IEEE ICASSP 2018 (April 15-20, 2018) @Calgary, Canada
Lecture Session SPTM-L7.2: Signal Processing on Networks

Distributed Approximate Message Passing with Summation Propagation

Ryo Hayakawa (Kyoto University, Japan)
Ayano Nakai (Kyoto University, Japan)
Kazunori Hayashi (Osaka City University, Japan)

Outline

1. Introduction

2. Preliminaries
   1. AMP Algorithm
   2. Consensus Propagation

3. Proposed Method: Distributed AMP Algorithm

4. Simulation Result

5. Conclusion

Outline

1. **Introduction**
2. Preliminaries
    1. AMP Algorithm
    2. Consensus Propagation
3. Proposed Method: Distributed AMP Algorithm
4. Simulation Result
5. Conclusion

Compressed Sensing [1]

reconstruct a **sparse** vector $\boldsymbol{x} \in \mathbb{R}^N$
from its **underdetermined** linear measurement $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{v} \in \mathbb{R}^M \, (M < N)$

$\boldsymbol{x} \in \mathbb{R}^N$: unknown sparse vector (most elements are zero)

$\boldsymbol{A} \in \mathbb{R}^{M \times N}$: measurement matrix $(M < N)$

$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \underbrace{\boldsymbol{v}}_{\text{noise vector}} \in \mathbb{R}^M$: measurement vector

Application

✦ magnetic resonance imaging (MRI) [2]
✦ wireless channel estimation [3]

---

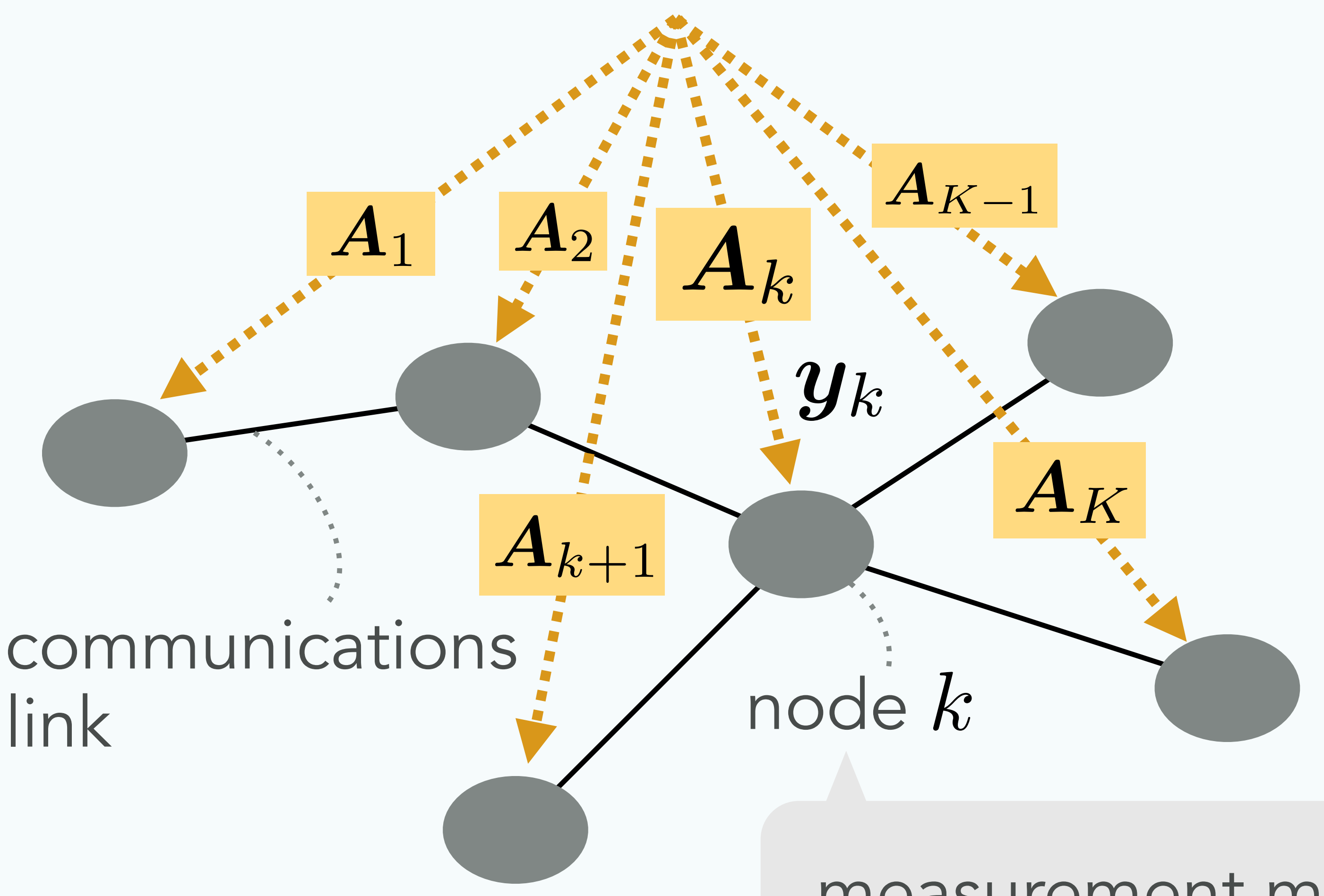
[1] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[2] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly, "Compressed sensing MRI," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 72–82, Mar. 2008.

[3] K. Hayashi, M. Nagahara, and T. Tanaka, "A user's guide to compressed sensing for communications systems," *IEICE Trans. Commun.*, vol. E96-B, no. 3, pp. 685–712, Mar. 2013.

Distributed Compressed Sensing

unknown sparse vector $\boldsymbol{x} \in \mathbb{R}^N$



communications link

node k

Application [4]

- ✦ sensor network
- ✦ video coding
- ✦ image fusion

$$\left(\sum_{k=1}^{K} M_k < N\right)$$

measurement matrix: $\boldsymbol{A}_k \in \mathbb{R}^{M_k \times N}$

measurement vector: $\boldsymbol{y}_k = \boldsymbol{A}_k \boldsymbol{x} + \boldsymbol{v}_k \in \mathbb{R}^{M_k}$

reconstruct $\boldsymbol{x}$ from $\boldsymbol{y}_k, \boldsymbol{A}_k \ (k = 1, \ldots, K)$

---

[4] H. Yin, J. Li, Y. Chai, and S. X. Yang, "A survey on distributed compressed sensing: theory and applications," *Frontiers of Computer Science*, vol. 8, no. 6, pp. 893–904, Dec. 2014.

Conventional Methods (1/2)

✦ D-LASSO [5]

(**D**istributed-**L**east **A**bsolute **S**hrinkage and **S**election **O**perator)

✦ D-ADMM [6]

(**D**istributed-**A**lternating **D**irection **M**ethod of **M**ultipliers)

- The computational complexity might be large

✦ D-IHT [7]

(**D**istributed-**I**terative **H**ard **T**hresholding)

✓ Each node performs simple calculations

- The sparsity level is required

---

[5] J. A. Bazerque and G. B. Giannakis, "Distributed spectrum sensing for cognitive radio networks by exploiting sparsity," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1847–1862, Mar. 2010.

[6] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Püschel, "Distributed basis pursuit," *IEEE Trans. Signal Process.*, vol. 60, no. 4, pp. 1942–1956, Apr. 2012.

[7] S. Patterson, Y. C. Eldar, and I. Keidar, "Distributed sparse signal recovery for sensor networks," in *Proc. IEEE ICASSP*, May 2013.

