Sparse Superposition Codes for the Gaussian Channel

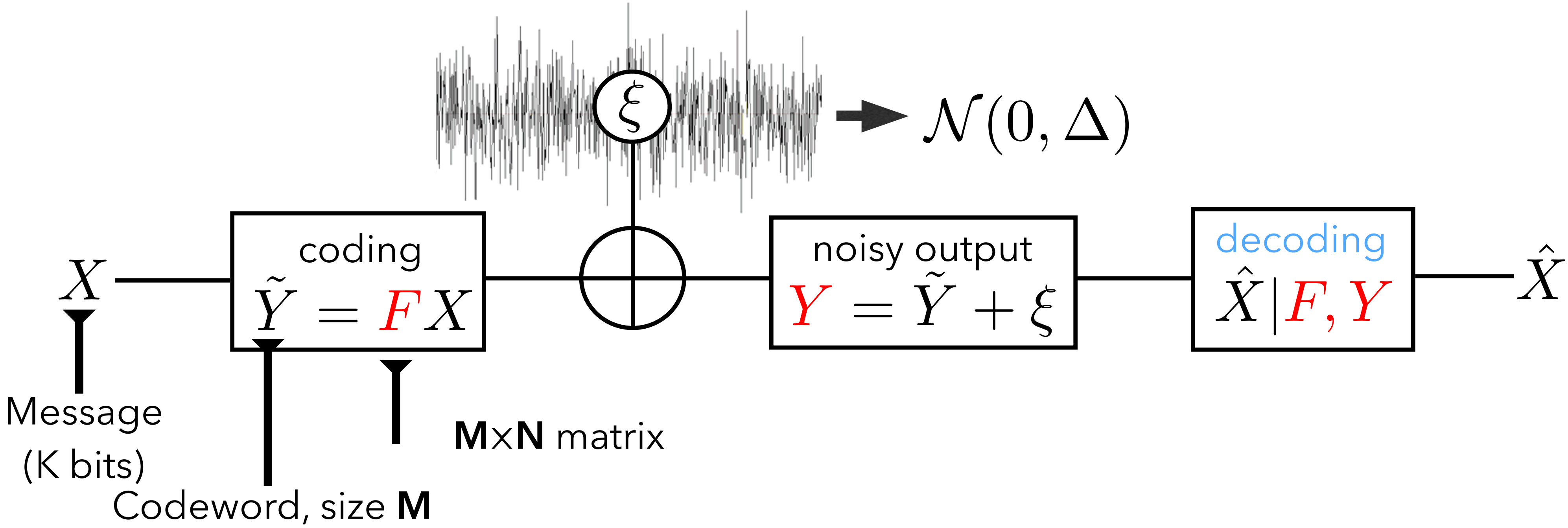
Florent Krzakala
(LPS, Ecole Normale Supérieure, France)

J. Barbier (ENS)

arXiv:1403.8024 presented at ISIT '14

Long version in preparation

Communication through the Gaussian Channel



Forney & Ungerboeck '98 Review
(modulation, shaping and coding)

Richardson & Urbanke '08
(LDPC and turbo codes)

Arikan '09, Abbe & Barron '11
(Polar codes)

Characteristics:

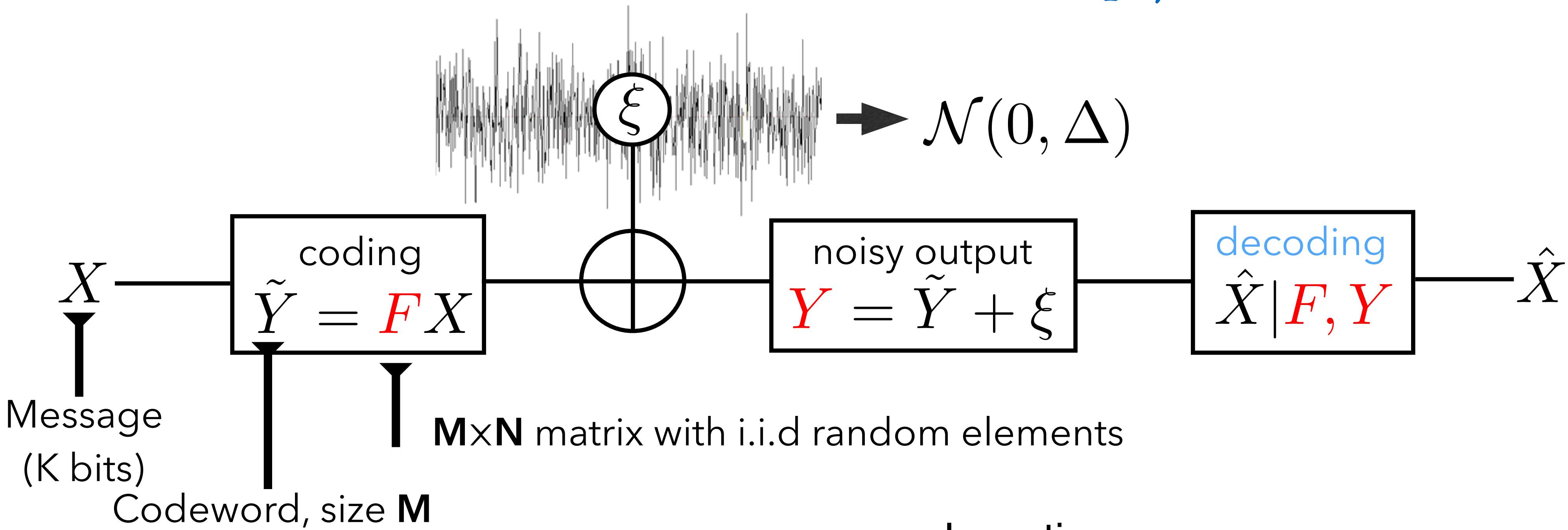
Power constraint: $\langle \tilde{Y}^2 \rangle = 1$

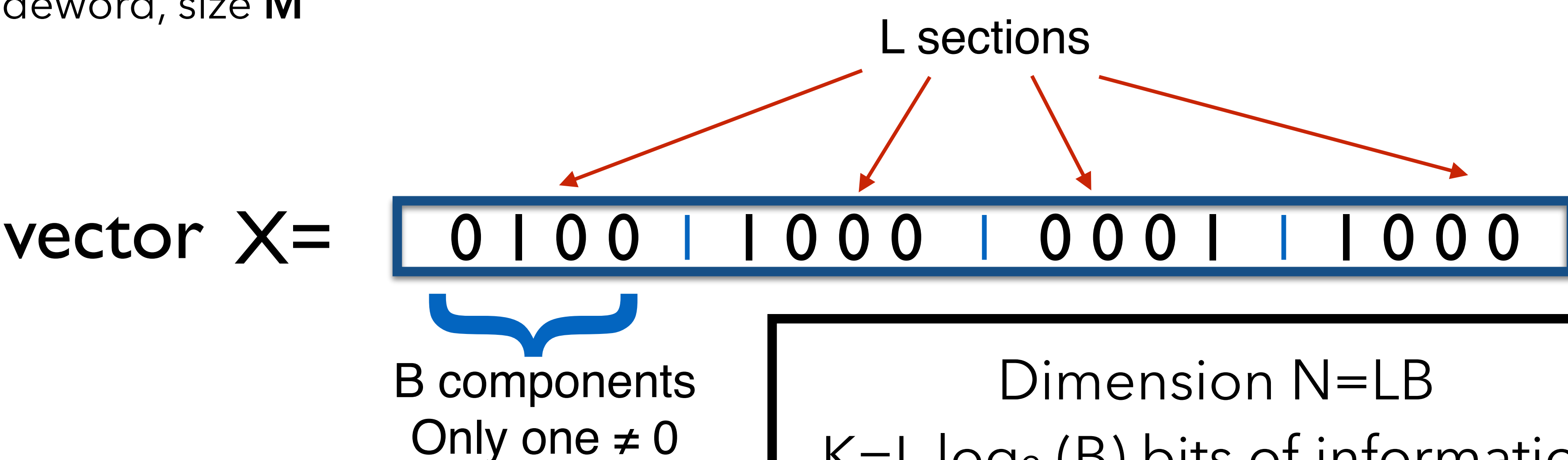
Signal to noise ratio: $\mathrm{SNR} = \frac{1}{\Delta}$

Shannon Capacity: $C = \frac{1}{2} \log_2 (1 + \mathrm{SNR})$

Sparse Superposition codes

Joseph, Barron ISIT '10





Dimension N=LB

K=L $\log_2$ (B) bits of information

Rate: $R = \dfrac{N \log_2(B)}{BM}$

Sparse Superposition codes

Joseph, Barron ISIT '10

- For large B, the maximum likelihood solution is capacity achieving

$\min_{\hat{x}} ||Y - F\hat{x}||_{\ell_2}$ such that $\hat{x}$ has a single 1 per section

Hard computational problem…

Sparse Superposition codes

Joseph, Barron ISIT '10

- For large B, the maximum likelihood solution is capacity achieving
- With proper power allocation, the "*adaptive successive decoder*" of Joseph and Barron achieve capacity when $B \rightarrow \infty$

vector X= | 0 1 0 0 | 1 0 0 0 | 0 0 0 1 | 1 0 0 0 |

Sparse Superposition codes

Joseph, Barron ISIT '10

- For large B, the maximum likelihood solution is capacity achieving
- With proper power allocation, the "*adaptive successive decoder*" of Joseph and Barron achieve capacity when B→∞

vector X= | 0 c_1 0 0 | c_2 0 0 0 | 0 0 0 c_3 | c_4 0 0 0 |

Power constraint: $\langle c_i^2 \rangle = 1$ $\quad c_i^2 \propto e^{-\frac{2Ci}{L}}$

Sparse Superposition codes

Joseph, Barron ISIT '10

- For large B, the maximum likelihood solution is capacity achieving
- With proper power allocation, the "*adaptive successive decoder*" of Joseph and Barron achieve capacity when B→∞

vector X= | 0 c_1 0 0 | c_2 0 0 0 | 0 0 0 c_3 | c_4 0 0 0 |

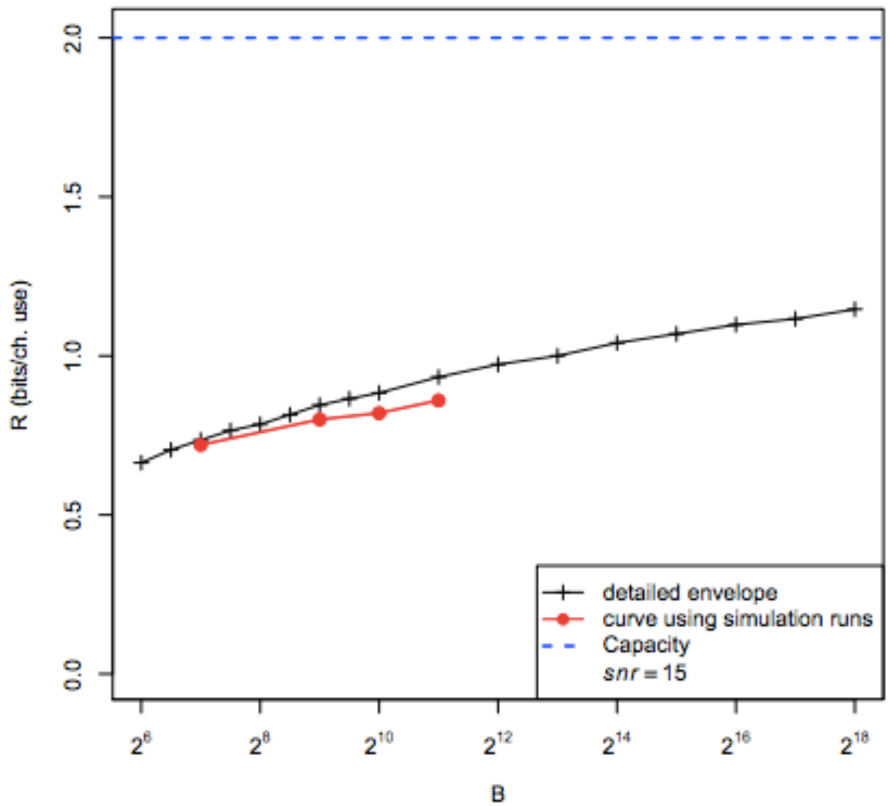
Power constraint: $\langle c_i^2 \rangle = 1$ $\qquad c_i^2 \propto e^{-\frac{2Ci}{L}}$

