Distributed Multi-View Sparse Vector Recovery

Zhuojun Tian, *Student Member, IEEE,* Zhaoyang Zhang, *Senior Member, IEEE,*
and Lajos Hanzo, *Life Fellow, IEEE*

***Abstract*—In this paper, we consider a multi-view compressed sensing problem, where each sensor can only obtain a partial view of the global sparse vector. Here the partial view means that some arbitrary and unknown indices of the global vector are unobservable to that sensor and do not contribute to the measurement outputs. The sensors aim to collaboratively recover the global state vector in a decentralized manner. We formulate this recovery problem as a bilinear optimization problem relying on a factored joint sparsity model (FJSM), in which the variables are factorized into a node-specific sparse local masking vector and the desired common sparse global vector. We first theoretically analyze the general conditions guaranteeing the global vector's successful recovery. Then we propose a novel in-network algorithm based on the powerful distributed alternating direction method of multipliers (ADMM), which can reconstruct the vectors and achieve consensus among nodes concerning the estimation of the global vector. Specifically, each node alternately updates the common global vector and its local masking vector, and then it transfers the estimated global vector to its neighboring nodes for further updates. To avoid potential divergence of the iterative algorithm, we propose an early stopping rule for the estimation of the local masking vectors and further conceive an estimation error-mitigation algorithm. The convergence of the proposed algorithms is theoretically proved. Finally, extensive simulations validate their excellent performance both in terms of the convergence and recovery accuracy.**

***Index Terms*—Sensor network, multi-view sparse vector recovery, distributed compressed sensing, distributed optimization, alternating direction method of multipliers (ADMM).**

I. Introduction

Emerging as a significant theme in the forthcoming B5G and 6G era, *Sensing* is envisioned as an enabler for learning and building intelligence in the future smart world [1, 2]. Among the technological methods for sensing, compressed sensing (CS) is an efficient framework successfully applied in sensing scenarios utilizing sparse characteristics [3]. Built on CS and relying on the increasing storage and computational capacity of devices, distributed compressed sensing (DCS) [4] is thus expected to implement robust large-scale sensing with sensory data collected by widely-distributed sensors.

In numerous sensing scenarios, multiple sensors cooperate to recover the global information, which, as shown in Fig. 1, can be interpreted as the global environment [5–7], the 3D sculpture or architecture [8, 9], as well as the global state vector [10, 11]. Through discretizing the environment scatters or 3D objective into cloud points, all of the three scenarios can be described by global sparse vectors. However, due to the occlusion effect and blind spots resulted from sensors' geographic locations, or the limited sensing capabilities resulted from energy cost, some arbitrary, unknown and different components of the global vector may be missing or invisible for them. Consequently, each sensor only has access to a *partial view* of the global sparse vector. When there is a lack of a fusion center which may demand larger communication and computing power, it is appealing to design an in-network recovery algorithm for the sensors, so that they can efficiently and collaboratively recover the global vector relying only on their partial-view measurements and necessary information exchange within their neighborhoods.

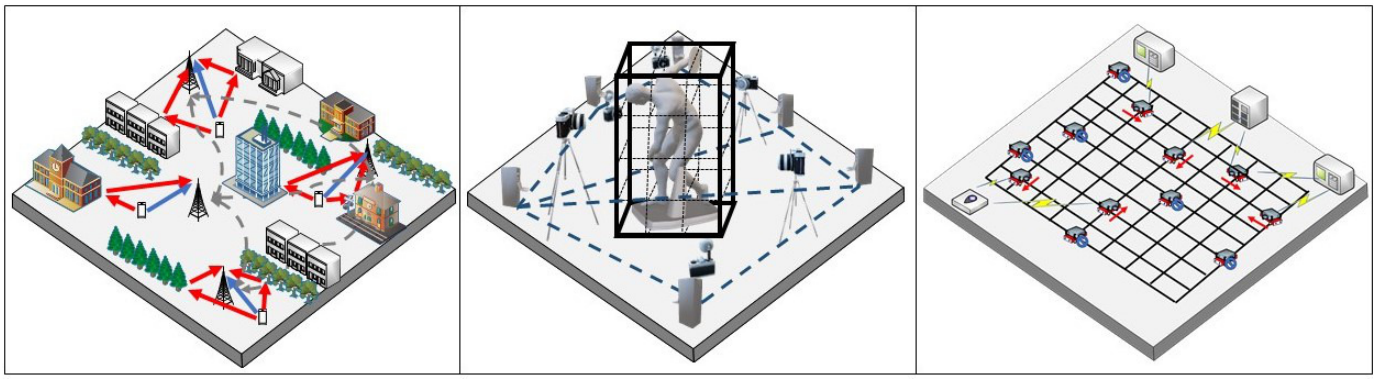
This in-network recovery problem can be generally modeled as a distributed compressed sensing problem, where the desired global information has certain structural sparsity and the observing nodes conduct independent sensing. The pivotal challenges are threefold, i.e., how to fully model the intrinsic sparsity among sensors; how to carry out the measurement; and how to design the recovery problem. An abundance of authors studied the DCS problem in the literature [4, 12–27]. As a meaningful approach to address the above challenges, Baron *et. al.* generalized the distributed compressed sensing process to the scenario in which the observed vectors of different nodes share a common component, namely the so-called Joint Sparsity Model 1 (JSM1). An alternative to that in which they have the same sparse support was termed as the so-called JSM2 [4]. Both centralized and decentralized recovery algorithms have been developed. The centralized schemes assume that a fusion center (FC) collects the measurement results as well as the sensing matrices from all the nodes, and based on these the center rebuilds the vectors. In a specific case where all the nodes share the same sensing matrix, the problem can be cast as the multiple measurement vector (MMV) problem [12–14]. On the other hand, the decentralized counterparts have no fusion center. Hence, the neighboring nodes exchange their necessary messages over the network collaboratively rebuilding the vectors, which is more robust to the nodes' failures. These approaches can be basically classified into three types: convex optimization [15–21], greedy pursuit [22–24] and Bayesian inference [25–27].

Among them, the optimization-based methods are popu-

Z. Tian and Z. Zhang's work was supported in part by National Natural Science Foundation of China under Grant U20A20158 and 61725104, National Key R&D Program of China under Grant 2020YFB1807101 and 2018YFB1801104, and Zhejiang Provincial Key R&D Program under Grant 2023C01021. L. Hanzo would like to acknowledge the financial support of the Engineering and Physical Sciences Research Council projects EP/W016605/1 and EP/X01228X/1 as well as of the European Research Council's Advanced Fellow Grant QuantCom (Grant No. 789028)

Z. Tian and Z. Zhang (Corresponding Author) are with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310007, China, and with Zhejiang Provincial Key Lab of Information Processing, Communication and Networking (IPCAN), Hangzhou 310007, China, and also with Zhejiang Provincial Key Lab of Collaborative Sensing and Autonomous Systems, Hangzhou 3100015, China. (e-mail: {dankotian, ning_ming}@zju.edu.cn)

L. Hanzo (e-mail: lh@ecs.soton.ac.uk) is with the Department of Electronics and Computer Science, University of Southampton, UK.



(a) Distributed Environment Sensing in Integrated Sensing and Communications

(b) Distributed 3D-Imaging from compressed sensing measurements

(c) Global State Monitoring in IoT with partial observations

Fig. 1. Illustration of three typical application scenarios with multi-view compressed sensing: (a) is a typical environment sensing problem in Integrated Sensing and Communications (ISAC), where each device (such as base station in [5] or radar system in [6]) obtains observations of the environment through wave propagation. (b) describes a 3D-Imaging problem where each sensor (like sonar system [8] or computational imaging system [9]) measures the objective within its partial view. (c) is a smart industrial IoT system, where distributed sensors monitor the global state space via compressed measurements [10, 11]. In all these scenarios, each client could only observe a partial view of the global vector due to geographic locations and the blockage as in (a) and (b), as well as the limited energy and sensing capabilities as in (c).

lar as a benefit of their efficient tractability and theoretical guarantees. Mateos *et al.* solved the reconstruction problem by formulating it as a distributed LASSO problem [15]. Explicitly, upon applying the distributed alternating direction method of multipliers (ADMM), Mateos *et al.* demonstrated the equivalence of the distributed algorithm and its centralized counterpart in terms of their convergence. As a further advance, a communication-efficient implementation of distributed ADMM, termed as D-ADMM, was proposed in [16] to recover the sparse vectors, which was further extended to other separable optimization objectives in [17]. Matamoros *et al.* applied distributed ADMM to solve the JSM1 problem in DCS, quantifying the efficiency of ADMM for the recovery of both the common and individual components [20].

Although certain differences of node observations have been considered in the above treatises, each of the nodes' measurements was assumed to be complete, i.e. it included contributions from all the different indices of the desired sparse vector. Thus, the above methods cannot deal with the incomplete observations in the sensor network. Explicitly, there is a paucity of literature on the general multi-view compressed sensing problem. Although in [28] the authors considered the case where each node has only partial components of the global variable, the observable components' indices are treated as fully known. However, in practice, the sensors may be randomly distributed or could be mobile. Consequently, the observable indices are generally unknown and they have to be estimated jointly in the recovery process. Authors in [5] considered the multi-view sensing problem while the recovery was implemented in a centralized manner.

In this work, we aim at solving the distributed recovery problem in the multi-view sensing network as shown in Fig. 1. Given its wide applications and excellent performance in distributed large-scale optimization problems [29, 30], especially its proven convergence [31–33], we opt for the distributed ADMM framework to address the problem. Before doing that, in order to represent the multi-view vectors observed by different sensors, we introduce a binary masking vector for each sensor and propose a factored joint sparsity model (FJSM). FJSM decouples the local observable vector for each sensor into the global vector and the local masking vector. This is inspired by [34], which formulates a bilinear factorization problem for the presence of missing data in the measurement. We observe that the factorized model and the corresponding bilinear problem may cause certain scaling ambiguity, making the solution non-unique and the problem non-convex [35–38]. To handle the problem, we constrain the elements of the masking vectors in the Boolean set similarly to [34, 35]. Moreover, we propose an early-stopping rule to improve the convergence of the algorithm.

The contributions of our work can thus be summarized as follows:

- We propose a general factored joint sparsity model for the multi-view sensing problem, which decouples the local vector into a global vector and a node-specific local masking vector. Based on the FJSM, we cast this problem as a bilinear factorization problem, which can be solved in a distributed manner. Additionally, we derive the theoretical measurement bounds for this problem under noiseless ℓ_0 norm optimization.
- Based on ADMM, a distributed optimization algorithm is proposed for solving the problem, in which each node iteratively updates both its masking vector and the global vector. A penalty term is added in the updating of the masking vectors, which allows solutions to approach the Boolean distribution. To guarantee the convergence, the updating of the masking vectors is terminated at some point guided by an early-stopping rule. An error-reduction step is carried out after that for mitigating the potential recovery error of the local vectors.
- Both the convergence and recovery performances are theoretically analyzed. Extensive simulations are conducted for comparing the proposed algorithms to other methods, validating their excellent performances both in terms of convergence and recovery accuracy. Moreover, we show that independent recovery at each sensor outperforms

the collaborative ones under severe blockage conditions, both theoretically and experimentally. Simulations are also conducted for evaluating the impact of the penalty term and guide its selection.

This paper is organized as follows. In Section II, we outline the system model and formulate the optimization problem. Section III gives the fundamental measurement bounds for ℓ_0 norm recovery. In Section IV, we develop our distributed algorithm for solving the problem, and propose an error-mitigation algorithm. Section V provides theoretical analysis w.r.t. the convergence and recovery results. Our numerical experiments are discussed in Section VI for validating the performance of the proposed algorithms. Section VII concludes the paper and discusses promising future research avenues.

Some notations: Given a vector $\boldsymbol{x}$, its support is denoted by $\mathcal{S}(\boldsymbol{x})$, which contains the indices of its nonzero entries. Its t-th entry is denoted by $\boldsymbol{x}(t)$. The cardinality of a given set Γ is denoted by $|\Gamma|$. Let $[\mathbf{1}]_{N\times 1}$ (resp. $[\mathbf{0.5}]_{N\times 1}$) be the $N\times 1$ vector with elements equal to 1 (resp. 0.5). We write $\mathbf{I}_N$ as the $N\times N$ identity matrix. The operator $\odot$ is the Hadamard product and $\otimes$ denotes the Kronecker product. The corresponding diagonal matrix of $\boldsymbol{x}$ is defined as $\mathbf{diag}(\boldsymbol{x})$, whose diagonal elements are components in $\boldsymbol{x}$. Likewise, the diagonal matrix for a set of matrices $\{\mathbf{X}_i \in \mathbb{R}^{m\times n}, i = 1, ..., N\}$ is defined as

$$\mathbf{diag}(\{\mathbf{X}_i\}) = \begin{bmatrix} \mathbf{X}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_2 & \cdots & \mathbf{0} \\ & & & \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{X}_N \end{bmatrix} \in \mathbb{R}^{mN\times nN}.$$

II. System Modeling and Problem Formulation

We consider a decentralized multi-sensor network and represent it by an undirected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, ..., N\}$ denotes the set of N distributed sensors and the edge set $\mathcal{E} = \{\varepsilon_{ij}\}_{i,j\in\mathcal{V}}$ indicates the communication links between them. Furthermore, $\mathcal{N}_i$ denotes the set of sensor i's neighboring sensors. The adjacency matrix of $\mathcal{G}$ is defined as $\mathbf{W}$, where $\mathbf{W}(i, j) = 1$ if $\varepsilon_{ij} \in \mathcal{E}$ and $\mathbf{W}(i, j) = 0$ otherwise. The diagonal degree matrix of $\mathcal{G}$ is denoted by $\mathbf{D}$, whose i-th diagonal element is the degree of node i, i.e., $\mathbf{D}(i, i) = |\mathcal{N}_i|$.

We denote the global state vector by $\boldsymbol{x} \in \mathbb{R}^T$ and the observable local vector of sensor i by $\boldsymbol{z}_i \in \mathbb{R}^T$. The support of $\boldsymbol{x}$ is denoted by $\mathcal{S}(\boldsymbol{x})$ and its sparsity is $|\mathcal{S}(\boldsymbol{x})| = K, K \ll T$. Each sensor in the network can only observe a partial view of the state vector and the missing or blocked entries become 0. Note that the blockage can happen in the zero entries, which however makes no impact on the results. So here to clarify, the blockage refers to those blocking the nonzero entries and leading to the loss of information. We can decouple each observable local vector $\boldsymbol{z}_i$ into the common global vector and the binary masking vector $\boldsymbol{v}_i \in \mathbb{R}^T$, where $\boldsymbol{v}_i(t) = 1$ indicates $\boldsymbol{x}(t), t \in \mathcal{S}(\boldsymbol{x})$ is observable for node i and $\boldsymbol{v}_i(t) = 0$ otherwise. Then we have

$$\boldsymbol{z}_i = \boldsymbol{v}_i \odot \boldsymbol{x},$$

which we term as the factored joint sparsity model. Here the global vector $\boldsymbol{x}$ is common for all sensors, while each sensor has its own masking vector $\boldsymbol{v}_i$.

