A statistical physics approach to compressed sensing

Florent Krzakala
ESPCI, PCT and Gulliver CNRS

in collaboration with
Marc Mézard & **François Sausset** (LPTMS)
Yifan Sun (ESPCI) and **Lenka Zdeborová** (IPhT Saclay)

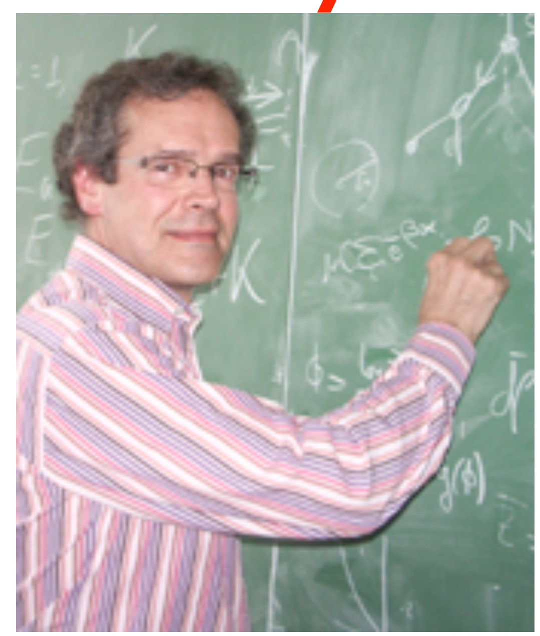
http://aspics.krzakala.org

Who are we? What do we do?

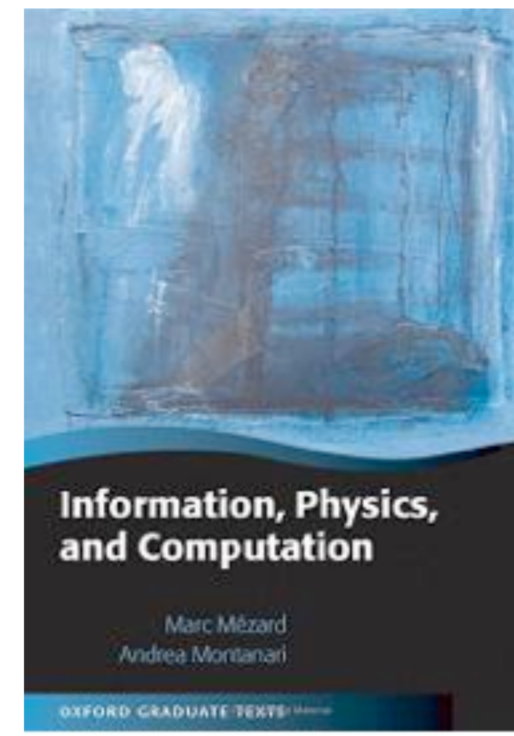
Interface between statistical physics, optimization, information theory and algorithms

Florent Krzakala
(ESPCI, Paris)

Lenka Zdeborová
(CNRS, Saclay)



Marc Mézard
(CNRS, Orsay)



http://aspics.krzakala.org

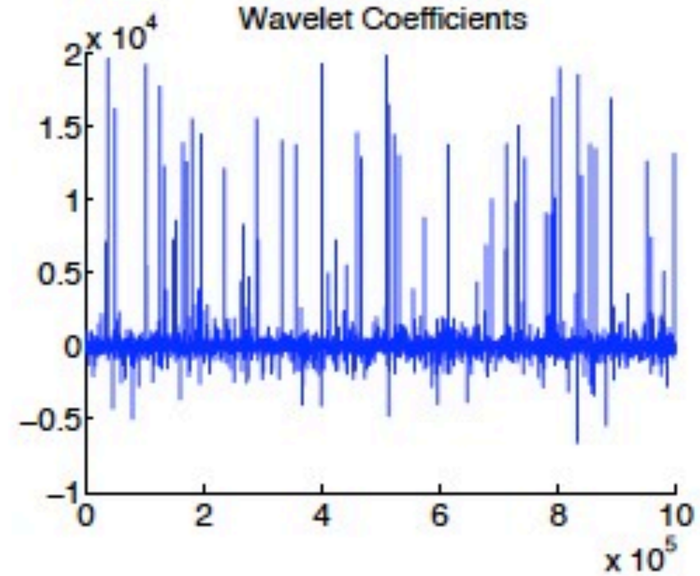
A statistical physics approach to compressed sensing

Florent Krzakala
ESPCI, PCT and Gulliver CNRS

in collaboration with
Marc Mézard & **François Sausset** (LPTMS)
Yifan Sun (ESPCI) and **Lenka Zdeborova** (IPhT Saclay)

http://aspics.krzakala.org

Sparse signals: what is compressed sensing?



From 10^6 wavelet coefficients, keep 25.000

Most signal of interest are sparse in an appropriated basis

⇒Exploited for data compression (JPEG2000).

Why do we record a huge amount of data, and then keep only the important bits?

Couldn't we record <u>only</u> the relevant information directly?

How does compressed sensing work?

How does compressed sensing work?

Image I

nxn pixels

vector of size N=n×n

$$\vec{I} = \begin{pmatrix} I^1 \\ . \\ . \\ . \\ . \\ I^N \end{pmatrix}$$

How does compressed sensing work?

M measurements

=

M linear operations on the vector

Image I

nxn pixels

vector of size $N=n\times n$

$$\vec{I} = \begin{pmatrix} I^1 \\ . \\ . \\ . \\ . \\ I^N \end{pmatrix}$$

How does compressed sensing work?

M measurements
=
M linear operations on the vector

Image I

nxn pixels

vector of size
N=n×n

$$\vec{I} = \begin{pmatrix} I^1 \\ . \\ . \\ . \\ . \\ I^N \end{pmatrix}$$

$$\vec{y} = \mathrm{G}\vec{\mathrm{I}}$$

G=M×N matrix

$$\vec{y} = \begin{pmatrix} y^1 \\ . \\ . \\ . \\ . \\ y^M \end{pmatrix}$$

vector of size M

How does compressed sensing work?

M measurements

=

M linear operations on the vector

Image I

nxn pixels

vector of size $N=n\times n$

$$\vec{I} = \begin{pmatrix} I^1 \\ . \\ . \\ . \\ . \\ I^N \end{pmatrix}$$

$$\vec{y} = G\vec{I}$$

G=M×N matrix

$$\vec{y} = \begin{pmatrix} y^1 \\ . \\ . \\ . \\ . \\ y^M \end{pmatrix}$$

vector of size M

Problem: you know y and G, how to reconstruct I ?

How does compressed sensing work?

M measurements
=
M linear operations on the vector

Image I

nxn pixels

vector of size N=n×n

$$\vec{I} = \begin{pmatrix} I^1 \\ . \\ . \\ . \\ . \\ I^N \end{pmatrix}$$

$$\vec{y} = G\vec{I}$$

G=M×N matrix

$$\vec{y} = \begin{pmatrix} y^1 \\ . \\ . \\ . \\ . \\ y^M \end{pmatrix}$$

vector of size M

Problem: you know y and G, how to reconstruct I ?

If M=N ☞ easy, just use: I $=G^{-1}y$

How does compressed sensing work?

M measurements
=
M linear operations on the vector

Image I

vector of size N=$n \times n$

nxn pixels

$$\vec{I} = \begin{pmatrix} I^1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ I^N \end{pmatrix}$$

$$\vec{y} = \mathrm{G}\vec{\mathrm{I}}$$

G=M×N matrix

$$\vec{y} = \begin{pmatrix} y^1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ y^M \end{pmatrix}$$

vector of size M

Problem: you know y and G, how to reconstruct I ?

If M<N ☞ under-constrained system of equations

Many solutions are possible

How does compressed sensing work?

M measurements
=
M linear operations on the vector

Image I

nxn pixels

vector of size N=n×n

$$\vec{I} = \begin{pmatrix} I^1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ I^N \end{pmatrix}$$

$$\vec{y} = G\vec{I}$$

G=M×N matrix

$$\vec{y} = \begin{pmatrix} y^1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ y^M \end{pmatrix}$$

vector of size M

The idea of compressed sensing is to use the a-priori knowledge that the signal is sparse in <u>some</u> appropriate basis

Problem: you know y and G, how to reconstruct I ?

If M<N ☞ under-constrained system of equations
Many solutions are possible

How does compressed sensing work?

M measurements
=
M linear operations on the vector

Image I

nxn pixels

vector of size N=n×n

$$\vec{I} = \begin{pmatrix} I^1 \\ . \\ . \\ . \\ . \\ I^N \end{pmatrix}$$

$$\vec{y} = G\vec{I}$$

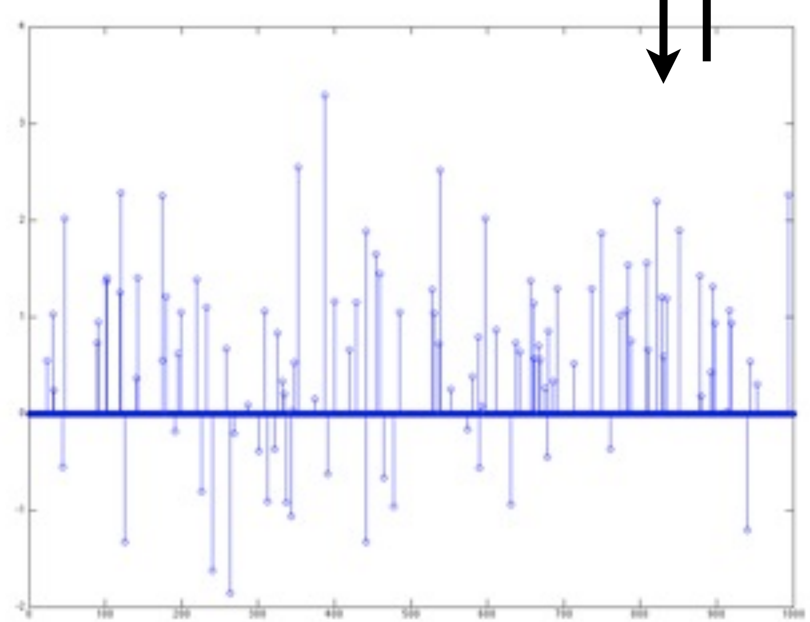
G=M×N matrix

$$\vec{y} = \begin{pmatrix} y^1 \\ . \\ . \\ . \\ . \\ y^M \end{pmatrix}$$

vector of size M

N×N matrix
Direct and inverse
Wavelet transforms

ψ^{-1} ψ



How does compressed sensing work?

M measurements
=
M linear operations on the vector

$$\vec{y} = \mathrm{G}\vec{\mathrm{I}}$$

G=M×N matrix

$$\vec{y} = \begin{pmatrix} y^1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ y^M \end{pmatrix}$$

vector of size M

Image I

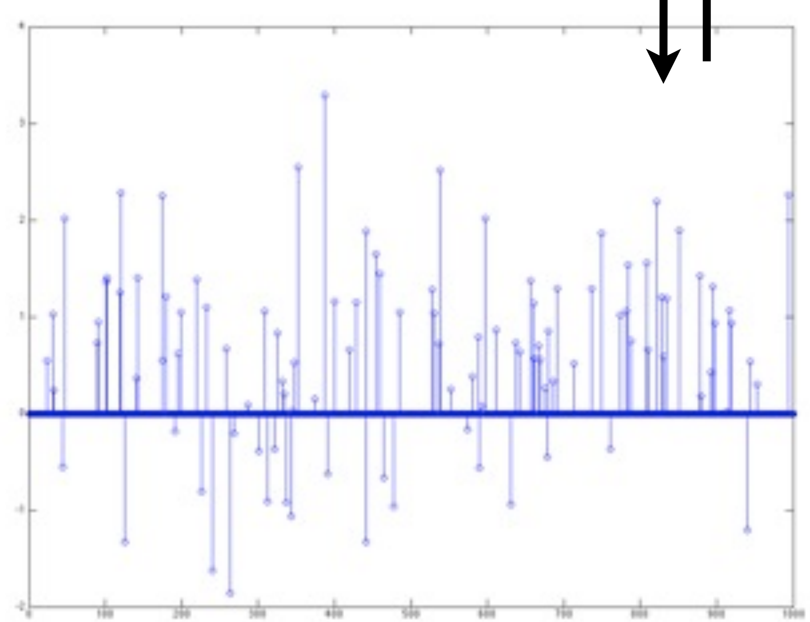
nxn pixels

vector of size N=n×n

$$\vec{I} = \begin{pmatrix} I^1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ I^N \end{pmatrix}$$

N×N matrix
Direct and inverse
Wavelet transforms

ψ^{-1} ψ



Sparse vector
of size N=n×n

$$\vec{x} = \begin{pmatrix} x^1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x^N \end{pmatrix}$$

How does compressed sensing work?

M measurements
=
M linear operations on the vector

Image I

nxn pixels

vector of size N=n×n

$$\vec{I} = \begin{pmatrix} I^1 \\ . \\ . \\ . \\ . \\ I^N \end{pmatrix}$$

$$\vec{y} = G\vec{I}$$

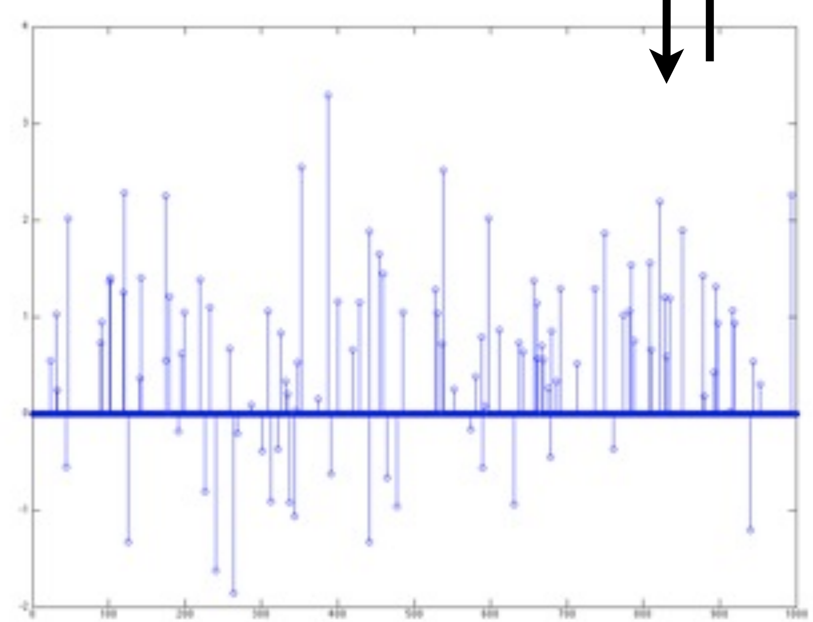
G=M×N matrix

$$\vec{y} = \begin{pmatrix} y^1 \\ . \\ . \\ . \\ . \\ y^M \end{pmatrix}$$

vector of size M

N×N matrix
Direct and inverse
Wavelet transforms

ψ^{-1} ψ



Sparse vector of size N=n×n

$$\vec{x} = \begin{pmatrix} x^1 \\ . \\ . \\ . \\ . \\ x^N \end{pmatrix}$$

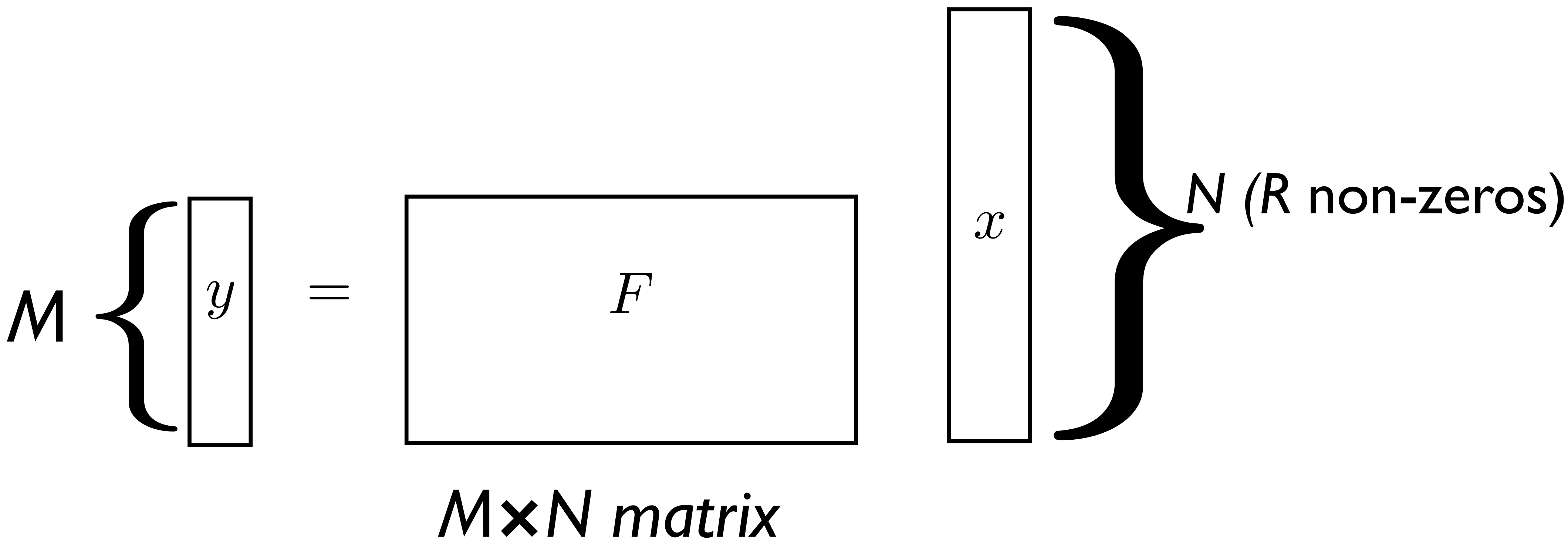
The problem to solve is now

$$\vec{y} = F\vec{x}$$

with $F = G\psi$

F=M×N matrix

How does compressed sensing work?



The problem to solve is now

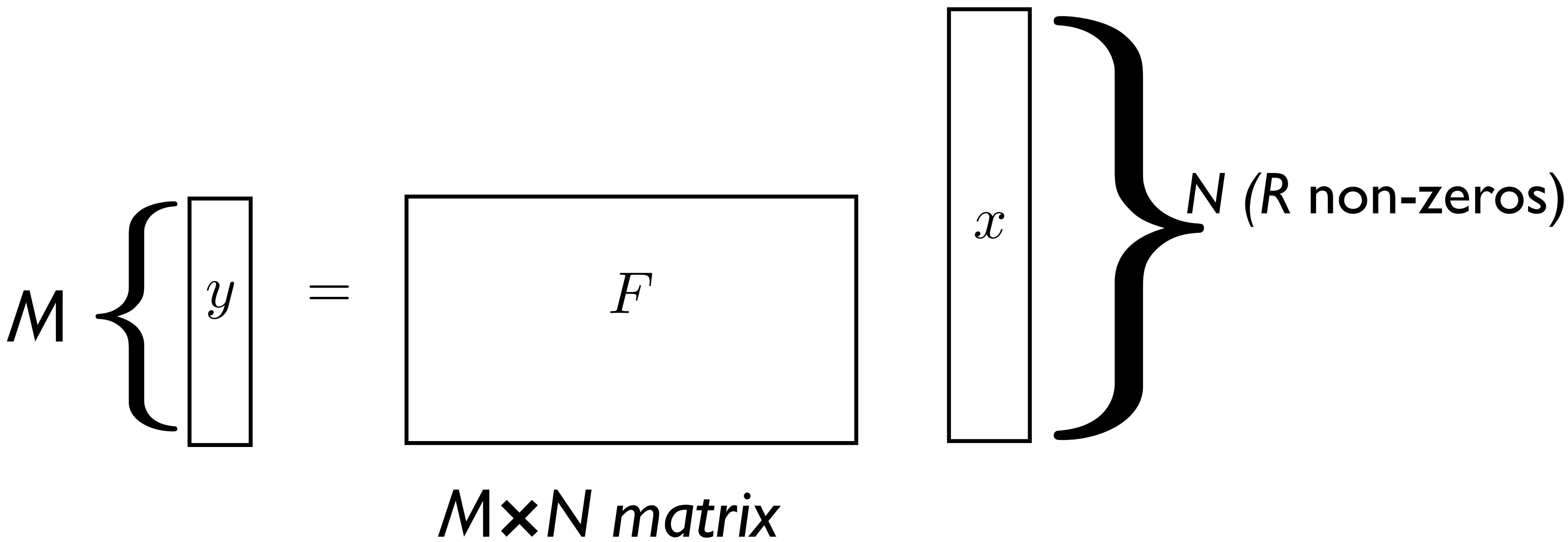
$$\vec{y} = F\vec{x}$$

with $F = G\psi$

F=M×N matrix

- Needs for a solver that finds sparse solutions of an under-constrained set of equations
- Ideally works as long as M>R
- Robust to noise

State of the art in CS



- Incoherent samplings (i.e. a random matrix F)
- Reconstruction by minimizing the L1 norm $||\vec{x}||_{L1} = \sum_i |x_i|$

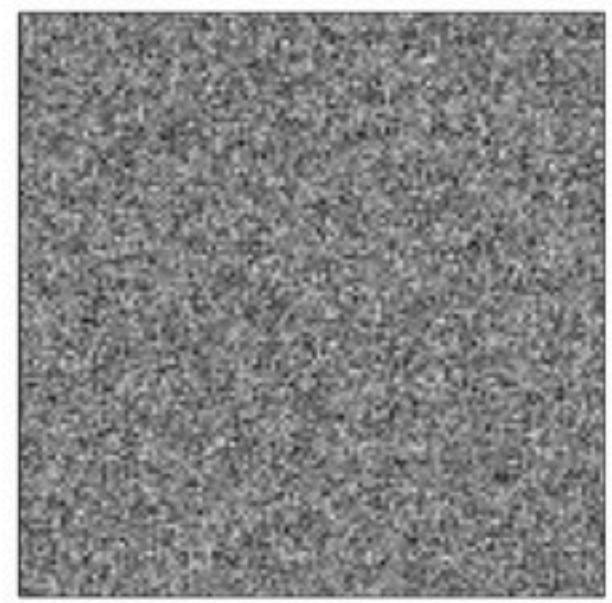
Candès & Tao (2005)
Donoho and Tanner (2005)

Example: measuring a picture

One measurement (scaling product with a random pattern)

$y_k = \langle$

,

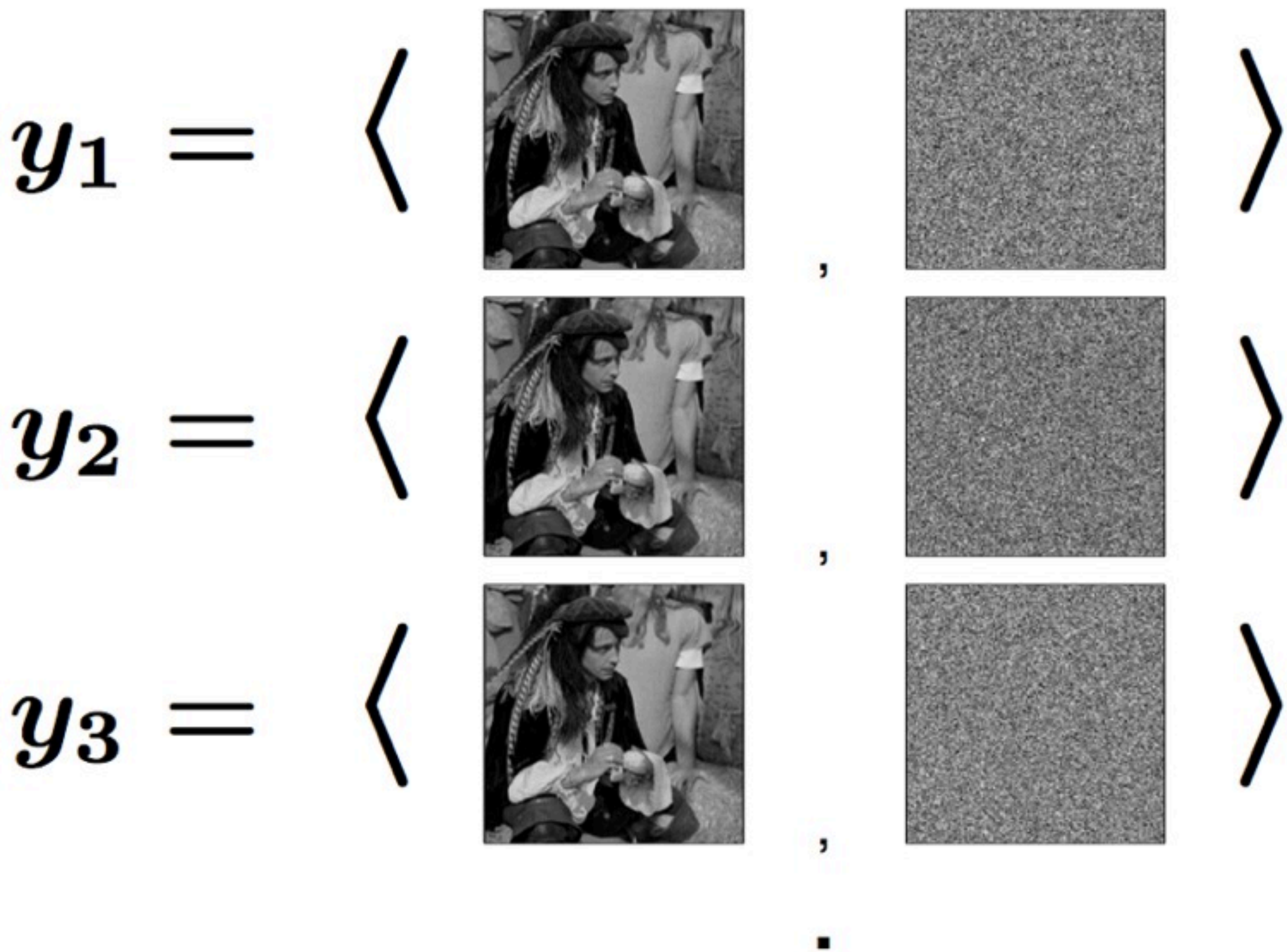


$\rangle$

- Each measurement touches every part of the underlying signal/image

Example: measuring a picture

Many measurements (scaling product with many random patterns)



Example: measuring a picture

Random matrix

$$[\,y\,] = [\,\mathbf{G}\,]\,[\,\mathbf{I}\,]$$

Measurements

signal

Example: measuring a picture

From 10^6 points, but only, 25.000 non zero

- Take $K = 96000$ incoherent measurements $y = \mathbf{G}\mathbf{I}$

- Solve

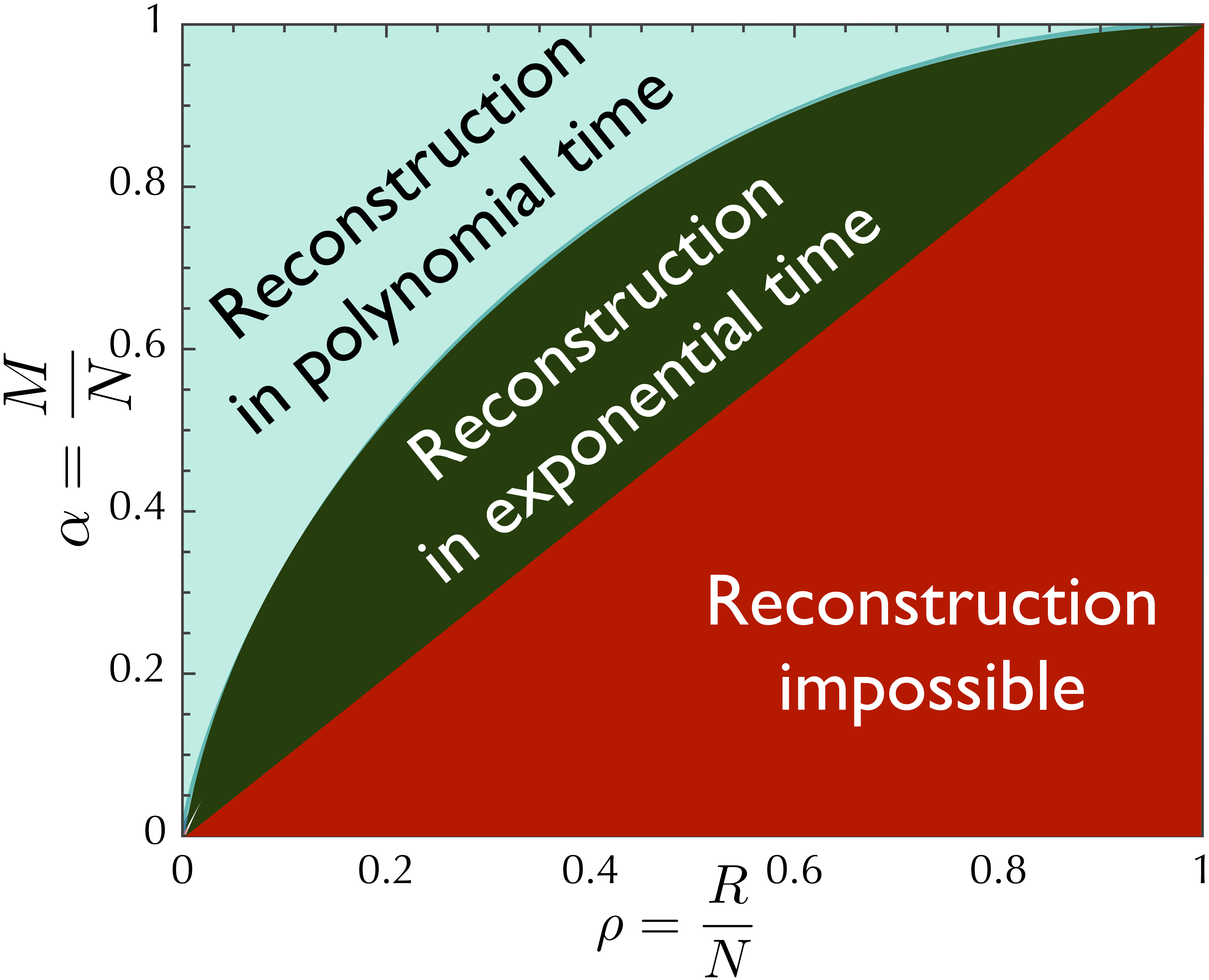
$$\min \|\mathbf{x}\|_{\ell_1} \quad \text{subject to} \quad \mathbf{G}\Psi\mathbf{x} = y$$