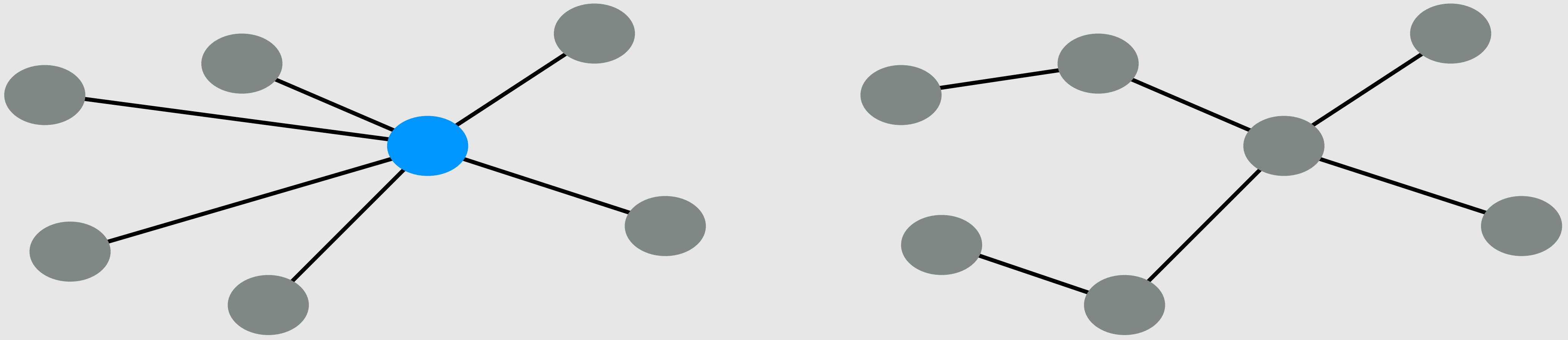
Conventional Methods (2/2)

✦ Distributed AMP [8], Multi-processor AMP [9]

(**A**pproximate **M**essage **P**assing)

✓ Each node performs simple calculations
✓ The sparsity level is **not** required
- A **fusion node** communicating with all nodes is required

- much energy at fusion node
- vulnerable to failure of node



applicable **not applicable**

---

[8] P. Han, R. Niu, M. Ren, and Y. C. Eldar, "Distributed approximate message passing for sparse signal recovery," in *Proc. IEEE GlobalSIP*, Dec. 2014.

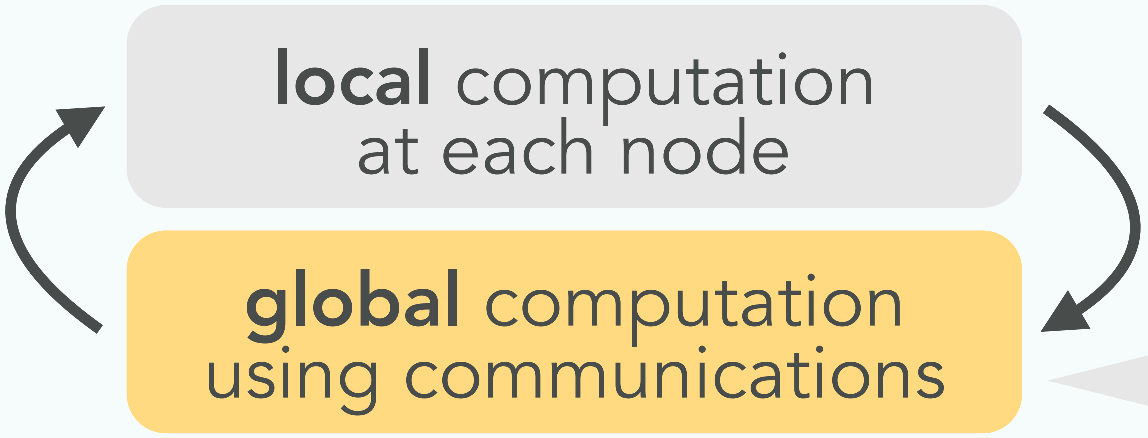
[9] J. Zhu, R. Pilgrim, and D. Baron, "An overview of multi-processor approximate message passing," in *Proc. IEEE CISS*, Mar. 2017.

Summary of This Study

Purpose of This Study

propose a **fully distributed** AMP algorithm,
which does not require any fusion node

① obtain update equations of the AMP algorithm
for distributed measurements



local computation
at each node

global computation
using communications

② propose **summation propagation**
for the global computation

③ show the validity of the proposed algorithm via computer simulation

Outline

1. Introduction
2. **Preliminaries**
   i. **AMP Algorithm**
   ii. **Consensus Propagation**
3. Proposed Method: Distributed AMP Algorithm
4. Simulation Result
5. Conclusion

AMP Algorithm (1/2)

$\boldsymbol{x} \in \mathbb{R}^N$: unknown sparse vector

$\boldsymbol{A} \in \mathbb{R}^{M \times N}$: measurement matrix $(M < N)$
i.i.d. elements with zero mean and unit variance

$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{v} \in \mathbb{R}^M$: measurement vector

ℓ_1 optimization

$$\hat{\boldsymbol{x}} = \arg \min_{\boldsymbol{z} \in \mathbb{R}^N} \|\boldsymbol{z}\|_1 \text{ subject to } \boldsymbol{y} = \boldsymbol{A}\boldsymbol{z}$$

approximate belief propagation for $\hat{\boldsymbol{x}}$ [10]

AMP Algorithm [10, 11]
(**A**pproximate **M**essage **P**assing)

- ✓ low complexity (no matrix inversion, $O(MN)$)
- ✓ asymptotic analysis

[10] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. motivation and construction," in *Proc. IEEE Inf. Theory Workshop*, Jan. 2010.

[11] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing," *Proc. Nat. Acad. Sci.*, vol. 106, no. 45, pp. 18 914–18 919, Nov. 2009.

AMP Algorithm (2/2)

$\boldsymbol{x} \in \mathbb{R}^N$: unknown sparse vector

$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{v} \in \mathbb{R}^M$: measurement vector

① Initialization: $t = 1, \hat{\boldsymbol{x}}(1) = \mathbf{0}, \boldsymbol{s}(0) = \mathbf{0}, \boldsymbol{r}(0) = \mathbf{0}, \hat{\sigma}^2(0) = 0$

② $\boldsymbol{s}(t) = \boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}(t) + \frac{1}{\Delta}\boldsymbol{s}(t-1)\left\langle \eta'\left(\boldsymbol{r}(t-1); \hat{\sigma}^2(t-1)\right)\right\rangle$

$t \leftarrow t+1$

$\Delta = M/N$: measurement ratio

$\langle \cdot \rangle$: mean

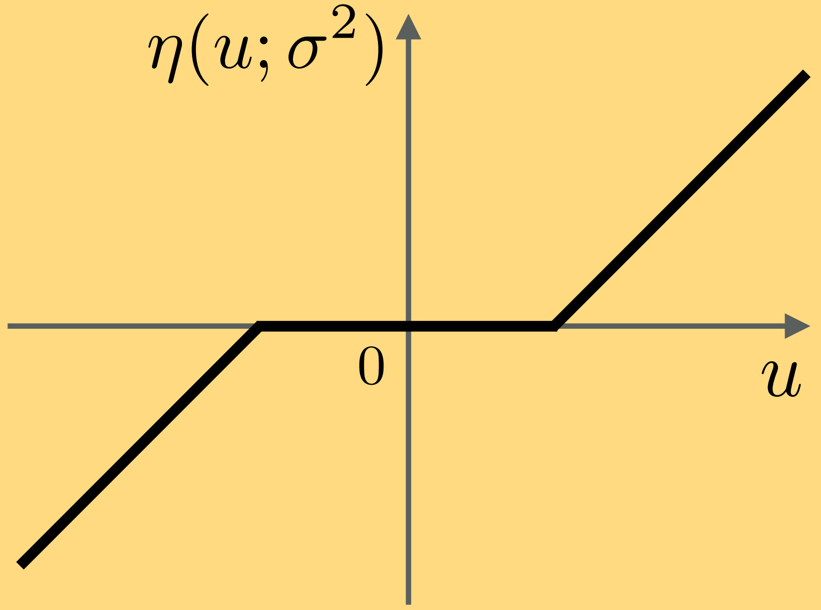
③ $\boldsymbol{r}(t) = \hat{\boldsymbol{x}}(t) + \frac{1}{M}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{s}(t)$