The sensor i in the network individually senses its observable sparse vector $\boldsymbol{z}_i$ through a set of linear and local measurements, i.e.,

$$\boldsymbol{y}_i = \mathbf{A}_i\boldsymbol{z}_i + \boldsymbol{w}_i = \mathbf{A}_i(\boldsymbol{v}_i \odot \boldsymbol{x}) + \boldsymbol{w}_i, \tag{1}$$

where $\mathbf{A}_i \in \mathbb{R}^{M_i\times T} (M_i \ll T)$ denotes the measurement matrix in sensor i and $\boldsymbol{w}_i \in \mathbb{R}^{M_i}$ is the additive noise. Here M_i is the number of measurements made by sensor i, and we employ the random i.i.d. Gaussian matrices [39] as the measurement matrix $\mathbf{A}_i$. Note that (1) is a bilinear model, where the observations are influenced by the two factors $\boldsymbol{v}_i$ and $\boldsymbol{x}$ with bilinear relationship. A practical example of such a model can be found in [5] (see also Fig.1 (a)) where a distributed ISAC scenario is considered. In this context, several base stations (BS) collaboratively sense the environment through EM-wave illumination during the communication process. Here $\boldsymbol{x}$ denotes the desired reflection coefficients of the cloud points discretizing the region of interest, $\boldsymbol{v}_i$ depicts user i's locally observable set of the cloud points, and $\mathbf{A}_i$ here is user i's observation matrix with each row represents the channel gains of all the path through the cloud points in an illumination. Refer to [5] for details.

The ultimate goal for each sensor is to reconstruct the global vector with the aid of its own measurements and the messages received from other nodes, in which process, the observable local vector could also be recovered. Additionally, through cooperation, the number of measurements in each sensor is expected to be much lower than that would be required for recovering the vectors independently. To achieve this ambitious goal, we formulate the following optimization problem (2) for estimating the common global vector $\boldsymbol{x}$ through cooperation among the nodes, as well as the local masking vector $\boldsymbol{v}_i$ in each sensor:

$$\begin{aligned} \min_{\boldsymbol{x}_i, \boldsymbol{v}_i} \quad & \sum_{i=1}^{N} \Big[\|\boldsymbol{y}_i - \mathbf{A}_i(\boldsymbol{v}_i \odot \boldsymbol{x}_i)\|_2^2 + \lambda \|\boldsymbol{x}_i\|_1 \Big], \\ s.t. \quad & \boldsymbol{x}_i = \boldsymbol{x}_j, \quad \forall i \in \mathcal{V}, j \in \mathcal{N}_i, \\ & \boldsymbol{v}_i \in \{0, 1\}^T, \quad \forall i \in \mathcal{V}, \end{aligned} \tag{2}$$

where $\{0, 1\}^T$ denotes the set of length-T vectors with each entry equal to 0 or 1. We have the following assumptions:

Assumption 1. *The sensor network is connected, i.e., there exists a path between any pair of sensors.*

Assumption 2. *We assume that each nonzero entry of the global vector can be observed by at least one sensor, i.e.,*

$$\mathcal{S}(\boldsymbol{z}_1) \cup ... \cup \mathcal{S}(\boldsymbol{z}_N) = \mathcal{S}(\boldsymbol{x}).$$

Before developing a practical algorithm, we first analyze the fundamental theoretical bounds on the measurement requirements for this multi-view sensing scenario, which are based on noiseless ℓ_0 norm minimization.

III. Conditions to Guarantee ℓ_0 Norm Recovery

For representing the vectors and measurements compactly in each node, we introduce $\tilde{M} = \sum_{i\in\mathcal{V}} M_i$ and define $\mathbf{Y} \in \mathbb{R}^{\tilde{M}}$,

$\mathbf{Z} \in \mathbb{R}^{TN}$, $\mathbf{\Phi} \in \mathbb{R}^{\tilde{M} \times TN}$ respectively as

$$\mathbf{Y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_N \end{bmatrix}, \quad \mathbf{Z} = \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \\ \vdots \\ \boldsymbol{z}_N \end{bmatrix}, \quad \mathbf{\Phi} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & ... & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & ... & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & ... & \mathbf{A}_N \end{bmatrix}. \tag{3}$$

Then in the noiseless condition, we have $\mathbf{Y} = \mathbf{\Phi Z}$.

Let us now define the stacked configuration matrix for all nodes as $\tilde{\mathbf{P}} = [\mathbf{P}_1^T, ..., \mathbf{P}_N^T]^T$. Here $\mathbf{P}_i$ is a $T \times K$ matrix, where $K = |\mathcal{S}(\boldsymbol{x})|$. We construct $\mathbf{P}_i$ in the following way. First, we delete $(T-K)$ columns from the $T \times T$ identity matrix outside the support of $\boldsymbol{x}$ and get $\check{\mathbf{P}}_i$. If the k-th index of the support is blocked for node i, then the k-th column of $\check{\mathbf{P}}_i$ is set to all-zero and we get $\mathbf{P}_i$. Under this definition, we have $\mathbf{Y} = \mathbf{\Phi}\tilde{\mathbf{P}}\boldsymbol{\theta}$, where $\boldsymbol{\theta} \in \mathbb{R}^K$ consists of the non-zero values of the global signal $\boldsymbol{x}$. Let us define the *visible set* $\Pi(\Gamma, \tilde{\mathbf{P}})$ as follows, to represent the indices of $\boldsymbol{x}$ that can only be observed by the nodes in the subset Γ.

$$\Pi(\Gamma, \tilde{\mathbf{P}}) = \{t \in 1, ..., T, \quad \exists i \in \Gamma, \boldsymbol{v}_i(t) \neq 0 \\ \& \, \forall j \in \mathcal{V} - \Gamma, \boldsymbol{v}_j(t) = 0\},$$

where Γ is an arbitrary subset of the node set $\mathcal{V}$.

Then we can formulate the theoretical measurement bounds for ℓ_0 norm recovery by Theorem 1.

Theorem 1. *Assume that the global vector $\boldsymbol{x}$ is partially observed by each node. Let furthermore $\mathcal{M} = (M_1, ..., M_N)$ be the measurement tuple containing the measurement sizes of all nodes and $\boldsymbol{A}_i$ be the random matrix for node i with M_i rows having i.i.d. Gaussian entries. Then we have:*

a) (Achievable, known $\tilde{\mathbf{P}}$) *If $\tilde{\boldsymbol{P}}$ is known and the measurement tuple satisfies:*

$$\sum_{j \in \Gamma} M_j \geq |\Pi(\Gamma, \tilde{\boldsymbol{P}})| \tag{4}$$

for all $\Gamma \subseteq \mathcal{V}$, then there exists a unique solution $\hat{\boldsymbol{\theta}}$ to $\boldsymbol{Y} = \mathbf{\Phi}\tilde{\boldsymbol{P}}\boldsymbol{\theta}$ with probability one over $\{\boldsymbol{A}_i\}_{i \in \mathcal{V}}$. Given the configuration matrix $\tilde{\boldsymbol{P}}$, both the global $\boldsymbol{x}$ and local $\boldsymbol{z}_i$ vectors can be uniquely recovered.

b) (Achievable, unknown $\tilde{\mathbf{P}}$) *If $\tilde{\boldsymbol{P}}$ is unknown and the measurement tuple satisfies:*

$$\sum_{j \in \Gamma} M_j \geq |\Pi(\Gamma, \tilde{\boldsymbol{P}})| + |\Gamma| \tag{5}$$

for all $\Gamma \subseteq \mathcal{V}$, then the global $\boldsymbol{x}$ and local $\boldsymbol{z}_i$ vectors can be uniquely recovered with probability one over $\{\boldsymbol{A}_i\}_{i \in \mathcal{V}}$.

c) (Converse) *If the measurement tuple satisfies:*

$$\sum_{j \in \Gamma} M_j < |\Pi(\Gamma, \tilde{\boldsymbol{P}})| \tag{6}$$

for any $\Gamma \subseteq \mathcal{V}$, then the global vector and the local vector cannot be uniquely recovered.

Proof. See Appendix A. □

Proposition 1. *We assume furthermore that the probabilities of blockage for all the nodes are the same, namely p. Provided that $p \leq 1 - \frac{1}{N}$ under Assumption 2, for larger p, each node needs more measurements for collaboratively recovering the global and local vectors in expectation, compared with smaller p.*

Proof. Let Γ contain a single one node. Then we have

$$\mathbb{E}\big[|\Pi(\Gamma, \mathbf{P})|\big] = K \times (1-p) \times p^{(N-1)}.$$

With $p \leq 1 - \frac{1}{N}$, it may be readily shown that $\partial \mathbb{E}[|\Pi(\Gamma, \mathbf{P})|]/\partial p \geq 0$. Thus the proposition is proven. □

Theorem 1 gives the fundamental recovery bounds of noiseless measurements based on ℓ_0 norm minimization. Specifically, Theorem 1 a) and b) reveal the effect of the local masking vector on the measurement complexity of the ℓ_0 norm based recovery. Theorem 1 c) gives the lower bound of the number of measurements, below which the vectors cannot be recovered with either ℓ_0 norm or ℓ_1 norm method. Moreover, Theorem 1 rigorously proves one information-theoretic intuition that the number of measurements required for each sensor must account for the observable features unique to that sensor, while at the same time, features observable among multiple sensors should be amortized over the group. Such rule is also partly validated by the simulations in Section VI-G. All of these remarks on Theorem 1 may serve as the rules of thumb, offering guidance on technological applications.

To circumvent the NP-hardness and intractability of ℓ_0 norm minimization, in the next section, we will develop algorithms based on ℓ_1 norm optimization as in Problem (2).

IV. Algorithm Derivation

To solve the optimization problem (2), in this section, we develop our algorithm based on the powerful distributed ADMM framework. Moreover, we design an estimation error-mitigation method.

A. Distributed ADMM with Value Penalty

We refer readers to [31, 40, 41] for a general discussion on applying ADMM for solving a set of non-convex programs and to [15] for its implementation in a decentralized network. Based on ADMM, each node alternately updates the primal variables $\{\boldsymbol{x}_i, \boldsymbol{v}_i\}$ and the dual variable. The challenges arise from two aspects. First, the Boolean constraints of $\boldsymbol{v}_i$ make the optimization problem non-convex and NP-hard; Secondly, the bilinear relationship between the two primal variables $\{\boldsymbol{x}_i, \boldsymbol{v}_i\}$ makes the problem non-convex and the distributed ADMM algorithm struggles to converge. Hence we will focus on addressing these two issues.

To begin with, a popular technique of tackling the elementwise Boolean constraint $\boldsymbol{v}_i(t) \in \{0, 1\}$ is to relax it into the inequality constraint $0 \leq \boldsymbol{v}_i(t) \leq 1$ and then project the result into the integer solution. In our algorithm, inspired by the ADMM penalized decoding method introduced in [42], we add the objective a penalty term $g_i(\boldsymbol{v}_i)$ into the optimization of $\boldsymbol{v}_i$, which aims for making the non-integer vertices more costly. In particular, the value of the penalty function is lower both at 0 and 1 compared to any other point in the interval $(0, 1)$. We slack the Boolean constraint and the optimization problem can be modified as (7):

$$\begin{aligned} \min_{\boldsymbol{x}_i, \boldsymbol{v}_i} \quad & \sum_{i=1}^{N} \big[\phi_i(\boldsymbol{v}_i, \boldsymbol{x}_i) + g_i(\boldsymbol{v}_i)\big], \\ s.t. \quad & \boldsymbol{x}_i = \boldsymbol{t}_{ij}, \quad \forall i \in \mathcal{V}, j \in \mathcal{N}_i, \\ & \boldsymbol{x}_j = \boldsymbol{t}_{ij}, \quad \forall i \in \mathcal{V}, j \in \mathcal{N}_i, \end{aligned} \tag{7}$$

$$\boldsymbol{v}_i \in [0,1]^T, \quad \forall i \in \mathcal{V},$$

where $\phi_i(\boldsymbol{v}_i, \boldsymbol{x}_i) = \|\boldsymbol{y}_i - \mathbf{A}_i(\boldsymbol{v}_i \odot \boldsymbol{x}_i)\|_2^2 + \lambda\|\boldsymbol{x}_i\|_1$. Here the slack variable $\boldsymbol{t}_{ij}$ is introduced for the consensus constraint $\boldsymbol{x}_i = \boldsymbol{x}_j$ between node i and its neighboring node $j \in \mathcal{N}_i$. Based on that, the augmented Lagrangian function of Problem (7) can be formulated as:

$$\mathcal{L} = \sum_{i=1}^{N}\big[\phi_i(\boldsymbol{v}_i, \boldsymbol{x}_i) + g_i(\boldsymbol{v}_i)\big] + \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i}\big[\bar{\boldsymbol{u}}_{ij}(\boldsymbol{x}_i - \boldsymbol{t}_{ij}) + \breve{\boldsymbol{u}}_{ij}(\boldsymbol{x}_j - \boldsymbol{t}_{ij})\big] + \frac{c}{2}\sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i}\Big[\|\boldsymbol{x}_i - \boldsymbol{t}_{ij}\|_2^2 + \|\boldsymbol{x}_j - \boldsymbol{t}_{ij}\|_2^2\Big],$$

where $\bar{\boldsymbol{u}}_{ij}$ and $\breve{\boldsymbol{u}}_{ij}$ are dual variables. Let us define $\boldsymbol{q}_i \triangleq \sum_{j\in\mathcal{N}_i}(\bar{\boldsymbol{u}}_{ij} + \breve{\boldsymbol{u}}_{ji})$ and initialize $\bar{\boldsymbol{u}}_{ij}^{(0)} + \breve{\boldsymbol{u}}_{ij}^{(0)} = \mathbf{0}, \forall i, j$. We apply the Gauss-Seidel strategy [20, 33] to solve the subproblem corresponding to the primal variables $\{\boldsymbol{x}_i, \boldsymbol{v}_i\}$ as follows.

$$(\boldsymbol{x}_i^{(k)}, \boldsymbol{v}_i^{(k)}) = \underset{0\le\boldsymbol{v}_i(t)\le 1}{\arg\min}\ \mathcal{L}(\boldsymbol{q}_i^{(k)}, \boldsymbol{x}_i^{(k-1)}, \boldsymbol{v}_i^{(k-1)}),$$

where the update of $\boldsymbol{v}_i^{(k)}$ is based on the updating result of $\boldsymbol{x}_i^{(k)}$. Then the update rules of $\boldsymbol{q}_i$ and the primal variables $\{\boldsymbol{x}_i, \boldsymbol{v}_i\}$ are:

$$\boldsymbol{q}_i^{(k)} = \boldsymbol{q}_i^{(k-1)} + c\sum_{j\in\mathcal{N}_i}\big(\boldsymbol{x}_i^{(k-1)} - \boldsymbol{x}_j^{(k-1)}\big), \tag{8a}$$

$$\boldsymbol{x}_i^{(k)} = \underset{\boldsymbol{x}_i}{\arg\min}\Big\{\phi_i(\boldsymbol{v}_i^{(k-1)}, \boldsymbol{x}_i) + \boldsymbol{x}_i^T q_i^{(k)} + c\sum_{j\in\mathcal{N}_i}\big\|\boldsymbol{x}_i - \frac{\boldsymbol{x}_i^{(k-1)} + \boldsymbol{x}_j^{(k-1)}}{2}\big\|_2^2\Big\}, \tag{8b}$$

$$\boldsymbol{v}_i^{(k)} = \underset{0\le\boldsymbol{v}_i(t)\le 1}{\arg\min}\Big[\big\|\boldsymbol{y}_i - \mathbf{B}_i^{(k)}\boldsymbol{v}_i\big\|_2^2 + g_i(\boldsymbol{v}_i)\Big], \tag{8c}$$

where $\mathbf{B}_i^{(k)} = \mathbf{A}_i \cdot \mathbf{diag}(\boldsymbol{x}_i^{(k)})$. Note that the slack variables $\{\boldsymbol{t}_{ij}\}_{i\in\mathcal{V}, j\in\mathcal{N}_i}$ do not appear in (8) because they can be expressed by $\{\boldsymbol{x}_i\}_{i\in\mathcal{V}}$; we refer readers to Appendix A in [15] for details. To retain the convexity of the subproblem (8c), we apply the ℓ_2 penalty function and define it as follows :

$$g_i(\boldsymbol{v}_i) = -\Big\|\mathbf{R}_i^{(k)}(\boldsymbol{v}_i - [\mathbf{0.5}]_{T\times 1})\Big\|_2^2.$$

