# Understanding transformers requires new physical systems: pseudo-likelihood and vector-spin models

Francesco D'Amico

SAPIENZA
UNIVERSITÀ DI ROMA

Consiglio Nazionale
delle Ricerche

# Transformer neural network and self-attention

## Transformer is the neural building block of modern AI

- Critical in most of generative AI, like LLMs and diffusion

- Typically easy to optimize, predictable to scale

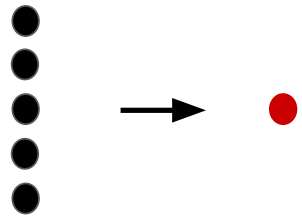
- Problems for theory: many differences with the past

  Different **task** (generative vs classification)
  Different **data** (sequences of vocabulary elements)
  Different **training procedure** (autoregressive vs fixed label)
  Different **representations** (embeddings vs scalars)

  But also different **questions**! Neural scaling laws, factual recall, ability of solving tasks, in-context learning...

# New training objective: token prediction

## Classification



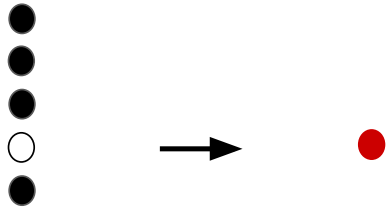
Input

$\{x_i^\mu\}_{i=1,\dots N}$

Output: supervision

$y^\mu$

## Masked Language Modeling (MLM)



Input

$\{x_i^\mu\}_{i(\neq k)}$

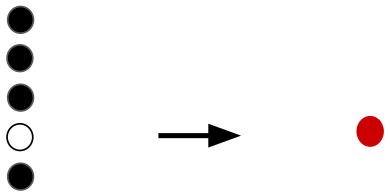
Output: self-supervision

$x_k^\mu$

What changes is the structure of data

# MLM loss

**Masked Language Modeling (MLM)**



Input

$\{x_i^\mu\}_{i(\neq k)}$

Output: self-supervision

$x_k^\mu$

- We require to learn the conditionals

$$P(x_k^\mu | \{x_i^\mu\}_{i(\neq k)})$$

- Max-entropy + conditionality:

cross-entropy loss

$$-\sum_{\mu,k} \log P(x_k^\mu | \{x_i^\mu\}_{i(\neq k)})$$

## Self-Attention operation

**Input:** sequence of N vectors in d dimension

$$(x_1, \dots, x_N) = X \in \mathbb{R}^{N \times d}$$

**Learnable parameters:**

$$W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$$

**Self-attention:**

calling $W = W_Q W_K^T$

$$A_{ij} = \frac{\exp([XWX^T]_{ij})}{\sum_{k=1}^{L} \exp([XWX^T]_{ik})} = \left[\mathrm{softmax}_{\mathrm{row}}(XWX^T)\right]_{ij}$$

**Output:**

$$O_i = \sum_{j=1}^{N} A_{ij} W_V x_j$$

Effective couplings

$$J \in \mathbb{R}^{N \times N \times d \times d}$$

# Transformer block

Many of them in parallel (heads)

$$\mathrm{MHA}(X) = \mathrm{Concat}\left(O^{(1)}, \ldots, O^{(H)}\right) W_O$$

Full transformer block

$$\begin{cases} a^{(\ell)} = X^{(\ell)} + \mathrm{MHA}\left(\mathrm{LN}\left(X^{(\ell)}\right)\right) \\ X^{(\ell+1)} = a^{(\ell)} + \mathrm{MLP}\left(\mathrm{LN}\left(a^{(\ell)}\right)\right) \end{cases}$$

With normalization

$$\mathrm{LN}(u) = \gamma \odot \frac{u - \bar{u}}{\sqrt{\sigma_u^2 + \epsilon}} + \beta$$

# Particles system interpretation

State at previous timestep

$$X^t \in \mathbb{R}^{N \times d}$$

---

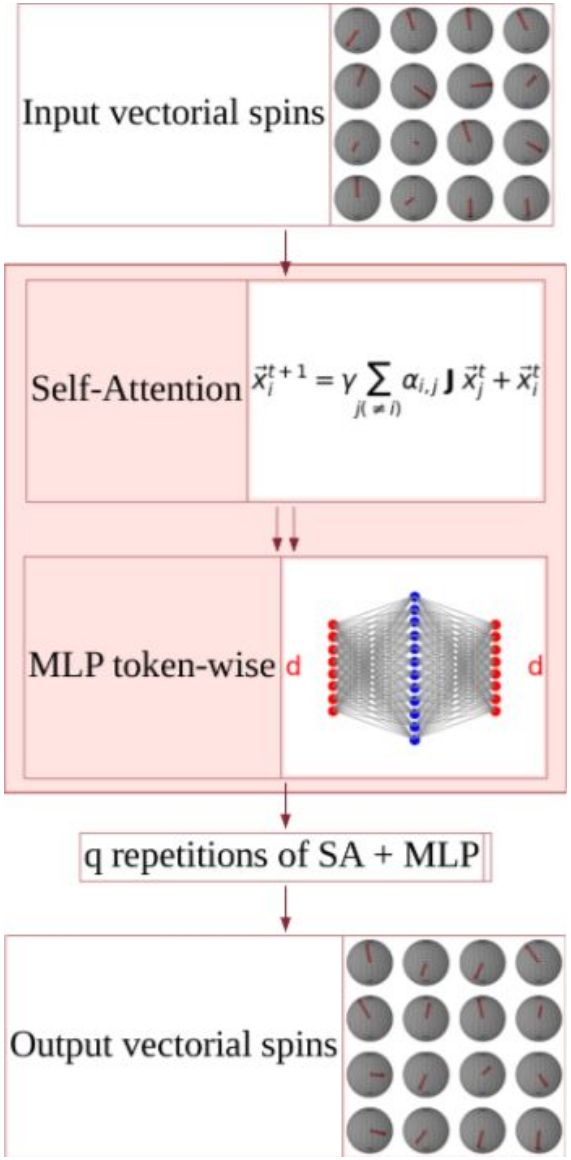
Self attention update with effective couplings

$$J \in \mathbb{R}^{N \times N \times d \times d}$$

---

New state

$$X^{t+1} \in \mathbb{R}^{N \times d}$$



**This dynamical system has not a proper energy**

$J_{ij\alpha\beta}$ asymmetric in $(i,j)$ $\Longrightarrow$ no Hamiltonian

Moreover: couplings are state-dependent

$\Longrightarrow$ Qucheng Gao, Zuyi Yang and Xiao Chen
*Clustered Attractor Manifolds and Dynamical Condensation in Self-Attention*

**Modern Hopfield networks are cross-attentions**

H. Ramsauer et al
*Hopfield Networks is All You Need*

M. Demircigil et al
*On a Model of Associative Memory with Huge Storage Capacity*

C. Lucibello, M. Mezard
*Exponential Capacity of Dense Associative Memories*

# Results

# (1)
# The role of pseudo-likelihood and recurrence in self-attention

# Question (1)

If not an energy, which cost minimized by self-attention?

What is produced by its minimization?

