Glass dynamics and Signal reconstruction in rough landscapes

Chiara Cammarota

Department of Physics, Sapienza, University of Rome

Baity-Jesi, Sagun, Geiger, Spiegler, Ben Arous, Cammarota, LeCun, Wyart, Biroli PMLR 2018
Ros, Ben Arous, Biroli, Cammarota PRX 2019
Sarao, Biroli, Cammarota, Krzakala, Urbani, Zdeborova PRX 2020
Sarao, Biroli, Cammarota, Krzakala, Zdeborova Spotlight at NeurIPS 2019
Biroli, Cammarota, Ricci-Tersenghi J. Phys. A: Math. and Theor. 2020
Sarao, Biroli, Cammarota, Krzakala, Urbani, Zdeborova NeurIPS 2020
Biroli, Cammarota, Ricci-Tersenghi in preparation

SAPIENZA UNIVERSITÀ DI ROMA

INFN

PROGETTI DI RICERCA DI RILEVANTE INTERESSE NAZIONALE

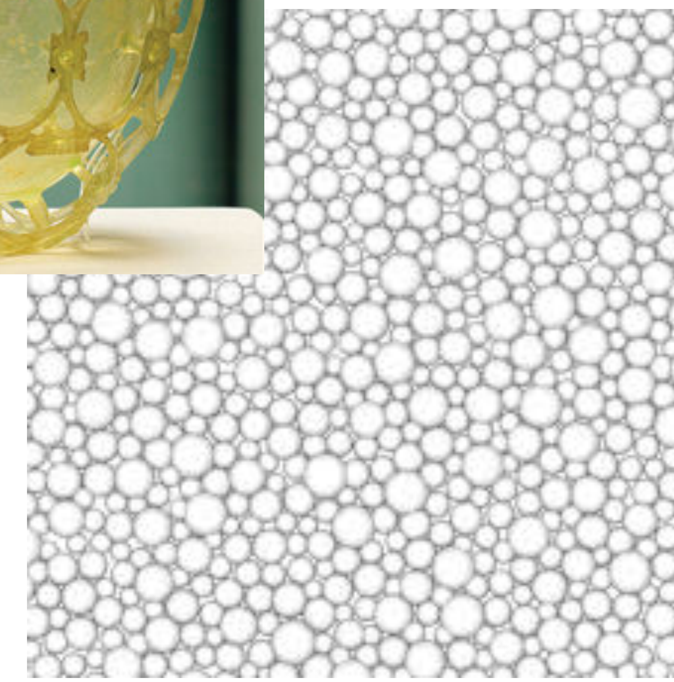
FUTURE AI RESEARCH

DISORDERED SYSTEMS DAYS AT KING'S COLLEGE LONDON

A workshop on disorder

To celebrate Reimer Kühn

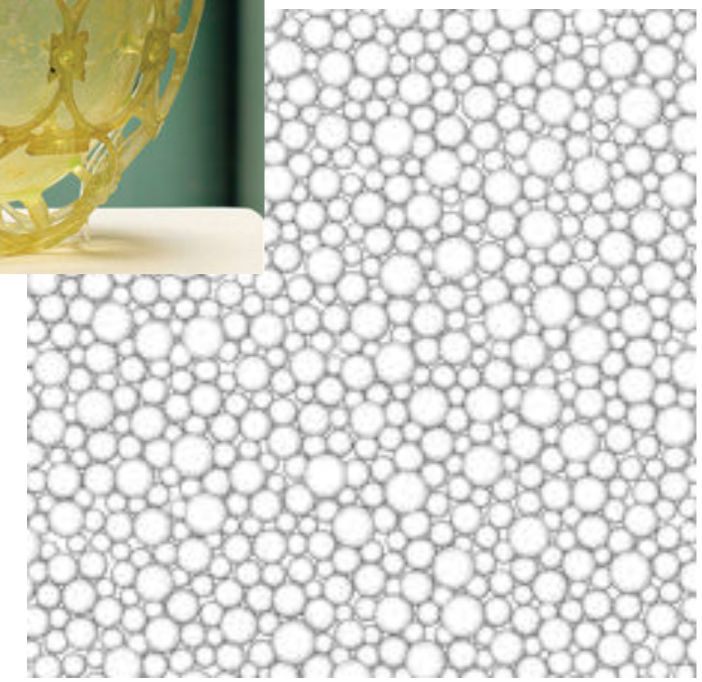
Glasses and aging dynamics



amorphous solids, or stuck liquids

$$H = \sum_{i<j} V(r_{ij}) \; ; \qquad\qquad r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$$

Glasses and aging dynamics

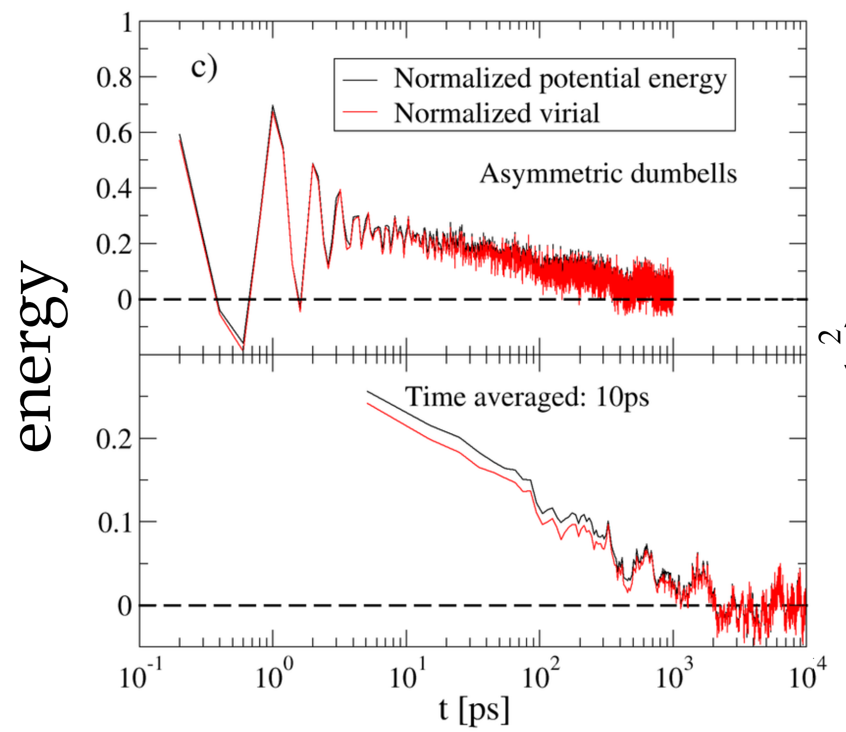


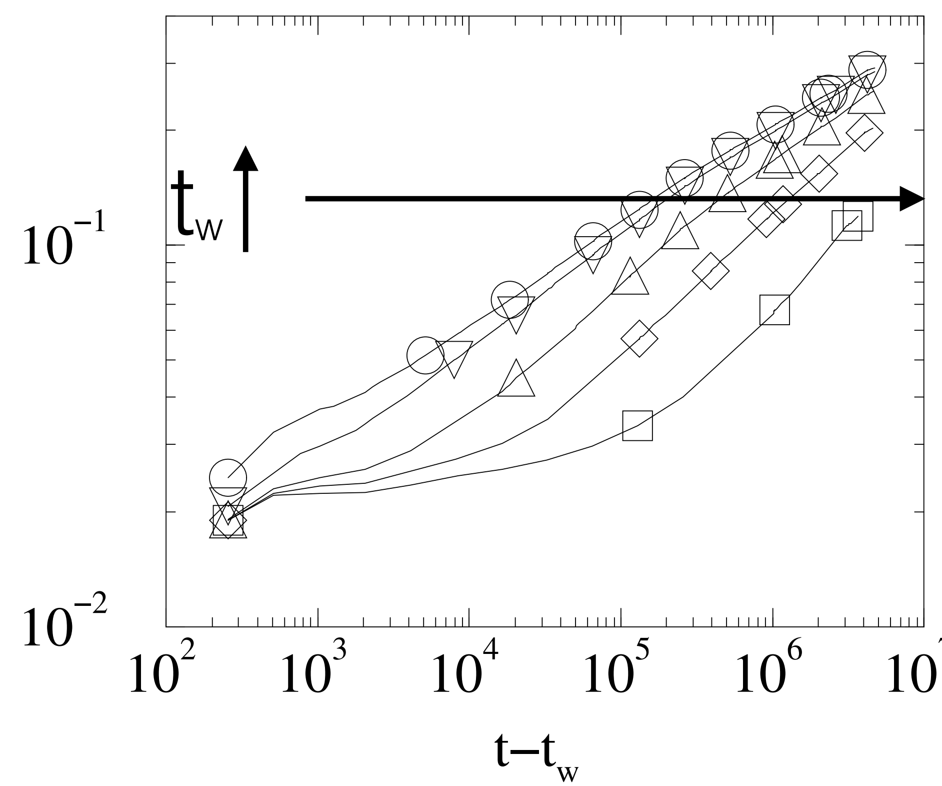
amorphous solids, or stuck liquids

$$H = \sum_{i<j} V(r_{ij}) \; ; \qquad r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$$

Relaxation dynamics $\dot{r}_{\alpha,i}(t) = -\nabla_{\alpha,i} H + \eta_{\alpha,i}(t)$

New dynamical properties, i.e. aging





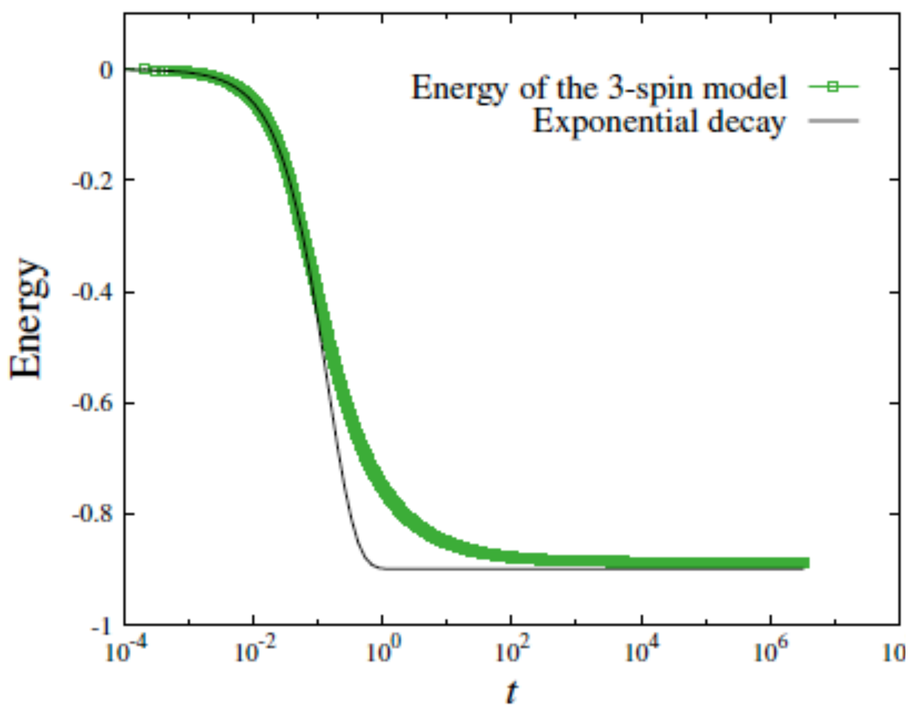
Sciortino 2005

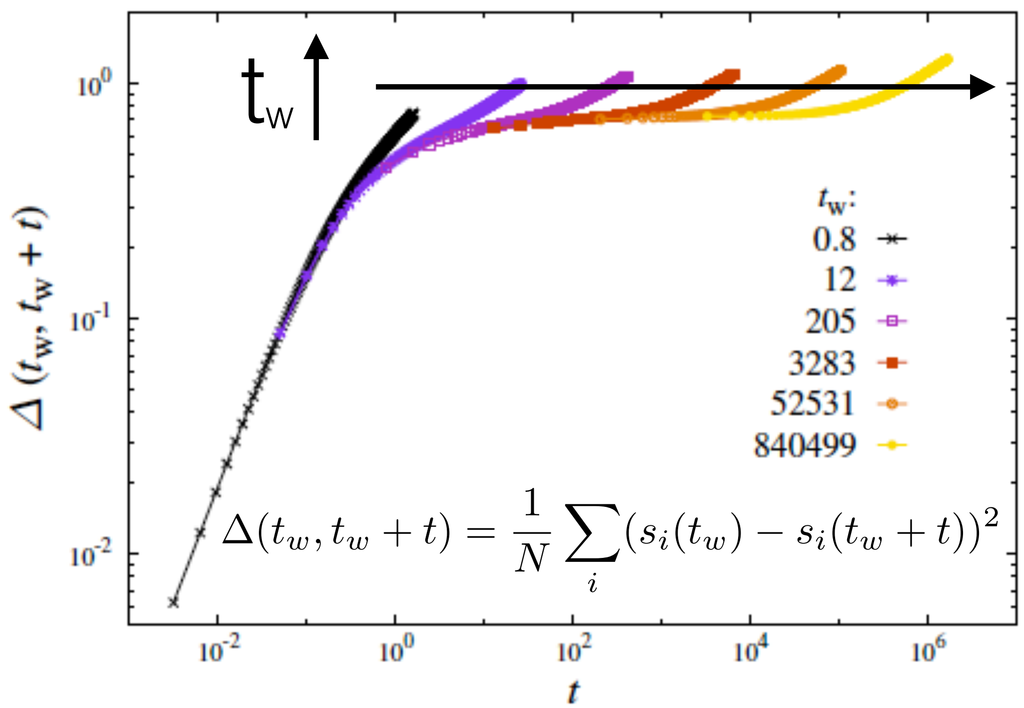
A mean field model of glass transition

p-spin model (p>2) $$H = -\sum_{(i_1,\ldots,i_p)} J_{i_1\ldots i_p} s_{i_1}\ldots s_{i_p}$$

Derrida 1980, Crisanti, Sommers 1992
Cugliandolo, Kurchan 1993

New dynamical properties, i.e. aging



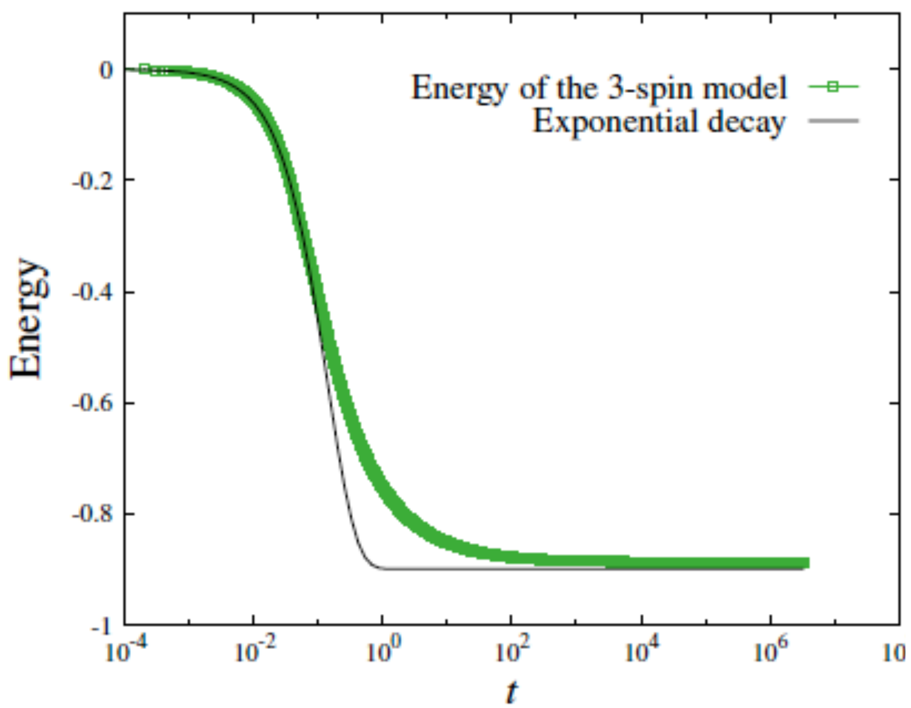


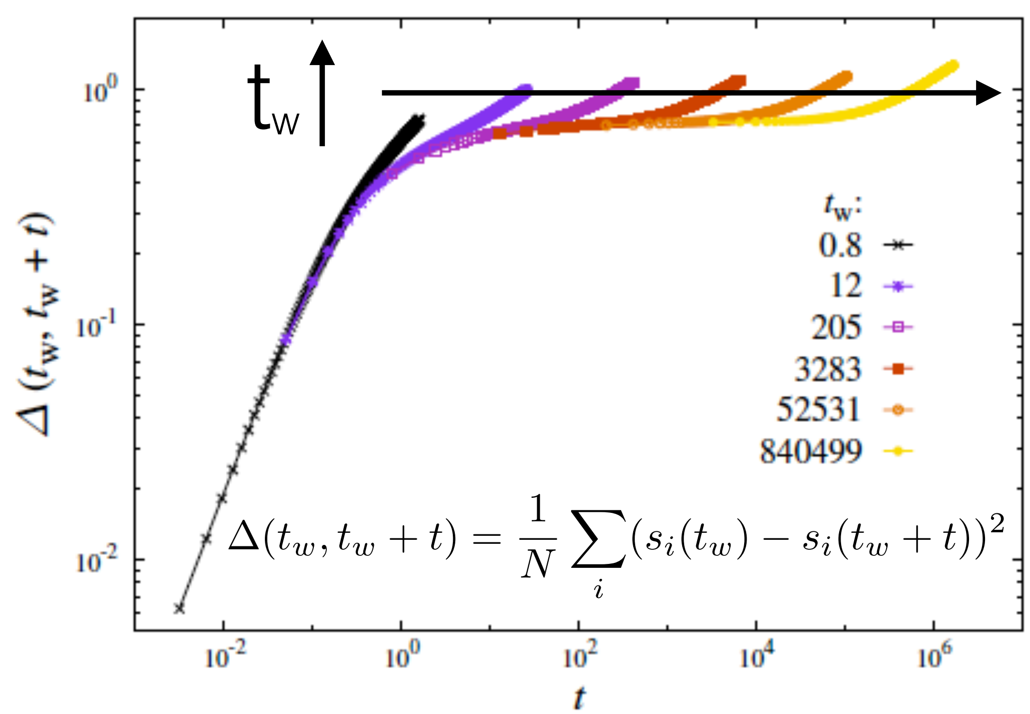
A mean field model of glass transition

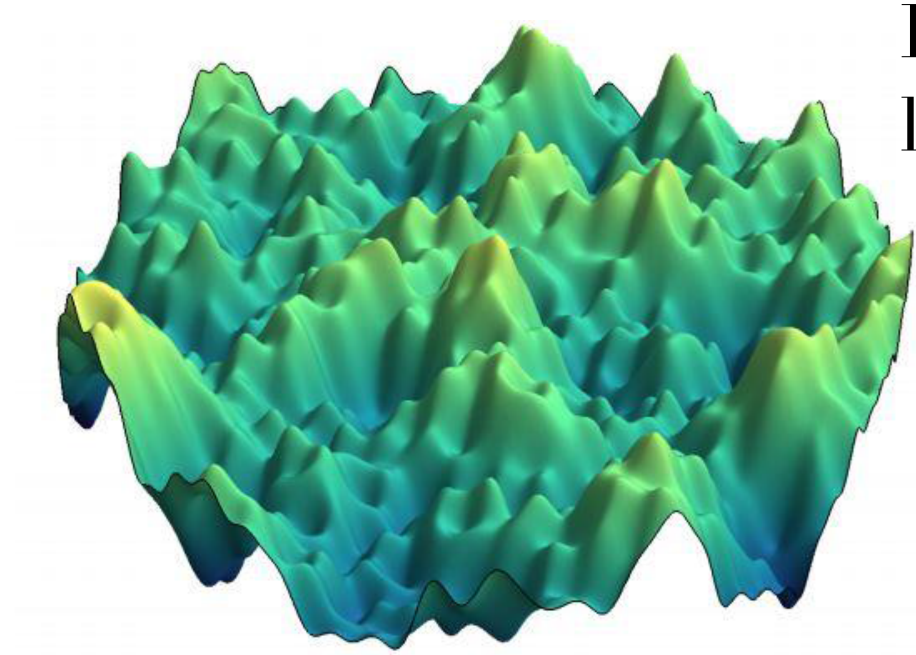
p-spin model (p>2) $$H = -\sum_{(i_1,\dots,i_p)} J_{i_1\dots i_p} s_{i_1}\dots s_{i_p}$$

Derrida 1980, Crisanti, Sommers 1992
Cugliandolo, Kurchan 1993

New dynamical properties, i.e. aging







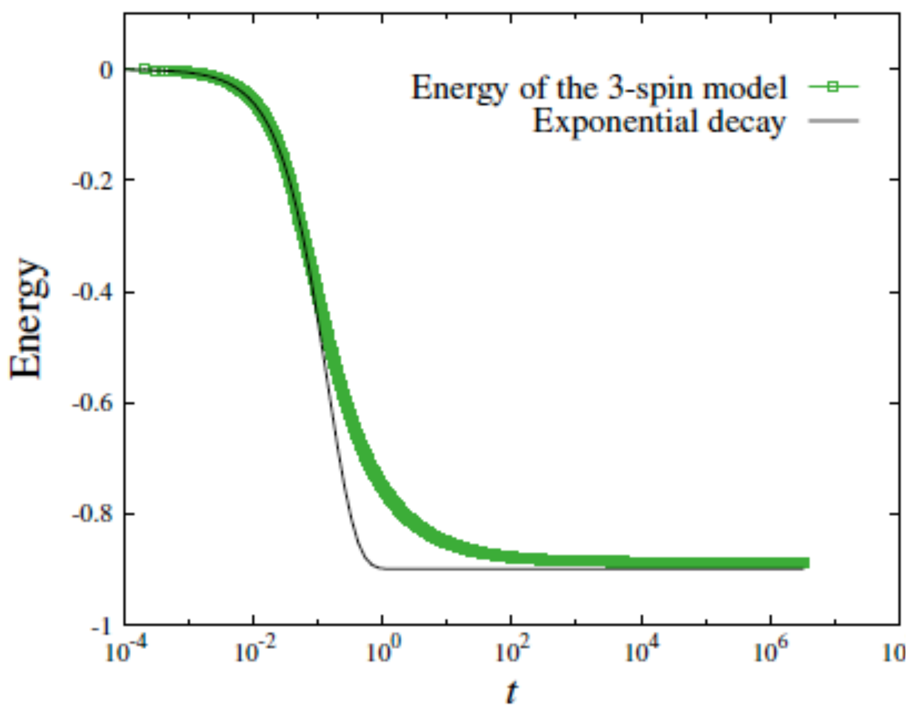
Energy landscape

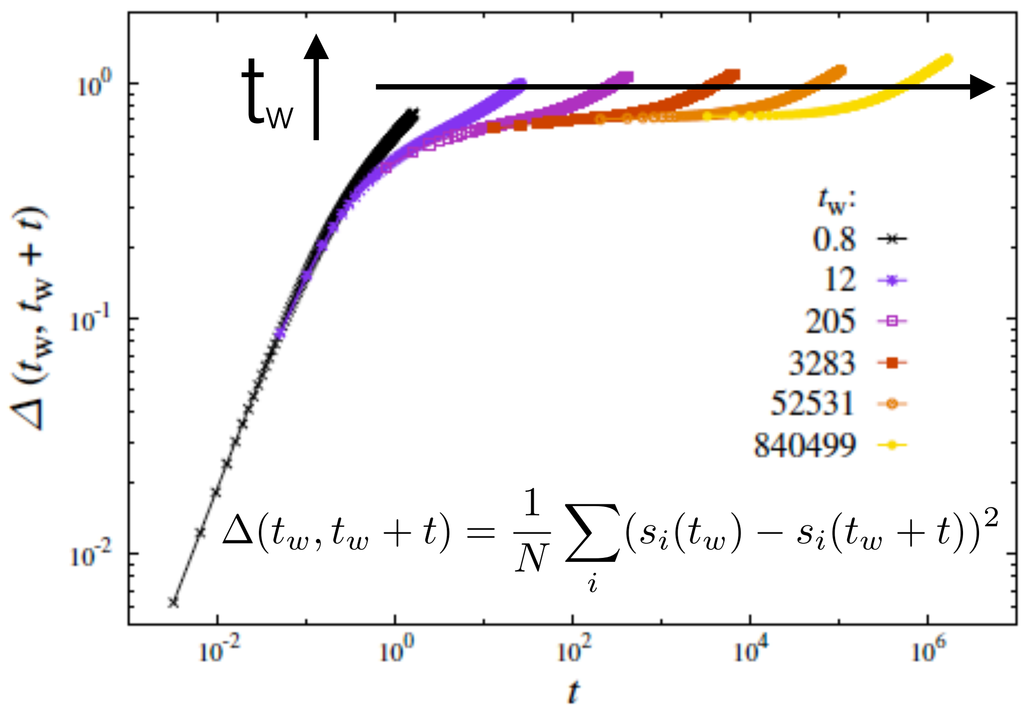
A mean field model of glass transition

p-spin model (p>2) $$H = -\sum_{(i_1,\ldots,i_p)} J_{i_1\ldots i_p} s_{i_1}\ldots s_{i_p}$$

Derrida 1980, Crisanti, Sommers 1992
Cugliandolo, Kurchan 1993

New dynamical properties, i.e. aging

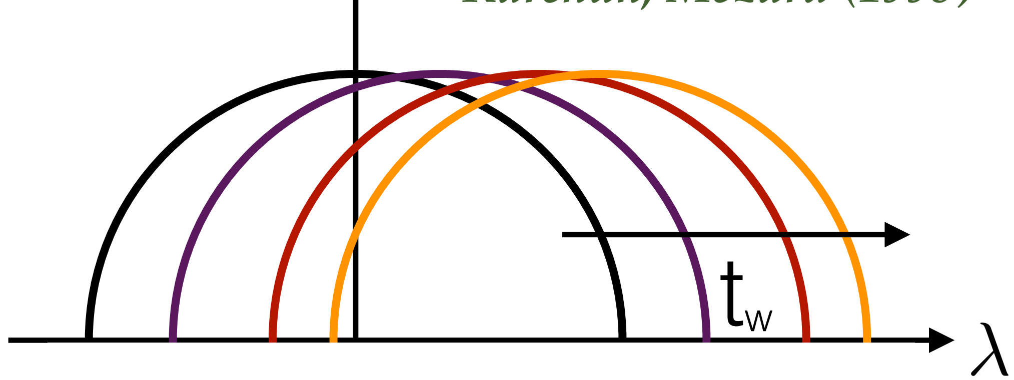


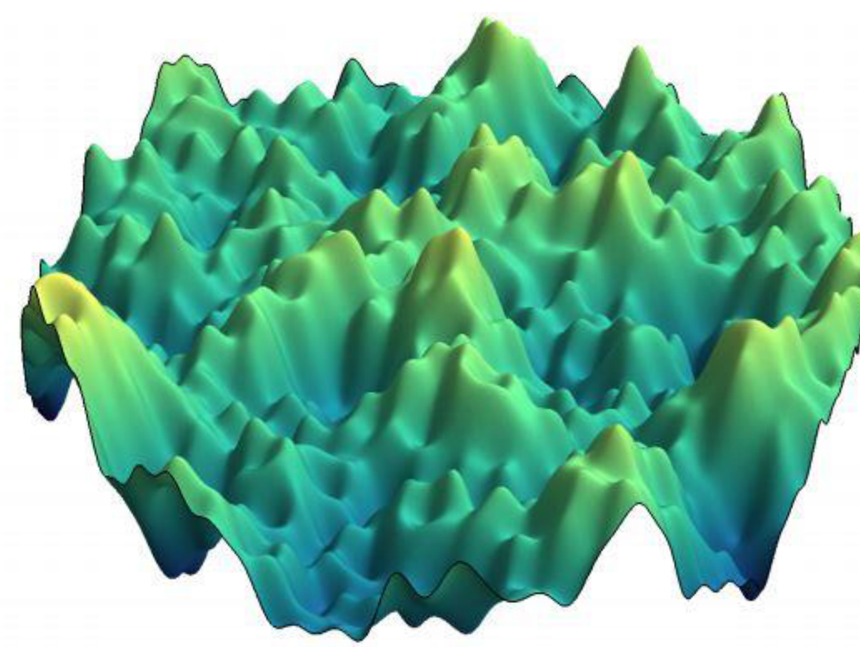


Spectrum of the Hessian

Bouchaud, Cugliandolo, Kurchan, Mezard (1998)