- In practice, however, results are **<u>FAR</u>** from capacity when B is finite

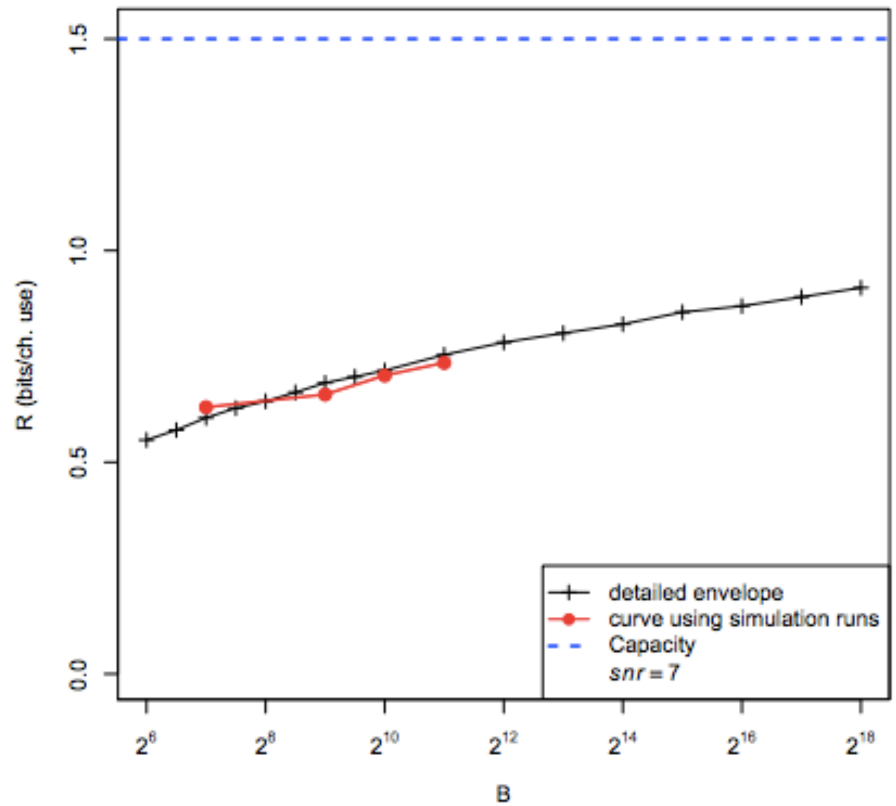
Sparse Superposition codes

Joseph, Barron ISIT '10

SNR=15



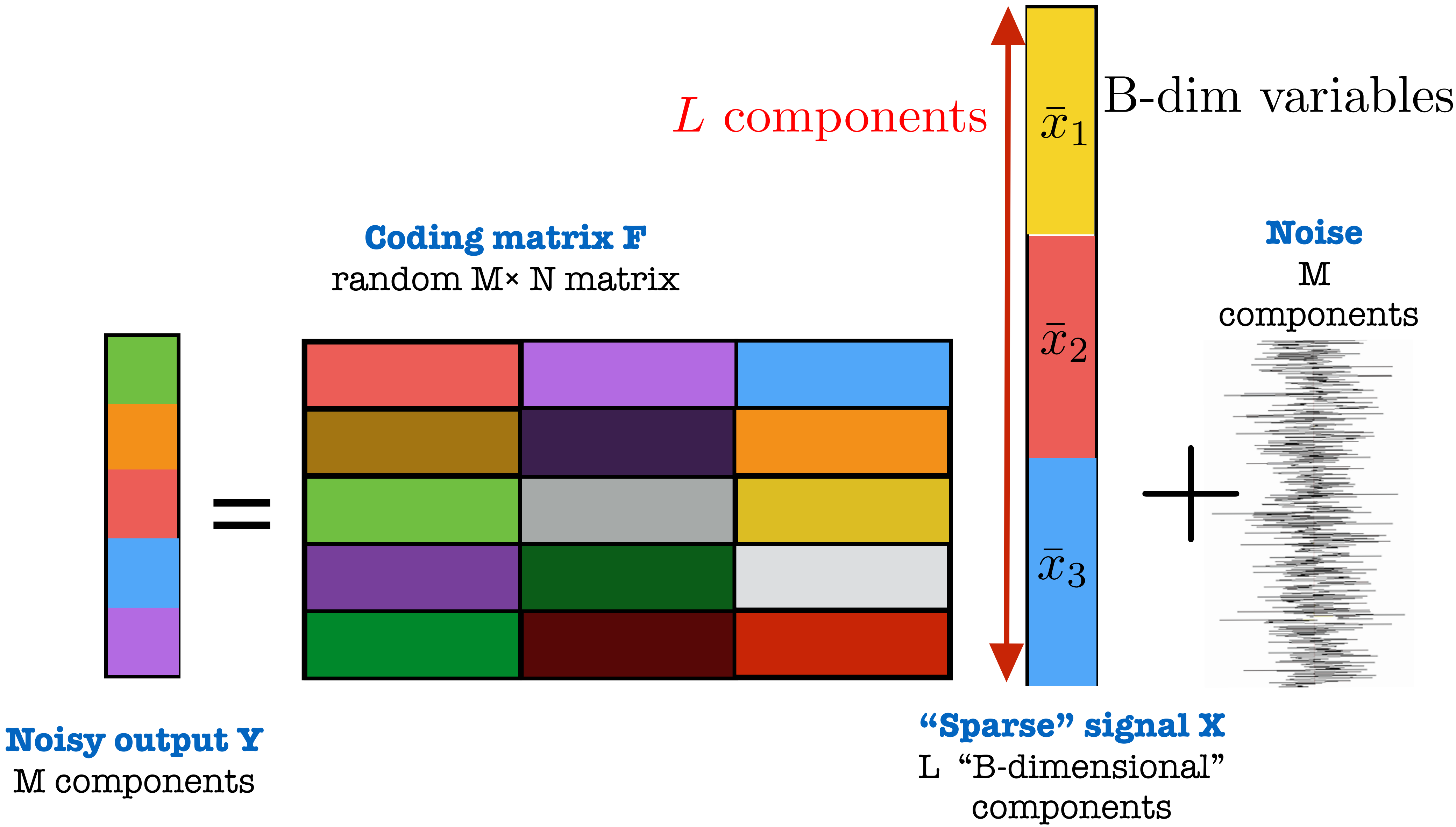
SNR=7



1

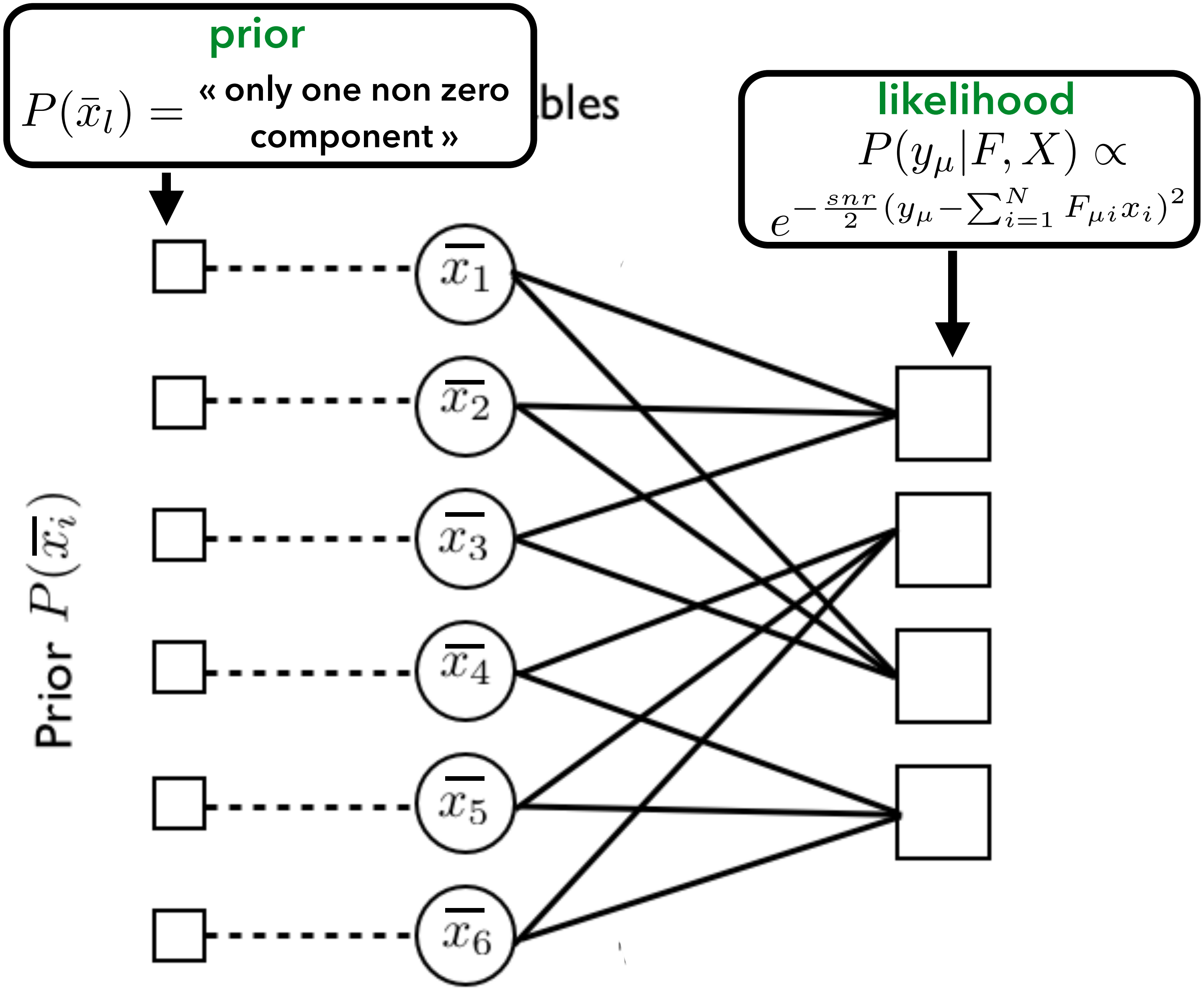
The approximate message-passing algorithm

