2016

Distributed sparse signal recovery in networked systems

Puxiao Han
Virginia Commonwealth University

DISTRIBUTED SPARSE SIGNAL RECOVERY IN NETWORKED SYSTEMS

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

by

PUXIAO HAN

B.S., Peking University, China - August 2004 to July 2008

M.S., Peking University, China - September 2008 to July 2011

Adviser: Ruixin Niu, Ph.D.,

Assistant Professor, Department of Electrical and Computer Engineering

Virginia Commonwealth University

Richmond, Virginia

December, 2016

ACKNOWLEDGMENTS

First, I am deeply thankful to my advisor Dr. Ruixin Niu, for his kind, thorough, and professional guidance and help during my PhD study, including all the work described in this dissertation. It is his vision in statistical and sparse signal processing that enlightens me in my PhD research, and leads me to all the academic successes I have achieved.

Second, I want to thank all the other members in my committee: Dr. Alen Docef, Dr. Yuichi Motai, Dr. Yongjia Song, and Dr. J. Paul Brooks, for all your constructive feedback and comments, and in-depth questions regarding my defense and dissertation.

Furthermore, I would like to thank all my fellow teammates in the lab, Mengqi Ren, Jingyang Lu, and Maitham Al-Salman, for all the meaningful insights and suggestions I got from our conversations and discussions.

Last but not least, I want to thank my family — my parents, and my beloved wife — it is your unconditional and continuous love and support that give me the courage, strength, and determination, to be fully dedicated to my research and make all these accomplishments.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

DISTRIBUTED SPARSE SIGNAL RECOVERY IN NETWORKED SYSTEMS

By Puxiao Han, Ph.D.

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

Virginia Commonwealth University, 2016.

Advisor: Ruixin Niu, Ph.D.,

Assistant Professor, Department of Electrical and Computer Engineering

In this dissertation, two classes of distributed algorithms are developed for sparse signal recovery in large sensor networks. All the proposed approaches consist of local computation (LC) and global computation (GC) steps carried out by a group of distributed local sensors, and do not require the local sensors to know the global sensing matrix. These algorithms are based on the original approximate message passing (AMP) and iterative hard thresholding (IHT) algorithms in the area of compressed sensing (CS), also known as sparse signal recovery. For distributed AMP (DiAMP), we develop a communication-efficient algorithm GCAMP. Numerical results demonstrate that it outperforms the modified thresholding algorithm (MTA), another popular GC algorithm for Top-K query from distributed large databases. For distributed IHT (DIHT), there is a step size μ which depends on the ℓ_2 norm of the global sensing matrix $\mathbf{A}$. The exact computation of $\|\mathbf{A}\|_2$ is non-separable. We propose a new method, based on the random matrix theory (RMT), to give a very tight statistical upper bound on $\|\mathbf{A}\|_2$, and the calculation of that upper bound is separable without any communication cost. In the GC step of DIHT, we develop an-

other algorithm named GC.K, which is also communication-efficient and outperforms MTA. Then, by adjusting the metric of communication cost, which enables transmission of quantized data, and taking advantage of the correlation of data in adjacent iterations, we develop quantized adaptive GCAMP (Q-A-GCAMP) and quantized adaptive GC.K (Q-A-GC.K) algorithms, leading to a significant improvement on communication savings.

Furthermore, we prove that state evolution (SE), a fundamental property of AMP that in high dimensionality limit, the output data are asymptotically Gaussian regardless of the distribution of input data, also holds for DiAMP. In addition, compared with the most recent theoretical results that SE holds for sensing matrices with independent subgaussian entries, we prove that the universality of SE can be extended to far more general sensing matrices. These two theoretical results provide strong guarantee of AMP's performance, and greatly broaden its potential applications.

CHAPTER 1

INTRODUCTION

1.1 An Overview of Compressed Sensing

Compressed sensing (CS) [1, 2, 3, 4] has wide applications in various areas of signal processing, such as multimedia processing [5, 6, 7], power systems [8, 9, 10], signal detection [11, 12, 13], and manifold learning [14, 15], etc.

Given a sparse N-dimensional signal $\mathbf{s}_0$ which has at most K non-zero components (we say $\mathbf{s}_0$ is K-sparse, or the sparsity level of $\mathbf{s}_0$ is K), and a sensing matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$, we can obtain M measurements composing the vector

$$\mathbf{y} = \mathbf{A}\mathbf{s}_0 + \mathbf{e}, \tag{1.1}$$

where $\mathbf{e}$ is an additive noise. Recovery of $\mathbf{s}_0$ from $\mathbf{y}$ and $\mathbf{A}$ turns out to be solving the following optimization problem:

$$\min_{\|\mathbf{x}\|_0 \leq K} ||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2. \tag{1.2}$$

Typically, solving (1.2) is a NP-hard problem. However, if $\mathbf{A}$ is a matrix with entries being independent and identically distributed (i.i.d.) random variables drawn from a sub-Gaussian distribution and $M = O\left(K \log \frac{N}{K}\right)$, then with a very high probability, restricted isometry property (RIP) [16] will be satisfied [17], and $\mathbf{s_0}$ can be reconstructed with high accuracy by solving the least absolute shrinkage and selection operator (LASSO) problem [18]:

$$\min_{\mathbf{x}} \frac{1}{2}||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2 + \lambda||\mathbf{x}||_1, \tag{1.3}$$

where $\lambda > 0$ is called the regularization parameter and needs to be tuned if the

sparsity level K is not known in advance. In the rest of this dissertation, we assume that all entries of $\mathbf{A}$ are i.i.d. random variables following $\mathcal{N}(0, 1/M)$.

1.2 Applications of CS

1.2.1 Power Systems

In power systems, each entry of the state vector $\mathbf{x} \in \mathbb{R}^n$ depicts the voltage phase angle of the corresponding bus, and the control center monitors the system by taking measurement of $\mathbf{x}$ using the so-called measurement Jacobian matrix $\mathbf{H} \in \mathbb{R}^{m \times n}$[10, 19]:

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{1.4}$$

Unlike network traffic systems, the measurement model in (1.4) is over-complete, i.e., $m \geq n$. Anomaly in power systems is typically defined as a false vector $\mathbf{a} \in \mathbb{R}^m$ injected onto the measurement domain by some malicious users, which may cause the control center obtaining a wrong estimate of $\mathbf{x}$.

According to [8] and [20], traditional statistical tests will fail to detect a malicious attack if there exists $\mathbf{c} \in \mathbb{R}^n$ such that

$$\mathbf{a} = \mathbf{H}\mathbf{c} \tag{1.5}$$

Due to limited resources and other constraints of the malicious users, it is reasonable to assume that $\mathbf{a}$ is sparse. In [8], the authors proposed a targeted attack model, in which the attackers can only control a subset of entries in $\mathbf{c}$, say $\mathbf{c}_{\mathcal{L}} := \{c_i, i \in \mathcal{L}\}$. Given $\mathbf{c}_{\mathcal{L}}$, the object of the attackers is to find a sparse $\mathbf{a}$ such that (1.5) holds, which can be formulated as a ℓ_1 or LASSO optimization problem.

In [9] a different model named strategic sparse attacks was proposed. Instead of an assumption on the controllability on the state domain $[n]$, this model divided

the measurement domain $[m]$ into two parts $\mathcal{A}$ and $\mathcal{S}$, where the attackers can only inject data in $\mathbf{a}_{\mathcal{A}}$, i.e., $\mathbf{a}_{\mathcal{S}} = 0$. The attackers then seek a vector $\mathbf{c}$ minimizing $\|\mathbf{a}_{\mathcal{A}}\|_0$ subject to $\mathbf{a}_{\mathcal{S}} = 0$ and $\|\mathbf{c}\|_\infty \geq \tau$, where $\tau > 0$ is some predefined constant.

In [10], these two models were summarized and their corresponding optimization problems were solved by using Alternating Direction Method of Multipliers (ADMM) algorithm [21]. Distributed optimization strategies for attackers and estimation algorithms for system protectors were also proposed in [10].

1.2.2 Video Anomaly Detection

In trajectory-based video monitoring systems, there are K pre-built dictionaries $\mathbf{D}_1, \mathbf{D}_2, \cdots, \mathbf{D}_K \in \mathbb{R}^{d\times n}$, where each column of $\mathbf{D}_k$ is a training sample labeled as the k-th type of trajectories [6, 7, 22]. For example, in a road traffic scenario, we can predefine K types of driving behaviors: legal/illegal straight, legal/illegal right turn, legal/illegal left turn, and legal/illegal U-turn, etc., and collect n sample trajectories for each category to build $\mathbf{D}_k$'s. A testing trajectory $\mathbf{y} \in \mathbb{R}^d$ is then approximated as a linear combination of columns in $\mathbf{D} := [\mathbf{D}_1, \mathbf{D}_2, \cdots, \mathbf{D}_K]$, that is,

$$\mathbf{y} \approx \mathbf{D}\mathbf{x}, \tag{1.6}$$

where $\mathbf{x}$ is called a representation of $\mathbf{y}$ on $\mathbf{D}$. Moreover, if we design $\mathbf{D}$ properly, we can have a sparse vector $\mathbf{x} \in \mathbb{R}^{Kn}$, which means that the dictionary is very comprehensive so that any normal $\mathbf{y}$ can be well "indexed" by very few items in it. Finding a sparse

$\hat{\mathbf{x}}$ satisfying (1.6) is a typical CS recovery problem. In [6], upon finding

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \\ \vdots \\ \hat{\mathbf{x}}_K \end{bmatrix}, \tag{1.7}$$

where $\hat{\mathbf{x}}_k = [\hat{\mathbf{x}}_{k,1}, \cdots, \hat{\mathbf{x}}_{k,n}]^T$ is the vector of coefficients corresponding to the k-th category, a residual r_k is calculated for each k:

$$r_k = \|\mathbf{y} - \mathbf{D}_k \hat{\mathbf{x}}_k\|_2, \tag{1.8}$$

and $\mathbf{y}$ is classified as the category corresponding to the minimal residual. If it falls into a category that is labeled abnormal, then we have a detection.

In [7], multiobject trajectory anomaly detection was proposed that exploits the interaction among multiple objects. For example, in a scenario where two moving vehicles are present, if both drivers drive carefully and follow the rules, then the trajectories of the two cars are all smooth and normal; however, if one car drives on a wrong way, then it may force an oncoming vehicle make a sudden change in direction, which leads both cars in an anomalous pattern. Mathematically, this can be formulated as a joint-sparsity structure, that is, for multiple trajectories $\mathbf{y}_1, \cdots, \mathbf{y}_P$ in one scenario, their sparse representations $\mathbf{x}_1, \cdots, \mathbf{x}_P$ under $\mathbf{D}$ shares a common support. Denote $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_P]$ and $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_P]$, the following $\ell_2 - \ell_0$ optimization problem was formulated to recover $\mathbf{X}$:

$$\begin{aligned} &\min \|\mathbf{X}\|_{2,0} \\ &\text{s.t.} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F \leq \epsilon, \end{aligned} \tag{1.9}$$

which is a common structured CS problem and has many existing solvers.

1.2.3 Manifold Learning

In the context of manifold-based compressed sensing (CS), a sparse signal means that it is in a low-dimensional manifold of a high-dimensional domain. In [3] and [14], the properties of manifold-based CS were analyzed theoretically. In [15], a mixture of factor analyzers (MFA) [23, 24] was applied to model manifolds with low intrinsic dimension. The authors modeled a manifold as a finite mixture of Gaussians with low-rank covariance matrices. The intuition of this model is straightforward geometrically, in the sense that the contour of distribution of a manifold can be well approximated a collection of flat ellipsoids — in statistics, a ellipsoids corresponds to a multivariate Gaussian distribution, and the flatness indicates that the covariance matrix is low-rank. In their paper, they drew n samples $\{\mathbf{x}_i\}_{i=1}^n$ from a manifold $\Omega \subset \mathbb{R}^N$ as the training data, and built a mixture low-rank Gaussian prior $p(\mathbf{x})$, where the number of mixtures and the rank of the Gaussian covariance matrices are inferred by non-parametric Bayesian approaches in [25] and [26]. Then, given a CS measurement of a vector $\mathbf{x} \in \Omega$, which is $\mathbf{y} = \mathbf{\Phi x} + \boldsymbol{\nu}$, where $\mathbf{\Phi}$ is the sensing matrix and $\boldsymbol{\nu}$ is a zero-mean Gaussian noise, the likelihood $p(\mathbf{y}|\mathbf{x})$ can be easily obtained in a closed form. With $p(\mathbf{x})$ and $p(\mathbf{y}|\mathbf{x})$ available, the posterior distribution $p(\mathbf{x}|\mathbf{y})$ can be inferred by the Bayes' rule.

1.3 Motivation of Distributed CS

In spite of its strength in dimensionality reduction, it is still demanding to perform CS on a single processor with limited memory and computational power when N and M are large. This urges the emergence of distributed CS (DCS), due to its nature of dividing and allocating large memory burden into a network of sensors, and its potential of speeding up the CS recovery process [27, 28, 29, 30, 31, 32].

In the literature, the term DCS may refer to two kinds of systems: one is linked to joint sparsity [33, 34, 35, 36], where P nodes take measurements of P correlated signals $\{\mathbf{s}_0^p\}_{p=1}^P$; in the other system, all the nodes in the network take measurements of a common $\mathbf{s}_0$ [37, 38, 39, 40, 41, 42], and the goal is to develop a distributed approach with the same accuracy as the centralized setting, that is, single-processor setting. For the former, each node has a sensing matrix $\mathbf{A}^p$ with enough rows such that the restricted isometry property (RIP) [16] is satisfied, and can perform signal recovery individually, although collaboration will further improve the recovery performance. On the other hand, the latter can be viewed as a distributed version of a centralized system, where the row-combination of all the P sensing matrices $\mathbf{A}^p$, is equivalent to a global sensing matrix $\mathbf{A}$ in centralized CS satisfying RIP. It is a typical assumption that each individual sensor has limited memory so that it cannot store the entire global sensing matrix.

In our work, we focus on the second DCS system, which contains two parts: (1) the local computation (LC) performed at each sensor, and (2) the global computation (GC) to obtain the estimate of the original sparse signal after sensors exchange the results of local computation. Among many efforts on the distributed computation, optimization, and network topology [37, 38] in DCS, only a few have been spent on addressing the communication issue in the GC step [40, 43, 41].

We are particularly interested in one category of CS recovery algorithms, where in every iteration t we have a sparse estimate $\mathbf{x}_t$ of $\mathbf{s}_0$, a residual or innovation $\mathbf{z}_t \in \mathbb{R}^M$ in the measurement domain, and a predetermined step size μ_t, based on which we can obtain a new estimate $\mathbf{x}_{t+1}$ of $\mathbf{s}_0$ as follows:

$$\mathbf{f}_t = \mathbf{x}_t + \mu_t \mathbf{A}^T \mathbf{z}_t, \tag{1.10}$$

$$\mathbf{x}_{t+1} = \eta_t(\mathbf{f}_t), \tag{1.11}$$

where $[\cdot]^T$ denotes transposition, and η_t is some predefined thresholder, for example, the hard thresholder in [44, 45], and the soft thresholder applied in [46, 47]. The reason we find these algorithms appealing is that we can develop DCS approaches based on them with a small communication cost and exactly the same recovery result as the centralized CS algorithms. As will be shown later, $\mathbf{f}_t$ in (1.10) can be expressed as the summation of P vectors $\mathbf{w}_t^p$, where each $\mathbf{w}_t^p$ can be calculated by the p-th sensor without communication. Due to the thresholding process in (1.11), it is not necessary to compute all the elements in $\mathbf{f}_t$ to obtain $\mathbf{x}_{t+1}$, which indicates that each sensor does not need to send the whole vector $\mathbf{w}_t^p$ to the fusion center in the GC step, hence yielding an amount of communication saving.

More specifically, in a DCS framework, the hard thresholding applied in [44, 45] can be modeled as an extension of Top-K problems [48, 49] in the field of distributed database querying, where there are P distributed agents indexed by $p \in \{1, \cdots, P\}$ and N objects indexed by $n \in \{1, \cdots, N\}$, with each object n having P partial scores $S^p(n)$'s distributed on the P sensors, and the object is to find the top K total scores $S(n) = \sum_{p=1}^{P} S^p(n)$ as well as the corresponding object indices with a minimum communication cost. On the other hand, the soft thresholding applied in [46, 47] can be modeled as a similar problem, which we name Top-β problem and is to find all the total scores with magnitudes greater than a threshold β as well as the corresponding object indices in pursuit of the highest communication savings.

Throughout the dissertation, we use bold capital letters to denote matrices and bold lower-case letters to denote vectors. $[\cdot]^T$ denotes matrix or vector transposition. $\mathbf{v}(k)$ denotes the k-th component of the vector $\mathbf{v}$, $\mathcal{T}_K(\mathbf{v})$ returns its K-th largest absolute element, $\mathcal{S}(\mathbf{v}) = \{k : \mathbf{v}(k) \neq 0\}$ denotes its support set, and $\|\mathbf{v}\|_0$ denotes $|\mathcal{S}(\mathbf{v})|$, where $|\cdot|$ on a set means its cardinality. $A \backslash B$ denotes the set difference between sets A and B. $[n]$ denotes the set $\{1, \cdots, n\}$. For $\Omega = \{n_1, \cdots, n_d\} \subset$

$[n]$, $\mathbf{v}(\Omega)$ denotes $[\mathbf{v}(n_1), \cdots, \mathbf{v}(n_d)]^T$; $\mathbf{W}(\Omega, :)$ and $\mathbf{W}(:, \Omega)$ denote the rows and columns of the matrix $\mathbf{W}$, specified by Ω respectively. The sign function is defined as $\text{sgn}(x) \stackrel{\Delta}{=} x/|x|$ if $x \neq 0$ and 0 otherwise; indicator function $\mathbb{I}(\texttt{STATEMENT}) = 1$ if the boolean `STATEMENT` is true and 0 otherwise, and $\mathbb{I}_S(x)$ means $\mathbb{I}(x \in S)$. $\mathbb{E}$ denotes the expectation of a random variable, $\mathcal{N}(\mu, \sigma^2)$ denotes the normal (Gaussian) distribution with mean μ and variance σ^2. For $Z \sim \mathcal{N}(0, 1)$, $\Phi(z) \stackrel{\Delta}{=} \Pr\{Z \leq z\}$ and $\phi(z) \stackrel{\Delta}{=} d\Phi(z)/dz$ denote its cumulative distribution function (CDF) and probability density function (PDF) respectively, $Q(z) \stackrel{\Delta}{=} 1 - \Phi(z)$ denotes the right tail probability $\Pr\{Z > z\}$ and $z_\alpha \stackrel{\Delta}{=} Q^{-1}(\alpha)$.

CHAPTER 2

DCS BASED ON APPROXIMATE MESSAGE PASSING

2.1 Introduction

Different from traditional CS recovery algorithms such as convex relaxation [50, 51], greedy pursuit [52, 53, 54], and iterative thresholding [46, 55, 44, 45], etc., approximate message passing (AMP) [47] is a statistical algorithm derived from the theory of probabilistic graphical models (PGM) [56]. The reason we choose AMP as a basic algorithm for developing DCS is that it is analytically convenient due to its state evolution (SE) formalism, as will be shown later in this chapter.

Instead of focusing on network topologies varying from applications, we are trying to propose some "universal" efficient data querying algorithms in the GC step of DiAMP. To this end, we assume a network of P sensors where Sensors 2 to P are connected to the fusion center Sensor 1, for better illustration of the proposed algorithms. We also assume that each individual sensor has limited memory, so that it cannot store the entire global sensing matrix, which has been assumed in [40, 43, 41]. In the LC step, each sensor only performs simple matrix operations, and in the GC step where communication cost is induced, we propose a algorithm named GC of DiAMP (GCAMP), to reduce the amount of data transmitted in the sensor network. Based on GCAMP, we take into consideration the correlation of data between adjacent iterations and incorporate quantization and propose a more sophisticated approach named quantized adaptive GCAMP (Q-A-GCAMP), which comes close to requiring the minimum bit rates.

In the rest of the chapter, we will give a brief review on AMP in Section 2.2, and

then present the framework of DiAMP and GCAMP algorithm in Section 2.3. An improved version of GCAMP, A-GCAMP is presented in Section 2.4, which is further improved by Q-A-GCAMP in Section 2.5. The proposed algorithms are evaluated in Section 2.6 and the dissertation is concluded in Section 2.7.

2.2 Overview of AMP

2.2.1 Centralized AMP

AMP starts from an initial estimate $\mathbf{x}_0 = 0$ and residual $\mathbf{z}_0 = \mathbf{y}$, and proceeds as follows:

$$\mathbf{f}_t = \mathbf{x}_t + \mathbf{A}^T \mathbf{z}_t, \tag{2.1}$$

$$\mathbf{x}_{t+1} = \eta(\mathbf{f}_t; \tau\sigma_t), \tag{2.2}$$

$$\mathbf{z}_{t+1} = \mathbf{y} - \mathbf{A}\mathbf{x}_{t+1} + \frac{||\mathbf{x}_{t+1}||_0}{M}\mathbf{z}_t, \tag{2.3}$$

where the square of σ_t, namely $\sigma_t^2 = \mathbb{E}\{\|\mathbf{f}_t - \mathbf{s}_0\|^2/N\}$ is the component-wise mean square error (MSE) of $\mathbf{f}_t$, and is often replaced by its estimate $\hat{\sigma}_t^2 = \|\mathbf{z}_t\|^2/M$ since $\mathbf{s}_0$ is unknown [57, 58], τ is a tunable parameter, and $\eta(\mathbf{x}; \beta)$ is a component-wise soft thresholding function and returns $\mathbf{u} \in \mathbb{R}^N$ with the i-th component computed by

$$\mathbf{u}(i) = \text{sgn}(\mathbf{x}(i))\mathbb{I}(|\mathbf{x}(i)| > \beta)(|\mathbf{x}(i)| - \beta). \tag{2.4}$$

Define $\kappa = M/N$ and $\rho = K/M$, and assume that each component of $\mathbf{s}_0$ comes from an i.i.d. source S_0 with unknown distribution. In the large system limit, that is, with $N \to \infty$ and κ, ρ keeping constant, each component in $\mathbf{f}_t - \mathbf{s}_0$ is i.i.d. Gaussian with mean 0 and variance σ_t^2, where σ_t^2 follows the State Evolution (SE) equation [47, 57, 58]:

$$\begin{aligned}\sigma_{t+1}^2 &= \sigma^2 + \frac{1}{\kappa}\mathbb{E}\frac{\|\mathbf{x}_t - \mathbf{s}_0\|_2^2}{N} \\ &= \sigma^2 + \frac{1}{\kappa}\mathbb{E}\left[\eta(S_0 + \sigma_t Z; \tau\sigma_t) - S_0\right]^2,\end{aligned} \tag{2.5}$$

where σ^2 is the variance of the noise $\mathbf{e}$ in (1.1) and $Z \sim \mathcal{N}(0,1)$.

Compared with RIP-based deterministic approaches, AMP has the following advantages:

1) The SE equation in AMP enables us to predict the recovery performance analytically in each iteration.