## Self-attention as an attractor network: transient memories without backpropagation


1st Francesco D'Amico
*Physics department*
*University of Rome Sapienza*
Rome, Italy
francesco.damico@uniroma1.it

2nd Matteo Negri
*University of Rome Sapienza, Physics department*
*and CNR Nanotech*
Rome, Italy
matteo.negri@uniroma1.it

## “Bare self-attention” simplified architecture

Dynamics $$\mathbf{x}_i^{(t+1)} = \sum_{j(\neq i)} \alpha_{i\leftarrow j} \mathbb{J}_{ij}\, \mathbf{x}_j^{(t)} + \gamma \mathbf{x}_i^{(t)}$$

where $$\alpha_{i\leftarrow j} = \mathrm{Softmax}_{[j]}(\mathbf{x}_i^{(t)} \mathbb{J}_{ij} \mathbf{x}_j^{(t)})$$

Dynamics can be written as gradient descent of a cost

$$\mathbf{x}_i^{(t+1)} = -\frac{\partial}{\partial x_i} e_i(\mathbf{x}_i, \{\mathbf{x}_j\})\Big|_{\mathbf{x}_i^{(t)}, \{\mathbf{x}_j^{(t)}\}} + \gamma \mathbf{x}_i^{(t)}$$

With cost

$$e_i(\mathbf{x}_i, \{\mathbf{x}_j\}) = -\log \sum_{j(\neq i)} \exp\left(\mathbf{x}_i \mathbb{J}_{ij} \mathbf{x}_j\right)$$

## “Bare self-attention” simplified architecture

This cost $$e_i(\mathbf{x}_i, \{\mathbf{x}_j\}) = -\log \sum_{j(\neq i)} \exp\left(\mathbf{x}_i \mathbb{J}_{ij} \mathbf{x}_j\right)$$

is exactly pseudo-likelihood training:

$$\mathcal{L}\left(X = (x_1, \ldots, x_N)\right) = \sum_i \ell(x_i | x_{/i})$$

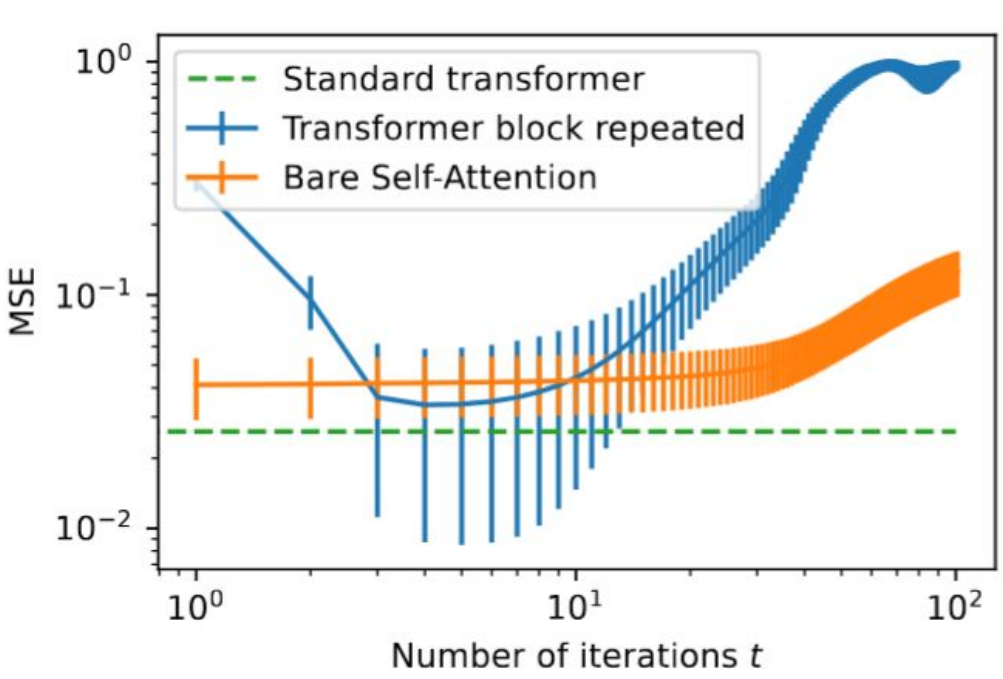
Equivalence already recognized in

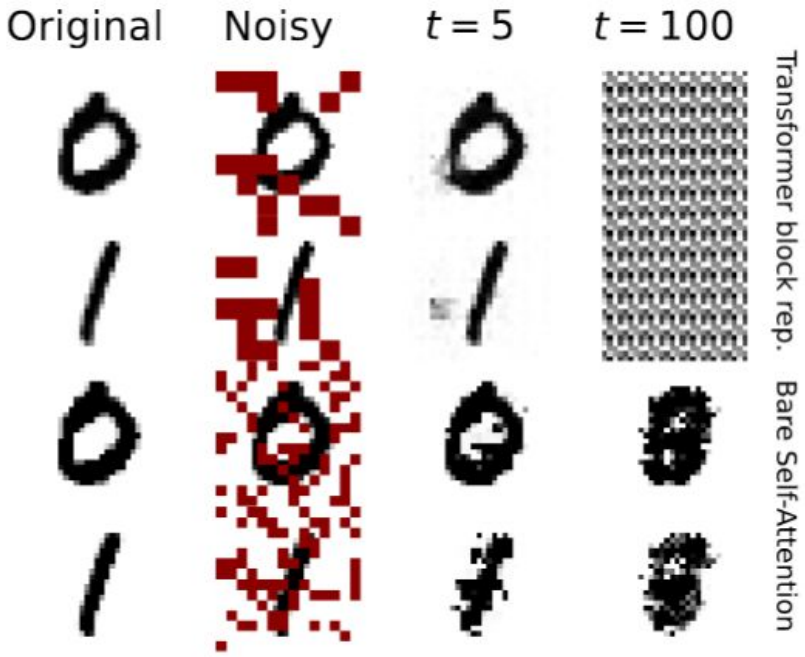
R. Rende et al
*Mapping of attention mechanisms to a generalized Potts model*

**Our experiment:**

- Pseudo-likelihood as a **training loss** (for bare self-attention)
- **Task-agnostic**: learns conditionals of one variable given the others
- **Task**: start from a noised test datapoint and run the dynamics