④ $\hat{\sigma}^2(t) = \frac{\|\boldsymbol{s}(t)\|_2^2}{MN}$

⑤ $\hat{\boldsymbol{x}}(t+1) = \eta\left(\boldsymbol{r}(t); \hat{\sigma}^2(t)\right)$

estimate of $\boldsymbol{x}$

example of $\eta(\cdot;\cdot)$: soft thresholding



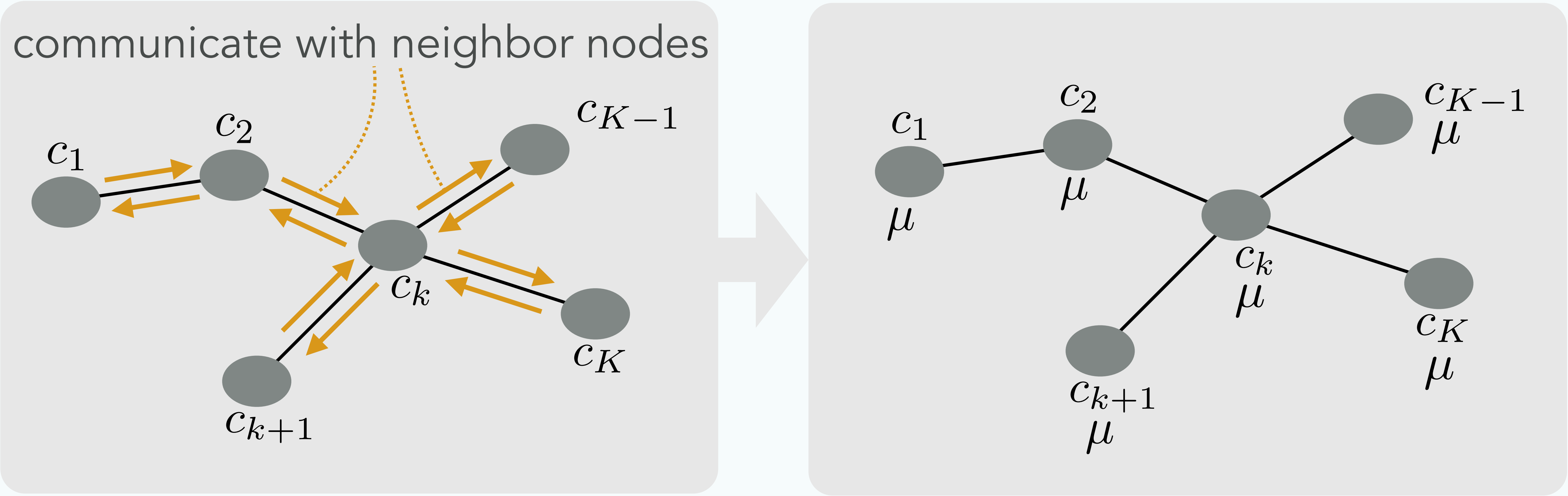
Consensus Propagation [12] (1/2)

A distributed algorithm for **average consensus** on undirected graphs

All nodes obtain the mean $\mu = \frac{1}{K}\sum_{k=1}^{K} c_k$

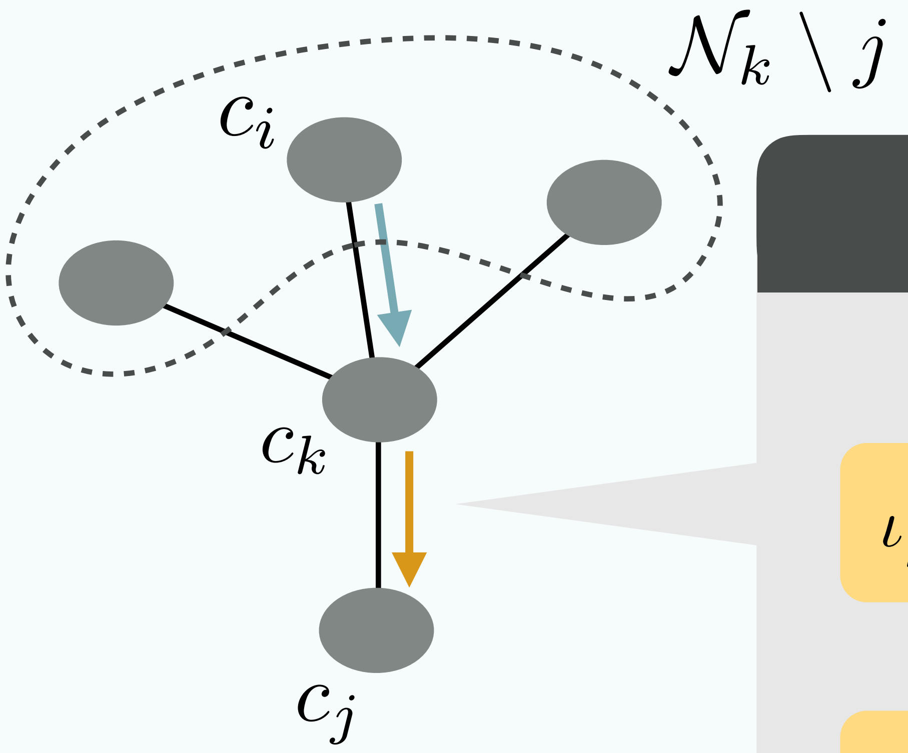
number of nodes

initial value at node k


communicate with neighbor nodes

[12] C. C. Moallemi and B. V. Roy, "Consensus propagation," *IEEE Trans. Inf. Theory*, vol. 52, no. 11, pp. 4753–4766, Nov. 2006.

Consensus Propagation [12] (2/2)



message from node k to node j

message from other nodes at the previous iteration

$$\iota_{k\to j}^{(t')} = 1 + \sum_{i\in\mathcal{N}_k\setminus j} \iota_{i\to k}^{(t'-1)} \qquad \left(\iota_{k\to j}^{(0)} = 0\right)$$

$$\nu_{k\to j}^{(t')} = \frac{c_k + \sum_{i\in\mathcal{N}_k\setminus j}\iota_{i\to k}^{(t'-1)}\nu_{i\to k}^{(t'-1)}}{1+\sum_{i\in\mathcal{N}_k\setminus j}\iota_{i\to k}^{(t'-1)}} \qquad \left(\nu_{k\to j}^{(0)} = 0\right)$$

- The graph is a tree
- # of iterations $\geq$ graph diameter

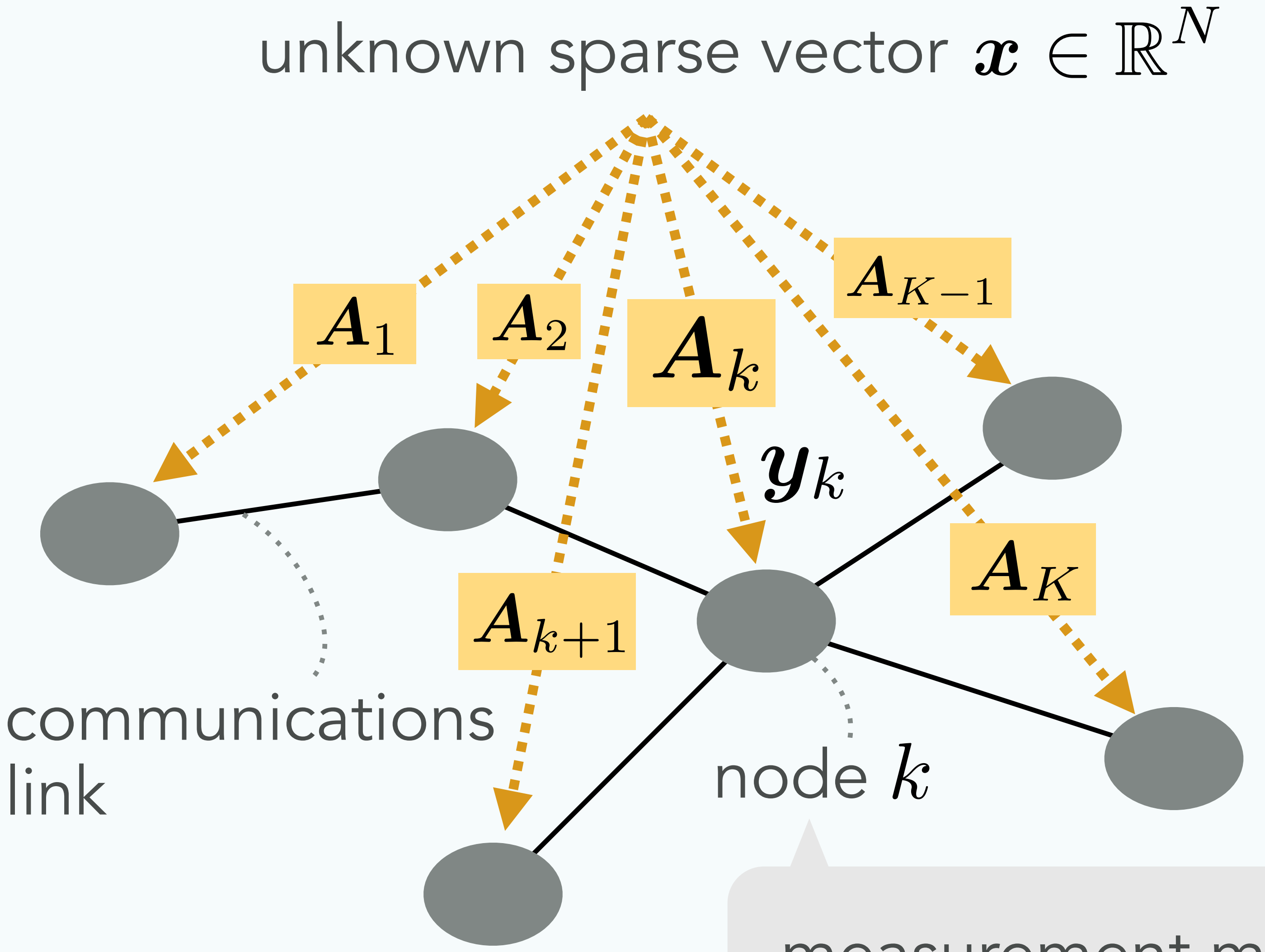
→ average consensus is achieved

[12] C. C. Moallemi and B. V. Roy, "Consensus propagation," *IEEE Trans. Inf. Theory*, vol. 52, no. 11, pp. 4753–4766, Nov. 2006.

