Turbo Compressed Sensing with Partial DFT Sensing Matrix

上海科技大学
ShanghaiTech University

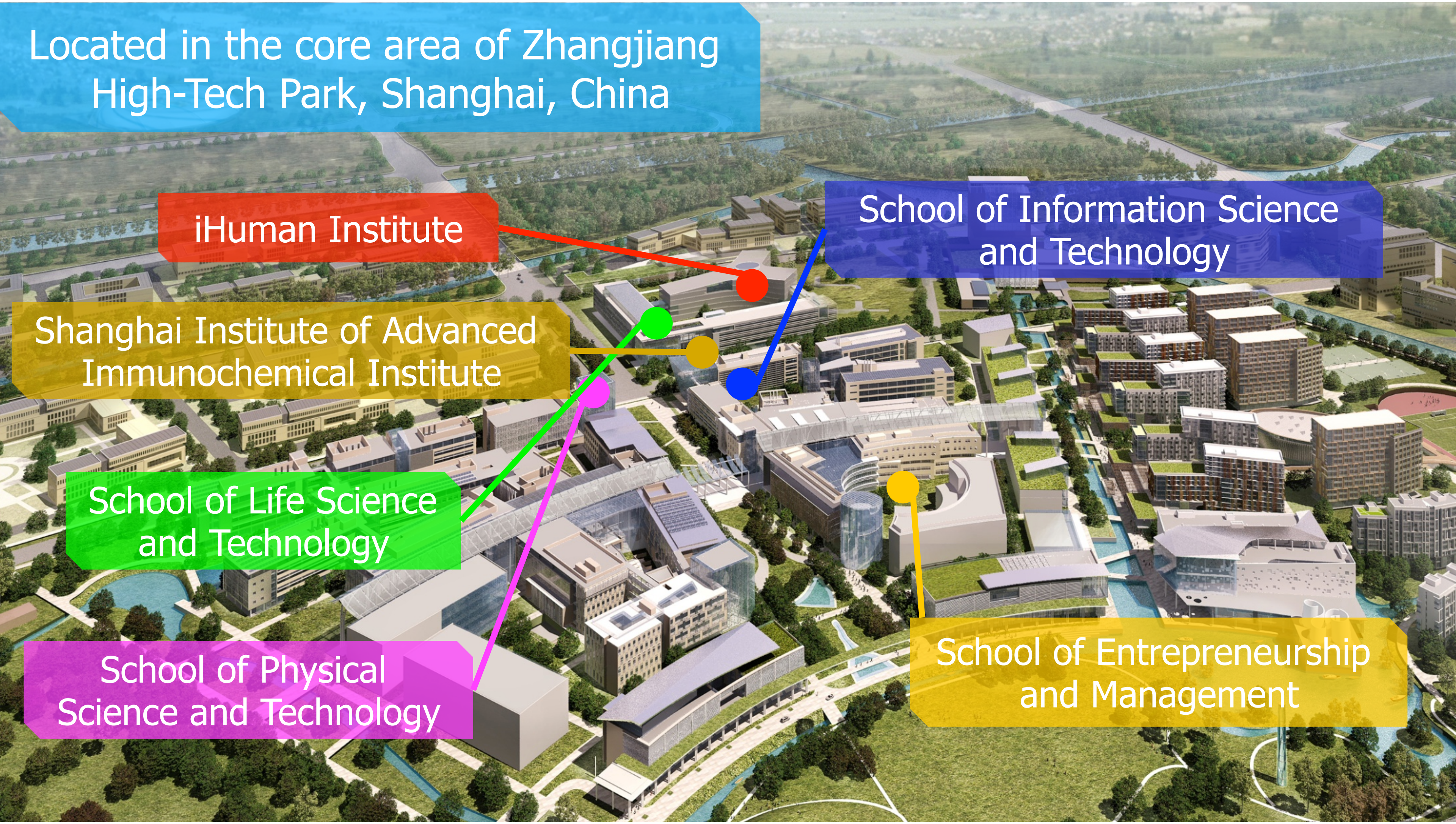
Xiaojun Yuan
School of Information Science and Technology
ShanghaiTech University

Presented on August 17, 2016

ShanghaiTech University

Located in the core area of Zhangjiang High-Tech Park, Shanghai, China

ShanghaiTech University



Junjie Ma, Xiaojun Yuan, and Li Ping, "Turbo compressed sensing with partial DFT sensing matrix," *IEEE Signal Processing Lett*, vol. 22, no. 2, pp. 158-161, Feb. 2015.

Junjie Ma, Xiaojun Yuan, and Li Ping, "On the performance of turbo signal recovery with partial DFT sensing matrices," *IEEE Signal Processing Lett*, vol. 22, no. 10, pp. 1580-1584, Oct. 2015.

Zhipeng Xue, Junjie Ma, and Xiaojun Yuan, "D-OAMP: A denoising-based signal recovery algorithm for compressed sensing," submitted to *GlobalSIP 2016*.

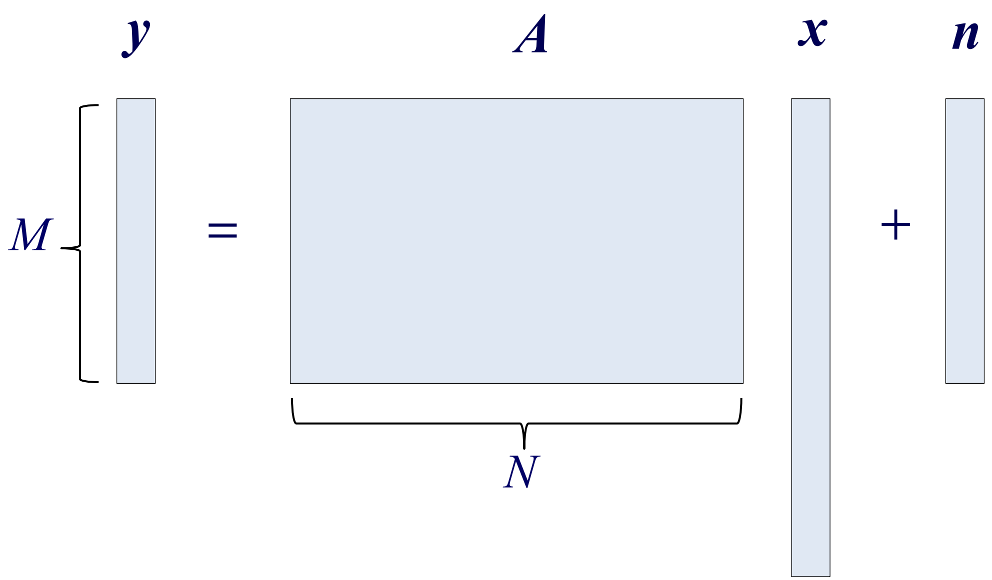
Junjie Ma and **Li Ping** are with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong.

Zhipeng Xue is with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China.

Background and Motivations

Compressed Sensing: Problem Formulation

Consider an underdetermined linear system ($M < N$):



- Sensing matrix $\boldsymbol{A}$: M-by-N, *a priori* known
- Problem: To determine $\boldsymbol{x}$ based on $\boldsymbol{y}$ and the knowledge that $\boldsymbol{x}$ is sparse.
- Applications: photography, facial recognition, network tomography, etc

Compressed Sensing Algorithms

- l_0-minimization
  - Non-convex problem

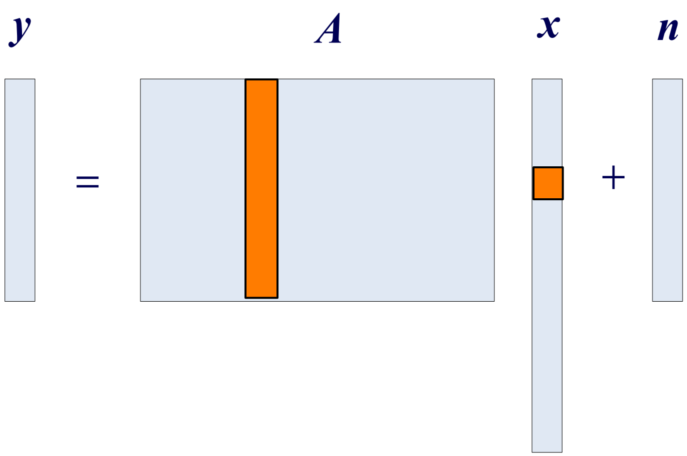
$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ \frac{1}{\lambda} \left\| \mathbf{y} - \mathbf{A}\mathbf{x} \right\|_2^2 + \left\| \mathbf{x} \right\|_0 \right\}$$

- l_1-minimization
  - Least absolute shrinkage and selection operator (LASSO)
  - Convex programming with polynomial time
  - Scalability is still an issue…

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ \frac{1}{\lambda} \left\| \mathbf{y} - \mathbf{A}\mathbf{x} \right\|_2^2 + \left\| \mathbf{x} \right\|_1 \right\}$$