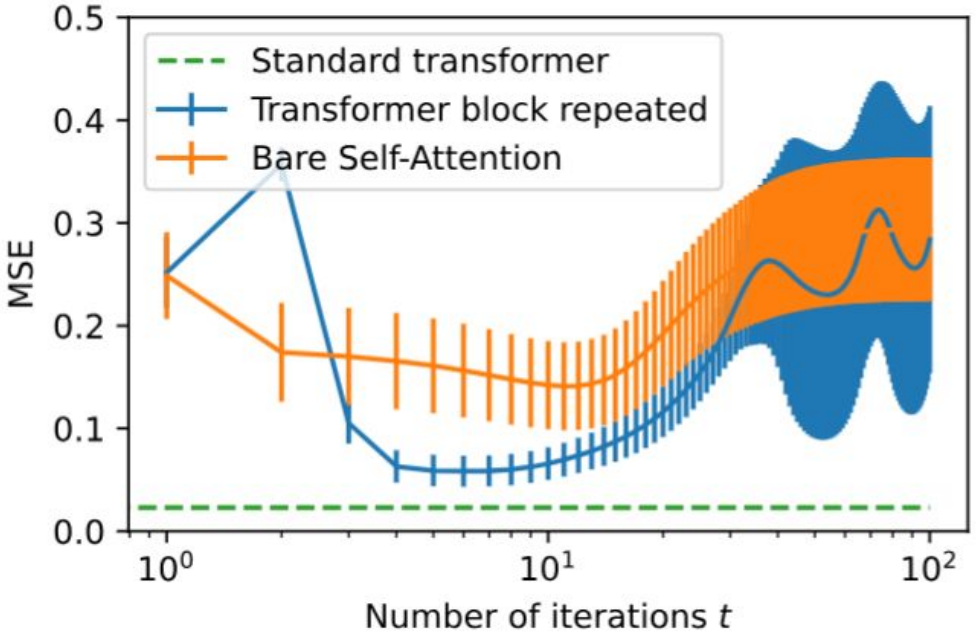
# Result: transient denoising ability

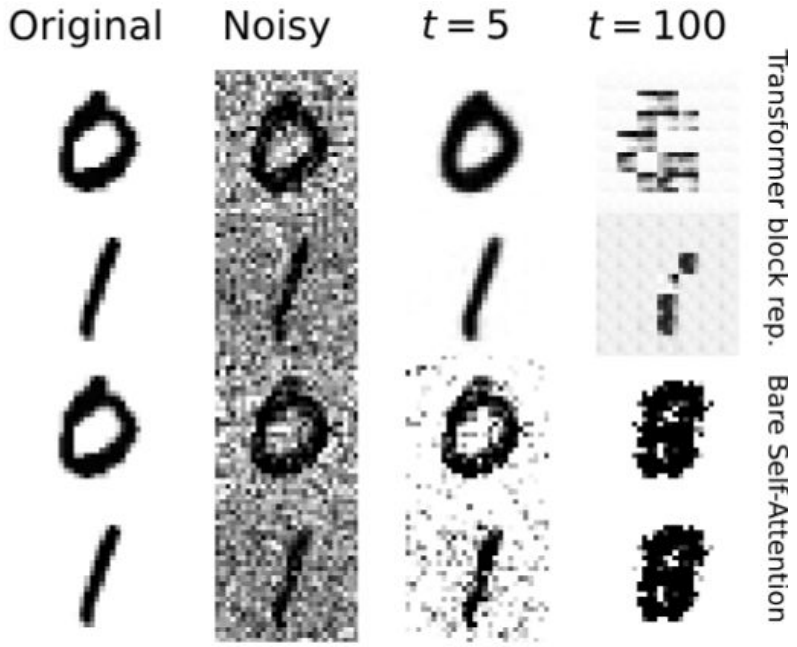



- Baseline: **5-layer transformer**
- Comparison: **iterated bare self-attention** vs **iterated transformer**

# Result: transient denoising ability




- Baseline: **5-layer transformer**
- Comparison: **iterated transformer** vs **iterated bare self-attention**

# (2)
# An analytical model to understand pseudo-likelihood

# Question (2)

Why it creates an associative memory?

How to study a recurrent physical system with energy not defined?

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa



Pseudo-likelihood produces associative memories able to generalize, even for asymmetric couplings☆

Francesco D'Amico [a,b,*], Dario Bocchi [a,b], Luca Maria Del Bono [a,b], Saverio Rossi [a], Matteo Negri [a,b,1]

[a] Physics Department, Sapienza University of Rome, Piazzale Aldo Moro 5, 00185 Rome, Italy
[b] Institute of Nanotechnology, National Research Council of Italy, CNR-NANOTEC, Rome Unit, Italy

## Analytical setting of pseudo-likelihood

Two-bodies spin system: $x_i \in \{\pm 1\}$ $J \in \mathbb{R}^{N \times N}$

with dynamics $x_i^{(t+1)} = \mathrm{sign}(\sum_{j(\neq i)}^{N} J_{ij} x_j^{(t)})$

Dataset of $P = \alpha N$ i.i.d. patterns $\xi^\mu, \qquad \mu = 1, \dots, P$

Couplings via gradient-descent on pseudo-likelihood loss:

$$\mathcal{L} = -\frac{1}{P} \sum_{i,\mu} \left[ \lambda \, \xi_i^\mu \sum_{j \neq i} J_{ij} \xi_j^\mu - \log 2 \cosh \left( \lambda \sum_{j \neq i} J_{ij} \xi_j^\mu \right) \right]$$

at fixed norms $|J_i| = \sqrt{N}$

# Analytical setting of pseudo-likelihood

Stabilities $\Delta_i^\mu = \xi_i^\mu \sum_{j(\neq i)} J_{ij}\xi_j^\mu$

dynamical update before sign

$$x_i^{(t+1)} = \text{sign}\left(\sum_{j(\neq i)}^{N} J_{ij}x_j^{(t)}\right)$$

Loss is a function of them

$$L_\lambda(J) = -\sum_{\mu=1}^{P}\sum_{i=1}^{N}\frac{1}{\lambda}\left(\lambda\Delta_i^\mu - \log 2\cosh\left(\lambda\Delta_i^\mu\right)\right)$$

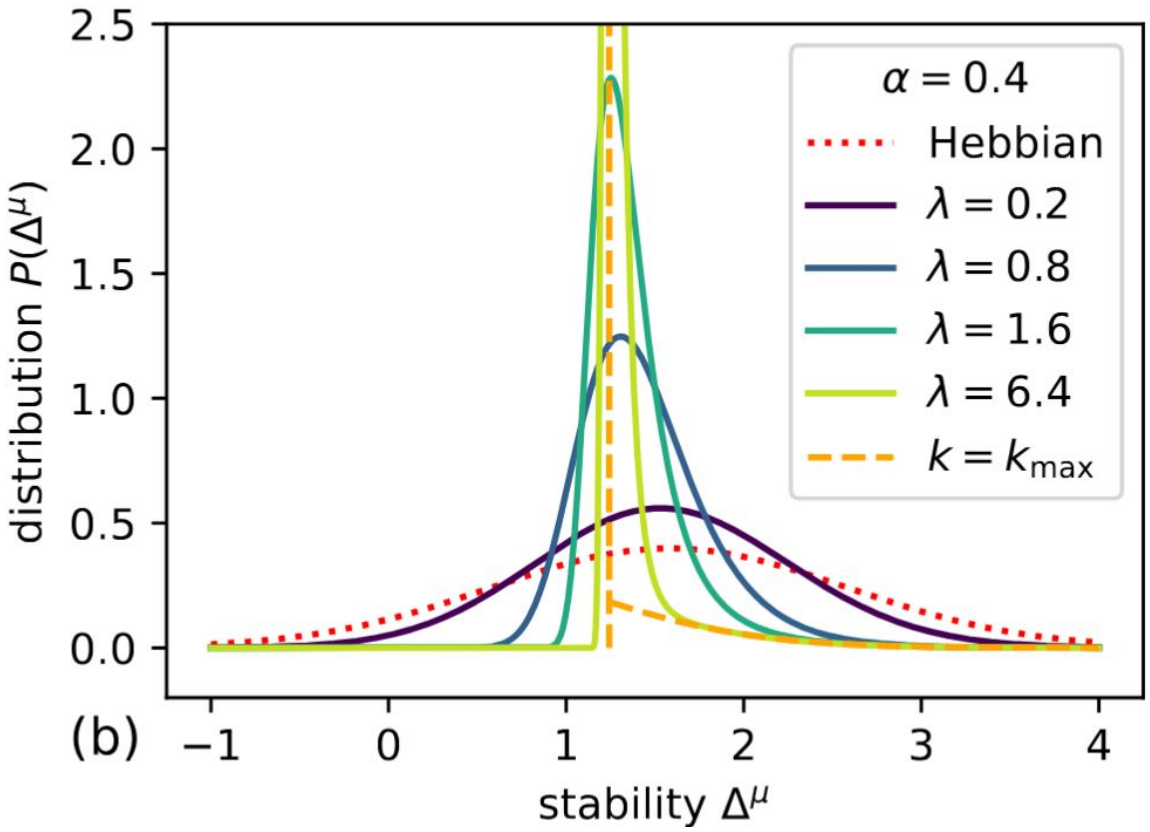
Gardner-like replica computation gives stabilities distribution