Given a well-designed parameter $\mathbf{R}_i^{(k)}$, where $\mathbf{R}_i^{(k)^T}\mathbf{R}_i^{(k)} = \alpha\mathbf{B}_i^{(k)^T}\mathbf{B}_i^{(k)}, 0 \le \alpha < 1$, the optimization problem in (8c) is convex. This is designed to make the variables' update mathematically tractable in the same way as [42]. On the other hand, such penalization is standard to account for a relaxed binary variable without limitation on the penalty parameter. The authors in [43, 44] formulated non-convex optimization problems based on the concave penalization to account for a relaxed binary variable and proposed solutions under mild conditions. Compared with [43, 44], a drawback may arise from the limited penalty effect because of the relatively small penalty parameter of $0 \le \alpha < 1$. However, under the bilinear relationship between the two primal variables, the convexity of the subproblem is vital for the tractability and computational complexity. Moreover, we do not strictly require the distribution of $\boldsymbol{v}_i$ to be Boolean in the early iterations for the stability of the algorithm. Rather, we would use a soft heuristic penalty term, which can enhance the tendency for the elements of $\boldsymbol{v}_i$ to approximate 0 or 1, making this a rational design. We can then obtain coordinate-wise update rule for $\boldsymbol{v}_i$:

$$\boldsymbol{v}_i^{(k)} = \prod_{[0,1]}\Big\{\frac{1}{1-\alpha}\big[\mathbf{B}_i^{(k)^\dagger}\boldsymbol{y}_i - \alpha(\mathbf{B}_i^{(k)^\dagger}\mathbf{B}_i^{(k)})^T[\mathbf{0.5}]_{T\times 1}\big]\Big\}, \tag{9}$$

where $\prod_{[0,1]} : \mathbb{R} \to [0,1]$ is the elementwise projection to the interval by mapping the elements smaller than 0 to 0 and those larger than 1 to 1. Note that the optimal solution of (8c) has to be obtained through KKT conditions with multiple inner iterations and unbearable computation cost. Thus, we apply the coordinate-wise approximated result (9) as that in [42]. It can be observed from (9) that the effect of the penalty term may be viewed as an elementwise soft mapping from the original solution $\mathbf{B}_i^{(k)^\dagger}\boldsymbol{y}_i$ towards $\{0,1\}$. With respect to the updating complexity, the pseudo inverse of $\mathbf{A}_i$, denoted by $\mathbf{A}_i^\dagger$, can be pre-computed at the beginning and stored in each node. Then $\mathbf{B}_i^{(k)^\dagger}$ can be readily computed at a modest computational cost, where $\mathbf{diag}(\boldsymbol{x}_i^{(k)})^\dagger$ is easy to acquire with the reciprocal of each non-zero diagonal element.

In the algorithm, if $\boldsymbol{v}_i$ is mapped into $\{0,1\}^T$ at the very beginning of iterations, then the process becomes unstable and prone to divergence. On the other hand, the scaling ambiguity problem still exits due to the bilinear relationship and the limited penalty parameter, implying that a hard projection into the binary set is necessary so as to narrow down the solution set. Hence we map the masking vector into the Boolean set after a few iterations. In the following we will discuss when to carry out the hard mapping for each node. We characterise the distribution of $\boldsymbol{v}_i^{(k)}$ by the distance of $\boldsymbol{v}_i^{(k)}$ from the Boolean subspace, defined as

$$d_i^{(k)} = \big\|\boldsymbol{v}_i^{(k)} \odot ([\mathbf{1}]_{T\times 1} - \boldsymbol{v}_i^{(k)})\big\|_2^2. \tag{10}$$

This is affected by the penalty parameter and furthermore as found by simulations, the value of $d_i^{(k)}$ is expected to decrease until the estimated value of $\boldsymbol{v}_i$ approaches some stationary point, where we narrow down the solution set of $\boldsymbol{v}_i$ into the Boolean set. Each node individually decides its instant of discretizing $\boldsymbol{v}_i$. Additionally, for the sake of stability, we set the minimum number $K_{\min}$ of iterations to be used before discretization, explicitly, if $k > K_{\min}$ and $d_i^{(k)} > d_i^{(k-1)}$, the solution set of $\boldsymbol{v}_i$ in node i is ready to be mapped into the Boolean set.

Upon projecting the solutions into the Boolean set using a hard mapping, we effectively remove the penalty term by setting $\alpha = 0$. This assumes that the soft mapping effect is not necessary, which may lead to sub-optimal solutions instead. Then the update of $\boldsymbol{v}_i$ can be formulated as:

$$\boldsymbol{v}_i^{(k)} = \prod_{\{0,1\}}\big(\mathbf{B}_i^{(k)^\dagger}\boldsymbol{y}_i\big), \tag{11}$$

where the projection $\prod_{\{0,1\}} : \mathbb{R} \to \{0,1\}$ rounds each entry to 0 or 1, whichever is closer.

Having addressed the problem of Boolean constraints, we now turn to the challenge of convergence. When applied in bilinear optimization problems, distributed ADMM lacks theoretical convergence guarantees and indeed, it often exhibits poor convergence in real implementations, and becomes

divergent after finding a local optimum. We observe that in the distributed implementations, the poor convergence typically owning to specific nodes exhibiting divergent tendency, even if the other nodes are converging. Inspired by the iterative water-filling algorithm of [45] that allocates more power to better channels in support of efficient power allocation, we also intend to focus on the strongly convergent nodes. Specifically, we propose to curtail the updating of the masking vectors in the poor-performance nodes, so as to concentrate the iterative process on the convergent nodes. Note that the updating of the global vector should continue so as to harness the measurement results from all nodes.

To decide whether the node achieves a local minimum point or prone to divergence, an early stopping rule is introduced, inspired by the Residual Balancing method of the ADMM algorithm [46]. The rule proves that the magnitude gap between the primal residual and the dual residual indicates the convergence rate of the algorithm. The authors of [47] further extended the method into distributed ADMM and formulated the expressions of primal and dual residuals as follows:

$$\|r_i^{(k)}\|_2^2 = \|\boldsymbol{x}_i^{(k)} - \bar{\boldsymbol{x}}_i^{(k)}\|_2^2, \quad j \in \mathcal{N}_i \tag{12a}$$

$$\|s_i^{(k)}\|_2^2 = c^2\|\bar{\boldsymbol{x}}_i^{(k)} - \bar{\boldsymbol{x}}_i^{(k-1)}\|_2^2, \quad j \in \mathcal{N}_i \tag{12b}$$

where $\bar{\boldsymbol{x}}_i \triangleq \frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i}\boldsymbol{x}_j$. The basic idea is that a larger difference between the two magnitudes indicates the slower convergence rate of the algorithm. When the difference is large enough, the convergence rate becomes sufficiently slow, which is indicative of approaching nearby a local minimum point. So the stopping rule of $\boldsymbol{v}_i$'s update is formulated as:

$$\|r_i^{(k)}\|_2^2 \geq \delta\|s_i^{(k)}\|_2^2 \quad \text{or} \quad \|s_i^{(k)}\|_2^2 \geq \delta\|r_i^{(k)}\|_2^2, \tag{13}$$

where the value of δ is chosen by experiments. A beneficial choice for the parameter is suggested to be $\delta = 10$ [47] for most situations. Additionally, for the sake of synchronization and stability, we set a maximum number of iterations $K_{\max}$ before early stopping for the nodes.

The proposed distributed ADMM associated with value penalty (VPD-ADMM), may thus be summarized in **Algorithm 1**. Here $\hat{\boldsymbol{x}}_i$ and $\hat{\boldsymbol{v}}_i$ denote the updated results of the primal variables $\boldsymbol{x}_i, \boldsymbol{v}_i$ respectively. a_i is the indicator on the early stopping of $\boldsymbol{v}_i$ while b_i is the indicator on the discretization of $\boldsymbol{v}_i$.

Through cooperation among nodes, the recovery of the global vector $\boldsymbol{x}$ can be guaranteed with a high probability, as illustrated by Theorem 3 of the next section. However, the recovery accuracy of the local vector, expressed as $\hat{\boldsymbol{x}}_i \odot \hat{\boldsymbol{v}}_i$, is highly dependent on that of the masking vector $\hat{\boldsymbol{v}}_i$, which may be curtailed prematurely and thus cannot guarantee its accuracy. To this end, after the early stopping of $\boldsymbol{v}_i$, we design an error-mitigation procedure for improving the recovery accuracy of the local vector and for further enhancing the estimation of the global vector, albeit at the expense of slightly increasing the computational cost.

B. Error Mitigation Algorithm

Although VPD-ADMM promises good performance for the recovery of the global vector, it cannot guarantee the estimation accuracy of the local vectors as discussed above. As a remedy, we propose an error-mitigation method applied by each node after the early-stopping of the iterations recovering the masking vector $\boldsymbol{v}_i$. The fixed estimated masking vector is denoted by $\hat{\boldsymbol{v}}_i$, whose diagonal representation is defined as $\hat{\mathbf{V}}_i = \mathbf{diag}(\hat{\boldsymbol{v}}_i)$. Upon denoting the real masking vector of node i by $\boldsymbol{v}_i^*$ and its corresponding diagonal matrix by $\mathbf{V}_i^* = \mathbf{diag}(\boldsymbol{v}_i^*)$, we have

$$\boldsymbol{y}_i = \mathbf{A}_i\mathbf{V}_i^*\boldsymbol{x} + \boldsymbol{w}_i = \mathbf{A}_i\hat{\mathbf{V}}_i\boldsymbol{x} + \mathbf{A}_i(\mathbf{V}_i^* - \hat{\mathbf{V}}_i)\boldsymbol{x} + \boldsymbol{w}_i. \tag{14}$$

Based on (14), we define $\boldsymbol{\beta}_i = (\mathbf{V}_i^* - \hat{\mathbf{V}}_i)\boldsymbol{x}$ as the error associated with the inequality $\mathbf{V}_i^* \neq \hat{\mathbf{V}}_i$. Then it may be deduced that when the estimated entry is the correct one, i.e., $\hat{\boldsymbol{v}}_i(t) = \boldsymbol{v}_i^*(t)$, the corresponding entry $\boldsymbol{\beta}_i(t) = 0$. Through our experiments, we observed that the accuracy of $\hat{\boldsymbol{v}}_i$ can be as high as 90%, which indicates the sparsity of the error vector $\boldsymbol{\beta}_i$. Introduce $\boldsymbol{z}_i$ as the local vector and we have $\boldsymbol{z}_i = \hat{\mathbf{V}}_i\boldsymbol{x} + \boldsymbol{\beta}_i$. Then for the specific nodes which stop updating $\boldsymbol{v}_i$, the local optimization problem can be formulated as:

$$\begin{aligned} \min_{\boldsymbol{z}_i,\boldsymbol{\beta}_i,\boldsymbol{x}_i} \quad & \|\boldsymbol{y}_i - \mathbf{A}_i\boldsymbol{z}_i\|_2^2 + \epsilon_1\|\boldsymbol{z}_i\|_1 + \epsilon_2\|\boldsymbol{\beta}_i\|_1 + \epsilon_3\|\boldsymbol{x}_i\|_1, \\ s.t. \quad & \boldsymbol{x}_i = \boldsymbol{x}_j, \quad j \in \mathcal{N}_i, \\ & \boldsymbol{z}_i = \hat{\mathbf{V}}_i\boldsymbol{x} + \boldsymbol{\beta}_i. \end{aligned} \tag{15}$$

Then the local Lagrangian function for these nodes is:

$$\begin{aligned} \mathcal{L}_i = & \|\boldsymbol{y}_i - \mathbf{A}_i\boldsymbol{z}_i\|_2^2 + \epsilon_1\|\boldsymbol{z}_i\|_1 + \epsilon_2\|\boldsymbol{\beta}_i\|_1 + \epsilon_3\|\boldsymbol{x}_i\|_1 \\ & + \frac{\rho}{2}\|\boldsymbol{z}_i - \hat{\mathbf{V}}_i\boldsymbol{x}_i - \boldsymbol{\beta}_i\|_2^2 + \boldsymbol{\tau}_i^T(\boldsymbol{z}_i - \hat{\mathbf{V}}_i\boldsymbol{x}_i - \boldsymbol{\beta}_i) \end{aligned}$$

Algorithm 1: VPD-ADMM

for *node* $i = 1, 2, ..., N$ **do**
  Initialize: $\boldsymbol{q}_i^{(0)} = \mathbf{0}$, $\boldsymbol{x}_i^{(0)} = \mathbf{0}$, $\boldsymbol{v}_i^{(0)} = \mathbf{1}$, $k = 0$.
  Set Indicator: $a_i = 1$, $b_i = 1$.
while *not converge* **do**
  $k = k + 1$
  for *node* $i = 1, 2, ...N$ **do**
    Update $\boldsymbol{q}_i^{(k)}$ according to (8a).
    Update $\boldsymbol{x}_i^{(k)}$ according to (8b).
    Compute $\|r_i^{(k)}\|_2^2$ and $\|s_i^{(k)}\|_2^2$.
    if $a_i = 0$ *or* $k > K_{max}$ *or ((13) is satisfied)* **then**
      Stop updating $\boldsymbol{v}_i$ and $a_i = 0$.
    else
      Compute $d_i^{(k)}$ according to (10).
      if $b_i = 0$ *or* ($k \leq K_{min}$ *and* $d_i^{(k)} > d_i^{(k-1)}$) **then**
        Update $\boldsymbol{v}_i^{(k)}$ according to (11) and set $b_i = 0$.
      else
        Update $\boldsymbol{v}_i^{(k)}$ according to (9).
    Transmit $\boldsymbol{x}_i^{(k)}$ to neighboring nodes $j \in \mathcal{N}_i$.
Output the estimation of global vector by $\hat{\boldsymbol{x}}_i$ and the local vector by $\hat{\boldsymbol{x}}_i \odot \hat{\boldsymbol{v}}_i$ for all node i.

$$+\frac{\theta}{2}\sum_{j\in\mathcal{N}_i}\left[\|\boldsymbol{x}_i-\boldsymbol{t}_{ij}\|_2^2+\|\boldsymbol{x}_j-\boldsymbol{t}_{ij}\|_2^2\right]$$
$$+\sum_{j\in\mathcal{N}_i}\left[\bar{\boldsymbol{u}}_{ij}^T(\boldsymbol{x}_i-\boldsymbol{t}_{ij})+\breve{\boldsymbol{u}}_{ij}^T(\boldsymbol{x}_j-\boldsymbol{t}_{ij})\right]. \quad (16)$$

Let us now define $\boldsymbol{q}_i \triangleq \sum_{j\in\mathcal{N}_i}(\bar{\boldsymbol{u}}_{ij}+\breve{\boldsymbol{u}}_{ij})$. Upon applying the distributed ADMM, we update the pair of dual variables $\{\boldsymbol{q}_i,\boldsymbol{\tau}_i\}$, and the three primal variables $\{\boldsymbol{z}_i,\boldsymbol{\beta}_i,\boldsymbol{x}_i\}$ in an alternative manner using the following rules:

$$\boldsymbol{q}_i^{(k)}=\boldsymbol{q}_i^{(k-1)}+\theta\sum_{j\in\mathcal{N}_i}(\boldsymbol{x}_i^{(k-1)}-\boldsymbol{x}_j^{(k-1)}), \quad (17a)$$

$$\boldsymbol{\tau}_i^{(k)}=\boldsymbol{\tau}_i^{(k-1)}+\rho(\boldsymbol{z}_i^{(k-1)}-\hat{\mathbf{V}}_i\boldsymbol{x}_i^{(k-1)}-\boldsymbol{\beta}_i^{(k-1)}). \quad (17b)$$

$$\boldsymbol{z}_i^{(k)}=\arg\min_{\boldsymbol{z}_i}\left\{\mathcal{L}_i(\boldsymbol{q}_i^{(k)},\boldsymbol{\tau}_i^{(k)},\boldsymbol{z}_i,\boldsymbol{\beta}_i^{(k-1)},\boldsymbol{x}_i^{(k-1)})\right\}, \quad (18a)$$

$$\boldsymbol{\beta}_i^{(k)}=\arg\min_{\boldsymbol{\beta}_i}\left\{\mathcal{L}_i(\boldsymbol{q}_i^{(k)},\boldsymbol{\tau}_i^{(k)},\boldsymbol{z}_i^{(k)},\boldsymbol{\beta}_i,\boldsymbol{x}_i^{(k-1)})\right\}, \quad (18b)$$

$$\boldsymbol{x}_i^{(k)}=\arg\min_{\boldsymbol{x}_i}\left\{\mathcal{L}_i(\boldsymbol{q}_i^{(k)},\boldsymbol{\tau}_i^{(k)},\boldsymbol{z}_i^{(k)},\boldsymbol{\beta}_i^{(k)},\boldsymbol{x}_i)\right\}. \quad (18c)$$

Under these updating rules, the VPD-ADMM relying on error mitigation (VPD-EM) is summarized as follows, which has two stages in each node: the first bilinear optimization stage and the second error-mitigation stage.