$\rho(\lambda)$





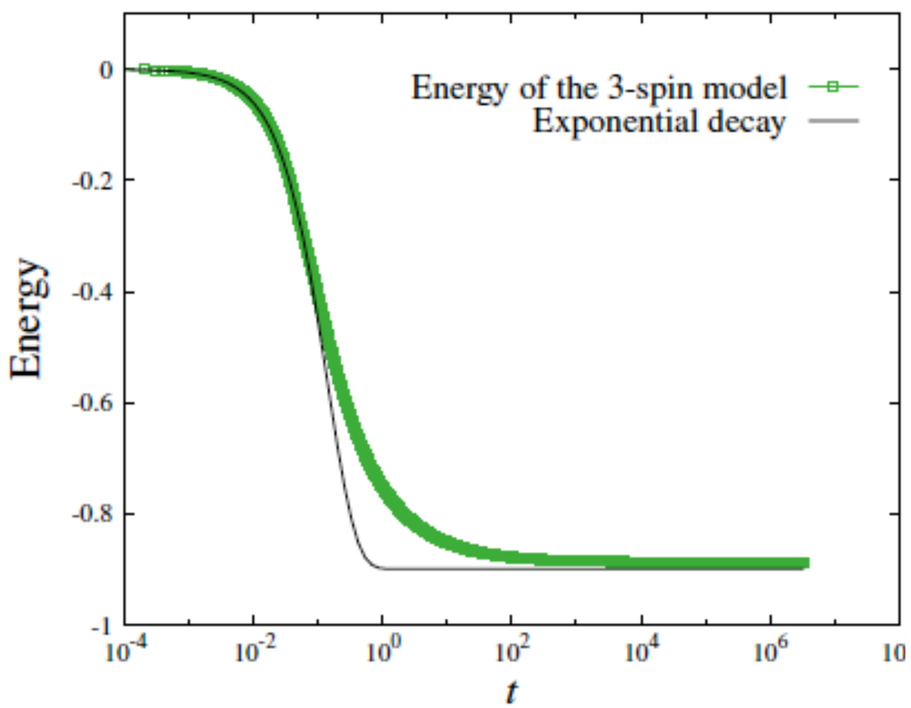
Energy landscape

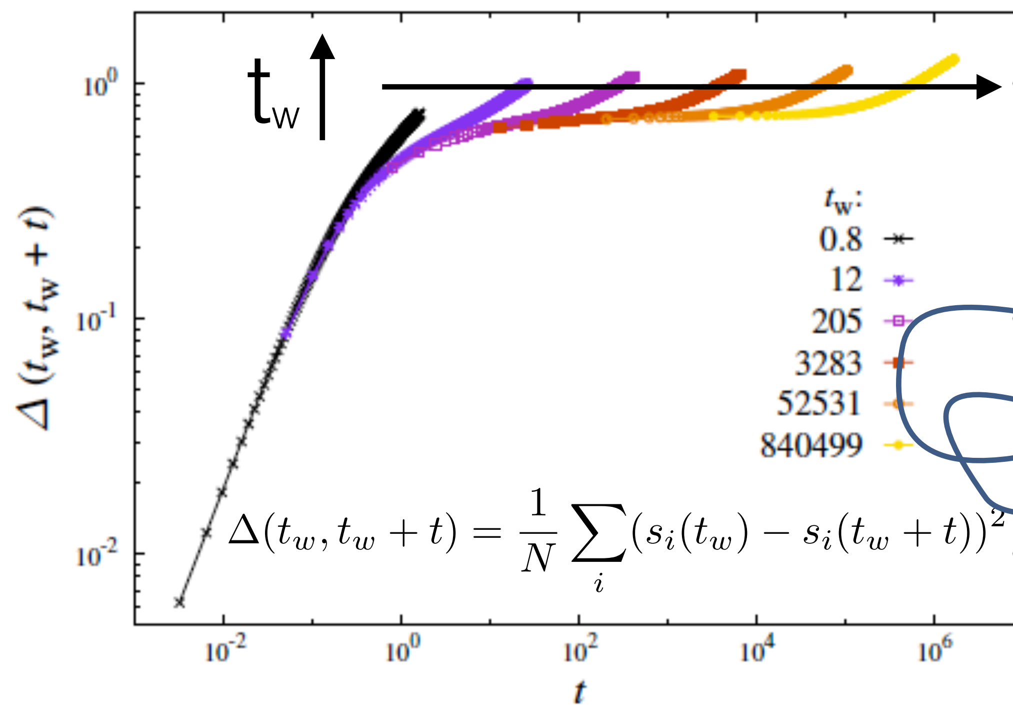
A mean field model of glass transition

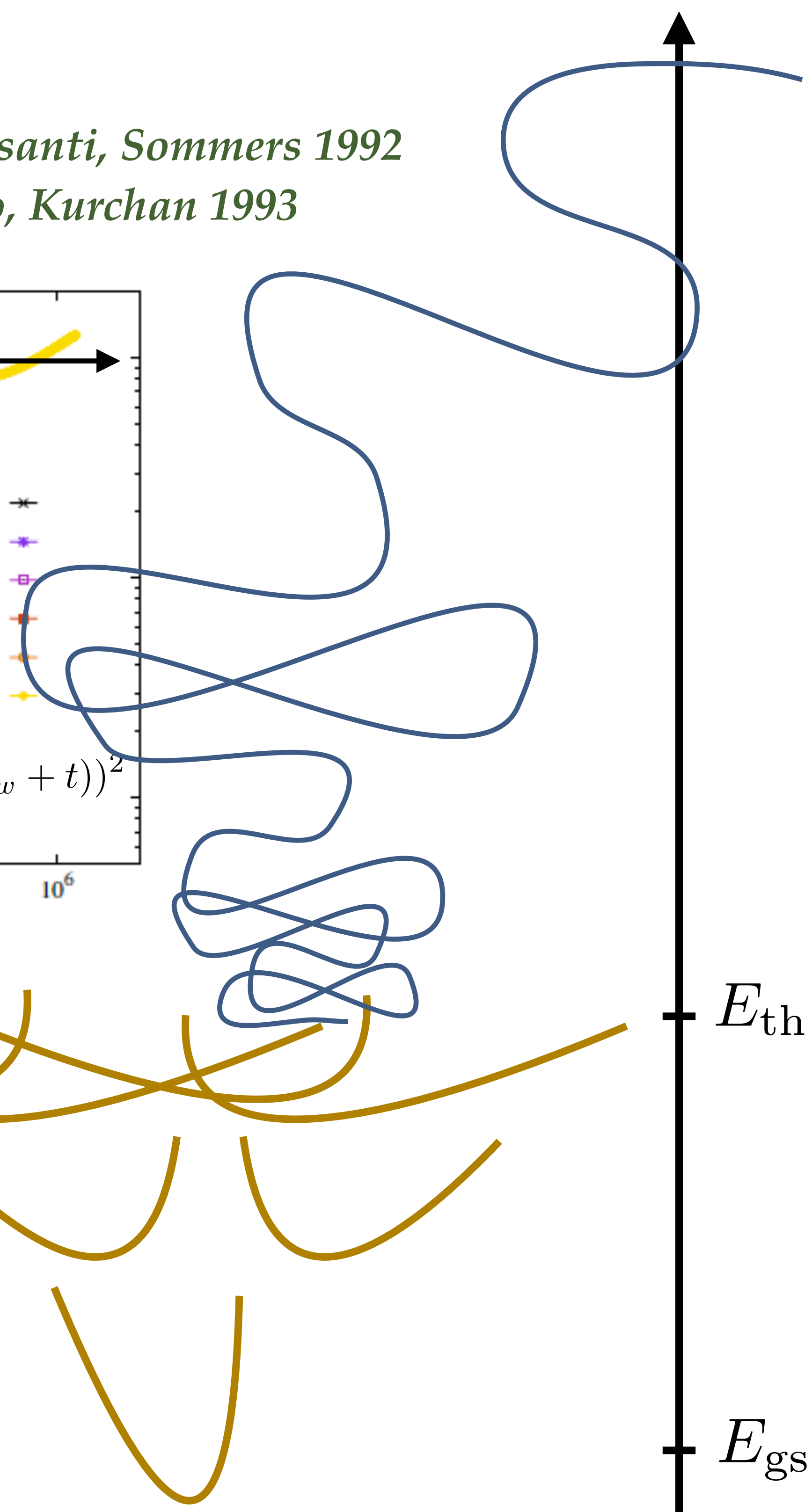
p-spin model (p>2) $$H = -\sum_{(i_1,\ldots,i_p)} J_{i_1\ldots i_p} s_{i_1}\ldots s_{i_p}$$

Derrida 1980, Crisanti, Sommers 1992
Cugliandolo, Kurchan 1993

New dynamical properties, i.e. aging

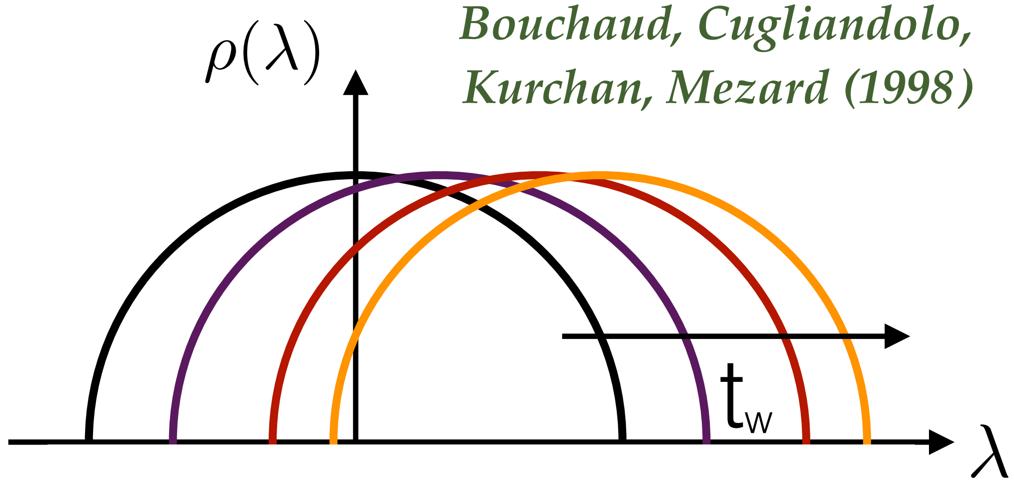






Spectrum of the Hessian

Bouchaud, Cugliandolo, Kurchan, Mezard (1998)

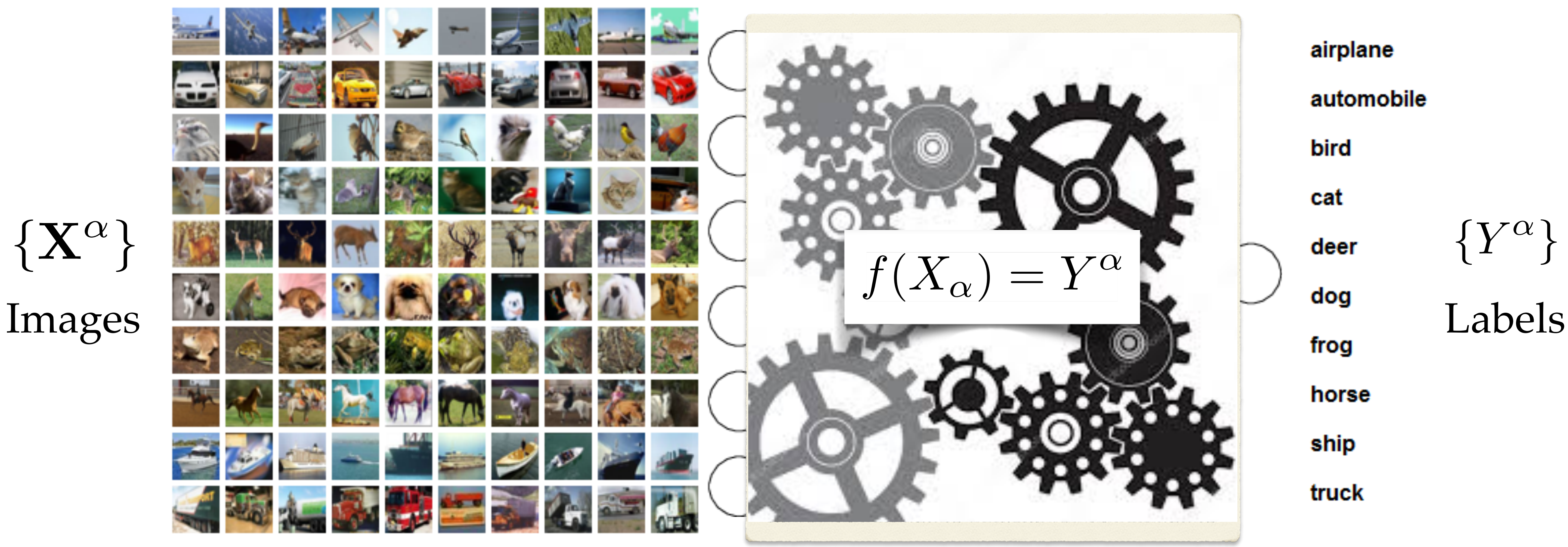


Machine Learning

Dynamical experiments to infer the landscape

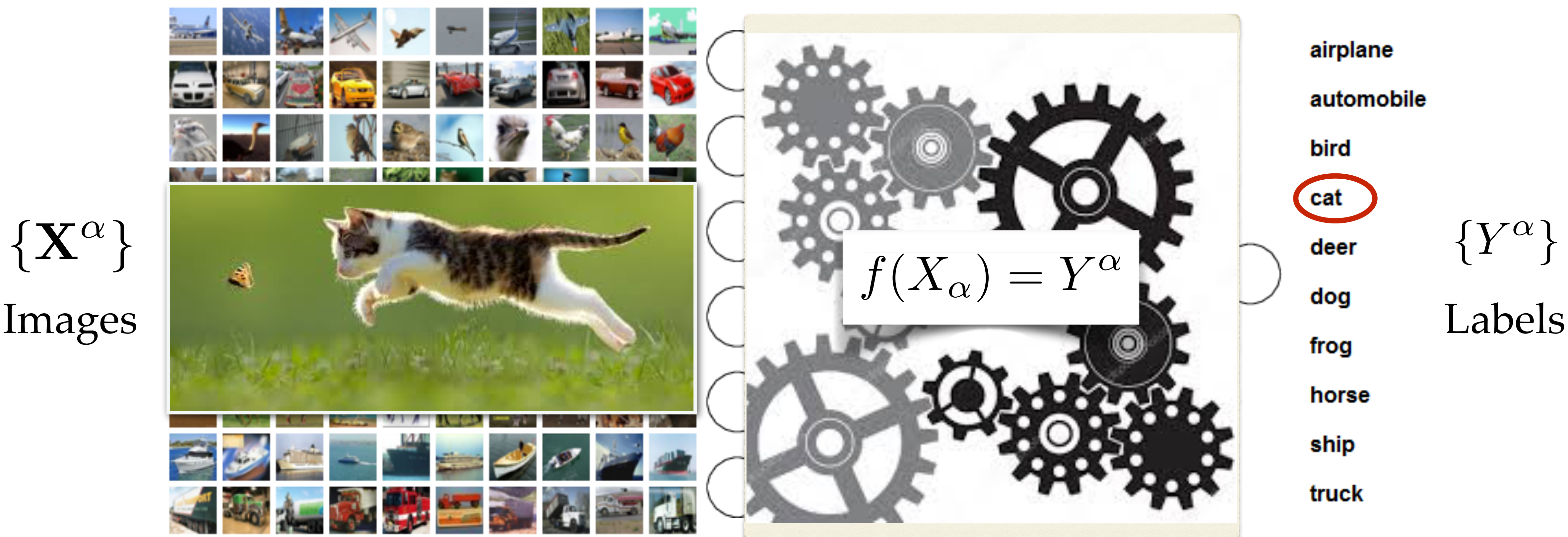
Machine Learning

Estimation of a function able to classify images



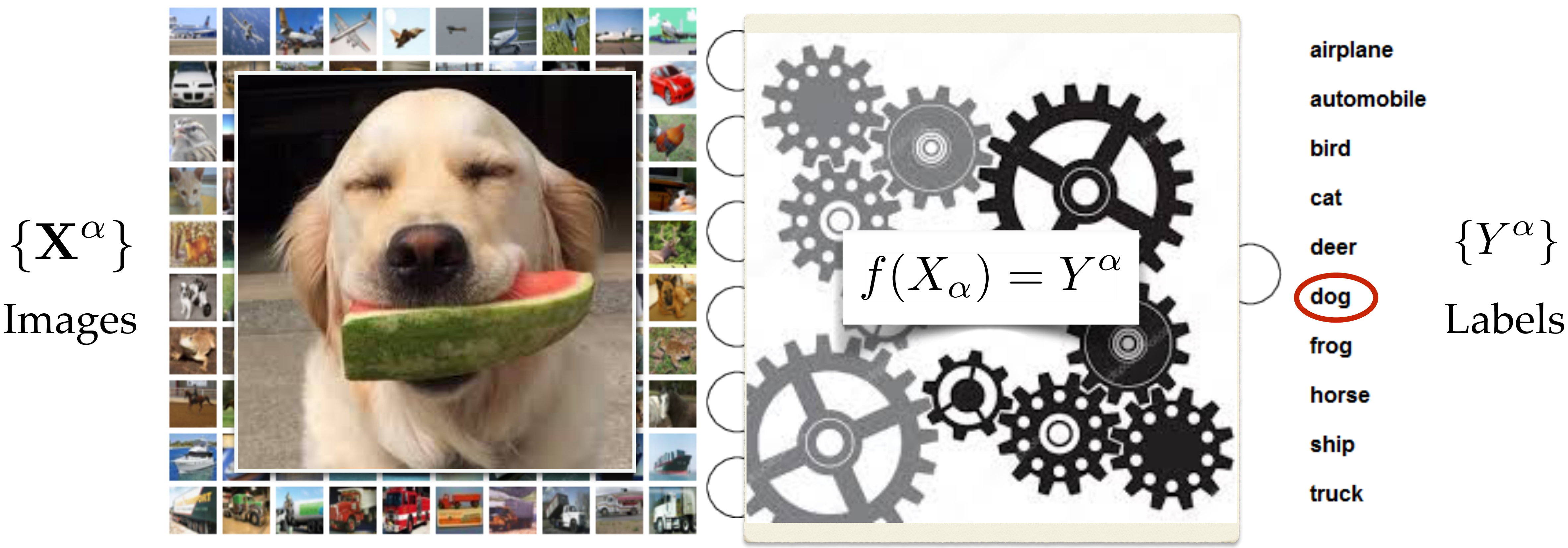
Machine Learning

Estimation of a function able to classify images



Machine Learning

Estimation of a function able to classify images

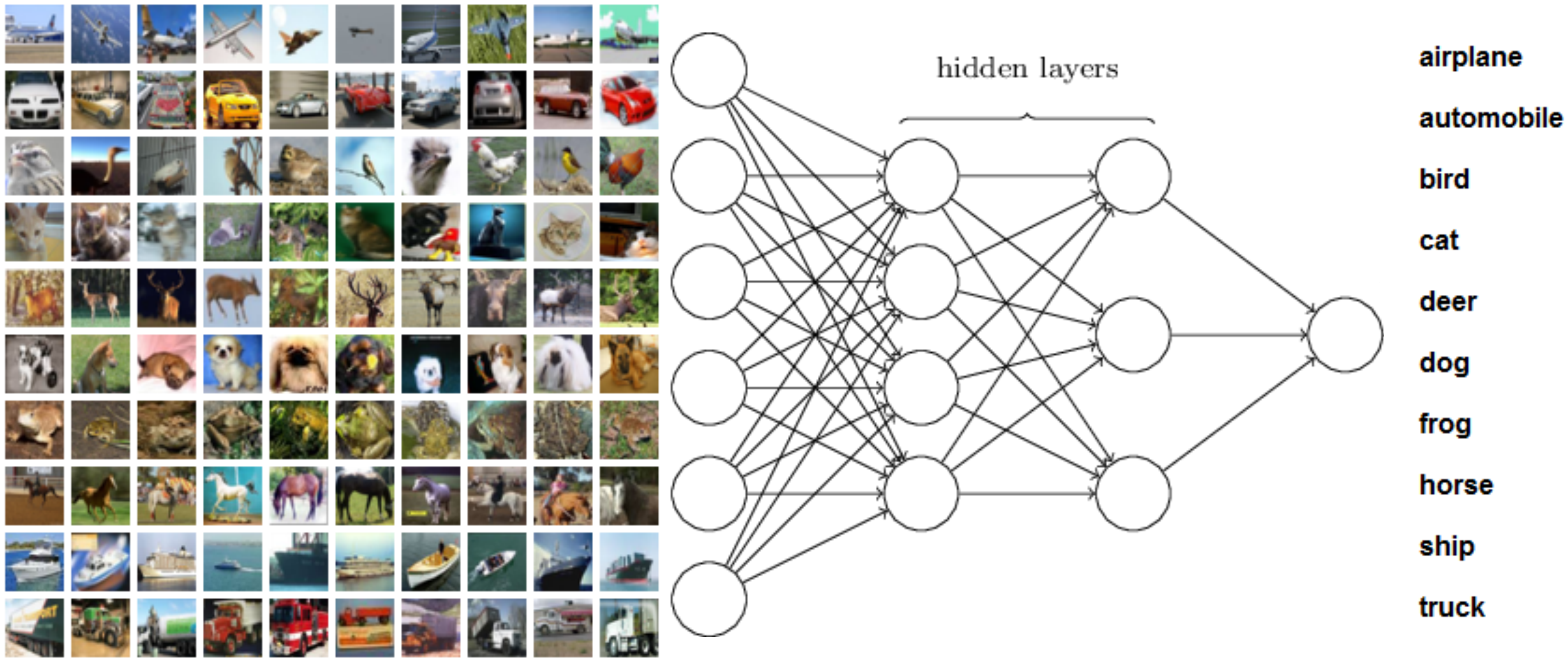


Machine Learning

Estimation of a function able to classify images

$\{\mathbf{X}^\alpha\}$

Images



$\{Y^\alpha\}$

Labels

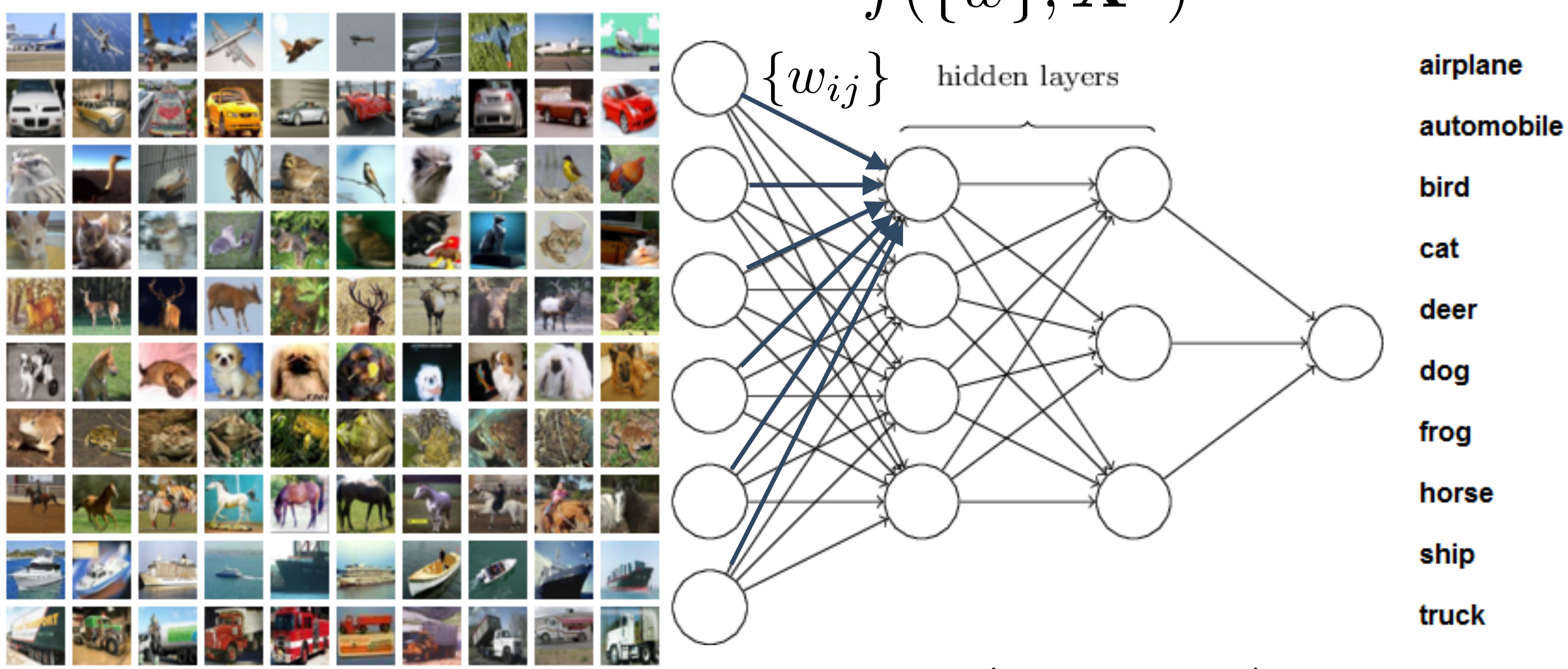
Machine Learning

Estimation of a function able to classify images

$\{\mathbf{X}^\alpha\}$

Images

$f(\{w\};\mathbf{X}^\alpha)$



$\{Y^\alpha\}$

Labels

$$x_i^{1,\alpha} = \sigma\left(\sum_j w_{ij} X_j^\alpha\right)$$

parameters $\{w_{ij}\}$

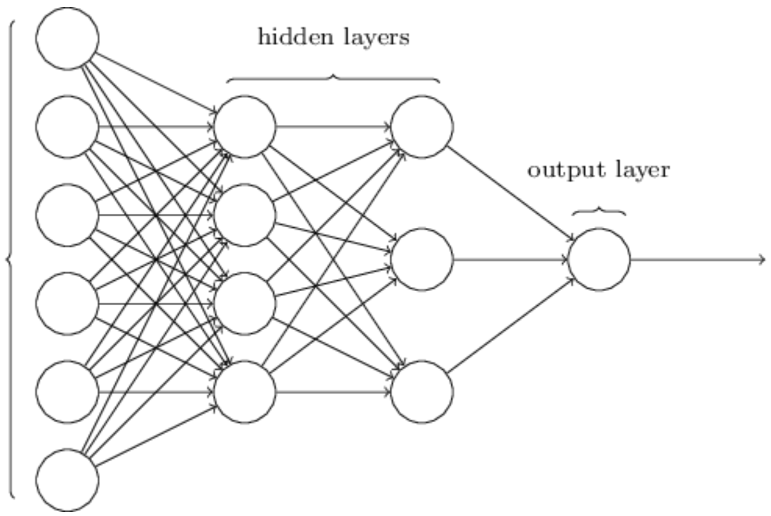
$\#\text{parameters} = 10^8$

Machine Learning *vs* glass quenches

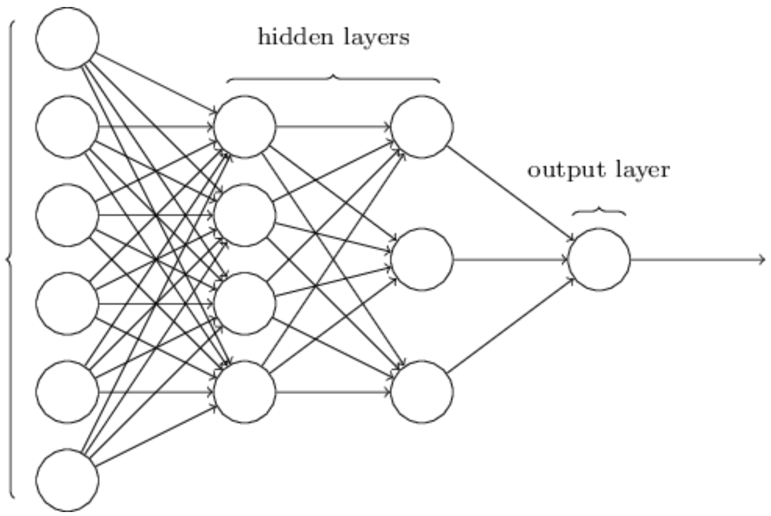
distance between output and correct answer, i.e.

$$\ell(\{w\};\mathbf{X}^\alpha,Y^\alpha)=(Y^\alpha-f(\{w\};\mathbf{X}^\alpha))^2$$

Loss function $$\mathcal{L}\{w\}=\frac{1}{M}\sum_\alpha^M \ell(\{w\};\mathbf{X}^\alpha,Y^\alpha)$$



Machine Learning *vs* glass quenches



distance between output and correct answer, i.e.

$$\ell(\{w\};\mathbf{X}^\alpha, Y^\alpha) = (Y^\alpha - f(\{w\};\mathbf{X}^\alpha))^2$$

Loss function $$\mathcal{L}\{w\} = \frac{1}{M}\sum_\alpha^M \ell(\{w\};\mathbf{X}^\alpha, Y^\alpha)$$

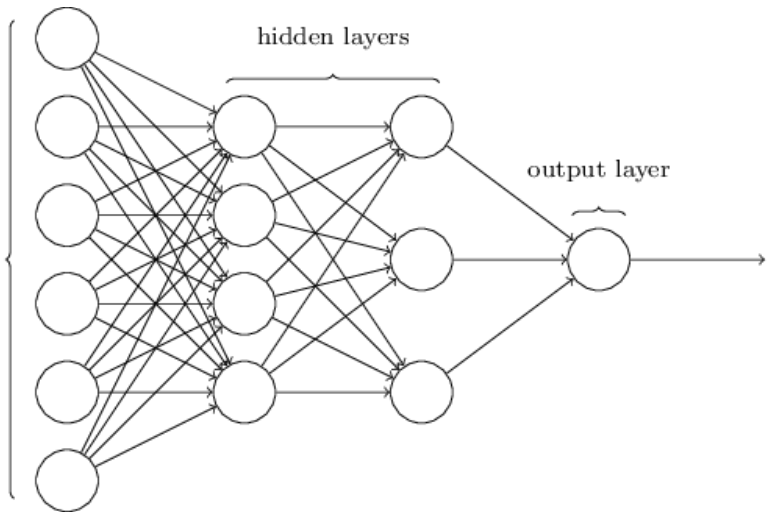
Learning (training): minimise the Loss function from random initial condition

Stochastic Gradient Descent $$\mathbf{w}(t+\Delta t) = \mathbf{w}(t) - \eta\nabla_w \sum_\alpha^B \ell(\{w\};\mathbf{X}^\alpha, Y^\alpha)$$

Machine Learning *vs* glass quenches

distance between output and correct answer, i.e.

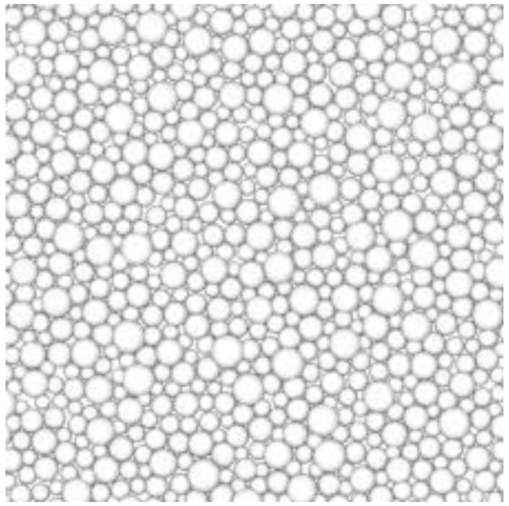
$$\ell(\{w\};\mathbf{X}^\alpha,Y^\alpha) = (Y^\alpha - f(\{w\};\mathbf{X}^\alpha))^2$$



Loss function $$\mathcal{L}\{w\} = \frac{1}{M}\sum_\alpha^M \ell(\{w\};\mathbf{X}^\alpha,Y^\alpha)$$

Learning (training): minimise the Loss function from random initial condition

Stochastic Gradient Descent $$\mathbf{w}(t+\Delta t) = \mathbf{w}(t) - \eta\nabla_w \sum_\alpha^B \ell(\{w\};\mathbf{X}^\alpha,Y^\alpha)$$



Quenches : rapid coolings from high temperature,
i.e. almost random initial configuration

Relaxation dynamics $$\dot{r}_{\alpha,i}(t) = -\nabla_{\alpha,i}H + \eta_{\alpha,i}(t)$$

How is learning dynamics? How the loss landscape?