$$P_{\alpha,\,\beta\to\infty}(\Delta) = \int_{-\infty}^{\infty}\frac{dt}{\sqrt{2\pi}}e^{-t^2/2}\,\delta\Big(\Delta - \Delta_0\big(t,\bar{x}(\alpha)\big)\Big)$$

with $\Delta_0(t,x) = \arg\min_{\Delta}\left[\mathcal{L}_\lambda(\Delta) + \frac{(\Delta-t)^2}{2x}\right]$

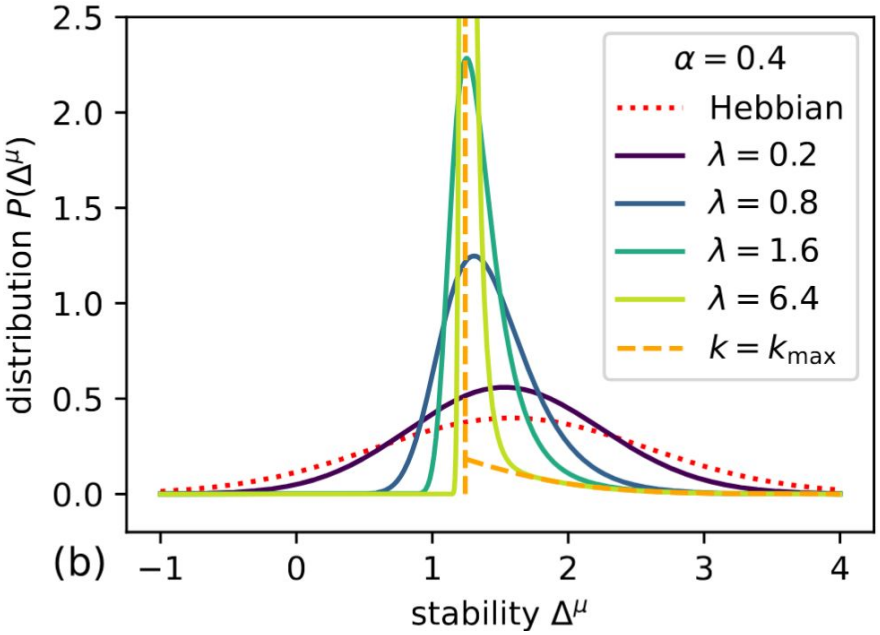
Distribution depends on weights scale

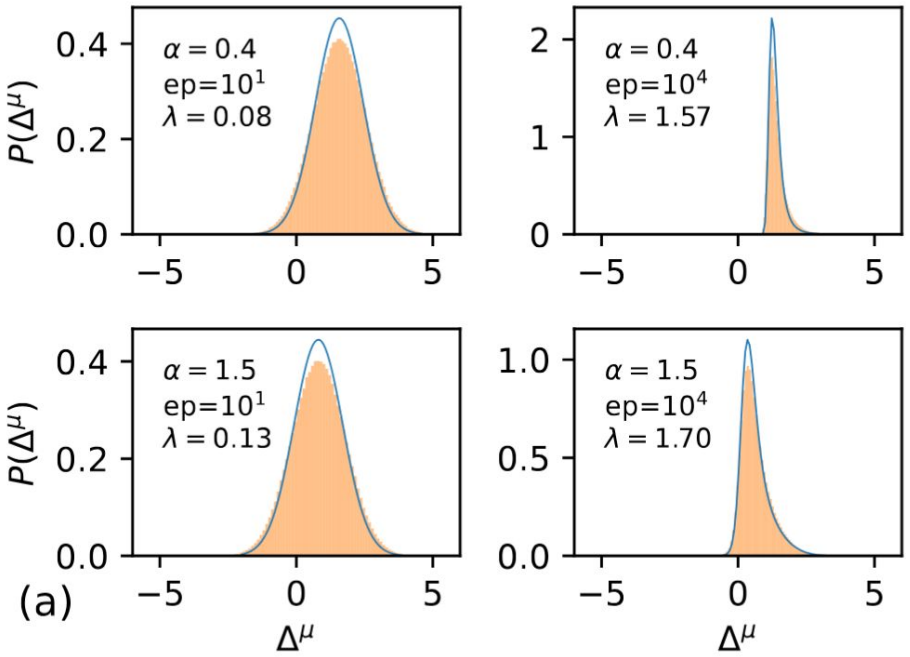
# Analytical solution




⇒ Scale interpolates from Hebb's to max-margin associative memory

# Analytical solution



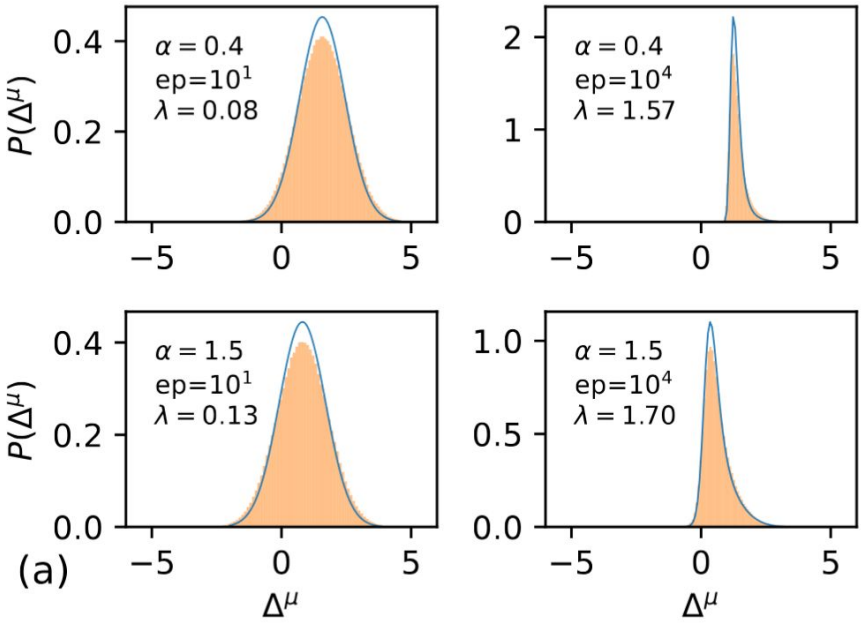



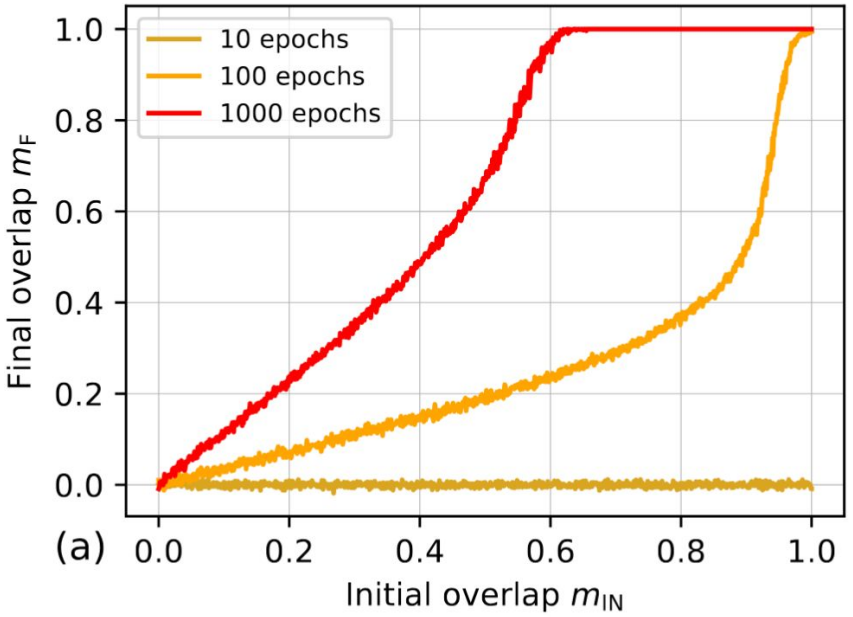

If not fixed, gradient-descent increases the scale

D. Soudry et al
*The implicit bias of gradient descent on separable data*

⇒ **dynamically, pseudo-likelihood increases Hebb's storage**

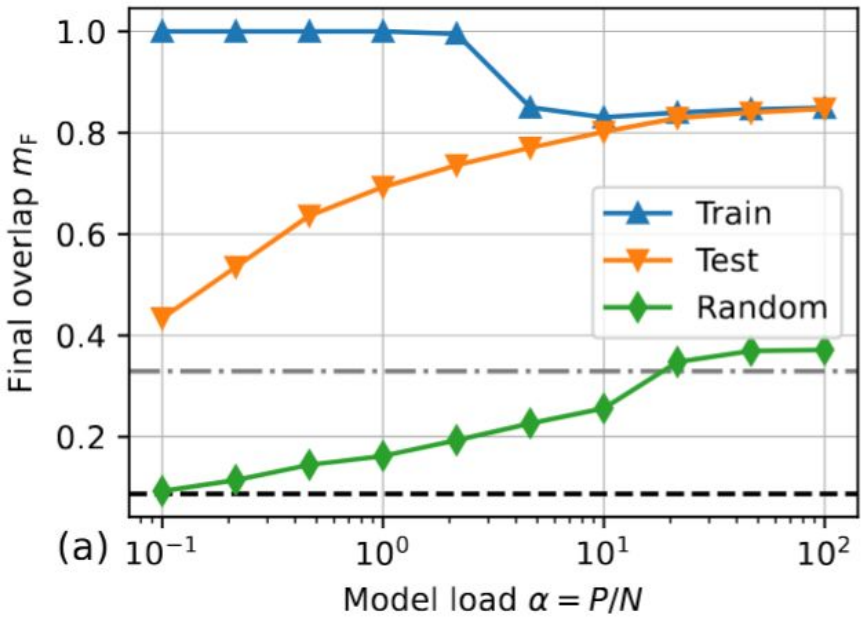
# Analytical solution



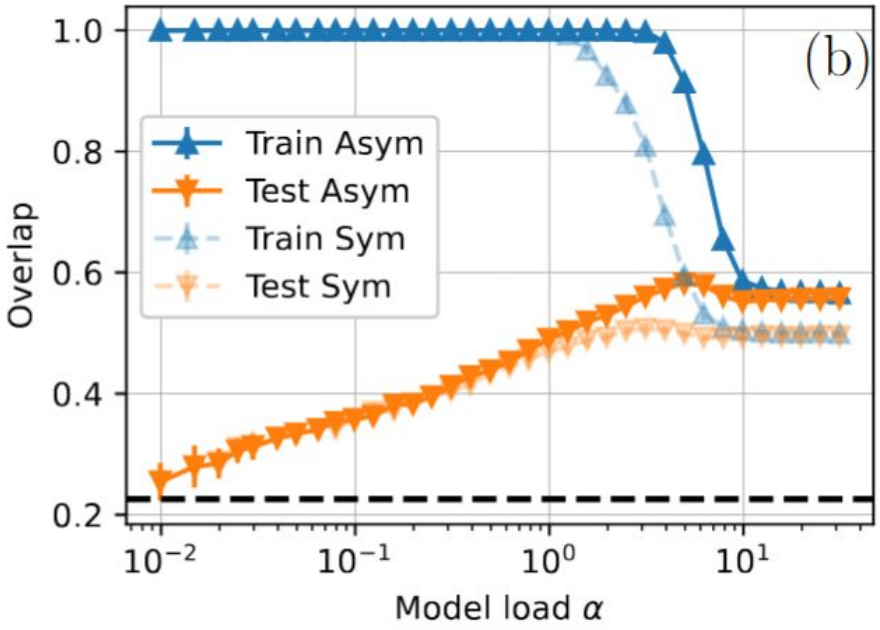



# Real data: memorization-to-generalization

Pseudo-likelihood **generalizes** with structured data (at variance with AMs)




**MNIST, binary variables**




**Beta-lactamase Potts proteins**

# (3)
# What is the effect of vector dimension

## Question (3)

What is the role of vector dimension of particles? How does it affect associative memory behavior?