Outline

1. Introduction
2. Preliminaries
    i. AMP Algorithm
    ii. Consensus Propagation
3. **Proposed Method: Distributed AMP Algorithm**
4. Simulation Result
5. Conclusion

Distributed Compressed Sensing



All measurements $\boldsymbol{y}_1, \dots, \boldsymbol{y}_K$ can be combined as

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_K \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \vdots \\ \boldsymbol{A}_K \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_K \end{bmatrix}$$

measurement matrix: $\boldsymbol{A}_k \in \mathbb{R}^{M_k \times N}$

measurement vector: $\boldsymbol{y}_k = \boldsymbol{A}_k \boldsymbol{x} + \boldsymbol{v}_k \in \mathbb{R}^{M_k}$

reconstruct $\boldsymbol{x}$ from $\boldsymbol{y}_k, \boldsymbol{A}_k \ (k = 1, \dots, K)$

AMP Algorithm for Distributed Model (1/2)

centralized model

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{v}$$

AMP algorithm

$$\boldsymbol{s}(t) = \boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}(t) + \frac{1}{\Delta}\boldsymbol{s}(t-1)\left\langle \eta'\left(\boldsymbol{r}(t-1);\hat{\sigma}^2(t-1)\right)\right\rangle$$

$$\boldsymbol{r}(t) = \hat{\boldsymbol{x}}(t) + \frac{1}{M}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{s}(t)$$

$$\hat{\sigma}^2(t) = \frac{\|\boldsymbol{s}(t)\|_2^2}{MN}$$

$$\hat{\boldsymbol{x}}(t+1) = \eta\left(\boldsymbol{r}(t);\hat{\sigma}^2(t)\right)$$

distributed model

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_K \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \vdots \\ \boldsymbol{A}_K \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_K \end{bmatrix}$$

AMP Algorithm

$$\boldsymbol{s}_k(t) = \boldsymbol{y}_k - \boldsymbol{A}_k\hat{\boldsymbol{x}}(t) + \frac{1}{\Delta}\boldsymbol{s}_k(t-1)\left\langle \eta'\left(\boldsymbol{r}(t-1);\hat{\sigma}^2(t-1)\right)\right\rangle$$

$$\boldsymbol{r}(t) = \sum_{k=1}^{K}\left(\frac{1}{K}\hat{\boldsymbol{x}}(t) + \frac{1}{M}\boldsymbol{A}_k^{\mathrm{T}}\boldsymbol{s}_k(t)\right)$$

$$\hat{\sigma}^2(t) = \sum_{k=1}^{K}\frac{\|\boldsymbol{s}_k(t)\|_2^2}{MN}$$

$$\hat{\boldsymbol{x}}(t+1) = \eta\left(\boldsymbol{r}(t);\hat{\sigma}^2(t)\right)$$

AMP Algorithm for Distributed Model (1/2)

centralized model

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{v}$$

AMP algorithm

$$\boldsymbol{s}(t) = \boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}(t) + \frac{1}{\Delta}\boldsymbol{s}(t-1)\left\langle \eta'\left(\boldsymbol{r}(t-1); \hat{\sigma}^2(t-1)\right)\right\rangle$$

$$\boldsymbol{r}(t) = \hat{\boldsymbol{x}}(t) + \frac{1}{M}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{s}(t)$$

$$\hat{\sigma}^2(t) = \frac{\|\boldsymbol{s}(t)\|_2^2}{MN}$$

$$\hat{\boldsymbol{x}}(t+1) = \eta\left(\boldsymbol{r}(t); \hat{\sigma}^2(t)\right)$$

$$\boldsymbol{s}(t) = \begin{bmatrix} \boldsymbol{s}_1(t) \\ \vdots \\ \boldsymbol{s}_K(t) \end{bmatrix}$$

distributed model

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_K \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \vdots \\ \boldsymbol{A}_K \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_K \end{bmatrix}$$

AMP Algorithm

$$\boldsymbol{s}_k(t) = \boldsymbol{y}_k - \boldsymbol{A}_k\hat{\boldsymbol{x}}(t) + \frac{1}{\Delta}\boldsymbol{s}_k(t-1)\left\langle \eta'\left(\boldsymbol{r}(t-1); \hat{\sigma}^2(t-1)\right)\right\rangle$$

$$\boldsymbol{r}(t) = \sum_{k=1}^{K}\left(\frac{1}{K}\hat{\boldsymbol{x}}(t) + \frac{1}{M}\boldsymbol{A}_k^{\mathrm{T}}\boldsymbol{s}_k(t)\right)$$

$$\hat{\sigma}^2(t) = \sum_{k=1}^{K}\frac{\|\boldsymbol{s}_k(t)\|_2^2}{MN}$$

$$\hat{\boldsymbol{x}}(t+1) = \eta\left(\boldsymbol{r}(t); \hat{\sigma}^2(t)\right)$$

AMP Algorithm for Distributed Model (1/2)

centralized model

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{v}$$

AMP algorithm

$$\boldsymbol{s}(t) = \boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}(t) + \frac{1}{\Delta}\boldsymbol{s}(t-1)\left\langle \eta'\left(\boldsymbol{r}(t-1); \hat{\sigma}^2(t-1)\right)\right\rangle$$

$$\boldsymbol{r}(t) = \hat{\boldsymbol{x}}(t) + \frac{1}{M}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{s}(t)$$

$$\hat{\sigma}^2(t) = \frac{\|\boldsymbol{s}(t)\|_2^2}{MN}$$

$$\hat{\boldsymbol{x}}(t+1) = \eta\left(\boldsymbol{r}(t); \hat{\sigma}^2(t)\right)$$

$$\boldsymbol{s}(t) = \begin{bmatrix} \boldsymbol{s}_1(t) \\ \vdots \\ \boldsymbol{s}_K(t) \end{bmatrix}$$

distributed model

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_K \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \vdots \\ \boldsymbol{A}_K \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_K \end{bmatrix}$$

AMP Algorithm

$$\boldsymbol{s}_k(t) = \boldsymbol{y}_k - \boldsymbol{A}_k\hat{\boldsymbol{x}}(t) + \frac{1}{\Delta}\boldsymbol{s}_k(t-1)\left\langle \eta'\left(\boldsymbol{r}(t-1); \hat{\sigma}^2(t-1)\right)\right\rangle$$

$$\boldsymbol{r}(t) = \sum_{k=1}^{K}\left(\frac{1}{K}\hat{\boldsymbol{x}}(t) + \frac{1}{M}\boldsymbol{A}_k^{\mathrm{T}}\boldsymbol{s}_k(t)\right)$$

$$\hat{\sigma}^2(t) = \sum_{k=1}^{K}\frac{\|\boldsymbol{s}_k(t)\|_2^2}{MN}$$

$$\hat{\boldsymbol{x}}(t+1) = \eta\left(\boldsymbol{r}(t); \hat{\sigma}^2(t)\right)$$

AMP Algorithm for Distributed Model (1/2)

centralized model

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{v}$$

AMP algorithm

$$\boldsymbol{s}(t) = \boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}(t) + \frac{1}{\Delta}\boldsymbol{s}(t-1)\left\langle \eta'\left(\boldsymbol{r}(t-1);\hat{\sigma}^2(t-1)\right)\right\rangle$$

$$\boldsymbol{r}(t) = \hat{\boldsymbol{x}}(t) + \frac{1}{M}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{s}(t)$$

$$\hat{\sigma}^2(t) = \frac{\|\boldsymbol{s}(t)\|_2^2}{MN}$$

$$\hat{\boldsymbol{x}}(t+1) = \eta\left(\boldsymbol{r}(t);\hat{\sigma}^2(t)\right)$$

$$\boldsymbol{s}(t) = \begin{bmatrix} \boldsymbol{s}_1(t) \\ \vdots \\ \boldsymbol{s}_K(t) \end{bmatrix}$$

distributed model

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_K \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \vdots \\ \boldsymbol{A}_K \end{bmatrix}\boldsymbol{x} + \begin{bmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_K \end{bmatrix}$$

