High-dimensional cost landscape and gradient descent in Tensor PCA and its generalisations

Chiara Cammarota
King's College London

- *Ros, Ben Arous, Biroli, Cammarota* **PRX (2019)**
- *Sarao, Biroli, Cammarota, Krzakala, Urbani, Zdeborova* **arXiv:1812.09066 (2018)**
- *Sarao, Biroli, Cammarota, Krzakala, Zdeborova* **Spotlight at NIPS (2019)**
- *Biroli, Cammarota, Ricci-Tersenghi* **arXiv:1905.12294 (2019)**

19.11.2019 IPAM UCLA Using Physical Insights for Machine Learning

High-dimensional non convex optimisation

Gradient Descent and its variants are valuable workhorses

e.g. Stochastic Gradient Descent for Machine Learning

$\mathcal{L}(\mathbf{x})$ Risk, Loss, Likelihood..

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) - \eta \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x})|_{\mathbf{x}(t)} + \mathrm{d}\xi$$

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

High-dimensional non convex optimisation

Gradient Descent and its variants are valuable workhorses

e.g. Stochastic Gradient Descent for Machine Learning

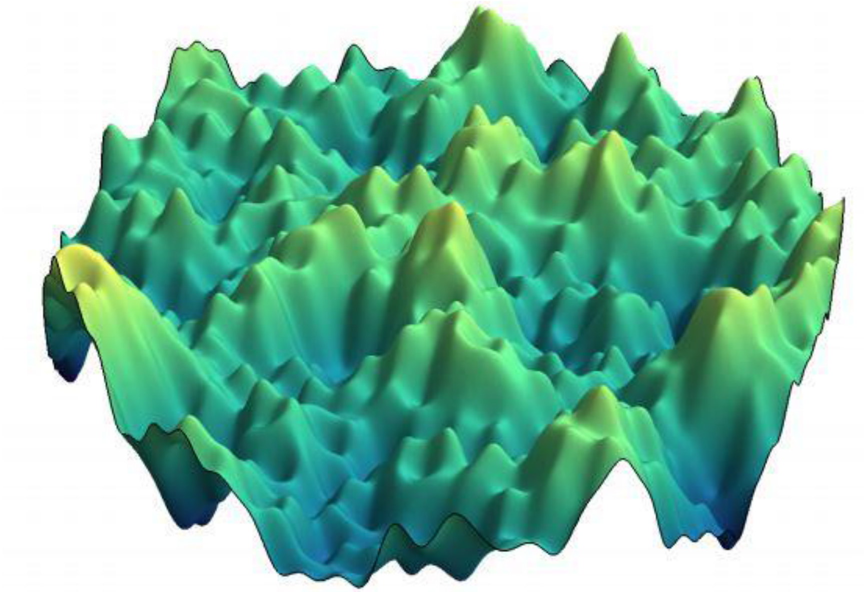
$\mathcal{L}(\mathbf{x})$ Risk, Loss, Likelihood..

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) - \eta \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x})|_{\mathbf{x}(t)} + \mathrm{d}\xi$$

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

Expected failure of GD for rough Risks with well-defined hard phase

planted cliques,
compressed sensing,
phase retrieval,
tensor decomposition/
factorisation/
completion



High-dimensional non convex optimisation

Gradient Descent and its variants are valuable workhorses

e.g. Stochastic Gradient Descent for Machine Learning

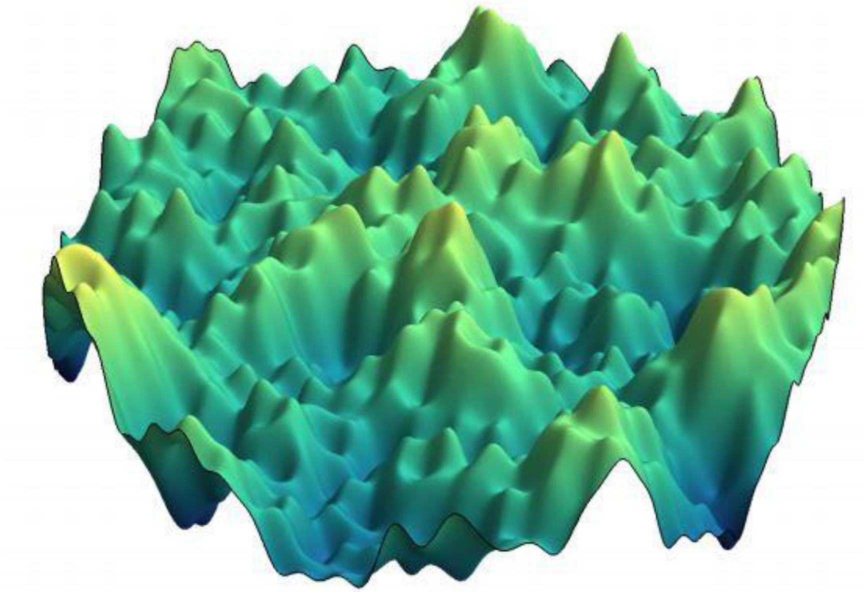
$\mathcal{L}(\mathbf{x})$ Risk, Loss, Likelihood..

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) - \eta \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x})|_{\mathbf{x}(t)} + \mathrm{d}\xi$$

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

Expected failure of GD for rough Risks with well-defined hard phase

planted cliques,
compressed sensing,
phase retrieval,
tensor decomposition/
factorisation/
completion



The plan

reveal details of GD/landscape connection
improve on GD (can it be versatile and competitive?)
trace origin of well-defined hard phase

Tensor PCA

Estimation of rank-one tensor from a noisy channel

Richard, Montanari 2014
Lesieur, Miolane, et al 2017
Perry, Wein, Bandeira 2016
Ben Arous, Mei et al 2017

Observation Corrupting noise Signal

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

$$\langle W^2_{i_1,\ldots,i_k}\rangle = \Delta$$

Bayesian approach $$P(\mathbf{x}|\mathbf{T}) \propto \prod_i e^{-\frac{x_i^2}{2}} \prod_{(i_1,\ldots,i_k)} e^{-\frac{1}{2\Delta}\left(T_{i_1,\ldots,i_k} - x_{i_1}\ldots x_{i_k}\right)^2}$$

$$\propto e^{-\beta H - \lambda\sum_i x_i^2}$$

Alike spin-glass model $$H = -\underbrace{\sum_{(i_1,\ldots,i_k)} J_{i_1,\ldots,i_k} x_{i_1}\ldots x_{i_k}}_{\text{p-spin model}} \underbrace{- rN\left(\sum_i \frac{x_i v_i}{N}\right)^k}_{\text{field}}$$

with $J_{i_1,\ldots,i_k} \propto W_{i_1,\ldots,i_k}$ such that $\langle J^2_{i_1,\ldots,i_k}\rangle = \dfrac{k!}{2N^{k-1}}$

Maximum likelihood estimate $$\mathbf{x}^* = \underset{x \text{ s.t. } ||x||_2^2 = N}{\operatorname{argmax}} \sum_{(i_1,\ldots,i_k)} T_{i_1,\ldots,i_k} x_{i_1}\ldots x_{i_k}$$

Tensor PCA and generalisations

Generalised Tensor PCA: $$H_{\overline{\mathrm{p,k}}} = -\sum_{(i_1,\dots,i_p)} J_{i_1,\dots,i_p} x_{i_1} \dots x_{i_p} - rN \left(\sum_i \frac{x_i v_i}{N} \right)^k$$

Mixed Matrix-Tensor PCA: $$H_{\text{tot}} = H_{p=2,k=2} + H_{p=3,k=3}$$

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

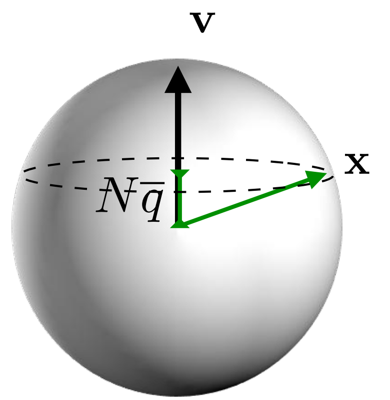
$$S_{k,l,m} = Z_{k,l,m} + v_k v_l v_m$$

r signal to noise ratio

$\mathbf{x}$ vector on a sphere

Distance from signal, latitude on the sphere

$$\overline{q} = \frac{1}{N} \sum_i x_i v_i$$



Ben Arous, Geissari, Jagannath (2018)

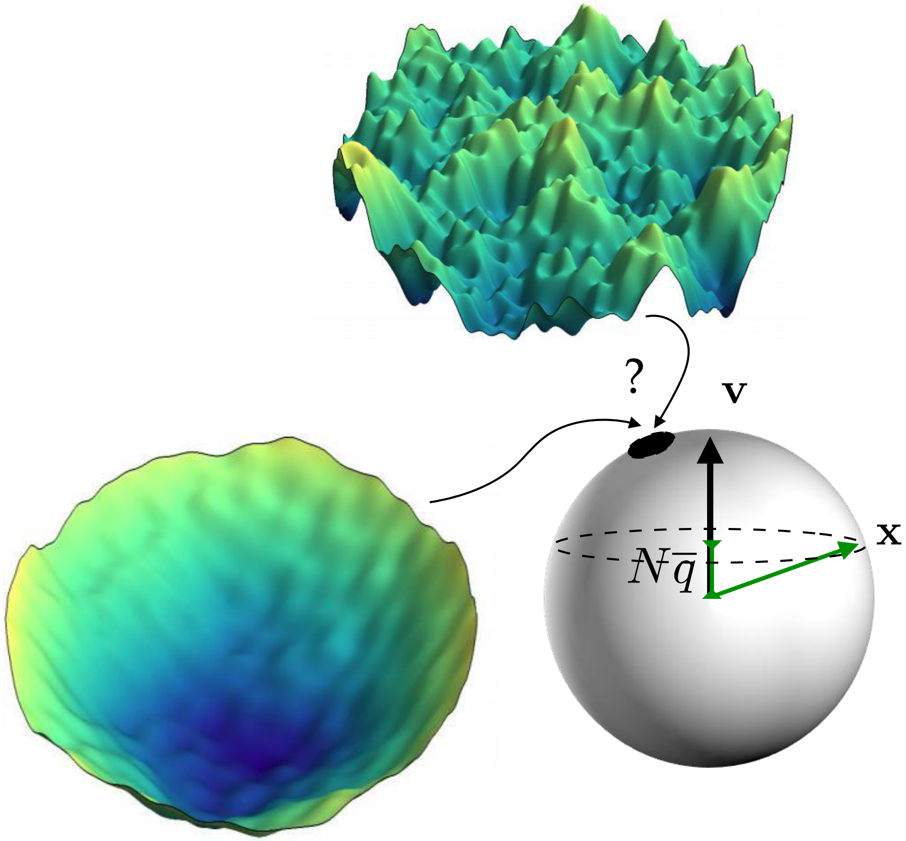
Tensor PCA and generalisations

Generalised Tensor PCA: $$H_{\mathrm{p,k}} = -\sum_{(i_1,\dots,i_p)} J_{i_1,\dots,i_p} x_{i_1} \dots x_{i_p} - rN \left(\sum_i \frac{x_i v_i}{N} \right)^k$$

Mixed Matrix-Tensor PCA: $$H_{\mathrm{tot}} = H_{p=2,k=2} + H_{p=3,k=3}$$

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

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



r signal to noise ratio

$\mathbf{x}$ vector on a sphere

Distance from signal, latitude on the sphere

$$\overline{q} = \frac{1}{N} \sum_i x_i v_i$$

Ben Arous, Geissari, Jagannath (2018)

How is Risk landscape?

The Risk Landscape

Enumerating stationary points

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

Kac-Rice formula to enumerate stationary points (at every risk level and latitude)

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

Annealed computation not always matching the quenched (correct) result

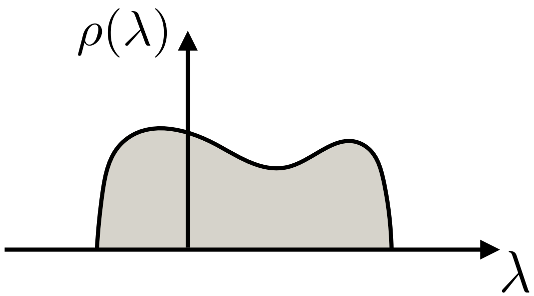
Subag (2015)

Introduction of Replicas (a formidable task)!