By introducing Stage Two, VPD-EM takes the estimation error of the masking vectors into consideration, which improves the recovery accuracy of the local vector. This procedure, however, increases the computational cost, because three sparse vectors have to be recovered in each iteration.

Algorithm 2: VPD-EM

for *node* $i=1,2,...,N$ **do**
  Initialize: $\boldsymbol{q}_i^{(0)}=\mathbf{0}$, $\boldsymbol{x}_i^{(0)}=\mathbf{0}$, $\boldsymbol{v}_i^{(0)}=\mathbf{1}$, $k=0$.
  Set Indicator: $b_i=1$, StageOne$_i$ = True, StageTwo$_i$ = False.
while *not converge* **do**
  $k=k+1$
  for *node* $i=1,2,...N$ **do**
    if *StageOne*$_i$ **then**
      Update $\boldsymbol{q}_i^{(k)}$ according to (8a).
      Update $\boldsymbol{x}_i^{(k)}$ according to (8b).
      Compute $\|r_i^{(k)}\|_2^2$ and $\|s_i^{(k)}\|_2^2$.
      if $k>K_{max}$ *or (13) is satisfied* **then**
        Stop updating $\boldsymbol{v}_i$ and initialize $\boldsymbol{\beta}_i^{(k)}=\mathbf{0}$, $\boldsymbol{z}_i^{(k)}=\boldsymbol{v}_i^{(k)}\odot\boldsymbol{x}_i^{(k)}+\boldsymbol{\beta}_i^{(k)}$
        StageOne$_i$ = False, StageTwo$_i$ = True.
      else
        Compute $d_i^{(k)}$ according to (10).
        if $b_i=0$ *or* ($k\le K_{min}$ *and* $d_i^{(k)}>d_i^{(k-1)}$) **then**
          Update $\boldsymbol{v}_i^{(k)}$ according to (11) and set $b_i=0$.
        else
          Update $\boldsymbol{v}_i^{(k)}$ according to (9).
    else
      Update $\boldsymbol{q}_i^{(k)},\boldsymbol{\tau}_i^{(k)}$ according to (17a-b).
      Update $\boldsymbol{z}_i^{(k)},\boldsymbol{\beta}_i^{(k)}$ and $\boldsymbol{x}_i^{(k)}$ according to (18a-c).
    Transmit $\boldsymbol{x}_i^{(k)}$ to neighboring nodes $j\in\mathcal{N}_i$.
Output the estimation of global vector by $\hat{\boldsymbol{x}}_i$ and the local vector by $\hat{\boldsymbol{z}}_i$ for all node i.

V. Convergence and Recovery Performance Analysis

In this section, we provide the theoretical analysis of the convergence properties as well as compressed sensing recovery of the VPD-ADMM algorithm. Furthermore, the convergence guarantee of the VPD-EM algorithm is also analyzed.

We commence the analysis of the VPD-ADMM's convergence properties. To the best of our knowledge, there lack theoretical guarantees for distributed ADMM with the two bilinear primal variables. Moreover, the update of distributed ADMM with two bilinear variables can be unstable and subject to diverge as shown in Section VI-A. For this reason, we propose an early-stopping rule to timely stop the updating of $\boldsymbol{v}_i$. Based on the early-stopping mechanism, we consider the convergence properties with the resultant $\boldsymbol{v}_i$ at all nodes, i.e., the eventual convergence after all the nodes stop updating their masking vectors. Let us define the corresponding centralized optimization problem as:

$$\hat{\boldsymbol{x}}_{Lasso}=\arg\min_{\boldsymbol{x}}\{\|\mathbf{Y}-\boldsymbol{\Phi}\tilde{\mathbf{V}}\boldsymbol{x}\|+\lambda\|\boldsymbol{x}\|_1\}, \quad (19)$$

where $\mathbf{Y}$ and $\boldsymbol{\Phi}$ was previously defined in (3) of Section III, with $\tilde{\mathbf{V}}=[\mathbf{V}_1^T,...,\mathbf{V}_N^T]^T$ and $\mathbf{V}_i=\mathbf{diag}(\boldsymbol{v}_i)$.

According to Proposition 2 in [15], it is easy to deduce the following Theorem 2 to build the equivalence between VPD-ADMM and its centralized counterpart, which represents the conditional convergence guarantee of the VPD-ADMM.

Theorem 2. (Conditional Convergence Guarantee) *Denote by $\hat{\boldsymbol{v}}_i$ the final result of $\boldsymbol{v}_i$ when early-stopping occurs. Then under Assumption 1, the updated results $\boldsymbol{x}_i^{(k)}$ of VPD-ADMM converge to the solution of the corresponding centralized problem (19) as $k\to\infty$, i.e.,*

$$\lim_{k\to\infty}\boldsymbol{x}_i^{(k)}=\hat{\boldsymbol{x}}_{Lasso},\forall i\in\mathcal{V},$$

where $\tilde{\boldsymbol{V}}=\tilde{\boldsymbol{V}}^{\triangle}\triangleq[\hat{\boldsymbol{V}}_1^T,...,\hat{\boldsymbol{V}}_N^T]^T$, $\hat{\boldsymbol{V}}_i=\boldsymbol{diag}(\hat{\boldsymbol{v}}_i)$.

Theorem 2 illustrates that the convergence result of VPD-ADMM is equivalent to the solution of its centralized LASSO problem (19) under the fixed estimation of $\tilde{\mathbf{V}}$. Moreover, it validates the robustness and stability of the VPD-ADMM under the proposed early-stopping rule. Building on Theorem 2, we will then formulate Theorem 3 as follows to provide its recovery guarantee. Since we do not focus on the measurement budget partitioning among nodes, in the following we assume that the number of measurements M in each node is identical. We clarify some of the definitions for a concise presentation of Theorem 3. Let $R=\{r_{\mathcal{S}_1},r_{\mathcal{S}_2},...r_{\mathcal{S}_K}\}$, where $\mathcal{S}_k$ denotes the

k-th element in the support $\mathcal{S}(\boldsymbol{x})$ and $r_{\mathcal{S}_k}$ denotes the number of nodes whose corresponding entry of $\hat{\boldsymbol{v}}_i$ is 1. The smallest value in R is denoted by R_{min}. Let

$$\mathbf{C} = \boldsymbol{\Phi}\tilde{\mathbf{V}}^{\Delta} = \begin{bmatrix} \mathbf{A}_1\hat{\mathbf{V}}_1 \\ \vdots \\ \mathbf{A}_N\hat{\mathbf{V}}_N \end{bmatrix} \in \mathbb{R}^{MN\times T}. \tag{20}$$

Upon using the subscript S to denote the sub-matrix concatenating the selected columns corresponding to the support $\mathcal{S}(\boldsymbol{x})$, we have $\mathbf{C}_S = \boldsymbol{\Phi}_S\tilde{\mathbf{V}}_S^{\Delta} \in \mathbb{R}^{MN\times K}$, where $\boldsymbol{\Phi}_S = \mathbf{diag}(\{\mathbf{A}_{iS}\})$, $\tilde{\mathbf{V}}_S^{\Delta} = [\hat{\mathbf{V}}_{1S}^T, ..., \hat{\mathbf{V}}_{NS}^T]^T$. Similarly, we define $\mathbf{C}_{S^c} = \boldsymbol{\Phi}_{S^c}\tilde{\mathbf{V}}_{S^c}^{\Delta}$, where the subscript S^c denotes the complement of the support set.

Theorem 3. (Recovery Guarantee) *Consider the linear sensing model (1) and the measurement matrix $\boldsymbol{A}_i \in \mathbb{R}^{M\times T}$ with i.i.d. random Gaussian entries obeying $\boldsymbol{N}(0, 1/M)$. The additive noise is $\boldsymbol{w}_i \sim \boldsymbol{N}(0, \sigma^2\boldsymbol{I}_M)$.*

Define C_{min} as the smallest eigenvalue of $\frac{1}{MN}\boldsymbol{C}_S^T\boldsymbol{C}_S$. If $R_{min} \geq 1$ and the regularization parameter λ satisfies $\lambda > 2\sqrt{\frac{2\zeta^2 logT}{MN}}$ with $\zeta^2 \triangleq \sigma^2 + \frac{1}{M}\max_i \|(\boldsymbol{V}_i^* - \hat{\boldsymbol{V}}_i)\boldsymbol{x}\|_2^2$, then as $MN \to \infty$ the following properties hold true with probability higher than $1 - 4e^{-c_1 MN\lambda^2} \to 1$ for some constant c_1:

a) *The VPD-ADMM converges to a unique solution $\hat{\boldsymbol{x}}$ with consensus achieved by nodes, i.e., $\hat{\boldsymbol{x}}_1 = .. = \hat{\boldsymbol{x}}_N = \hat{\boldsymbol{x}}$. The solution satisfies $\mathcal{S}(\hat{\boldsymbol{x}}) \subseteq \mathcal{S}(\boldsymbol{x})$ and the ℓ_∞ bound*

$$\|\hat{\boldsymbol{x}}_S - \boldsymbol{x}_S\|_\infty \leq \lambda\Big[\big\|(\boldsymbol{C}_S^T\boldsymbol{C}_S/MN)^{-1}\big\|_\infty + \frac{4\zeta}{\sqrt{C_{min}}}\Big] = h(\lambda).$$

Furthermore, if we assume that $\big\|(\boldsymbol{C}_S^T\boldsymbol{C}_S/MN)^{-1}\big\|_\infty = \mathcal{O}(1)$ for simplicity and choose $\lambda = \mathcal{O}(\sqrt{\frac{logT}{MN}})$, the ℓ_∞ bound can be formulated as the ℓ_2 norm expression of

$$\|\hat{\boldsymbol{x}}_S - \boldsymbol{x}_S\|_2 = \mathcal{O}(\lambda\sqrt{K}) = \mathcal{O}(\sqrt{\frac{K\log T}{MN}}).$$

b) *If the minimum magnitude of $\boldsymbol{x}$ is bounded by $|\boldsymbol{x}|_{min} > h(\lambda)$, then we have $\mathcal{S}(\hat{\boldsymbol{x}}) = \mathcal{S}(\boldsymbol{x})$ with correct sign.*

Proof. See Appendix B. □

To recover the global vector $\boldsymbol{x}$, Theorem 3 gives the necessary condition $R_{min} \geq 1$ imposed on the estimated masking vector $\hat{\boldsymbol{v}}_i$, requiring that the entries of $\hat{\boldsymbol{v}}_i$ corresponding to the support $\mathcal{S}(\boldsymbol{x})$ should necessarily be 1 in at least one node. Theorem 3 a) gives the upper bound of the global vector's ℓ_∞ estimation error, which is positively correlated with the estimation error of $\hat{\mathbf{V}}_i$ as indicated by the expression of ζ. More specifically, due to the sparsity of $\boldsymbol{x}$, the upper bound can be further reduced to be only related to the estimation error of $\hat{\mathbf{V}}_i$ within the support. Moreover, if we assume the total power of the global vector is bounded by $\|\boldsymbol{x}\|_2^2 \leq P$, then the expression of ζ^2 can be simplified to $\sigma^2 + \frac{P}{M}$. Note that we consider the Gaussian sensing matrix for this problem, which is typical in the analysis of compressed sensing. The performance on other sensing matrices should be considered in the case of specific applications, since it is tightly related to the specific structures of the matrices.

Theorem 2 and 3 show the effectiveness of the VPD-ADMM algorithm. We now turn to the analysis of the convergence property of the VPD-EM error-mitigation algorithm, also considering the eventual convergence where all the nodes step into Stage Two. We formulate Theorem 4 as follows.

Theorem 4. *Provided that the stepsize of the dual variables $\rho = \theta$ is sufficiently small. We define the Laplacian matrix of the communication network as $\boldsymbol{L} \triangleq \boldsymbol{D} - \boldsymbol{W}$. If the matrix $\boldsymbol{diag}(\{\hat{\boldsymbol{V}}_i\}) + \boldsymbol{L} \otimes \boldsymbol{I}_T$ has full rank, the primal and dual variables optimized by the VPD-EM algorithm converge to an optimal primal-dual solution for problem (15) for each node.*

Proof. See Appendix C. □

VI. Experiments

In this section, we evaluate the performance of the proposed algorithms in the multi-view sensing scenario. Consider the linear measurement model of (1). Again, since we do not focus on managing the measurement budget among sensors, we set $M_1 = M_2 = ... = M_N = M$ for simplicity. For all of our experiments, the entries of the sensing matrices $\mathbf{A}_i \in \mathbb{R}^{M\times T}$ are generated randomly according to the Gaussian distribution $\mathbf{N}(0, 1/M)$. We fix the length of the global state vector to $T = 500$ and its sparsity to $K = 50$. Its support $\mathcal{S}(\boldsymbol{x})$ is generated uniformly at random and the nonzero values are drawn from a standard Gaussian distribution. The probability of blockage is identical for all sensors and denoted by p. In the support $\mathcal{S}(x)$, $\boldsymbol{v}_i$'s entries are generated using a binomial distribution with probability $1 - p$. The other entries of $\boldsymbol{v}_i$ outside the support are set to 0. Then the local observable vector $\boldsymbol{z}_i$ is derived along with $\boldsymbol{z}_i = \boldsymbol{x} \odot \boldsymbol{v}_i$. The additive noise $\boldsymbol{w}_i \in \mathbb{R}^M$ is drawn from $\mathbf{N}(0, \sigma^2)$. We define $\text{SNR}_i \triangleq \frac{1}{T}\|\boldsymbol{z}_i\|_2^2/\sigma^2$, which is identical for all sensors.

