Neural Networks learning landscapes and post-quantum cryptographic primitives : a statistical physics approach

Marco Benedetti, Andrej Bogdanov, Enrico Malatesta, Marc Mezard, Gianmarco Perrupato, Alon Rosen, Nikolaj I. Schwartzbach , and Riccardo Zecchina*

Department of Computing Sciences, Bocconi University, Milan

**University of Ottawa*

Work in progress!

Several natural computational challenges in NN

$\mathbf{A} \in \mathbf{R}^{P\times N}$ *random matrix composed by* P N*-dim random rows* $\mathbf{x}^{\mu}$

$$\mathbf{A} := \begin{pmatrix} — \mathbf{x}^1 — \\ \mathbf{x}^2 \\ \vdots \\ \mathbf{x}^P \end{pmatrix}$$

Several natural computational challenges in NN

$\mathbf{A} \in \mathbf{R}^{P\times N}$ *random matrix composed by* P N*-dim random rows* $\mathbf{x}^\mu$

$$\mathbf{A} := \begin{pmatrix} — \mathbf{x}^1 — \\ \mathbf{x}^2 \\ \vdots \\ \mathbf{x}^P \end{pmatrix}$$

- Inversion (learning): given $\mathbf{A} \in \mathbf{R}^{P\times N}$ and the labels $\mathbf{y} \in \{-1,1\}^P$, find any set of weights $W \in \{-1,1\}^N$ such that $y_\mathbf{A}(W) = \mathbf{y}$, assuming such W exists.

- Teacher-student: given $\mathbf{A} \in \mathbf{R}^{P\times N}$ and the labels $\hat{\mathbf{y}} = y_\mathbf{A}(W) \in \{-1,1\}^P$ for uniformly sampled $W \in \{-1,1\}^N$, find any $W' \in \{-1,1\}^N$ such that $y_\mathbf{A}(W') = \hat{\mathbf{y}}$

Several natural computational challenges in NN

$\mathbf{A} \in \mathbf{R}^{P \times N}$ *random matrix composed by* P N*-dim random rows* $\mathbf{x}^\mu$

$$\mathbf{A} := \begin{pmatrix} — \mathbf{x}^1 — \\ \mathbf{x}^2 \\ \vdots \\ \mathbf{x}^P \end{pmatrix}$$

- Inversion (learning): given $\mathbf{A} \in \mathbf{R}^{P \times N}$ and the labels $\mathbf{y} \in \{-1,1\}^P$, find any set of weights $W \in \{-1,1\}^N$ such that $y_\mathbf{A}(W) = \mathbf{y}$, assuming such W exists.

- Teacher-student: given $\mathbf{A} \in \mathbf{R}^{P \times N}$ and the labels $\hat{\mathbf{y}} = y_\mathbf{A}(W) \in \{-1,1\}^P$ for uniformly sampled $W \in \{-1,1\}^N$, find any $W' \in \{-1,1\}^N$ such that $y_\mathbf{A}(W') = \hat{\mathbf{y}}$

- Collision finding: given $\mathbf{A} \in \mathbf{R}^{P \times N}$, find any two $W \neq W' \in \{-1,1\}^N$ such that $y_\mathbf{A}(W) = y_\mathbf{A}(W')$ (unexplored so far).

Plank of the talk

- Local entropy in non convex Neural Networks
- The *Overlap Gap Property* (OGP) and limiting performance of stable algorithms
- Random functions and post-quantum cryptography
- Collision Robust Hash function from random NN and their OGP transition

Training large deep neural network is in principle a non-convex hard computational problem.

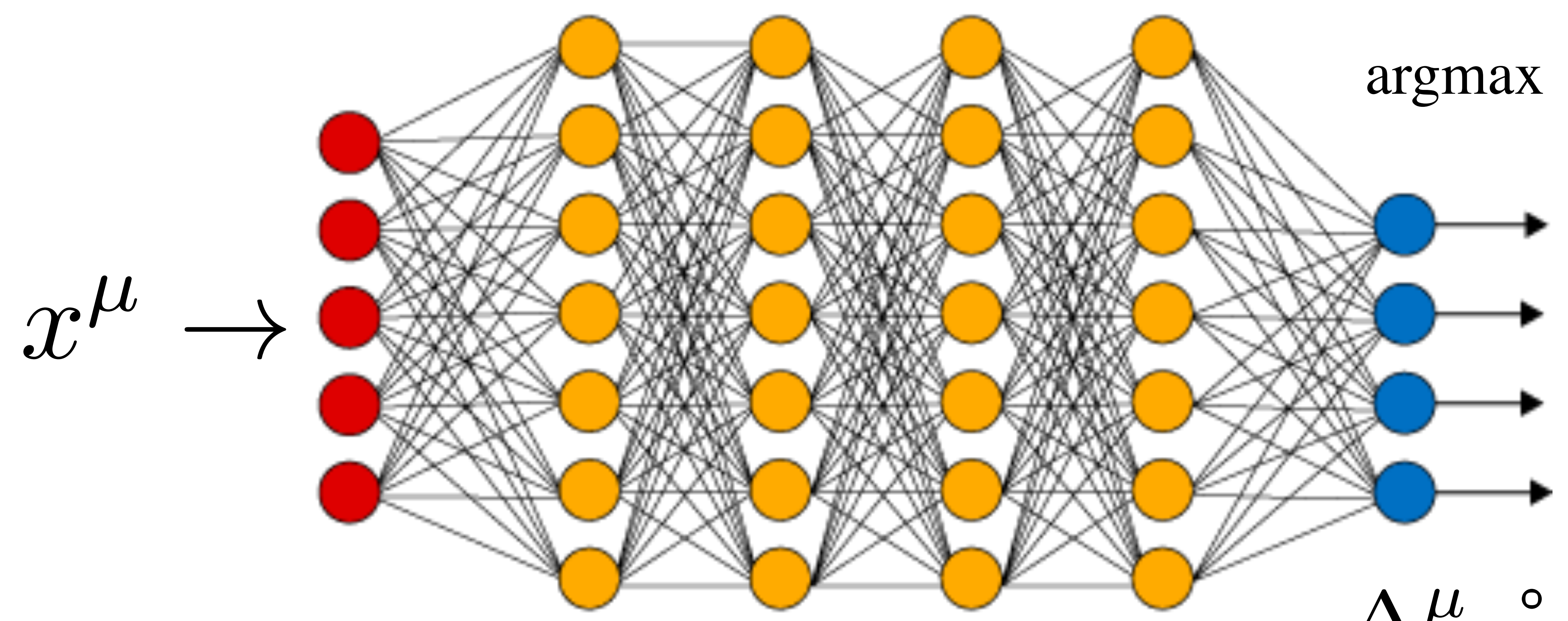
Evidence about learning huge data sets with largely overparametrized networks:

1. Algorithmically easy for relatively simple algorithms (e.g. gradient based algorithms)

2. Lead to solutions which have good generalisation properties

3. “Benign” overfitting even in presence of noise!

Given an input vector of size N, the network computes an output by alternating layers of linear transformations with non-linear activation functions.



Training set: $\{(x^\mu, y^\mu)\}_{\mu=1,\dots,M}$

$$\hat{y}^\mu \longleftrightarrow y^\mu$$

compare with

$$\Delta^\mu \doteq \underbrace{\left(W^K \cdot \sigma_K(x^\mu)\right)}_{\text{pre-activations at layer K}} y^\mu$$

$$\boxed{\hat{y}^\mu = \operatorname{argmax}\left(W^K \sigma\left(W^{K-1} \sigma\left(\dots \sigma\left(W^2 \sigma\left(W^1 x^\mu\right)\right)\right)\right)\right)}$$

output (label)

weights (matrices)

input (vector)

Energy function and surrogate energy functions

- Energy = “0-1 loss”: number of errors on the training set (not differentiable)

$$\boxed{\mathcal{L}_{NE} = \sum_{\mu} (1 - \delta(\hat{y}^{\mu}, y^{\mu}))}$$

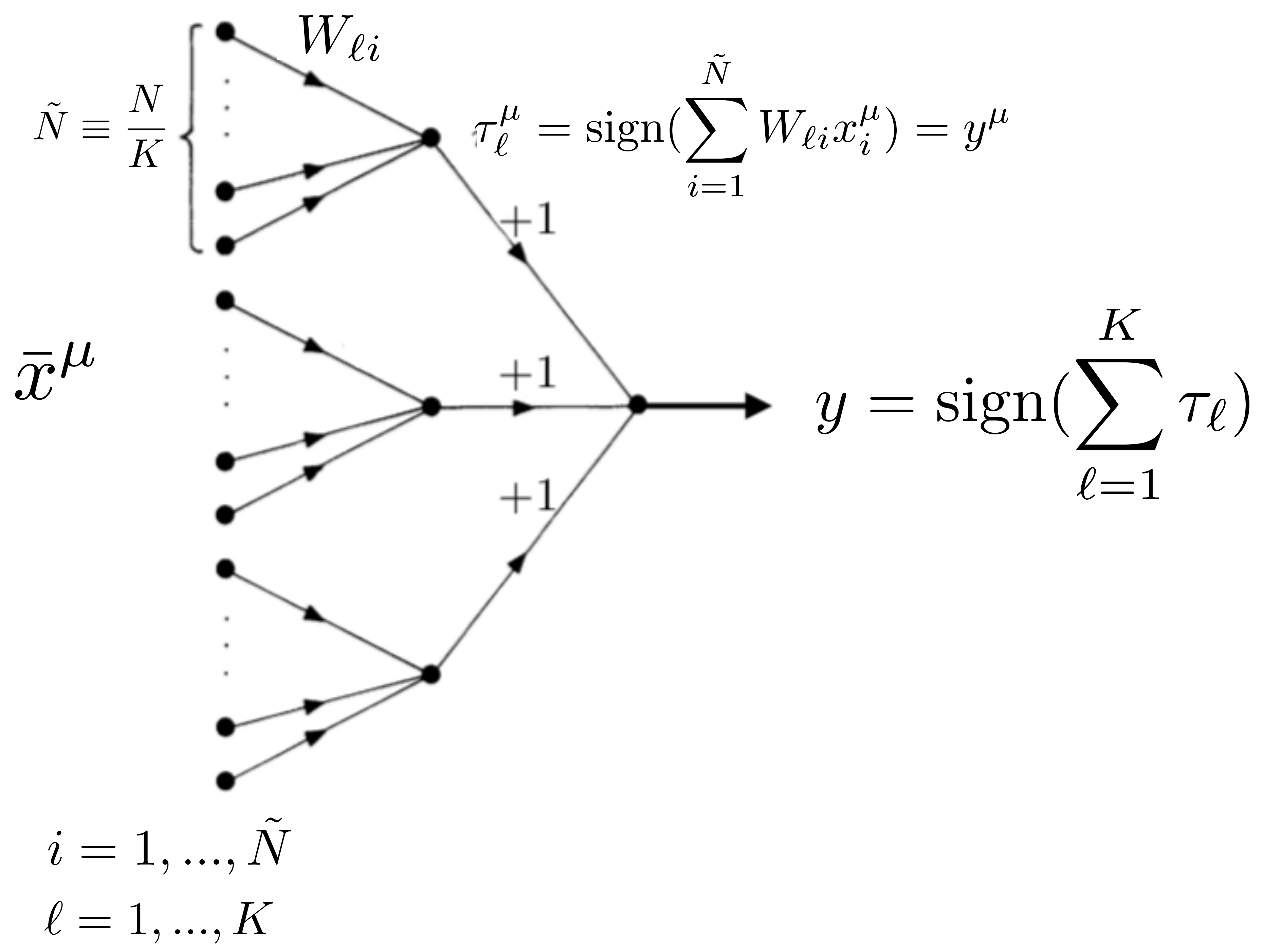
- Surrogate differentiable energies

Mean Square error
$$\mathcal{L}_{MSE} = \sum_{\mu} (\hat{\Delta}^{\mu} - \Delta^{\mu})^2$$

Cross-entropy: softmax
$$\mathcal{L}_{CE} = -\sum_{\mu} (\hat{\Delta}^{\mu}_{y^{\mu}} - \log \sum_{k} \exp \gamma \hat{\Delta}^{\mu}_{k})$$

$$\Delta^{\mu} \circeq (W^{K} \cdot \sigma_{K}(x^{\mu}))\, y^{\mu}$$

Simplest non convex neural device : 1-hidden layer, i.i.d. random associations



training set: $\{(\bar{x}^\mu, y^\mu)\} \quad \mu = 1, ..., P = \alpha N$

$$x_{\ell i}^\mu = \pm 1 \;\; (i.i.d. \;\; p = 1/2)$$

$$y^\mu = \pm 1 \;\; (i.i.d. \;\; p = 1/2)$$

control parameter: $\alpha = \dfrac{\#\ \text{patterns}}{\#\ \text{weights}}$

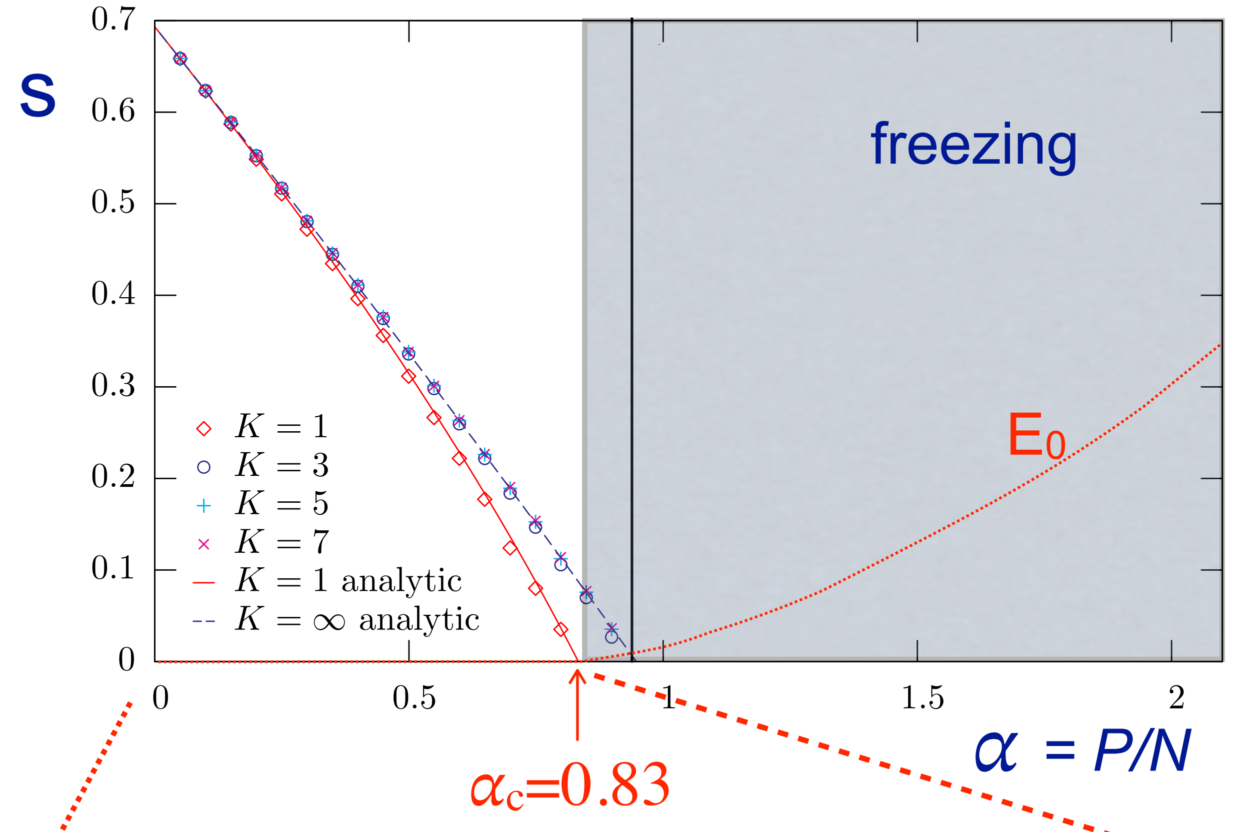
Non convex also for K=1

Results generalise to networks with continuous weights

Learning in the K=1 binary perceptron

In the large N, P limit (with $\alpha \equiv P/N$ fixed):

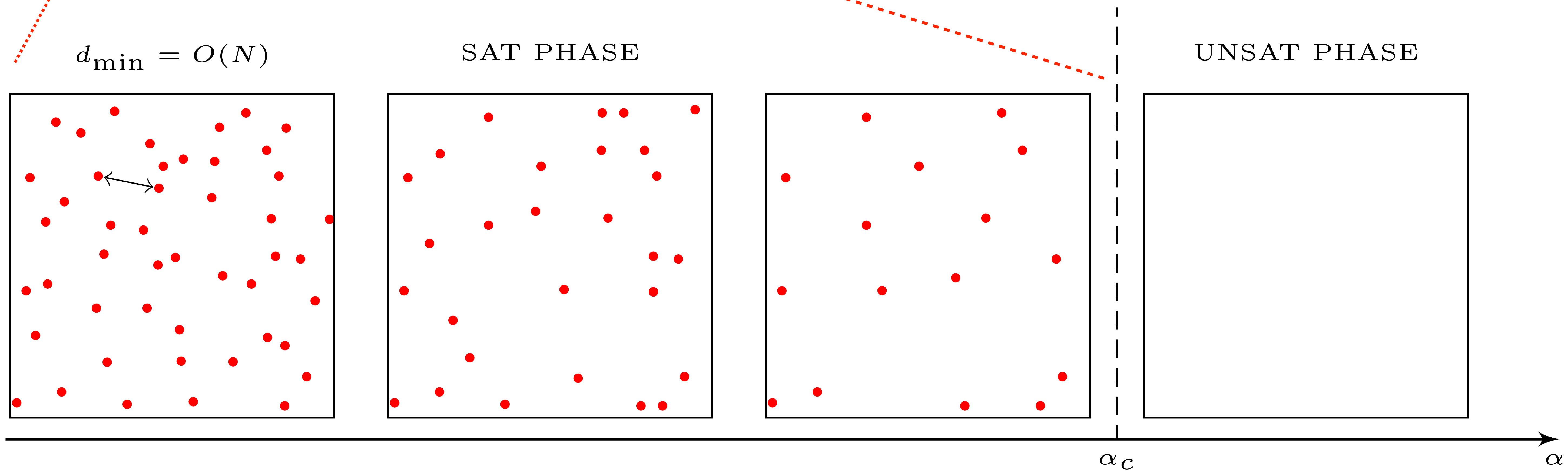
- the space of solution splits into separated states of vanishing entropy (Gardner, Derrida, (1988); Krauth, Mézard (1989));
- $\forall \alpha > 0$ typical solutions are isolated (Huang, Kabashima (2014));
- Rigorous Proofs: Abbe, Li, Sly (2021), Perkins, Xu (2021), Nakajima, Sun (2022).



some classical papers:

E. Gardner, E. Gardner B. Derrida, +

W. Krauth, M. Mézard, *J. de Physique* **50**, 3057-3066 (1989) ;E. Barkai, D. Hansel, H. Sompolinsky, *Phys. Rev. A* **45**, 4146-4160 (1992) ; M. Mezard, J. Phys. A 22, 2181 (1989); H.S. Seung, H. Sompolinsky, N. Tishby,Phys. Rev. A 45, 6056 (1992); E. Barkai, I. Kanter, Europhys. Lett14, 107 (1991); R. Penney and D. Sherrington, J. Phys. A 26, 6173(1993)
M. Tsodyks , *Mod.Phys. Lett. B* 4, 713 (1990); D.J. Amit, S. Fusi *NETWORK* 3, 443 (1992); D.J. Amit, S. Fusi, *Neural Computation* **6**, 957-982 (1994);
H. Horner, *Z. Phys. B* **86**, 291-308 (1992)
...



What has early Stat. Mech. brought to the field?

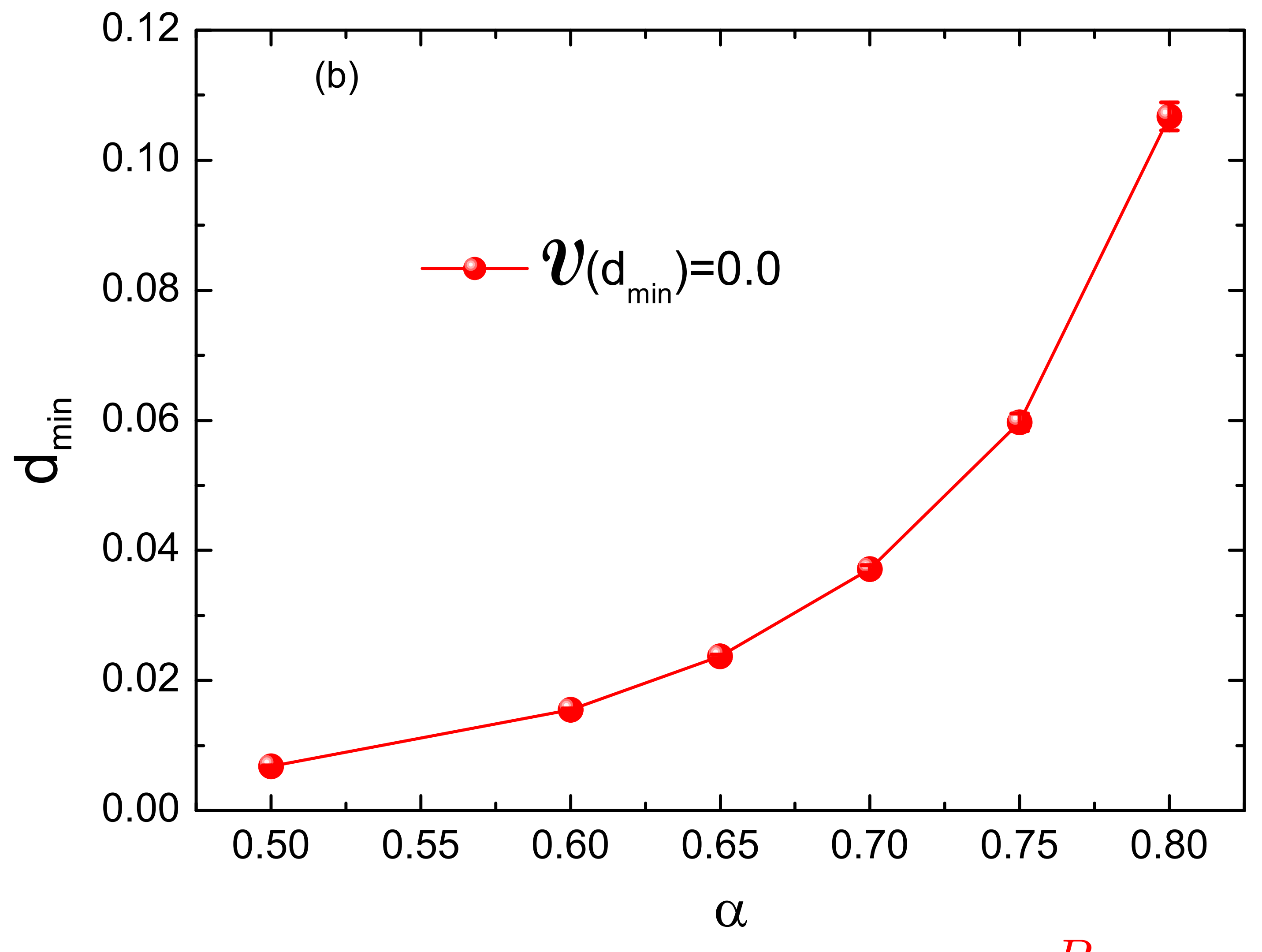
- Phase transitions
- Probabilities in large dimensions
- Dynamics
- New algorithms for convex perceptrons
- …

Geometry of the space of solutions:

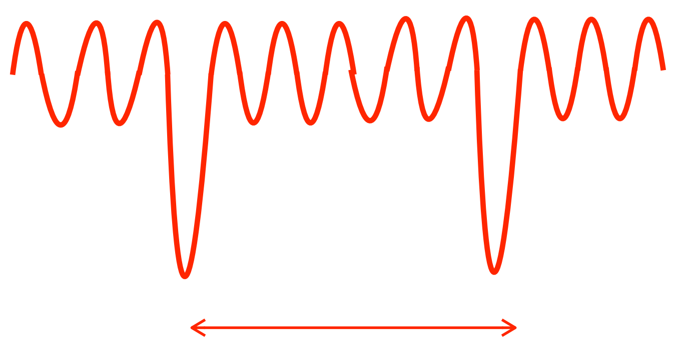
Franz-Parisi potential: entropy at distance **d**, sampling from typical solution **J**

$$F(x) = \left\langle \frac{1}{Z(T')} \sum_{\mathbf{J}} \Theta\left(\frac{1}{\sqrt{N}} \sum_{i=1}^{N} J_i \xi_i^\mu\right) \ln \sum_{\mathbf{w}} \Theta\left(\frac{1}{\sqrt{N}} \sum_{i=1}^{N} w_i \xi_i^\mu\right) e^{x \mathbf{J} \cdot \mathbf{w}} \right\rangle_{\boldsymbol{\xi}}$$