Ψ = wavelet transform

original (25k wavelets)

perfect recovery

State of the art in CS

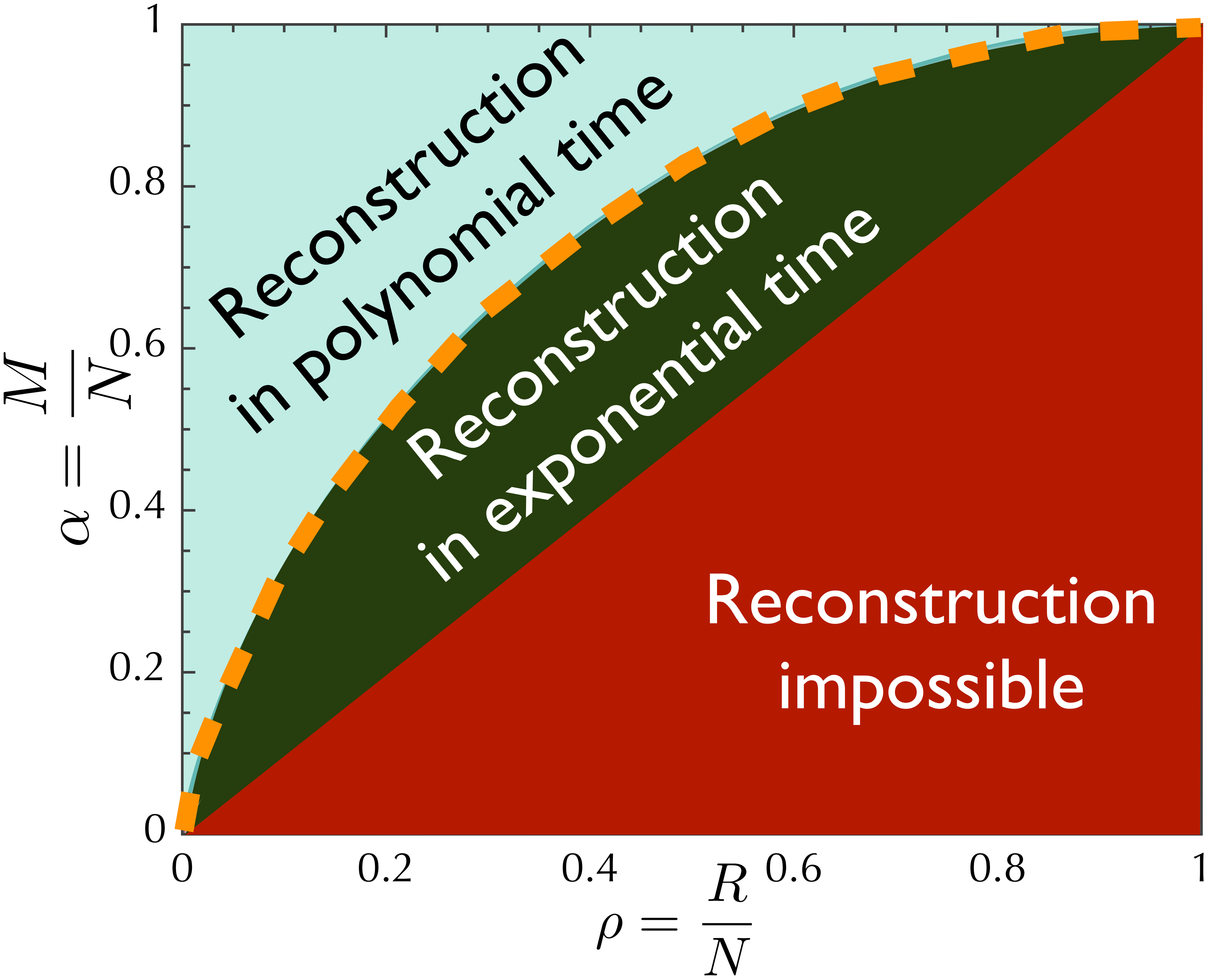


For a signal with

- (1-ρ)N zeros
- R=ρN non zeros

and a random iid matrix with M =α N

State of the art in CS

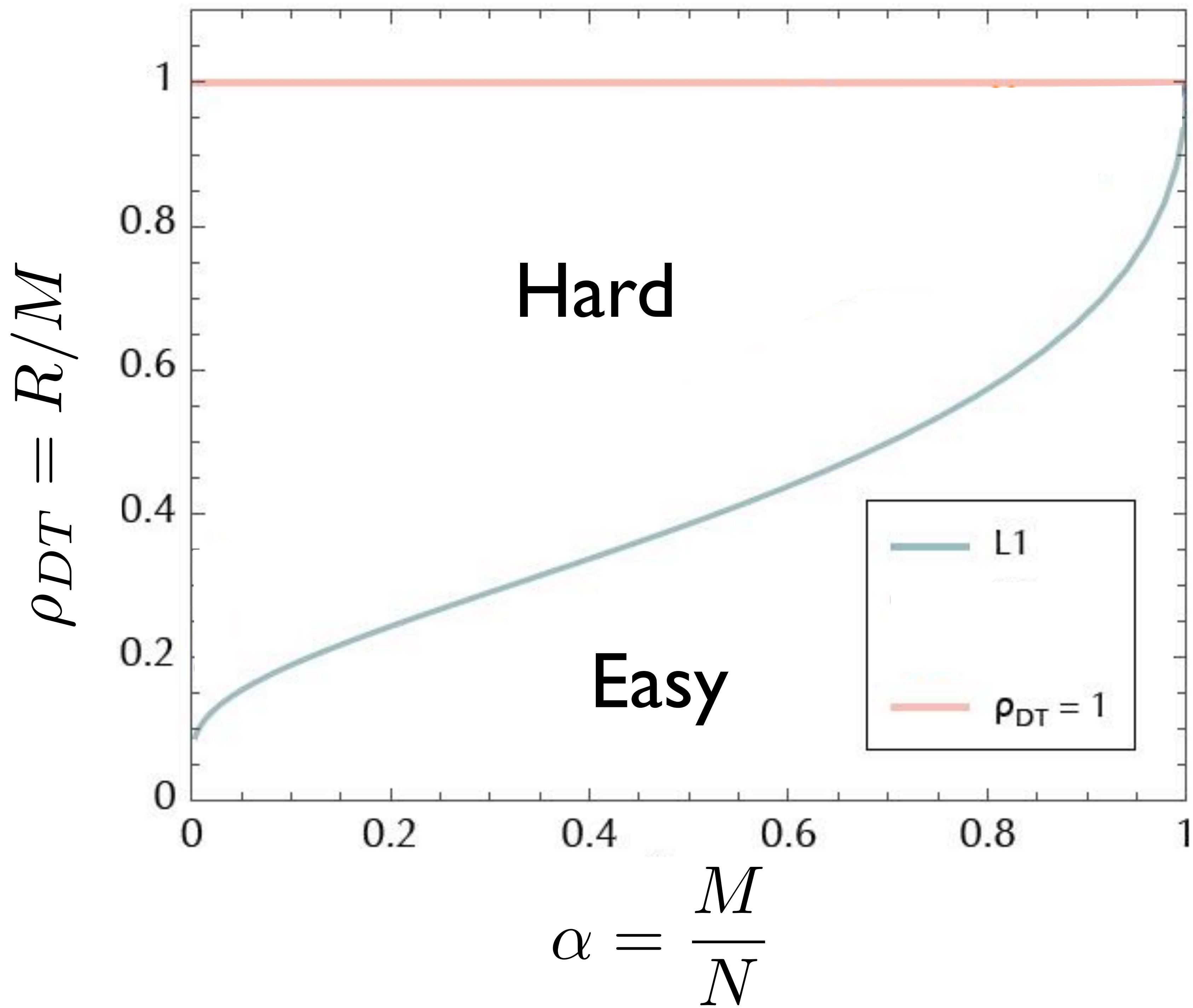


For a signal with
- (1-ρ)N zeros
- R=ρN non zeros

and a random iid matrix with $M = \alpha N$

Reconstruction limited by the Donoho-Tanner transition for the L_1 norm minimization

State of the art in CS

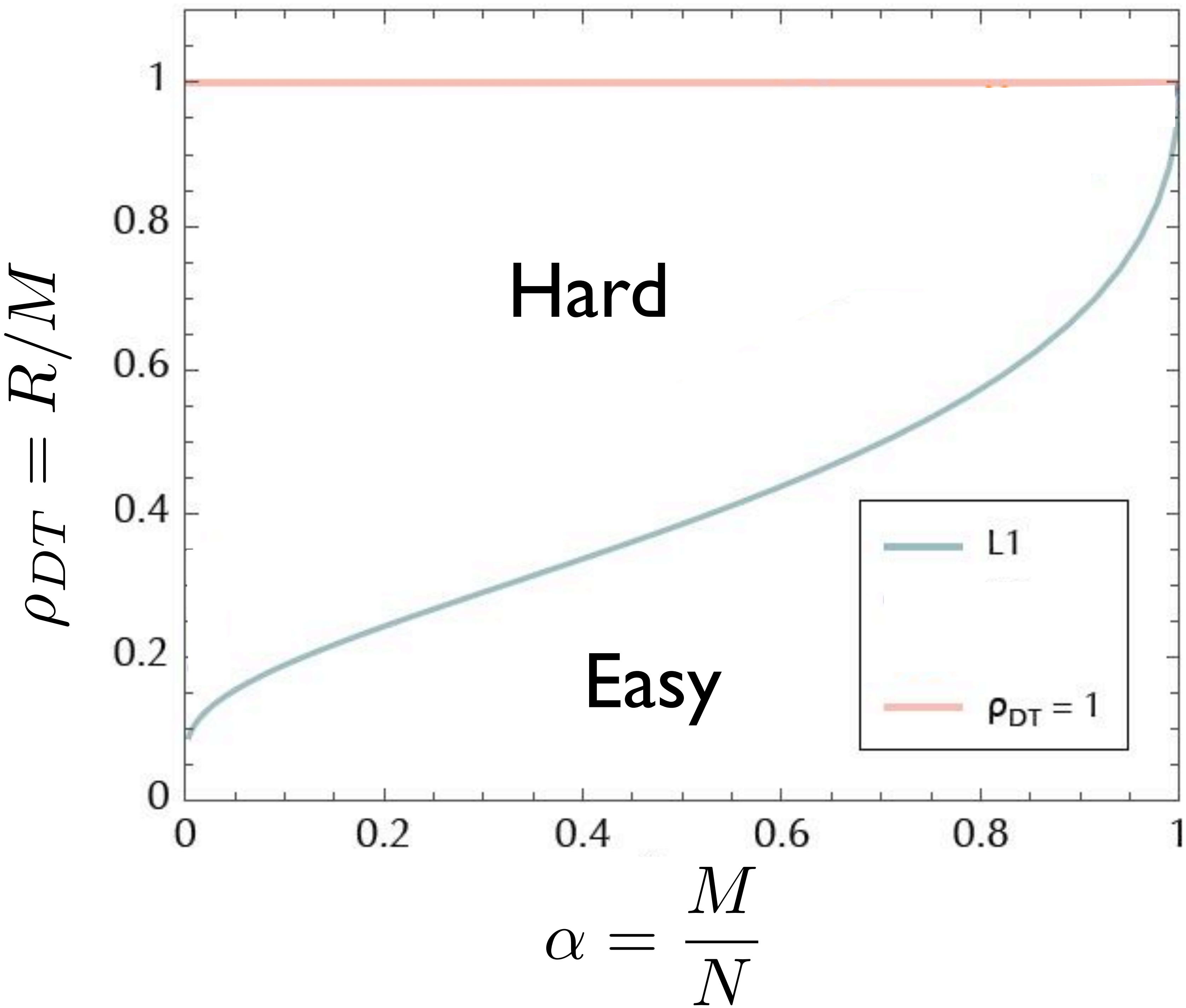


For a signal with

- (1-ρ)N zeros
- R=ρN non zeros

and a random iid matrix with $M = \alpha N$

State of the art in CS



For a signal with

- (1-ρ)N zeros
- R=ρN non zeros

and a random iid matrix with M =α N

A different representation of the same transition

Our work

A statistical physics approach to compressed sensing

- A probabilistic approach to reconstruction
- The Belief Propagation algorithm
- Seeded measurements matrices

Our work

A statistical physics approach to compressed sensing

- **A probabilistic approach to reconstruction**
- The Belief Propagation algorithm
- Seeded measurements matrices

A probabilistic approach to compressed sensing

We want to sample from this distribution:

$$P(\vec{x}|\vec{y}) = \frac{1}{Z} \prod_{i=1}^{N} \left[(1-\rho)\,\delta(x_i) + \rho\phi(x_i)\right] \prod_{\mu=1}^{M} \delta\left(y_\mu - \sum_{i=1}^{N} F_{\mu i} x_i\right)$$

A probabilistic approach to compressed sensing

We want to sample from this distribution:

$$P(\vec{x}|\vec{y}) = \frac{1}{Z} \prod_{i=1}^{N} \left[(1-\rho)\,\delta(x_i) + \rho\phi(x_i)\right] \underbrace{\prod_{\mu=1}^{M} \delta\left(y_\mu - \sum_{i=1}^{N} F_{\mu i} x_i\right)}$$

Solution of
the linear system

A probabilistic approach to compressed sensing

We want to sample from this distribution:

$$P(\vec{x}|\vec{y}) = \frac{1}{Z}\prod_{i=1}^{N}\left[(1-\rho)\,\delta(x_i)+\rho\phi(x_i)\right]\prod_{\mu=1}^{M}\delta\left(y_\mu-\sum_{i=1}^{N}F_{\mu i}x_i\right)$$

$$\underbrace{\qquad\qquad\qquad\qquad}_{\text{Sparse vector}}\qquad\underbrace{\qquad\qquad\qquad\qquad}_{\text{Solution of the linear system}}$$

Sparse vector

Solution of
the linear system

A probabilistic approach to compressed sensing

We want to sample from this distribution:

$$P(\vec{x}|\vec{y}) = \frac{1}{Z} \prod_{i=1}^{N} \left[(1-\rho)\,\delta(x_i) + \rho\phi(x_i) \right] \prod_{\mu=1}^{M} \delta\left(y_\mu - \sum_{i=1}^{N} F_{\mu i} x_i \right)$$

<u>Theorem</u>: sampling from P(x|y) gives the correct solution in the large N limit as long as $\alpha > \rho_0$ if: a) $\Phi(x) > 0\ \forall x$ and b) $1 > \rho > 0$

A probabilistic approach to compressed sensing

We want to sample from this distribution:

$$P(\vec{x}|\vec{y}) = \frac{1}{Z}\prod_{i=1}^{N}\left[(1-\rho)\,\delta(x_i) + \rho\phi(x_i)\right]\prod_{\mu=1}^{M}\delta\left(y_\mu - \sum_{i=1}^{N}F_{\mu i}x_i\right)$$

<u>Theorem</u>: sampling from P(x|y) gives the correct solution in the large N limit as long as $\alpha > \rho_0$ if: a) $\Phi(x) > 0$ $\forall x$ and b) $1 > \rho > 0$

Sampling from P(x|y) is optimal,

(even if we do not know the correct $\Phi(x)$ or the correct ρ)

A probabilistic approach to compressed sensing

We want to sample from this distribution:

$$P(\vec{x}|\vec{y}) = \frac{1}{Z}\prod_{i=1}^{N}\left[(1-\rho)\,\delta(x_i) + \rho\phi(x_i)\right]\prod_{\mu=1}^{M}\delta\left(y_\mu - \sum_{i=1}^{N}F_{\mu i}x_i\right)$$

Theorem: sampling from P(x|y) gives the correct solution in the large N limit as long as $\alpha > \rho_0$ if: a) $\Phi(x) > 0$ $\forall x$ and b) $1 > \rho > 0$

Sampling from P(x|y) is optimal,

(even if we do not know the correct $\Phi(x)$ or the correct ρ)

In practice, we use a Gaussian distribution for $\Phi(x)$, with mean m and variance σ^2, and "learn" the best value for ρ, σ and m.

A sketch of the proof

Consider the system constrained to be at distances larger than D with respect to the solution

$$Y(D,\epsilon) = \int \prod_{i=1}^{N} (dx_i \; [(1-\rho)\delta(x_i) + \rho\phi(x_i)]) \prod_{\mu=1}^{M} \delta_\epsilon \left(\sum_i F_{\mu i}(x_i - s_i) \right) \mathbb{I} \left(\sum_{i=1}^{N} (x_i - s_i)^2 > ND \right)$$

1) Y(0) is infinite if $\alpha > \rho_0$ (equivalently if M>R)

(just count the delta functions! N-R+M deltas versus N integrals...)

2) Y(D) is finite for any D>0

(bound by a first moment method, or "annealed" computation)

A sketch of the proof

Consider the system constrained to be at
distances larger than D with respect to the solution

$$Y(D,\epsilon) = \int \prod_{i=1}^{N} \left(dx_i \; [(1-\rho)\delta(x_i) + \rho\phi(x_i)]\right) \prod_{\mu=1}^{M} \delta_\epsilon\left(\sum_i F_{\mu i}(x_i - s_i)\right) \mathbb{I}\left(\sum_{i=1}^{N}(x_i - s_i)^2 > ND\right)$$

1) Y(0) is infinite if $\alpha>\rho_0$ (equivalently if M>R)

(just count the delta functions! N-R+M deltas versus N integrals...)

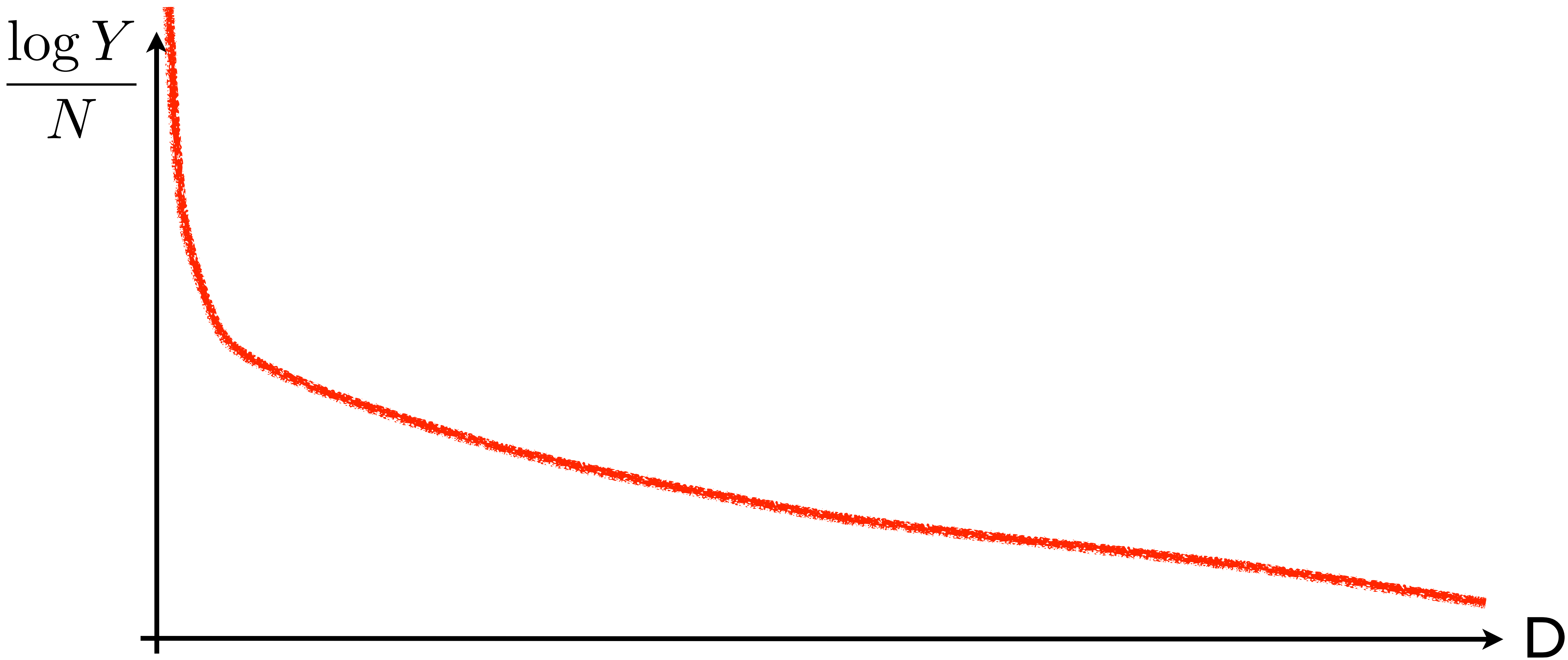
2) Y(D) is finite for any D>0

(bound by a first moment method, or "annealed" computation)

If $\alpha>\rho_0$, the measure is always dominated by the solution

A sketch of the proof

Consider the system constrained to be at distances larger than D with respect to the solution



$$Y(D,\epsilon) = \int \prod_{i=1}^{N} \left(dx_i \;\left[(1-\rho)\delta(x_i) + \rho\phi(x_i)\right]\right) \prod_{\mu=1}^{M} \delta_\epsilon \left(\sum_i F_{\mu i}(x_i - s_i) \right) \mathbb{I}\left(\sum_{i=1}^{N} (x_i - s_i)^2 > ND \right)$$

A probabilistic approach to compressed sensing

Probabilistic reconstruction using:

$$P(\vec{x}|\vec{y}) = \frac{1}{Z} \prod_{i=1}^{N} \left[(1-\rho)\,\delta(x_i) + \rho\phi(x_i)\right] \prod_{\mu=1}^{M} \delta\left(y_\mu - \sum_{i=1}^{N} F_{\mu i} x_i\right)$$

Sampling from P(x|y) is optimal,

(even if we do not know the correct Φ(x) or the correct ρ)

Statistical physics and information theory tools can be readily used for

- Sampling
- Computing phase diagram
- etc etc...

The link with statistical physics and spin glasses

$$P(\vec{x}|\vec{y}) = \frac{1}{Z}\prod_{i=1}^{N} P(x_i) \prod_{\mu=1}^{M} \delta\left(y_\mu - \sum_{i=1}^{N} F_{\mu i} x_i\right)$$ with $$P(x_i) = (1-\rho)\delta(x_i) + \rho\phi(x_i)$$

$$P(\vec{x}|\vec{y}) = \frac{1}{Z} e^{-\sum_{i=1}^{N} \log P(x_i) - \frac{1}{2\Delta}\sum_{\mu=1}^{M}(y_\mu - \sum_{i=1}^{N} F_{\mu i} x_i)^2}$$

In physics, this is called a spin glass problem
This is studied since the early 80's

The link with statistical physics and spin glasses

$$P(\vec{x}|\vec{y}) = \frac{1}{Z}\prod_{i=1}^{N} P(x_i)\prod_{\mu=1}^{M}\delta\left(y_\mu - \sum_{i=1}^{N} F_{\mu i} x_i\right)$$ **with** $$P(x_i) = (1-\rho)\delta(x_i) + \rho\phi(x_i)$$

Hamiltonian

$$P(\vec{x}|\vec{y}) = \frac{1}{Z} e^{-\sum_{i=1}^{N}\log P(x_i) - \frac{1}{2\Delta}\sum_{\mu=1}^{M}(y_\mu - \sum_{i=1}^{N} F_{\mu i} x_i)^2}$$

In physics, this is called a spin glass problem
This is studied since the early 80's

The link with statistical physics and spin glasses

$$P(\vec{x}|\vec{y}) = \frac{1}{Z}\prod_{i=1}^{N} P(x_i)\prod_{\mu=1}^{M}\delta\left(y_\mu - \sum_{i=1}^{N} F_{\mu i}x_i\right)$$ **with** $$P(x_i) = (1-\rho)\delta(x_i) + \rho\phi(x_i)$$

Hamiltonian

Partition sum

$$P(\vec{x}|\vec{y}) = \frac{1}{Z} e^{-\sum_{i=1}^{N}\log P(x_i) - \frac{1}{2\Delta}\sum_{\mu=1}^{M}(y_\mu - \sum_{i=1}^{N} F_{\mu i}x_i)^2}$$

In physics, this is called a spin glass problem
This is studied since the early 80's

The link with statistical physics and spin glasses

$$P(\vec{x}|\vec{y}) = \frac{1}{Z}\prod_{i=1}^{N} P(x_i) \prod_{\mu=1}^{M} \delta\left(y_\mu - \sum_{i=1}^{N} F_{\mu i} x_i\right)$$ with $$P(x_i) = (1-\rho)\delta(x_i) + \rho\phi(x_i)$$

Disordered interaction

$$P(\vec{x}|\vec{y}) = \frac{1}{Z} e^{-\sum_{i=1}^{N} \log P(x_i) - \frac{1}{2\Delta}\sum_{\mu=1}^{M}(y_\mu - \sum_{i=1}^{N} F_{\mu i} x_i)^2}$$

In physics, this is called a spin glass problem
This is studied since the early 80's

The link with statistical physics and spin glasses

$$P(\vec{x}|\vec{y}) = \frac{1}{Z}\prod_{i=1}^{N} P(x_i) \prod_{\mu=1}^{M} \delta\left(y_\mu - \sum_{i=1}^{N} F_{\mu i} x_i\right)$$ with $P(x_i) = (1-\rho)\delta(x_i) + \rho\phi(x_i)$

Mean-field
long-range interactions

$$P(\vec{x}|\vec{y}) = \frac{1}{Z} e^{-\sum_{i=1}^{N} \log P(x_i) - \frac{1}{2\Delta}\sum_{\mu=1}^{M} (y_\mu - \sum_{i=1}^{N} F_{\mu i} x_i)^2}$$

In physics, this is called a spin glass problem
This is studied since the early 80's

Our work

A statistical physics approach to compressed sensing

- A probabilistic approach to reconstruction
- **The Belief Propagation algorithm**
- Seeded measurements matrices

Our work

A statistical physics approach to compressed sensing

- A probabilistic approach to reconstruction
- **The Belief Propagation algorithm**
- Seeded measurements matrices

Statistical Physics approach (FK et al.)
+rigorous (Donoho, Montanari et al.)

How to sample?

$$P(\vec{x}|\vec{y}) = \frac{1}{Z} \prod_{i=1}^{N} \left[(1-\rho)\,\delta(x_i) + \rho\phi(x_i)\right] \prod_{\mu=1}^{M} \delta\left(y_\mu - \sum_{i=1}^{N} F_{\mu i} x_i\right)$$

How to sample?

$$P(\vec{x}|\vec{y}) = \frac{1}{Z}\prod_{i=1}^{N}\left[(1-\rho)\,\delta(x_i)+\rho\phi(x_i)\right]\prod_{\mu=1}^{M}\delta\left(y_\mu-\sum_{i=1}^{N}F_{\mu i}x_i\right)$$

Solution number 1:using Markov-Chain Monte-Carlo

How to sample?

$$P(\vec{x}|\vec{y}) = \frac{1}{Z} \prod_{i=1}^{N} \left[(1-\rho)\,\delta(x_i) + \rho\phi(x_i)\right] \prod_{\mu=1}^{M} \delta\left(y_\mu - \sum_{i=1}^{N} F_{\mu i} x_i\right)$$

Solution number 1:using Markov-Chain Monte-Carlo

Used in the litterature for ultrasound imaging (cf : Quinsac et al, 2011...)

How to sample?

$$P(\vec{x}|\vec{y}) = \frac{1}{Z} \prod_{i=1}^{N} \left[(1-\rho)\,\delta(x_i) + \rho\phi(x_i) \right] \prod_{\mu=1}^{M} \delta\left(y_\mu - \sum_{i=1}^{N} F_{\mu i} x_i \right)$$

Solution number 1:using Markov-Chain Monte-Carlo

Very long!

Used in the litterature for ultrasound imaging
(cf : Quinsac et al, 2011...)

How to sample?

$$P(\vec{x}|\vec{y}) = \frac{1}{Z} \prod_{i=1}^{N} \left[(1-\rho)\,\delta(x_i) + \rho\phi(x_i) \right] \prod_{\mu=1}^{M} \delta\left(y_\mu - \sum_{i=1}^{N} F_{\mu i} x_i \right)$$

Solution number 1:using Markov-Chain Monte-Carlo

Very long!

Used in the litterature for ultrasound imaging
(cf : Quinsac et al, 2011...)

How to sample?

$$P(\vec{x}|\vec{y}) = \frac{1}{Z} \prod_{i=1}^{N} \left[(1-\rho)\,\delta(x_i) + \rho\phi(x_i) \right] \prod_{\mu=1}^{M} \delta\left(y_\mu - \sum_{i=1}^{N} F_{\mu i} x_i \right)$$

Solution number 2: estimate the marginal probabilities with a message passing algorithm

If we do it correctly, then the solution is given by $a_i = \int dx_i P_i(x_i) x_i$

How to sample?

$$P(\vec{x}|\vec{y}) = \frac{1}{Z}\prod_{i=1}^{N}\left[(1-\rho)\,\delta(x_i)+\rho\phi(x_i)\right]\prod_{\mu=1}^{M}\delta\left(y_\mu - \sum_{i=1}^{N} F_{\mu i}x_i\right)$$

Solution number 2: estimate the marginal probabilities with a message passing algorithm

If we do it correctly, then the solution is given by $a_i = \int dx_i P_i(x_i) x_i$

In this model, this can be done <u>exactly</u> (for large N,M) using an approach known as:

1. Thouless-Anderson-Parlmer, or Cavity method in physics
   Bethe-Peierls, Onsager ('35) Parisi and Mezard ('02)
2. Belief propagation in artificial intelligence (Pearl, '82)
3. Sum-product in coding theory (Gallager, '60)
3. Approximate Message Passing in compressed sensing
   Rangan, Montanari...