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

> Structure of stationary points

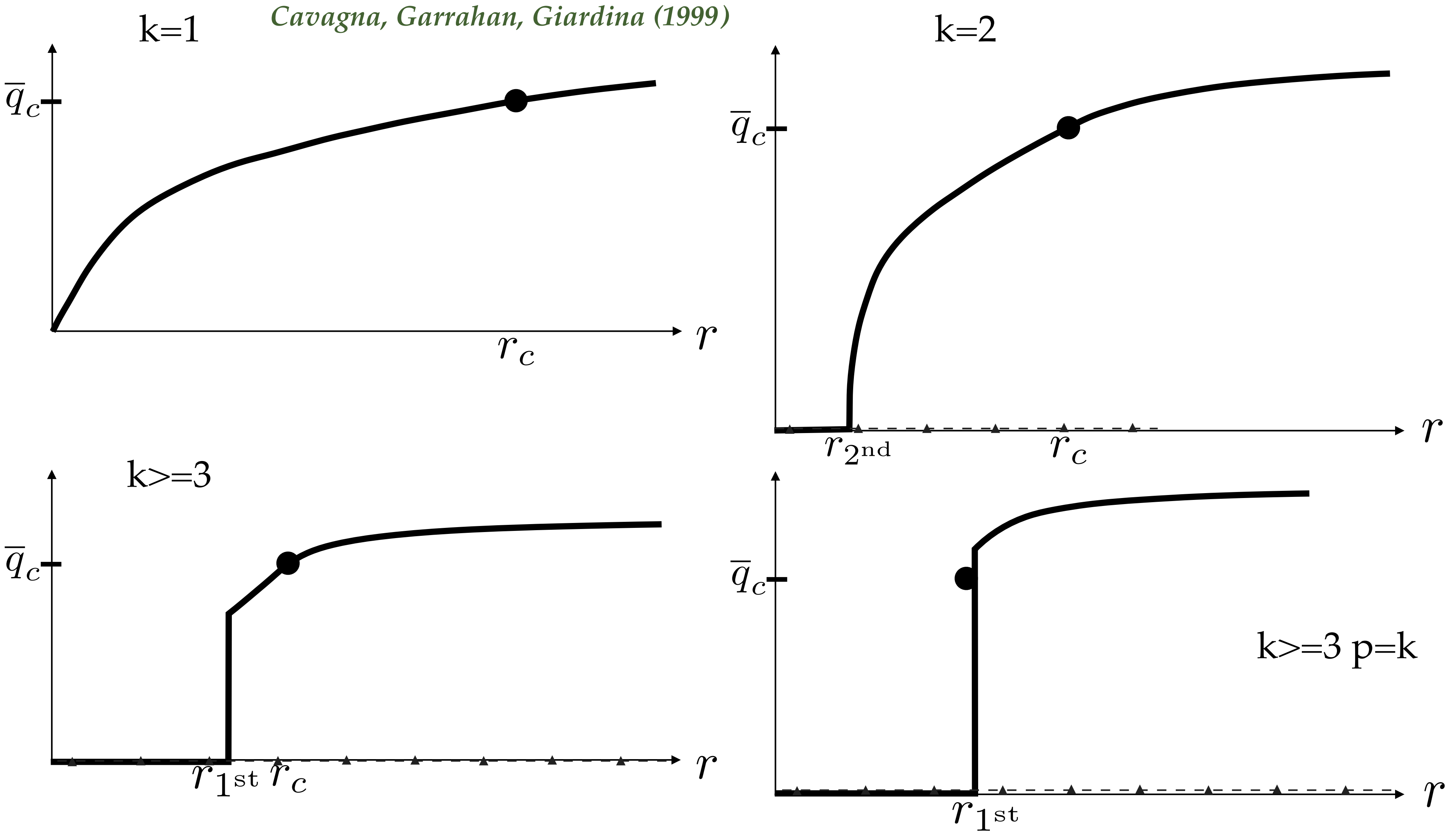
> Distribution of Hessians eigenvalues



Generalised Tensor PCA: the rough landscape

Ros, Ben Arous, Biroli, Cammarota PRX 2019

Showing the latitude range for which a band of minima exist

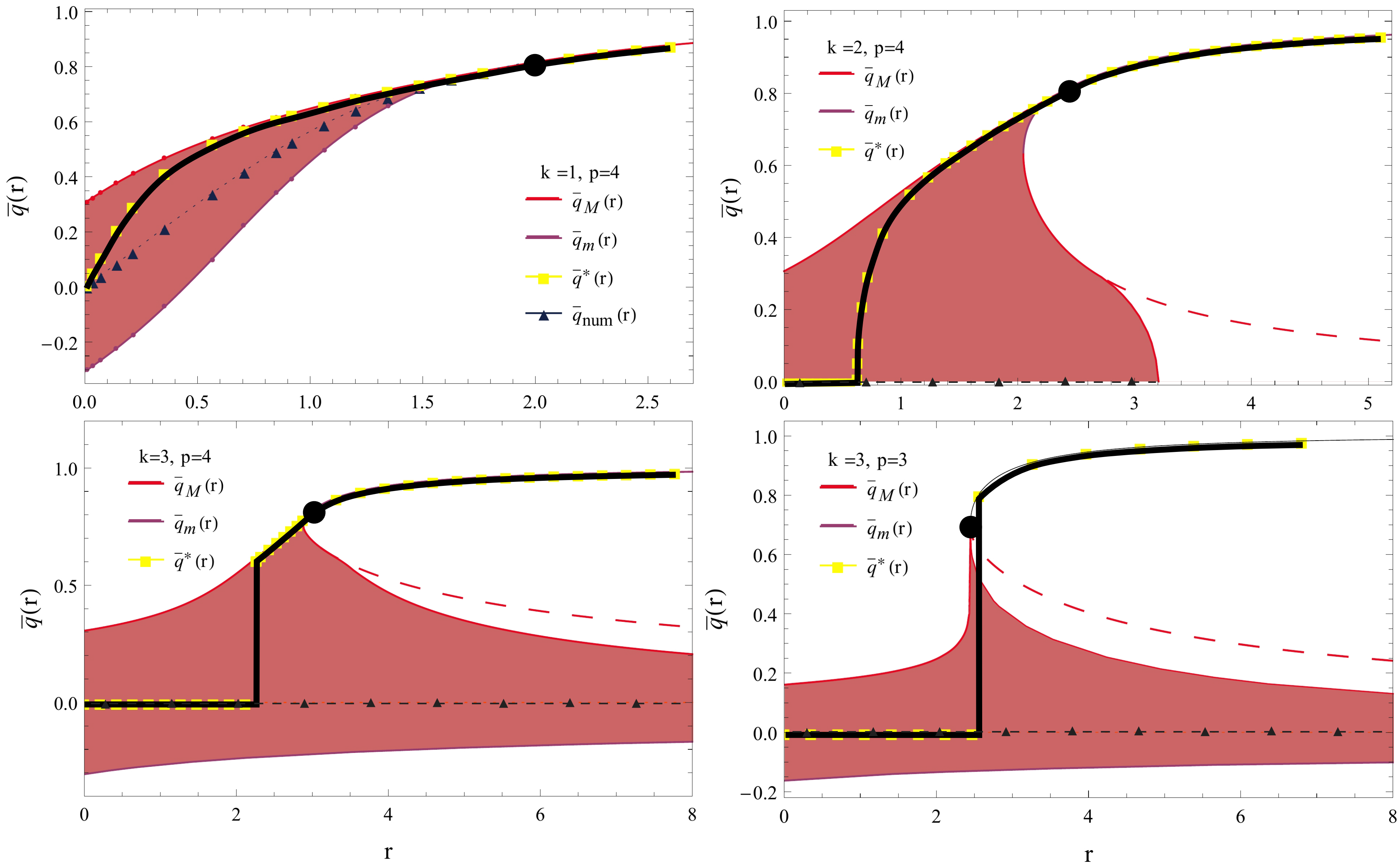


Generalised Tensor PCA: the rough landscape

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

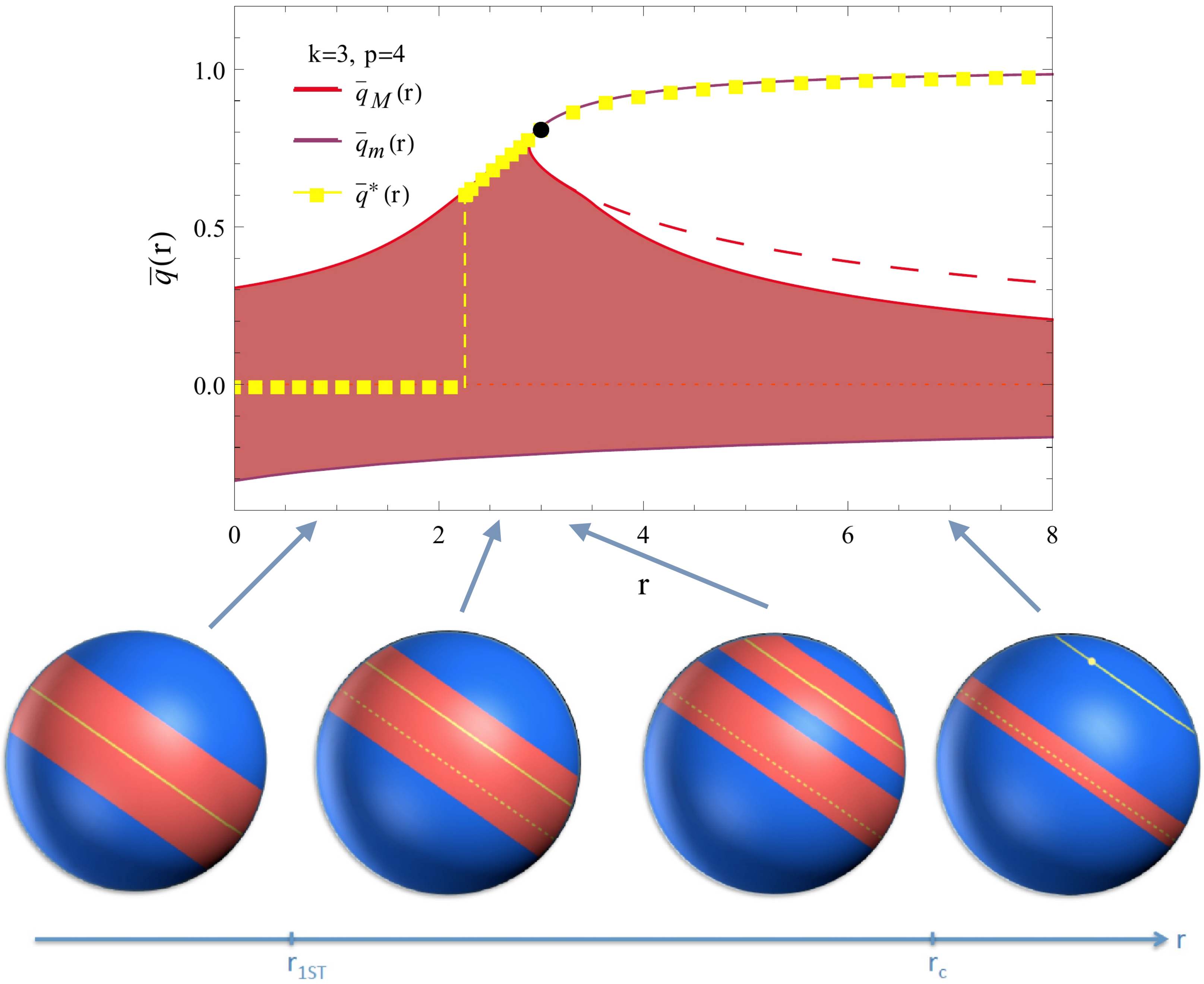
Showing the latitude range for which a band of minima exist

Cavagna, Garrahan, Giardina (1999)