$d_{\min}(\alpha)$

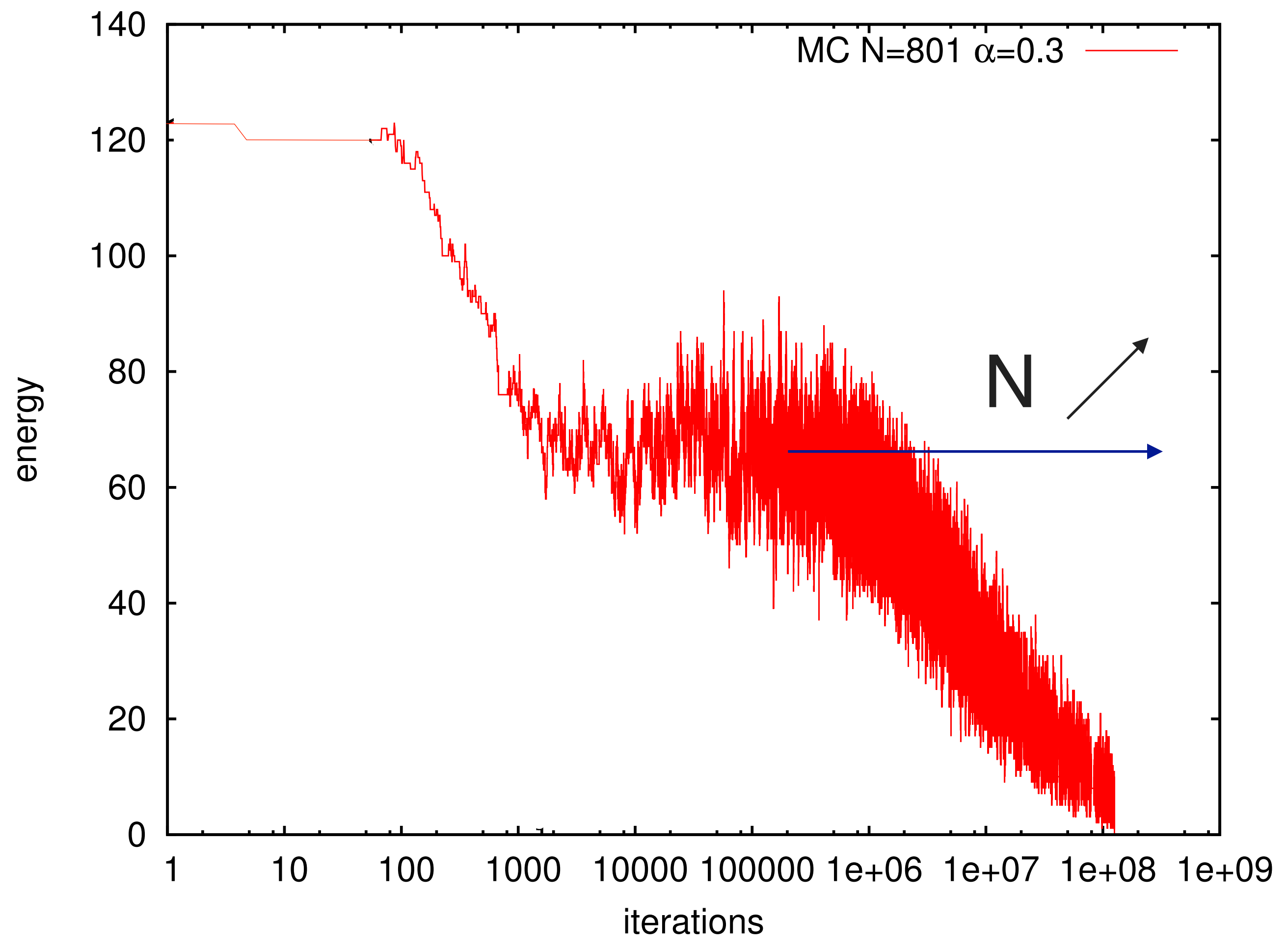


$\alpha_c = \frac{P_{max}}{N} \simeq 0.83$



$d_{\min}(\alpha) \sim O(N)$

H. Huang, Y. Kabashima (2014)



Sampling from the Gibbs distribution is not a good algorithm (as expected)

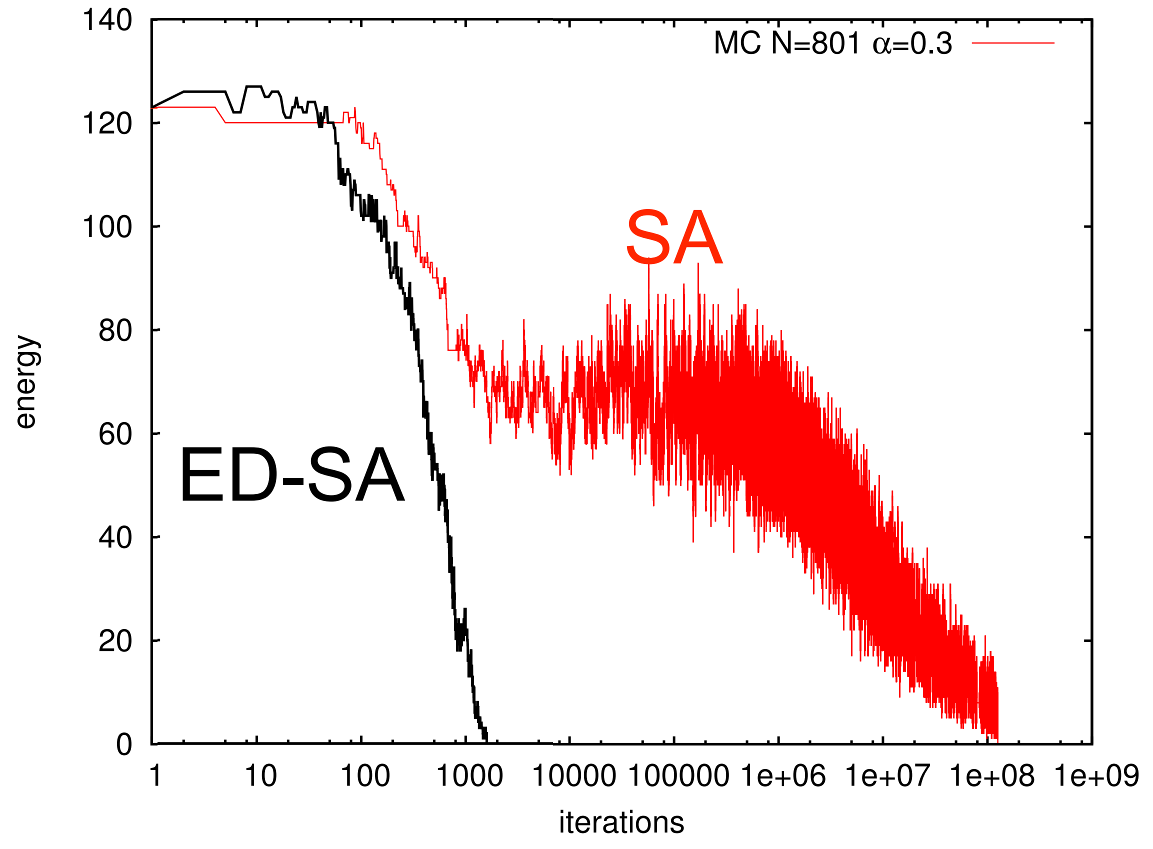
Golf course for any **α** ? Efficient learning impossible ?

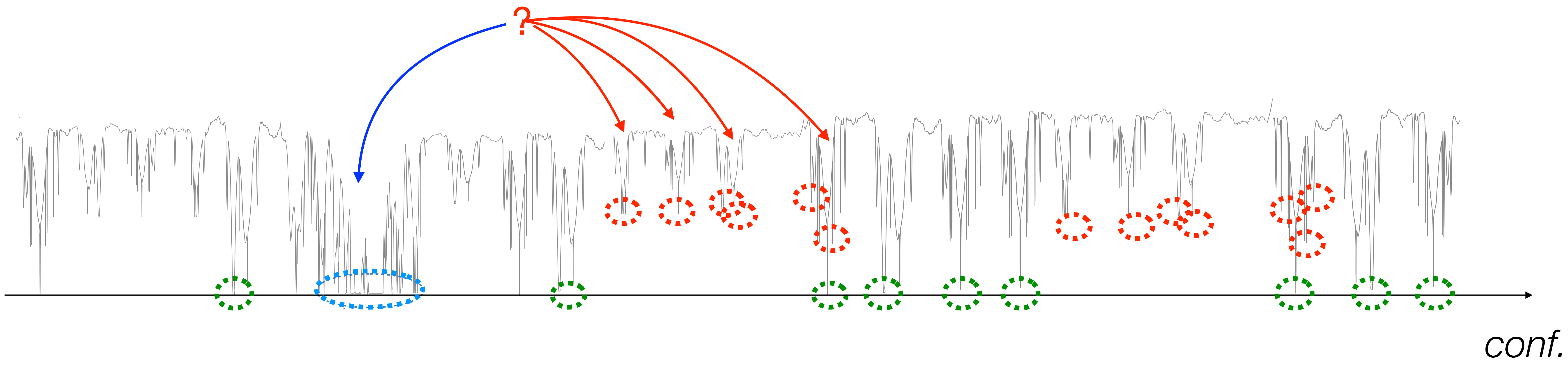
However other algorithms find solution efficiently!

The learning problem is predicted to be typically computationally difficult

- Typical global minima are isolated (mutual distance of O(N))
- Glassy landscape: exponentially many local minima
- Learning should be hopeless

This contradicts empirical evidence!





Braunstein, Z. PRL 2006

Local entropy measure: large deviation 1-RSB techniques

Bias the statistical measure towards dense (wide, flat) regions (large deviation analysis)

$$\mathcal{N}\left(\tilde{W}, d\right) = \sum_{\{W\}} \mathbb{X}_\xi\left(W\right) \delta\left(W \cdot \tilde{W}, N\left(1 - 2d\right)\right)$$

solution at distance d

$$\mathbb{X}_\xi\left(W\right) = \lim_{\beta \to \infty} e^{-\beta \mathcal{L}_{NE}(W)}$$

indicator function

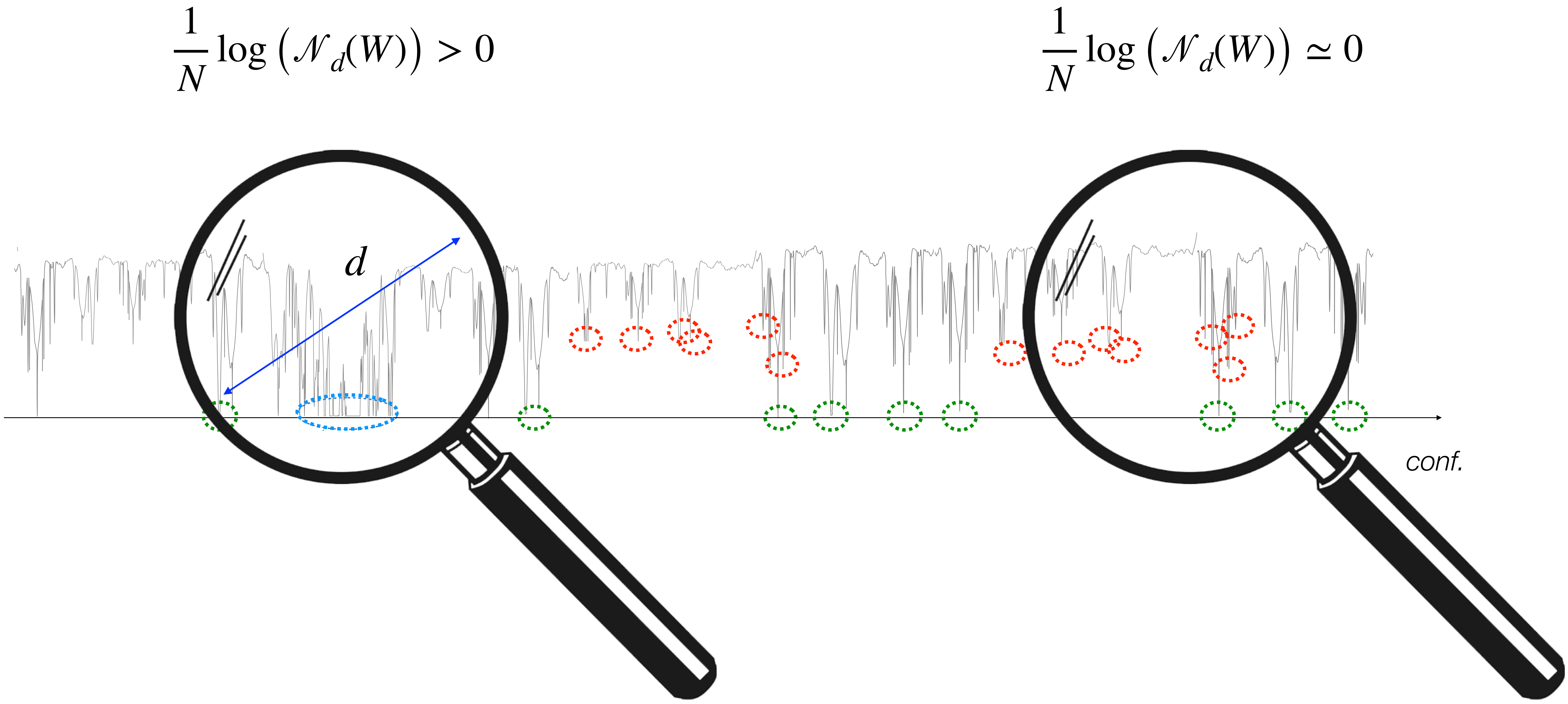
$$\boxed{\mathcal{E}_d(\tilde{W}) \doteq -\log \mathcal{N}(\tilde{W}, d)}$$

"**local entropy**" (the log of the number of solutions in hypersphere of radius *d*)

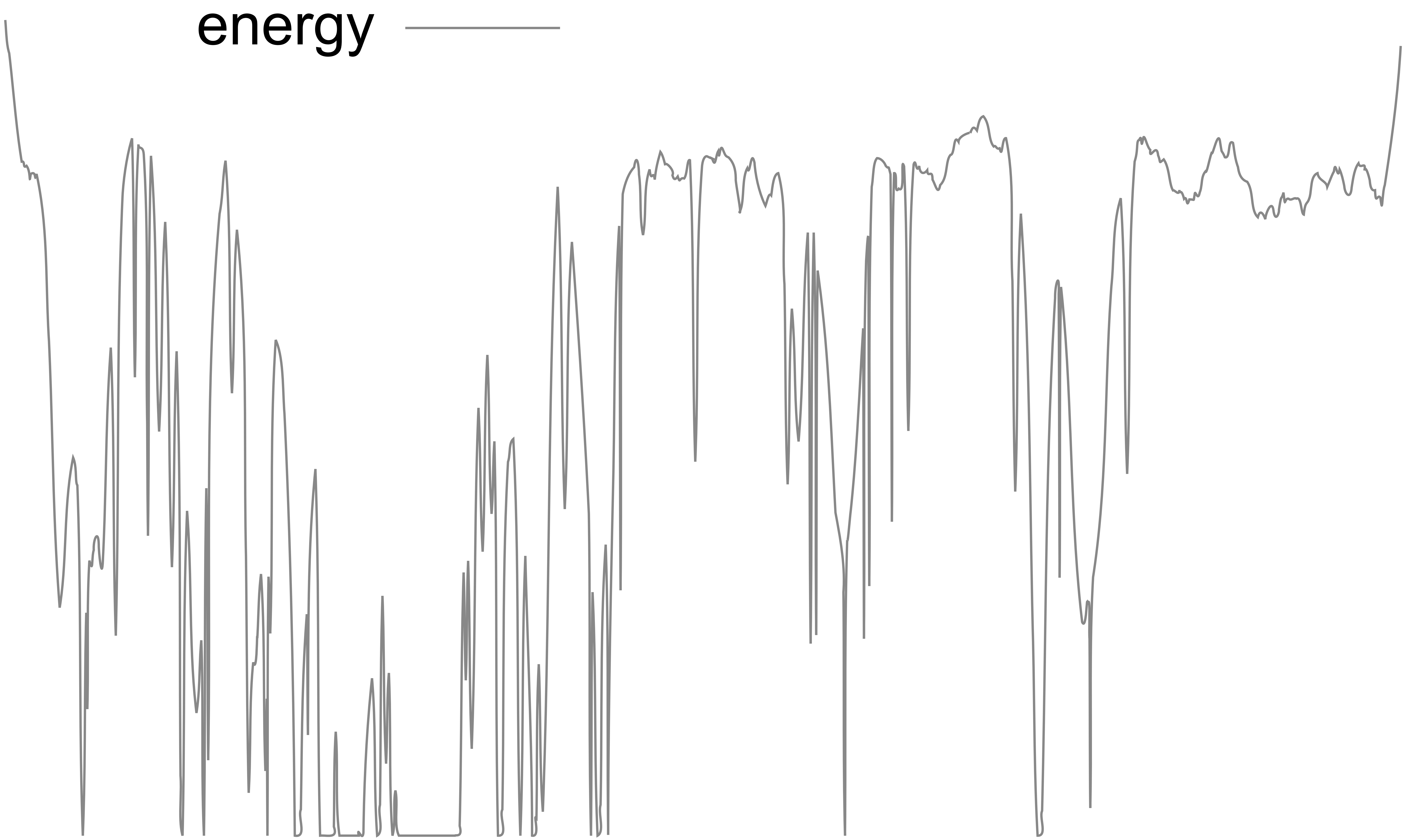
$$\boxed{\Phi(\tilde{W}, \beta, \gamma) = \ln \sum_{\{W\}} e^{-\beta \mathcal{L}_{NE}(W) - \gamma d(\tilde{W}, W)}}$$

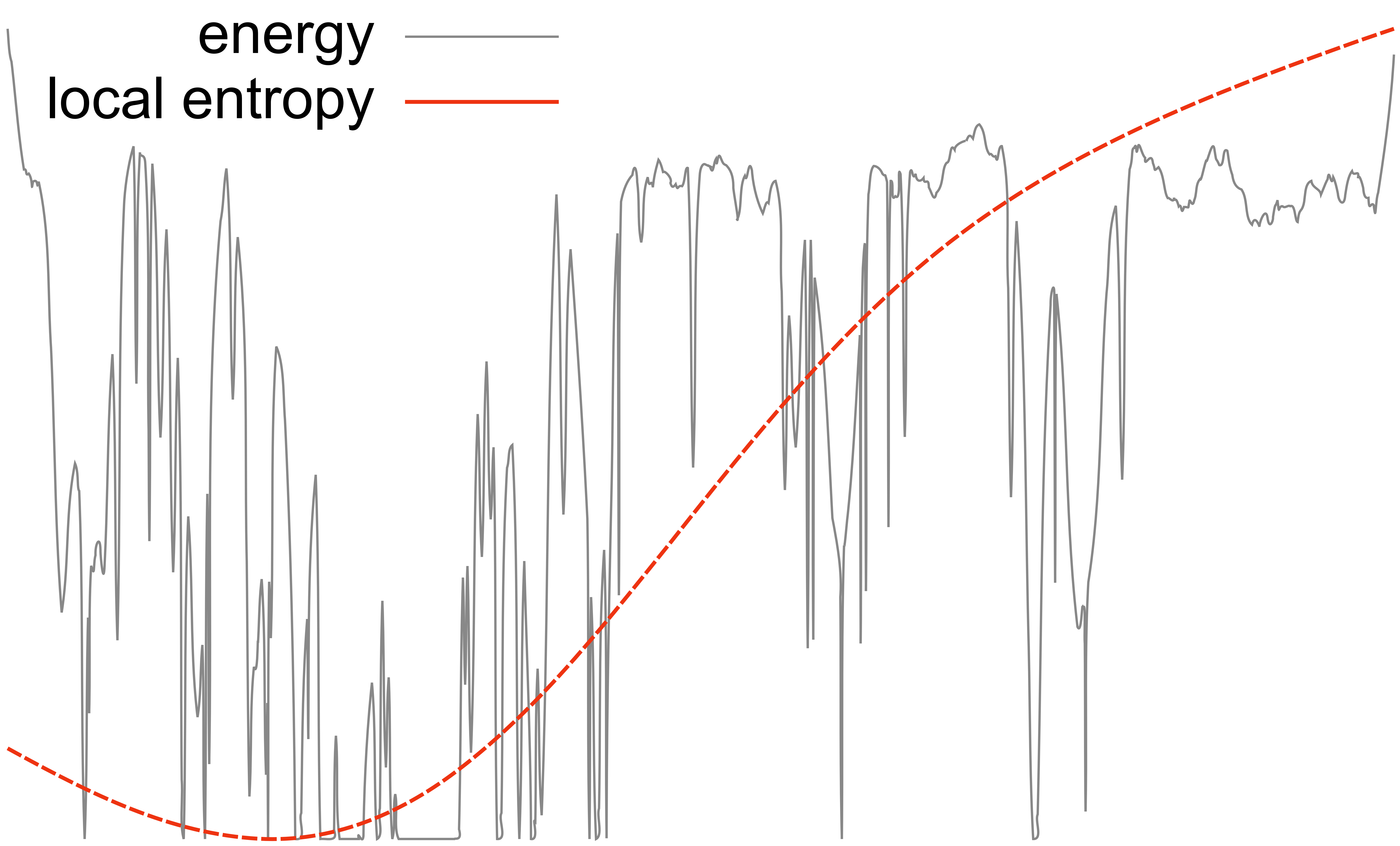
"**local free entropy**"