2) The soft thresholding denoiser in (2.4) is optimal in the minimax sense, i.e., it minimizes the recovery error in the worst-case scenario when the distribution of $\mathbf{s}_0$ is unknown. In the Bayesian framework, when the prior distribution of $\mathbf{s}_0$ is given, the denoiser can be replaced by the Minimum-Mean-Square-Error (MMSE) estimator, which is the conditional mean

$$\begin{aligned}\mathbf{x}_{t+1}(n) &= \eta(\mathbf{f}_t(n); \sigma_t) \\ &= \mathbb{E}\{\mathbf{s}_0(n) | \mathbf{s}_0(n) + \sigma_t Z = \mathbf{f}_t(n)\},\end{aligned} \tag{2.6}$$

and the SE still holds [58, 59, 60, 61].

2.2.2 Tuning of τ in Centralized AMP

We adopt the tuning framework in [62] to determine τ, which inherits the idea of continuation from the area of optimization [63]. First, a candidate list of candidate values of τ, $\{\tau_\ell\}_{\ell=1}^L$ is generated. Then, for each candidate τ_ℓ, we run iterations in (2.2) and (2.3) until $\mathbf{x}_t$ and σ_t converge to $\mathbf{x}_\ell^*$ and $\hat{\sigma}_\ell^*$, and use them as the initial estimates for the iterations using the next candidate $\tau_{\ell+1}$. Among all the L estimates $\mathbf{x}_\ell^*$, we choose the one that corresponds to the minimal $\hat{\sigma}_\ell^*$.

Since τ is directly related to the sparsity level of the estimate $\mathbf{x}_t$ — the larger τ is, the more sparse $\mathbf{x}_t$ becomes, and vice versa — we can determine the upper and

lower bounds on τ, namely $\tau_{\max}$ and $\tau_{\min}$, due to this property. Note that at the beginning, $\mathbf{x}_0 = 0$, $\mathbf{z}_0 = \mathbf{y}$, and $\mathbf{f}_0 = \mathbf{A}^T\mathbf{y}$, if $\tau \geq \|\mathbf{f}_0\|_\infty = \|\mathbf{A}^T\mathbf{y}\|_\infty/\hat{\sigma}_0$, then we will come to the zero solution. Therefore $\tau_{\max} = \|\mathbf{f}_0\|_\infty/\hat{\sigma}_0$. If we set the first candidate value $\tau_1 = \tau_{\max}$, then we will still stay at the zero solution after the first iteration. To prevent it from happening, we set $\tau_1 = \mathcal{T}_2(\mathbf{f}_0)/\hat{\sigma}_0$, which is guaranteed to yield a non-zero solution $\mathbf{x}_1$.

To determine $\tau_{\min}$, notice that for a given $\kappa = M/N$, the most dense $\mathbf{s}_0$ AMP can guarantee to recover is with the sparsity level $N\kappa\rho(\kappa)$, where $\rho(\kappa)$ is given by the phase-transition formalism [47, 57]:

$$\rho(\kappa) = \max_{z\geq 0}\left\{\frac{1 - 2/\kappa\left[(1+z^2)\Phi(-z) - z\phi(z)\right]}{1 + z^2 - 2\left[(1+z^2)\Phi(-z) - z\phi(z)\right]}\right\}, \tag{2.7}$$

and the value of τ in this case is given by

$$\tau(\kappa) = \arg\max_{z\geq 0}\left\{\frac{1 - 2/\kappa\left[(1+z^2)\Phi(-z) - z\phi(z)\right]}{1 + z^2 - 2\left[(1+z^2)\Phi(-z) - z\phi(z)\right]}\right\}. \tag{2.8}$$

Therefore, we can set $\tau_{\min} = \tau(\kappa)$, and set the last candidate value $\tau_L = \tau_{\min}$. Let $\Delta\tau = (\tau_1 - \tau_L)/(L-1)$, we can then set $\tau_\ell = \tau_1 - (\ell - 1)\Delta\tau$, $\forall \ell \in \{2, \cdots, L-1\}$.

Regarding when to terminate the algorithm, we use the convergence criterion $|\hat{\sigma}_{t+1} - \hat{\sigma}_t| \leq \zeta\hat{\sigma}_t$. Besides, we set a "budget" T_1 for the total number of iterations of DiAMP; for each candidate τ_ℓ, we set its maximum number of iterations as $\min\{T_2, (T_1 - \sum_i^{\ell-1} t_i)/(L - \ell + 1)\}$, where t_i is the number of iterations of running DiAMP with parameter τ_i. In this way, the total number of iterations would not exceed T_1 and the number of iterations corresponding to each τ_ℓ would not exceed T_2.

The pseudo code of AMP algorithm is shown in Table 1.

2.3 Distributed AMP Framework

Let us consider a sensor network with P distributed sensors. Each sensor p ($p = 1, \cdots, P$) takes M_p rows of $\mathbf{A}$, namely $\mathbf{A}^p$, and obtains $\mathbf{y}^p = \mathbf{A}^p\mathbf{s}_0 + \mathbf{e}^p$. Then

Table 1. Centralized AMP algorithm

Input $\mathbf{y}$, $\mathbf{A}$, $\tau_1, \cdots, \tau_L$, T_1, T_2

Initialization $\mathbf{x}_0 = 0, \mathbf{z}_0 = \mathbf{y}$
for $\ell = 1{:}L$
  initialize $t_\ell = 0$;
  while $t_\ell < \max\{T_2, (T_1 - \sum_{i=1}^{\ell-1} t_i)/(\ell - 1)\}$
    obtain an estimator $\widehat{\sigma}_{t_\ell}^2$ of $\sigma_{t_\ell}^2$;
    $\mathbf{x}_{t_\ell+1} = \eta(\mathbf{x}_{t_\ell} + \mathbf{A}^T\mathbf{z}_{t_\ell}; \tau_\ell\widehat{\sigma}_{t_\ell})$
    $\mathbf{z}_{t_\ell+1} = \mathbf{y} - \mathbf{A}\mathbf{x}_{t_\ell+1} + \frac{||\mathbf{x}_{t_\ell+1}||_0}{M}\mathbf{z}_{t_\ell}$
    if some convergence cretieron is met
      $\widehat{\sigma}(\tau_\ell) = \widehat{\sigma}_{t_\ell}, \mathbf{x}(\tau_\ell) = \mathbf{x}_{t_\ell+1}, \mathbf{z}(\tau_\ell) = \mathbf{z}_{t_\ell+1}$
      break
    else
      update $t_\ell \leftarrow t_\ell + 1$
      continue
    endif
  endwhile
  update $\mathbf{x}_0 = \mathbf{x}_{t_\ell+1}, \mathbf{z}_0 = \mathbf{z}_{t_\ell+1}$;
endfor

Output $\hat{\sigma}(\tau_\ell), \mathbf{x}(\tau_\ell)$, $\ell = 1, \cdots, L$.

(1.1) can be rewritten as:

$$\begin{bmatrix} \mathbf{y}^1 \\ \vdots \\ \mathbf{y}^P \end{bmatrix} = \begin{bmatrix} \mathbf{A}^1 \\ \vdots \\ \mathbf{A}^P \end{bmatrix} \mathbf{s}_0 + \begin{bmatrix} \mathbf{e}^1 \\ \vdots \\ \mathbf{e}^P \end{bmatrix}. \tag{2.9}$$

Let us introduce an intermediate matrix $\mathbf{W}_t = \left[\mathbf{w}_t^1, \ldots, \mathbf{w}_t^P\right]$ with each column computed by the corresponding sensor as [41]:

$$\mathbf{w}_t^p = \begin{cases} \mathbf{x}_t + (\mathbf{A}^p)^T\mathbf{z}_t^p, \text{ if } p = 1, \\ (\mathbf{A}^p)^T\mathbf{z}_t^p, \text{ otherwise.} \end{cases} \tag{2.10}$$

It is easy to show that AMP can be run in a distributed manner:

$$\mathbf{f}_t = \mathbf{x}_t + (\mathbf{A}^1)^T \mathbf{z}_t^p + \sum_{p=2}^{P} \left[(\mathbf{A}^p)^T \mathbf{z}_t^p \right] = \sum_{p=1}^{P} \mathbf{w}_t^p, \tag{2.11}$$

$$\mathbf{x}_{t+1} = \eta \left(\mathbf{f}_t; \tau \hat{\sigma}_t \right), \tag{2.12}$$

$$\mathbf{z}_{t+1}^p = \mathbf{y}^p - \mathbf{A}^p \mathbf{x}_{t+1} + \frac{||\mathbf{x}_{t+1}||_0}{M} \mathbf{z}_t^p, \ \forall p \in [P]. \tag{2.13}$$

In (2.10), if we modify the definition of $\mathbf{w}_t^p$ as

$$\mathbf{w}_t^p = \omega_p \mathbf{x}_t + (\mathbf{A}^p)^T \mathbf{z}_t^p, \tag{2.14}$$

where $\omega_p = M_p/M$ with M_p being the number of rows of $\mathbf{A}^p$, then it is easy to verify that (2.11) still holds. The new definition is adopted not only because it generates more balanced data, but also due to our recent proof on distributed AMP (DiAMP) that $\mathbf{w}_t^p - \omega_p \mathbf{s}_0$ is asymptotically $\mathcal{N}(0, \omega_p \sigma_t^2 \mathbf{I}_N)$ as $N \to \infty$, where $\mathbf{I}_N$ is the $N \times N$ identity matrix. The proof will be shown later in Chapter 3. Due to this property, we can obtain a new estimator $\hat{\sigma}_t^2$ of σ_t^2 as follows.

On Sensor p, we further partition $\mathbf{A}^p$ equally by rows and obtain $\mathbf{A}^{p,1}$, $\mathbf{A}^{p,2} \in \mathbf{R}^{M_p/2 \times N}$, and the corresponding $\mathbf{y}^{p,1}$, $\mathbf{y}^{p,2}$, $\mathbf{z}_t^{p,1}$, $\mathbf{z}_t^{p,2}$, etc. Denoting $\mathbf{w}_t^{p,i} = (\omega_p/2)\mathbf{x}_t + (\mathbf{A}^{p,i})^T \mathbf{z}_t^{p,i}$ $(i = 1, 2)$, then the $2P$ random vectors $\mathbf{r}_t^{p,i} = \mathbf{w}_t^{p,i} - (\omega_p/2)\mathbf{s}_0$ is asymptotically $\mathcal{N}(\mathbf{0}, (\omega_p/2)\sigma_t^2 \mathbf{I}_N)$. Defining $\mathbf{d}_t^p = (\mathbf{w}_t^{p,1} - \mathbf{w}_t^{p,2})/\sqrt{\omega_p}$, it is easy to show that

$$\mathbf{d}_t^p = \frac{\mathbf{w}_t^{p,1} - \mathbf{w}_t^{p,2}}{\sqrt{\omega_p}} = \frac{\mathbf{r}_t^{p,1} - \mathbf{r}_t^{p,2}}{\sqrt{\omega_p}} \sim \mathcal{N}(0, \sigma_t^2 \mathbf{I}_N). \tag{2.15}$$

We can obtain an estimator for σ_t^2:

$$\hat{\sigma}_t^2 = \frac{\sum_{p=1}^{P} \|\mathbf{d}_t^p\|_2^2}{NP}. \tag{2.16}$$

Note that $\mathbf{d}_t^p$ can be computed locally without communication. In order to obtain $\hat{\sigma}_t^2$, each sensor $p \geq 2$ only needs to send a positive number $\|\mathbf{d}_t^p\|_2^2$ to Sensor 1; Sensor

1 then computes $\hat{\sigma}_t^2$ and broadcasts it back to other sensors. This communication cost is negligible compared with that of obtaining $\mathbf{x}_{t+1}$, as will be shown later.

It can be seen that DiAMP can be divided into two parts: local computation (LC) of $\mathbf{w}_t^p$ and $\mathbf{z}_{t+1}^p (p = 1, \cdots, P)$, where there is no communication; and global computation (GC) of $\mathbf{x}_{t+1}$, where communication is needed. For the latter, a natural approach is to send all the components in $\mathbf{w}_t^p$ for $p \geq 2$ to sensor 1, which will induce a high communication cost when N is large. Therefore, how to reduce the communication cost, meanwhile without incurring any loss of accuracy compared with the original AMP, is the main goal of the distributed approach.

In DiAMP, by (2.10), $\mathbf{x}_{t+1}(n) = 0$ if $|\mathbf{f}_t(n)| \leq \beta_t = \tau\hat{\sigma}_t$. Therefore, we only need to know all $(n, \mathbf{f}_t(n))$ such that $|\mathbf{f}_t(n)| = |\sum_{p=1}^P \mathbf{w}_t^p(n)| > \beta_t$ in the GC. As introduced in Chapter 1, this is a Top-β problem. The n-th row of the intermediate matrix $\mathbf{W}_t$ can be viewed as an object with index n and partial scores $\mathbf{w}_t^1(n), \cdots, \mathbf{w}_t^P(n)$ stored on agents (sensors) $1, \cdots, P$ respectively, and the total score of object n is $\mathbf{f}_t(n) = \sum_{p=1}^P \mathbf{w}_t^p(n)$. Our objective is to find all the total scores with magnitudes greater than the threshold β_t as well as the objects they correspond to, meanwhile trying to reduce the communication cost induced. The Top-β problem has a very similar structure to the Top-K problem.

For the Top-K problems, one of the most popular algorithms is known as thresholding algorithm (TA) [48], which requires a known K and all the entries in $\mathbf{W}_t$ to be non-negative, and cannot be directly applied in DCS. In [40], a modified TA (MTA) was proposed to deal with the positivity issue. Notice that for DiAMP, TA can also be modified in a similar way to solve the induced Top-β algorithm, namely, MTA, which is shown in Table 2.

Theorem 1 *In each iteration, MTA gives exactly the same* $\mathbf{x}_{t+1}$ *as that of original*

Table 2. MTA Algorithm for DiAMP

Input $\mathbf{w}_t^1, \cdots, \mathbf{w}_t^P$, $\beta_t = \tau\hat{\sigma}_t$;

Initialization $\mathbf{x}_{t+1} = 0$, `count`$= 0$;
for sensor $p = 1{:}P$
  sort components of $\mathbf{w}_t^p$ in descending order of magnitudes;
  define the sorted vector as $\mathbf{s}_t^p$ and $\mathbf{I}_t^p(n) := \ell$ s.t. $\mathbf{w}_t^p(\ell) = \mathbf{s}_t^p(n)$;
  mark all $(\mathbf{I}_t^p(n), \mathbf{s}_t^p(n))$ pairs as "unsent";
endfor
while `TRUE`
  for $p = 1{:}P$
    find the first $(\mathbf{I}_t^p(n), \mathbf{s}_t^p(n))$ pair marked "unsent" from top;
    set $u_p = \mathbf{s}_t^p(n)$, broadcast $(\mathbf{I}_t^p(n), u_p)$ to other sensors;
    mark $(\mathbf{I}_t^p(n), \mathbf{s}_t^p(n))$ as "sent";
    for sensor $q \neq p$
      store u_p and send $(\mathbf{I}_t^p(n), \mathbf{w}_t^q(\mathbf{I}_t^p(n)))$ to sensor p;
      mark $(\mathbf{I}_t^p(n), \mathbf{w}_t^q(\mathbf{I}_t^p(n)))$ as "sent";
    endfor
    update $\mathbf{x}_{t+1}(\mathbf{I}_t^p(n)) = \eta_t(\sum_{p=1}^P \mathbf{w}_t^p(I_t^p(n)); \beta)$;
    `count`=`count`+1;
    if `count`$\geq P$ and $\sum_{p=1}^P |u_p| \leq \beta$, or if `count`$\geq N$
      set $N_s =$ `count`, the algorithm terminates;
    endif
  endfor
endwhile

Output $\mathbf{x}_{t+1}$

AMP algorithm computed by (2.2).

Proof of Theorem 1: As we can see, MTA is composed of a series of global summation, where a global summation means computing a total score $\mathbf{f}_t(n)$ for some n. N_s is a counter recording the number of global summations. At the very end of one global summation, for each n, either the $(n, \mathbf{w}_t^p(n))$ pairs for all p are marked as "sent"; or all are marked as "unsent". So we can just say n is marked as "sent" or "unsent". It is easy to show that, $\sum_{p=1}^P |u_p|$ is an upper bound on $|\mathbf{f}_t(n)|$ for all the n's that have not been marked as "sent"; if $\sum_{p=1}^P |u_p| \leq \beta_t$, then we have $|\mathbf{f}_t(n)| \leq \beta_t$ for these n's. Therefore, as the algorithm terminates, we do not lose any non-zero components of

$\mathbf{x}_{t+1}$. Q.E.D.

2.3.1 Proposed GC Algorithm in DiAMP: GCAMP

As discussed in the introduction of this chapter, the GC step of DiAMP can be modeled as a Top-β problem, which has a structure similar to the Top-K problem. There are many mature algorithms, e.g., TA [48], and the three-phase uniform threshold (TPUT) algorithm [49] . Similar to TA, the original TPUT cannot be applied in DiAMP, not only because it solves a Top-K problem instead of a Top-β problem, but also due to the fact that it requires all partial scores to be non-negative, which does not hold in our problem settings. The point is that some essence of TPUT, which is to get an upper bound on the total score, really helps us derive our own Top-β algorithm. Before we proceed, we first introduce the following Lemma:

Lemma 1 *Given arbitary $T > 0$, define*

$$S_n = \{p \in [P] \setminus \{1\} : |\mathbf{w}_t^p(n)| > T\} \tag{2.17}$$

for $n \in [N]$, then

$$U(n) = |\mathbf{w}_t^1(n) + \sum_{p \in S_n} \mathbf{w}_t^p(n)| + (P - 1 - |S_n|)T \tag{2.18}$$

is an upper bound on $|\mathbf{f}_t(n)|$.

Proof of Lemma 1: For any $n = 1, \cdots, N$, we have

$$\mathbf{f}_t(n) = \mathbf{w}_t^1(n) + \sum_{p \in S_n} \mathbf{w}_t^p(n) + \sum_{p \geq 2, p \notin S_n} \mathbf{w}_t^p(n) \tag{2.19}$$

Then, applying the triangular inequality, we have

$$\begin{aligned} |\mathbf{f}_t(n)| &\leq \left| \mathbf{w}_t^1(n) + \sum_{p \in S_n} \mathbf{w}_t^p(n) \right| + \left| \sum_{p \geq 2, p \notin S_n} \mathbf{w}_t^p(n) \right| \\ &\leq \left| \mathbf{w}_t^1(n) + \sum_{p \in S_n} \mathbf{w}_t^p(n) \right| + (P - 1 - |S_n|)T = U(n) \end{aligned} \tag{2.20}$$

According to Lemma 1, we develop the following GCAMP algorithm.

First, sensors 2 to P only send the partial scores with magnitudes greater than a predefined threshold T, as well as the corresponding indices of objects to sensor 1. Then sensors 1 computes an upper bound $U(n)$ for each n according to (2.18) and obtains the set of n's with $U(n) > \beta_t$. Finally, sensor 1 requests all the partial scores for all objects $n \in F$ from other sensors, computes total scores $\mathbf{f}_t(n)$ for all objects $n \in F$, and obtains the new estimate $\mathbf{x}_{t+1}$, where $\mathbf{x}_{t+1}(F) = \eta(\mathbf{f}_t(F); \beta_t)$ and $\mathbf{x}_{t+1}([N] \setminus F) = 0$.

Theorem 2 *In each iteration, GCAMP gives exactly the same* $\mathbf{x}_{t+1}$ *as that of the centralized AMP algorithm computed by* (2.2)*.*

Proof of Theorem 2: Let $\mathbf{x}_{t+1}^G$ and $\mathbf{x}_{t+1}^A$ denote the result obtained by the GCAMP and the Centralized AMP respectively. For any $n \in F$, we have $\mathbf{x}_{t+1}^G(n) = \eta(\mathbf{f}_t(n); \beta_t) = \mathbf{x}_{t+1}^A(n)$; for any $n \notin F$, we have $\mathbf{x}_{t+1}^G(n) = 0$ and $U(n) \leq \beta_t$, while according to Lemma 1, we know $|\mathbf{f}_t(n)| \leq U(n) \leq \beta_t$, so $\mathbf{x}_{t+1}^A(n) = 0$. Therefore, $\mathbf{x}_{t+1}^G = \mathbf{x}_{t+1}^A$.

A remaining problem is how to choose a proper T. Theoretically, we can use arbitrary $T > 0$. However, if T is too large, then we may get a very loose upper bound $U(n)$ for n. Moreover, considering the case that all entries in $\mathbf{W}_t$ have the same sign but we are not aware of it in advance, and we use $T \geq \frac{\beta_t}{P-1}$, then it is easy to show that we will have $U(n) > \beta_t$ for all $n \in [N]$, that is, we need to send all partial scores from other sensors to sensor 1 and cannot save any communication. To

prevent such case from happening, we need to constrain $T < \frac{\beta_t}{P-1}$. Therefore, we can choose some $\theta \in (0, 1)$, and set $T = \theta\frac{\beta_t}{P-1}$.

The pseudo code of GCAMP algorithm is shown in Table 3. It can be shown that the total numbers of messages is $\sum_{p=1}^{P} |\Omega_p \bigcup F| + |F|$, where the first part is the number of data other sensors send to Sensor 1, and the second part is the number of broadcasting messages Sensor 1 sends to others. For MTA, in each global summation, there are 1 broadcasting message from some sensor to others and $P-1$ incoming messages, so the total number of messages is PN_s.

In every inner loop of DiAMP, after GCAMP and MTA return $\mathbf{x}_{t+1}$, it takes $\|\mathbf{x}_{t+1}\|_0$ messages for all the sensors to know the non-zero components in $\mathbf{x}_{t+1}$. Once knowing $\mathbf{x}_{t+1}$, each local sensor can obtain $\mathbf{z}_{t+1}^p$ using (2.13) and $\sigma_{t+1}^p = ||\mathbf{z}_{t+1}^p||_2$ ($p = 1, \cdots, P$). Next, each sensor $p \geq 2$ just sends a scalar σ_{t+1}^p to Sensor 1, which needs $P-1$ messages. Then, sensor 1 computes $\hat{\sigma}_{t+1} = \sqrt{\sum_{p=1}^{P}(\sigma_{t+1}^p)^2/M}$, updates β_t and T, and broadcasts the scalar T to other sensors, so the total number of messages in DiAMP is that of GCAMP (MTA) plus $\|\mathbf{x}_{t+1}\|_0 + P$.

