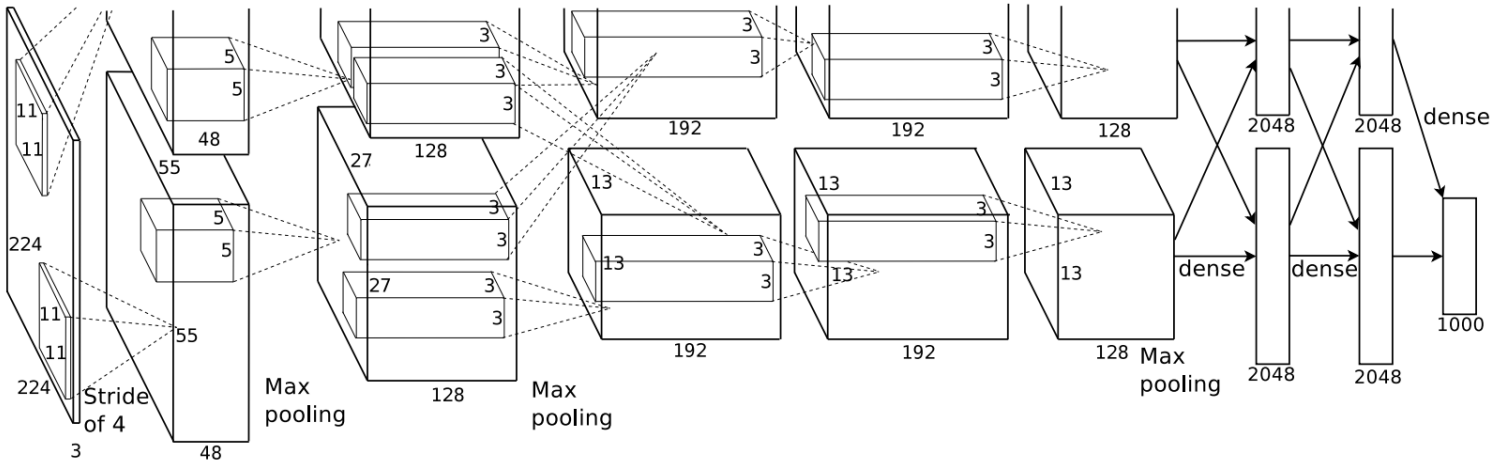
2. High dimensional statistics: theory and experiment

    1. The best way to do regression in high dimensions (Replica theory)
    2. Recovering neural state space dynamics (Rand proj / Matrices / Free prob)
    3. Figuring out how neural circuits learn (Tensor decompositions)

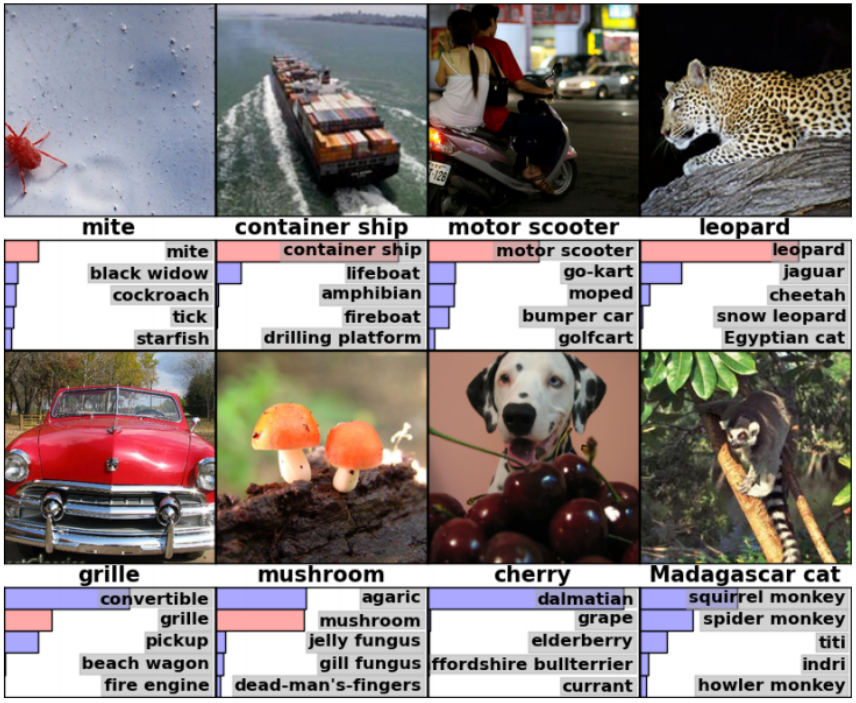
3. Deep learning: theory and practice

    1. Speeding up deep learning (Dynamic criticality)
    2. Error landscape of deep networks (Stat mech of random Gaussian fields)
    3. Deep generative models (Non-equilbrium thermodynamics)
    4. Expressive power of deep networks (Riemannian geometry and chaos theory)
    5. Application: deep models of the retina: the first step in seeing

An interesting artificial neural circuit for image classification



Alex Krizhevsky
Ilya Sutskever
Geoffrey E. Hinton
NIPS 2012

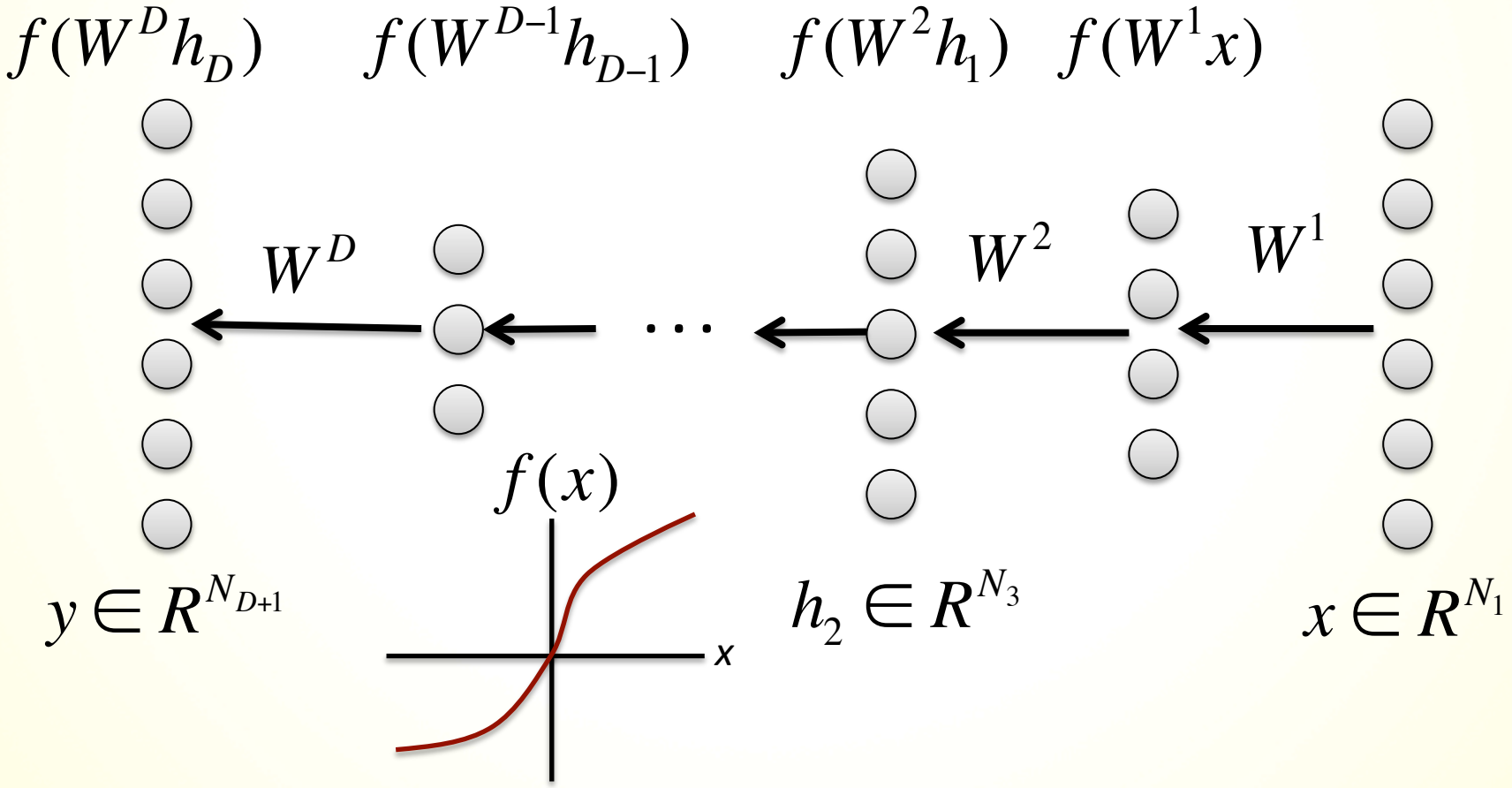


Towards a theory of deep learning dynamics

- The dynamics of learning in deep networks is non-trivial – i.e. plateaus and sudden transitions to better performance
- How does training time scale with depth?
- How should the learning rate scale with depth?
- How do different weight initializations impact learning speed?
- We will find that weight initializations with *critical dynamics* can aid deep learning and generalization.

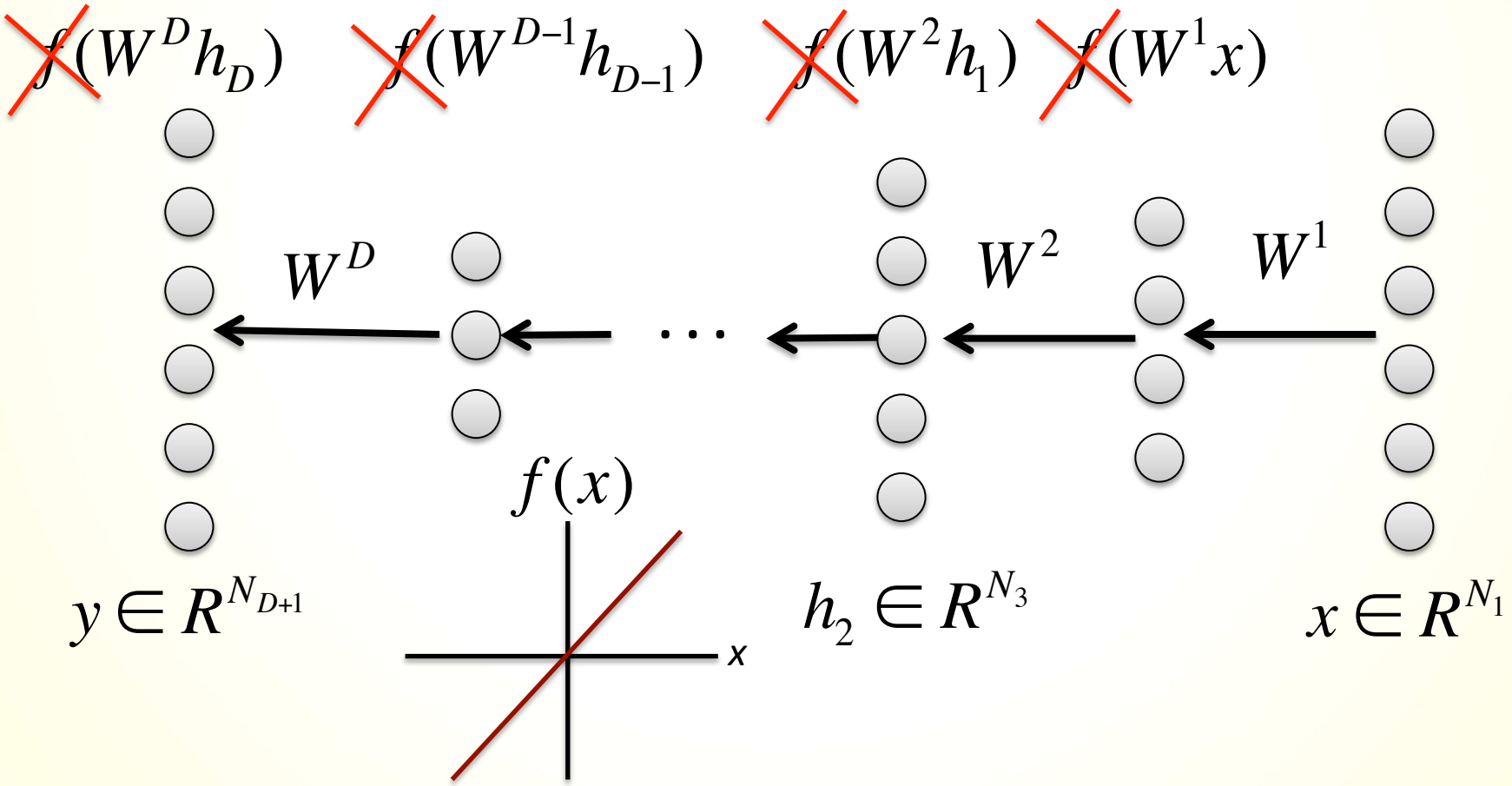
Deep network

- Little hope for a complete theory with arbitrary nonlinearities



Deep *linear* network

- Use a deep *linear* network as a starting point



Deep *linear* network

- Input-output map: Always linear

$$y = \left(\prod_{i=1}^{D} W^i \right) x \equiv W^{tot} x$$

- Gradient descent dynamics: Nonlinear; coupled; nonconvex

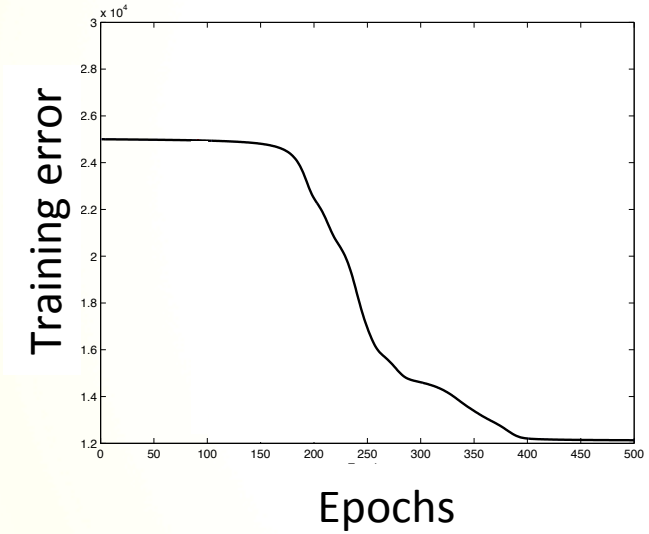
$$\Delta W^l = \lambda \sum_{\mu=1}^{P} \left(\prod_{i=l+1}^{D} W^i \right)^T \left[y^\mu x^{\mu T} - \left(\prod_{i=1}^{D} W^i \right) x^\mu x^{\mu T} \right] \left(\prod_{i=1}^{l-1} W^i \right)^T$$

$$l = 1, \cdots, D$$

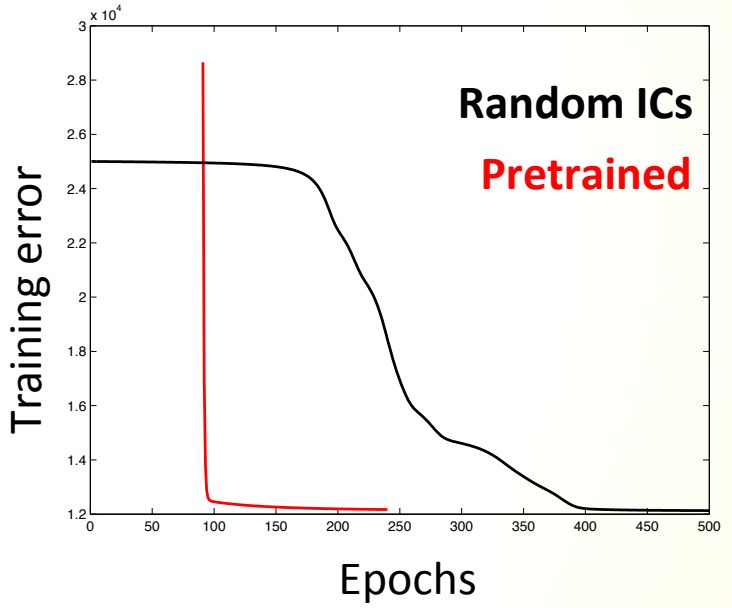
- Useful for studying *learning dynamics*, not representation power.

Nontrivial learning dynamics

Plateaus and sudden transitions

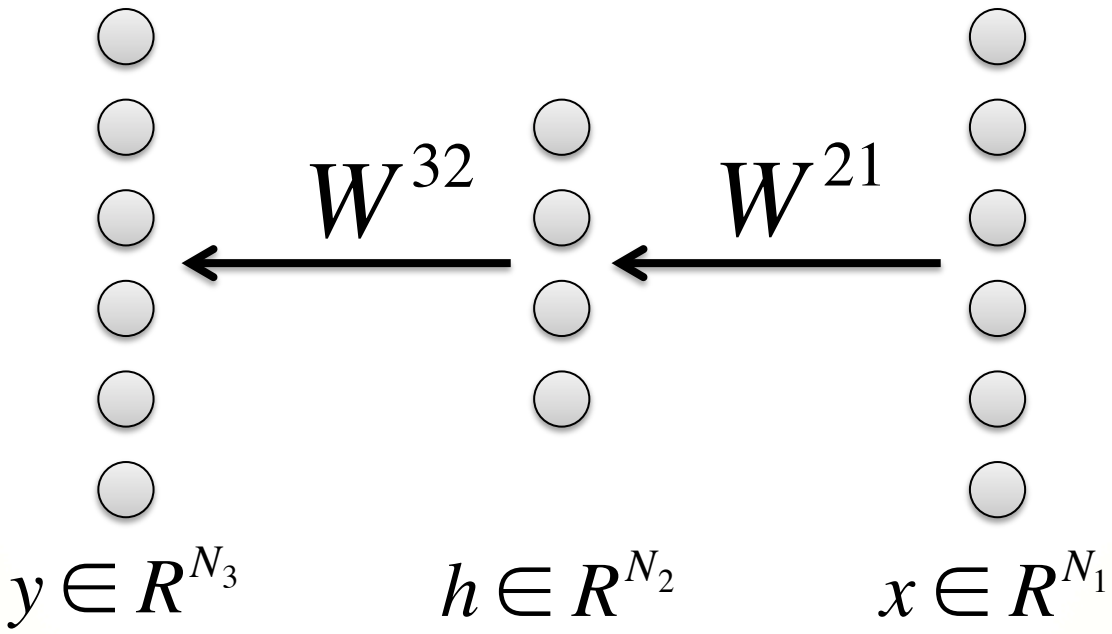


Faster convergence from pretrained initial conditions



- Build intuitions for nonlinear case by analyzing linear case

Three layer dynamics



Problem formulation

- Network trained on patterns $\{x^\mu, y^\mu\}, \mu = 1, \ldots, P.$
- Batch gradient descent on squared error $\|Y - W^{32}W^{21}X\|_F^2$
- Dynamics

$$\tau \frac{d}{dt} W^{21} = W^{32^T} \left(\Sigma^{31} - W^{32} W^{21} \Sigma^{11} \right)$$

$$\tau \frac{d}{dt} W^{32} = \left(\Sigma^{31} - W^{32} W^{21} \Sigma^{11} \right) W^{21^T}$$

Input correlations: $\Sigma^{11} \equiv E[xx^T] = I$

Input-output correlations: $\Sigma^{31} \equiv E[yx^T]$

(see paper for more general input correlations)

Analytic learning trajectory

SVD of input-output correlations:

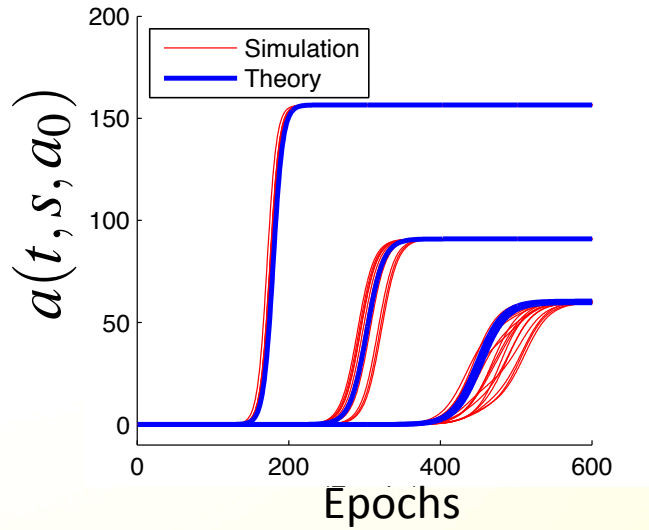
$$\Sigma^{31} = U^{33} S^{31} V^{11\,T} = \sum_{\alpha=1}^{N_1} s_\alpha u^\alpha v^{\alpha T}$$