Low-Complexity Approaches

- Greedy algorithms
  - Orthogonal matching pursuit (OMP)
  - Iterative hard thresholding
- Iterative algorithms
  - Iterative soft-thresholding (IST)
  - Fast iterative soft-threholding
- Probabilistic inference
  - Approximate message passing (AMP) [Donoho09]: near-optimal with linear complexity



Approximate Message-Passing (AMP) Algorithm

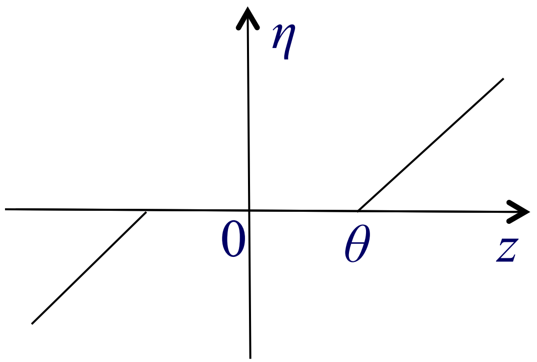
Algorithm:

$$\boldsymbol{u}^t = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^t + \frac{1}{N}\sum_{i=1}^{N}\eta'\left(\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{u}^{t-1} + \boldsymbol{x}^{t-1}\right)_i\right)\cdot \boldsymbol{u}^{t-1}$$

$$\boldsymbol{x}^{t+1} = \eta\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{u}^t + \boldsymbol{x}^t\right)$$

where $\eta(\cdot)$ is the scalar threshold function (or denoiser) defined as

$$\eta(z;\theta) = \begin{cases} z-\theta & \text{if } z > \theta \\ z+\theta & \text{if } z < -\theta \\ 0 & \text{otherwise} \end{cases}$$



and $\eta'(\cdot)$ is its derivative.

Main Idea of AMP

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ \frac{1}{\lambda} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \|\mathbf{x}\|_1 \right\}$$

Main Idea of AMP

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ \frac{1}{\lambda} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \|\mathbf{x}\|_1 \right\}$$

Assign a probability model

$$\arg\max_{\mathbf{x}} \exp\left\{ -\frac{1}{\lambda} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 - \|\mathbf{x}\|_1 \right\} = \arg\max_{\mathbf{x}} p(\mathbf{x} \mid \mathbf{y})$$

Main Idea of AMP

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ \frac{1}{\lambda} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \|\mathbf{x}\|_1 \right\}$$

Assign a probability model

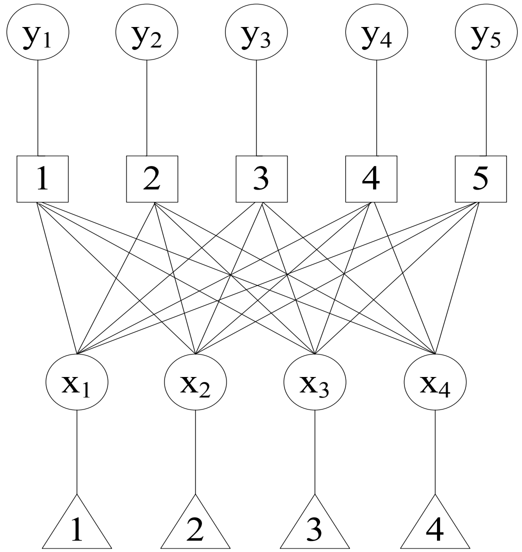
$$\arg\max_{\mathbf{x}} \exp\left\{ -\frac{1}{\lambda} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 - \|\mathbf{x}\|_1 \right\} = \arg\max_{\mathbf{x}} p(\mathbf{x} \mid \mathbf{y})$$

Factor graph + belief propagation

$p(y_n \mid \mathbf{x})$

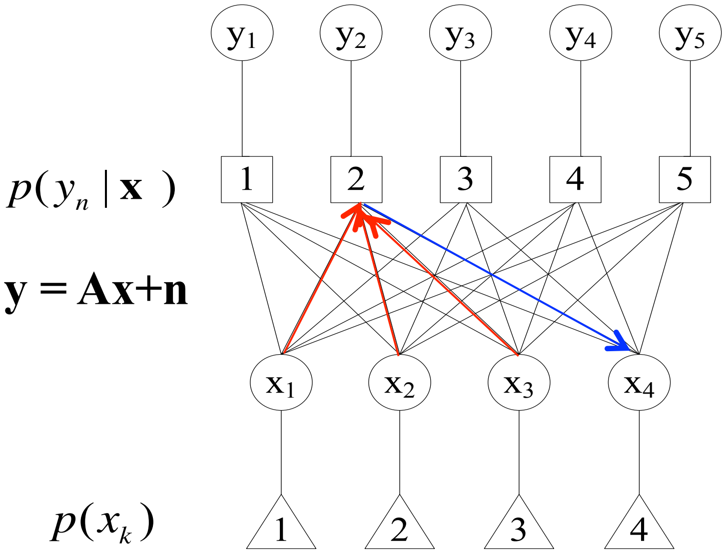
$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$

$p(x_k)$



Complexity of AMP

- The complexity of message passing is proportional to the total number of edges in the graph.
- In general, the sensing matrix **A** is a dense matrix, implying a high complexity.

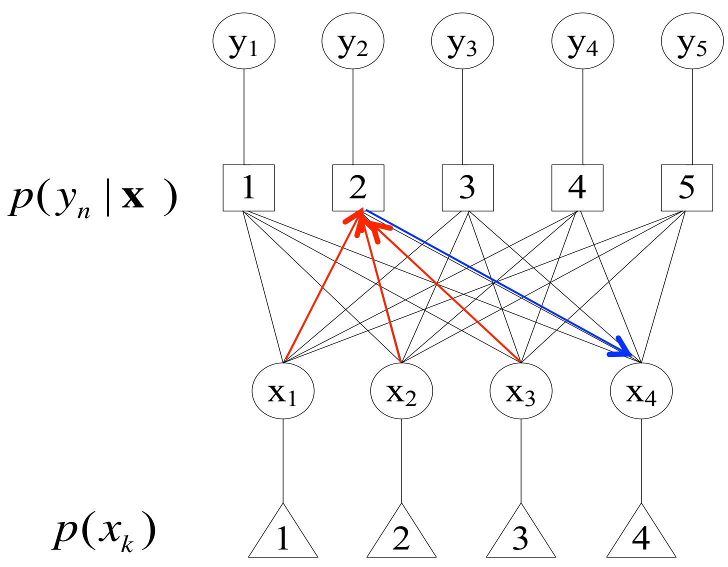


AMP takes two approximations to reduce complexity:

- Gaussian approximation → only need to track mean and variance
- First-order Taylor approximation

Performance of AMP

- Belief propagation depends on the independence of messages.
- The factor graph contains many short loops, which may compromise the independence of messages.
- When the entries of **A** are iid, AMP is near-optimal.
- For a structured **A**, the performance of AMP is not guaranteed.



Good news: Randomness of **A** ensures the approximate independence of messages.

Structured Sensing Matrix A

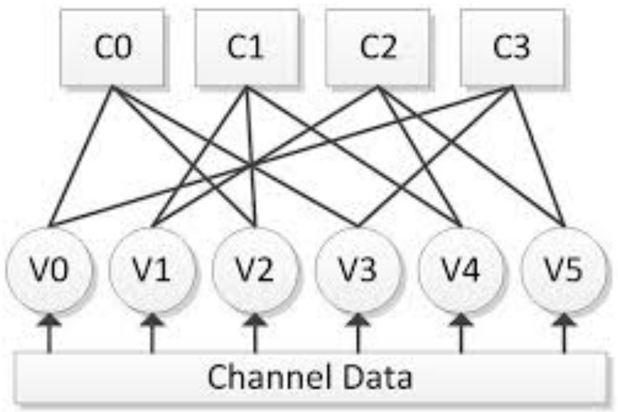
- In many applications, **A** is structured rather than iid random.
- For example, **A** consists of random rows of the DFT matrix in image processing, such as magnetic resonance imaging (MRI).
- AMP doesn't work well when **A** is a partial DFT matrix.