How capacity and order parameters scale with d?

**Statistical mechanics of vector Hopfield network near and above saturation**

**Flavio Nicoletti**[1,2,4], **Francesco D'Amico**[2,3,*] and **Matteo Negri**[2,3,5]

[1] Department of Computer Science and Engineering, Chalmers University of Technology and University of Gothenburg, SE-41296 Gothenburg, Sweden
[2] Dipartimento di Fisica, Sapienza Università di Roma, 00185 Rome, Italy
[3] Institute of Nanotechnology, National Research Council of Italy, CNR-NANOTEC, Rome Unit, Italy

# Vector Hopfield model

Vector-spin states $(\vec{x}_1, \dots, \vec{x}_N) = \boldsymbol{X} \in \mathbb{R}^{N \times d}$

P i.i.d. vector-spin patterns $P(\xi_i^\mu) = \dfrac{1}{\mathcal{S}_d(1)} \delta(|\xi_i^\mu| - 1)$

Tensor Hebb's rule $\mathbb{J}_{ij} = \dfrac{1}{N} \sum_{\mu=1}^{P} \xi_{i;\mu} \xi_{j;\mu}^\intercal \implies \boldsymbol{J} \in \mathbb{R}^{N \times N \times d \times d}$

Dynamics $\vec{x}_i(t+1) = \dfrac{\sum_{j:j\neq i} \mathbb{J}_{ij} \vec{x}_j(t)}{|\sum_{j:j\neq i} \mathbb{J}_{ij} \vec{x}_j(t)|}$