In Fig. 1, an example is provided to illustrate how GCAMP works, in which each sensor p already sorts $\mathbf{w}_t^p(n)$ in descending order of magnitudes, and stores the data in the form of $(n,\ \mathbf{w}_t^p(n))$ pairs ($p = 1, \cdots, 3, n = 1, \cdots, 10$). Suppose $\beta_t = 20$ and $\theta = 0.8$, since we have $P = 3$ sensors, we get $T = \beta_t\theta/(P-1) = 8$. In Step I, Sensors 2 to P send all the $(n,\ \mathbf{w}_t^p(n))$ pairs with $|\mathbf{w}_t^p(n)| > T$ (red boxes in the figure) to Sensor 1 (red arrows). In Step II, Sensor 1 receives the data and computes upper bounds $U(n)$ for $n = 1, \cdots, 10$. As we can see, only $U(4)$, $U(6)$ and $U(7) > \beta_t$, which means that we are sure that $\mathbf{x}_{t+1}(n) = 0$ for all $n \notin F = \{4, 6, 7\}$. So Sensor 1 only needs to broadcast requests of partial scores of objects $n \in V$. In Step III, Sensor 2 sends $\mathbf{w}_t^2(4)$ and $\mathbf{w}_t^2(7)$, and Sensor 3 sends $\mathbf{w}_t^3(4)$ and $\mathbf{w}_t^3(6)$ to Sensor 1. Finally, in Step IV, Sensor 1 computes $\mathbf{x}_{t+1}(n)$ for $n \in F$ by (2.11), and outputs the non-zero

Table 3. GCAMP algorithm

Input $\mathbf{w}_t^1, \cdots, \mathbf{w}_t^P$, $\beta_t = \tau\sigma_t$, θ

Step I Set $T = \beta_t\theta/(P-1)$
for sensor $p = 2{:}P$
  denote $\Omega_p = \{n : |\mathbf{w}_t^p(n)| > T\}$;
  send all $(n, \mathbf{w}_t^p(n))$ pairs for $n \in \Omega_p$ to sensor 1.
endfor
Step II for sensor 1,
for $n = 1{:}N$
  get S_n as defined in (2.17);
  Compute the upper bound $U(n)$ for $|\mathbf{f}_t(n)|$ according to (2.18)
  if $U(n) > \beta_t$
    broadcast the index n to other sensors
  endif
endfor
Step III denote $F = \{n : U(n) > \beta_t\}$
for sensor $p = 2{:}P$
  send all $(n, \mathbf{w}_t^p(n))$ pairs for $n \in F \backslash R_p$ to sensor 1.
endfor
Step IV for sensor 1, denote $\Gamma = \{n \in F : |\mathbf{f}_t(n)| > \beta_t\}$
assign $\mathbf{x}_{t+1}(\Gamma) = \eta(\mathbf{f}_t(\Gamma); \beta_t)$ and $\mathbf{x}_{t+1}([N] \backslash \Gamma) = 0$;

Output $\mathbf{x}_{t+1}$

---

components of $\mathbf{x}_{t+1}$, which is $\mathbf{x}_{t+1}(6) = 3$ and $\mathbf{x}_{t+1}(6) = -1$. It is easy to verify that the gap Δ_{t+1} obtained for this example is 1. Overall, in this example, only 9 data points are sent from the sensors to sensor 1, and the total number of messages is 12 (9 data points plus 3 broadcast requests). For the data set in Fig. 1, MTA needs 9 global summations to get the final results, which leads to 27 messages, much greater than of that of GCAMP.

Recall that the centralized AMP starts with $\mathbf{x}_0 = 0$, $\mathbf{z}_0 = \mathbf{y}$, and $\tau = \mathcal{T}_2(\mathbf{f}_0)/\hat{\sigma}_0$, which requires the second largest magnitude in the vector $\mathbf{f}_0$. This requires us to solve an extended Top-2 problem to obtain $\mathcal{T}_2(\mathbf{f}_0)$ at the beginning of DiAMP.

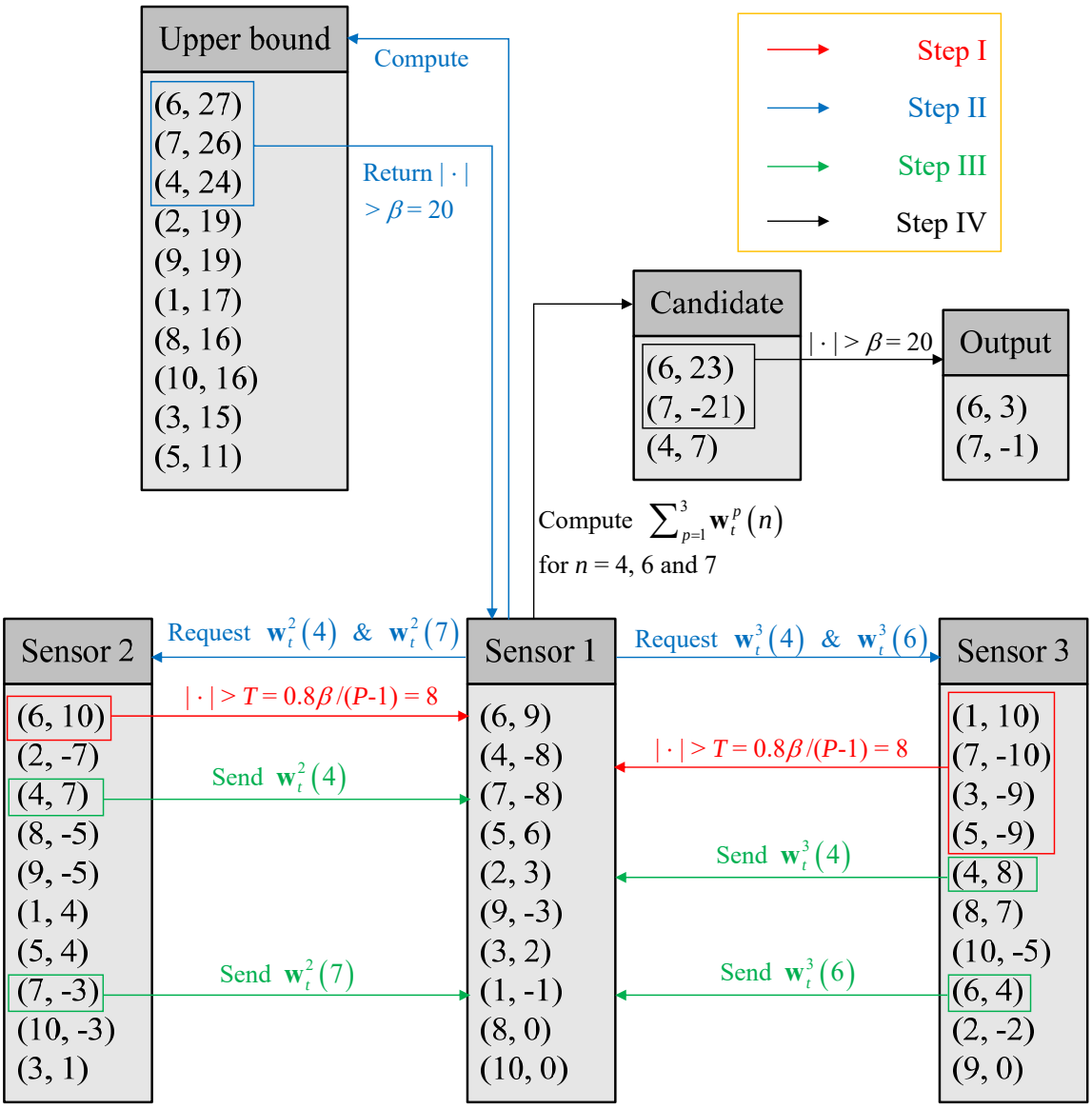

Fig. 1. GCAMP algorithm

2.3.2 The optimal value of θ in GCAMP

It can be shown in Table 3 that the number of communication messages needed in GCAMP is $N_G = \sum_{p=2}^{P} |F \cup \Omega_p| + |F| + \|\mathbf{x}_{t+1}\|_0$. If we model the elements in $\mathbf{s}_0$ and measurement noise $\mathbf{e}$ as i.i.d. scalar random variables $S_0 \sim p_{S_0}$ and $E \sim p_E$

respectively, then N_G is also a random variable, where its expectation

$$\mathbb{E}\left[N_G\right] = \sum_{p=2}^{P} \mathbb{E}\left[|F \cup \Omega_p|\right] + \mathbb{E}\left[|F|\right] + \mathbb{E}\left[\|\mathbf{x}_{t+1}\|_0\right] \tag{2.21}$$

is a function of the parameter θ. More specifically, only the items $\mathbb{E}|F \cup \Omega_p|$ and $\mathbb{E}|F|$ depend on θ. So we can model the following optimization problem to determine the value of θ:

$$\min_{\theta} \sum_{p=2}^{P} \mathbb{E}\left[|F \cup \Omega_p|\right] + \mathbb{E}\left[|F|\right]. \tag{2.22}$$

Recalling the Gaussianity in DiAMP as described in Section 2.3, we know that each element $\mathbf{w}_t^p(n)$ follows i.i.d. $\omega_p S_0 + \sigma_t\sqrt{\omega_p}Z_p$ with $Z_1, \cdots, Z_P \sim$ i.i.d. $\mathcal{N}(0,1)$. We can derive

$$\begin{aligned}\mathbb{E}\left[|F|\right] = N\Pr\{U(n) > \beta_t\} = N\Pr\Bigg\{\Bigg|\omega_p S_0 + \sigma_t\sqrt{\omega_1}Z_1 + \\ \sum_{p=2}^{P}\left(\omega_p S_0 + \sigma_t\sqrt{\omega_p}Z_p\right)\mathbb{I}\left(\left|\omega_p S_0 + \sigma_t\sqrt{\omega_p}Z_p\right| > \frac{\theta\beta_t}{P-1}\right)\Bigg| \\ + T\sum_{p=2}^{P}\mathbb{I}\left(\left|\omega_p S_0 + \sigma_t\sqrt{\omega_p}Z_p\right| \leq \frac{\theta\beta_t}{P-1}\right) > \beta_t\Bigg\},\end{aligned} \tag{2.23}$$

and

$$\begin{aligned}\mathbb{E}\left[|F \cup \Omega_p|\right] = N\Bigg[\Pr\left\{U(n) > \beta_t\right\} + \Pr\Bigg\{\Bigg|\omega_p S_0 + \sigma_t\sqrt{\omega_p}Z_p\Bigg| > \\ \frac{\theta\beta_t}{P-1}\Bigg\} - \Pr\left\{U(n) > \beta_t, \left|\omega_p S_0 + \sigma_t\sqrt{\omega_p}Z_p\right| > \frac{\theta\beta_t}{P-1}\right\}\Bigg].\end{aligned} \tag{2.24}$$

For the most common cases where the prior distribution p_{S_0} is unknown, we can perform worst-case analysis to obtain a mini-max solution:

$$\min_{\theta} \sup_{p_{S_0}} \sum_{p=2}^{P} \mathbb{E}\left[|F \cup \Omega_p|\right] + \mathbb{E}\left[|F|\right]. \tag{2.25}$$

Unfortunately, there is no closed-form solution for either (2.22) or (2.25). While it may be possible to get some approximated solutions given particular assumptions, we will tune θ empirically in this dissertation.

2.4 Improvement on GCAMP: Adaptive Approach

In GCAMP, the intermediate $\mathbf{W}_t$ in each iteration t is considered as totally new data, i.e., correlation between $\mathbf{W}_t$ and $\mathbf{W}_{t-1}$ is not considered in the algorithm. However, due to the convergence of AMP, $\mathbf{W}_t$ and $\mathbf{W}_{t-1}$ will become closer and closer through iterations. If an adaptive approach based on GCAMP can be developed by taking advantage of this property, then further improvements in terms of communication savings may be achievable without loss of recovery accuracy.

From Table 3, it can be shown that GCAMP has the following outcomes:

i) Total scores for all objects $n \in \Gamma$, where Γ is the support of $\mathbf{x}_{t+1}$. They have one-to-one mapping to non-zero entries in $\mathbf{x}_{t+1}$.

ii) Total scores for all objects $n \in F \backslash \Gamma$, where Γ is the support of $\mathbf{x}_{t+1}$. These objects have $U(n) > \beta_t$ but $|\mathbf{f}_t(n)| \leq \beta_t$. The total scores for all objects in i) and ii), that is, in F, are saved in the column vector x^h_{t+1} as shown in Table 3.

iii) A gap Δ_{t+1} between $\{|\mathbf{f}_t(n)| \colon n \notin \Gamma\}$ and the threshold β_t. For $n \notin \Gamma$, if $n \in F$, which means that GCAMP obtains $\mathbf{f}_t(n)$ for n, then

$$\beta_t - |\mathbf{f}_t(n)| \geq \beta_t - \max_{n \in F} |\mathbf{f}_t(n)|; \tag{2.26}$$

if $n \notin F$, which means $U(n) \leq \beta_t$, then

$$\beta_t - |\mathbf{f}_t(n)| \geq \beta_t - U(n) \geq \beta_t - \max_{n \notin F} U(n). \tag{2.27}$$

Overall, we have

$$\beta_t - |\mathbf{f}_t(n)| \geq \Delta_{t+1} = \beta_t - \max\{\max_{n\in F\setminus\Gamma} |\mathbf{f}_t(n)|, \max_{n\notin F} U(n)\} \tag{2.28}$$

for all $n \notin \Gamma$.

For GCAMP itself, i) is the only outcome of interest, and ii) and iii) seem to be meaningless byproducts. However, if an adaptive approach is developed as discussed above, where GCAMP is used as a subroutine, then outcomes in ii) and iii) will become important as shown later in the dissertation.

2.4.1 Question: Can We Save More?

In the previous sections we discussed the framework of DiAMP consisting of a series of local and global computations, where the global computation (GC) step consumes communication bandwidths. To reduce the communication cost in the GC step, we proposed GCAMP, which sifts a candidate set for the support of $\mathbf{x}_{t+1}$ and then sends the partial scores of all the objects within the candidate set to the fusion center (Sensor 1).

While the aforementioned framework is communication efficient, it still induces more communication cost than what is necessary. This is because GCAMP is a non-adaptive approach, i.e., in each iteration t, it obtains $\mathbf{x}_{t+1}$ only based on the current intermediate data $\mathbf{W}_t$, with no memory about $\mathbf{W}_{t-1}$ nor $\mathbf{x}_t$; on the other hand, AMP has a linear convergence rate such that after a few iterations, $\mathbf{W}_{t-1}$ and $\mathbf{W}_t$ will become very close, so it is the case for $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$. This indicates that it is possible to adaptively learn some useful knowledge from previous data $\mathbf{W}_{t-1}$ and $\mathbf{x}_t$ when computing $\mathbf{x}_{t+1}$, in return for a further reduction in communication cost.

To see this, we first define $\Delta\mathbf{w}_t^p = \mathbf{w}_t^p - \mathbf{w}_{t-1}^p$, $p \in [P]$, $\Delta\mathbf{W}_t = \left[\Delta\mathbf{w}_t^1, \cdots, \Delta\mathbf{w}_t^P\right]$, and rewrite $\mathbf{f}_t(n)$ in the following way:

$$\mathbf{f}_t(n) = \sum_{p=1}^{P} \left(\mathbf{w}_{t-1}^p(n) + \Delta\mathbf{w}_t^p(n)\right) = \mathbf{f}_{t-1}(n) + \Delta\mathbf{f}_t(n), \tag{2.29}$$

where we use $\Delta\mathbf{f}_t(n)$ to denote $\sum_{p=1}^{P} \Delta\mathbf{w}_t^p(n)$.

By applying the triangular inequality, we can easily establish an upper bound on $|\mathbf{f}_t(n)|$:

$$|\mathbf{f}_t(n)| \leq |\mathbf{f}_{t-1}(n)| + |\Delta\mathbf{f}_t(n)|. \tag{2.30}$$

Remember that after iteration $t-1$, we have obtained the support set $\mathcal{S}(\mathbf{x}_t)$, and for any $n \notin \mathcal{S}(\mathbf{x}_t)$, we are certain that $|\mathbf{f}_{t-1}(n)| \leq \beta_{t-1}$. Suppose in iteration t, we have a threshold $\beta_t > \beta_{t-1}$, then for any $n \notin \mathcal{S}(\mathbf{x}_t)$ satisfying $|\Delta\mathbf{f}_t(n)| \leq \beta_t - \beta_{t-1}$, we have

$$|\mathbf{f}_t(n)| \leq \beta_{t-1} + (\beta_t - \beta_{t-1}) = \beta_t, \tag{2.31}$$

which means that n will not be in the support set of $\mathbf{x}_{t+1}$ either. In other words, only two groups of n's can be in $\mathcal{S}(\mathbf{x}_{t+1})$: (i) n's$\in \mathcal{S}(\mathbf{x}_t)$, (ii) n's$\notin \mathcal{S}(\mathbf{x}_t)$ satisfying $|\Delta\mathbf{f}_t(n)| > \beta_t - \beta_{t-1}$. Therefore, we can use the following adaptive procedure named adaptive GCAMP (A-GCAMP) to obtain $\mathcal{S}(\mathbf{x}_{t+1})$.

For group i, we can run GCAMP on $\mathbf{W}_t(\mathcal{S}(\mathbf{x}_t), :)$ to find all the n's with $|\mathbf{f}_t(n)| > \beta_t$, which does not consume much communication since $\mathbf{x}_t$ is sparse, i.e., $|\mathcal{S}(\mathbf{x}_t)| \ll N$.

For group ii, we first need to run GCAMP on $\Delta\mathbf{W}_t([N] \setminus \mathcal{S}(\mathbf{x}_t), :)$ to find all the n's with $|\Delta\mathbf{f}_t(n)| > \beta_t - \beta_{t-1}$. Let V_{t+1} be the set of these n's, we further run GCAMP on $\mathbf{W}_t(V_{t+1}, :)$ to find all the n's with $|\mathbf{f}_t(n)| > \beta_t$. Note that if most entries' magnitudes in $\Delta\mathbf{W}_t([N] \setminus \mathcal{S}(\mathbf{x}_t), :)$ are much smaller than $\beta_t - \beta_{t-1}$, then by the mechanism of GCAMP, the induced communication cost will be negligible. But how likely is this assumption to be valid?

First, let us consider the magnitudes in $\mathbf{W}_t$. According to [47, 64, 65], AMP has a linear convergence rate, which implies that after a few iterations, $\mathbf{W}_{t-1}$ and $\mathbf{W}_t$ will be very close, i.e., most entries in $\Delta\mathbf{W}_t$ will be close to 0.

Second, we need to evaluate $\beta_t - \beta_{t-1}$. Note that the procedure mentioned above only makes sense when $\beta_t > \beta_{t-1}$. However, this seems over-optimistic since β_t

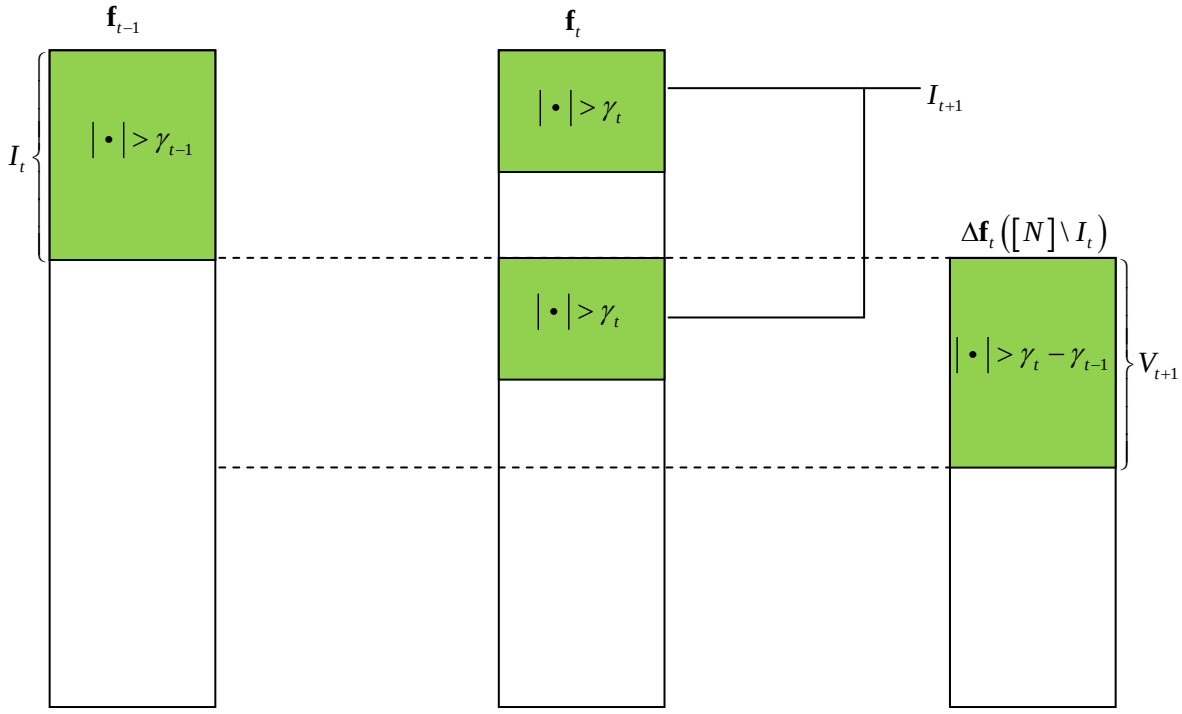

Fig. 2. A diagram of A-GCAMP.

proportional to $\hat{\sigma}_t$ is generally decreasing. To circumvent this conundrum, we propose a *First-Lose-Then-Win* strategy in the following subsection.

2.4.2 First-Lose-Then-Win Strategy

The basic idea of this strategy is to design a sequence $\{\alpha_t\} \subset (0, 1]$ and transform the threshold to $\gamma_t = \alpha_t \beta_t$, so that $\gamma_t > \gamma_{t-1}$ and A-GCAMP introduced in Section 2.4.1 can be applied, as shown in Fig. 2, where I_{t+1} denotes the set of objects with total scores $|\mathbf{f}_t(n)| > \gamma_t$.

Note that after we find I_{t+1}, we still plug the original threshold β_t into the soft thresholding function to obtain $\mathbf{x}_{t+1}$. Therefore, to maintain accuracy we need to constrain $\alpha_t \leq 1$ so that any $|\mathbf{f}_t(n)| > \beta_t$ will satisfy $|\mathbf{f}_t(n)| > \gamma_t$. However, lowering the threshold may cause additional communication costs, which is the "First-Lose"

stage; on the other hand, we can have $\gamma_t > \gamma_{t-1}$ hold, which may lead to a significant reduction in communication cost as discussed in Section 2.4.1, and this is the "Then-Win" stage. The key is to properly design $\{\alpha_t\}$ so that what we win is more than what we lose, which will be described in the next subsection.

We provide an overview of A-GCAMP based on the strategy below:

(i) Input: $\mathbf{W}_{t-1}$, $\mathbf{W}_t$, β_t.

(ii) Intermediate: α_t, γ_t, $I_{t+1} = \{n : |\mathbf{f}_t(n)| > \gamma_t\}$.

(iii) Output: $\mathbf{x}_{t+1}$.

Theorem 3 *A-GCAMP obtains* $\mathbf{x}_{t+1}$ *which is exactly the same as that obtained by the centralized AMP using* (2.1) *and* (2.2).

Proof of Theorem 3: We know that all the n's in $I_{t+1} = \{n\colon |\mathbf{f}_t(n)| > \gamma_t\}$ will either satisfy $n \in I_t$, or $n \notin I_t$ but $|\Delta\mathbf{f}_t(n)| > \Delta\gamma_t$. The latter group is found by GCAMP. Denoting $V_{t+1} = \{n\colon n \notin I_t, |\Delta\mathbf{f}_t(n)| > \Delta\gamma_t\}$, GCAMP will lose no accuracy when only searching in $I_t \cup V_{t+1}$ for n's such that $|\mathbf{f}_t(n)| > \gamma_t$. Since $\gamma_t \leq \beta_t$ and $\mathbf{f}_t(n)$'s for all the n's in I_{t+1} are calculated, we know that A-GCAMP will return the same $\mathbf{x}_{t+1}$ as that of the centralized AMP. Q.E.D.