How does BP works?

Gibbs free energy approach: $\log Z = \max_{\{\mathcal{P}(\vec{x})\}} f_{Gibbs}\left(\{\mathcal{P}(\vec{x})\}\right)$

With $f_{Gibbs}\left(\{\mathcal{P}(\vec{x})\}\right) = -\langle \log P(\vec{x}|\vec{y}) \rangle_{\mathcal{P}(\vec{x})} - \int d\vec{x}\mathcal{P}(\vec{x}) \log \mathcal{P}(\vec{x})$

How does BP works?

Gibbs free energy approach: $\log Z = \max_{\{\mathcal{P}(\vec{x})\}} f_{Gibbs}\left(\{\mathcal{P}(\vec{x})\}\right)$

With $f_{Gibbs}\left(\{\mathcal{P}(\vec{x})\}\right) = -\langle \log P(\vec{x}|\vec{y})\rangle_{\mathcal{P}(\vec{x})} - \int d\vec{x}\mathcal{P}(\vec{x}) \log \mathcal{P}(\vec{x})$

Mean-Field $\Rightarrow$ $\mathcal{P}(\vec{x}) = \prod_i \mathcal{P}_i(\vec{x}_i)$

How does BP works?

Gibbs free energy approach: $$\log Z = \max_{\{\mathcal{P}(\vec{x})\}} f_{Gibbs}\left(\{\mathcal{P}(\vec{x})\}\right)$$

With $$f_{Gibbs}\left(\{\mathcal{P}(\vec{x})\}\right) = -\langle \log P(\vec{x}|\vec{y})\rangle_{\mathcal{P}(\vec{x})} - \int d\vec{x}\mathcal{P}(\vec{x}) \log \mathcal{P}(\vec{x})$$

Mean-Field $\Rightarrow$ $$\mathcal{P}(\vec{x}) = \prod_i \mathcal{P}_i(\vec{x}_i)$$

Not correct
+Convergence problems

How does BP works?

Gibbs free energy approach: $\log Z = \max_{\{\mathcal{P}(\vec{x})\}} f_{Gibbs}\left(\{\mathcal{P}(\vec{x})\}\right)$

With $f_{Gibbs}\left(\{\mathcal{P}(\vec{x})\}\right) = -\langle \log P(\vec{x}|\vec{y})\rangle_{\mathcal{P}(\vec{x})} - \int d\vec{x}\mathcal{P}(\vec{x}) \log \mathcal{P}(\vec{x})$

Mean-Field $\Rightarrow$ $\mathcal{P}(\vec{x}) = \prod_i \mathcal{P}_i(\vec{x}_i)$

Not correct
+Convergence problems

Belief-Propagation$\Rightarrow$ $\mathcal{P}(\vec{x}) = \frac{\prod_{ij} \mathcal{P}_{ij}(\vec{x}_i, \vec{x}_j)}{\prod_i \mathcal{P}_i(\vec{x}_i)^{M-1}}$

How does BP works?

Gibbs free energy approach: $\log Z = \max_{\{\mathcal{P}(\vec{x})\}} f_{Gibbs}\left(\{\mathcal{P}(\vec{x})\}\right)$

With $f_{Gibbs}\left(\{\mathcal{P}(\vec{x})\}\right) = -\langle \log P(\vec{x}|\vec{y})\rangle_{\mathcal{P}(\vec{x})} - \int d\vec{x}\mathcal{P}(\vec{x}) \log \mathcal{P}(\vec{x})$

Mean-Field $\Rightarrow$ $\mathcal{P}(\vec{x}) = \prod_i \mathcal{P}_i(\vec{x}_i)$

Not correct
+Convergence problems

Belief-Propagation$\Rightarrow$ $\mathcal{P}(\vec{x}) = \frac{\prod_{ij} \mathcal{P}_{ij}(\vec{x}_i, \vec{x}_j)}{\prod_i \mathcal{P}_i(\vec{x}_i)^{M-1}}$

(asymptotically)
exact in CS
with random matrices

How does BP works?

Gibbs free energy approach: $$\log Z = \max_{\{\mathcal{P}(\vec{x})\}} f_{Gibbs}\left(\{\mathcal{P}(\vec{x})\}\right)$$

With $$f_{Gibbs}\left(\{\mathcal{P}(\vec{x})\}\right) = -\langle \log P(\vec{x}|\vec{y})\rangle_{\mathcal{P}(\vec{x})} - \int d\vec{x}\mathcal{P}(\vec{x}) \log \mathcal{P}(\vec{x})$$

Mean-Field $\Rightarrow$ $$\mathcal{P}(\vec{x}) = \prod_i \mathcal{P}_i(\vec{x}_i)$$ Not correct +Convergence problems

Belief-Propagation$\Rightarrow$ $$\mathcal{P}(\vec{x}) = \frac{\prod_{ij} \mathcal{P}_{ij}(\vec{x}_i, \vec{x}_j)}{\prod_i \mathcal{P}_i(\vec{x}_i)^{M-1}}$$ (asymptotically) exact in CS with random matrices

How does BP works?

Simplification thanks to the large connectivity limit:
Projection on first two moments is enough :

$$f\left(\{\mathcal{P}_i(x_i), \mathcal{P}_{ij}(x_i, x_j)\}\right) \Rightarrow f\left(\{\langle x_i\rangle, \langle x_i^2\rangle\}\right)$$

Belief-Propagation equations $\Rightarrow$

$$\begin{cases} \langle x_i\rangle^{t+1} = \langle x_i\rangle^t + \dfrac{\partial f}{\partial\langle x_i\rangle} \\ \langle x_i^2\rangle^{t+1} = \langle x_i^2\rangle^t + \dfrac{\partial f}{\partial\langle x_i^2\rangle} \end{cases}$$

The Belief-Propagation algorithm

Iterate these variables

$$U_i^{(t+1)} = \frac{\alpha}{M} \sum_\mu \frac{1}{\Delta_\mu + \gamma^{(t)}}$$

$$V_i^{(t+1)} = \sum_\mu F_{\mu i} \frac{(y_\mu - \alpha_\mu^{(t)})}{\Delta_\mu + \gamma_\mu^{(t)}} + f_a\left(U_i^{(t)}, V_i^{(t)}\right) \frac{\alpha}{M} \sum_\mu \frac{1}{\Delta_\mu + \gamma^{(t)}}$$

$$\alpha_\mu^{(t+1)} = \sum_i F_{\mu i} f_a(U_i^{(t+1)}, V_i^{(t+1)}) - \frac{(y_\mu - \alpha_\mu^{(t)})}{\Delta_\mu + \gamma^{(t)}} \frac{1}{N} \sum_i \frac{\partial f_a}{\partial Y}\left(U_i^{(t+1)}, V_i^{(t+1)}\right)$$

$$\gamma^{(t+1)} = \frac{1}{N} \sum_i f_c(U_i^{(t+1)}, V_i^{(t+1)})$$

Using these functions:

$$f_a(X,Y) = \left[\frac{\rho Y}{(1+X)^{3/2}} e^{Y^2/(2(1+X))}\right] \left[1 - \rho + \frac{\rho}{(1+X)^{1/2}} e^{Y^2/(2(1+X))}\right]^{-1}$$

$$f_c(X,Y) = \left[\frac{\rho}{(1+X)^{3/2}} e^{Y^2/(2(1+X))} \left(1 + \frac{Y^2}{1+X}\right)\right] \left[1 - \rho + \frac{\rho}{(1+X)^{1/2}} e^{Y^2/(2(1+X))}\right]^{-1} - f_a(X,Y)^2$$

And finally at the end:

$$\langle x_i \rangle = f_a\left(U_i, V_i\right)$$

The Belief-Propagation algorithm

Iterate these variables

$$U_i^{(t+1)} = \frac{\alpha}{M}\sum_\mu \frac{1}{\Delta_\mu + \gamma^{(t)}}$$

$$V_i^{(t+1)} = \sum_\mu F_{\mu i}\frac{(y_\mu - \alpha_\mu^{(t)})}{\Delta_\mu + \gamma_\mu^{(t)}} + f_a\left(U_i^{(t)}, V_i^{(t)}\right)\frac{\alpha}{M}\sum_\mu \frac{1}{\Delta_\mu + \gamma^{(t)}}$$

$$\alpha_\mu^{(t+1)} = \sum_i F_{\mu i} f_a(U_i^{(t+1)}, V_i^{(t+1)}) - \frac{(y_\mu - \alpha_\mu^{(t)})}{\Delta_\mu + \gamma^{(t)}}\frac{1}{N}\sum_i \frac{\partial f_a}{\partial Y}\left(U_i^{(t+1)}, V_i^{(t+1)}\right)$$

$$\gamma^{(t+1)} = \frac{1}{N}\sum_i f_c(U_i^{(t+1)}, V_i^{(t+1)})$$

Simple algebraic operations!

Using these functions:

$$f_a(X,Y) = \left[\frac{\rho Y}{(1+X)^{3/2}} e^{Y^2/(2(1+X))}\right]\left[1 - \rho + \frac{\rho}{(1+X)^{1/2}} e^{Y^2/(2(1+X))}\right]^{-1}$$

$$f_c(X,Y) = \left[\frac{\rho}{(1+X)^{3/2}} e^{Y^2/(2(1+X))}\left(1 + \frac{Y^2}{1+X}\right)\right]\left[1 - \rho + \frac{\rho}{(1+X)^{1/2}} e^{Y^2/(2(1+X))}\right]^{-1} - f_a(X,Y)^2$$

And finally at the end:

$$\langle x_i \rangle = f_a\left(U_i, V_i\right)$$

The Belief-Propagation algorithm

Iterate these variables

$$U_i^{(t+1)} = \frac{\alpha}{M}\sum_{\mu}\frac{1}{\Delta_\mu + \gamma^{(t)}}$$

$$V_i^{(t+1)} = \sum_{\mu} F_{\mu i}\frac{(y_\mu - \alpha_\mu^{(t)})}{\Delta_\mu + \gamma_\mu^{(t)}} + f_a\left(U_i^{(t)}, V_i^{(t)}\right)\frac{\alpha}{M}\sum_{\mu}\frac{1}{\Delta_\mu + \gamma^{(t)}}$$

$$\alpha_\mu^{(t+1)} = \sum_i F_{\mu i} f_a(U_i^{(t+1)}, V_i^{(t+1)}) - \frac{(y_\mu - \alpha_\mu^{(t)})}{\Delta_\mu + \gamma^{(t)}}\frac{1}{N}\sum_i \frac{\partial f_a}{\partial Y}\left(U_i^{(t+1)}, V_i^{(t+1)}\right)$$

$$\gamma^{(t+1)} = \frac{1}{N}\sum_i f_c(U_i^{(t+1)}, V_i^{(t+1)})$$

Simple algebraic operations!

Using these functions:

$$f_a(X,Y) = \left[\frac{\rho Y}{(1+X)^{3/2}} e^{Y^2/(2(1+X))}\right]\left[1 - \rho + \frac{\rho}{(1+X)^{1/2}} e^{Y^2/(2(1+X))}\right]^{-1}$$

$$f_c(X,Y) = \left[\frac{\rho}{(1+X)^{3/2}} e^{Y^2/(2(1+X))}\left(1 + \frac{Y^2}{1+X}\right)\right]\left[1 - \rho + \frac{\rho}{(1+X)^{1/2}} e^{Y^2/(2(1+X))}\right]^{-1} - f_a(X,Y)^2$$

And finally at the end:

$$\langle x_i \rangle = f_a\left(U_i, V_i\right)$$

Complexity is O(N^2×convergence time)

The Belief-Propagation algorithm: How to learn the parameter in the Prior?

Three parameters $\rho, \overline{x}, \sigma$ $\phi(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\overline{x})^2/(2\sigma^2)}$

Compute the Bethe free-entropy $F = \log Z$ using the BP messages.

Compute the gradient $\left(\frac{\partial F}{\partial \rho}, \frac{\partial F}{\partial \overline{x}}, \frac{\partial F}{\partial \sigma} \right)$

and update parameters to maximize F at each step of the BP iteration

Learning makes the algorithm faster
(equivalent to Expectation-Maximization)

Analysis of the algorithm

The performance of the algorithm for a given distribution of signals can be analyzed using a method knows as density evolution (coding theory) or replica method (physics)

$$Z(y) = \int \prod_{i=1}^{N} dx_i P(x|y) \qquad F(\vec{y}) = -\log Z(\vec{y})$$

Analysis of the algorithm

The performance of the algorithm for a given distribution of signals can be analyzed using a method knows as density evolution (coding theory) or replica method (physics)

$$Z(y) = \int \prod_{i=1}^{N} dx_i P(x|y) \qquad F(\vec{y}) = -\log Z(\vec{y})$$

Averaging over a signal distributiion (ex: Gauss Bernoulli)

$F_{\mu i}$ iid Gaussian, variance $1/N$

$y_\mu = \sum_{i=1}^{N} F_{\mu i} x_i^0$ where x_i^0 are iid distributed from $(1-\rho_0)\delta(x_i^0) + \rho_0 \phi_0(x_i)$

Analysis of the algorithm

The performance of the algorithm for a <u>given distribution of signals</u> can be analyzed using a method knows as density evolution (coding theory) or replica method (physics)

$$Z(y) = \int \prod_{i=1}^{N} dx_i P(x|y) \qquad F(\vec{y}) = -\log Z(\vec{y})$$

<u>Averaging over a signal distributiion (ex: Gauss Bernoulli)</u>

$F_{\mu i}$ iid Gaussian, variance $1/N$

$y_\mu = \sum_{i=1}^{N} F_{\mu i} x_i^0$ where x_i^0 are iid distributed from $(1-\rho_0)\delta(x_i^0) + \rho_0 \phi_0(x_i)$

Replica method ⇒ $\overline{\log Z} = \lim_{n \to 0} \frac{\overline{Z^n} - 1}{n}$

Analysis of the algorithm

Averaging over a signal distributiion (ex: Gauss Bernoulli)

$F_{\mu i}$ iid Gaussian, variance $1/N$

$y_\mu = \sum_{i=1}^{N} F_{\mu i} x_i^0$ where x_i^0 are iid distributed from $(1-\rho_0)\delta(x_i^0) + \rho_0 \phi_0(x_i)$

$$\Phi(Q,q,m,\hat{Q},\hat{q},\hat{m}) = -\frac{1}{2N}\sum_\mu \frac{q-2m+\rho+\Delta_\mu}{\Delta_\mu+Q-q} - \frac{1}{2N}\sum_\mu \log\left(\Delta_\mu+Q-q\right) + \frac{Q\hat{Q}}{2} - m\hat{m} + \frac{q\hat{q}}{2}$$

$$+\int \mathcal{D}z \int \mathrm{d}x_0 \left[(1-\rho_0)\delta(x_0)+\rho_0\phi_0(x_0)\right] \log\left\{\int \mathrm{d}x\, e^{-\frac{\hat{Q}+\hat{q}}{2}x^2 + \hat{m}x x_0 + z\sqrt{\hat{q}}x} \left[(1-\rho)\delta(x)+\rho\phi(x)\right]\right\}$$

Order parameters:

$$Q = \frac{1}{N}\sum_i \langle x_i^2\rangle \qquad q = \frac{1}{N}\sum_i \langle x_i\rangle^2 \qquad m = \frac{1}{N}\sum_i x_i^0\langle x_i\rangle$$

Mean square error:

$$E = \frac{1}{N}\sum_i \left(\langle x_i\rangle - x_i^0\right)^2 = q - 2m + \langle (x_i^0)^2\rangle_0$$

Analysis of the algorithm

Averaging over a signal distributiion (ex: Gauss Bernoulli)

$F_{\mu i}$ iid Gaussian, variance $1/N$

$y_\mu = \sum_{i=1}^{N} F_{\mu i} x_i^0$ where x_i^0 are iid distributed from $(1-\rho_0)\delta(x_i^0) + \rho_0 \phi_0(x_i)$

$$\Phi(Q,q,m,\hat{Q},\hat{q},\hat{m}) = -\frac{1}{2N}\sum_\mu \frac{q-2m+\rho+\Delta_\mu}{\Delta_\mu+Q-q} - \frac{1}{2N}\sum_\mu \log\left(\Delta_\mu+Q-q\right) + \frac{Q\hat{Q}}{2} - m\hat{m} + \frac{q\hat{q}}{2}$$

$$+\int \mathcal{D}z \int \mathrm{d}x_0 \left[(1-\rho_0)\delta(x_0)+\rho_0\phi_0(x_0)\right] \log\left\{\int \mathrm{d}x\, e^{-\frac{\hat{Q}+\hat{q}}{2}x^2+\hat{m}xx_0+z\sqrt{\hat{q}}x}\left[(1-\rho)\delta(x)+\rho\phi(x)\right]\right\}$$

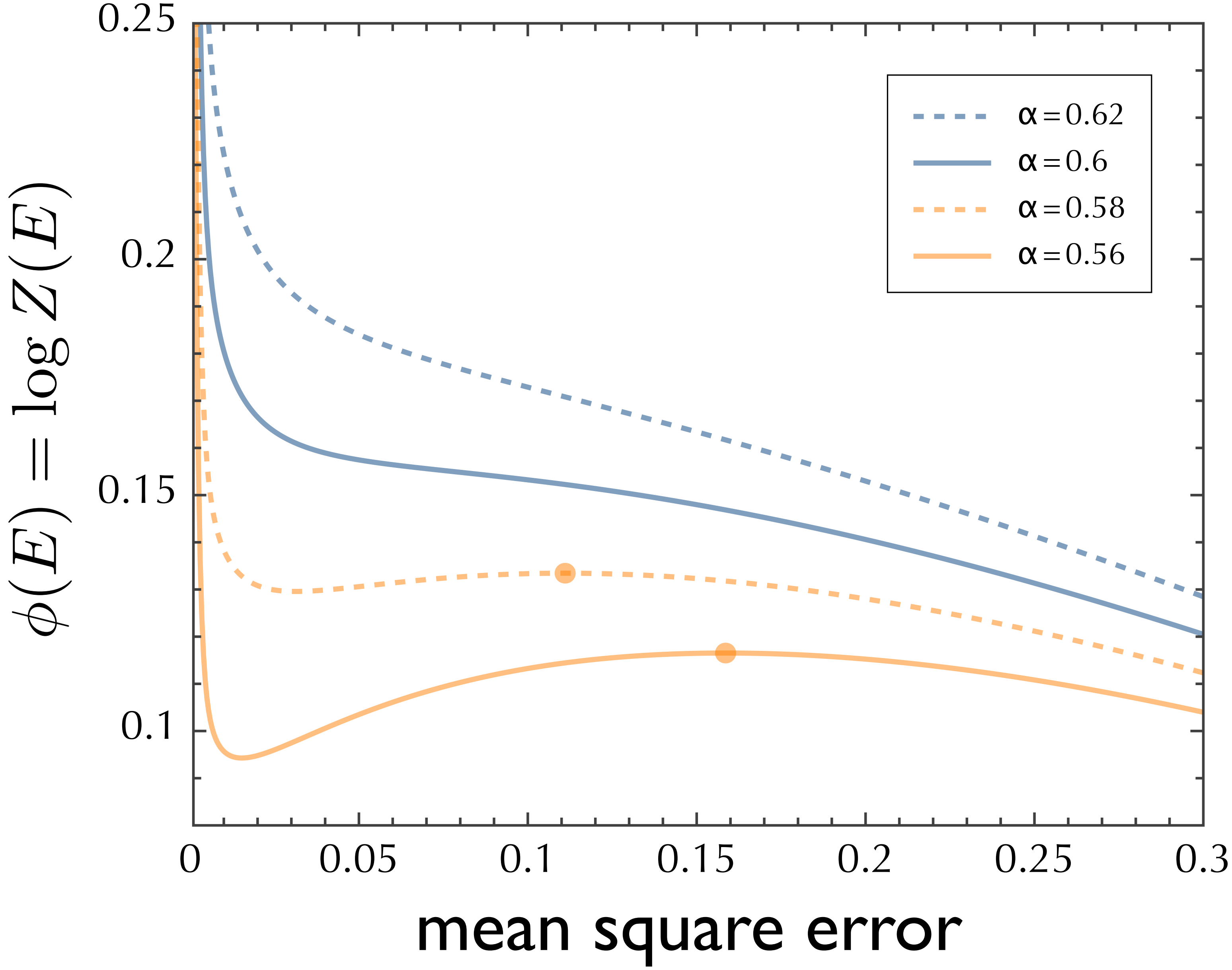
Order parameters:

$$Q = \frac{1}{N}\sum_i \langle x_i^2\rangle \qquad q = \frac{1}{N}\sum_i \langle x_i\rangle^2 \qquad m = \frac{1}{N}\sum_i x_i^0\langle x_i\rangle$$

Mean square error: $$E = \frac{1}{N}\sum_i \left(\langle x_i\rangle - x_i^0\right)^2 = q - 2m + \langle (x_i^0)^2\rangle_0$$

Computing the free entropy

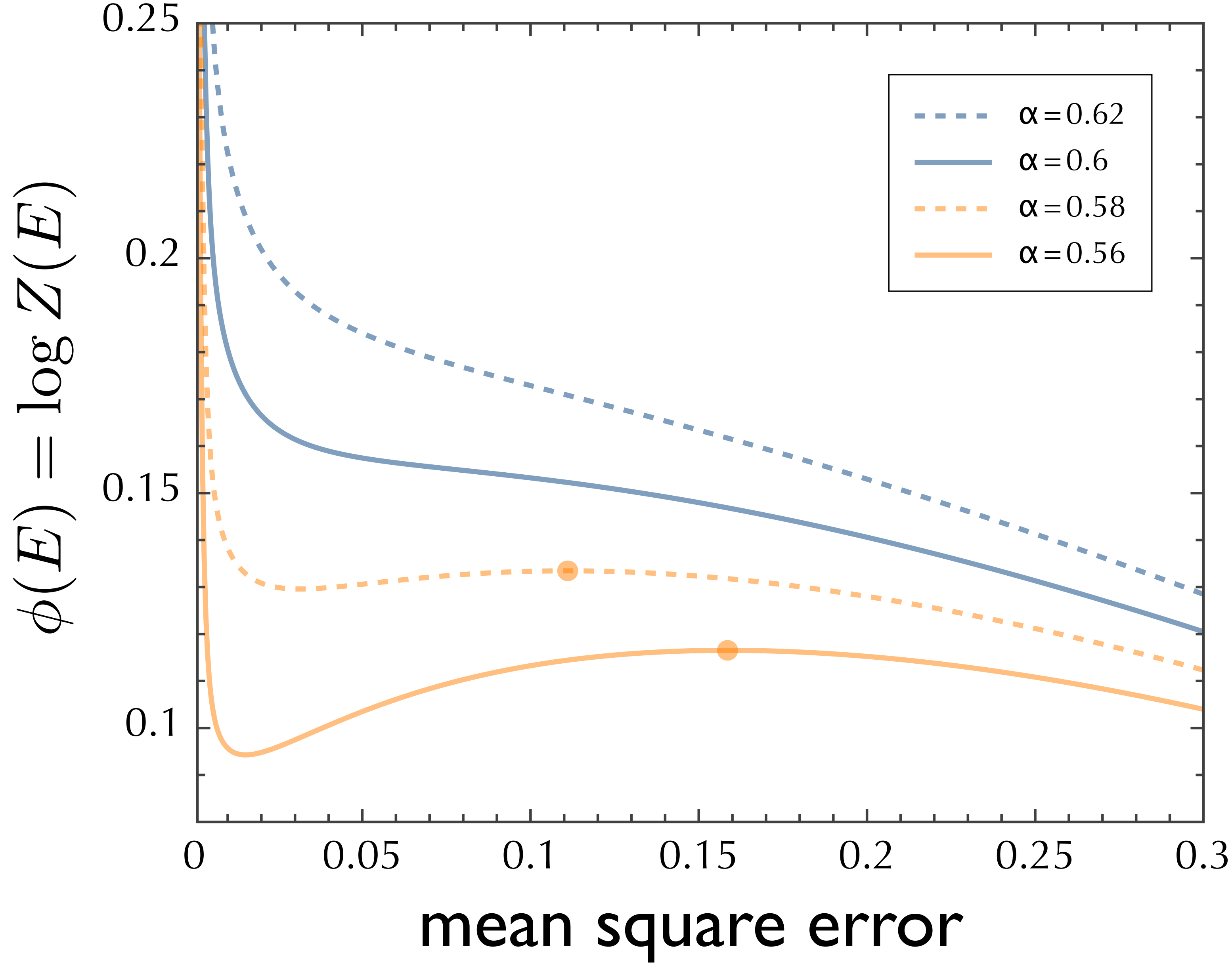
Example with ρ_0=0.4, and Φ_0 a Gaussian distribution with zero mean and unit variance



$$E = \frac{1}{N} \sum_i \left(\langle x_i \rangle - x_i^0 \right)^2$$

Computing the free entropy

Example with ρ_0=0.4, and Φ_0 a Gaussian distribution with zero mean and unit variance

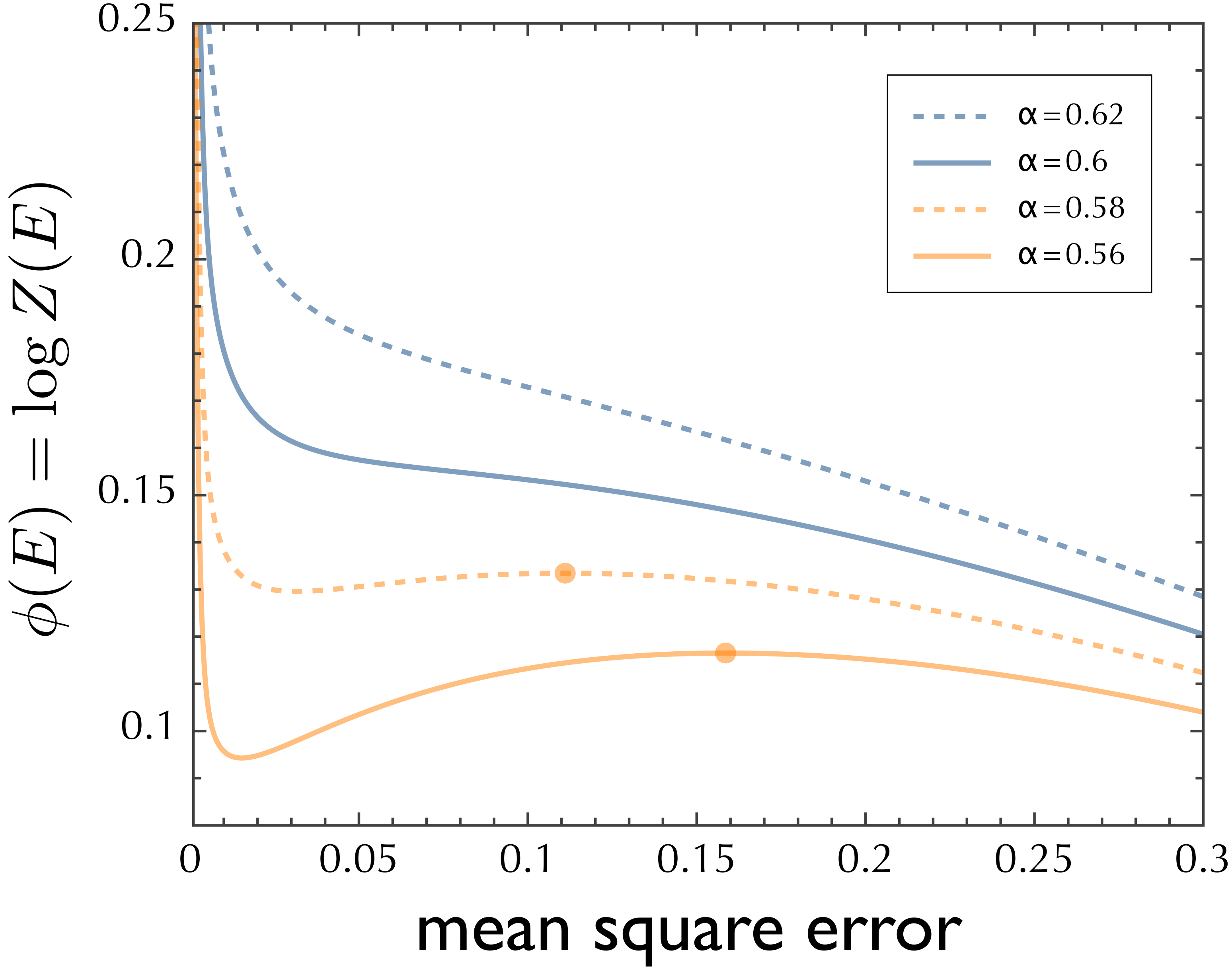


$$E = \frac{1}{N} \sum_i \left(\langle x_i \rangle - x_i^0 \right)^2$$

- Maximum is at E=0 (as long as $\alpha > \rho_0$): Equilibrium behavior dominated by the original signal

Computing the free entropy

Example with ρ_0=0.4, and Φ_0 a Gaussian distribution with zero mean and unit variance