Learning as interrupted Aging and Diffusion

Baity-Jesi, Sagun, Geiger, Spiegler, Ben Arous, Cammarota, LeCun, Wyart, Biroli PMLR 2018

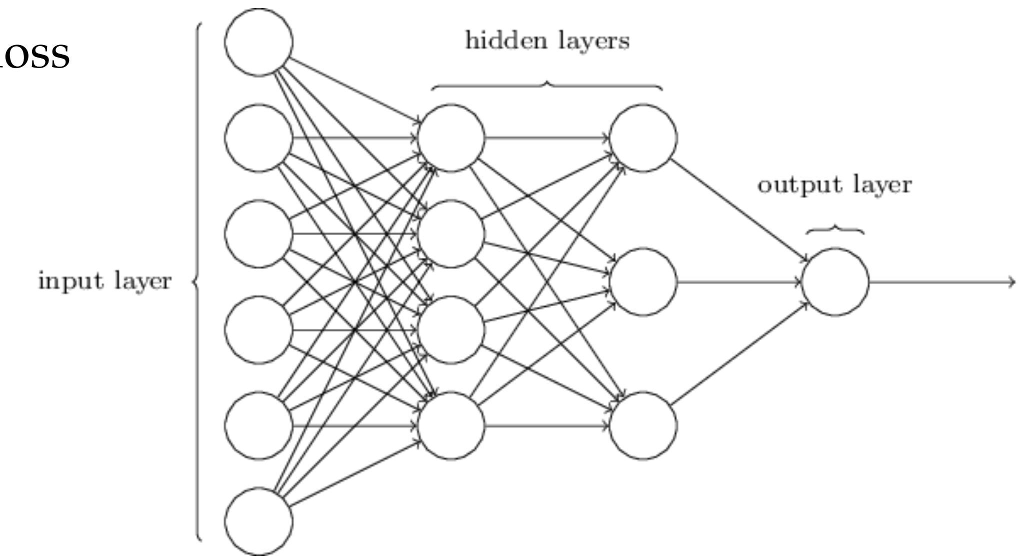
Toy model: 1 hidden layer, ReLU, sigmoid in output, MSE as a loss

Fully connected: 3 hidden layers, ReLU, log likelihood

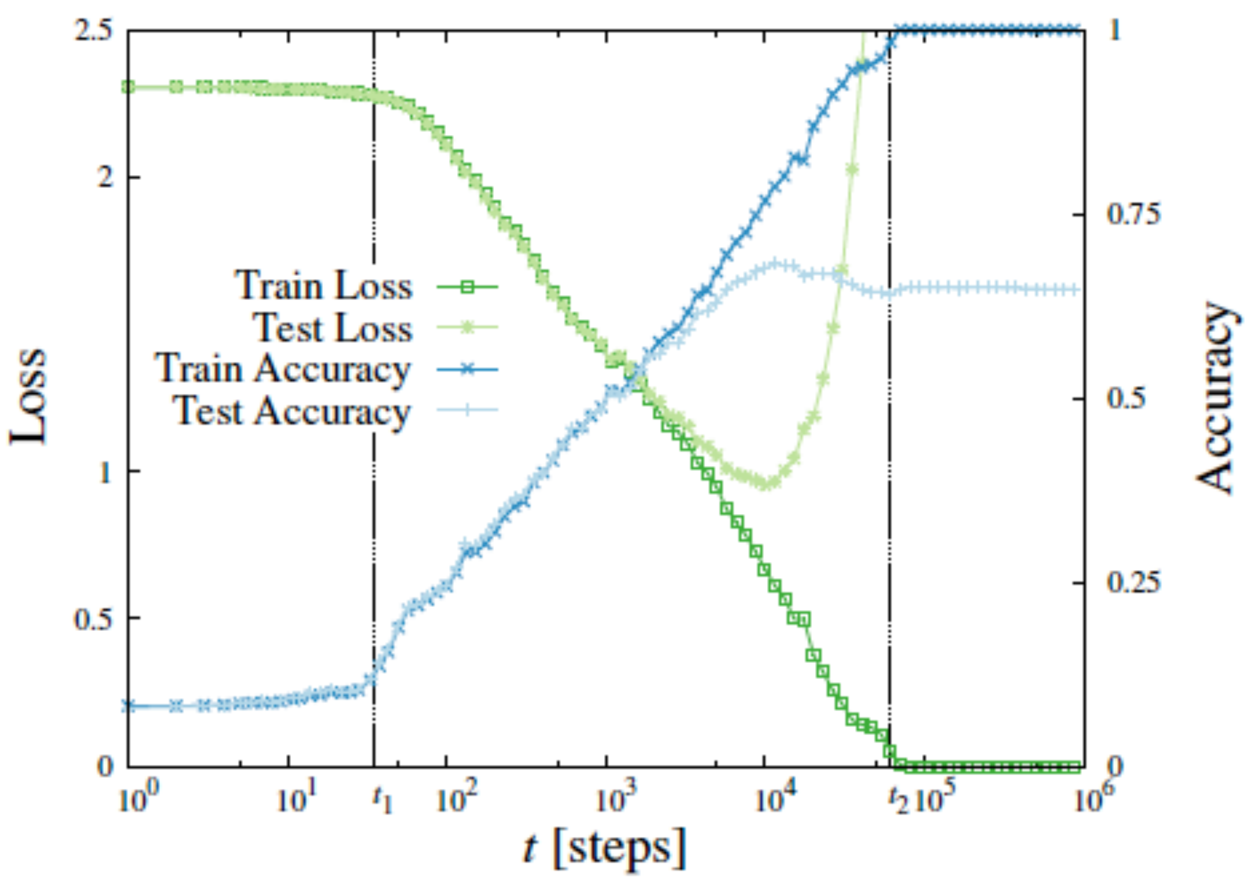
Small Net: 2 hidden convolutional layers,
2 fully connected ReLU, log likelihood

ResNet18: 18 hidden convolutional layers

MNIST, CFAR-10, CFAR-100

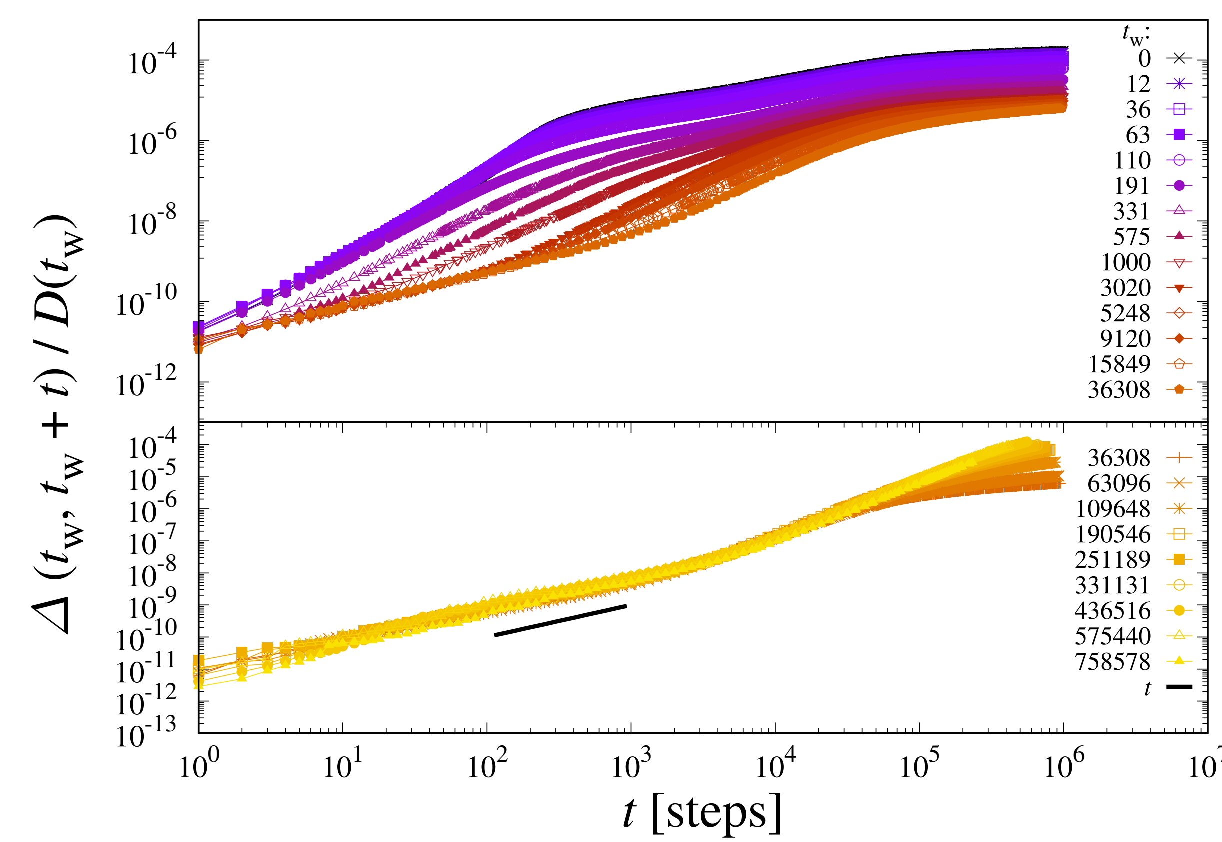


Slow decay of Loss function



(c) Small Net on CIFAR-10, $B = 100$, $\alpha = 0.01$.

Mean Square displacement



Dr. Chiara Cammarota

Learning as interrupted Aging and Diffusion

Baity-Jesi, Sagun, Geiger, Spiegler, Ben Arous, Cammarota, LeCun, Wyart, Biroli PMLR 2018

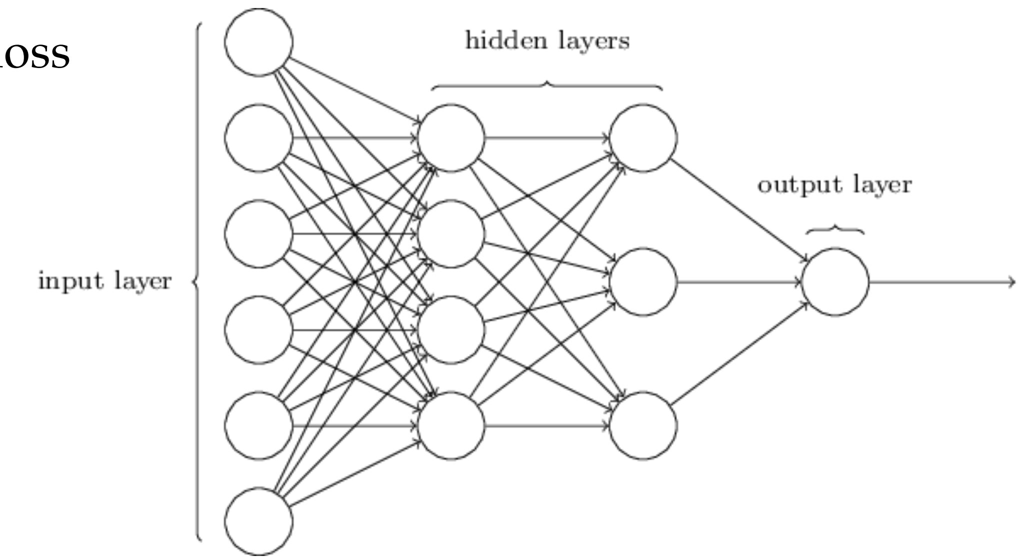
Toy model: 1 hidden layer, ReLU, sigmoid in output, MSE as a loss

Fully connected: 3 hidden layers, ReLU, log likelihood

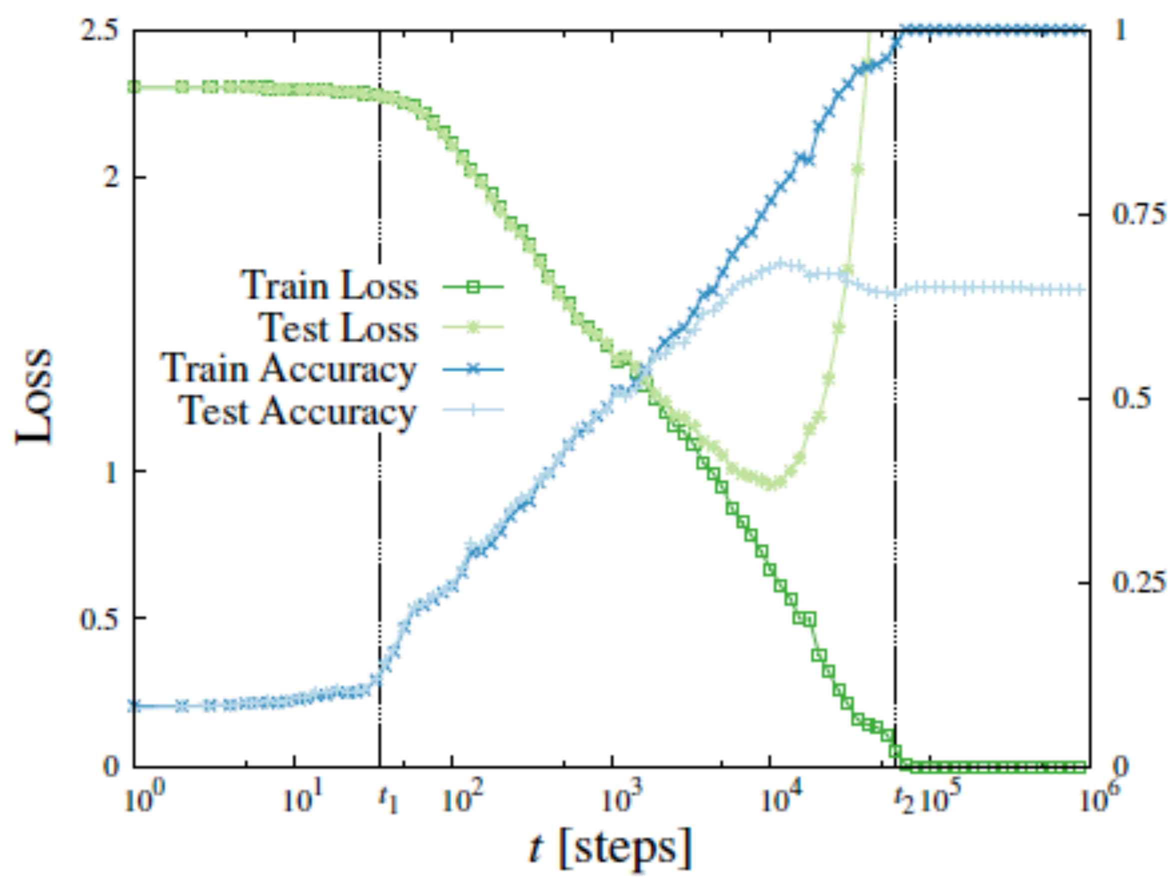
Small Net: 2 hidden convolutional layers,
2 fully connected ReLU, log likelihood

ResNet18: 18 hidden convolutional layers

MNIST, CFAR-10, CFAR-100

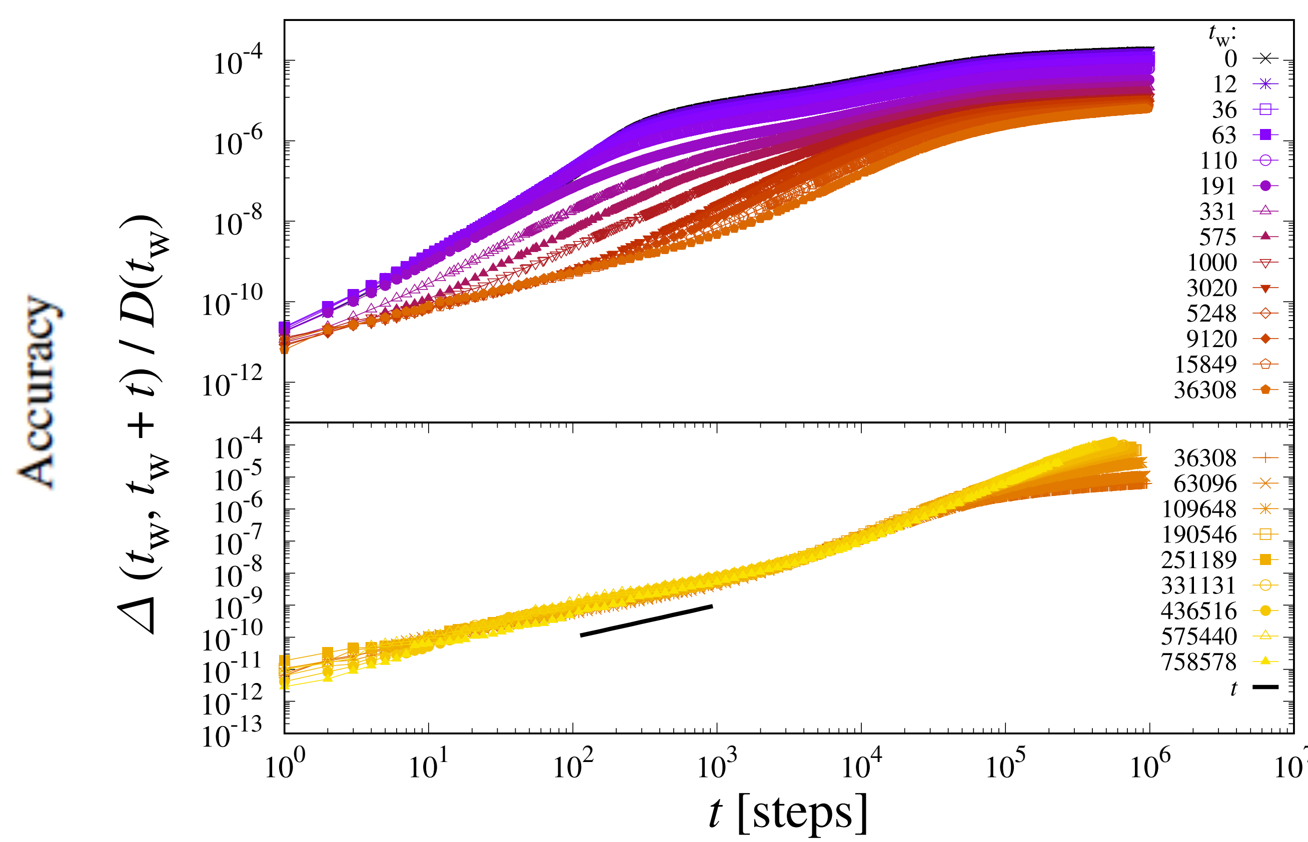


Slow decay of Loss function



(c) Small Net on CIFAR-10, $B = 100$, $\alpha = 0.01$.

Mean Square displacement

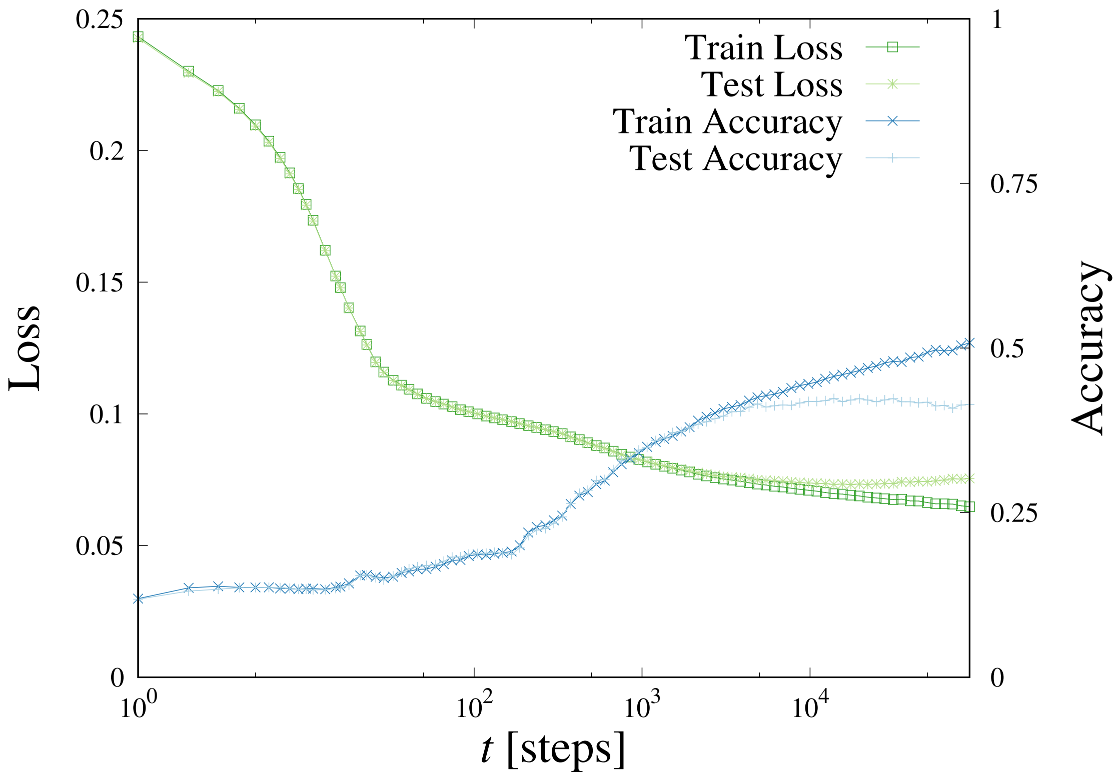


Flat bottom of the Loss landscape!

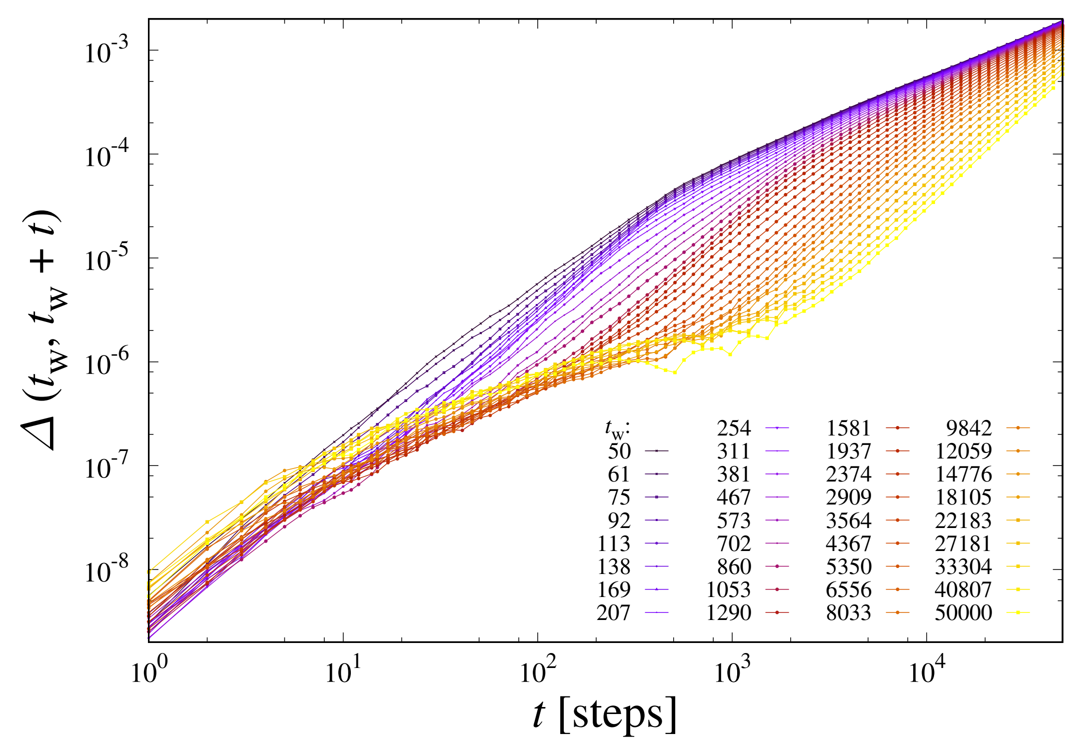
Aging is restored for under-parametrised NN!

Baity-Jesi, Sagun, Geiger, Spiegler, Ben Arous, Cammarota, LeCun, Wyart, Biroli PMLR 2018

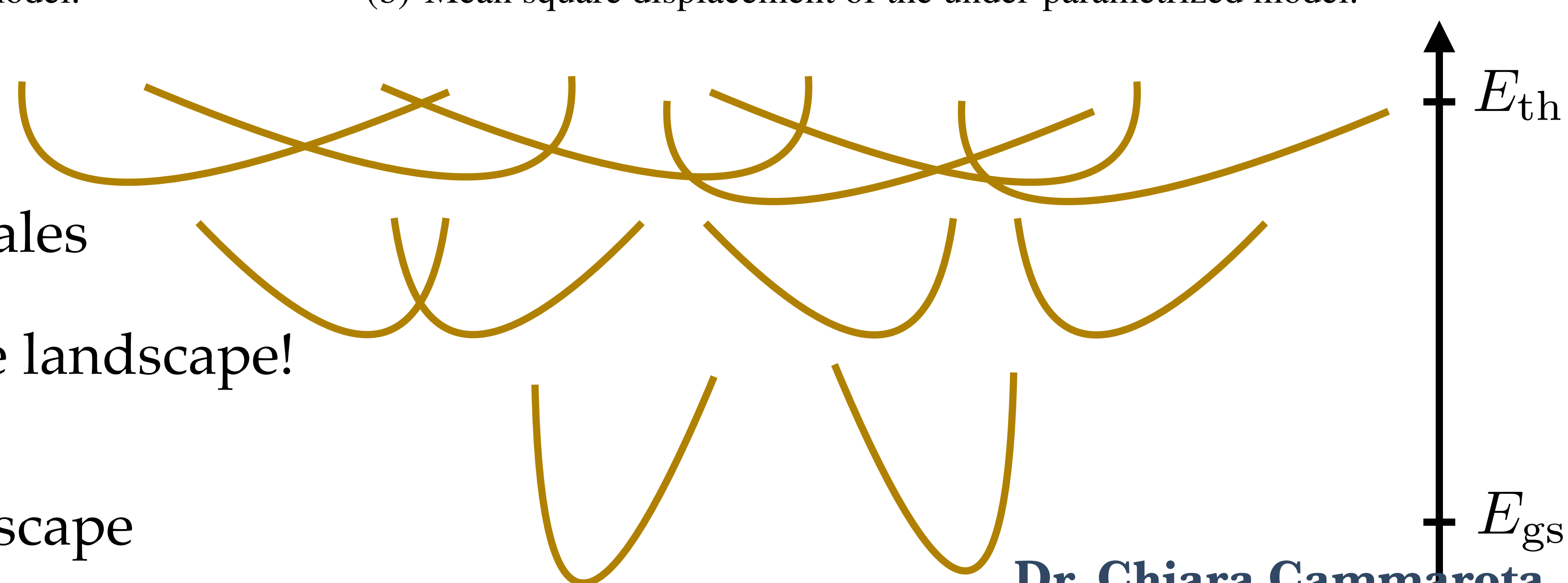
Toy model: 1 hidden layer (MUCH SMALLER), ReLU, sigmoid in output, MSE as a loss



(a) Loss of the under-parametrized model.



(b) Mean square displacement of the under-parametrized model.



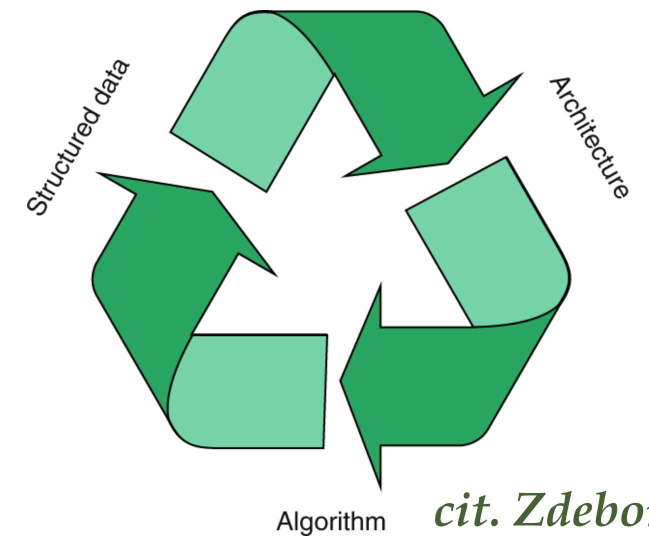
Aging on infinitely long timescales

Not getting to the bottom of the landscape!

Rough bottom of the Loss landscape

Much more on Machine Learning

Three intertwined elements in machine learning:
training algorithm
data structure
network structure



cit. Zdeborova

How SGD works in state of the art machine learning? (path)

Many people (Franz Goldt Saad Saxe Urbani etc)

How generalisation is achieved? (outcome)

Many people (Biroli Montanari Zecchina etc)

How all this can be improved?

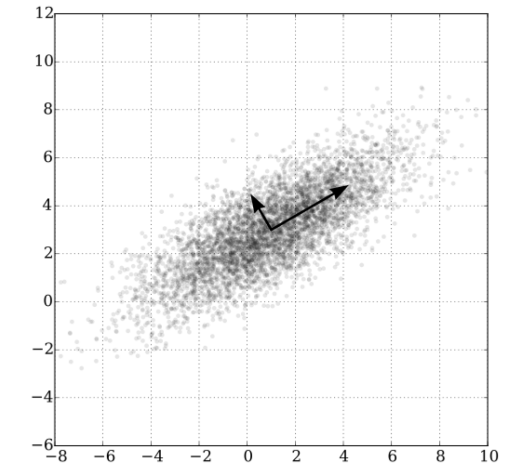
Milder overparametrization
Optimised algorithm (mostly SGD)
Improved use of the data

Inference

From landscape structure to algorithmic predictions..and optimisation

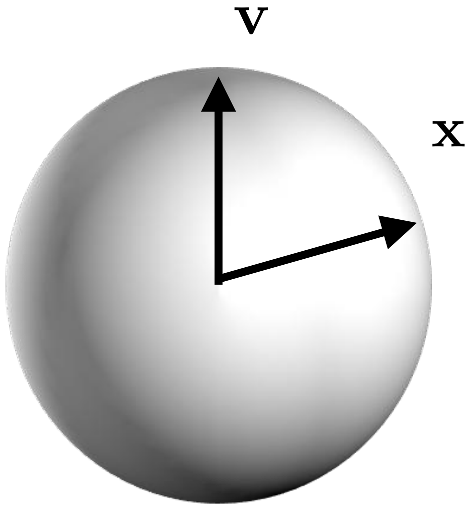
An example of signal reconstruction

MATRIX PCA, TENSOR PCA, MIXED MODELS



An example of signal reconstruction

MATRIX PCA, TENSOR PCA, MIXED MODELS



Estimation of rank-one k-tensor from a noisy channel(s)

Observation Corrupting noise Signal

$$T_{i_1,\ldots,i_k} = W_{i_1,\ldots,i_k} + v_{i_1} \ldots v_{i_k}$$

Maximum likelihood estimator: minimum squared distance

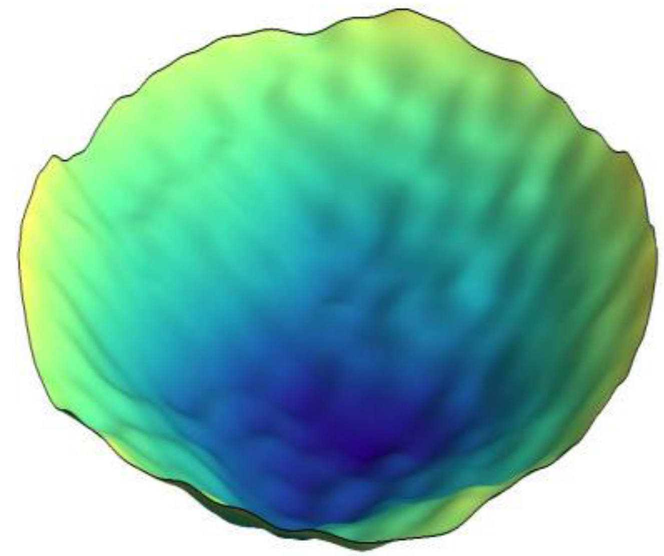
$$H_k = - \sum_{(i_1,\ldots,i_k)} (T_{i_1,\ldots,i_k} - x_{i_1} \ldots x_{i_k})^2 \propto - \sum_{(i_1,\ldots,i_k)} J_{i_1,\ldots,i_k} x_{i_1} \ldots x_{i_k} - rN \left(\sum_i \frac{x_i v_i}{N} \right)^k + const$$

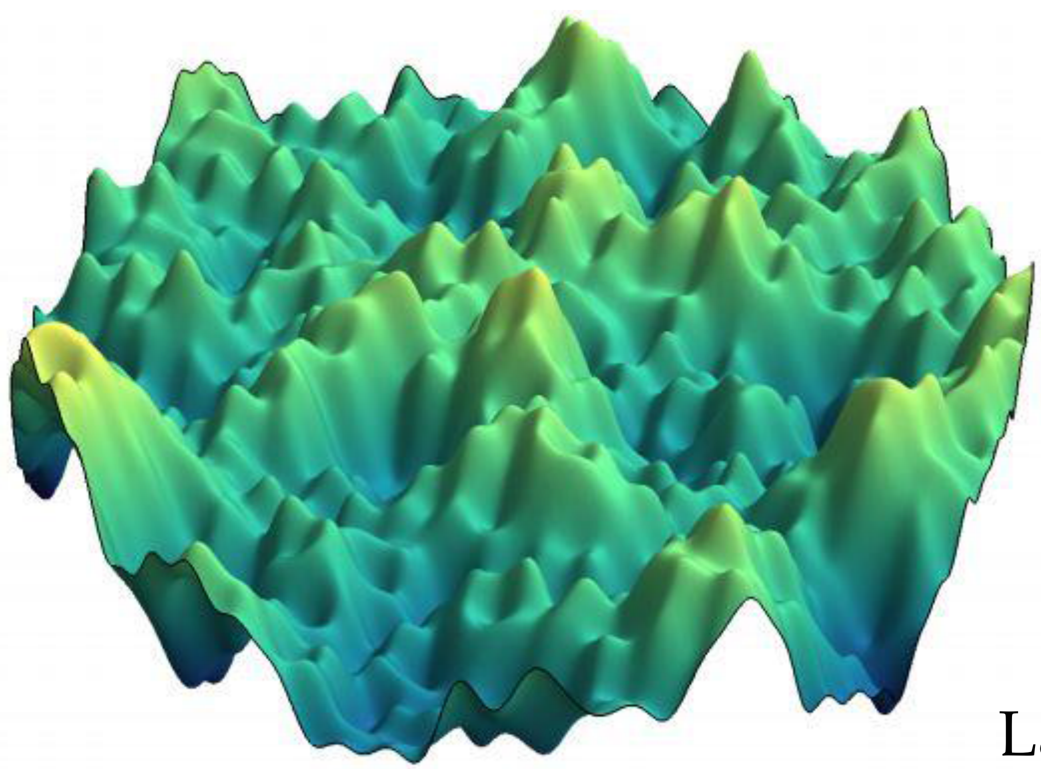
with $J_{i_1,\ldots,i_k} \propto W_{i_1,\ldots,i_k}$ and r signal to noise ratio ..also MIXED matrix / tensor models

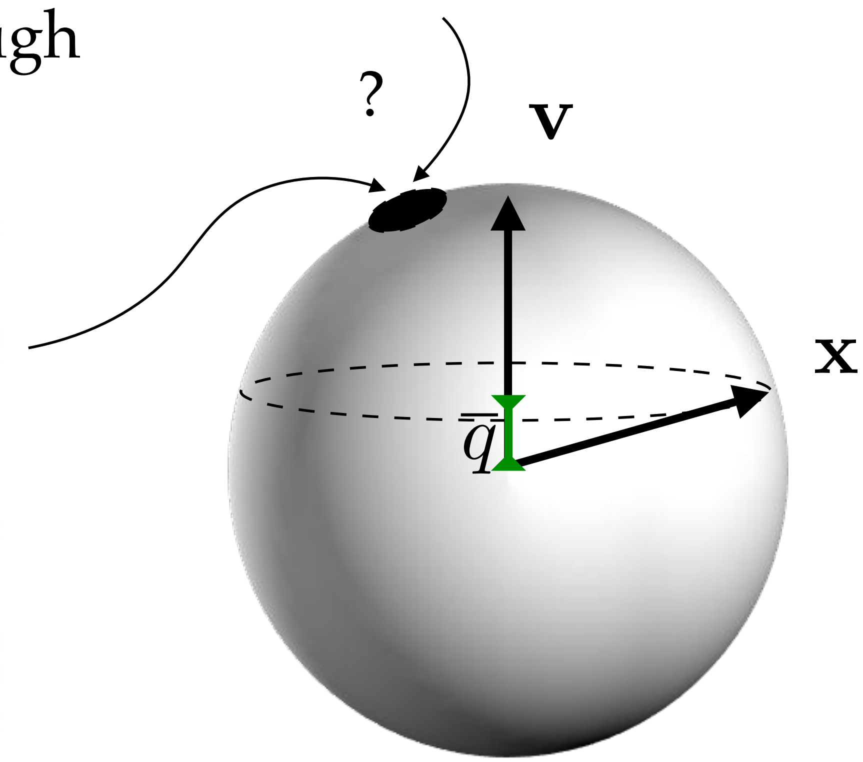
Landscape hints of signal reconstruction

$$\dot{\mathbf{x}} = -\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x}(t)) + \mu(t)\mathbf{x}(t)$$