C. Baldassi, A. Ingrosso, C. Lucibello, L. Saglietti, and R. Zecchina Phys. Rev. Lett. 115, 128101 (2015)

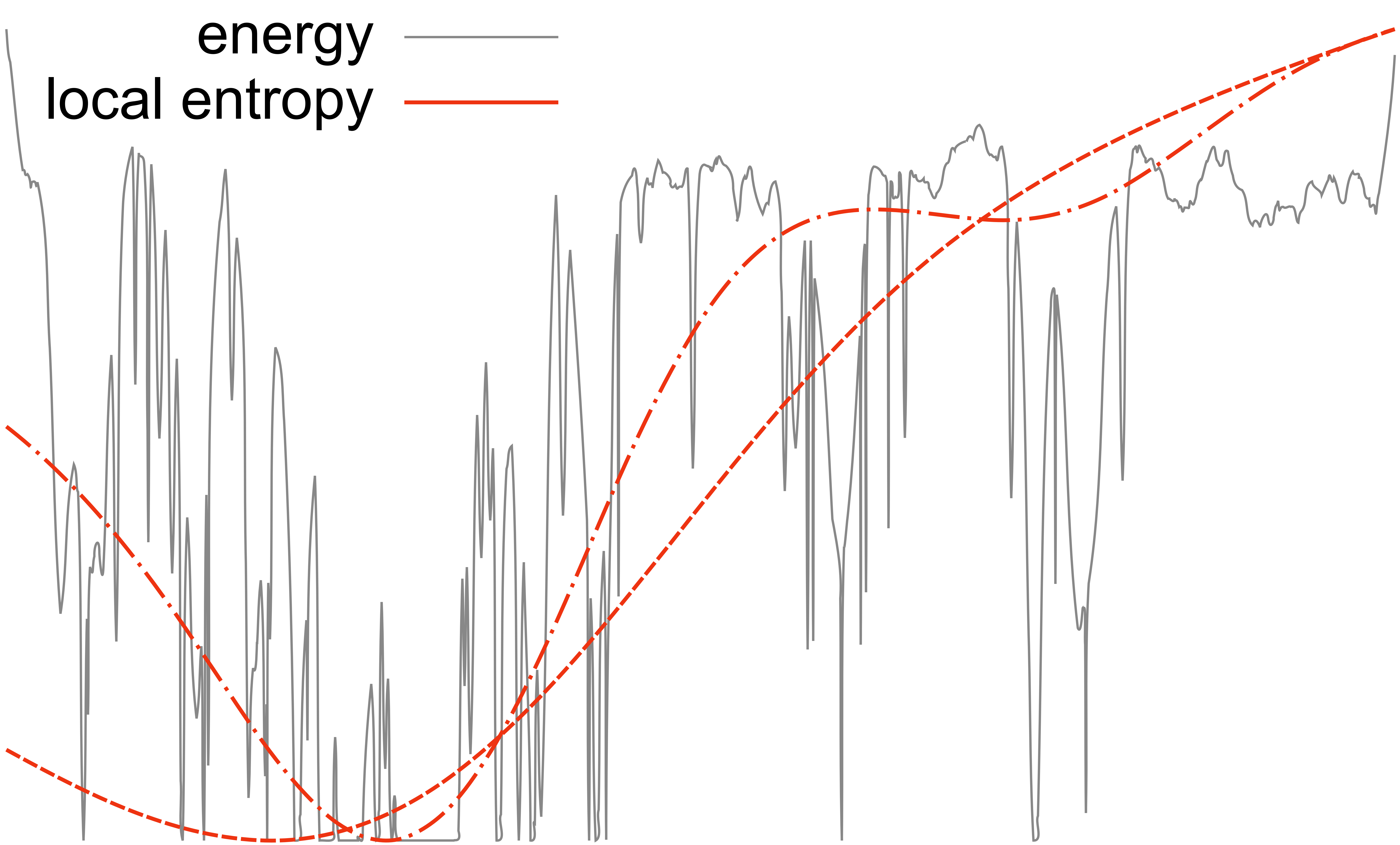


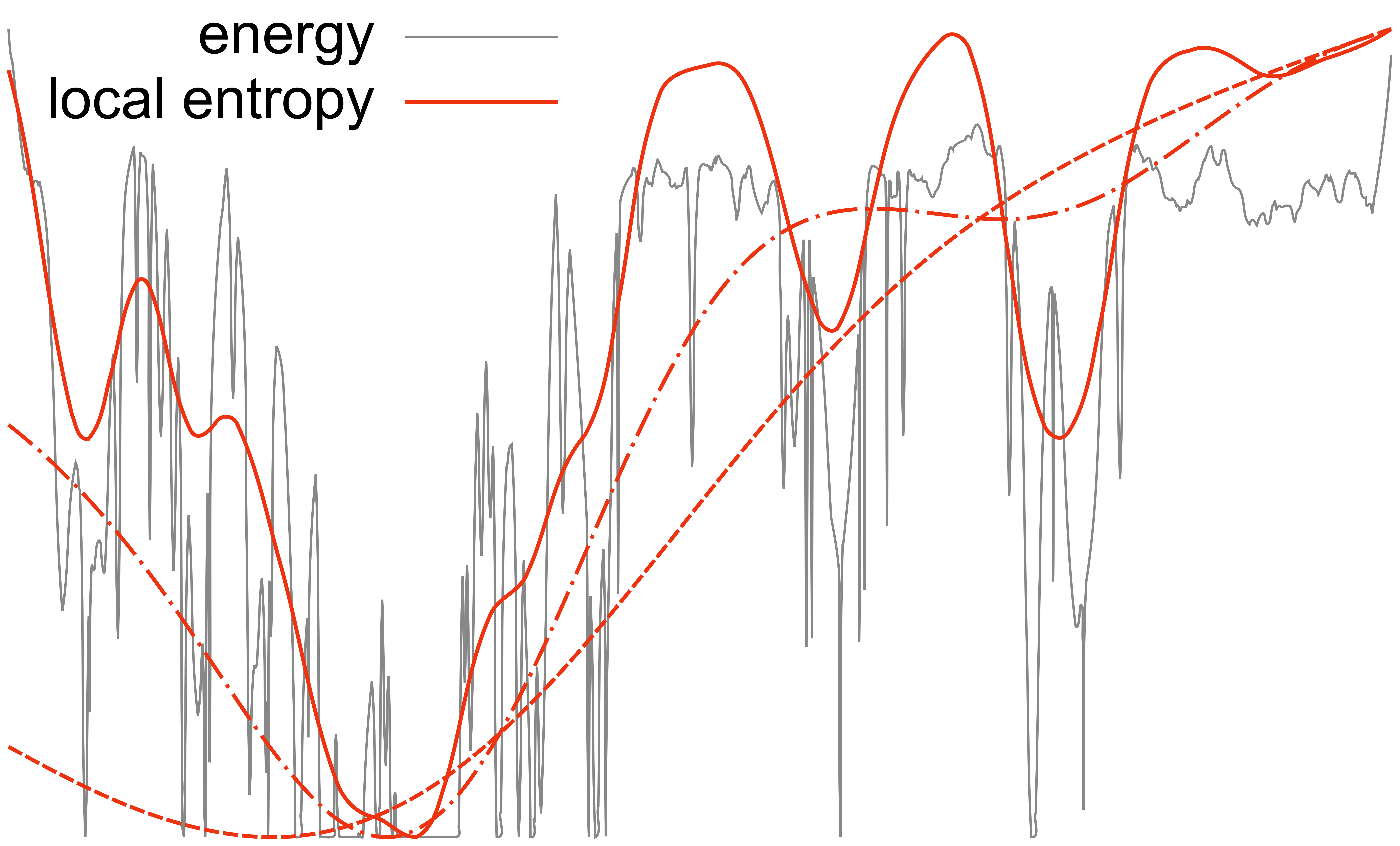
$\frac{1}{N}\log\left(\mathcal{N}_d(W)\right) > 0$
$\frac{1}{N}\log\left(\mathcal{N}_d(W)\right) \simeq 0$
d
conf.


energy



energy
local entropy

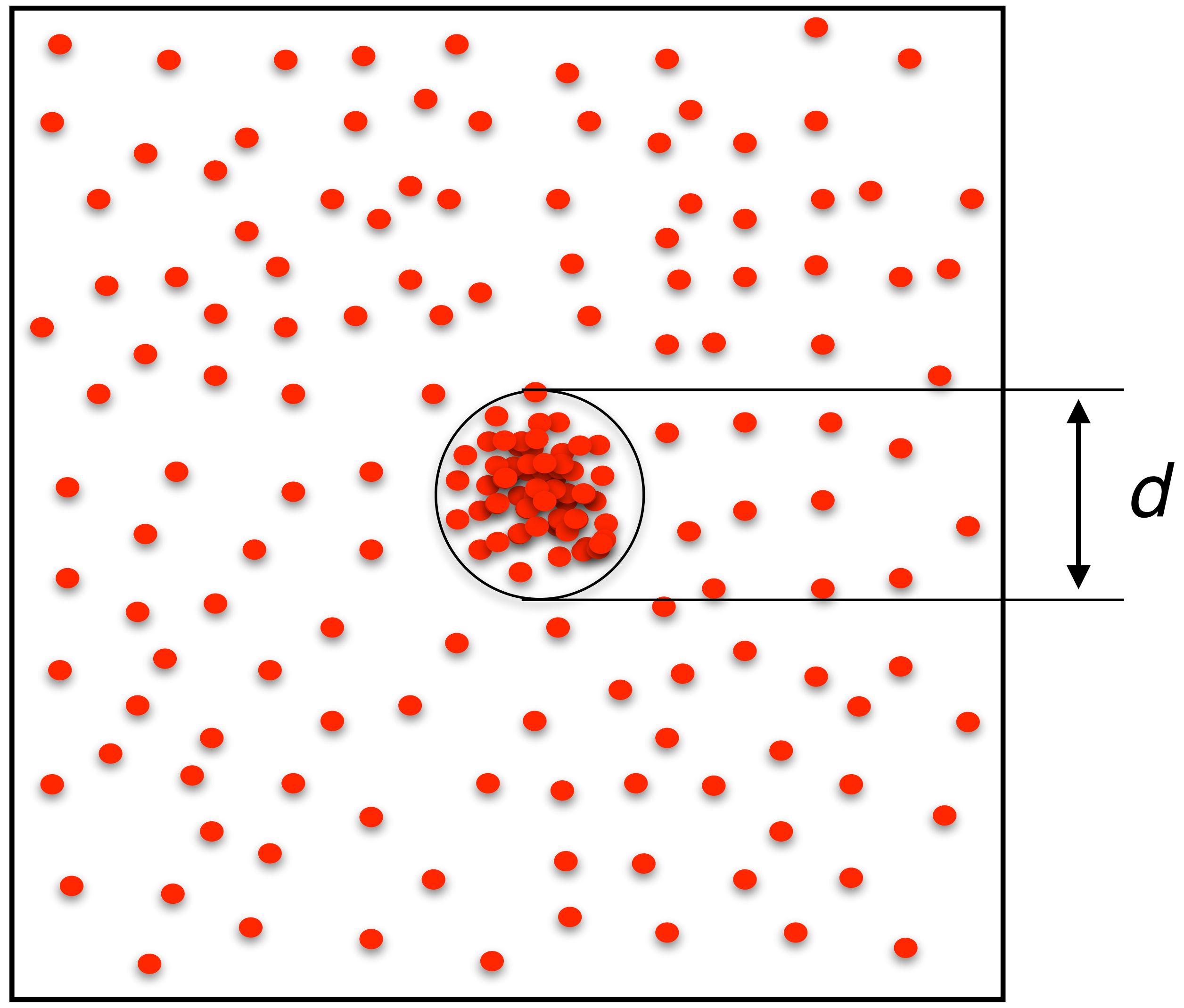

energy
local entropy


energy
local entropy

Check the existence of subdominant dense regions of solutions in the Binary Perceptron

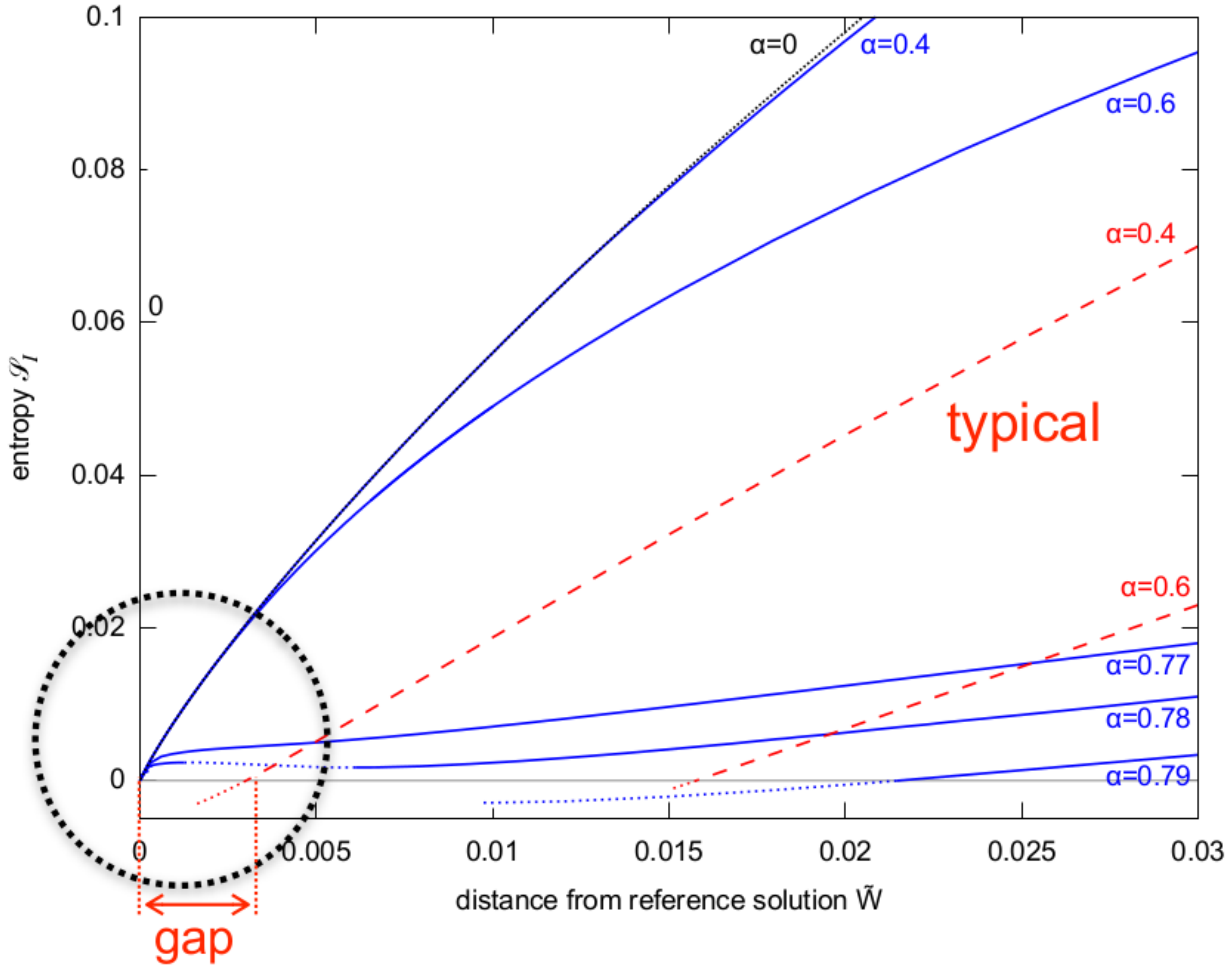
Finite temperature version (not only zero error states) : free local entropy

$$P(\tilde{W}) = \frac{e^{y\Phi(\tilde{W},\beta,\gamma)}}{Z} \qquad P(\tilde{W}) = \frac{e^{-y\mathcal{E}_d(\tilde{W},d)}}{Z} \quad (\beta \to \infty)$$

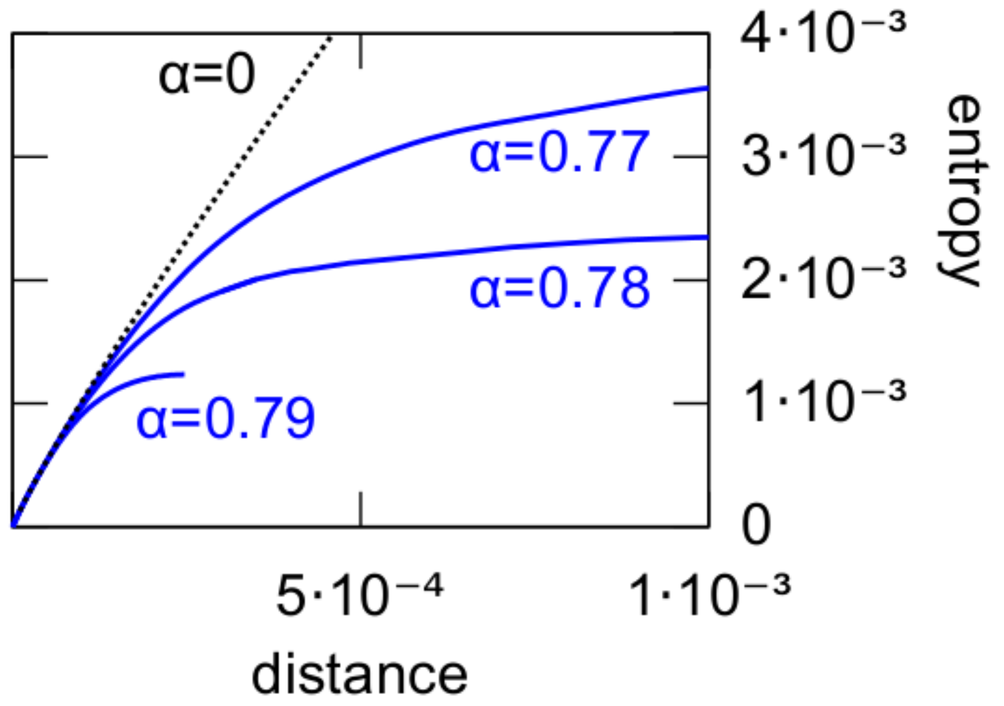


Find $\tilde{W}$ that maximises the local free entropy the: $\operatorname{argmin}_{\tilde{W}} \langle \Phi(\tilde{W},\beta,\gamma) \rangle$

$$P(\tilde{W}) = \lim_{y,\beta\to\infty} \frac{e^{y\Phi(\tilde{W},\beta,\gamma)}}{Z}$$



ultra-dense cluster



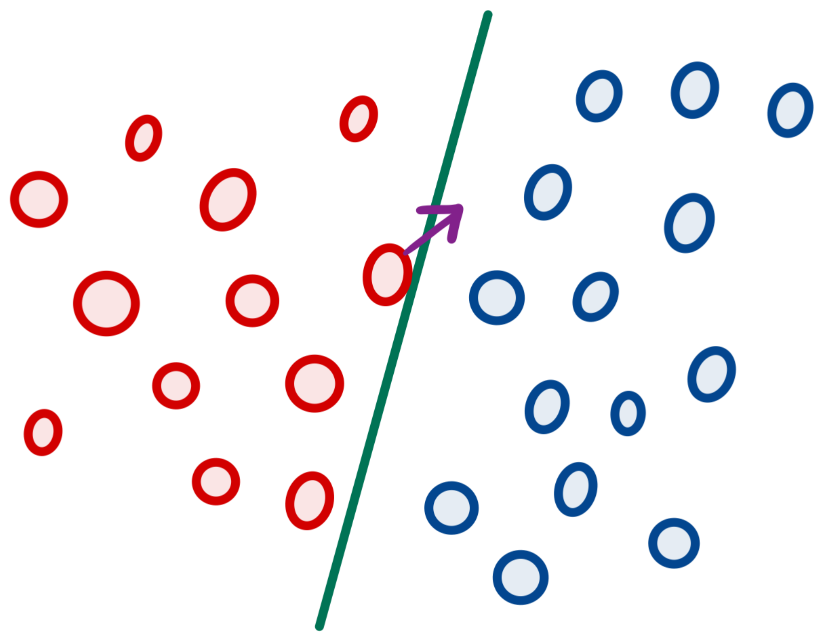
Geometrical phase discontinuous transition

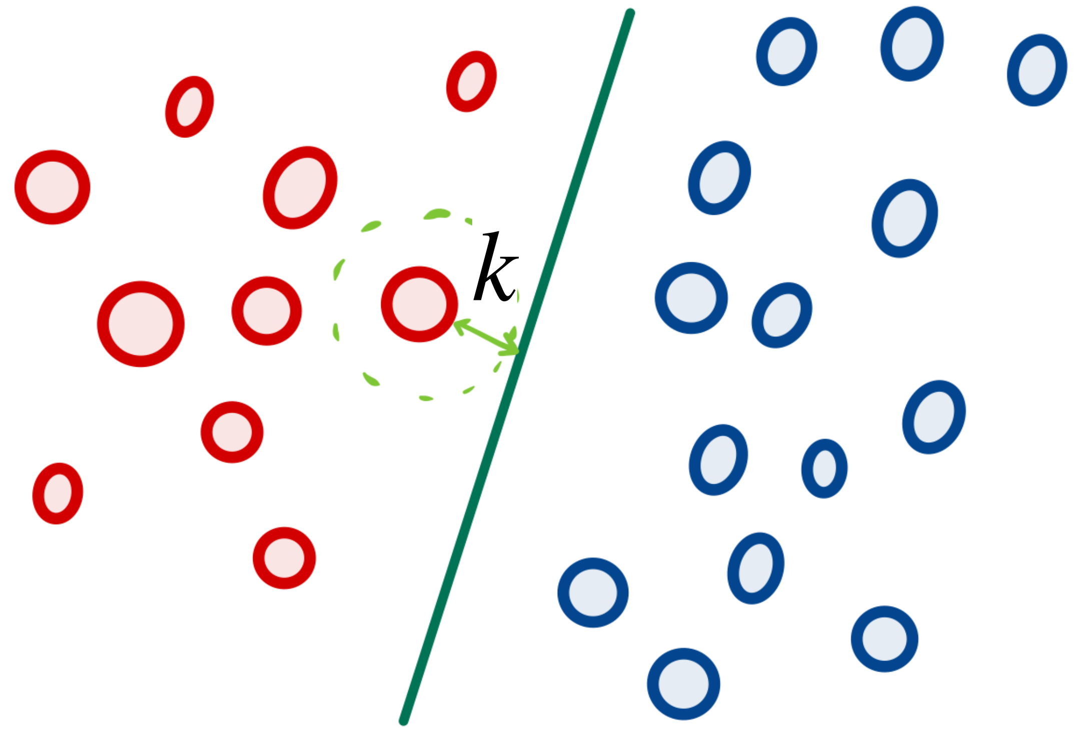
$$\alpha_u \simeq 0.77$$

How to connected the results with the “traditional” maximum margin studies ?

Solutions with margin k:

$$\mathbb{X}_{\xi,F}(W;\kappa) = \prod_{\mu=1}^{P} \Theta\left(y^\mu \sigma^\mu_{\mathrm{out}} - \kappa\right)$$



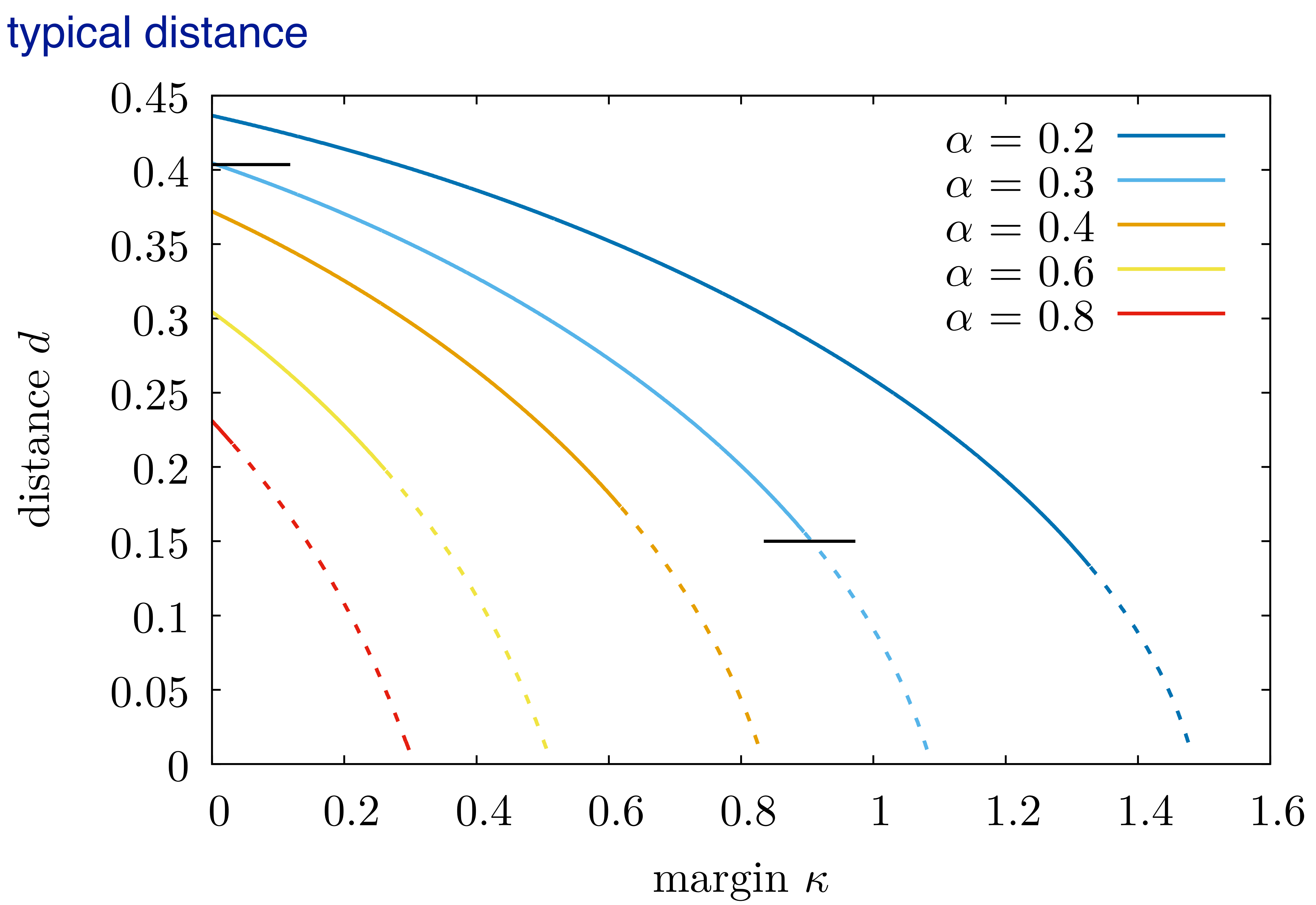


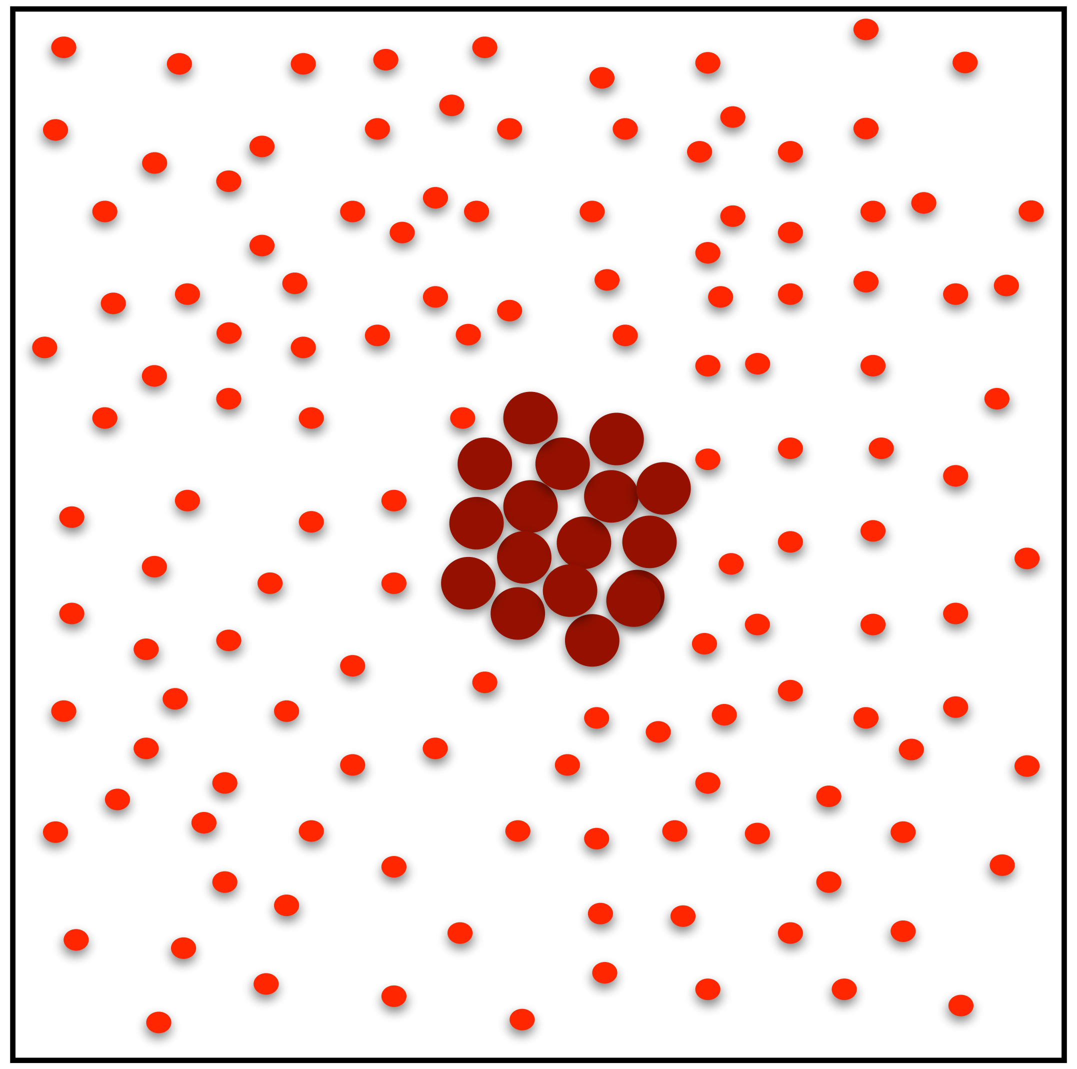
$$E = \sum_{\mu=1}^{P} \Theta\left(-y^\mu \sigma^\mu_{\mathrm{out}}\right)$$

$$E = \sum_{\mu=1}^{P} \Theta\left(-y^\mu \sigma^\mu_{\mathrm{out}} + \kappa\right)$$