AMP Algorithm

$$\boldsymbol{s}_k(t) = \boldsymbol{y}_k - \boldsymbol{A}_k\hat{\boldsymbol{x}}(t) + \frac{1}{\Delta}\boldsymbol{s}_k(t-1)\left\langle \eta'\left(\boldsymbol{r}(t-1);\hat{\sigma}^2(t-1)\right)\right\rangle$$

$$\boldsymbol{r}(t) = \sum_{k=1}^{K}\left(\frac{1}{K}\hat{\boldsymbol{x}}(t) + \frac{1}{M}\boldsymbol{A}_k^{\mathrm{T}}\boldsymbol{s}_k(t)\right)$$

$$\hat{\sigma}^2(t) = \sum_{k=1}^{K}\frac{\|\boldsymbol{s}_k(t)\|_2^2}{MN}$$

$$\hat{\boldsymbol{x}}(t+1) = \eta\left(\boldsymbol{r}(t);\hat{\sigma}^2(t)\right)$$

AMP Algorithm for Distributed Model (2/2)

centralized model

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{v}$$

AMP algorithm

$$\boldsymbol{s}(t) = \boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}(t) + \frac{1}{\Delta}\boldsymbol{s}(t-1)\left\langle \eta'\left(\boldsymbol{r}(t-1); \hat{\sigma}^2(t-1)\right)\right\rangle$$

$$\boldsymbol{r}(t) = \hat{\boldsymbol{x}}(t) + \frac{1}{}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{s}(t)$$

cannot be computed locally

$$\hat{\sigma}^2(t) = \frac{\|\boldsymbol{s}(t)\|_2}{MN}$$

$$\hat{\boldsymbol{x}}(t+1) = \eta\left(\boldsymbol{r}(t); \hat{\sigma}^2(t)\right)$$

distributed model

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_K \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \vdots \\ \boldsymbol{A}_K \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_K \end{bmatrix}$$

AMP Algorithm

$$\boldsymbol{s}_k(t) = \boldsymbol{y}_k - \boldsymbol{A}_k\hat{\boldsymbol{x}}(t) + \frac{1}{\Delta}\boldsymbol{s}_k(t-1)\left\langle \eta'\left(\boldsymbol{r}(t-1); \hat{\sigma}^2(t-1)\right)\right\rangle$$

$$\boldsymbol{r}(t) = \sum_{k=1}^{K}\left(\frac{1}{K}\hat{\boldsymbol{x}}(t) + \frac{1}{M}\boldsymbol{A}_k^{\mathrm{T}}\boldsymbol{s}_k(t)\right)$$

$$\hat{\sigma}^2(t) = \sum_{k=1}^{K}\frac{\|\boldsymbol{s}_k(t)\|_2^2}{MN}$$

$$\hat{\boldsymbol{x}}(t+1) = \eta\left(\boldsymbol{r}(t); \hat{\sigma}^2(t)\right)$$

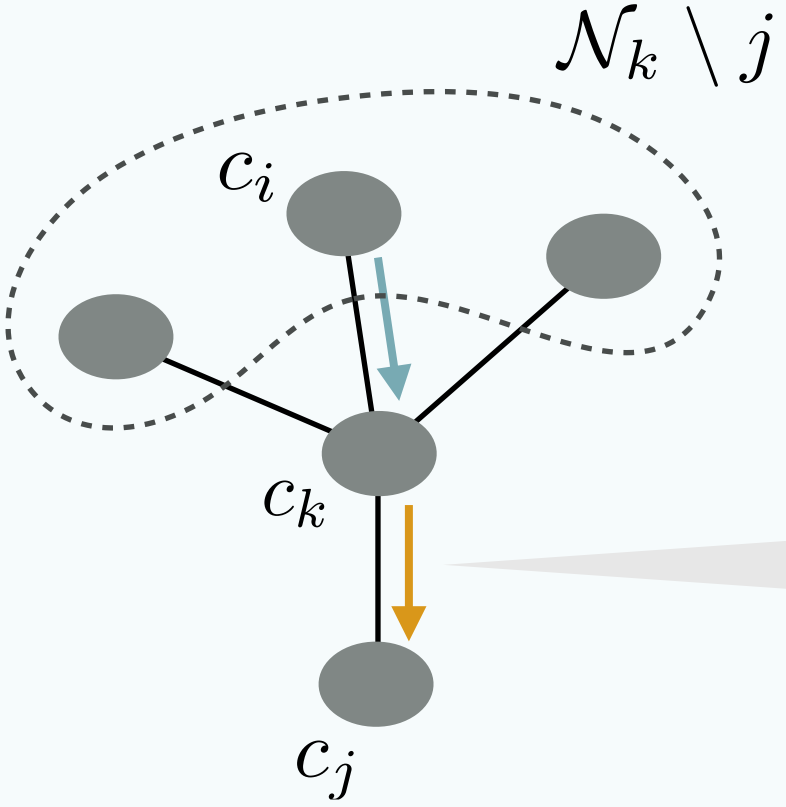
Proposed Method: Distributed AMP Algorithm

Summation Propagation

We propose **summation propagation** to compute

$$\boldsymbol{r}(t) = \sum_{k=1}^{K} \left(\frac{1}{K} \hat{\boldsymbol{x}}(t) + \frac{1}{M} \boldsymbol{A}_k^{\mathrm{T}} \boldsymbol{s}_k(t) \right) \qquad \hat{\sigma}^2(t) = \sum_{k=1}^{K} \frac{\|\boldsymbol{s}_k(t)\|_2^2}{MN}$$

by using the idea of consensus propagation



summation propagation

send the **summation** of c_k and messages from other nodes

$$\boxed{\xi_{k\to j}^{(t')}} = c_k + \sum_{i\in\mathcal{N}_k\setminus j} \boxed{\xi_{i\to k}^{(t'-1)}} \qquad \left(\xi_{k\to j}^{(0)} = 0\right)$$

- The graph is a tree
- # of iterations $\geq$ graph diameter

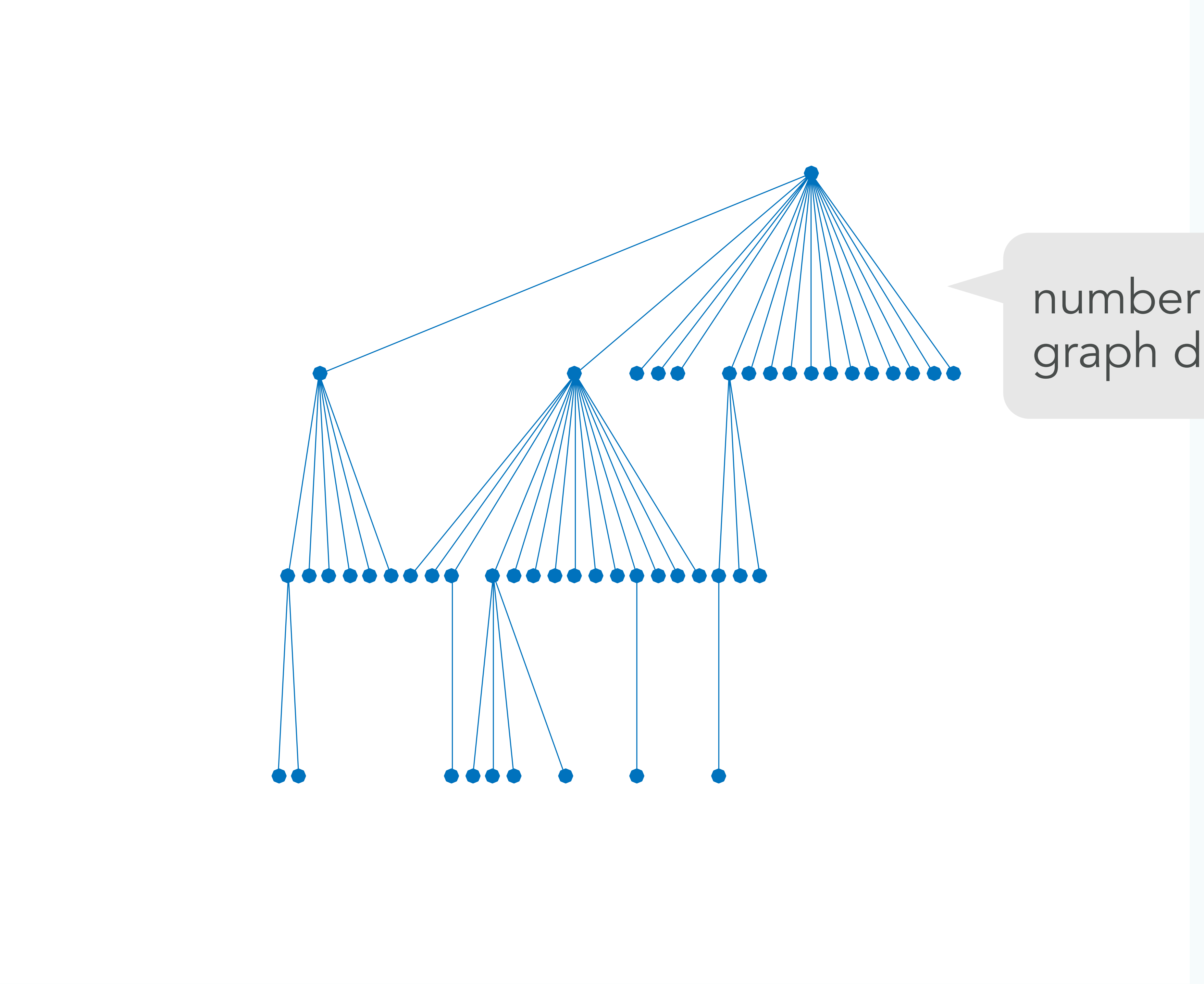
➡ All nodes obtain the summation $\sum_{k=1}^{K} c_k$

