# Local and Global Methods to Identify Local and Global Structure in Data Science

**Michael W. Mahoney**

**ICSI and Dept of Statistics, UC Berkeley**

*( For more info, see:*
*http://www.stat.berkeley.edu/~mmahoney*

February 2020

# Start with the conclusions

Practice:

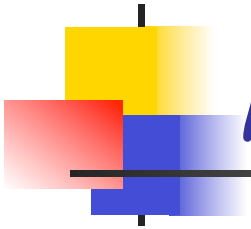
- Some problems are "global":
  - total click-through rate; total number of stars; human population genetics
- Some problems are "local":
  - meaningful communities; quasars that identify new physics; personalized medicine

Theory:

- Most objectives are global:
  - global partition, global eigenvetor, global minimum
- Some objectives are local:
  - find good cluster near a seed set
- Most algorithms are local:
  - take one step of gradient descent
  - but theory then makes a statement about global objective
  - but practice then deviates from theory and applies it locally

*Question: How to go forward?*

# Outline



# Matrix computations

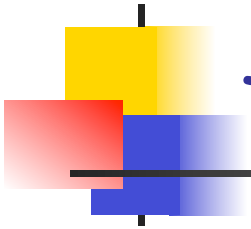
Eigendecompositions, QR, SVD, least-squares, etc.

Traditional algorithms:

- compute "exact" answers to, say, 10 digits as a black box
- assume the matrix is in RAM and minimize flops

But they are NOT well-suited for:

- with missing or noisy entries
- problems that are very large
- distributed or parallel computation
- when communication is a bottleneck
- when the data must be accessed via "passes"



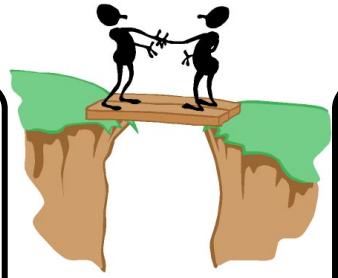
# The general idea …

- Randomly sample columns/rows/entries of the matrix, with carefully-constructed *importance sampling probabilities*, to form a randomized sketch
- Preprocess the matrix with random projections, to form a randomized sketch by sampling columns/rows *uniformly*
- Use the sketch to compute an approximate solution to the original problem w.h.p.
- Resulting sketches are "similar" to the original matrix in terms of singular value and singular vector structure, e.g., w.h.p. are bounded distance from the original matrix

# History of Randomized Matrix Algs

## Theoretical origins

- theoretical computer science, convex analysis, etc.
- Johnson-Lindenstrauss
- Additive-error algs
- Good worst-case analysis
- No statistical analysis

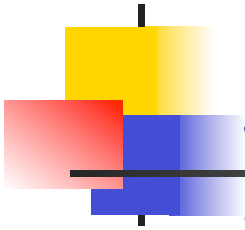


## Practical applications

- NLA, ML, statistics, data analysis, genetics, etc
- Fast JL transform
- Relative-error algs
- Numerically-stable algs
- Good statistical properties

## How to "bridge the gap"?

- decouple randomization from linear algebra
- importance of statistical leverage scores!



# Statistical leverage, coherence, etc.

Mahoney and Drineas (2009, PNAS); Drineas, Magdon-Ismail, Mahoney, and Woodruff (2012, ICML)

**Definition**: Given a "tall" n x d matrix A, i.e., with n > d, let U be *any* n x d orthogonal basis for span(A), & let the d-vector $U_{(i)}$ be the $i^{th}$ *row* of U. Then:

- the statistical leverage scores are $\lambda_i = ||U_{(i)}||_2^2$ , for $i \in \{1,\ldots,n\}$
- the coherence is $\gamma = \max_{i \in \{1,\ldots,n\}} \lambda_i$
- the (i,j)-cross-leverage scores are $U_{(i)}^T U_{(j)} = \langle U_{(i)}, U_{(j)} \rangle$

**Note**: There are extension of this to:

- "fat" matrices A, with n, d are large and low-rank parameter k
- L1 and other p-norms

# Outline

# Local versus global

Local or small-scale properties (the proverbial needle in haystack):

- In machine learning: nearest neighbor models/rules
- In graph analytics: ego networks near a person in a social network
- This information is often the most reliable

Global or large-scale properties (the proverbial haystack):

- In machine learning: global latent factor models
- In graph analytics: large-scale community structure in social network

Algorithmic/statistical tools make strong local-global assumptions:

- Data are some large-scale structure with lots (enough to average over to go to some asymptotic limit) of small-scale noise
- Recursive spectral paritioning
- Typical ML objectives, e.g., MSE, bias toward large classes

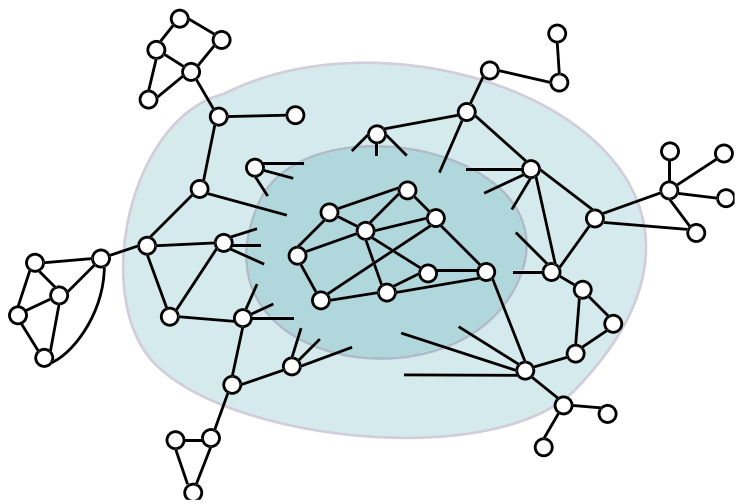
# Networks and networked data

Lots of "networked" data!!

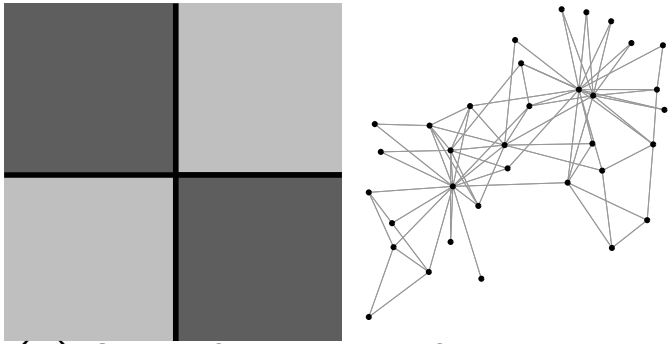
- technological networks (AS, power-grid, road networks)
- biological networks (food-web, protein networks)
- social networks (collaboration networks, friendships)
- information networks (co-citation, blog cross-postings, advertiser-bidded phrase graphs ...)
- language networks (semantic networks ...)
- ...

Interaction graph model of networks:

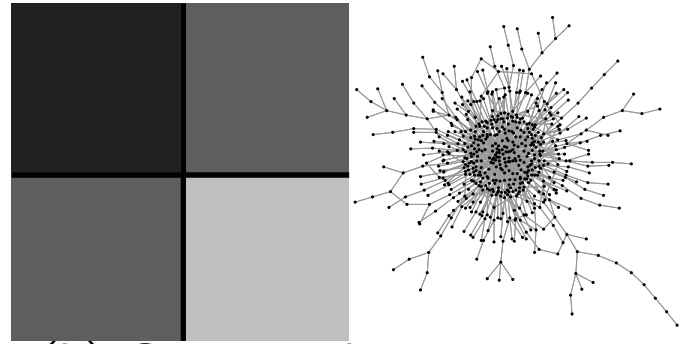
- Nodes represent "entities"
- Edges represent "interaction" between pairs of entities