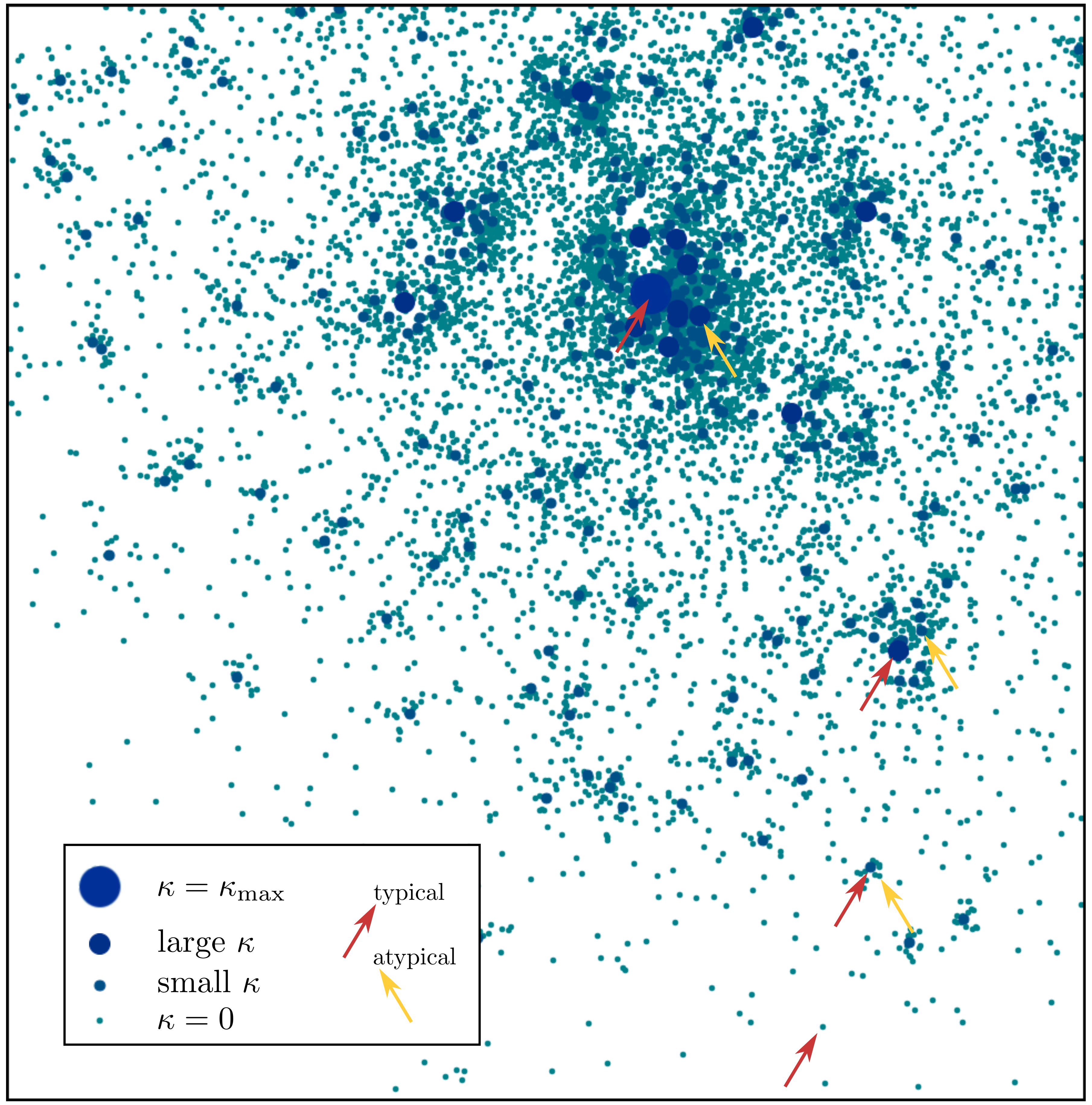
High margin solutions are less but tend to be much closer to each other!

The lines change from solid to dashed when the entropy of solutions becomes negative, i.e. when $\kappa = \kappa_{max}$





A wide flat minima arises by the coalescence of (atypical) high margin minima!

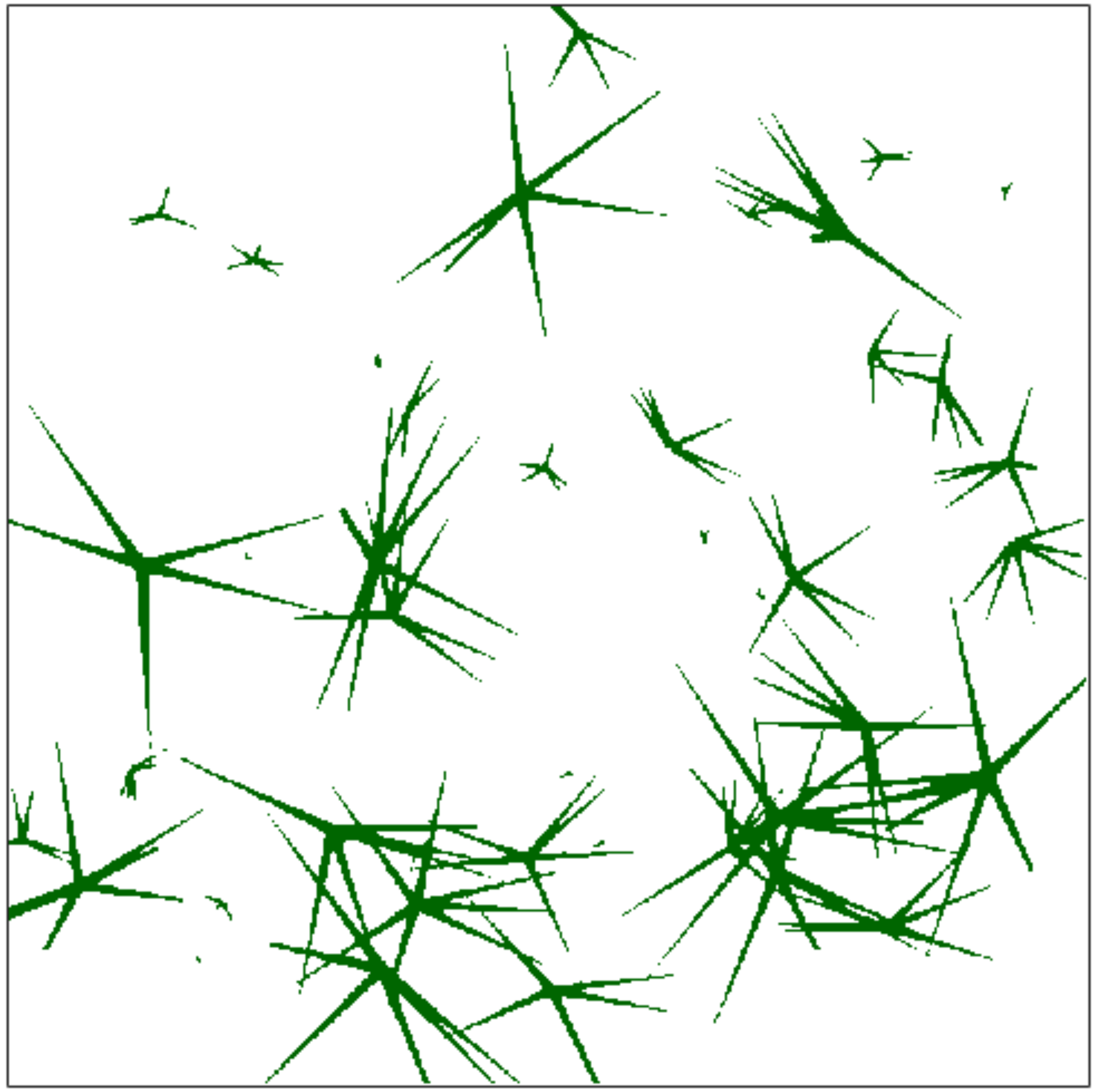


Unveiling the structure of wide flat minima in neural networks

Carlo Baldassi,[1] Clarissa Lauditi,[2] Enrico M. Malatesta,[1] Gabriele Perugini,[1] and Riccardo Zecchina[1]

[1] *Artificial Intelligence Lab, Bocconi University, 20136 Milano, Italy*
[2] *Department of Applied Science and Technology, Politecnico di Torino, 10129 Torino, Italy*

arXiv:2107.01163 PRL, 2021



Binary perceptron: efficient algorithms can find solutions in a rare well-connected cluster

Emmanuel Abbe * Shuangping Li † Allan Sly ‡

arXiv:2111.03084

Algorithmic follow-up

Local free entropy: $$\phi(W,\gamma,\beta) = \log \sum_{W'} e^{-\beta \mathcal{L}_{NE}(W') - \frac{\gamma}{2} d(W,W')}$$

Large-deviation partition function: $$Z(y,\gamma,\beta',\beta) = \sum_{W} e^{-\beta' \mathcal{L}_{NE}(W) + y\,\phi(W,\gamma,\beta)}$$

Assume **y integer**: $$Z(y,\gamma,\beta',\beta) = \sum_{W,\{W_a\}} e^{-\beta' \mathcal{L}_{NE}(W) - \beta \sum_{a=1}^{y} \mathcal{L}_{NE}(W_a) - \frac{\gamma}{2} \sum_{a=1}^{y} d(W,W_a)}$$

interaction

center

y (real) replicas

Local entropy algorithms

- Local Entropy driven or replicated Simulated Annealing
- Replicated Message-Passing (Belief Propagation)
- Replicated Stochastic Gradient Descent (SGD)
- **Entropy-SGD:** Langevin dynamics to estimate local entropy
- Replicated Greedy Algorithms
- **Sharpness Aware Minimization**
- Stochastic weights +gradient on the probabilities
- Quantum Annealing delocalization mechanism for finding NN ground states
- …

some known algorithms for DNNs

PNAS ARTICLES FRONT MATTER AUTHORS TOPICS +

RESEARCH ARTICLE | COMPUTER SCIENCES

Unreasonable effectiveness of learning neural networks: From accessible states and robust ensembles to basic algorithmic schemes

Carlo Baldassi, Christian Borgs, Jennifer T. Chayes, +3, and Riccardo Zecchina Authors Info & Affiliations

Edited by William Bialek, Princeton University, Princeton, NJ, and approved October 14, 2016 (received for review May 20, 2016)

November 15, 2016 | 113 (48) E7655-E7662 | https://doi.org/10.1073/pnas.1608103113

Published as a conference paper at ICLR 2017

ENTROPY-SGD: BIASING GRADIENT DESCENT INTO WIDE VALLEYS

Pratik Chaudhari[1], Anna Choromanska[2], Stefano Soatto[1], Yann LeCun[3,4], Carlo Baldassi[5], Christian Borgs[6], Jennifer Chayes[6], Levent Sagun[3], Riccardo Zecchina[5]

Published as a conference paper at ICLR 2021

SHARPNESS-AWARE MINIMIZATION FOR EFFICIENTLY IMPROVING GENERALIZATION

Pierre Foret *
Google Research
pierre.pforet@gmail.com

Ariel Kleiner
Google Research
akleiner@gmail.com

Hossein Mobahi
Google Research
hmobahi@google.com

Behnam Neyshabur
Blueshift, Alphabet
neyshabur@google.com

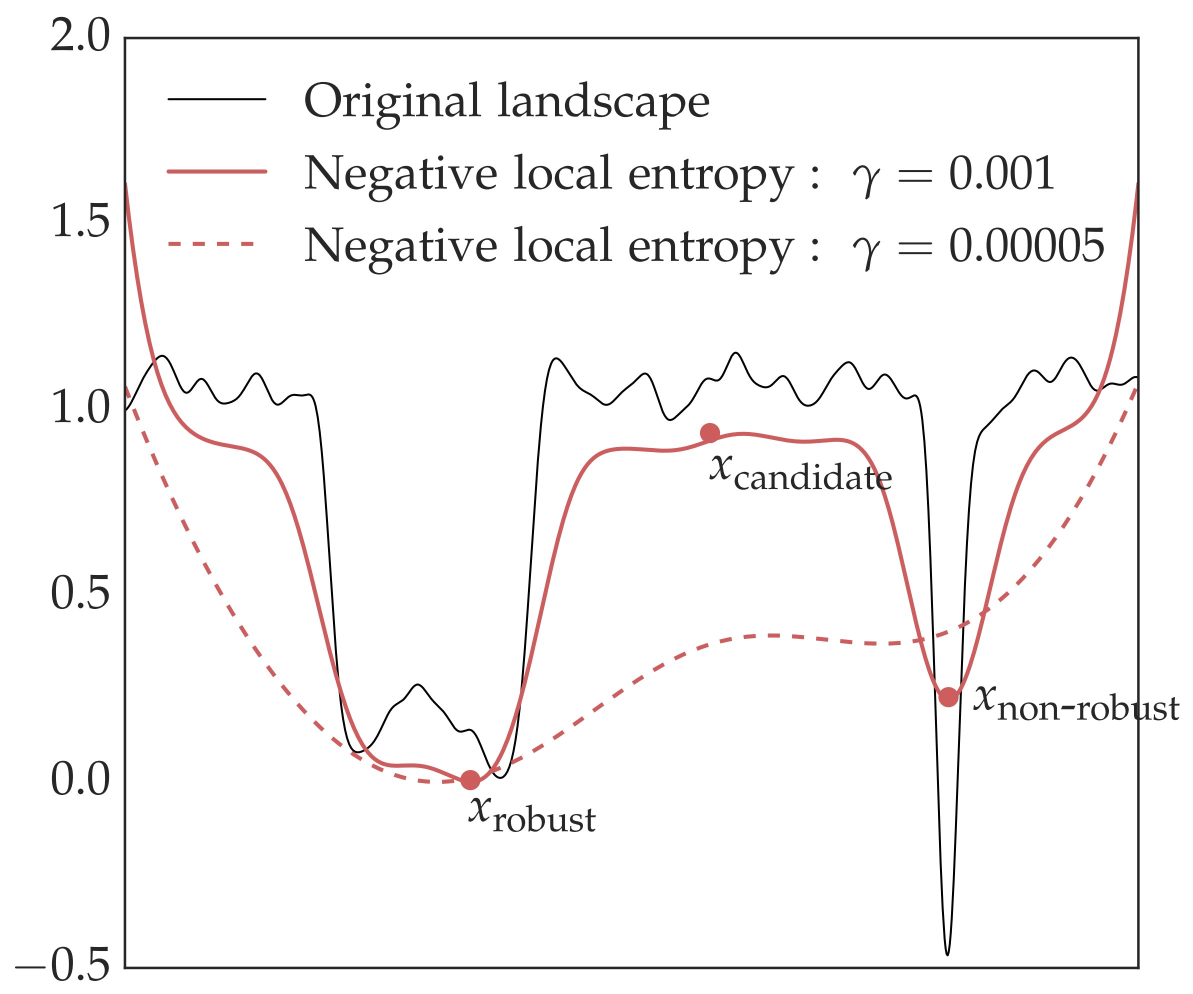
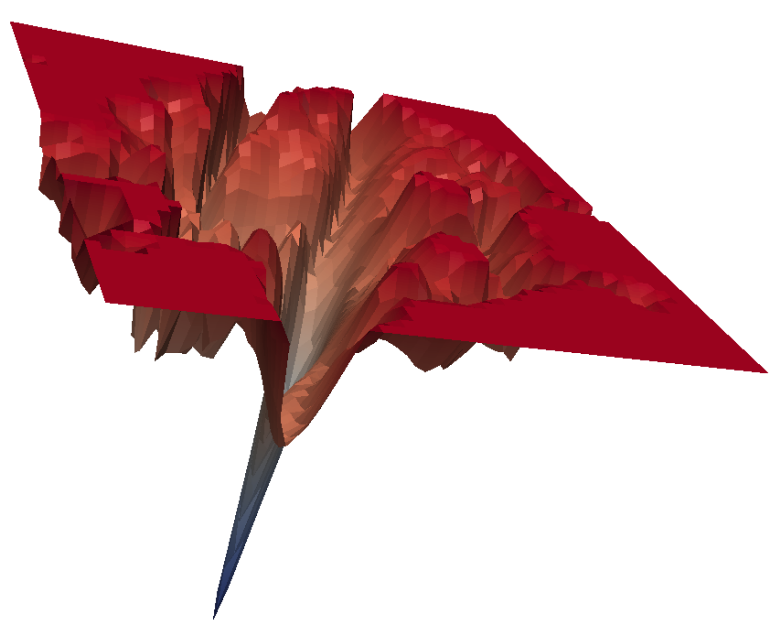
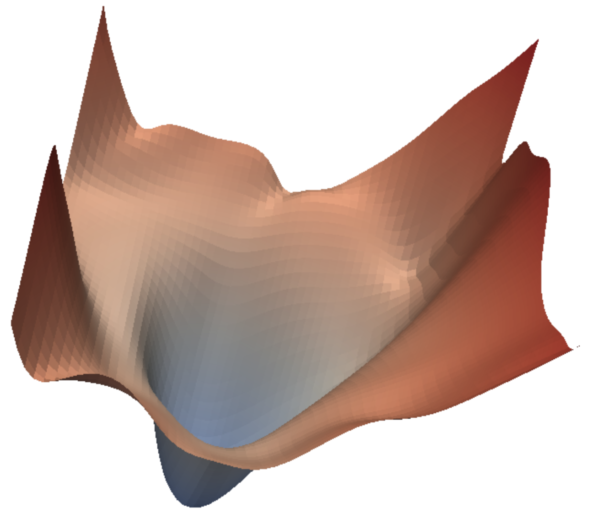


Figure 2: Local entropy concentrates on wide valleys in the energy landscape.



SGD



SAM

Similar analytical results hold for 1-hidden layer NN with continuous weights and for overparametrized NN

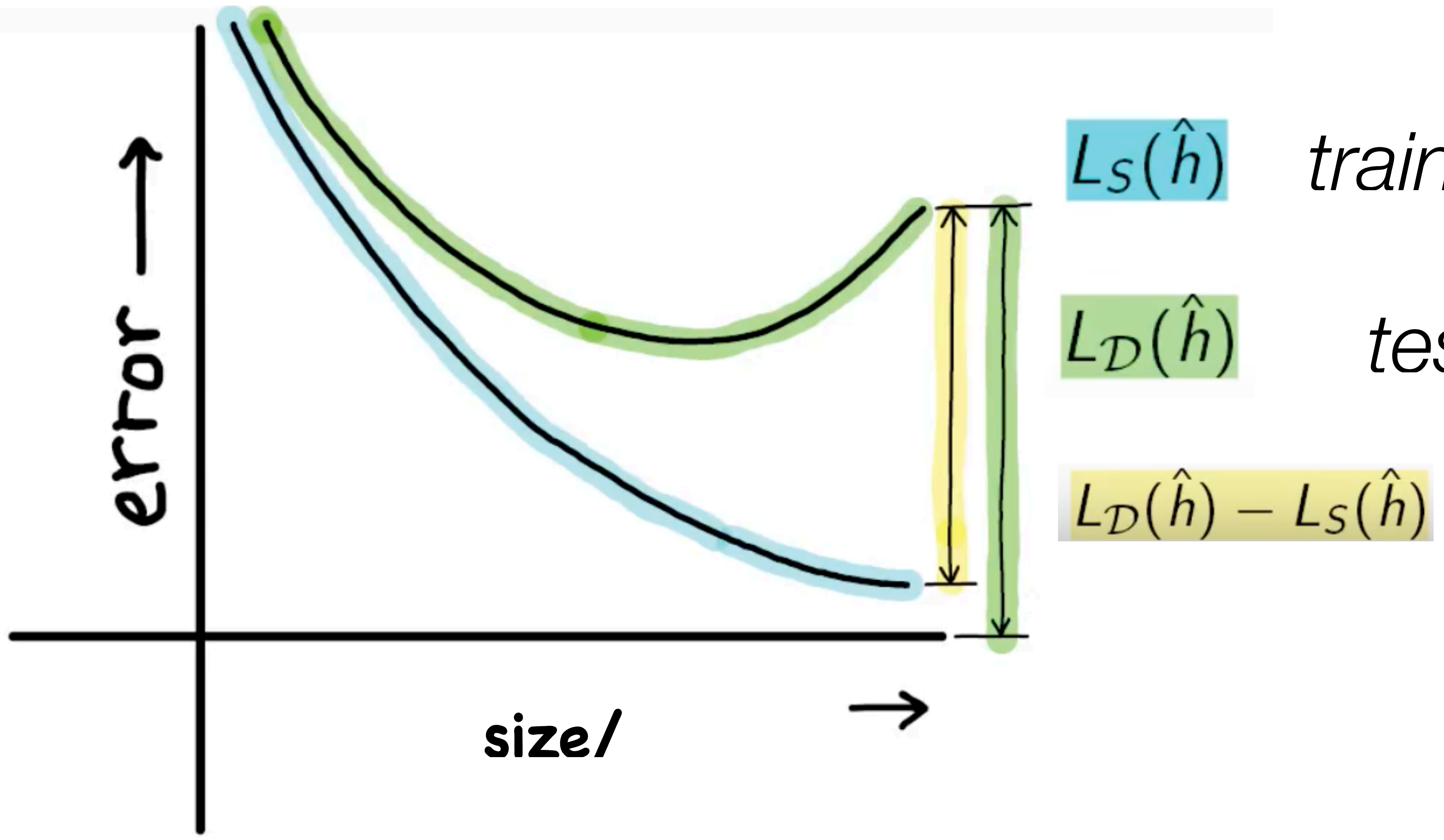
High Local Entropy regions ↔ Wide Flat Minima (WFM)

Analytical Results: Rare Wide Flat Minima (WFM) exist in non-convex networks with continuous weights storing random patterns.