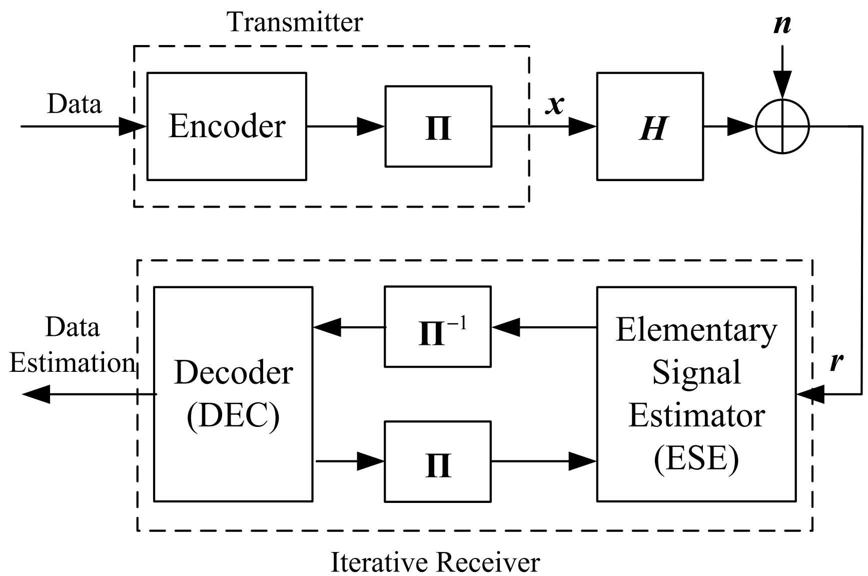
Challenge

For structured sensing matrices, how to design a linear-complexity compressed sensing algorithm with near-optimal performance?

Turbo Compressed Sensing

LDPC Decoding vs. Turbo Detection

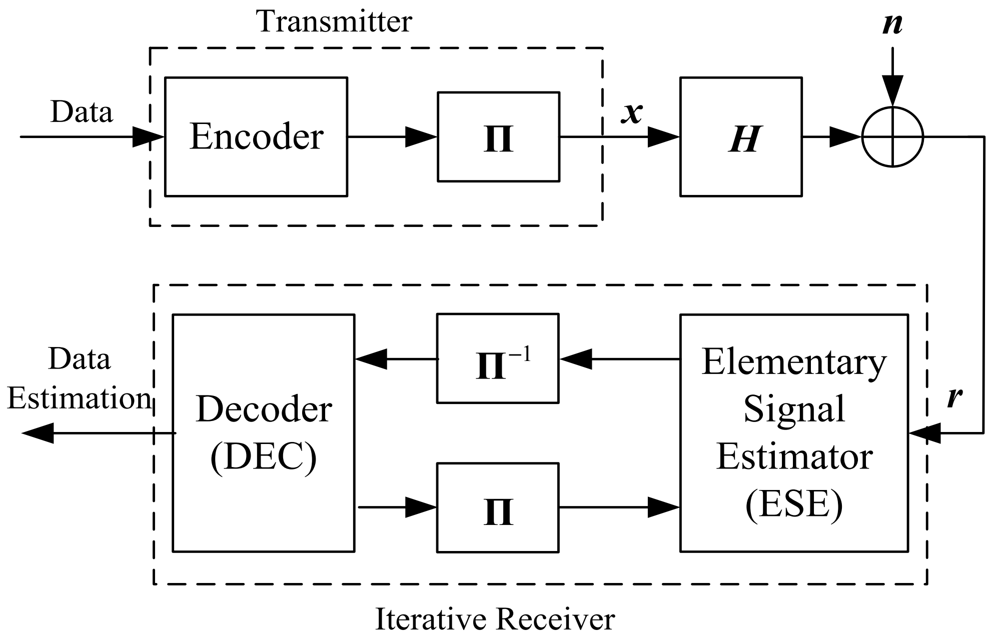




LDPC decoding as an analogy to AMP

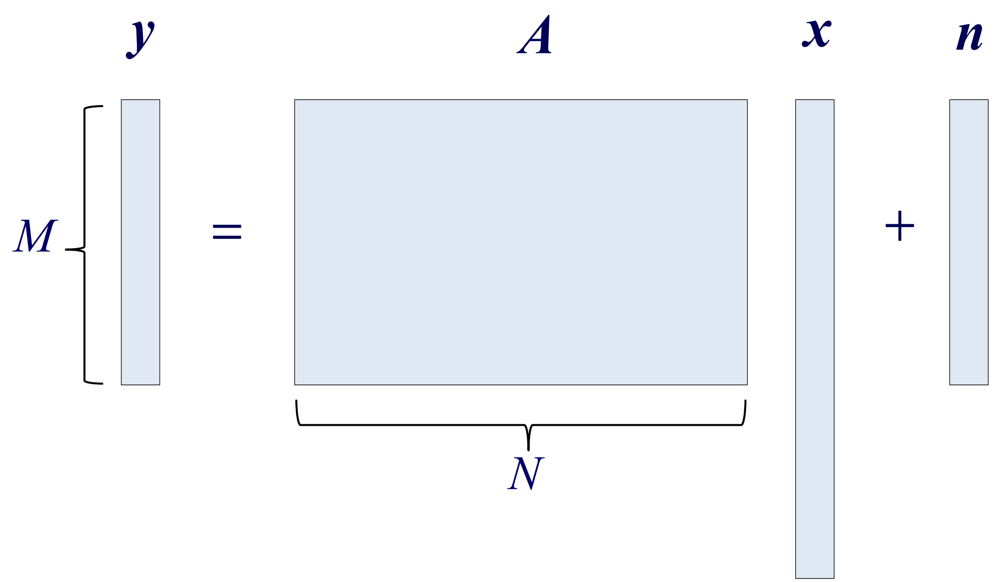
Turbo detection

Turbo Detection for MIMO Systems



- The main idea of turbo detection is to divide the whole inference problem into two component problems, and then do detection for each component iteratively.
- The following two features guarantee the success of turbo detection:
  - Use random interleaver Π
  - Pass extrinsic messages

Problem Formulation Revisited

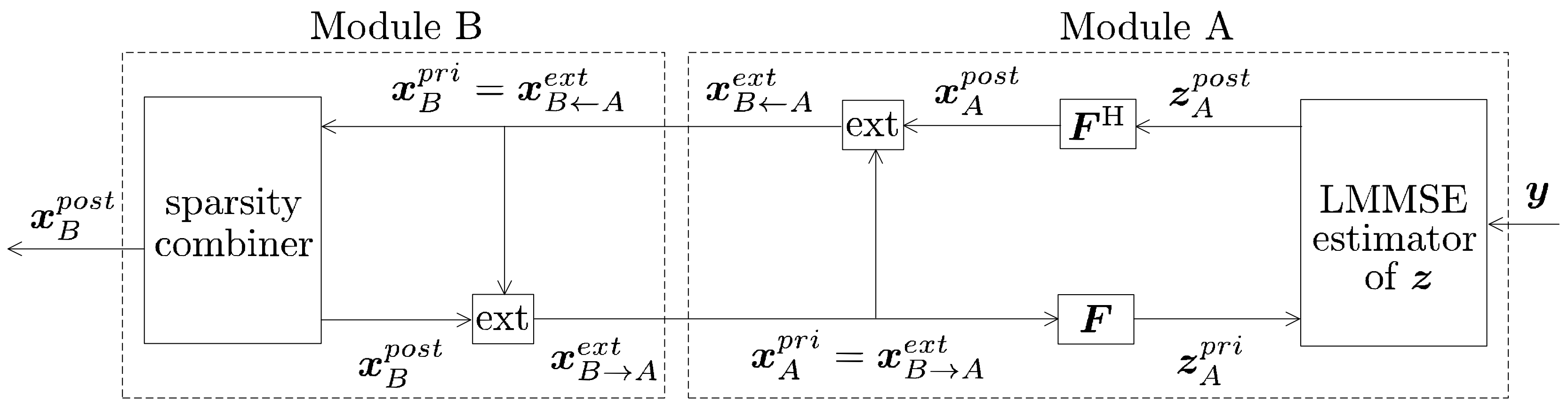


- Our goal is to estimate $\boldsymbol{x}$ with partial orthogonal matrix $\boldsymbol{A} = \boldsymbol{F}_{\text{partial}}$
- Stakes at hand:
  - The measurement vector $\boldsymbol{y}$
  - $\boldsymbol{x}$ is a sparse signal

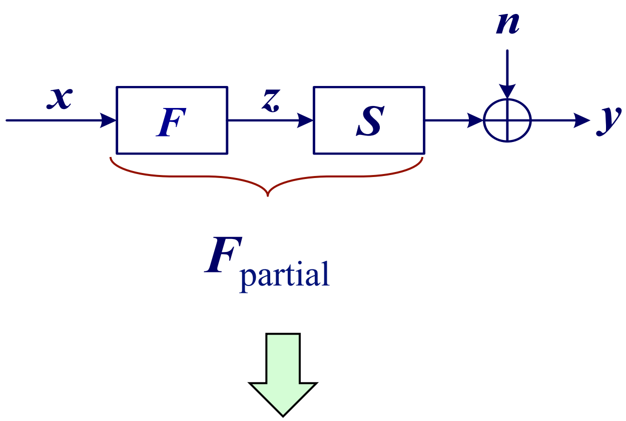
Turbo Detector: A First Attempt

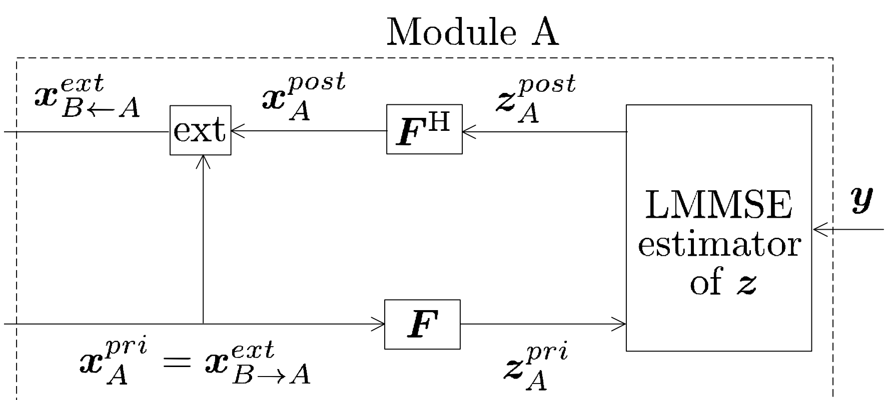
Module A: $\boldsymbol{y} = \boldsymbol{F}_{\text{partial}}\boldsymbol{x} + \boldsymbol{n}$