τ	1/Learning rate
s	Singular value
a_0	Initial mode strength

Network input-output map:

$$W^{32}(t) W^{21}(t) = \sum_{\alpha=1}^{N_2} a(t, s_\alpha, a_\alpha^0)\, u^\alpha v^{\alpha T} \quad \text{where} \quad a(t,s,a_0) = \frac{s e^{2st/\tau}}{e^{2st/\tau} - 1 + s/a_0}$$

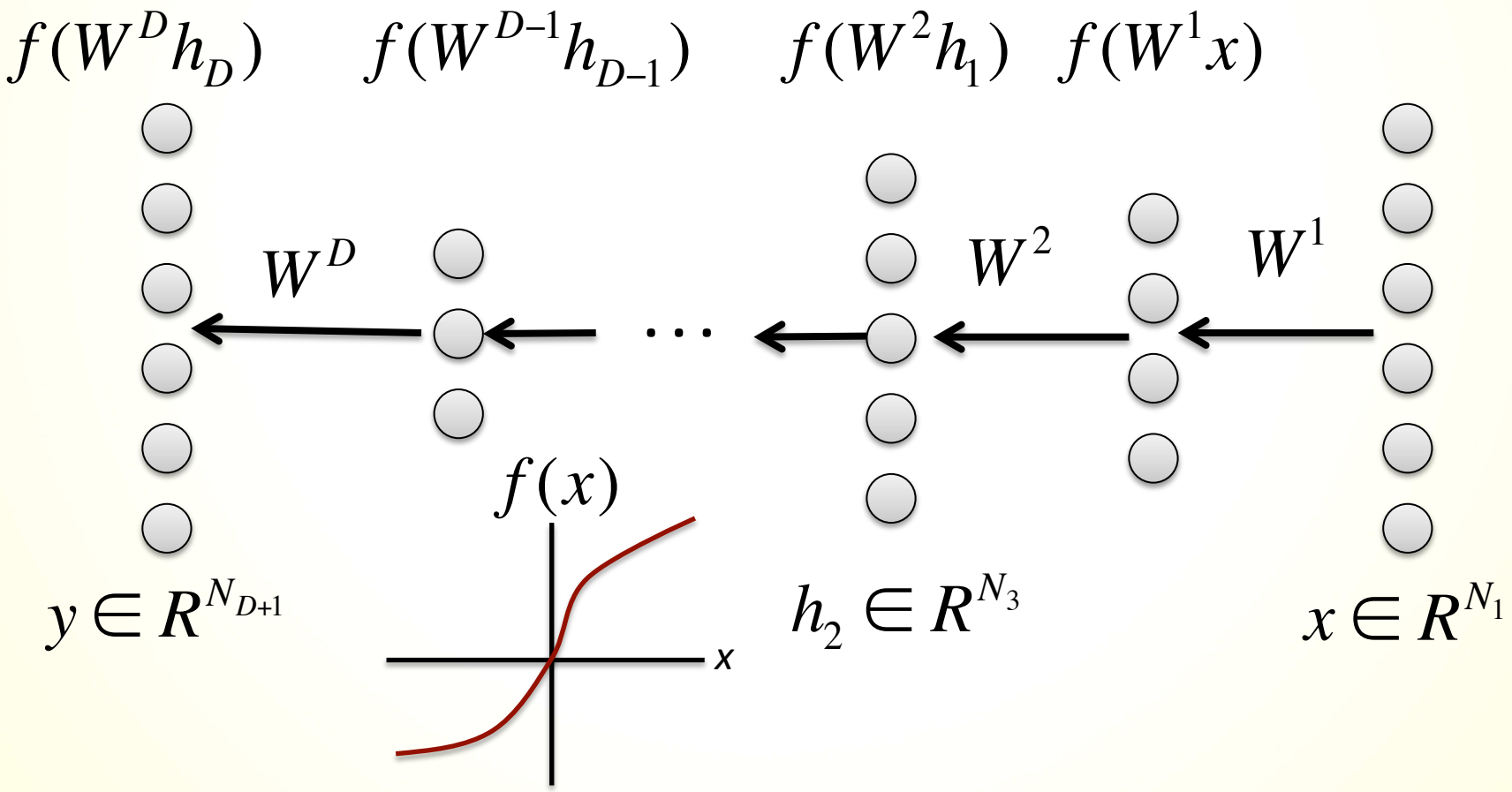
- Starting from decoupled initial conditions.
- Each 'connectivity mode' evolves independently
- Singular value s learned at time O(1/s)



Saxe, McCelland, Ganguli, ICLR, 2014

Deeper network learning dynamics

- Jacobian that back-propagates gradients can explode or decay



Deeper networks

- Can generalize to arbitrary depth network

- Each effective singular value *a* evolves independently

$$\tau \frac{d}{dt} a = (N_l - 1) a^{2-2/(N_l-1)} (s - a)$$

τ	1/Learning rate
s	Singular value
N_l	# layers

- In deep networks, combined gradient is $O(N_l/\tau)$

$$a = \prod_{i=1}^{N_l - 1} W_i$$

Deep linear learning speed

- Intuition (see paper for details):
  - Gradient norm $O(N_l)$
  - Learning rate $O(1/N_l)$ (N_l = # layers)
  - Learning time $O(1)$
- Deep learning *can be fast* with the right ICs.

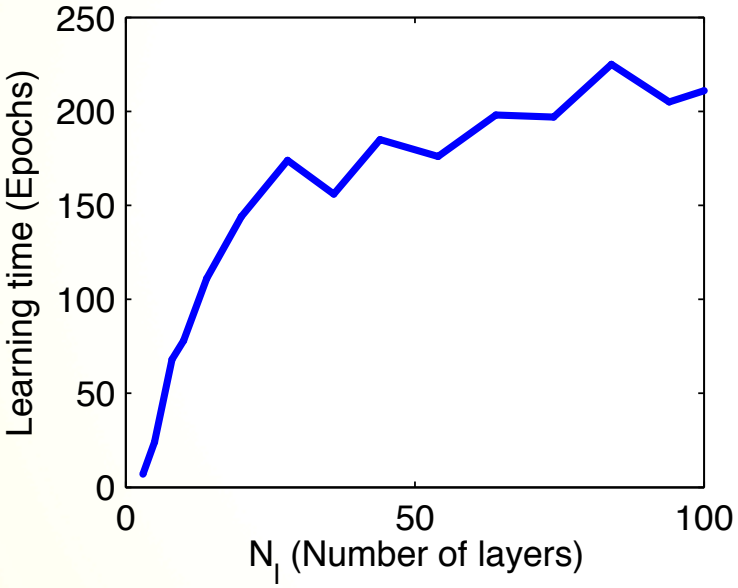
Saxe, McClelland, Ganguli ICLR 2014

MNIST learning speeds

- Trained deep *linear* nets on MNIST
- Depths ranging from 3 to 100
- 1000 hidden units/layer (overcomplete)
- Decoupled initial conditions with fixed initial mode strength
- Batch gradient descent on squared error
- Optimized learning rates for each depth
- Calculated epoch at which error falls below fixed threshold

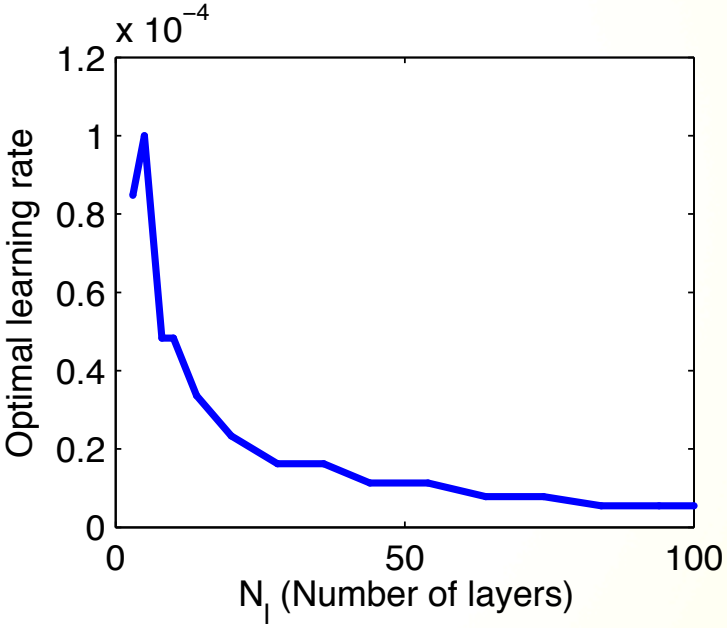
MNIST depth dependence

Time to criterion



Depth

Optimal learning rate



Depth

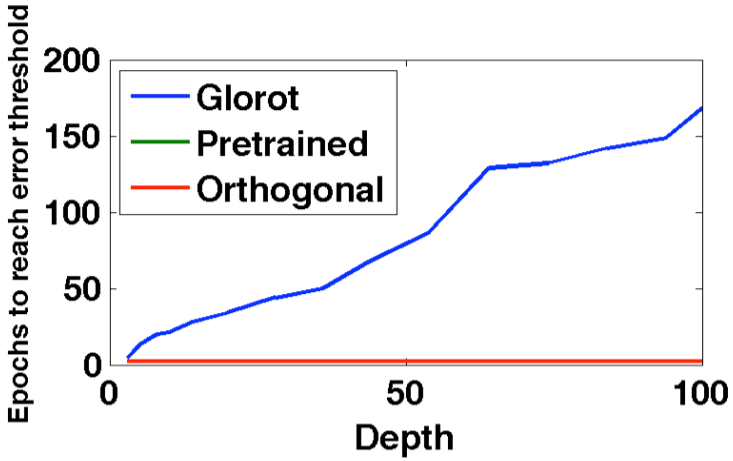
Deep linear networks

- Deep learning *can be fast* with decoupled ICs and O(1) initial mode strength. **How to find these?**
- Answer: Pre-training and random orthogonal initializations can find these special initial conditions that allow depth independent training times!!
- But scaled random Gaussian initial conditions on weights cannot.

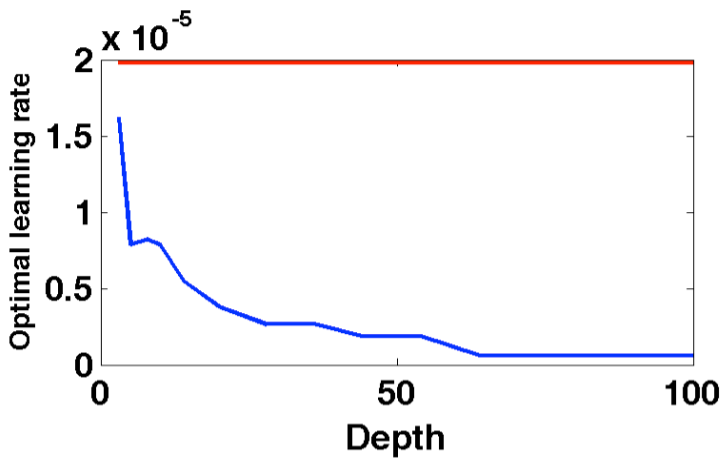
Depth-independent training time

- Deep *linear* networks on MNIST
- Scaled random Gaussian initialization (Glorot & Bengio, 2010)

Time to criterion



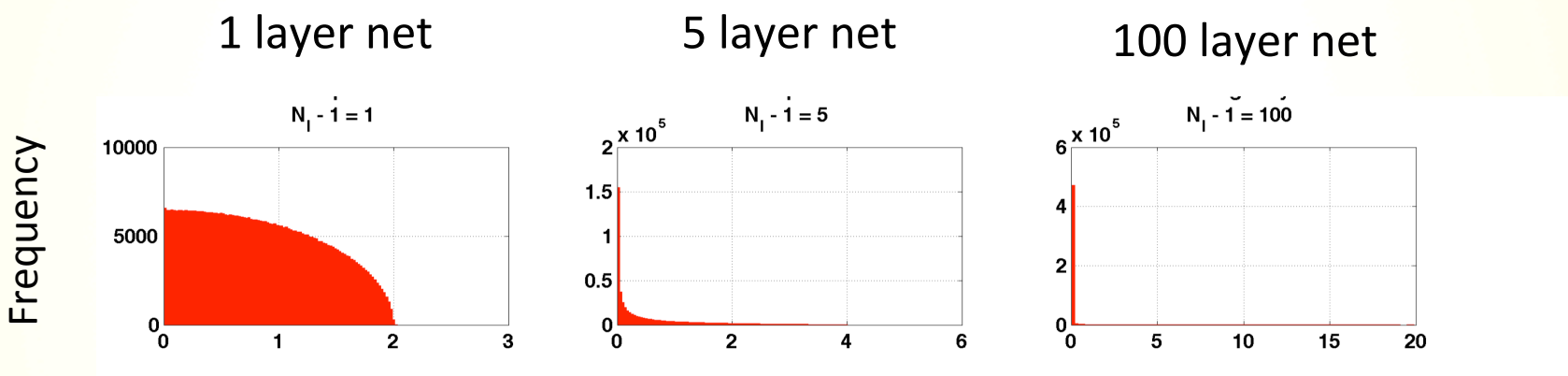
Optimal learning rate



- Pretrained and orthogonal have fast **depth-independent** training times!

Random vs orthogonal

- Gaussian preserves norm of random vector *on average*

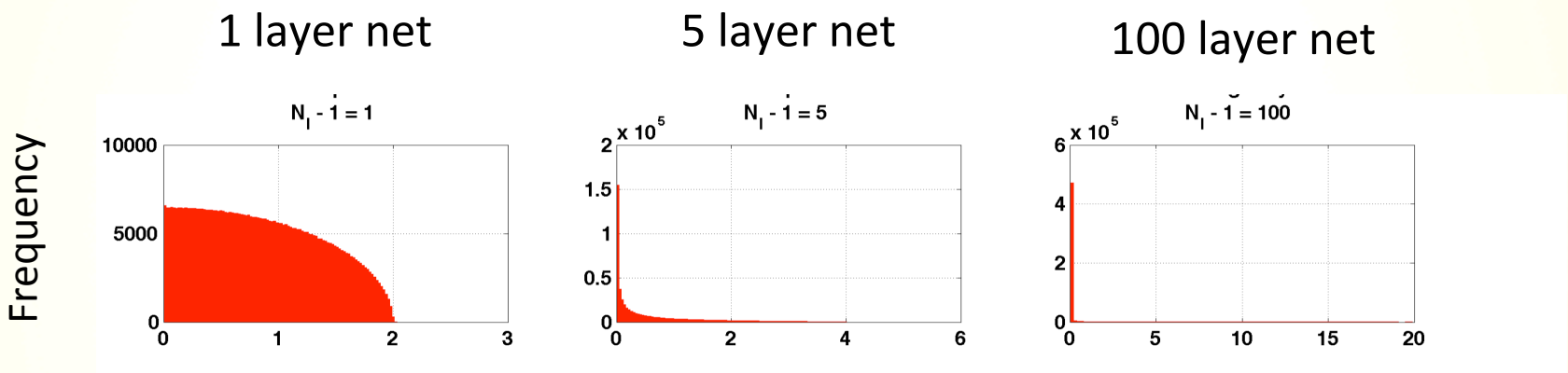


Singular values of $W^{tot} = \prod_{i=1}^{N_l - 1} W^i$

- *Attenuates* on subspace of high dimension
- *Amplifies* on subspace of low dimension

Random vs orthogonal

- Glorot preserves norm of random vector *on average*



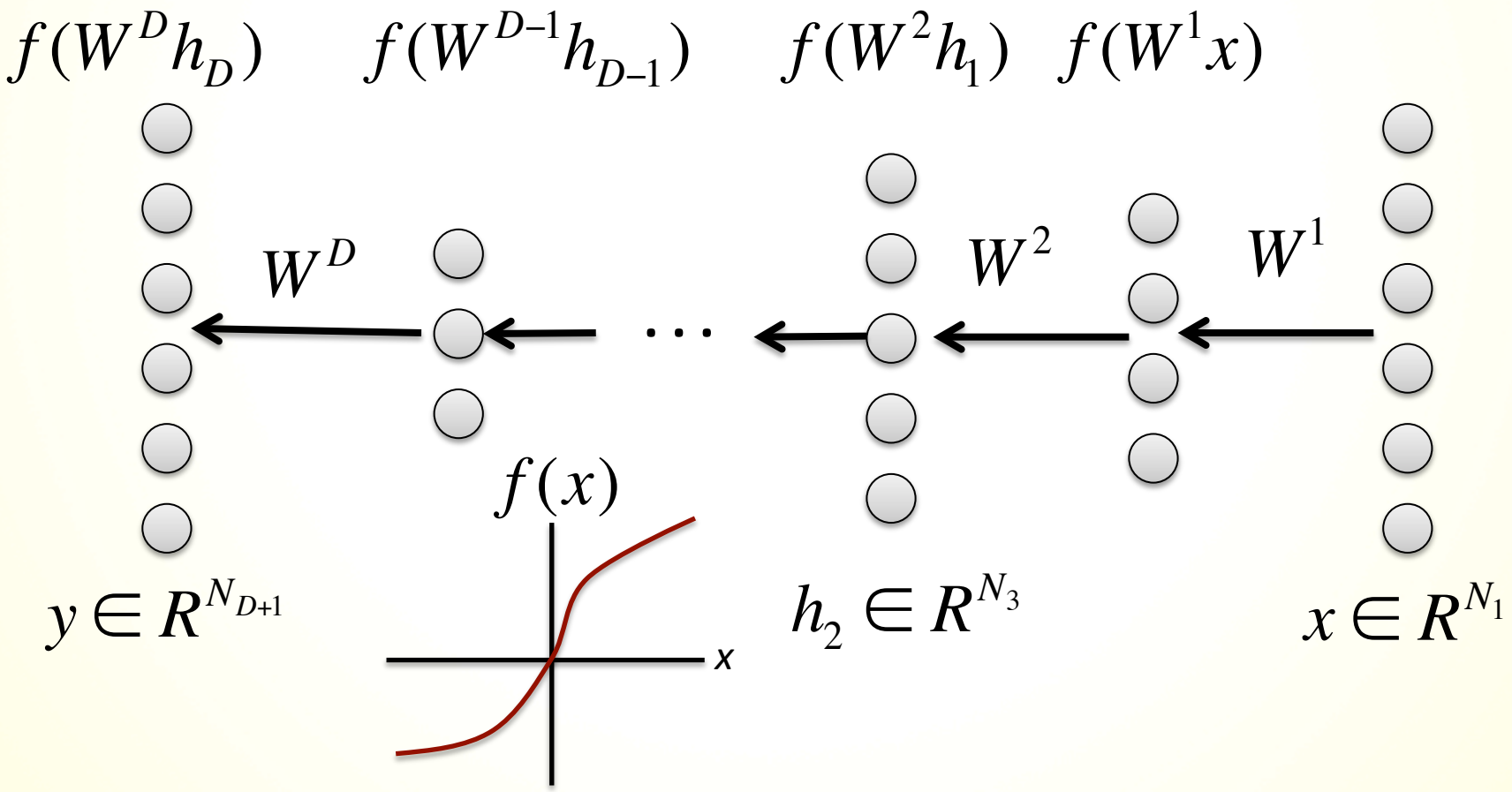
Singular values of $W^{tot} = \prod_{i=1}^{N_l - 1} W^i$

- Orthogonal preserves norm of all vectors *exactly*

All singular values of $W^{tot} = 1$

Deeper network learning dynamics

- Jacobian that back-propagates gradients can explode or decay



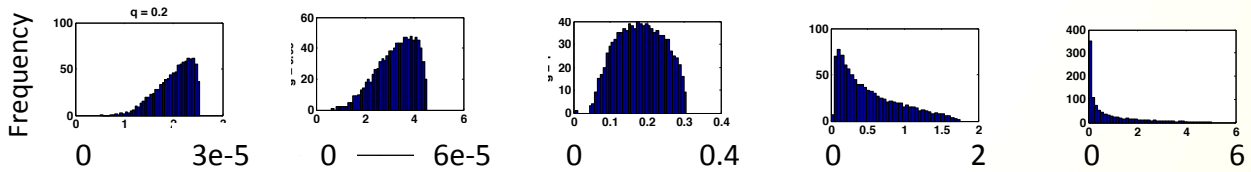
Extensive Criticality yields Dynamical Isometry in *nonlinear* nets

Suggests initialization for *nonlinear* nets

- near-isometry on subspace of large dimension
- Singular values of *end-to-end* Jacobian $J_{ij}^{N_l,1}(x^{N_l}) \equiv \left.\frac{\partial x_i^{N_l}}{\partial x_j^1}\right|_{x^{N_l}}$ concentrated around 1.