Admit Hamiltonian $\mathcal{H} = -\dfrac{1}{2} \sum_{i\neq j}^{1,N} \vec{x}_i \cdot \mathbb{J}_{ij} \vec{x}_j$

# Analytical solution at equilibrium

**Mattis overlap** (magnetization of memories)

$$m = \overline{\int \frac{d\boldsymbol{h}\, e^{-\frac{h^2}{2\alpha r}}}{[2\pi\alpha r]^{d/2}} \frac{(\boldsymbol{h}+\sigma m\boldsymbol{\xi})\cdot\boldsymbol{\xi}}{\left|\boldsymbol{h}+\sigma m\boldsymbol{\xi}\right|} \mathrm{g}_d\left(\beta\sigma\left|\boldsymbol{h}+\sigma m\boldsymbol{\xi}\right|\right)}$$

$$= \int_0^\infty dh\; P_h(h)\mathrm{g}_d\left(\frac{m\sigma h}{\alpha r}\right)\mathrm{g}_d(\beta\sigma h).$$

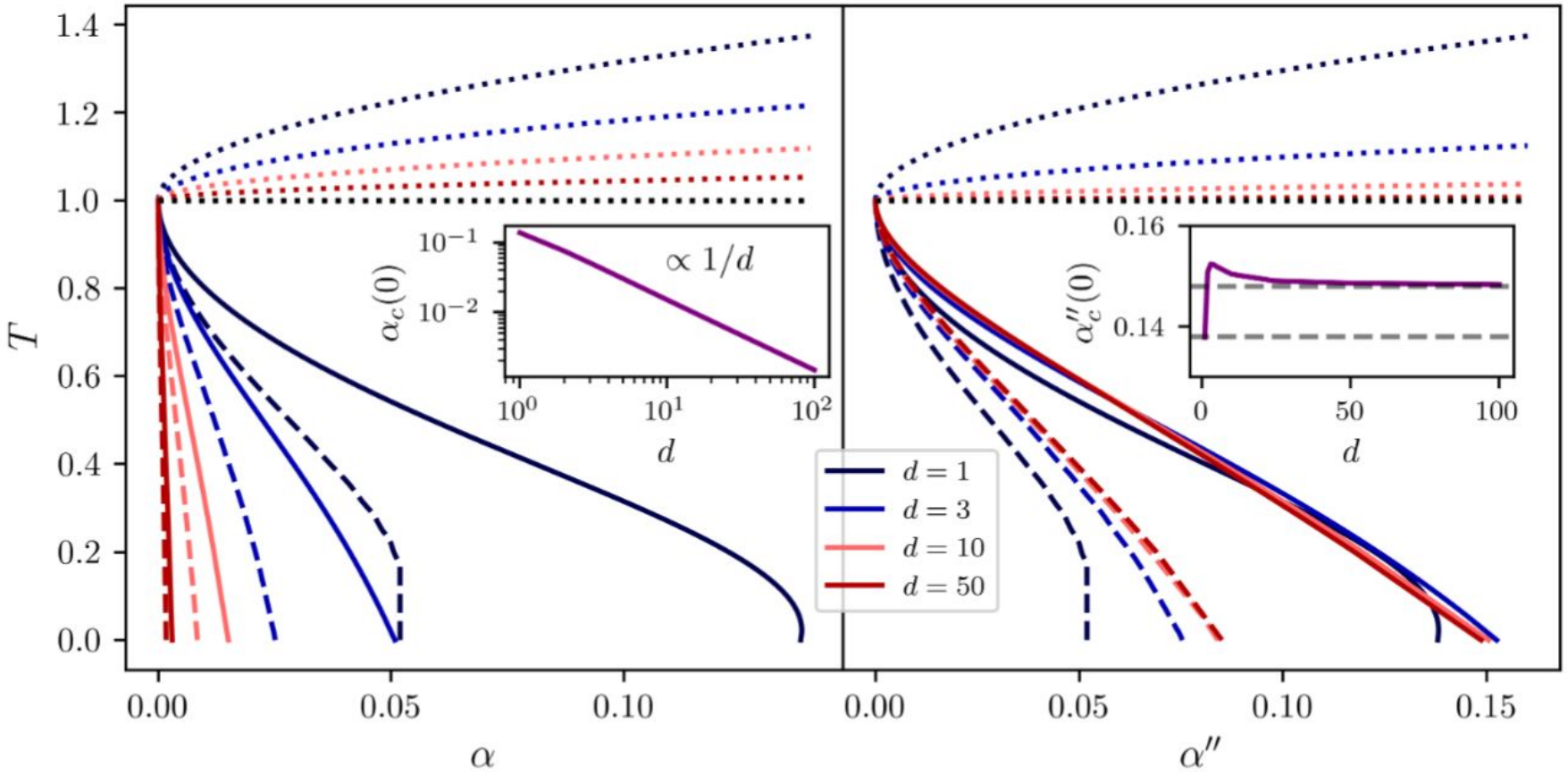
**Replica overlap**

$$q = \overline{\int \frac{d\boldsymbol{h}\, e^{-\frac{h^2}{2\alpha r}}}{[2\pi\alpha r]^{d/2}} \mathrm{g}_d\left(\beta\sigma\left|\boldsymbol{h}+\sigma m\boldsymbol{\xi}\right|\right)^2} = \int_0^\infty dh\; P_h(h)\mathrm{g}_d(\beta\sigma h)^2$$