Sparse Superposition codes =
Multi-dimensional estimation problem



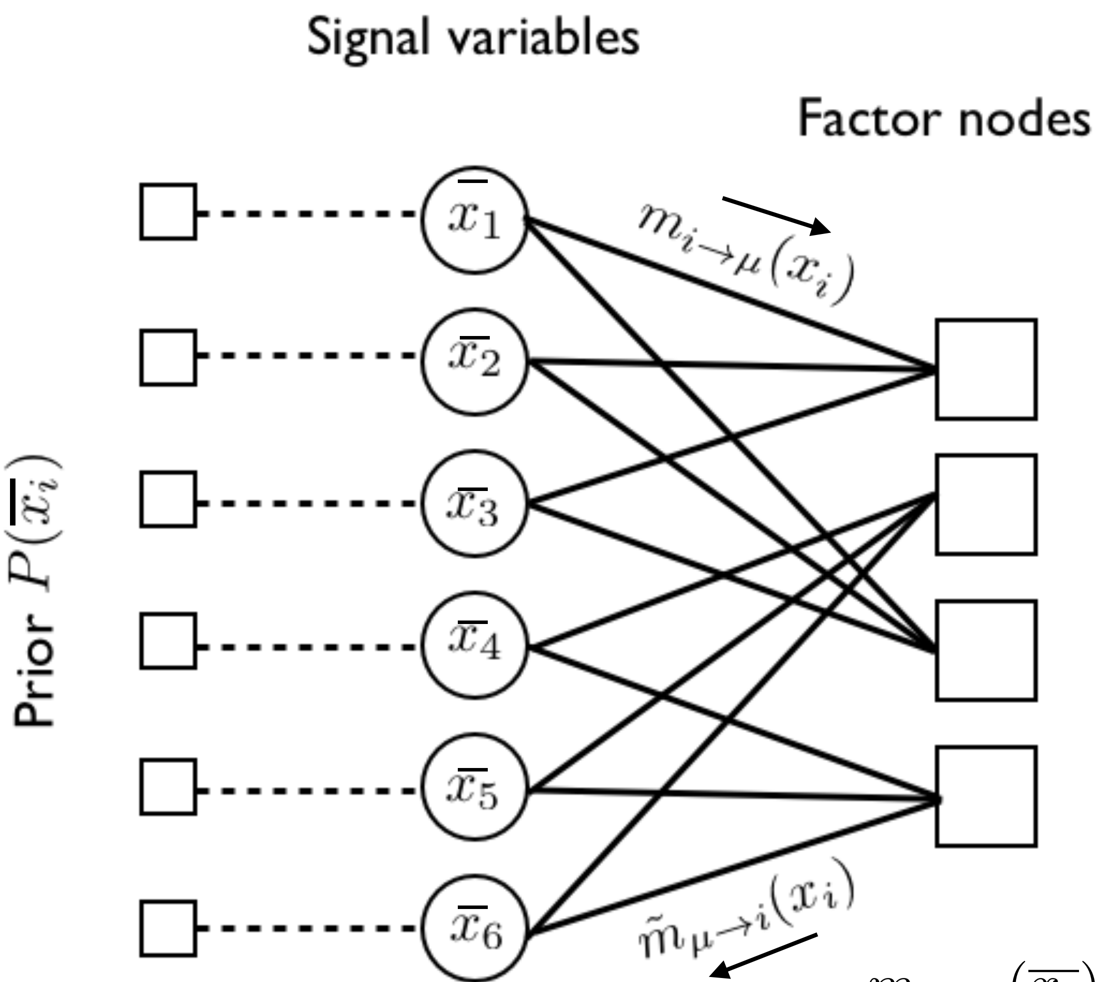
Graphical model

Compute the marginal of $P(X|F,Y) \propto P(X)P(Y|F,X)$



Graphical model

Compute the marginal of $P(X|F,Y) \propto P(X)P(Y|F,X)$



Belief Propagation?

Not really an option …

i) Densely connected model
ii) $O(L^2)$ messages
iii) Combinatoric factor $O(B^L)$
iv) Way too slow!

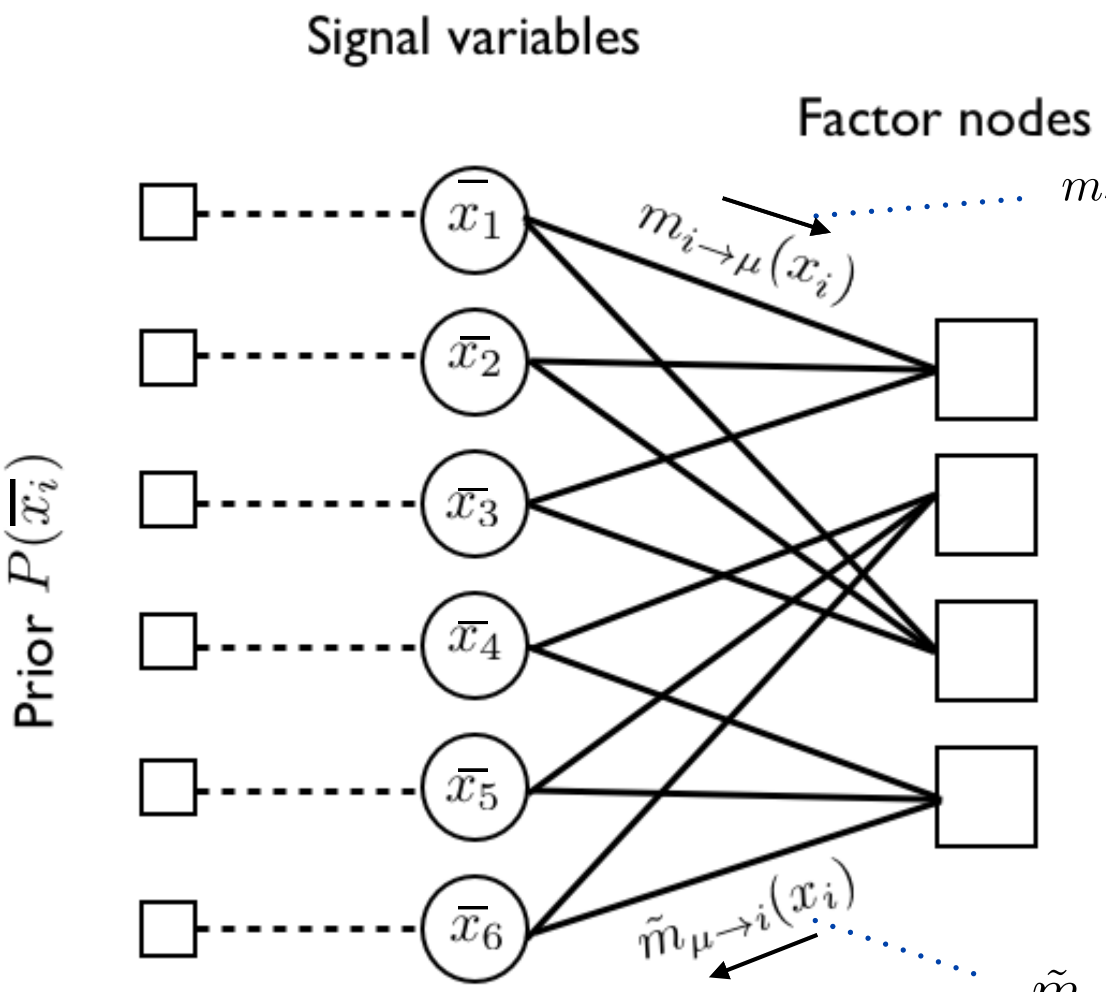
$$m_{i\to\mu}(\overline{x_i}) \propto P(\overline{x_i}) \prod_{\gamma\neq\mu} m_{\gamma\to i}(\overline{x_i})$$

$$m_{\mu\to i}(\overline{x_i}) \propto \sum_{\{x_j\}} \prod_{j\neq i} m_{j\to\mu}(\overline{x_i}) e^{-\frac{SNR}{2}\left(y_\mu - \sum_l F_{\mu l} x_l\right)^2}$$

From belief propagation to AMP

Compute the marginal of $P(X|F,Y) \propto P(X)P(Y|F,X)$

(i) A Gaussian relaxation (aka Gaussian-BP)



$$m_{i\to\mu}(x_i) \propto P(x_i) e^{-\frac{x_i^2}{2}\sum_{\gamma\neq\mu} A_{\gamma\to i} + \sum_{\gamma\neq\mu} B_{\mu\to i} x_i}$$

A Gaussian-like relaxation

$$A_{\mu\to i} = \frac{F_{\mu i}^2}{\Delta_\mu + \sum_{j\neq i} F_{\mu j}^2 v_{j\to\mu}},$$

$$B_{\mu\to i} = \frac{F_{\mu i}(y_\mu - \sum_{j\neq i} F_{\mu j} a_{j\to\mu})}{\Delta_\mu + \sum_{j\neq i} F_{\mu j}^2 v_{j\to\mu}}$$

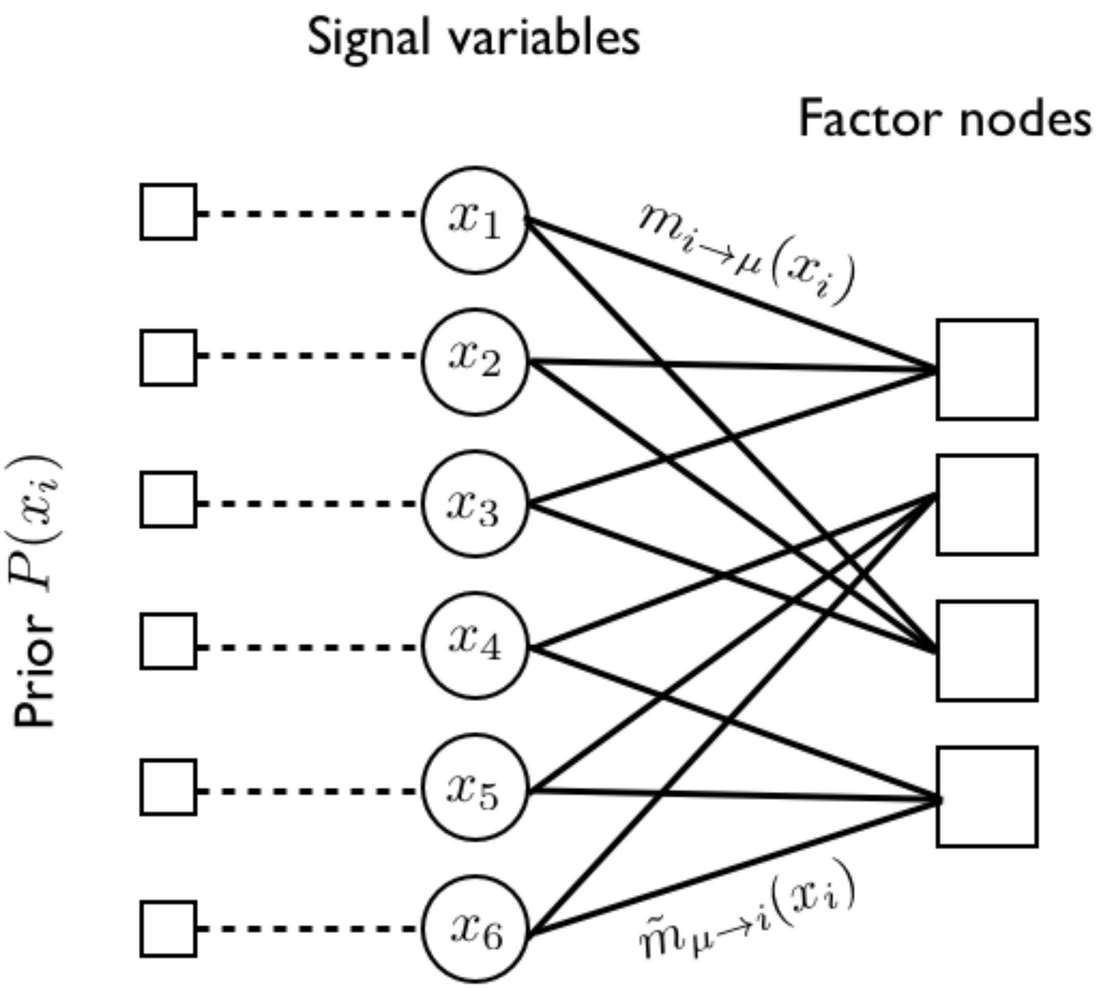
$$\tilde{m}_{\mu\to i}(x_i) \propto e^{-\frac{x_i^2}{2} A_{\mu\to i} + B_{\mu\to i} x_i}$$