Scale orthogonal matrices by gain *g* to counteract contractive nonlinearity

Singular values of *J*



Gain

g=0.9 *g=0.95* *g=1* *g=1.05* *g=1.1*

Just beyond *edge of chaos (g>1)* may be good initialization

Dynamic Isometry Initialization

- g>1 speeds up **30 layer nonlinear** nets
  - Tanh network, softmax output, 500 units/layer
  - No regularization (weight decay, sparsity, dropout, etc)

MNIST Classification error, epoch 1500	**Train Error (%)**	**Test Error (%)**
Gaussian (g=1, random)	2.3	3.4
g=1.1, random	1.5	3.0
g=1, orthogonal	2.8	3.5
Dynamic Isometry (g=1.1, orthogonal)	**0.095**	**2.1**

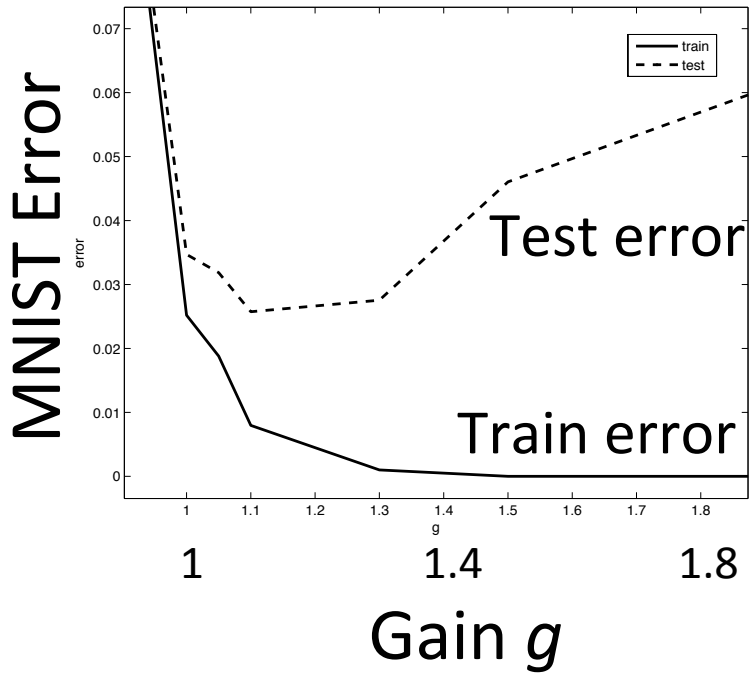
- Dynamic isometry reduces test error by 1.4% pts

Summary

- Deep linear nets have **nontrivial nonlinear learning dynamics.**
- Learning time inversely proportional to strength of input-output correlations.
- With the right initial weight conditions, number of training epochs can remain finite as depth increases.
- Dynamically critical networks just beyond the edge of chaos enjoy **depth-independent** learning times.

Beyond learning: criticality and generalization

- Deep networks + large gain factor g train exceptionally quickly
- But large g incurs heavy cost in generalization performance



- Suggests small initial weights regularize towards smoother functions

Outline of talks I, II and III

1. Oldies but goodies

    1. Models of single neurons: Hodgkin Huxley to Hopfield
    2. The Hopfield model
    3. The perceptron learning algorithm: memorization and generalization
    4. Unsupervised learning: PCA, ICA, Sparse Coding

2. High dimensional statistics: theory and experiment

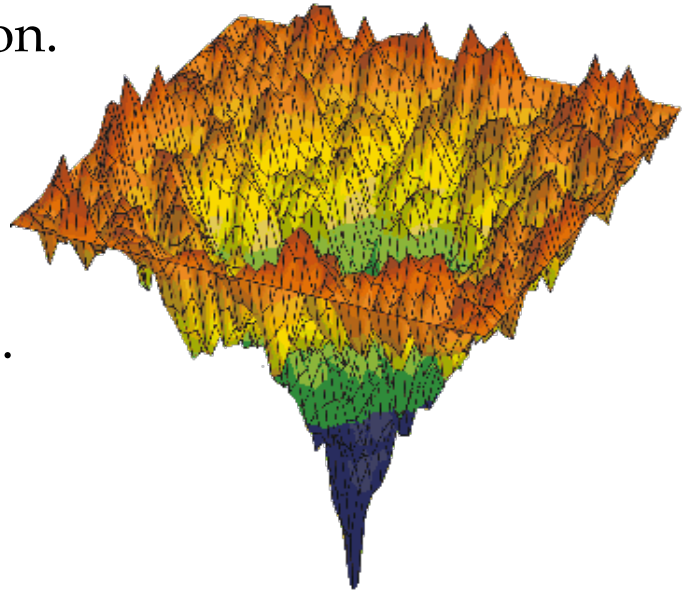
    1. The best way to do regression in high dimensions (Replica theory)
    2. Recovering neural state space dynamics (Rand proj / Matrices / Free prob)
    3. Figuring out how neural circuits learn (Tensor decompositions)

3. Deep learning: theory and practice

    1. Speeding up deep learning (Dynamic criticality)
    2. Error landscape of deep networks (Stat mech of random Gaussian fields)
    3. Deep generative models (Non-equilbrium thermodynamics)
    4. Expressive power of deep networks (Riemannian geometry and chaos theory)
    5. Application: deep models of the retina: the first step in seeing

High dimensional nonconvex optimization

It is often thought that local minima at high error stand as as a major impediment to non-convex optimization.



In random non-convex error surfaces over high dimensional spaces, local minima at high error are exponentially rare in the dimensionality.

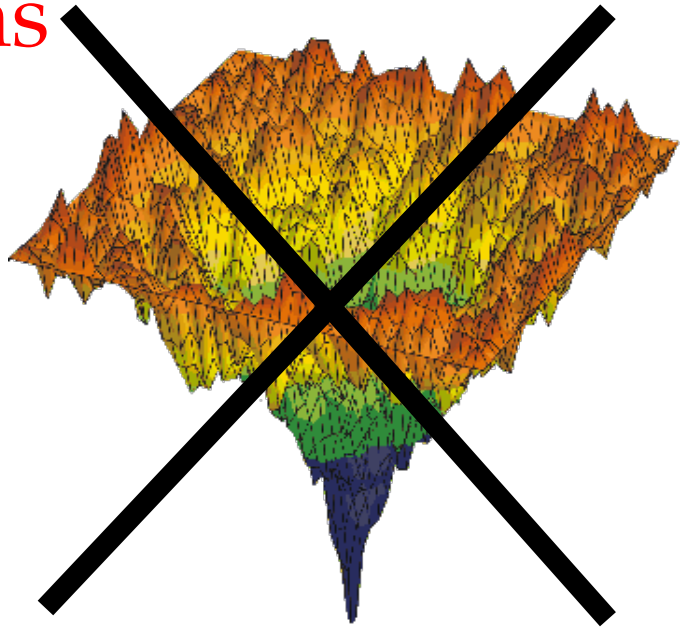
Instead saddle points proliferate.

We developed an algorithm that rapidly escapes saddle points in high dimensional spaces.

Identifying and attacking the saddle point problem in high dimensional non-convex optimization. Y. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, Y. Bengio. NIPS 2014.

The loss surfaces of multilayer networks. A. Chormanska, M. Henaff, M. Mathieu, G. Ben Arous, Y. LecCun, AISTATS 2015.

General properties of error landscapes in high dimensions



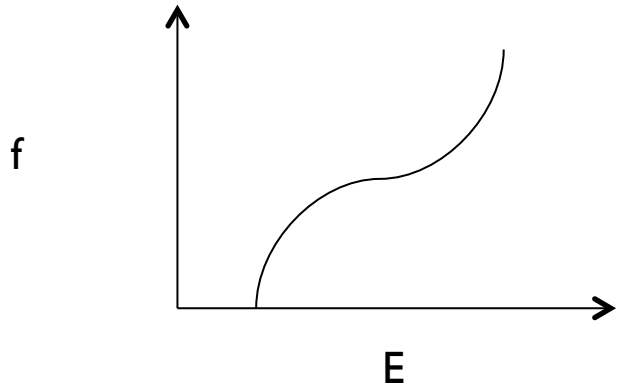
From statistical physics:

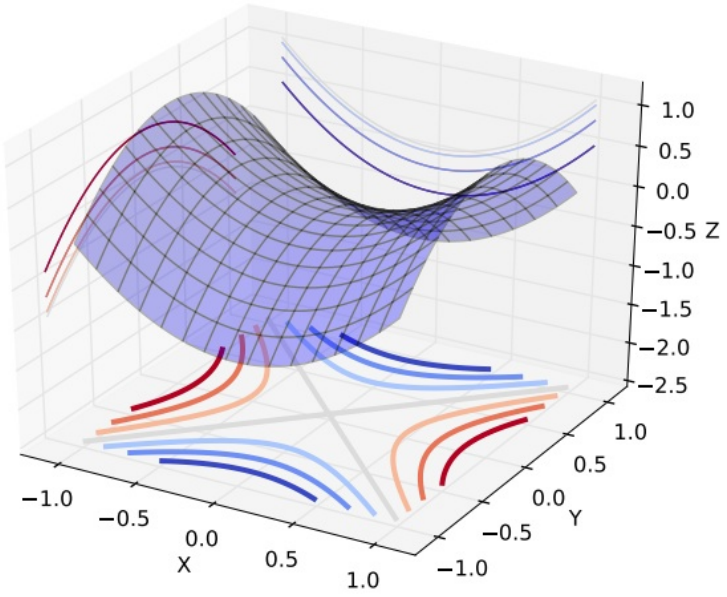
Consider a random Gaussian error landscape over N variables.

Let x be a critical point.

Let E be its error level.

Let f be the fraction of negative curvature directions.

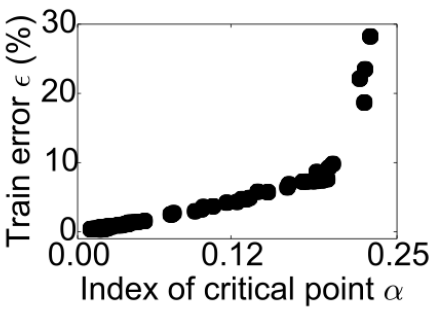


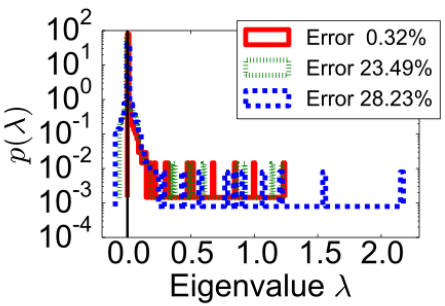


Bray and Dean, Physical Review Letters, 2007

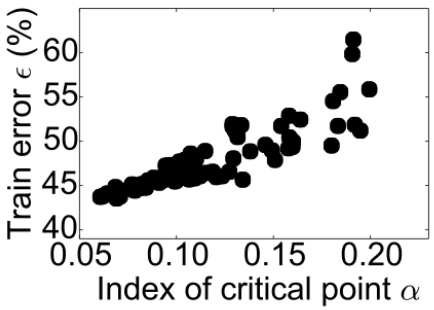
Properties of Error Landscapes on the Synaptic Weight Space of a Deep Neural Net

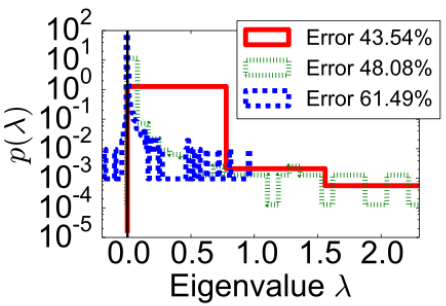
MNIST





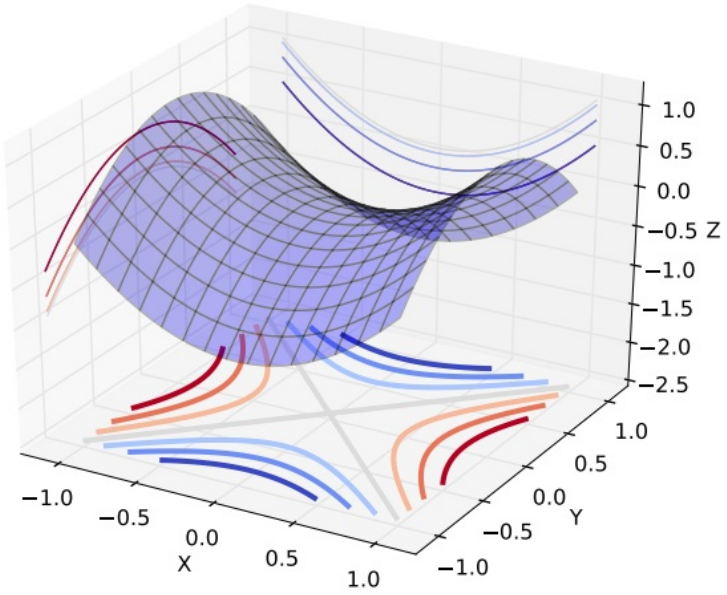
CIFAR-10





Qualitatively consistent with the
statistical physics theory of random error landscapes

How to descend saddle points



Newton's Method

$$\Delta x = -H^{-1} \nabla f(x)$$

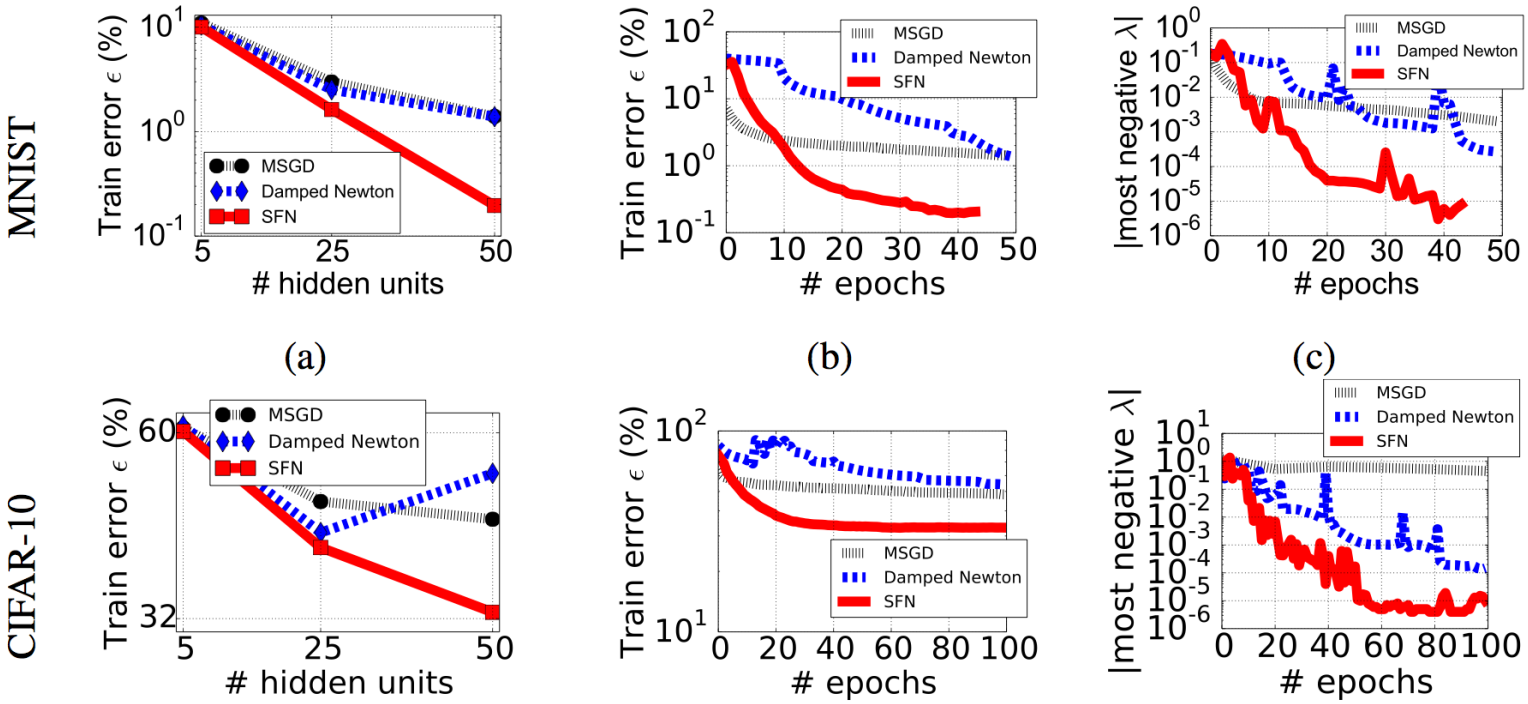
Saddle Free Newton's Method

$$\Delta x = -|H|^{-1} \nabla f(x)$$

Intuition: saddle points attract Newton's method, but repel saddle free Newton's method.

Derivation: minimize a linear approximation to f(x) within a trust region in which the linear and quadratic approximations agree

Performance of saddle free Newton in learning deep neural networks.



SFN out-performs

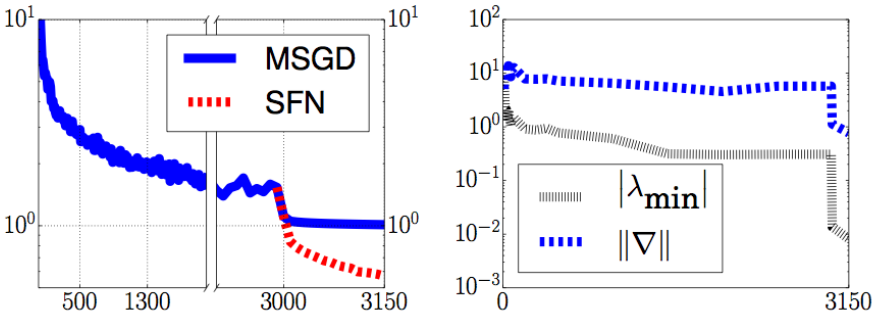
(1) minibatch stochastic gradient descent and

(2) damped Newton's method

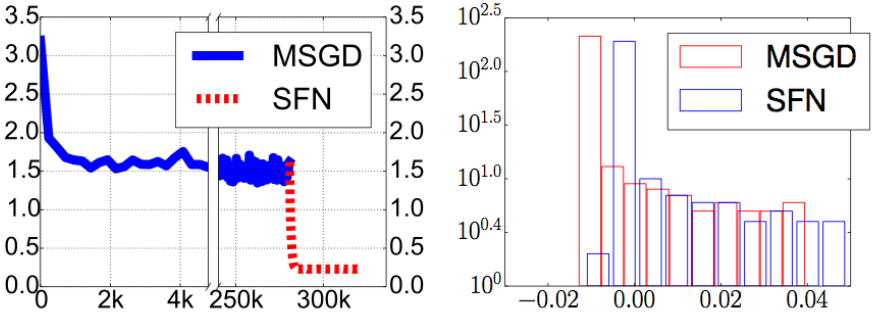
The performance advantage increases with the problem dimensionality.

Performance of saddle free Newton in learning deep neural networks.

Deep Autoencoder



Recurrent Neural Network



When stochastic gradient descent appears to plateau, switching to saddle Free newton escapes the plateau.