2.4.3 Design of α_t

The aim of designing α_t in A-GCAMP is to yield a threshold difference $\Delta\gamma_t = \gamma_t - \gamma_{t-1}$ which is much greater than most magnitudes in $\Delta\mathbf{f}_t([N]\setminus I_t)$, so that the communication cost of running GCAMP on $\Delta\mathbf{W}_t([N]\setminus I_t, :)$ can be negligible.

In Fig. 3 the flow chart of determining γ_t is shown, where q is a small number and is set to 0.05 in this dissertation, and θ is the parameter in GCAMP which is invoked to find all the n's not in I_t such that $|\Delta\mathbf{f}_t(n)| > \Delta\gamma_t$. As shown in the figure, we first obtain the P quantiles, take the maximal, namely $\max_p \mathcal{T}_{\lfloor qN_t^r \rfloor}(\Delta\mathbf{w}_t^p([N]\setminus I_t))$,

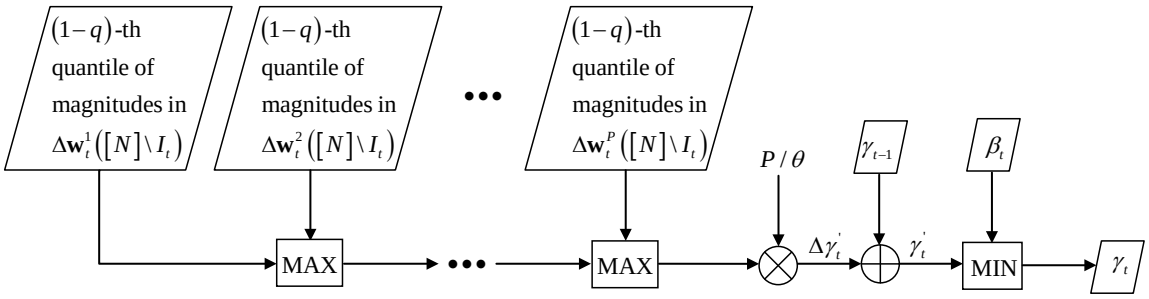

Fig. 3. Flow chart of determining γ_t.

where $N_t^r = N - |I_t|$, and obtain a candidate value $\Delta\gamma_t'$ for $\Delta\gamma_t$:

$$\Delta\gamma_t' = \frac{P \max_p \mathcal{T}_{\lfloor qN_t^r \rfloor}(\Delta\mathbf{w}_t^p([N]\setminus I_t))}{\theta}, \tag{2.32}$$

based on which we can obtain a candidate value

$$\gamma_t' = \gamma_{t-1} + \Delta\gamma_t'. \tag{2.33}$$

Since γ_t is upper bounded by β_t, we set

$$\gamma_t = \min\{\gamma_t', \beta_t\} \text{ and } \alpha_t = \min\left\{1, \frac{\gamma_t'}{\beta_t}\right\}. \tag{2.34}$$

Since β_t is generally decreasing, the sequence $\{\alpha_t\}$ is typically increasing. Once $\alpha_{t-1} = 1$, we will reset $\alpha_t = \alpha^*$, where α^* is a predefined parameter less than 1 and set to 0.94 in this dissertation, and repeat the above process to generate α_{t+1}.

Define

$$\Delta\Omega_t^p = |\{n \notin I_t \colon |\Delta\mathbf{w}_t^p(n)|\}| \tag{2.35}$$

for any $p \in [P]$, and

$$\Delta\Omega_t = \left|\{n \notin I_t : |\Delta\mathbf{f}_t(n)| > \theta\Delta\gamma_t\}\right|. \tag{2.36}$$

It is easy to verify that if $\Delta\gamma_t = \Delta\gamma_t'$, then

$$|\Delta\Omega_t^p| \le \left|\{n \notin I_t \colon |\Delta\mathbf{w}_t^p(n)| > \mathcal{T}_{\lfloor qN_t^r \rfloor}(\Delta\mathbf{w}_t^p([N]\setminus I_t))\}\right| \le qN_t^r \tag{2.37}$$

for any $p \in [P]$, and

$$
\begin{aligned}
|\Delta\Omega_t| \leq & \left| \left\{ n \notin I_t : \max_p |\Delta \mathbf{w}_t^p(n)| > \frac{\theta \Delta\gamma_t}{P} \right\} \right| \\
= & \left| \bigcup_{p=1}^{P} \left\{ n \notin I_t : |\Delta \mathbf{w}_t^p(n)| > \max_p \mathcal{T}_{\lfloor qN_t^r \rfloor}(\Delta \mathbf{w}_t^p([N] \setminus I_t)) \right\} \right| \\
\leq & \sum_{p=1}^{P} \left| \left\{ n \notin I_t : |\Delta \mathbf{w}_t^p(n)| > \max_p \mathcal{T}_{\lfloor qN_t^r \rfloor}(\Delta \mathbf{w}_t^p([N] \setminus I_t)) \right\} \right| \\
\leq & \sum_{p=1}^{P} \left| \left\{ n \notin I_t : |\Delta \mathbf{w}_t^p(n)| > \mathcal{T}_{\lfloor qN_t^r \rfloor}(\Delta \mathbf{w}_t^p([N] \setminus I_t)) \right\} \right| \leq qN_t^r P,
\end{aligned}
\tag{2.38}
$$

where the notation $|\{\cdot\}|$ on a set $\{\cdot\}$ denotes its cardinality, and $|\cdot|$ on a number, for example, $\mathbf{w}_t^p(n)$, denotes its magnitude.

As we can see, (2.37) sets an upper bound on number of the data points Sensor p sends to Sensor 1 at Stage I of GCAMP, and (2.38) obtains an upper bound on number of n's with $|\Delta \mathbf{f}_t(n)| > \Delta\gamma_t$. The equality in (2.38) holds because the statement "the maximal of the P numbers is greater than some value" is equivalent to "at least one of the P numbers is greater than the same value".

It is clear that if we choose the starting point $\alpha_{\min}$ properly, then we can have $\alpha_{t-1} < 1$ and $\alpha_t < 1$, which makes $\Delta\gamma_t = \frac{P \max_p T_{\lfloor qN_t^r \rfloor}(|\Delta \mathbf{w}_t^p|)}{\theta}$ hold in most iterations. In computing $\mathbf{x}_t$, we find all the n's such that $|\mathbf{f}_{t-1}(n)| > \gamma_{t-1}$, which includes the n that $\gamma_{t-1} < |\mathbf{f}_{t-1}(n)| \leq \beta_{t-1}$, this seems a waste of computation since we are sure $\mathbf{x}_t(n) = 0$ for these n's without knowing their total scores, that is why we call it the "first-lose" stage; however, what we win are thresholds $\{\gamma_t\}$ and gaps Δ_t which satisfy $\gamma_t + \Delta_t - \gamma_{t-1} = \Delta\gamma_t > 0$, which implies reduction of communication cost in computing $\mathbf{x}_{t+1}$.

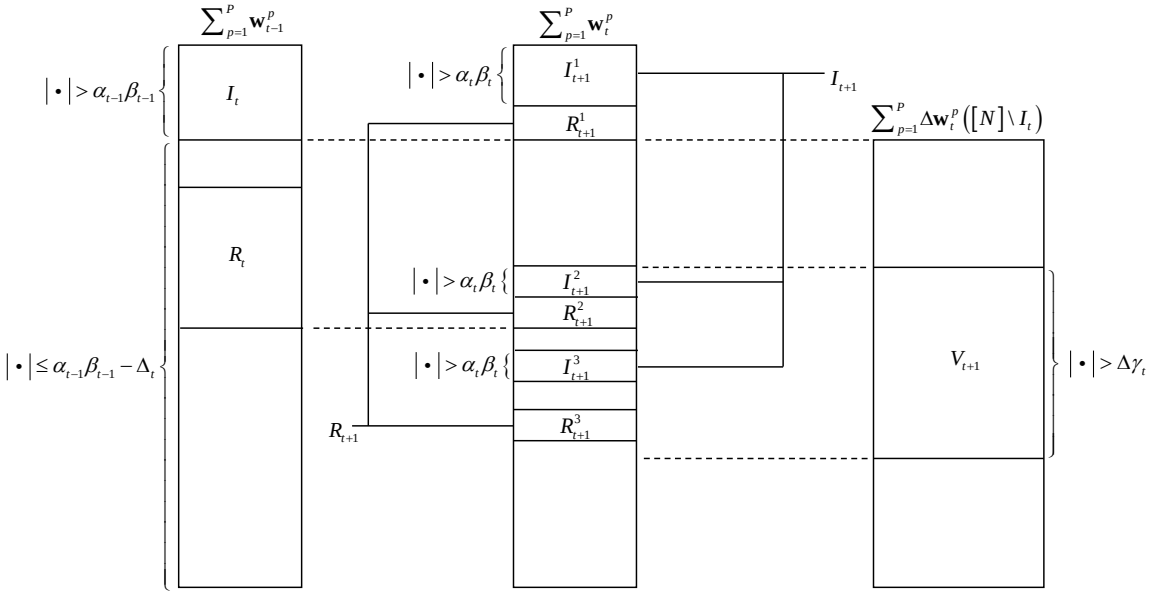

Fig. 4. A detailed diagram of A-GCAMP algorithm

2.4.4 DiAMP Based on A-GCAMP

In Fig. 4 a detailed diagram of DiAMP is shown based on A-GCAMP. In each iteration t, we will get $I_{t+1} = \{n : |\sum_{p=1}^{P} \mathbf{w}_t^p(n)| > \gamma_t\}$, and will also get a group of total scores for $n \in R_{t+1} \subset [N] \setminus I_{t+1}$; furthermore, we will get a gap Δ_{t+1} between $|\mathbf{f}_t(n)|$ for $n \notin I_{t+1}$ and γ_t.

In the first iteration $t = 0$, we get these by running the original GCAMP algorithm. Starting from $t = 1$, we first need to check whether $\Delta\Gamma_t = \max(\gamma_t - \gamma_{t-1} + \Delta_t, 0)$ is a large positive number compared with the magnitudes of partial scores for $n \in [N] \setminus I_t$, by checking whether

$$\sum_{p=1}^{P} |\Delta\Omega_t^p| \le \rho_w N_t^r P \tag{2.39}$$

holds, where $\Delta\Omega_t^p$ is the same as defined in (2.35) and ρ_w is within $(0, 1)$. This process just needs P communication messages. If (2.39) does not hold, then we still run the

original GCAMP algorithm on $\mathbf{W}_t$ to obtain I_{t+1}, R_{t+1} as well as the total scores $\mathbf{f}_t(I_{t+1})$, $\mathbf{f}_t(R_{t+1})$, and a gap Δ_{t+1}. Now we mainly focus on the case that (2.39) holds. If so, then we proceed as follows:

We first calculate total scores for all $n \in I_t = I_{t+1}^1 \cup R_{t+1}^1$, where $|\mathbf{f}_t(n)| > \gamma_t, \forall n \in I_{t+1}^1$ and $|\mathbf{f}_t(n)| \leq \gamma_t, \forall n \in R_{t+1}^1$; for $n \notin I_t$, we run GCAMP on $\Delta\mathbf{W}_t$ to get $V_{t+1} = \{n\colon |\sum_{p=1}^{P} \Delta\mathbf{w}_t^p(n)| > \Delta\gamma_t$, and a gap Δ_{t+1}^1 between $\{|\Delta w_t^p(n)|\colon n \notin I_t \bigcup V_{t+1}\}$ and $\Delta\Gamma_t$, where $\Delta\Gamma_t = \max(\gamma_t - \gamma_{t-1} + \Delta_t, 0)$.

For each $n \in V_{t+1} \bigcap R_t$, we know the total scores $\mathbf{f}_{t-1}(n)$ since $n \in R_t$ and $\sum_{p=1}^{P} \Delta\mathbf{w}_t^p(n)$ since $n \in V_{t+1}$, so we can easily calculate total scores $\mathbf{f}_t(n)$. Now partition the set $V_{t+1} \bigcap R_t$ into I_{t+1}^2 and R_{t+1}^2, where $|\mathbf{f}_t(n)| > \gamma_t, \forall n \in I_{t+1}^2$ and $|\mathbf{f}_t(n)| \leq \gamma_t, \forall n \in R_{t+1}^2$. So far, we have obtained the total scores for $n \in I_{t+1}^1 \bigcup I_{t+1}^2 \bigcup R_{t+1}^1 \bigcup R_{t+1}^2$. We then calculate

$$\Delta_{t+1}^2 = \gamma_t - \max |\mathbf{f}_t(R_{t+1}^1 \bigcup R_{t+1}^2)| \tag{2.40}$$

For $n \in V_{t+1} \backslash R_t$, we run GCAMP on $\mathbf{w}_t^1(V_{t+1} \backslash R_t)$, $\cdots$, $\mathbf{w}_t^P(V_{t+1} \backslash R_t)$ to get a group of total scores for $n \in I_{t+1}^3 \bigcup R_{t+1}^3$, where $|\mathbf{f}_t(n)| > \gamma_t, \forall n \in I_{t+1}^3$ and $|\mathbf{f}_t(n)| \leq \gamma_t, \forall n \in R_{t+1}^3$, and a gap Δ_{t+1}^3 between $\{|\mathbf{f}_t(n)| : n \in V_{t+1} \backslash R_t, |\mathbf{f}_t(n)| \leq \gamma_t\}$ and γ_t. Then we have the following outcomes for GCAMP:

i) $I_{t+1} = I_{t+1}^1 \cup I_{t+1}^2 \cup I_{t+1}^3$ and total scores $f_t(I_{t+1})$, $x_{t+1}(I_{t+1}) = \eta^S(f_t(I_{t+1}); \beta_t)$ and $\mathbf{x}_{t+1}([N] \setminus I_{t+1}) = 0$;

ii) $R_{t+1} = R_{t+1}^1 \cup R_{t+1}^2 \cup R_{t+1}^3$ and total scores $\mathbf{f}_t(R_{t+1})$;

iii) $\Delta_{t+1} = \min(\Delta_{t+1}^1, \Delta_{t+1}^2, \Delta_{t+1}^3)$.

The total scores $\mathbf{f}_t(I_{t+1} \bigcup R_{t+1})$ are saved in vector $\mathbf{x}_{t+1}^h$.

In order to show the correctness of A-GCAMP, we first need to prove the following lemma.

Lemma 2 *Δ_{t+1} calculated in A-GCAMP is a gap between $\{|\mathbf{f}_t(n)| : n \notin I_{t+1}, |\mathbf{f}_t(n)| \leq$*

$\gamma_t\}$ *and* γ_t.

Proof of Lemma 2: It is clear that

$$\begin{aligned}[N]\setminus I_{t+1} = [[N]\setminus(\ I_t \cup V_{t+1})] \cup \\ (R^1_{t+1} \cup R^2_{t+1}) \cup \left[V_{t+1}\setminus(I^3_{t+1} \cup R_t)\right]\end{aligned} \tag{2.41}$$

Obviously, Δ^2_{t+1} is a gap between $\{|\mathbf{f}_t(n)| : n \in (R^1_{t+1} \bigcup R^2_{t+1})\}$ and γ_t, and Δ^3_{t+1} is a gap between $\{|\mathbf{f}_t(n)| : n \in V_{t+1}\setminus(I^3_{t+1} \bigcup R_t)\}$ and γ_t. Now we need to show that Δ^1_{t+1} is a gap between $\{|\mathbf{f}_t(n)| : n \in [N]\setminus(\ I_t \bigcup V_{t+1})\}$ and γ_t.

By the definition of Δ^1_{t+1} in GCAMP algorithm, we know that

$$|\sum_{p=1}^{P} \Delta\mathbf{w}^p_t(n)| \leq \Delta\gamma_t - \Delta^1_{t+1}, \forall n \in [N]\setminus(\ I_t \bigcup V_{t+1}) \tag{2.42}$$

Therefore, we have

$$\begin{aligned}|\mathbf{f}_t(n)| &= |\mathbf{f}_{t-1}(n) + \sum_{p=1}^{P} \Delta\mathbf{w}^p_t(n)| \\ &\leq |\mathbf{f}_{t-1}(n)| + |\sum_{p=1}^{P} \Delta\mathbf{w}^p_t(n)| \\ &\leq (\beta_{t-1} - \Delta_t) + (\Delta\gamma_t - \Delta^1_{t+1}) \\ &= \beta_t - \Delta^1_{t+1}, \forall n \in [N]\setminus(\ I_t \bigcup V_{t+1})\end{aligned} \tag{2.43}$$

In summary, $\Delta_{t+1} = \min(\Delta^1_{t+1}, \Delta^2_{t+1}, \Delta^3_{t+1})$ is a gap between $\{|\mathbf{f}_t(n)| : n \notin I_{t+1}, |\mathbf{f}_t(n)| \leq \gamma_t\}$ and γ_t. Q.E.D.

Now we can prove the correctness of A-GCAMP.

Theorem 4 *A-GCAMP algorithm obtains* $\mathbf{x}_{t+1}$ *which is exactly the same as that of the centralized AMP algorithm computed by* (2.2).

Proof of Theorem 4: Applying Lemma 2, we know that Δ_t calculated in the A-GCAMP algorithm at iteration $t-1$ is a gap between $\{|\mathbf{f}_{t-1}(n)| : n \notin I_t, |\mathbf{f}_{t-1}(n)| \leq$

$\gamma_{t-1}\}$ and γ_{t-1}. Therefore, $I_{t+1} \subset I_t \cup V_{t+1} = I_t \cup (V_{t+1} \cap R_t) \cup (V_{t+1} \backslash R_t)$.

According to the algorithm, for all $n \in I_t \cup (V_{t+1} \cap R_t)$, we get the total scores $\mathbf{f}_t(n)$, and find $I_{t+1}^1 = \{n : |\mathbf{f}_t(n)| > \gamma_t, n \in I_t\}$ and $I_{t+1}^2 = \{n : |\mathbf{f}_t(n)| > \gamma_t, n \in V_{t+1} \cap R_t\}$. For all $n \in V_{t+1} \backslash R_t$, $I_{t+1}^3 = n : |\mathbf{f}_t(n)| > \gamma_t, n \in V_{t+1} \backslash R_t$ and $\mathbf{f}_t(I_{t+1}^3)$ are found by running the GCAMP algorithm. Therefore, $\mathbf{f}_t(I_{t+1})$ with $I_{t+1} = I_{t+1}^1 \cup I_{t+1}^2 \cup I_{t+1}^3$ contains all the total scores with magnitude greater than γ_t. In other words, $|\mathbf{f}_t(n)| \leq \gamma_t \leq \beta_t, \forall n \notin I_{t+1}$. Similar to the Proof of Theorem 2, we know that the $\mathbf{x}_{t+1}$ GCAMP algorithm obtains is exactly the same as that of the centralized AMP algorithm obtains. Q.E.D.

The pseudo code of A-GCAMP algorithm for iterations $t \geq 2$ is given in Table 4

.

Table 4.: A-GCAMP Algorithm

Input $\mathbf{w}_t^p$, $\Delta\mathbf{w}_t^p$, $\mathbf{x}_t^h$, γ_{t-1}, α_t, β_t, Δ_t, θ, ρ_w

---

$I_t = \{n : |\mathbf{x}_t^h(n)| > \gamma_{t-1}\}$;
$R_t = \{n : 0 < |\mathbf{x}_t^h(n)| \leq \gamma_{t-1}\}$;
$C_t^1 = [N] \backslash I_t$;
$\gamma_t = \alpha_t \beta_t$;
Compute $\Delta\gamma_t$;
if $\Delta\gamma_t > 0$
  for sensor $p = 1 : P$
    obtain $\Delta\Omega_p$;
    if $p \geq 2$

send the cardinality $|\Delta\Omega_p|$ to sensor 1;

endif

endfor

$N^g = \sum_{p=1}^{P} |\Delta\Omega_p|$;

else

$N^g = |C_t^1|P$;

endif

if $N^g \leq \rho_w |C_t^1| P$

initialize $\mathbf{x}_{t+1}^h = 0, \Delta\mathbf{x}_{t+1}^h = 0$;

$\mathbf{x}_{t+1}^h(I_t) = \sum_{p=1}^{P} \mathbf{w}_t^p(I_t)$;

$I_{t+1}^1 = \{n : n \in I_t, |\mathbf{x}_{t+1}^h(n)| > \gamma_t\}$;

$R_{t+1}^1 = I_t \backslash I_{t+1}^1$;

$\left[\sim, \Delta\mathbf{x}_{t+1}^h(C_t^1), \Delta_{t+1}^1\right] = \text{GCAMP}(\Delta\mathbf{w}_t^1(C_t^1), \cdots, \Delta\mathbf{w}_t^P(C_t^1), \Delta\gamma_t, \theta)$;

$V_{t+1} = \{n : |\Delta\mathbf{x}_{t+1}^h(n)| > \Delta\gamma_t\}$;

$C_t^2 = V_{t+1} \cap R_t$;

$\mathbf{x}_{t+1}^h(C_t^2) = \mathbf{x}_t^h(C_t^2) + \Delta\mathbf{x}_{t+1}^h(C_t^2)$;

$I_{t+1}^2 = \{n : n \in C_t^2, |\mathbf{x}_{t+1}^h(n)| > \gamma_t\}$;

$R_{t+1}^2 = C_t^2 \backslash I_{t+1}^2$;

$\Delta_{t+1}^2 = \gamma_t - \max(|\mathbf{x}_{t+1}^h(R_{t+1}^1 \cup R_{t+1}^2)|)$;

$C_t^3 = V_{t+1} \backslash R_t$;

$\left[\sim, \mathbf{x}_{t+1}^h(C_t^3), \Delta_{t+1}^3\right] = \text{GCAMP}\left(\mathbf{w}_t^1(C_t^3), \cdots, \mathbf{w}_t^P(C_t^3), \gamma_t, \theta\right)$;

$I_{t+1}^3 = \{n : n \in C_t^3, |\mathbf{x}_{t+1}^h(n)| > \gamma_t\}$;

$R_{t+1}^3 = \{n : n \in C_t^3, 0 < |\mathbf{x}_{t+1}^h(n)| \leq \gamma_t\}$;

$I_{t+1} = I_{t+1}^1 \cup I_{t+1}^2 \cup I_{t+1}^3$;

$R_{t+1} = R_{t+1}^1 \cup R_{t+1}^2 \cup R_{t+1}^3$;

$\Delta_{t+1} = \min(\Delta_{t+1}^1, \Delta_{t+1}^2, \Delta_{t+1}^3)$;

else

$\left[\sim, \mathbf{x}_{t+1}^h, \Delta_{t+1}\right] = \text{GCAMP}\left(\mathbf{w}_t^1, \cdots, \mathbf{w}_t^P, \gamma_t, \theta\right)$;

endif

$\mathbf{x}_{t+1} = \eta(\mathbf{x}_{t+1}^h; \beta_t)$;

Output $\mathbf{x}_{t+1}, \mathbf{x}_{t+1}^h, \Delta_{t+1}$

With A-GCAMP, the adaptive DiAMP approach is shown in Table 5,

Table 5.: Adaptive DiAMP Algorithm

Input $\{\mathbf{y}^p\}_{p=1}^P$, $\{\mathbf{A}^p\}_{p=1}^P$, $\{\tau_\ell\}_{\ell=1}^L$, $\alpha_{\min}$, q, θ, ρ_w, T_1, T_2

Initialize $\mathbf{x}_0 = 0$, $\mathbf{z}_0 = \mathbf{y}$, $\alpha_0 = \alpha_{\min}$, $\tau = \tau_1$;

for $\ell = 1{:}L$

initialize $t_\ell = 0$;

while $t_\ell < \max\{T_2, (T_1 - \sum_{i=1}^{\ell-1} t_i)/(\ell - 1)\}$

set $t = \sum_{i=1}^{\ell-1} t_i + t_\ell$;

if $t \geq 1$

Compute w_t^p and $\Delta w_t^p = w_t^p - w_{t-1}^p$ by (3.251) for each p;

if $\alpha_{t-1} = 1$

$\alpha_t = \alpha_{\min}$;

else

Compute α_t according to (2.34);

endif

obtain an estimator $\hat{\sigma}_t^2$ of σ_t^2, and set $\beta_t = \tau\hat{\sigma}_t$ and $\gamma_t = \alpha_t\beta_t$;

if $t \geq 1$

$\left[\mathbf{x}_{t+1}, \mathbf{x}_{t+1}^h, \Delta_{t+1}\right] =$

A-GCAMP$(\mathbf{w}_t^1, \cdots, \mathbf{w}_t^P, \Delta\mathbf{w}_t^1, \cdots, \Delta\mathbf{w}_t^P, \mathbf{x}_t^h, \gamma_{t-1}, \alpha_t, \beta_t, \Delta_t, \theta, \rho_w)$;

else

$\left[\sim, \mathbf{x}_{t+1}^h, \Delta_{t+1}\right] =$ GCAMP$(\mathbf{w}_t^1, \cdots, \mathbf{w}_t^P, \gamma_t)$;

$\mathbf{x}_{t+1} = \eta(\mathbf{x}_{t+1}^h; \beta_t)$;

endif

Compute $\mathbf{z}_{t+1}^p$ by (2.13) for each p;

if some convergence cretieron is met

set $\hat{\sigma}(\tau_\ell) = \sigma_{t+1}, \mathbf{x}(\tau_\ell) = x_{t+1}, \mathbf{z}^p(\tau_\ell) = \mathbf{z}_{t+1}^p$ for $p = 1 \cdots P$

update $\tau = \tau_{\ell+1}$;

break;

else

$t_\ell \leftarrow t_\ell + 1$;

continue;

endif

endwhile

endfor

Output $\hat{\sigma}(\tau_\ell)$, $\mathbf{x}(\tau_\ell)$, $\mathbf{z}^p(\tau_\ell)$

2.5 Improvement on GCAMP: Quantization

2.5.1 Intuition: A Sign-Aware Approach

In GCAMP and A-GCAMP, we aimed to reduce the number of communication messages, also known as units of communication [49] needed in the GC step. Seemingly a reasonable metric of communication cost to some extent, it does not account for the difference in structures of various types of data, for example, while broadcasting an object index and sending a $(n, \mathbf{w}_t^p(n))$ pair consumes different number of bits, they are both viewed as 1 message. Furthermore, evaluating the communication cost in terms of the number of communication messages does not show the advantage of quantization, a powerful technique in data compression, since one data point, regardless of being quantized or not, needs one message to be transmitted.