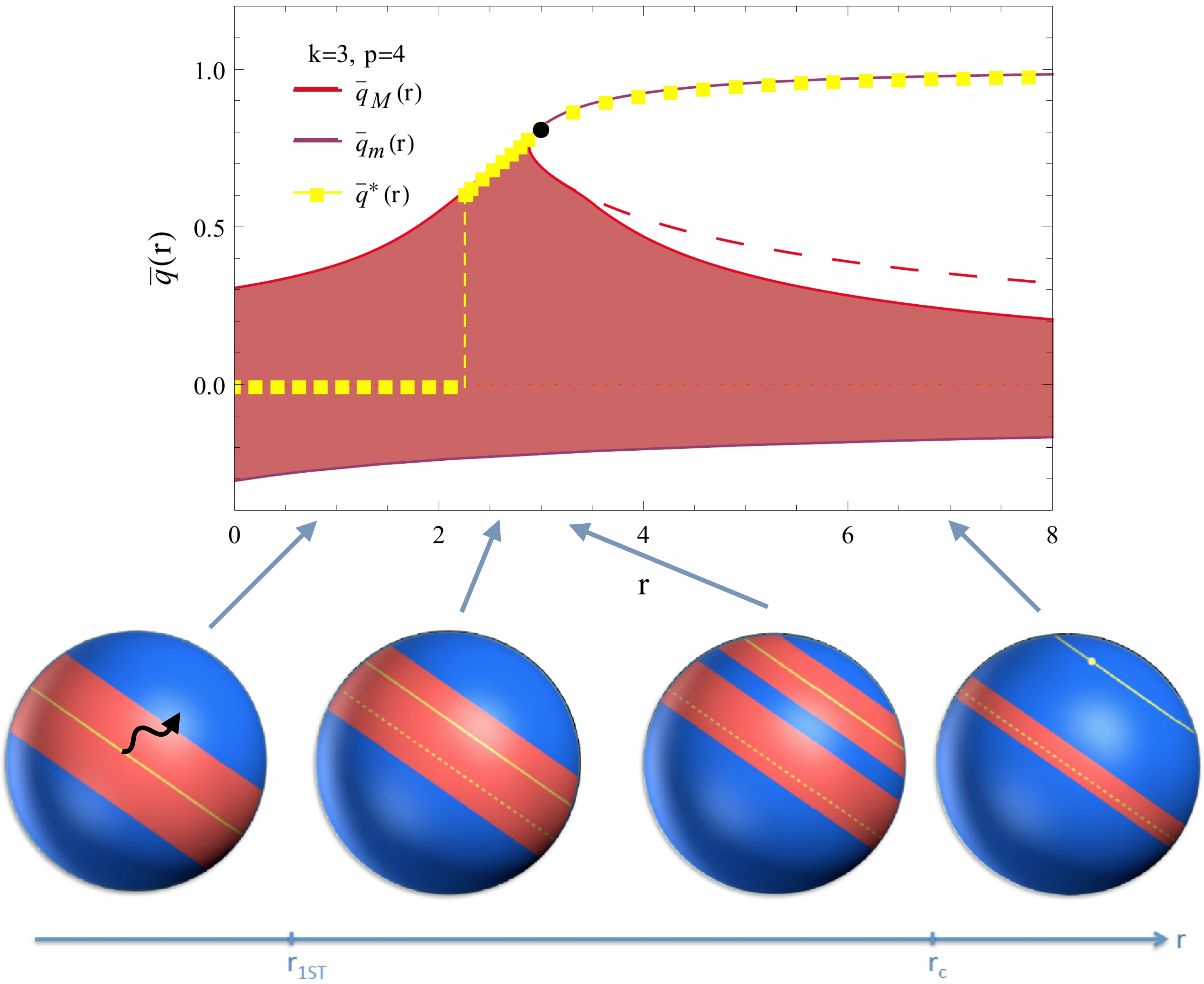
A first dynamical perspective

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



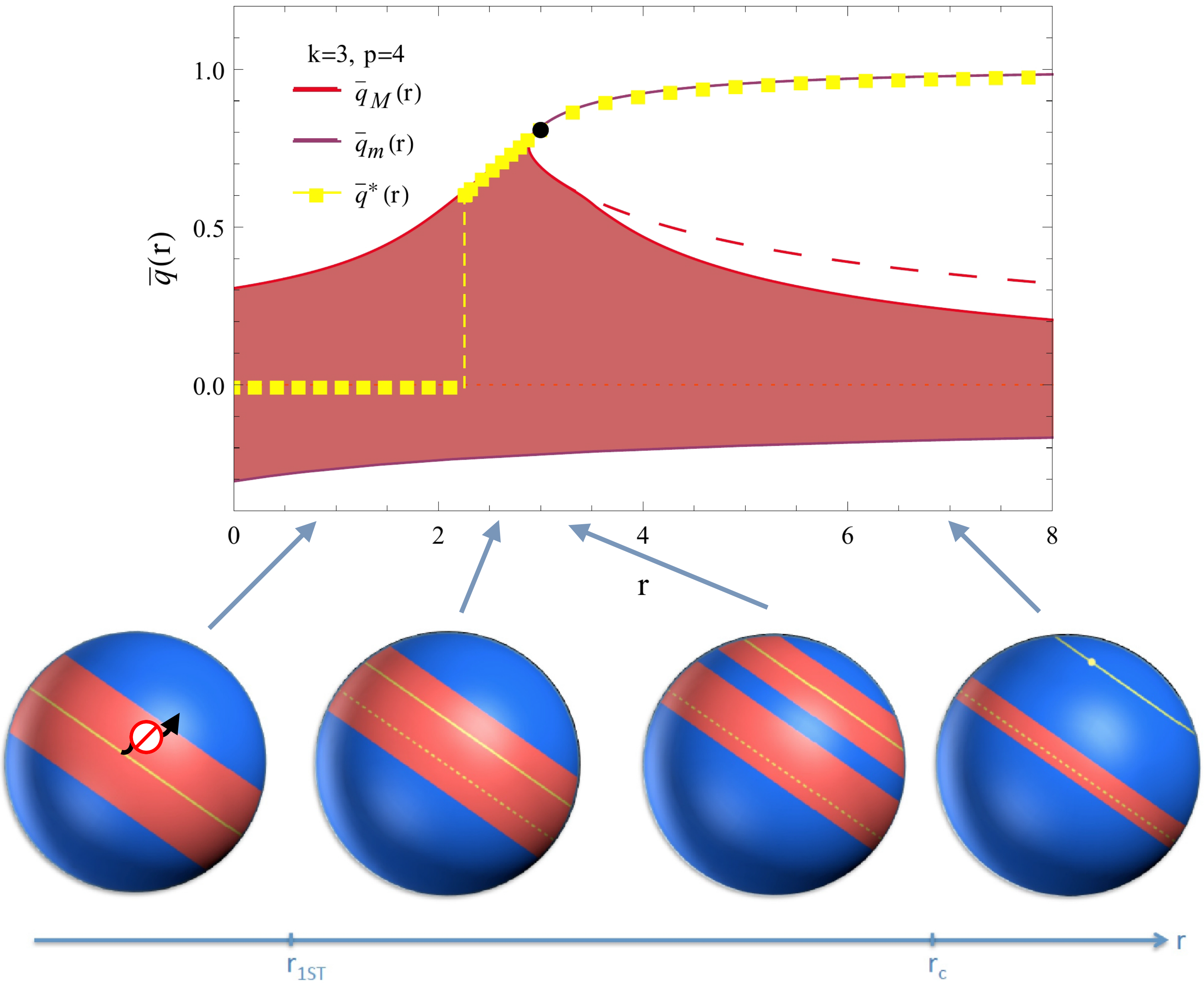
A first dynamical perspective

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



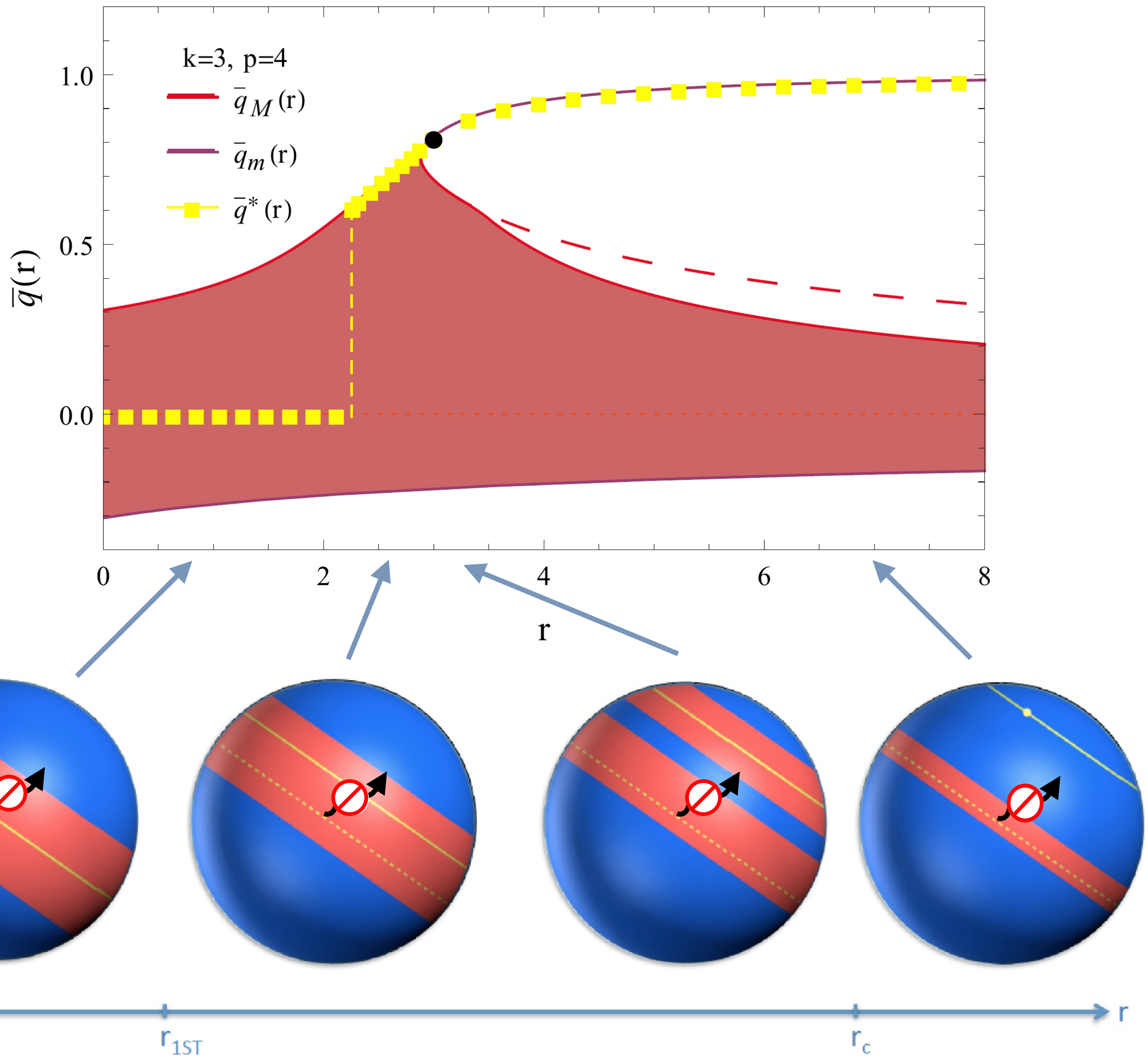
A first dynamical perspective

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



A first dynamical perspective

Ros, Ben Arous, Biroli, Cammarota PRX 2019



Gradient Flow & Langevin

Mixed matrix-tensor PCA

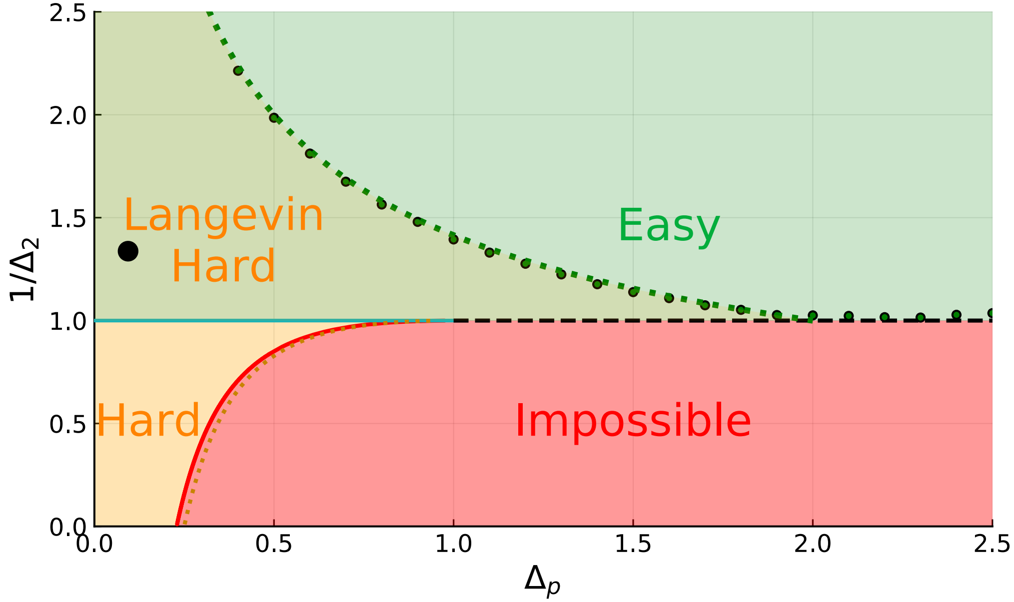
Sarao, Biroli, Cammarota, Krzakala, Urbani, Zdeborova **arXiv:1812.09066 2018**

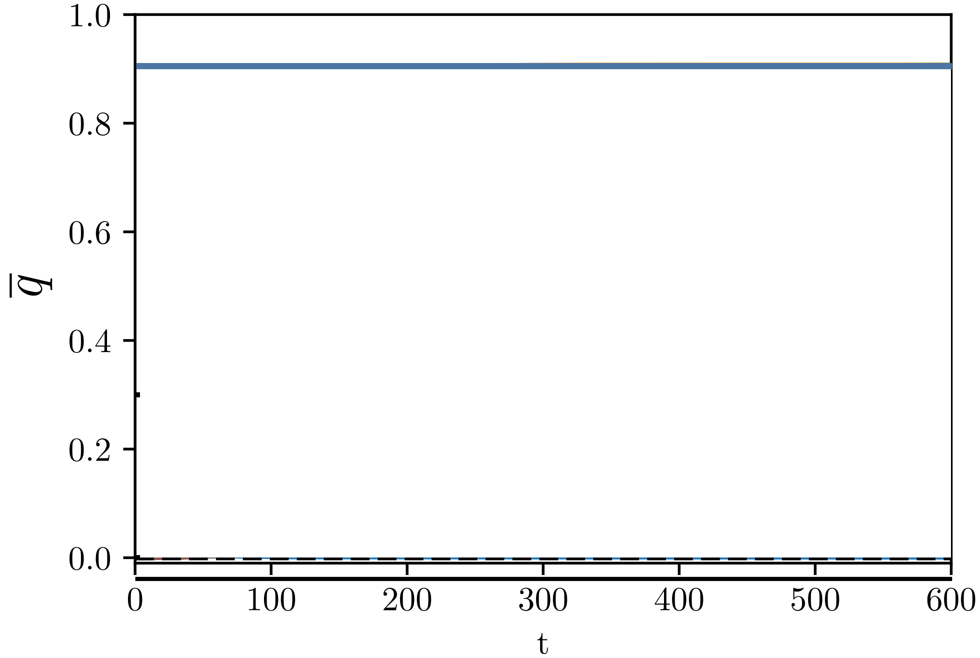
$$T_{i,j} = W_{i,j} + v_i v_j \qquad \langle W_{i,j}^2 \rangle = \Delta_2$$

$$S_{k,l,m} = Z_{k,l,m} + v_k v_l v_m \qquad \langle Z_{k,l,m}^2 \rangle = \Delta_p$$

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

Approximate Message Passing much better than Langevin





Mixed matrix-tensor PCA

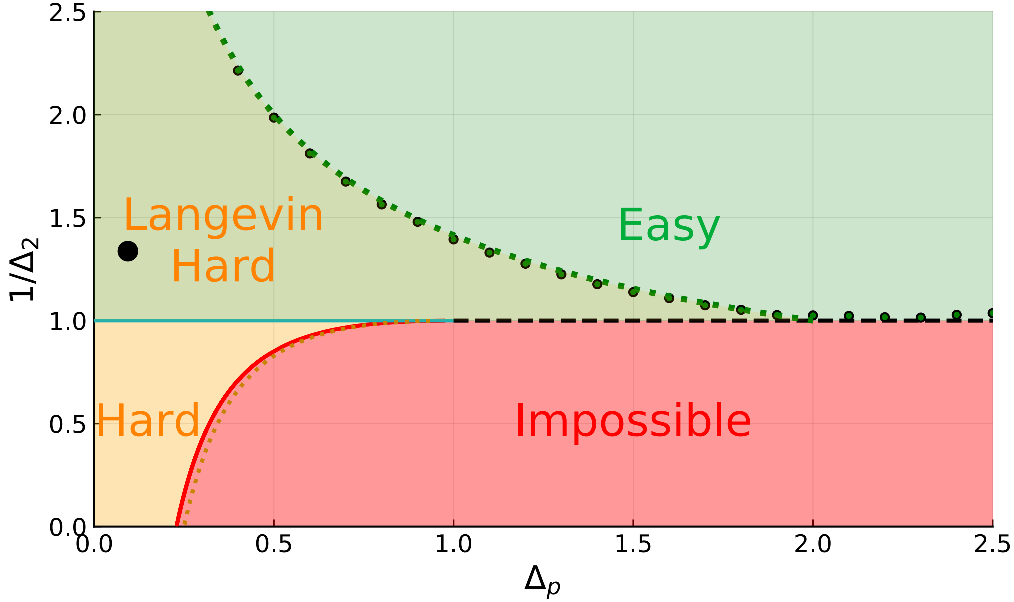
Sarao, Biroli, Cammarota, Krzakala, Urbani, Zdeborova **arXiv:1812.09066 2018**