Outline of talks I, II and III

1. Oldies but goodies

    1. Models of single neurons: Hodgkin Huxley to Hopfield
    2. The Hopfield model
    3. The perceptron learning algorithm: memorization and generalization
    4. Unsupervised learning: PCA, ICA, Sparse Coding

2. High dimensional statistics: theory and experiment

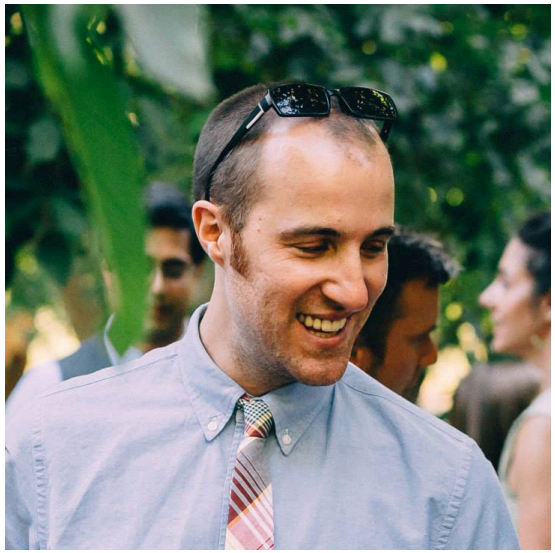
    1. The best way to do regression in high dimensions (Replica theory)
    2. Recovering neural state space dynamics (Rand proj / Matrices / Free prob)
    3. Figuring out how neural circuits learn (Tensor decompositions)

3. Deep learning: theory and practice

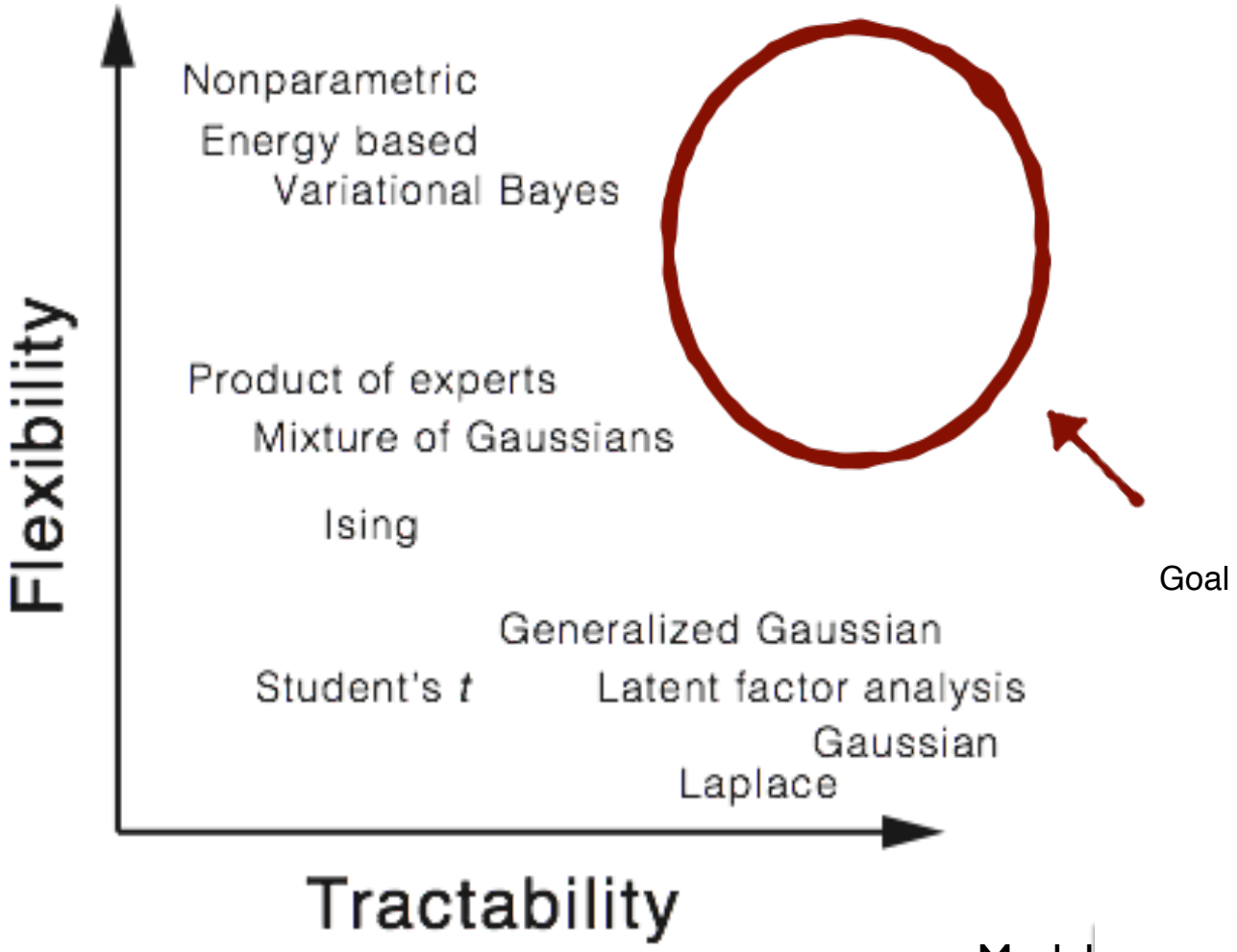
    1. Speeding up deep learning (Dynamic criticality)
    2. Error landscape of deep networks (Stat mech of random Gaussian fields)
    3. Deep generative models (Non-equilbrium thermodynamics)
    4. Expressive power of deep networks (Riemannian geometry and chaos theory)
    5. Application: deep models of the retina: the first step in seeing

Modeling Complex Data by ReversingTime

with Jascha Sohl-Dickstein
Eric Weiss, Niru Maheswaranathan



Flexibility-Tractability Tradeoff in Probabilistic Models



stein

Model

Achieving Flexibility and Tractability

- Physical motivation
  - Destroy structure in data through a diffusive process.
  - Carefully record the destruction.
  - Use deep networks to **reverse time and create structure from noise.**
- Inspired by recent results in non-equilibrium statistical mechanics which show that entropy can transiently decrease for short time scales (violations of second law)

stein Modeli

Physical Intuition: Destruction of Structure through Diffusion

- Dye density represents probability density
- Goal: Learn structure of probability density
- Observation: Diffusion destroys structure

Data distribution → Uniform distribution

stein

Model

Physical Intuition: Recover Structure by Reversing Time

- What if we could reverse this process?
- Recover data distribution by starting from uniform distribution and running a new type of reverse dynamics (using a trained deep network)

Data distribution

Uniform distribution

stein

Model

Physical Intuition: Recover Structure by Reversing Time

- What if we could reverse time?
- Recover data distribution by starting from uniform distribution and running dynamics backwards (using a trained deep network)

Data distribution

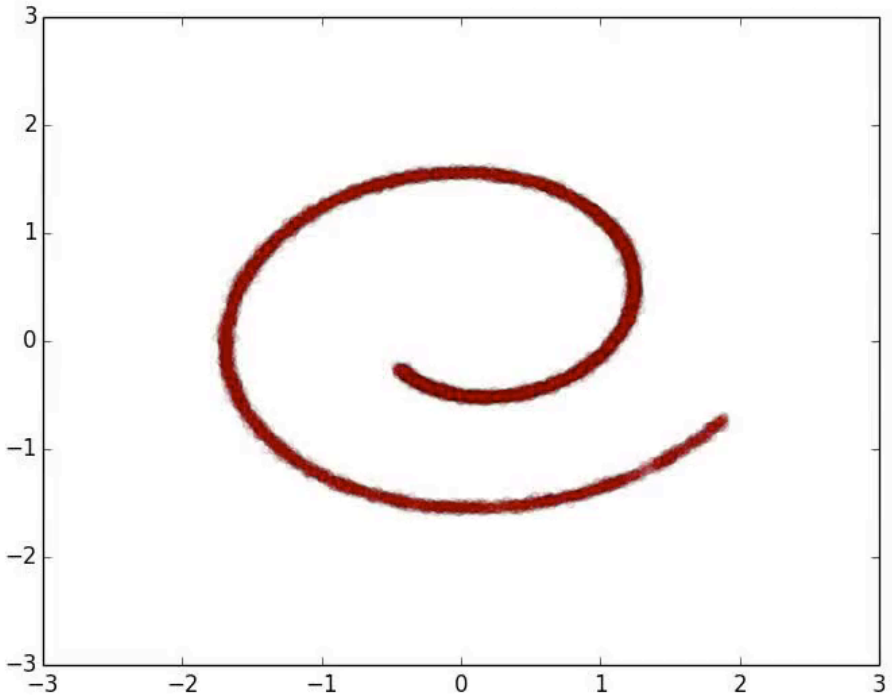
Uniform distribution

stein

Model

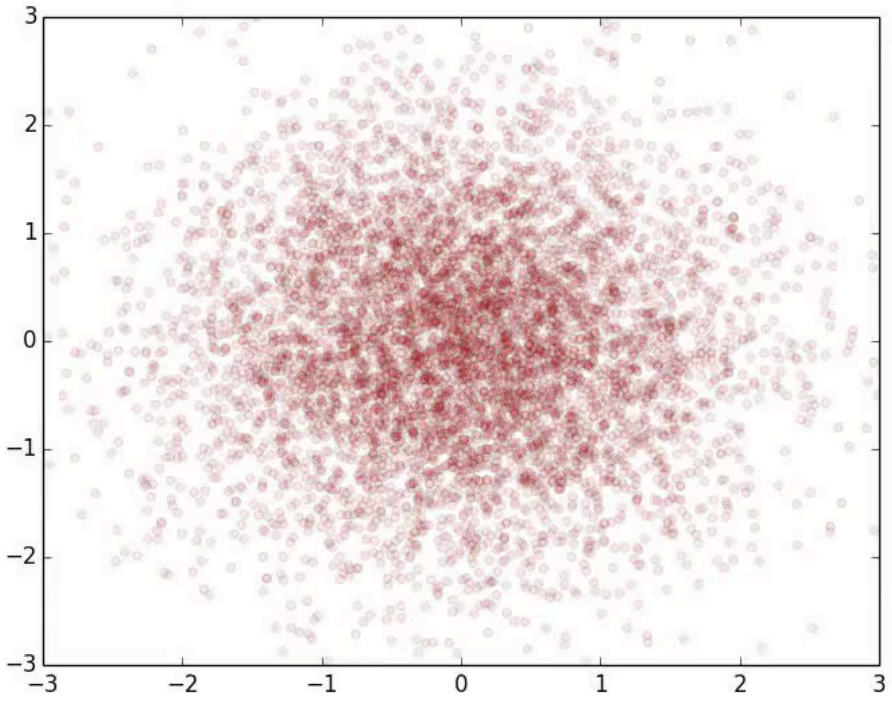
Swiss Roll

- Forward diffusion process
  - Start at data
  - Run Gaussian diffusion until samples become Gaussian blob

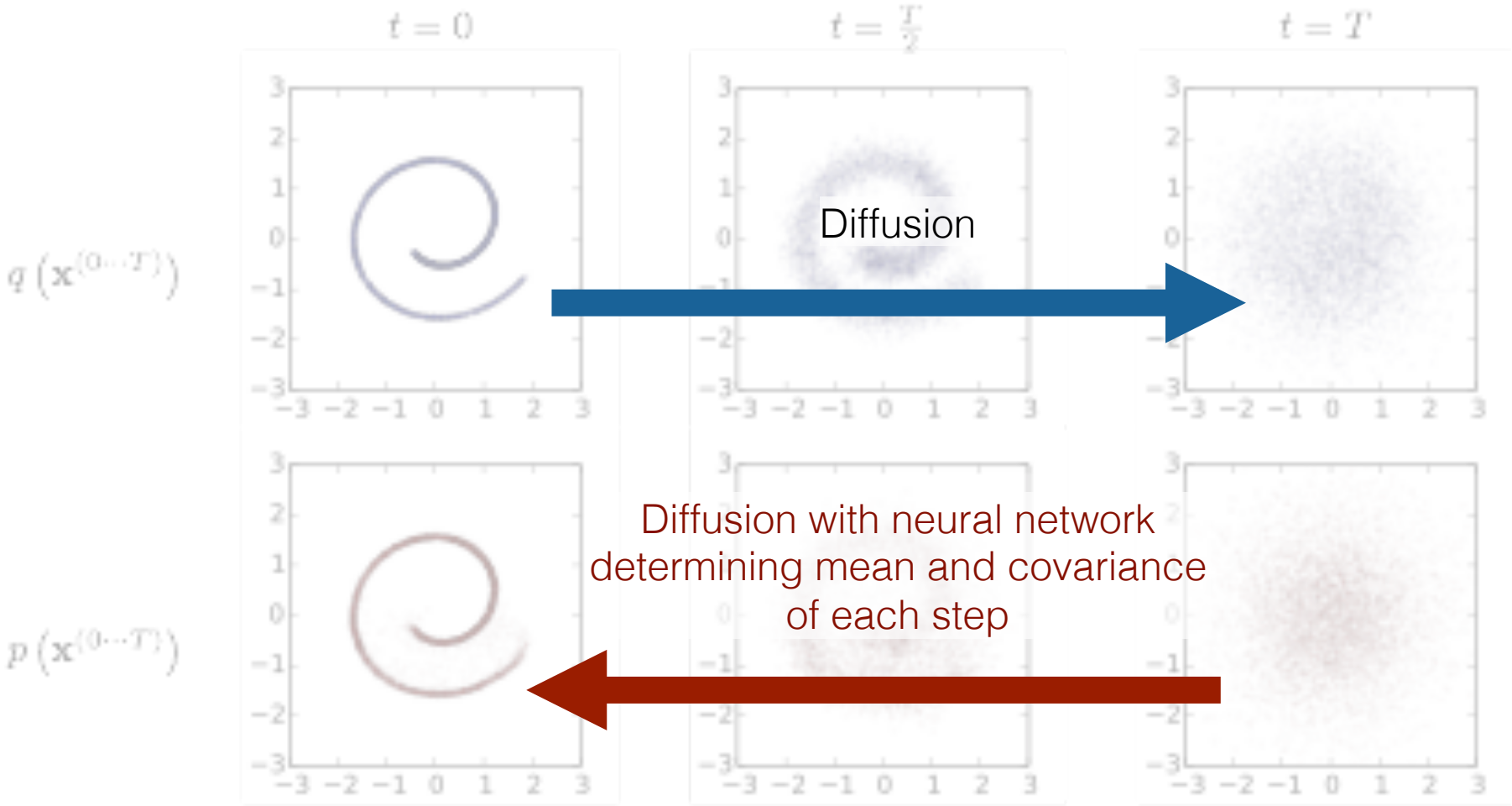


Swiss Roll

- Reverse diffusion process
  - Start at Gaussian blob
  - Run Gaussian diffusion until samples become data distribution



Swiss Roll

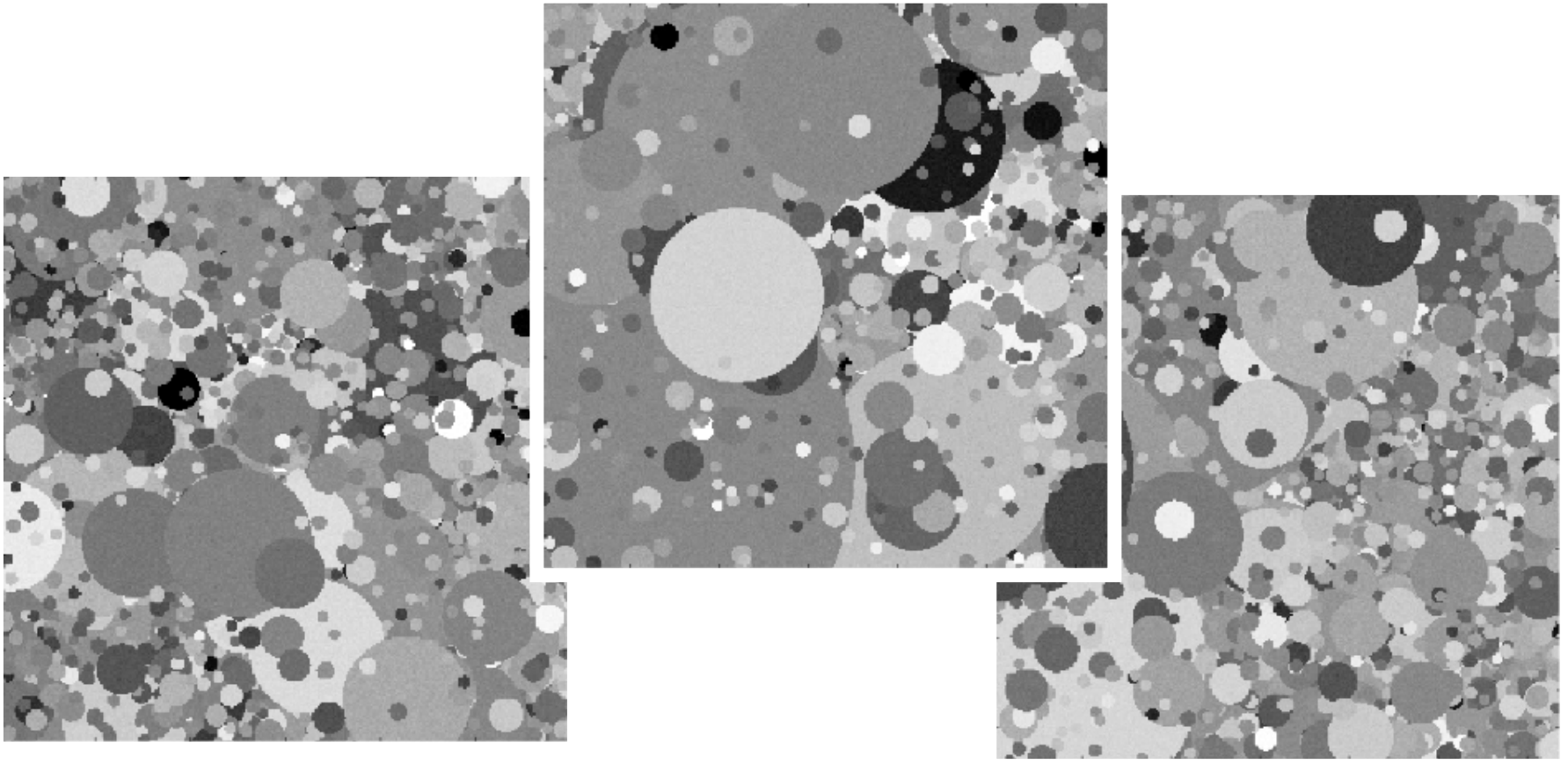


stein

Model

Dead Leaf Model

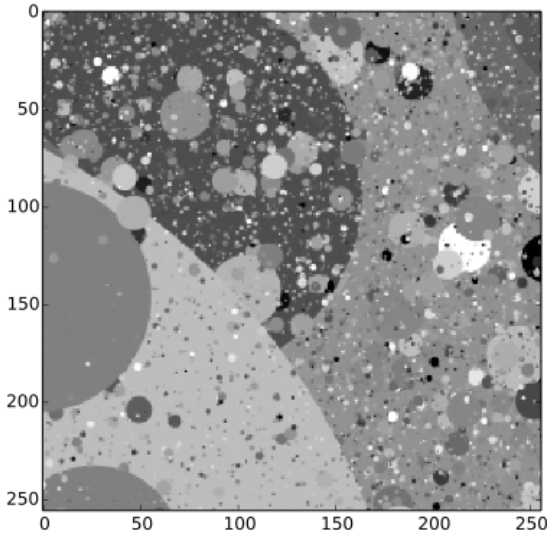
- Training data



stein

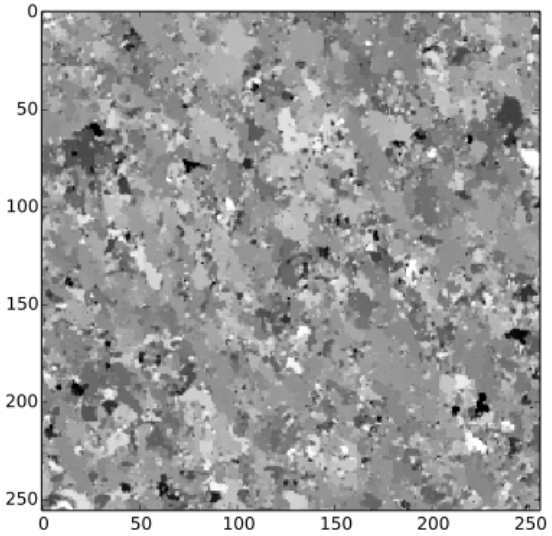
Model

Diffusion Probabilistic Model on Dead Leaves



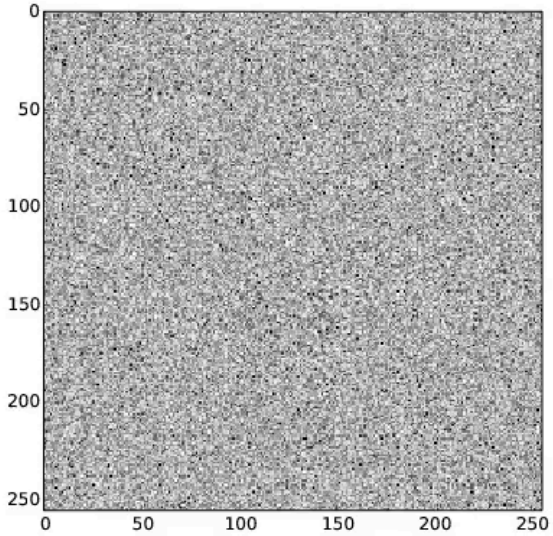
Training Data

Log likelihood
1.24 bits/pixel



Sample from
[Theis *et al*, 2012]