**Noise from other memories**

$$r = \frac{q}{\sigma^2\left[\frac{d}{\sigma^2} - \beta(1-q)\right]^2}$$

# Analytical solution at equilibrium

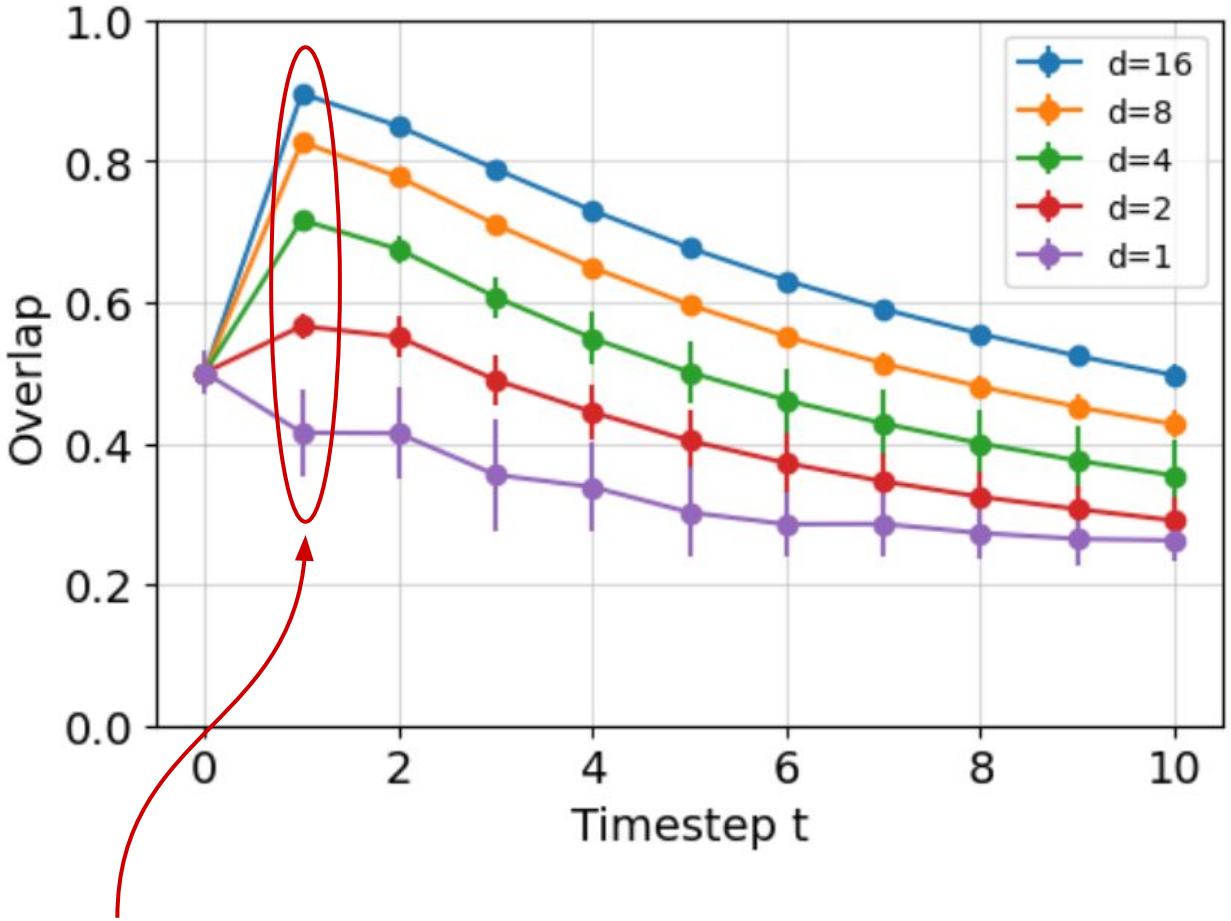



**With** $\alpha = \dfrac{P}{N}$

**With** $\alpha'' = \dfrac{Pd}{N}$

$\Rightarrow$ Dimension of vector spin decreases critical load

# Out-of-equilibrium: transient denoising




**Denoising peak** increases with d

## Out-of-equilibrium: transient denoising

At very first step $\sum_{j(\neq i)} \mathbb{J}_{ij}\vec{x}_j(0) = m_0\vec{\xi}_i^1 + \vec{\mathrm{n}}_i$

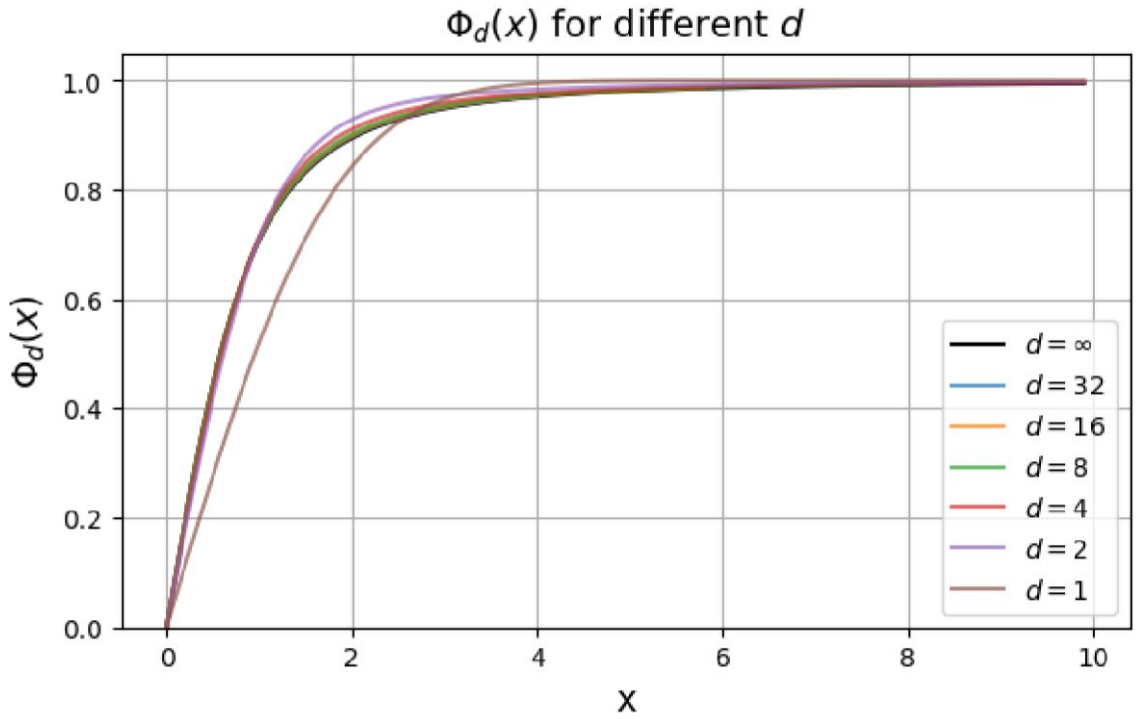
where noise $\vec{\mathrm{n}}_i$ is isotropic Gaussian, with variance $\alpha/d^2$

$\Longrightarrow$ predicted first-step overlap

$$m_1 = \frac{1}{N}\sum_{i=1}^{N}\vec{\xi}_i^1 \cdot \vec{x}_i(1) = \int \frac{d\vec{\mathrm{n}}\, e^{-\frac{|\vec{\mathrm{n}}|^2 d}{2(\alpha/d)}}}{[(2\frac{\pi}{d})(\frac{\alpha}{d})]^{d/2}} \overline{\frac{m_0 + \vec{\mathrm{n}}\cdot\vec{\xi}}{|m_0 + \vec{\mathrm{n}}\cdot\vec{\xi}|}}$$

$$= \Phi_d\left(\frac{m_0}{\sqrt{\alpha/d}}\right) = \Phi_d(x)$$

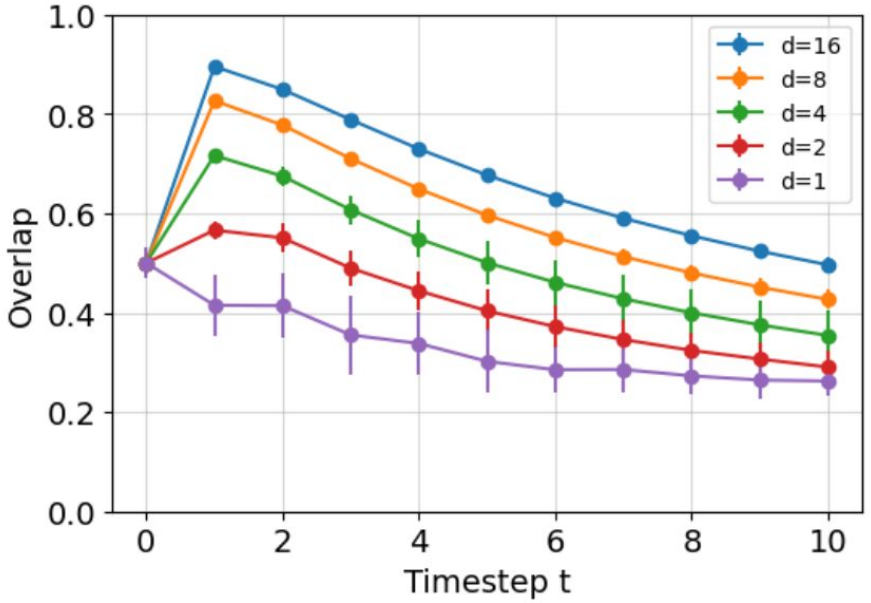
First-step overlap $m_1 = \Phi_d\left(\frac{m_0}{\sqrt{\alpha/d}}\right) = \Phi_d(x)$