Minimisation via gradient flow on the sphere from random initial condition, where likelihood / cost landscape is rough







Landscape matter: gradient, Hessian

Tensor PCA: the full landscape structure

Ros, Ben Arous, Biroli, Cammarota **PRX 2019**

Kac-Rice formula to enumerate stationary points (at any risk/likelyhood level and latitude)

$$\mathcal{N}_N(E,\overline{q};r) = \int \prod_i dx_i \delta(\nabla_x H_r) |\det \nabla^2 H| \delta(H-E) \delta\left(\sum_i v_i x_i - N\overline{q}\right)$$

Beyond annealed computation: Replicated Kac-Rice

Subag 2015

$$\langle \log \mathcal{N}_N(E,\overline{q};r)\rangle = \lim_{n\to 0} \frac{\langle \mathcal{N}(E,\overline{q};r)^n\rangle - 1}{n}$$

> Structure of stationary points

> Distribution of Hessians eigenvalues

Tensor PCA: the full landscape structure

Ros, Ben Arous, Biroli, Cammarota **PRX 2019**

Kac-Rice formula to enumerate stationary points (at any risk/likelyhood level and latitude)

$$\mathcal{N}_N(E,\overline{q};r) = \int \prod_i dx_i \delta(\nabla_x H_r) |\det \nabla^2 H| \delta(H-E)\delta\left(\sum_i v_i x_i - N\overline{q}\right)$$

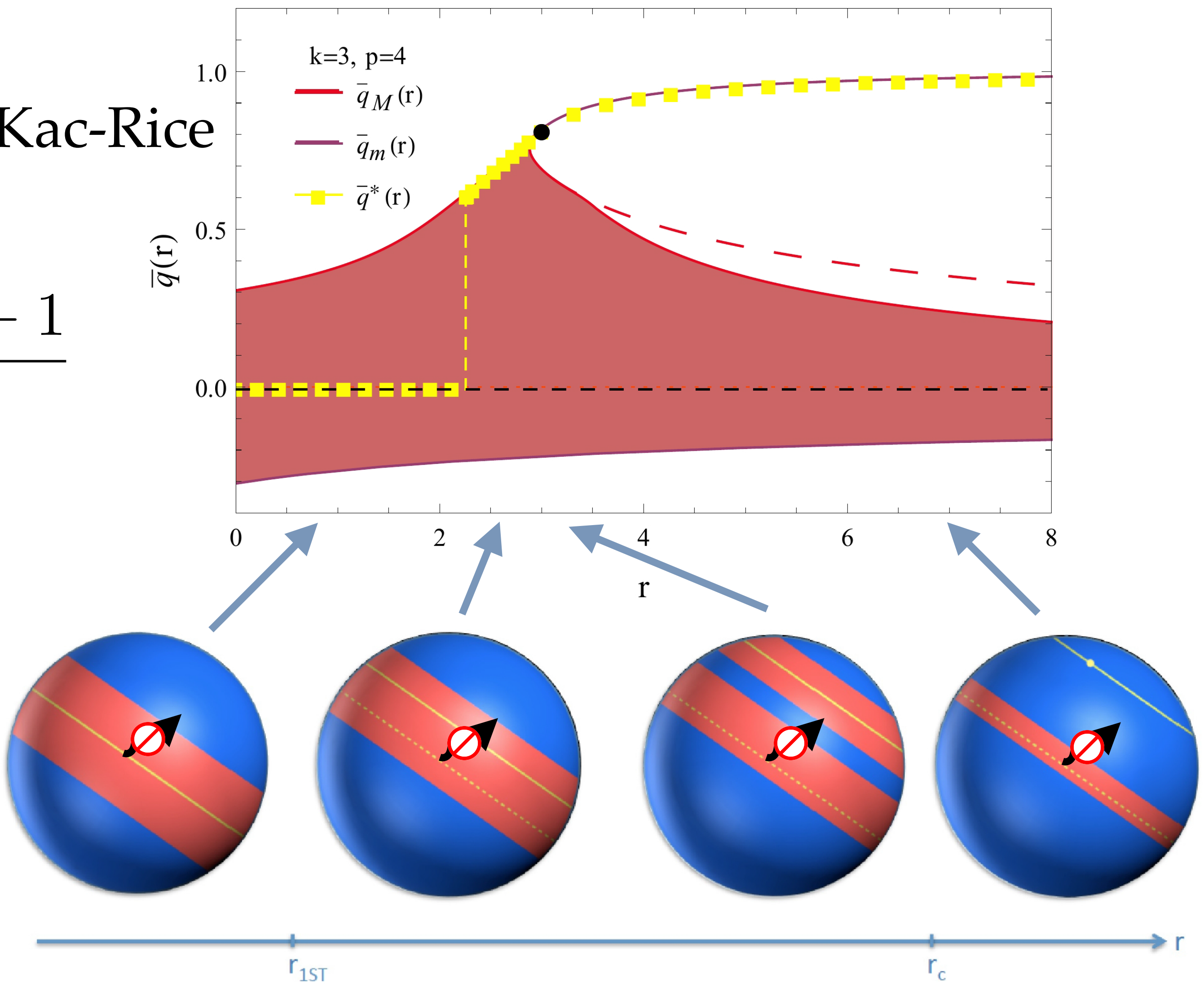
Beyond annealed computation: Replicated Kac-Rice

Subag 2015

$$\langle \log \mathcal{N}_N(E,\overline{q};r)\rangle = \lim_{n\to 0} \frac{\langle \mathcal{N}(E,\overline{q};r)^n\rangle - 1}{n}$$

> Structure of stationary points

> Distribution of Hessians eigenvalues



Tensor PCA: the full landscape structure

Ros, Ben Arous, Biroli, Cammarota **PRX 2019**

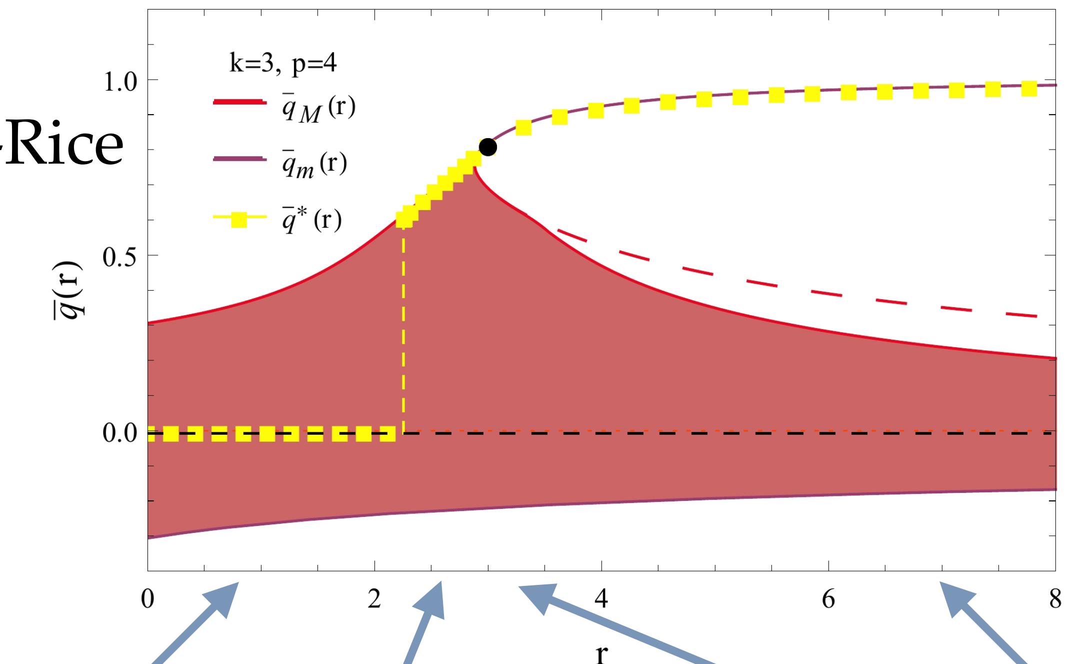
Kac-Rice formula to enumerate stationary points (at any risk/likelyhood level and latitude)

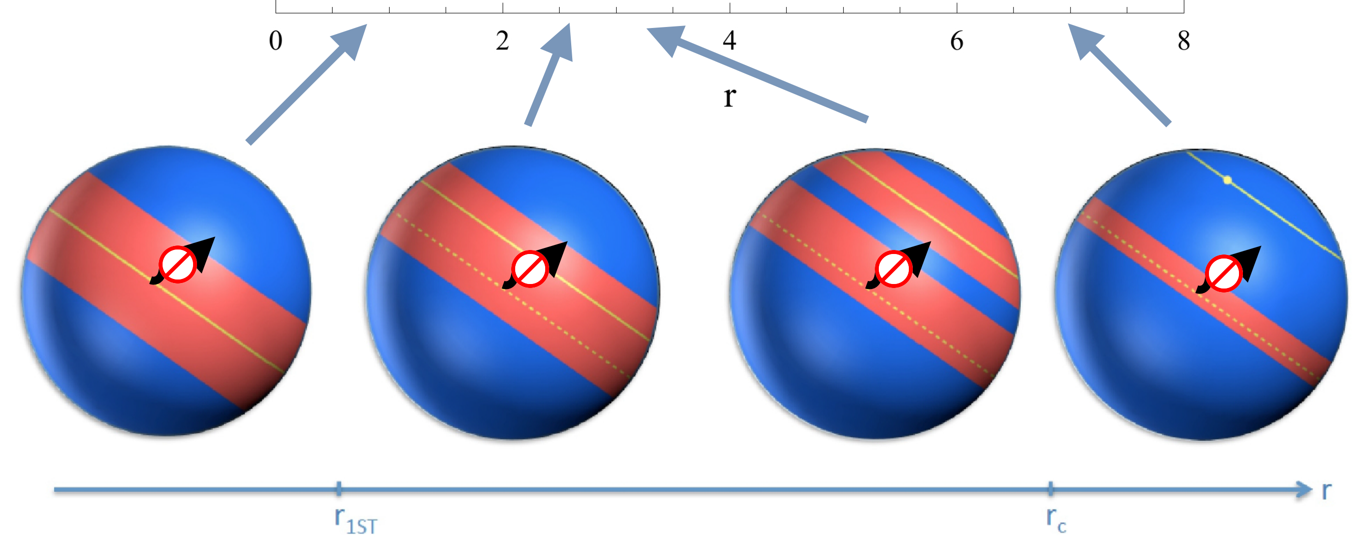
$$\mathcal{N}_N(E,\overline{q};r) = \int \prod_i dx_i \delta(\nabla_x H_r)|\det \nabla^2 H|\delta(H-E)\delta\left(\sum_i v_i x_i - N\overline{q}\right)$$

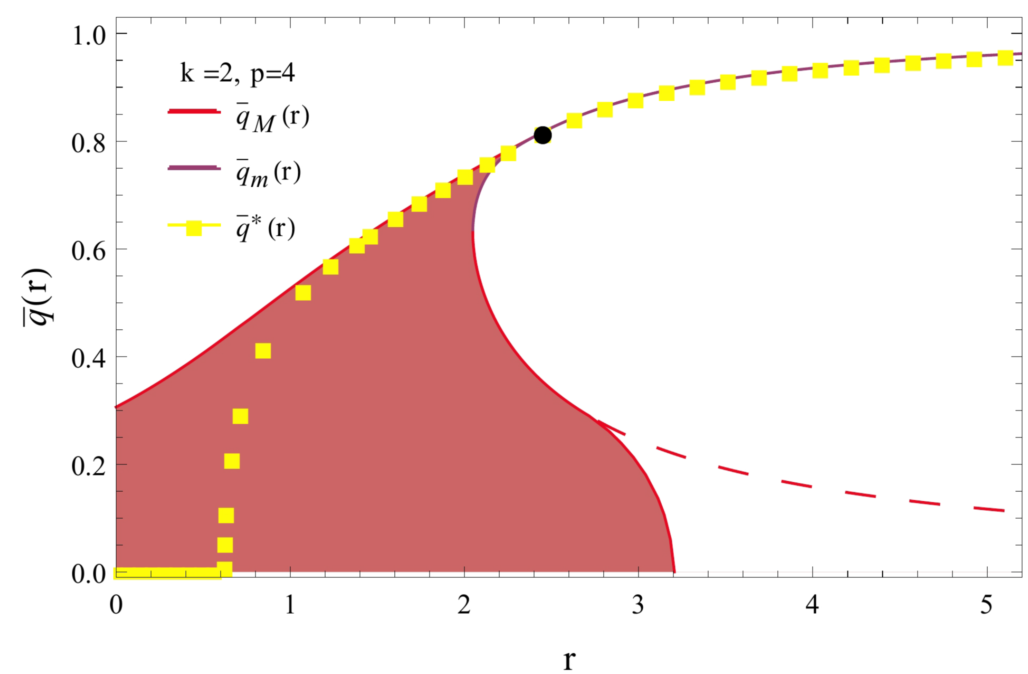
Beyond annealed computation: Replicated Kac-Rice

Subag 2015

$$\langle \log \mathcal{N}_N(E,\overline{q};r)\rangle = \lim_{n\to 0} \frac{\langle \mathcal{N}(E,\overline{q};r)^n\rangle - 1}{n}$$







Matrix-Tensor PCA: how gradient flow escapes minima

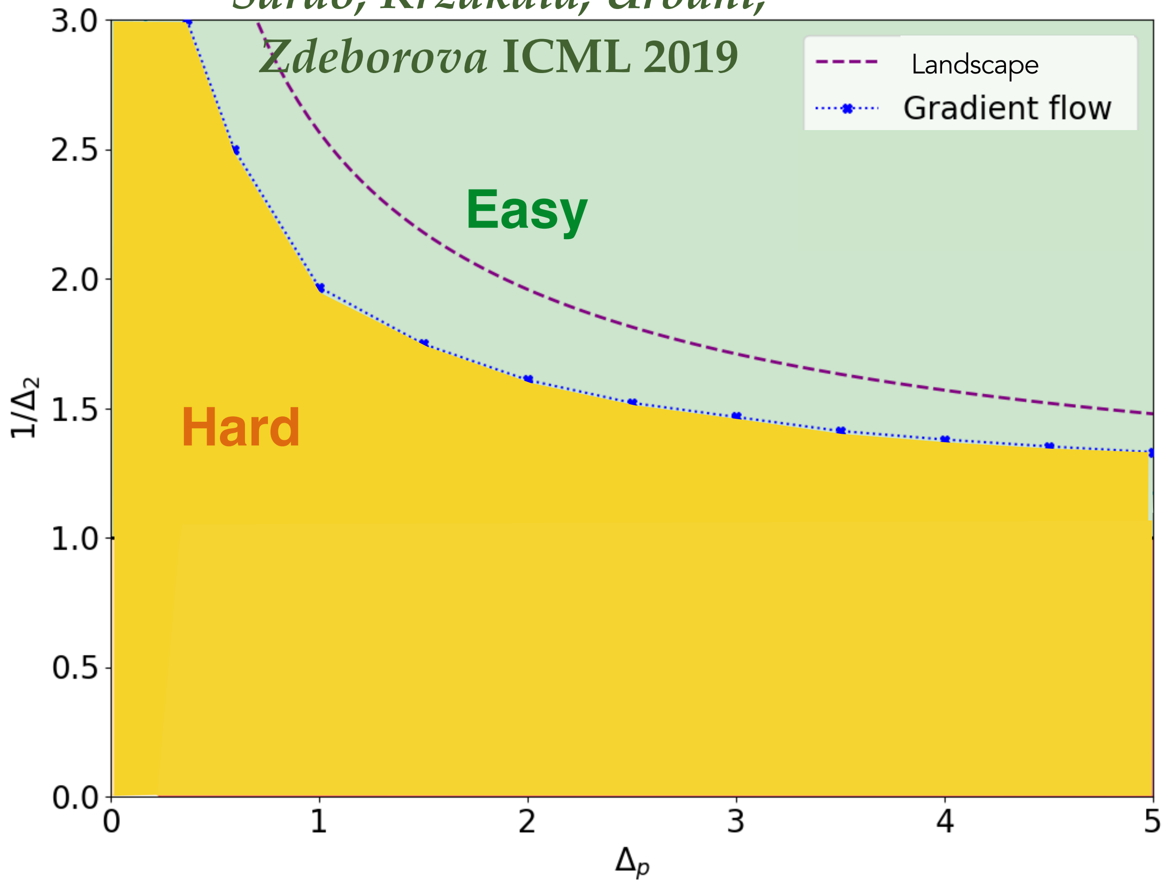
Sarao, Biroli, Cammarota, Krzakala, Zdeborova **Spotlight at NIPS 2019**

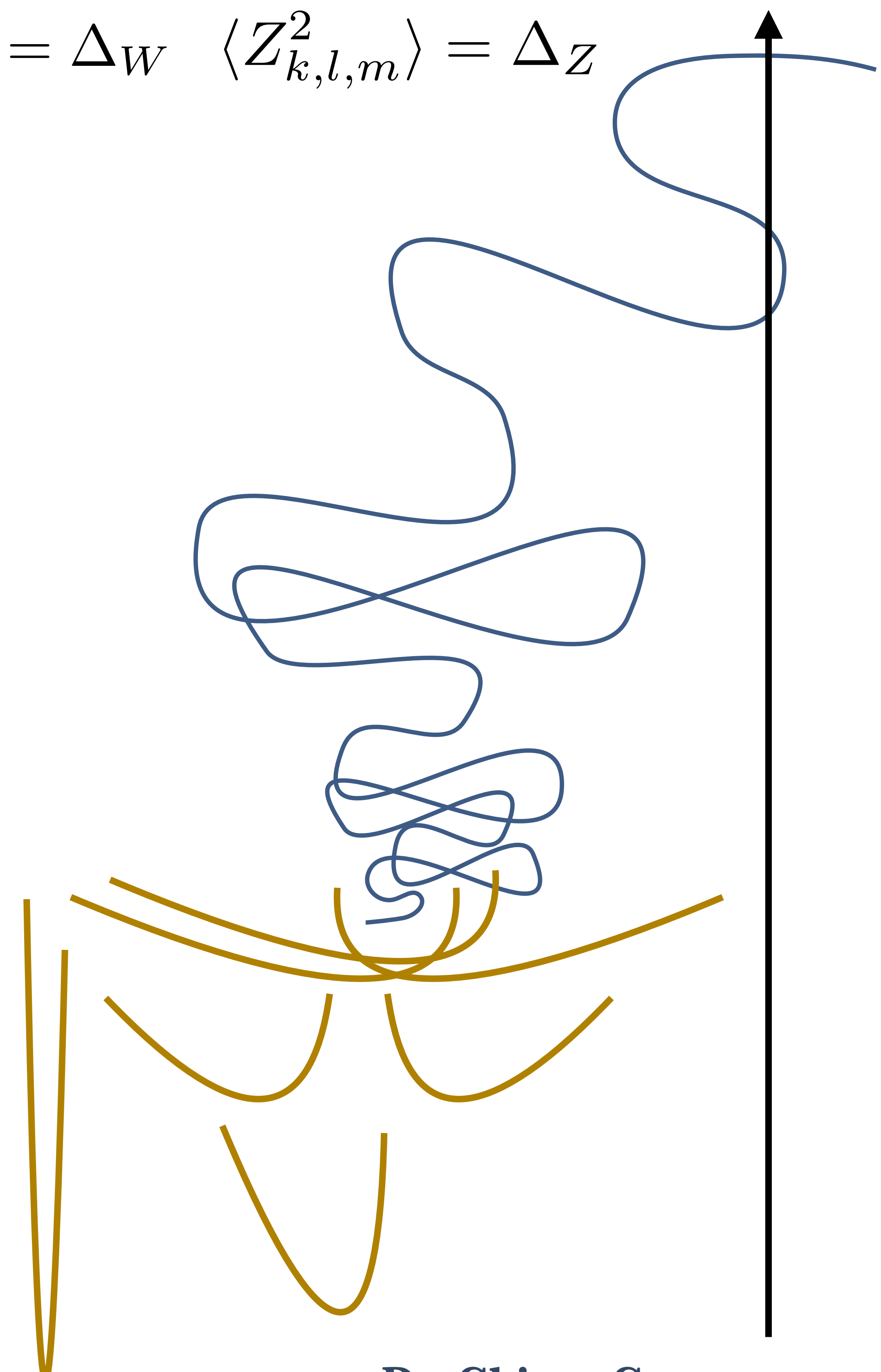
$$T_{i,j} = W_{i,j} + v_i v_j$$

$$S_{k,l,m} = Z_{k,l,m} + v_k v_l v_m$$

$$\langle W_{i,j}^2 \rangle = \Delta_W \quad \langle Z_{k,l,m}^2 \rangle = \Delta_Z$$

Sarao, Krzakala, Urbani, Zdeborova **ICML 2019**





Matrix-Tensor PCA: how gradient flow escapes minima

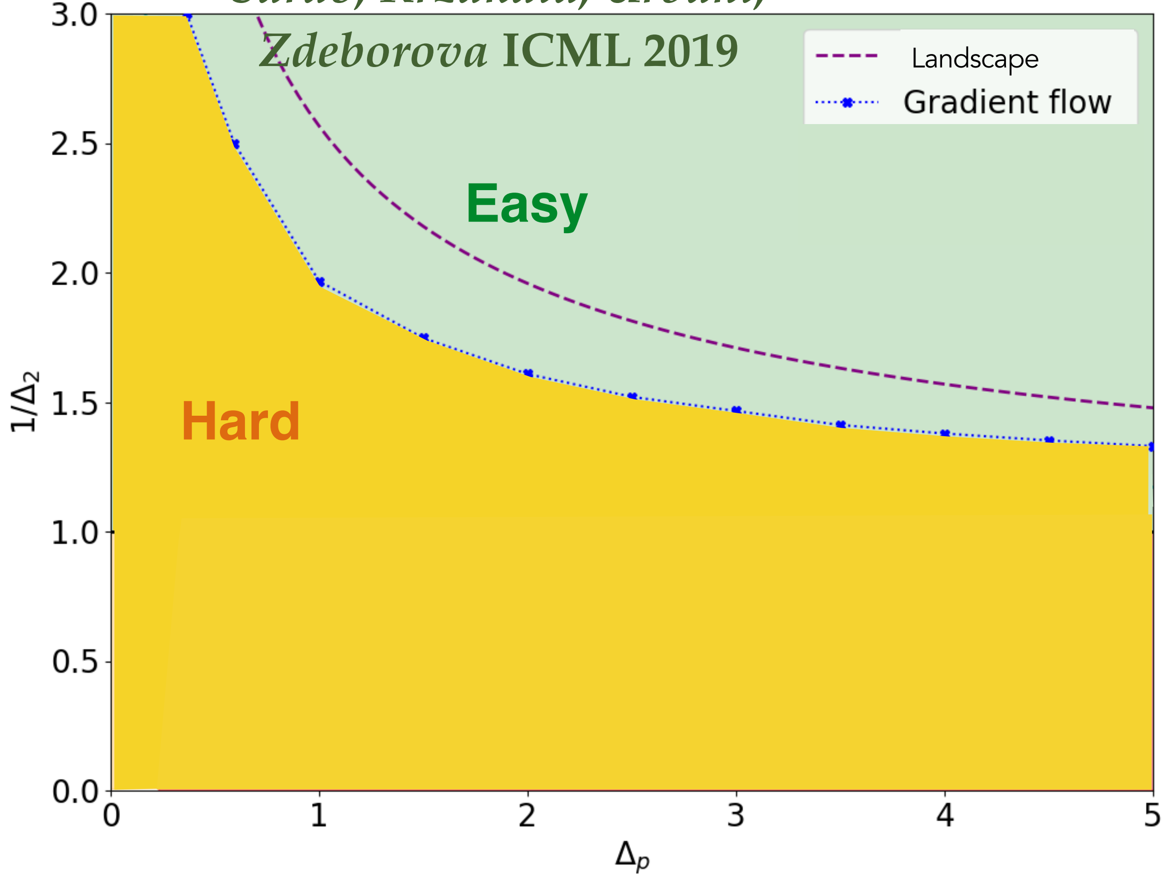
Sarao, Biroli, Cammarota, Krzakala, Zdeborova Spotlight at NIPS 2019

$$T_{i,j} = W_{i,j} + v_i v_j$$

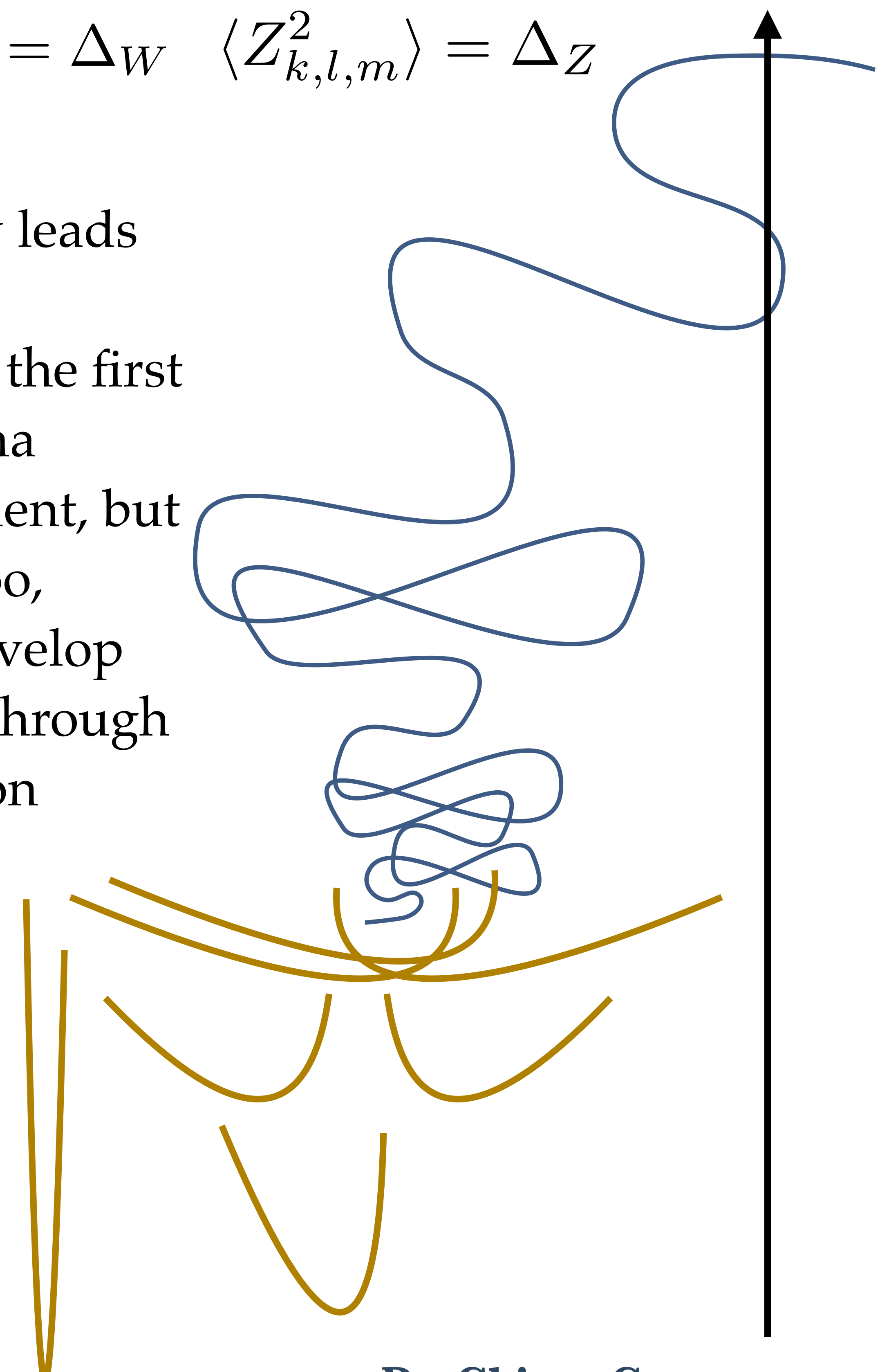
$$S_{k,l,m} = Z_{k,l,m} + v_k v_l v_m$$

$$\langle W_{i,j}^2 \rangle = \Delta_W \quad \langle Z_{k,l,m}^2 \rangle = \Delta_Z$$

Sarao, Krzakala, Urbani, Zdeborova ICML 2019



Gradient Flow leads to thorough exploration of the first layer of minima the most frequent, but most fragile too, …and they develop an instability through a BBP transition