Outline

1. Introduction
2. Preliminaries
   1. AMP Algorithm
   2. Consensus Propagation
3. Proposed Method: Distributed AMP Algorithm
4. **Simulation Result**
5. Conclusion

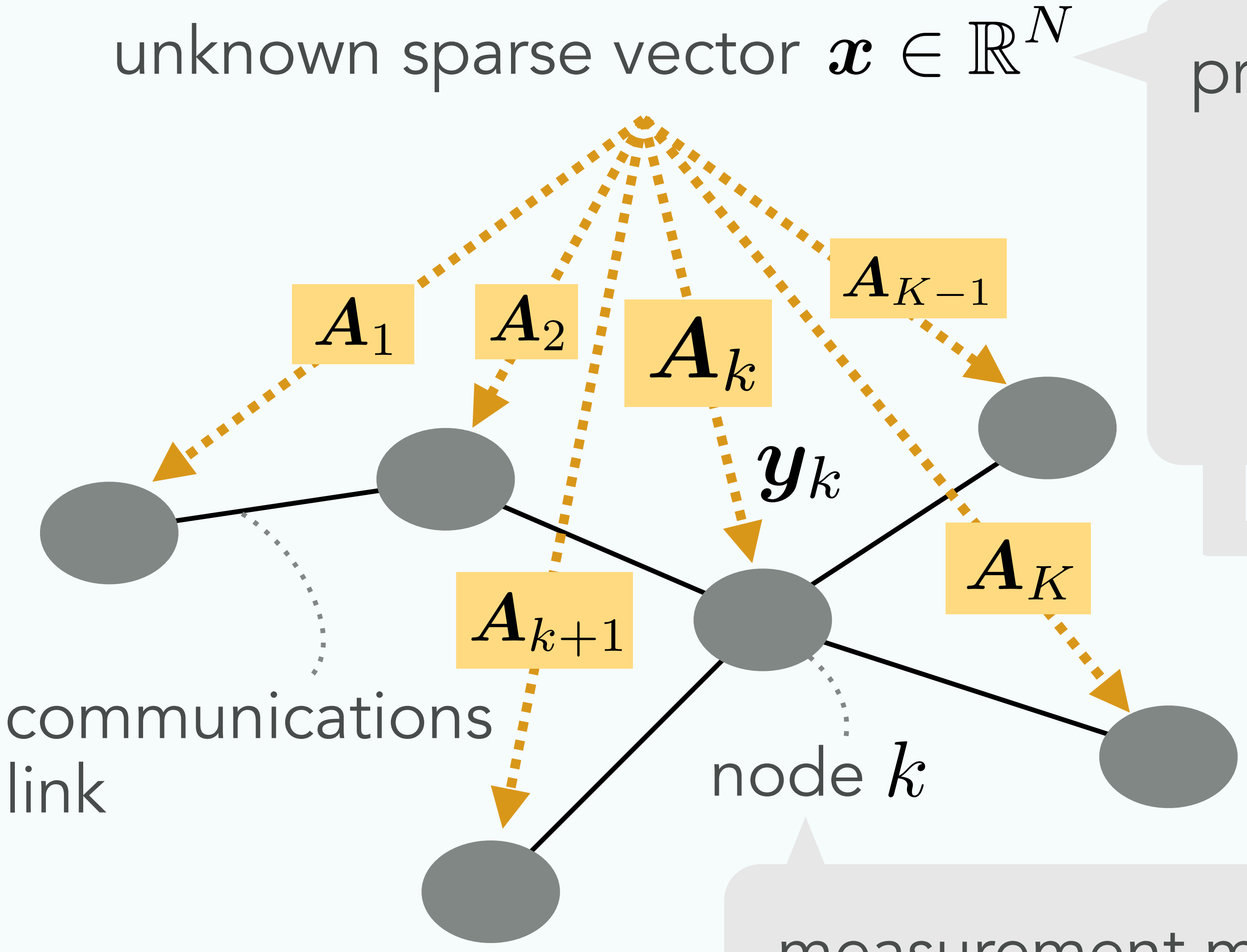
Simulation Result

Graph Structure



number of nodes: $K = 50$
graph diameter: 6

Problem Settings (Sparse Vector Reconstruction)



probability distribution (unknown):

$$p(x_n) = \underbrace{q}_{\text{parameter}}\delta(x_n) + (1-q)\underbrace{\phi}_{\text{probability density function of standard Gaussian distribution}}(x_n)$$

⇒ $\boldsymbol{x}$ is sparse when q is large

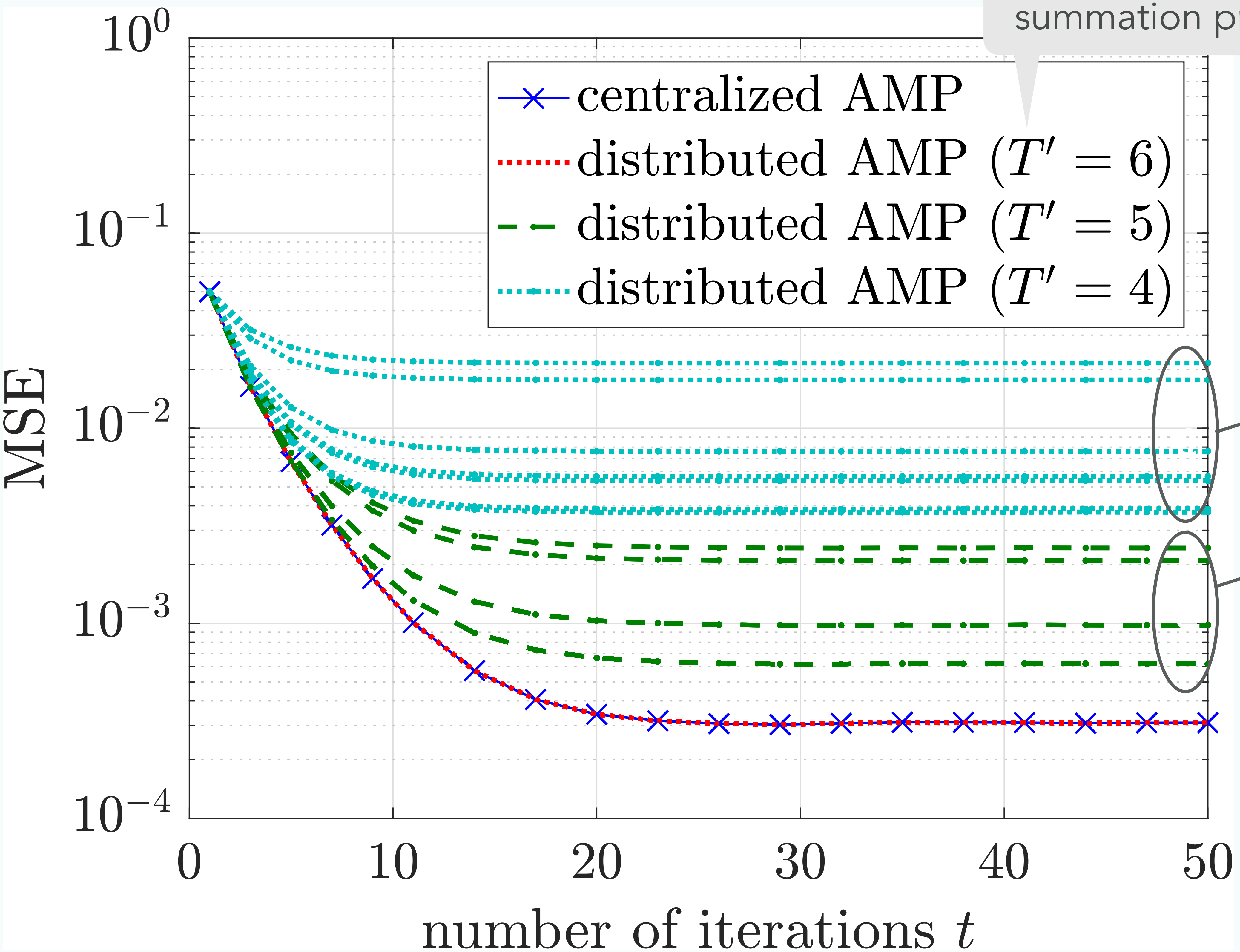
measurement matrix: $\boldsymbol{A}_k \in \mathbb{R}^{M_k \times N}$