As there is no existing approach applicable to this scenario, we consider the standard LASSO with ADMM solver and design another two baselines for comparison:

- Standard LASSO with Distributed ADMM (D-LASSO): Each sensor ignores the its unobservable components, which are considered as an additive noise on the measurements. The problem is formulated as standard LASSO and solved via distributed ADMM.
- Independent Recovery and Average Scheme (IRAS): Each sensor independently estimates the locally observable vector $\boldsymbol{z}_i$ using a conventional compressed sensing algorithm (solving ℓ_1 optimization). To recover the global state vector, we assume a fusion center collecting all the estimated $\hat{\boldsymbol{z}}_i$. The support of the global vector is recovered by the union of the supports of $\hat{\boldsymbol{z}}_i, i \in \mathcal{V}$, and its value is recovered by averaging the corresponding non-zero values in $\hat{\boldsymbol{z}}_i, i \in \mathcal{V}$.
- Distributed Optimization with Additive Joint Sparsity Model (D-AJSM) [20]: Assume that the missing or blocked entries resulted from the additive noise $\boldsymbol{e}_i$, which is sparse and has negative values of the blocked entries of $\boldsymbol{x}$, i.e., $\boldsymbol{z}_i = \boldsymbol{e}_i + \boldsymbol{x}$. Then the problem is formulated into a modified conventional JSM-1 problem, with additional sparsity constraints subject to the locally observable vectors $\boldsymbol{z}_i$. Applying the distributed algorithm proposed in

[20] with the additional sparsity constraints, we have our second baseline scheme, namely D-AJSM.

We define a pair of metrics for characterizing the recovery performance: the average mean square error (AMSE) of the global vector and AMSE of the local vector as:

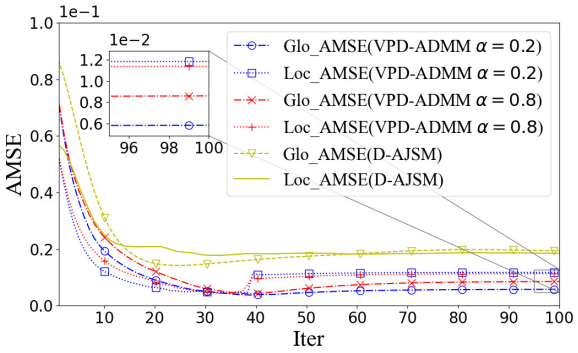
$$\text{Glo\_AMSE} = \frac{1}{N}\sum_{i\in\mathcal{V}}\frac{1}{T}\|\hat{\boldsymbol{x}}_i - \boldsymbol{x}\|_2^2,$$

$$\text{Loc\_AMSE} = \frac{1}{N}\sum_{i\in\mathcal{V}}\frac{1}{T}\|\hat{\boldsymbol{z}}_i - \boldsymbol{z}_i\|_2^2.$$

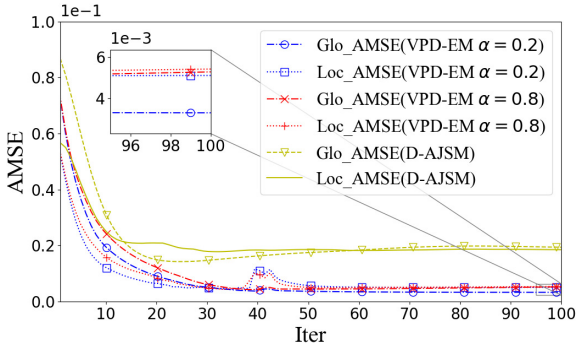
The network is generated randomly relying on the *Erdos_Renyi* network model using *networkx* package in Python, where the number of nodes is set to $N = 6$ and the probability of connectivity is 0.8.

A. Convergence Curve

In this part we characterize the convergence curves of both VPD-ADMM and VPD-EM algorithms and compare them to D-AJSM. We fix the number of measurements in each node to $M = 80$ and the probability of blockage in each node to $p = 0.2$ using SNR $= 12dB$. The penalty parameter in (9) is chosen either as $\alpha = 0.2$ or $\alpha = 0.8$. For a concise representation, we plot the convergence curves of VPD-ADMM and VPD-EM in two figures respectively, as seen in Fig. 2.



(a) Convergence Curves of VPD-ADMM v.s. D-AJSM



(b) Convergence Curves of VPD-EM v.s. D-AJSM

Fig. 2. Convergence Curves

Observe from Fig. 2 that both of the proposed algorithms perform better than D-AJSM w.r.t. their final convergence results. Additionally, Fig. 2(a) shows that the selection of α has an impact on the convergence result of VPD-ADMM, where $\alpha = 0.2$ outperforms $\alpha = 0.8$ in this setting. A more comprehensive comparison of different penalty parameters will be given in Section VI-F. In contrast to VPD-ADMM, VPD-EM is less sensitive to the selection of α and compensates for the estimation error imposed by the early stopping on the local vector, which benefits from its error-mitigation mechanism. A comparison between the enlarged excerpts in Fig. 2(a) and (b) indicates that VPD-EM has lower recovery error.

In addition, to represent the effectiveness and necessity of the discretization and early-stopping rule of $\boldsymbol{v}_i$ as proposed in Section IV-A, the convergence curves of the proposed algorithms are compared with the implementation without discretization (termed as NoDiscre) and another one with discretization but without early-stopping (termed as NoEarlyStop). The results are shown in Fig. 3. It could be observed that there exist two key points in the convergence curves, one is the discretization point and another one is the early-stopping point. Without discretization of $\boldsymbol{v}_i$ determined by the metric in (10), the implementation NoDiscre tends to become divergent since then. Meanwhile, without the early-stopping rule decided by (13), the implementation NoEarlyStop is divergent and unstable since then. This comparison also validates the effectiveness of the determination rule in (10) and (13).

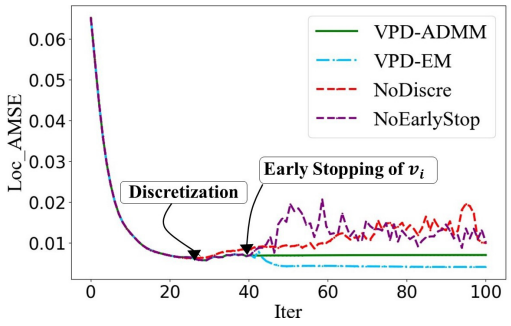
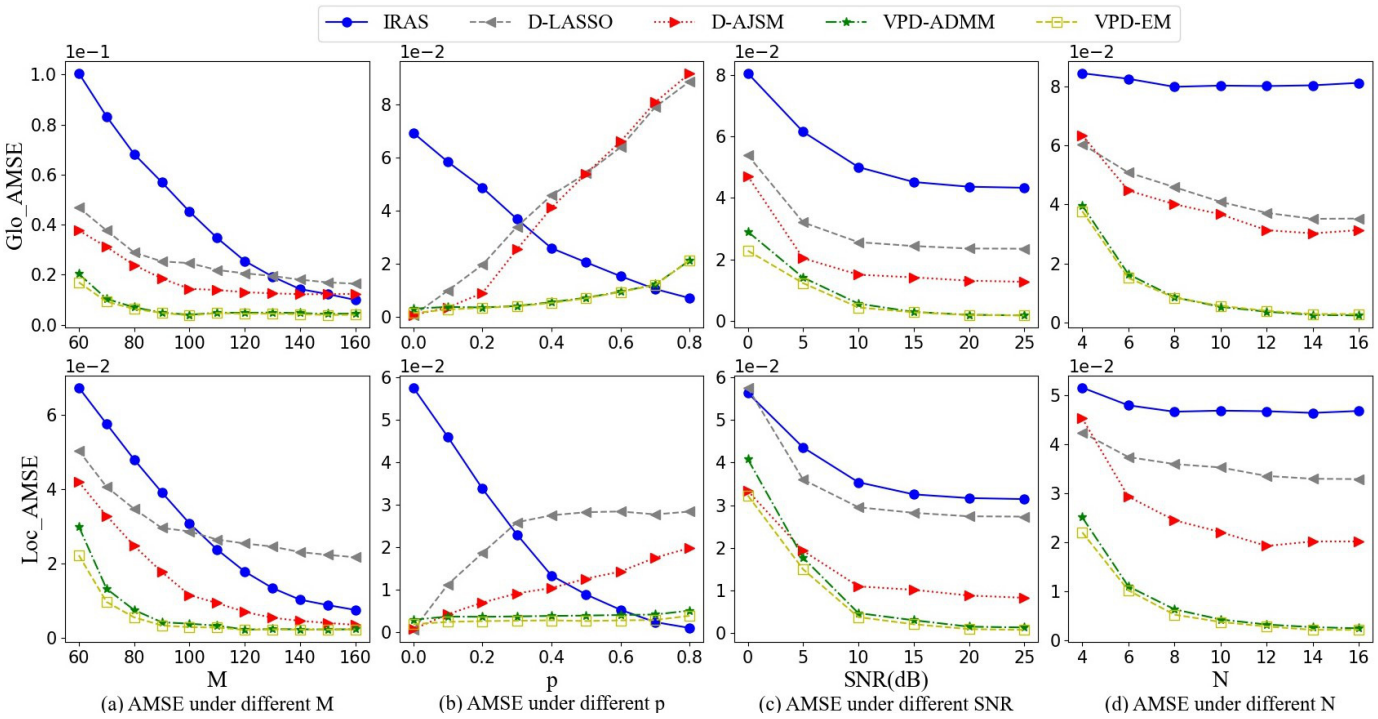


Fig. 3. Comparison of Convergence Curves

B. Performance Under Different Number of Measurements

In this subsection, we evaluate the performance of the proposed algorithms for different number M of measurements in each node. The two baseline methods (IRAS and D-AJSM) described above are also evaluated for comparison. We fix $p = 0.2$ and set SNR $= 12dB$. For each node, we generate 5 different $\mathbf{A}_i$ for each M, 2 different $\boldsymbol{x}$ and 4 different $\boldsymbol{v}_i$ for each $\boldsymbol{x}$. The results are averaged over the $5 \times 2 \times 4 = 40$ trials and are shown in Fig. 4(a).

It can be observed from Fig. 4(a) that the recovery error decreases upon increasing the number of measurements in each node, both for the global and the local vectors. Among all the five algorithms, IRAS has the worst recovery accuracy when M is small, which is resulted from its independent estimation in each node, i.e., without making any use of the measurements gleaned from other nodes. This highlights the effectiveness of the cooperation among nodes. On the other hand, the standard LASSO shows poor performance when M is large, especially for the estimation of local vector. This is because that the standard LASSO simply ignores the blocked components without considering the individual local vector in each sensor. Thus D-LASSO cannot be applied



(a) AMSE under different M (b) AMSE under different p (c) AMSE under different SNR (d) AMSE under different N

Fig. 4. AMSE under different conditions

in effectively estimating the local vector. Additionally, the proposed algorithms have better performance than the D-AJSM as also shown in Section VI-A. The main reason lies in the different joint sparsity models of D-AJSM and the proposed FJSM. By exploiting a more explicit relationship between the local observable vector $\boldsymbol{z}_i$ and the global vector $\boldsymbol{x}$, FJSM actually puts more constraints on the data space of $\boldsymbol{z}_i$, compared with the simple summation of two independent variables $\boldsymbol{e}_i$ and $\boldsymbol{x}$ in D-AJSM. Thus, the proposed algorithms are more efficient than D-AJSM, while D-AJSM needs more measurements to recover the vectors.

Meanwhile, compared to VPD-ADMM, VPD-EM shows a better performance, which is notable w.r.t. the local recovery, but not so pronounced w.r.t. the global recovery. The reason lies in the error-mitigation mechanism, which is designed mainly for reducing the local estimation error. Furthermore, both Theorem 2 and 3 guarantee a good recovery of the global vector for VPD-ADMM, even under inaccurate $\hat{\boldsymbol{v}}_i$. Last but not least, the performance difference between VPD-ADMM and VPD-EM becomes smaller upon increasing M, which is as expected because a larger M gives a more accurate $\hat{\boldsymbol{v}}_i$ for VPD-ADMM and the estimation error becomes smaller.

C. Performance Under Different Probability of Blockage

In this subsection, we compare the recovery performance of the four algorithms for different probability of blockage p. We fix the number of measurements in each node as $M = 100$ and set SNR $= 12dB$. We generate 5 different $\mathbf{A}_i$, 2 different $\boldsymbol{x}$ and 4 different $\boldsymbol{v}_i$ for each $\boldsymbol{x}$ and each p. The results are averaged over 40 trials and shown in Fig. 4(b).

Fig. 4(b) gives us valuable insights. To begin with, we compare the performance of the four algorithms in Fig. 4(b). It can be seen that in the cases of $0 < p < 0.7$, both the VPD-ADMM and VPD-EM outperform the other three methods, and as p increases, the two proposed algorithms have more remarkable advantages over the D-AJSM and D-LASSO. In fact, even under the severe blockage condition where $p = 0.5$, both of the proposed algorithms have remarkable recovery performance. This indicates that our proposed methods are more efficient and stable in hostile blocking scenario than D-AJSM or D-LASSO. This accrues from the specifically designed FJSM, resulting in better exploitation of the measurements in each node. Furthermore, similar to the results and analysis of Subsection VI-B, VPD-EM has lower estimation error than VPD-ADMM.

Secondly, we now consider the specific trends of the different algorithms. In Fig. 4(b), the curves of IRAS show different trends from the other four algorithms. This is because the IRAS recovers the local vector independently in each node, and the local vector's sparsity increases with larger p, hence resulting in more accurate estimation under the same M. Thus the global vector, estimated by the union of all the nodes' recovered local vector, has lower recovery error for larger p. Under hostile blockage conditions, where the observable entries of nodes have limited overlap and the information gleaned from other nodes is not useful enough for local estimation, the individual recovery based IRAS naturally has better performance than the collaborative ones. In the following proposition, we theoretically formulate the conditions, where the independent recovery outperforms the proposed algorithms w.r.t. the upper bound of the recovery error, providing guidance for selecting the most appropriate algorithm under realistic conditions.

Proposition 2. *When the probability of blockage is higher than* $1 - \frac{c\cdot\zeta^2}{\sigma^2 N}$*, i.e.,* $1 - \frac{c\cdot\zeta^2}{\sigma^2 N} \leq p \leq 1 - \frac{1}{N}$*, the upper bound of the recovery error of IRAS is smaller than that of the proposed collaborative methods. Here* ζ, σ *are defined in Theorem 3,*

and c is a constant concerned with the sensing matrix.

Proof. See Appendix D. □

On the other hand, Fig. 4(b) shows that all of the other four algorithms exhibit increased recovery error for the global vector for larger p. This can be understood by Proposition 1 or from the perspective of the amount of information available. As p increases, under the same M in nodes, each entry of the global vector has a reduced number of observations, hence leading to increased errors. Compared to D-AJSM, as shown in Fig. 4(b), both VPD-ADMM and VPD-EM exhibit a relatively flat trend under larger p, while D-AJSM has escalating errors.

Last but not least, we can observe that under the perfectly non-blocked scenario of $p = 0$, D-AJSM and D-LASSO outperform our two proposed methods. This is because D-AJSM is based on the additive model and the recovery can be readily accomplished with the additive noise equal to $\mathbf{0}$. Meanwhile, when $p = 0$, the problem degrades into the standard distributed LASSO problem with each sensor observing the global vector, which can be efficiently solved by D-LASSO. while the pair of proposed methods are specifically designed for scenarios of missing entries and the estimation error of $\boldsymbol{v}_i$ leads to increased recovery error.

D. Performance Under Different SNRs

In this subsection, we fix $M = 100$ and $p = 0.3$, while the SNR is set to $[0, 5, 10, 15, 20, 25]$ in dB. The results are averaged over 40 trials and are shown in Fig. 4(c).

Observe from Fig. 4(c) that the estimation error reduces upon increasing the SNR. As shown in Fig. 4(c), in the case of SNR $= 0$, compared to D-AJSM, VPD-ADMM has a higher estimation error w.r.t. the local vectors, which is caused by the inaccurate estimation of the masking vectors at low SNRs. The error-mitigation method VPD-EM overcomes this drawback and attains better performance than D-AJSM. Additionally, as the SNR increases, the estimation errors of IRAS, D-AJSM and D-LASSO converge to higher values than those of the proposed algorithms. This is because IRAS recovers the vector in each node independently, while D-AJSM does not fully exploit the measurements among nodes due to the additive joint sparsity model. D-LASSO treats the blockage as measurement noise, which increases the power of noise and cannot make good estimation of the local vector. Thus the three baseline methods are inefficient for this multi-view scenario.