Matrix-Tensor PCA: how gradient flow escapes minima

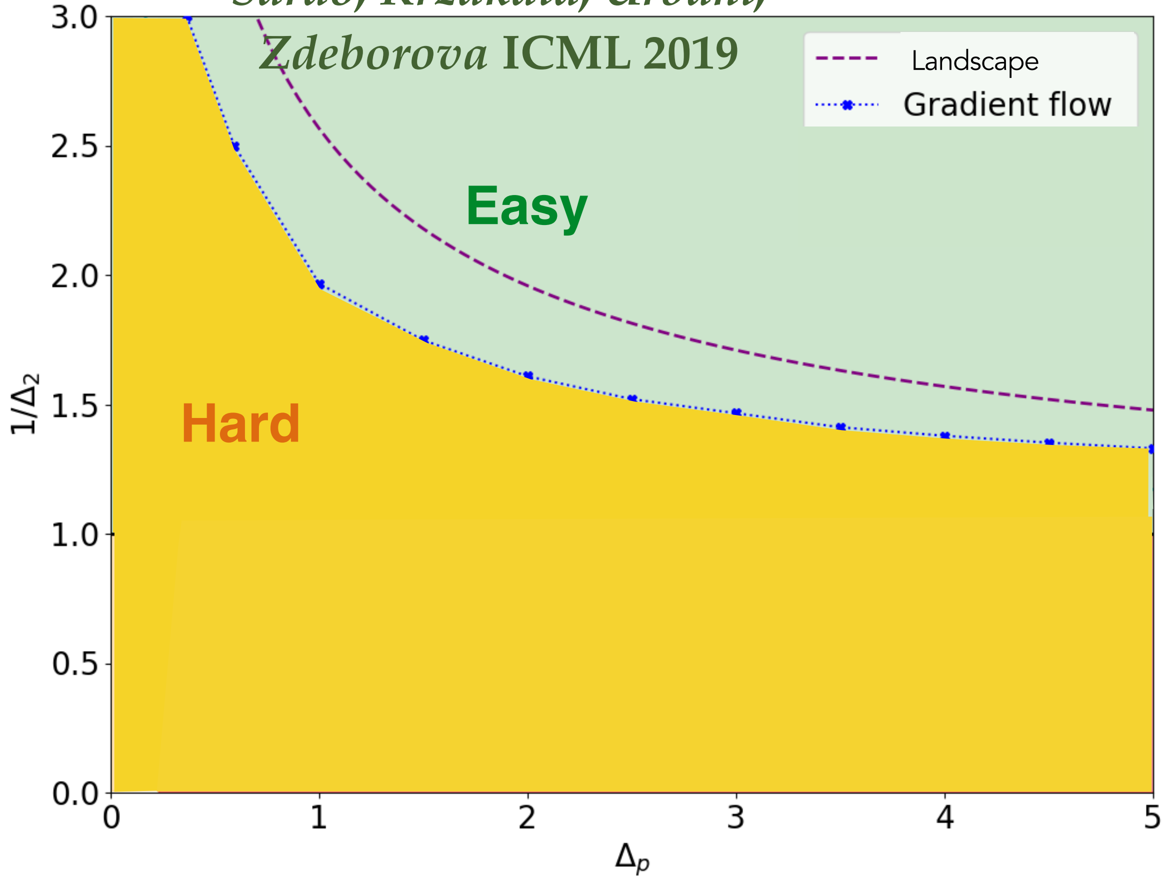
Sarao, Biroli, Cammarota, Krzakala, Zdeborova Spotlight at NIPS 2019

$$T_{i,j} = W_{i,j} + v_i v_j$$

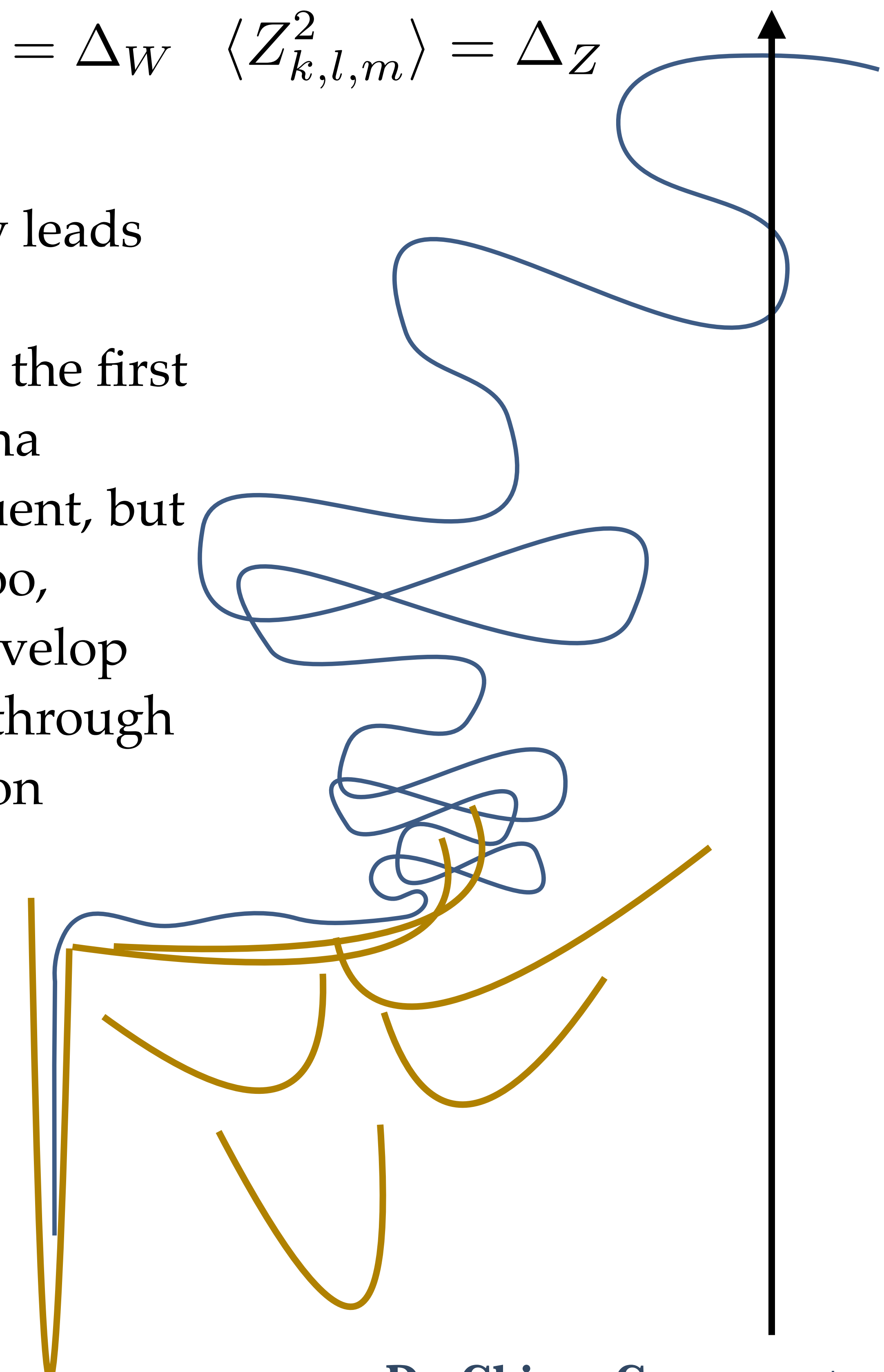
$$S_{k,l,m} = Z_{k,l,m} + v_k v_l v_m$$

$$\langle W_{i,j}^2 \rangle = \Delta_W \quad \langle Z_{k,l,m}^2 \rangle = \Delta_Z$$

Sarao, Krzakala, Urbani, Zdeborova ICML 2019



Gradient Flow leads to thorough exploration of the first layer of minima the most frequent, but most fragile too, …and they develop an instability through a BBP transition



Matrix-Tensor PCA: how gradient flow escapes minima

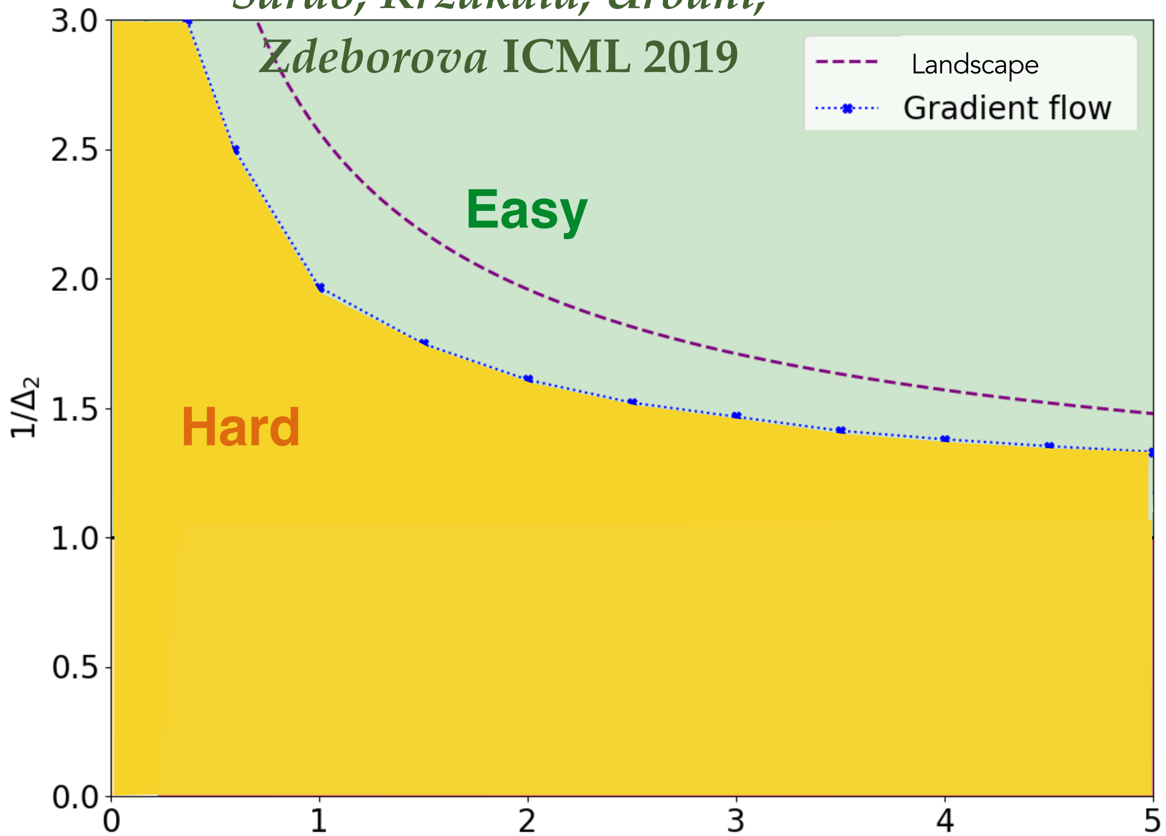
Sarao, Biroli, Cammarota, Krzakala, Zdeborova **Spotlight at NIPS 2019**

$$T_{i,j} = W_{i,j} + v_i v_j$$

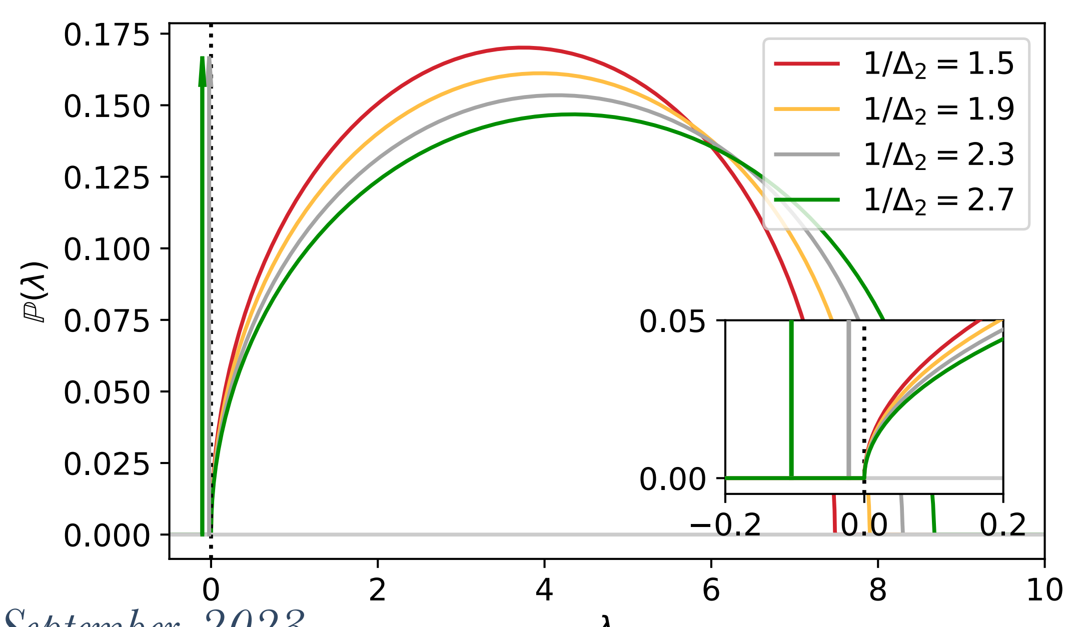
$$S_{k,l,m} = Z_{k,l,m} + v_k v_l v_m$$

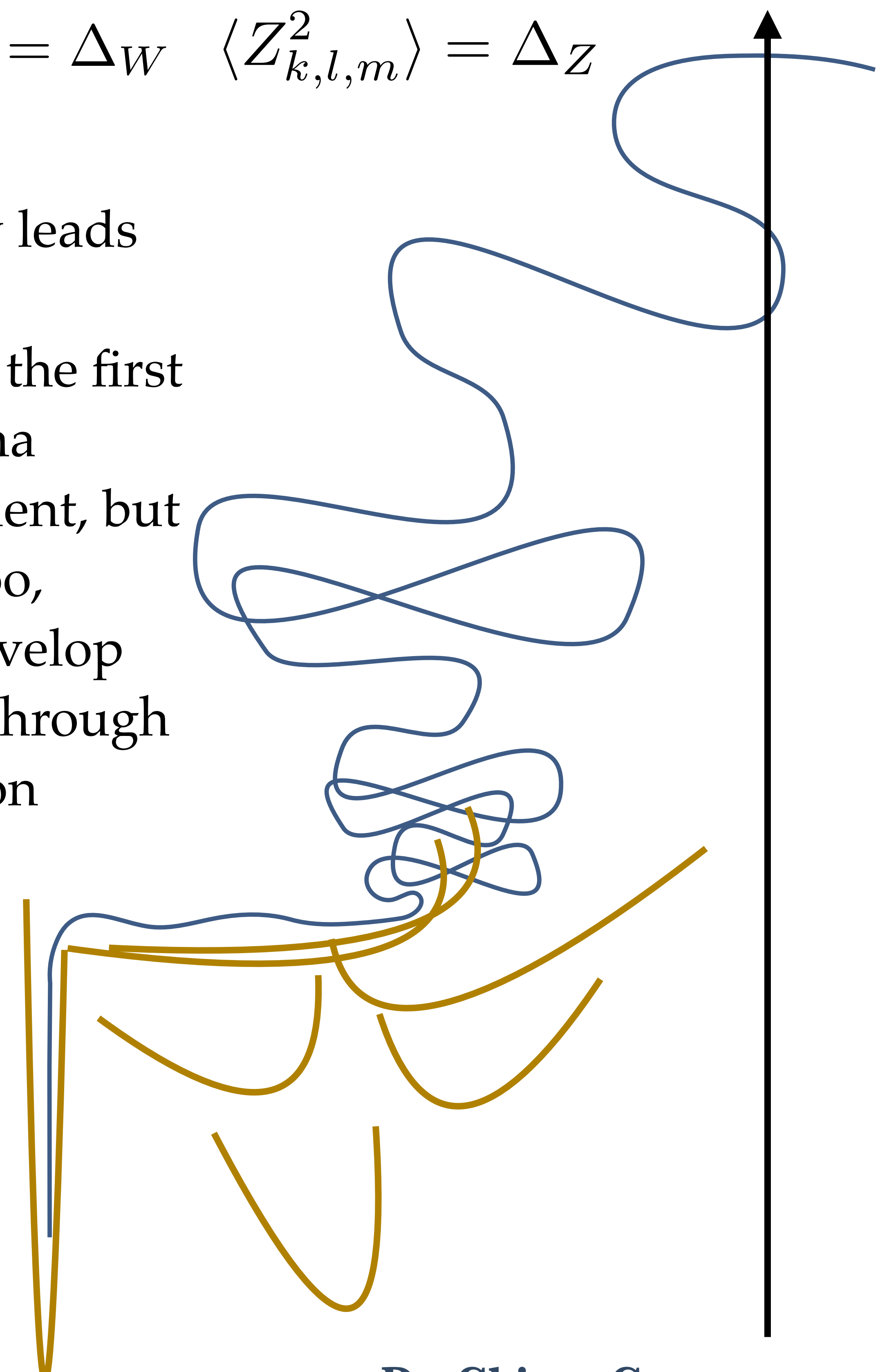
$$\langle W_{i,j}^2 \rangle = \Delta_W \quad \langle Z_{k,l,m}^2 \rangle = \Delta_Z$$

Sarao, Krzakala, Urbani, Zdeborova **ICML 2019**



Gradient Flow leads to thorough exploration of the first layer of minima the most frequent, but most fragile too,
…and they develop an instability through a BBP transition





Matrix-Tensor PCA: how gradient flow escapes minima

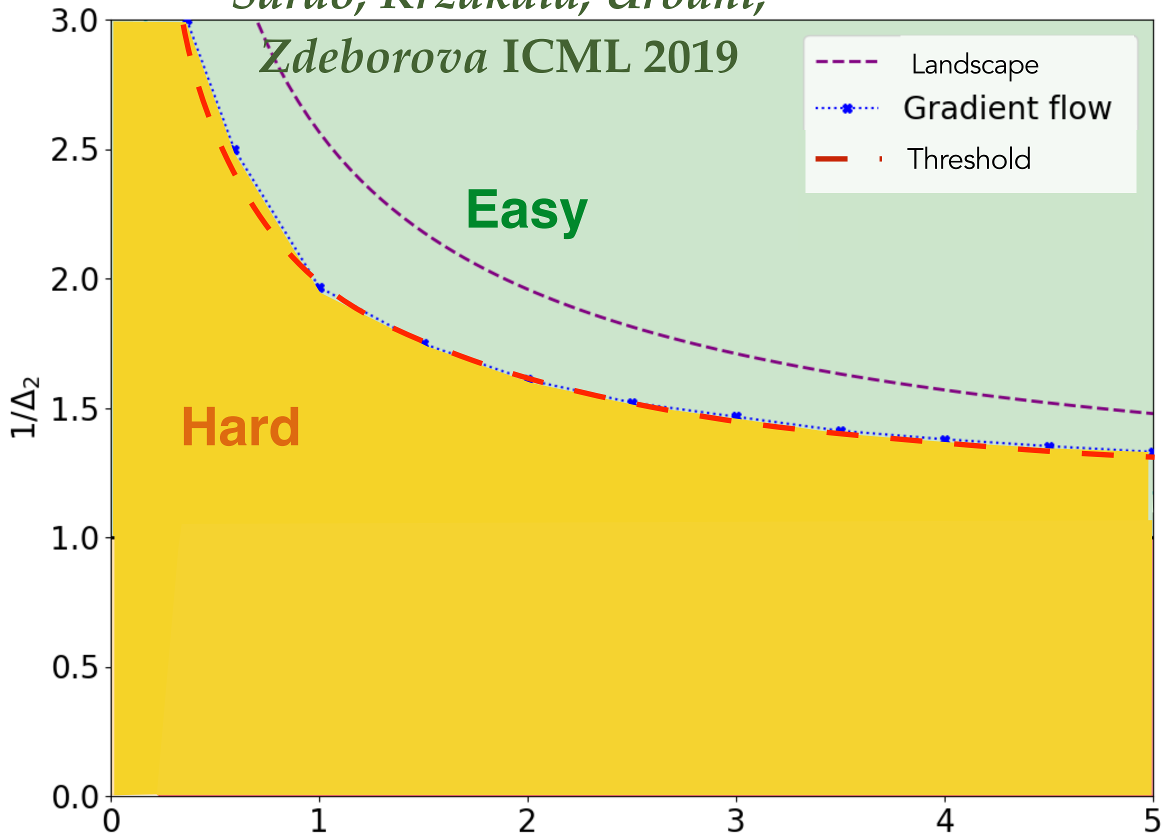
Sarao, Biroli, Cammarota, Krzakala, Zdeborova **Spotlight at NIPS 2019**

$$T_{i,j} = W_{i,j} + v_i v_j$$

$$S_{k,l,m} = Z_{k,l,m} + v_k v_l v_m$$

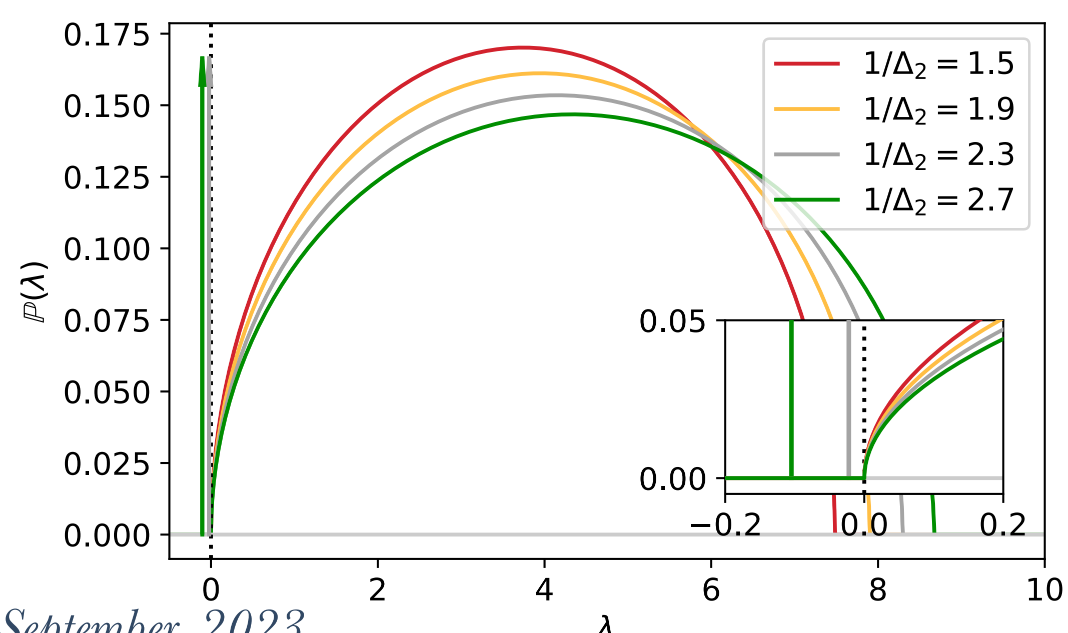
$$\langle W_{i,j}^2 \rangle = \Delta_W \quad \langle Z_{k,l,m}^2 \rangle = \Delta_Z$$

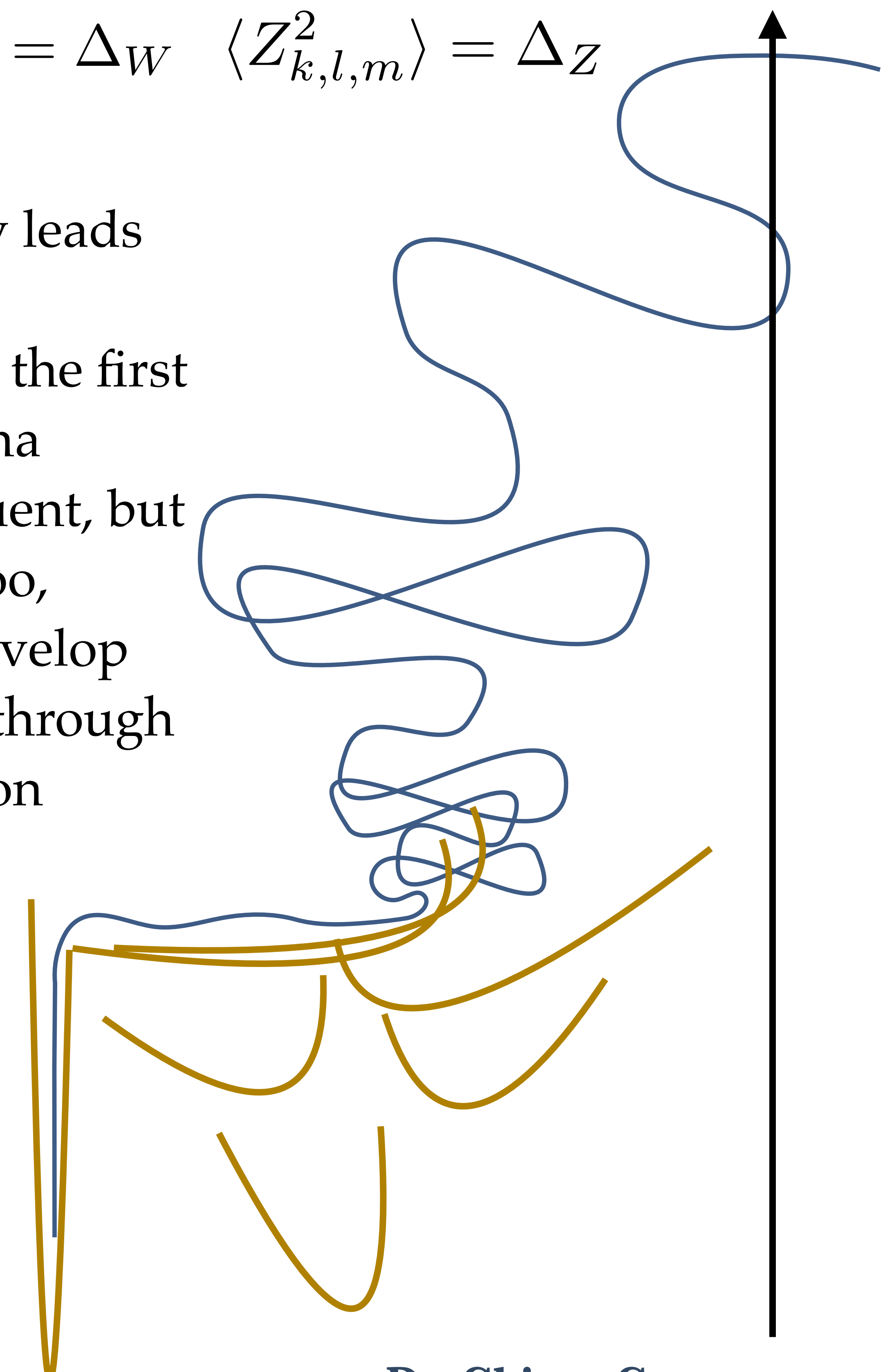
Sarao, Krzakala, Urbani, Zdeborova **ICML 2019**



Gradient Flow leads to thorough exploration of the first layer of minima the most frequent, but most fragile too,

…and they develop an instability through a BBP transition





When less is better: AMP vs Langevin

Sarao, Biroli, Cammarota, Krzakala, Urbani, Zdeborova **PRX 2020**

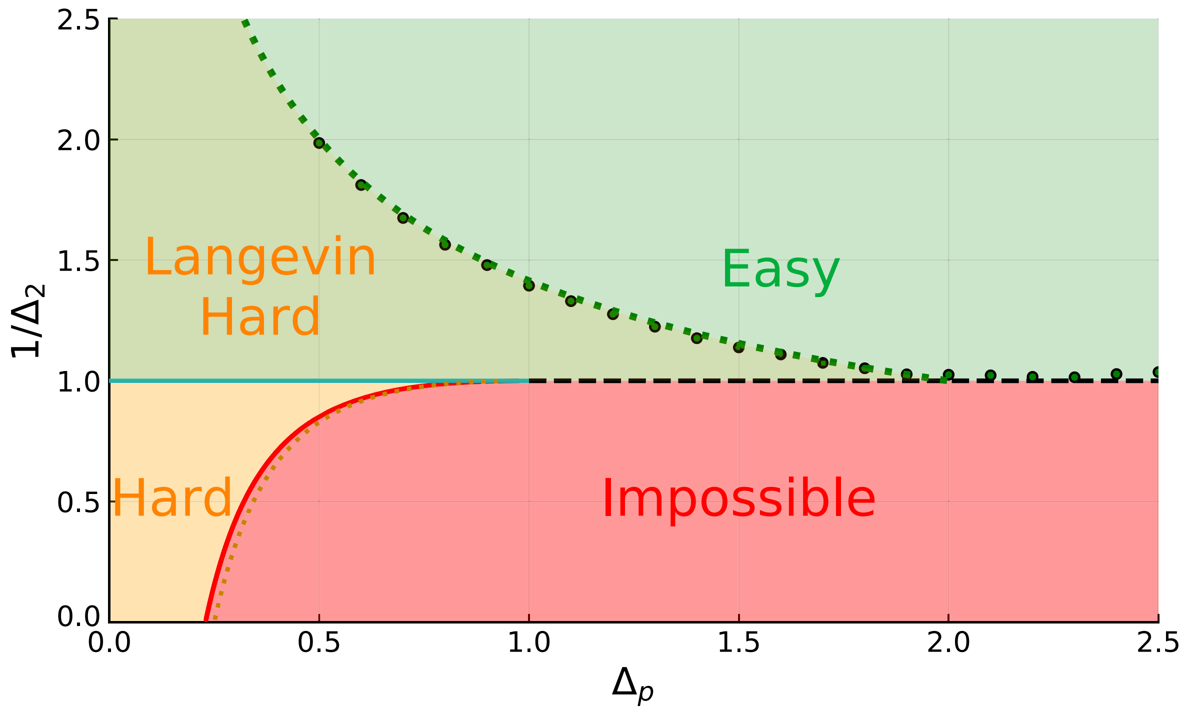
$$T_{i,j} = W_{i,j} + v_i v_j$$

$$S_{k,l,m} = Z_{k,l,m} + v_k v_l v_m$$

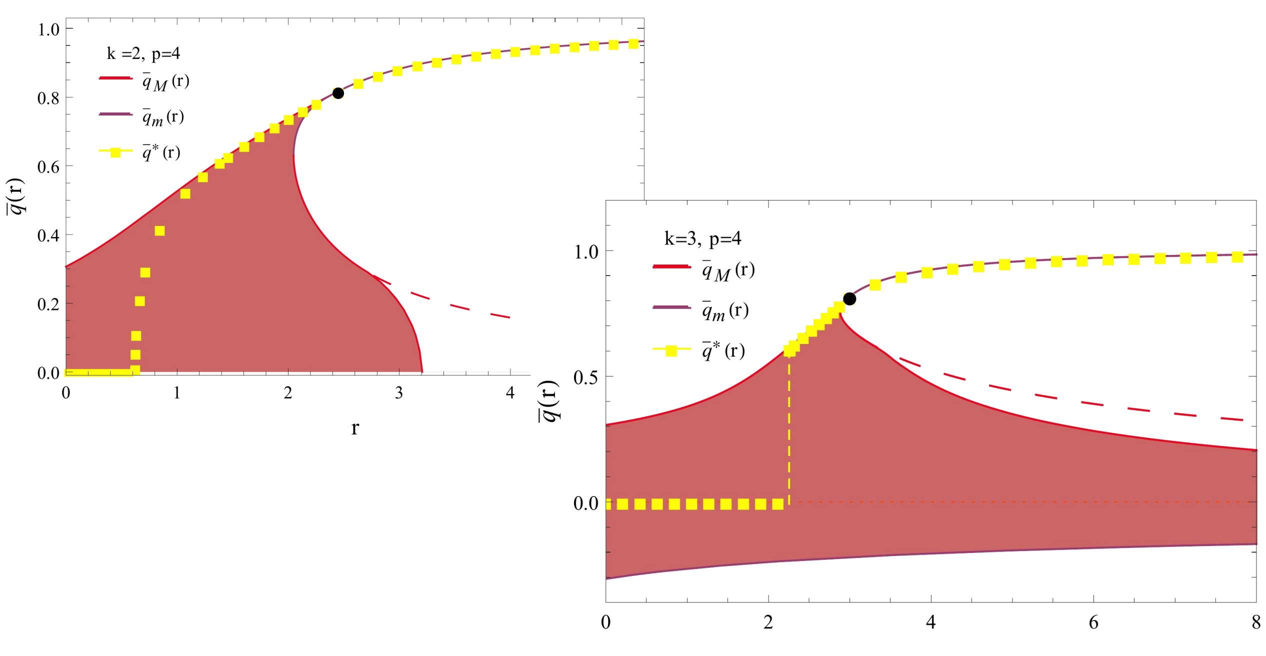
$$\langle W_{i,j}^2 \rangle = \Delta_W \quad \langle Z_{k,l,m}^2 \rangle = \Delta_Z$$

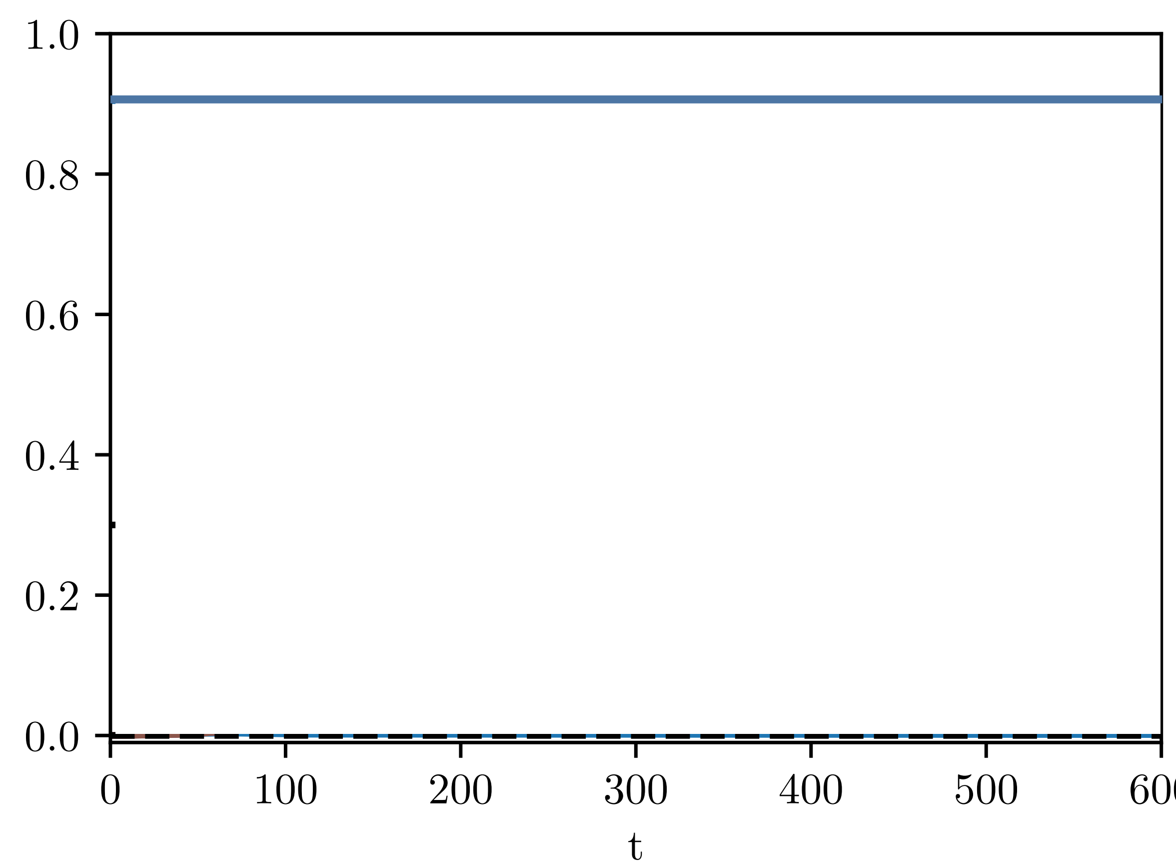
$$H = -\sum_{(i_1,\dots,i_p)} J_{i_1,\dots,i_p} x_{i_1} \dots x_{i_p} - rN \left(\sum_i \frac{x_i v_i}{N} \right)^k$$

$$H_{\text{tot}} = H_{p=2,k=2} + H_{p=3,k=3}$$



AMP much better than Langevin!