From this perspective, we update the communication cost metric to the number of communication bits. The new metric is not only more practical, but also motivates us to develop more sophisticated algorithms taking account of the structure of transmitted data. When developing GCAMP in Section 2.3, we restricted the parameter θ to be strictly less than 1, to prevent sending all the partial scores under the extreme cases where all of them have the same sign but no sensor is aware of that in advance. This restriction sometime may limit the availability of a finer upper bound $U(n)$ on $|\mathbf{f}_t(n)|$. However, with the new communication cost metric in terms of the communication bits, we can get rid of this restriction by adding a "sign-awareness" step at the beginning of GCAMP, where Sensors $p \geq 2$ send the signs of all the partial scores to

Sensor 1, with a communication bit of only 1 bit per element. After this step, Sensor 1 is aware of all the partial scores' signs and hence has more flexibility in choosing θ, which helps obtaining a tighter upper bound $U(n)$ and reducing the communication cost. This is the essence of the sign-aware GC (SAGC) algorithm.

We assume that all the components in $\mathbf{w}_t^p$ are stored as IEEE double-precision floating-point numbers, with a 1-bit sign, an 11-bit exponent, and a 52-bit significand [66], which will be discussed in detail in Section 2.5.3. Now, we present the SAGC algorithm as follows:

Step I: Define $T = \theta\beta_t/(P-1)$ and $\Omega_p = \{n : |\mathbf{w}_t^p(n)| > T\}$, where $\theta > 0$ is a tunable parameter for the trade-off between the communication cost in this step and that in Step III. Each sensor $p \geq 2$ sends to Sensor 1 the package shown in Fig. 5.

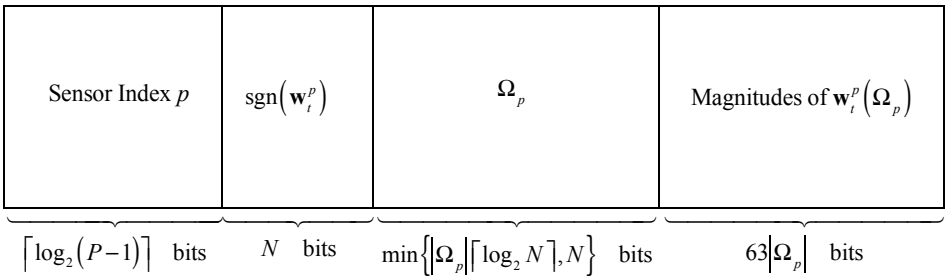

Fig. 5. The structure of the package Sensor p sends to Sensor 1 in Step I

As shown in Fig. 5, the package contains four parts:

1) the sensor index p, since there are $P-1$ sensors other than Sensor 1, we can use $\lceil \log_2(P-1) \rceil$ bits to encode p.

2) the 1-bit signs ("+" or "-") of all the components in $\mathbf{w}_t^p$, which totally require N bits to represent.

3) To represent Ω_p, we can either directly encode each index ($\lceil \log_2 N \rceil$ bits per index) in Ω_p with $|\Omega_p| \lceil \log_2 N \rceil$ bits in total, or use a 1/0 flag to denote whether each index $n = 1, \cdots, N$ is in Ω_p, which requires totally N bits. We finally choose the one

using fewer bits. For the case where $|\Omega_p|\lceil \log_2 N \rceil = N$, the latter will be chosen.

4) all the absolute values of $\mathbf{w}_t^p(\Omega_p)$, since they are all positive numbers, the total number of bits required is $63|\Omega_p|$.

Therefore, the total number of bits in Step I is

$$\begin{aligned} B_1 &= (P-1)\left[\lceil \log_2(P-1) \rceil + N\right] \\ &+ \sum_{p=2}^{P} \min\{|\Omega_p|\lceil \log_2 N \rceil, N\} + 63 \sum_{p=2}^{P} |\Omega_p|. \end{aligned} \tag{2.44}$$

Proposition 1 *Sensor* 1 *can decode the package from Sensor p without ambiguity.*

Proof of Proposition 1: It is easy to show that there is no ambiguity decoding Parts 1) and 2): the first $\lceil \log_2(P-1) \rceil$ bits of the package indicate which sensor the package is from, and the next N bits reveal the signs of $\mathbf{w}_t^p$. Now we need to show that there is no ambiguity decoding Ω_p and $\mathbf{w}_t^p(\Omega_p)$, which takes $L = \min\{|\Omega_p|\lceil \log_2 N \rceil, N\} + 63|\Omega_p|$ bits in total.

The possible ambiguity, if any, happens only if there exists two different values x and y for $|\Omega_p|$ leading to different encoding patterns of Ω_p yet ending up with the same L. Without loss of generality, let us assume that $x\lceil \log_2 N \rceil < N$, $y\lceil \log_2 N \rceil \geq N$, and $x\lceil \log_2 N \rceil + 63x = N + 63y = L$. We can easily show that this will not happen: since $x\lceil \log_2 N \rceil < N$ and $y\lceil \log_2 N \rceil \geq N$, we have $x < y$; on the other hand, because $x\lceil \log_2 N \rceil < N$ and $x\lceil \log_2 N \rceil + 63x = N + 63y$, we have $x > y$, which is a contradiction. Therefore, the solution $|\Omega_p|$ to the equation $L = \min\{|\Omega_p|\lceil \log_2 N \rceil, N\} + 63|\Omega_p|$ is unique, that is, there is no ambiguity in decoding Parts 3) and 4) of the package.

Step II: Sensor 1 decodes the packages from other sensors as discussed above, and obtains a range $\left[R^L(n), R^U(n)\right]$ of $\mathbf{f}_t(n)$:

$$R^L(n) = \mathbf{w}_t^1(n) + \sum_{p=2}^{P} \mathbf{w}_t^p(n)\mathbb{I}(n \in \Omega_p) - T\sum_{p=2}^{P} \mathbb{I}(-T \le \mathbf{w}_t^p(n) < 0) \tag{2.45}$$

and

$$R^U(n) = \mathbf{w}_t^1(n) + \sum_{p=2}^{P} \mathbf{w}_t^p(n)\mathbb{I}(n \in \Omega_p) + T\sum_{p=2}^{P} \mathbb{I}(0 < \mathbf{w}_t^p(n) \le T). \tag{2.46}$$

It can be shown that

$$U(n) = \max\left\{R^U(n), -R^L(n)\right\} \tag{2.47}$$

is an upper bound on $|\mathbf{f}_t(n)|$. Sensor 1 obtains the set $\Pi = \{n\colon U(n) \ge \beta_t\}$, represents Π in the same way as encoding Ω_p, and broadcasts it to all the other sensors, resulting in the number of bits used in Step II as

$$B_2 = \min\{|\Pi|\lceil \log_2 N\rceil, N\}. \tag{2.48}$$

Step III: Each sensor $p \ge 2$ reads all the indices in Π, and sends the package shown in Fig. 6 to Sensor 1.

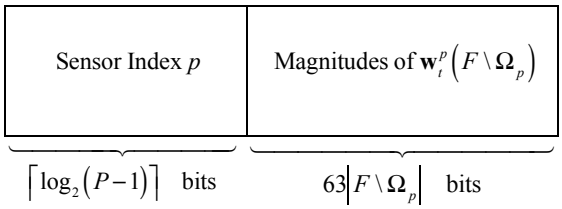

Fig. 6. The structure of the package Sensor p sends to Sensor 1 in Step III.

Note that both Sensors 1 and p know Π and Ω_p, and there is no need to send the object indices in the set $\Pi\setminus\Omega_p$.

Now Sensor 1 can compute $\mathbf{f}_t(\Pi)$ and obtain

$$\mathbf{x}_{t+1}(n) = \begin{cases} \eta(\mathbf{f}_t(n); \beta_t) & \text{if } n \in \Pi, \\ 0 & \text{otherwise.} \end{cases} \tag{2.49}$$

Finally, Sensor 1 needs to broadcast the support set of $\mathbf{x}_{t+1}$, which is $\Gamma = \{n \colon \mathbf{x}_{t+1}(n) \neq 0\}$, and the non-zero components $\mathbf{x}_{t+1}(\Gamma)$ to other sensors. Since $\Gamma \subset \Pi$, and each Sensor $p \geq 2$ already knows Π, Sensor 1 only needs $\min\{|\Gamma|\lceil \log_2 |\Pi| \rceil, |\Pi|\}$ bits to represent Γ. Therefore, the number of bits in Step III is

$$\begin{aligned} B_3 = {} & (P-1)\lceil \log_2(P-1) \rceil + 63 \sum_{p=2}^{P} |\Pi \backslash \Omega_p| \\ & + \min\{|\Gamma|\lceil \log_2 |\Pi| \rceil, |\Pi|\} + 64|\Gamma|, \end{aligned} \tag{2.50}$$

and the total number of bits in SAGC is $\sum_{i=1}^{3} B_i$.

As we can see, θ is a parameter controlling the trade-off between the number of communication bits in Steps I and III: increasing θ leads to a larger T, which means less data sent to Sensor 1 in Step I; however, this can cause a more conservative upper bound $U(n)$, which increases the number of bits in Step III, and vice versa. In this dissertation, we tune θ empirically.

Regarding the recovery accuracy of SAGC, we have the following theorem:

Theorem 5 *Given the same threshold $\beta_t = \tau\hat{\sigma}_t$, SAGC algorithm obtains exactly the same $\mathbf{x}_{t+1}$ as that of the centralized AMP algorithm computed by using* (2.2).

In other words, DiAMP based on SAGC has the same accuracy and convergence rate as the centralized AMP.

In order to prove Theorem 2, we first introduce the following lemma:

Lemma 3 *$U(n)$ is an upper bound on $|\mathbf{f}_t(n)|$.*

Proof of Lemma 3: First, we show that $R^L(n) \leq \mathbf{f}_t(n) \leq R^U(n)$ for any $n = 1, \cdots, N$:

$$
\begin{aligned}
\mathbf{f}_t(n) &= \mathbf{w}_t^1(n) + \sum_{p=2}^{P} \mathbf{w}_t^p(n)\mathbb{I}(n \in \Omega_p) \\
&+ \sum_{p=2}^{P} \mathbf{w}_t^p(n)\mathbb{I}(-T \le \mathbf{w}_t^p(n) < 0) \\
&+ \sum_{p=2}^{P} \mathbf{w}_t^p(n)\mathbb{I}(0 \le \mathbf{w}_t^p(n) \le T) \\
&\ge \mathbf{w}_t^1(n) + \sum_{p=2}^{P} \mathbf{w}_t^p(n)\mathbb{I}(n \in \Omega_p) \\
&+ \sum_{p=2}^{P} (-T) \times \mathbb{I}(-T \le \mathbf{w}_t^p(n) < 0) + 0 = R^L(n).
\end{aligned}
\tag{2.51}
$$

Similarly, we can prove that $\mathbf{f}_t(n) \le R^U(n)$. Since $R^L(n) \le \mathbf{f}_t(n) \le R^U(n)$, we have $-R^U(n) \le -\mathbf{f}_t(n) \le -R^L(n)$ and hence

$$
\begin{aligned}
|\mathbf{f}_t(n)| &= \max\{\mathbf{f}_t(n), -\mathbf{f}_t(n)\} \\
&\le \max\{R^U(n), -R^L(n)\} = U(n).
\end{aligned}
\tag{2.52}
$$

Now we can prove Theorem 2 as follows.

Proof of Theorem 2: Let $\mathbf{x}_{t+1}^G$ and $\mathbf{x}_{t+1}^A$ denote the results obtained by the SAGC and the centralized AMP respectively. For any $n \in \Pi$, according to (2.49), we have $\mathbf{x}_{t+1}^G(n) = \eta(\mathbf{f}_t(n); \beta_t) = \mathbf{x}_{t+1}^A(n)$; for any $n \notin \Pi$, according to (2.49), we have $\mathbf{x}_{t+1}^G(n) = 0$ and $U(n) \le \beta_t$, and according to Lemma 1, we know that $|\mathbf{f}_t(n)| \le U(n) \le \beta_t$, so $\mathbf{x}_{t+1}^A(n) = 0$. Therefore, $\mathbf{x}_{t+1}^G = \mathbf{x}_{t+1}^A$.

2.5.2 Quantized GCAMP Algorithm

In SAGC we calculate $U(n)$ based on $\mathbf{W}_t$ and T. Since $U(n)$ in SAGC is only an upper bound on $|\mathbf{f}_t(n)|$, we can use quantized $\mathbf{W}_t$ for its calculation, and for those n's with $U(n) > \beta_t$, we send all the floating-point numbers $\mathbf{w}_t^p(n)$ to Sensor 1, which could lead to a further reduction in communication cost. For a better understanding of the proposed approach, we first give a brief review of the format of floating-point

numbers.

2.5.3 Preliminary on Floating-Point Numbers

In the IEEE standard [67], to obtain the floating-point format of a real number a, we first need 1 bit to denote its sign, and then express $|a|$ using scientific notation with base 2:

$$|a| = S_a \times 2^{E_a}, \tag{2.53}$$

where $E_a = \lfloor \log_2 |a| \rfloor$ is called the exponent, and $S_a \in [1, 2)$ is called the significand in the form of $1.a_1a_2 \cdots a_k \cdots$ $(a_k \in \{0, 1\}, \forall k)$ with base 2. Note that $\{a_k\}$ can be an infinite sequence.

Since E_a is an integer, if we know that $E_a \in [E_{\min}, E_{\max}]$, then we need

$$B_e = \lceil \log_2(E_{\max} - E_{\min} + 1) \rceil \tag{2.54}$$

bits to represent E_a. For examples, the exponents of all the 64-bits floating-point numbers are within $[-1022, 1023]$, so the corresponding $B_e = \lceil \log_2(1022+1023+1) \rceil = 11$ [67].

Regarding S_a, if we only choose the first ℓ items in $\{a_k\}$, then we will obtain a number

$$S_a^\ell = 1 + \frac{\lfloor 2^\ell (S_a - 1) \rfloor}{2^\ell}. \tag{2.55}$$

It can be shown that

$$0 \leq S_a - S_a^\ell < \frac{1}{2^\ell}, \tag{2.56}$$

that is, S_a^ℓ converges to S_a at least exponentially fast. In 64-bit floating-point numbers, the first $B_s = 52$ bits are used for representing the significand [67].

2.5.4 Computing $U(n)$ Using Quantization

Here, we present the quantized GCAMP (Q-GCAMP) algorithm, which has the same structure as SAGC, only by introducing quantization in Steps I and II, and

modifying the calculation of $U(n)$:

Step I: Define T and Ω_p the same as in SAGC, and define $N_p = |\Omega_p|$, $N_p' = N - N_p$, $U_p = \max_n |\mathbf{w}_t^p(n)|$, and $L_p = \min_n |\mathbf{w}_t^p(n)|$. We can carefully choose the parameter θ so that $L_p < T < U_p$ holds. Before sending anything to Sensor 1, each Sensor $p \geq 2$ uniformly quantizes $(T, U_p]$ into $\lfloor rN_p \rfloor$ bins, and $[L_p, T]$ into $\lfloor r'N_p' \rfloor$ bins, where $r, r' \in (0, 1]$ are scaling parameters and tuned empirically. Each bin inside $(T, U_p]$ or $[L_p, T]$ is indexed from 0 to $\lfloor rN_p - 1 \rfloor$ or $\lfloor r'N_p' - 1 \rfloor$.

Then, Sensor $p \geq 2$ finds the quantization index of $\mathbf{w}_t^p(n)$, which is denoted as $\mathbf{q}_t^p(n)$, for each $n \in [N]$. It is easy to show that

$$\begin{aligned} T + \mathbf{q}_t^p(n) \frac{U_p - T}{\lfloor rN_p \rfloor} &\leq |\mathbf{w}_t^p(n)| \\ &\leq T + (\mathbf{q}_t^p(n) + 1) \frac{U_p - T}{\lfloor rN_p \rfloor}, \ \forall n \in \Omega_p, \end{aligned} \tag{2.57}$$

and

$$\begin{aligned} L_p + \mathbf{q}_t^p(n) \frac{T - L_p}{\lfloor r'N_p' \rfloor} &\leq |\mathbf{w}_t^p(n)| \\ &\leq L_p + (\mathbf{q}_t^p(n) + 1) \frac{T - L_p}{\lfloor r'N_p' \rfloor}, \ \forall n \notin \Omega_p. \end{aligned} \tag{2.58}$$

Now, Sensor p sends a package containing the sensor index p, $\mathrm{sgn}(\mathbf{w}_t^p)$, L_p, U_p, N_p, Ω_p, and $\mathbf{q}_t^p(\Omega_p)$ to Sensor 1.

Note that both L_p and U_p take 63 bits, N_p takes $\lceil \log_2 N \rceil$ bits, and $\mathbf{q}_t^p(\Omega_p)$ are integers, which take $N_p \lceil \log_2 \lfloor rN_p \rfloor \rceil$ bits. Therefore, the total number of communication bits used in Step I is

$$\begin{aligned} B_1 = &(P-1) \left[\lceil \log_2(P-1) \rceil + N + 126 + \lceil \log_2 N \rceil \right] \\ &+ \sum_{p=2}^{P} \left(\min\{ N_p \lceil \log_2 N \rceil, N \} + N_p \lceil \log_2 \lfloor rN_p \rfloor \rceil \right). \end{aligned} \tag{2.59}$$

Step II: On receiving a package, Sensor 1 can obtain a range $\left[R_{Q_1}^L(p,n), R_{Q_1}^U(p,n) \right]$ on $\mathbf{w}_t^p(n)$: for any $n \in \Omega_p$, $R_{Q_1}^L(p,n)$ and $R_{Q_1}^U(p,n)$ can be obtained according to (2.57);

for any $n \notin \Omega_p$, $R^L_{Q_1}(p,n) = L_p$ and $R^U_{Q_1}(p,n) = T$ if $w^p_t(n) > 0$, and $R^L_{Q_1}(p,n) = -T$ and $R^U_{Q_1}(p,n) = -L_p$ otherwise.

Sensor 1 can then obtain a range $\left[R^L_{Q_1}(n), R^U_{Q_1}(n)\right]$ on $\mathbf{f}_t(n)$:

$$R^L_{Q_1}(n) = \mathbf{w}^1_t(n) + \sum_{p=2}^{P} R^L_{Q_1}(p,n), \tag{2.60}$$

and

$$R^U_{Q_1}(n) = \mathbf{w}^1_t(n) + \sum_{p=2}^{P} R^U_{Q_1}(p,n). \tag{2.61}$$

Further,

$$U_{Q_1}(n) = \max\left\{R^U_{Q_1}(n), -R^L_{Q_1}(n)\right\} \tag{2.62}$$

is an upper bound on $|\mathbf{f}_t(n)|$. Sensor 1 broadcasts the set $F_{Q_1} = \{n\colon U_{Q_1}(n) \geq \beta_t\}$ to other sensors, which requires $\min\{|F_{Q_1}|\lceil\log_2 N\rceil, N\} + 1$ bits, and receives a package containing p and $\mathbf{q}^p_t(F_{Q_1}\backslash\Omega_p)$ from each Sensor p, of the size $\lceil\log_2(P-1)\rceil + |F_{Q_1}\backslash\Omega_p|\left\lceil\log_2\lfloor r' N'_p\rfloor\right\rceil$ bits.

Now, Sensor 1 can obtain a new range $\left[R^L_{Q_2}(p,n), R^U_{Q_2}(p,n)\right]$ on each $\mathbf{w}^p_t(n)$ for each $n \in F_{Q_1}$: for $n \in \Omega_p$, the range remains the same; for $n \in F_{Q_1}\backslash\Omega_p$, the range is updated according to (2.58). Therefore, we can obtain a new range $\left[R^L_{Q_2}(n), R^U_{Q_2}(n)\right]$ on $\mathbf{f}_t(n)$ and a new upper bound $U(n)$ on $|\mathbf{f}_t(n)|$ for each $n \in F_{Q_1}$, in a similar way as (2.60)-(2.62).

Sensor 1 broadcasts the "refined" set $\Pi = \{n\colon U(n) \geq \beta_t\}$ to other sensors, which requires $\min\{|\Pi|\lceil\log_2|F_{Q_1}|\rceil, |F_{Q_1}|\}$ bits.