Analytical/numerical results: they have good generalisation properties

(Baldassi, Pittorino, Zecchina, PNAS 2019)

Classical overfitting problem



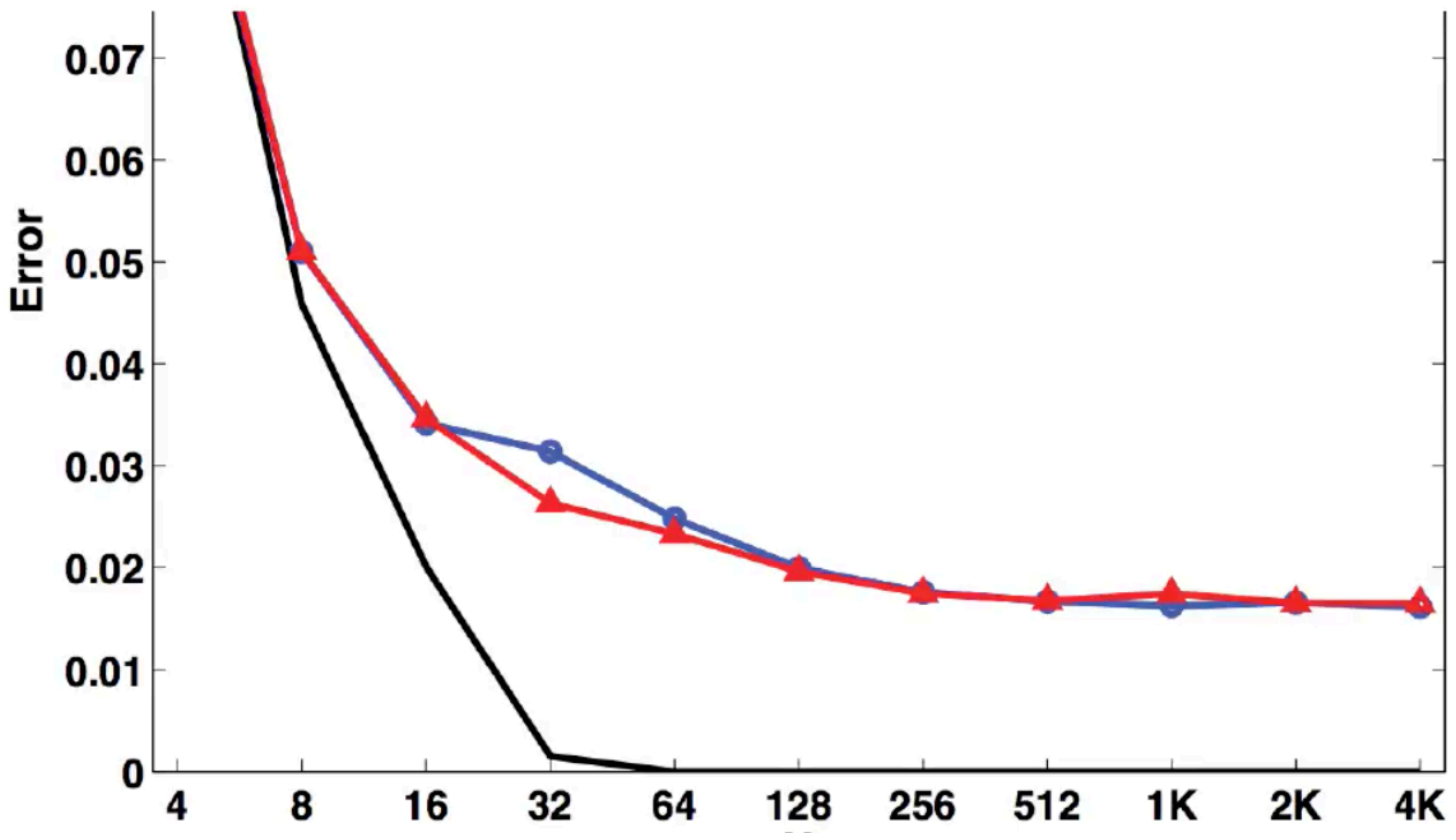
training

test

generalization

Overfitting under control in DNNs !?

Using **stochastic gradient descent (SGD)**, trained networks of increasing size on MNIST until convergence.



Training error at convergence.

Test error at convergence and for **early stopping**.

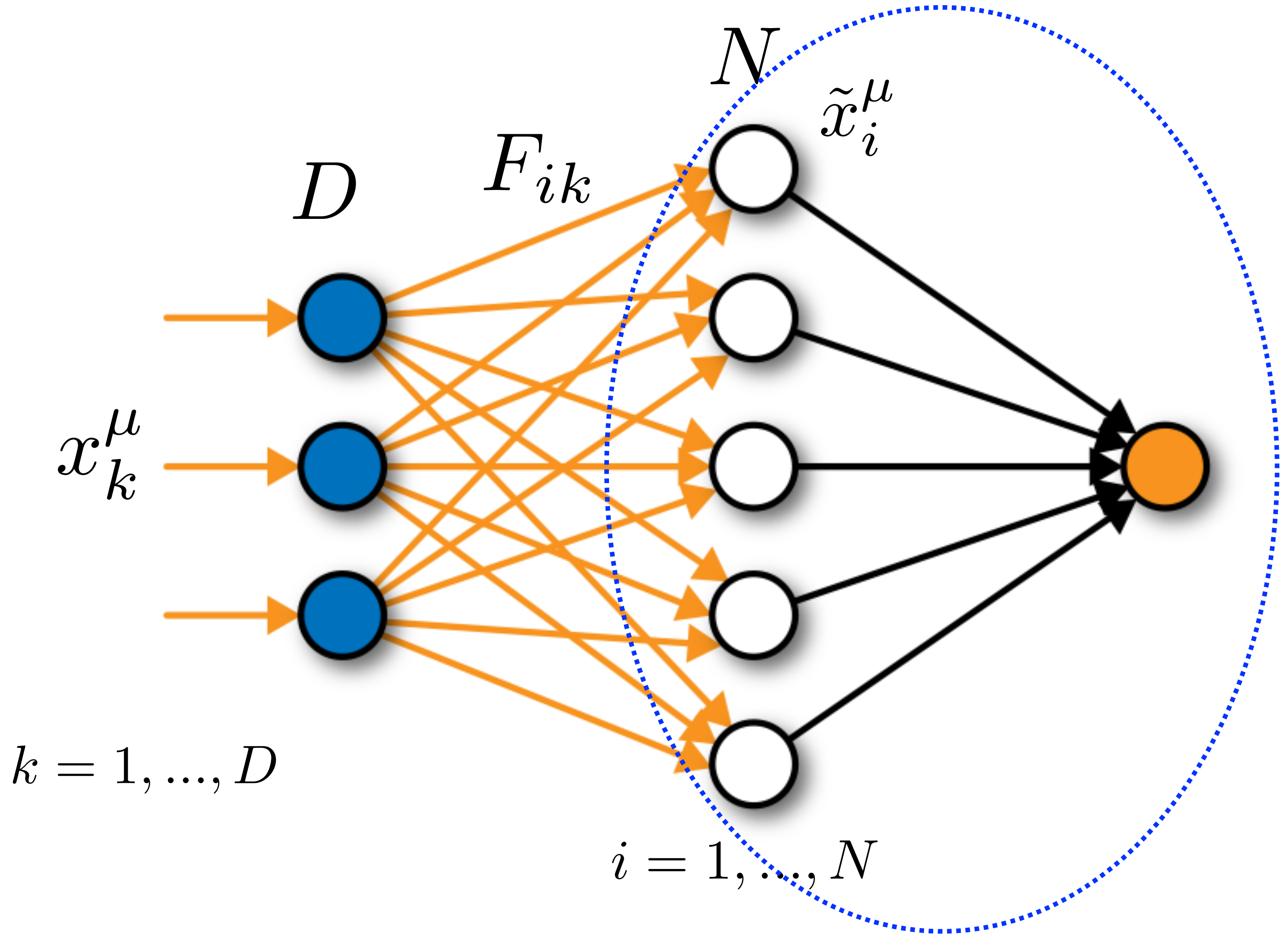
size hidden layers

(Neyshabur, Tomioka, and Srebro, 2014)

Random Feature Model

Neal, 1996; Balcan, Blum, Vempala 2006; Rahimi, Recht; 2008; Bach, 2016

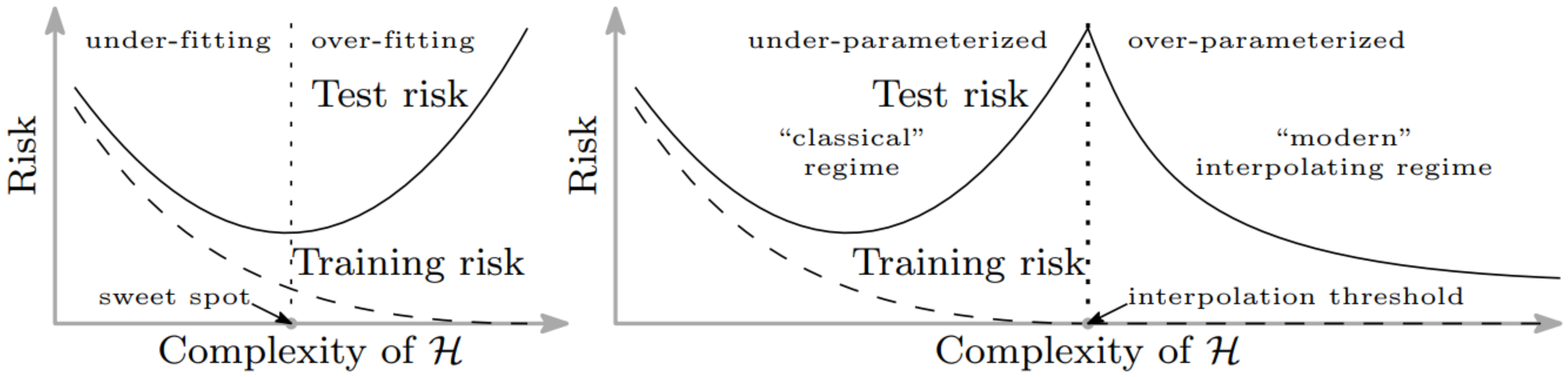
Nonlinear (random) projection of the data (from dimension D to N)



$$\tilde{x}_i^\mu = \sigma\left(\frac{1}{\sqrt{D}} \sum_{k=1}^{D} F_{ki} x_k^\mu\right)$$

$$F_{ki} \sim \mathcal{N}(0,1)$$

- Interpolation does not necessarily lead to poor generalization, as long as you go "deep" enough in the interpolation regime
- Reconciling the modern practice with a statistical point-of-view
- Explicit analysis for Linear Models



(a) U-shaped "bias-variance" risk curve (b) "double descent" risk curve

Belkin, Rakhlin, Tsybakov, 2018

- Connection with the Hidden Manifold Model

Goldt, Mezard, Krzakala, Zdeborova, 2020

Montanari, Mei, 2019

Deep Double Descent

From: Deep Double Descent: Where Bigger Models and More Data Hurt
P Nakkiran, G Kaplun, Y Bansal, T Yang, B Barak, I Sutskever (2019)

Geometric and algorithmic phase transitions on non-convex overparametrized NN

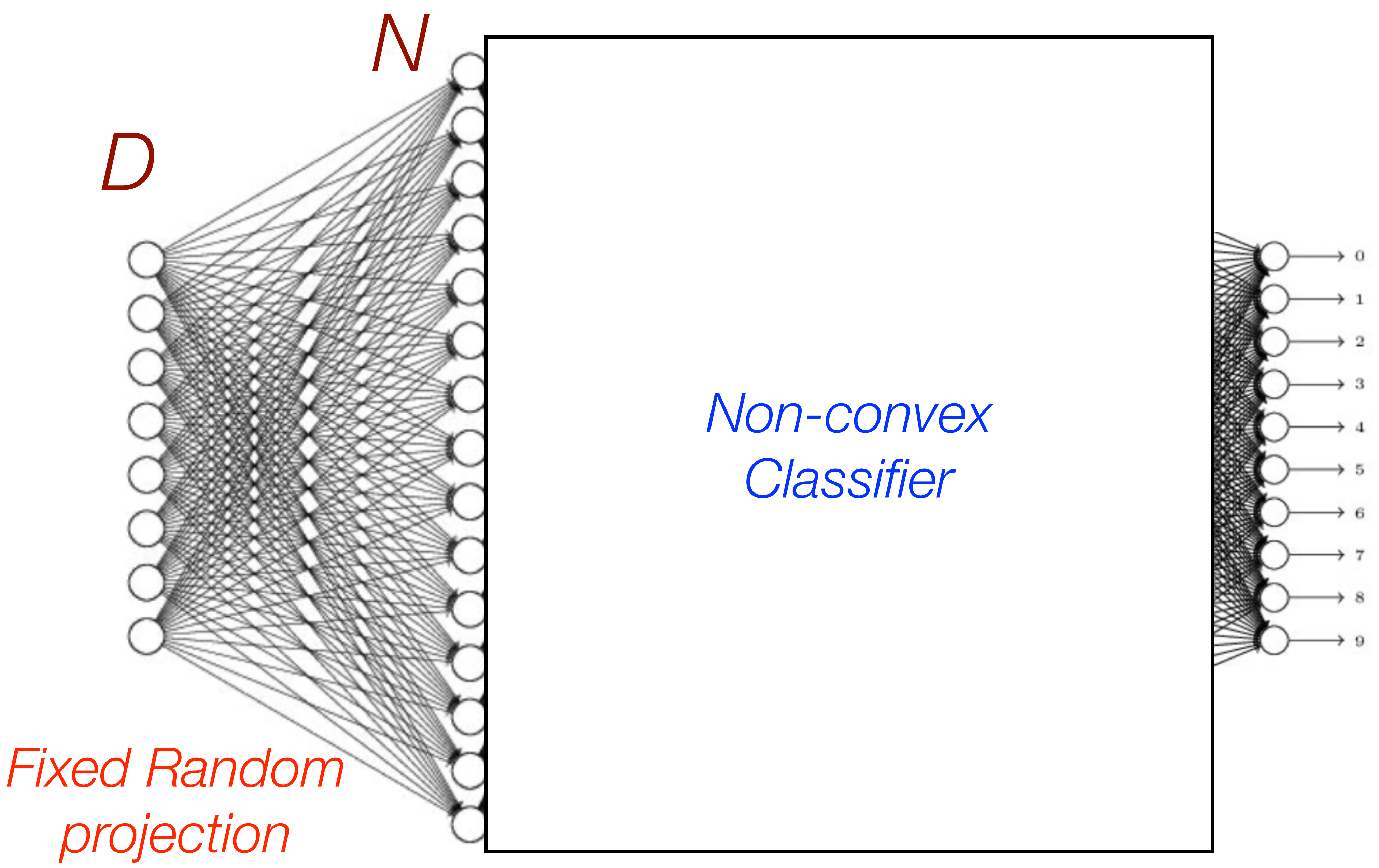
Overparametrized Regime

Input: $D \to \infty$

$N \to \infty$

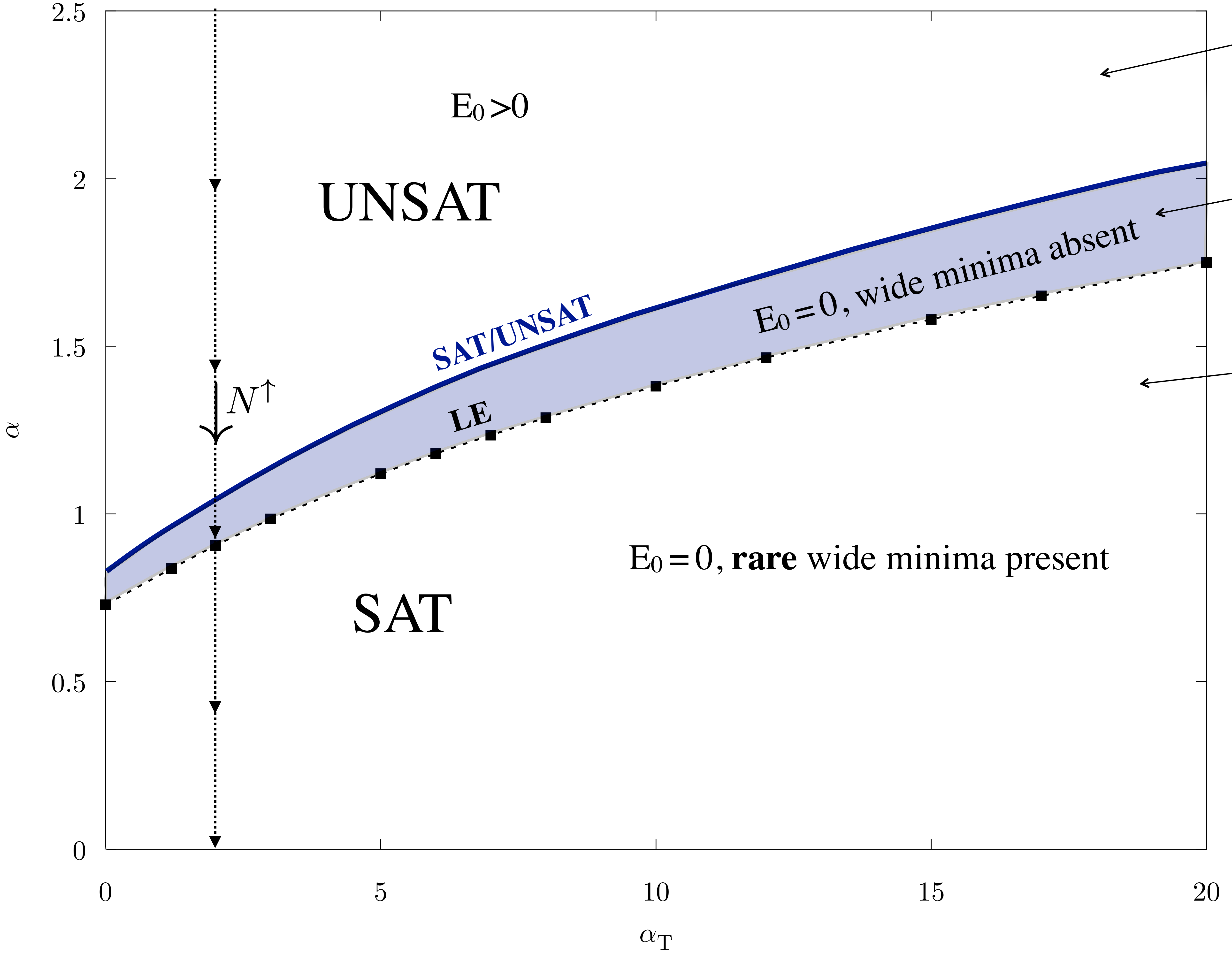
$$\frac{D}{N} \to \psi \ll 1$$

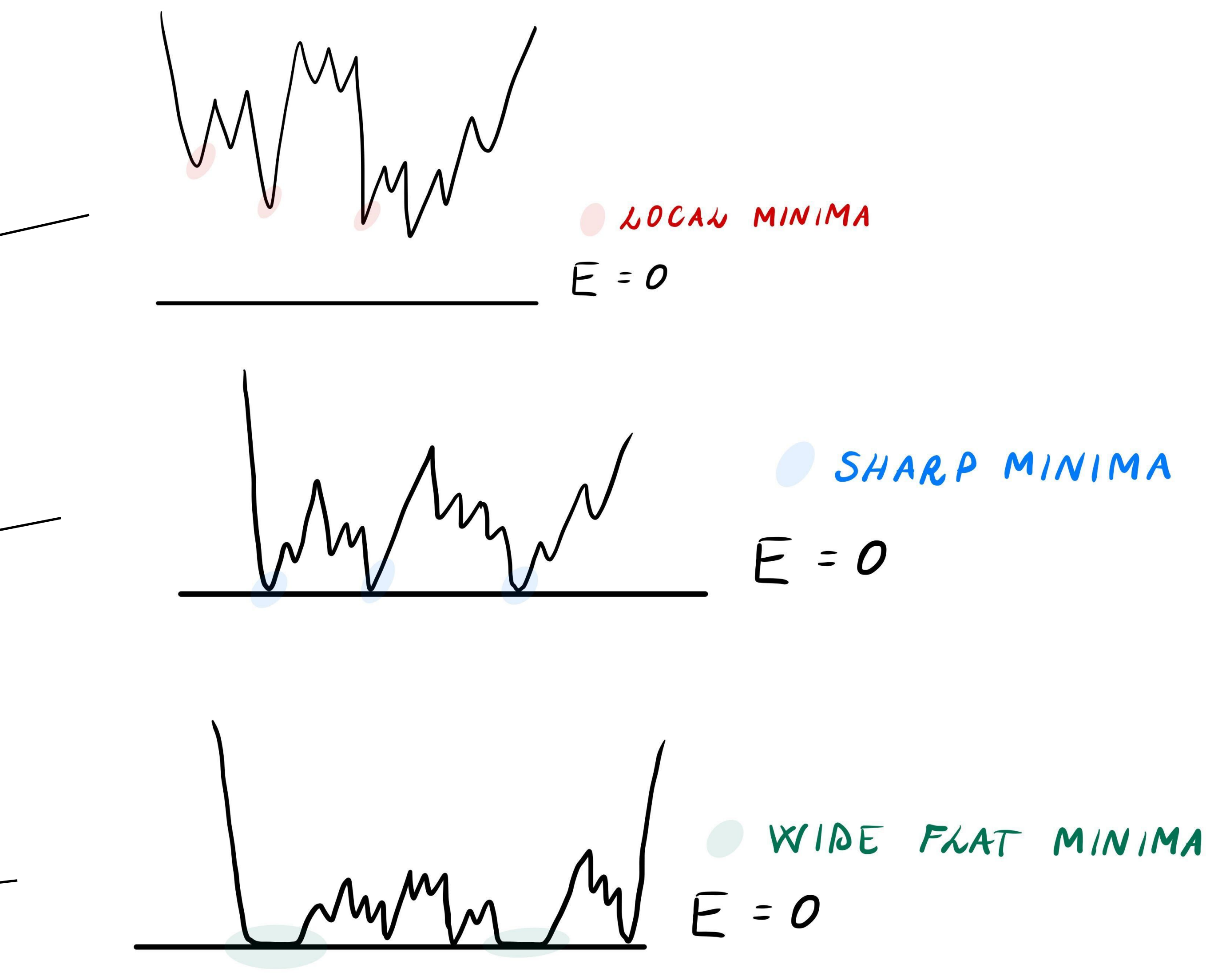
(D intrinsic dimension of the data)



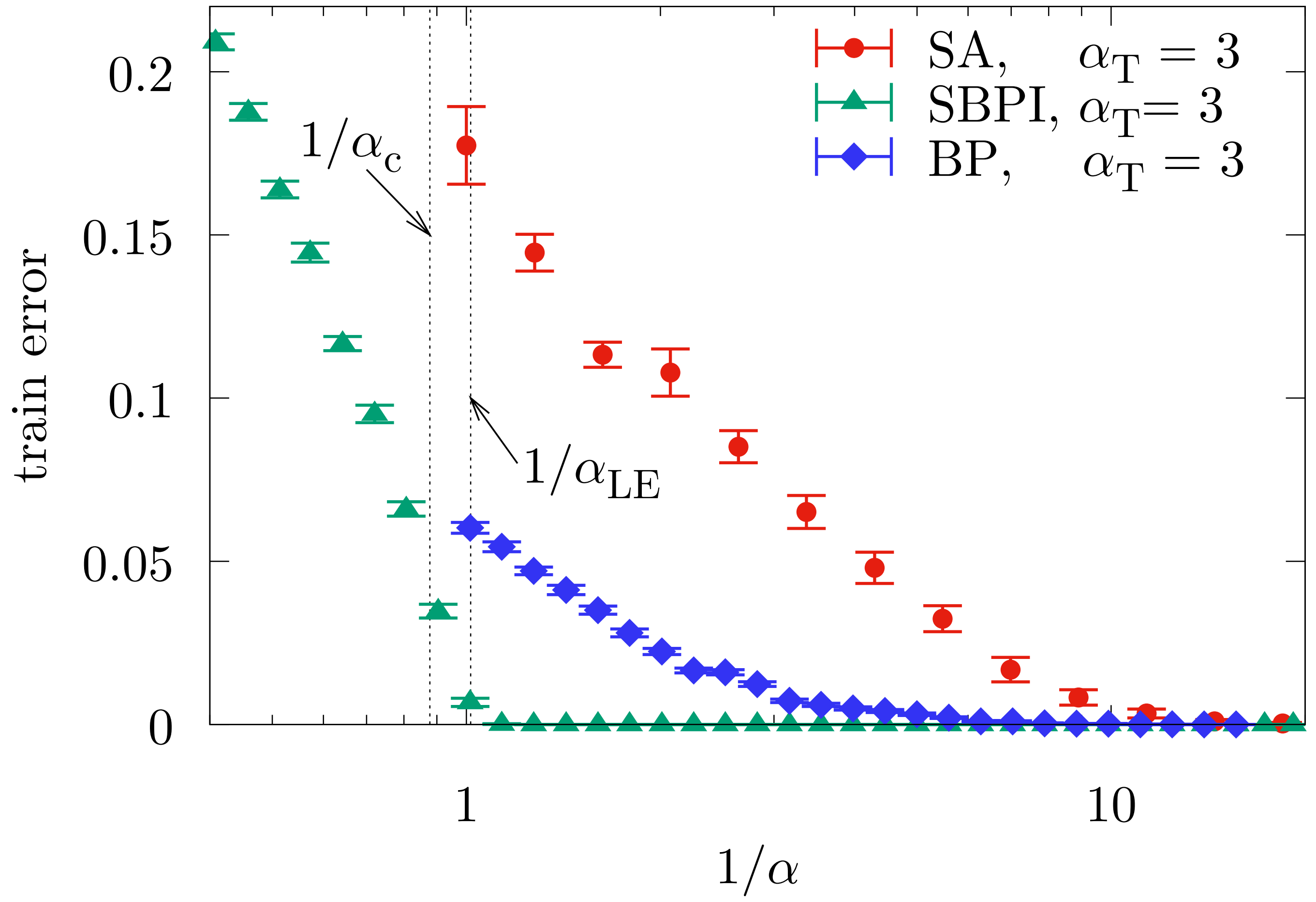
Learning through atypical "phase transitions" in overparametrized neural networks

P= # data points $\alpha \equiv \frac{P}{N}$, $\alpha_T \equiv \frac{P}{D}$, $\alpha_D \equiv \frac{N}{D}$,





Algorithmic experiment



Baldassi, Malatesta, Lauditi, Pacelli, Perugini, RZ , 2021

Analytics vs Numerics in large scale DNNs

PNAS | **January 7, 2020** | vol. 117 | no. 1 | **161–170**

Shaping the learning landscape in neural networks around wide flat minima

Carlo Baldassi[a,b,1,2], Fabrizio Pittorino[a,c], and Riccardo Zecchina[a,d,1,2]