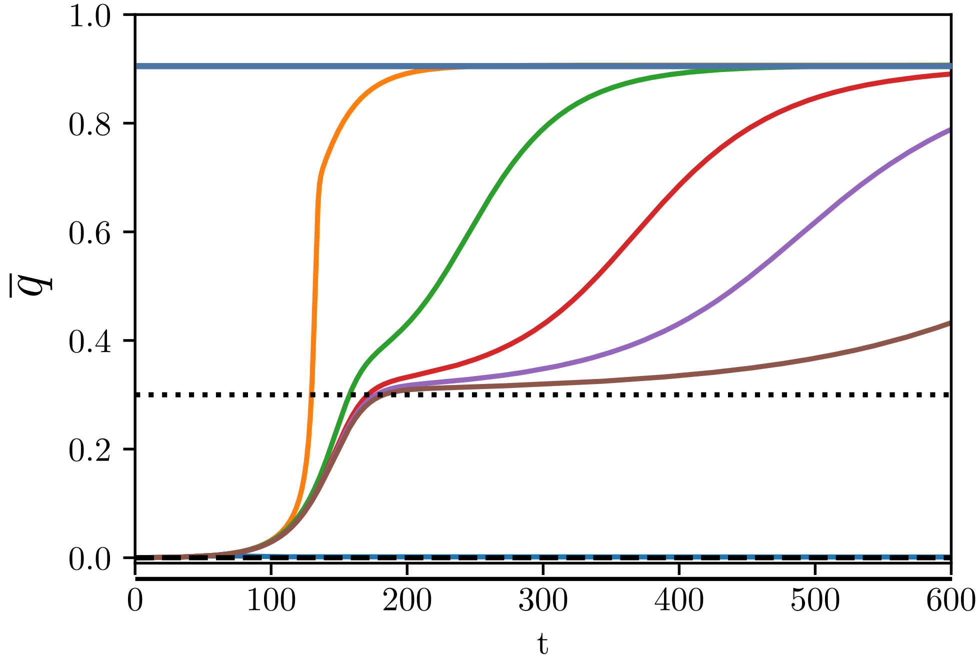
$$T_{i,j} = W_{i,j} + v_i v_j \qquad \langle W_{i,j}^2 \rangle = \Delta_2$$

$$S_{k,l,m} = Z_{k,l,m} + v_k v_l v_m \qquad \langle Z_{k,l,m}^2 \rangle = \Delta_p$$

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

Approximate Message Passing much better than Langevin





Mixed matrix-tensor PCA

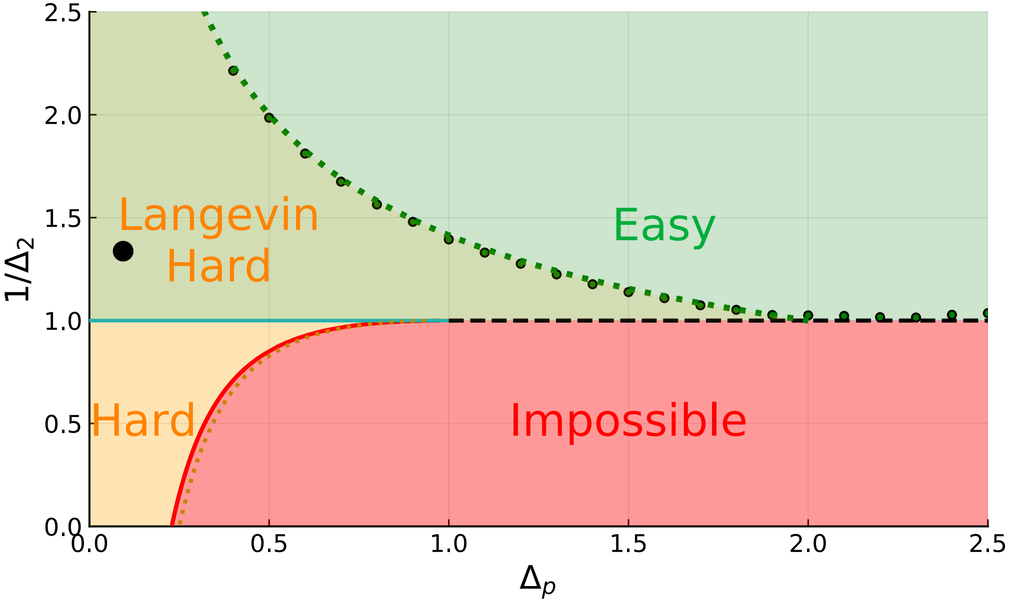
Sarao, Biroli, Cammarota, Krzakala, Urbani, Zdeborova arXiv:1812.09066 2018

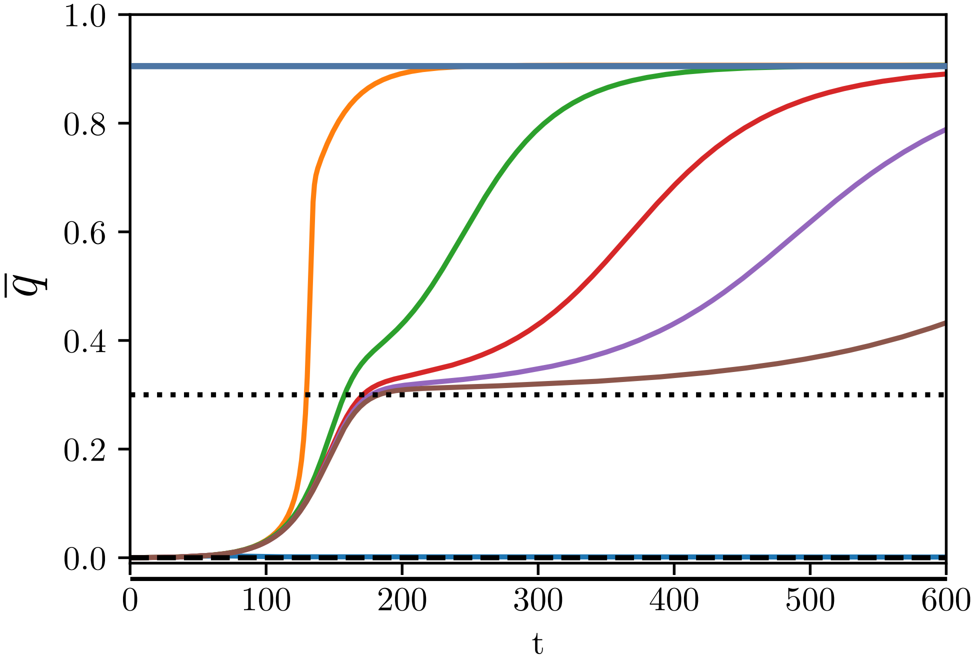
$$T_{i,j} = W_{i,j} + v_i v_j \qquad \langle W_{i,j}^2 \rangle = \Delta_2$$

$$S_{k,l,m} = Z_{k,l,m} + v_k v_l v_m \qquad \langle Z_{k,l,m}^2 \rangle = \Delta_p$$

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

Approximate Message Passing much better than Langevin

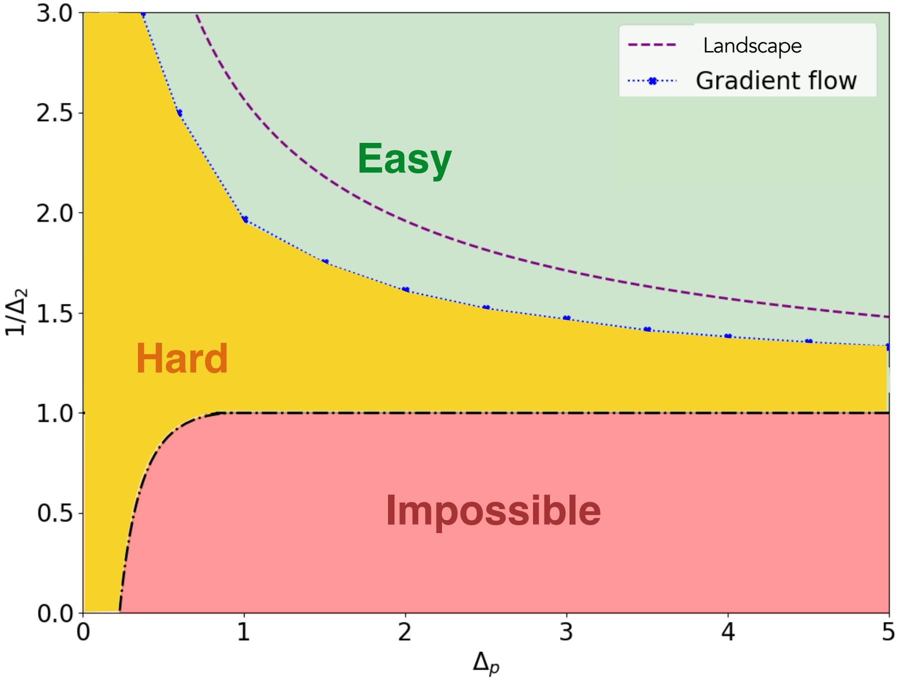




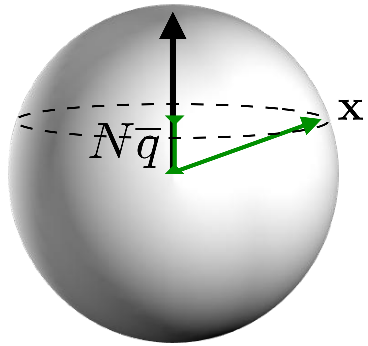
Given problem / algorithm used, landscape info can help to chose the best strategy

The puzzle of non trapping minima

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

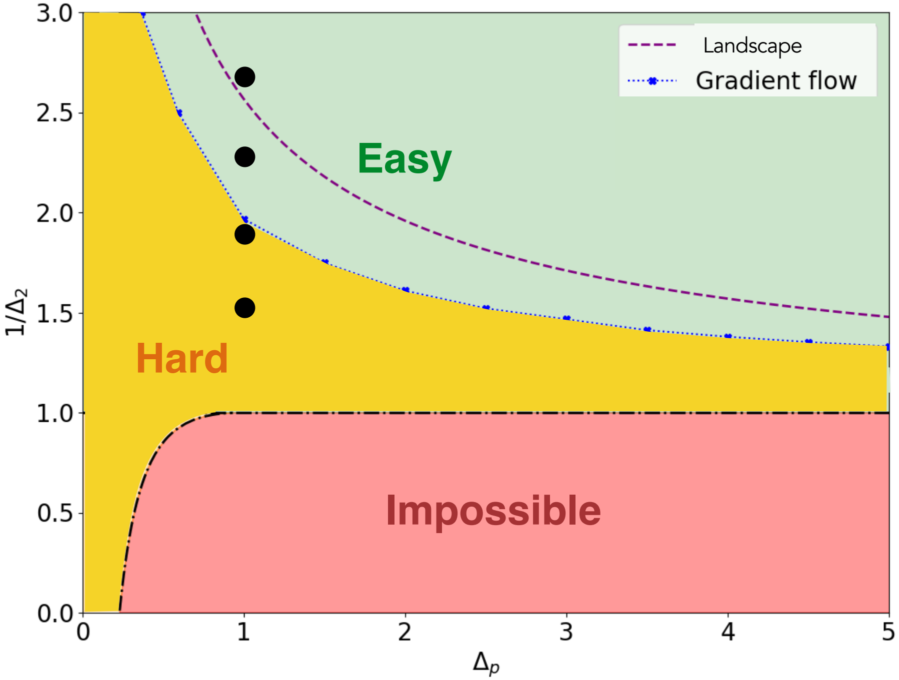


Landscape trivialisation
transition
lies strangely
in the EASY phase

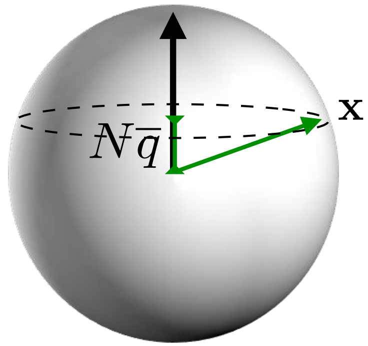


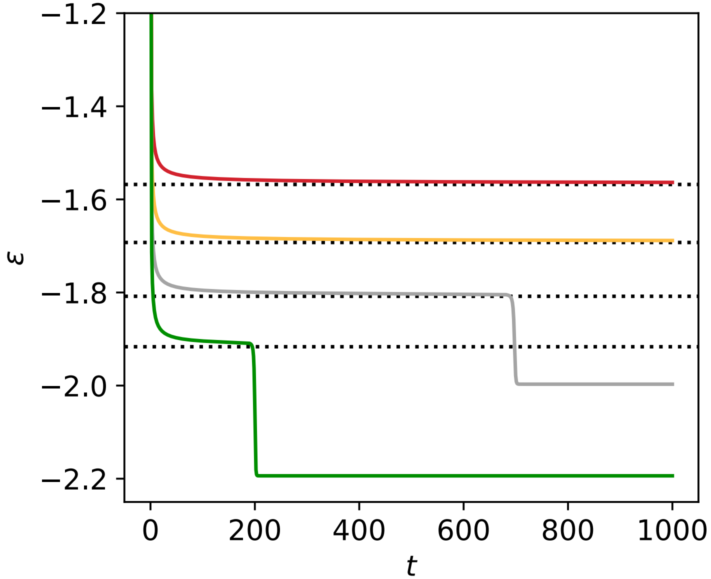
The puzzle of non trapping minima

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



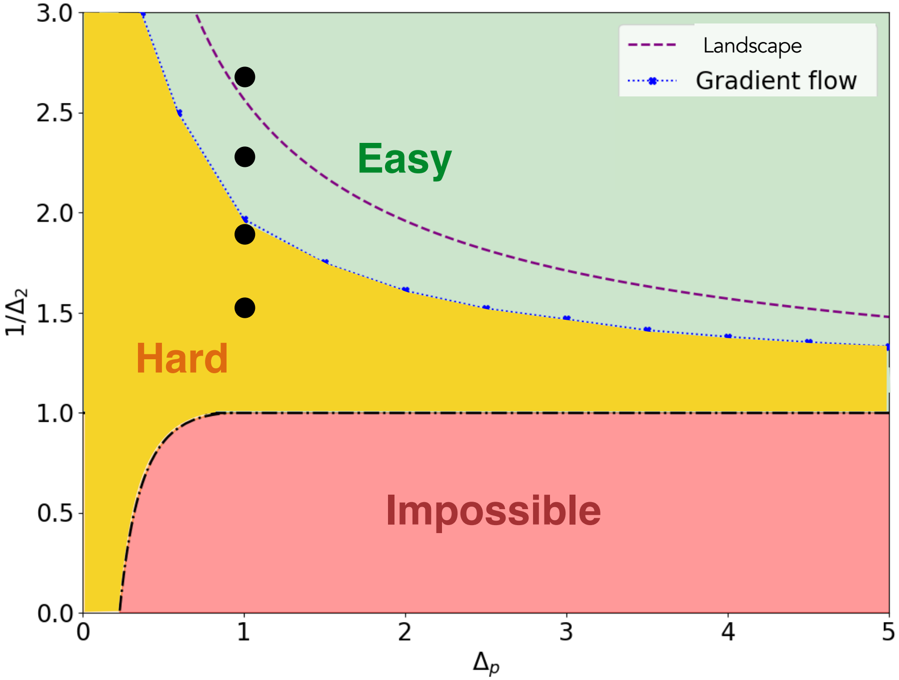
Landscape trivialisation transition lies strangely in the EASY phase