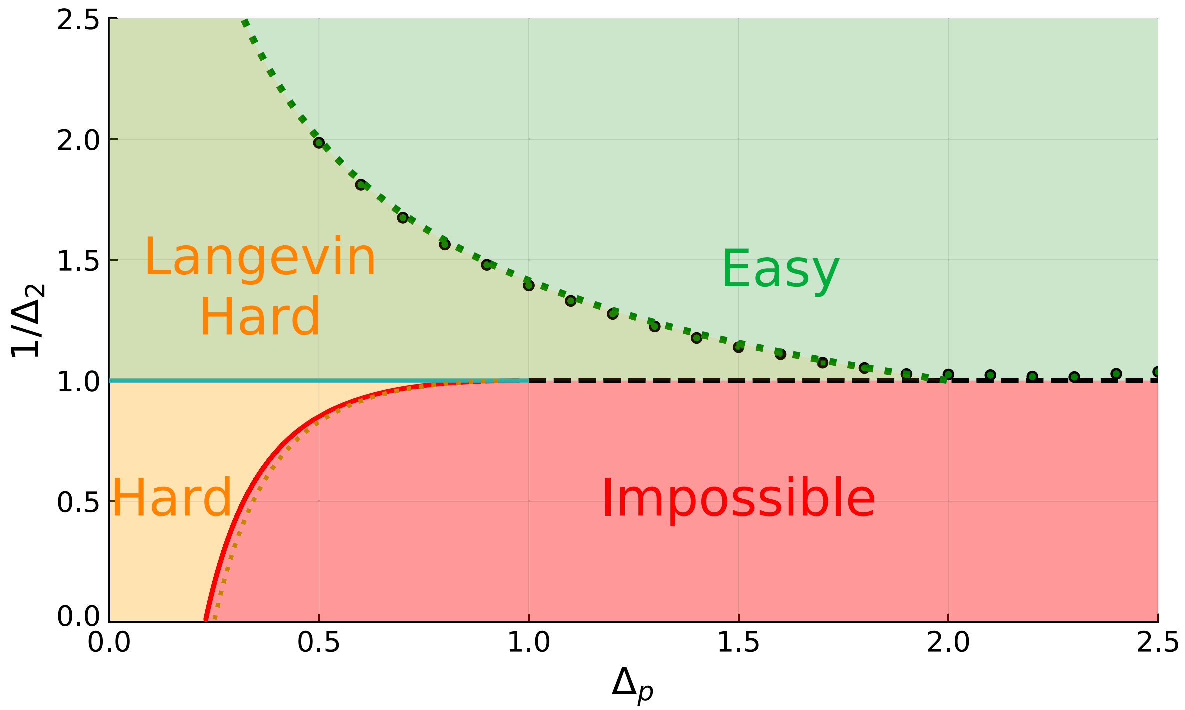


When less is better: AMP vs Langevin

Sarao, Biroli, Cammarota, Krzakala, Urbani, Zdeborova **PRX 2020**

$$T_{i,j} = W_{i,j} + v_i v_j$$
$$S_{k,l,m} = Z_{k,l,m} + v_k v_l v_m$$

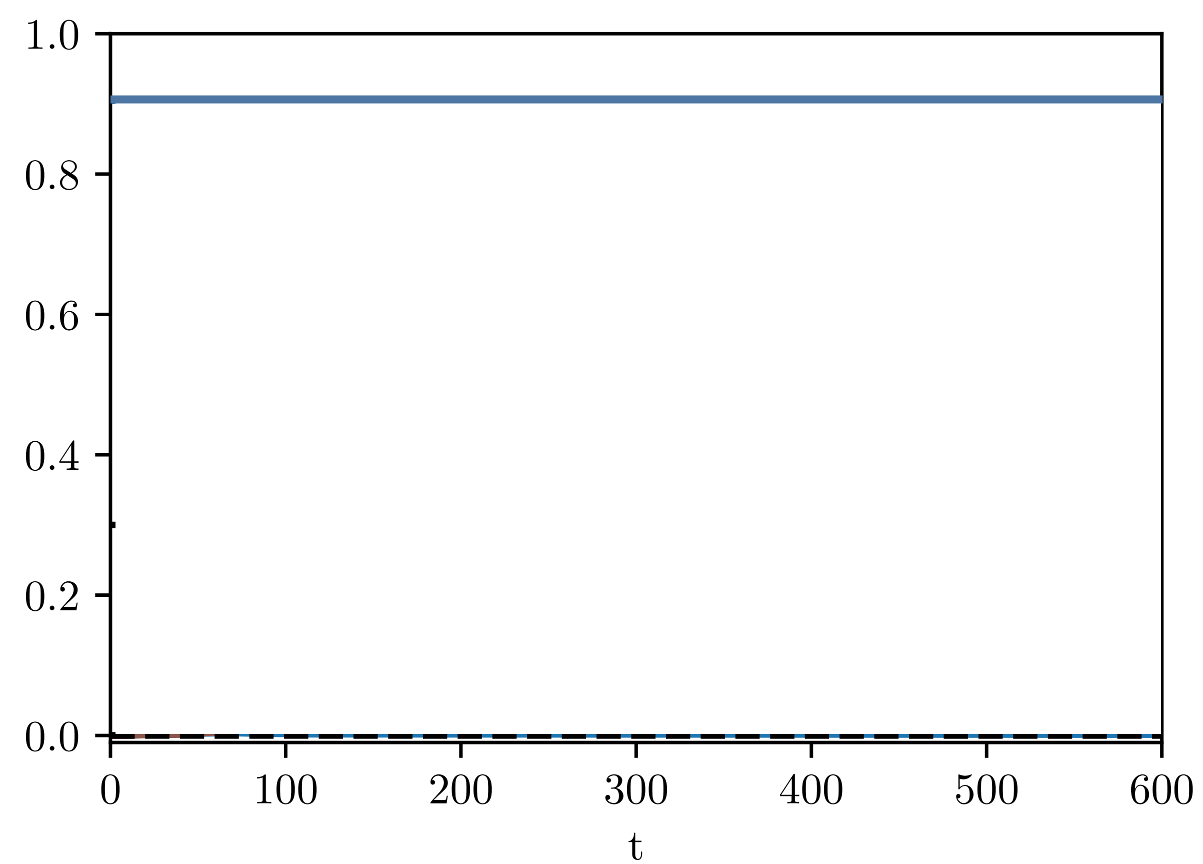
$$\langle W_{i,j}^2 \rangle = \Delta_W \quad \langle Z_{k,l,m}^2 \rangle = \Delta_Z$$



$$H = - \sum_{(i_1,\dots,i_p)} J_{i_1,\dots,i_p} x_{i_1} \dots x_{i_p} - rN \left(\sum_i \frac{x_i v_i}{N} \right)^k$$

$$H_{\text{tot}} = H_{p=2,k=2} + H_{p=3,k=3}$$

AMP much better than Langevin!



However Langevin would work more efficiently on $H_{p=2,k=2}$ only
… at least in the vicinity of the equator

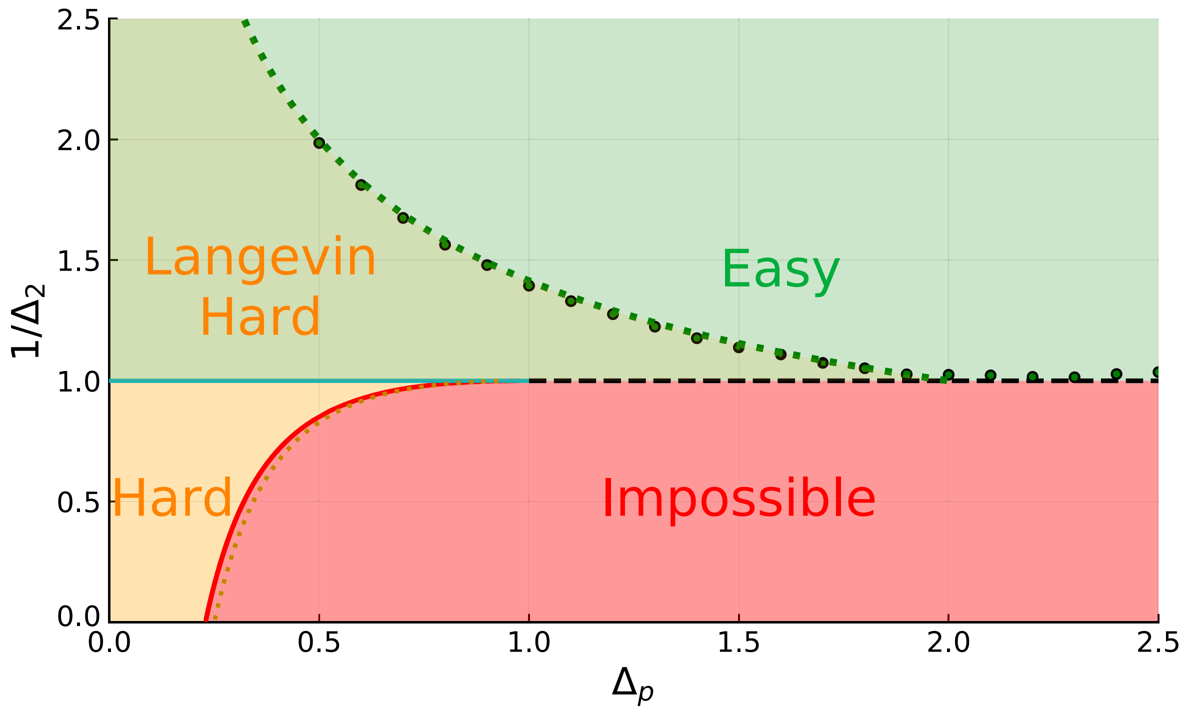
When less is better: AMP vs Langevin

Sarao, Biroli, Cammarota, Krzakala, Urbani, Zdeborova **PRX 2020**

$$T_{i,j} = W_{i,j} + v_i v_j$$

$$S_{k,l,m} = Z_{k,l,m} + v_k v_l v_m$$

$$\langle W_{i,j}^2 \rangle = \Delta_W \quad \langle Z_{k,l,m}^2 \rangle = \Delta_Z$$

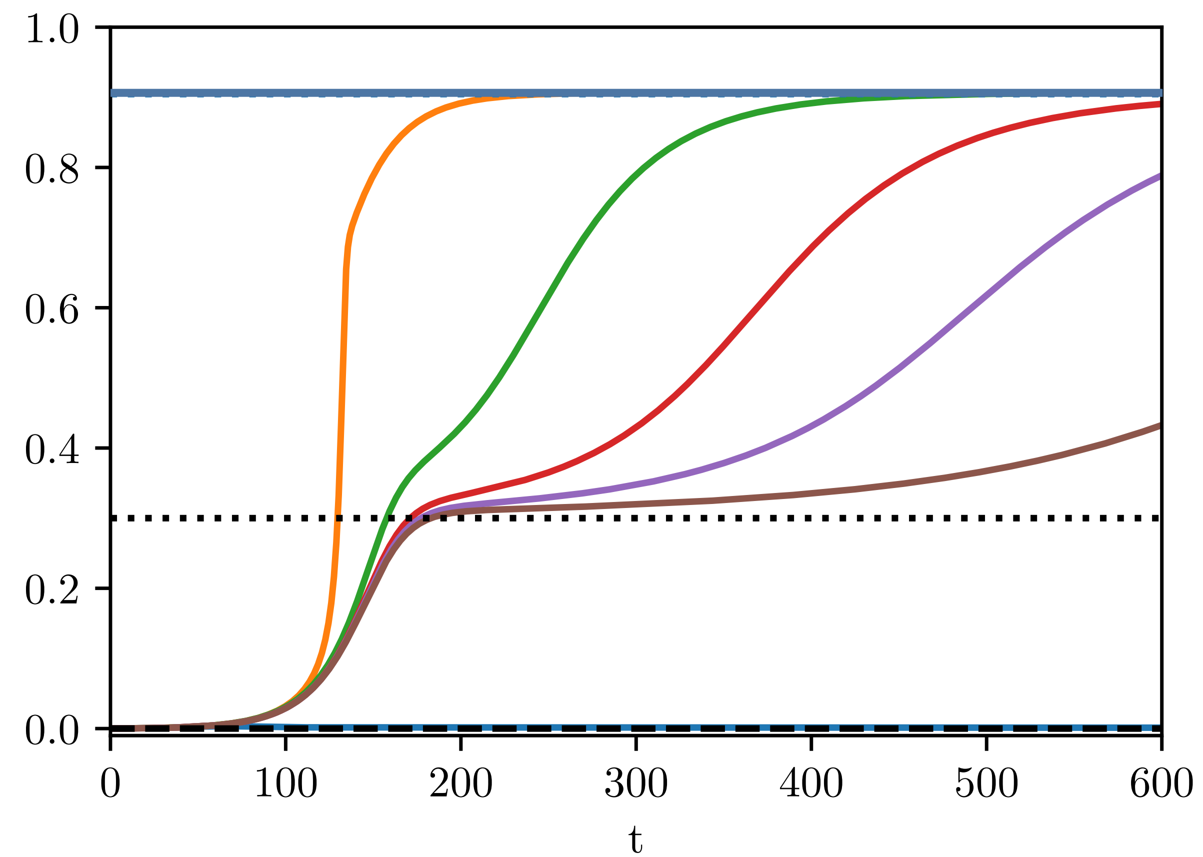


$$H = -\sum_{(i_1,\dots,i_p)} J_{i_1,\dots,i_p} x_{i_1} \dots x_{i_p} - rN \left(\sum_i \frac{x_i v_i}{N} \right)^k$$

$$H_{\text{tot}} = H_{p=2,k=2} + H_{p=3,k=3}$$

AMP much better than Langevin!

However Langevin would work more efficiently on $H_{p=2,k=2}$ only
… at least in the vicinity of the equator



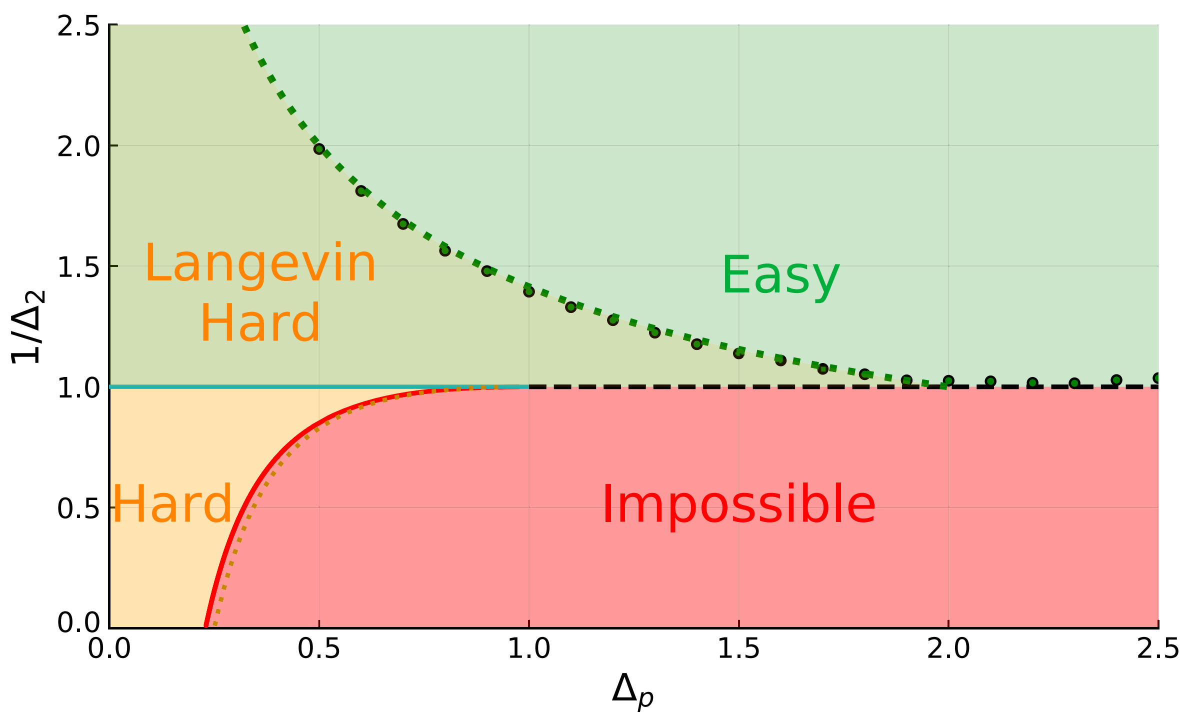
When less is better: AMP vs Langevin

Sarao, Biroli, Cammarota, Krzakala, Urbani, Zdeborova **PRX 2020**

$$T_{i,j} = W_{i,j} + v_i v_j$$

$$S_{k,l,m} = Z_{k,l,m} + v_k v_l v_m$$

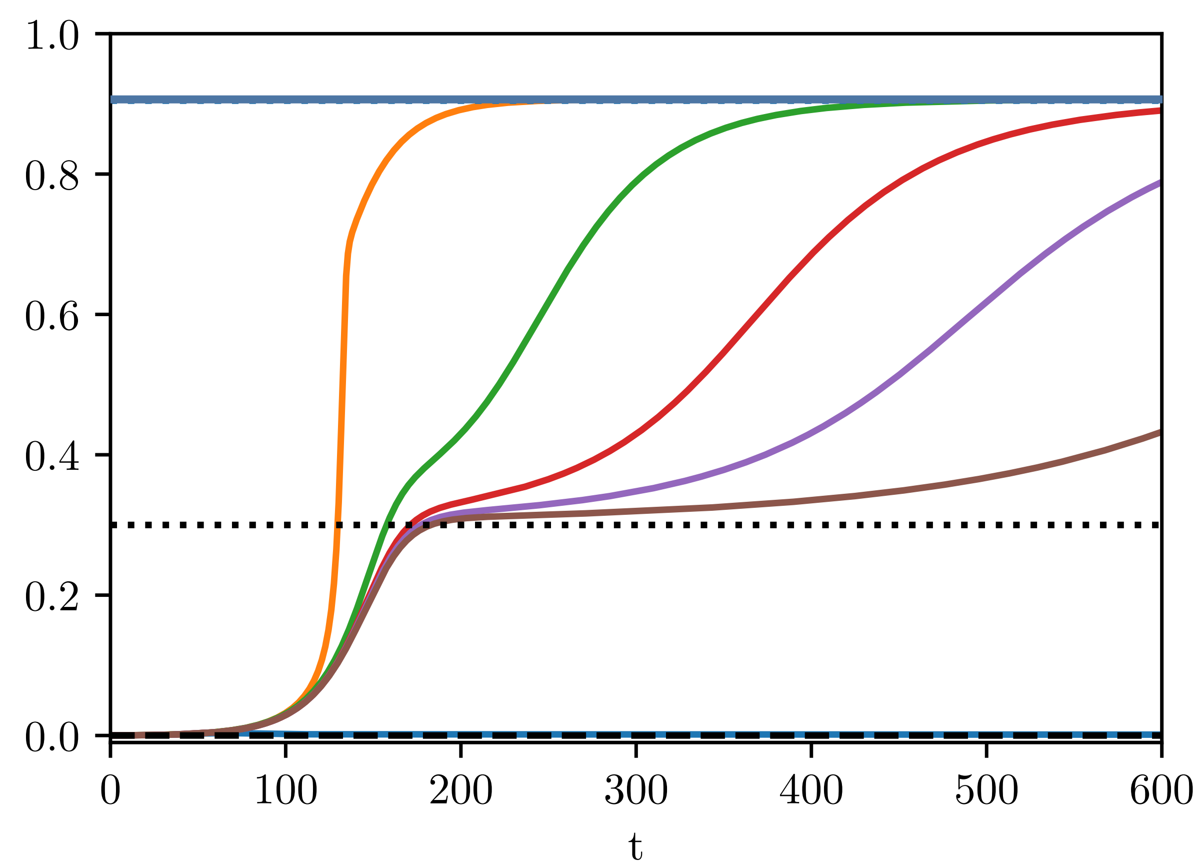
$$\langle W_{i,j}^2 \rangle = \Delta_W \quad \langle Z_{k,l,m}^2 \rangle = \Delta_Z$$



$$H = -\sum_{(i_1,\dots,i_p)} J_{i_1,\dots,i_p} x_{i_1}\dots x_{i_p} - rN\left(\sum_i \frac{x_i v_i}{N}\right)^k$$

$$H_{\text{tot}} = H_{p=2,k=2} + H_{p=3,k=3}$$

AMP much better than Langevin!



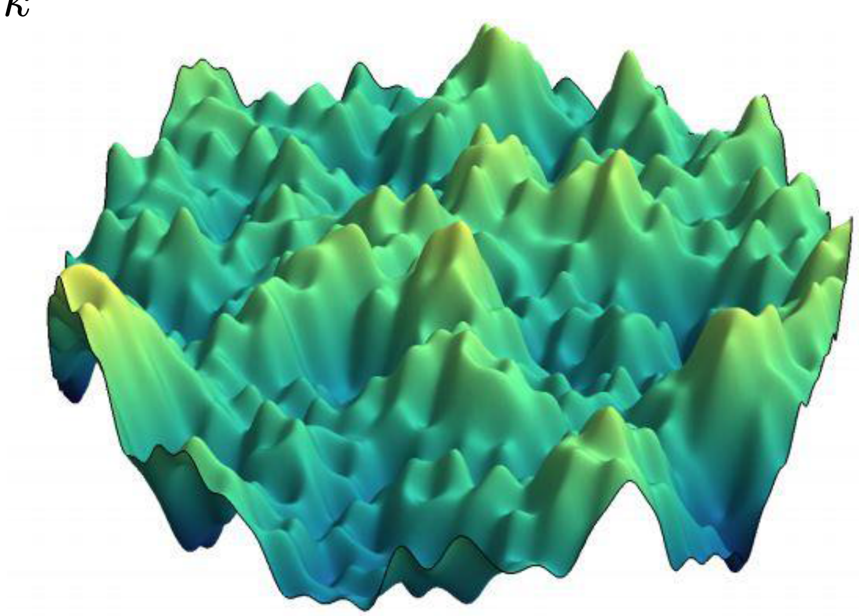
However Langevin would work more efficiently on $H_{p=2,k=2}$ only
… at least in the vicinity of the equator

Given problem / algorithm used, landscape info can help to chose the best strategy

Ironing the landscape

Biroli, Cammarota, Ricci-Tersenghi **J. Phys. A: Math. and Theor. 2020**

$$H = - \sum_{(i_1,\dots,i_k)} J_{i_1,\dots,i_k} x_{i_1} \dots x_{i_k} - rN \left(\sum_i \frac{x_i v_i}{N} \right)^k$$

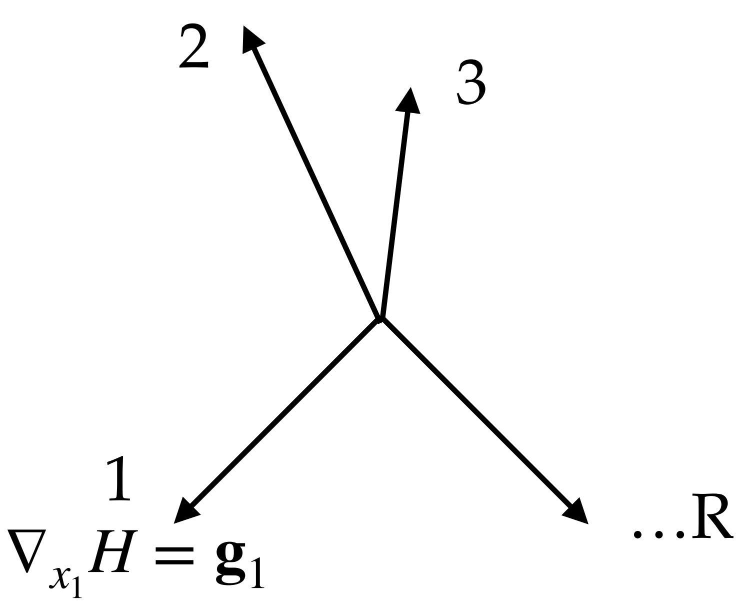


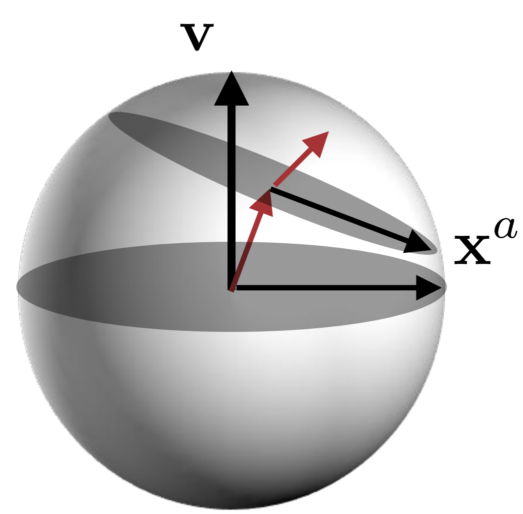
IT $\lambda_{IT} \sim O(1)$

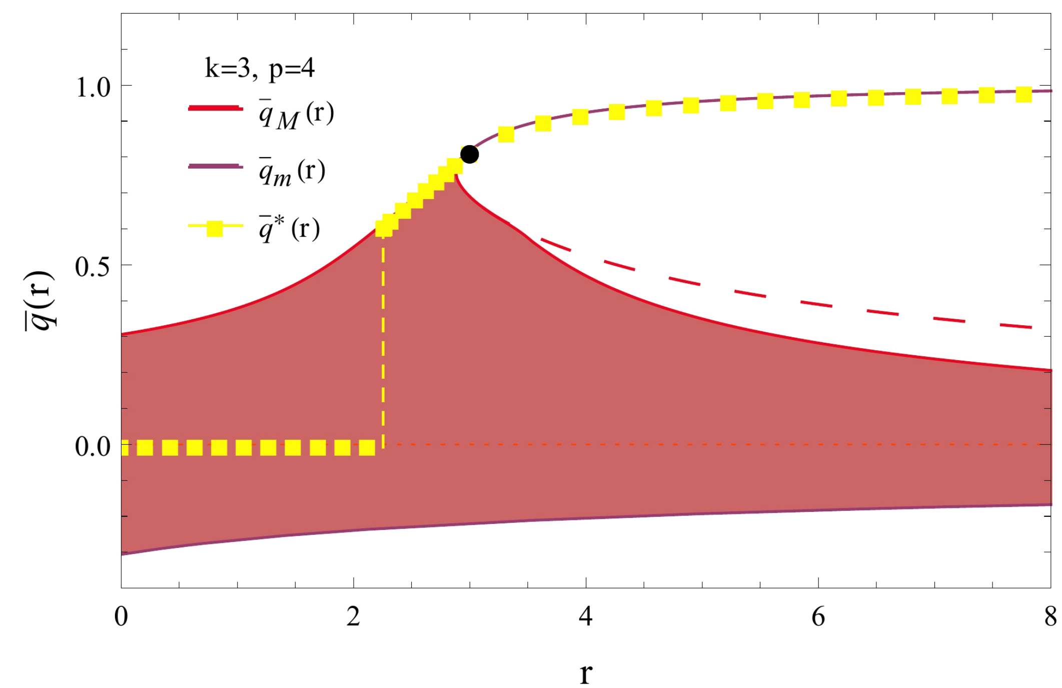
AMP, GD $\lambda_{AL} \sim N^{\frac{k-2}{2}}$

Tensor Unfolding, SOS $\lambda_{AL} \sim N^{\frac{k-2}{4}}$

Idea: sample the landscape on R points







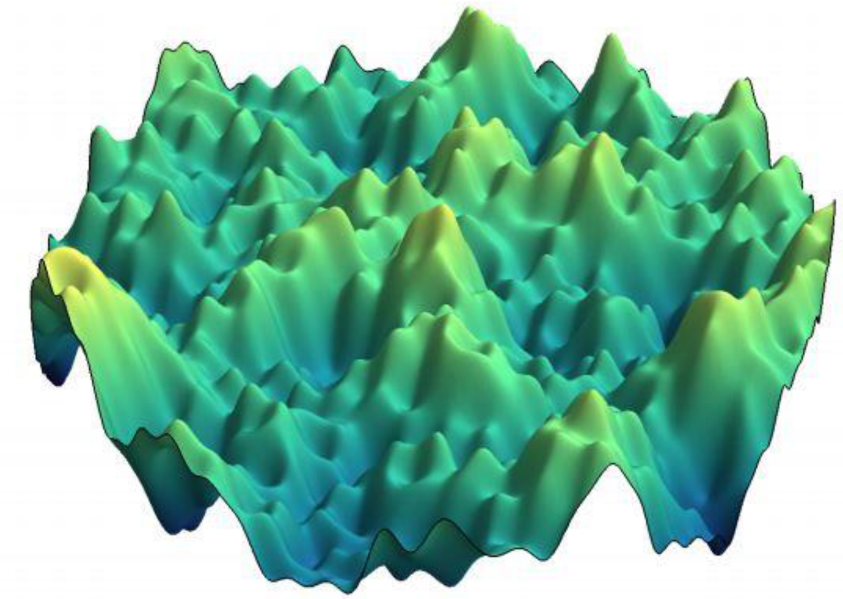
$$-\frac{x_{CM}(t+1) - x_{CM}(t)}{\eta} = \frac{1}{R}\sum_{a=1}^{R} \mathbf{g}_a = \frac{1}{R}\sum_{a=1}^{R} (r\mathbf{g_s} + \mathbf{g_n}_a) = r\mathbf{g_s} + \mathbf{g_n}_R$$

$$\mathbf{g_n}_R \sim \frac{\mathbf{g_n}_a}{\sqrt{R}}$$

Ironing the landscape

Biroli, Cammarota, Ricci-Tersenghi **J. Phys. A: Math. and Theor. 2020**

$$H = -\sum_{(i_1,\dots,i_k)} J_{i_1,\dots,i_k} x_{i_1} \dots x_{i_k} - rN \left(\sum_i \frac{x_i v_i}{N} \right)^k$$