Module B: $\boldsymbol{x}$ is sparse



Operations of Module A





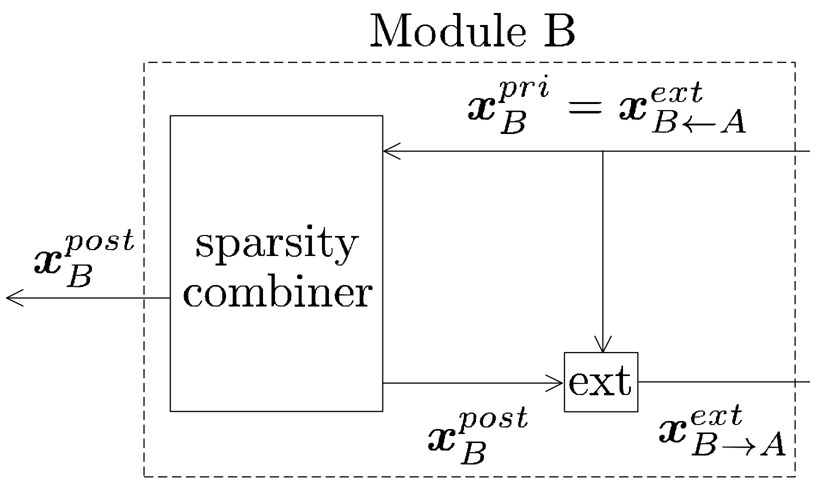
$$x^{ext} = x^{pri} + F^{H}_{\text{partial}}\left(y - F_{\text{partial}} x^{pri}\right)$$

z is approximately Gaussian,
MMSE = LMMSE

Extrinsic message passing is the key to the success of turbo codes [Berrou93]

Extrinsic-message computation rules for LMMSE filtering can be found in [Loeliger07]

Operations of Module B



↓

symbol-wise MMSE denoiser

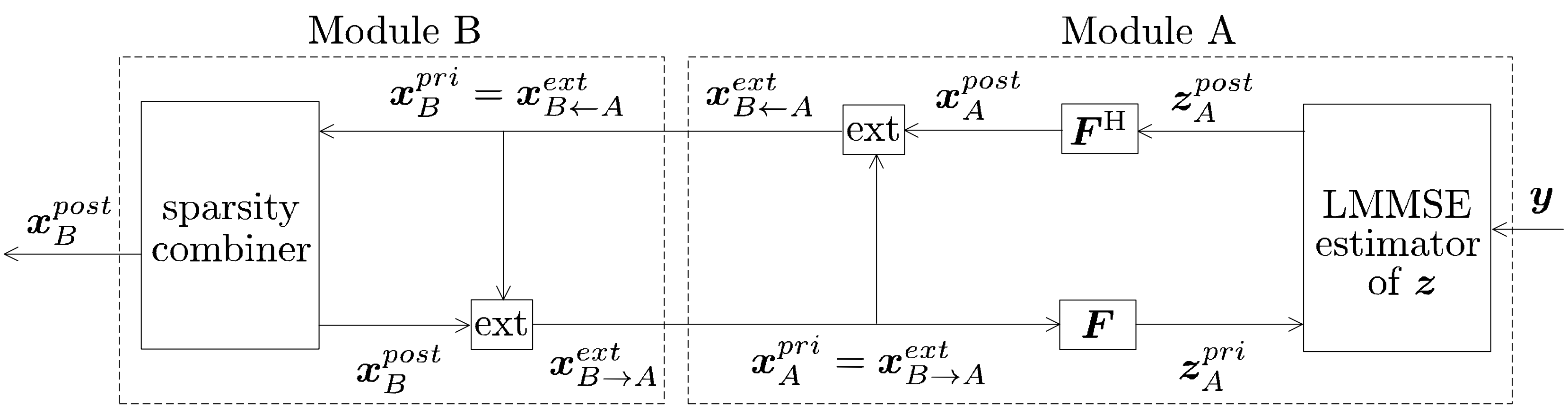
For each entry x_i, the sparsity combiner combines the message of x_i from Module A and the *a priori* of x_i

Extrinsic message passing is the key to the success of turbo detection

Turbo Detector: A First Attempt

Module A: $\boldsymbol{y} = \boldsymbol{F}_{\text{partial}}\boldsymbol{x} + \boldsymbol{n}$

Module B: $\boldsymbol{x}$ is sparse



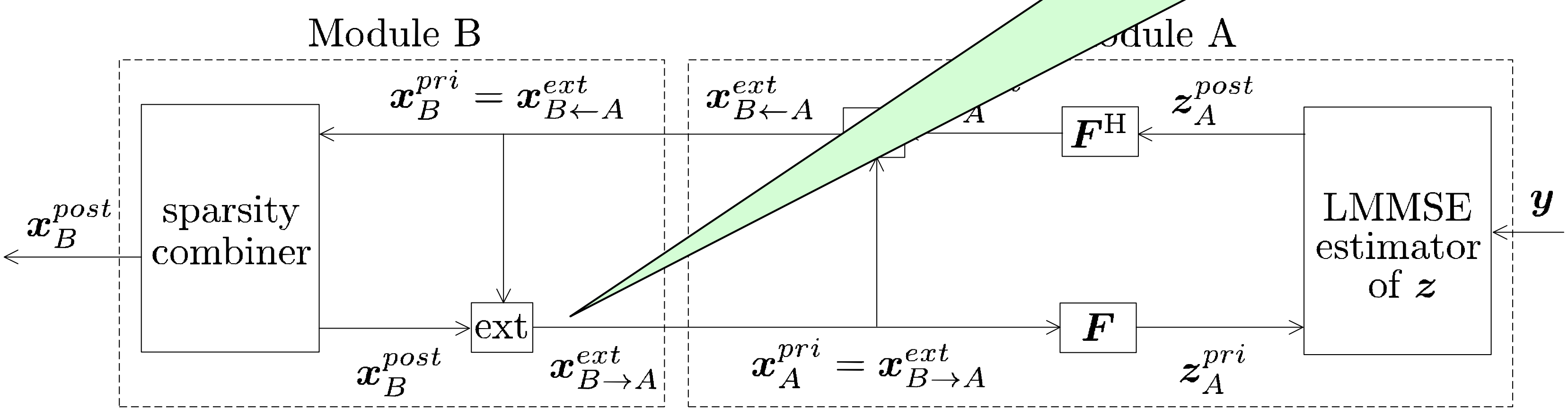
- The function of interleaving is achieved by the DFT transform.
- The problem is with the calculation of extrinsic messages.

Turbo Detector: A First Attempt

Module A: $\boldsymbol{y} = \boldsymbol{F}_{\text{partial}}\boldsymbol{x} + \boldsymbol{n}$

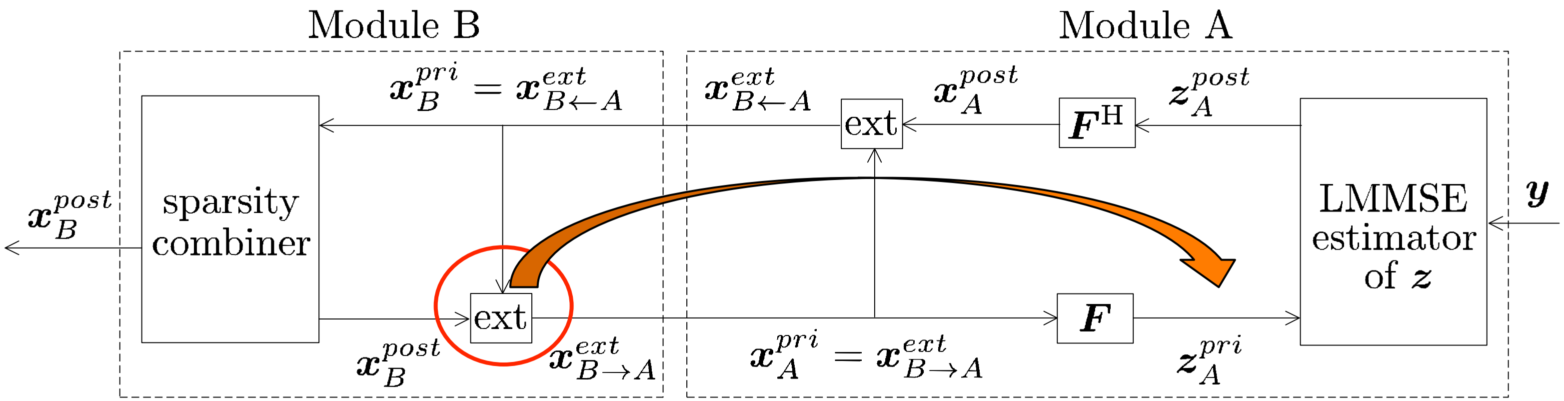
Module B: $\boldsymbol{x}$ is sparse

The extrinsic messages are constants in iteration.