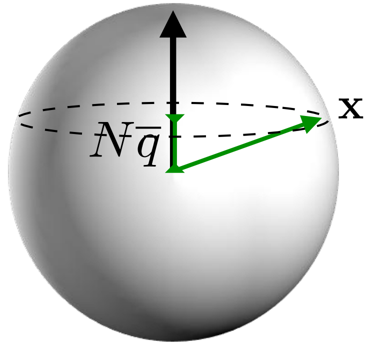


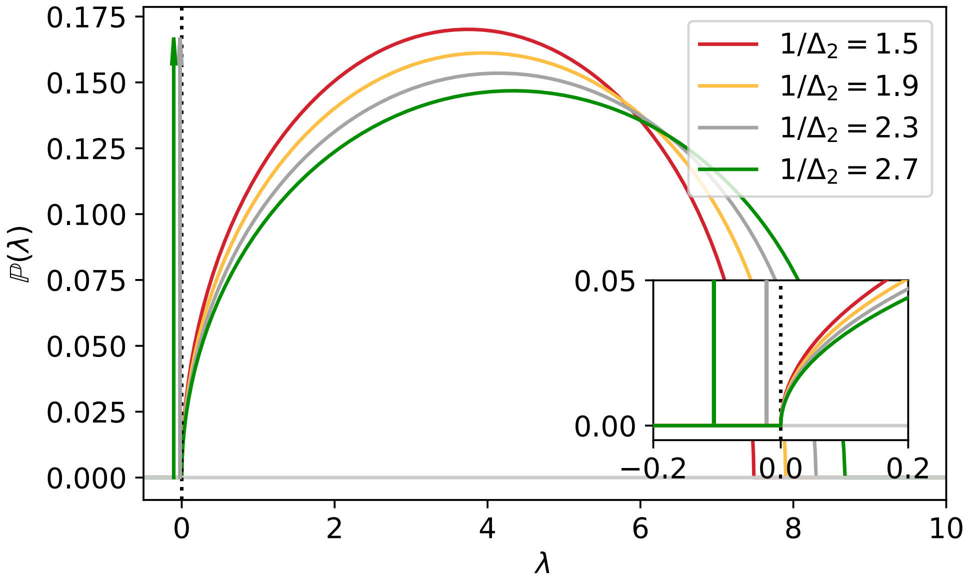
The puzzle of non trapping minima

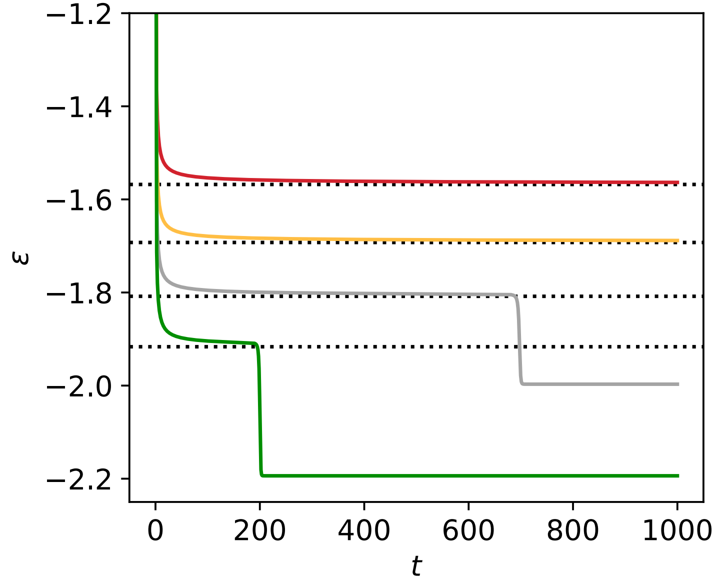
Sarao, Biroli, Cammarota, Krzakala, Zdeborova **Spotlight at NIPS 2019**



Landscape trivialisation transition lies strangely in the EASY phase

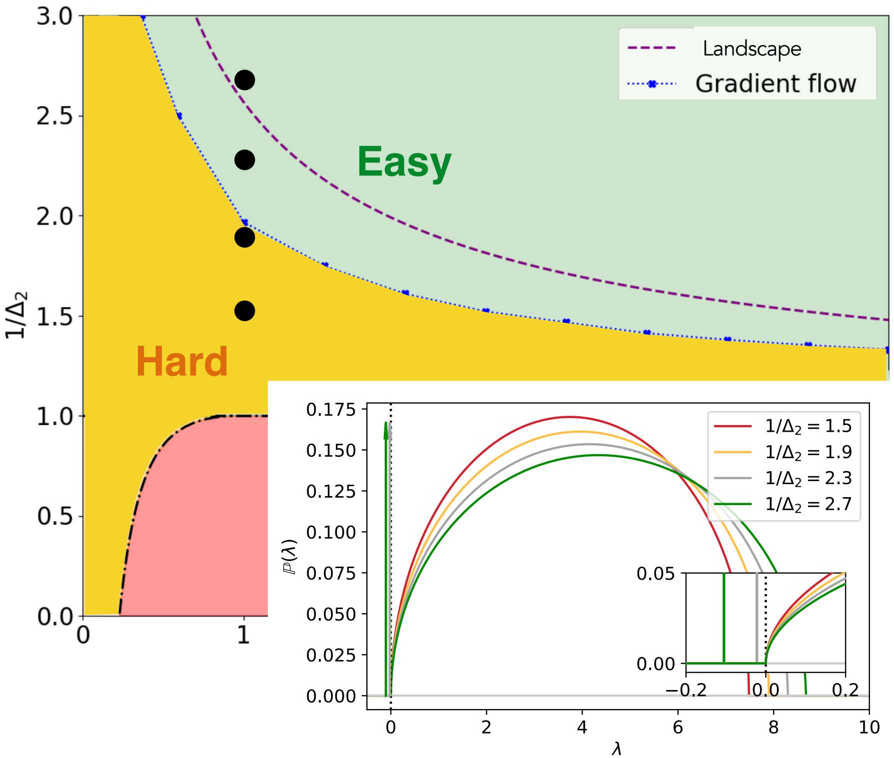




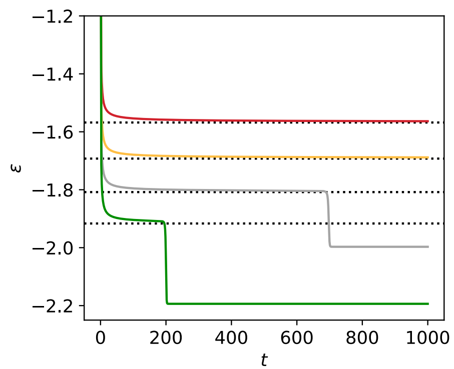


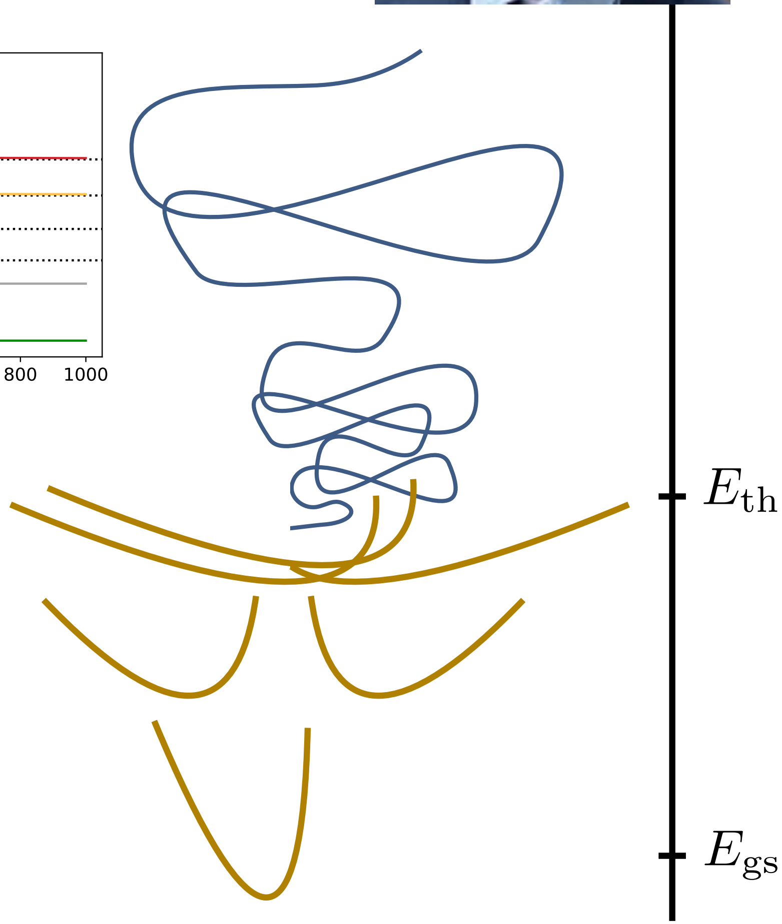
The puzzle of non trapping minima

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



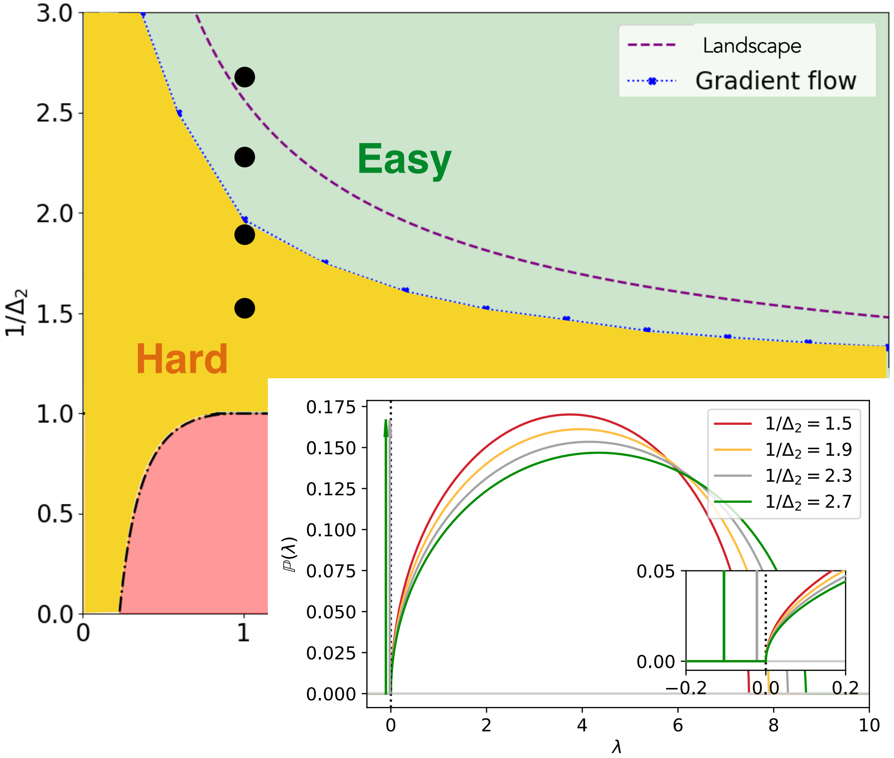
Landscape trivialisation
transition
lies strangely
in the EASY phase