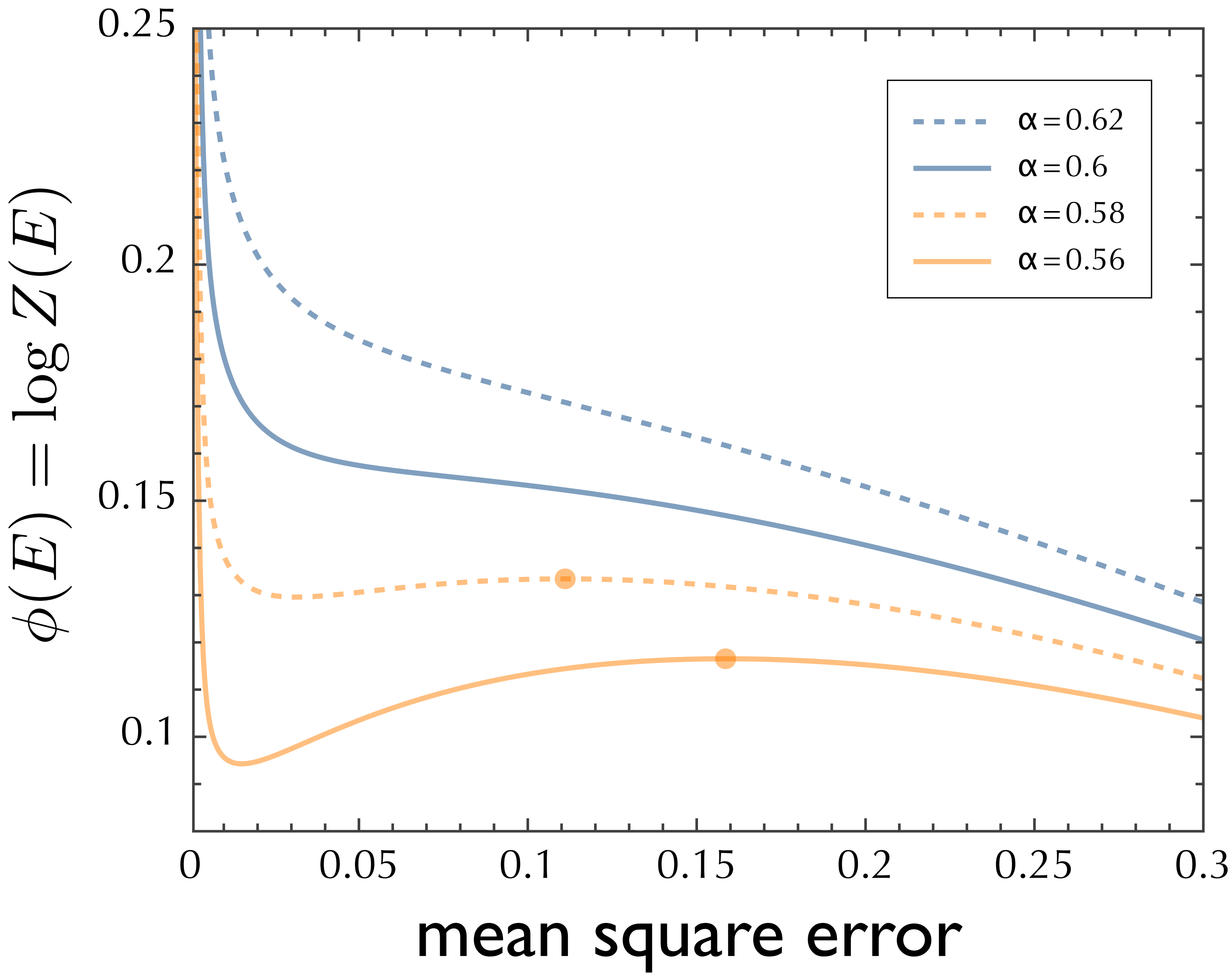


$$E = \frac{1}{N}\sum_i \left(\langle x_i \rangle - x_i^0\right)^2$$

- Maximum is at E=0 (as long as $\alpha>\rho 0$): Equilibrium behavior dominated by the original signal
- For $\alpha<0.58$, a secondary maximum appears (meta-stable state): spinodal point

Computing the free entropy

Example with ρ_0=0.4, and Φ_0 a Gaussian distribution with zero mean and unit variance

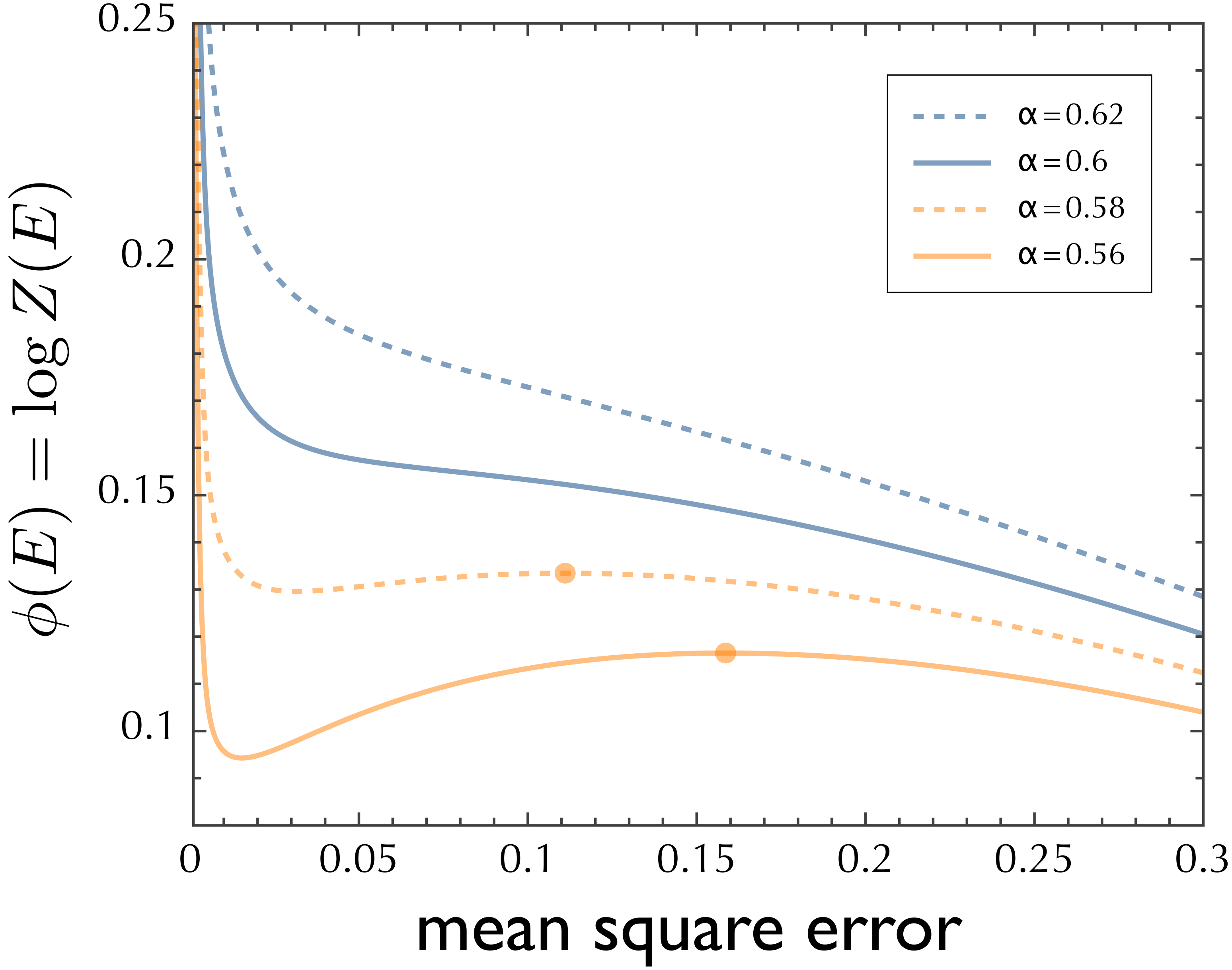


$$E = \frac{1}{N} \sum_i \left(\langle x_i \rangle - x_i^0 \right)^2$$

- Maximum is at E=0 (as long as α>ρ0): Equilibrium behavior dominated by the original signal
- For α<0.58, a secondary maximum appears (meta-stable state): spinodal point
- A steepest ascent dynamics starting from large E would reach the signal for α>0.58, but would stay block in the meta-stable state for α<0.58, even if the true equilibrium is at E=0.

Computing the free entropy

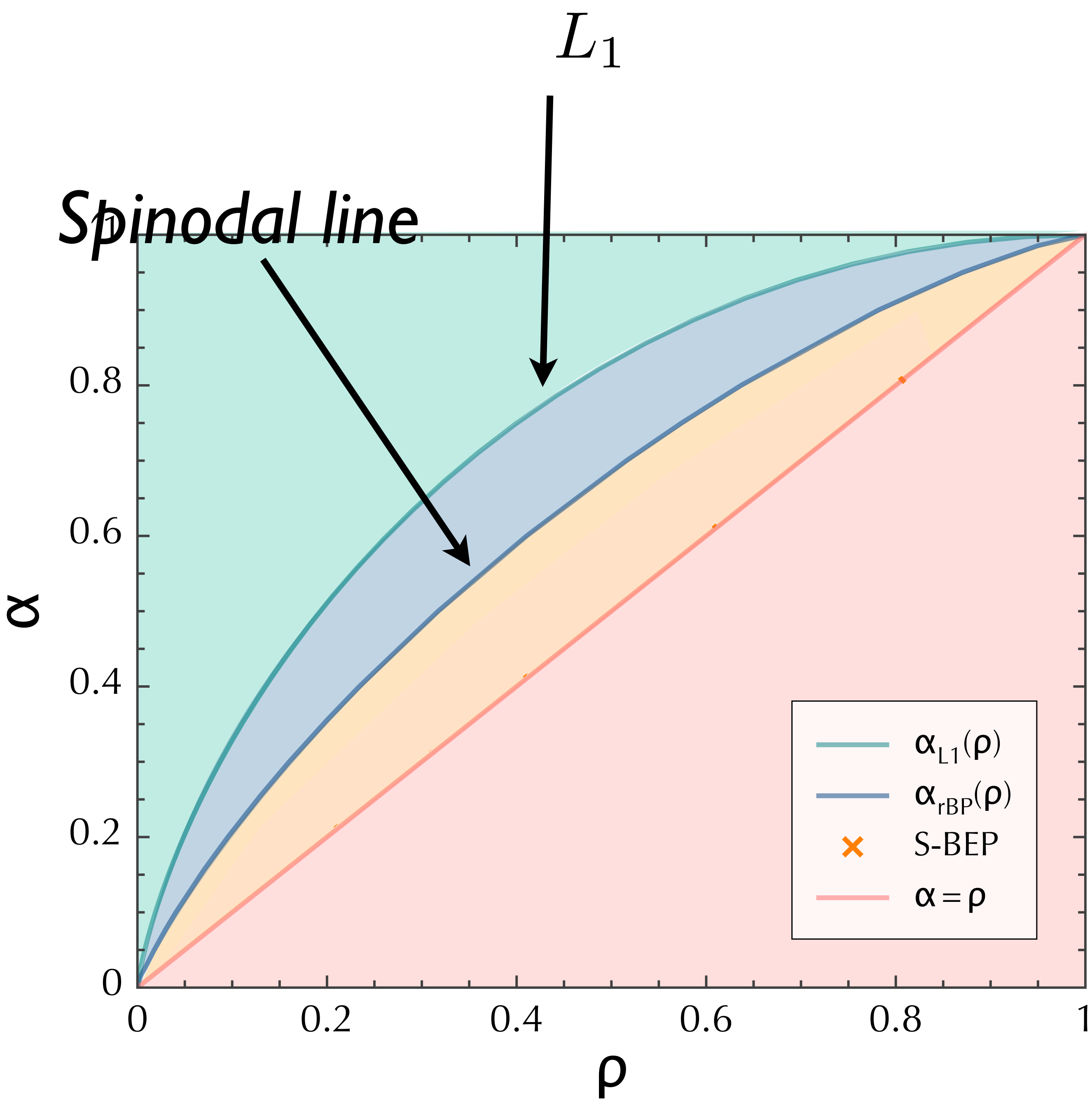
Example with ρ_0=0.4, and Φ_0 a Gaussian distribution with zero mean and unit variance

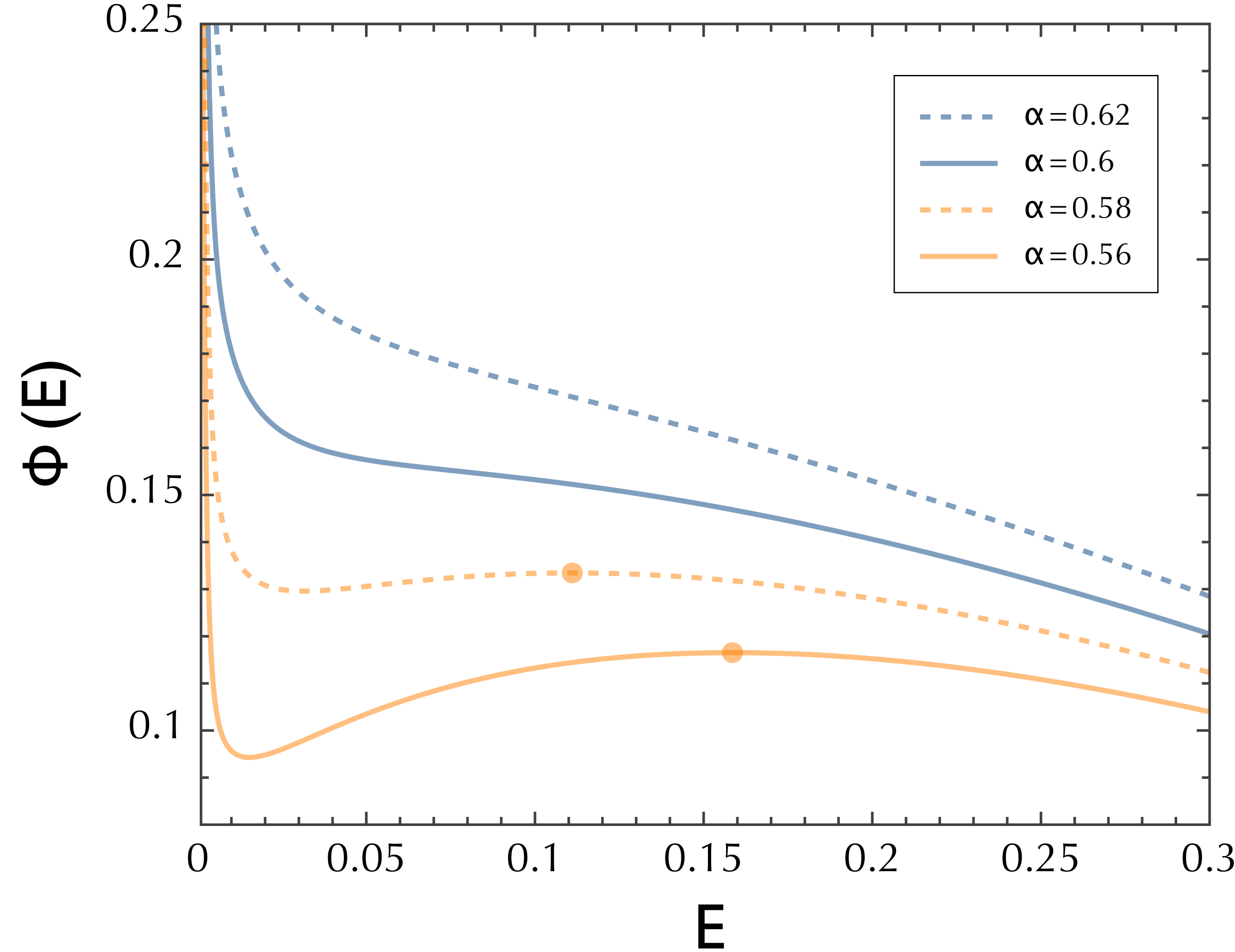


$$E = \frac{1}{N}\sum_i \left(\langle x_i \rangle - x_i^0\right)^2$$

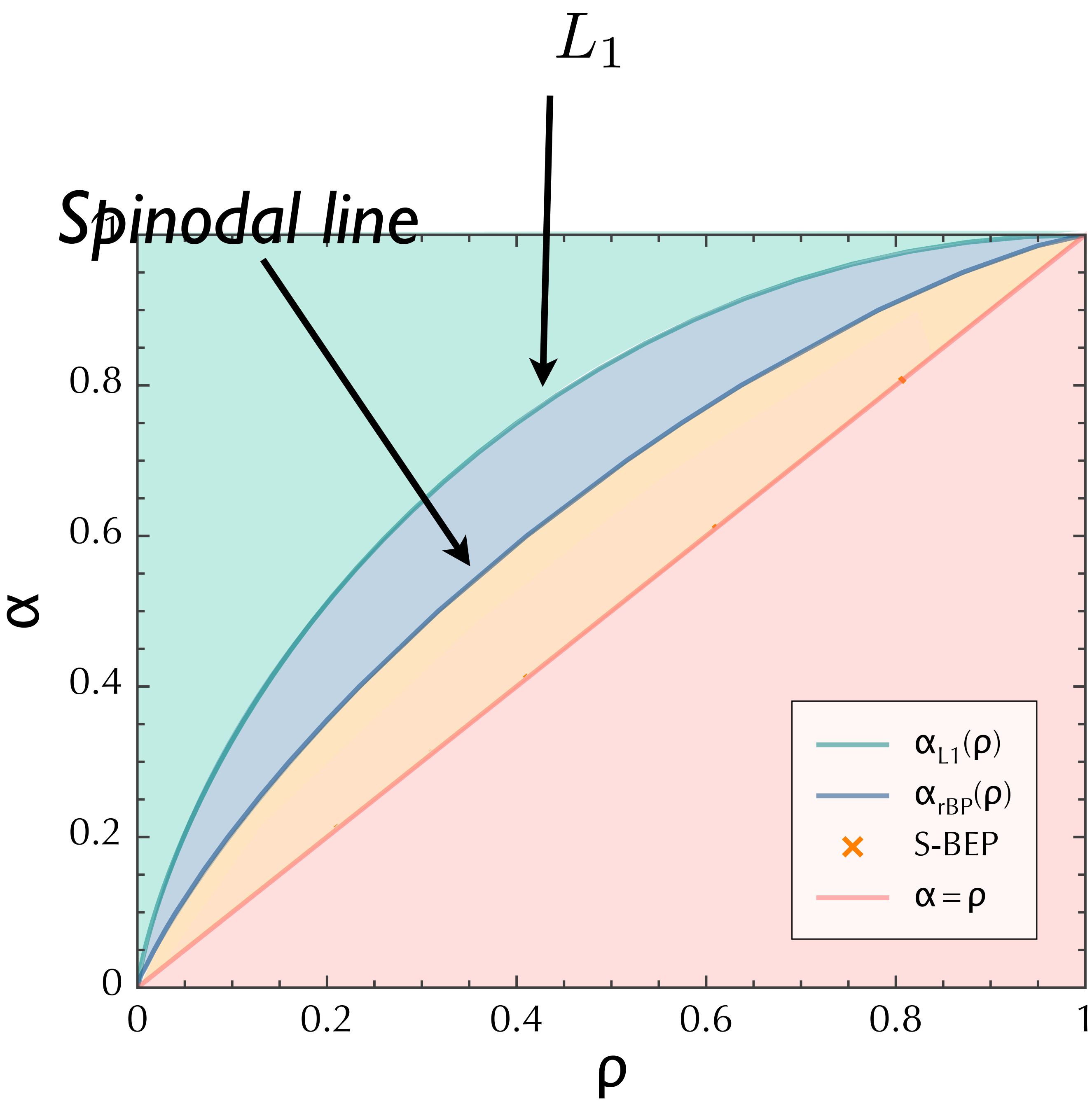
- Maximum is at E=0 (as long as $\alpha>\rho_0$): Equilibrium behavior dominated by the original signal
- For $\alpha<0.58$, a secondary maximum appears (meta-stable state): spinodal point
- A steepest ascent dynamics starting from large E would reach the signal for $\alpha>0.58$, but would stay block in the meta-stable state for $\alpha<0.58$, even if the true equilibrium is at E=0.
- Similarity with supercooled liquids

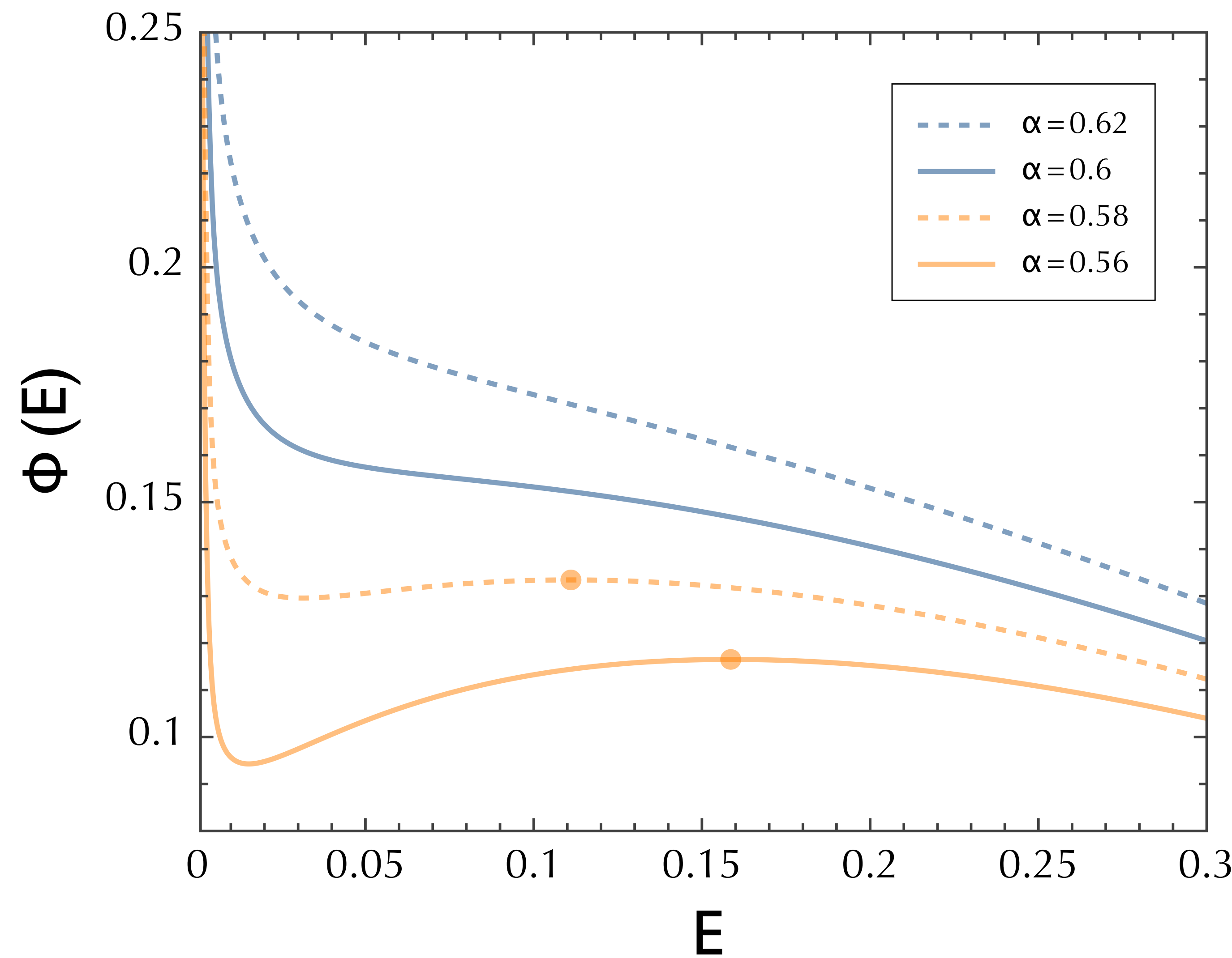
Computing the Phase Diagram





Computing the Phase Diagram

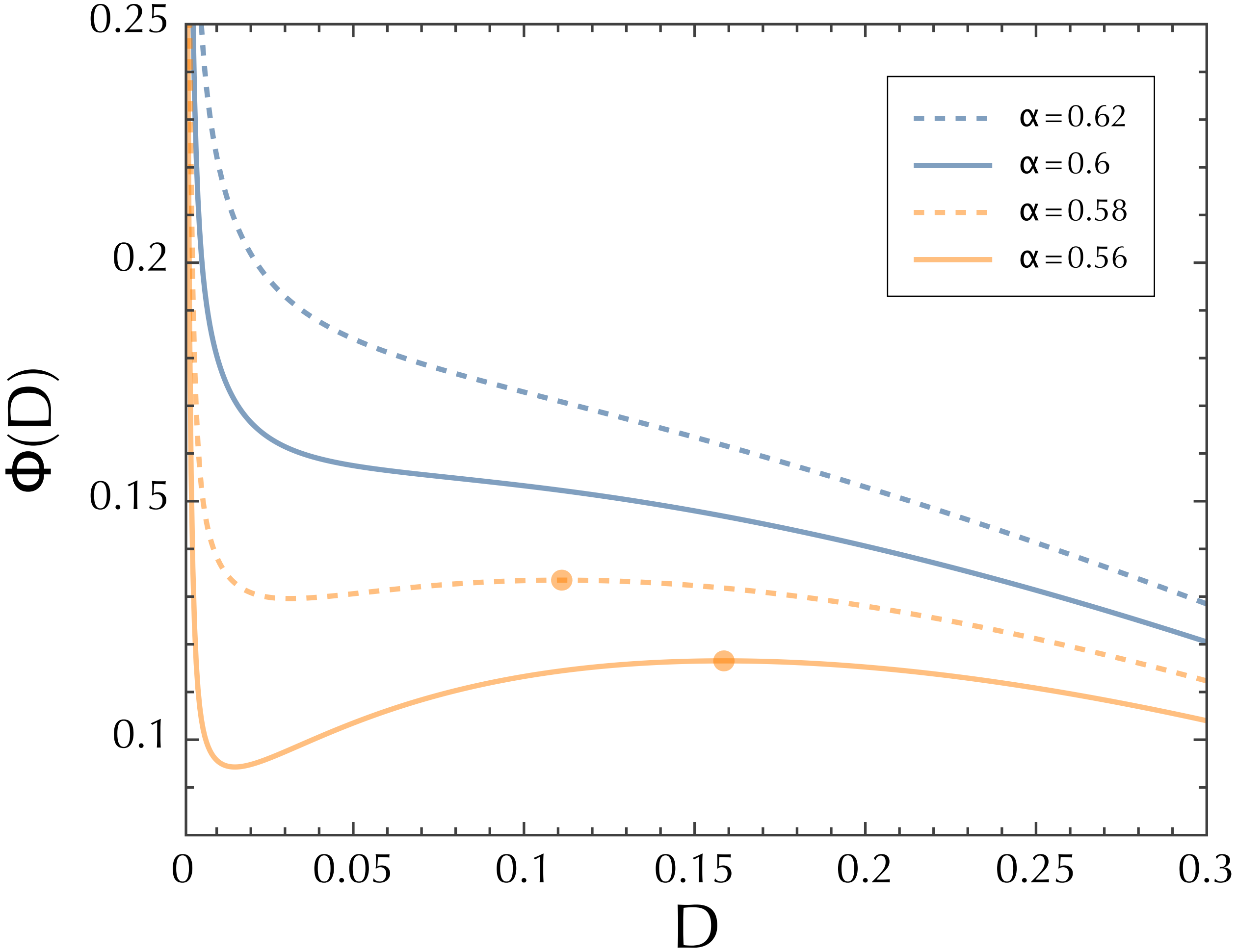




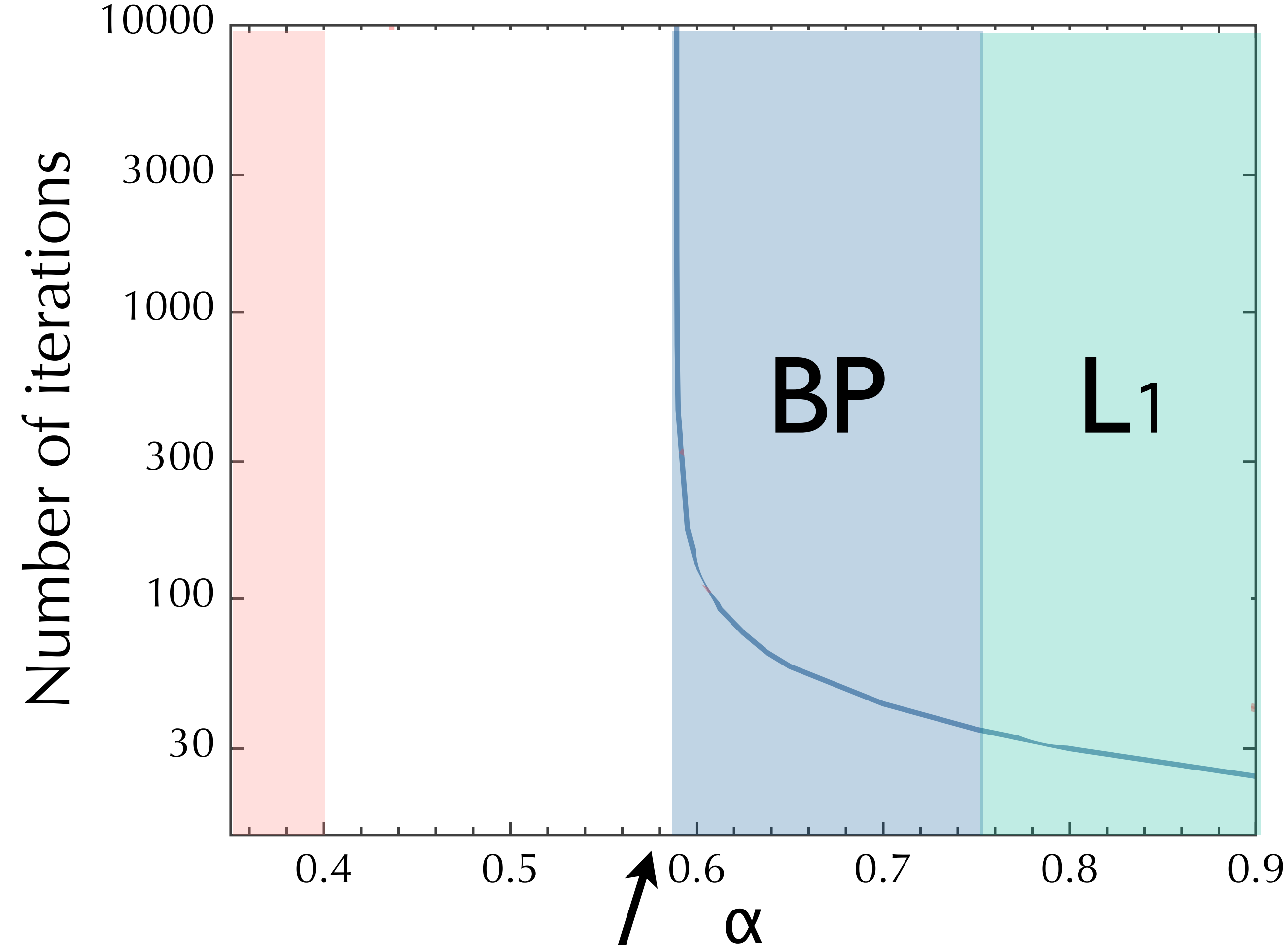
A steepest ascent of the free entropy allows
a perfect reconstruction until the spinodal line.
This is more efficient than L_1-minimization