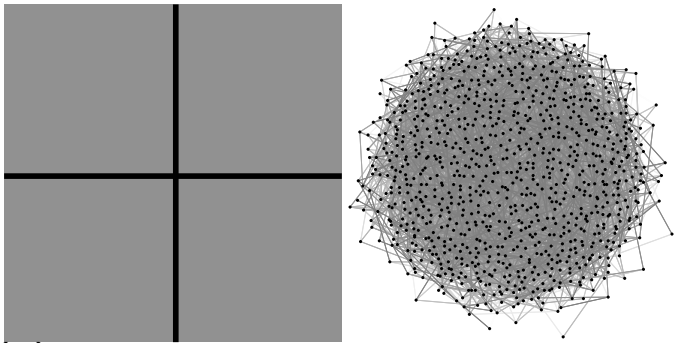
# Possible ways a graph might look



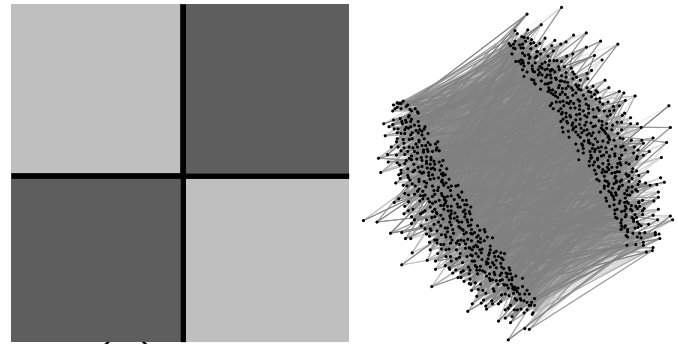
(a) Low-dimensional structure



(b) Core-periphery structure

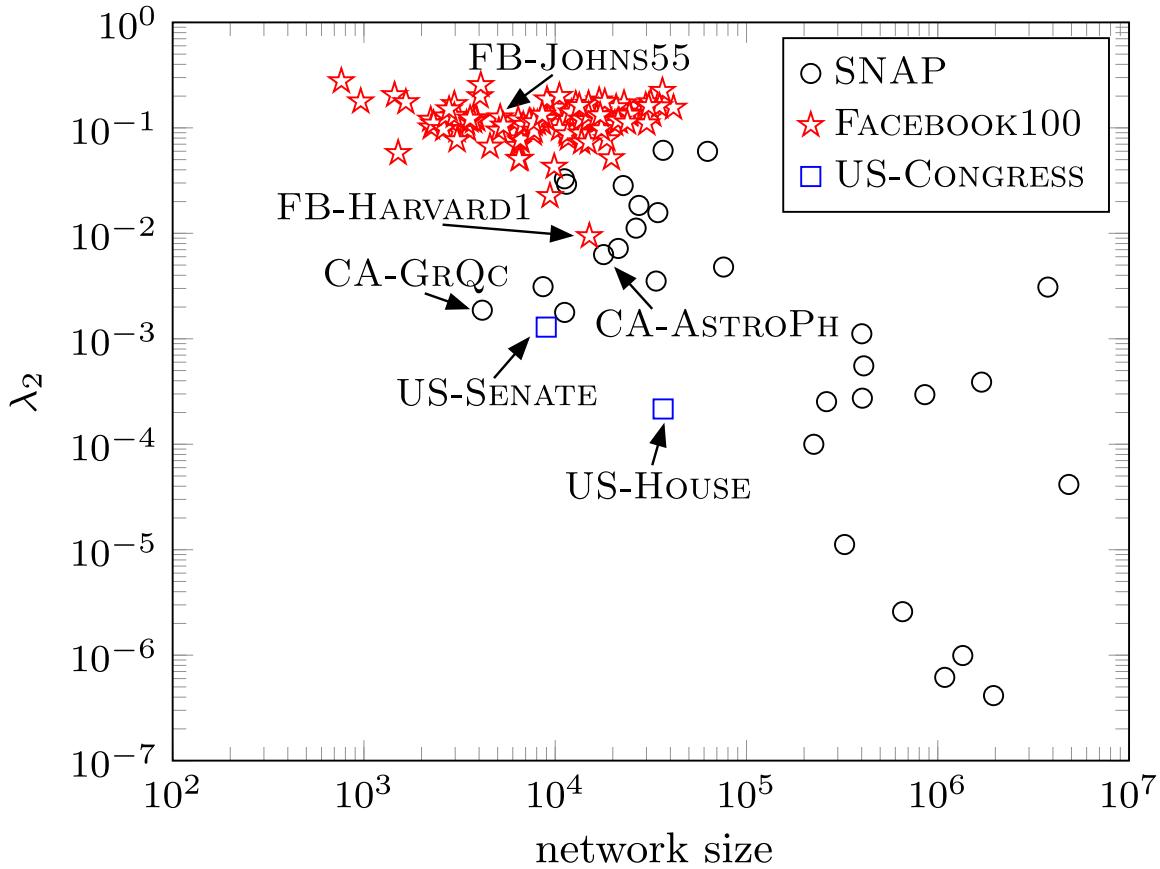


(c) Expander or complete graph



(d) Bipartite structure

# Scatter plot of $\lambda_2$ (the "Fiedler value") for real networks




Conventional wisdom: down is good.

*Question: does this plot really tell us much about these networks?*

# Communities, conductances, and NCPs

Let $A$ be the adjacency matrix of $G = (V, E)$.
The *conductance* $\phi$ of a set $S$ of nodes is

$$\phi(S) = \frac{\sum_{i \in S, j \notin S} A_{ij}}{\min\{A(S), A(\bar{S})\}}, \quad \text{where} \quad A(S) = \sum_{i \in S} \sum_{j \in V} A_{ij}$$

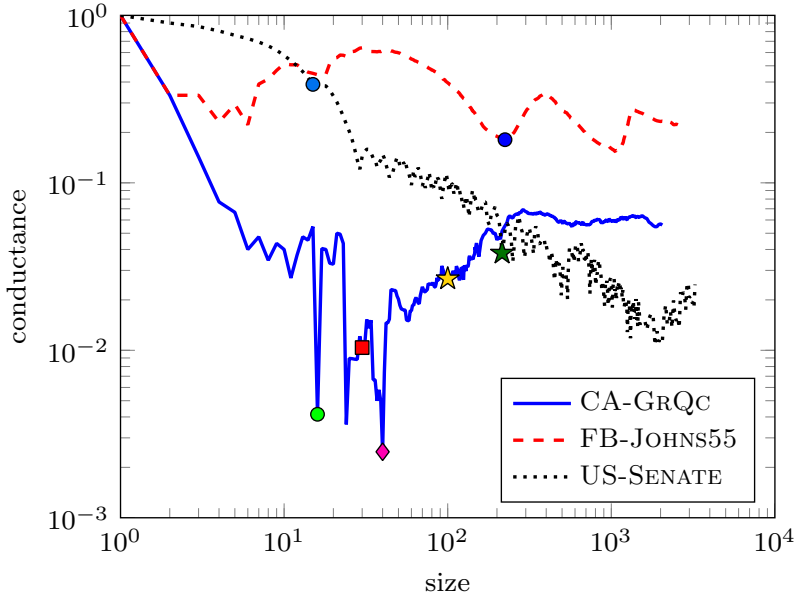
The Network Community Profile (NCP) of the graph is

$$\Phi(k) = \min_{S \subset V, |S| = k} \phi(S)$$

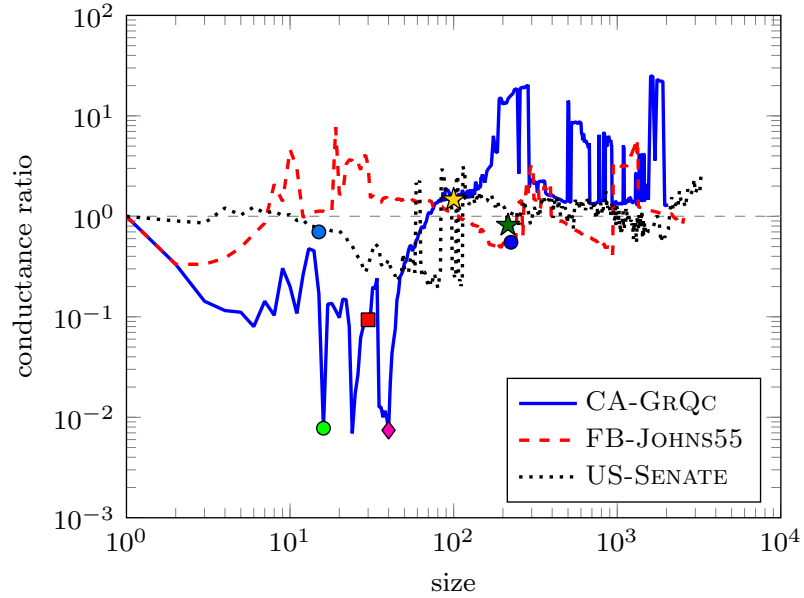
*Just as conductance captures a Surface-Area-To-Volume, the NCP*

- *captures a size-resolved Surface-Area-To-Volume notion*
- *captures the idea of local size-resolved bottlenecks to diffusion*

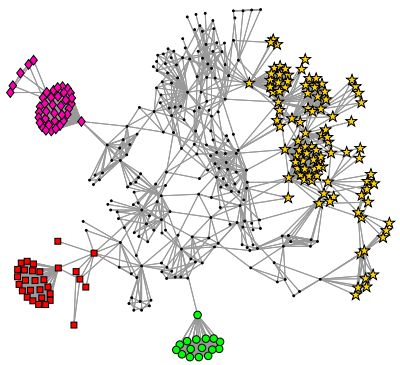
# Three different types of real networks



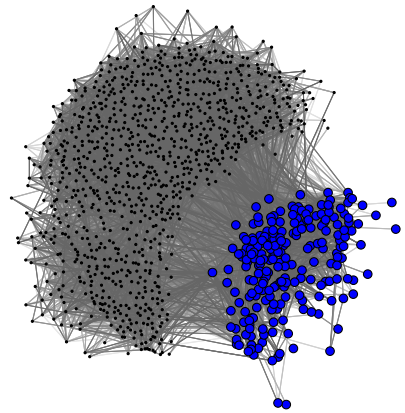

(e) NCP: conductance value of best conductance set, as function of size



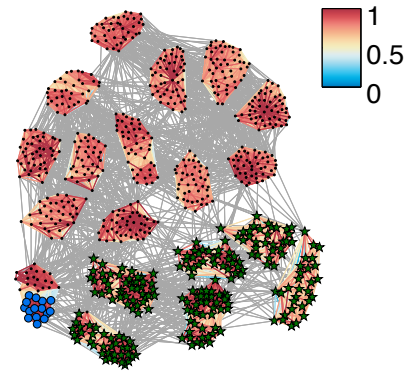

(f) CRP: ratio of internal to external conductance, as function of size



(g) CA-GrQc

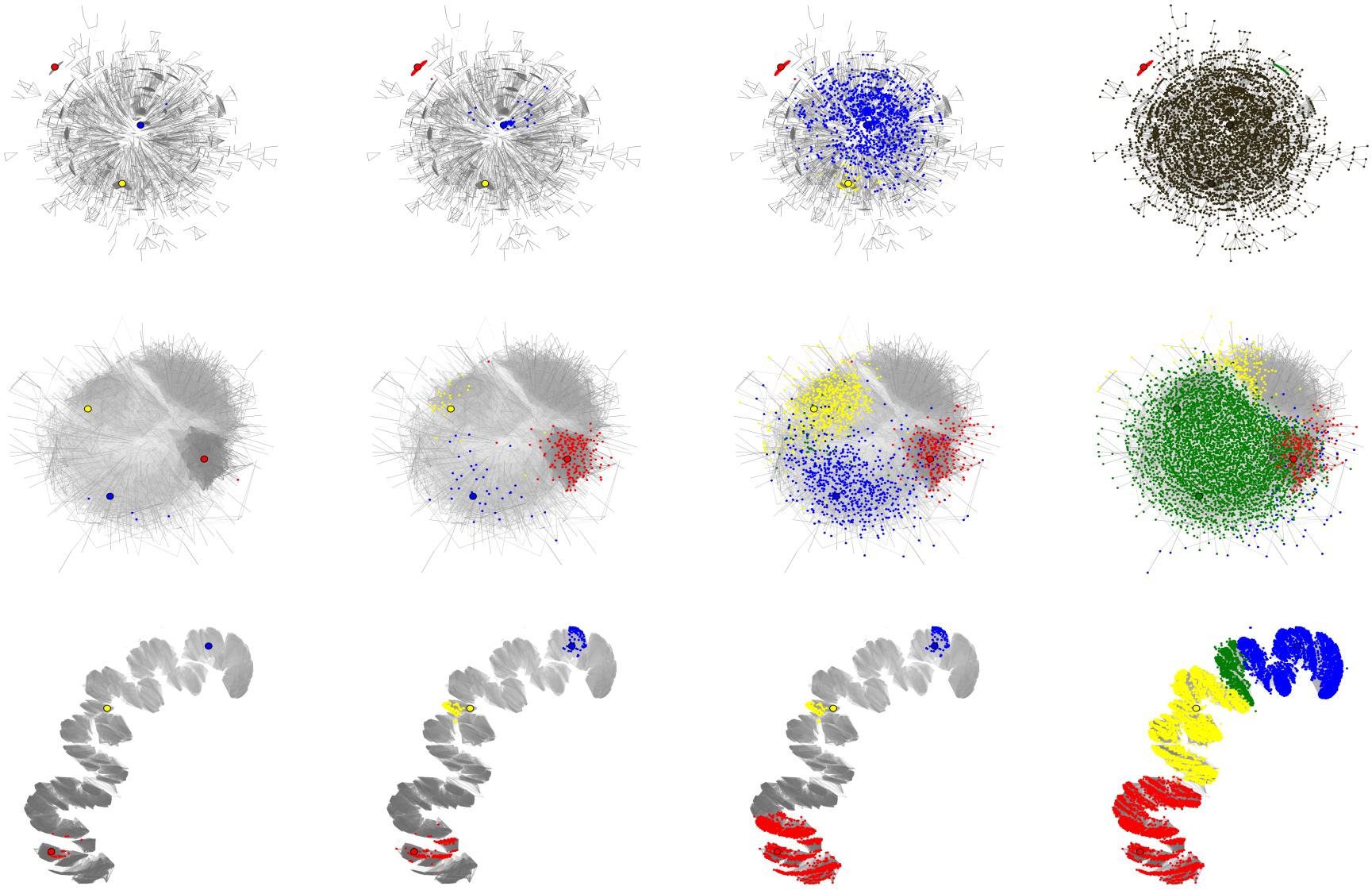


(h) FB-Johns55




(i) US-Senate

# Information propagates local-to-global in different networks in different ways



# Summary of lessons learned

**Local-global properties** of real data are *very* different

- than practical/theoretical people implicitly/explicitly assume

**Local graph algorithms** (local spectral methods) were big winner

- For both algorithmic and statistical reasons

**Little design decisions** made a big difference

- Details of how deal with truncation and boundary conditions are not second-order issues when graphs are expander-like