From belief propagation to AMP

Compute the marginal of $P(X|F,Y) \propto P(X)P(Y|F,X)$

(i) A Gaussian relaxation (aka Gaussian-BP)

(ii) Write the recursion in terms of single marginals (equivalent to "TAP" in Stat. Phys.)



$$m_{i\to\mu}(x_i) \approx m_i(x_i) + \text{correction}$$

Closing the equations on
two moments for each variables:
2N variables (AMP)
vs
N^2 variables (BP)

From belief propagation to AMP

Compute the marginal of $P(X|F,Y) \propto P(X)P(Y|F,X)$

$$
\begin{aligned}
V_\mu^{t+1} &= \sum_i F_{\mu i}^2 v_i^t \\
\omega_\mu^{t+1} &= \sum_i F_{\mu i} a_i^t - (y_\mu - \omega_\mu^t) \frac{V_\mu^{t+1}}{1/\mathrm{snr} + V_\mu^t} \\
(\Sigma_i^{t+1})^2 &= \left[\sum_\mu \frac{F_{\mu i}^2}{1/\mathrm{snr} + V_\mu^{t+1}} \right]^{-1} \\
R_i^{t+1} &= a_i^t + (\Sigma_i^{t+1})^2 \sum_\mu F_{\mu i} \frac{(y_\mu - \omega_\mu^{t+1})}{1/\mathrm{snr} + V_\mu^{t+1}} \\
a_i^{t+1} &= f_{a_i}\left(\{\Sigma_j^{t+1}, R_j^{t+1}\}_{j \in l}\right) \\
v_i^{t+1} &= f_{c_i}\left(\{\Sigma_j^{t+1}, R_j^{t+1}\}_{j \in l}\right)
\end{aligned}
$$

$$
a_i^t := f_{a_i}(\{\Sigma_j^t, R_j^t\}_{j \in l}) = \frac{e^{-\frac{1-2R_i^t}{2(\Sigma_i^t)^2}}}{\sum_{\{j \in l\}}^B e^{-\frac{1-2R_j^t}{2(\Sigma_j^t)^2}}}
$$

$$
v_i^t := f_{c_i}(\{\Sigma_j^t, R_j^t\}_{j \in l}) = a_i^t(1 - a_i^t)
$$

Matlab implementation + demo: https://github.com/jeanbarbier/BPCS_common

From belief propagation to AMP

Compute the marginal of $P(X|F,Y) \propto P(X)P(Y|F,X)$

$$V_\mu^{t+1} = \sum_i F_{\mu i}^2 v_i^t$$
$$\omega_\mu^{t+1} = \sum_i F_{\mu i} a_i^t - (y_\mu - \omega_\mu^t)\frac{V_\mu^{t+1}}{1/\mathrm{snr} + V_\mu^t}$$
$$(\Sigma_i^{t+1})^2 = \left[\sum_\mu \frac{F_{\mu i}^2}{1/\mathrm{snr} + V_\mu^{t+1}}\right]^{-1}$$
$$R_i^{t+1} = a_i^t + (\Sigma_i^{t+1})^2 \sum_\mu F_{\mu i}\frac{(y_\mu - \omega_\mu^{t+1})}{1/\mathrm{snr} + V_\mu^{t+1}}$$
$$a_i^{t+1} = f_{a_i}\left(\{\Sigma_j^{t+1}, R_j^{t+1}\}_{j\in l}\right)$$
$$v_i^{t+1} = f_{c_i}\left(\{\Sigma_j^{t+1}, R_j^{t+1}\}_{j\in l}\right)$$

A (perhaps) more familiar form from monday's talk

(Donoho, Montanari, Maleki '09)

$$a_{t+1} = f_a^t(F' z_t + a_t)$$
$$z_t = y - F a_t + \frac{1}{\alpha} < \dot{f}_a^{t+1}(F' z^{t-1} + a_{t-1}) > z^{t-1}$$

$$a_i^t := f_{a_i}(\{\Sigma_j^t, R_j^t\}_{j\in l}) = \frac{e^{-\frac{1-2R_i^t}{2(\Sigma_i^t)^2}}}{\sum_{\{j\in l\}}^B e^{-\frac{1-2R_j^t}{2(\Sigma_j^t)^2}}}$$
$$v_i^t := f_{c_i}(\{\Sigma_j^t, R_j^t\}_{j\in l}) = a_i^t(1 - a_i^t)$$

Complexity is N^2 for iid matrix
Nlog(N) for Hadamard/Fourier

Matlab implementation + demo: https://github.com/jeanbarbier/BPCS_common

2

Asymptotic analysis

A heuristic computation of the optimal SER

Bayes-optimal estimate:
Marginalize w.r.t.

$$P(X|Y,F) = \frac{P(X)}{Z} \prod_{\mu=0}^{M} e^{-\frac{\mathrm{SNR}}{2}\left(y_\mu - \sum_{i=1}^{N} F_{\mu i}\hat{x}_i\right)^2}$$

The **replica method** : Asymptotic estimates when N→∞
(statistical physics, information theory, optimization, etc..)

In CDMA
- Tanaka '02
- Guo, Verdu '06

In Compressed sensing
- Rangan et al. '09
- Kabashima '09
- Guo Baron Shamai '09
- Krzakala et al. '11

A heuristic computation of the optimal SER

Bayes-optimal estimate:
Marginalize w.r.t.

$$P(X|Y,F) = \frac{P(X)}{Z}\prod_{\mu=0}^{M} e^{-\frac{\mathrm{SNR}}{2}\left(y_\mu - \sum_{i=1}^{N} F_{\mu i}\hat{x}_i\right)^2}$$

The **replica method** : Asymptotic estimates when N→∞
(statistical physics, information theory, optimization, etc..)

1) Assume that log(Z) is concentrated

NON RIGOROUS

2) Use the following identity:

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

NON RIGOROUS

3) After (a bit) of work:

$$\overline{\log Z} \propto \int dE e^{N\Phi(E)} \quad \Rightarrow \quad MMSE = \max_E \Phi(E)$$

4) From the MSE E, one can compute the average SER

Single letter characterization of the SER

B-dimensional integral

$$\Phi_B(E) = -\frac{\log_2(B)}{2R}\left(\log(1/\mathrm{snr}+E) + \frac{1-E}{1/\mathrm{snr}+E}\right) + \underbrace{\int \mathcal{D}\overline{z}\log\left(e^{\frac{1}{2\Sigma(E)^2}+\frac{z_1}{\Sigma(E)}} + \sum_{i=2}^{B} e^{-\frac{1}{2\Sigma(E)^2}+\frac{z_i}{\Sigma(E)}}\right)}$$

with $\Sigma^t = \sqrt{\left(\frac{1}{\mathrm{snr}}+E\right)\frac{R}{\log_2 B}}$ and $\mathcal{D}\overline{z} = \prod dz_1 \frac{e^{-\frac{z_1^2}{2}}}{\sqrt{2\pi}} \dots dz_B \frac{e^{-\frac{z_B^2}{2}}}{\sqrt{2\pi}}$

The SER can be computed from E as

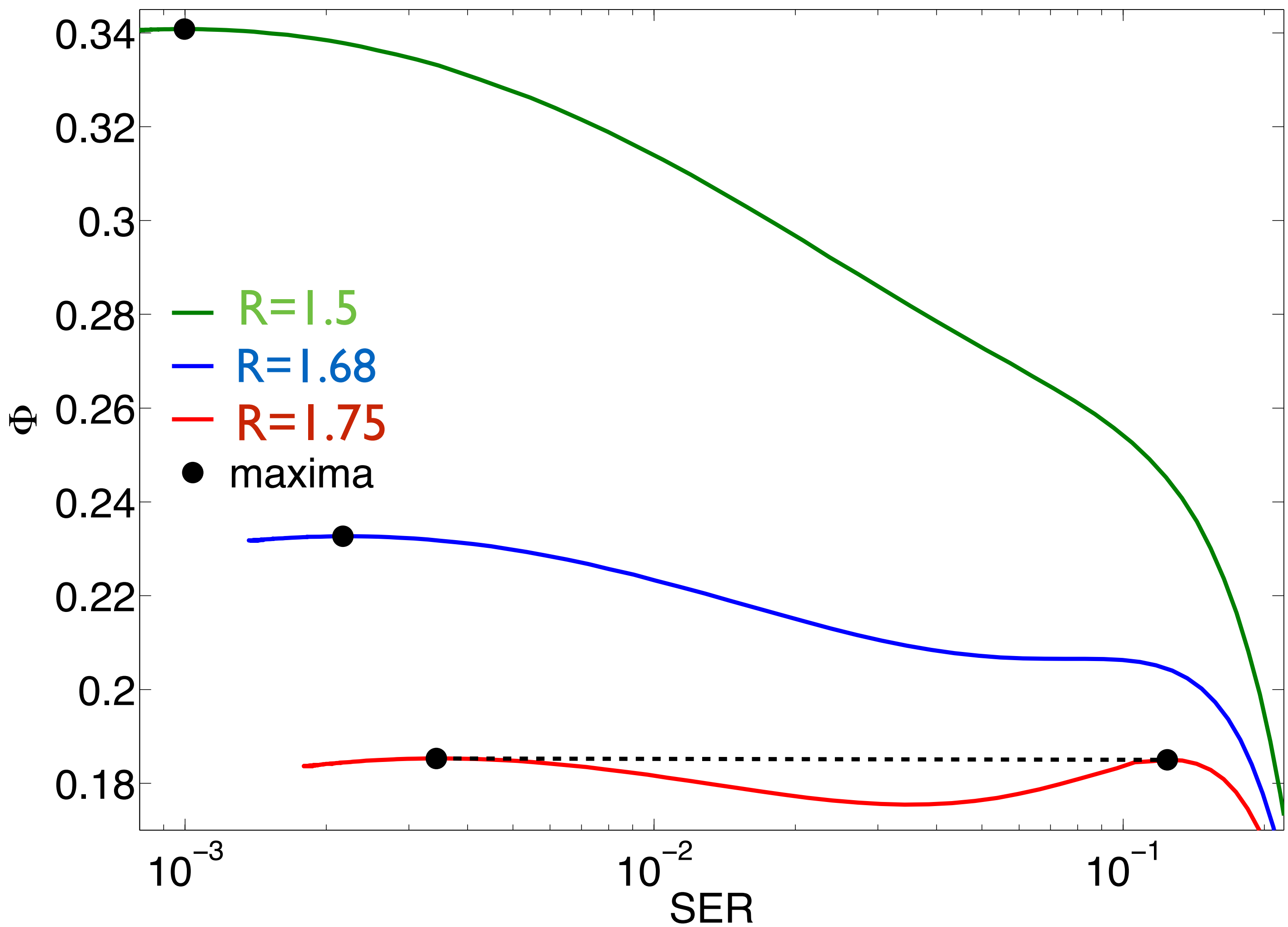
$$\mathrm{SER}^t = \int \mathcal{D}\overline{z}\ \mathbb{I}\left(\exists\ j \in \{2,\dots,B\} : f^{(0)}_{a_{j,1}}(\Sigma^t,\overline{z}) > f^{(1)}_{a_1}(\Sigma^t,\overline{z})\right)$$