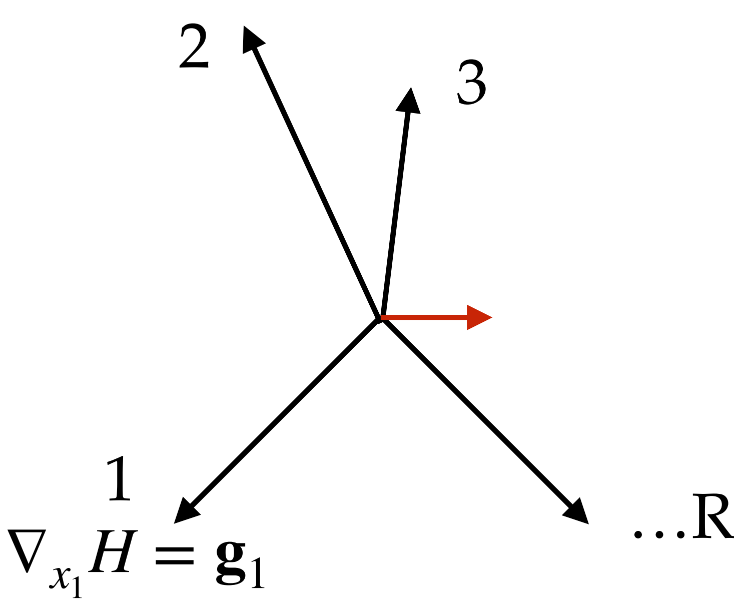


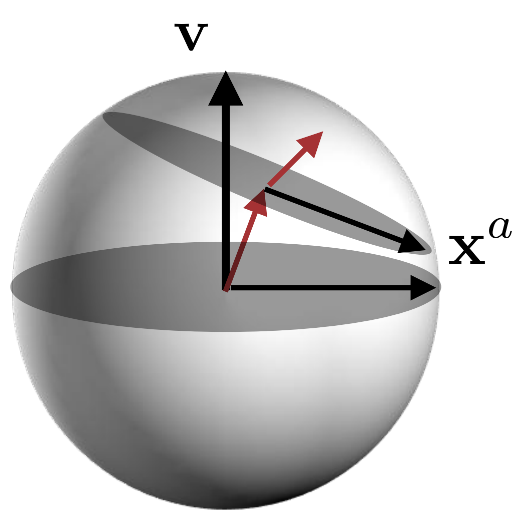
IT $\quad \lambda_{IT} \sim O(1)$

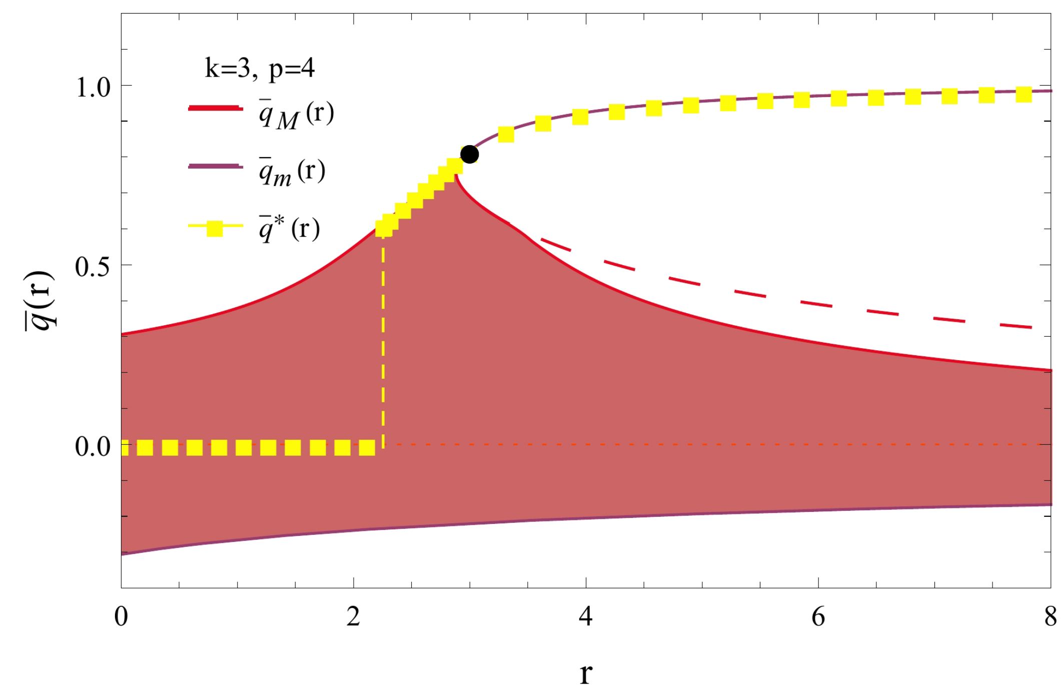
AMP, GD $\quad \lambda_{AL} \sim N^{\frac{k-2}{2}}$

Tensor Unfolding, SOS $\quad \lambda_{AL} \sim N^{\frac{k-2}{4}}$

Idea: sample the landscape on R points





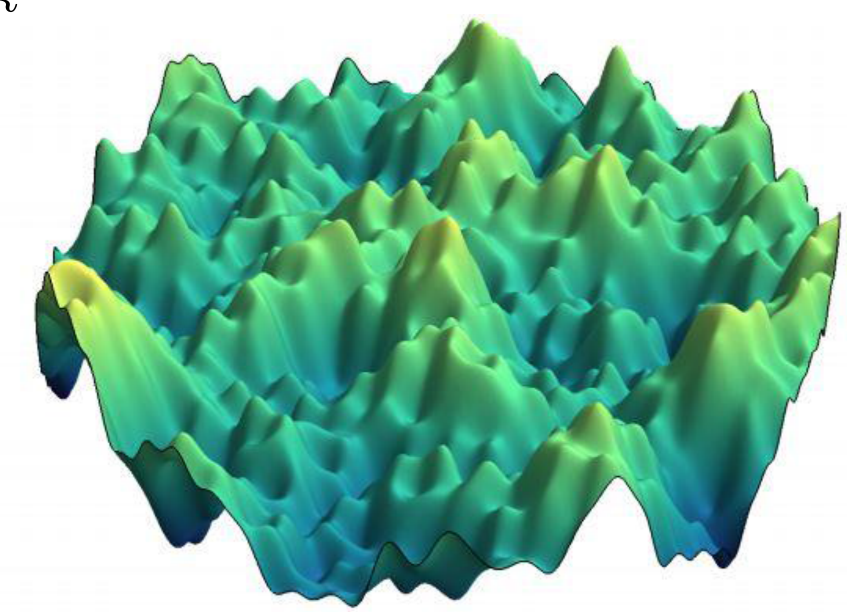


$$-\frac{x_{CM}(t+1) - x_{CM}(t)}{\eta} = \frac{1}{R}\sum_{a=1}^{R} \mathbf{g}_a = \frac{1}{R}\sum_{a=1}^{R} (r\mathbf{g_s} + \mathbf{g_n}_a) = r\mathbf{g_s} + \mathbf{g_n}_R \qquad \mathbf{g_n}_R \sim \frac{\mathbf{g_n}_a}{\sqrt{R}}$$

Ironing the landscape

Biroli, Cammarota, Ricci-Tersenghi **J. Phys. A: Math. and Theor. 2020**

$$H = -\sum_{(i_1,\ldots,i_k)} J_{i_1,\ldots,i_k} x_{i_1}\ldots x_{i_k} - rN\left(\sum_i \frac{x_i v_i}{N}\right)^k$$

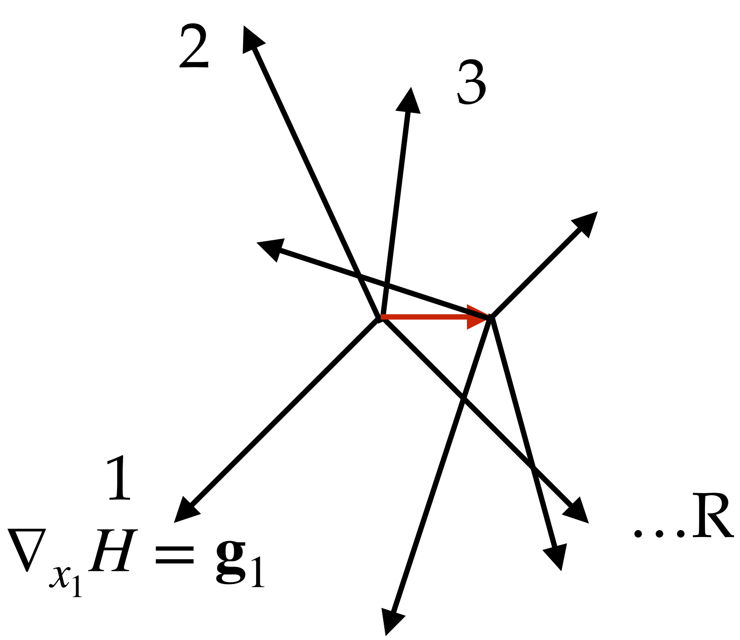


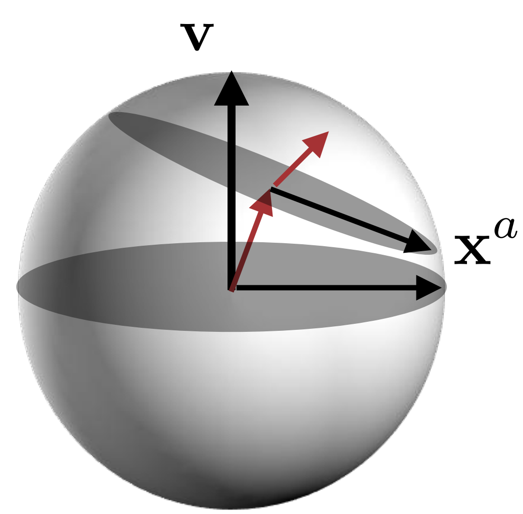
IT $\lambda_{IT} \sim O(1)$

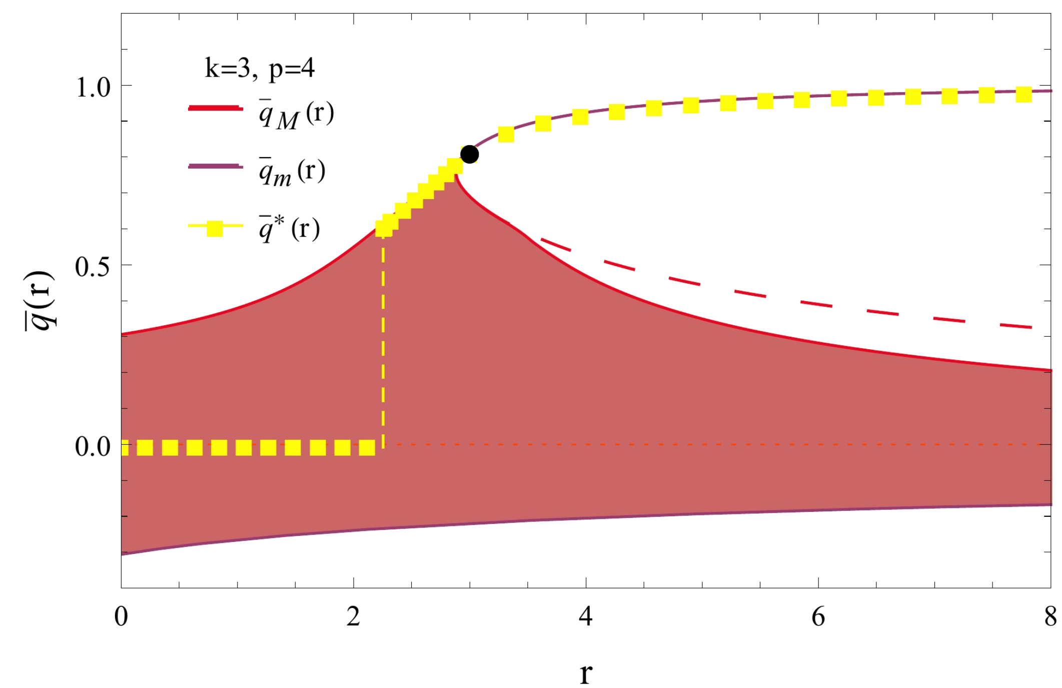
AMP, GD $\lambda_{AL} \sim N^{\frac{k-2}{2}}$

Tensor Unfolding, SOS $\lambda_{AL} \sim N^{\frac{k-2}{4}}$

Idea: sample the landscape on R points







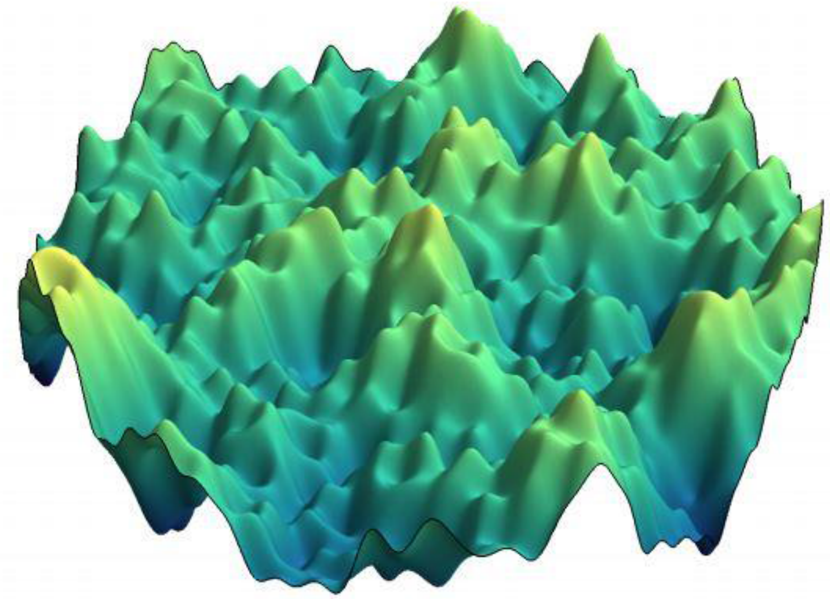
$$-\frac{x_{CM}(t+1) - x_{CM}(t)}{\eta} = \frac{1}{R}\sum_{a=1}^{R} \mathbf{g}_a = \frac{1}{R}\sum_{a=1}^{R}(r\mathbf{g_s} + \mathbf{g_n}_a) = r\mathbf{g_s} + \mathbf{g_n}_R$$

$$\mathbf{g_n}_R \sim \frac{\mathbf{g_n}_a}{\sqrt{R}}$$

Ironing the landscape

Biroli, Cammarota, Ricci-Tersenghi **J. Phys. A: Math. and Theor. 2020**

$$H = -\sum_{(i_1,\ldots,i_k)} J_{i_1,\ldots,i_k} x_{i_1} \ldots x_{i_k} - rN \left(\sum_i \frac{x_i v_i}{N} \right)^k$$

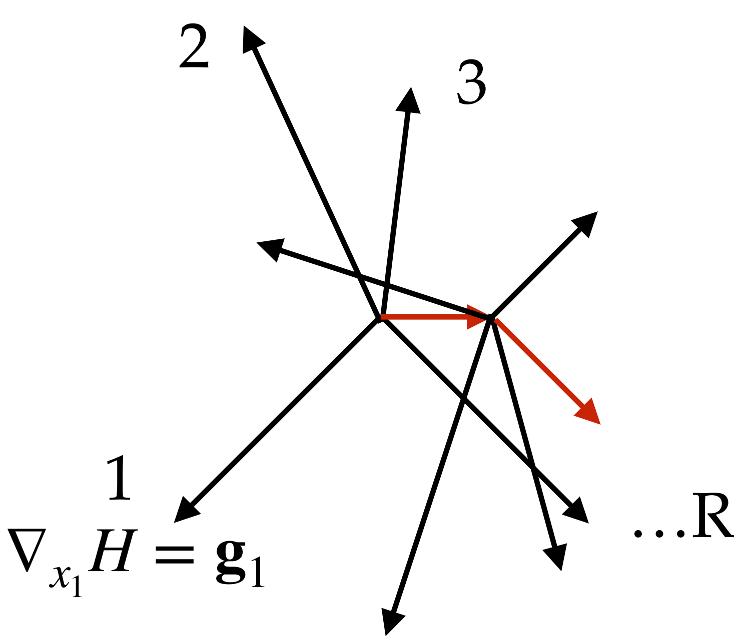


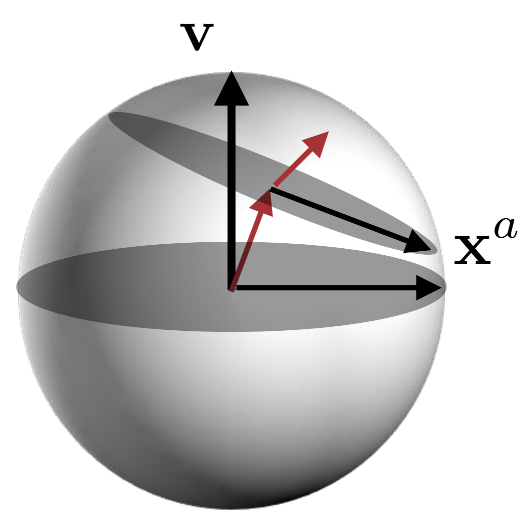
IT $\quad \lambda_{IT} \sim O(1)$

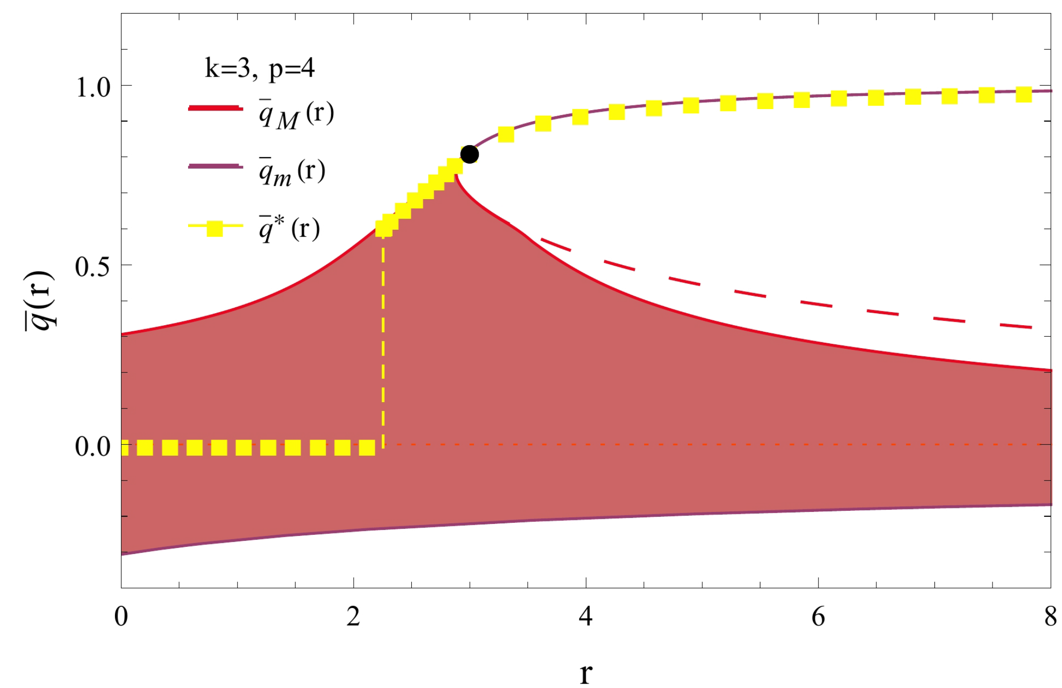
AMP, GD $\quad \lambda_{AL} \sim N^{\frac{k-2}{2}}$

Tensor Unfolding, SOS $\quad \lambda_{AL} \sim N^{\frac{k-2}{4}}$

Idea: sample the landscape on R points





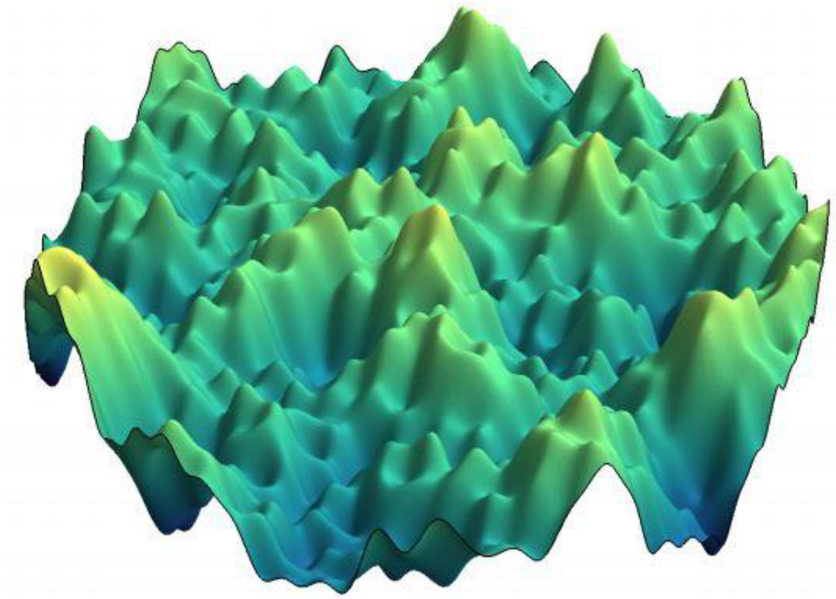


$$-\frac{x_{CM}(t+1) - x_{CM}(t)}{\eta} = \frac{1}{R}\sum_{a=1}^{R} \mathbf{g}_a = \frac{1}{R}\sum_{a=1}^{R} (r\mathbf{g_s} + \mathbf{g_n}_a) = r\mathbf{g_s} + \mathbf{g_n}_R \qquad \mathbf{g_n}_R \sim \frac{\mathbf{g_n}_a}{\sqrt{R}}$$

Ironing the landscape

Biroli, Cammarota, Ricci-Tersenghi **J. Phys. A: Math. and Theor. 2020**

$$H = -\sum_{(i_1,\ldots,i_k)} J_{i_1,\ldots,i_k} x_{i_1} \ldots x_{i_k} - rN\left(\sum_i \frac{x_i v_i}{N}\right)^k$$

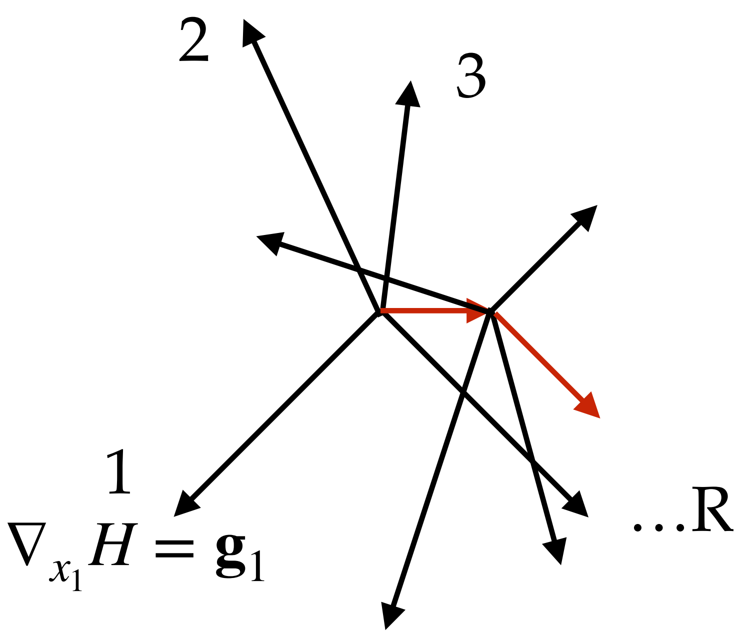


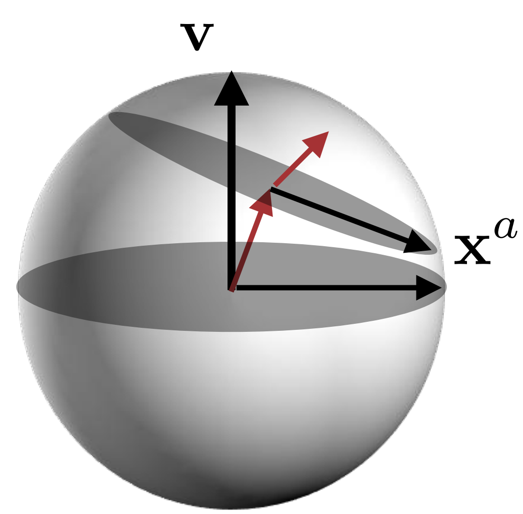
IT $\quad \lambda_{IT} \sim O(1)$

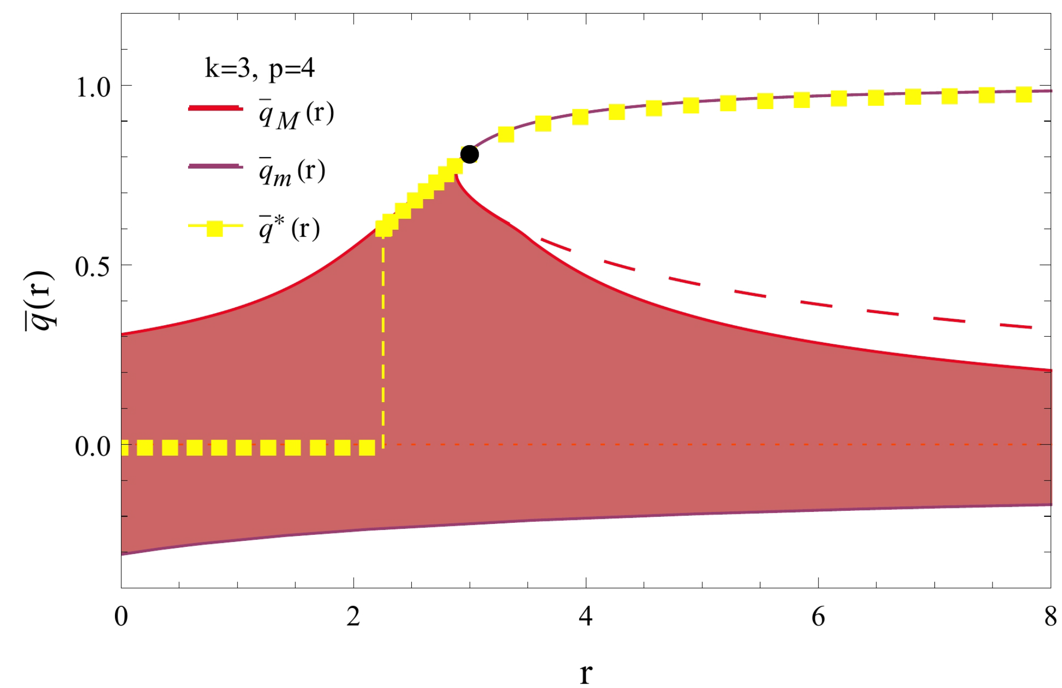
AMP, GD $\quad \lambda_{AL} \sim N^{\frac{k-2}{2}}$

Tensor Unfolding, SOS $\quad \lambda_{AL} \sim N^{\frac{k-2}{4}}$

Idea: sample the landscape on R points





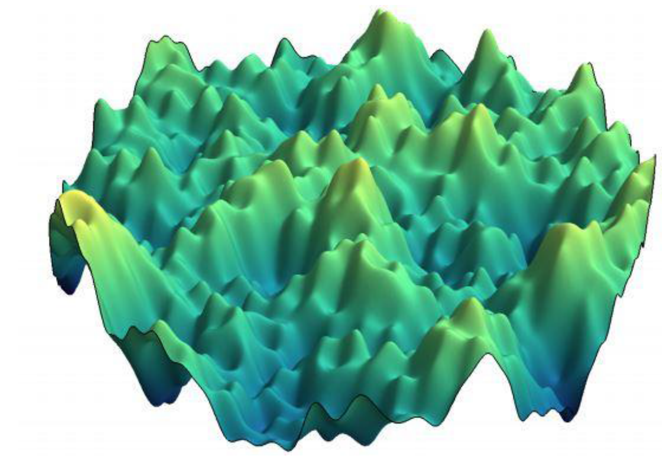


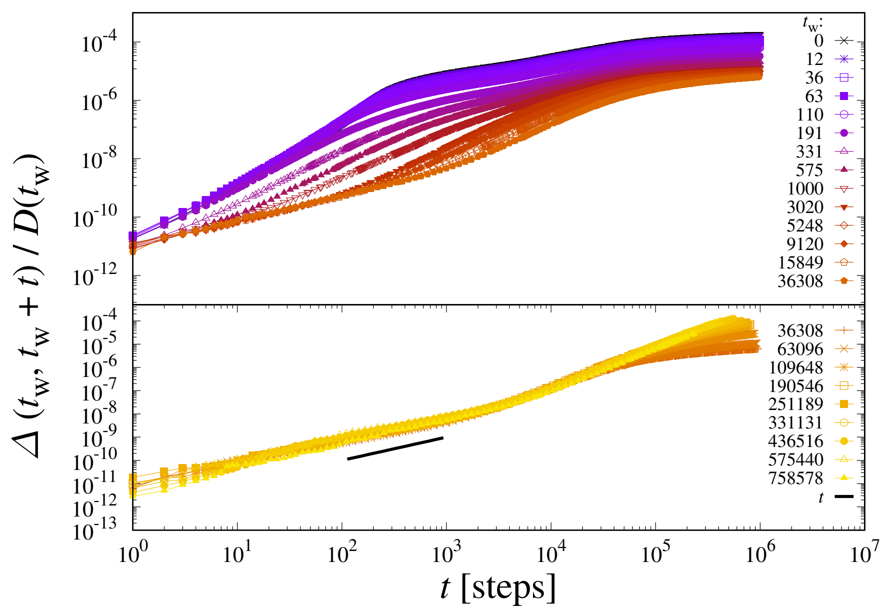
$$-\frac{x_{CM}(t+1) - x_{CM}(t)}{\eta} = \frac{1}{R}\sum_{a=1}^{R} \mathbf{g}_a = \frac{1}{R}\sum_{a=1}^{R}(r\mathbf{g_s} + \mathbf{g_n}_a) = r\mathbf{g_s} + \mathbf{g_n}_R \qquad \mathbf{g_n}_R \sim \frac{\mathbf{g_n}_a}{\sqrt{R}}$$

Averaged landscape descent: $\quad \lambda_{AL} \sim N^{\frac{k-2}{4}}$ $\quad$ as good as it can get!

Learning dynamics in rough landscapes

Learning as rough loss/risk/cost landscapes exploration

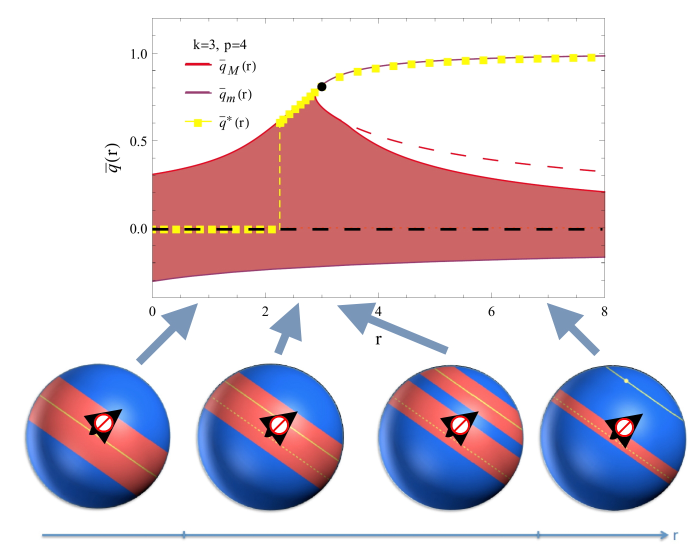




Machine Learning as interrupted aging (slowed down by glassy landscape) and diffusion

Tensor PCA: detailed information on landscape structure and accurate prediction of algorithmic transition

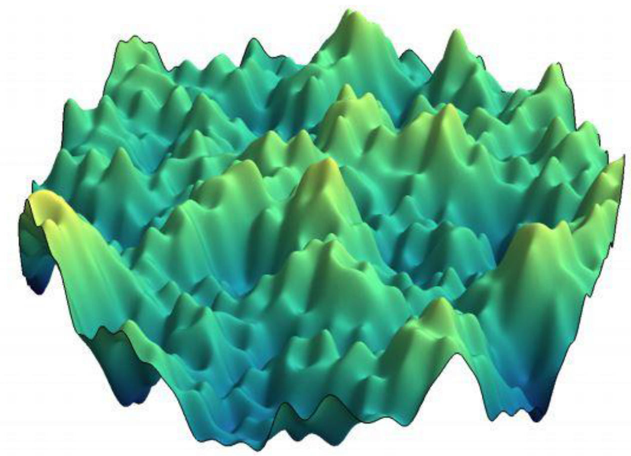
Tensor PCA: two strategies (one is very general!) to optimise GD

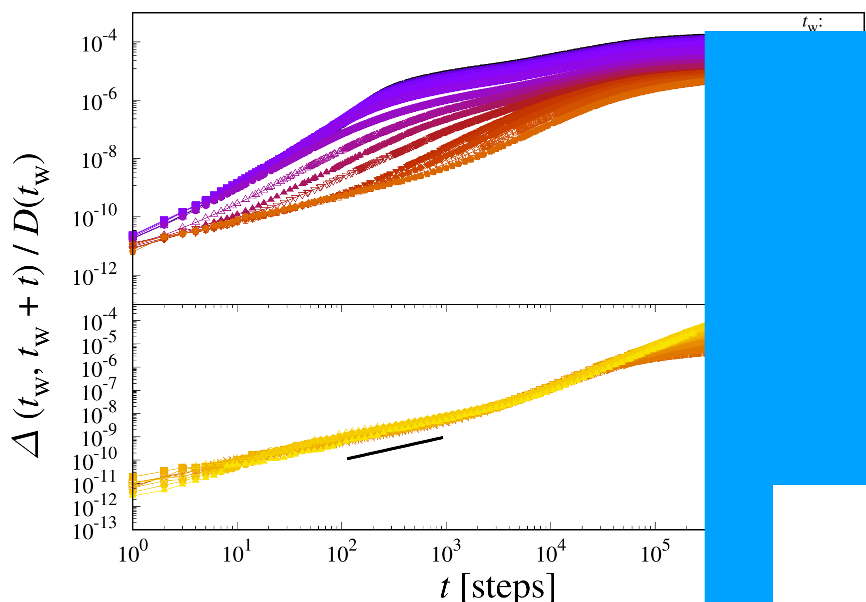


To which extent are these concepts general (e.g. phase retrieval) and/or applicable to ML?
Can we reduce overparametrization, dataset's size, propose more efficient versions of SGD?