Thermodynamic potential

BP convergence time



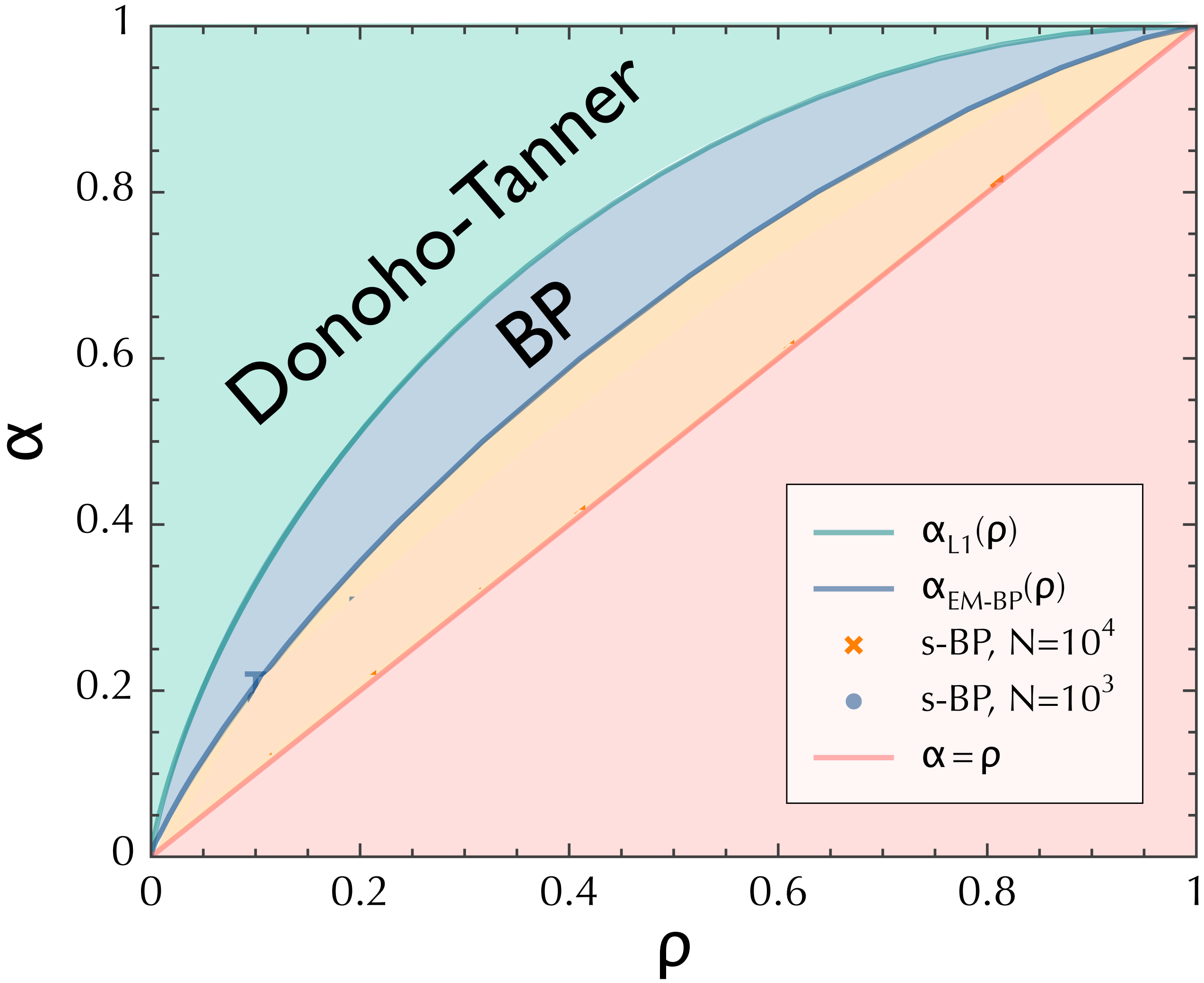
distance to native state



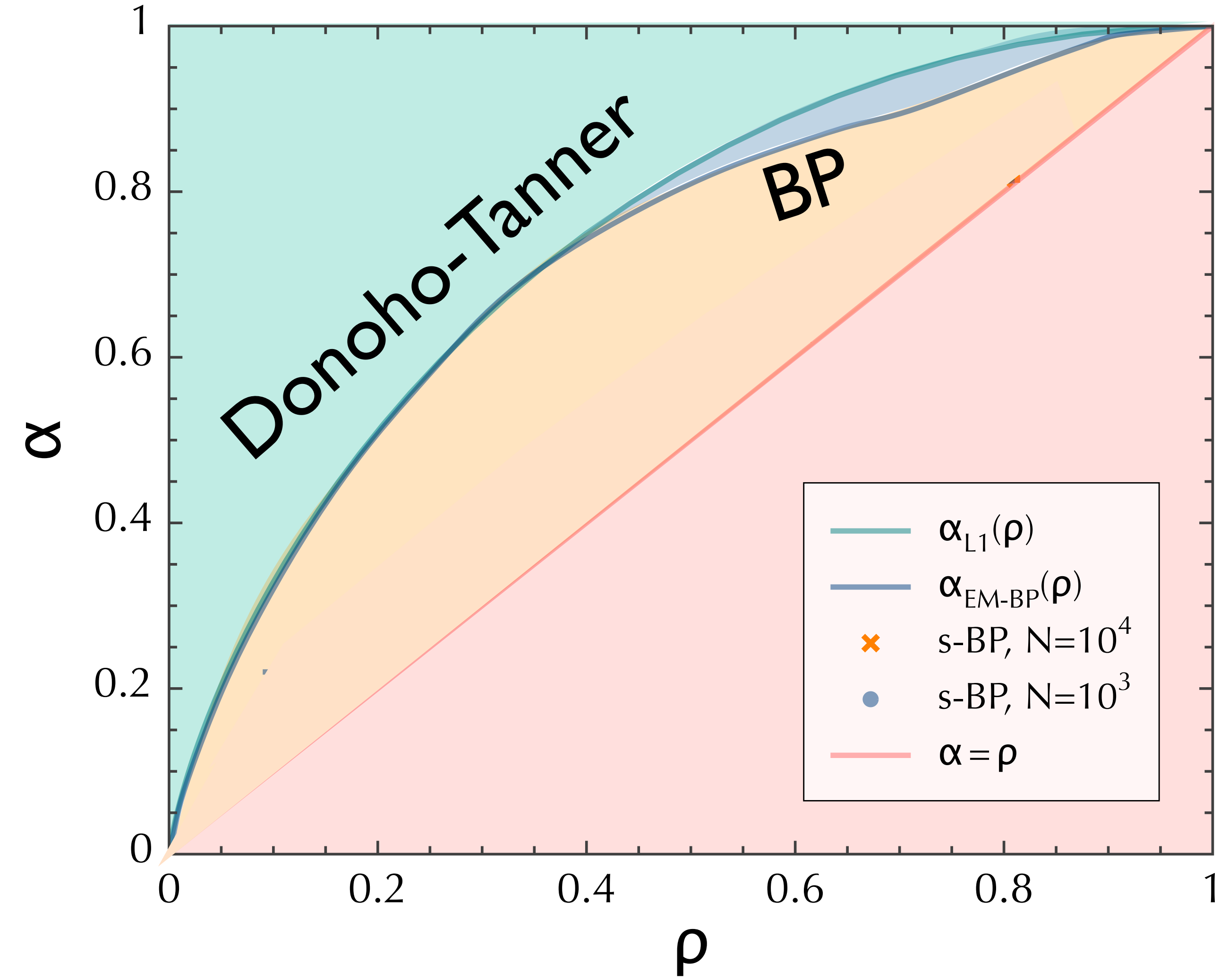
Spinodal transition
(supercooled limit)

Trying different type of signals

The limit depends on the type of signal
(while the Donoho-Tanner is universal)



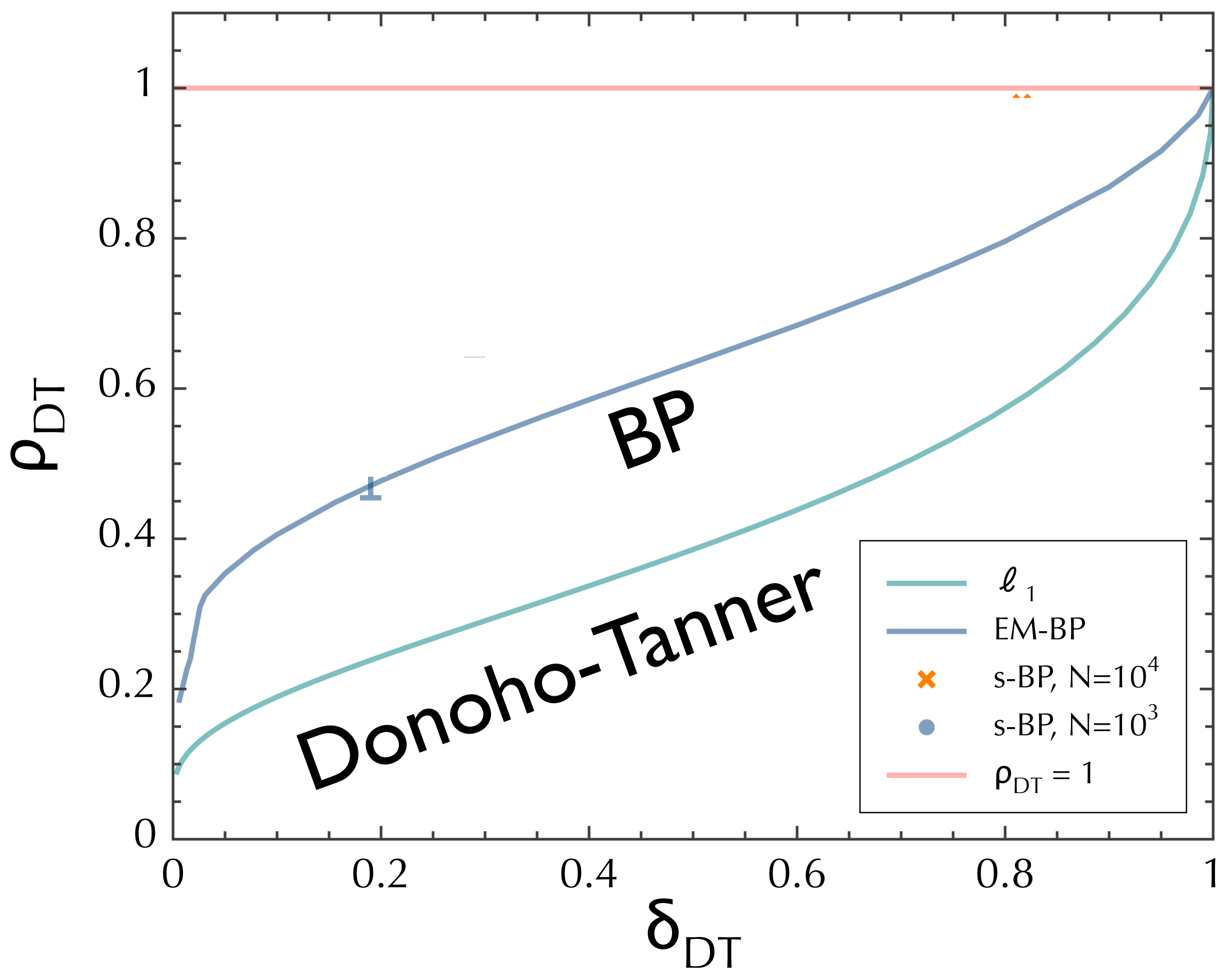
Gauss-Bernoulli signal



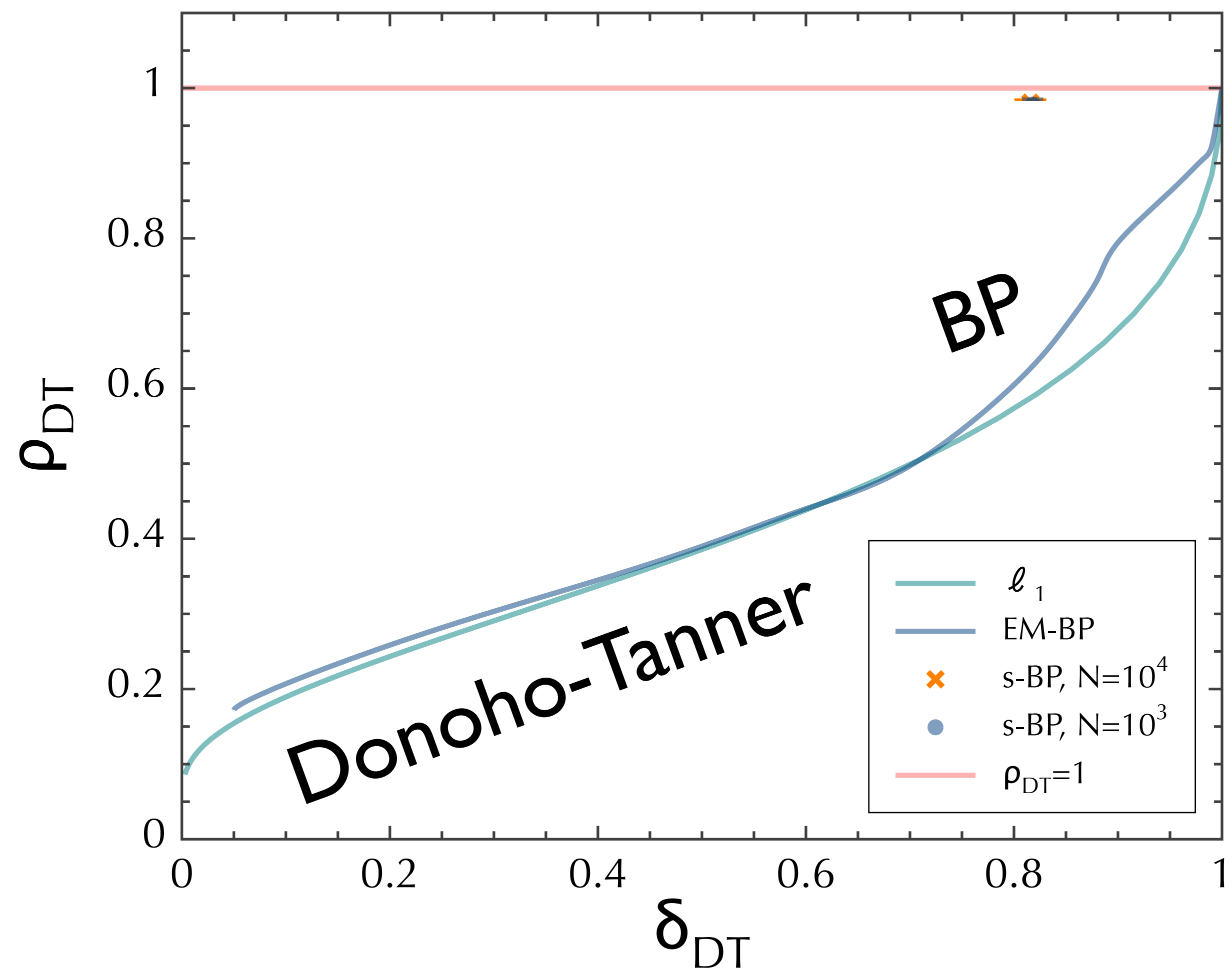
Binary signals

Trying different type of signals

The limit depends on the type of signal
(while the Donoho-Tanner is universal)

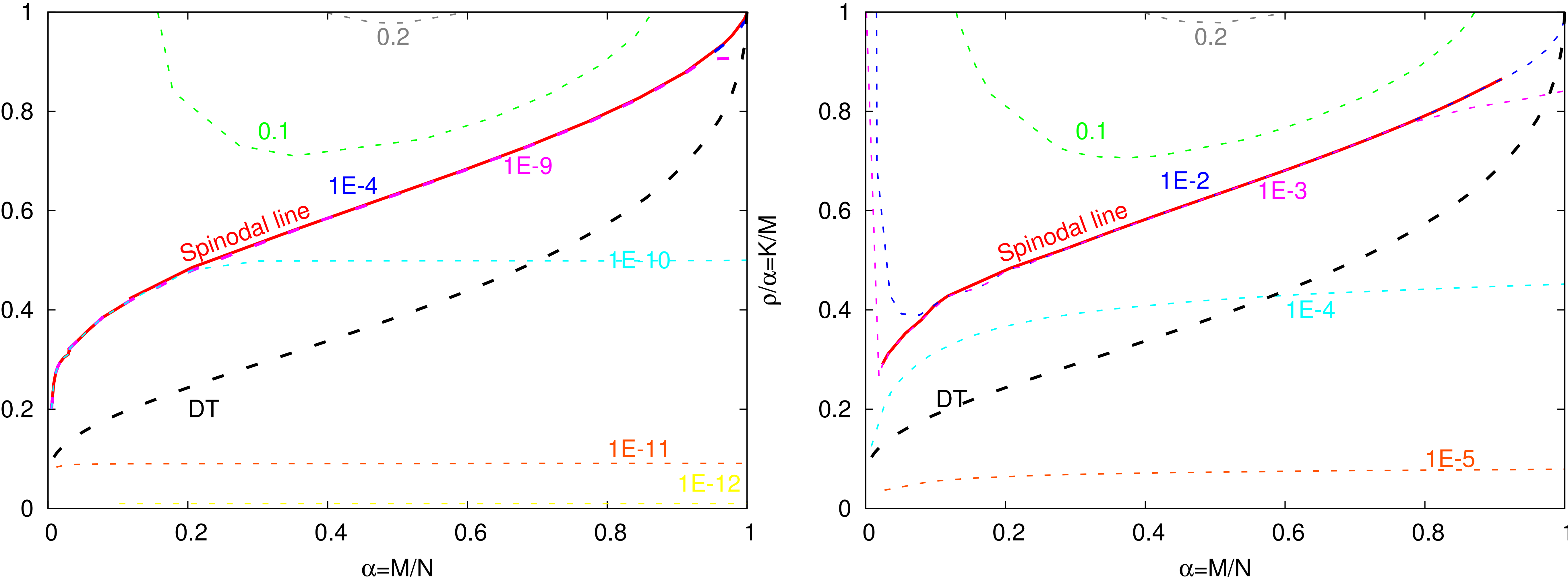


Gauss-Bernoulli signal



Binary signals

BP is Robust to noise



Noise with $\sigma = 10^{-5}$

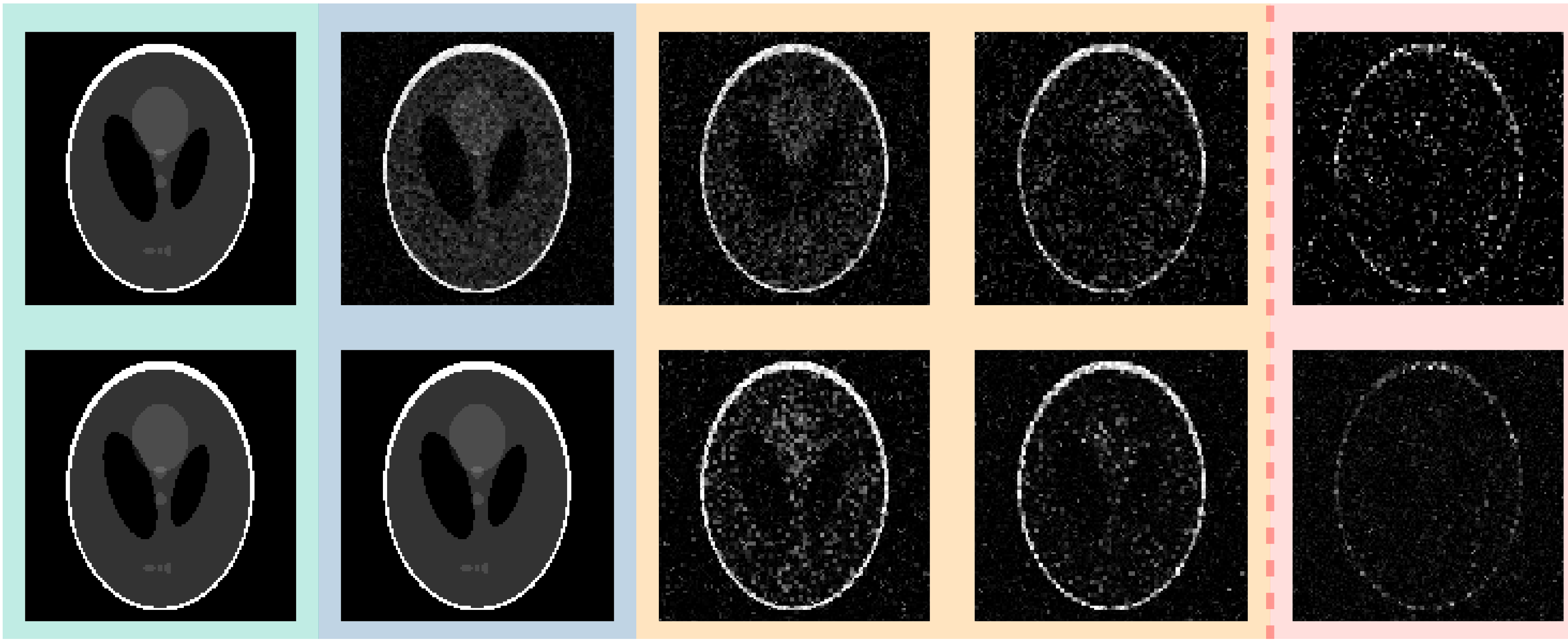
Noise $\sigma = 10^{-2}$

A more complex signal

$\alpha = \rho \approx 0.15$

L_1

BP



Shepp-Logan phantom, in the Haar-wavelet representation

$\alpha = 0.5$ $\alpha = 0.4$ $\alpha = 0.3$ $\alpha = 0.2$ $\alpha = 0.1$

BP + probabilistic approach

- Efficient and fast
- Robust to noise
- Very flexible (more information can be put in the prior)

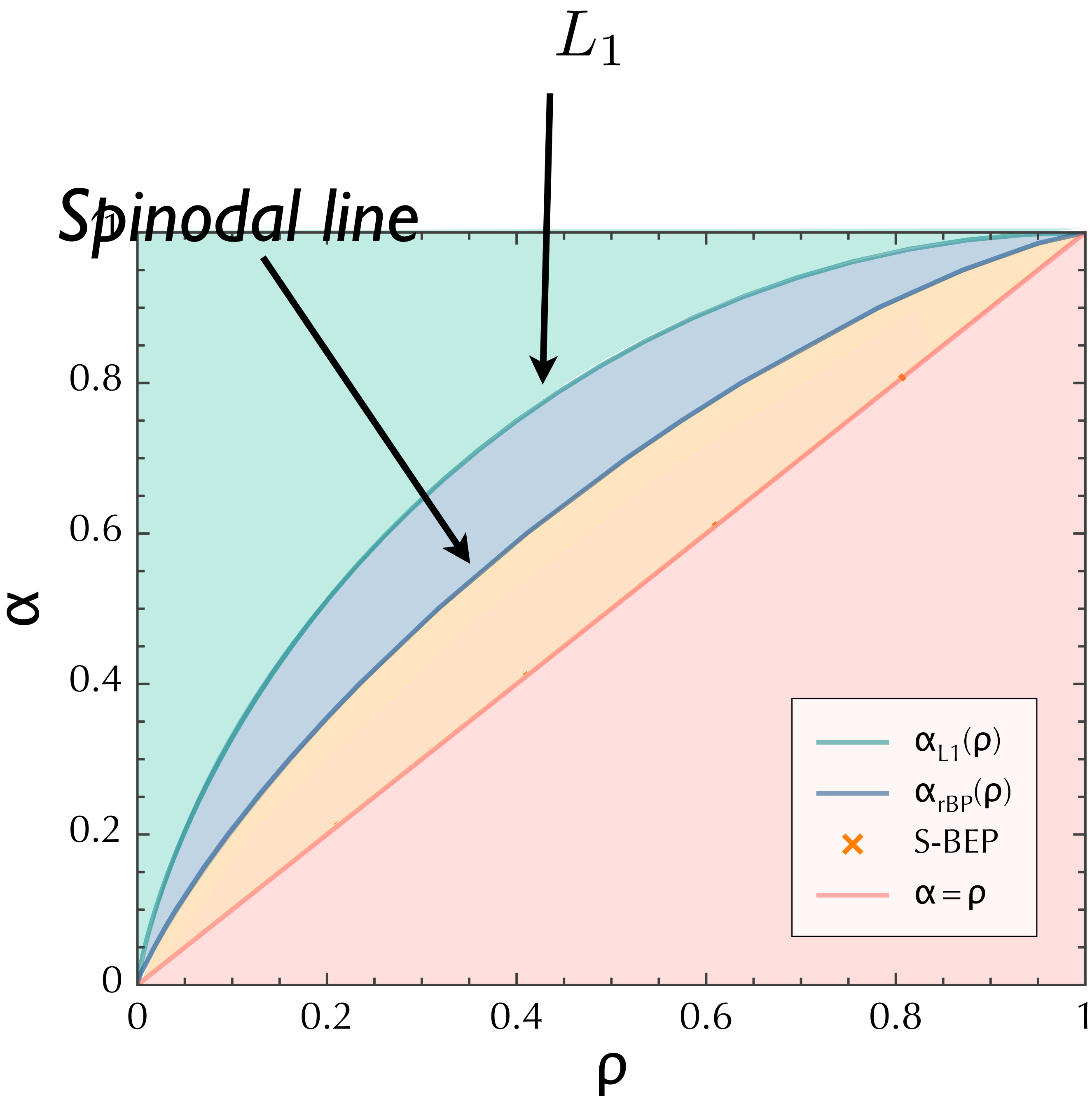
$$P(\vec{x}|\vec{y}) = \frac{1}{Z} \prod_{i=1}^{N} \left[(1-\rho)\,\delta(x_i) + \rho\phi(x_i)\right] \prod_{\mu=1}^{M} \delta\left(y_\mu - \sum_{i=1}^{N} F_{\mu i} x_i\right)$$