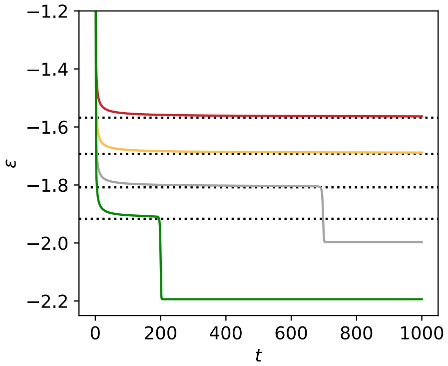


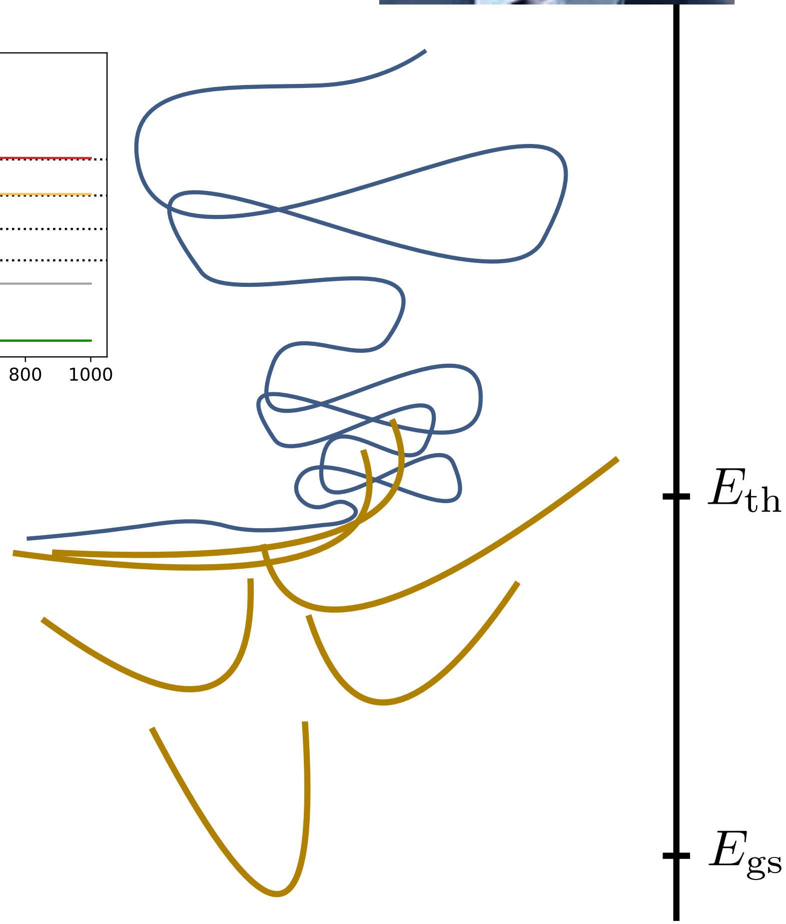
The puzzle of non trapping minima

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



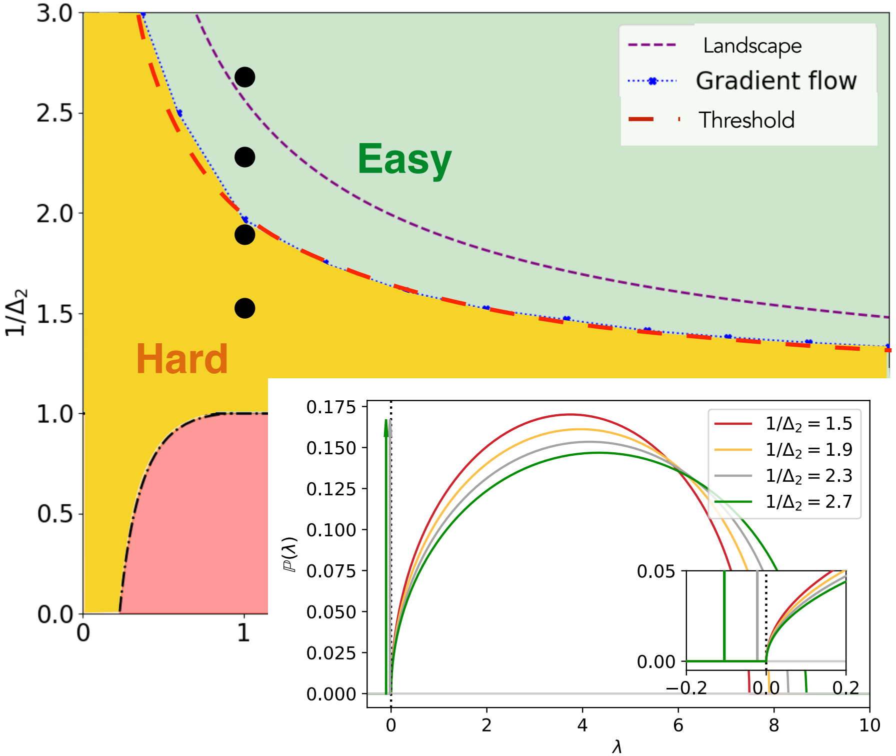
Landscape trivialisation transition lies strangely in the EASY phase



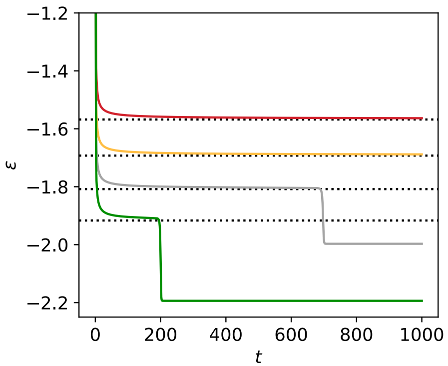


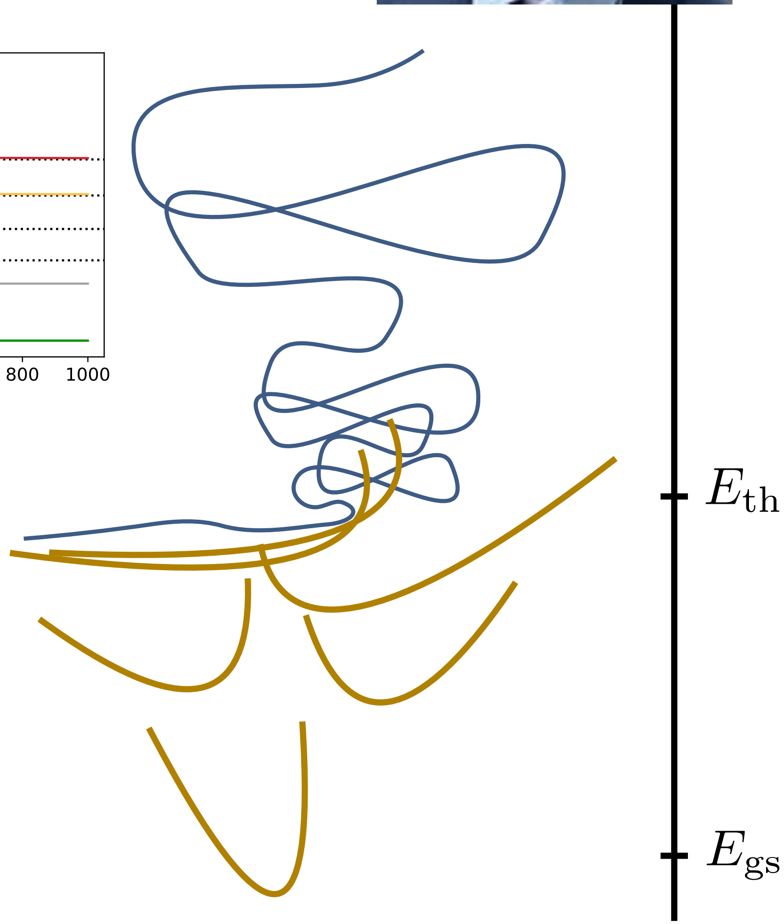
The puzzle of non trapping minima

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



Landscape trivialisation
transition
lies strangely
in the EASY phase





Answer in

Aging dynamics

and

details of Landscape structure

Ben Arous, Baik, Peche 2004

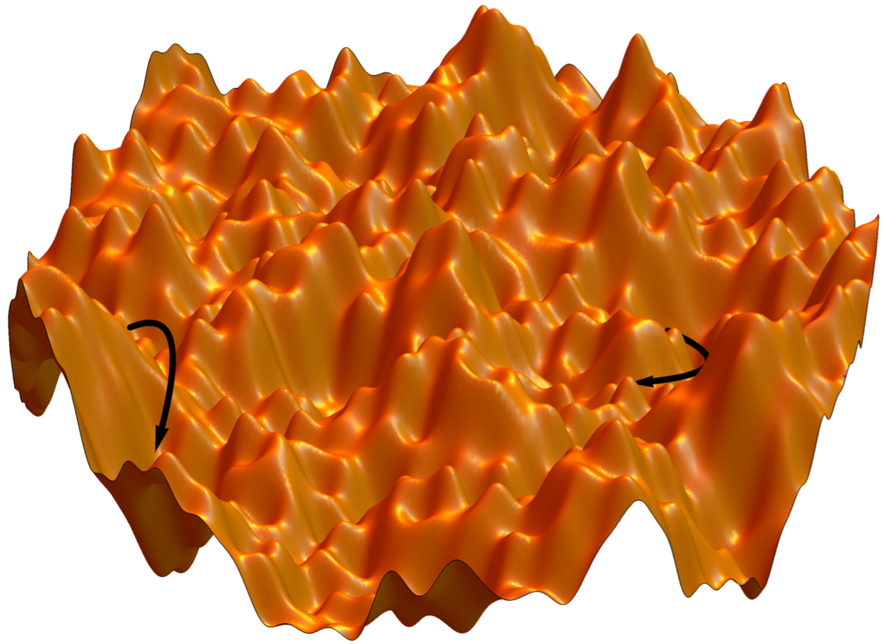
thorough exploration of the first layer of minima
the most frequent, but the most fragile too!

Can GD be competitive?

Best known algorithms for Tensor PCA

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

Information Theoretic transition $r_{\mathrm{IT}} \sim O(1)$



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

Richard and Montanari (2014),
Ben Arous Gheissari and Jagannath (2018)

Tensor Unfolding, SOS, Homotopy based method $r_{\mathrm{AL}} \sim N^{\frac{k-2}{4}}$

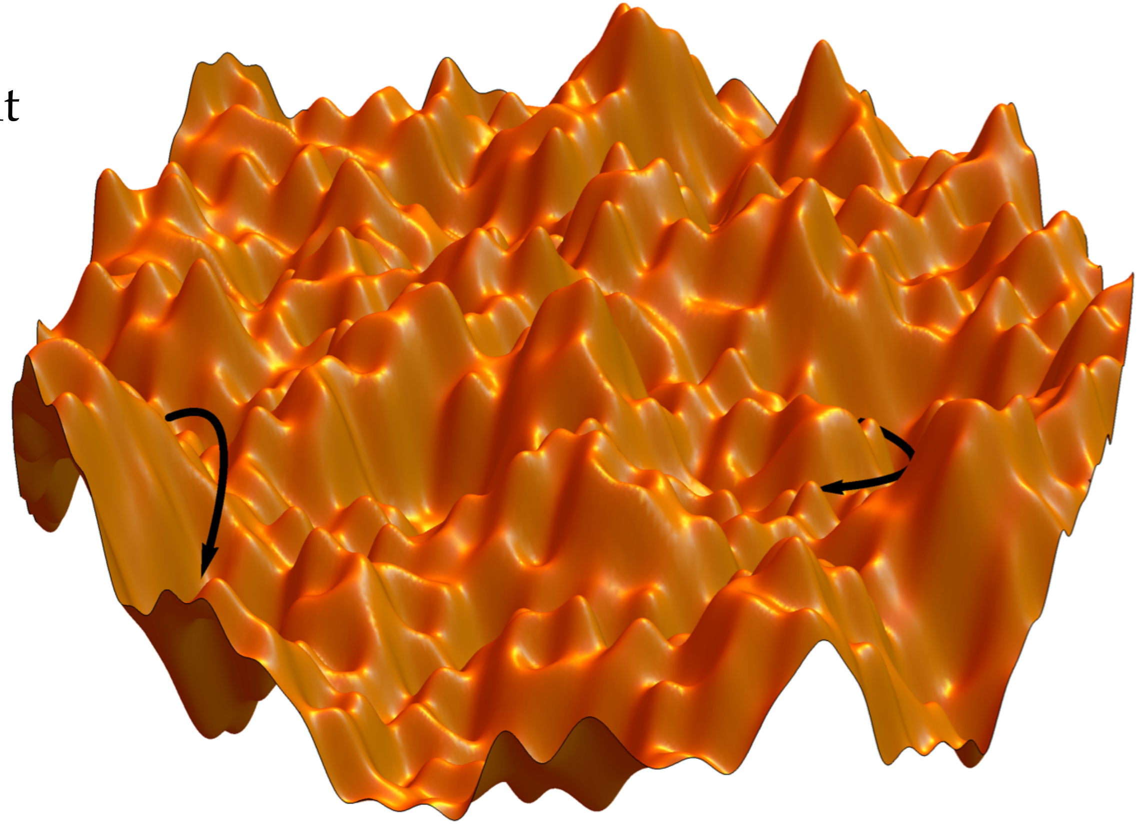
Richard and Montanari (2014),
Hopkins Shi and Steurer (2015),
Anandkumar Deng Ge and Mobahi (2016),
Wein Alaoui and Moore (2019)

Ironing the landscape

Biroli, Cammarota, Ricci-Tersenghi **arXiv:1905.12294 2019**

A traditional way to Iron the landscape -> Big Data!

Each data point carries an independent realisation of the noise component

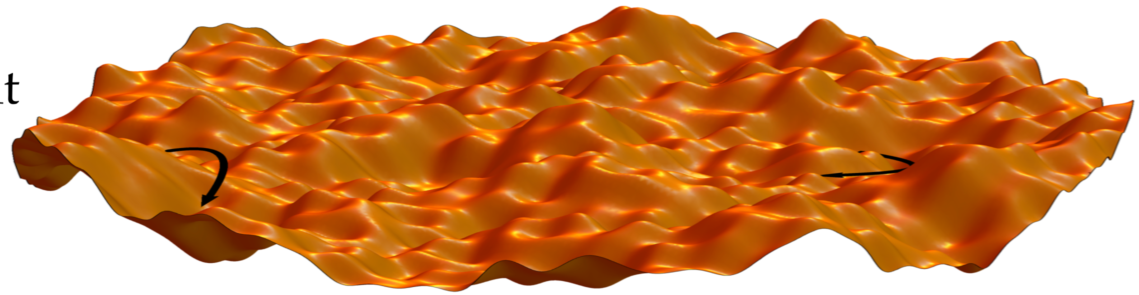


Ironing the landscape

Biroli, Cammarota, Ricci-Tersenghi **arXiv:1905.12294 2019**

A traditional way to Iron the landscape -> Big Data!