**Approximation algorithm usefulness uncoupled from theory**

- Often useful when they implicitly regularize

# Some basics on kernels and SPSD matrices

- Given $\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$ and $\kappa : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, the $n \times n$ matrix with elements $A_{ij} = \kappa(\vec{x}_i, \vec{x}_j)$ is the kernel matrix of $\kappa$ w.r.t. $\vec{x}_1, \ldots, \vec{x}_n$.
  - Appropriate $\kappa$ ensures $A$ is SPSD
  - $A_{ij}$ measure a ($\kappa$-defined) similarity between points $i$ and $j$.
  - $\kappa$ determines a *feature map* $\Phi_\kappa : \mathbb{R}^d \to \mathbb{R}^\infty$ s.t. similarity of $\vec{x}_i$ and $\vec{x}_j$ in feature space is measure by $A_{ij} = \langle \Phi_\kappa(\vec{x}_i), \Phi_\kappa(\vec{x}_j) \rangle$.

- When $\kappa$ is the usual Euclidean inner-product, so that $A_{ij} = \langle \vec{x}_i, \vec{x}_k \rangle$, then $A$ is called a **Linear Kernel** matrix.

- **Gaussian RBF Kernel** matrices, defined by $A^\sigma_{ij} = \exp\left(\frac{-\|\vec{x}_i - \vec{x}_j\|_2^2}{\sigma^2}\right)$, correspond to the similarity measure $\kappa(\vec{x}, \vec{y}) = \exp(-\|\vec{x} - \vec{y}\|_2^2/\sigma^2)$.
  - $\sigma$ defines "size scale" over which points "see" each other.
  - Can "sparsify" $A$ by decreasing $\sigma$.
  - Can also sparsify the matrix $A$ by zeroing out entries.

# Size-dependence of optimal neighbor width

Wang, Li, Mahoney, and Darve (2015)

EMG Physical Action Data Set

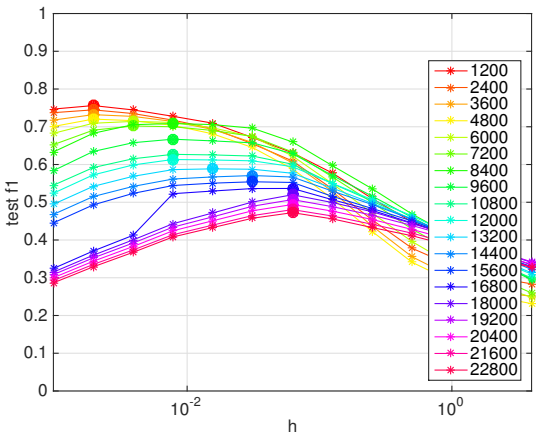
- 10 normal & 10 aggressive physical actions that measure human activity.

| Class | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | |
|---|---|---|---|---|---|---|---|---|---|---|
| $r_i^2$ | 1e-4 | 3e-4 | 1.2e-3 | 2.8e-3 | 2.6e-2 | 2.7e-2 | 3.6e-2 | 4.6e-1 | 1.0 | |
| $d_i^2$ | 3e-4 | 2.3e-3 | 9.9e-3 | 1.7e-2 | 1.9e-1 | 2.3e-1 | 1.4e-1 | 1.4 | 2.1 | |
| Class | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| $r_i^2$ | 1.0 | 1.6 | 1.9 | 2.2 | 2.2 | 2.6 | 2.8 | 2.9 | 3.0 | 3.1 |
| $d_i^2$ | 2.0 | 3.4 | 4.4 | 4.2 | 4.3 | 5.3 | 5.4 | 5.4 | 5.7 | 5.9 |

Table: Pair-wise distance ($d_i$) and distance to center ($r_i$) for each class

We order the classes by their "size" $r_i$ and group them as follows:

```
g1 = smallest class,
g2 = union(smallest & 2nd smallest class),
..., g20 = all the data
```



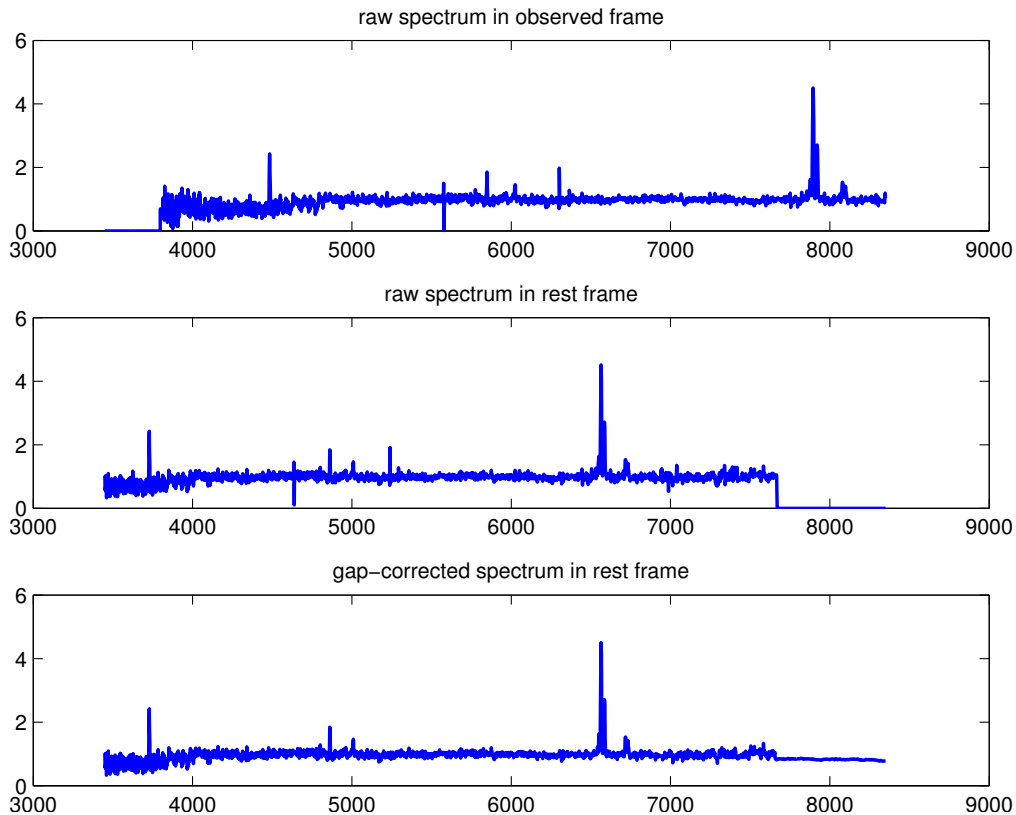
# Use case[1]: Galactic spectra from SDSS

$x_i \in \mathbb{R}^{3841}$, $N \approx 500k$

photon fluxes in $\approx 10$ Å wavelength bins

preprocessing corrects for redshift, gappy regions

normalized by median flux at certain wavelengths




[1] Also results in neuroscience as well as genetics and mass spec imaging.