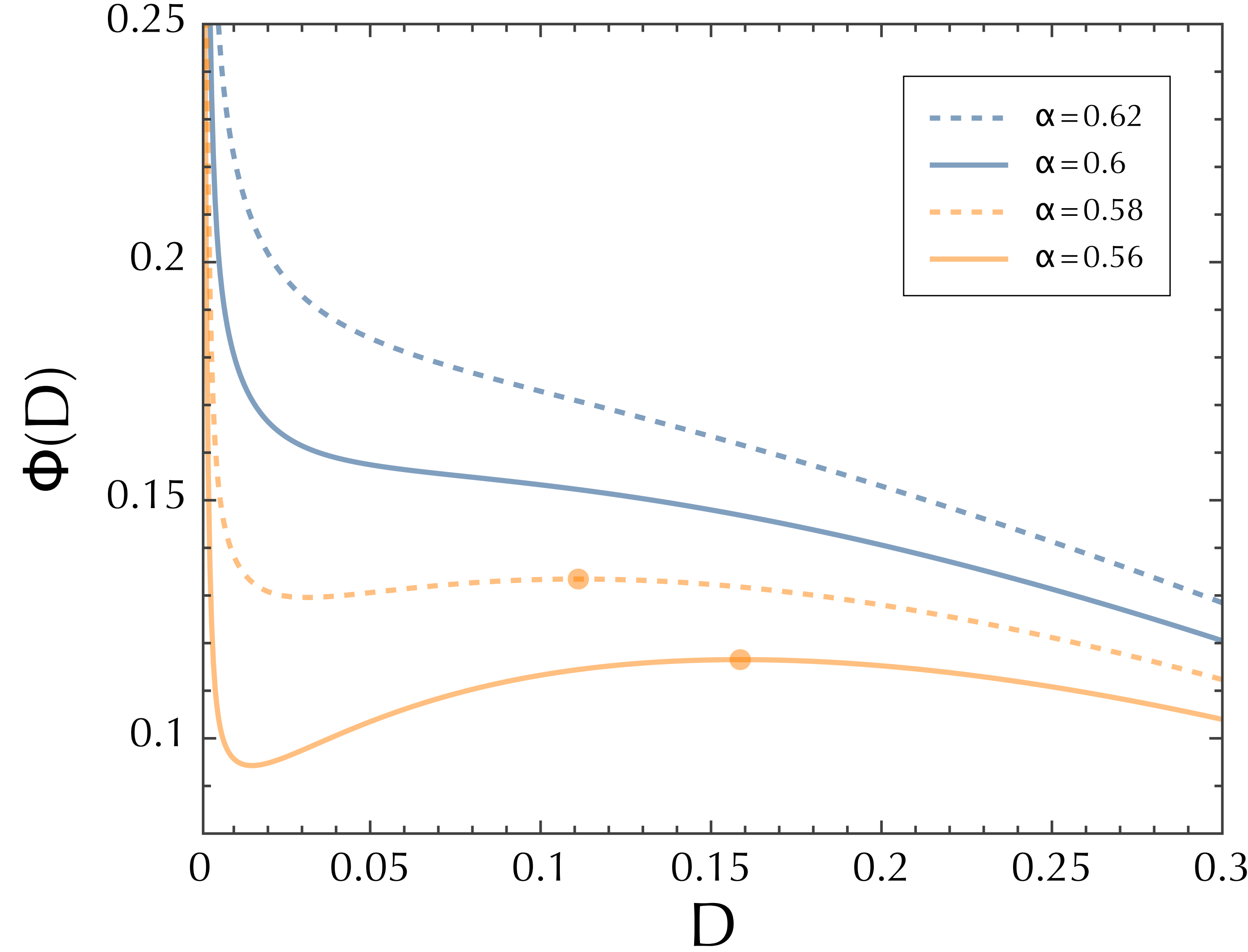
Our work

A statistical physics approach to compressed sensing

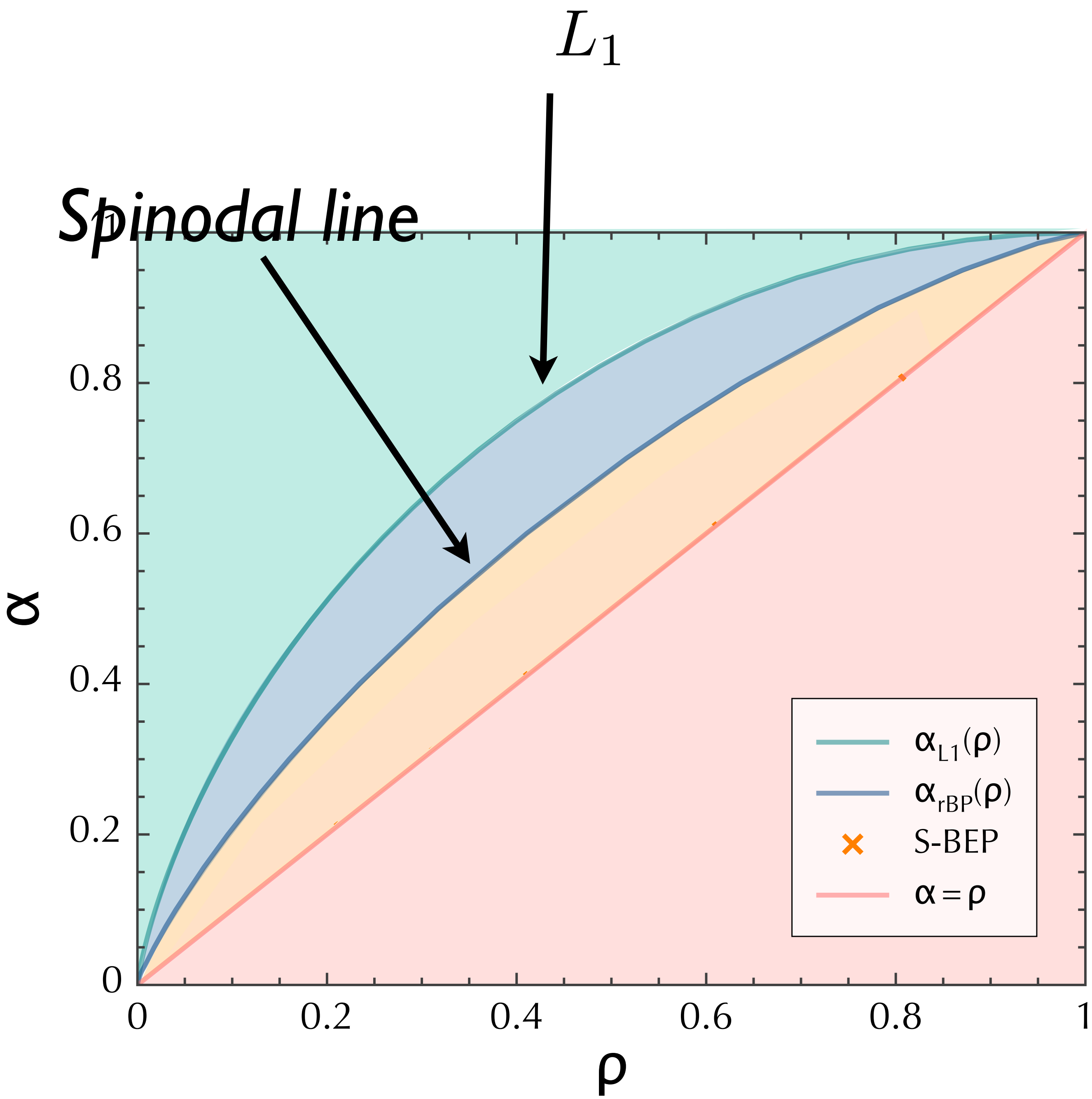
- A probabilistic approach to reconstruction
- The Belief Propagation algorithm
- **Seeded measurements matrices**

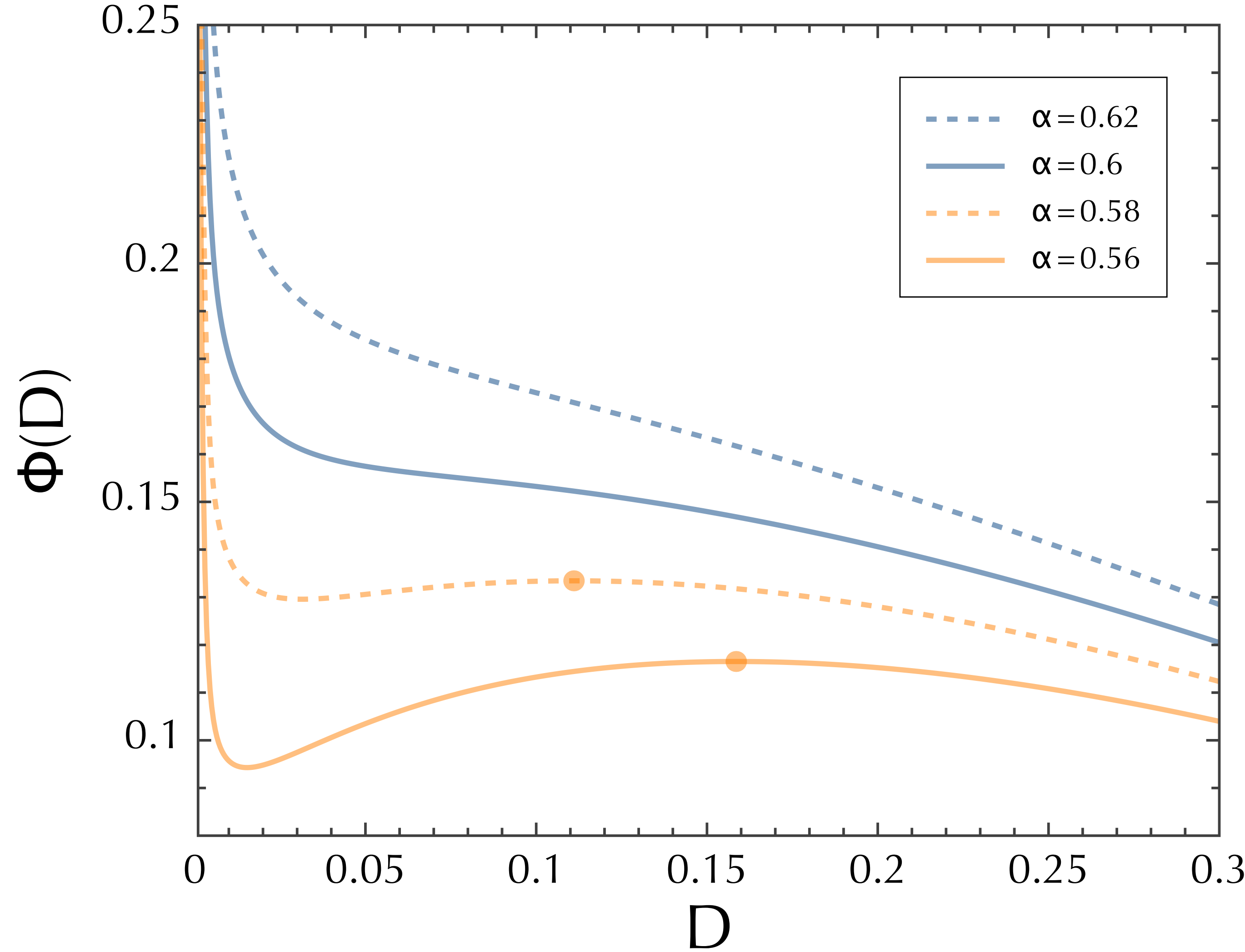
This is good, but not good enough





This is good, but not good enough

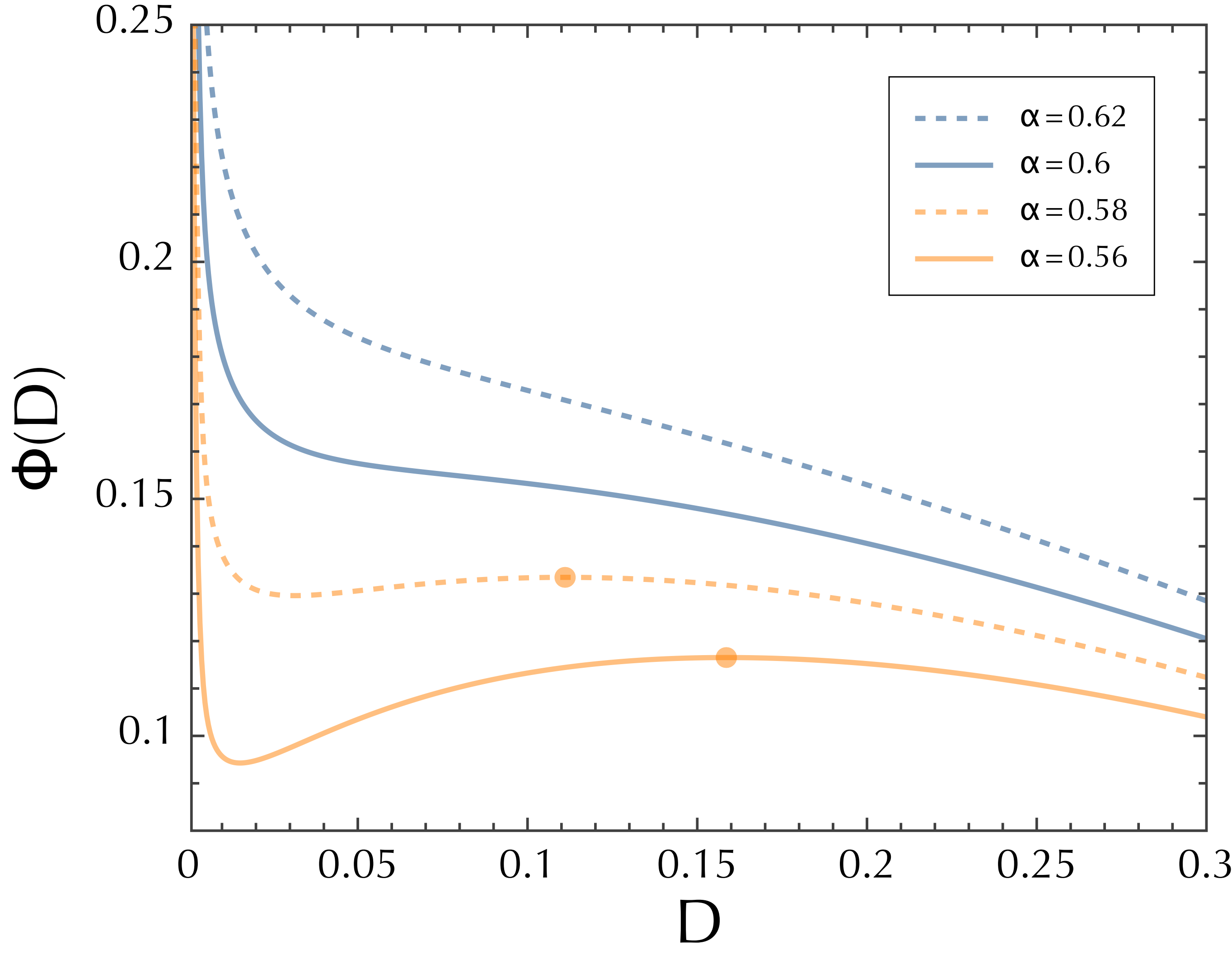




The dynamics is stuck in a metastable state, just as a liquid cooled too fast remains in a supercooled liquid state instead of crystalizing

This is good, but not good enough

How to pass the spinodal point?



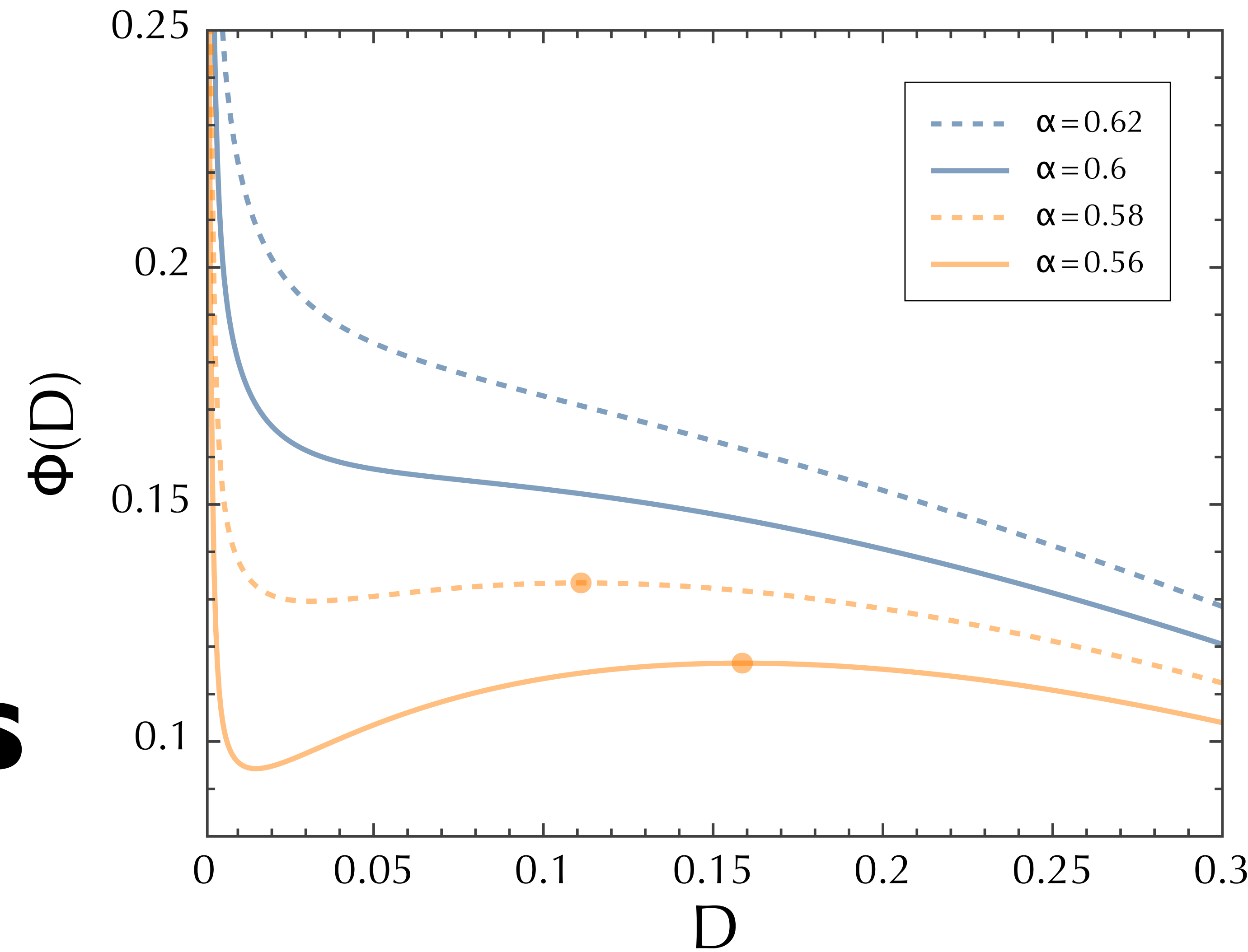
The dynamics is stuck in a metastable state, just as a liquid cooled too fast remains in a supercooled liquid state instead of crystalizing

This is good, but not good enough

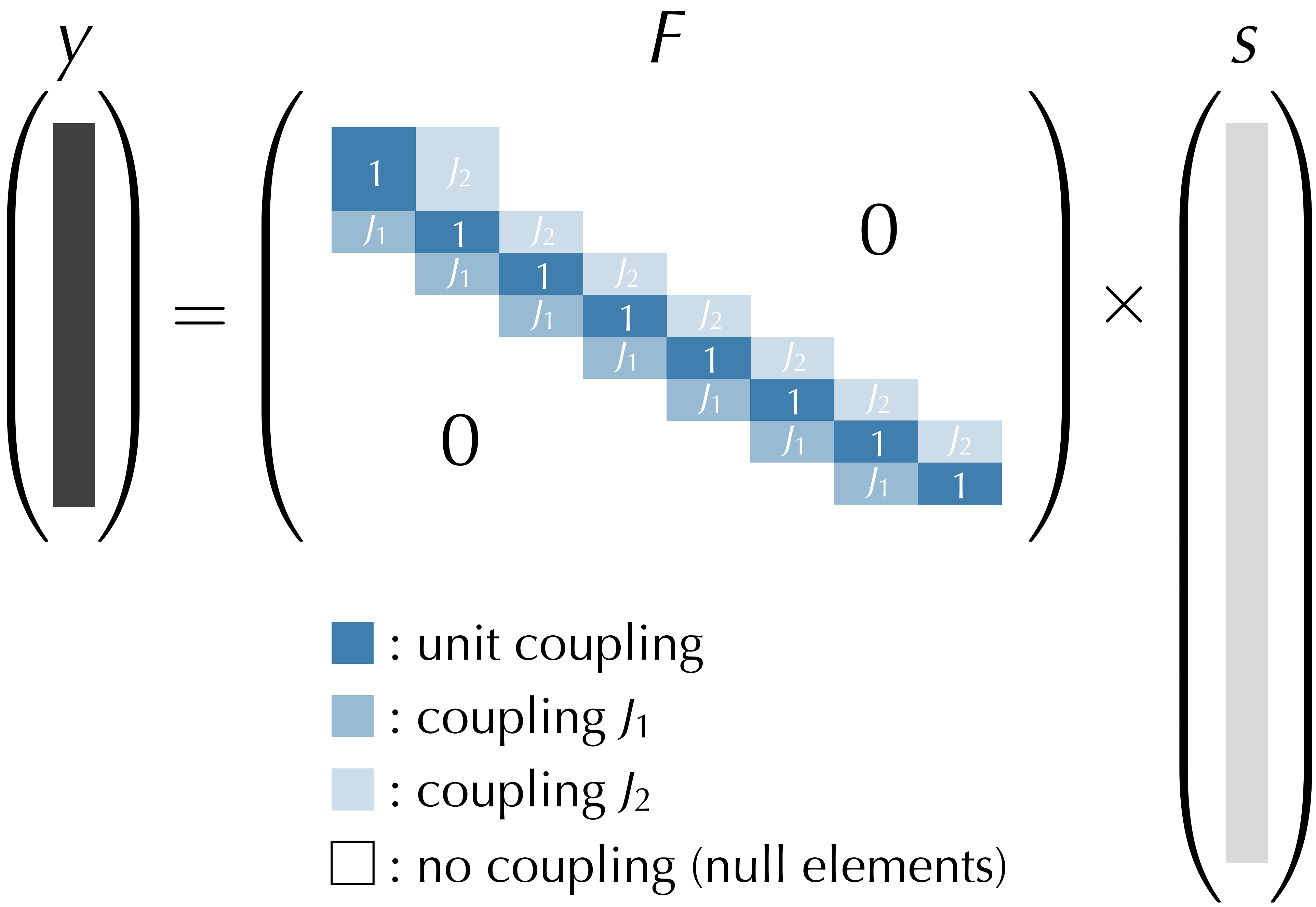
How to pass the spinodal point?

By nucleation!

Special design of "seeded" matrices



The dynamics is stuck in a metastable state, just as a liquid cooled too fast remains in a supercooled liquid state instead of crystalizing