The total number of bits in Step II is

$$\begin{aligned} B_2 = &\min\{|F_{Q_1}|\lceil\log_2 N\rceil, N\} \\ &+ (P-1)\lceil\log_2(P-1)\rceil \\ &+ \sum_{p=2}^{P} |F_{Q_1}\backslash\Omega_p|\left\lceil\log_2\lfloor r' N'_p\rfloor\right\rceil \\ &+ \min\{|\Pi|\lceil\log_2|F_{Q_1}|\rceil, |F_{Q_1}|\}. \end{aligned} \tag{2.63}$$

Step III: Up to now, both Sensors 1 and p know Π, and $\left[R_{Q_2}^L(p,n), R_{Q_2}^U(p,n)\right]$ for each $n \in \Pi$, so they can determine the minimum number of bits $B(p,n)$ required for Sensor 1 to know $\mathbf{w}_t^p(n)$, in the following way:

1) Let $B_e^L(p,n)$ and $B_e^U(p,n)$ be the exponents of $R_{Q_2}^L(p,n)$ and $R_{Q_2}^U(p,n)$, if they are not the same, then Sensor p needs to send $\lceil \log_2(|B_e^U(p,n) - B_e^L(p,n)| + 1) \rceil$ bits representing the exponent of $\mathbf{w}_t^p(n)$, and all the 52 bits of the significand of $\mathbf{w}_t^p(n)$, i.e., $B(p,n) = \lceil \log_2(|B_e^U(p,n) - B_e^L(p,n)| + 1) \rceil + 52$.

2) if $B_e^L(p,n) = B_e^U(p,n)$, then Sensor 1 knows that the exponent of $\mathbf{w}_t^p(n)$ is also $B_e^L(p,n)$. Let $\{L_n^p(k)\}$ and $\{U_n^p(k)\}$ $(k = 1, 2, \cdots, 52)$ denote the 52-bit significands of $R_{Q_2}^L(p,n)$ and $R_{Q_2}^U(p,n)$ respectively. Sensor p compares $\{L_n^p(k)\}$ and $\{U_n^p(k)\}$, and counts $B_{\text{same}} = \max\{k\colon L_n^p(j) = U_n^p(j), \forall j = 1, \cdots, k\}$. In this case, Sensor p only needs to send the last $B(p,n) = 52 - B_{\text{same}}$ bits of the significand of $\mathbf{w}_t^p(n)$ to Sensor 1.

Finally, Sensor 1 computes $\mathbf{f}_t(n)$ for each $n \in \Pi$, and obtains $\mathbf{x}_{t+1}$ correspondingly. The support set Γ takes $\min\{|\Gamma|\lceil \log_2 |\Pi| \rceil, |\Pi|\}$ bits, and $\mathbf{x}_{t+1}(\Gamma)$ takes $64|\Gamma|$ bits to broadcast.

Therefore, the number of bits in Step III is

$$\begin{aligned} B_3 = {} & (P-1)\lceil \log_2(P-1) \rceil + \sum_{p=2}^{P} \sum_{n \in \Pi} B(p,n) \\ & + \min\{|\Gamma|\lceil \log_2 |\Pi| \rceil, |\Pi|\} + 64|\Gamma|. \end{aligned} \tag{2.64}$$

As we can see, the major modification of Q-GCAMP based on SAGC is that in Steps I and II there is no communication of floating-point numbers other than L_p and U_p in Step I, the two rounds of sifting in Step II leaves a "good" candidate set Π, and in Step III, we further reduce the communication of floating-point numbers by incorporating the information of their ranges obtained in the first two steps.

Now we prove the correctness of Q-GCAMP.

Theorem 6 *Given the same threshold $\beta_t = \tau\hat{\sigma}_t$, Q-GCAMP algorithm obtains exactly the same $\mathbf{x}_{t+1}$ as that of the centralized AMP algorithm computed by* (2.2).

It is straightforward to prove Theorem 6 due to the fact that for any $n \in \Gamma$, $\mathbf{f}_t(n)$ is calculated since $U(n) \geq |\mathbf{f}_t(n)| \geq \beta_t$.

2.6 Numerical Results

2.6.1 Simulation Setup

In the simulation, we set $N = 10,000$, and choose (κ, ρ) pairs below the phase transition curve $\rho = \rho(\kappa) = \max_{\tau \geq 0} \rho(\kappa, \tau)$ to generate $\mathbf{s}_0$ with each entry drawn from i.i.d. S_0 which follows the Bernoulli-Gaussian distribution:

$$f_{S_0}(s) = \epsilon\phi(s) + (1-\epsilon)\delta(s), \tag{2.65}$$

where $\epsilon = \kappa\rho$ denotes the normalized sparsity level of $\mathbf{s}_0$, and $\phi(s)$ is the PDF of the standard normal distribution $\mathcal{N}(0,1)$.

The sensing matrix $\mathbf{A}$ with i.i.d. entries $\sim \mathcal{N}(0, \frac{1}{M})$ is equally divided into P parts by rows, with each sensor having a $(M/P) \times N$ measurement matrix, and the measurement noise $\mathbf{e}$ consists of i.i.d. $\mathcal{N}(0, \sigma^2)$ entries. We determine the values of σ^2 based on given signal-to-noise ratios (SNR), which is defined as

$$\text{SNR} = 10\log_{10}\frac{\mathbb{E}(\|\mathbf{A}\mathbf{s}_0\|_2^2)}{\mathbb{E}(\|\mathbf{e}\|_2^2)} \approx 10\log_{10}\frac{\mathbb{E}(\|\mathbf{s}_0\|_2^2)}{\mathbb{E}(\|\mathbf{e}\|_2^2)}. \tag{2.66}$$

It can be shown that for the Bernoulli-Gaussian in (2.65),

$$\text{SNR} = 10\log_{10}\frac{\rho}{\sigma^2}. \tag{2.67}$$

In order to evaluate how efficient the proposed GC algorithms are in terms of communication savings, we first consider the naive approach where each Sensor $p \geq 2$ sends the whole vector $\mathbf{w}_{t-1}^p$ to Sensor 1 to compute $\mathbf{x}_t$, which requires

$$B_{t,\max} = (P-1)\left[\lceil \log_2(P-1) \rceil + 64N\right] + \min\{\|\mathbf{x}_{t+1}\|_0 \lceil \log_2 N \rceil, N\} + 64\|\mathbf{x}_{t+1}\|_0 \tag{2.68}$$

bits, where the first line of the right hand side (RHS) is the number of bits required for sending $\mathbf{w}_t^p$ and sensor indices p to Sensor 1, and the second line refers to that used for broadcasting the support and non-zero elements of $\mathbf{x}_{t+1}$ from Sensor 1 to others. For large N, $B_{t,\max}$ is dominated by the item $64N(P-1)$, which does not depend on t.

We then define the normalized bit rate $\mathsf{NR}_t = B_t/B_{t,\max}$ for DiAMP, where B_t is the number of bits required by DiAMP in the t-th iteration.

For every proposed GC algorithm, we tune their parameters θ empirically and choose the ones that yield the best simulation results (Different algorithms may have different "optimal" θ). The tuned parameters are given as follows: in GCAMP, $\theta = 0.8$; in A-GCAMP, when GCAMP is invoked on $\mathbf{W}_t$, we set $\theta = \theta_1 = 0.8$, and when when GCAMP is invoked on $\Delta\mathbf{W}_t$, we set $\theta = \theta_2 = 0.85$; in Q-A-GCAMP, we set $\theta_1 = \theta_2 = 1.1$. In A-GCAMP and Q-A-GCAMP, ρ_w is set to 0.5.

Regarding when to terminate the algorithm, we use the convergence criterion $|\hat{\sigma}_{t+1} - \hat{\sigma}_t| \leq \zeta\hat{\sigma}_t$, where $\zeta = 1 \times 10^{-4}$. In tuning the parameter τ, we set $L = 10$, $T_1 = 100$, and $T_2 = 20$. For each (κ, ρ) pair, we have 100 Monte-Carlo simulations.

2.6.2 Accuracy of $\widehat{\sigma}_t^2$ in DiAMP

In this section we evaluate the accuracy of the proposed estimator $\hat{\sigma}_t^2$ in (2.16). In Fig. 7 we compare the proposed estimator $\widehat{\sigma}_t^2$ with $\sigma_{t,S}^2$ obtained by the SE equation (2.5), and the empirical $\sigma_{t,E}^2 = \sqrt{\|\mathbf{x}_t + \mathbf{A}^T\mathbf{z}_t - \mathbf{s}_0\|_2^2/N}$. As shown in the figure, $\widehat{\sigma}_t^2$ is very close to $\sigma_{t,E}^2$, and matches $\sigma_{t,S}^2$ reasonably well, where the match of $\widehat{\sigma}_t^2$ and $\sigma_{t,E}^2$ indicates that the proposed $\widehat{\sigma}_t^2$ is an accurate estimator of σ_t^2, and the match of $\widehat{\sigma}_t^2$

and $\sigma^2_{t,S}$ verifies that the simulation setup fits well into the large system limit where the SE equation holds.

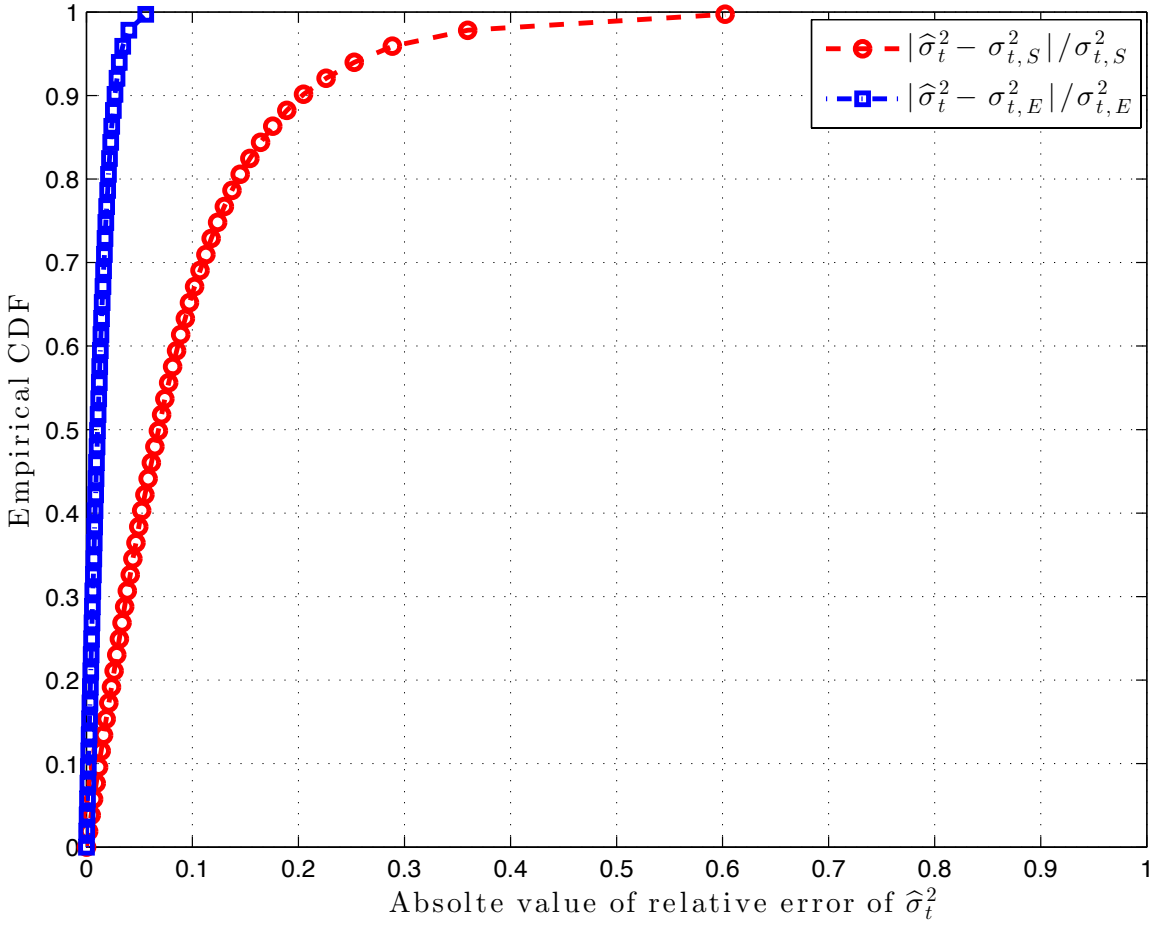

Fig. 7. Empirical CDF of the absolute values of relative errors of $\widehat{\sigma}^2_t$ with respect to $\sigma^2_{t,S}$ and $\sigma^2_{t,E}$.

2.6.3 Performance of Proposed GC Algorithms

Since the communication cost of GC algorithms is directly related to the sparsity of $\mathbf{x}_{t+1}$, to observe their relation, we compute the normalized sparsity level of $\mathbf{x}_{t+1}$, namely $\epsilon_t = \|\mathbf{x}_{t+1}\|_0/N$.

In Table 6 we list the normalized sparsity levels of $\mathbf{x}_{t+1}$ and the normalized bit rates of the proposed GC algorithms averaged over iterations, which are $\bar{\epsilon} =$

$\sum_{t=1}^{100} \epsilon_t/100$ and $\overline{\mathsf{NR}} = \sum_{t=1}^{100} B_t / \sum_{t=1}^{100} B_{t,\max}$ respectively. It can be seen that the trend (increasing or decreasing) of $\overline{\mathsf{NR}}$ agrees well with that of $\bar{\epsilon}$, which is not surprising: the more sparse $\mathbf{x}_{t+1}$ is, the lower communication cost it takes to find $\mathcal{S}(\mathbf{x}_{t+1})$.

It can be shown that MTA becomes inefficient when the total score is no longer monotone with partial scores. On the other hand, the proposed GCAMP works reasonably well and saves roughly 50% of the communication cost compared with MTA. The adaptive approach A-GCAMP improves another 7%-14% of communication savings based on GCAMP. With quantization incorporated, Q-A-GCAMP achieves further significant reduction of communication cost, ranging from 13%-26%.

More insight can be obtained from Table 6. Note that ϵ_t is factually the lower bound on NR_t. This is because the former is the percentage of $\mathbf{f}_t(n)$'s with magnitudes greater than β_t, while the latter can be approximately interpreted from another perspective — the percentage of $\mathbf{f}_t(n)$'s calculated in GC algorithms. In order not to lose any accuracy, the latter must be greater than or equal to the former, and the equality holds only if $\mathcal{S}(\mathbf{x}_{t+1})$ is known in advance. In this aspect, we can see that the $\overline{\mathsf{NR}}$ of Q-A-GCAMP comes close to its lower bound $\bar{\epsilon}$.

In Fig. 8 we provide more detailed information of the NR_t values obtained by different GC algorithms. The more top-left the empirical CDF curve of NR_t is located, the better communication saving a GC algorithm has achieved. It is clear that Q-A-GCAMP > A-GCAMP > GCAMP > MTA in terms of communication savings.

2.7 Conclusion

Assuming the sparsity of the original signal to be unknown, several DiAMP algorithms have been proposed for performing compressed sensing in distributed sensor networks, consisting a series of local and global computations. All of these algorithms have exactly the same recovery result as the centralized algorithms, given the same

Table 6. Performance of GC algorithms

	$\rho = 0.1$,SNR = 20dB			$\rho = 0.2$,SNR = 20dB		
κ	0.20	0.30	0.40	0.20	0.30	0.40
# of iterations	98.33	96.61	94.54	94.34	93.86	95.18
MTA (%)	96.65	96.40	96.32	109.23	108.47	108.31
GCAMP (%)	42.02	44.11	45.60	53.29	55.57	57.70
A-GCAMP (%)	33.90	36.25	38.15	40.50	43.18	45.43
Q-A-GCAMP (%)	10.05	12.64	14.87	12.94	15.94	18.84
$\bar{\epsilon}$ (%)	4.23	6.41	8.45	6.13	8.92	11.75
	$\rho = 0.1$,SNR = 10dB			$\rho = 0.2$,SNR = 10dB		
κ	0.20	0.30	0.40	0.20	0.30	0.40
# of iterations	83.43	81.58	80.64	75.76	73.94	74.62
MTA (%)	97.03	96.40	96.47	108.52	107.48	107.07
GCAMP (%)	43.77	45.77	47.57	53.23	55.13	57.03
A-GCAMP (%)	33.89	36.44	38.32	40.17	42.37	44.90
Q-A-GCAMP (%)	10.36	13.06	15.37	13.18	15.83	18.59
$\bar{\epsilon}$ (%)	4.78	7.17	9.40	6.25	9.00	11.84

thresholds. To reduce the communication cost in the GC step for both non-quantized and quantized data, we developed GCAMP, which is a communication-efficient data-querying algorithm and outperforms another popular algorithm MTA significantly. By taking into consideration the correlation of data between adjacent iterations and incorporating quantization steps, a more sophisticated algorithm Q-A-GCAMP is developed, which comes close to requiring the minimum bit rates stipulated by the sparsity of $\mathbf{x}_{t+1}$.

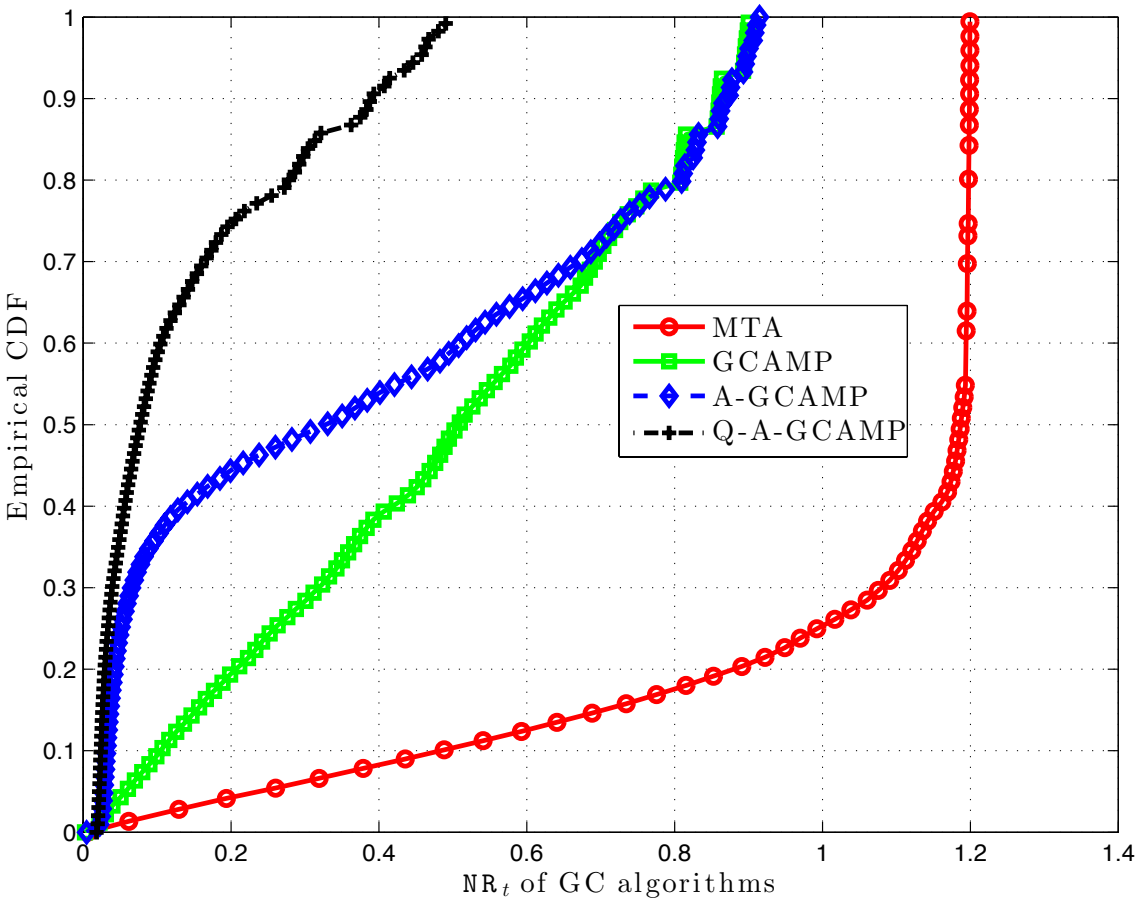

Fig. 8. Empirical CDF of NR_t for GC algorithms.

CHAPTER 3

GAUSSIANITY IN DIAMP

3.1 Introduction

Approximate message passing (AMP) [47], a statistical algorithm originally developed for compressed sensing (CS) recovery [1], turns out to have more promising applications for general linear inverse problems, including low-rank matrix completion [68], non-negative principal component analysis (PCA) [69], and code division multiple access (CDMA) systems [70, 71], etc. The popularity of AMP is due to the universality of state evolution (SE) [72], a one-dimensional recursion that determines AMP's performance.

Consider the linear inverse problem

$$\mathbf{y} = \mathbf{A}\mathbf{s}_0 + \mathbf{w}, \tag{3.1}$$

where $\mathbf{s}_0 \in \mathbb{R}^N$ is unknown and $\mathbf{w} \in \mathbb{R}^n$ is additive measurement noise, and $\mathbf{A} \in \mathbb{R}^{n\times N}$ and $\mathbf{y} \in \mathbb{R}^n$ are given, which are known as the sensing matrix and the measurement respectively. AMP obtains a sequence of estimators $\{\mathbf{x}^t\}_{t\geq 0}$ for $\mathbf{s}_0$, by starting from an initial estimate $\mathbf{x}_0 = \mathbf{0}$ and residual $\mathbf{z}_0 = \mathbf{y}$, and proceeding as follows:

$$\mathbf{u}^t = \mathbf{x}^t + \mathbf{A}^T\mathbf{z}^t, \tag{3.2}$$

$$\mathbf{x}^{t+1} = \boldsymbol{\eta}_t(\mathbf{u}^t), \tag{3.3}$$

$$\mathbf{z}^{t+1} = \mathbf{y} - \mathbf{A}\mathbf{x}^t + \frac{1}{\kappa}\gamma_t\mathbf{z}^t, \tag{3.4}$$

where $\kappa = n/N$ is called the measurement ratio, $[\cdot]^T$ denotes transposition, $\boldsymbol{\eta}_t : \mathbb{R}^N \to$

$\mathbb{R}^N$ is a separable denoiser, i.e.,

$$\boldsymbol{\eta}_t(\mathbf{u}^t) = \left[\eta_t(u_1^t), \eta_t(u_2^t), \cdots, \eta_t(u_N^t)\right]^T, \tag{3.5}$$

and

$$\gamma_t = \frac{1}{N}\sum_{i=1}^{N} \eta_t'(u_i^t), \tag{3.6}$$

where $\eta_t'(u_i^t)$ denotes the derivative of $\eta_t(u_i^t)$ with respect to u_i^t.

In AMP literature, it is a common assumption that the sensing matrix $\mathbf{A}$ consists of i.i.d. Gaussian entries with zero-mean and variance $1/n$, and the entries of $\mathbf{s}_0$ and $\mathbf{w}$ follow i.i.d. $S_0 \sim p_{S_0}$ and $W \sim p_W$ respectively, where W is zero-mean with variance σ^2; $\mathbf{u}^t$ and $\mathbf{z}^t$ in (3.2) and (3.4) are called pseudo data and measurement residual respectively, and $\frac{1}{\kappa}\gamma_t \mathbf{z}_t$ in (3.4) is known as the Onsager term.