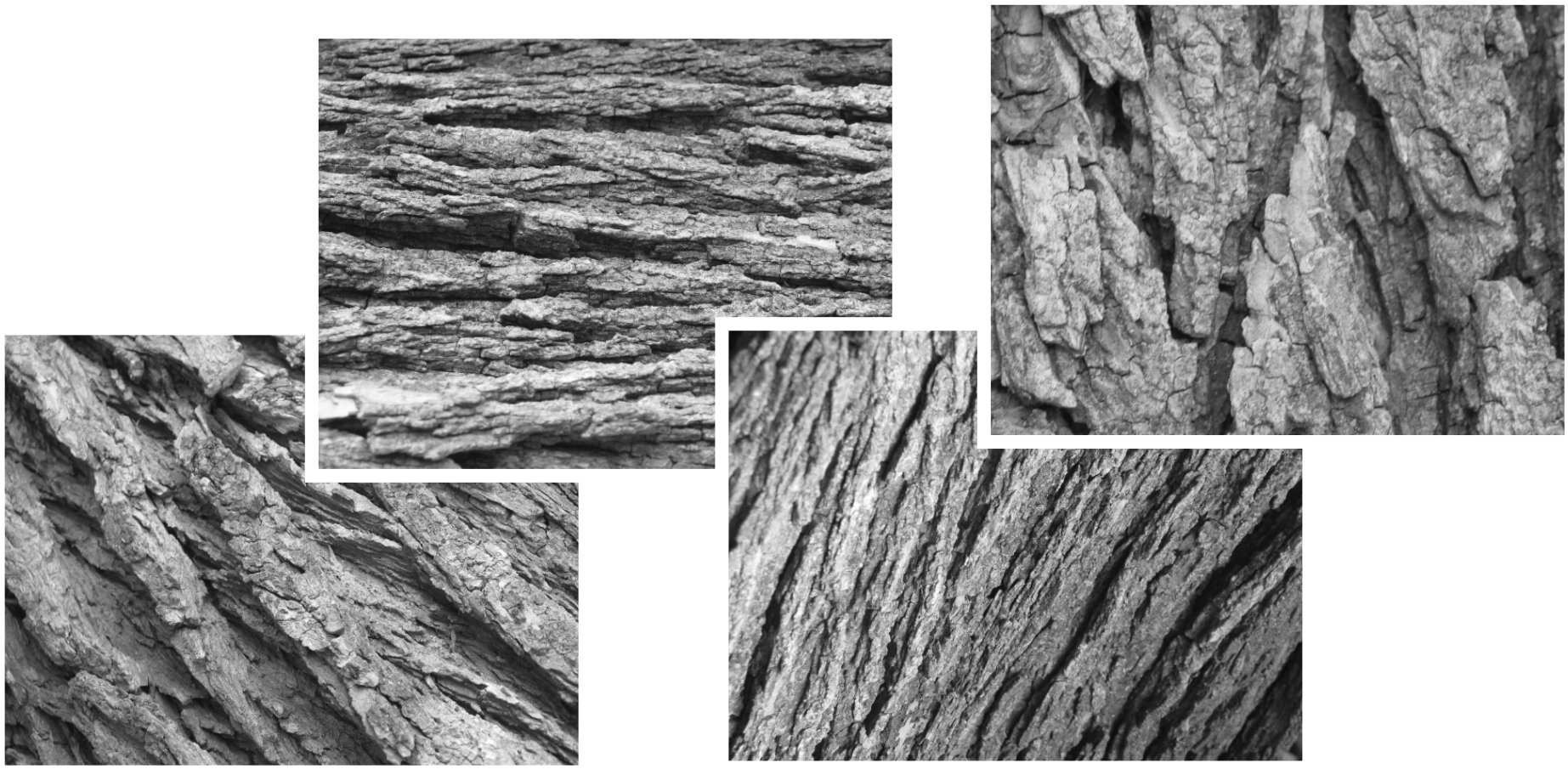
Log likelihood
1.49 bits/pixel



Sample from
diffusion model

Natural Images

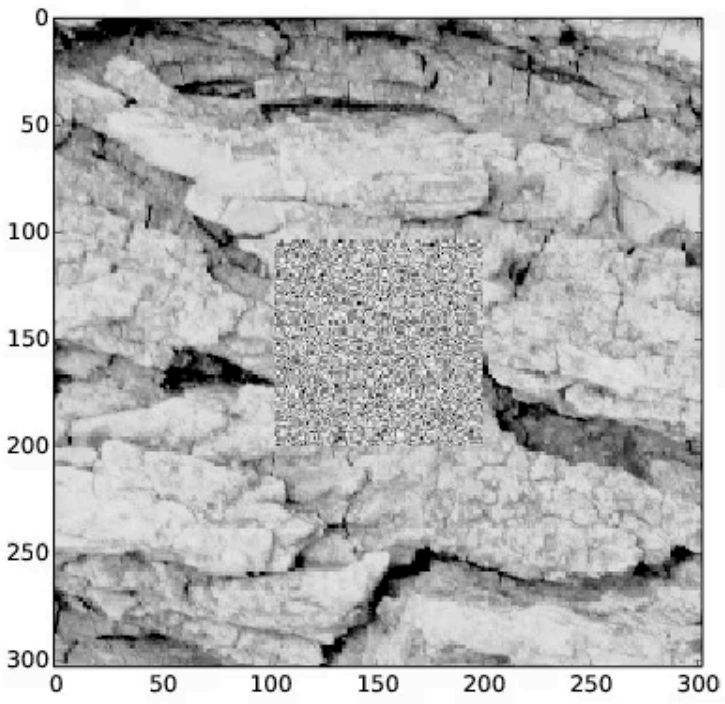
- Training data

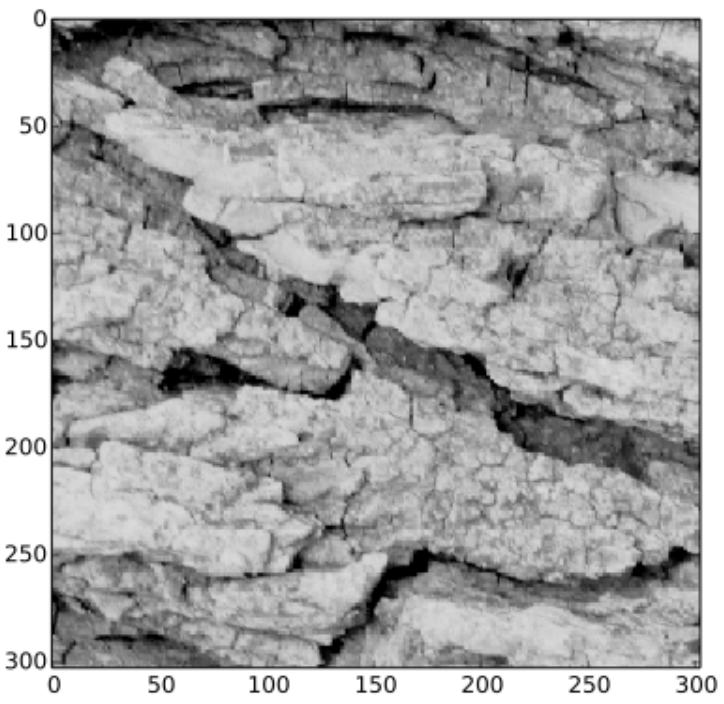


stein

Model

Diffusion Probabilistic Model Inpainting





Flexible and Tractable Learning of Probabilistic Models

- Flexible
  - Every distribution has a diffusion process (ongoing work applying to binary spike trains, and full color natural images from diverse scenes)
- Tractable
  - Training: Estimate mean and covariance of Gaussian
  - Sampling: Exact - model defined by sampling chain
  - Inference: Via sampling
  - Evaluation: Cheap - compute probability of sequence of Gaussians

Outline of talks I, II and III

1. Oldies but goodies

    1. Models of single neurons: Hodgkin Huxley to Hopfield
    2. The Hopfield model
    3. The perceptron learning algorithm: memorization and generalization
    4. Unsupervised learning: PCA, ICA, Sparse Coding

2. High dimensional statistics: theory and experiment

    1. The best way to do regression in high dimensions (Replica theory)
    2. Recovering neural state space dynamics (Rand proj / Matrices / Free prob)
    3. Figuring out how neural circuits learn (Tensor decompositions)

3. Deep learning: theory and practice

    1. Speeding up deep learning (Dynamic criticality)
    2. Error landscape of deep networks (Stat mech of random Gaussian fields)
    3. Deep generative models (Non-equilbrium thermodynamics)
    4. Expressive power of deep networks (Riemannian geometry and chaos theory)
    5. Application: deep models of the retina: the first step in seeing

A theory of deep neural expressivity through transient chaos

Stanford

Google

Ben Poole

Subhaneil Lahiri

Maithra Raghu

Jascha Sohl-Dickstein

Funding:

Bio-X Neuroventures
Burroughs Wellcome
Genentech Foundation
James S. McDonnell Foundation
McKnight Foundation
National Science Foundation

NIH
Office of Naval Research
Simons Foundation
Sloan Foundation
Swartz Foundation
Stanford Terman Award

http://ganguli-gang.stanford.edu

Twitter: @SuryaGanguli

References

Saxe, J. McClelland, S. Ganguli, Learning hierarchical category structure in deep neural networks, Cog Sci. 2013.

Saxe, J. McClelland, S. Ganguli, Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, ICLR 2014.

Identifying and attacking the saddle point problem in high dimensional non-convex optimization, Yann Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, Yoshua Bengio, NIPS 2014.

Modelling arbitrary probability distributions using non-equilibrium thermodynamics, J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, S. Ganguli, ICML 2015.

Exponential expressivity in deep neural networks through transient chaos, B. Poole, S. Lahiri,M. Raghu, J. Sohl-Dickstein, S. Ganguli, under review, arxiv/1606.05340

On the expressive power of deep neural networks, M.Raghu, B. Poole,J. Kleinberg, J. Sohl-Dickstein, S. Ganguli, under review, arxiv/1606.05336

Deep Knowledge Tracing, C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. Guibas, J. Sohl-Dickstein, NIPS 2015.

http://ganguli-gang.stanford.edu

Seminal works on the expressive power of depth

Networks with one hidden layer are universal function approximators.

So why do we need depth?

Universal function approximation theorems yield no guarantees on the size
of the hidden layer needed to approximate a function well.

Overall idea: there exist certain (special?) functions that can be computed:

a) efficiently using a deep network (poly # of neurons in input dimension)

b) but not by a shallow network (requires exponential # of neurons)

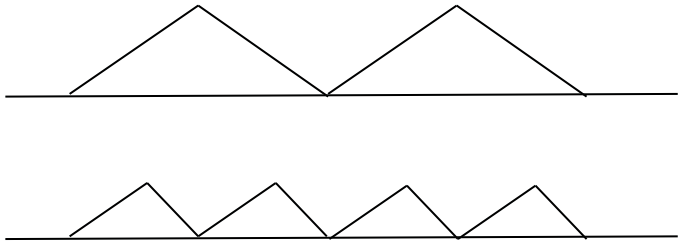
Intellectual traditions in boolean circuit theory: parity function is such a function for boolean circuits.

Seminal works on the expressive power of depth

Nonlinearity	Measure of Functional Complexity
Rectified Linear Unit (ReLu)	Number of linear regions

There exists a function computable by a deep network where the number of linear regions is exponential in the depth.

To approximate this function with a shallow network, one would require exponentially many more neurons.



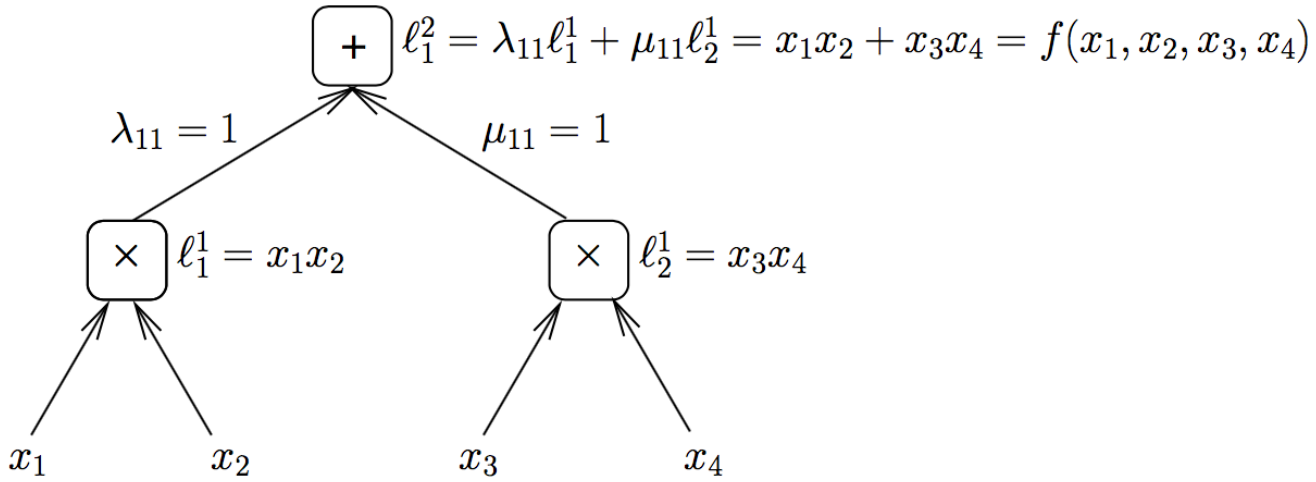
Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio.
On the number of linear regions of deep neural networks, NIPS 2014

Seminal works on the expressive power of depth

Nonlinearity	Measure of Functional Complexity
Sum-product network	Number of monomials

There exists a function computable by a deep network where the number of unique monomials is exponential in the depth.

To approximate this function with a shallow network, one would require exponentially many more neurons.



Olivier Delalleau and Yoshua Bengio. Shallow vs. deep sum-product networks, NIPS 2011.

Questions

The particular functions exhibited by prior work do not seem natural?

(see Tommy Poggio's talk later today!)

Are such functions rare curiosities?

Or is this phenomenon much more generic than these specific examples?

In some sense, is **any** function computed by a **generic** deep network not efficiently computable by a shallow network?

If so we would like a theory of deep neural expressivity that demonstrates this for

1) Arbitrary nonlinearities

2) A natural, general measure of functional complexity.

Limitations of prior work

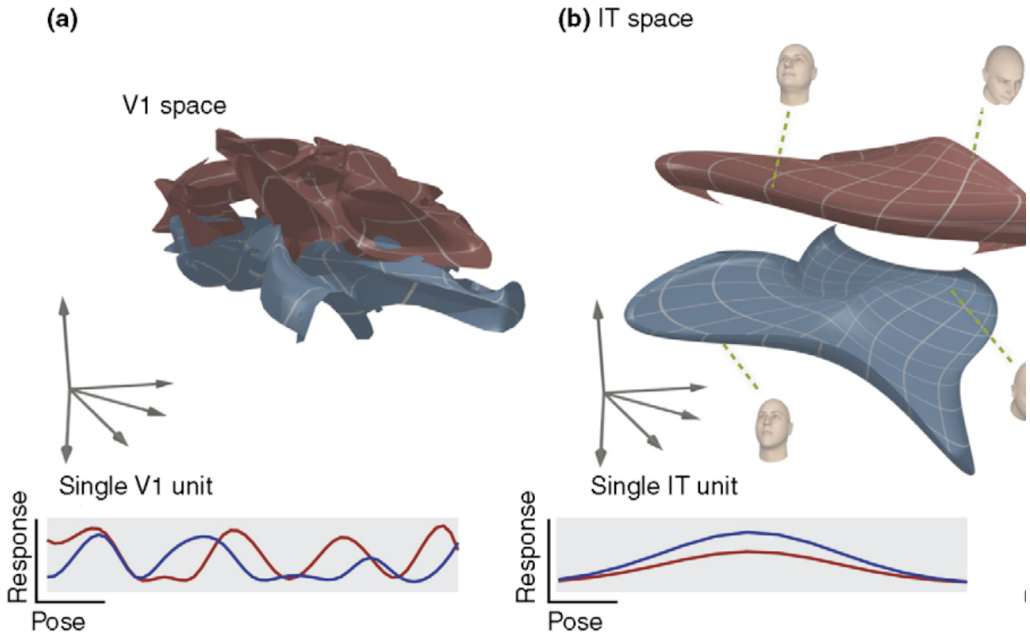
Theoretical technique	Nonlinearity	Measure of Functional Complexity
Combinatorics/ Hyperplane Arrangements	ReLU	Number of linear regions
Polynomial expansion	Sum-product	Number of monomials
Algebraic topology	Pfaffian	Sum of betti numbers
Riemannian geometry + Dynamical mean field theory	**Arbitrary**	**Extrinsic Curvature**

Monica Bianchini and Franco Scarselli. On the complexity of neural network classifiers: A comparison between shallow and deep architectures. *Neural Networks and Learning Systems, IEEE Transactions on*, 2014.
ier

We will show that even in generic, random deep neural networks, measures of functional curvature grow exponentially with depth but not width!

More over the origins of this exponential growth can be traced to chaos theory.

Another perspective on the advantage of depth: disentangling

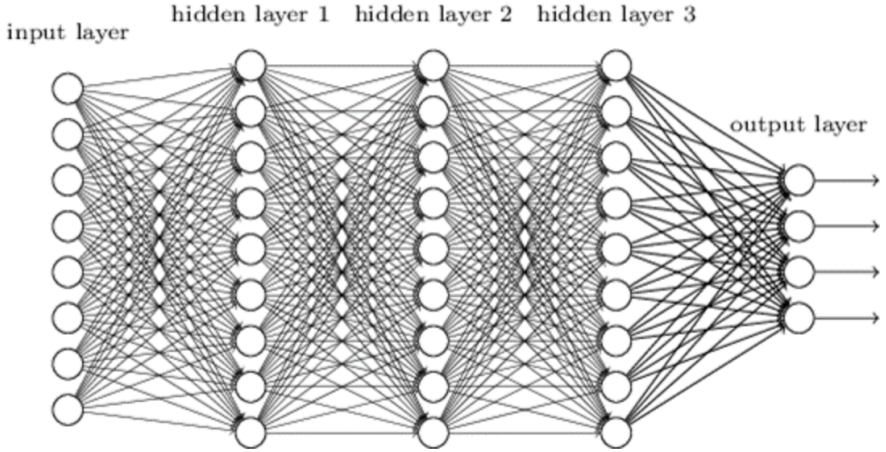


How can we mathematically formalize the notion of disentangling in deep networks?

How do we use this mathematical formalization to quantitatively assess the disentangling power of deep versus shallow networks?

We will show that deep networks can disentangle manifolds whose curvature grows exponentially with depth!

A maximum entropy ensemble of deep random networks



$$N_l = \text{number of neurons in layer l}$$

$$D = \text{depth}(l = 1, \ldots, D)$$

$$\mathbf{x}^l = \phi(\mathbf{h}^l)$$

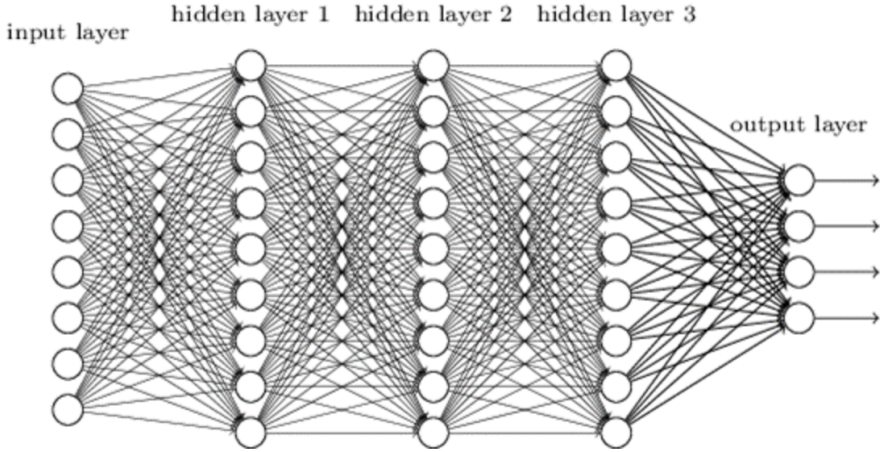
$$\mathbf{h}^l = \mathbf{W}^l \, \mathbf{x}^{l-1} + \mathbf{b}^l$$

Structure: i.i.d. random Gaussian weights and biases:

$$\mathbf{W}_{ij}^l \leftarrow \mathcal{N}\left(0, \frac{\sigma_w^2}{N^{l-1}}\right)$$

$$\mathbf{b}_i^l \leftarrow \mathcal{N}(0, \sigma_b^2)$$

Emergent, deterministic signal propagation in random neural networks



$$N_l = \text{number of neurons in layer l}$$

$$D = \text{depth}(l = 1, \ldots, D)$$

$$\mathbf{x}^l = \phi(\mathbf{h}^l)$$

$$\mathbf{h}^l = \mathbf{W}^l \, \mathbf{x}^{l-1} + \mathbf{b}^l$$

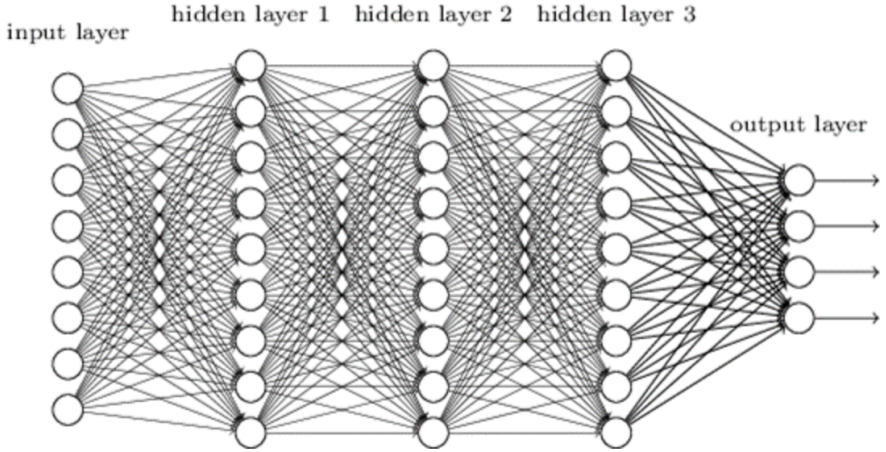
Question: how do simple input manifolds propagate through the layers?