# Global embedding: effect of $k$

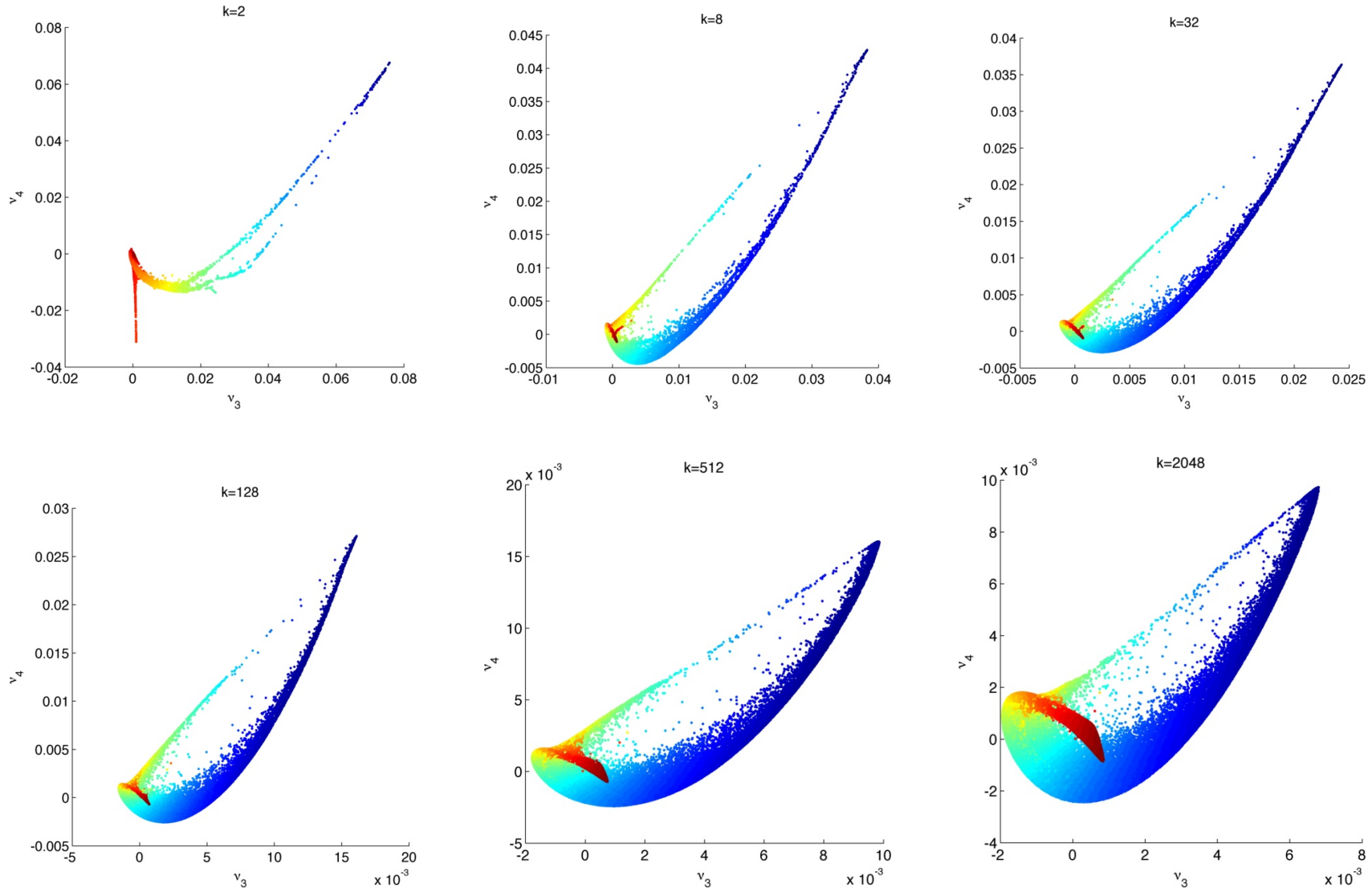
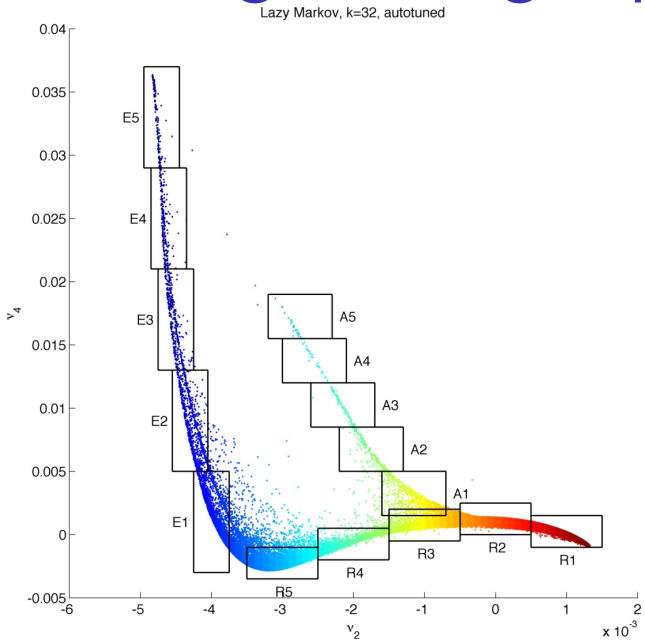
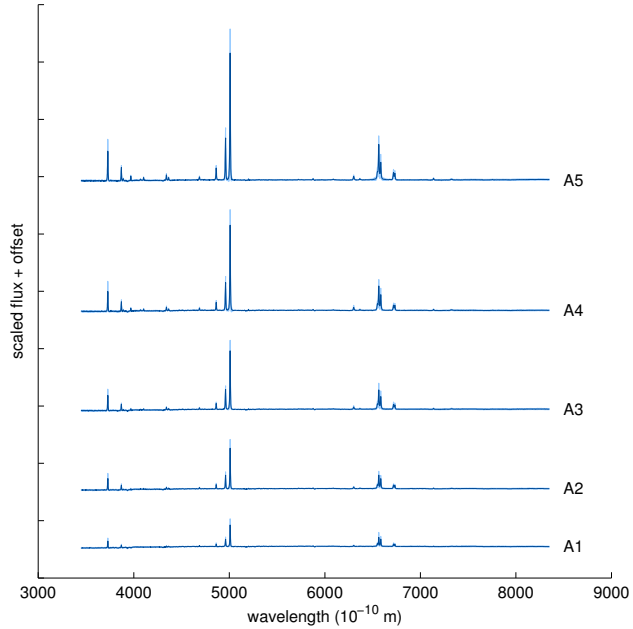
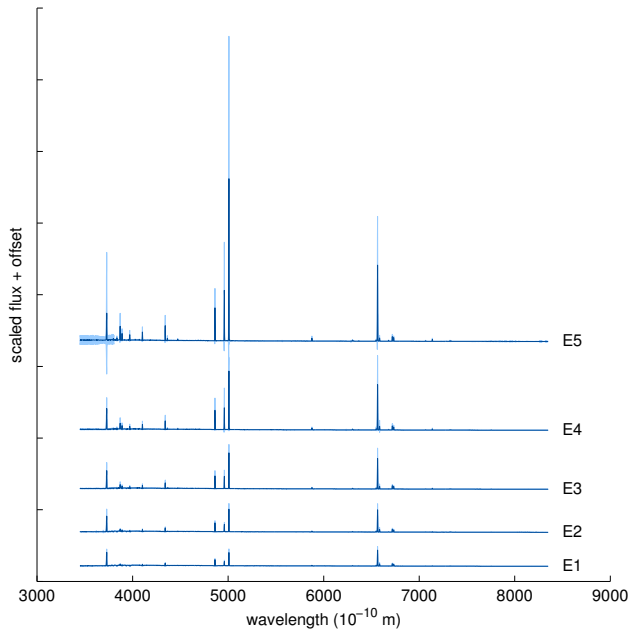


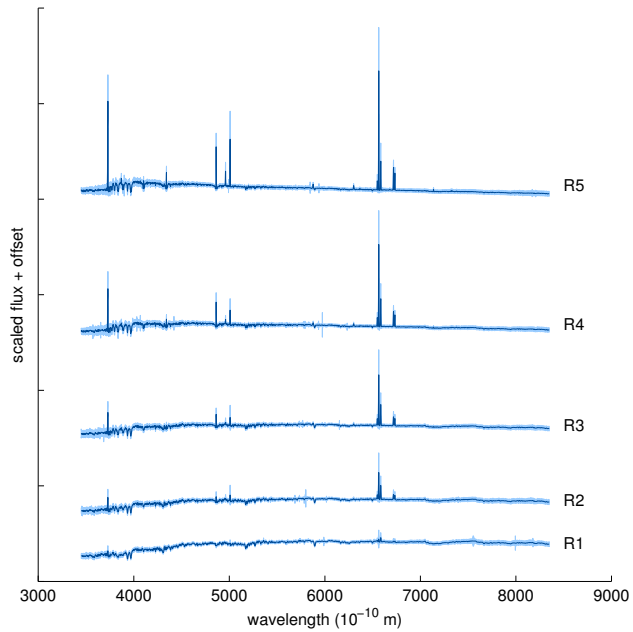

Figure: Eigenvectors 3 and 4 of Lazy Markov operator, $k = 2 : 2048$

# Global embedding: average spectra



# MOV optimization approach to local spectral methods

(Mahoney, Orecchia, and Vishnoi, 2009; Hansen and Mahoney, 2013; Lawlor, Budavari, Mahoney, 2015)

Suppose we have:

1. a seed vector $s = \chi_S$, where $S$ is a subset of data points
2. a correlation parameter $\kappa$

**MOV objective.** The first semi-supervised eigenvector $w_2$ solves:

$$\begin{aligned} \text{minimize} \quad & x^T L x \\ \text{subject to} \quad & x^T D x = 1 \\ & x^T D \mathbb{1} = 0 \\ & x^T D s \geq \sqrt{\kappa} \end{aligned}$$

Similarly for $w_t$ with addition constraints $x^T D w_j = 0,\ j < t$.

**Theorem.** Solution can be found by solving a linear equation. It is "quadratically good" (with a local version of Cheeger's Inequality).

# Local embedding: scale parameter and effect of seed

For an appropriate choice of $c$ and $\gamma = \gamma(\kappa) < \lambda_2$, one can show

$$\begin{aligned} w_2 &= c(L - \gamma D)^+ Ds \\ &= c(L_G - \gamma L_{k_n})^+ Ds \end{aligned}$$

(In practice, binary search to find "correct" $\gamma$.)

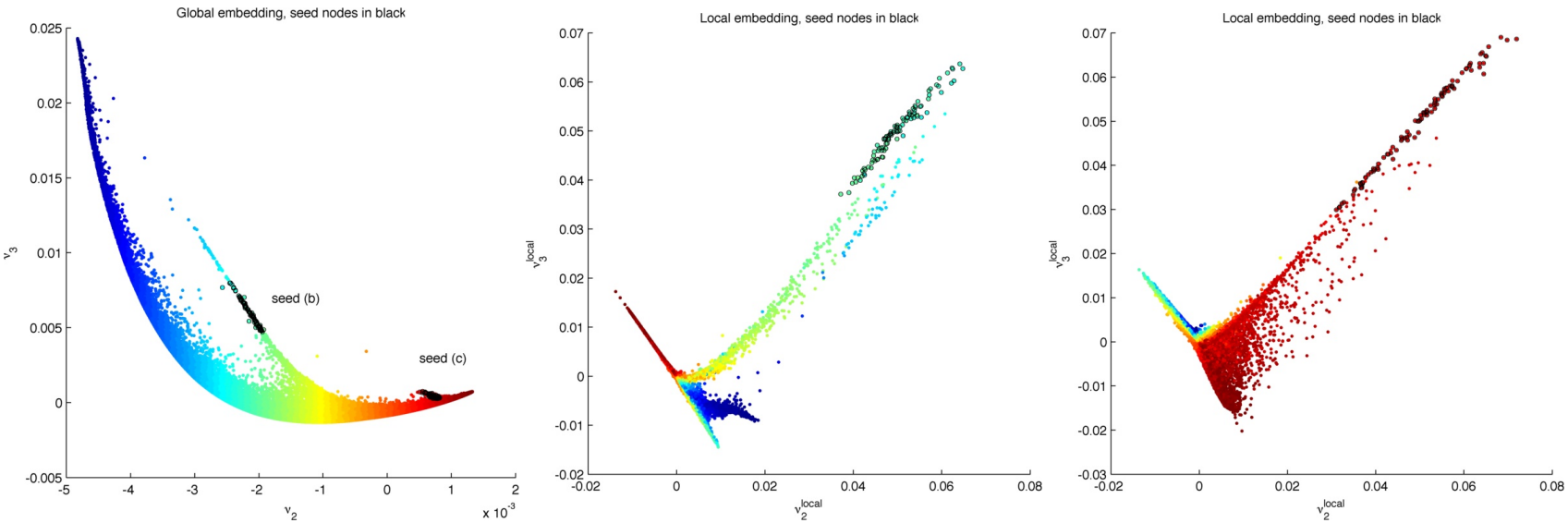
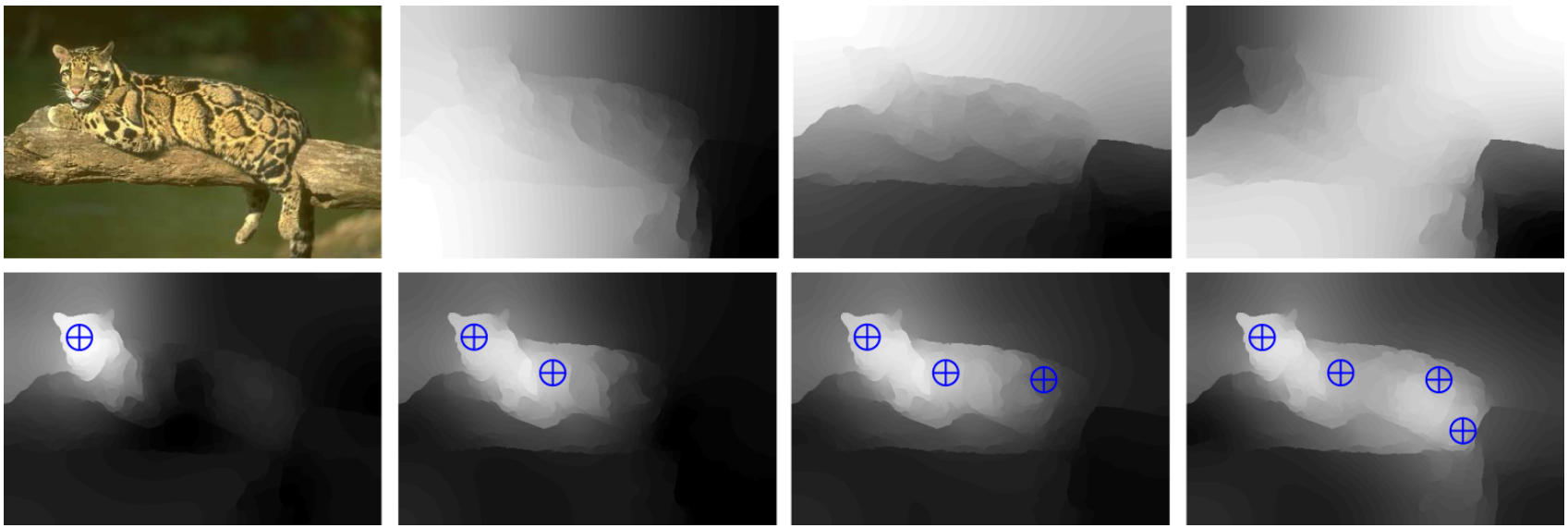