The popularity of AMP lies in its theoretical support — state evolution (SE), which depicts the performance of AMP through an one-dimensional recursive equation. More specifically, it shows that $\mathbf{u}_t - \mathbf{s}_0$ behaves as $\mathcal{N}(\mathbf{0}, \tau_t^2 \mathbf{I}_N)$ [58], where $\mathcal{N}(\cdot,\cdot)$ denotes the normal (Gaussian) distribution, $\mathbf{I}_N$ denotes the $N \times N$ identity matrix, and τ_t^2 satisfies

$$\tau_{t+1}^2 = \sigma^2 + \frac{1}{\kappa}\mathbb{E}\left[\eta_t(S_0 + \tau_t Z) - S_0\right]^2, \tag{3.7}$$

where $Z \sim \mathcal{N}(0,1)$ is independent of S_0. Note that $\mathbb{E}\left[\eta_t(S_0 + \tau_t Z) - S_0\right]^2$ in (3.7) is exactly the mean square error (MSE) of $\mathbf{x}_{t+1}$. Therefore, the SE equation in (3.7) can also be written as

$$\sigma_{t+1}^2 = \frac{1}{\kappa}\mathrm{MSE}(\mathbf{x}_{t+1}) = \frac{1}{\kappa}\mathbb{E}\left[\eta_t(S_0 + \tau_t Z) - S_0\right]^2, \tag{3.8}$$

and

$$\tau_{t+1}^2 = \sigma^2 + \sigma_{t+1}^2. \tag{3.9}$$

The Gaussianity of $\mathbf{u}_t - \mathbf{s}_0$ comes from the Onsager term in (3.4), which is the key difference between AMP and the iterative soft thresholding (IST) algorithm in [46, 55].

SE makes AMP favorable in two aspects. First, it converts the linear inverse problem in (3.1) into a sequence of sub-problems in scalar channels:

$$\mathbf{u}_t = \mathbf{s}_0 + \boldsymbol{\theta}_t, \text{ with } \boldsymbol{\theta}_t \sim \mathcal{N}(\mathbf{0}, \tau_t^2 \mathbf{I}^N), \tag{3.10}$$

or simply

$$U_t = S_0 + \tau_t Z, \tag{3.11}$$

which avoids complicated matrix operations such as inversion, and provides a handful of choices for the denoiser η_t. For example, if $\mathbf{s}_0$ is assumed to be sparse yet with unknown prior distribution, then the soft thresholding function [47] yields the minimax reconstruction MSE [57]; otherwise, if p_{S_0} is given, then the minimum MSE (MMSE) estimator conditioning on $\mathbf{u}_t$ in (3.10), or U_t in (3.11) will be the optimal denoiser in the MSE sense.

Second, the SE equations (3.8) and (3.9) can be performed offline, which makes the performance of AMP more predictable. For example, an important parameter in AMP is the compression ratio κ, to determine what κ is good enough in order to achieve a recovery MSE below a designed threshold, one can find a corresponding solution by evaluating SE equations with binary search.

3.1.1 Existing Results about SE

In order to show the universality of SE, a generalized algorithm based on AMP was considered in [58] , which we denote as AMP-G1:

$$\mathbf{b}^t = \mathbf{A}\boldsymbol{f}_t(\mathbf{h}^t, \mathbf{s}_0) - \lambda_t \boldsymbol{g}_{t-1}(\mathbf{b}^{t-1}, \mathbf{w}), \text{ and } \mathbf{h}^{t+1} = \mathbf{A}^T \boldsymbol{g}_t(\mathbf{b}^t, \mathbf{w}) - \xi_t \boldsymbol{f}_t(\mathbf{h}^t, \mathbf{s}_0), \tag{3.12}$$

where all the elements of $\mathbf{s}_0 \in \mathbb{R}^N$ and $\mathbf{w} \in \mathbb{R}^n$ follow i.i.d. p_{S_0} and p_W respectively, $\mathbf{A} \in \mathbb{R}^{n\times N}$ is with entries following i.i.d. $\mathcal{N}(0, 1/n)$, $f_t : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$ and $g_t : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ are assumed to be separable, i.e., $\boldsymbol{f}_t(\mathbf{h}, \mathbf{s}) = [f_t(h_1, s_1), \cdots, f_t(h_N, s_N)]^T$ and $\boldsymbol{g}_t(\mathbf{b}, \mathbf{w}) = [g_t(b_1, w_1), \cdots, g_t(b_n, w_n)]^T$, with $f_t(h, s)$ and $g_t(b, w)$ being Lipschitz continuous[1], and λ_t and ξ_t are given by

$$\lambda_t = \sum_{i=1}^{N} f_t'(h_i^t, s_{0,i})/n, \text{ and } \xi_t = \sum_{i=1}^{n} g_t'(b_i^t, w_i)/n, \tag{3.13}$$

where the derivatives are with respect to the first arguments.

Denote

$$\mathbf{q}^t = f_t(\mathbf{h}^t, \mathbf{s}_0), \text{ and } \mathbf{m}^t = g_t(\mathbf{b}^t, \mathbf{w}). \tag{3.14}$$

It can be shown that the original AMP is a special case of AMP-G1 with

$$\begin{aligned}
&\mathbf{h}^{t+1} = \mathbf{s}_0 - \mathbf{u}^t,\\
&\mathbf{b}^t = \mathbf{w} - \mathbf{z}^t,\\
&\mathbf{q}^t = \mathbf{s}_0 - \mathbf{x}^t,\\
&\mathbf{m}^t = -\mathbf{z}^t,\\
&\boldsymbol{f}_t\left(\mathbf{h}^t, \mathbf{s}_0\right) = \boldsymbol{\eta}_{t-1}(\mathbf{s}_0 - \mathbf{h}^t) - \mathbf{s}_0,\\
&\boldsymbol{g}_t\left(\mathbf{b}^t, \mathbf{w}\right) = \mathbf{b}^t - \mathbf{w}.
\end{aligned} \tag{3.15}$$

AMP-G1 starts with initial values $\mathbf{q}^0 \in \mathbb{R}^N$ satisfying some statistical property, and $\mathbf{m}^{-1} = \mathbf{0}$, while the original AMP starts with $\mathbf{x}^0 = \mathbf{0}$, corresponding to $\mathbf{q}^0 = \mathbf{s}_0$. By comparison, it is useful to model the initial value $\mathbf{q}^0$ in AMP-G1 as a function of $\mathbf{s}_0$ and another random vector $\mathbf{x}^0$ independent of $\mathbf{s}_0$.

By applying a so-called conditioning technique that originates from Spin Glass

[1]A function $\boldsymbol{f} : \mathbb{R}^p \to \mathbb{R}^q$ is said to be Lipschitz continuous if there exists $L > 0$ only depending on $\boldsymbol{f}$ such that $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^p$, $||\boldsymbol{f}(\mathbf{x}) - \boldsymbol{f}(\mathbf{y})|| \leq L\|\mathbf{x} - \mathbf{y}\|$.

Theory [73], in [58] it was shown that elements in $\mathbf{h}^{t+1}$ and $\mathbf{b}^t$ are asymptotically i.i.d. Gaussian as $N \to \infty$ and $n/N \to \kappa > 0$, with their variances τ_t^2 and σ_t^2 satisfying the generalized SE:

$$\sigma_t^2 = \frac{1}{\kappa}\mathbb{E}\left[f_t(\tau_{t-1}Z, S_0\right]^2, \tag{3.16}$$

and

$$\tau_t^2 = \mathbb{E}\left[g_t(\sigma_t Z, W)\right]^2. \tag{3.17}$$

The essence of the conditioning technique is to view AMP-G1 as a generator of linear constraints for $\mathbf{A}$:

$$\begin{aligned} \mathbf{A}\mathbf{q}^t &= \mathbf{b}^t + \lambda_t \mathbf{m}^{t-1}, \\ \mathbf{A}^T\mathbf{m}^t &= \mathbf{h}^{t+1} + \xi_t \mathbf{q}^t, \end{aligned} \tag{3.18}$$

Note that AMP-G1 can be divided into two stages: i) given

$$\mathfrak{S}_{t,t} \stackrel{\triangle}{=} \{\mathbf{w}, \mathbf{s}_0, \mathbf{q}^0, \mathbf{b}^0, \mathbf{m}^0, \mathbf{h}^1, \mathbf{q}^1, \cdots, \mathbf{b}^{t-1}, \mathbf{m}^{t-1}, \mathbf{h}^t, \mathbf{q}^t\},$$

it obtains $\mathbf{b}^t$ and $\mathbf{m}^t$, and ii) given

$$\mathfrak{S}_{t+1,t} \stackrel{\triangle}{=} \mathfrak{S}_{t,t} \cup \{\mathbf{b}^t, \mathbf{m}^t\},$$

it obtains $\mathbf{h}^{t+1}$ and $\mathbf{q}^{t+1}$, and has

$$\mathfrak{S}_{t+1,t+1} \stackrel{\triangle}{=} \mathfrak{S}_{t+1,t} \cup \{\mathbf{h}^{t+1}, \mathbf{q}^{t+1}\},$$

The proof in [58] is sketched as follows: since $\mathbf{A}$ is Gaussian, its conditional distribution under linear constraints determined by $\mathfrak{S}_{t,t}$ and $\mathfrak{S}_{t+1,t}$ is still Gaussian. Based on $\mathbf{A}|\mathfrak{S}_{t,t}$ and $\mathbf{A}|\mathfrak{S}_{t+1,t}$, one can further obtain the conditional distributions $\mathbf{b}^t|\mathfrak{S}_{t,t}$ and $\mathbf{h}^{t+1}|\mathfrak{S}_{t+1,t}$. Since they are linear in $\mathbf{A}$, their conditional distributions are also Gaussian. Then, one can take the limit $N \to \infty$ and $n/N \to \kappa > 0$,

and show that their conditional distributions become irrelevant to $\mathfrak{S}_{t,t}$ and $\mathfrak{S}_{t+1,t}$ asymptotically, which finishes the proof.

In [74] a Bayesian optimal AMP was proposed, where the denoiser is the minimum-mean-squared-error (MMSE) estimator, for solving the linear inverse problem in (3.1), by assuming a prior knowledge of p_{S_0}. In that paper the sensing matrix $\mathbf{A}$ is replaced by a so-called spatially-coupled sensing matrices, where each element is zero-mean and Gaussian, yet with different variances. In [74] it was shown that SE still holds in this case, and proved that the Bayesian optimal AMP achieves a mean-squared-error (MSE) of $O(\sigma^2)$ asymptotically as $N \to \infty$, $n/N \to \kappa > 0$, and $t \to \infty$, where σ^2 is the variance of noise.

In [75] AMP was further combined with spatially-coupled sensing matrices into a more general algorithm, which we denote as AMP-G2:

$$\begin{aligned} \mathbf{\Pi}_t &= \mathbf{A}\boldsymbol{F}_t(\mathbf{\Psi}^t, \mathbf{S}) - \frac{1}{\kappa}\boldsymbol{G}_{t-1}(\mathbf{\Pi}^{t-1}, \mathbf{Y})\mathbf{J}_{\psi,t}^T, \\ \mathbf{\Psi}_{t+1} &= \mathbf{A}^T\boldsymbol{G}_t(\mathbf{\Pi}^t, \mathbf{Y}) - \boldsymbol{F}_t(\mathbf{\Psi}^t, \mathbf{S})\mathbf{J}_{\pi,t}^T, \end{aligned} \tag{3.19}$$

where $\mathbf{A} \in \mathbb{R}^{n\times N}$ consists of i.i.d. $\mathcal{N}(0, 1/n)$ entries, $\mathbf{S}$ and $\mathbf{Y}$ are $N \times q$ and $n \times q$ matrices, with each of their rows $\mathbf{s}_i \in \mathbb{R}^q$ and $\mathbf{y}_j \in \mathbb{R}^q$ following i.i.d. $p_{\mathbf{S}}$ and $p_{\mathbf{Y}}$ correspondingly, $\boldsymbol{F}_t : \mathbb{R}^{N\times q} \times \mathbb{R}^{N\times q} \to \mathbb{R}^{N\times q}$ and $\boldsymbol{G}_t : \mathbb{R}^{n\times q} \times \mathbb{R}^{n\times q} \to \mathbb{R}^{n\times q}$ are assumed to be row-separable, i.e., $\boldsymbol{F}_t(\mathbf{\Psi}^t, \mathbf{S}) = [\boldsymbol{f}_t^{row}(\boldsymbol{\psi}_1^t, \mathbf{s}_1), \cdots, \boldsymbol{f}_t^{row}(\boldsymbol{\psi}_N^t, \mathbf{s}_N)]^T$, and $\boldsymbol{G}_t(\mathbf{\Pi}^t, \mathbf{Y}) = [\boldsymbol{g}_t^{row}(\boldsymbol{\pi}_1^t, \mathbf{y}_1), \cdots, \boldsymbol{g}_t^{row}(\boldsymbol{\pi}_n^t, \mathbf{y}_n)]^T$, with $\boldsymbol{f}_t^{row} : \mathbb{R}^q \to \mathbb{R}^q$ and $\boldsymbol{g}_t^{row} : \mathbb{R}^q \to \mathbb{R}^q$ being Lipschitz continuous, and $\mathbf{J}_{\psi,t} \in \mathbb{R}^{q\times q}$ and $\mathbf{J}_{\pi,t} \in \mathbb{R}^{q\times q}$ are the empirical averages of Jacobian of $\boldsymbol{f}_t^{row}(\boldsymbol{\psi}_i^t, \mathbf{s}_i)$ and $\boldsymbol{g}_t^{row}(\boldsymbol{\pi}_j^t, \mathbf{y}_j)$ with respect to their

first arguments, i.e.,

$$
\begin{aligned}
\mathbf{J}_{\psi,t} &= \frac{1}{N}\sum_{i=1}^{N} \nabla_{\boldsymbol{\psi}_i^t}^T \boldsymbol{f}_t^{row}(\boldsymbol{\psi}_i^t, \mathbf{s}_i) \\
&= \frac{1}{N}\sum_{i=1}^{N} \begin{bmatrix} \frac{\partial f_{t,1}^{row}(\boldsymbol{\psi}_i^t,\mathbf{s}_i)}{\partial \psi_{i,1}^t} & \cdots, & \frac{\partial f_{t,1}^{row}(\boldsymbol{\psi}_i^t,\mathbf{s}_i)}{\partial \psi_{i,q}^t} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{t,q}^{row}(\boldsymbol{\psi}_i^t,\mathbf{s}_i)}{\partial \psi_{i,1}^t} & \cdots, & \frac{\partial f_{t,q}^{row}(\boldsymbol{\psi}_i^t,\mathbf{s}_i)}{\partial \psi_{i,q}^t} \end{bmatrix}, \\
\mathbf{J}_{\pi,t} &= \frac{1}{n}\sum_{i=1}^{n} \nabla_{\boldsymbol{\pi}_i^t}^T \boldsymbol{g}_t^{row}(\boldsymbol{\pi}_i^t, \mathbf{y}_i) \\
&= \frac{1}{n}\sum_{i=1}^{n} \begin{bmatrix} \frac{\partial g_{t,1}^{row}(\boldsymbol{\pi}_i^t,\mathbf{y}_i)}{\partial \pi_{i,1}^t} & \cdots, & \frac{\partial g_{t,1}^{row}(\boldsymbol{\pi}_i^t,\mathbf{y}_i)}{\partial \pi_{i,q}^t} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_{t,q}^{row}(\boldsymbol{\pi}_i^t,\mathbf{y}_i)}{\partial \pi_{i,1}^t} & \cdots, & \frac{\partial g_{t,q}^{row}(\boldsymbol{\pi}_i^t,\mathbf{y}_i)}{\partial \pi_{i,q}^t} \end{bmatrix}.
\end{aligned}
\tag{3.20}
$$

It can be verified that for the case $q = 1$, AMP-G2 is the same as AMP-G1. By applying a conditioning technique similar to [58], in [75] it was shown that rows of $\mathbf{\Pi}_t$ and $\mathbf{\Psi}_{t+1}$ are asymptotically Gaussian as $N \to \infty$ and $n/N \to \kappa > 0$. Finally, the state-of-art work in [72] showed that SE still holds for AMP-G2 when $\mathbf{A}$ consists of independent zero-mean subgaussian[2] entries with variance $1/n$.

3.1.2 Our Contribution: State Evolution in Distributed AMP

Recently, the advance of distributed processing techniques has helped increase more attention to large-scale linear inverse problems, due to its potential to speed up the recovery process. In our earlier work [41], a distributed AMP (DiAMP) algorithm was developed, which decomposes $\mathbf{u}^t$ in (3.2) into a summation of P vectors $\mathbf{u}_p^t \in \mathbb{R}^N$,

[2]The definition of subgaussian random variables will be introduced in Section 3.3.1.

with

$$\mathbf{u}_p^t = \begin{cases} \mathbf{x}^t + \mathbf{A}_p^T \mathbf{z}_p^t, & \text{if } p = 1, \\ \mathbf{A}_p^T \mathbf{z}_p^t, & \text{otherwise,} \end{cases} \tag{3.21}$$

where $\begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_p \\ \vdots \\ \mathbf{A}_P \end{bmatrix} = \mathbf{A}$, and $\begin{bmatrix} \mathbf{z}_1^t \\ \vdots \\ \mathbf{z}_p^t \\ \vdots \\ \mathbf{z}_P^t \end{bmatrix} = \mathbf{z}^t$ are row partitions of $\mathbf{A}$ and $\mathbf{z}^t$ respectively. In the designed system, P sensors take the corresponding partitions $\mathbf{A}_p$, and compute $\mathbf{u}_p^t$, which dissertation AMP running in a parallel pattern. For presentation purposes, we use some different notations from the previous chapter, for example, $\mathbf{A}_p$ instead of $\mathbf{A}^p$ when denoting the p-th row partition of $\mathbf{A}$ at most places of this chapter, where the reason should be straightforward as will be shown later.

On the meantime, the summation of P vectors $\mathbf{u}_p^t$ implies inter-sensor communication, which can be a challenging issue as the problem size goes up. In [76], lossy compression is introduced in DiAMP in order to save communication cost, where each sensor quantizes $\mathbf{u}_p^t$ into

$$Q(\mathbf{u}_p^t) = \mathbf{u}_p^t + \mathbf{v}_p^t, \tag{3.22}$$

with the quantization error $\mathbf{v}_p^t$, and sends $Q(\mathbf{u}_p^t)$ to the fusion center. If the uniform scalar quantization is performed and the characteristic function of $\mathbf{u}_p^t$ is band limited, or if vector quantization is performed, it is a good approximation to model $\mathbf{v}_p^t$ as a random vector with i.i.d. zero-mean elements in addition that $\mathbf{v}_p^t$ is independent of $\mathbf{u}_p^t$ [66, 77].

Finally, the fusion center obtains

$$\widetilde{\mathbf{u}}^t = \sum_{p=1}^{P} Q(\mathbf{u}_p^t) = \mathbf{u}^t + \mathbf{v}^t \tag{3.23}$$

with $\mathbf{v}^t = \sum_{p=1}^{P} \mathbf{v}_p^t$ independent of $\mathbf{u}^t$, and updates the estimation of $\mathbf{s}_0$ as

$$\mathbf{x}^{t+1} = \eta_t(\widetilde{\mathbf{u}}^t). \tag{3.24}$$

Furthermore, compared with [41], the computation of $\mathbf{u}_p^t$ in [76] is slightly changed:

$$\mathbf{u}_p^t = \omega_p \mathbf{x}^t + \mathbf{A}_p^T \mathbf{z}_p^t, \tag{3.25}$$

where $\omega_p = n_p/n$, with n_p being the number of rows of $\mathbf{A}_p$. The modification was made because numerical results indicate that $\mathbf{u}_p^t - \omega_p \mathbf{s}_0$ behaves like a random vector following $\mathcal{N}(\mathbf{0}, \omega_p \tau_{t,Q}^2 \mathbf{I}_N)$, where

$$\tau_{t+1,Q}^2 = \sigma^2 + \frac{1}{\kappa}\mathbb{E}\left[\eta_t(S_0 + \tau_{t,Q} Z + V_t) - S_0\right]^2, \tag{3.26}$$

and V_t has the same distribution as all the elements in $\mathbf{v}^t$. As we can see, if there is no quantization error ($V_t = 0$), then $\tau_{t,Q}^2$ is exactly the same as that in (3.7). In other words, SE still holds for DiAMP, even in the presence of quantization error, which is exactly what we are trying to prove in this dissertation.

In light of the universality of SE for centralized AMP, we have good reason to conjecture that SE holds for the following more general DiAMP algorithms, which we denote as DiAMP-G1(G2):

local computation:

$$\text{DiAMP-G1}\begin{cases} \mathbf{b}_p^t = \mathbf{A}_p\mathbf{q}^t - \lambda_t \boldsymbol{g}_{t-1}(\mathbf{b}_p^{t-1}, \mathbf{w}_p), \\ \mathbf{h}_p^{t+1} = \mathbf{A}_p^T \boldsymbol{g}_t(\mathbf{b}_p^t, \mathbf{w}_p) - \omega_p \xi_t \mathbf{q}_t, \end{cases}$$

$$\text{DiAMP-G2}\begin{cases} \boldsymbol{\Pi}_p^t = \mathbf{A}_p\boldsymbol{\Phi}^t - \frac{1}{\kappa}\boldsymbol{G}_{t-1}(\boldsymbol{\Psi}_p^{t-1}, \mathbf{Y}_p)\mathbf{J}_{\psi,t}^T, \\ \boldsymbol{\Psi}_p^{t+1} = \mathbf{A}_p^T \boldsymbol{G}_t(\boldsymbol{\Psi}_p^t, \mathbf{Y}_p) - \omega_p \boldsymbol{\Phi}^t \mathbf{J}_{\pi,t}^T, \end{cases} \tag{3.27}$$

global computation:

$$\begin{aligned} \text{DiAMP-G1: } \mathbf{q}^t &= \boldsymbol{f}_t\left(\sum_{p=1}^P \mathbf{h}_p^t, \mathbf{v}^t, \mathbf{s}_0\right), \\ \text{DiAMP-G2: } \boldsymbol{\Phi}^t &= \boldsymbol{F}_t\left(\sum_{p=1}^P \boldsymbol{\Psi}_p^t, \mathbf{V}^t, \mathbf{S}\right), \end{aligned} \tag{3.28}$$

where $\mathbf{v}^t$ or $\mathbf{V}^t$ accounts for quantization noise, which is assumed to be independent of $\mathbf{h}_p^{t+1}$ or $\boldsymbol{\Psi}_p^{t+1}$, and mutually independent for different t. As we can see, when $\mathbf{v}^t = \mathbf{0}$ or $\mathbf{V}^t = \mathbf{0}$, the proposed DiAMP algorithms are equivalent to centralized AMP algorithms.

In this dissertation, we will show stronger results for DiAMP than previous work. For simplicity of illustration, we are only going to prove that DiAMP-G1 has the following properties: (This proof can be easily extended for DiAMP-G2)

i) In addition to the Gaussianity of $\mathbf{h}^{t+1} = \sum_{p=1}^P \mathbf{h}_p^{t+1}$ and $\mathbf{b}^t$ in centralized AMP where there is no quantization noise, we show that each individual $\mathbf{h}_p^{t+1}$ and $\mathbf{b}_p^t$ are also asymptotically zero-mean Gaussian with variances $\omega_p\tau_t^2$ and σ_t^2 respectively, where

$$\sigma_t^2 = \frac{1}{\kappa}\mathbb{E}\left[f_t(\tau_{t-1}Z, V_t, S_0)\right]^2, \tag{3.29}$$

and

$$\tau_t^2 = \mathbb{E}\left[g_t(\sigma_t Z, W)\right]^2. \tag{3.30}$$

ii) We provide a broader class of sensing matrices than [72] where SE holds for DiAMP and the centralized AMP. The proof is based on our own developed technique called ***augmenting***.