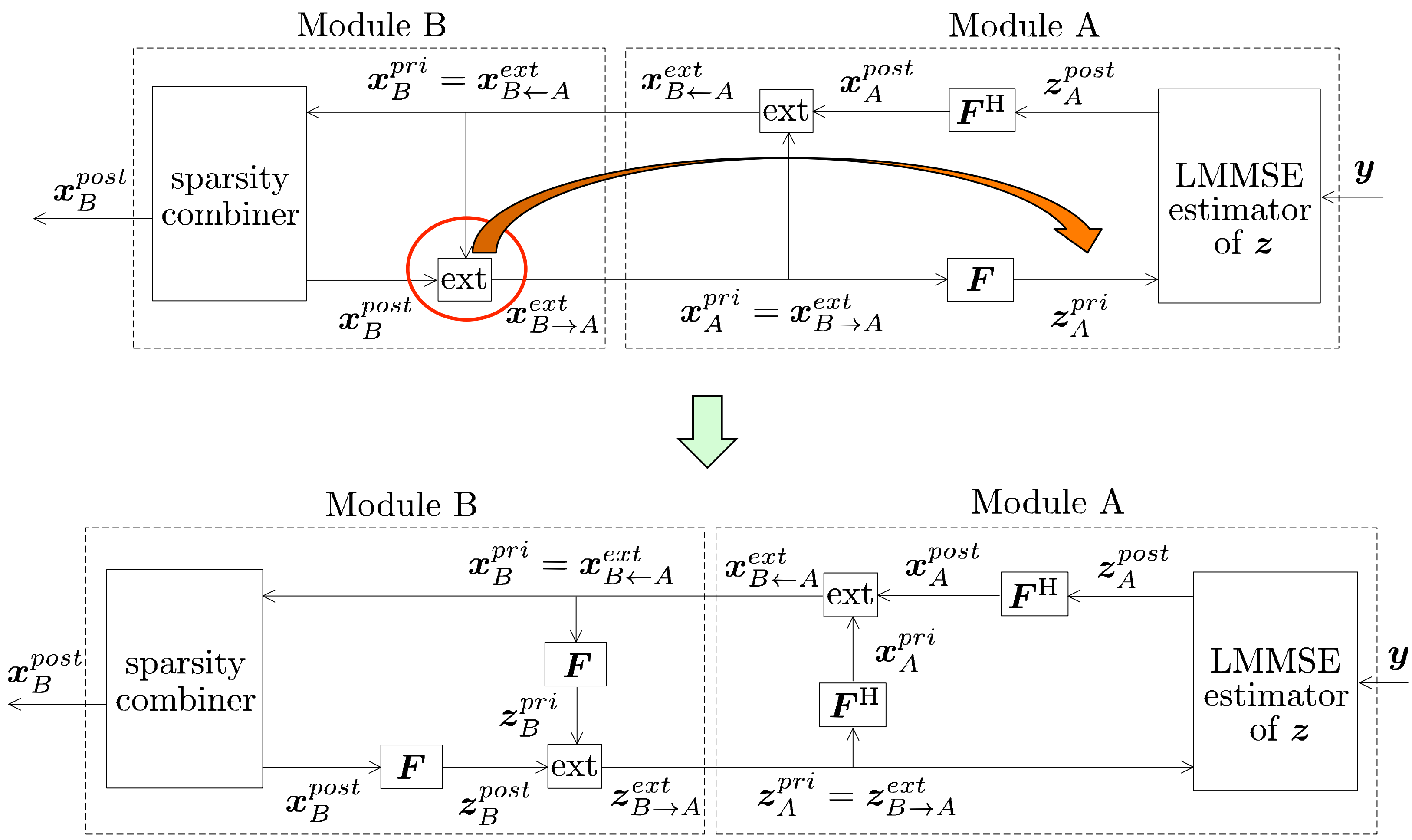


- The function of interleaving is replaced by the DFT transform.
- The problem is with the calculation of extrinsic messages.

Proposed Turbo Detector



Proposed Turbo Detector



Turbo Compressed Sensing - Algorithm

Module B:

$$z_B^{pri} \leftarrow F x_B^{pri}$$

$$x_{B,j}^{post} \leftarrow \mathrm{E}\left\{ x_j \mid x_{B,j}^{pri} \right\}$$

$$v_{B,j}^{post} \leftarrow \mathrm{var}\left\{ x_j \mid x_{B,j}^{pri} \right\}$$

$$z_B^{post} \leftarrow \boldsymbol{F}\boldsymbol{x}_B^{post}$$

$$v_B^{post} \leftarrow \frac{1}{N}\sum_{j=1}^{N} v_{B,j}^{post}$$

$$v_A^{pri} \leftarrow v_B^{ext} \leftarrow \left(\frac{1}{v_B^{post}} - \frac{1}{v_B^{pri}} \right)^{-1}$$

$$z_B^{pri} \leftarrow z_B^{ext} \leftarrow v_B^{ext} \left(\frac{z_B^{post}}{v_B^{post}} - \frac{z_B^{pri}}{v_B^{pri}} \right)$$

Module A:

$$\boldsymbol{x}_A^{pri} \leftarrow \boldsymbol{F}^{\mathrm{H}} z_A^{pri}$$

$$z_A^{post} \leftarrow z_A^{pri} + \frac{v_A^{pri}}{v_A^{pri} + \sigma^2} \boldsymbol{S}^{\mathrm{H}} \left(\boldsymbol{y} - \boldsymbol{S} z_A^{pri} \right)$$

$$v_{A,j}^{post} \leftarrow v_A^{pri} - \frac{\left(v_A^{pri}\right)^2}{v_A^{pri} + \sigma^2} \left(\boldsymbol{S}^{\mathrm{H}} \boldsymbol{S} \right)_j$$

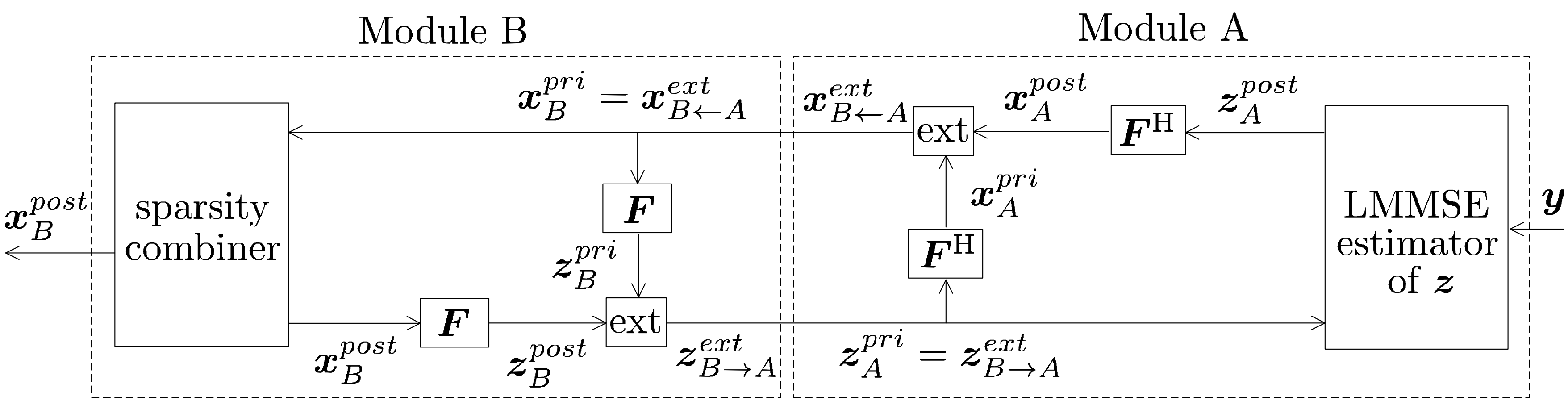
$$\boldsymbol{x}_A^{post} \leftarrow \boldsymbol{F}^{\mathrm{H}} z_A^{post}$$

$$v_A^{post} \leftarrow \frac{1}{N}\sum_{j=1}^{N} v_{A,j}^{post}$$

$$v_B^{pri} \leftarrow v_A^{ext} \leftarrow \left(\frac{1}{v_A^{post}} - \frac{1}{v_A^{pri}} \right)^{-1}$$

$$\boldsymbol{x}_B^{pri} \leftarrow \boldsymbol{x}_A^{ext} \leftarrow v_A^{ext} \left(\frac{\boldsymbol{x}_A^{post}}{v_A^{post}} - \frac{\boldsymbol{x}_A^{pri}}{v_A^{pri}} \right)$$

Computational Complexity



↓ symbol-wise nonlinear MMSE denoiser

↓ symbol-wise linear MMSE estimation

- Module A: symbol-wise estimate of $\boldsymbol{z}$ + extrinsic of $\boldsymbol{x}$
- Module B: symbol-wise estimate of $\boldsymbol{x}$ + extrinsic of $\boldsymbol{z}$
- The complexity is dominated by the DFT transform ($o(N\log N)$).