Each data point carries an independent realisation of the noise component



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

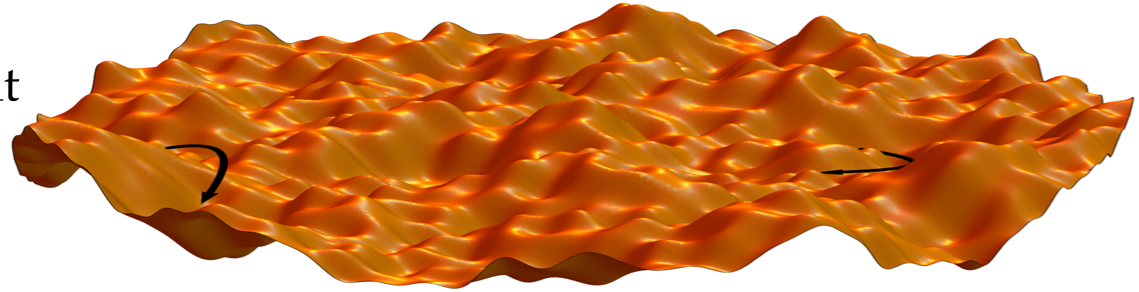
$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) - \eta \nabla_{\mathbf{x}} \mathcal{L}$$

Ironing the landscape

Biroli, Cammarota, Ricci-Tersenghi **arXiv:1905.12294 2019**

A traditional way to Iron the landscape -> Big Data!

Each data point carries an independent realisation of the noise component



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

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) - \eta \nabla_{\mathbf{x}} \mathcal{L}$$

What if only one data point is available? $\ell(\mathbf{x}; \mathbf{X}^1, Y^1) = H(\mathbf{x})$

IDEA: use the fact that noise could be <u>uncorrelated</u> in different regions of the landscape

Central Limit Theorem will do the rest

Replicated Gradient Descent

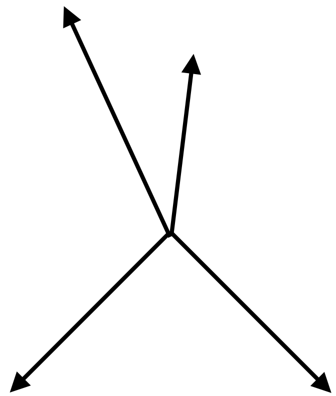
Biroli, Cammarota, Ricci-Tersenghi **arXiv:1905.12294 2019**

$$\ell(\mathbf{x}; \mathbf{X}^1, Y^1) = H(\mathbf{x})$$

$a \in [1, R]$ $\quad$ $\mathbf{x}^a(t=0)$ uniformly in Ω

$$\mathbf{x}_{CM}(0) = \frac{1}{R}\sum_{a=1}^{R} \mathbf{x}^a(0)$$

$$\nabla_{\mathbf{x}} \mathcal{L}_R = \frac{1}{R}\sum_{a=1}^{R} \nabla_{\mathbf{x}} H(\mathbf{x})|_{\mathbf{x}_a}$$



Replicated Gradient Descent

Biroli, Cammarota, Ricci-Tersenghi **arXiv:1905.12294 2019**

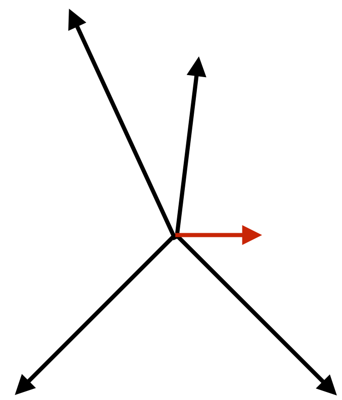
$$\ell(\mathbf{x}; \mathbf{X}^1, Y^1) = H(\mathbf{x})$$

$a \in [1, R]$ $\quad \mathbf{x}^a(t=0)$ uniformly in Ω

$$\mathbf{x}_{CM}(0) = \frac{1}{R}\sum_{a=1}^{R} \mathbf{x}^a(0)$$

$$\nabla_{\mathbf{x}} \mathcal{L}_R = \frac{1}{R}\sum_{a=1}^{R} \nabla_{\mathbf{x}} H(\mathbf{x})|_{\mathbf{x}_a}$$

$$\mathbf{x}_{CM}(t+\Delta t) = \mathbf{x}_{CM}(t) - \eta \nabla_{\mathbf{x}} \mathcal{L}_R$$



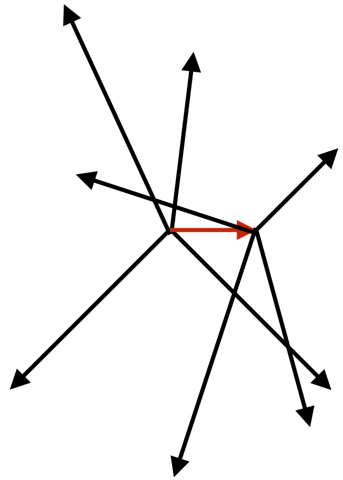
Replicated Gradient Descent

Biroli, Cammarota, Ricci-Tersenghi **arXiv:1905.12294 2019**

$$\ell(\mathbf{x}; \mathbf{X}^1, Y^1) = H(\mathbf{x})$$

$a \in [1, R]$ $\quad \mathbf{x}^a(t = 0)$ uniformly in Ω

$$\mathbf{x}_{CM}(0) = \frac{1}{R}\sum_{a=1}^{R} \mathbf{x}^a(0)$$



$$\nabla_{\mathbf{x}} \mathcal{L}_R = \frac{1}{R}\sum_{a=1}^{R} \nabla_{\mathbf{x}} H(\mathbf{x})|_{\mathbf{x}_a}$$

$$\mathbf{x}_{CM}(t + \Delta t) = \mathbf{x}_{CM}(t) - \eta \nabla_{\mathbf{x}} \mathcal{L}_R$$

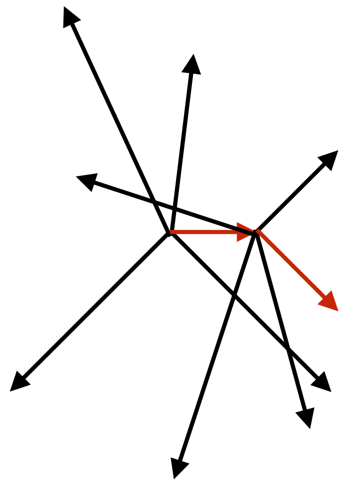
$\mathbf{x}^a(t + \Delta t)$ uniformly in Ω

such that $\quad \mathbf{x}_{CM}(t + \Delta t) = \frac{1}{R}\sum_{a=1}^{R} \mathbf{x}^a(t + \Delta t)$

Replicated Gradient Descent

Biroli, Cammarota, Ricci-Tersenghi **arXiv:1905.12294 2019**

$$\ell(\mathbf{x}; \mathbf{X}^1, Y^1) = H(\mathbf{x})$$



$a \in [1, R]$ $\qquad \mathbf{x}^a(t=0)$ uniformly in Ω

$$\mathbf{x}_{CM}(0) = \frac{1}{R}\sum_{a=1}^{R} \mathbf{x}^a(0)$$

$$\nabla_{\mathbf{x}} \mathcal{L}_R = \frac{1}{R}\sum_{a=1}^{R} \nabla_{\mathbf{x}} H(\mathbf{x})|_{\mathbf{x}_a}$$

$$\mathbf{x}_{CM}(t+\Delta t) = \mathbf{x}_{CM}(t) - \eta \nabla_{\mathbf{x}} \mathcal{L}_R$$

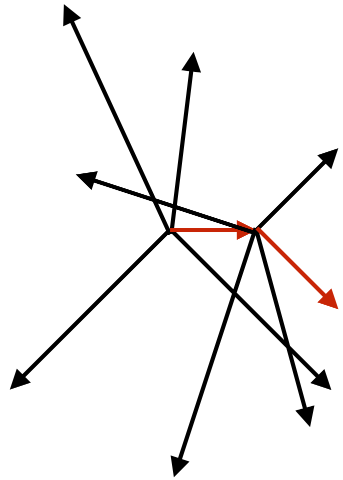
$\mathbf{x}^a(t+\Delta t)$ $\qquad$ uniformly in Ω

such that $\quad \mathbf{x}_{CM}(t+\Delta t) = \frac{1}{R}\sum_{a=1}^{R} \mathbf{x}^a(t+\Delta t)$

Replicated Gradient Descent

Biroli, Cammarota, Ricci-Tersenghi arXiv:1905.12294 2019

$$\ell(\mathbf{x}; \mathbf{X}^1, Y^1) = H(\mathbf{x})$$



$a \in [1, R]$ $\quad\quad$ $\mathbf{x}^a(t=0)$ uniformly in Ω

$$\mathbf{x}_{CM}(0) = \frac{1}{R}\sum_{a=1}^{R} \mathbf{x}^a(0)$$

$$\nabla_{\mathbf{x}}\mathcal{L}_R = \frac{1}{R}\sum_{a=1}^{R} \nabla_{\mathbf{x}} H(\mathbf{x})|_{\mathbf{x}_a}$$

$$\mathbf{x}_{CM}(t+\Delta t) = \mathbf{x}_{CM}(t) - \eta \nabla_{\mathbf{x}}\mathcal{L}_R$$

$\mathbf{x}^a(t+\Delta t)$ $\quad\quad$ uniformly in Ω

such that $\quad \mathbf{x}_{CM}(t+\Delta t) = \dfrac{1}{R}\displaystyle\sum_{a=1}^{R} \mathbf{x}^a(t+\Delta t)$

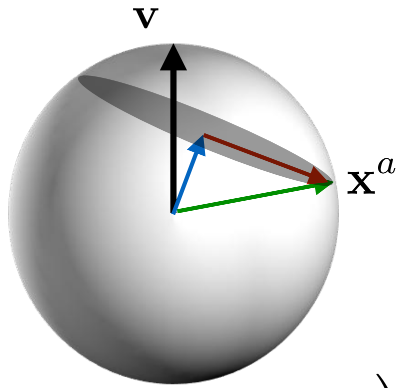
$$\nabla_{\mathbf{x}}\mathcal{L}_R = \frac{1}{R}\sum_{a=1}^{R} \mathbf{g}_a = \frac{1}{R}\sum_{a=1}^{R} (r\mathbf{g_s} + \mathbf{g_n}_a) = r\mathbf{g_s} + \mathbf{g_n}_R \qquad \mathbf{g_n}_R \sim \frac{\mathbf{g_n}_a}{\sqrt{R}}$$

algorithms based on similar ideas: *Anandkumar Deng Ge and Mobahi (2016)*
Baldassi et al. (2016)

RGD matches best algorithms on Tensor PCA

Biroli, Cammarota, Ricci-Tersenghi **arXiv:1905.12294 2019**

$$H = -\sum_{(i_1,\ldots,i_k)} J_{i_1,\ldots,i_k} x_{i_1}\ldots x_{i_k} - rN\left(\sum_i \frac{x_i v_i}{N}\right)^k \qquad \text{for} \quad k=3$$