E. Performance Under Different Number of Sensors

In this part, we evaluate the performance under different number of sensors, to show their effectiveness under larger scale of network. Specifically, we fix the number of measurements in each sensor as $M = 60$ and the connectivity probability between sensors as 0.5. The probability of blockage is fixed as $p = 0.4$ and $SNR = 12dB$. The network topology is generated using the same way as illustrated before and the results after averaging 40 trials are shown as Fig. 4(d).

As shown in Fig. 4(d), with the number of sensors increasing, the collaborative methods shows better performance while the independent recovery cannot benefit from the larger number of sensors. Meanwhile, compared with the baseline methods, the proposed methods shows a greater advantage with more sensors. Moreover, as N increases, the effectiveness of VPD-EM reduces, due to more accurate estimation of the masking vectors in VPD-ADMM.

F. Impact of the Penalty Term

Before mapping the estimated masking vectors into the Boolean set, the penalty term $g_i(\boldsymbol{v}_i)$ is added in the updating to make the non-integer values more costly. As discussed in Section IV-A, the effect of the penalty term may be viewed as a soft mapping towards the Boolean set $\{0, 1\}$, accelerating the process for $\boldsymbol{v}_i$ to approximate the Boolean distribution. As it transpires from our analysis, a larger value of α leads to faster convergence. However, the analysis of the penalty term on the final estimation accuracy remains an open problem, but we evaluate its impact experimentally in this subsection.

We select the penalty parameter α from the set of $\{0, 0.3, 0.6, 0.9, 0.99\}$, where the last term is added for considering $\alpha < 1$. Let SNR $= 10dB$. We compare the resultant recovery errors of VPD-ADMM under these penalty parameters, given different number of measurements M. We set $p = 0.2$ and $p = 0.4$ respectively. The results are averaged over 40 trials and are shown in Table I and II.

We can observe from Table I that the gain obtained by the penalty term reduces for larger M. Specifically, when $M = 60$, the reduction of the estimation error is approximately 4×10^{-3} for the global vector and 6×10^{-3} for the local vector. However, the reduction for $M = 90$ is as small as 0.6×10^{-3}. This is because with more measurements, the calculated values of $\boldsymbol{v}_i$ are located nearer to the boolean set with higher probability and thus the benefit of the penalty term reduces. Moreover, we can see that the best penalty parameter reduces upon increasing M. For example, $\alpha = 0.9$ performs best for $M = 60$ and 70, while $\alpha = 0.6$ for $M = 80$. It is indeed expected since having more measurements diminishes the benefit of penalty term, as mentioned before, and a smaller penalty parameter works better with less bias.

Secondly, comparing Table I and II, we can observe that the best-performed penalty parameter reduces with larger p. This is because a larger p indicates a larger sparsity of the local vector, whose recovery accuracy increases and the estimated masking vectors are closer to the boolean set. As analyzed before, a smaller α works better in such conditions. In general, the results show that the penalty term is more beneficial in the condition of fewer measurements or smaller blockage, which also gives us insights about selecting the penalty parameter α. As M or p increases, α is suggested to be reduced. When M or p is large enough, the penalty term can be removed by letting $\alpha = 0$.

G. Performance Under Unbalanced Number of Measurements Among Sensors

The above simulations consider the simple condition where the number of measurements is equal among sensors. To provide a comprehensive evaluation, we investigate the performance of the algorithms under unbalanced number of measurements. Specifically, for each node $i \in \mathcal{V}$, it randomly chooses

TABLE I. Recovery AMSE comparison under $p = 0.2$ (10^{-3})

	$M = 60$		$M = 70$		$M = 80$		$M = 90$	
α	Glo_AMSE	Loc_AMSE	Glo_AMSE	Loc_AMSE	Glo_AMSE	Loc_AMSE	Glo_AMSE	Loc_AMSE
0	19.08	26.57	13.26	17.92	7.31	7.15	6.47	4.70
0.3	19.64	26.38	12.73	16.52	6.97	6.32	5.88	4.22
0.6	18.78	23.98	10.66	13.98	7.66	6.87	6.76	4.45
0.9	15.49	20.08	10.44	14.09	7.66	7.61	8.22	6.08
0.99	15.93	21.42	10.42	13.80	9.08	8.01	8.43	6.13

TABLE II. Recovery AMSE comparison under $p = 0.4$ (10^{-3})

	$M = 60$		$M = 70$		$M = 80$		$M = 90$	
α	Glo_AMSE	Loc_AMSE	Glo_AMSE	Loc_AMSE	Glo_AMSE	Loc_AMSE	Glo_AMSE	Loc_AMSE
0	29.58	24.74	18.34	15.99	11.28	7.34	9.00	4.59
0.3	28.31	23.08	17.80	15.92	10.85	6.55	10.07	5.08
0.6	27.10	22.56	19.02	14.28	12.68	7.45	10.82	5.40
0.9	29.18	20.27	19.38	13.84	15.04	9.14	13.15	7.04
0.99	28.86	23.05	20.43	15.08	14.92	9.62	14.78	7.91

its M_i from the Number set $[50, 60, 70, 80, 90, 100]$ for 5 times and generates 5 different $\mathbf{A}_i$. The probability of blockage is set to $p = 0.2$, equal for all sensors and we set SNR $= 12dB$. The results are averaged over 40 trailswith 5 different $\mathbf{A}_i$, 2 different $\boldsymbol{x}$ and 4 different $\boldsymbol{v}_i$, and are shown in Table III. Table III verifies the effectiveness and efficiency of the proposed algorithms under unbalanced number of measurement among sensors.

TABLE III. AMSE Comparison Under Unbalanced M_i (10^{-2})

	IRAS	D-LASSO	D-AJSM	VPD-ADMM	VPD-EM
Glo	9.18	3.37	3.00	0.89	0.84
Loc	6.40	3.53	3.10	0.71	0.64

Additionally, the measurement budget among sensors is another important problem to be considered, where M_i could be unbalanced among sensors. The measurement budget should be highly related to the sensing capability of each sensor, as proved rigorously in Theorem 1, i.e., the number of measurements required for each sensor must account for the observable features unique to that sensor. To give a simple yet effective investigation of this rule, we simulate on an extreme condition. Specifically, we suppose that there are a certain number of indices in $\boldsymbol{x}$ only observable to sensor 0, and this number is denoted by $|\Pi(\{0\})|$ as defined in Section III. We set the number as $5, 15, 20, 25$ respectively, and simulate the recovery results under them. Meanwhile, to ensure that the recovery performance reflects the impact of M_0, we set $M_i = 100$, enough for other sensors. The averaged recovery accuracy of the global vector over 40 trails is shown in Fig. 5.

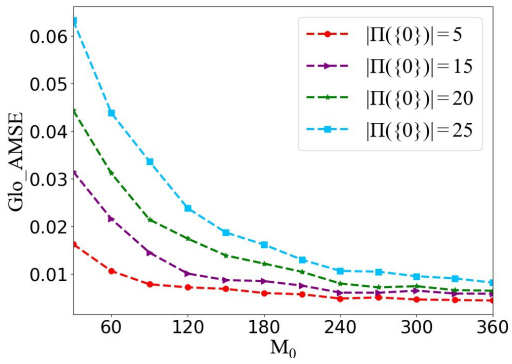


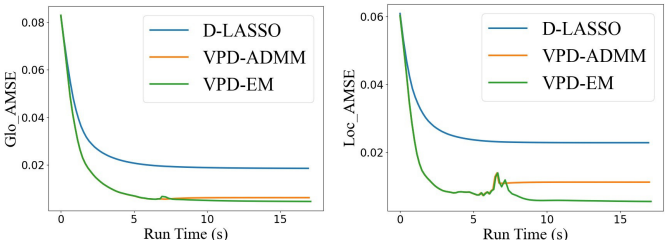
Fig. 5. AMSE under different M_0

Fig. 5 shows that the number of measurements in sensor 0 is vital for the recovery performance of the global vector, resulted from the fact that there are some indices in $\boldsymbol{x}$ only observable to it. Moreover, as $|\Pi(\{0\})|$ becomes larger, more measurements are required for sensor 0, which is consistent with the conclusion in Theorem 1.

H. Performance Enhancement Versus Run time

Although the proposed algorithms show better recovery performance compared with the standard method D-LASSO, they are more complex on computational cost Jwith two primal variables to update. Thus, in the part, we compare the recovery error of the D-LASSO and the proposed algorithms versus run time. Specifically, we consider $N = 6$, $p = 0.2$ and SNR $= 12dB$ as the setting in Section VI-A. We set $M = 80$ and $M = 160$ respectively to see the impact of M.

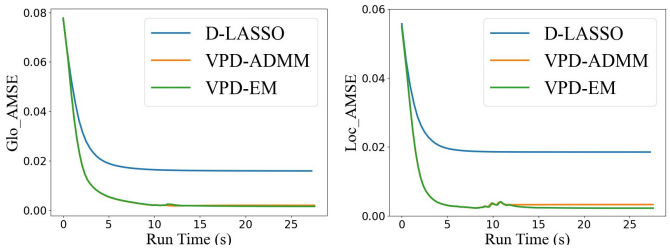
After one iteration of local update, each node transmits the updated results of $\boldsymbol{x}_i$ to its neighboring nodes. Since such transmission mechanism is the same for the algorithms, including D-LASSO and our proposed algorithms, we fairly ignore the transmission/communication time and only consider the computation time. Besides, for each distributed algorithm, we consider the synchronous setting. Thus, in each iteration, we take the time consumption of the slowest node as the time cost in this iteration. The algorithms are implemented in Python 2.7 with Intel Core 15-9400F CPU with 2.9Hz and the recovery error comparisons versus run time are shown in Fig. 6 and Fig. 7.



(a) Glo_AMSE versus run time (b) Loc_AMSE versus run time

Fig. 6. Performance comparison versus run time (M=80)

It can be observed from Fig. 6 and 7 that the two proposed algorithms are more efficient than the conventional D-LASSO method which treats the blocked components as measurement noise. Moreover, as the number of measurements increases, all of the algorithms take more time in each iteration, and



(a) Glo_AMSE versus run time (b) Loc_AMSE versus run time

Fig. 7. Performance comparison versus run time (M=160)

the enhancement of VPD-EM reduces compared with VPD-ADMM, due to more accurate estimation of $\boldsymbol{v}_i$.

VII. Conclusion

Novel DCS solutions have been conceived for recovering sparse vectors from multi-view compressed sensing measurements. To tackle the problem, a distributed recovery algorithm termed as VPD-ADMM was proposed, where each node alternately updates the local masking vector and the common global vector. The estimated global vector is shared among neighboring nodes over the network for achieving consensus. By appropriately curtailing the update of the masking vectors, the algorithm guarantees convergence, as shown by our theoretical analysis. However, this early termination way increases the estimation error of the masking vectors, hence VPD-EM was specifically designed for the estimation error of the local vectors and the associated convergence analysis was also provided. Compared to the independent recovery in each node and to another baseline method relying on the additive joint sparsity model, the proposed algorithms make better use of the measurements in each node and are more efficient in the specific DCS problem considered. Numerical experiments validate the superiority of our algorithms.

There are numerous open research topics, such as the study of the measurement budgets sharing among sensors, taking into consideration the number of observable entries determined by sensing ability of each sensor. Moreover, to achieve the best performance of the sensing system, an integrated procedure should be designed for striking tradeoff among the sensing, computation and communication cost. Furthermore, it is worthwhile applying the proposed algorithms in real sensing problems and further improving the algorithms by exploiting specific scenarios.

Appendix

A. Proof of Theorem 1

Proof. We prove each part of Theorem 1 sequentially, which is partially inspired by [4].

a) Since $\mathbf{Y} = \boldsymbol{\Phi}\tilde{\mathbf{P}}\boldsymbol{\theta}$, given known $\tilde{\mathbf{P}}$, there sufficiently exists a unique solution $\hat{\boldsymbol{\theta}} \in \mathbb{R}^K$ for this equation, as long as the rank of $\boldsymbol{\Phi}\tilde{\mathbf{P}}$ is K. So in the following we only have to prove the K-rank nature of $\boldsymbol{\Phi}\tilde{\mathbf{P}}$. Define

$$\boldsymbol{\Upsilon} = \boldsymbol{\Phi}\mathbf{P} = \begin{bmatrix} \mathbf{A}_1\mathbf{P}_1 \\ \vdots \\ \mathbf{A}_N\mathbf{P}_N \end{bmatrix} \in \mathbb{R}^{\tilde{M}\times K}.$$

Naturally, we have $\text{rank}(\boldsymbol{\Upsilon}) \leq K$. For each node i, we select m_i arbitrary rows from $\mathbf{A}_i\mathbf{P}_i$ so that

$$\sum_{i\in\Gamma} m_i = |\Pi(\Gamma, \mathbf{P})|, \quad m_i \leq M_i$$

for all the subset of nodes $\Gamma \subseteq \mathcal{V}$. This process is guaranteed by the condition (4). Then there exists a mapping $\mathcal{C} : \{1, 2, .., K\} \to \mathcal{V}$, assigning each element of the support to one of the sensors. Then we concatenate these rows and construct another matrix $\boldsymbol{\Upsilon}_1$. Since $\boldsymbol{\Upsilon}_1$ is constructed by selecting certain rows of $\boldsymbol{\Upsilon}$, we have $\text{rank}(\boldsymbol{\Upsilon}) \geq \text{rank}(\boldsymbol{\Upsilon}_1)$.

Now we reordering the columns of $\boldsymbol{\Upsilon}_1$ in the following way. Let $\boldsymbol{\Upsilon}_1(k)$ denote the k-th column of $\boldsymbol{\Upsilon}_1$. Following we introduce another matrix $\boldsymbol{\Upsilon}_2$ and for better illustration, we represent Algorithm 3 to show the method to construct it.

Algorithm 3: Construction of $\boldsymbol{\Upsilon}_2$

```
Initialize Υ_2 = [], Ψ = {1, 2, ..., K}.
for j = 1, ..., N do
    for k in Ψ do
        if C(k) = j then
            Ψ ← Ψ − {k},
            Υ_2 ← [Υ_2, Υ_1(k)].
Output Υ_2.
```

It then follows that $\text{rank}(\boldsymbol{\Upsilon}_1) = \text{rank}(\boldsymbol{\Upsilon}_2)$. We then observe three properties of $\boldsymbol{\Upsilon}_2$. Each entry is either 0 or a Gaussian random variable; All the Gaussian random variables are i.i.d.; All diagonal entries are Gaussian random variables. The first two properties hold true naturally with the construction of the matrix. The last property is confirmed because each diagonal entry of $\boldsymbol{\Upsilon}_2$ is an entry of some k-th column of $\mathbf{A}_j\mathbf{P}_j$, where we have $\mathcal{C}(k) = j$ according to the construction of $\boldsymbol{\Upsilon}_2$, and thus it is nonzero and remains Gaussian. Given these three properties, according to Lemma 3 of [4], $\boldsymbol{\Upsilon}_2$ is of full rank, i.e., $\text{rank}(\boldsymbol{\Upsilon}_1) = \text{rank}(\boldsymbol{\Upsilon}_2) = K$. So we have $\text{rank}(\boldsymbol{\Upsilon}) \geq \text{rank}(\boldsymbol{\Upsilon}_1) = K$ and $\text{rank}(\boldsymbol{\Upsilon}) = K$.

b) The unknown $\tilde{\mathbf{P}}$ and the corresponding vector $\boldsymbol{\theta}$ can be recovered using the algorithm of Appendix D in [4]. The main idea is to add a step of cross-validation, which takes the last measurement of each sensor and assembles the N measurements as a test set for validation. We refer the motivated readers to Appendix D in [4] for the detailed proof. This algorithm is akin to ℓ_0 norm minimization apart from an additional validation step, both of which are intractable in practical implementations and here they are only applied for the analysis of measurement bounds.

c) In this part, we prove that $\boldsymbol{\theta}$ cannot be uniquely recovered under the condition of (6). This is arranged by showing that the linear equations $\mathbf{Y} = \boldsymbol{\Phi}\tilde{\mathbf{P}}\boldsymbol{\theta}$ is under-determined even for a known $\tilde{\mathbf{P}}$. Equally, we prove that $\text{rank}(\boldsymbol{\Upsilon}) < K$.