Learning dynamics in rough landscapes

Learning as rough loss/risk/cost landscapes exploration



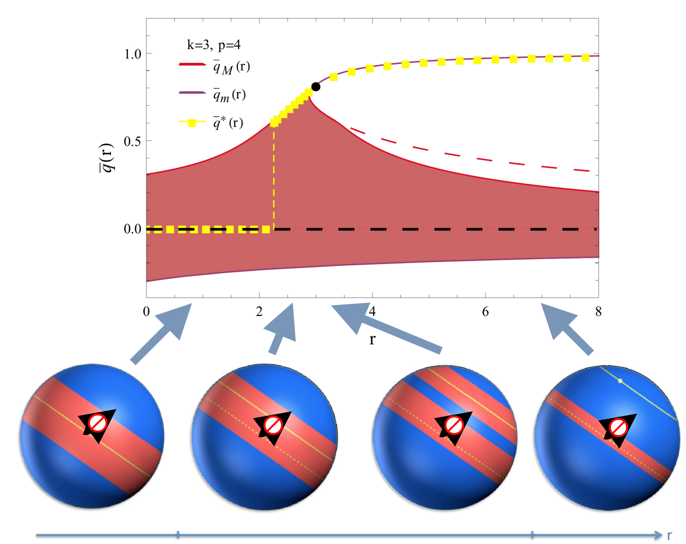


upted aging (slowed down
y landscape) and diffusion

PROGETTI DI RICERCA DI
RILEVANTE INTERESSE NAZIONALE

FUTURE AI RESEARCH

Tensor PCA: detailed infor
accurate prediction of algo

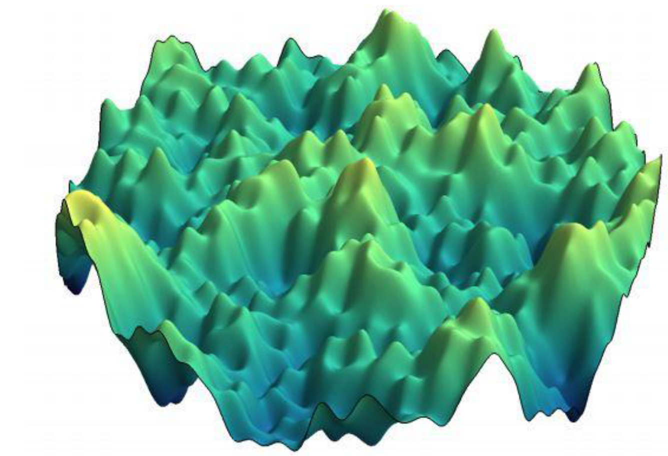


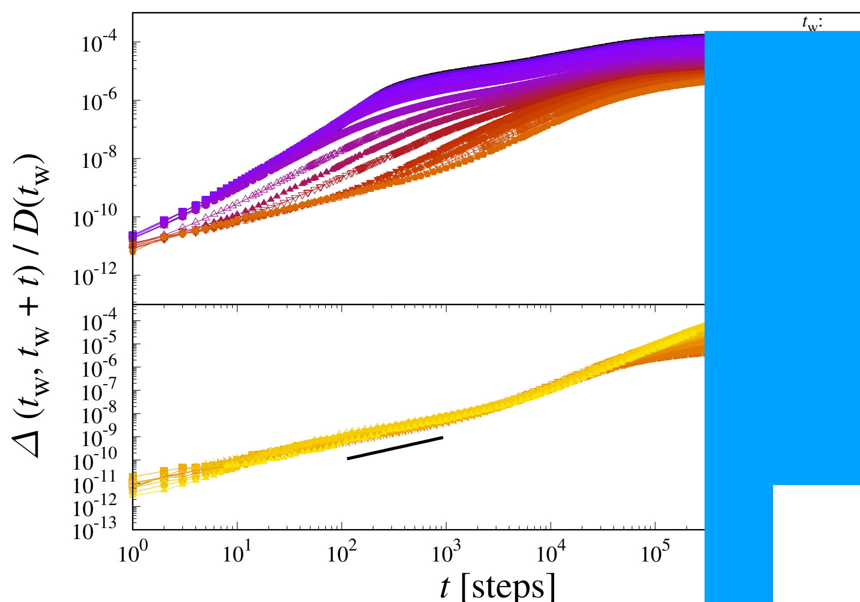
Tensor PCA: two strategies (one is very general!) to optimise GD

To which extent are these concepts general (e.g. phase retrieval) and/or applicable to ML?
Can we reduce overparametrization, dataset's size, propose more efficient versions of SGD?

Learning dynamics in rough landscapes

Learning as rough loss/risk/cost landscapes exploration

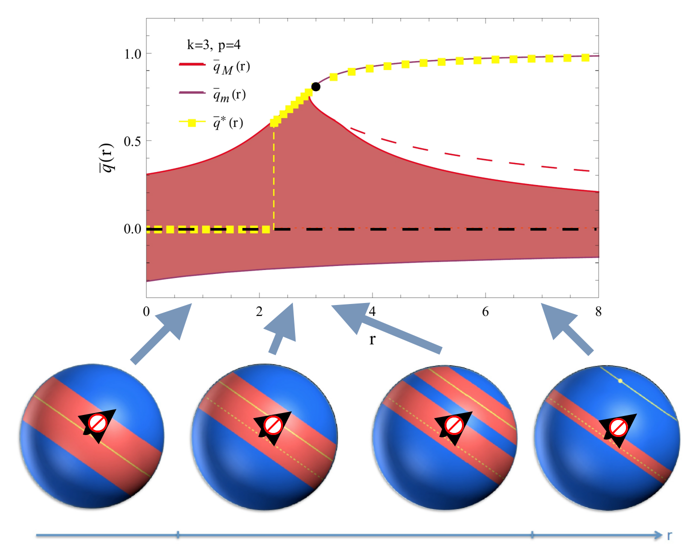




...upted aging (slowed down
...y landscape) and diffusion

Thank you!

Tensor PCA: detailed infor...
accurate prediction of algo...



Tensor PCA: two strategies (one is very general!) to optimise GD

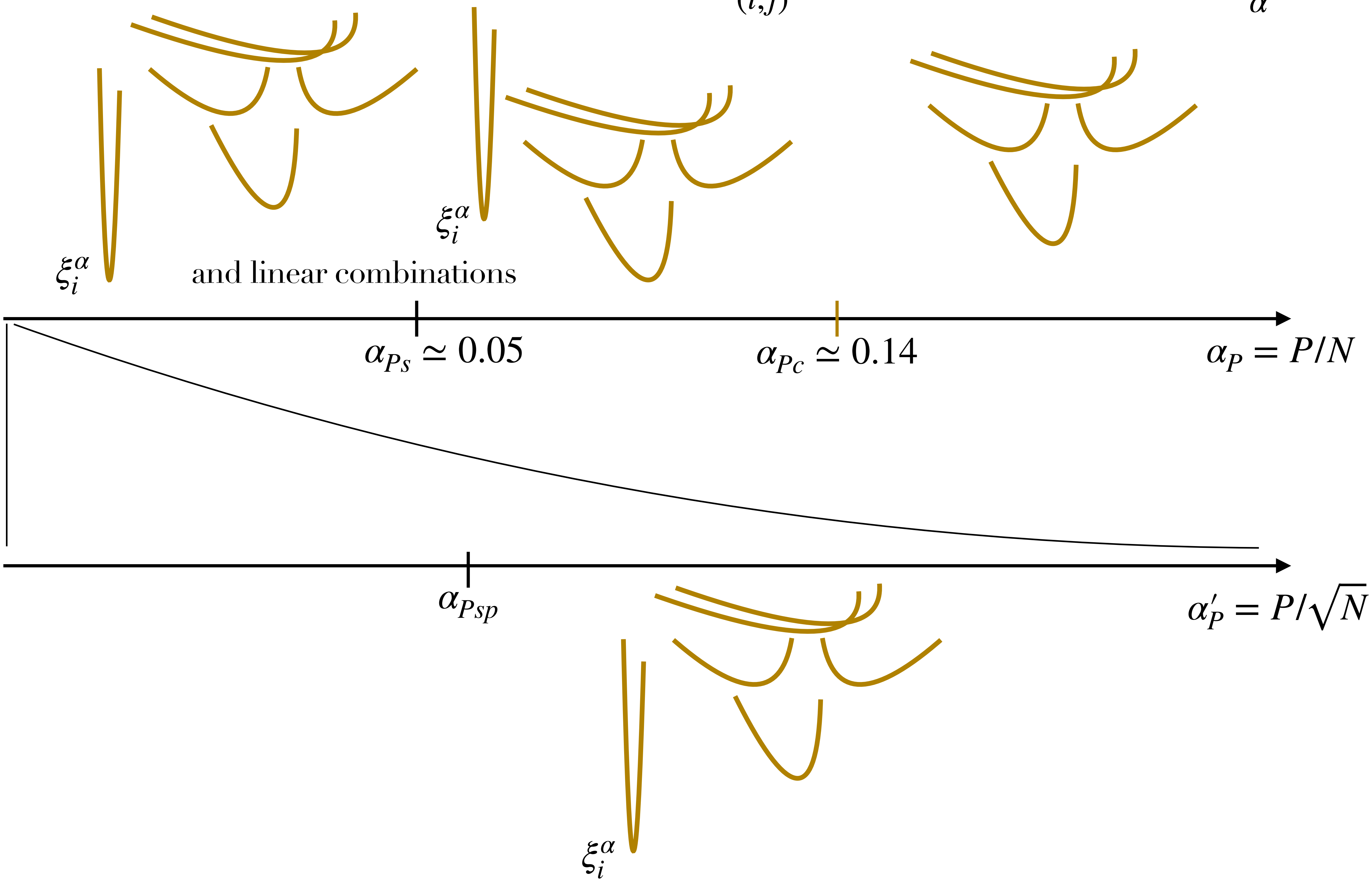
To which extent are these concepts general (e.g. phase retrieval) and/or applicable to ML?
Can we reduce overparametrization, dataset's size, propose more efficient versions of SGD?

Dynamics, data structure…and Hopfield

Consider the Hopfield model $H = -\sum_{(i,j)}^{N} J_{ij} s_i s_j$ $J_{ij} = \frac{1}{N} \sum_{\alpha}^{P} \xi_i^\alpha \xi_j^\alpha$

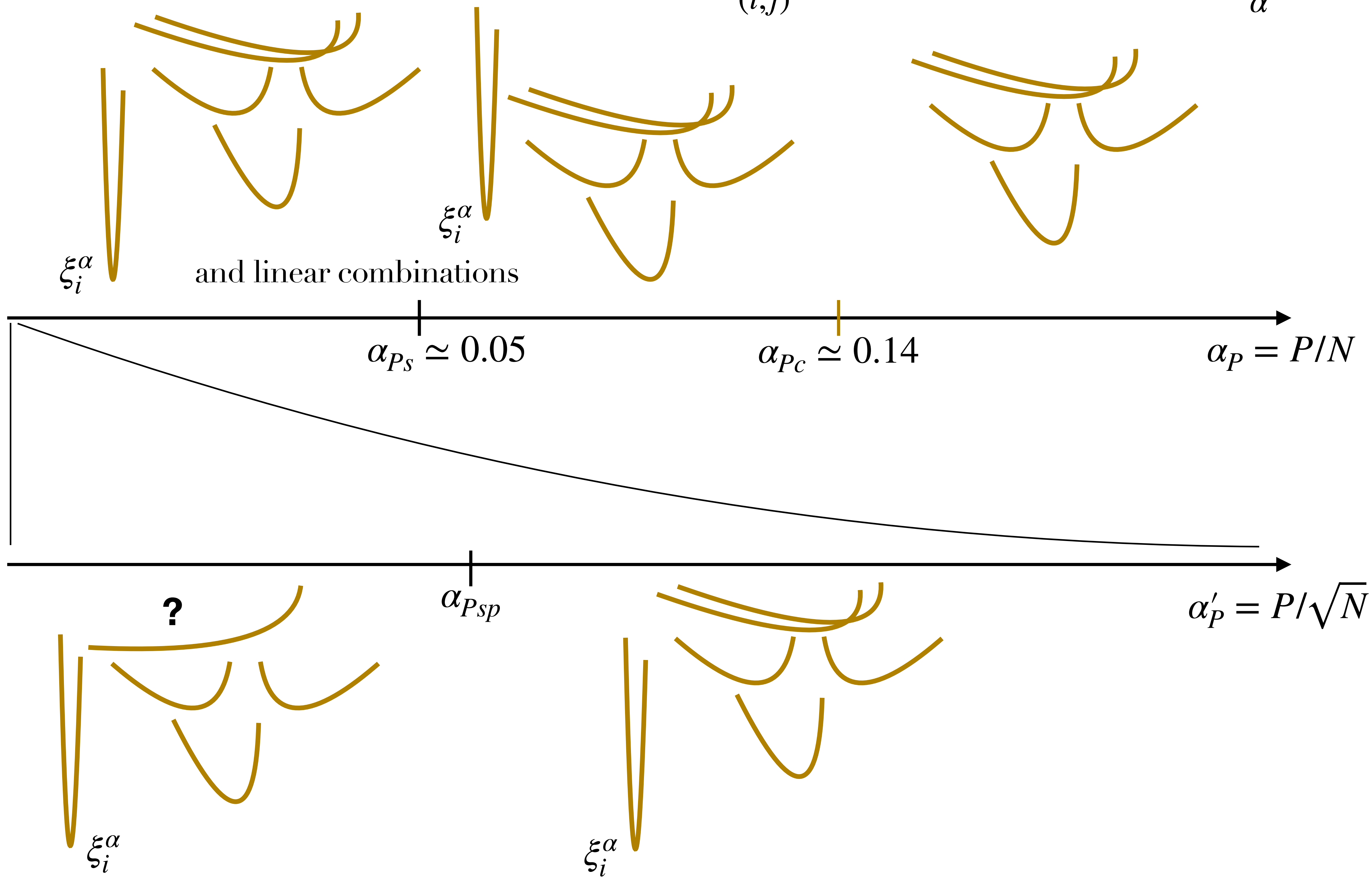
Dynamics …and Hopfield

Consider the Hopfield model $H = -\sum_{(i,j)}^{N} J_{ij} s_i s_j$ $J_{ij} = \frac{1}{N} \sum_{\alpha}^{P} \xi_i^\alpha \xi_j^\alpha$



Dynamics …and Hopfield

Consider the Hopfield model $H = -\sum_{(i,j)}^{N} J_{ij} s_i s_j$ $J_{ij} = \frac{1}{N} \sum_{\alpha}^{P} \xi_i^\alpha \xi_j^\alpha$

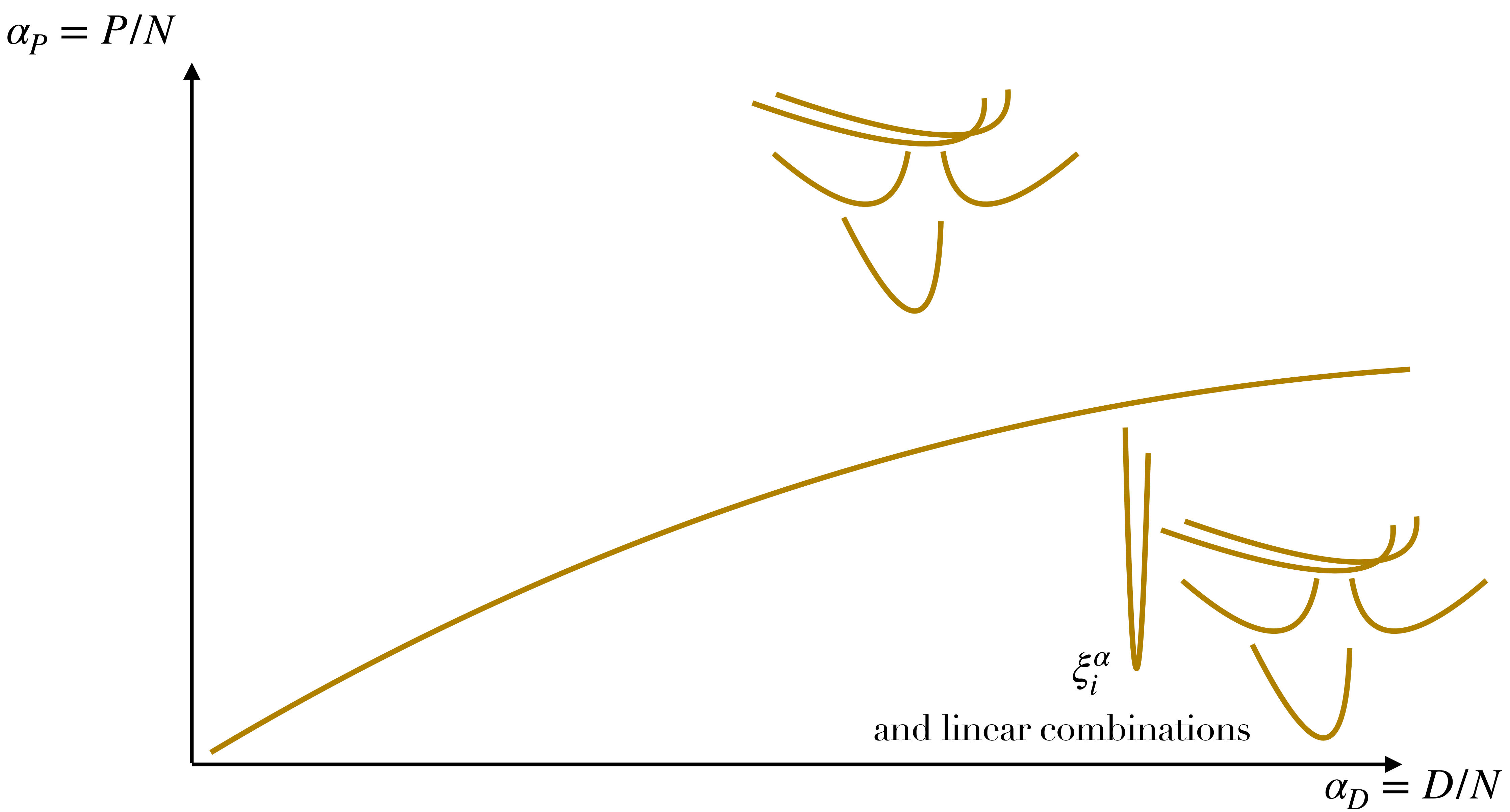


Dynamics, data structure…and Hopfield

Negri Lauditi Perugini Lucibello Malatesta arXiv:2303.16880 (2023)

Consider the Hopfield model $H = -\sum_{(i,j)}^{N} J_{ij} s_i s_j$ $\qquad J_{ij} = \frac{1}{N} \sum_{\alpha}^{P} \xi_i^\alpha \xi_j^\alpha$

Add correlation $\xi_i^\alpha = \mathrm{sign}\left(\sum_{\mathrm{k}}^{\mathrm{D}} \mathrm{c}_{\mathrm{k}}^{\alpha} \mathrm{f}_{\mathrm{i}}^{\mathrm{k}} \right)$

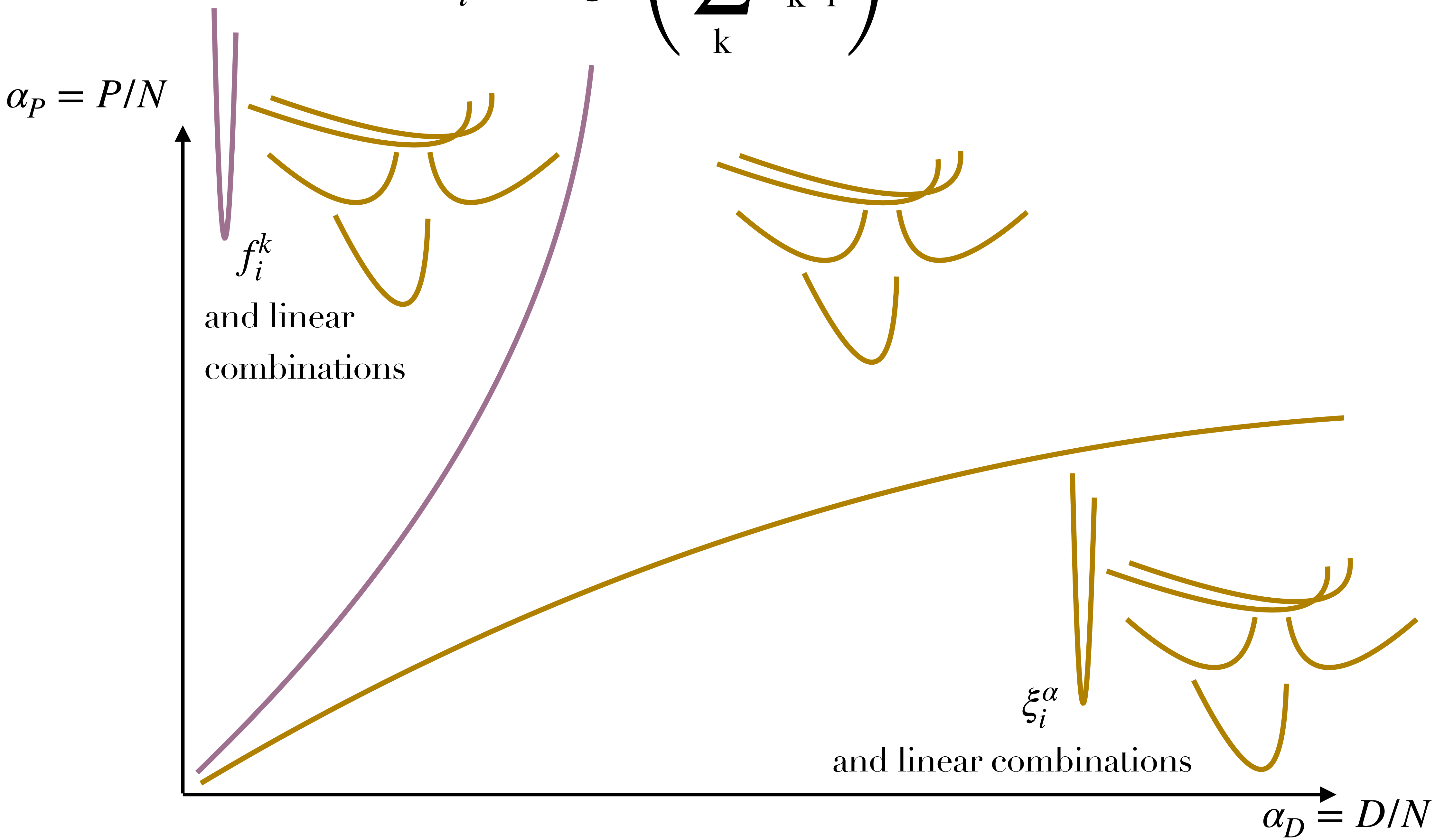


Dynamics, data structure…and Hopfield

Negri Lauditi Perugini Lucibello Malatesta arXiv:2303.16880 (2023)

Consider the Hopfield model $H = -\sum_{(i,j)}^{N} J_{ij} s_i s_j$ $\quad J_{ij} = \frac{1}{N} \sum_{\alpha}^{P} \xi_i^\alpha \xi_j^\alpha$

Add correlation $\xi_i^\alpha = \mathrm{sign}\left(\sum_{\mathrm{k}}^{\mathrm{D}} \mathrm{c}_{\mathrm{k}}^{\alpha} \mathrm{f}_{\mathrm{i}}^{\mathrm{k}} \right)$

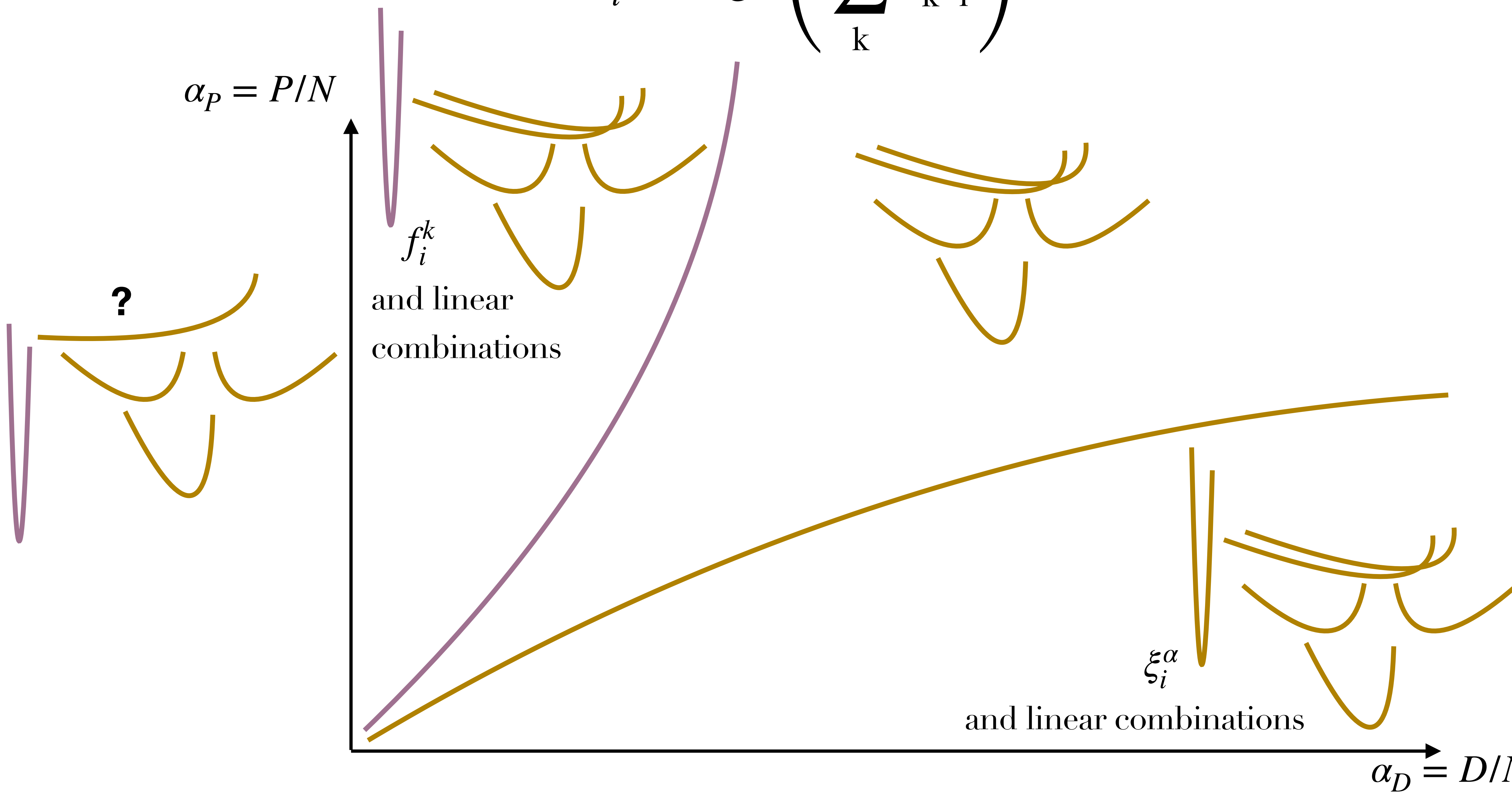


Dynamics, data structure…and Hopfield

Negri Lauditi Perugini Lucibello Malatesta arXiv:2303.16880 (2023)

Consider the Hopfield model $$H = -\sum_{(i,j)}^{N} J_{ij} s_i s_j \qquad J_{ij} = \frac{1}{N}\sum_{\alpha}^{P} \xi_i^\alpha \xi_j^\alpha$$

Add correlation $$\xi_i^\alpha = \mathrm{sign}\left(\sum_{\mathrm{k}}^{\mathrm{D}} \mathrm{c}_{\mathrm{k}}^{\alpha} \mathrm{f}_{\mathrm{i}}^{\mathrm{k}}\right)$$

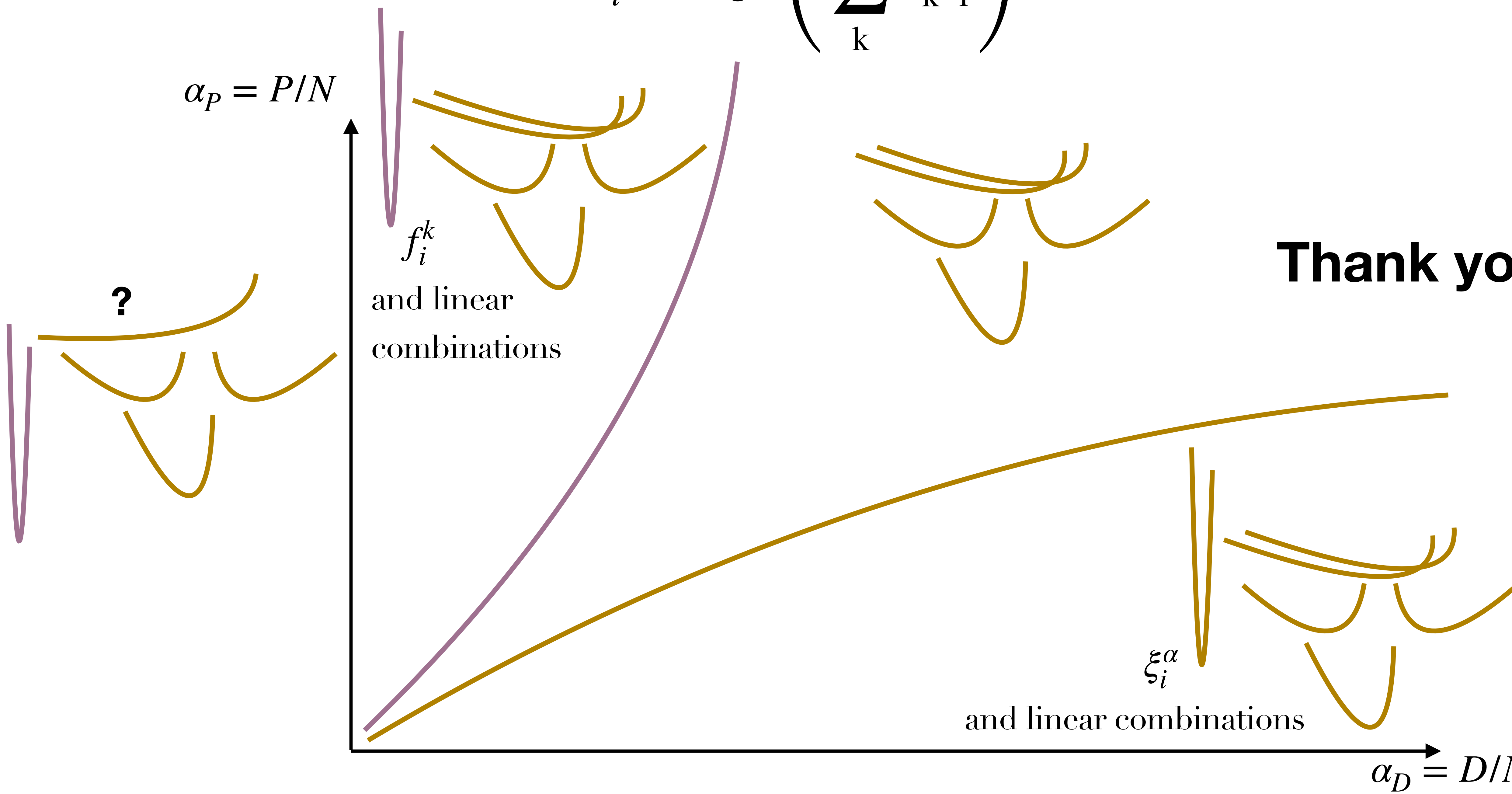


Dynamics, data structure…and Hopfield

Negri Lauditi Perugini Lucibello Malatesta arXiv:2303.16880 (2023)

Consider the Hopfield model $H = -\sum_{(i,j)}^{N} J_{ij} s_i s_j$ $J_{ij} = \frac{1}{N}\sum_{\alpha}^{P} \xi_i^\alpha \xi_j^\alpha$

Add correlation $\xi_i^\alpha = \mathrm{sign}\left(\sum_{\mathrm{k}}^{\mathrm{D}} \mathrm{c}_{\mathrm{k}}^\alpha \mathrm{f}_{\mathrm{i}}^{\mathrm{k}}\right)$



Thank you!