Published as a conference paper at ICLR 2021

ENTROPIC GRADIENT DESCENT ALGORITHMS AND WIDE FLAT MINIMA

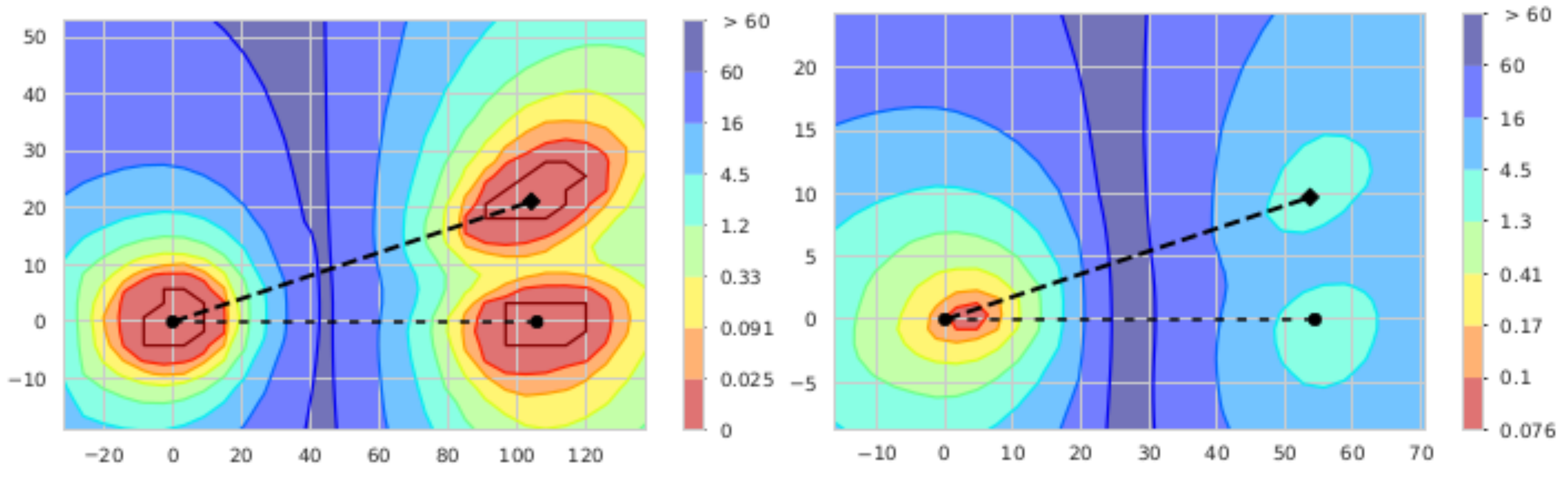
Fabrizio Pittorino[1,2], Carlo Lucibello[1], Christoph Feinauer[1], Gabriele Perugini[1], Carlo Baldassi[1], Elizaveta Demyanenko[1], Riccardo Zecchina[1]

ICML 2022

Deep Networks on Toroids: Removing Symmetries Reveals the Structure of Flat Regions in the Landscape Geometry

Fabrizio Pittorino[1] Antonio Ferraro[1] Gabriele Perugini[1 2] Christoph Feinauer[1] Carlo Baldassi[1] Riccardo Zecchina[1]

VGG16 on Cifar10



- Left Panel: Unnormalized
- Right Panel: Normalized
- Left Points: RSGD (finds flatter minima)
- Right Points: unaligned/aligned SGD with adversarial initialization

Difference is only visible *after* symmetry removal

The quite expensive Google experiments

(Y. Jiang, B.Neyshabur, H. M. D.Krishnan, S.Bengio, 2019)

- *Trained more than **10,000 models** over two image classification datasets (CIFAR-10, Street View House Numbers).*
- *Training under all combination of hyperparameters and optimization resulted in a large pool of models.*
- *For any such model, we considered **40 complexity measures***.*

Findings:

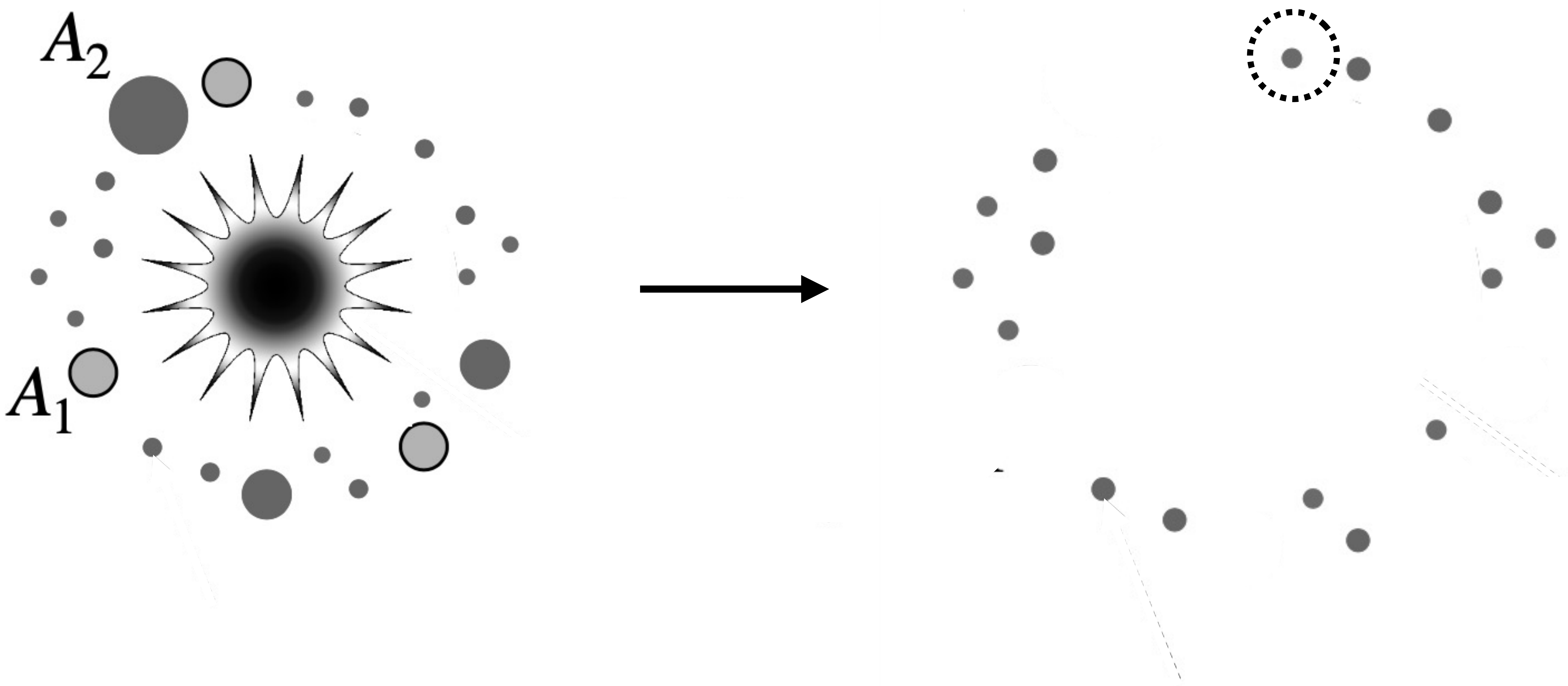
> *" … the relative success of sharpness-based and optimization-based complexity measures for predicting the generalization gap can provoke further study of these measures."*

* ***complexity measure in machine learning:*** *a quantity that monotonically relates to some aspect of generalization. Typically it depends on the trained model and the training data, **but should not have access to a validation set.** Lower complexity should often imply smaller **generalization gap**.*

Our aim is to use random Neural Network models to build cryptographic system, based on what we know about the geometry of solutions.

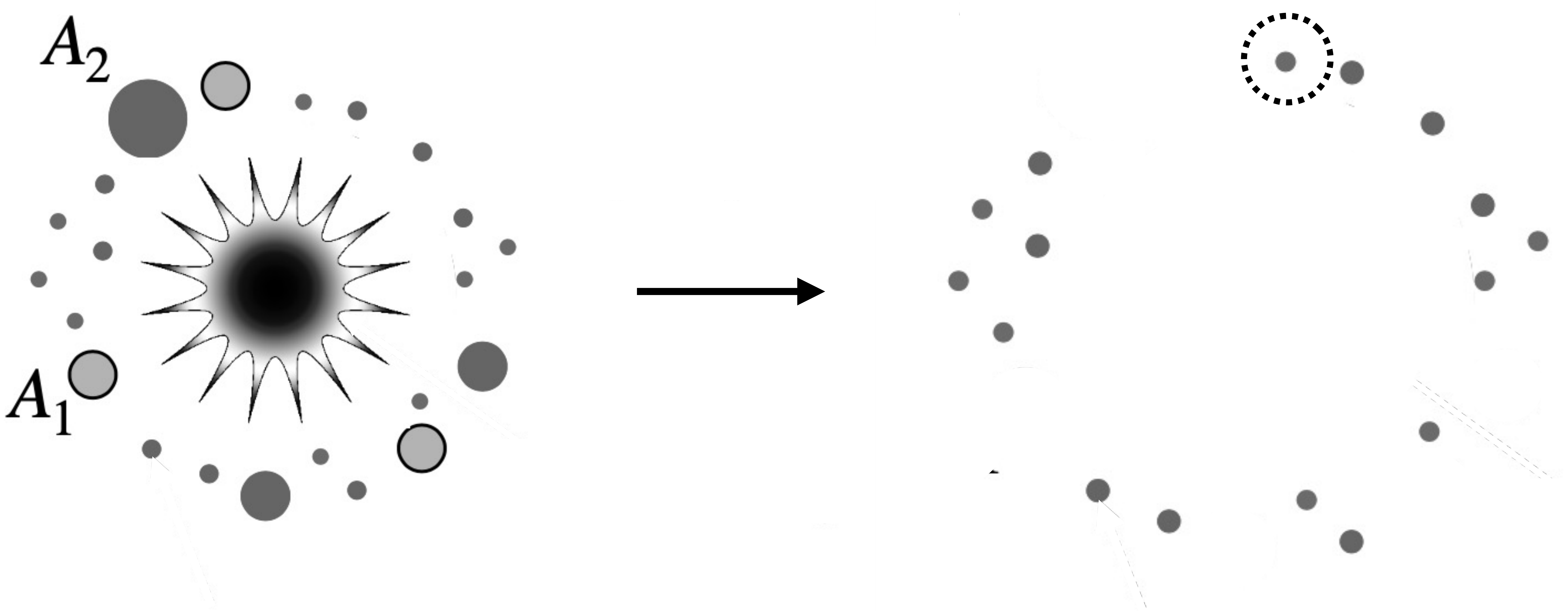
The Overlap Gap Property (OGP)

OGP Definition: In high-dimensional optimization problems, the solution space exhibits a "gap" structure.



The Overlap Gap Property (OGP)

OGP Definition: In high-dimensional optimization problems, the solution space exhibits a "gap" structure.



This scenario has to be studied for *collisions*

The Overlap Gap Property (OGP) *D. Gamarnik, 2018*

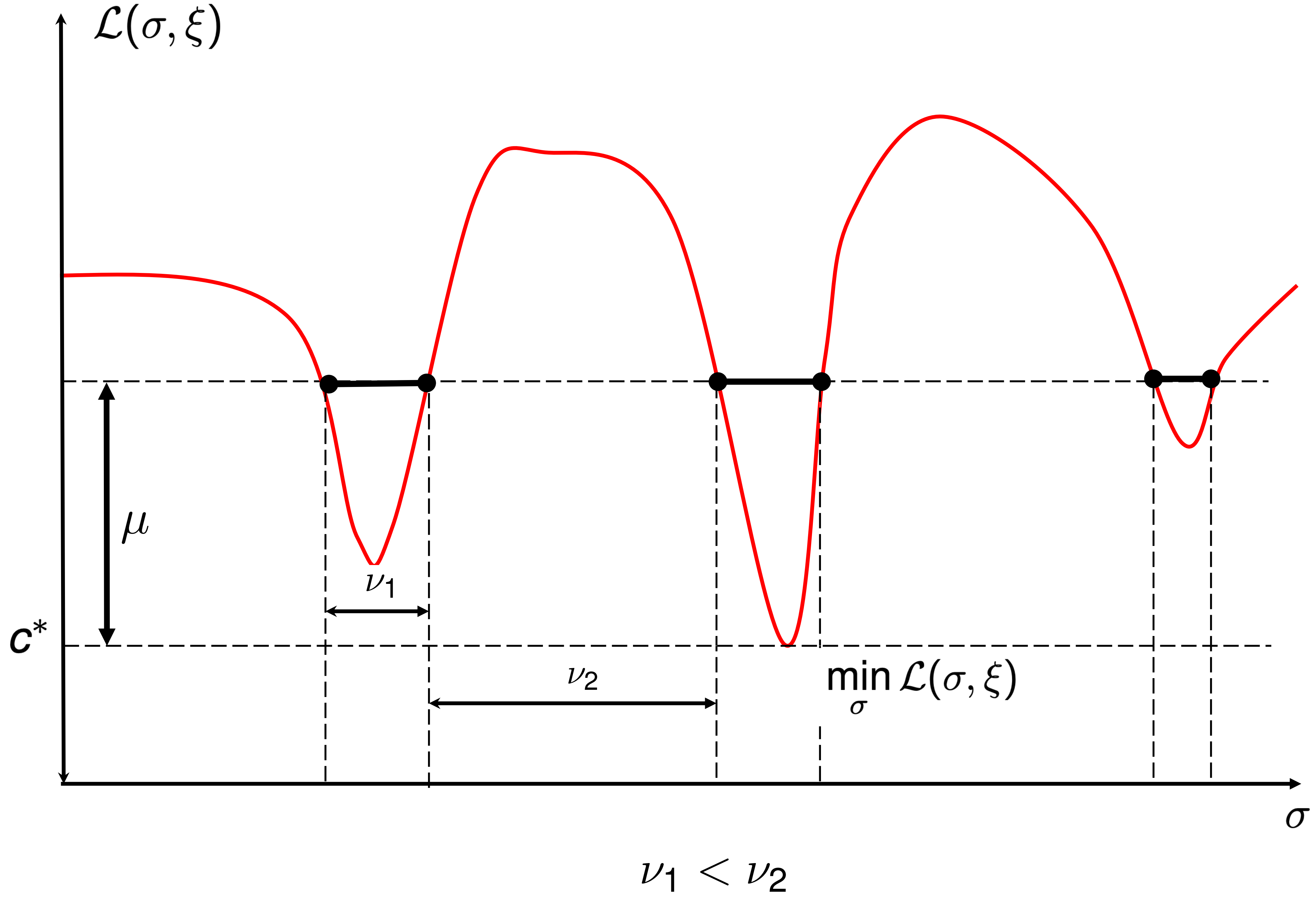
Theorem: (OGP and Stable algorithms), informal statement:

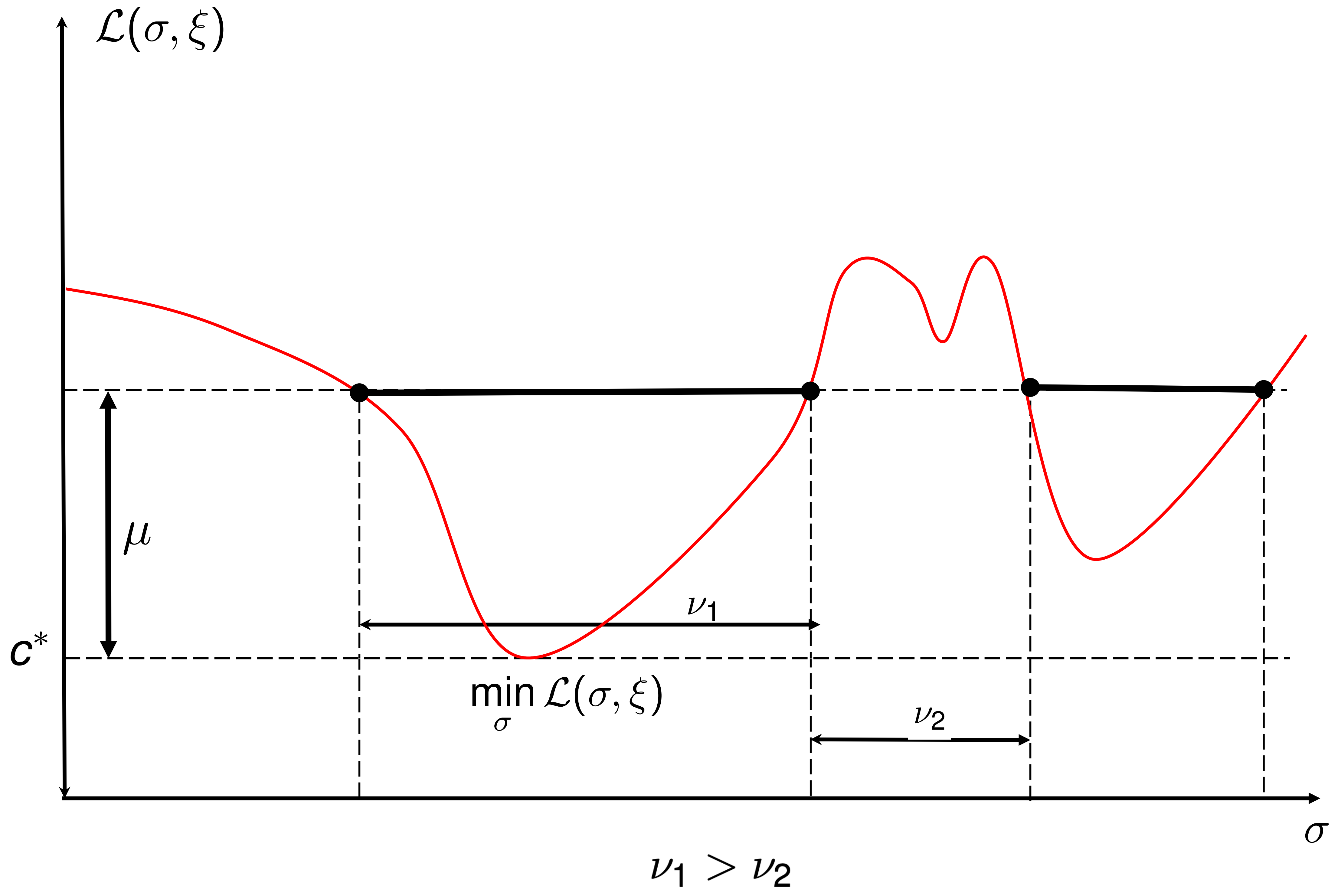
Let **P** be a combinatorial optimization problem exhibiting OGP.

If a *stable algorithm* is applied to solve **P**, then under typical conditions the algorithm will fail to find an optimal solution with high probability.

Generalised to multi-OGP: 2-OGP ⇒ y-OGP (y-OGP is sufficient)

Intuition: if there is OGP a small perturbation can result in a large change in the output, then the algorithm cannot be stable.

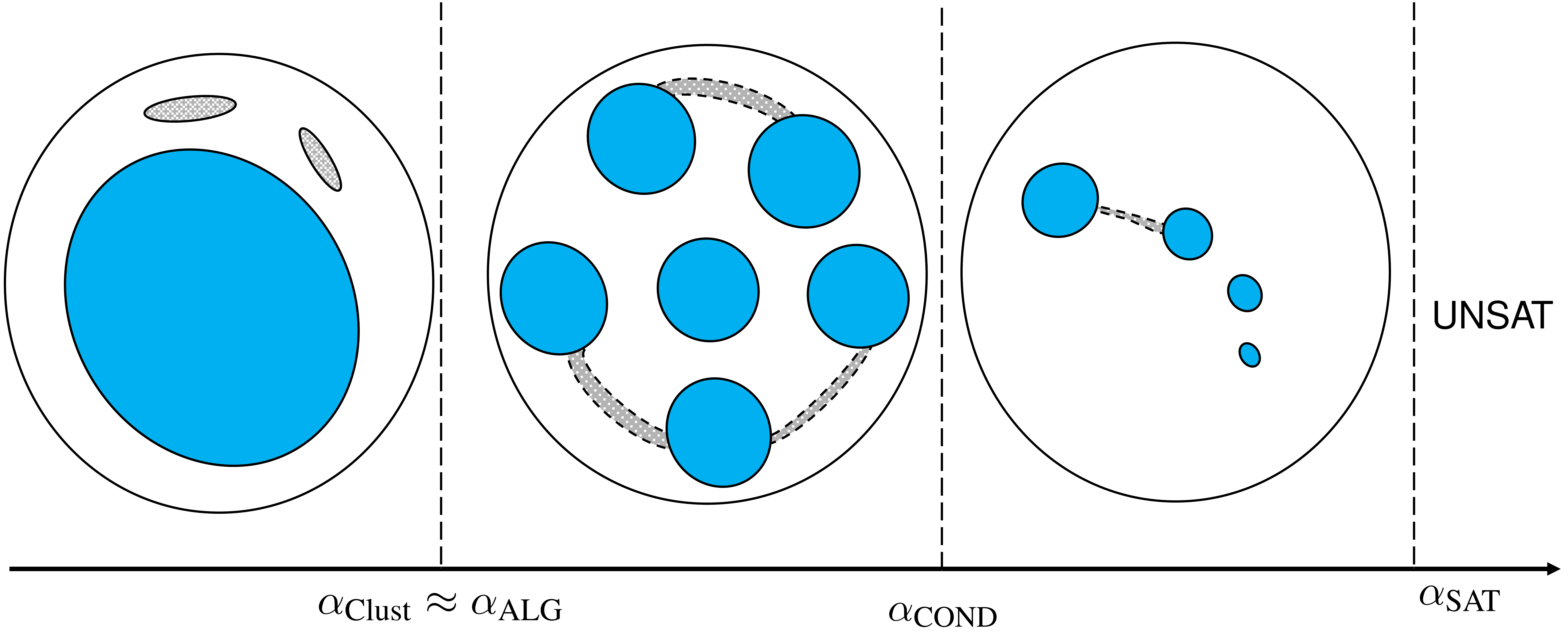




Landscape exhibiting the OGP: Solutions are split into clusters, with diameter of each cluster smaller than the distance between any pair clusters.

Landscape not exhibiting the OGP: diameter of one cluster is larger than distance between one pair of clusters.

from D. Gamarnik, PNAS 2018



from D. Gamarnik, PNAS 2018

What is a Stable Algorithm?

- ***Definition:*** it refers to an algorithm whose output does not change significantly when small, random perturbations are made to its input. The algorithm is *robust* to minor changes or noise in the input, meaning its performance or result is not highly sensitive to such perturbations.

Examples:

- *Gradient Descent*
- *SVMs*
- *PCS*
- *Convex Optimization Algorithms (Interior-Point Methods or Simplex Method)*
- *Low degree polynomials (hence message-passing)*
- *QAOA*
- *…*

Post-Quantum Cryptography (PQC)

PQC: Cryptographic algorithms designed to be secure against attacks by quantum computers.

Quantum computers can break widely-used cryptographic systems (e.g., RSA, ECC) by leveraging algorithms like Shor's algorithm.

- Lattice-based Cryptography: Utilizes hard problems in lattice structures, such as Learning With Errors (LWE).
- Code-based Cryptography: Based on the difficulty of decoding random linear codes, such as McEliece cryptosystem.
- **Hash-based Cryptography: Relies on the security of hash functions, used for digital signatures.**
- Multivariate Cryptography: Solves systems of multivariate polynomial equations, considered hard for quantum computers.

Ajtai's Function and high-dimensional Lattice problems

Lattice Problem: Consider a lattice $\mathscr{L}$ defined as $\mathscr{L} = \{B \cdot z : z \in \mathbb{Z}^n\}$, where B is a (random) basis matrix. Ajtai's function is related to the **shortest vector problem (SVP)** in this lattice.

Ajtai's Theorem:

Worst-Case to Average-Case Reduction: Ajtai showed that there exists an algorithmic function $f(x)$ that maps a random instance of a lattice problem (like SVP) to a generic solution in polynomial time if and only if the corresponding worst-case problem is solvable in polynomial time.

Hardness Guarantee: The function $f(x)$ in Ajtai's theorem ensures that if an efficient algorithm solves the average case of this function, then it can also solve the hardest instances of the problem.

Generating Hard Instances of Lattice Problems
Extended abstract
M. Ajtai
IBM Almaden Research Center

(1996)

Explicit case: *Short Integer Solution (SIS) Problem*

Find a short vector $x \in \mathbb{Z}^n$ such that $A \cdot x = 0 \mod q$ for a given **random matrix** A.
This problem is hard under Ajtai's worst-case to average-case reduction.

Function Definition:

The SIS function could be formally written as

$$f(A) = \{x : A \cdot x = 0 \mod q,\ \|x\| \text{ is small}\}.$$

Observations:

- *the hardness of these problems (and the functions derived from them) is believed to hold even against quantum computers.*
- *not very efficient …*

Remark:

Post-Quantum Encryption Standards

NIST PQC Standardization: The National Institute of Standards and Technology (NIST) has just approved a standard for PQC algorithms, with the first crypto system officially approved.