A single point: When does its length grow or shrink and how fast?

A pair of points: Do they become more similar or more different, and how fast?

A smooth manifold: How does its curvature and volume change?

Propagation of a single point through a deep network



$$N_l = \text{number of neurons in layer l}$$
$$D = \text{depth}(l = 1, \ldots, D)$$
$$\mathbf{x}^l = \phi(\mathbf{h}^l)$$
$$\mathbf{h}^l = \mathbf{W}^l\, \mathbf{x}^{l-1} + \mathbf{b}^l$$

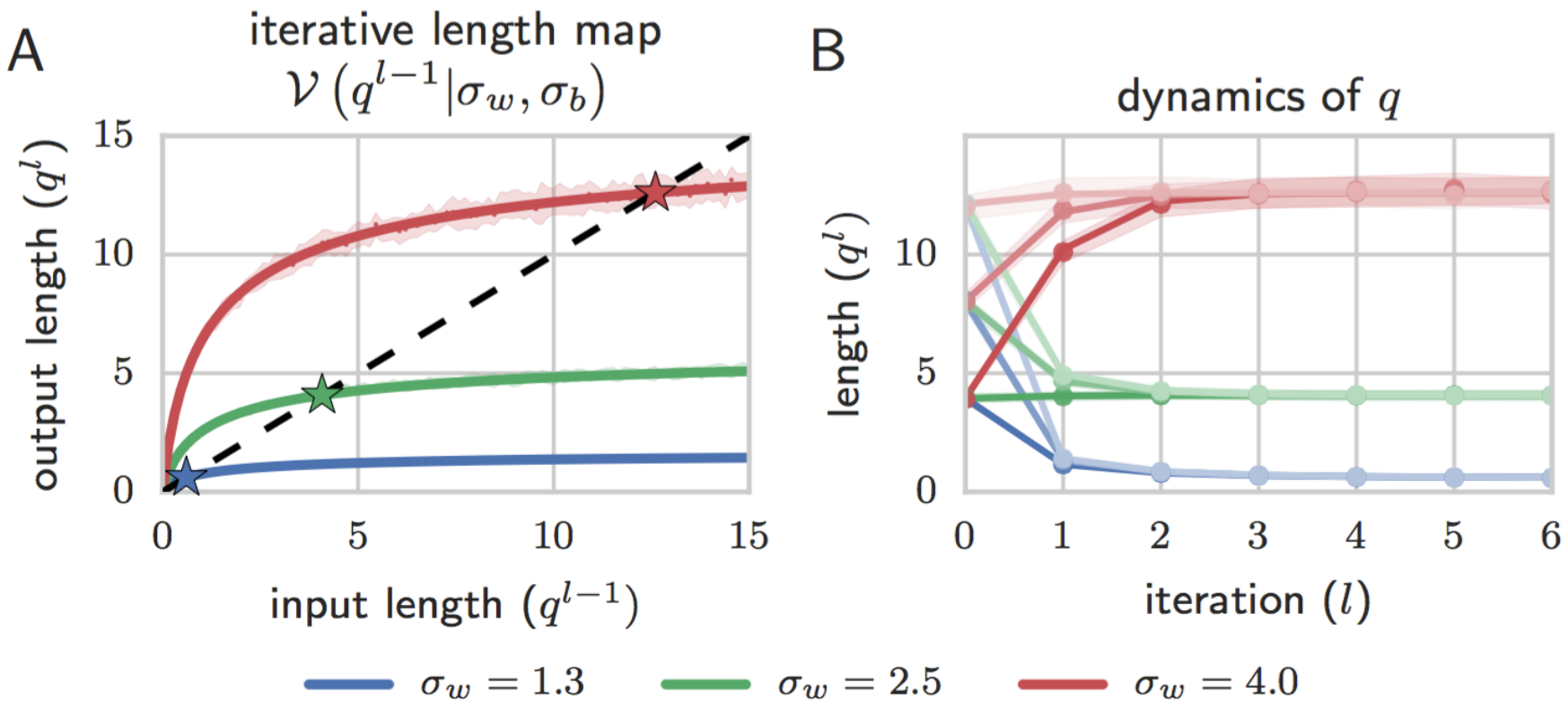
$$\mathbf{h}^l = \mathbf{W}^l\, \phi(\mathbf{h}^{l-1}) + \mathbf{b}^l$$

$$q^l = \frac{1}{N_l} \sum_{i=1}^{N_l} (\mathbf{h}_i^l)^2$$

$$q^l = \mathcal{V}(q^{l-1} \,|\, \sigma_w, \sigma_b) \equiv \sigma_w^2 \int \mathcal{D}z\, \phi\left(\sqrt{q^{l-1}} z\right)^2 + \sigma_b^2$$

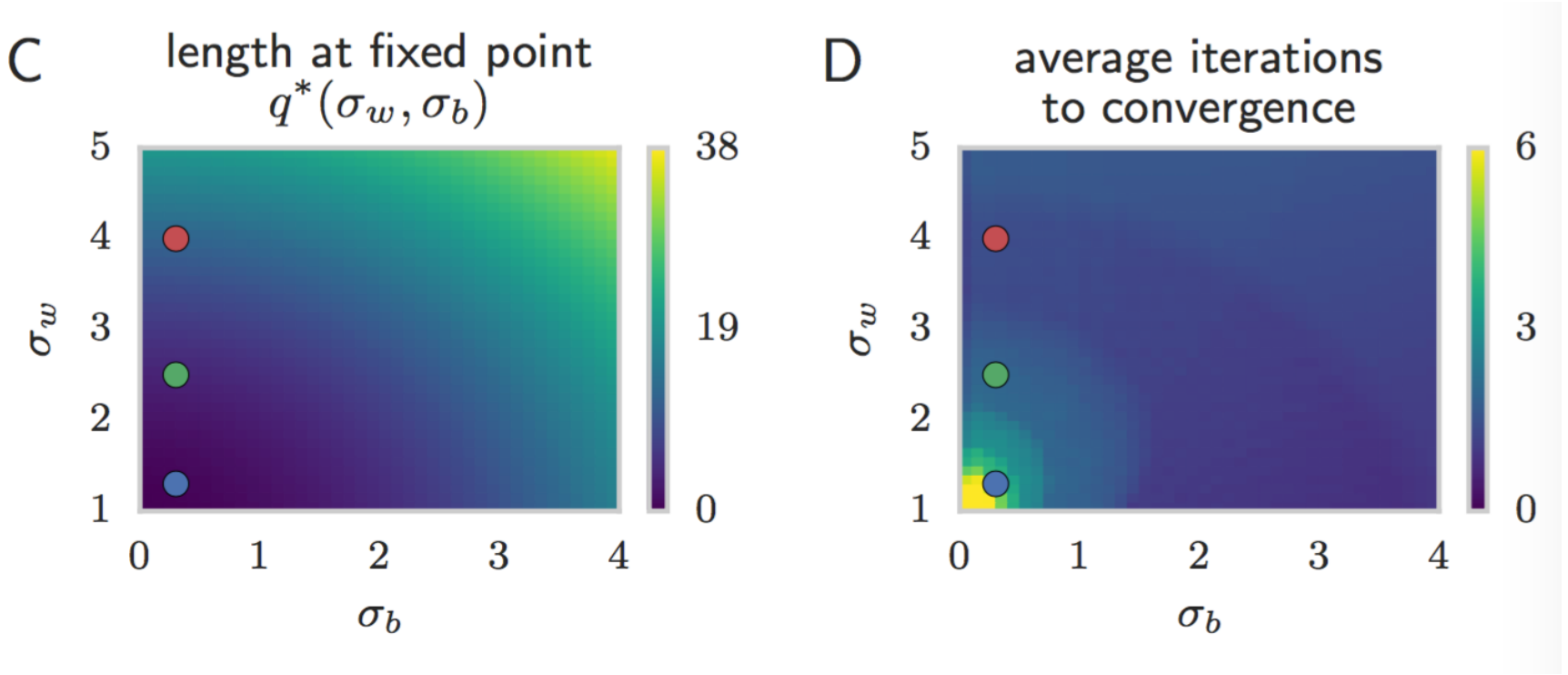
A recursion relation for the length of a point as it propagates through the network

Propagation of a single point through a deep network



$$\sigma_b = 0.3$$

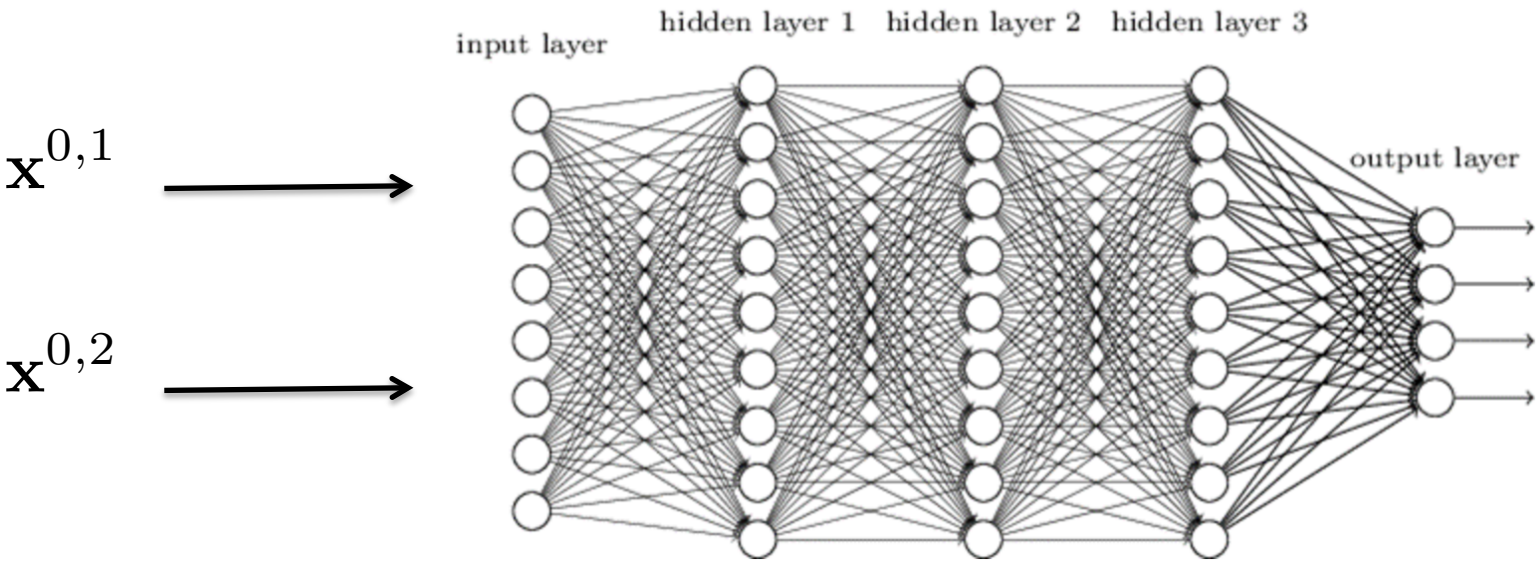
Propagation of a single point through a deep network



$$\sigma_w < 1 \quad \sigma_b = 0 : \quad q^l \to 0$$

$$\sigma_w > 1 \quad \sigma_b = 0 \quad \text{or} \quad \sigma_b \neq 0 : \quad q^l \to q^*$$

Propagation of two points through a deep network



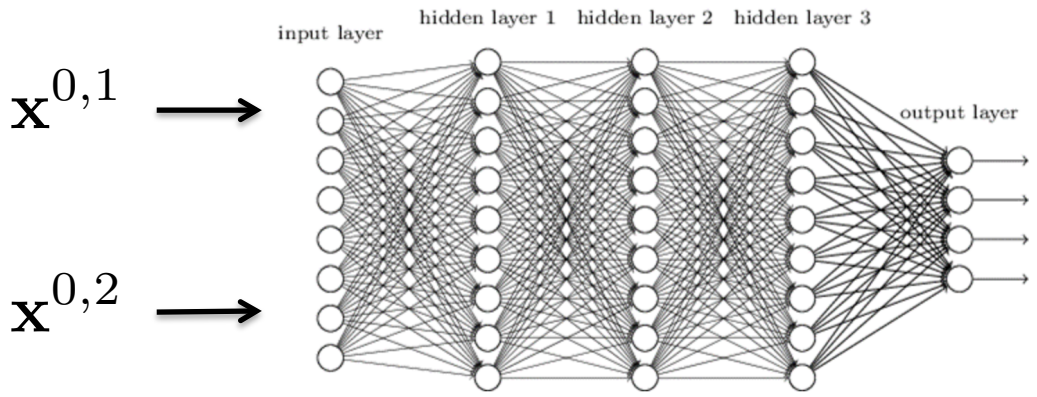
The geometry of two points in a hidden layer l is captured by the two by two matrix of inner products:

$$q^l_{ab} = \frac{1}{N_l} \sum_{i=1}^{N_l} \mathbf{h}^l_i(\mathbf{x}^{0,a})\, \mathbf{h}^l_i(\mathbf{x}^{0,b}) \qquad a, b \in \{1, 2\}.$$

Of particular interest: the correlation coefficient or cosine of the angle between the two points:

$$c^l_{12} = \frac{q^l_{12}}{\sqrt{q^l_{11}}\sqrt{q^l_{22}}}$$

A theory of correlation propagation in a deep network



The geometry of two points:

$$q^l_{ab} = \frac{1}{N_l}\sum_{i=1}^{N_l} \mathbf{h}^l_i(\mathbf{x}^{0,a})\,\mathbf{h}^l_i(\mathbf{x}^{0,b}) \qquad a,b \in \{1,2\}.$$

Correlation coefficient between two points:

$$c^l_{12} = \frac{q^l_{12}}{\sqrt{q^l_{11}}\sqrt{q^l_{22}}}$$

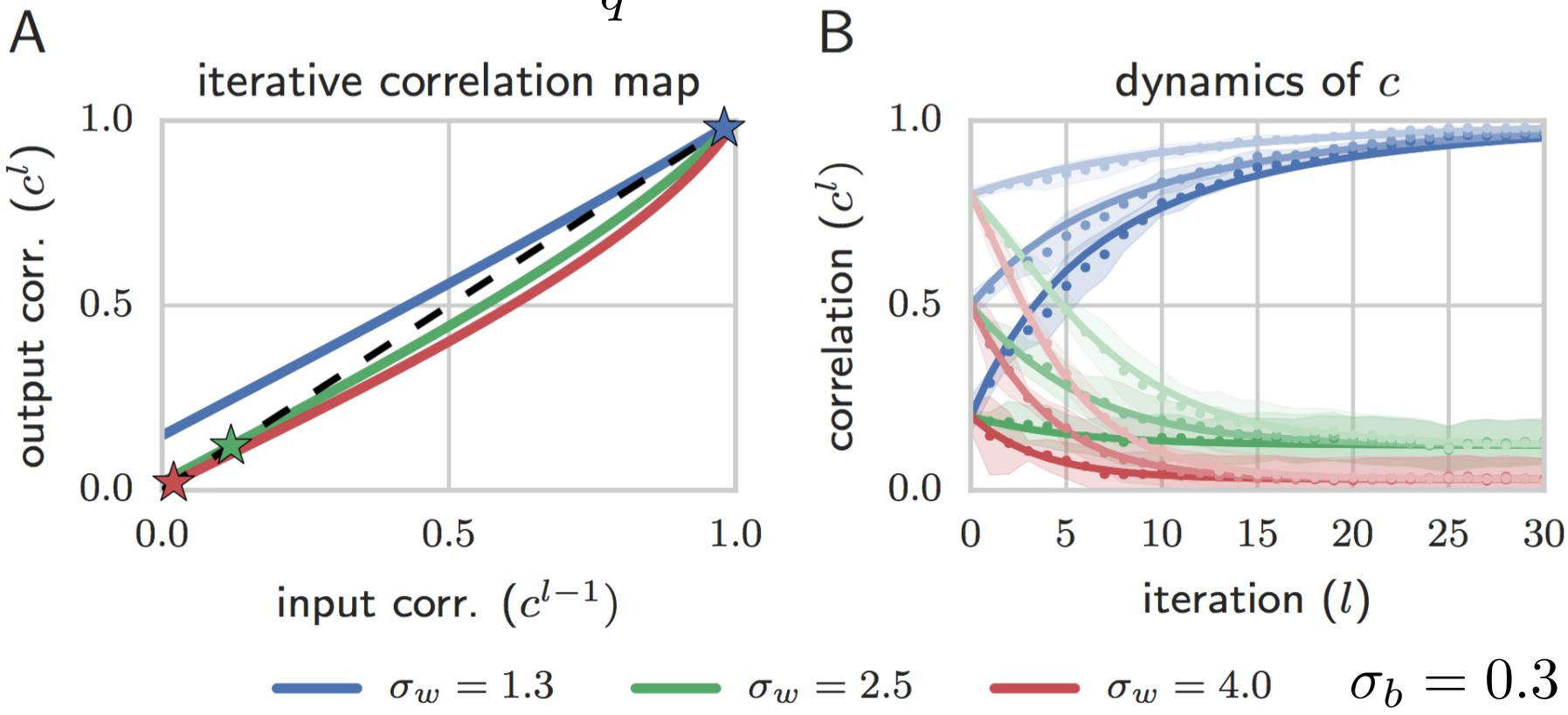
$$q^l_{12} = \mathcal{C}(c^{l-1}_{12},\, q^{l-1}_{11},\, q^{l-1}_{22}\,|\,\sigma_w w, \sigma_b) \equiv \sigma_w^2 \int \mathcal{D}z_1\,\mathcal{D}z_2\,\phi\left(u_1\right)\phi\left(u_2\right) + \sigma_b^2,$$

$$u_1 = \sqrt{q^{l-1}_{11}}\,z_1, \qquad u_2 = \sqrt{q^{l-1}_{22}}\left[c^{l-1}_{12}\,z_1 + \sqrt{1-(c^{l-1}_{12})^2}\,z_2\right],$$

A recursion relation for the correlation coeff. between two points in a deep net!