$L = 8$

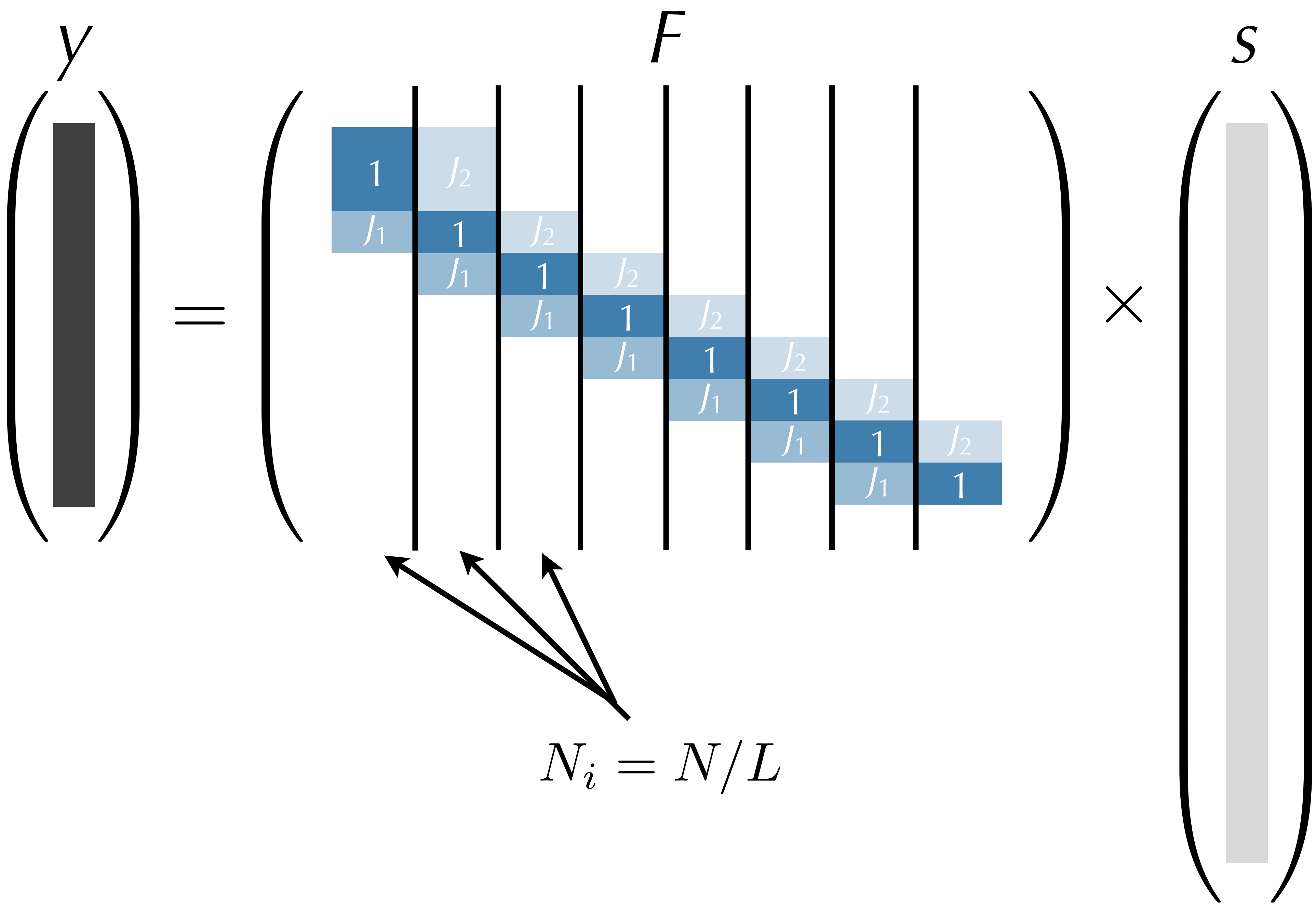
$N_i = N/L$

$M_i = \alpha_i N/L$

$\alpha_1 > \alpha_{BP}$

$\alpha_j = \alpha' < \alpha_{BP} \qquad j \geq 2$

$$\alpha = \frac{1}{L}\left(\alpha_1 + (L-1)\alpha'\right)$$



$$N_i = N/L$$

$$L = 8$$

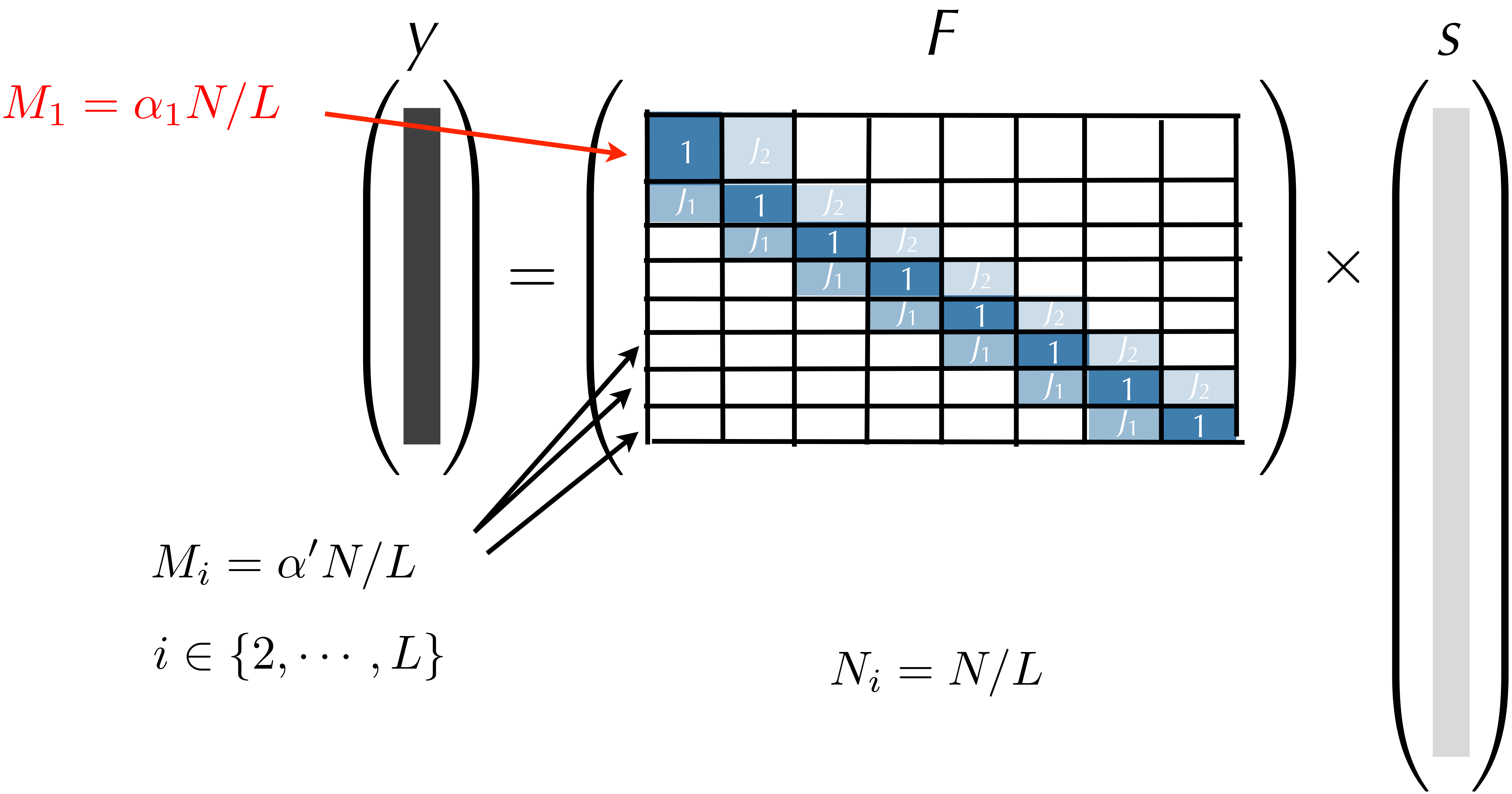
$$N_i = N/L$$

$$M_i = \alpha_i N/L$$

$$\alpha_1 > \alpha_{BP}$$

$$\alpha_j = \alpha' < \alpha_{BP} \qquad j \geq 2$$

$$\alpha = \frac{1}{L}\left(\alpha_1 + (L-1)\alpha'\right)$$



$L = 8$

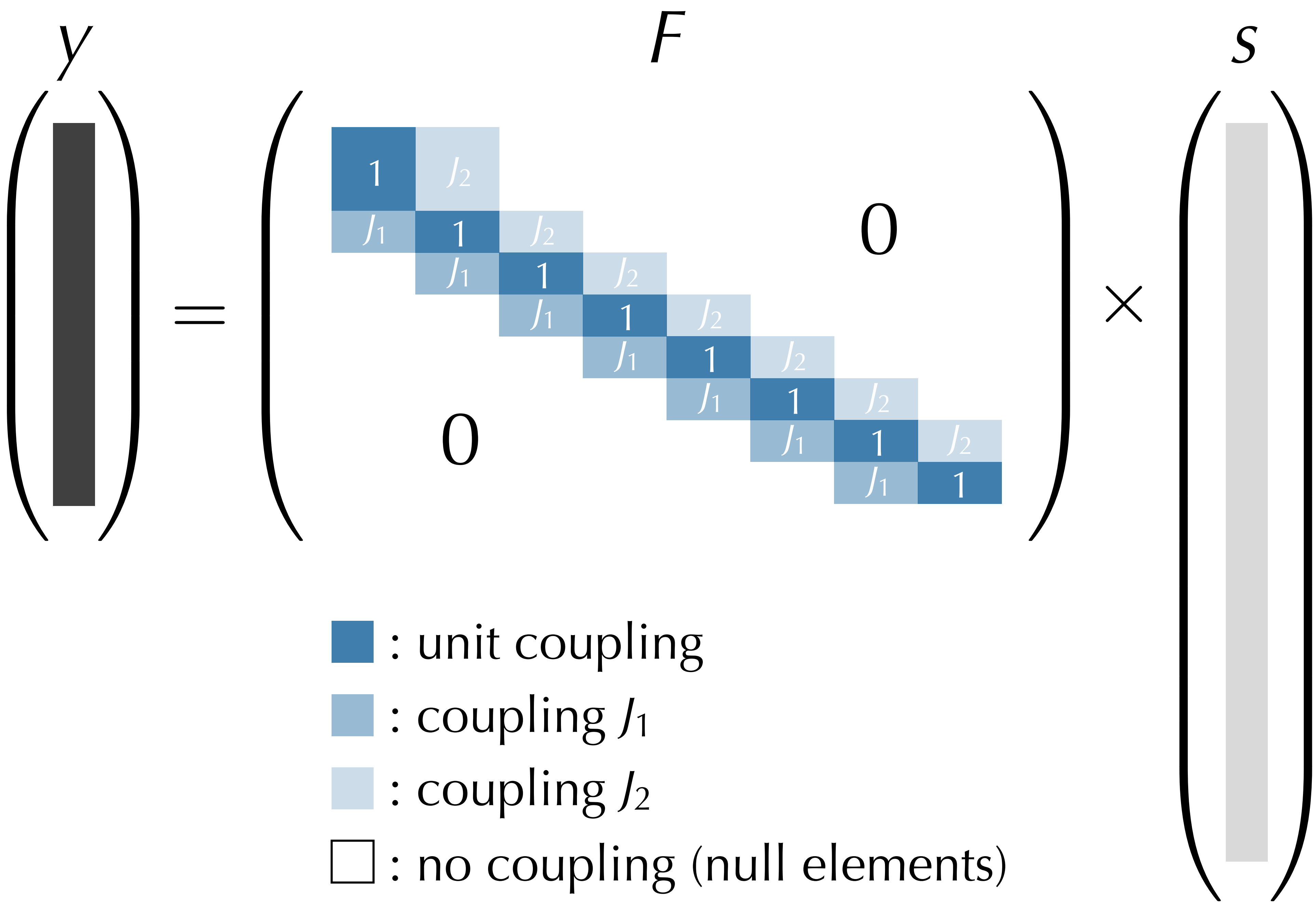
$N_i = N/L$

$M_i = \alpha_i N/L$

$\alpha_1 > \alpha_{BP}$

$\alpha_j = \alpha' < \alpha_{BP} \qquad j \geq 2$

$$\alpha = \frac{1}{L}\left(\alpha_1 + (L-1)\alpha'\right)$$



$$L = 8$$

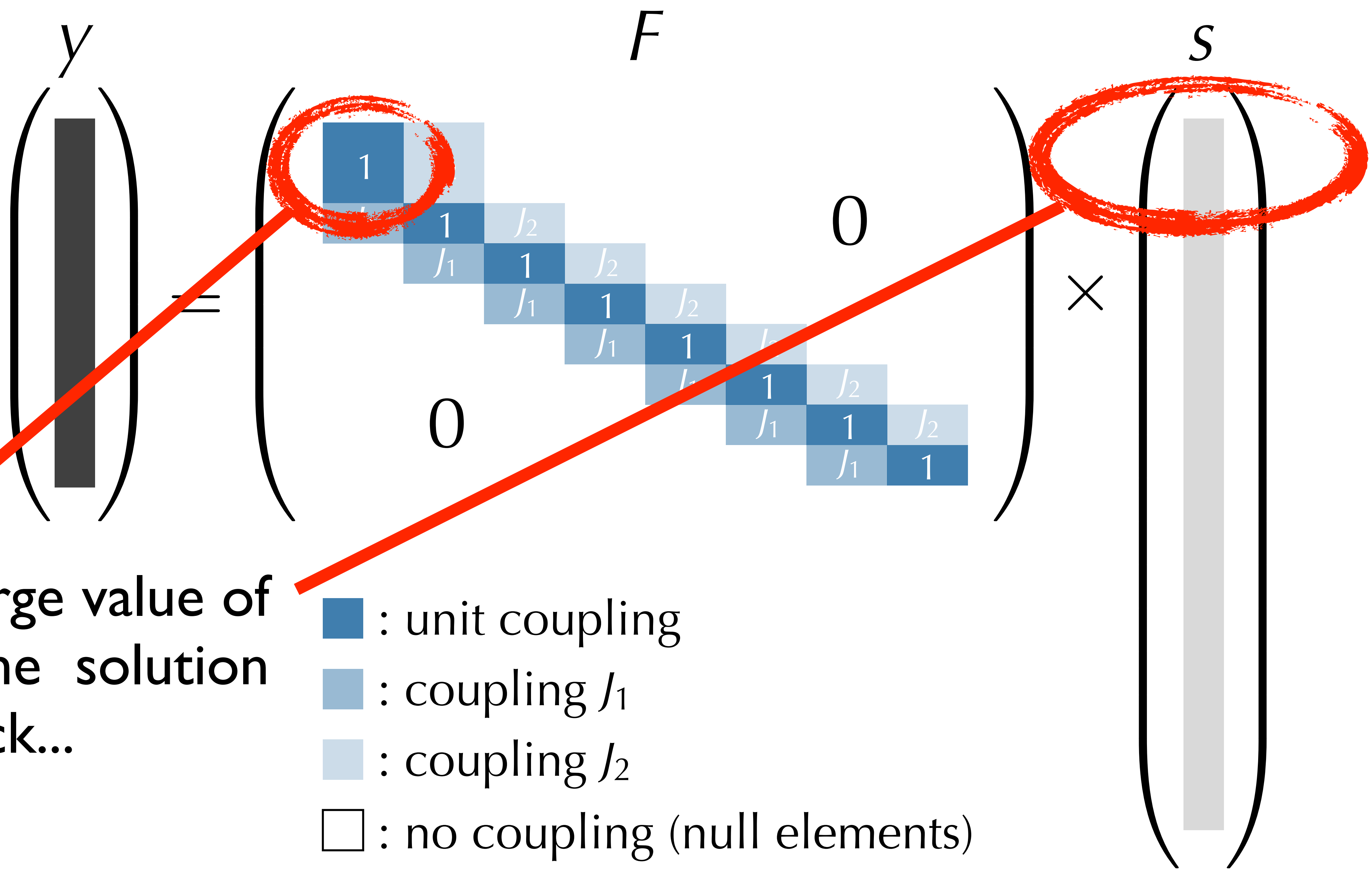
$$N_i = N/L$$

$$M_i = \alpha_i N/L$$

$$\alpha_1 > \alpha_{BP}$$

$$\alpha_j = \alpha' < \alpha_{BP} \qquad j \geq 2$$

$$\alpha = \frac{1}{L}\left(\alpha_1 + (L-1)\alpha'\right)$$



Block 1 has a large value of M such that the solution arise in this block...

$$L = 8$$

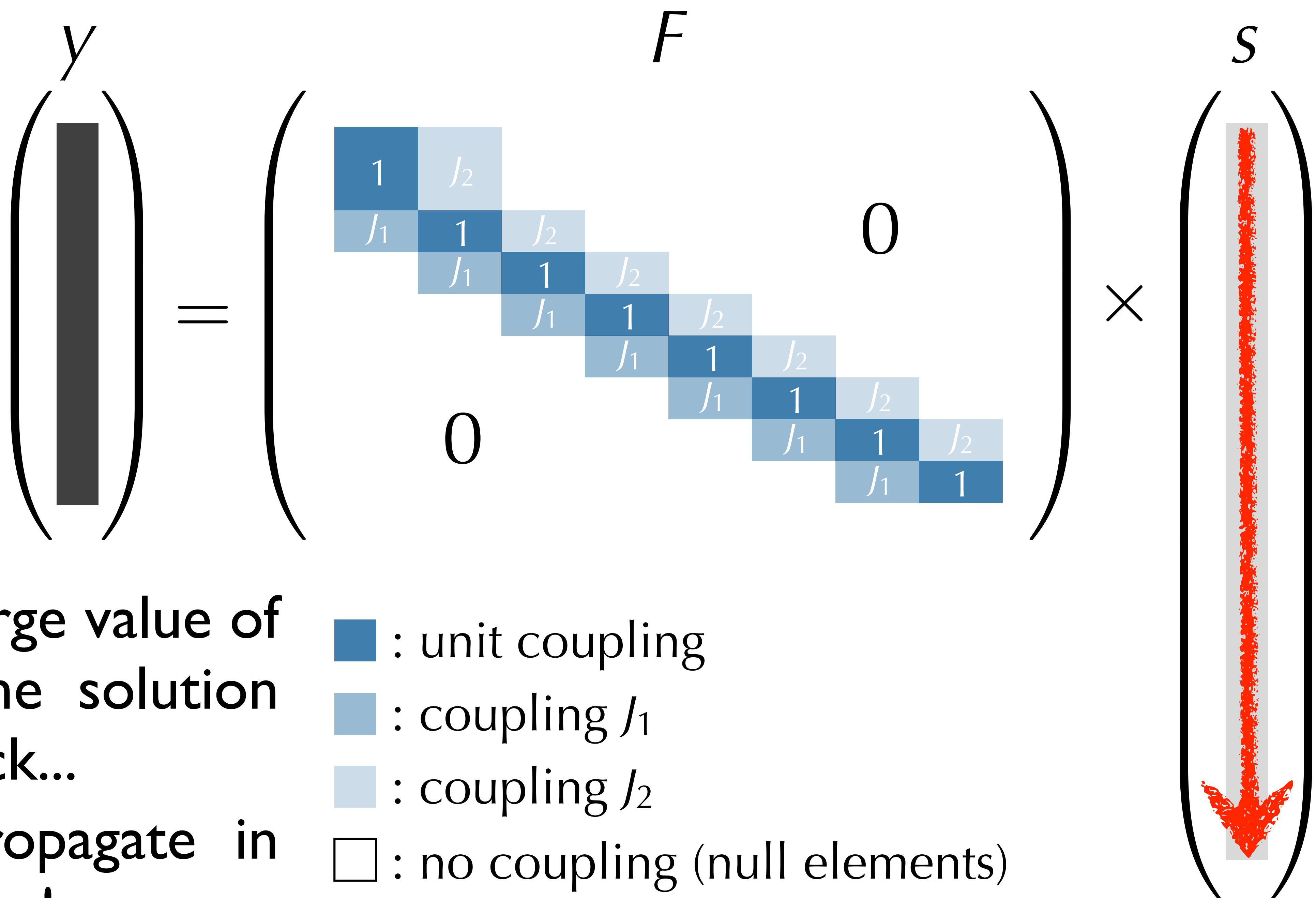
$$N_i = N/L$$

$$M_i = \alpha_i N/L$$

$$\alpha_1 > \alpha_{BP}$$

$$\alpha_j = \alpha' < \alpha_{BP} \qquad j \geq 2$$

$$\alpha = \frac{1}{L}\left(\alpha_1 + (L-1)\alpha'\right)$$



Block 1 has a large value of M such that the solution arise in this block...

... and then propagate in the whole system!

$$L = 8$$

$$N_i = N/L$$

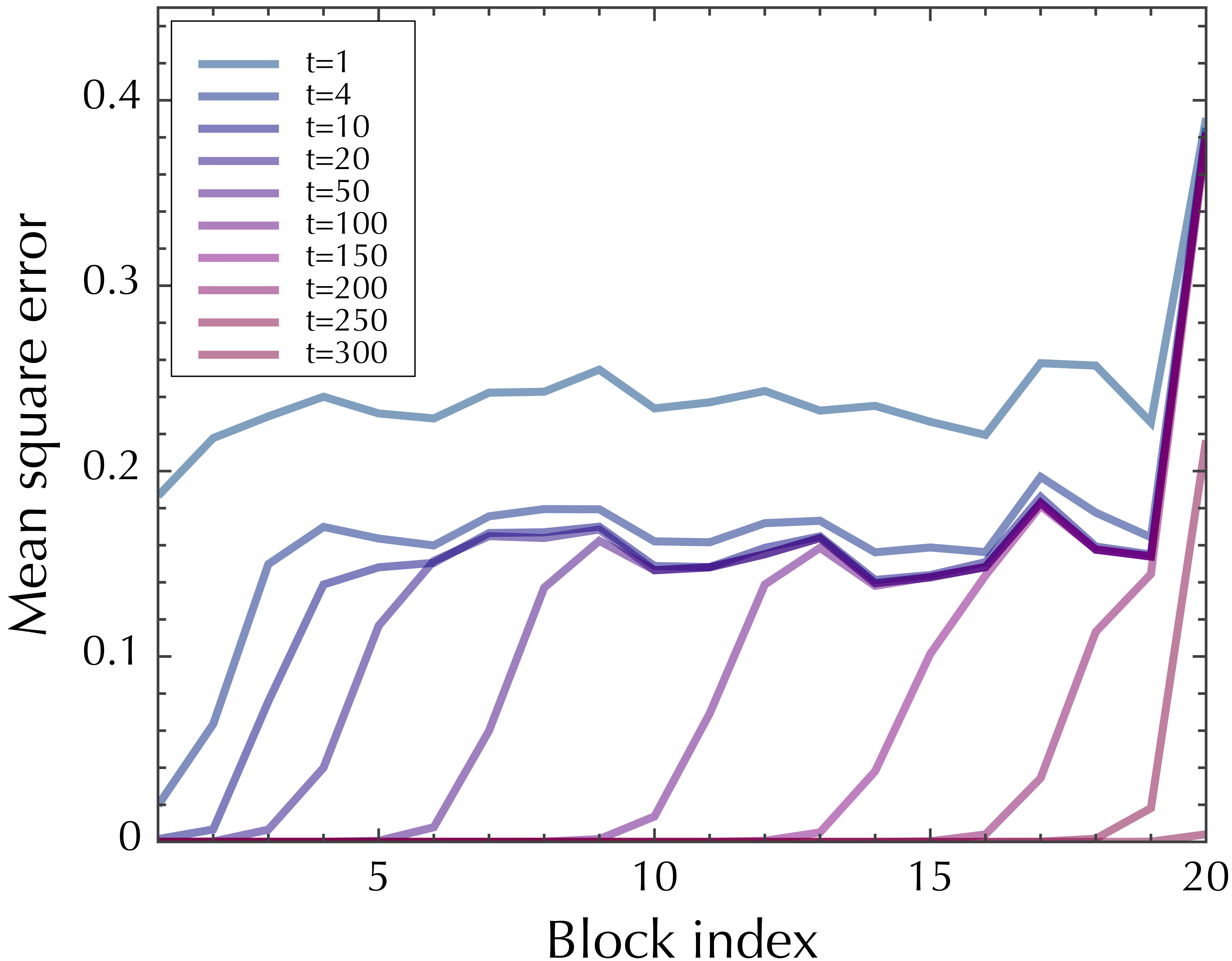
$$M_i = \alpha_i N/L$$

$$\alpha_1 > \alpha_{BP}$$

$$\alpha_j = \alpha' < \alpha_{BP} \qquad j \geq 2$$

$$\alpha = \frac{1}{L}\left(\alpha_1 + (L-1)\alpha'\right)$$

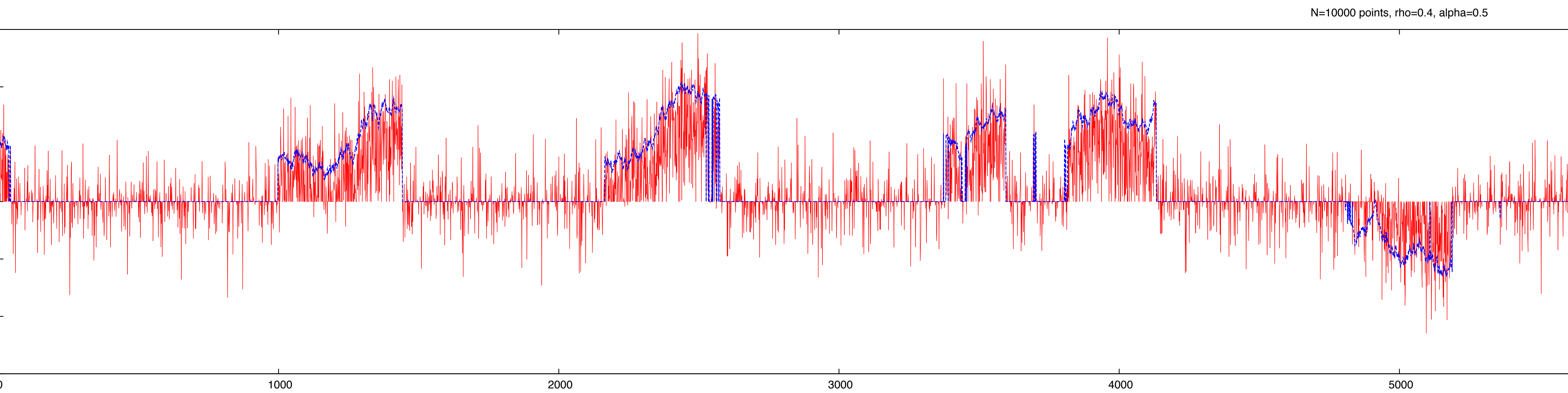
Example with ρ_0=0.4, and Φ_0 a Gaussian distribution with 0 mean and unit variance



$L = 20$ $\quad N = 50000$ $\quad \rho = .4$ $\quad J_1 = 20$ $\quad \alpha_1 = 1$

$J_2 = .2$ $\quad \alpha = .5$

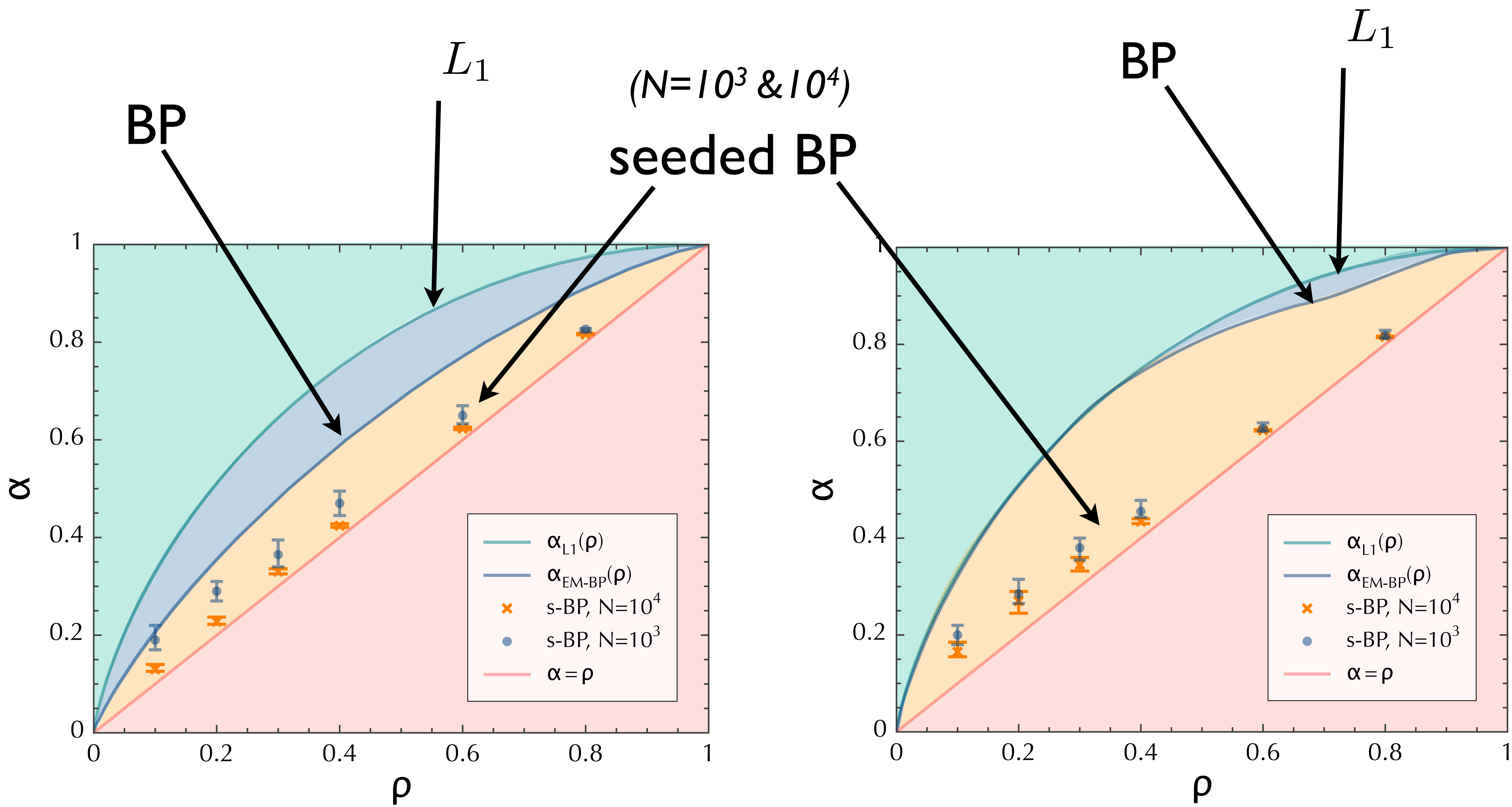
A signal with α=0.5 and ρ=0.4



Blue is the true signal reconstructed by s-BP

Red is the signal found by L_1

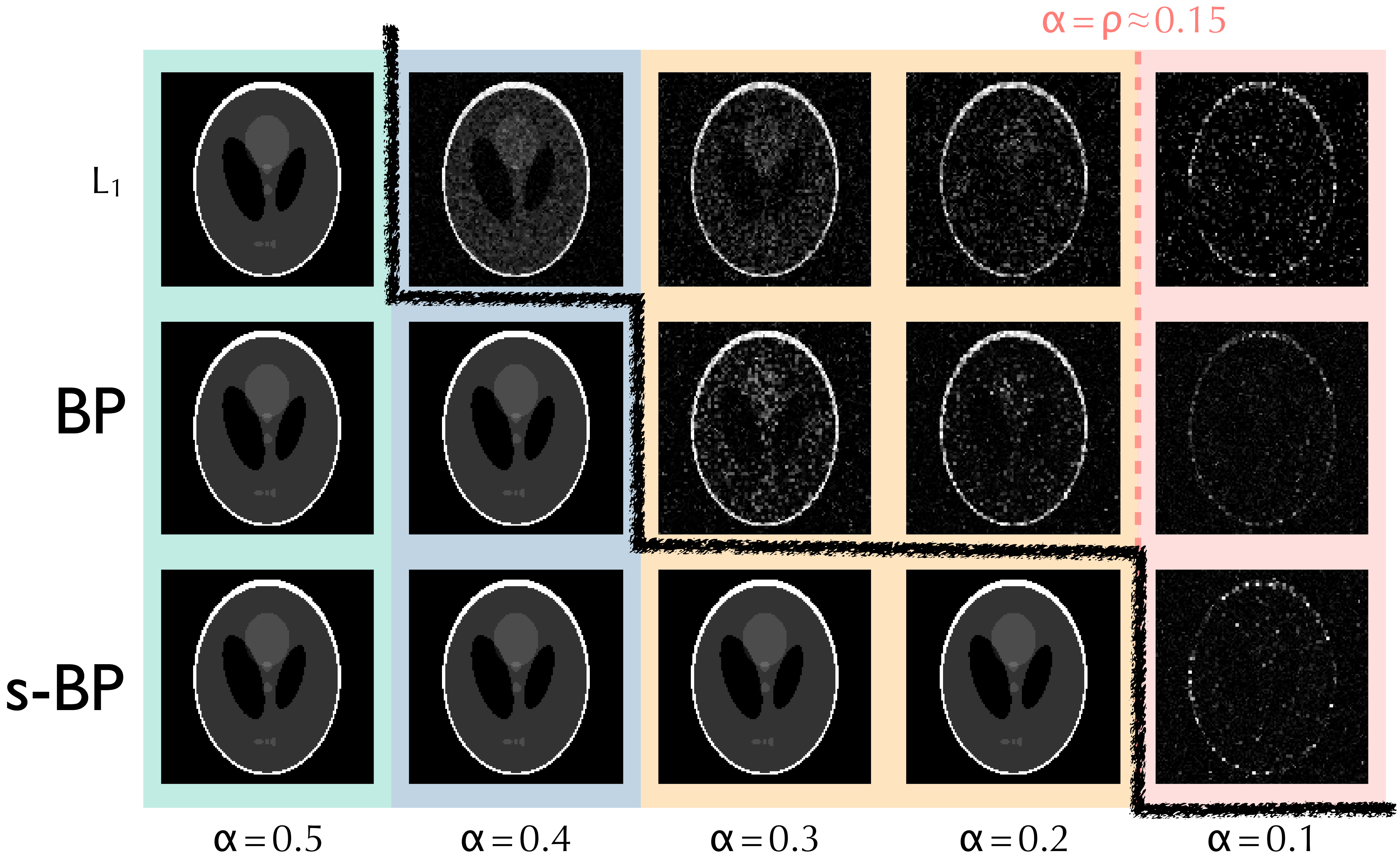
Phase Diagrams



Gauss-Bernoulli signal

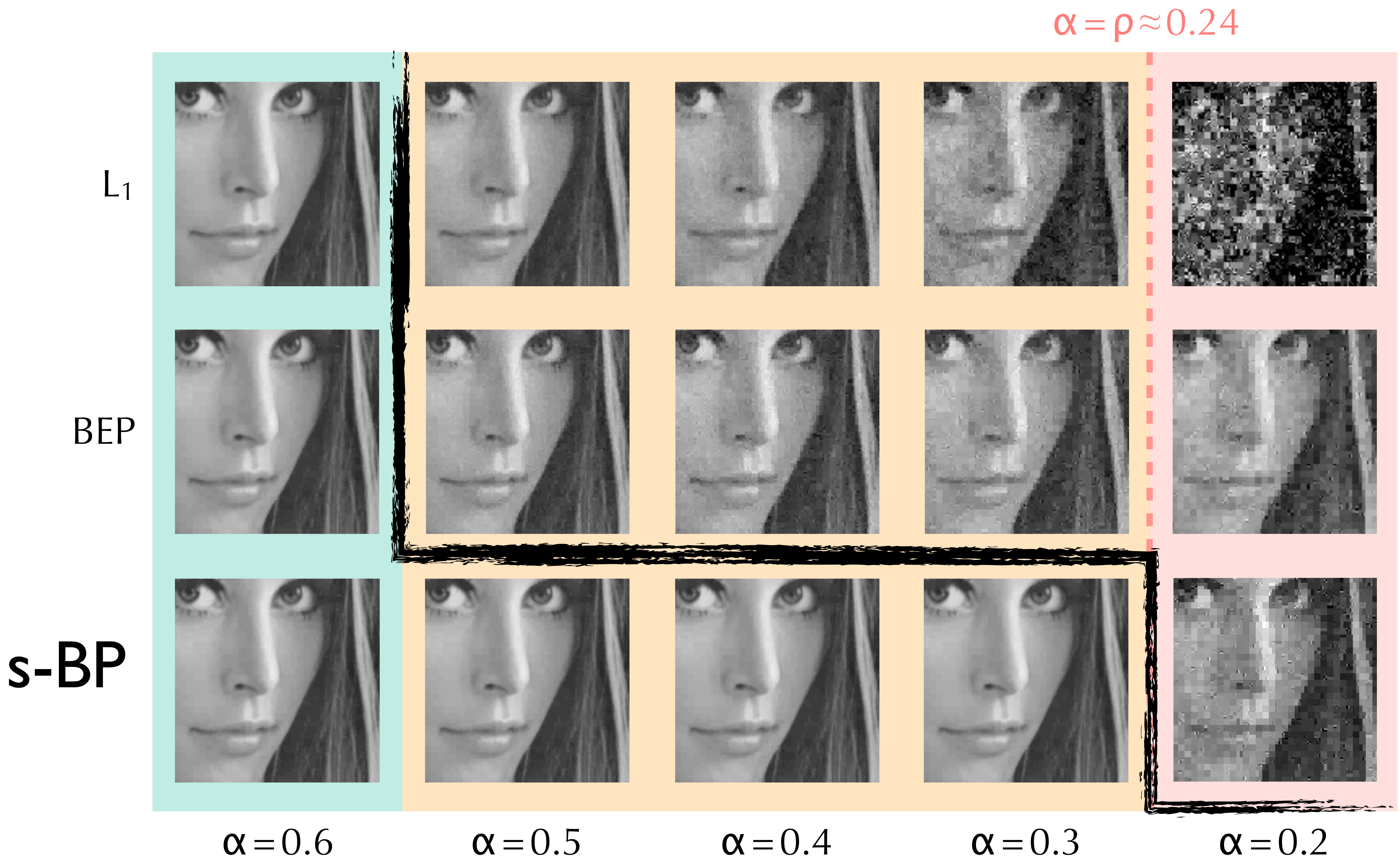
Binary signals

A more interesting example



Shepp-Logan phantom, in the Haar-wavelet representation

A EVEN more interesting example



The Lena picture in the Haar-wavelet representation

Analytical results for seeding matrices

- One can repeat the replica analysis for the seeded matrices, and the performance of the algorithm can be studied analytically, leading to $\alpha > \rho$ in the large N limit:

- These results have been recently confirmed by a rigorous analysis by Donoho, Montanari and Javanmard **(arXiv:1112.0708)**

- There is a lot of liberty in the design of the seeded matrices.

Analytical results for seeding matrices

- One can repeat the replica analysis for the seeded matrices, and the performance of the algorithm can be studied analytically, leading to $\alpha>\rho$ in the large N limit:

- These results have been recently confirmed by a rigorous analysis by Donoho, Montanari and Javanmard **(arXiv:1112.0708)**

- There is a lot of liberty in the design of the seeded matrices.