Figure: (left) Global embedding with seeds in black. (middle, right) Local embeddings using specified seeds.

# Pictorial illustration of what can be discovered

Mahoney, Orecchia, and Vishnoi, (2009); Maji, Vishnoi,and Malik (2011); Hansen and Mahoney, (2013)



- Cannot find the tiger with global eigenvectors.
- Can find the tiger with our LocalSpectral method!

# Outline

# Motivations: towards a Theory of Deep Learning

**Theoretical**: deeper insight into *Why Deep Learning Works*?

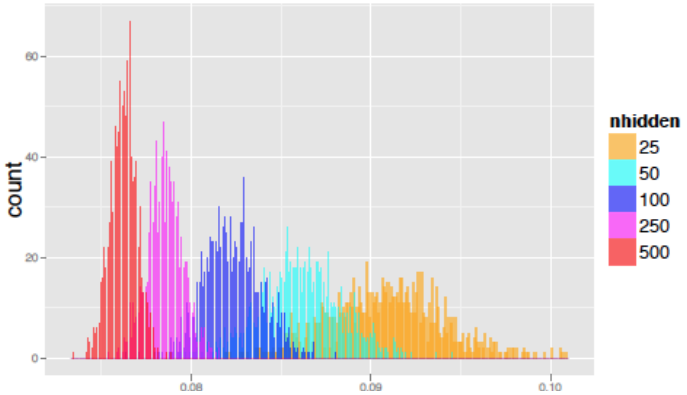
- convex versus non-convex optimization?
- explicit/implicit regularization?
- is / why is / when is deep better?
- VC theory versus Statistical Mechanics theory?
- ...

**Practical**: use insights to improve engineering of DNNs?

- when is a network fully optimized?
- can we use labels and/or domain knowledge more efficiently?
- large batch versus small batch in optimization?
- designing better ensembles?
- ...

# Motivations: towards a Theory of Deep Learning

DNNs as spin glasses, Choromanska et al. 2015




Looks exactly like old protein folding results (late 90s)

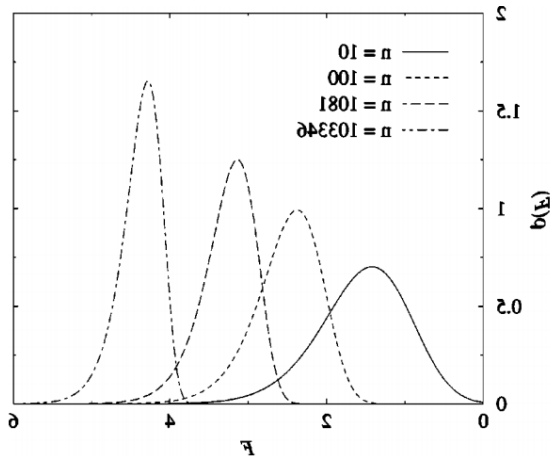
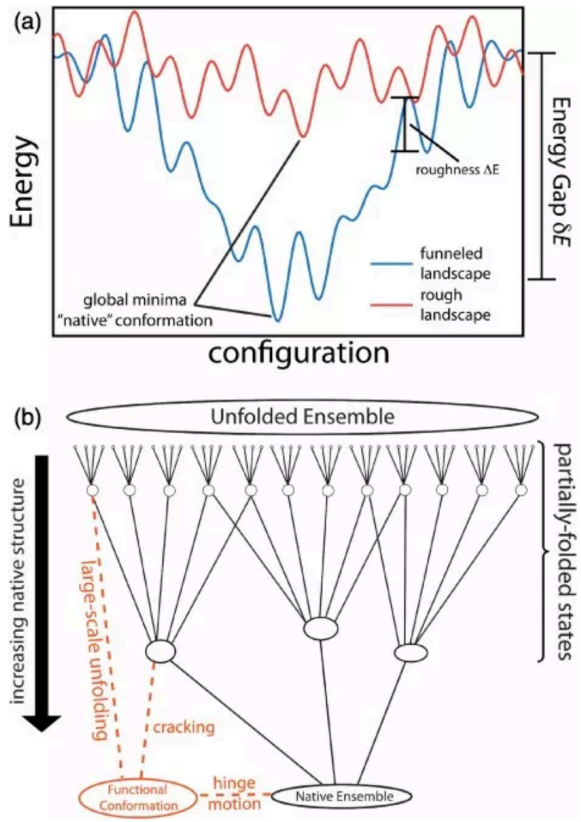



FIG. 1. Plot of the foldability distribution $p(\mathcal{F})$ for different numbers of compact states ($n = 10$, 100, 1081, 10 346), calculated using the random energy model.

Energy Landscape Theory




Energy Landscape of MultiScale Spin Glass Model

Completely different picture of DNNs

Raises broad questions about Why Deep Learning Works

# Set up: the Energy Landscape

Energy/Optimization function:

$$E_{DNN} = h_L(\mathbf{W}_L \times h_{L-1}(\mathbf{W}_{L-1} \times h_{L-2}(\cdots) + \mathbf{b}_{L-1}) + \mathbf{b}_L)$$

Train this on labeled data $\{d_i, y_i\} \in \mathcal{D}$, using Backprop, by minimizing loss $\mathcal{L}$:

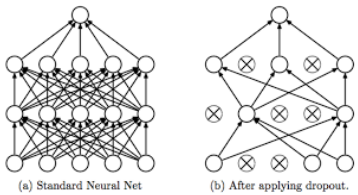
$$\min_{W_l, b_l} \mathcal{L} \left( \sum_i E_{DNN}(d_i) - y_i \right)$$

$E_{DNN}$ is "the" *Energy Landscape*:

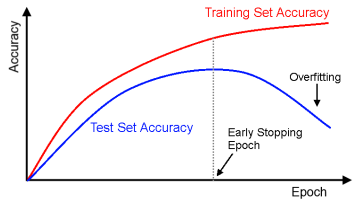
- The part of the optimization problem parameterized by the heretofore unknown elements of the weight matrices and bias vectors, and as defined by the data $\{d_i, y_i\} \in \mathcal{D}$
- Pass the data through the Energy function $E_{DNN}$ multiple times, as we run Backprop training
- The Energy Landscape* is *changing* at each epoch

*i.e., the optimization function that is *nominally* being optimized

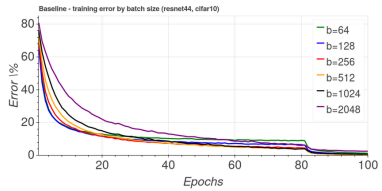
# Motivations: what is regularization?



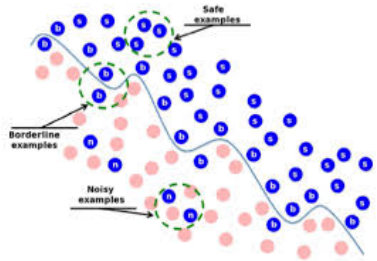

(a) Dropout.




(b) Early Stopping.




(c) Batch Size.




(d) Noisify Data.

Every adjustable *knob* and *switch*—and there are *many*[†]—is regularization.

[†] https://arxiv.org/pdf/1710.10686.pdf

# How we will study regularization

The Energy Landscape is *determined* by layer weight matrices $\mathbf{W}_L$:

$$E_{DNN} = h_L(\mathbf{W}_L \times h_{L-1}(\mathbf{W}_{L-1} \times h_{L-2}(\cdots) + \mathbf{b}_{L-1}) + \mathbf{b}_L)$$

Traditional regularization is applied to $\mathbf{W}_L$:

$$\min_{W_l, b_l} \mathcal{L}\left(\sum_i E_{DNN}(d_i) - y_i\right) + \alpha \sum_l \|\mathbf{W}_l\|$$

*Different types of regularization, e.g., different norms* $\|\cdot\|$*, leave different empirical signatures on* $\mathbf{W}_L$*.*

What we do:

- Turn off "all" regularization.
- Systematically turn it back on, explicitly with $\alpha$ or implicitly with knobs/switches.
- Study empirical properties of $\mathbf{W}_L$.

# *Lots* of DNNs Analyzed

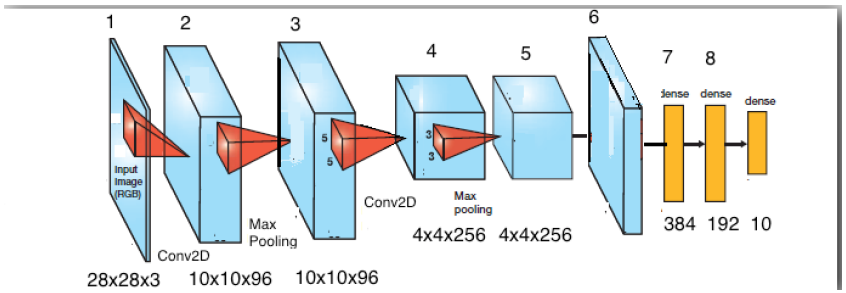
**Question:** What happens to the layer weight matrices $\mathbf{W}_L$?

*(Don't evaluate your method on one/two/three NN, evaluate it on a dozen/hundred.)*

Retrained LeNet5 on MINST using Keras.