measurement vector: $\boldsymbol{y}_k = \boldsymbol{A}_k \boldsymbol{x} + \boldsymbol{v}_k \in \mathbb{R}^{M_k}$

Simulation Result

MSE for Sparse Vector Reconstruction

(Mean-Square-Error)



\# of iterations in summation propagation

$N = 1000$
$M_k = 6$
$\sigma_v^2 = 0.1$
$q = 0.95$

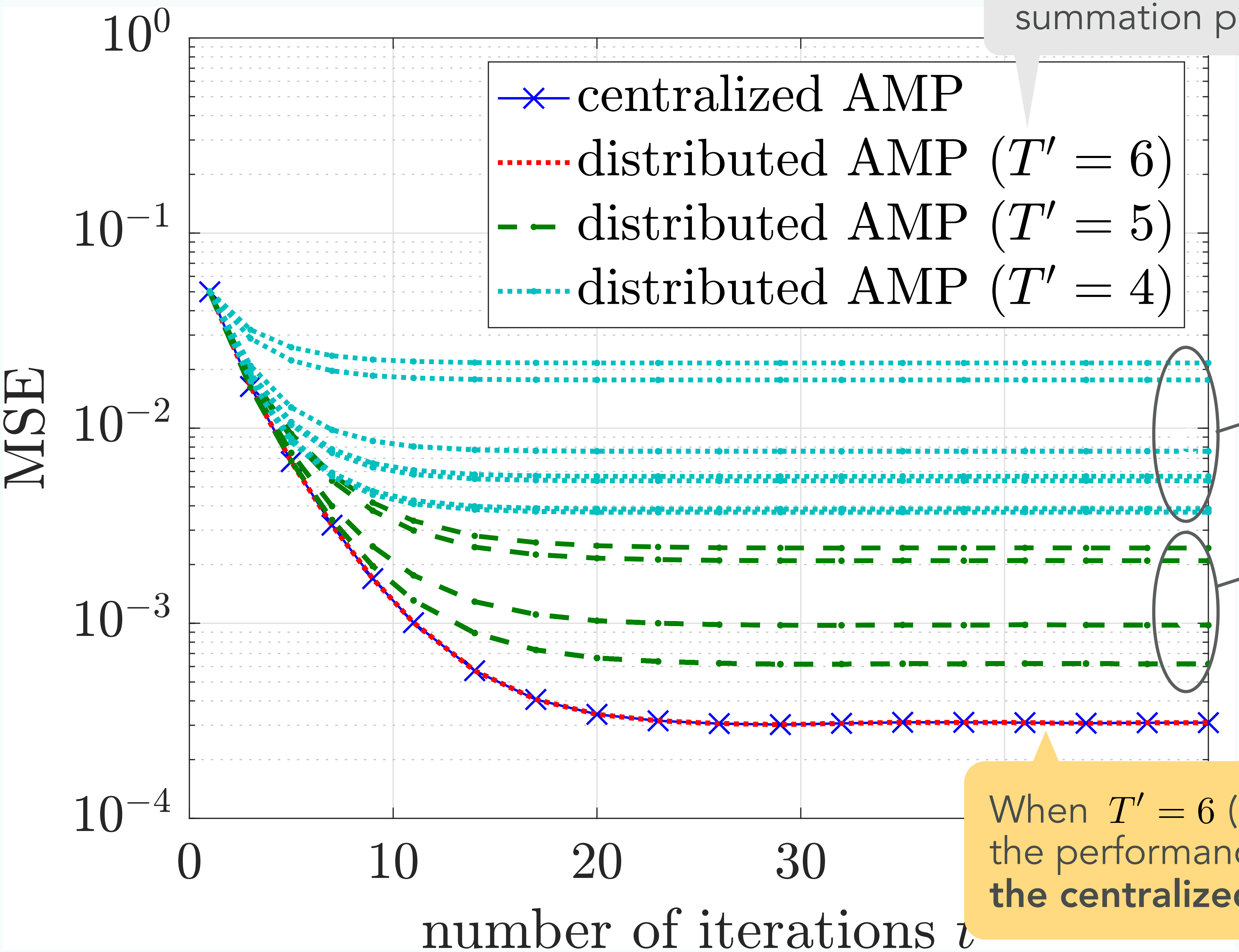
MSEs at $K = 50$ nodes when $T' = 4$

MSEs at $K = 50$ nodes when $T' = 5$

Simulation Result

MSE for Sparse Vector Reconstruction

(Mean-Square-Error)



\# of iterations in summation propagation

$N = 1000$
$M_k = 6$
$\sigma_v^2 = 0.1$
$q = 0.95$

MSEs at $K = 50$ nodes when $T' = 4$

MSEs at $K = 50$ nodes when $T' = 5$

When $T' = 6$ (graph diameter), the performance is **the same as the centralized AMP algorithm**

Outline

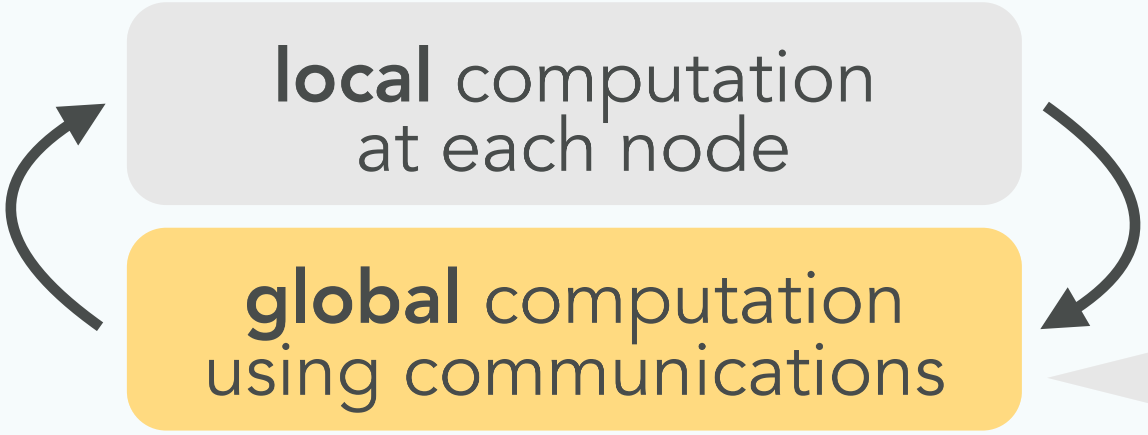
1. Introduction
2. Preliminaries
    1. AMP Algorithm
    2. Consensus Propagation
3. Proposed Method: Distributed AMP Algorithm
4. Simulation Result
5. **Conclusion**

Conclusion

Purpose of This Study

propose a **fully distributed** AMP algorithm,
which does not require any fusion node