with

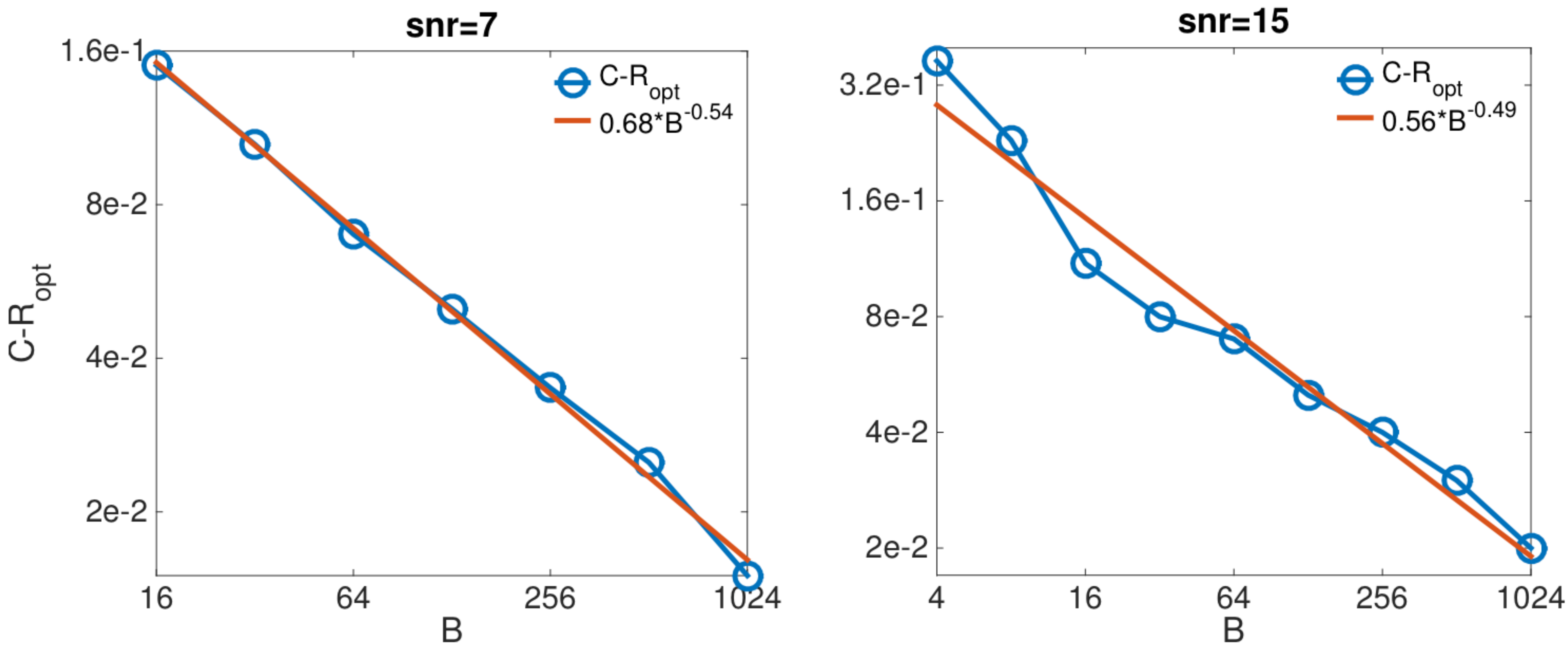
$$\begin{cases} f^{(1)}_{a_i}(\Sigma,\overline{z}) = \left[1 + e^{-\frac{1}{\Sigma^2}} \sum_{\{1\le j\le B: j\neq i\}} e^{\frac{z_j - z_i}{\Sigma}}\right]^{-1} \\ f^{(0)}_{a_{i,j}}(\Sigma,\overline{z}) = \left[1 + e^{\frac{1}{\Sigma^2}+\frac{z_j-z_i}{\Sigma}} + \sum_{\{1\le k\le B: k\neq i,j\}} e^{\frac{z_k-z_i}{\Sigma}}\right]^{-1} \end{cases}$$

Single letter characterization of the SER

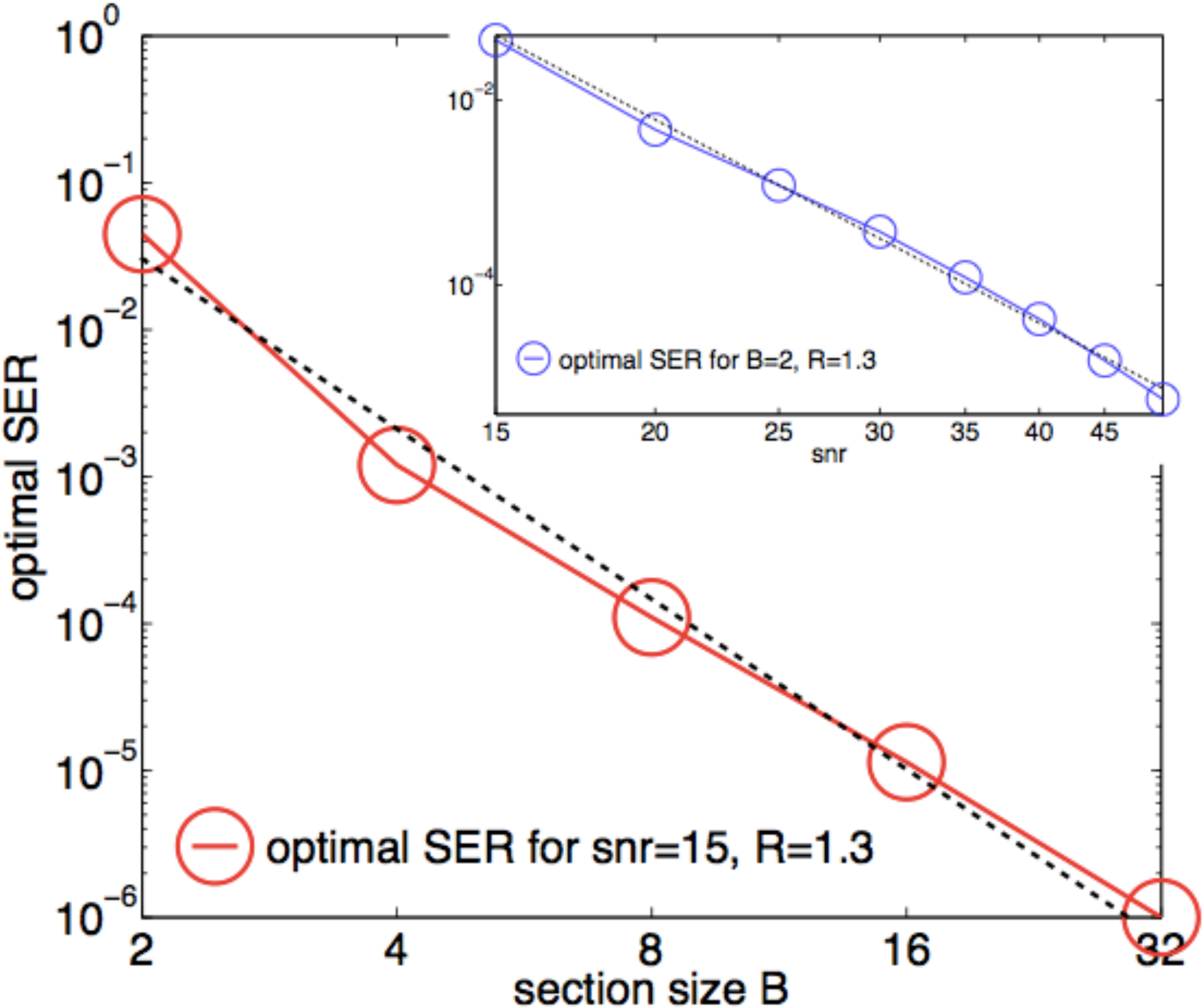
Here B=2, SNR=15 , C=2



Gap to capacity closing polynomially



Error floor decay polynomially with B



State Evolution Analysis of AMP

(As in Bayati-Montanari '10 in the scalar case)

The MSE obey the following recursion:

$$E^t = \int \mathcal{D}\bar{z} \left([f^{(1)}_{a_1}(\Sigma^{t-1}, \bar{z}) - 1]^2 + (B-1)[f^{(0)}_{a_{1,2}}(\Sigma^{t-1}, \bar{z})]^2 \right)$$

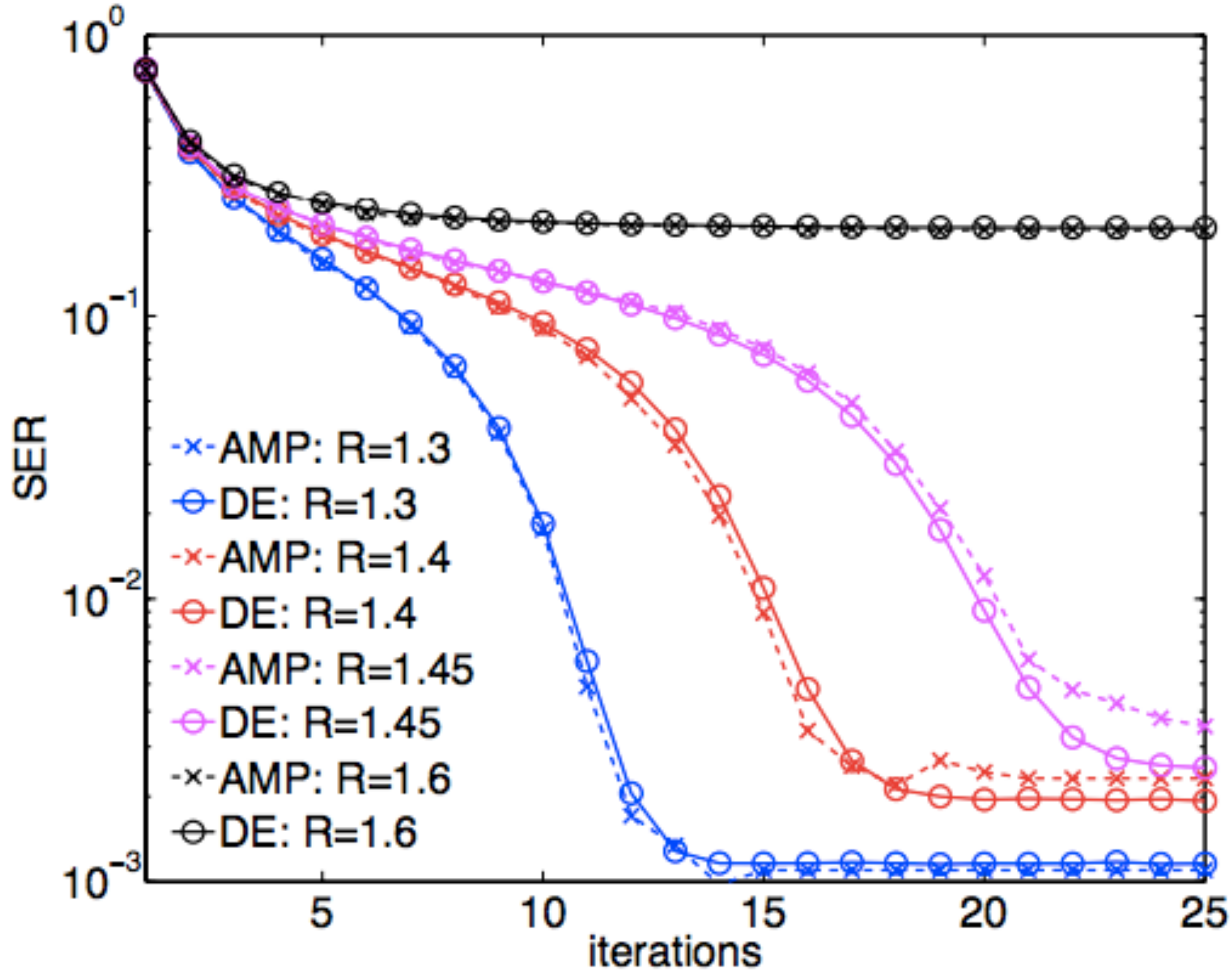
with: $$f^{(1)}_{a_i}(\Sigma, \bar{z}) := \left[1 + e^{-\frac{1}{\Sigma^2}} \sum_{\{1 \le j \le B : j \ne i\}} e^{\frac{z_j - z_i}{\Sigma}} \right]^{-1} \quad f^{(0)}_{a_{i,j}}(\Sigma, \bar{z}) := \left[1 + e^{\frac{1}{\Sigma^2} + \frac{z_j - z_i}{\Sigma}} + \sum_{\{1 \le k \le B : k \ne i,j\}} e^{\frac{z_k - z_i}{\Sigma}} \right]^{-1}$$