Performance Analysis

State Evolution for AMP

- For large **iid Gaussian** sensing matrix, it was proved in [Bayati11] that AMP is characterized by the following *state evolution* equations:

linear filter: $$\rho^t = \frac{1}{v^t + \sigma^2}$$

non-linear denoiser: $$v^{t+1} = \psi\left(\rho^t\right)$$

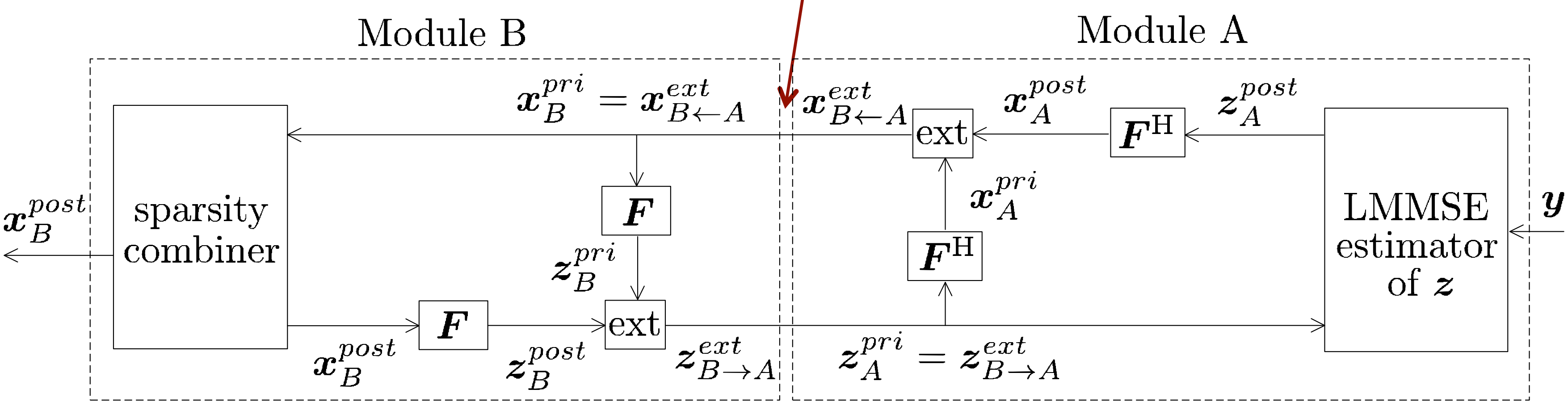
where $$\psi(\rho) \equiv \mathrm{E}\left[\left|\eta\left(y = x + \rho^{-1/2} w\right) - x\right|^2\right]$$

- Similar to density evolution in the analysis of LDPC decoding
- Differences: (i) The factor graph for AMP is dense; (ii) $\boldsymbol{x}$ for AMP is real or complex-valued.

State Evolution for Turbo CS

Gaussian distortion assumption:

$$\boldsymbol{x}_B^{pri} = \boldsymbol{x} + \text{Gaussian noise}$$

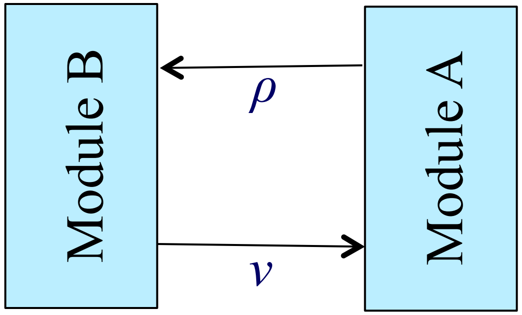


State Evolution for Turbo CS

Based on the Gaussian assumption

Module A: $$\rho^t = \frac{1}{\frac{N-M}{M}\cdot v^t + \frac{N}{M}\cdot\sigma^2}$$

Module B: $$v^{t+1} = \left(\frac{1}{mmse(\rho^t)} - \rho^t\right)^{-1}$$



The fixed point is given by

$$\rho^* = \frac{mmse(\rho^*) + \sigma^2 - \sqrt{\left(mmse(\rho^*)+\sigma^2\right)^2 - 4\cdot\sigma^2\cdot mmse(\rho^*)\cdot\frac{M}{N}}}{2\cdot\sigma^2\cdot\frac{M}{N}}$$

Consistent with MMSE prediction based on the replica method [Tulino13]

Turbo CS vs. AMP-MMSE

$$\rho^t = \frac{1}{\frac{N-M}{M} \cdot v^t + \frac{N}{M} \cdot \sigma^2}$$

$$v^{t+1} = \left(\frac{1}{mmse(\rho^t)} - \rho^t \right)^{-1}$$

Turbo CS for partially orthogonal **A**

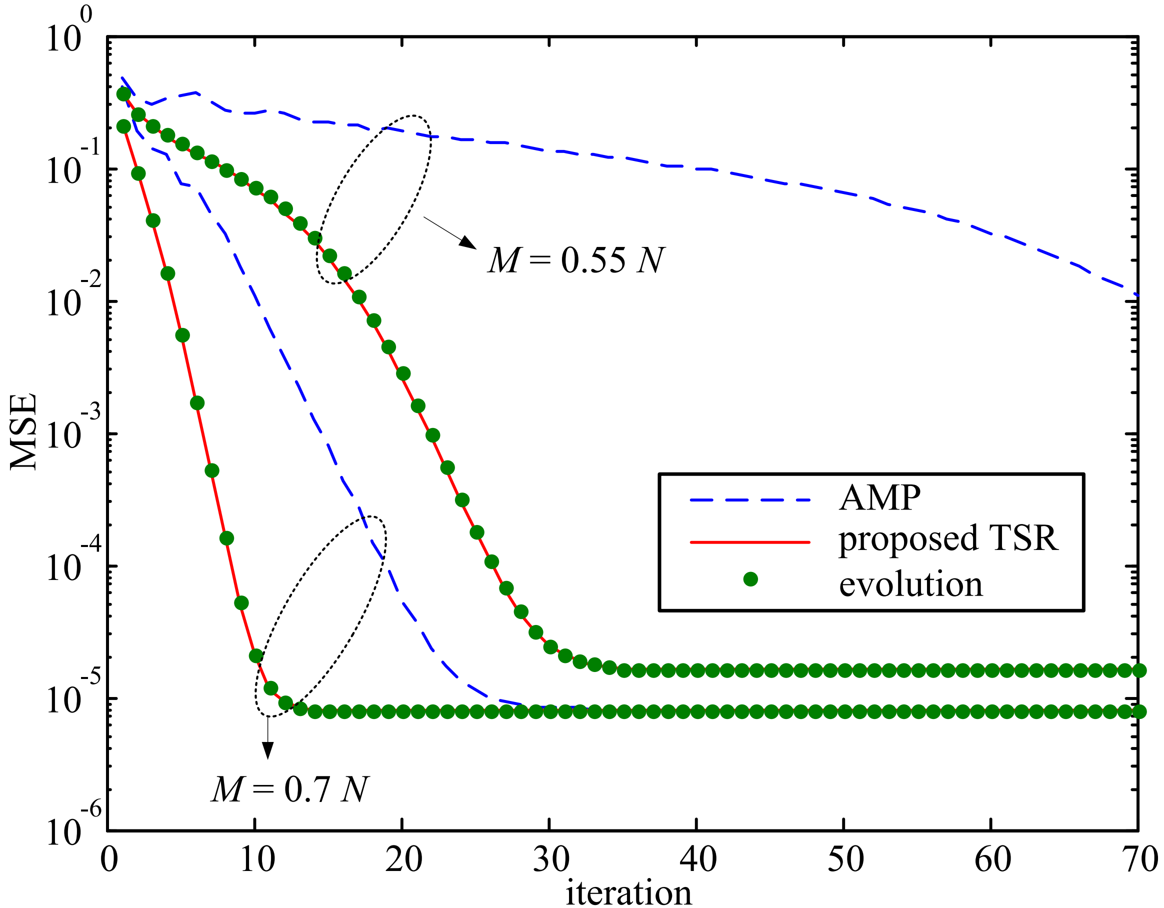
$$\rho^t = \frac{M}{N} \cdot \frac{1}{v^t + \sigma^2}$$