Two other small models:

- 3-Layer MLP
- Mini AlexNet




Conv2D MaxPool Conv2D MaxPool FC1 FC2 FC

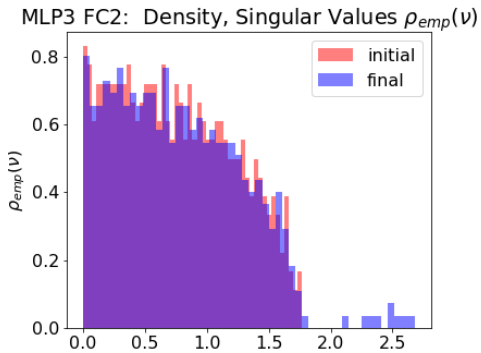
Wide range of state-of-the-art pre-trained models:

- AlexNet, Inception, etc.

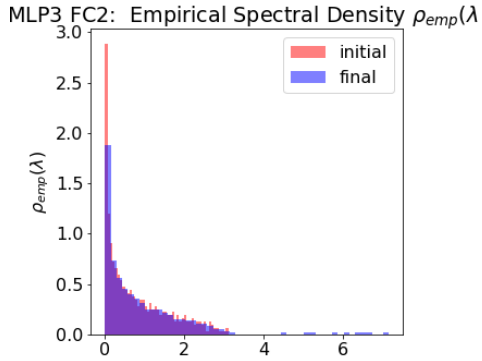
# Matrix complexity: Singular/Eigen Value Densities

$$\mathbf{W} = \mathbf{U}\Sigma\mathbf{V}^T \qquad \nu_i = \Sigma_{ii} \qquad p_i = \nu_i^2 / \sum_i \nu_i^2$$

$$\mathcal{S}(\mathbf{W}) = \frac{-1}{\log(R(\mathbf{W}))} \sum_i p_i \log\ p_i \qquad \mathcal{R}_s(\mathbf{W}) = \frac{\|\mathbf{W}\|_F^2}{\|\mathbf{W}\|_2^2} = \frac{\sum_i \nu_i^2}{\nu_{max}^2}$$




(a) Singular val. density




(b) Eigenvalue density

Figure: Histograms of the Singular Values $\nu_i$ and associated Eigenvalues $\lambda_i = \nu_i^2$.

# Random Matrix Theory 103: Heavy-tailed RMT

Go beyond the (relatively easy) Gaussian Universality class:

- *model* strongly-correlated systems ("signal") with heavy-tailed random matrices.

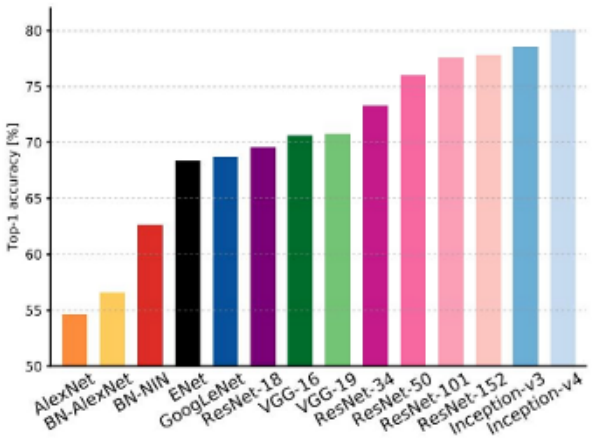
| | Generative Model w/ elements from Universality class | Finite-$N$ Global shape $\rho_N(\lambda)$ | Limiting Global shape $\rho(\lambda),\ N \to \infty$ | Bulk edge Local stats $\lambda \approx \lambda^+$ | (far) Tail Local stats $\lambda \approx \lambda_{max}$ |
|---|---|---|---|---|---|
| Basic MP | Gaussian | MP distribution | MP | TW | No tail. |
| Spiked-Covariance | Gaussian, + low-rank perturbations | MP + Gaussian spikes | MP | TW | Gaussian |
| Heavy tail, $4 < \mu$ | (Weakly) Heavy-Tailed | MP + PL tail | MP | Heavy-Tailed* | Heavy-Tailed* |
| Heavy tail, $2 < \mu < 4$ | (Moderately) Heavy-Tailed (or "fat tailed") | PL** $\sim \lambda^{-(a\mu+b)}$ | PL $\sim \lambda^{-(\frac{1}{2}\mu+1)}$ | No edge. | Frechet |
| Heavy tail, $0 < \mu < 2$ | (Very) Heavy-Tailed | PL** $\sim \lambda^{-(\frac{1}{2}\mu+1)}$ | PL $\sim \lambda^{-(\frac{1}{2}\mu+1)}$ | No edge. | Frechet |

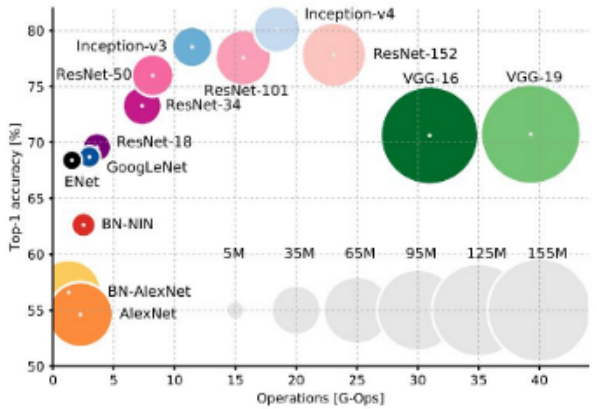
Basic MP theory, and the spiked and Heavy-Tailed extensions we use, including known, empirically-observed, and conjectured relations between them. Boxes marked "*" are best described as following "TW with large finite size corrections" that are likely Heavy-Tailed, leading to bulk edge statistics and far tail statistics that are indistinguishable. Boxes marked "**" are phenomenological fits, describing large ($2 < \mu < 4$) or small ($0 < \mu < 2$) finite-size corrections on $N \to \infty$ behavior.

# Experiments: just apply this to pre-trained models

https://medium.com/@siddharthdas_32104/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-...

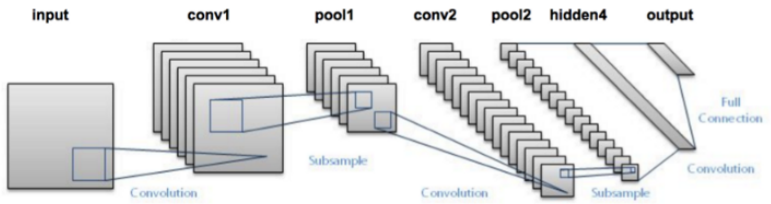
| Year | CNN | Developed by | Place | Top-5 error rate | No. of parameters |
|---|---|---|---|---|---|
| 1998 | LeNet(8) | Yann LeCun et al | | | 60 thousand |
| 2012 | AlexNet(7) | Alex Krizhevsky, Geoffrey Hinton, Ilya Sutskever | 1st | 15.3% | 60 million |
| 2013 | ZFNet() | Matthew Zeiler and Rob Fergus | 1st | 14.8% | |
| 2014 | GoogLeNet(19) | Google | 1st | 6.67% | 4 million |
| 2014 | VGG Net(16) | Simonyan, Zisserman | 2nd | 7.3% | 138 million |
| 2015 | ResNet(152) | Kaiming He | 1st | 3.6% | |

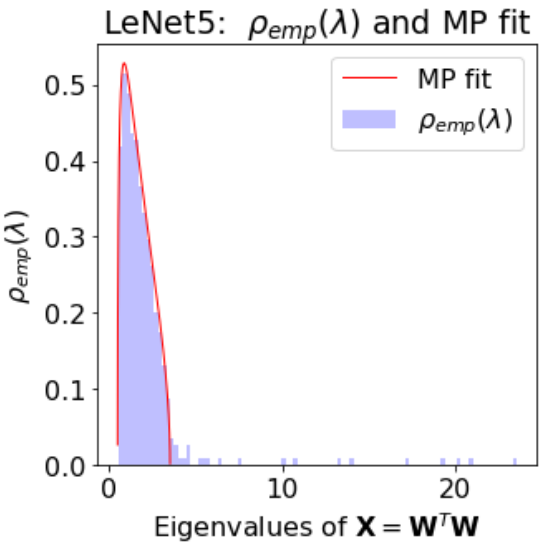



An Analysis of Deep Neural Network Models for Practical Applications, 2017.

# RMT: LeNet5 (an old/small example)



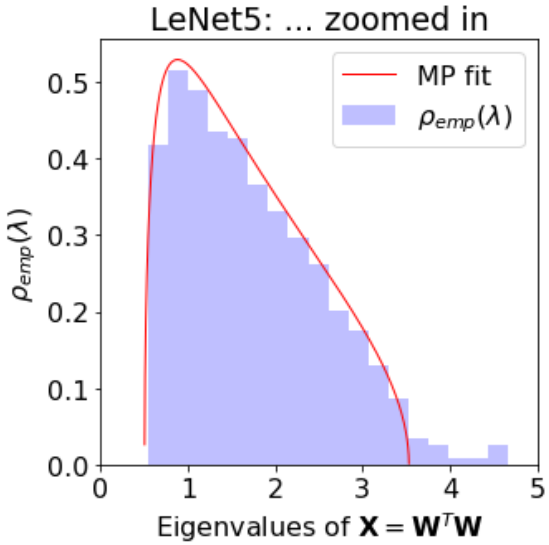
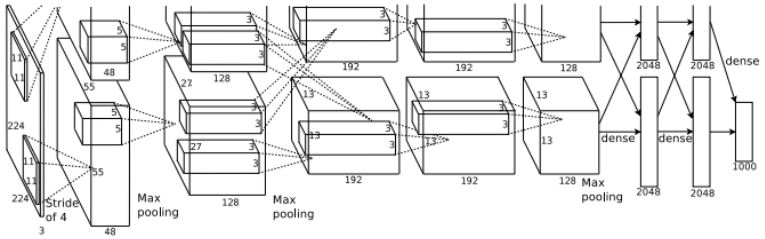
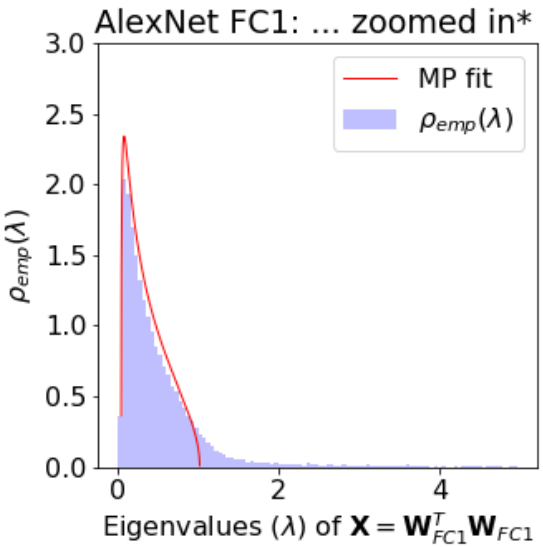




Figure: Full and zoomed-in ESD for LeNet5, Layer FC1.