Open problem:

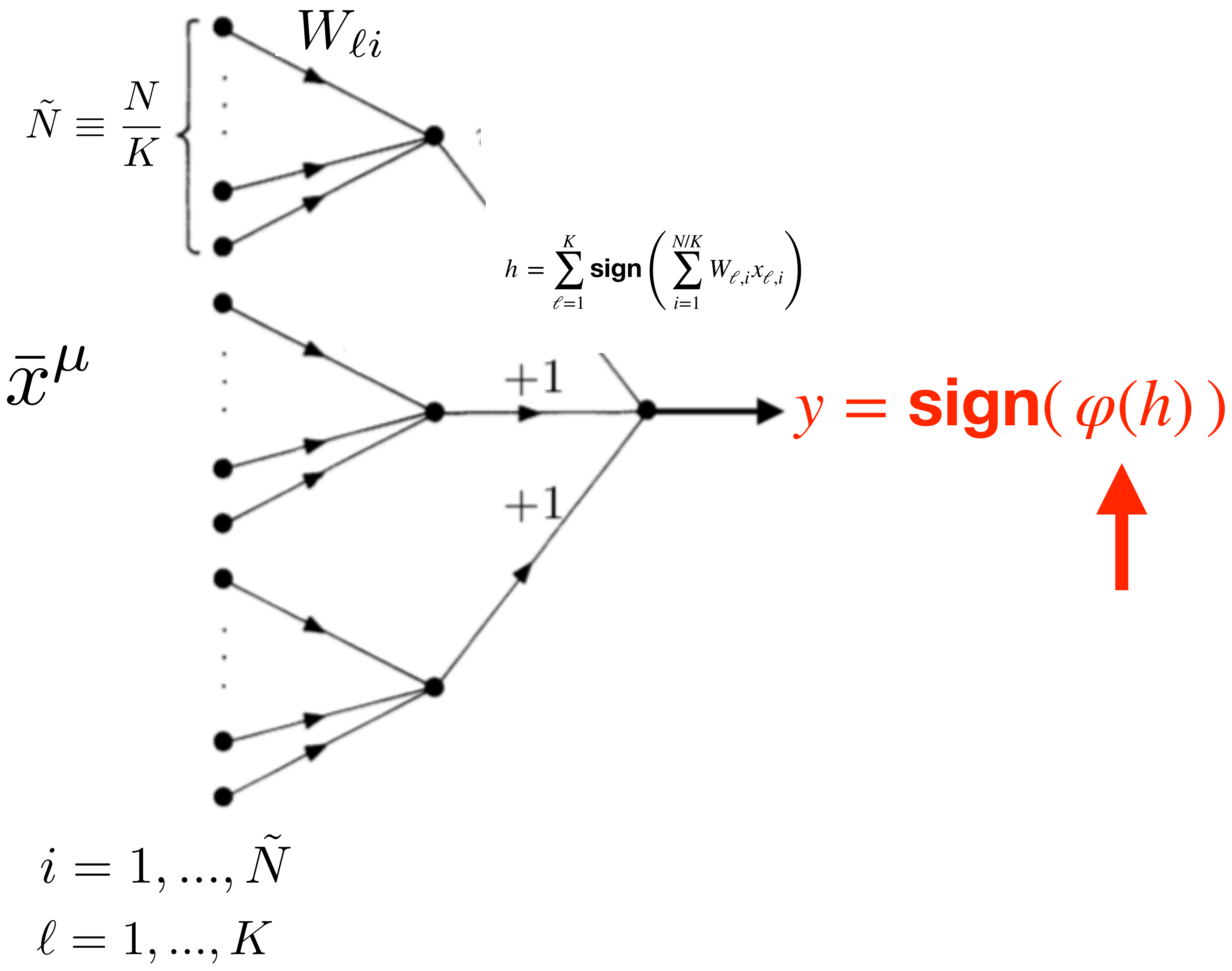
PQC algorithms may have larger key sizes and slower performance compared to traditional algorithms. There is a need for efficient crypto systems.

Collision problem

$\Downarrow$

these rare events should not exist!

Simplest non convex neural device : 1-hidden layer, i.i.d. random associations



training set: $\{(\bar{x}^{\mu}, y^{\mu})\} \quad \mu = 1, \ldots, P = \alpha N$

$$x^{\mu}_{\ell i} = \pm 1 \ \ (i.i.d. \ \ p = 1/2)$$

$$y^{\mu} = \pm 1 \ \ (i.i.d. \ \ p = 1/2)$$

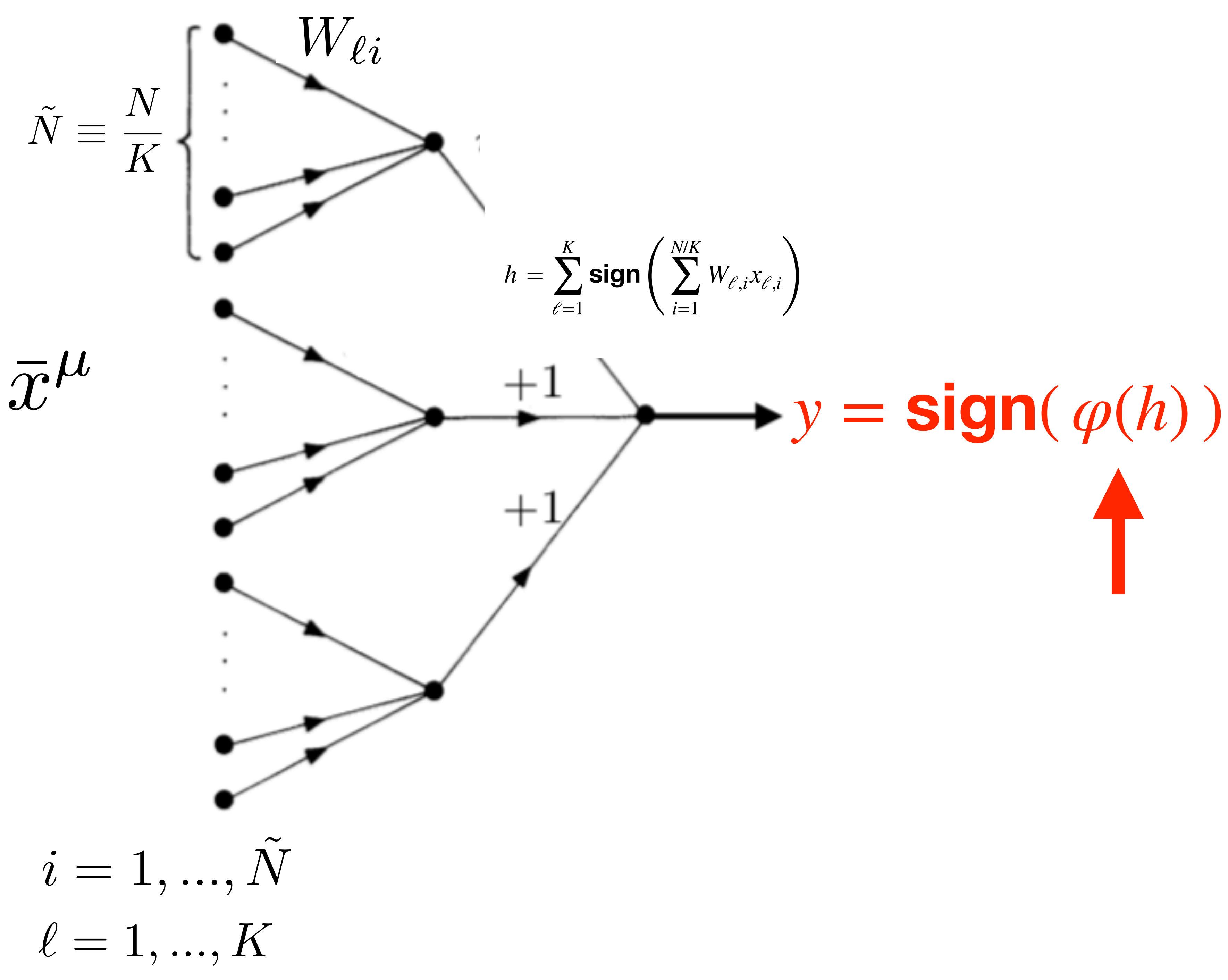
$$\mathbf{A} := [\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^P]$$

control parameter: $\alpha = \dfrac{\#\ \text{patterns}}{\#\ \text{weights}}$

Non convex also for K=1 (perceptron)

Find W such that $y_{\mathbf{A}}(W) = \mathbf{y}$ with $\{W_i = \pm 1\}$, i.e. $\mathrm{sign}(\varphi(\sum_i W_i x_i^{\mu})) = y^{\mu} \quad \forall \mu$

Simplest non convex neural device : 1-hidden layer, i.i.d. random associations



training set: $\{(\bar{x}^{\mu}, y^{\mu})\} \quad \mu = 1, \ldots, P = \alpha N$

$$x^{\mu}_{\ell i} = \pm 1 \ \ (i.i.d. \ \ p = 1/2)$$

$$y^{\mu} = \pm 1 \ \ (i.i.d. \ \ p = 1/2)$$

$$\mathbf{A} := [\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^P]$$

control parameter: $\alpha = \dfrac{\#\ \text{patterns}}{\#\ \text{weights}}$

Non convex also for K=1 (perceptron)

Find W such that $y_{\mathbf{A}}(W) = \mathbf{y}$ with $\{W_i = \pm 1\}$, i.e. $\mathrm{sign}(\varphi(\sum_i W_i x_i^{\mu})) = y^{\mu} \quad \forall \mu$

We will need to generalize this model through φ to obtain the computational bounds we need for the collision problem

Computational challenges in non-convex NN:

$\mathbf{A} \in \mathbf{R}^{P \times N}$ *random matrix composed by* P N*-dim random rows* $\mathbf{x}^\mu$

$$\mathbf{A} := \begin{pmatrix} — \mathbf{x}^1 — \\ \mathbf{x}^2 \\ \vdots \\ \mathbf{x}^P \end{pmatrix}$$

- Inversion (learning): given disorder $\mathbf{A} \in \mathbf{R}^{P \times N}$ and labels $\mathbf{y} \in \{-1,1\}^P$, find any set of weights $W \in \{-1,1\}^N$ such that $y_{\mathbf{A}}(W) = \mathbf{y}$, assuming such W exists.

- Teacher-student: given disorder $\mathbf{A} \in \mathbf{R}^{P \times N}$ and labels $\hat{\mathbf{y}} = y_{\mathbf{A}}(W) \in \{-1,1\}^P$ for uniformly sampled $W \in \{-1,1\}^N$, find any $W' \in \{-1,1\}^N$ such that $y_{\mathbf{A}}(W') = \hat{\mathbf{y}}$

- Collision finding: given disorder $\mathbf{A} \in \mathbf{R}^{P \times N}$, find any two $W \neq W' \in \{-1,1\}^N$ such that $y_{\mathbf{A}}(W) = y_{\mathbf{A}}(W')$ (unexplored so far).

Collision finding:

The input is simply the function $y_{\mathbf{A}}$ itself, and the problem is to find a collision, defined as a pair of distinct $W \neq W'$ such that $y_{\mathbf{A}}(W) = y_{\mathbf{A}}(W')$.

Collision Resistant Hash Functions

Def.: A hash function family $\mathscr{H} = \{h : X \to Y\}$ is said to be *collision resistant*, if for any polynomial-time algorithm $A(\cdot)$ and any constant $c > 0$, it holds that,

$$\Pr_{A,\, h \in_R \mathscr{H}} \left[h(x) = h(y) \wedge x \neq y \mid (x, y) \leftarrow A \right] = o\left(n^{-c}\right),$$

where the randomness is taken over a uniform random choice of h, and the random coins used by A.

The Generalised Binary Perceptron model(s)

$$\hat{y} = \text{sign}\,(\,\varphi(h)\,)$$

where $h \equiv \frac{1}{\sqrt{N}} \sum_i w_i x_i$, w_i are binary variables, and $\boldsymbol{x}$ is a N-dimensional pattern

In order to fit our model to a set of random inputs $\boldsymbol{x}^\mu$ and labels y^μ, we need to impose that the stability Δ^μ is larger than zero

$$\Delta^\mu(\boldsymbol{w}) \equiv y^\mu \varphi\left(\frac{1}{\sqrt{N}} \sum_i w_i x_i^\mu\right) \geq 0\,, \qquad \text{for any } \mu \in [P].$$

Construction of a Hash Function Using a random NN

Random Function Generation: Generate a set of $P = \alpha N$ random patterns

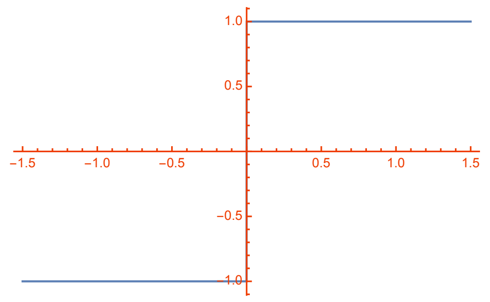
Input: Consider an input vector W from the space of possible inputs (e.g., a message or file).

Hash Function: The hash function based on the GRW model can be defined as follows.

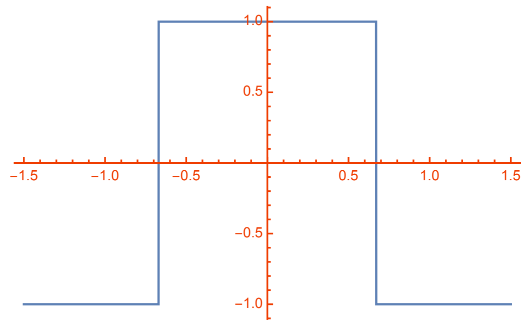
$$y : \mathbf{B}^N \to \mathbf{B}^P \quad \left\{ \begin{array}{l} y_1(x) = \textbf{sign}\left(\varphi\left(\frac{1}{\sqrt{N}}\sum_i w_i x_i^{(1)}\right)\right) \\ y_2(x) = \textbf{sign}\left(\varphi\left(\frac{1}{\sqrt{N}}\sum_i w_i x_i^{(2)}\right)\right) \\ \vdots \\ y_P(x) = \textbf{sign}\left(\varphi\left(\frac{1}{\sqrt{N}}\sum_i w_i x_i^{(P)}\right)\right) \end{array} \right.$$

Some relevant examples of non-linearities

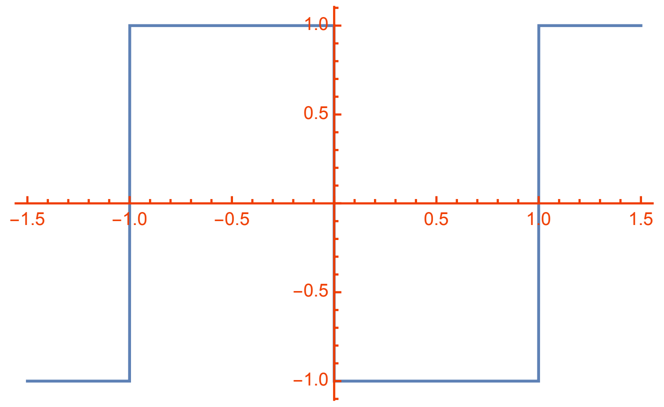
- $\varphi(h) = h$ *standard binary perceptron* model;



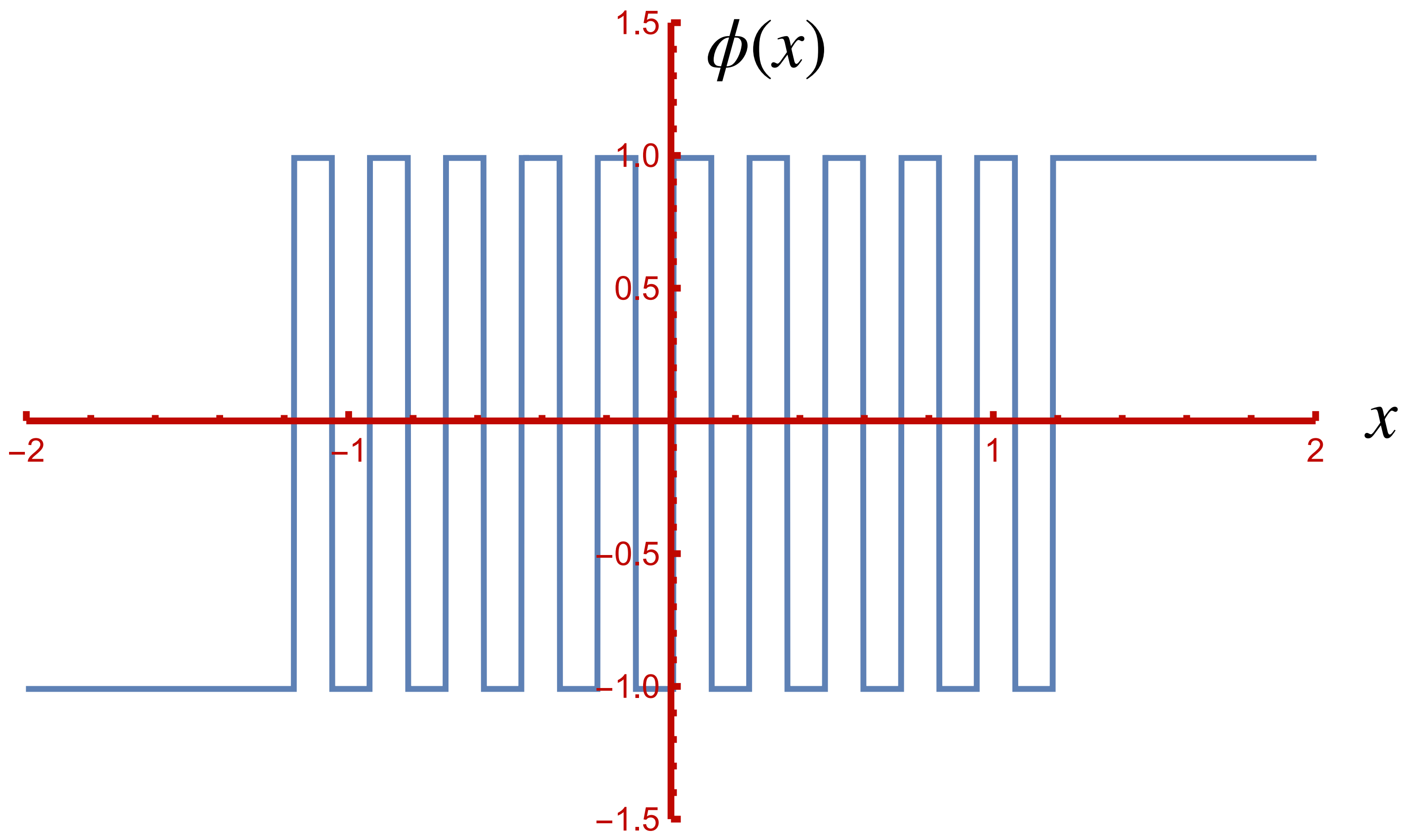
- $\varphi(h) = \kappa - \left| h \right|$ and $y^\mu = 1$, *symmetric perceptron*;



- $\varphi(h) = (h-\gamma)h(h+\gamma)$ *reversed wedge perceptron*;



- $$\varphi(h) = \prod_{l=-K}^{K} \left(h + \frac{l\gamma}{K} \right)$$ *generalizalized reverse wedge perceptron*, with K oscillations in $[-\gamma, \gamma]$.



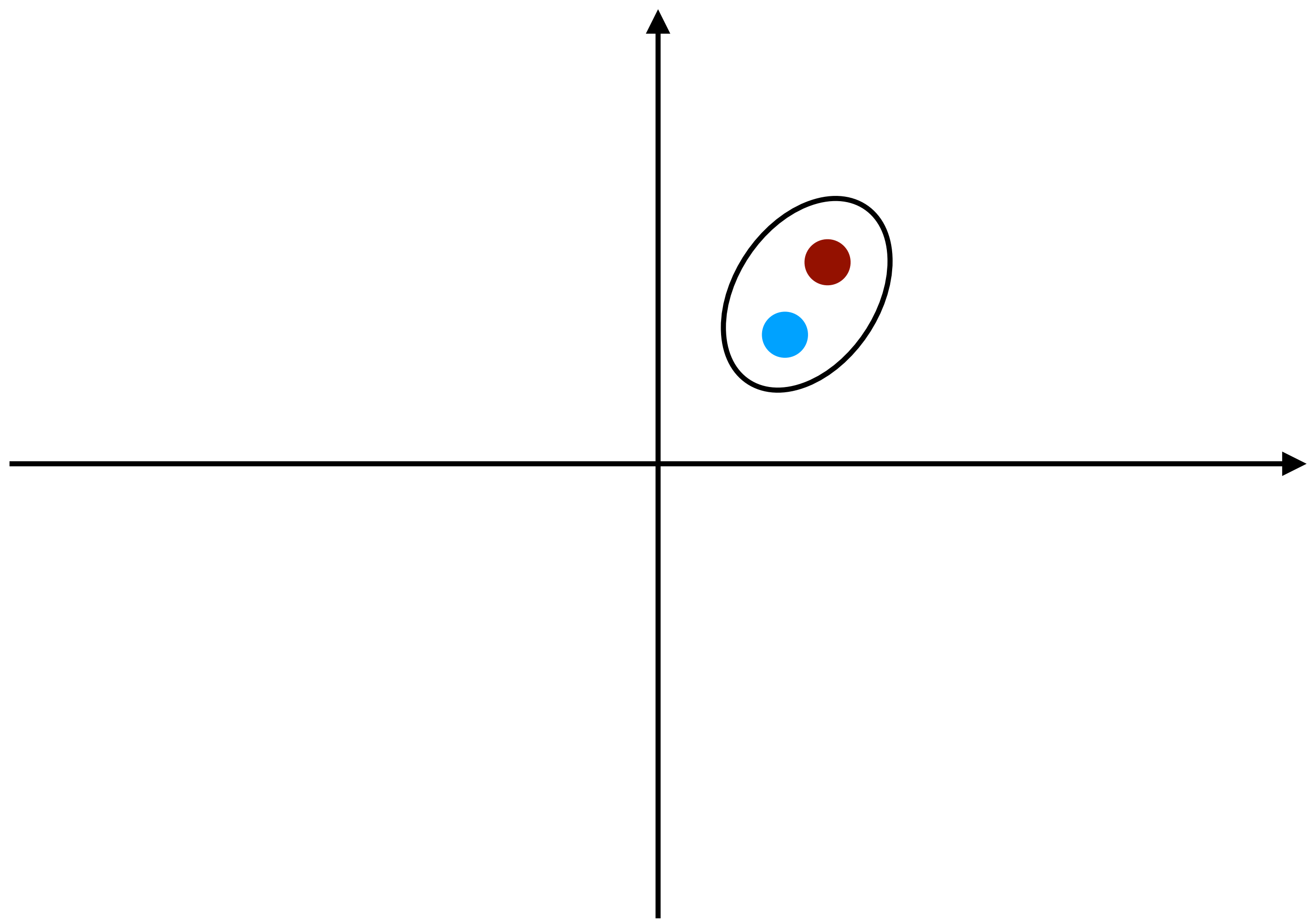
Indicator function $\mathbb{X}_{\boldsymbol{x}}(\boldsymbol{w};\kappa) \equiv \prod_{\mu=1}^{P} \Theta\left(\Delta^\mu(\boldsymbol{w})\right)$,

$x_i^\mu \sim \mathcal{N}(0,1)$ and $\alpha \equiv \dfrac{P}{N}$.

The problem of Collisions

Given $\boldsymbol{x}^\mu$, $\mu = 1,\ldots,P$, and $x_i^\mu \sim \mathcal{N}(0,1)$, find $\boldsymbol{w}^{(1)}, \boldsymbol{w}^{(2)}$ s.t.

$$\textbf{sgn}\left[\varphi\left(\frac{1}{\sqrt{N}}\boldsymbol{w}^{(1)}\cdot\boldsymbol{x}^\mu\right)\right] = \textbf{sgn}\left[\varphi\left(\frac{1}{\sqrt{N}}\boldsymbol{w}^{(2)}\cdot\boldsymbol{x}^\mu\right)\right] \quad \forall\mu$$



Indicator function $\mathbb{X}_{\boldsymbol{x}}(\boldsymbol{c})$:

$$\mathbb{X}_{\boldsymbol{x}}(\boldsymbol{c}) \equiv \prod_{\mu=1}^{P}\sum_{y^\mu}\Theta\left(y^\mu\varphi\left(\frac{1}{\sqrt{N}}\boldsymbol{w}^{(1)}\cdot\boldsymbol{x}^\mu\right)\right)\Theta\left(y^\mu\varphi\left(\frac{1}{\sqrt{N}}\boldsymbol{w}^{(2)}\cdot\boldsymbol{x}^\mu\right)\right) \qquad \text{with } \boldsymbol{c} \equiv (\boldsymbol{w}^{(1)}, \boldsymbol{w}^{(2)}).$$

Partition function of collisions is $Z_{\boldsymbol{x}} \equiv \int d\boldsymbol{c}\ \mathbb{X}_{\boldsymbol{x}}(\boldsymbol{c};\kappa_c)$ where $d\boldsymbol{c} \equiv d\boldsymbol{w}^{(1)}d\boldsymbol{w}^{(2)}$.