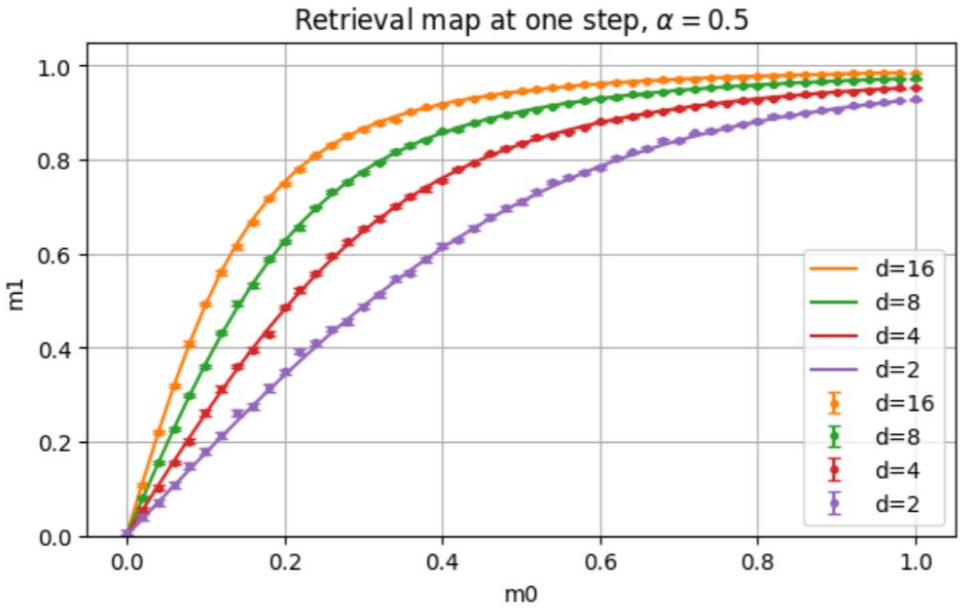



Correct scaling $\alpha' = \frac{\alpha}{d} = \frac{P}{Nd}$

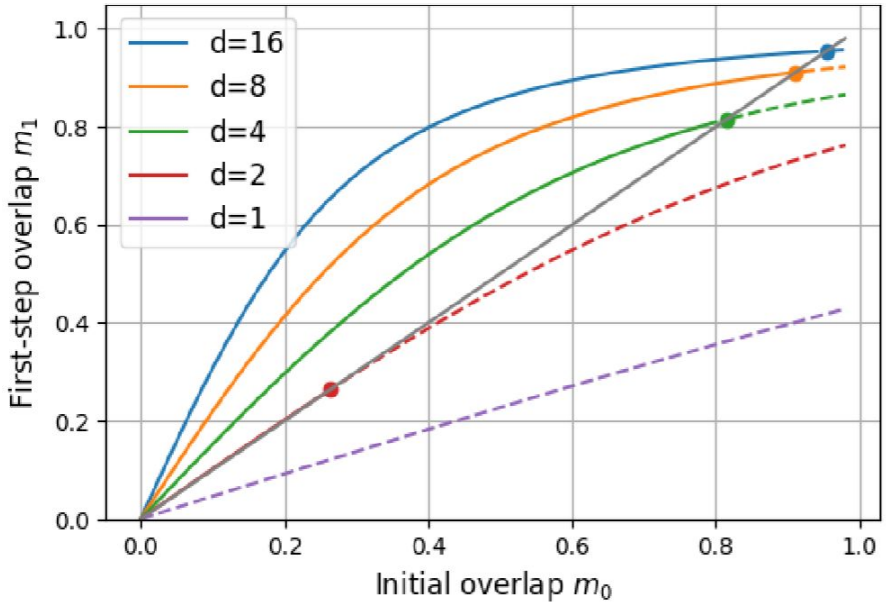
$\Rightarrow$ **dimension enhances transient denoising**

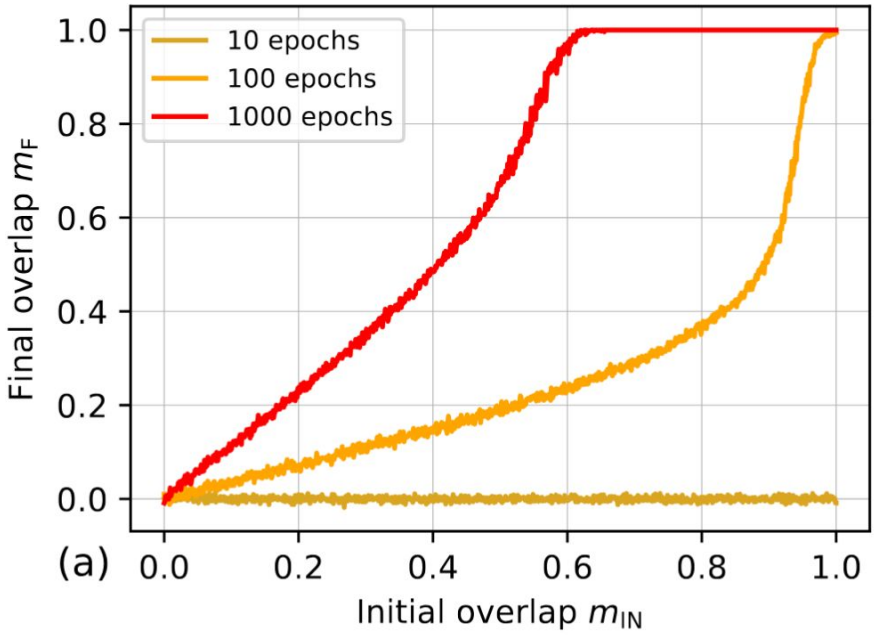
# Out-of-equilibrium: transient denoising



# Out-of-equilibrium: transient denoising




**Transient denoising**

**Associative memory**

Denoising genuinely new phenomenon

# Conclusion

1) Particle system interpretation leads to physical models addressable with statistical mechanics

2) Self-attention minimizes a pseudo-likelihood

3) In models considered, pseudo-likelihood produces associative memories able to generalize

4) New transient-denoising phenomenon enhanced by spin dimension

Francesco D'Amico

SAPIENZA
UNIVERSITÀ DI ROMA

Consiglio Nazionale
delle Ricerche

# Back-up

# Generative Pre-trained Transformers (GPT)

## Next-token prediction trick




- Autoregressive loss: $-\sum_{\mu,T} \log P(x_T^\mu | \{x_t^\mu\}_{t=1,\ldots,T})$

- **Training:**

  same network does all the predictions at once → ***causal masking***

  need the information of token position in the sequence → ***positional encoding***

  network fed with fixed-length subsequences → ***context window***

- **Generation:**

  sequential generation of one token at a time starting from a ***prompt*** subsequence