Marchenko-Pastur Bulk + Spikes

# RMT: AlexNet (a typical modern DNN example)



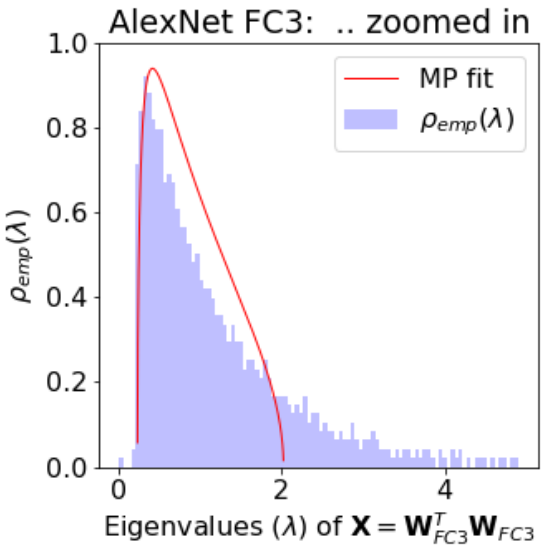
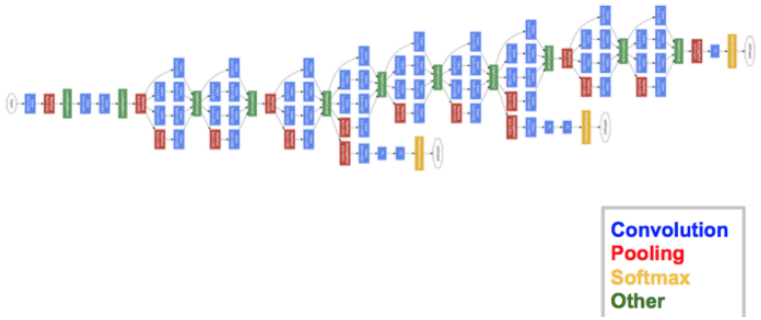
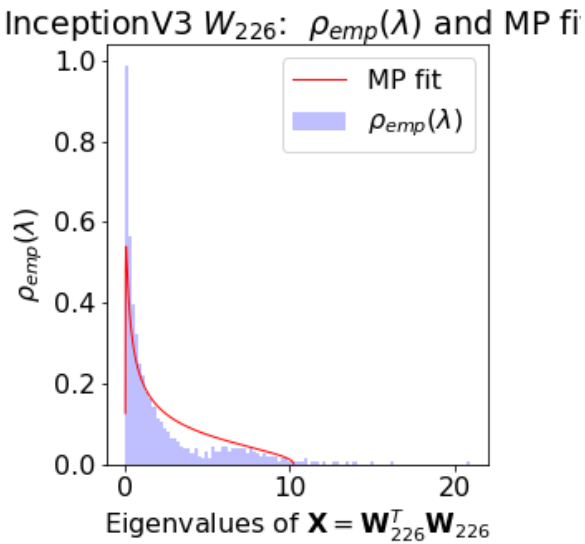




Figure: Zoomed-in ESD for Layer FC1 and FC3 of AlexNet.

Marchenko-Pastur Bulk-decay + Heavy-tailed

# RMT: InceptionV3 (a particularly unusual example)



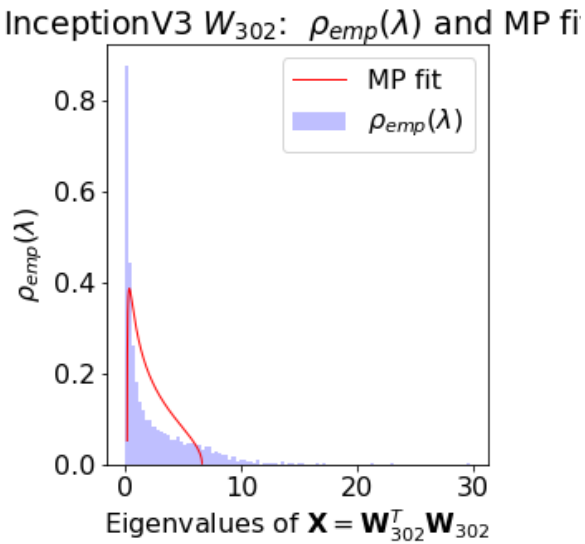
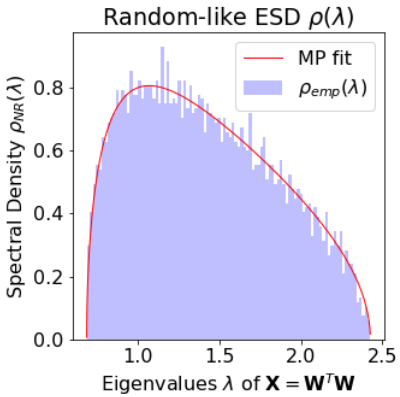



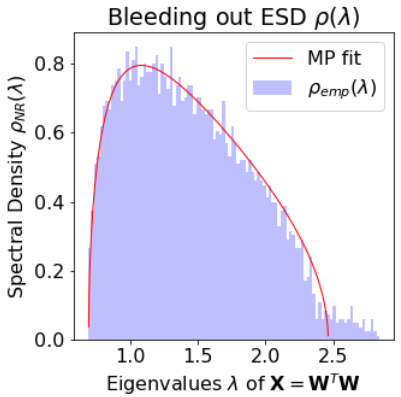

Figure: ESD for Layers L226 and L302 in InceptionV3, as distributed w/ pyTorch.

Marchenko-Pastur bulk decay, onset of Heavy Tails

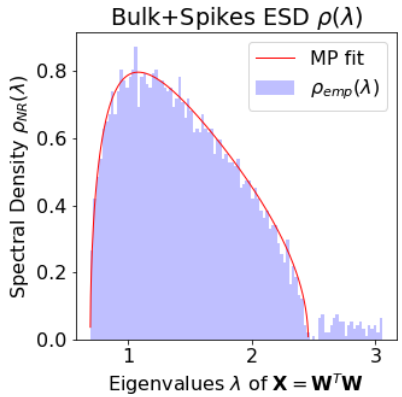
# RMT-based 5+1 Phases of Training



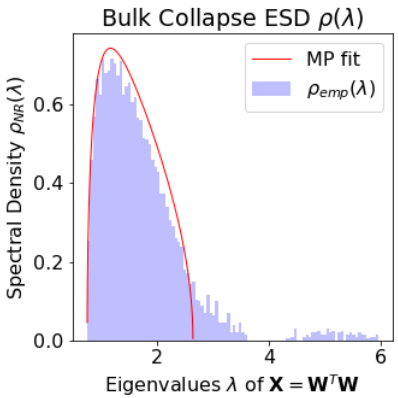

(a) RANDOM-LIKE.



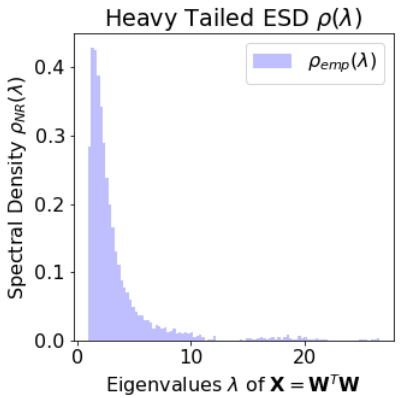

(b) BLEEDING-OUT.



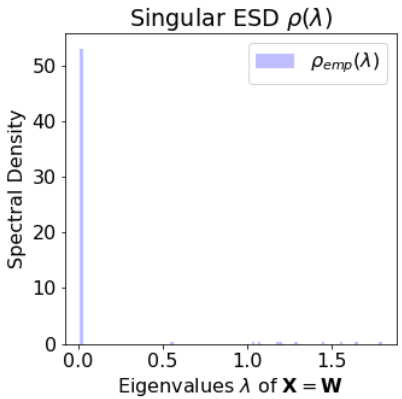

(c) BULK+SPIKES.




(d) BULK-DECAY.




(e) HEAVY-TAILED.




(f) RANK-COLLAPSE.

Figure: The 5+1 phases of learning we identified in DNN training.

# RMT-based 5+1 Phases of Training

We *model* "noise" and also "signal" with random matrices:

$$\mathbf{W} \simeq \mathbf{W}^{rand} + \Delta^{sig}. \tag{1}$$

| | Operational Definition | Informal Description via Eqn. (1) | Edge/tail Fluctuation Comments | Illustration and Description |
|---|---|---|---|---|
| RANDOM-LIKE | ESD well-fit by MP with appropriate $\lambda^+$ | $\mathbf{W}^{rand}$ random; $\|\Delta^{sig}\|$ zero or small | $\lambda_{max} \approx \lambda^+$ is sharp, with TW statistics | Fig. 10(a) |
| BLEEDING-OUT | ESD RANDOM-LIKE, excluding eigenmass just above $\lambda^+$ | $\mathbf{W}$ has eigenmass at bulk edge as spikes "pull out"; $\|\Delta^{sig}\|$ medium | BPP transition, $\lambda_{max}$ and $\lambda^+$ separate | Fig. 10(b) |
| BULK+SPIKES | ESD RANDOM-LIKE plus $\geq 1$ spikes well above $\lambda^+$ | $\mathbf{W}^{rand}$ well-separated from low-rank $\Delta^{sig}$; $\|\Delta^{sig}\|$ larger | $\lambda^+$ is TW, $\lambda_{max}$ is Gaussian | Fig. 10(c) |
| BULK-DECAY | ESD less RANDOM-LIKE; Heavy-Tailed eigenmass above $\lambda^+$; some spikes | Complex $\Delta^{sig}$ with correlations that don't fully enter spike | Edge above $\lambda^+$ is not concave | Fig. 10(d) |
| HEAVY-TAILED | ESD better-described by Heavy-Tailed RMT than Gaussian RMT | $\mathbf{W}^{rand}$ is small; $\Delta^{sig}$ is large and strongly-correlated | No good $\lambda^+$; $\lambda_{max} \gg \lambda^+$ | Fig. 10(e) |
| RANK-COLLAPSE | ESD has large-mass spike at $\lambda = 0$ | $\mathbf{W}$ very rank-deficient; over-regularization | — | Fig. 10(f) |

The 5+1 phases of learning we identified in DNN training.