① obtain update equations of the AMP algorithm for distributed measurements


local computation at each node

global computation using communications

② propose **summation propagation** for the global computation

③ show the validity of the proposed algorithm via computer simulation

Future Work

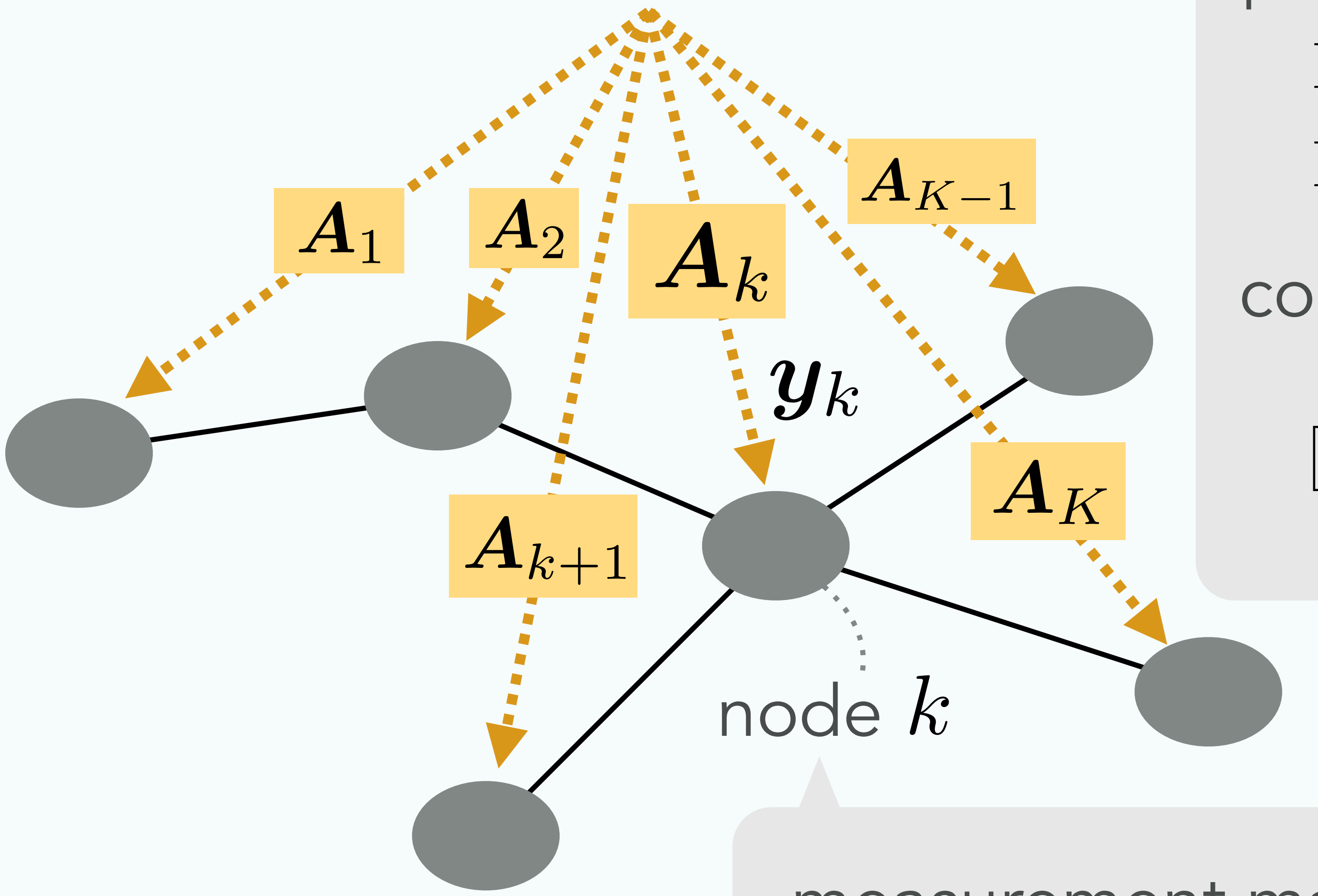
- ✦ extension for generalized AMP algorithm
- ✦ comparison with conventional methods

Appendix

Problem Settings (Binary Vector Reconstruction)

We can apply the AMP algorithm for binary vector reconstruction by using another function as $\eta(\cdot;\cdot)$

unknown vector $\boldsymbol{x} \in \{0,1\}^N$



probability distribution (known):

$$\Pr(x_n = 0) = p_1$$

$$\Pr(x_n = 1) = p_2$$

corresponding $\eta(\cdot;\cdot)$:

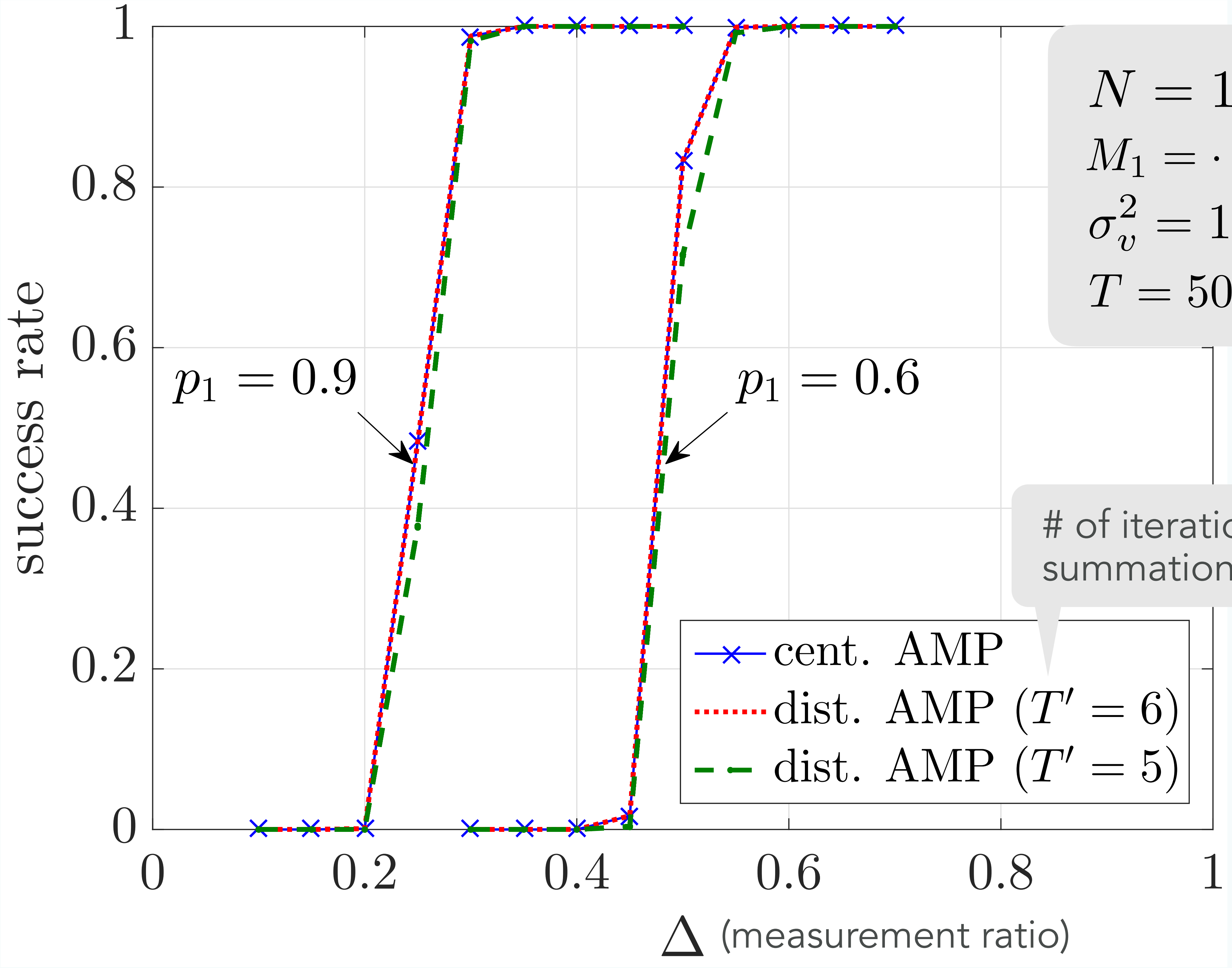
$$[\eta(\boldsymbol{r};\sigma^2)]_n = \frac{p_2\phi\left(\frac{\sqrt{\Delta}}{\sigma}(r_n-1)\right)}{p_1\phi\left(\frac{\sqrt{\Delta}}{\sigma}r_n\right) + p_2\phi\left(\frac{\sqrt{\Delta}}{\sigma}(r_n-1)\right)}$$

measurement matrix: $\boldsymbol{A}_k \in \mathbb{R}^{M_k \times N}$

measurement vector: $\boldsymbol{y}_k = \boldsymbol{A}_k\boldsymbol{x} + \boldsymbol{v}_k \in \mathbb{R}^{M_k}$

Appendix

Success Rate for Binary Vector Reconstruction



$N = 1000$

$M_1 = \cdots = M_K = \dfrac{\Delta N}{K}$

$\sigma_v^2 = 1$

$T = 50$

# of iterations in summation propagation