Assume that $\Gamma \subseteq \mathcal{V}$ is a set which satisfies (6). We then define $\boldsymbol{\Upsilon}_1$ as the concatenation of $|\Pi(\Gamma, \tilde{\mathbf{P}})|$ columns selected from $\boldsymbol{\Upsilon}$, each of which corresponds to entries of $\Pi(\Gamma, \tilde{\mathbf{P}})$. To prove that $\text{rank}(\boldsymbol{\Upsilon}) < K$, we can equally prove that $\boldsymbol{\Upsilon}$ is not of full column rank. Since $\boldsymbol{\Upsilon}_1$ is constructed by selecting columns from $\boldsymbol{\Upsilon}$, we can prove that $\boldsymbol{\Upsilon}$ is not of full column rank by showing that the columns of $\boldsymbol{\Upsilon}_1$ cannot be linearly independent.

By exploiting the construction of $\boldsymbol{\Upsilon}_1$, we can formally show that the rows corresponding to nodes $j, j \notin \Gamma$ have all zero entries since these nodes cannot observe the indices in $\Pi(\Gamma, \tilde{\mathbf{P}})$ according to its definition. So the matrix $\boldsymbol{\Upsilon}_1$ has a total of $\sum_{i\in\Gamma} M_i$ nonzero rows and $|\Pi(\Gamma, \tilde{\mathbf{P}})|$ columns. The condition (6) shows that we have $\text{rank}(\boldsymbol{\Upsilon}_1) \leq \sum_{i\in\Gamma} M_i < |v(\Gamma, \tilde{\mathbf{P}})|$. Thus, the $|\Pi(\Gamma, \tilde{\mathbf{P}})|$ columns of $\boldsymbol{\Upsilon}_1$ cannot be linearly independent, which proves that $\boldsymbol{\Upsilon}$ is not of full column rank, i.e., we have $\text{rank}(\boldsymbol{\Upsilon}) < K$.

Note that the authors in [3, 39, 48] also gave analysis on the ℓ_0 norm recovery of sparse vector, which are based on other analytical methods or conditions. We refer readers to these works for further interest. □

B. Proof of Theorem 3

Proof. After terminating the update of $\boldsymbol{v}_i$, we have $\hat{\boldsymbol{v}}_i$ as the estimation result. Upon defining $\hat{\mathbf{V}}_i = \textbf{diag}(\hat{\boldsymbol{v}}_i)$, the measurement model in node i can be written as:

$$\boldsymbol{y}_i = \mathbf{A}_i\mathbf{V}_i^*\boldsymbol{x} + \boldsymbol{w}_i = \mathbf{A}_i\hat{\mathbf{V}}_i\boldsymbol{x} + \mathbf{A}_i(\mathbf{V}_i^* - \hat{\mathbf{V}}_i)\boldsymbol{x} + \boldsymbol{w}_i. \quad (21)$$

Let us now define the sum of the last two terms in (21) as

$$\begin{aligned}\tilde{\boldsymbol{w}}_i &\triangleq \mathbf{A}_i(\mathbf{V}_i^* - \hat{\mathbf{V}}_i)\boldsymbol{x} + \boldsymbol{w}_i \\ &= \sum_{t=1}^{T} \mathbf{A}_{i,(:,t)}(\boldsymbol{v}_i^*(t) - \hat{\boldsymbol{v}}_i(t))\boldsymbol{x}(t) + \boldsymbol{w}_i,\end{aligned} \quad (22)$$

where $\mathbf{A}_{i,(:,t)}$ denotes the t-th column of $\mathbf{A}_i$.

Define furthermore the centralized measurement model as the stacked measurements of all the nodes as:

$$\mathbf{Y} \triangleq \boldsymbol{\Phi}\tilde{\mathbf{V}}^*\boldsymbol{x} + \mathbf{W} = \boldsymbol{\Phi}\tilde{\mathbf{V}}^{\triangle}\boldsymbol{x} + \boldsymbol{\Phi}(\tilde{\mathbf{V}}^* - \tilde{\mathbf{V}}^{\triangle})\boldsymbol{x} + \mathbf{W}, \quad (23)$$

where $\mathbf{W} = [\boldsymbol{w}_1^T, ..., \boldsymbol{w}_N^T]^T$ denotes the stacked noise vector. Similarly to (22), we define

$$\widetilde{\mathbf{W}} \triangleq \boldsymbol{\Phi}(\tilde{\mathbf{V}}^* - \tilde{\mathbf{V}}^{\triangle})\boldsymbol{x} + \mathbf{W}.$$

Then it may be readily shown that $\widetilde{\mathbf{W}} = [\tilde{\boldsymbol{w}}_1^T, ..., \tilde{\boldsymbol{w}}_N^T]^T$.

The proof of the theorem is built on Theorem 1 of [49]. After the termination of $\boldsymbol{v}_i$'s update, we treat $\mathbf{C} = \boldsymbol{\Phi}\tilde{\mathbf{V}}^{\triangle}$ defined in (20) as the new measurement matrix along with the fixed estimated $\hat{\boldsymbol{v}}_i$. Note that the matrix $\mathbf{C}$ is determined after the termination of $\boldsymbol{v}_i$'s update, and it contains some submatrix $\mathbf{A}_i\hat{\mathbf{V}}_i$ having some all-zero columns, corresponding to the diagonal zero values estimated in $\hat{\mathbf{V}}_i$. So we harness the deterministic analysis of Theorem 1 in [49] and prove the theory asymptotically along with $MN \to \infty$, following three steps:

- Firstly, by treating the estimation error $\boldsymbol{\Phi}(\tilde{\mathbf{V}}^* - \tilde{\mathbf{V}}^{\triangle})\boldsymbol{x}$ as noise, we prove that $\widetilde{\mathbf{W}}$ is sub-Gaussian satisfying the distribution of noise in Theorem 1 of [49].
- Secondly, we prove the three necessary conditions of Theorem 1 in [49] are satisfied in this scenario and thus can derive the theorem for the centralized implementation.
- Lastly, according to Theorem 2, we arrive at the same converged solution as the VPD-ADMM algorithm.

We first derive the sub-Gaussian distribution of $\tilde{\boldsymbol{w}}_i$, where $\tilde{\boldsymbol{w}}_i$ is the sum of $T+1$ M-length vectors, where the first T vectors are denoted by

$$\boldsymbol{a}_{i,t} = \mathbf{A}_{i,(:,t)}\Big[\boldsymbol{v}_i^*(t) - \hat{\boldsymbol{v}}_i(t)\Big]\boldsymbol{x}(t), \quad t = 1, ..., T$$

and the last vector is the noise vector $\boldsymbol{w}_i \sim \mathbf{N}(0, \sigma^2)$.

Since each entry in $\mathbf{A}_i$ obeys $\mathbf{N}(0, 1/M)$, each entry in $\boldsymbol{a}_{i,t}$ obeys $\mathbf{N}(0, [(\boldsymbol{v}_i^*(t) - \hat{\boldsymbol{v}}_i(t))\boldsymbol{x}(t)]^2/M)$ with fixed $\hat{\boldsymbol{v}}_i(t)$, where $\boldsymbol{x}(t)$ can be treated as some unknown but fixed values. The zero-mean Gaussian variable with variance δ^2 is sub-Gaussian with δ. Then according to Lemma 1.7 of [50], $\tilde{\boldsymbol{w}}_i$ is also sub-Gaussian along with

$$\sum_{t=1}^{T} \frac{1}{M}[(\boldsymbol{v}_i^*(t) - \hat{\boldsymbol{v}}_i(t))\boldsymbol{x}(t)]^2 + \sigma^2 = \sigma^2 + \frac{1}{M}\|(\mathbf{V}_i^* - \hat{\mathbf{V}}_i)\boldsymbol{x}\|_2^2.$$

Define $\xi_i^2 = \sigma^2 + \frac{1}{M}\|(\mathbf{V}_i^* - \hat{\mathbf{V}}_i)\boldsymbol{x}\|_2^2$. Then according to the definition of sub-Gaussian distribution in [49], each entry z in $\tilde{\boldsymbol{w}}_i$ satisfies

$$\mathbb{E}[\exp(az)] \leq \exp[a^2\xi_i^2/2], \quad \text{for all} \quad a \in \mathbb{R}. \quad (24)$$

For each $\tilde{\boldsymbol{w}}_i$, its entries satisfy the property (24). Hence for each entry z in $\widetilde{\mathbf{W}}$, we have:

$$\mathbb{E}[\exp(az)] \leq \exp[a^2\zeta^2/2], \quad \text{for all} \quad a \in \mathbb{R},$$

where $\zeta^2 \triangleq \max_i \xi_i = \sigma^2 + \frac{1}{M}\max_i \|(\mathbf{V}_i^* - \hat{\mathbf{V}}_i)\boldsymbol{x}\|_2^2$. Then it transpires that each entry in $\widetilde{\mathbf{W}}$ is sub-Gaussian with ζ^2.

Secondly, we prove that the three conditions of Theorem 1 in [49] are satisfied in this scenario. When the total number of measurements in the network is $MN \to \infty$, we have:

$$|||\mathbf{C}_{S^c}^T\mathbf{C}_S(\mathbf{C}_S^T\mathbf{C}_S)^{-1}|||_\infty \leq (1-\gamma), \quad (25)$$

with probability one and $\gamma = 1$. For an $m \times n$ matrix O, this norm is given by $|||O|||_\infty = \max_{i=1,...,m}\sum_{j=1}^{n}|O_{i,j}|$. This is because as $MN \to \infty$, we have $\mathbf{C}_{S^c}^T\mathbf{C}_S \to \mathbf{0}$ with probability one as a result of the i.i.d. Gaussian random $\mathbf{A}_i$ and the linear sparsity.

Furthermore, if $\mathbf{C}_S^T\mathbf{C}_S \in \mathbb{R}^{K\times K}$ has K nonzero eigenvalues, Condition 2 of [49], namely,

$$\Lambda_{\min}\big(\frac{1}{MN}\mathbf{C}_S^T\mathbf{C}_S\big) \geq C_{\min}, \quad (26)$$

associated with $C_{min} > 0$ is satisfied. We will prove that the matrix $\mathbf{C}_S \in \mathbb{R}^{MN\times K}$ is of full column rank, when $R_{min} \geq 1$, which means that $\mathbf{C}_S$ has K nonzero singular values and equally means that $\mathbf{C}_S^T\mathbf{C}_S$ has K nonzero eigenvalues. We prove this in a similar way to the proof of Theorem 1. Let us now construct an auxiliary matrix $\mathbf{C}_S'$ in the following way. For $k = 1, ...K$, select an arbitrary row from the i-th submatrix of $\mathbf{C}_S$ (corresponding to node i), where $\hat{\boldsymbol{v}}_i$ has the entry of 1 corresponding to k. This process is guaranteed under the condition $R_{min} \geq 1$. Then the diagonal entries of $\mathbf{C}_S'$ are non-zero, and the other two properties in Lemma 3 of [4] are satisfied naturally. According to Lemma 3 of [4], the matrix $\mathbf{C}_S'$ is of full rank and thus $\text{rank}(\mathbf{C}_S) = K$.

Thirdly, when $\hat{\boldsymbol{v}}_i(t) = 1$ for all $t \in \mathcal{S}^c$, we have

$$\|\mathbf{C}_{:,j}\|_2^2 = \|\mathbf{A}_{1,(:,j)}\|_2^2 + ... + \|\mathbf{A}_{N,(:,j)}\|_2^2,$$

for all $j \in \mathcal{S}^c$. Based on the law of large numbers and the variance $\frac{1}{M}$ of $\mathbf{A}_i$'s entries, we have $\frac{1}{MN}\|\mathbf{C}_{:,j}\|_2^2 \to \frac{1}{M}$ with probability one as $MN \to \infty$. Then we have $(MN)^{-1/2}\max_{j\in\mathcal{S}^c}\|\mathbf{C}_{:,j}\| \to M^{-1/2} < 1$ with probability one. In any other cases where $\hat{\boldsymbol{v}}_i(t) = \boldsymbol{v}_i^*(t) = 0, t \in \mathcal{S}^c$, the

t-th column of $\mathbf{A}_i$ is all-zero and we have $\frac{1}{MN}\|\mathbf{C}_{:,j}\|_2^2 < \frac{1}{M}$. Thus, when $MN \to \infty$, with probability one, we have

$$(MN)^{-1/2} \max_{j \in \mathcal{S}^c} \|\mathbf{C}_{:,j}\| < 1. \tag{27}$$

According to the sub-Gaussian distribution of $\widetilde{\mathbf{W}}$ and (25-27), the conditions of Theorem 1 in [49] are all satisfied. Let us now consider the recovery results of the centralized model (23), which is the solution of the Lasso problem in (19). By exploiting the equivalence of the distributed implementation and its centralized counterpart given in Theorem 2, Theorem 3 is proved. □

C. Proof of Theorem 4

Proof. We rely on Theorem 3.1 of [33], which outlines the convergence properties of the ADMM associated with more than two blocks of functions or variables. According to Theorem 3.1 of [33], the variables $\{\boldsymbol{q}_i, \boldsymbol{\tau}_i\}$ and $\{\boldsymbol{z}_i, \boldsymbol{\beta}_i, \boldsymbol{x}_i\}$ converge to an optimal primal-dual solution, if the assumptions (a)-(g) in [33] hold true. Below we will modify the problem (15) for all nodes as in [33] and then show that the assumptions (a)-(g) in [33] are satisfied in the modified problem.