# Bulk+Spikes: Small Models

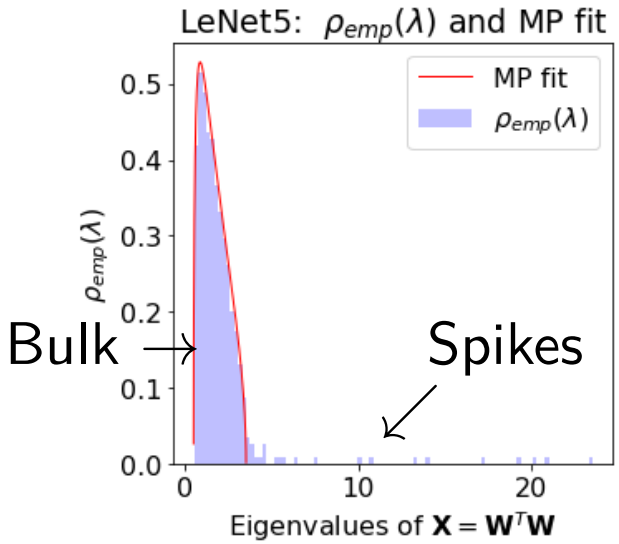
Low-rank perturbation

$$\mathbf{W}_l \simeq \mathbf{W}_l^{rand} + \Delta^{large}$$

Perturbative correction

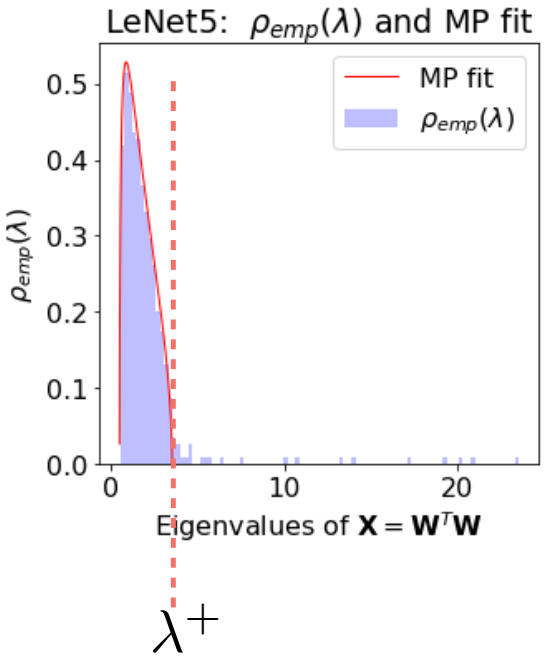
$$\lambda_{max} = \sigma^2 \left( \frac{1}{Q} + \frac{|\Delta|^2}{N} \right) \left( 1 + \frac{N}{|\Delta|^2} \right)$$

$$|\Delta| > (Q)^{-\frac{1}{4}}$$




Smaller, older models can be described perturbatively with Gaussian RMT

# Bulk+Spikes: Small Models $\sim$ Tikhonov regularization




simple scale threshold

$$\mathbf{x} = \left(\hat{\mathbf{X}} + \alpha\mathbf{I}\right)^{-1} \hat{\mathbf{W}}^T\mathbf{y}$$

eigenvalues $> \alpha$ (Spikes) carry most of the signal/information

Smaller, older models like LeNet5 exhibit traditional regularization

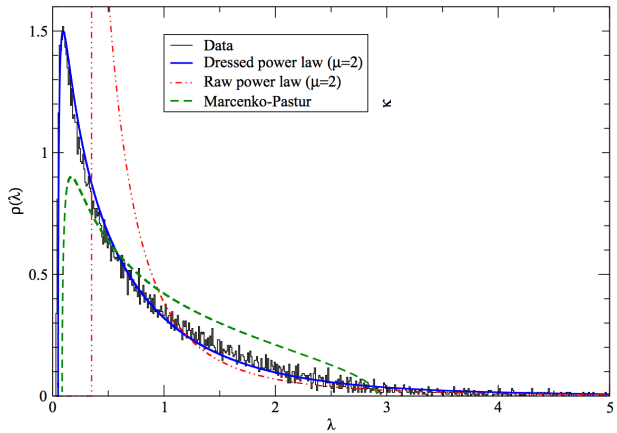
# Heavy-tailed Self-regularization

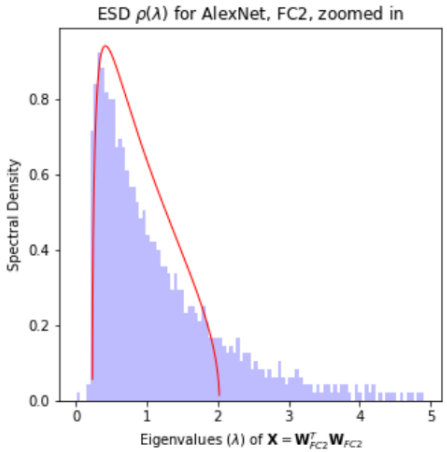
**W** is *strongly-correlated* and highly non-random

- Can *model* strongly-correlated systems by heavy-tailed random matrices

Then RMT/MP ESD will also have heavy tails

Known results from RMT / polymer theory (Bouchaud, Potters, etc)




AlexNet
ReseNet50
Inception V3
DenseNet201
...

Larger, modern DNNs exhibit novel Heavy-tailed self-regularization

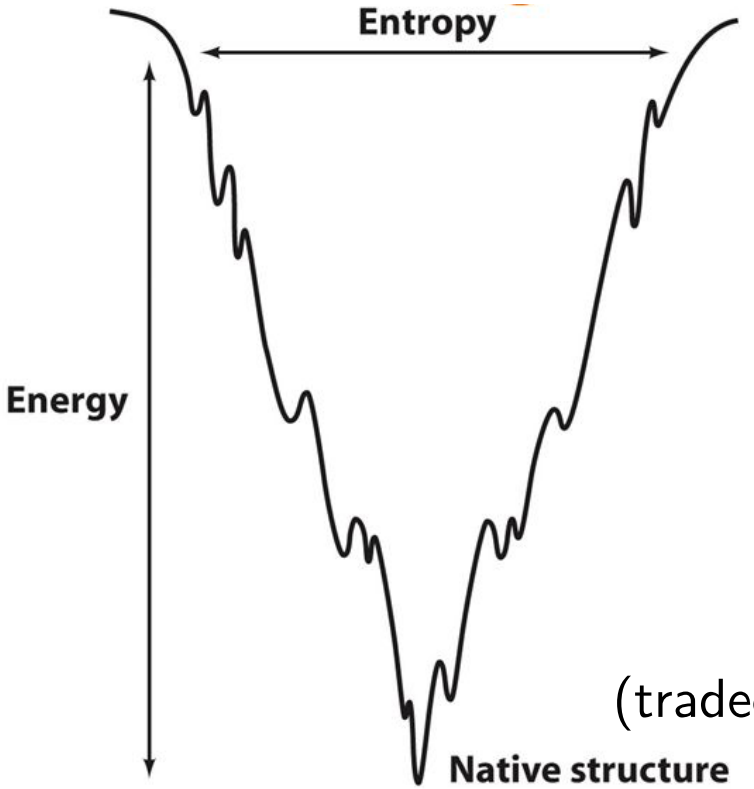
# Heavy-tailed Self-regularization

Summary of what we "suspect" today

- No single scale threshold.
- No simple low rank approximation for $\mathbf{W}_L$.
- Contributions from correlations at all scales.
- Can *not* be treated perturbatively.

Larger, modern DNNs exhibit novel Heavy-tailed self-regularization

# Implications: RMT and Deep Learning



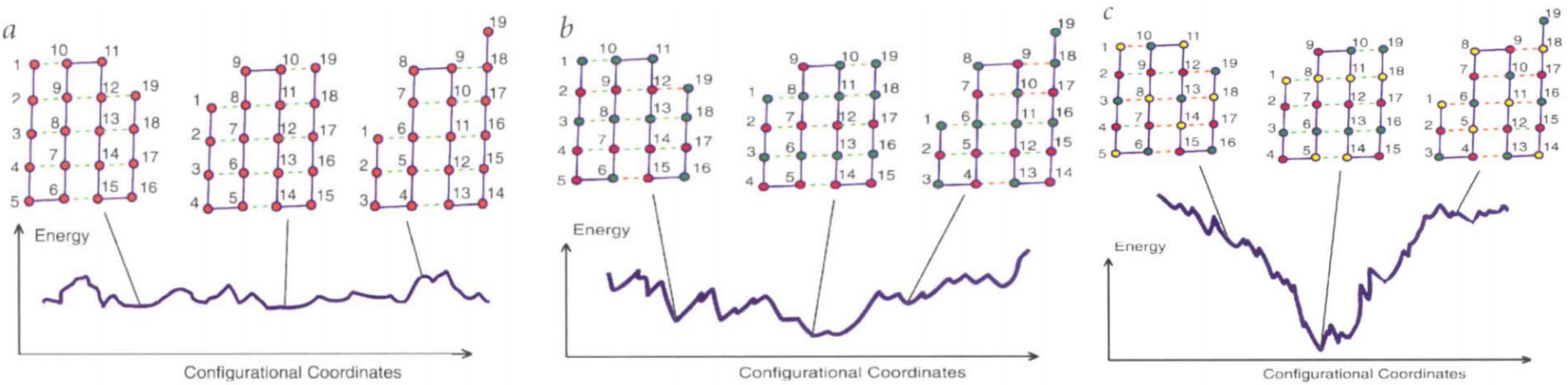

- Where are the local minima?
- How is the Hessian behaved?
- Are simpler models misleading?
- Can we design better learning strategies?

(tradeoff between Energy and Entropy minimization)

How can RMT be used to understand the Energy Landscape?

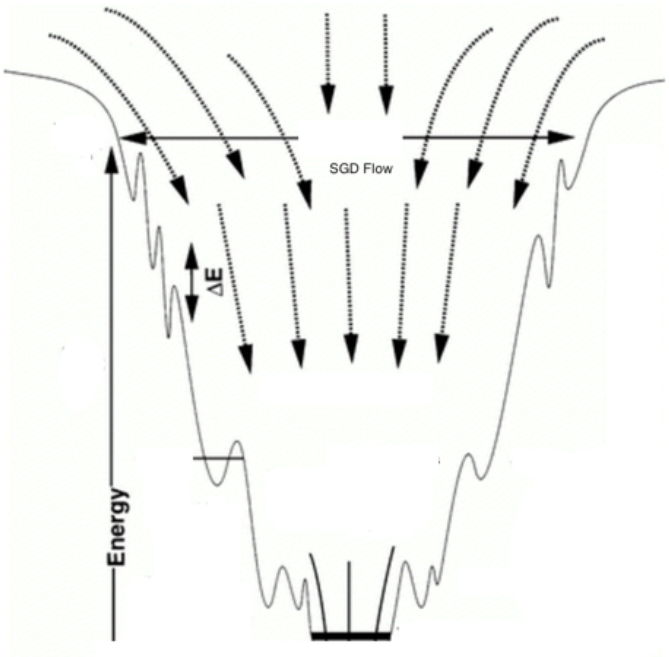
# Implications: Minimizing Frustration and Energy Funnels

As simple as can be?, Wolynes, 1997




Energy Landscape Theory: "random heteropolymer" versus "natural protein" folding

# Implications: Energy Landscapes of Heavy-tailed Models?




Compare with (Gaussian) Spin Glass model of Choromanska et al. 2015

Spin Glasses with Heavy Tails?

- Local minima do not concentrate near the ground state (Cizeau and Bouchaud 1993)

*If Energy Landscape is more funneled, then no "problems" with local minima!*

# Conclusions

Practice:

- Some problems are "global":
  - total click-through rate; total number of stars; human population genetics
- Some problems are "local":
  - meaningful communities; quasars that identify new physics; personalized medicine

Theory:

- Most objectives are global:
  - global partition, global eigenvetor, global minimum
- Some objectives are local:
  - find good cluster near a seed set
- Most algorithms are local:
  - take one step of gradient descent
  - but theory then makes a statement about global objective
  - but practice then deviates from theory and applies it locally

*Question: How to go forward?*