with $$\Sigma^t := \sqrt{(1/\mathrm{snr} + E^t) R / \log_2(B)}, \; E^t := \frac{1}{L} \sum_{i=1}^{N} (a_i^t - x_i)^2$$

The Section Error Rate can be deduced from the MSE at all times by

$$\mathrm{SER}^t = \int \mathcal{D}\bar{z} \, \mathbb{I} \left(\exists \, j \in \{2, ..., B\} : f^{(0)}_{a_{j,1}}(\Sigma^t, \bar{z}) > f^{(1)}_{a_1}(\Sigma^t, \bar{z}) \right)$$

State Evolution vs AMP for finite sizes



B= 4, SNR=15, N=32768, R_{BP}=1.55

State Evolution Analysis and the replica potential

$$E^t = \int \mathcal{D}\overline{z} \left([f^{(1)}_{a_1}(\Sigma^{t-1}, \overline{z}) - 1]^2 + (B-1)[f^{(0)}_{a_{1,2}}(\Sigma^{t-1}, \overline{z})]^2 \right)$$

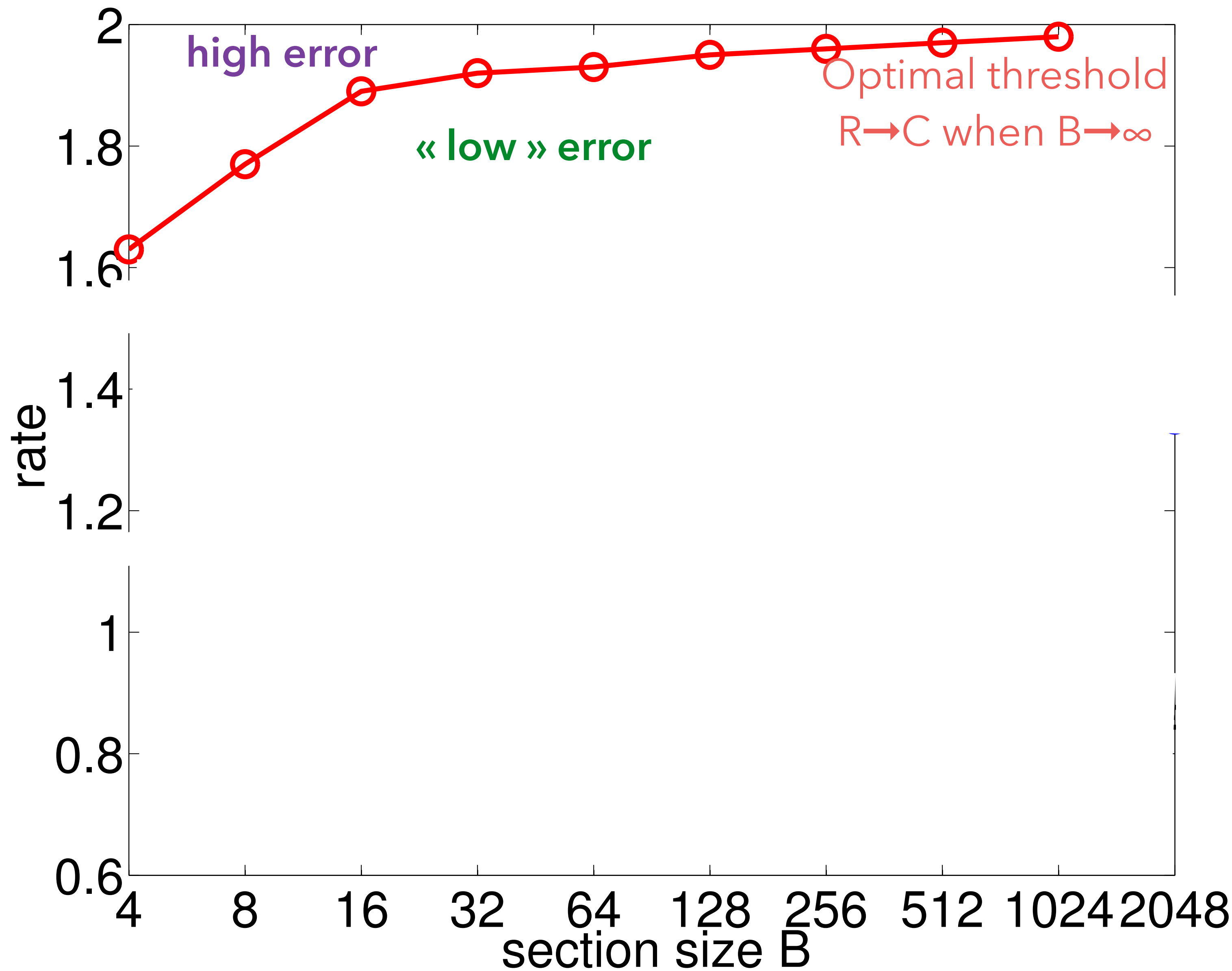
with: $f^{(1)}_{a_i}(\Sigma, \overline{z}) := \left[1 + e^{-\frac{1}{\Sigma^2}} \sum_{\{1 \le j \le B : j \ne i\}} e^{\frac{z_j - z_i}{\Sigma}} \right]^{-1}$ $\quad f^{(0)}_{a_{i,j}}(\Sigma, \overline{z}) := \left[1 + e^{\frac{1}{\Sigma^2} + \frac{z_j - z_i}{\Sigma}} + \sum_{\{1 \le k \le B : k \ne i,j\}} e^{\frac{z_k - z_i}{\Sigma}} \right]^{-1}$

Replica Potential:

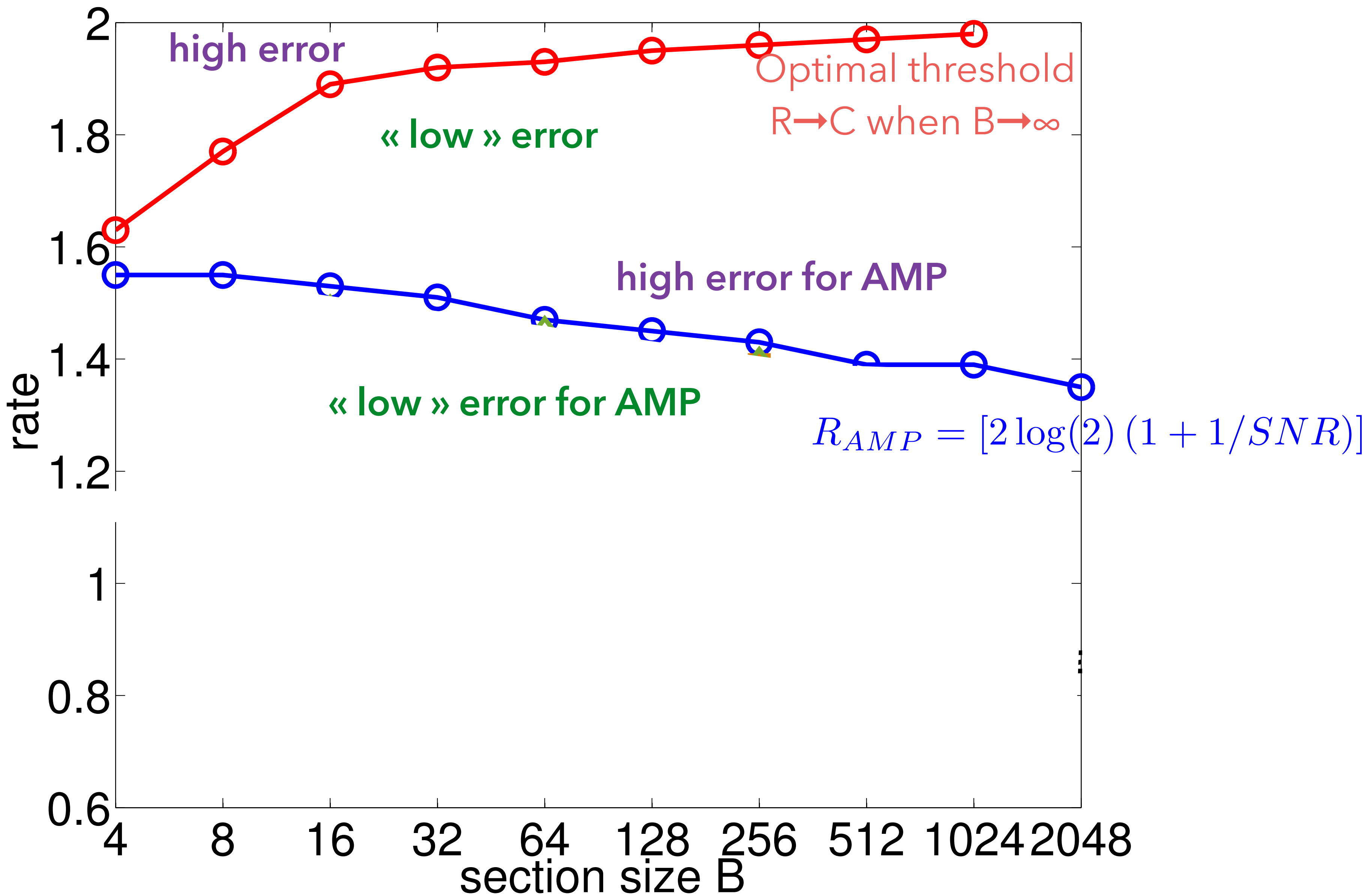
$$\Phi_B(E) = -\frac{\log_2(B)}{2R} \left(\log(1/\mathrm{snr} + E) + \frac{1-E}{1/\mathrm{snr} + E} \right) + \int \mathcal{D}\overline{z} \log \left(e^{\frac{1}{2\Sigma(E)^2} + \frac{z_1}{\Sigma(E)}} + \sum_{i=2}^{B} e^{-\frac{1}{2\Sigma(E)^2} + \frac{z_i}{\Sigma(E)}} \right)$$

The fixed point of the state evolution corresponds = extrema of the potential
Potential = Bethe free energy