Geometric landscape collisions

Local entropy of collisions $\widetilde{\boldsymbol{w}}_1, \widetilde{\boldsymbol{w}}_2$:

$$\ln \mathcal{N}_{\boldsymbol{\xi}}(\widetilde{\boldsymbol{w}}_1, \widetilde{\boldsymbol{w}}_2; d) \equiv \ln \int d\boldsymbol{w}_1 d\boldsymbol{w}_2 \, \mathbb{X}_{\boldsymbol{\xi}}(\boldsymbol{w}_1, \boldsymbol{w}_2) \, \delta\left(d\left[(\widetilde{\boldsymbol{w}}^{(1)}, \widetilde{\boldsymbol{w}}^{(2)}), (\boldsymbol{w}^{(1)}, \boldsymbol{w}^{(2)})\right] - d\right)$$

where $d\left[(\widetilde{\boldsymbol{w}}^{(1)}, \widetilde{\boldsymbol{w}}^{(2)}), (\boldsymbol{w}^{(1)}, \boldsymbol{w}^{(2)})\right]$ is a permutation invariant distance between two collisions.

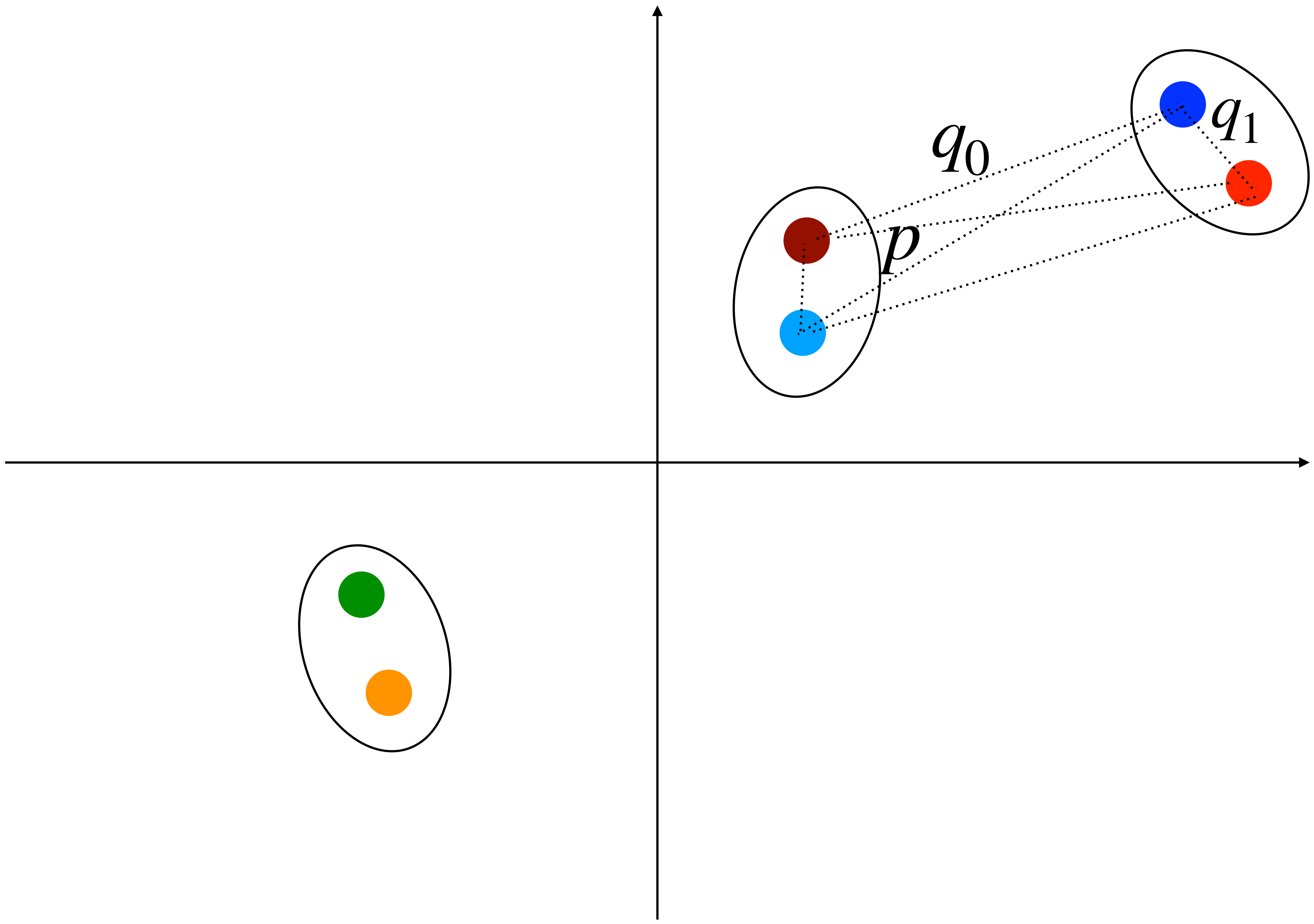
Consider $\boldsymbol{c}_a = (\boldsymbol{w}_a^{(1)}, \boldsymbol{w}_a^{(2)})$ and $\boldsymbol{c}_b = (\boldsymbol{w}_b^{(1)}, \boldsymbol{w}_b^{(2)})$ with $a \neq b$.

$$d\left(\boldsymbol{c}_a, \boldsymbol{c}_b\right) = \min_{\pi \in \mathcal{S}_2} \frac{1}{2} \sum_{s=1}^{2} d\left(\boldsymbol{w}_a^{(s)} - \boldsymbol{w}_b^{\pi(s)}\right) = \min_{\pi \in \mathcal{S}_2} \frac{1}{2} \sum_{s=1}^{2} \frac{1}{4N} \sum_{i=1}^{N} \left(w_{ai}^{(s)} - w_{bi}^{\pi(s)}\right)^2$$

$$= \min_{\pi \in \mathcal{S}_2} \frac{1}{2} \sum_{s=1}^{2} \frac{1}{2} \left(1 - \frac{1}{N} \boldsymbol{w}_a^{(s)} \cdot \boldsymbol{w}_b^{\pi(s)}\right) = \frac{1}{4} \max_{\pi \in \mathcal{S}_2} \sum_{s=1}^{2} \left(1 - q_{s\pi(s)}^{ab}\right) = \frac{1}{2}(1 - p)$$

with $\boldsymbol{p}$ the overlap on the diagonal of the overlap matrix q_{st}^{ab}

(thanks to the symmetry, we have that the overlap $q_{1\pi(1)}^{ab} = q_{2\pi(2)}^{ab}$, and we can choose $\boldsymbol{\pi}$ to be the identity)



q_0
q_1
p

Free entropy $\phi_y(d)$ in the annealed approximation, i.e.

$$\phi_y(d) \leq \phi_y^A(d) = \lim_{yN\to\infty} \frac{1}{yN} \ln \mathbb{E}_{\boldsymbol{x}} \mathcal{N}_y(d;\boldsymbol{x})$$

Since $\mathcal{N}_y(d;\boldsymbol{x})$ is a non-negative and integer valued random variable, by Markov inequality we get

$$P(\mathcal{N}_y(d;\boldsymbol{x}) > 0) \leq \mathbb{E}_{\boldsymbol{x}} \mathcal{N}_y(d;\boldsymbol{x}) = e^{yN\phi_y^A(d)}$$

If $\phi_y^A(d) \leq 0$ *for* $\alpha \geq \alpha_c^{UB}(d)$ *then* $P(\mathcal{N}_y(d;\boldsymbol{x}) > 0) \to 0$ *for large* N.

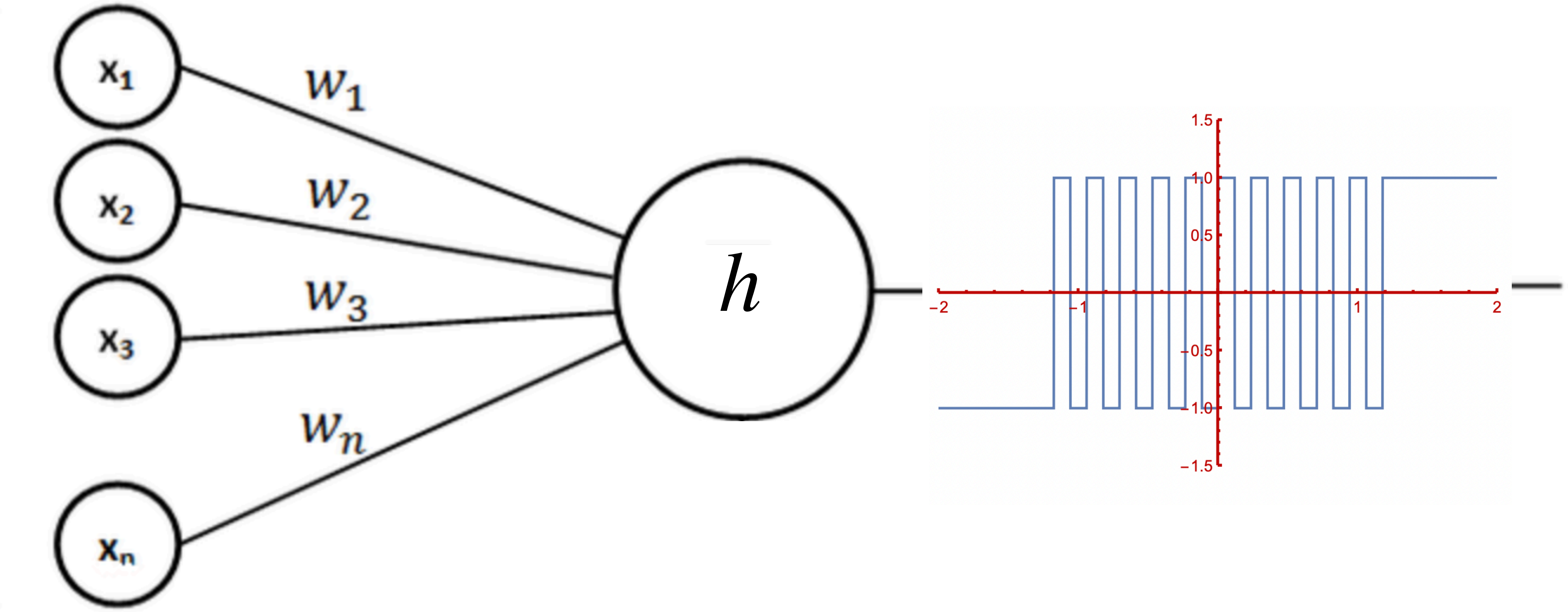
$\alpha \geq \alpha_c^{UB}(d) \quad \Rightarrow$ no collisions at a fixed distance d to one another.

Note: that $\alpha_c^{UB}(d)$ *is only an upper bound to the true value (i.e. it might be that the true* α_c *is lower than that).*

Generalised Reverse Wedge model

$$\varphi(h) = \prod_{l=-K}^{K} \left(h + \frac{l\gamma}{K} \right)$$

generalizalized reverse wedge perceptron, with K oscillations in $[-\gamma, \gamma]$.



In the large K limit, the computation simplifies: $\lim_{N\to\infty} \frac{K}{N} \to 0$, and next $K \gg 1$

Conclusions

Statistical physics of highly non-convex random systems and Crypto are very close

Spin glass theory used for crypto-systems design

These are just first steps, we conjecture we can prove CRH w.r.t. stable algorithms

Marco Benedetti, Andrej Bogdanov, Enrico Malatesta, Marc Mezard, Gianmarco Perrupato, Alon Rosen, Nikolaj I. Schwartzbach , and Riccardo Zecchina

Quantum Annealing for non convex learning devices

- Quantum annealing strategy: use quantum fluctuations (rather than thermal fluctuations) to overcome energetic barriers
  - Classical energy function + quantum perturbation, slowly send the perturbation to zero

classical part

$$\hat{H} = E\left(\{\hat{\sigma}_j^z\}\right) - \Gamma \sum_{j=1}^{N} \hat{\sigma}_j^x$$

transverse field (send Γ to 0)

- Thus far: unclear if "true" QA really helps, compared to standard annealing, in any relevant concrete scenario

QA: Suzuki-Trotter transformation

- Partition function transformation → "effective" replicated classical Hamiltonian (with infinite replicas, $y \to \infty$)

$$H_{\text{eff}}\left(\left\{\sigma_j^a\right\}_{j,a}\right) = \underbrace{\frac{1}{y}\sum_{a=1}^{y} E\left(\left\{\sigma_j^a\right\}_j\right)}_{\text{replicated classical part}} \underbrace{- \frac{\gamma}{\beta}\sum_{a=1}^{y}\sum_{j=1}^{N}\sigma_j^a\sigma_j^{a+1}}_{\text{interaction}} - \frac{NK}{\beta}$$

replicated classical part

interaction
$\Gamma \to 0 \iff \gamma \to \infty$

$$\gamma = \frac{1}{2}\log\coth\left(\frac{\beta\Gamma}{y}\right)$$

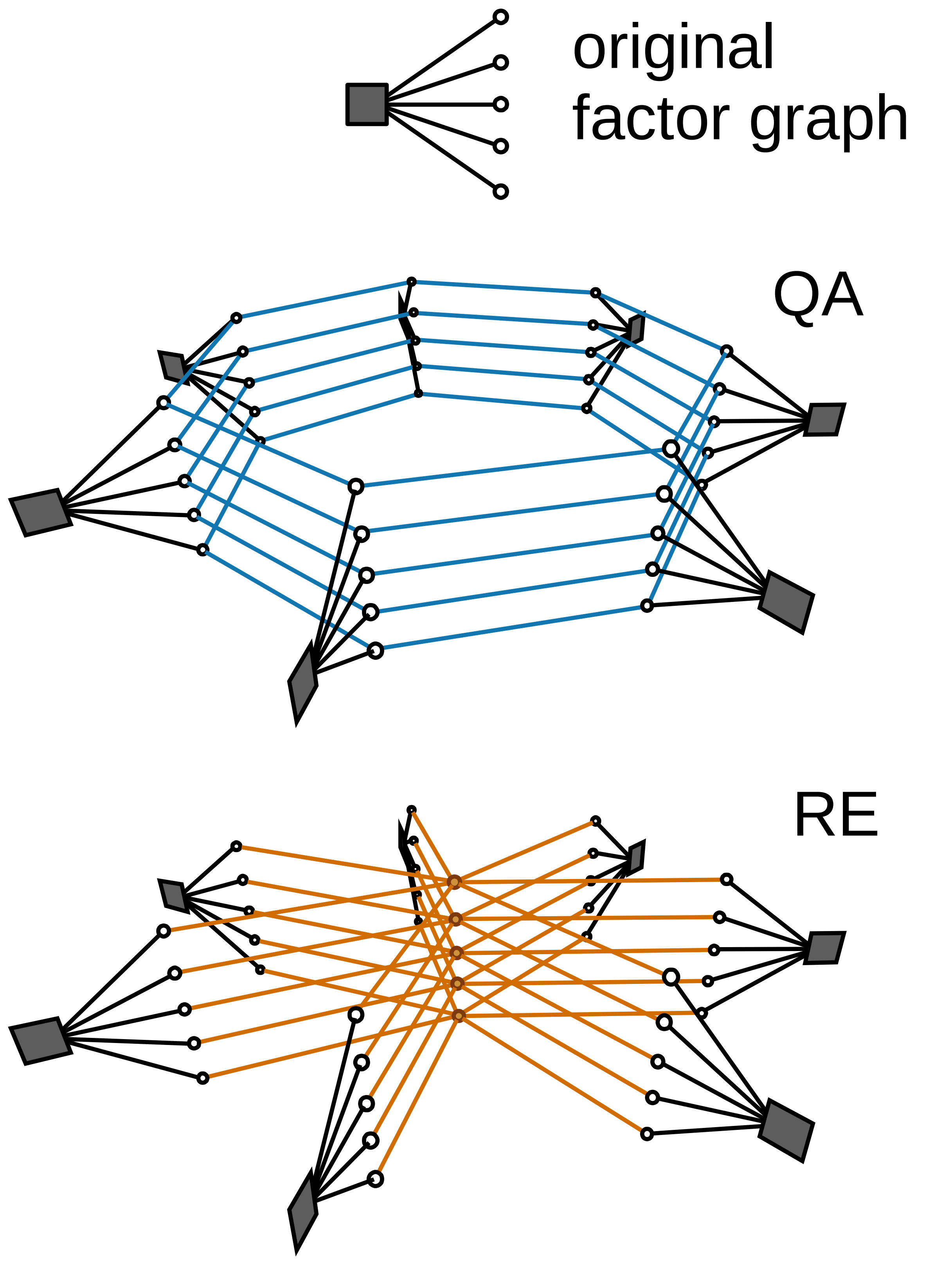
- Can be simulated with MCMC (finite y) → Quantum Simulated Annealing (QSA)

Quantum annealing vs Robust ensemble

- Effective Hamiltonian after Suzuki-Trotter transformation: very similar to the robust ensemble description...

$$H_{\text{eff}}\left(\left\{\sigma_j^a\right\}_{j,a}\right) = \frac{1}{y}\sum_{a=1}^{y} E\left(\left\{\sigma_j^a\right\}_j\right) - \frac{\gamma}{\beta}\sum_{a=1}^{y}\sum_{j=1}^{N}\sigma_j^a\sigma_j^{a+1} - \frac{NK}{\beta}$$

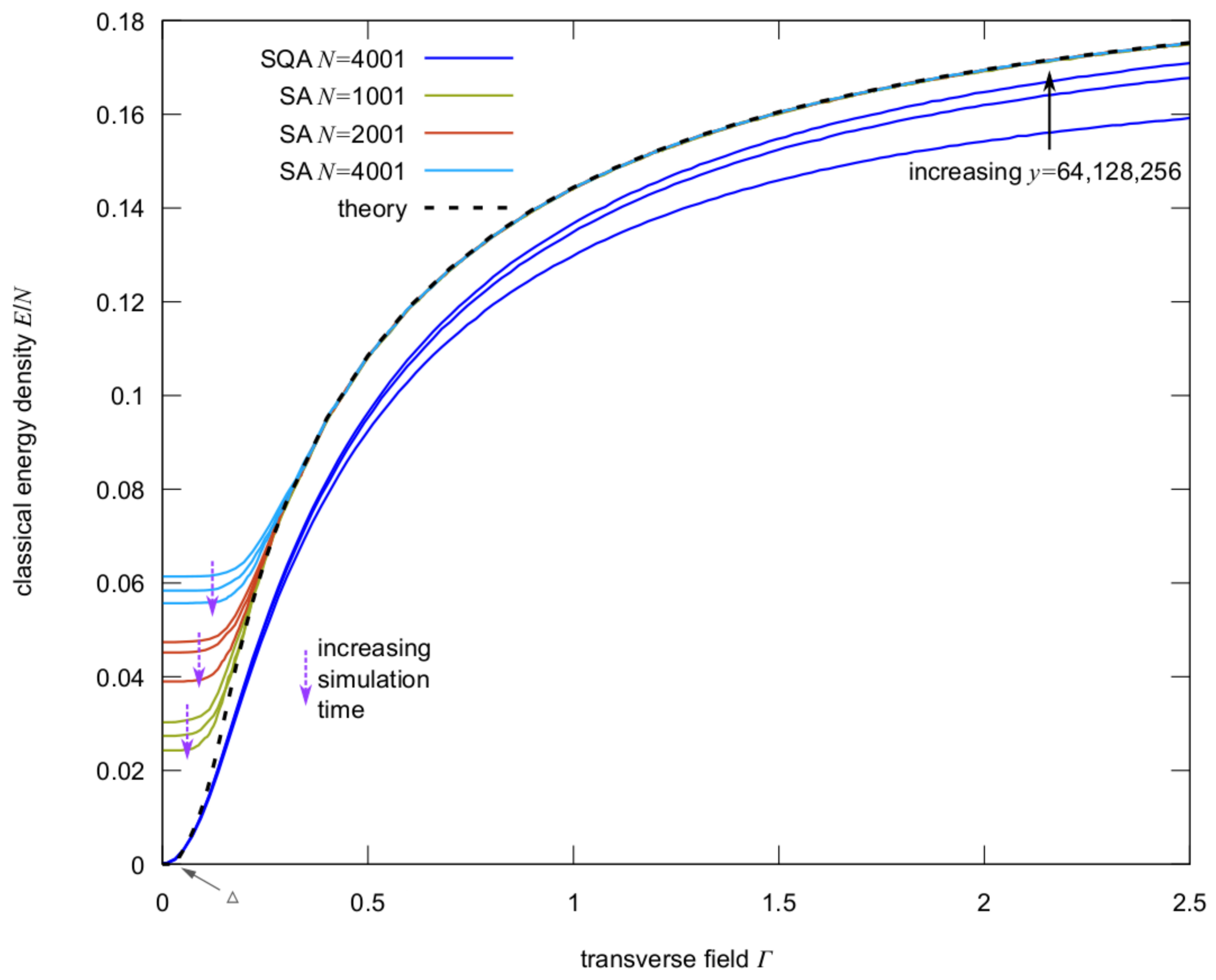
$$H_{\text{eff}}^{\text{RE}}\left(\sigma^\star, \left\{\sigma_j^a\right\}_{j,a}\right) = \sum_{a=1}^{y} E\left(\left\{\sigma_j^a\right\}_j\right) - \frac{\lambda}{\beta}\sum_{a=1}^{y}\sum_{j=1}^{N}\sigma_j^a\sigma_j^\star$$

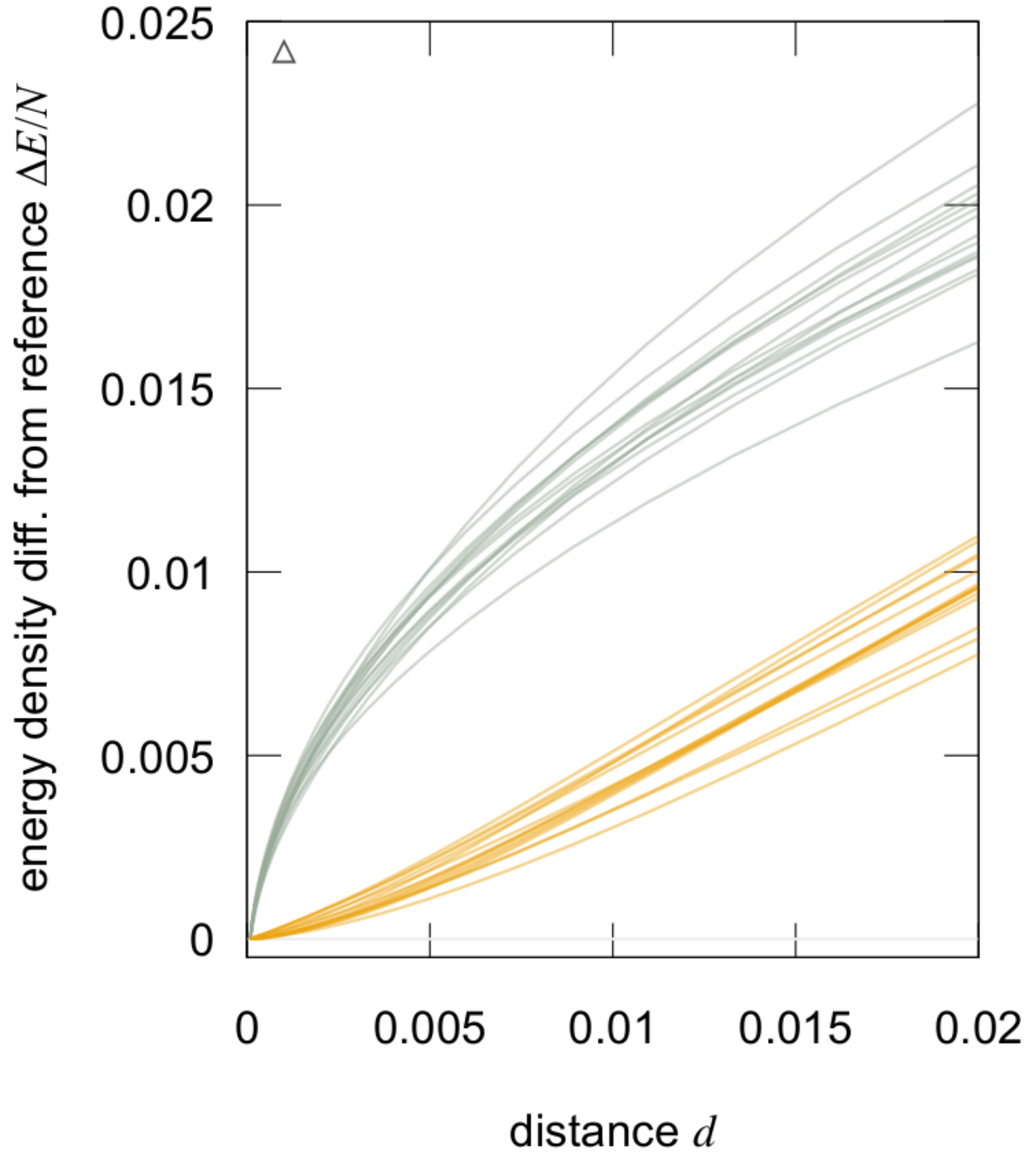


C. Baldassi, R. Zecchina, PNAS 2018

QSA on binary neural networks study

- Analytical calculations + numerical experiments + comparison with true QA in small instances
- Ends up in the dense states (exponential speed-up w.r.t. thermal annealing – a physical device would work in $\sim O(1)$...) (DWave-like)
- **QA lowers kinetic energy by delocalizing → favors dense regions**





C. Baldassi, R. Zecchina, PNAS 2018