$$v^{t+1} = mmse(\rho^t)$$

AMP for iid **A**

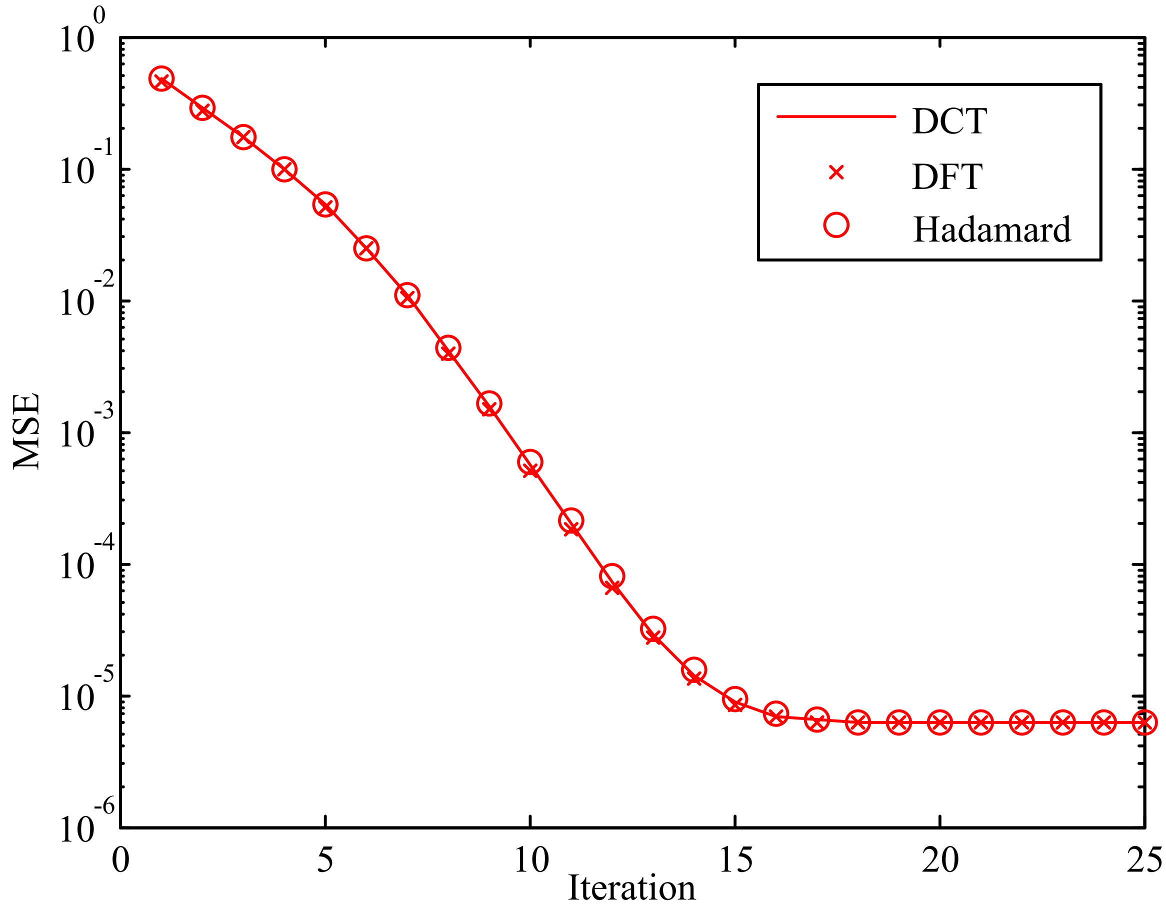
- Turbo CS outperforms AMP in every iteration.
- The advantage is partly due to the use of different sensing matrices: an orthogonal matrix is better conditioned.

Noisy MSE Performance



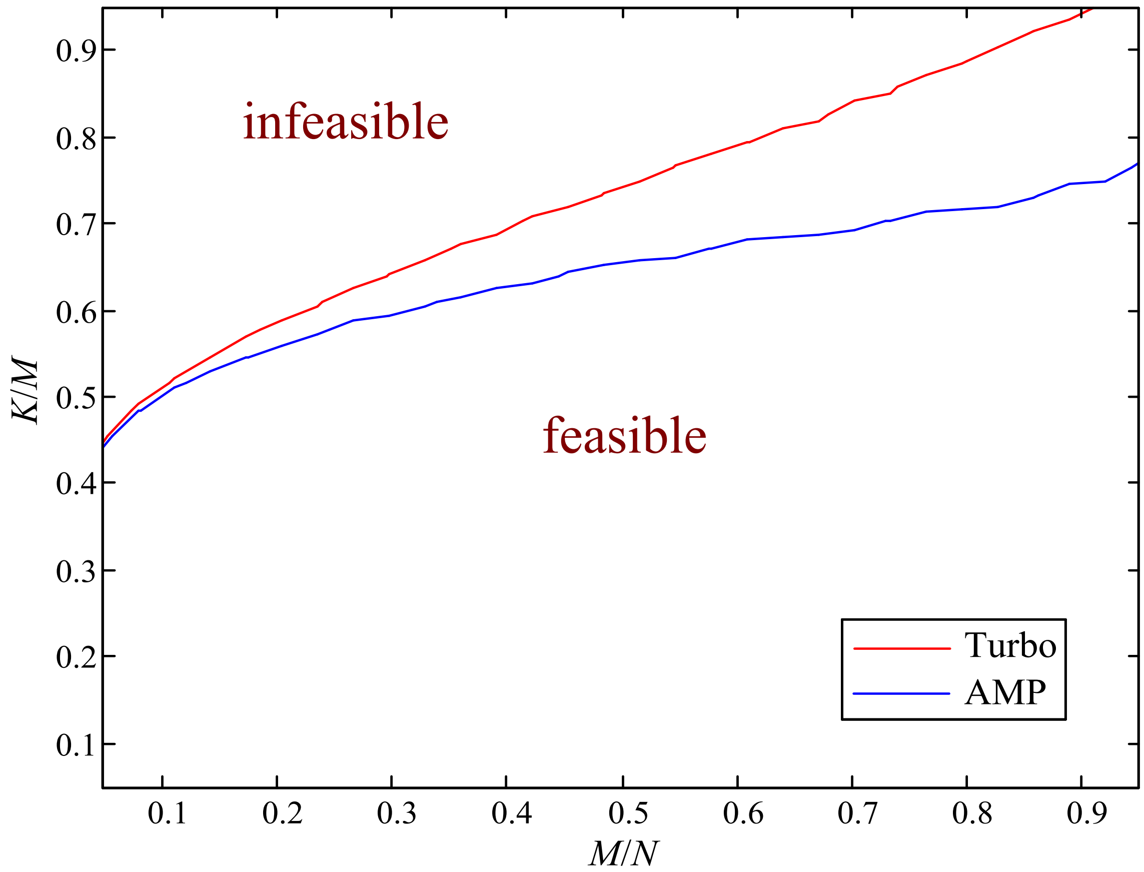
Bernulli-Gaussian prior. Sparsity level = 0.4. SNR = 50dB.

State Evolution for Different Sensing Matrix A



Bernulli-Gaussian prior. Sparsity level = 0.1. $M = 0.25N$. SNR = 50dB.

Noiseless Empirical Phase Transition: 50 Iterations



Partial DFT matrix; Bernulli-Gaussian prior; $N = 8192$;

200 realization. In each realization, success if MSE < 10^{-6}; contour average success rate = 0.5

Conclusions

Summary

- Proposed the turbo compressed sensing (CS) algorithm
- Established one-letter state evolution for turbo CS
- Showed by state evolution that orthogonal sensing with turbo CS always outperforms iid sensing with AMP
- Showed that turbo CS achieves the optimal MMSE predicted by the replica method
- Demonstrated that turbo CS outperforms AMP when both involves orthogonal sensing matrices

References

[Donoho09] Donoho, David L., Arian Maleki, and Andrea Montanari. “Message-passing algorithms for compressed sensing,” *Proceedings of the National Academy of Sciences* 106.45 (2009): 18914-18919.

[Bayati11] M. Bayati and A. Montanari, “The dynamics of message passing on dense graphs, with applications to compressed sensing,” *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 764–785, Feb. 2011.

[Krzakala10] F. Krzakala, M. Mézard, F. Sausset, Y. Sun, and L. Zdeborová, “Probabilistic reconstruction in compressed sensing: algorithms, phase diagrams, and threshold achieving matrices,” J. Statist. Mech.-Theory Exp., 2012.