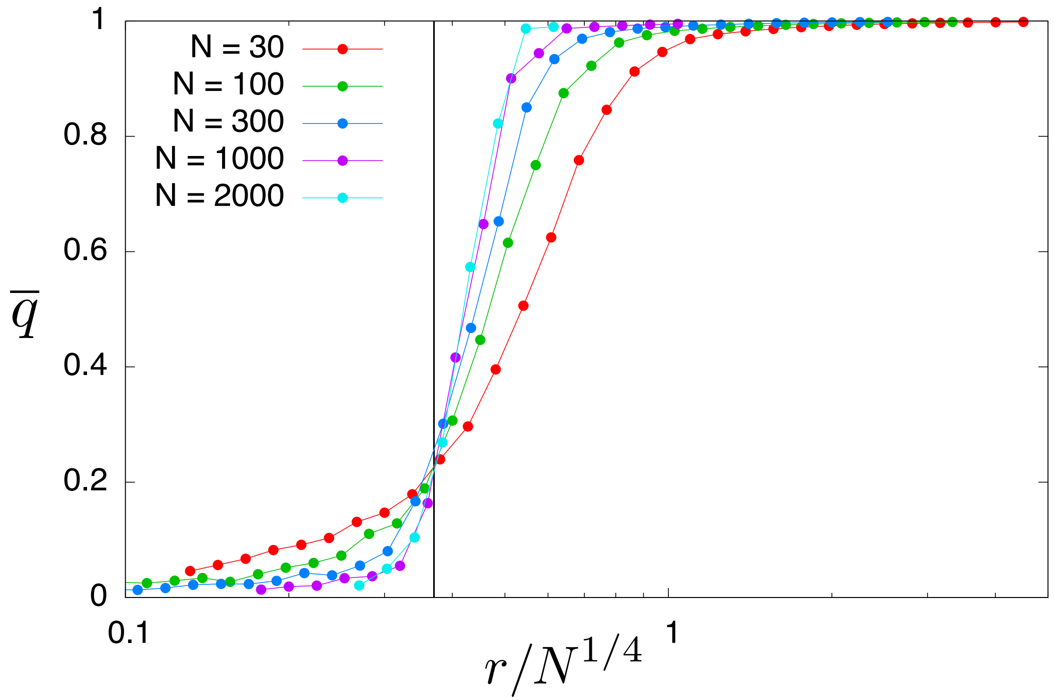


$$\mathbf{x}^a(t) = \mathbf{x}_{CM}(t) + (1-n^2(t))\,\mathbf{x}^{\perp a}(t)$$

with $Nn^2(t) = ||\mathbf{x}_{CM}||_2^2$

and $N = ||\mathbf{x}^{\perp}(t)||_2^2$

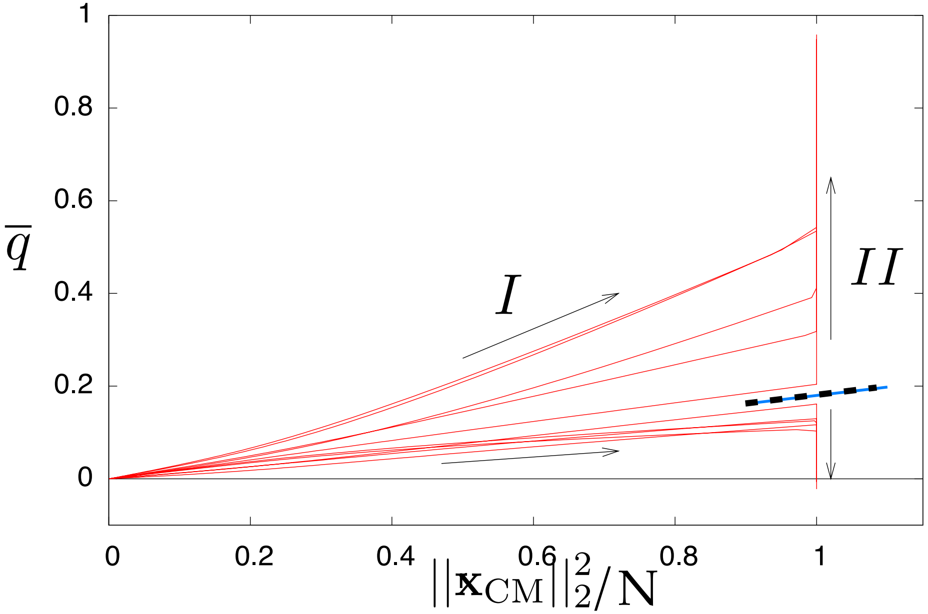
$$x_{CM,i}(t+\Delta t) = x_{CM,i}(t) - \eta\mathbb{E}[\nabla_{x_i} H] = x_{CM,i}(t) - \eta\left((1-n^2(t))\sum_j T_{ijj} + \sum_{j\le k} T_{ijk} x_{CM,j}(t) x_{CM,k}(t)\right)$$

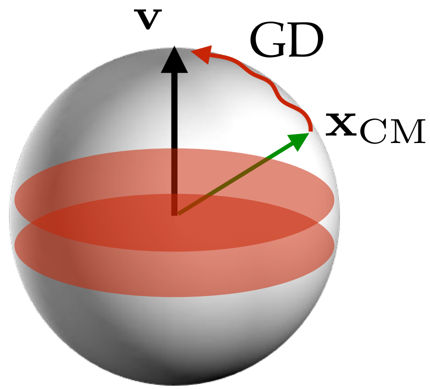


$\rightarrow$ $r_{\mathrm{RGD}} \sim N^{1/4}$

Landscape-based explanation

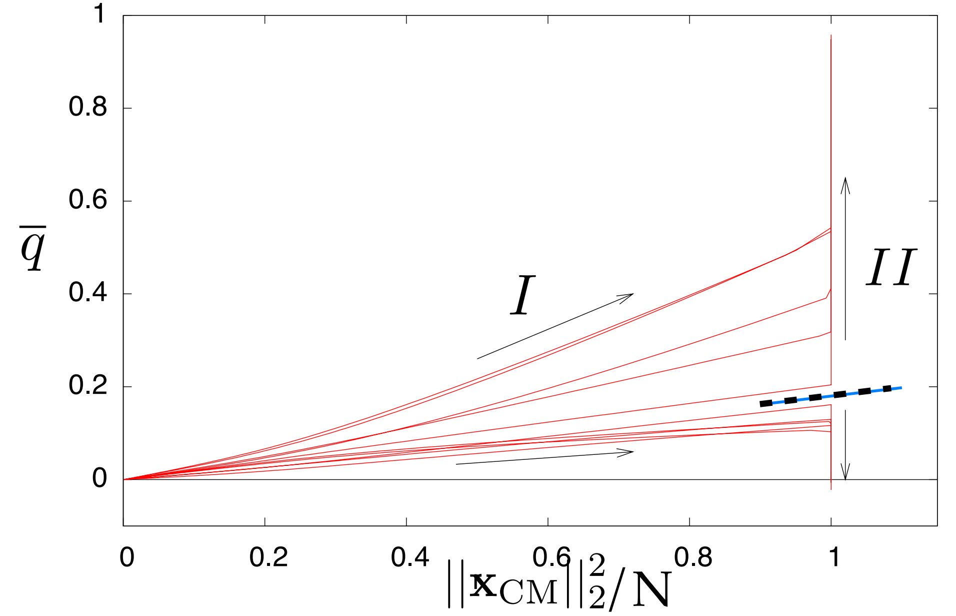
Biroli, Cammarota, Ricci-Tersenghi **arXiv:1905.12294 2019**

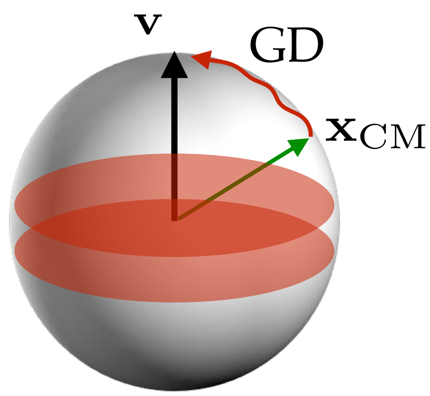


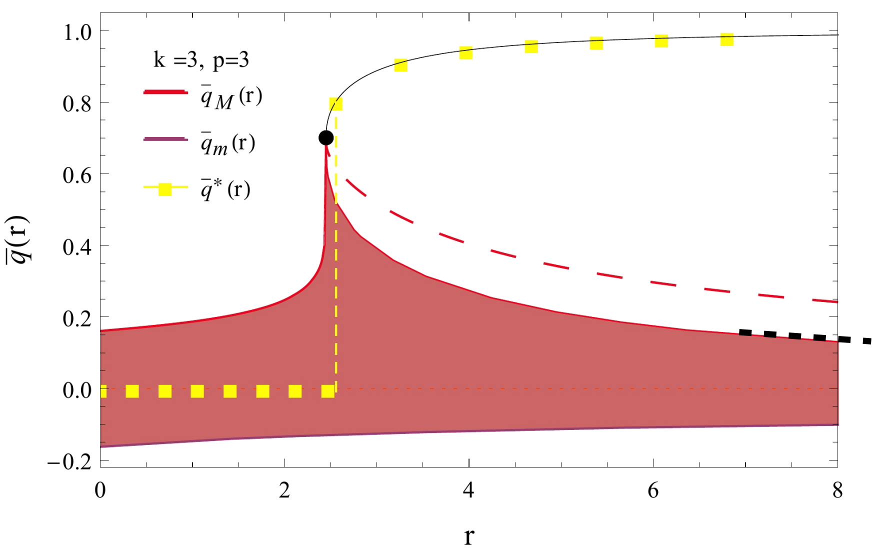


Landscape-based explanation

Biroli, Cammarota, Ricci-Tersenghi **arXiv:1905.12294 2019**

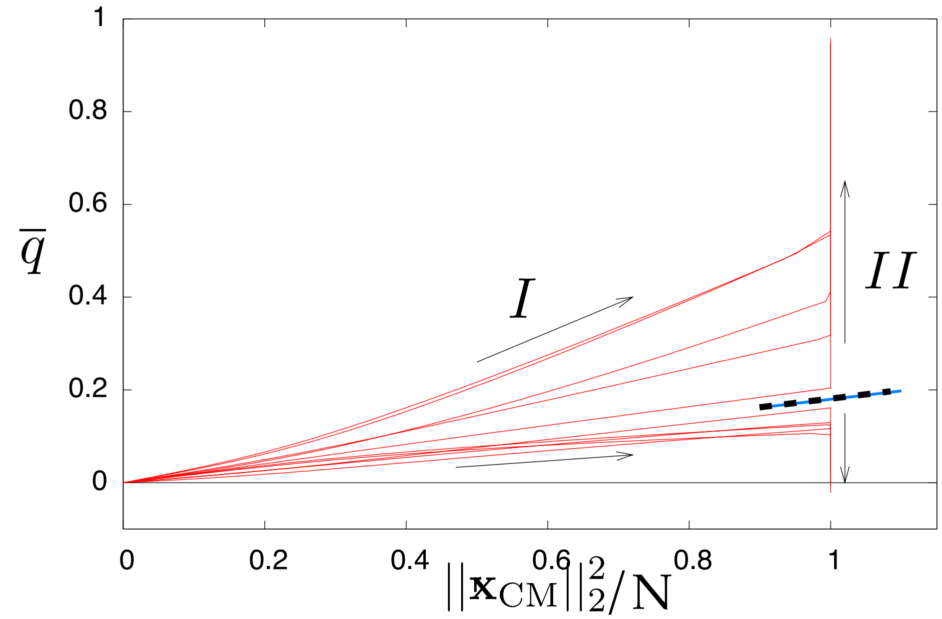


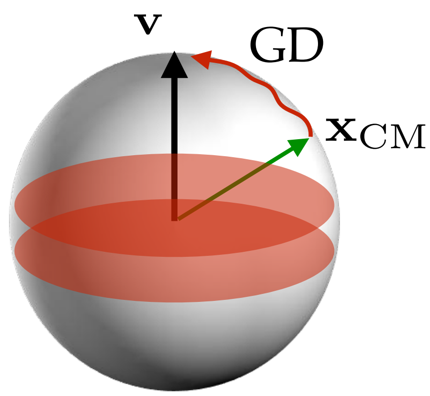


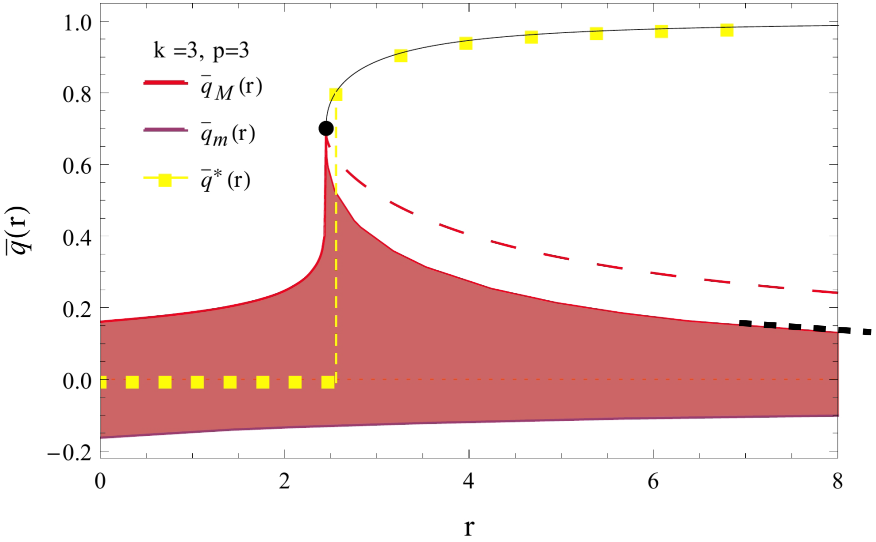


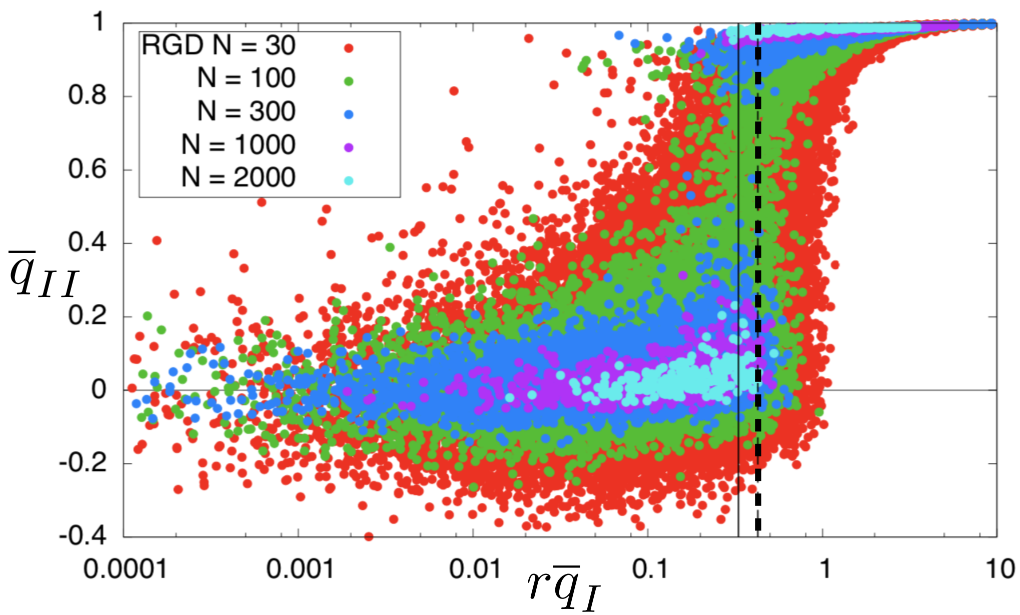
Landscape-based explanation

Biroli, Cammarota, Ricci-Tersenghi arXiv:1905.12294 2019









Conclusion/Discussion

Focus on the intimate connection between landscape and GD-based dynamics

Only the stability of the most numerous minima does matter

Smart use of the knowledge of landscape structure allows GD to match AMP

Replicated Gradient Descent makes GD competitive, keeping its versatility

Is all the info available contained in landscape?

Can we do better than that or it is not possible because RGD is already exploiting it at best?

Conclusion/Discussion

Focus on the intimate connection between landscape and GD-based dynamics

Only the stability of the most numerous minima does matter

Smart use of the knowledge of landscape structure allows GD to match AMP

Replicated Gradient Descent makes GD competitive, keeping its versatility

Thank you!

Is all the info available contained in landscape?

Can we do better than that or it is not possible because RGD is already exploiting it at best?