Propagation of correlations through a deep network

$$c_{12}^{l} = \frac{1}{q^*}\mathcal{C}(c_{12}^{l-1},\, q^*,\, q^* \,|\, \sigma_w, \sigma_b)$$



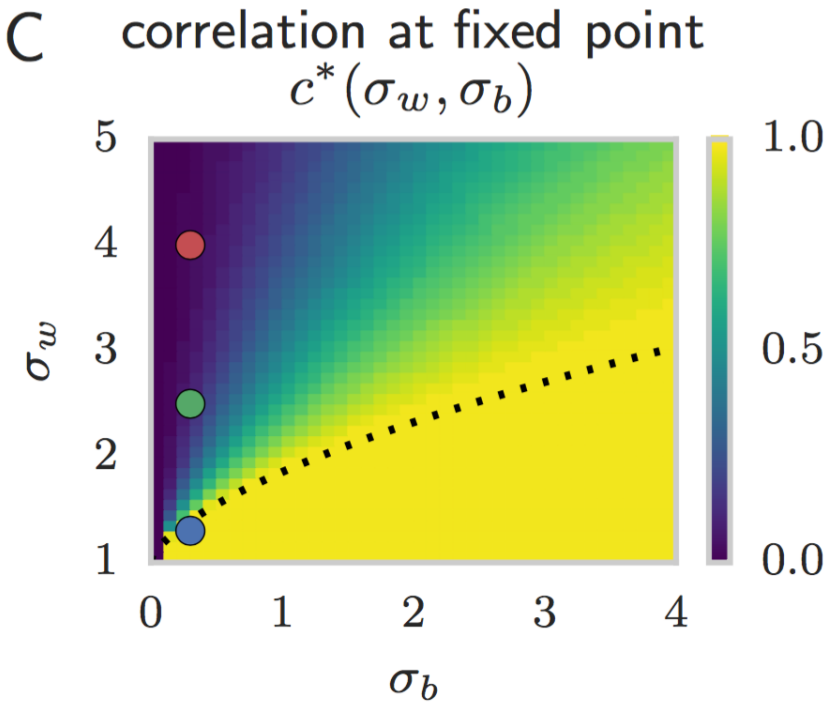
$$\chi_1 \equiv \left.\frac{\partial c_{12}^{l}}{\partial c_{12}^{l-1}}\right|_{c=1} = \sigma_w^2 \int \mathcal{D}z \left[\phi'\left(\sqrt{q^*}z\right)\right]^2$$

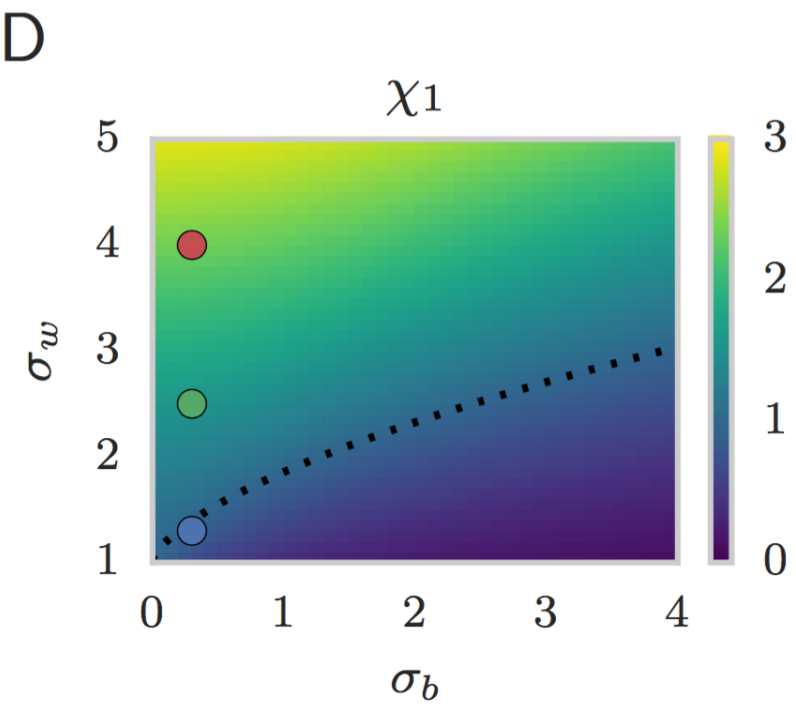
Interpretation: χ_1 is a multiplicative stretch factor:

$\chi_1 < 1$: nearby points come closer together

$\chi_1 > 1$: nearby points are driven apart

Propagation of two points through a deep network





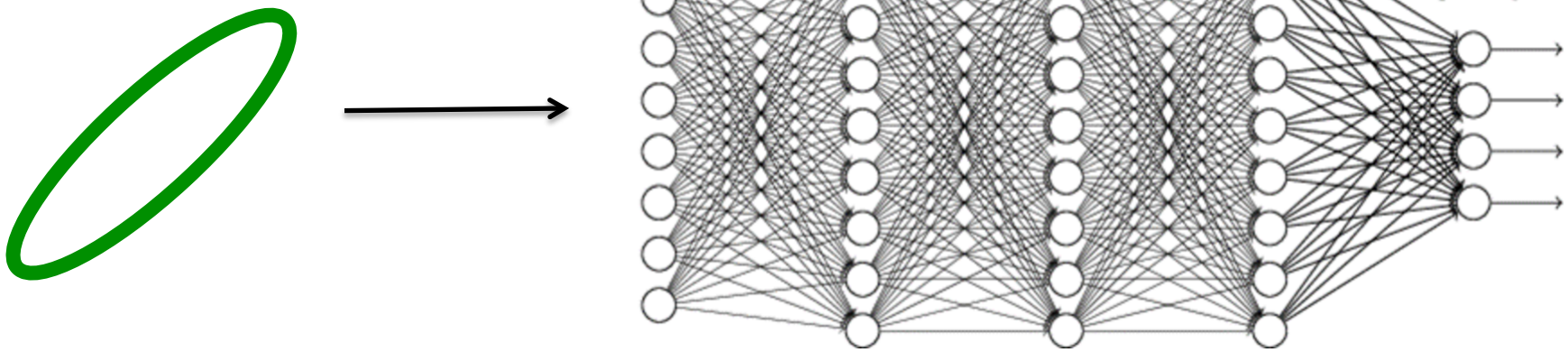
$$\text{Small } \sigma_w \text{ relative to } \sigma_b : \quad \chi_1 < 1 \quad c_{12}^l \to 1$$

$$\text{Intermediate } \sigma_w \text{ relative to } \sigma_b : \quad \chi_1 > 1 \quad c_{12}^l \to c^*$$

$$\text{Large } \sigma_w \text{ relative to } \sigma_b : \quad \chi_1 > 1 \quad c_{12}^l \to 0$$

Propagation of a manifold through a deep network

$\mathbf{x}^0(\theta)$



The geometry of the manifold is captured by the similarity matrix - How similar two points are in internal representation space):

$$q^l(\theta_1, \theta_2) = \frac{1}{N_l} \sum_{i=1}^{N_l} \mathbf{h}_i^l[\mathbf{x}^0(\theta_1)]\, \mathbf{h}_i^l[\mathbf{x}^0(\theta_2)]$$

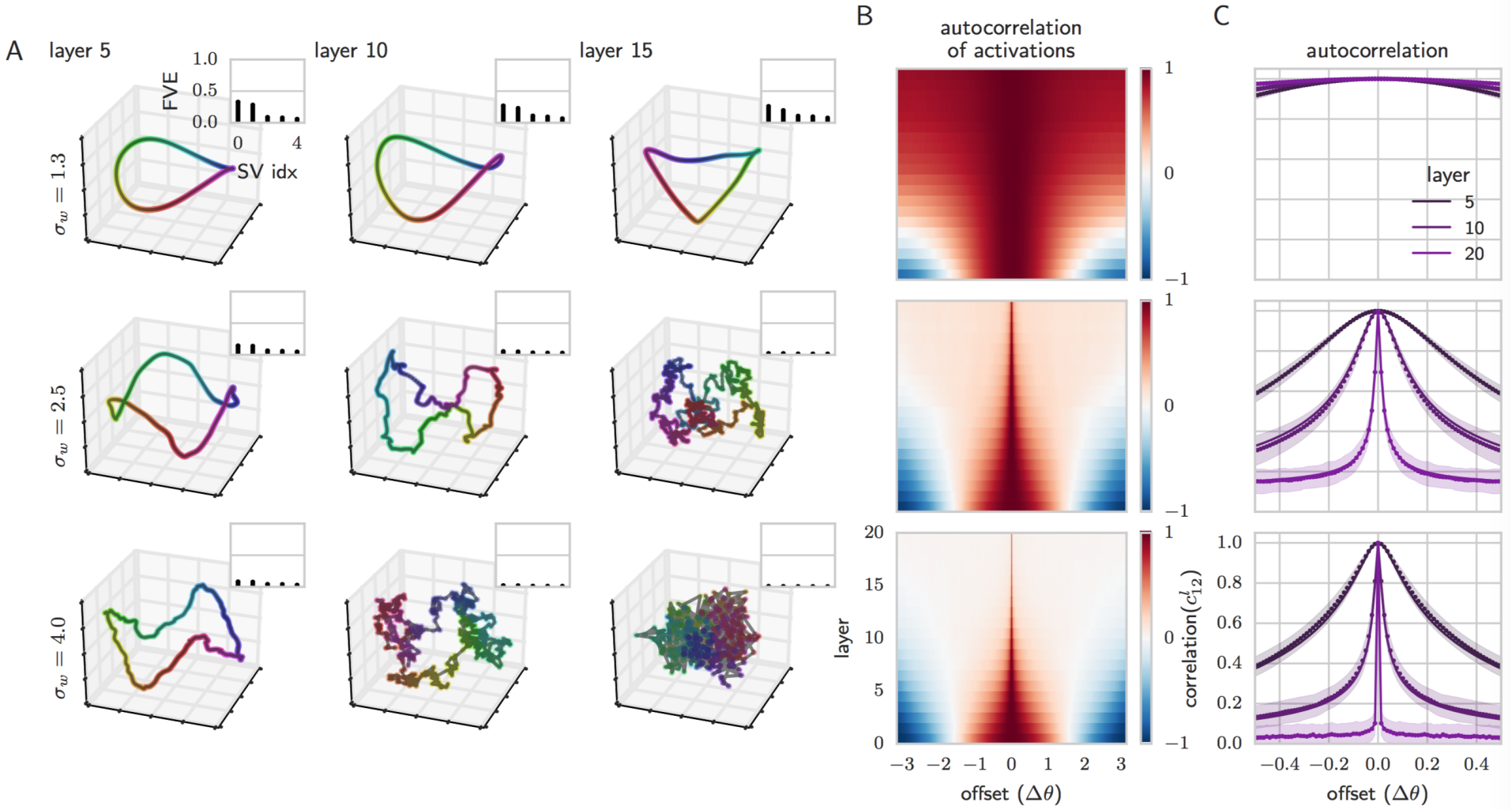
Or autocorrelation function:

$$q^l(\Delta\theta) = \int d\theta\, q^l(\theta, \theta + \Delta\theta)$$

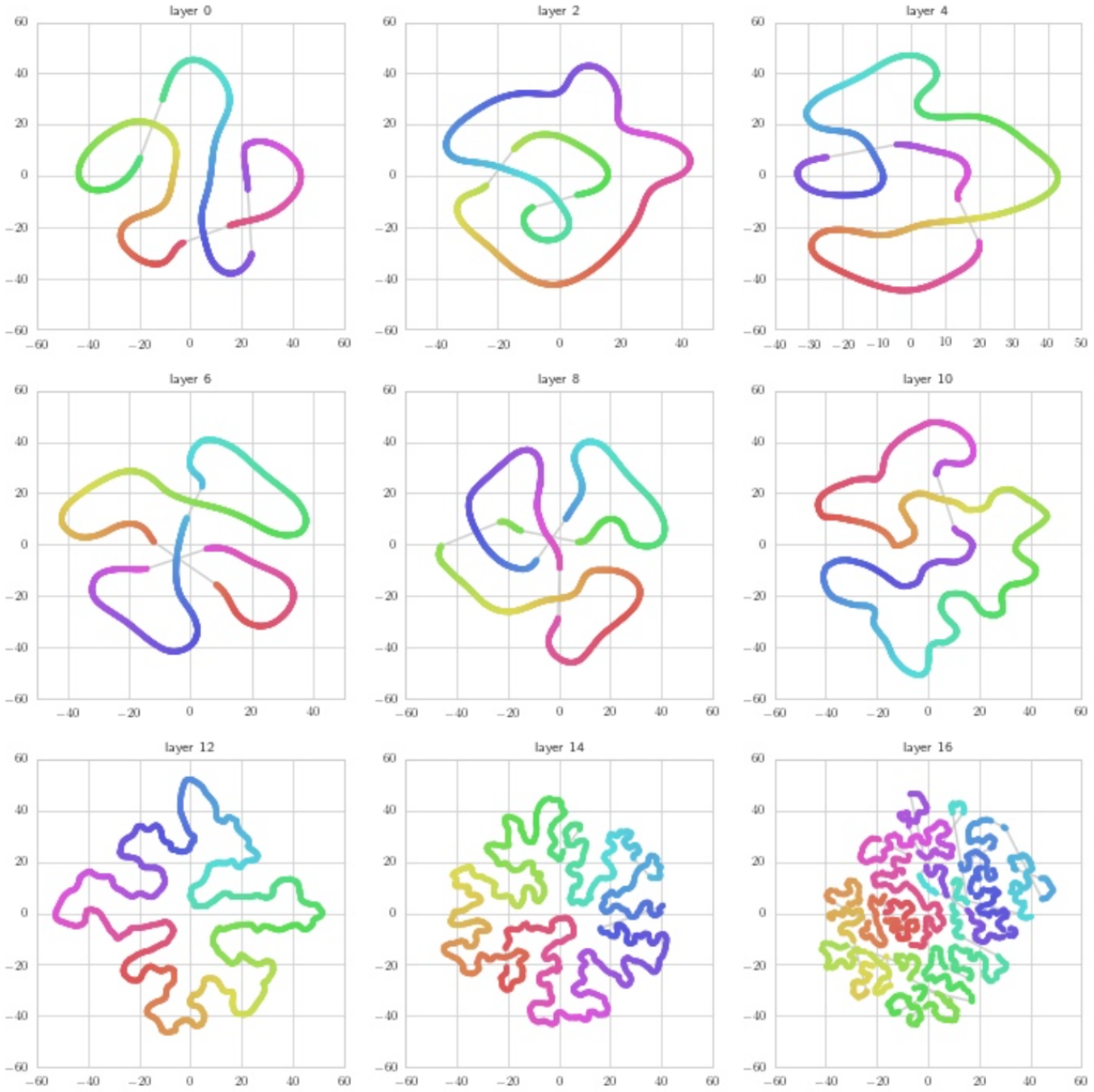
Propagation of a manifold through a deep network

$$\mathbf{h}^1(\theta) = \sqrt{N_1 q^*}\left[\mathbf{u}^0\cos(\theta) + \mathbf{u}^1\sin(\theta)\right]$$

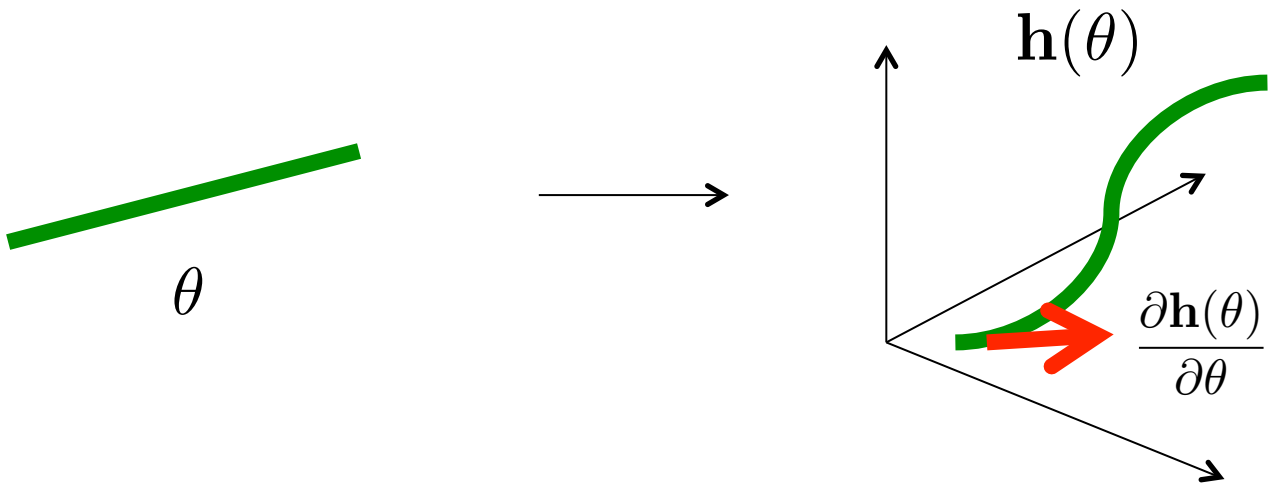
A great circle input manifold



Propagation of a manifold through a deep network



Riemannian geometry I: Euclidean length



$$g^E(\theta) = \frac{\partial \mathbf{h}(\theta)}{\partial \theta} \cdot \frac{\partial \mathbf{h}(\theta)}{\partial \theta}$$

Metric on manifold coordinate θ induced by Euclidean metric in internal representation space **h**.

$$d\mathcal{L}^E = \sqrt{g^E(\theta)}d\theta$$

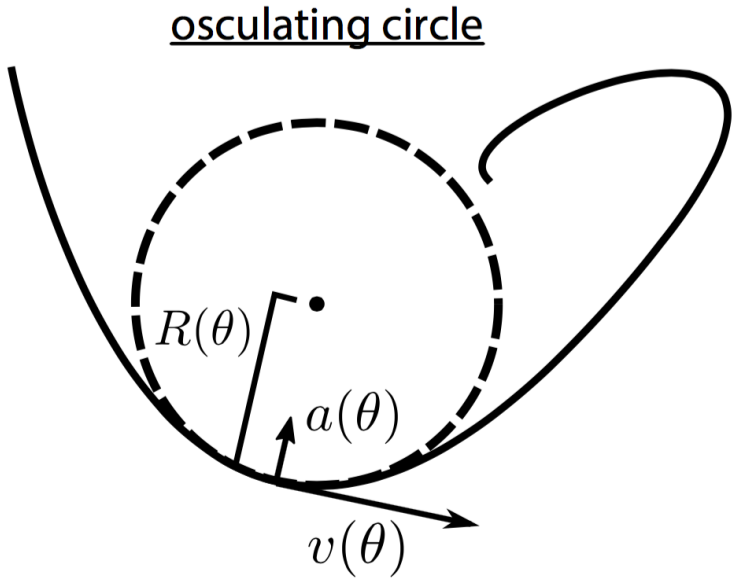
Length element: if one moves from Θ to $\Theta + d\Theta$ along the manifold, then one moves a distance dL^E in internal representation space

Riemannian geometry II: Extrinsic Gaussian Curvature

$\mathbf{h}(\theta)$ Point on the curve

$\mathbf{v}(\theta) = \frac{\partial \mathbf{h}(\theta)}{\partial \theta}$ Tangent or velocity vector

$\mathbf{a}(\theta) = \frac{\partial \mathbf{v}(\theta)}{\partial \theta}$ Acceleration vector



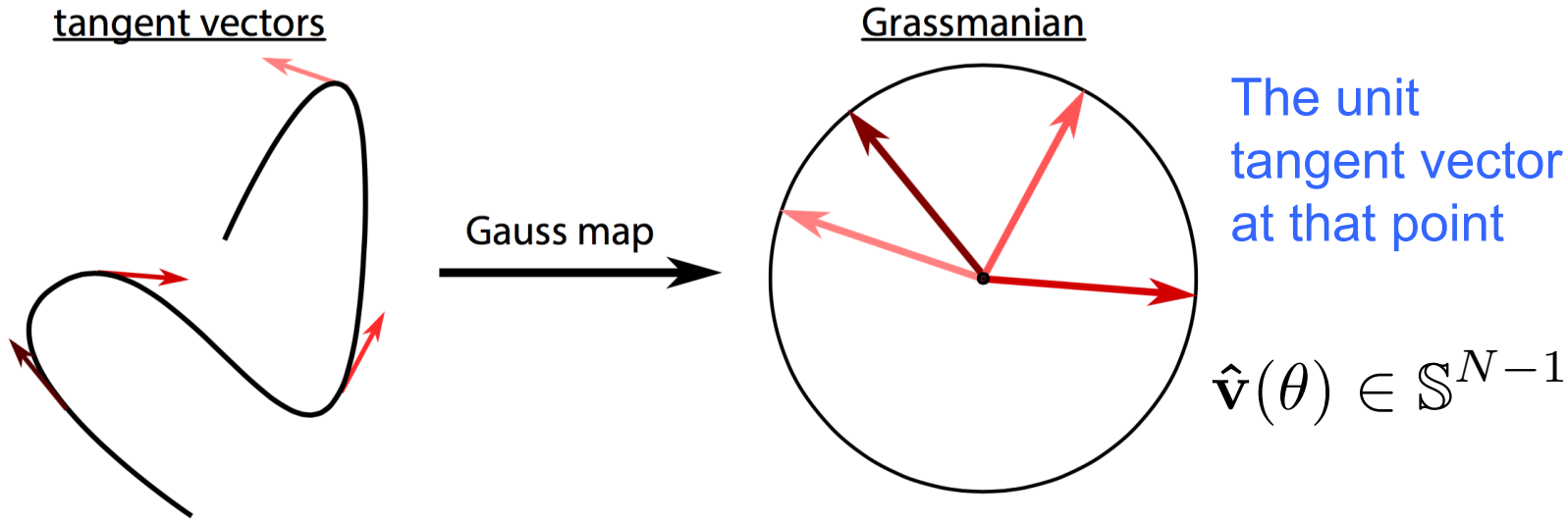
The velocity and acceleration vector span a 2 dimensional plane in N dim space.

Within this plane, there is a unique circle that touches the curve at $\mathbf{h}(\theta)$, with the same velocity and acceleration.