[Rangan15] Rangan, Sundeep, et al. "Inference for generalized linear models via alternating directions and Bethe free energy minimization." arXiv preprint arXiv:1501.01797 (2015).

[Donoho14] D. Donoho, A. Javanmard, A. Montanari, “Information-theoretically optimal compressed sensing via spatial coupling and approximate message passing,” *IEEE Trans. Information Theory*, Dec. 2014.

[Berrou93] Berrou, Claude, and Alain Glavieux. "Near optimum error correcting coding and decoding: Turbo-codes." *Communications, IEEE Transactions on* 44.10 (1996): 1261-1271.

[Loeliger07] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. Kschischang, “The factor graph approach to model-based signal processing,” *Proc. IEEE*, vol. 95, no. 6, pp. 1295–1322, Jun. 2007.

[Tulino13] Tulino, Antonia M., et al. "Support recovery with sparsely sampled free random matrices." *Information Theory, IEEE Transactions on* 59.7 (2013): 4243-4271.

Thank you!

http://sist.shanghaitech.edu.cn/faculty/yuanxj/

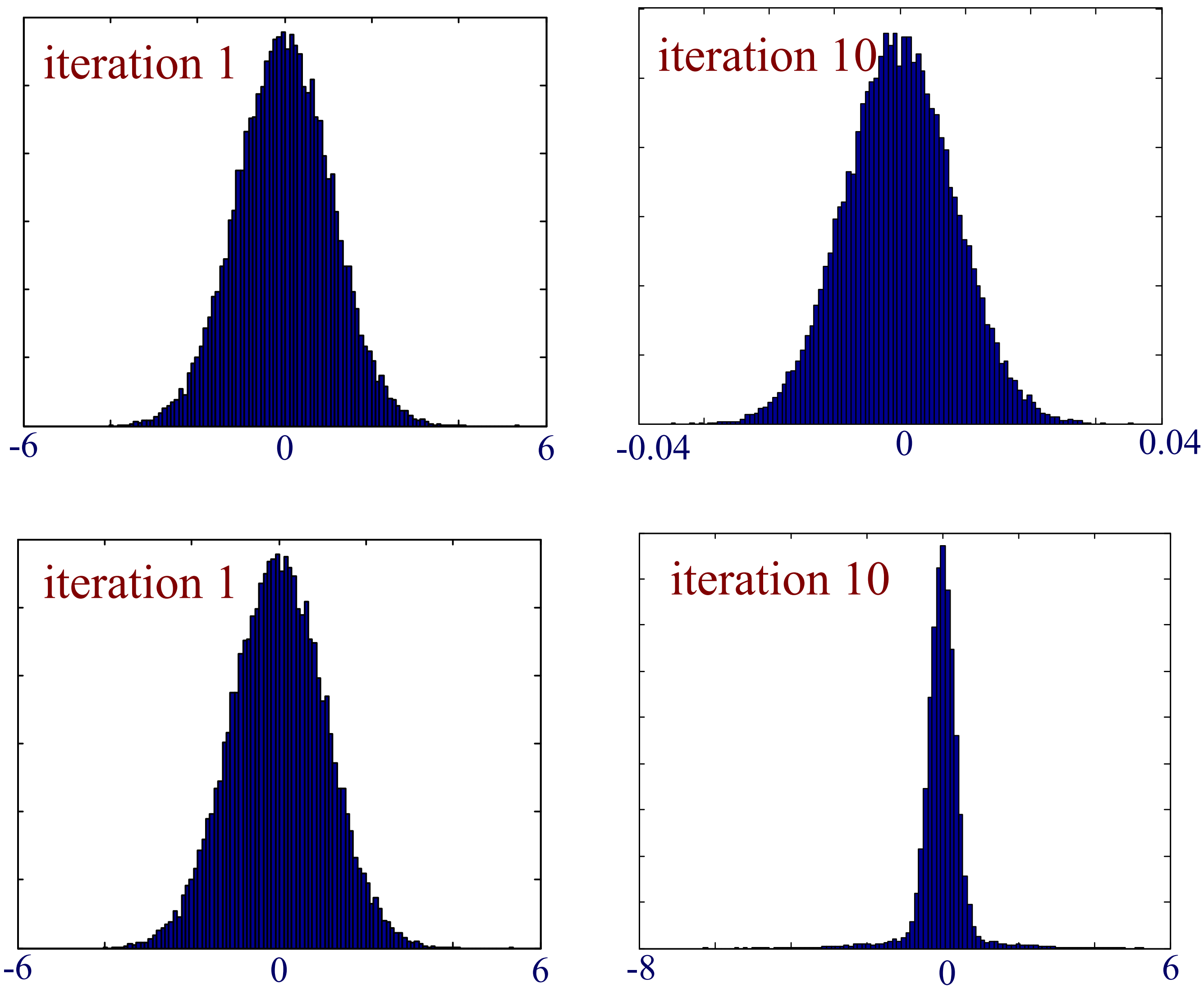
Gaussian Distortion Assumption

Module B

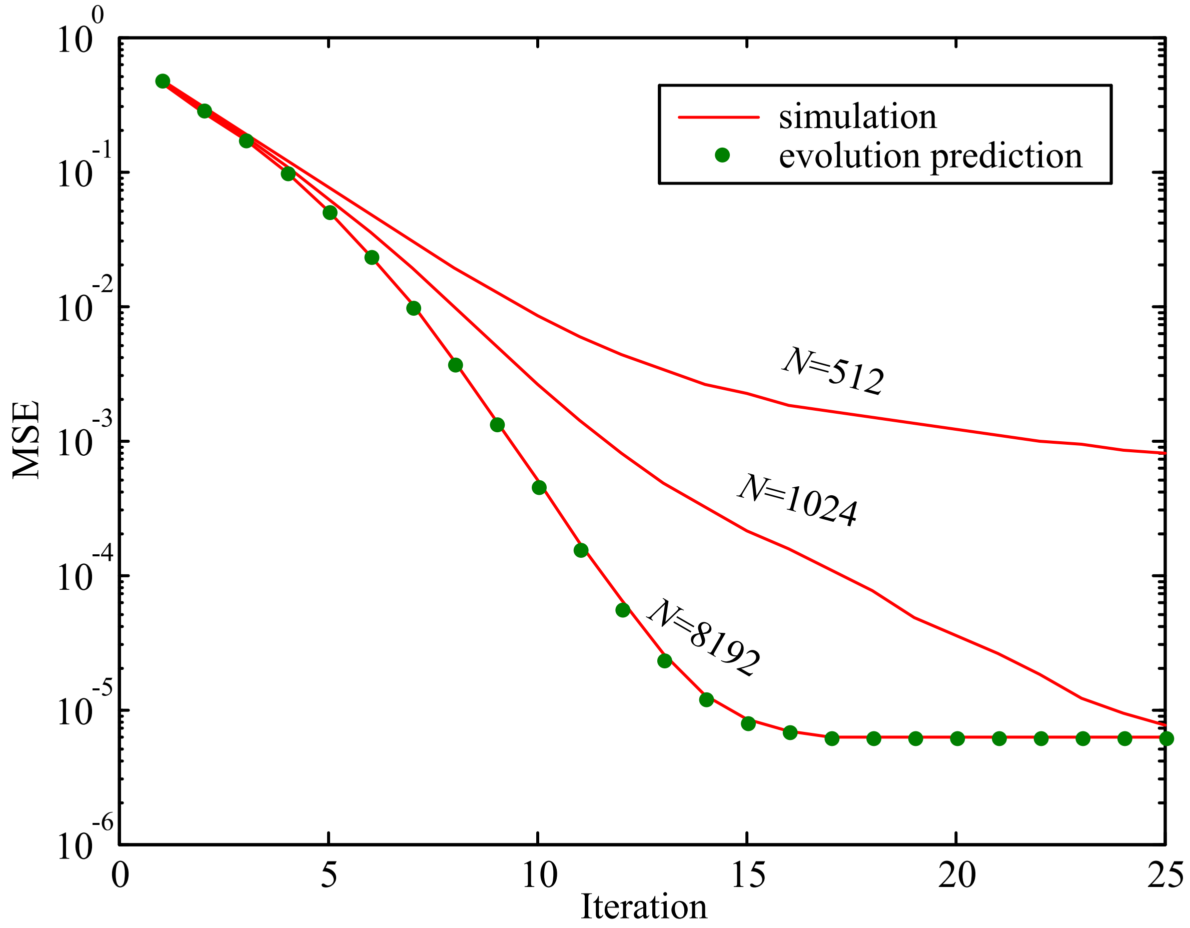
extrinsic

Module B

direct feedback



State Evolution and Simulation



Bernulli-Gaussian prior. Sparsity level = 0.1. $M = 0.25N$. SNR = 50dB.

Future Work

- Rigorous proof of the convergence of turbo CS
- Extension to non-Bayesian settings without requiring priors
- Turbo CS for matrix completion, phase retrieval, etc
- Applications
- ……