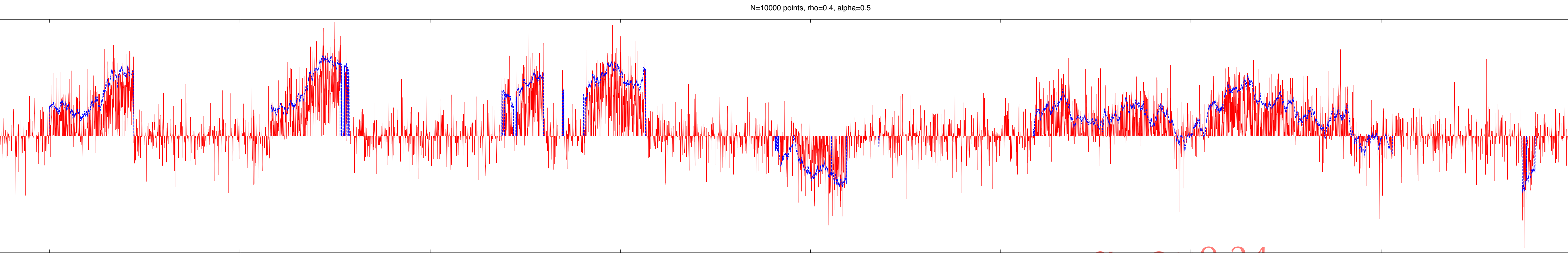
Asymptotically optimal measurements

Conclusions...

- A probabilistic approach to reconstruction
- The Belief Propagation algorithm
- Seeded measurements matrices

... and perspectives:

- More information in the prior?
- Other matrix with asymptotic measurements?
- Calibration noise, additive noise, quasi-sparsity, etc... ?
- Applications ?



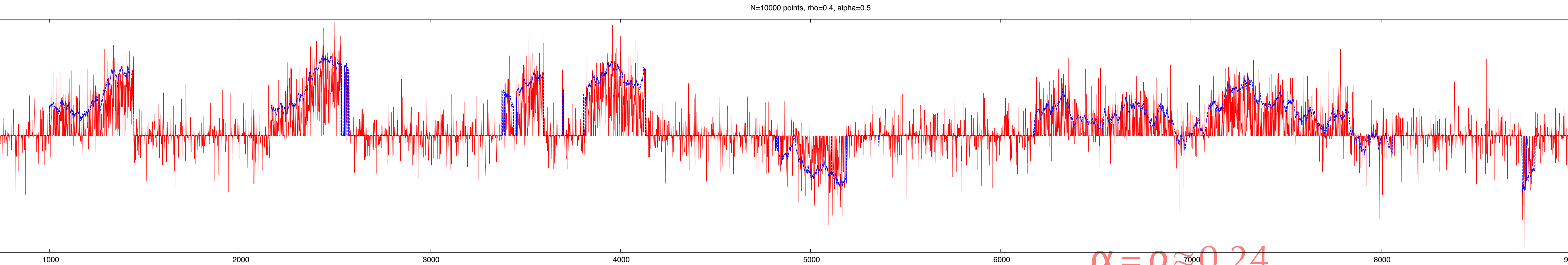
$\alpha = \rho \approx 0.24$

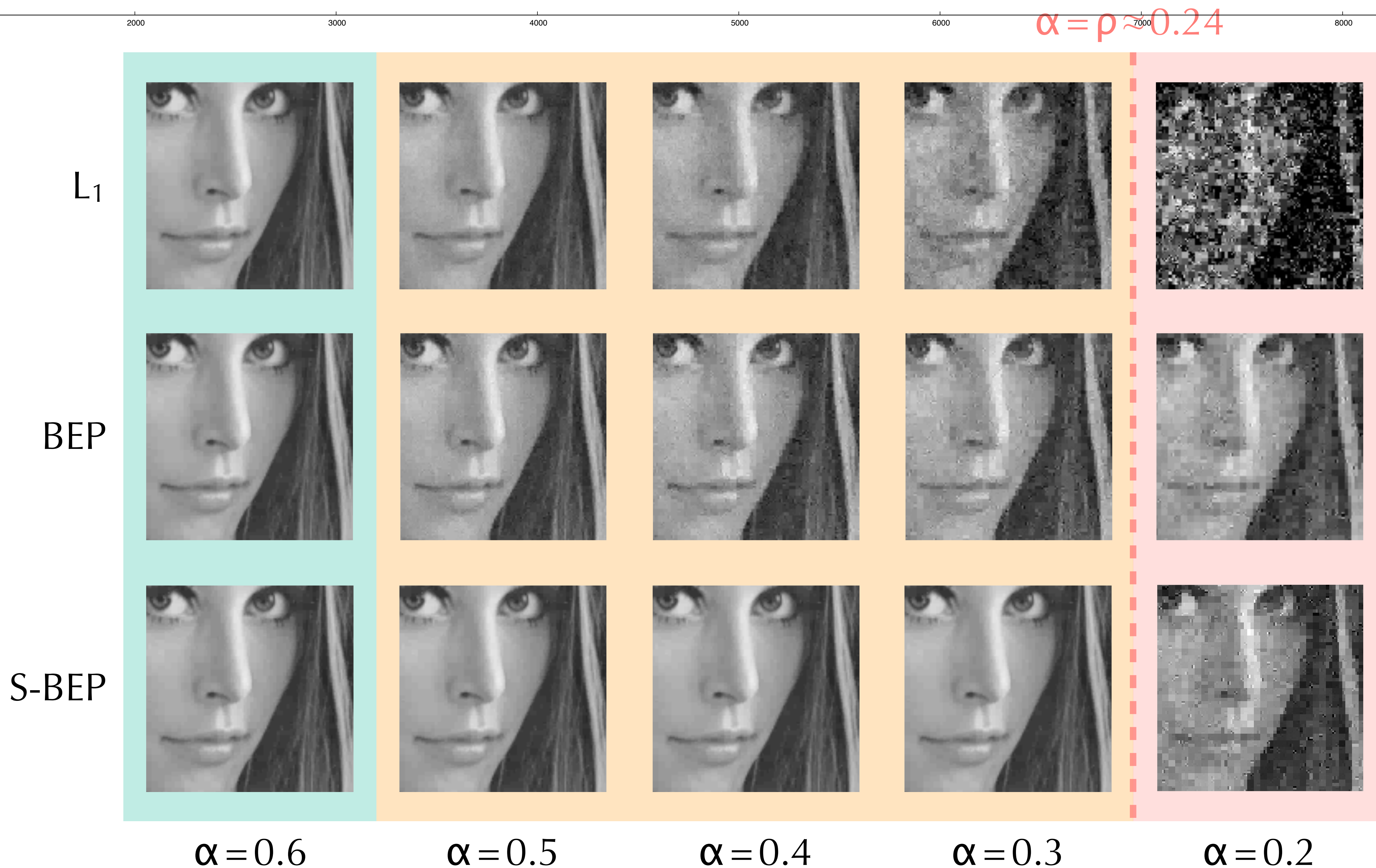
L_1

BEP

S-BEP

$\alpha = 0.6$ $\alpha = 0.5$ $\alpha = 0.4$ $\alpha = 0.3$ $\alpha = 0.2$

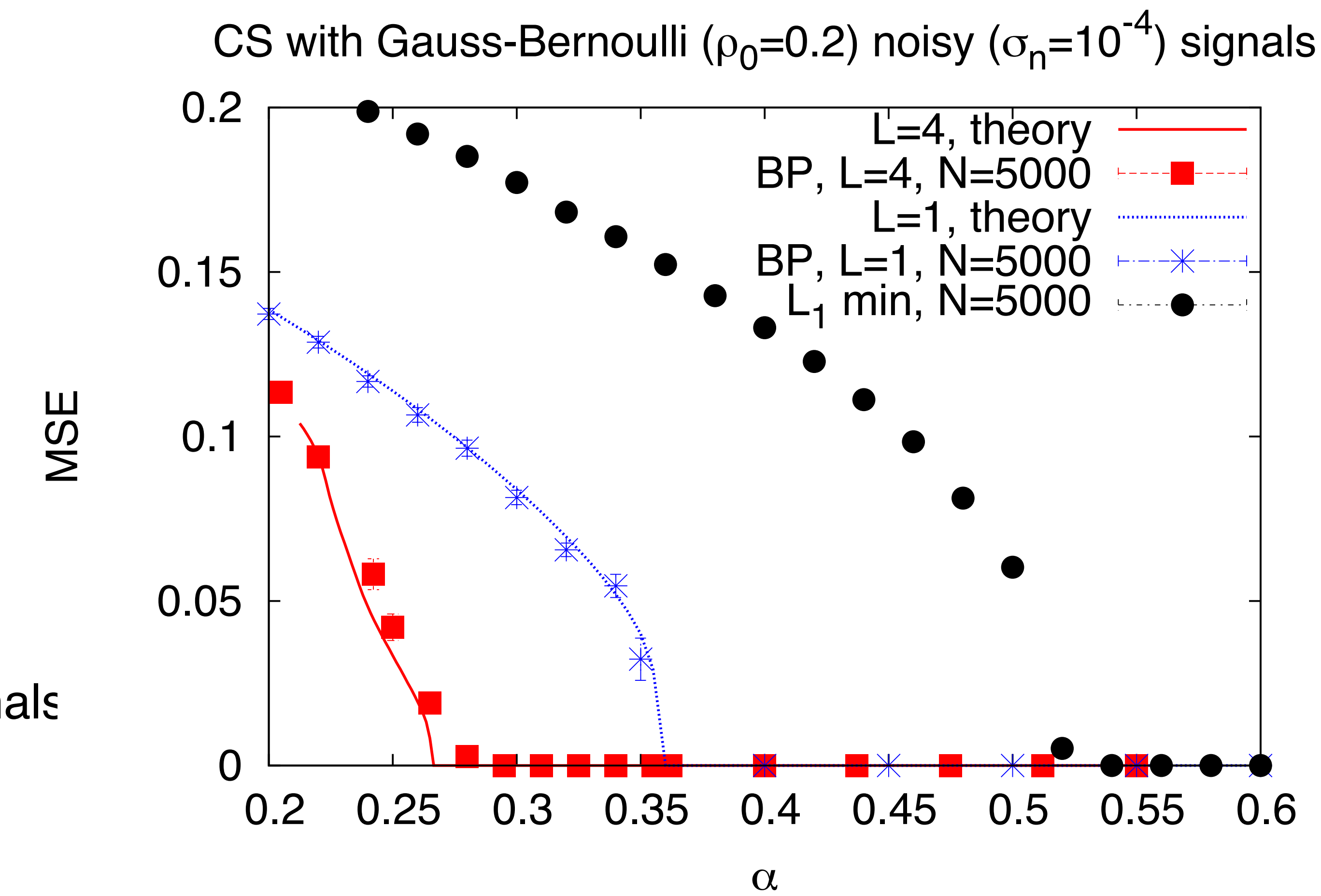


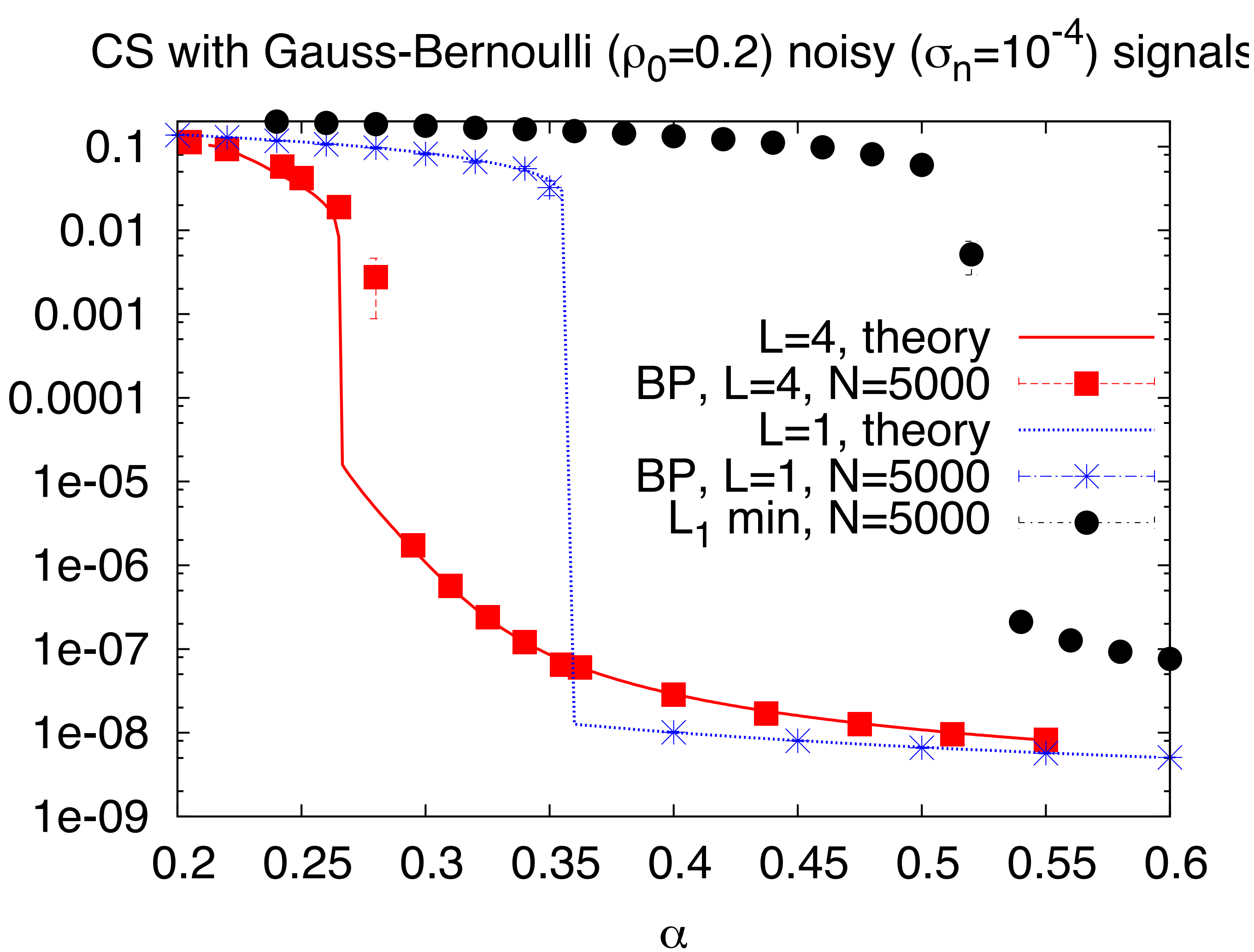


Thank you for your attention!

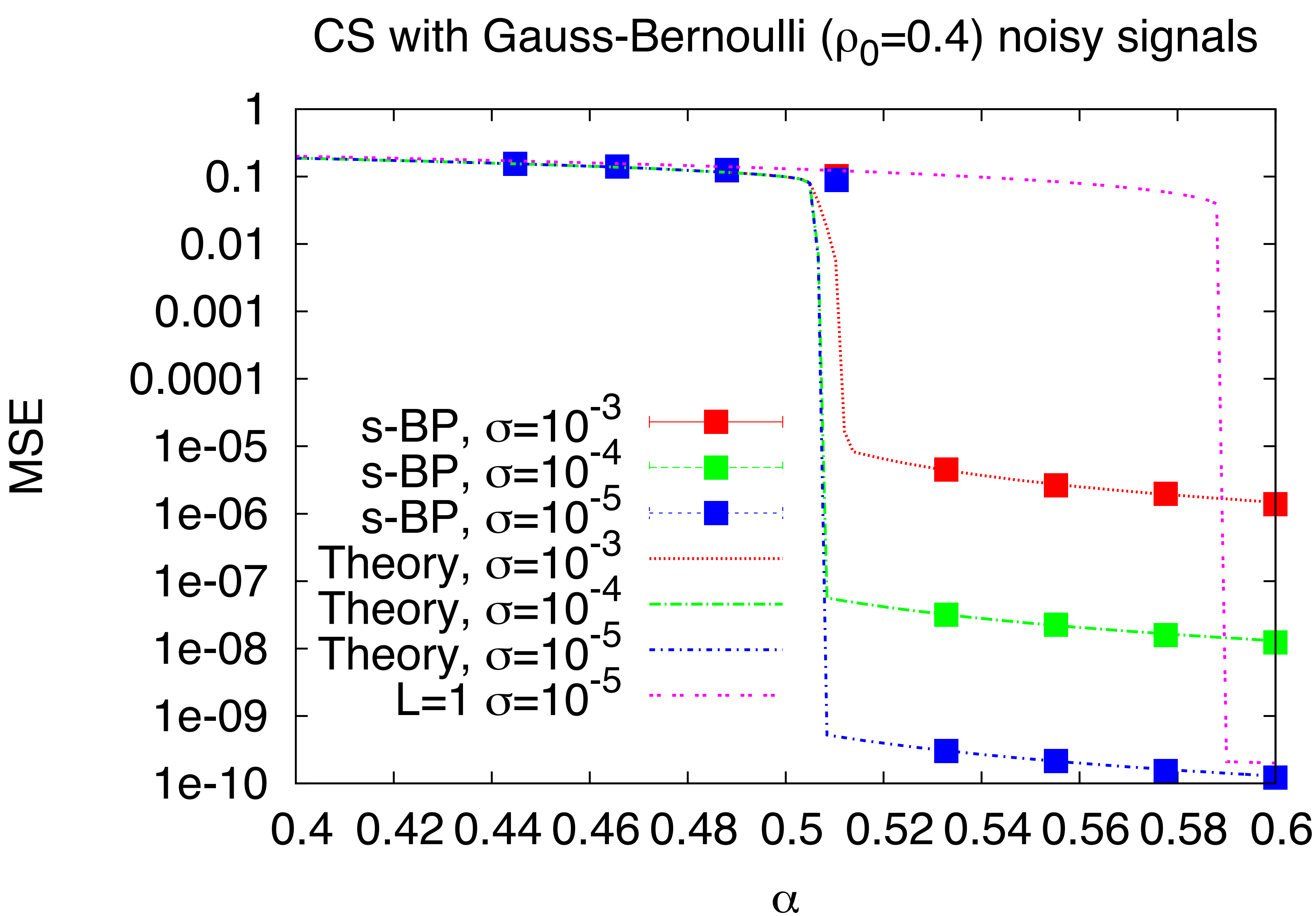
BONUS

Noise sensitivity

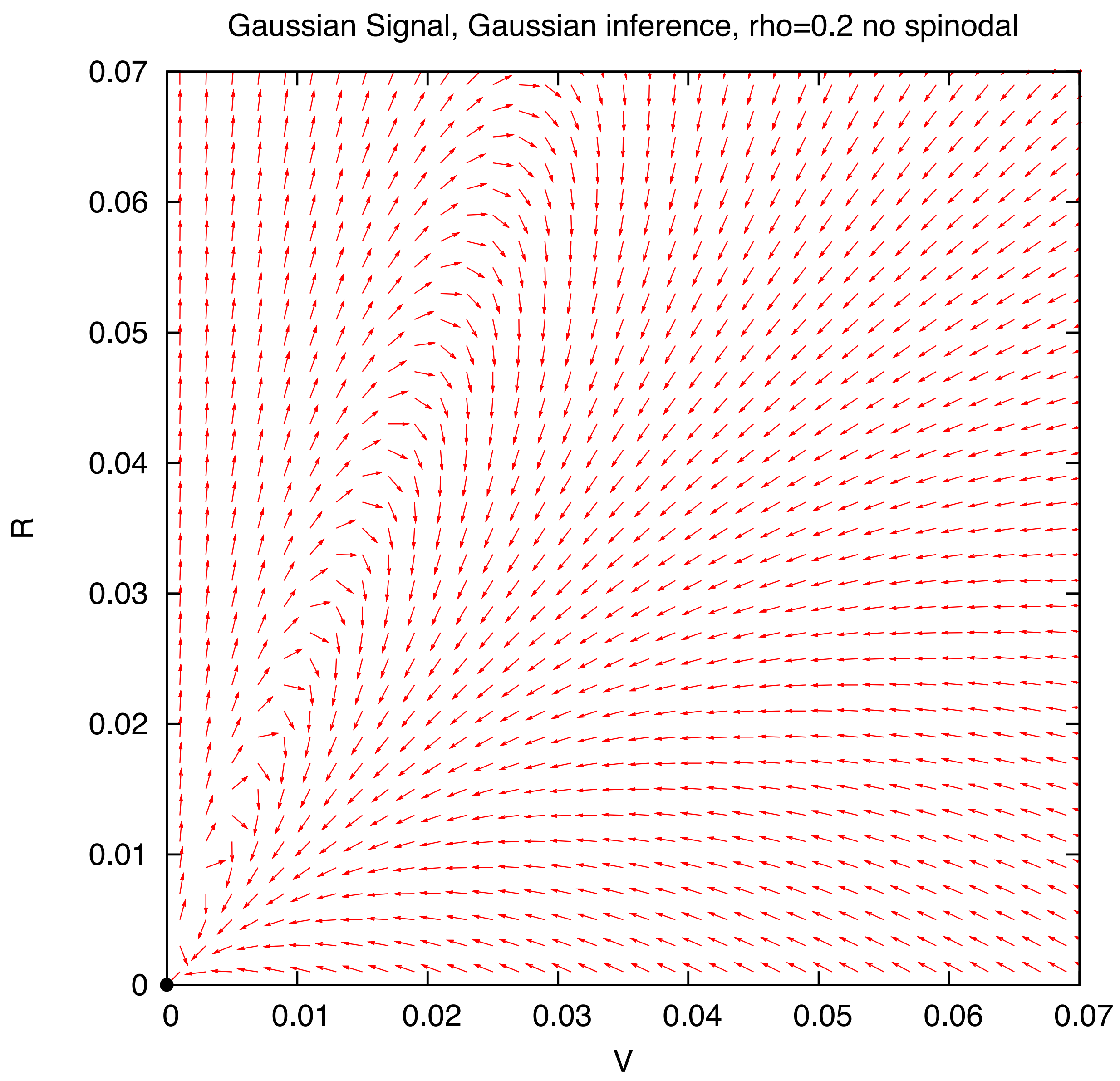


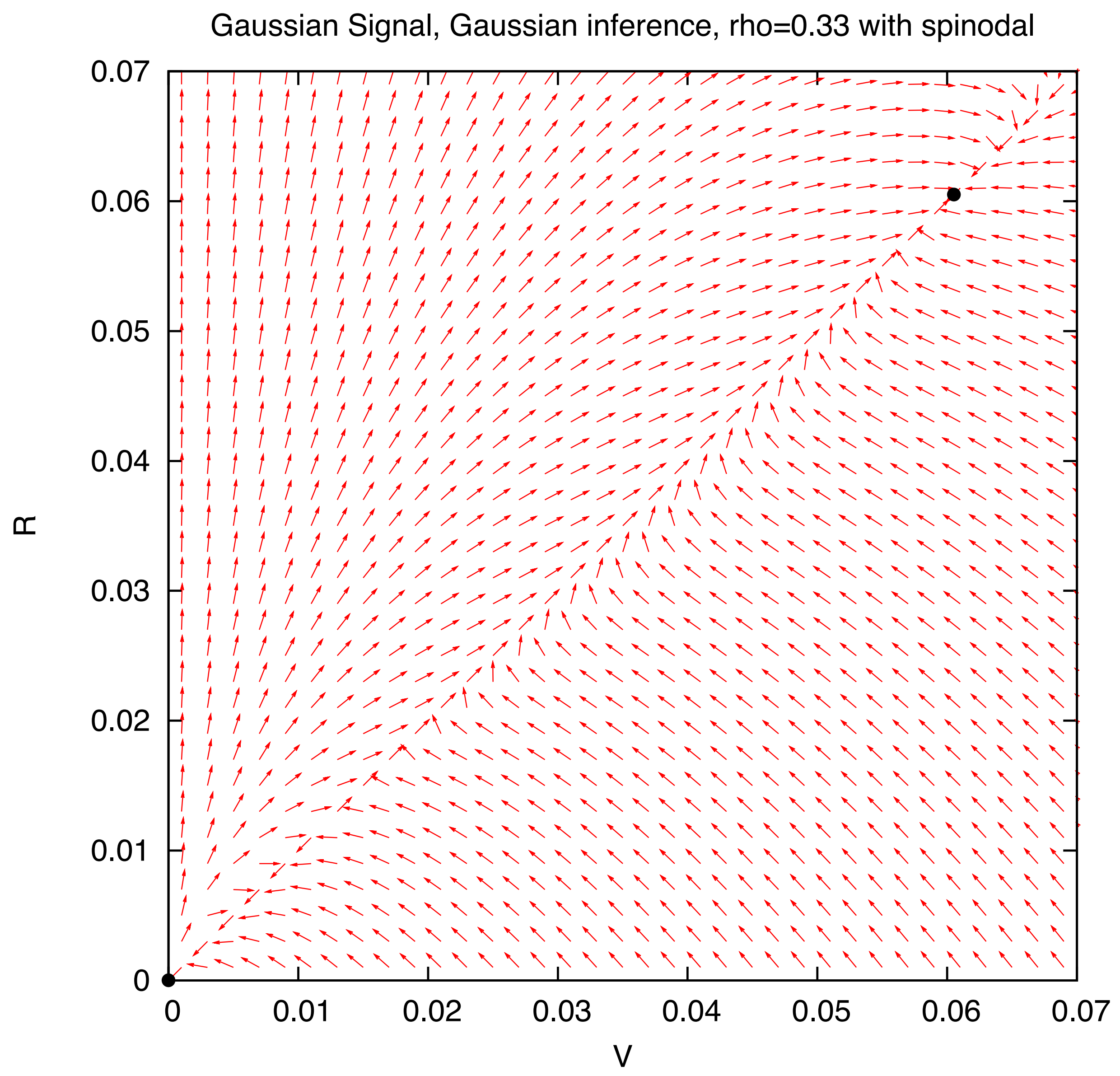


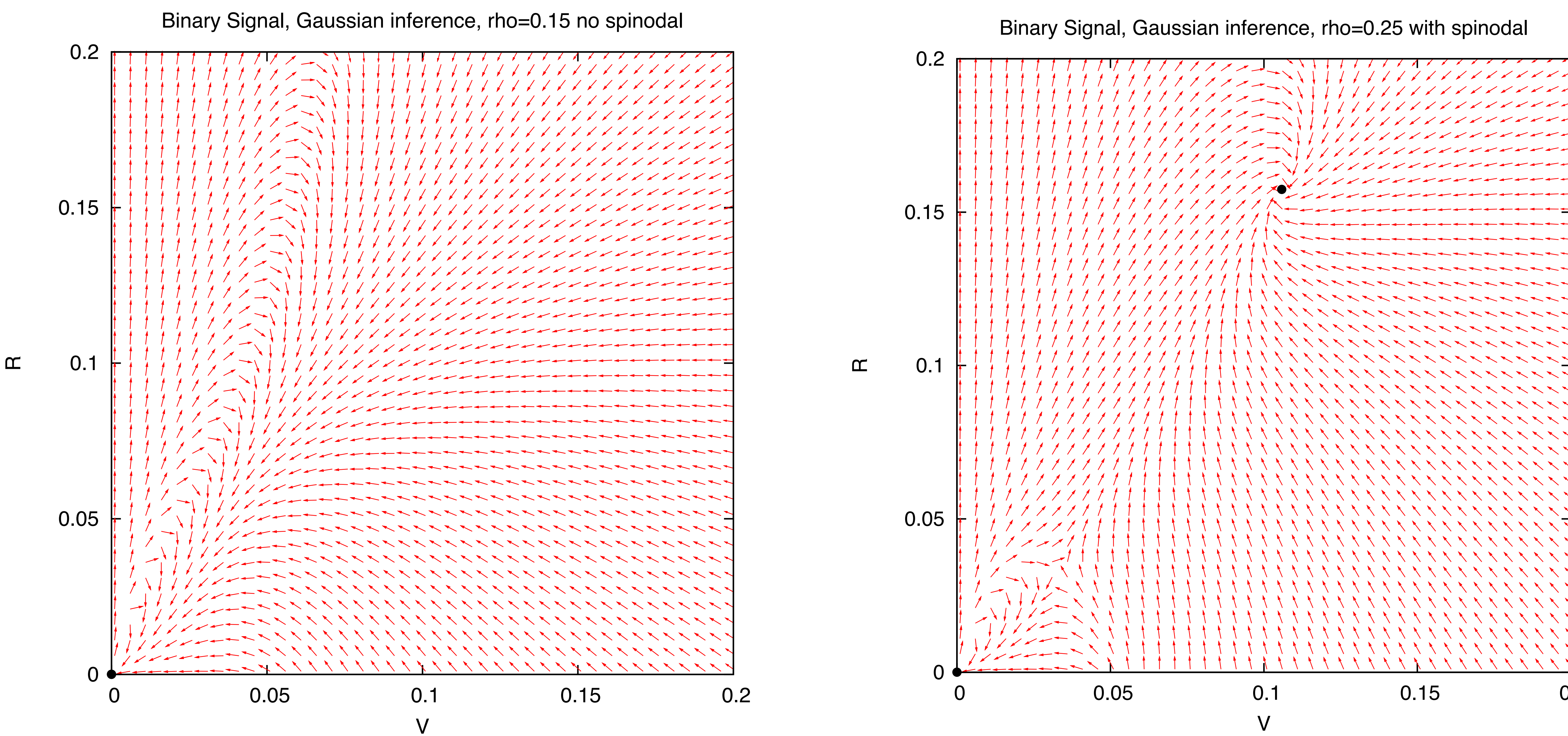
Noise sensitivity



Flow in the space $Q-q$, $E = q - 2m + \langle (x_i^0)^2 \rangle_0$






Binary Signal, Gaussian inference, rho=0.15 no spinodal
R
V
0.2
0.15
0.1
0.05
0
0
0.05
0.1
0.15
0.2
Binary Signal, Gaussian inference, rho=0.25 with spinodal
R
V
0.2
0.15
0.1
0.05
0
0
0.05
0.1
0.15