Let us define the vectors $\tilde{\boldsymbol{z}} = [\boldsymbol{z}_1^T, ..., \boldsymbol{z}_N^T]$, $\tilde{\boldsymbol{\beta}} = [\boldsymbol{\beta}_1^T, ..., \boldsymbol{\beta}_N^T]$, $\tilde{\boldsymbol{x}} = [\boldsymbol{x}_1^T, ..., \boldsymbol{x}_N^T]$ and functions:

$$f_1(\tilde{\boldsymbol{z}}) = \sum_{i=1}^{N} \|\boldsymbol{y}_i - \mathbf{A}_i \boldsymbol{z}_i\|_2^2 + \epsilon_1 \|\boldsymbol{z}_i\|_1,$$
$$f_2(\tilde{\boldsymbol{\beta}}) = \sum_{i=1}^{N} \epsilon_2 \|\boldsymbol{\beta}_i\|_1, \quad f_3(\tilde{\boldsymbol{x}}) = \sum_{i=1}^{N} \epsilon_3 \|\boldsymbol{x}_i\|_3.$$

Then the sum of (15) for all nodes can be recast as follows:

$$\begin{aligned} \min_{\tilde{\boldsymbol{z}}, \tilde{\boldsymbol{\beta}}, \tilde{\boldsymbol{x}}} \quad & f_1(\tilde{\boldsymbol{z}}) + f_2(\tilde{\boldsymbol{\beta}}) + f_3(\tilde{\boldsymbol{x}}), \\ s.t. \quad & \mathbf{E}_1(\tilde{\boldsymbol{z}}) + \mathbf{E}_2(\tilde{\boldsymbol{\beta}}) + \mathbf{E}_3(\tilde{\boldsymbol{x}}) = \mathbf{0}_{(N+D)T \times 1}, \end{aligned} \tag{28}$$

where $D = |\mathcal{E}|$ is the number of edges in the network $\mathcal{G}$. Since the graph is undirected, having D constraints is sufficient to describe all of the consensus constraints $\boldsymbol{x}_i = \boldsymbol{x}_j, j \in \mathcal{N}_i$. The matrices $\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3$ are defined as follows to represent the constraints:

$$\mathbf{E}_1 = [\check{\mathbf{E}}_1^T, \mathbf{0}_{NT \times DT}^T]^T, \mathbf{E}_2 = [\check{\mathbf{E}}_2^T, \mathbf{0}_{NT \times DT}^T]^T, \mathbf{E}_3 = [\check{\mathbf{E}}_3^T, \mathbf{B}^T]^T.$$

Here $\check{\mathbf{E}}_1$ is an $NT \times NT$ identity matrix, i.e., $\check{\mathbf{E}}_1 = \mathbf{I}_{NT}$ and $\check{\mathbf{E}}_2 = -\check{\mathbf{E}}_1$. $\check{\mathbf{E}}_3 = -\mathbf{diag}(\{\hat{\mathbf{V}}_i\}) \in \mathbb{R}^{NT \times NT}$, while $\mathbf{B}$ is the $DT \times NT$ matrix defined as

$$\mathbf{B} = [\mathbf{B}_{1,:}^T \cdots \mathbf{B}_{D,:}^T]^T \otimes \mathbf{I}_T,$$

where $\mathbf{B}_{t,:}$ is the t-th row of $\mathbf{B}$, associated with a single edge of the network. In $\mathbf{B}_{t,:}$, corresponding to the edge $\varepsilon_{ij}, i < j$, its i-th entry is set to 1 and j-th entry is -1. The definition of $\mathbf{B}$ describes the consensus constraints of $\boldsymbol{x}$ among nodes.

Then we only have to prove that problem (28) satisfies the Assumptions (a)-(g) in [33]. A similar proof was adopted in [20]. With the formulation of (28), we omit the proof of assumptions (a)-(e) and (g), which are naturally satisfied and are also discussed in the Appendix of [20]. We now turn to prove *(f) Each submatrix $\boldsymbol{E}_k$ has full column rank.*

According to the definition of $\mathbf{E}_1$ and $\mathbf{E}_2$, it may be readily seen that they both have full column rank. Based on the definition of $\mathbf{E}_3$, we have

$$\begin{aligned} \mathbf{E}_3^T \mathbf{E}_3 &= \left[\check{\mathbf{E}}_3^T, \mathbf{B}^T\right] \cdot \begin{bmatrix} \check{\mathbf{E}}_3 \\ \mathbf{B} \end{bmatrix} = \check{\mathbf{E}}_3^T \check{\mathbf{E}}_3 + \mathbf{B}^T \mathbf{B} \\ &= \mathbf{diag}(\{\hat{\mathbf{V}}_i\}) + (\mathbf{D} - \mathbf{W}) \otimes \mathbf{I}_T \\ &= \mathbf{diag}(\{\hat{\mathbf{V}}_i\}) + \mathbf{L} \otimes \mathbf{I}_T. \end{aligned}$$

Since the matrix $\mathbf{diag}(\{\hat{\mathbf{V}}_i\}) + \mathbf{L} \otimes \mathbf{I}_T$ is of full rank according to the condition, it is easy to show that $\mathbf{E}_3$ is of full column rank. □

D. Proof of Proposition 2

Proof. According to Theorem 3, we assume that $\left\|\left(\mathbf{C}_S^T \mathbf{C}_S / MN\right)^{-1}\right\|_\infty \leq c_1$ for simplicity and set its regularization as λ_1 then we have

$$\|\hat{\boldsymbol{x}}_{S,1} - \boldsymbol{x}_S\|_2 \leq \lambda_1 \sqrt{K} \big[c_1 + \frac{4\zeta}{\sqrt{C_{min,1}}}\big].$$

Here the subscript 1 indicates the recovery result and relative parameters of VPD-ADMM.

On the other hand, since IRAS works by taking average of the estimated non-zero values, its recovery performance can be bounded by that of the single node whose recovery error is the largest among nodes. Thus we can apply Theorem 1 in [49] to analyze IRAS, where the sparsity of signal is $K(1-p)$ for the local node. We use $\mathbf{A}_{i,S}$ to represent the matrix composed of the selected columns from the sensing matrix $\mathbf{A}_i$ corresponding to the support set $\mathcal{S}(\boldsymbol{z}_i)$, for an arbitrary node i. We assume that for all nodes $i \in \mathcal{V}$, there exists a constant c_2 such that

$$\max_{i \in \mathcal{V}} \left\|\left(\mathbf{A}_{i,S}^T \mathbf{A}_{i,s} / M\right)^{-1}\right\|_\infty \leq c_2.$$

If we choose the regularization λ_2, according to Theorem 1 in [49], we have

$$\|\hat{\boldsymbol{x}}_{S,2} - \boldsymbol{x}_S\|_2 \leq \lambda_2 \sqrt{K(1-p)} \big[c_2 + \frac{4\sigma}{\sqrt{C_{min,2}}}\big],$$

where the subscript 2 indicates the recovery result of IRAS. Here $C_{min,2}$ denotes the minimum value of $\frac{1}{M}\mathbf{A}_{i,S}^T \mathbf{A}_{i,S}$ for all node i.

Note that $\lambda_1 > 2\sqrt{\frac{2\zeta^2 logT}{MN}}$ according to Theorem 3 and $\lambda_2 > 2\sqrt{\frac{2\sigma^2 logT}{M}}$. We choose $\lambda_1 = c_3\sqrt{\frac{\zeta^2 logT}{MN}}$ and $\lambda_2 = c_3\sqrt{\frac{\sigma^2 logT}{M}}$ where $c_3 > 2\sqrt{2}$. Then it may be readily deduced that when

$$\sigma^2 \times (1-p) \times \log T \leq \frac{c \times \zeta^2 \log T}{N},$$

i.e., $p \geq 1 - \frac{c \cdot \zeta^2}{\sigma^2 N}$, the upper bound of $\|\hat{\boldsymbol{x}}_{S,2} - \boldsymbol{x}_S\|_2^2$ is smaller than that of $\|\hat{\boldsymbol{x}}_{S,1} - \boldsymbol{x}_S\|_2^2$. Here c is a constant concerned with the sensing matrix and it can be explicitly expressed as

$$c = \frac{c_1 + \frac{4\zeta}{\sqrt{C_{min,1}}}}{c_2 + \frac{4\sigma}{\sqrt{C_{min,2}}}}.$$

□

REFERENCES

[1] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, 2019.

[2] F. Liu, Y. Cui, C. Masouros, J. Xu, T. X. Han, Y. C. Eldar, and S. Buzzi, "Integrated sensing and communications: toward dual-Functional wireless networks for 6G and beyond," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 6, pp. 1728–1767, 2022.

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

[4] D. Baron, M. F.Duarte, M. B.Wakin, S. Sarvotham and R. G. Baraniuk, "Distributed compressed sensing," arXiv: 0901.3403 [cs.IT], 2009.

[5] X. Tong, Z. Zhang, Y. Zhang, Z. Yang, C. Huang, K. Wong and M. Debbah, "Environment Sensing Considering the Occlusion Effect: A Multi-View Approach," *IEEE Trans. Signal Process.*, pp. 1–17, 2022.

[6] N. Sugavanam, S. Baskar and E. Ertin, "High resolution MIMO radar sensing With compressive illuminations," *IEEE Trans. Signal Process.*, vol. 70, pp. 1448–1463, 2022.

[7] M. A. Herman and T. Strohmer, "High-Resolution Radar via Compressed Sensing," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2275–2284, 2009.

[8] Z. Lin, Y. Chen, X. Liu, R. Jiang, B. Shen, and G. Xin-Xin, "Optimized design for sparse arrays in 3-D imaging sonar systems based on perturbed Bayesian compressive sensing," *IEEE Sensors J.*, vol. 20, no. 10, pp. 5554–5565, 2020.

[9] Y. Qian, R. He, Q. Chen, G. Gu, F. Shi, and W. Zhang, "Adaptive compressed 3D ghost imaging based on the variation of surface normals," *Opt Express.*, vol. 27, no. 20, pp. 27862–27872, 2019.

[10] S. Cai and V. K. N. Lau, "Zero MAC latency sensor networking for cyber-physical systems," *IEEE Trans. Signal Process.*, vol. 66, no. 14, pp. 3814–3823, 2018.

[11] S. Cai and V. K. N. Lau, "Modulation-free M2M communications for mission-critical applications," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 2, pp. 248–263, 2018.

[12] J. Chen and X. Huo, "Theoretical results on sparse representations of multiple-measurement vectors," *IEEE Trans. Signal Process.*, vol. 54, pp. 4634–4643, 2006.

[13] J. D. Blanchard, M. Cermak, D. Hanle, and Y. Jing, "Greedy algorithms for joint sparse recovery," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1694–1704, 2014.

[14] S. H. Hsieh, W. J. Liang, C. Lu, and S. C. Pei, "Distributed compressive sensing: Performance analysis with diverse signal ensembles," *IEEE Trans. Signal Process.*, vol. PP, no. 99, pp. 1–1, 2020.

[15] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5262–5276, 2010.

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

[17] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Püschel, "D-ADMM: A communication-efficient distributed algorithm for separable optimization," *IEEE Trans. Signal Process.*, vol. 61, no. 10, pp. 2718–2723, 2013.

[18] 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, 2010.

[19] S. Patterson, Y. C. Eldar, and I. Keidar, "Distributed compressed sensing for static and time-varying networks," *IEEE Trans. Signal Process.*, vol. 62, no. 19, pp. 4931–4946, 2014.

[20] J. Matamoros, S. M. Fosson, E. Magli, and C. Antón-Haro, "Distributed ADMM for in-network reconstruction of sparse signals with innovations," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 4, pp. 225–234, 2015.

[21] S. M. Fosson, J. Matamoros, C. Antón-Haro, and E. Magli, "Distributed recovery of jointly sparse signals under communication constraints," *IEEE Trans. Signal Process.*, vol. 64, no. 13, pp. 3470–3482, 2016.

[22] D. Sundman, S. Chatterjee, and M. Skoglund, "Design and analysis of a greedy pursuit for distributed compressed sensing," *IEEE Trans. Signal Process.*, vol. 64, no. 11, pp. 2803–2818, 2016.

[23] G. Li, T. Wimalajeewa, and P. K. Varshney, "Decentralized and collaborative subspace pursuit: A communication-efficient algorithm for joint sparsity pattern recovery with sensor networks," *IEEE Trans. Signal Process.*, vol. 64, no. 3, pp. 556–566, 2016.

[24] K. Huang and J. Liu, "Inner–outer support set pursuit for distributed compressed sensing," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 3024–3039, 2018.

[25] S. Khanna and C. R. Murthy, "Decentralized joint-sparse signal recovery: A sparse Bayesian learning approach," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 1, pp. 29–45, 2017.

[26] G. Hannak, A. Perelli, N. Goertz, G. Matz, and M. E. Davies, "Performance analysis of approximate message passing for distributed compressed sensing," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 5, pp. 857–870, 2018.

[27] R. Nassif, S. Vlaski, and A. H. Sayed, "Distributed inference over networks under subspace constraints," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2019, pp. 5232–5236.

[28] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Püschel, "Distributed optimization with local domains: Applications in MPC and network flows," *IEEE Trans. Autom. Control*, vol. 60, no. 7, pp. 2004–2009, 2015.

[29] Y. Liu, W. Xu, G. Wu, Z. Tian, and Q. Ling, "Communication-censored ADMM for decentralized consensus optimization," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2565–2579, 2019.

[30] Z. Tian, Z. Zhang, J. Wang, X. Chen, W. Wang, and H. Dai, "Distributed ADMM with synergetic communication and computation," *IEEE Trans. Commun.*, vol. 69, no. 1, pp. 501–517, 2021.

[31] S. Boyd, N. Parikh and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010.

[32] T.-H. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus ADMM," *IEEE Trans. Signal Process.*, vol. 63, no. 2, pp. 482–497, 2015.

[33] Z. Luo, M. Hong, "On the linear convergence of the alternating direction method of multipliers," *Math. Programming*, 2017.

[34] A. Del Bue, J. Xavier, L. Agapito, and M. Paladini, "Bilinear modeling via augmented Lagrange multipliers (BAML)," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 8, pp. 1496–1508, 2012.

[35] A. Ahmed, B. Recht, and J. Romberg, "Blind deconvolution using convex programming," *IEEE Trans. Inf. Theory*, vol. 60, no. 3, pp. 1711–1732, 2014.

[36] I. Tosic and P. Frossard, "Dictionary learning," *IEEE Signal Process. Magazine*, vol. 28, no. 2, pp. 27–38, 2011.

[37] A. Aghasi, A. Ahmed, P. Hand, and B. Joshi, "Bilinear compressed sensing under known signs via convex programming," *IEEE Trans. Signal Process.*, vol. 68, pp. 6366–6379, 2020.

[38] K. Lee, Y. Wu, and Y. Bresler, "Near-optimal compressed sensing of a class of sparse low-rank matrices via sparse power factorization," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1666–1698, 2018.

[39] E. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[40] Y. Wang, W. Yin, J. Zeng, "Global convergence of ADMM in nonconvex nonsmooth Optimization," *J. Sci. Comput.*, vol. 78, no. 1, pp.29–63, 2019.

[41] J. Wang, L. Zhao, "Nonconvex generalization of Alternating Direction Method of Multipliers for nonlinear equality constrained problems," *Results Control and Optimization*, vol. 2, pp.100009, 2021.

[42] X. Liu and S. C. Draper, "The ADMM penalized decoder for LDPC codes," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 2966–2984, 2016.

[43] S. M. Fosson, "Non-convex approach to binary compressed sensing," *Proc. Asilomar Conf. Signals Syst. Comput.*, pp.1959-1963, 2018.

[44] S. M. Fosson and M. Abuabiah, "Recovery of binary sparse signals from compressed linear measurements via polynomial optimization," *IEEE Signal Process. Lett.*, vol. 26, no. 7, pp.1070-1074, 2019.

[45] Y. Wei, R. Wonjong, S. Boyd and J. M.Cioffi, "Iterative water-filling for Gaussian vector multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 50, no. 1, pp. 145–152, 2004.

[46] B. S. He, H. Yang, and S. L. Wang, "Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities," *J. Optimization Theory Applicat.*, vol. 106, 2000.

[47] S. Y. C. Song and V. Pavlovic, "Fast ADMM algorithm for distributed optimization with adaptive penalty," *Proc. 30th AAAI Conf. Artificial Intell.*, p. 753–759, 2016.

[48] M. Elad and A. M. Bruckstein, "A generalized uncertainty principle and sparse representations in pairs of bases," *IEEE Trans. Inf. Theory*, vol. 49, no. 9, pp. 2558–2567, 2002.

[49] M. J. Wainwright, "Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (LASSO)," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2183–2202, 2009.

[50] V. V. Buldygin and Y. V. Kozachenko, "Metric characterization of random variables and random processes. Transl. from the Russian by V. Zaiats," *Amer. Math. Soc.*, 2000.