Phase diagram SNR=15 (C=2)

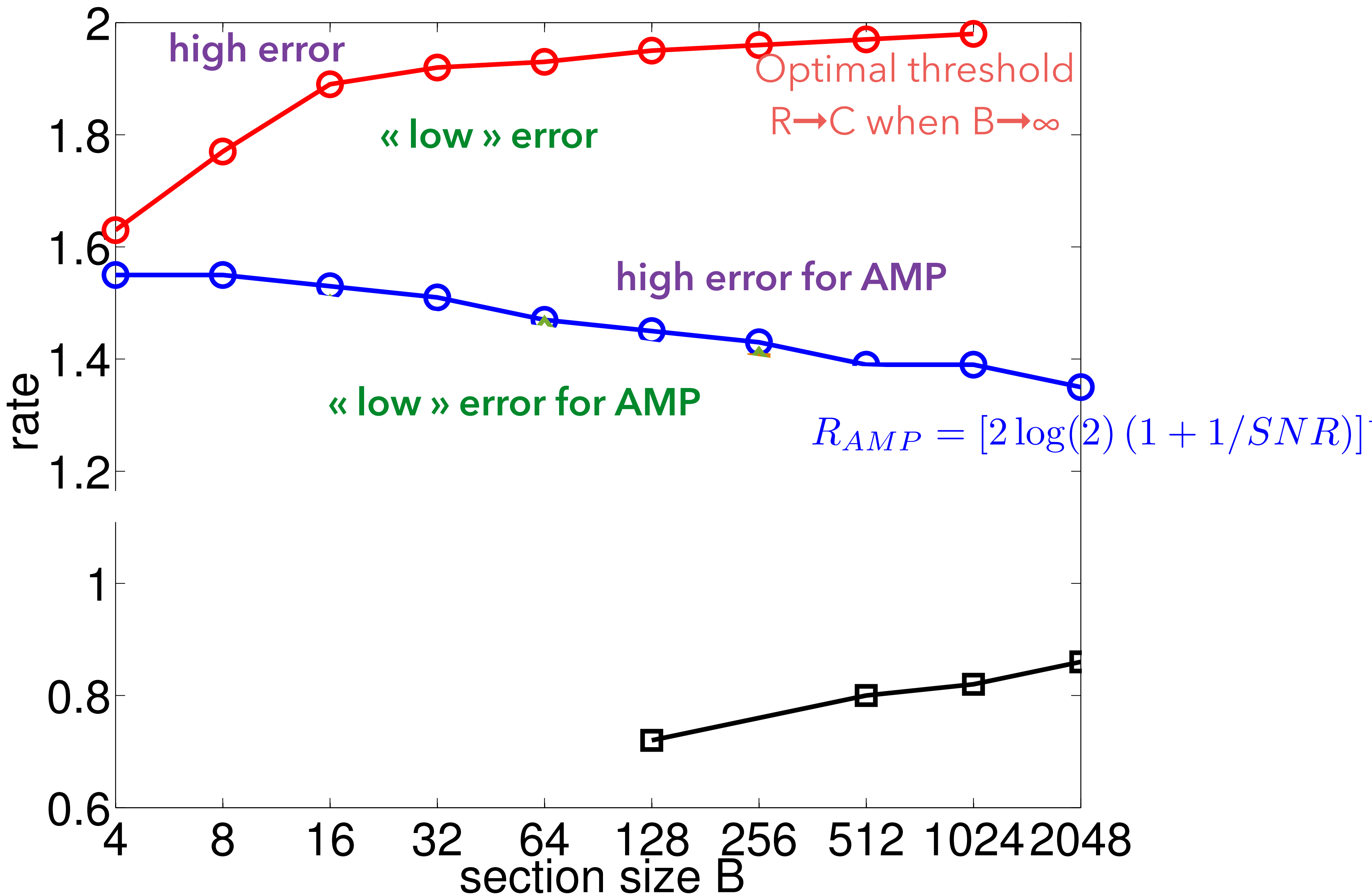


Phase diagram SNR=15 (C=2)



AMP gets worst as B increases :-(

Phase diagram SNR=15 (C=2)



… but is still way better than the alternative :-)

3

Two strategies to reach capacity

Two ways to make AMP capacity achieving:

Power Allocation

Two ways to make AMP capacity achieving:

Power Allocation

vector X= | 0 1 0 0 | 1 0 0 0 | 0 0 0 1 | 1 0 0 0 |

vector X= | 0 c_1 0 0 | c_2 0 0 0 | 0 0 0 c_3 | c_4 0 0 0 |

Generalization the State evolution
for structured matrices shows
that bAMP can reach capacity as B➡∞

$$c_i^2 \propto 2^{-2Ci/L}$$

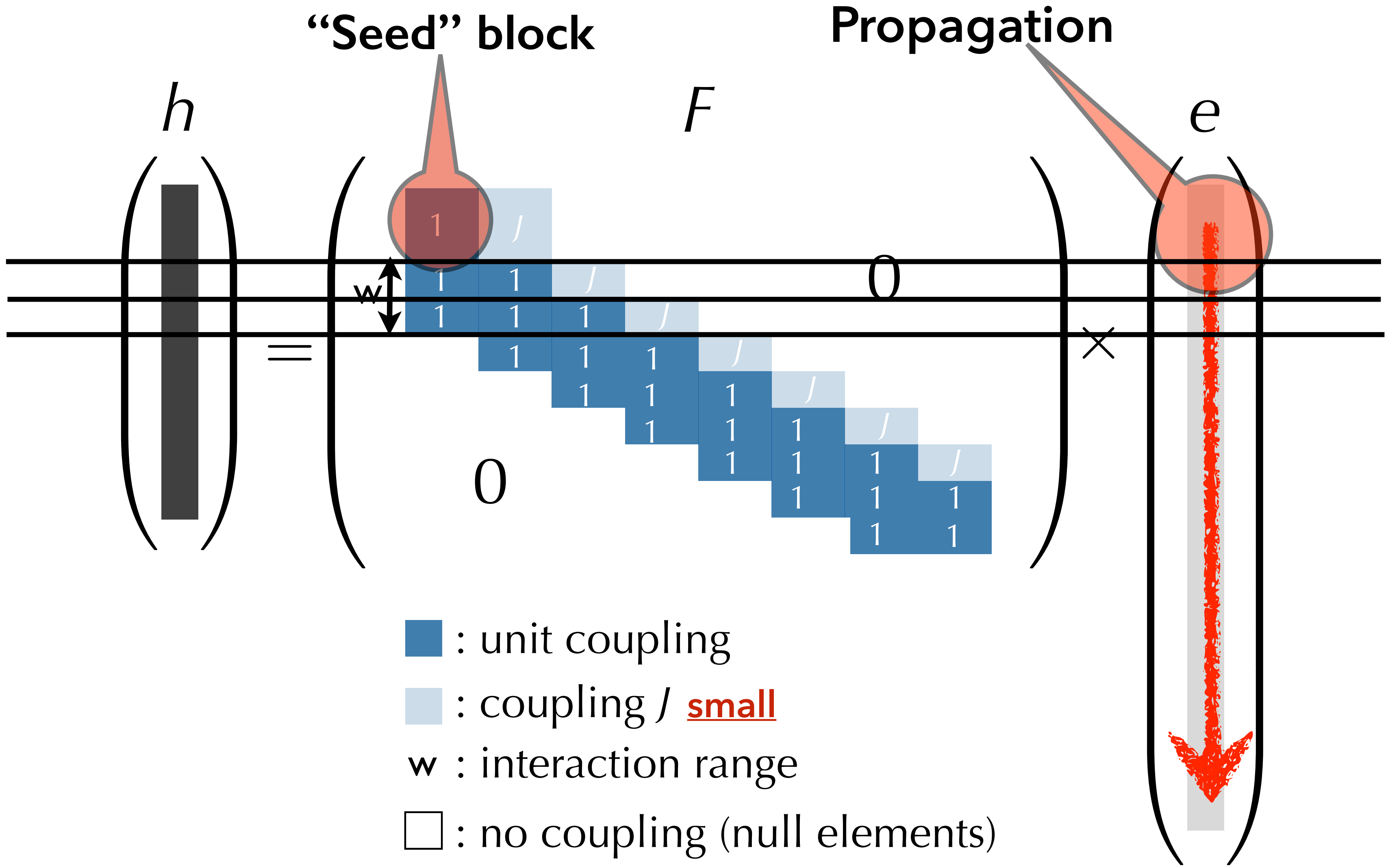
Power constraint: $\langle c_i^2 \rangle = 1$

In practice, however, this turns out to be not so efficient…

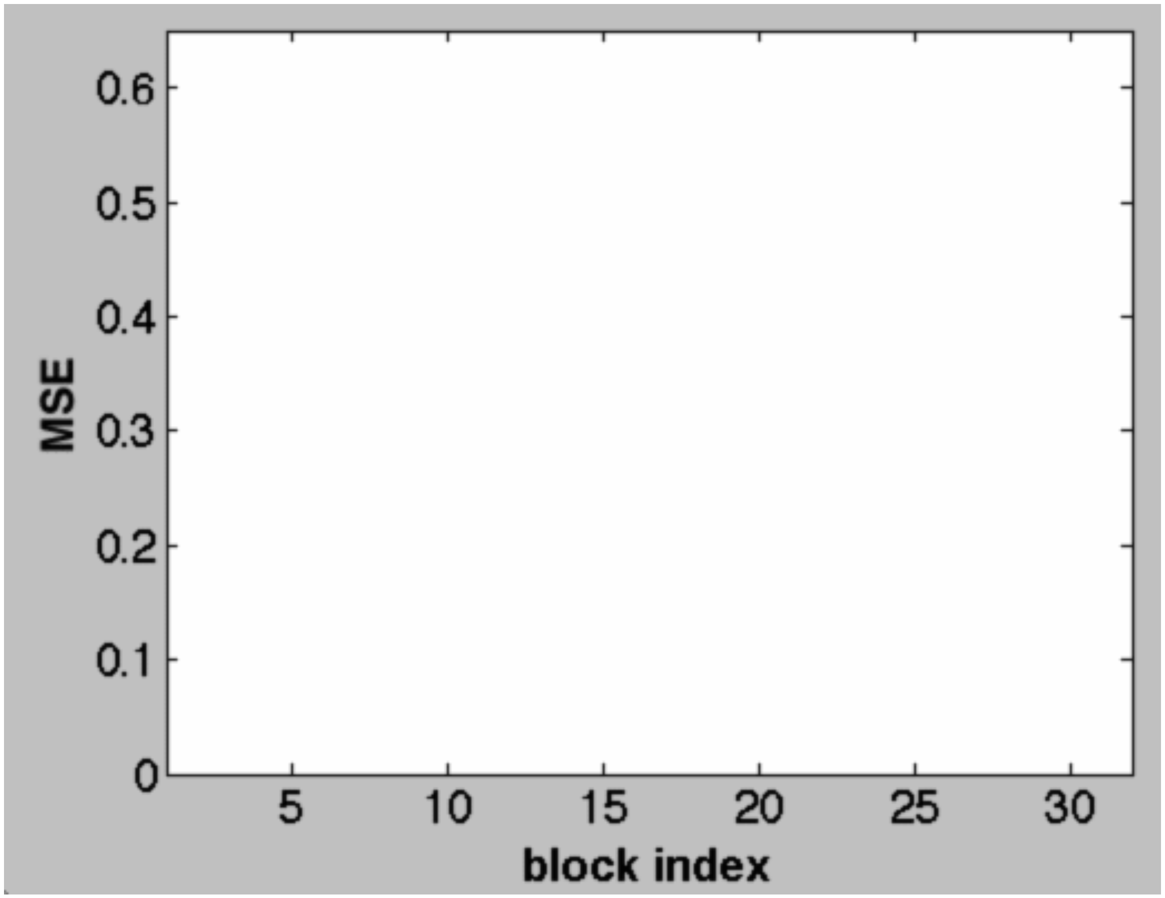
Two ways to make AMP capacity achieving:

Spatial coupling

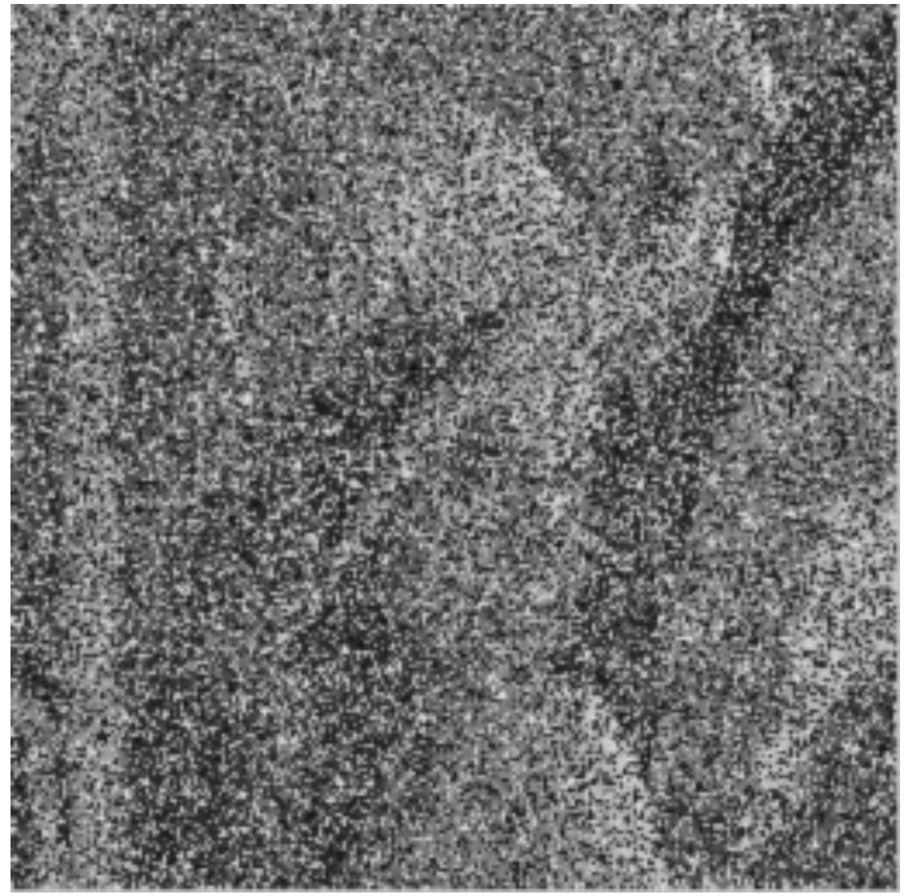
Use of a structured matrix
(Similar constructions in compressed sensing):
Pfister-Kudekar ’09
Krzakala et al ’11
Donoho et al ’11



Spatial coupling



Lena: L=256^2
B=256 levels of grey

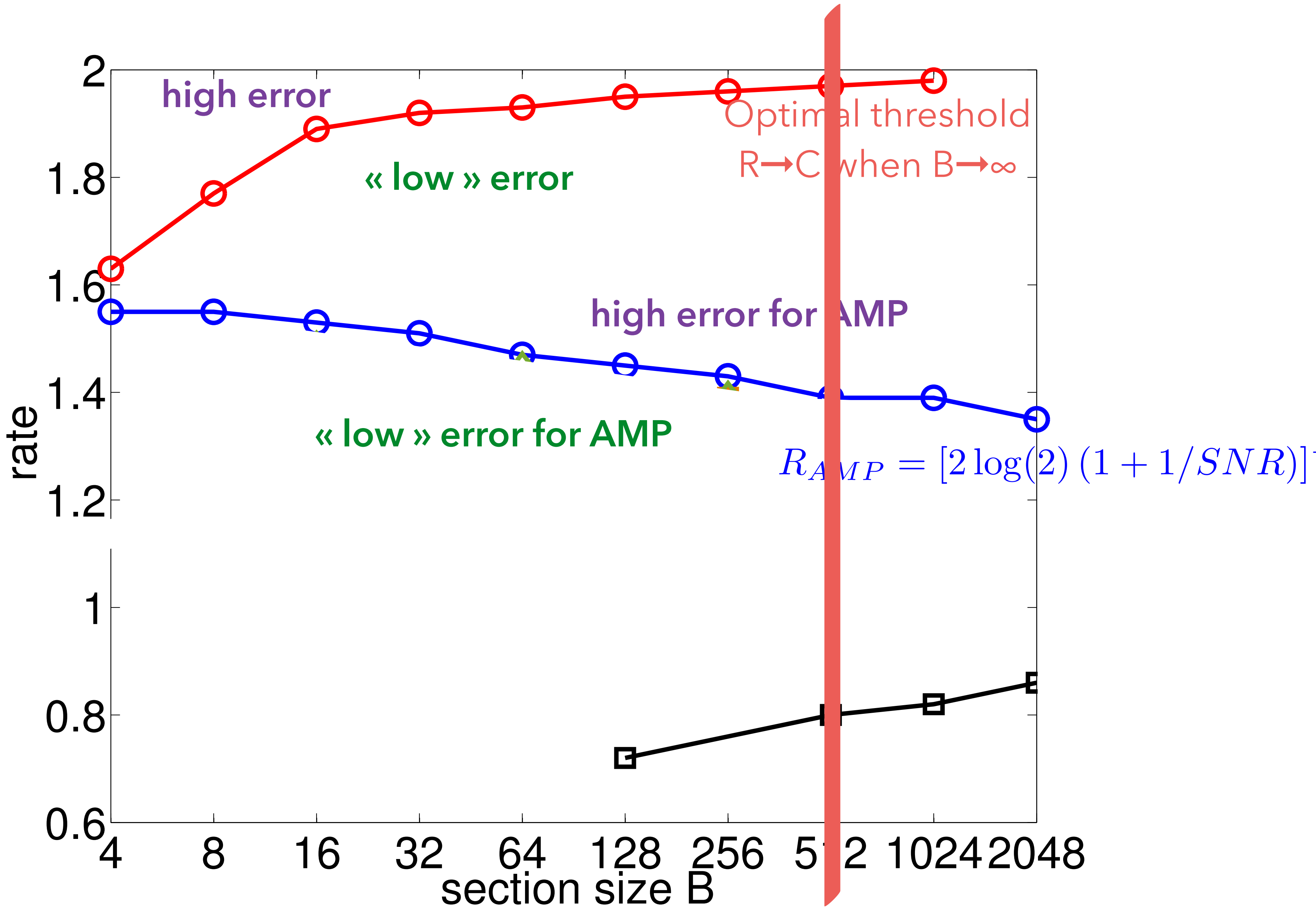


Matlab implementation + demo: https://github.com/jeanbarbier/BPCS_common

4

Practical tests!

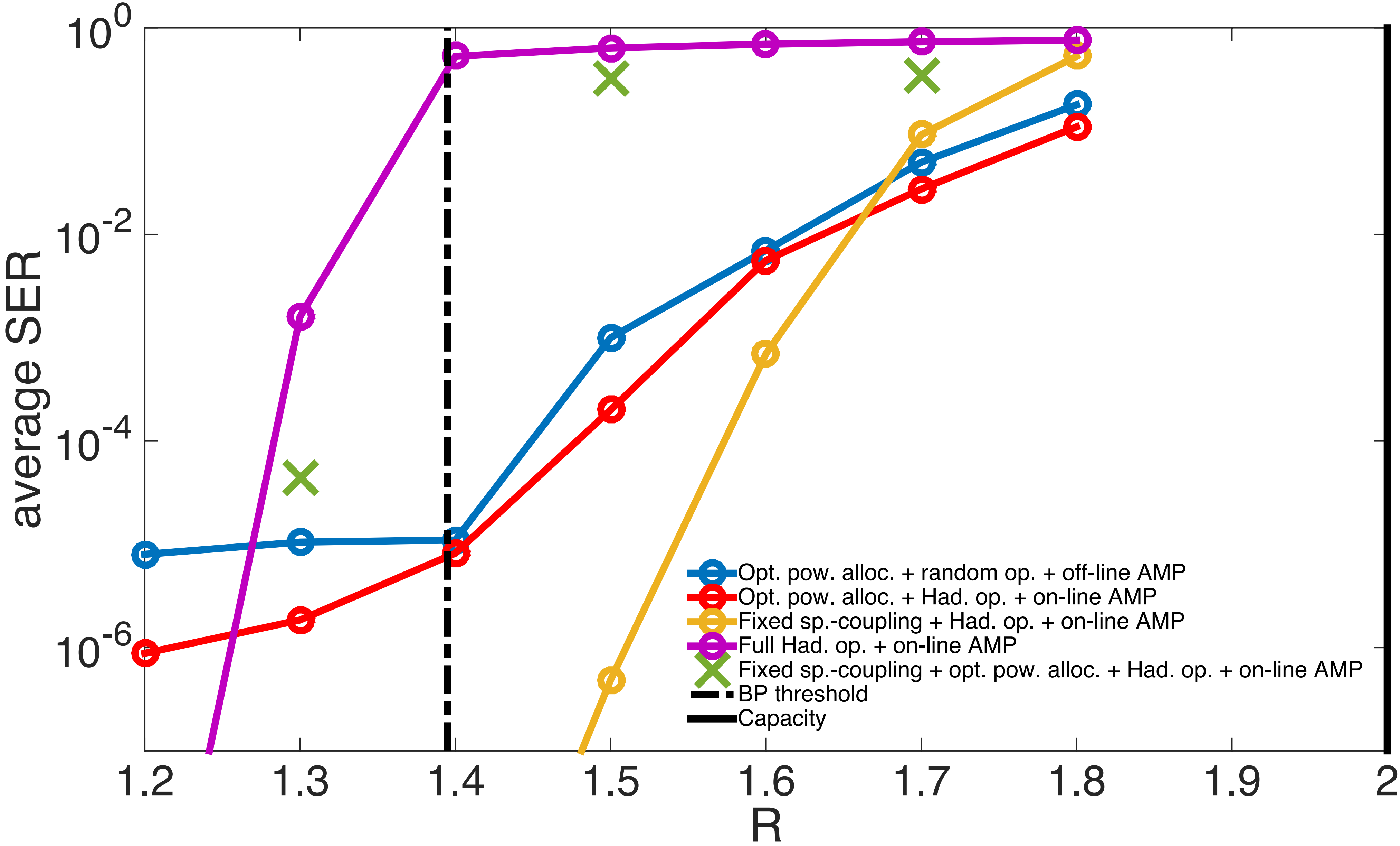
Phase diagram SNR=15 (C=2)



… but is still way better than the alternative :-)

Comparing everything
snr = 15, B = 512, L = 1024

Blocklenght ~ 5000



Conclusion

- **Simple** coding, **capacity achieving** with spatial coupling/power allocation
- Similar phenomenology as LDPC codes
- **Efficient when used with structured operator** such as Hadamard
- Can be **analyzed** by state evolution/replica method
- Interesting **finite size** performances ?