The Gaussian curvature $\kappa(\theta)$ is ~~the inverse of the radius of th~~is circle.

$$\kappa(\theta) = \sqrt{\frac{(\mathbf{v}\cdot\mathbf{v})(\mathbf{a}\cdot\mathbf{a}) - (\mathbf{v}\cdot\mathbf{a})^2}{(\mathbf{v}\cdot\mathbf{v})^3}}$$

Riemannian geometry III: The Gauss map and Grassmannian length



A point on the curve

θ

The unit tangent vector at that point

$\hat{\mathbf{v}}(\theta) \in \mathbb{S}^{N-1}$

$$g^G(\theta) = \frac{\partial \hat{\mathbf{v}}(\theta)}{\partial \theta} \cdot \frac{\partial \hat{\mathbf{v}}(\theta)}{\partial \theta}$$

Metric on manifold coordinate θ induced by metric on the Grassmannian: how quickly unit tangent vector changes

$$d\mathcal{L}^G = \sqrt{g^G(\theta)}d\theta$$

Length element: if one moves from Θ to Θ+ dΘ along the manifold, then one moves a distance dL^G Along the Grassmanian

$$g^G(\theta) = \kappa(\theta)^2 g^E(\theta)$$

Grassmannian length, Gaussian curvature and Euclidean length

An example: the great circle

$$\mathbf{h}^1(\theta) = \sqrt{Nq}\left[\mathbf{u}^0\cos(\theta) + \mathbf{u}^1\sin(\theta)\right]$$

A great circle input manifold

Euclidean length	Gaussian Curvature	Grassmannian Length
$g^E(\theta) = Nq$	$\kappa(\theta) = 1/\sqrt{Nq}$	$g^G(\theta) = 1$
$\mathcal{L}^E = 2\pi\sqrt{Nq}$		$\mathcal{L}^G = 2\pi$

Behavior under isotropic linear expansion via multiplicative stretch χ_1:

	$\mathcal{L}^E \to \sqrt{\chi_1}\,\mathcal{L}^E$	$\kappa \to \frac{1}{\sqrt{\chi_1}}\,\kappa$	$\mathcal{L}^G \to \mathcal{L}^G$
$\chi_1 < 1$	Contraction	Increase	Constant
$\chi_1 > 1$	Expansion	Decrease	Constant

Theory of curvature propagation in deep networks

$$\bar{g}^{E,l} = \chi_1 \, \bar{g}^{E,l-1} \qquad \bar{g}^{E,1} = q^*$$

$$(\bar{\kappa}^l)^2 = 3\frac{\chi_2}{\chi_1^2} + \frac{1}{\chi_1}(\bar{\kappa}^{l-1})^2 \qquad (\bar{\kappa}^1)^2 = \frac{1}{q^*}$$

$$\chi_1 = \sigma_w^2 \int \mathcal{D}z \left[\phi'\left(\sqrt{q^*}z\right)\right]^2$$

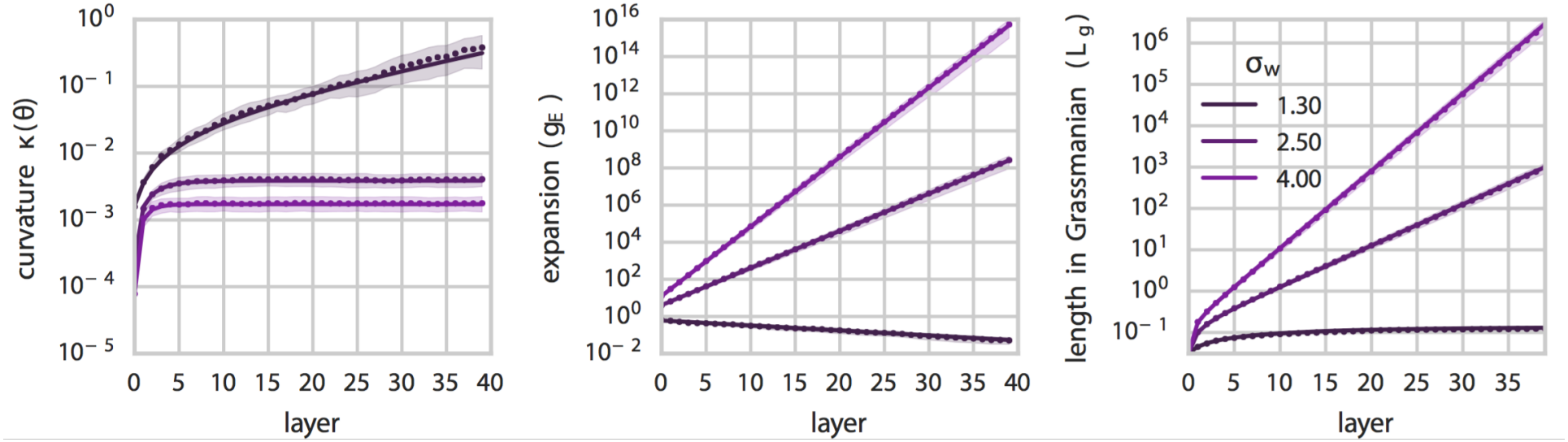
$$\chi_2 = \sigma_w^2 \int \mathcal{D}z \left[\phi''\left(\sqrt{q^*}z\right)\right]^2$$

Modification of existing curvature due to stretch

Addition of new curvature due to nonlinearity

		Local Stretch	Gaussian Curvature	Grassmannian Length
Ordered:	$\chi_1 < 1$	Contraction	Explosion	Constant
Chaotic:	$\chi_1 > 1$	Expansion	Attentuation + Addition	Exponential Growth

Curvature propagation: theory and experiment

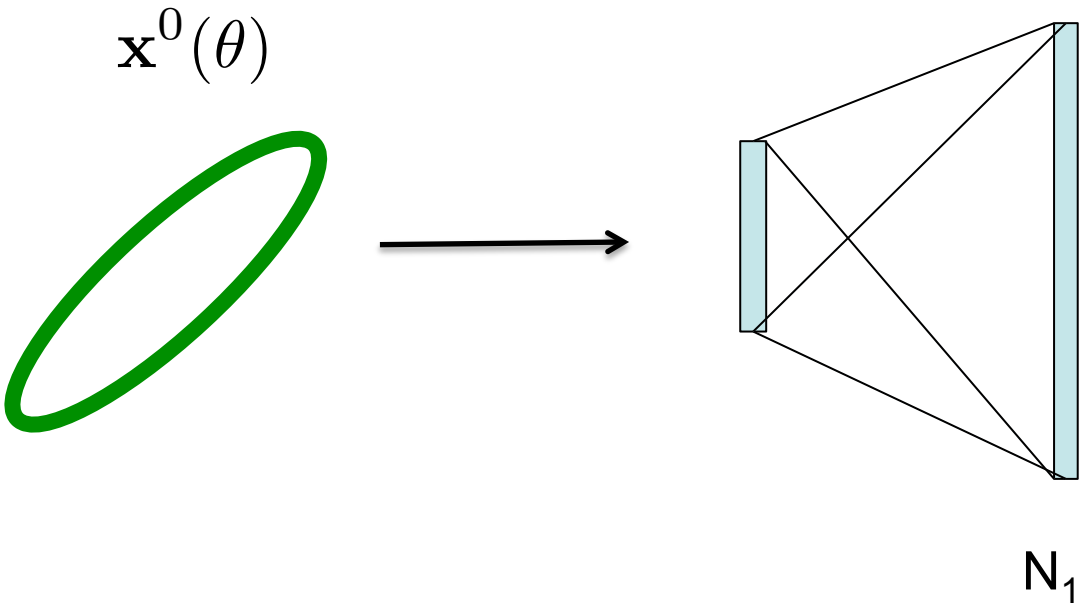


Unlike linear expansion, deep neural signal propagation can:

1) exponentially expand length,
2) without diluting Gaussian curvature,
3) thereby yielding exponential growth of Grassmannian length.

As a result, the circle will become space filling as it winds around at a constant rate of curvature to explore many dimensions!

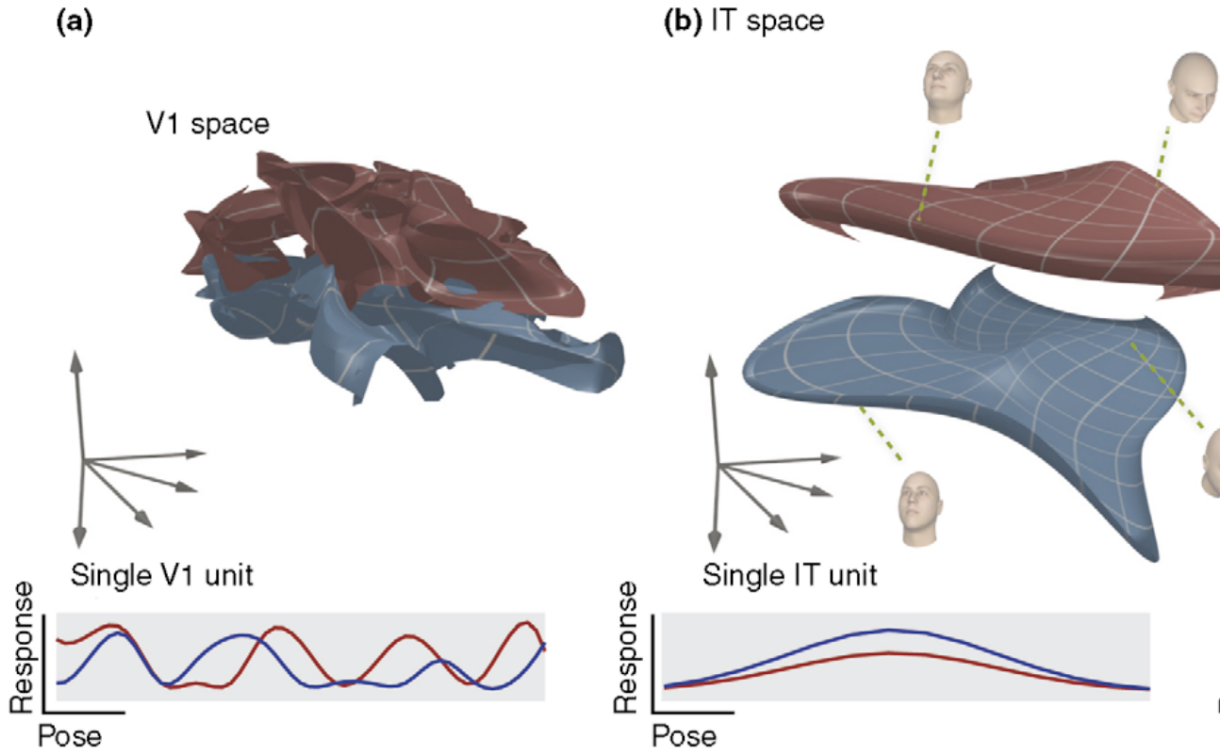
Exponential expressivity is not achievable by shallow nets



Consider a shallow network with 1 hidden layer $\mathbf{x}^1$, one input layer $\mathbf{x}^0$, with $\mathbf{x}^1 = \phi(\mathbf{W}^1\mathbf{x}^0) + \mathbf{b}^1$, and a linear readout layer. How complex can the hidden representation be as a function of its width N_1, relative to the results above for depth? We prove a general upper bound on $\mathcal{L}^E$ (see SM):

Theorem 1. *Suppose $\phi(h)$ is monotonically non-decreasing with bounded dynamic range R, i.e. $\max_h \phi(h) - \min_h \phi(h) = R$. Further suppose that $\mathbf{x}^0(\theta)$ is a curve in input space such that no 1D projection of $\partial_\theta \mathbf{x}(\theta)$ changes sign more than s times over the range of θ. Then for any choice of $\mathbf{W}^1$ and $\mathbf{b}^1$ the Euclidean length of $\mathbf{x}^1(\theta)$, satisfies $\mathcal{L}^E \leq N_1(1+s)R$.*

Boundary disentangling: theory



How can we mathematically formalize the notion of disentangling in deep networks?

How do we use this mathematical formalization to quantitatively assess the disentangling power of deep versus shallow networks?

Boundary disentangling: theory

$$y = \mathrm{sgn}(\boldsymbol{\beta} \cdot \mathbf{x}^D - \beta_0)$$

$$(\boldsymbol{\beta} \cdot \mathbf{x}^D - \beta_0) = 0$$

A linear classifier in the top layer

Implements a hyperplane decision boundary in final layer

$$G(\mathbf{x^0}) = (\boldsymbol{\beta} \cdot \mathbf{x}^D(\mathbf{x^0}) - \beta_0) = 0$$

Yielding a curved co-dimension 1 decision boundary in the input layer

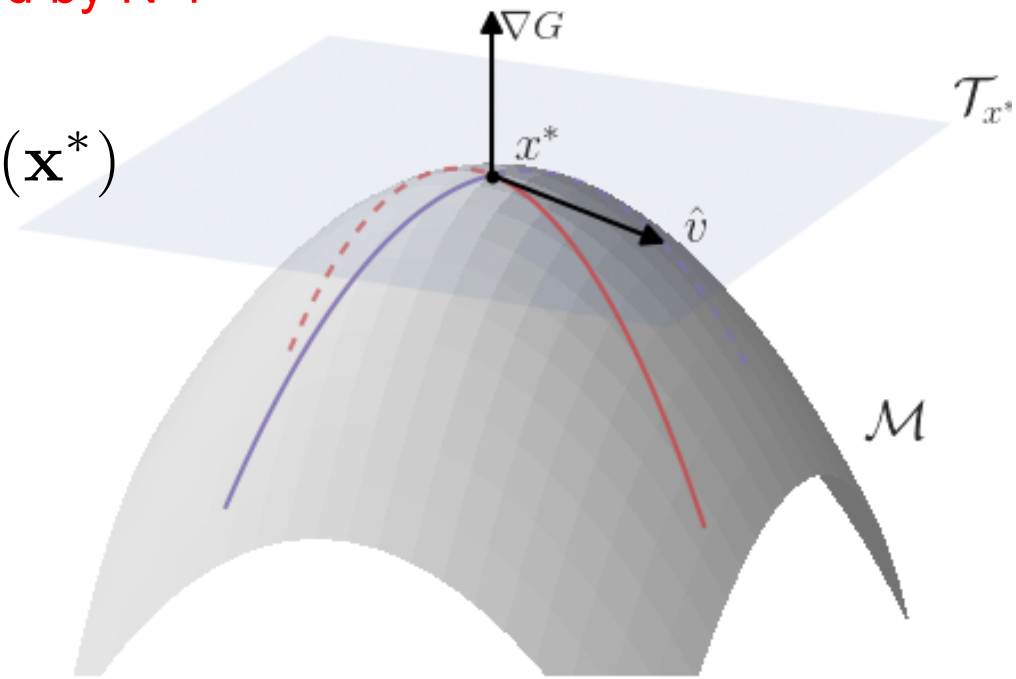
Its curvature at a point is characterized by N-1 principal curvatures:

$$\kappa_1(\mathbf{x}^*) \geq \kappa_2(\mathbf{x}^*) \geq \cdots \geq \kappa_{N-1}(\mathbf{x}^*)$$

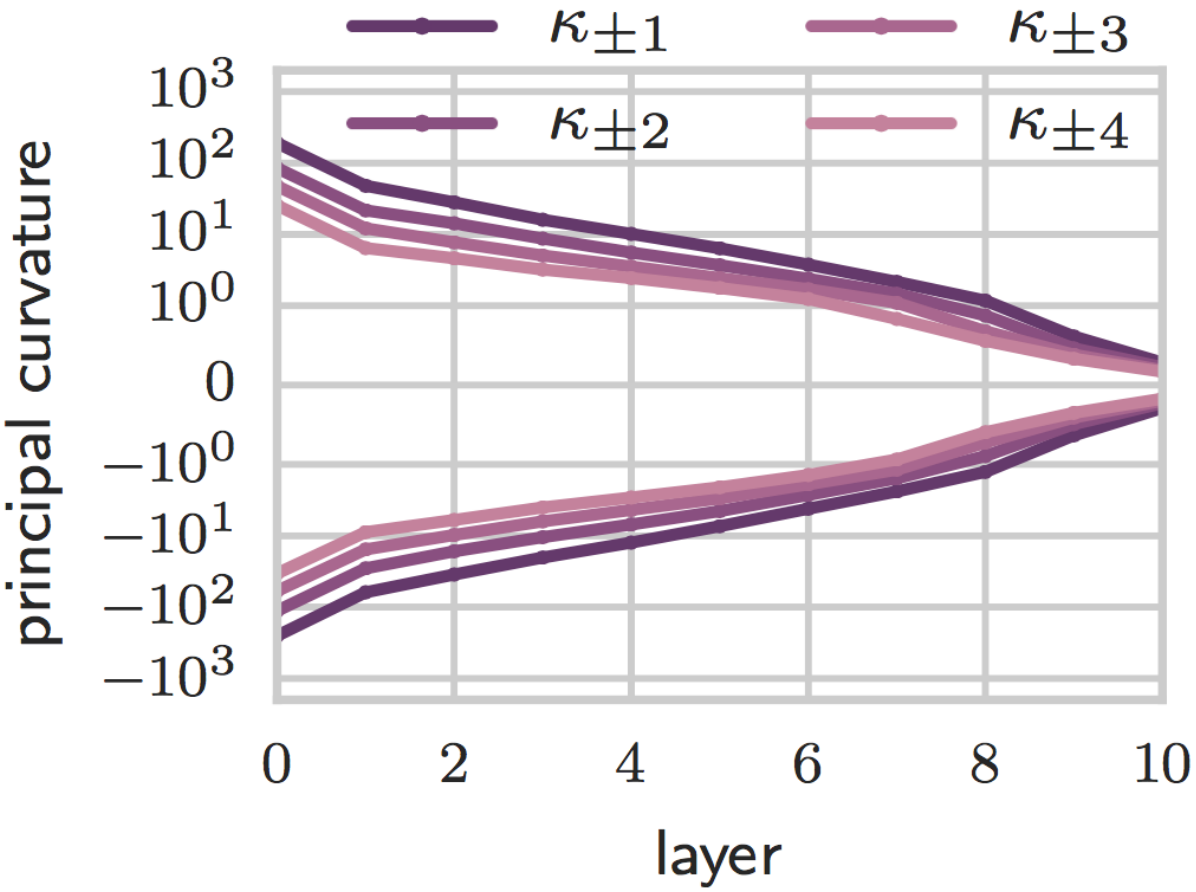
They are the eigenvalues of:

$$\mathcal{H} = ||\vec{\nabla} G||_2^{-1} \mathbf{P} \frac{\partial^2 G}{\partial \mathbf{x} \partial \mathbf{x}^T} \mathbf{P}$$

$$\mathbf{P} = \mathbf{I} - \widehat{\nabla G} \widehat{\nabla G}^T$$



Boundary disentangling: experiment



The principal curvatures of decision boundaries in the chaotic regime grow ***exponentially*** with depth!

Thus exponentially curved manifolds in input space can be flattened to hyperplanes even by deep random networks!

Summary

We have combined Riemannian geometry with dynamical mean field theory to study the emergent deterministic properties of signal propagation in deep nonlinear nets.

We derived analytic recursion relations for Euclidean length, correlations, curvature, and Grassmannian length as simple input manifolds propagate forward through the network.

We obtain an excellent quantitative match between theory and simulations.

Our results reveal the existence of a transient chaotic phase in which the network expands input manifolds without straightening them out, leading to "space filling" curves that explore many dimensions while turning at a constant rate. The number of turns grows exponentially with depth.

Such exponential growth does not happen with width in a shallow net.

Chaotic deep random networks can also take exponentially curved N-1 Dimensional decision boundaries in the input and flatten them into Hyperplane decision boundaries in the final layer: exponential disentangling!

Outline of talks I, II and III

1. Oldies but goodies

    1. Models of single neurons: Hodgkin Huxley to Hopfield
    2. The Hopfield model
    3. The perceptron learning algorithm: memorization and generalization
    4. Unsupervised learning: PCA, ICA, Sparse Coding

2. High dimensional statistics: theory and experiment

    1. The best way to do regression in high dimensions (Replica theory)
    2. Recovering neural state space dynamics (Rand proj / Matrices / Free prob)
    3. Figuring out how neural circuits learn (Tensor decompositions)

3. Deep learning: theory and practice

    1. Speeding up deep learning (Dynamic criticality)
    2. Error landscape of deep networks (Stat mech of random Gaussian fields)
    3. Deep generative models (Non-equilbrium thermodynamics)
    4. Expressive power of deep networks (Riemannian geometry and chaos theory)
    5. Application: deep models of the retina: the first step in seeing

References

- M. Advani and S. Ganguli, An equivalence between high dimensional Bayes optimal inference and M-estimation, NIPS 2016.
- M. Advani and S. Ganguli, Statistical mechanics of optimal convex inference in high dimensions, Physical Review X, 6, 031034, 2016.
- A. Saxe, J. McClelland, S. Ganguli, Learning hierarchical category structure in deep neural networks, Proc. of the 35th Cognitive Science Society, pp. 1271-1276, 2013.
- A. Saxe, J. McClelland, S. Ganguli, Exact solutions to the nonlinear dynamics of learning in deep neural networks, ICLR 2014.
- M. Advani and S. Ganguli, An equivalence between high dimensional Bayes optimal inference and M-estimation, NIPS 2016.
- Y. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, Y. Bengio, Identifying and attacking the saddle point problem in high-dimensional non-convex optimization, NIPS 2014.
- B. Poole, S. Lahiri, M. Raghu, J. Sohl-Dickstein, and S. Ganguli, Exponential expressivity in deep neural networks through transient chaos, NIPS 2016.
- S. Schoenholz, J. Gilmer, S. Ganguli, and J. Sohl-Dickstein, Deep information propagation, https://arxiv.org/abs/1611.01232, under review at ICLR 2017.
- S. Lahiri, J. Sohl-Dickstein and S. Ganguli, A universal tradeoff between energy speed and accuracy in physical communication. https://arxiv.org/abs/1603.07758
- A memory frontier for complex synapses, S. Lahiri and S. Ganguli, NIPS 2013.
- Modelling arbitrary probability distributions using non-equilibrium thermodynamics, J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, S. Ganguli, ICML 2015.
- Deep Knowledge Tracing, C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. Guibas, J. Sohl-Dickstein, NIPS 2015.
- Deep learning models of the retinal response to natural scenes, L. McIntosh, N. Maheswaranathan, S. Ganguli, S. Baccus, NIPS 2016.
- http://ganguli-gang.stanford.edu