Specifically, we have the following theorem:

Theorem 7 *For DiAMP-G1 in* (3.27) *and* (3.28)*, if* $\mathbf{A} \in \mathbb{R}^{n\times N}$*,* $\mathbf{s}_0 \in \mathbb{R}^N$*, and* $\mathbf{w} \in \mathbb{R}^n$ *satisfy the following conditions:*

i) $\mathbf{A}$*,* $\mathbf{s}_0$*, and* $\mathbf{w}$ *are independent;*

ii) $\mathbf{A}$ *consists of independent zero-mean entries with variance* $1/n$*, and each row (column) of* $\mathbf{A}$ *satisfies Lindeberg's condition*[3]*;*

iii) The empirical distribution[4] *of* $\mathbf{s}_0$*,* $\widehat{F}_{\mathbf{s}_0}(x)$ *converges weakly (in distribution) to a probability measure* $F_{S_0}(x)$ *with bounded moments up to the order of* $2k-2$ *for* $k \geq 2$ *as* $N \to \infty$*, and the empirical* $(2k-2)$*-th moment of* $\mathbf{s}_0$*,* $\widehat{\mathbb{E}}_{\widehat{F}_{\mathbf{s}_0}}|S_0|^{2k-2} \triangleq \sum_{i=1}^N s_{0,i}^{2k-2}/N \to \mathbb{E}_{F_{\mathbf{S}_0}}|S_0|^{2k-2}$*;*

iv) The empirical distribution of the initial value $\mathbf{q}^0$*,* $\widehat{F}_{\mathbf{q}^0}(x)$*,* $\forall p \in [P]$*, converges weakly to a probability measure* $F_{Q_0}(x)$*, with* Q_0 *being a Lipschitz continuous function of* S_0 *and another random variable* X_0 *independent of* S_0*, and assume that* $\langle \mathbf{q}^0, \mathbf{q}^0\rangle \to \kappa\sigma_0^2$ *and* $\sum_{i=1}^N |q_i^0|^{2k-2}/N < \infty$ *almost surely;*

v) the empirical distribution of $\mathbf{w}_p$*,* $\widehat{F}_{\mathbf{w}_p}(x)$*,* $\forall p \in [P]$*, converges weakly to a common probability measure* $F_W(x)$ *with bounded moments up to the order of* $2k-2$ *for* $k \geq 2$ *as* $n_p \to \infty$*, where* n_p *is the number of rows of* $\mathbf{A}_p$*, and the empirical* $(2k-2)$*-th moment of* $\mathbf{w}_p$*,* $\widehat{\mathbb{E}}_{\widehat{F}_{\mathbf{w}_p}}|W|^{2k-2} \triangleq \sum_{i=1}^N w_{p,i}^{2k-2}/n_p \to \mathbb{E}_{F_{\mathbf{W}}}|W|^{2k-2}$*;*

vi) $\mathbf{v}^t$ *in* (3.28) *is independent of* $\mathbf{A}$*,* $\mathbf{s}_0$*, and* $\mathbf{w}$*; in addition, all the elements of* $\mathbf{v}^t$ *follow i.i.d.* F_{V_t}*, where* V_t *are independent for different* t*,*

[3] *Lindeberg's condition will be introduced in Section 3.3.1.*

[4] *Empirical distribution is defined in Section 3.2, Corollary 1.*

then as $N \to \infty$ *and* $n/N \to \kappa > 0$ *with* t, P, $\omega_1, \cdots,$ *and* ω_P *fixed, for any pseudo Lipschitz functions*[5] $\psi_h : \mathbb{R}^{P+2} \to \mathbb{R}$ *and* $\psi_b : \mathbb{R}^2 \to \mathbb{R}$ *of order* k*, the following holds:*

$$\begin{aligned} &\frac{1}{N}\sum_{i=1}^{N} \psi_h(\left[\mathbf{h}_1^{t+1}\right]_i, \cdots, \left[\mathbf{h}_P^{t+1}\right]_i, \left[\mathbf{v}^t\right]_i, [\mathbf{s}_0]_i) \overset{a.s.}{\to} \\ &\mathbb{E}\psi_h\left(\tau_t\sqrt{\omega_1}Z_1^t, \cdots, \tau_t\sqrt{\omega_P}Z_P^t, V_t, S_0\right), \end{aligned} \tag{3.31}$$

where $Z_1^t, \cdots, Z_P^t \sim$ *i.i.d.* $\mathcal{N}(0,1)$*, and*

$$\frac{1}{n_p}\sum_{i=1}^{N} \psi_b(\left[\mathbf{b}_p^t\right]_i, [\mathbf{w}_p]_i) \overset{d}{\to} \mathbb{E}\psi_b\left(\sigma_t\widehat{Z}_p^t, W\right), \tag{3.32}$$

where $\widehat{Z}_1^t, \cdots, \widehat{Z}_P^t \sim$ *i.i.d.* $\mathcal{N}(0,1)$*.*

Moreover, if $\mathbf{A}$ *only consists of i.i.d.* $\mathcal{N}(0, 1/n)$ *entries, the weak convergences in* (3.31) *and* (3.32) *can be replaced by almost sure convergences.*

Our focus in this chapter is to prove Theorem 7.

3.1.3 Organization, Definitions, and Notations

In the following, we will first prove the asymptotic Gaussianity of $\mathbf{h}_p^{t+1}$ and $\mathbf{b}_p^t$, under the case where $\mathbf{A}$ consists of i.i.d. $\mathcal{N}(0, 1/n)$ entries, by applying similar techniques in [58] and [75]. Then, we will present our proposed augmenting technique to show the universality of SE for more general cases, thus finishing the proof of Theorem 7.

Throughout the proof, we use the following definitions and notations similar to [58].

Upper-case and lower-case bold letters with (without) superscript and subscript denote matrices and vectors respectively. $[\mathbf{v}_{sub}^{sup}]_i$, or simply $v_{sub,i}^{sup}$ denotes the i-th

[5] *A function* $\boldsymbol{f} : \mathbb{R}^p \to \mathbb{R}^q$ *is said to be pseudo Lipschitz continuous of order* k *if there exists* $L > 0$ *only depending on* $\boldsymbol{f}$ *such that* $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^p$, $||\boldsymbol{f}(\mathbf{x}) - \boldsymbol{f}(\mathbf{y})|| \leq L(1 + \|\mathbf{x}\|^{k-1} + \|\mathbf{y}\|^{k-1})\|\mathbf{x} - \mathbf{y}\|$*.[58]*

element of the vector $\mathbf{v}_{sub}^{sup}$. Similarly, $[\mathbf{M}_{sub}^{sup}]_{i,j}$ and $M_{sub,i,j}^{sup}$ both denote the element at the i-th row and j-th column of the matrix $\mathbf{M}_{sub}^{sup}$. $f'(x, \cdots)$ denotes the partial derivative of the function f with respect to the first argument x, and for $\mathbf{x} = [x_1, \cdots, x_n]^T$, $\boldsymbol{f}(\mathbf{x}, \cdots)$ denotes $[f(x_1, \cdots), \cdots, f(x_n, \cdots)]^T$, and $\boldsymbol{f}'(\mathbf{x}, \cdots)$ denotes $[f'(x_1, \cdots), \cdots, f'(x_n, \cdots)]^T$. $\|\cdot\|_p$ denotes ℓ_p norm, and without specification, $\|\cdot\|$ denotes ℓ_2 norm. Indicator function $\mathbb{I}_{\texttt{bool}}$ or $\mathbb{I}(\texttt{bool})$ returns 1 if `bool` is true and 0 otherwise, and $\mathbb{I}_{\mathcal{A}}(x)$ means $\mathbb{I}(x \in \mathcal{A})$. $\delta(x)$ denotes Dirac Delta function, while δ_{ij} means $\mathbb{I}(i = j)$, which is the Kronecker delta. $\langle\cdot\rangle$ denotes the empirical average of elements in a vector, while $\langle\cdot,\cdot\rangle$ for $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ denotes $\langle \mathbf{u}, \mathbf{v}\rangle = \mathbf{u}^T\mathbf{v}/n$. $\overrightarrow{o}_t^n(1)$ denotes a sequence of length-t vectors where all the elements converge to 0 almost surely as $n \to \infty$. For simplicity, we will omit the superscript n, and when $t = 1$, we use $o(1)$ to denote $\overrightarrow{o}_1(1)$. $[n]$ denotes the set $\{1, 2, \cdots, n\}$. PL_k denotes the set of pseudo Lipschtiz continuous functions of order k.

Define $\mathbf{m}_p^t = \boldsymbol{g}_t(\mathbf{b}_p^t, \mathbf{w}_p)$ and the following:

$$\mathbf{H}_p^t = \left[\mathbf{h}_p^1 | \cdots | \mathbf{h}_p^t\right], \mathbf{M}_p^t = \left[\mathbf{m}_p^0 | \cdots | \mathbf{m}_p^{t-1}\right], \tag{3.33}$$

$$\mathbf{B}_p^t = \left[\mathbf{b}_p^0 | \cdots | \mathbf{b}_p^{t-1}\right], \mathbf{Q}^t = \left[\mathbf{q}^0 | \cdots | \mathbf{q}^{t-1}\right], \tag{3.34}$$

$$\mathbf{X}_p^t = \left[\mathbf{h}_p^1 + \xi_0\omega_p\mathbf{q}^0 | \cdots | \mathbf{h}_p^t + \xi_{t-1}\omega_p\mathbf{q}^{t-1}\right], \tag{3.35}$$

$$\mathbf{Y}_p^t = \left[\mathbf{b}_p^0 | \mathbf{b}_p^1 + \lambda_1\mathbf{m}_p^0 | \cdots | \mathbf{b}_p^{t-1} + \lambda_{t-1}\mathbf{m}_p^{t-2}\right]. \tag{3.36}$$

It is easy to show that

$$\mathbf{X}_p^t = \mathbf{H}_p^t + \mathbf{Q}^t\boldsymbol{\Xi}_p^t, \mathbf{Y}_p^t = \mathbf{B}_p^t + \left[\mathbf{0} | \mathbf{M}_p^{t-1}\right]\boldsymbol{\Lambda}^t, \tag{3.37}$$

where

$$\mathbf{\Xi}_p^t = \omega_p \begin{bmatrix} \xi_0 & & \\ & \ddots & \\ & & \xi_{t-1} \end{bmatrix}, \ \mathbf{\Lambda}^t = \begin{bmatrix} \lambda_0 & & \\ & \ddots & \\ & & \lambda_{t-1} \end{bmatrix}, \tag{3.38}$$

further,

$$\mathbf{X}_p^{t_2} = \mathbf{A}_p^T \mathbf{M}_p^{t_2}, \ \mathbf{Y}_p^{t_1} = \mathbf{A}_p \mathbf{Q}^{t_1}, \ \forall t_1, t_2. \tag{3.39}$$

From (3.39) it can be shown that

$$(\mathbf{X}_p^{t_2})^T \mathbf{Q}^{t_1} = (\mathbf{M}_p^{t_2})^T \mathbf{Y}_p^{t_1} = (\mathbf{M}_p^{t_2})^T \mathbf{A}_p \mathbf{Q}^{t_1}. \tag{3.40}$$

Define the following σ-algebras

$$\begin{aligned} &\mathfrak{S}_{t_1,t_2}^p \triangleq \{\mathbf{H}_p^{t_1}, \mathbf{Q}^{t_1}, \mathbf{v}^1, \cdots, \mathbf{v}^{t_1}, \mathbf{B}_p^{t_2}, \mathbf{M}_p^{t_2}, \mathbf{s}_0, \mathbf{w}_p\}, \\ &\text{and } \mathfrak{S}_{t_1,t_2} \triangleq \bigcup_{p=1}^{P} \mathfrak{S}_{t_1,t_2}^p. \end{aligned} \tag{3.41}$$

For a full-column matrix $\mathbf{M}$, $\mathbf{M}^\dagger = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T$ denotes its pseudo inverse, $span(\mathbf{M})$ denotes its column space, and $\mathbf{P}_\mathbf{M} = \mathbf{M}\mathbf{M}^\dagger$ and $\mathbf{P}_\mathbf{M}^\perp = \mathbf{I} - \mathbf{P}_\mathbf{M}$ denote the projectors onto $span(\mathbf{M})$, and the orthogonal complement of $span(\mathbf{M})$ respectively.

Based on the above notations, we can write

$$\mathbf{q}^t = \mathbf{q}_{||}^t + \mathbf{q}_\perp^t, \ \mathbf{m}_p^t = \mathbf{m}_{p,||}^t + \mathbf{m}_{p,\perp}^t, \tag{3.42}$$

where

$$\begin{aligned} &\mathbf{q}_{||}^t \triangleq \mathbf{P}_{\mathbf{Q}^t} \mathbf{q}^t, \ \mathbf{q}_\perp^t \triangleq \mathbf{P}_{\mathbf{Q}^t}^\perp \mathbf{q}^t, \\ &\mathbf{m}_{p,||}^t \triangleq \mathbf{P}_{\mathbf{M}_p^t} \mathbf{m}_p^t, \ \mathbf{m}_{p,\perp}^t \triangleq \mathbf{P}_{\mathbf{M}_p^t}^\perp \mathbf{m}_p^t. \end{aligned} \tag{3.43}$$

By definition, we have

$$(\mathbf{Q}^t)^T \mathbf{q}_\perp^t = \mathbf{0}, \ (\mathbf{M}_p^t)^T \mathbf{m}_{p,\perp}^t = \mathbf{0}, \tag{3.44}$$

and

$$\mathbf{m}_{p,\text{II}}^t = \mathbf{M}_p^t \boldsymbol{\alpha}_p^t,\ \mathbf{q}_{\text{II}}^t = \mathbf{Q}^t \boldsymbol{\beta}^t, \tag{3.45}$$

where

$$\begin{aligned} \boldsymbol{\alpha}_p^t &= \left[\alpha_p^0, \alpha_p^1, \cdots, \alpha_p^{t-1}\right]^T = (\mathbf{M}_p^t)^\dagger \mathbf{m}_p^t, \\ \boldsymbol{\beta}^t &= \left[\beta_0, \beta_1, \cdots, \beta_{t-1}\right]^T = (\mathbf{Q}^t)^\dagger \mathbf{q}^t. \end{aligned} \tag{3.46}$$

Further, define

$$\widetilde{\mathbf{Q}}^t = \mathbf{Q}^t \left[\frac{(\mathbf{Q}^t)^T \mathbf{Q}^t}{N}\right]^{-\frac{1}{2}} \text{ and } \widetilde{\mathbf{M}}_p^t = \mathbf{M}_p^t \left[\frac{(\mathbf{M}_p^t)^T \mathbf{M}_p^t}{n_p}\right]^{-\frac{1}{2}},$$

it is easy to show that

$$\widetilde{\mathbf{Q}}^t (\widetilde{\mathbf{Q}}^t)^T = N \mathbf{P}_{\mathbf{Q}^t},\ \widetilde{\mathbf{M}}_p^t (\widetilde{\mathbf{M}}_p^t)^T = n_p \mathbf{P}_{\mathbf{M}_p^t}, \tag{3.47}$$

and

$$(\widetilde{\mathbf{Q}}^t)^T \widetilde{\mathbf{Q}}^t = N \mathbf{I}_t,\ (\widetilde{\mathbf{M}}_p^t)^T \widetilde{\mathbf{M}}_p^t = n_p \mathbf{I}_t, \tag{3.48}$$

i.e., columns of $\widetilde{\mathbf{Q}}^t/\sqrt{N}$ and $\widetilde{\mathbf{M}}_p^t/\sqrt{n_p}$ form orthonormal basis of $span(\mathbf{Q}^t)$ and $span(\mathbf{M}_p^t)$ respectively.

3.2 Proof for i.i.d. Gaussian Sensing Matrices

In this section, we focus on DiAMP where $\mathbf{A}$ consists of i.i.d. Gaussian entries. In order to show that Theorem 7 holds for this case, we first need to prove a more thorough lemma, similar to Lemma 1 in [58].

Lemma 4 *For DiAMP-G1, given the sequence $\{\sigma_t^2\}$ and $\{\tau_t^2\}$ generated according to* (3.29) *and* (3.30)*, where $\mathbf{A}$, $\mathbf{s}_0$, $\mathbf{w}_p$, $\mathbf{q}^0$, and $\mathbf{v}^t$ satisfy all the conditions in Theorem 7, and that $\mathbf{A}$ consists of i.i.d. $\mathcal{N}(0, 1/n)$ entries, the following holds:*

(a)

$$\begin{aligned}
&\mathbf{h}_p^{t+1}|\mathfrak{S}_{t+1,t}^p \overset{d}{=} \mathbf{H}_p^t\boldsymbol{\alpha}_p^t + \widetilde{\mathbf{A}}_p^T\mathbf{m}_{p,\perp}^t + \widetilde{\mathbf{Q}}^{t+1}\overrightarrow{o}_{t+1}(1)\\
&= \sum_{i=0}^{t-1}\alpha_p^i\mathbf{h}_p^{i+1} + \widetilde{\mathbf{A}}_p^T\mathbf{m}_{p,\perp}^t + \widetilde{\mathbf{Q}}^{t+1}\overrightarrow{o}_{t+1}(1),\\
&\mathbf{b}_p^t|\mathfrak{S}_{t+1,t}^p \overset{d}{=} \mathbf{B}_p^t\boldsymbol{\beta}^t + \widetilde{\mathbf{A}}_p\mathbf{q}_\perp^t + \widetilde{\mathbf{M}}_p^t\overrightarrow{o}_t(1)\\
&= \sum_{i=0}^{t-1}\beta_i\mathbf{b}_p^i + \widetilde{\mathbf{A}}_p\mathbf{q}_\perp^t + \widetilde{\mathbf{M}}_p^t\overrightarrow{o}_t(1),
\end{aligned} \tag{3.49}$$

(b) for any $\phi_h, \phi_b \in PL_k$,

$$\begin{aligned}
&\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}\phi_h\left(h_{1,i}^1,\cdots,h_{P,i}^1,h_{1,i}^2,\cdots,h_{P,i}^2,\cdots,\right.\\
&\left.h_{1,i}^{t+1},\cdots,h_{P,i}^{t+1},v_i^1,\cdots,v_i^t,s_{0,i}\right)\\
&\overset{a.s.}{=} \mathbb{E}\left\{\phi_h\left(\tau_0\sqrt{\omega_1}Z_1^0,\cdots,\tau_0\sqrt{\omega_P}Z_P^0,\cdots,\right.\right.\\
&\left.\left.\tau_t\sqrt{\omega_1}Z_1^t,\cdots,\tau_t\sqrt{\omega_P}Z_P^t,V^0,\cdots,V^t,S_0\right)\right\},
\end{aligned} \tag{3.50}$$

where Z_p^t *are mutually independent for different* p.

$$\begin{aligned}
&\lim_{n\to\infty}\frac{1}{n_p}\sum_{i=1}^{n_p}\phi_b\left(b_{p,i}^0,b_{p,i}^1,\cdots,b_{p,i}^t,w_{p,i}\right)\\
&\overset{a.s.}{=} \mathbb{E}\left\{\phi_b\left(\sigma_0\widehat{Z}_p^0,\cdots,\sigma_t\widehat{Z}_p^t,W_p\right)\right\},
\end{aligned} \tag{3.51}$$

where $\widehat{Z}_p^t$ *are mutually independent for different* p.

(c) $\forall 0 \le r, s \le t$,

$$\lim_{N\to\infty}\left\langle\mathbf{h}_p^{r+1},\mathbf{h}_p^{s+1}\right\rangle \overset{a.s.}{=} \omega_p\lim_{n\to\infty}\left\langle\mathbf{m}_p^r,\mathbf{m}_p^s\right\rangle, \tag{3.52}$$

$$\lim_{n\to\infty}\left\langle\mathbf{b}_p^r,\mathbf{b}_p^s\right\rangle \overset{a.s.}{=} \frac{1}{\kappa}\lim_{n\to\infty}\left\langle\mathbf{q}^r,\mathbf{q}^s\right\rangle. \tag{3.53}$$

(d) $\forall 0 \leq r, s \leq t$, $\forall \phi \in PL_k$,

$$\begin{aligned} &\lim_{N\to\infty} \left\langle \mathbf{h}_p^{r+1}, \phi\left(\sum_{q=1}^{P} \mathbf{h}_q^{s+1}, \mathbf{v}^s, \mathbf{s}_0\right)\right\rangle \overset{a.s.}{=} \\ &\lim_{N\to\infty} \left\langle \mathbf{h}_p^{r+1}, \mathbf{h}_p^{s+1}\right\rangle \left\langle \phi'\left(\sum_{q=1}^{P} \mathbf{h}_q^{s+1}, \mathbf{v}^s, \mathbf{s}_0\right)\right\rangle, \end{aligned} \tag{3.54}$$

$$\lim_{n\to\infty} \left\langle \mathbf{b}_p^r, \phi(\mathbf{b}_p^s, \mathbf{w}_p)\right\rangle \overset{a.s.}{=} \lim_{N\to\infty} \left\langle \mathbf{b}_p^r, \mathbf{b}_p^s\right\rangle \left\langle \phi'(\mathbf{b}_p^s, \mathbf{w}_p)\right\rangle. \tag{3.55}$$

(e) for $\ell = k-1$,

$$\limsup_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} (\left[\mathbf{h}_p^{t+1}\right]_i)^{2\ell} < \infty, \tag{3.56}$$

$$\limsup_{n\to\infty} \frac{1}{n_p} \sum_{i=1}^{n_p} (\left[\mathbf{b}_p^{t}\right]_i)^{2\ell} < \infty. \tag{3.57}$$

(f) $0 \leq r \leq t$,

$$\lim_{N\to\infty} \left\langle \mathbf{h}_p^{r+1}, \mathbf{q}^0\right\rangle \overset{a.s.}{=} 0. \tag{3.58}$$

(g) $0 \leq r, s \leq t$,

$$\lim_{N\to\infty} \left\langle \mathbf{h}_p^{r+1}, \mathbf{h}_q^{s+1}\right\rangle \overset{a.s.}{=} 0, \forall p \neq q. \tag{3.59}$$

(h) $\forall 0 \leq r \leq t, 0 \leq s \leq t-1$, $\exists \rho_r, \zeta_s > 0$, *s.t.*

$$\lim_{N\to\infty} \langle \mathbf{q}_\perp^r, \mathbf{q}_\perp^r\rangle > \rho_r, \tag{3.60}$$

$$\lim_{n\to\infty} \left\langle \mathbf{m}_{p,\perp}^s, \mathbf{m}_{p,\perp}^s\right\rangle > \zeta_s. \tag{3.61}$$

It can be shown that under the condition that $\mathbf{A}$ consists of i.i.d. $\mathcal{N}(0, 1/n)$, Theorem 7 holds immediately once proving Lemma 4, as it is a special case of Lemma 4 (b).

In Lemma 4, denote $\mathcal{B}_t$ as all the conclusions regarding $\mathbf{b}_p^t$ and $\mathbf{m}_p^t$ conditioning on $\mathfrak{S}_{t,t}^p$, and denote $\mathcal{H}_{t+1}$ as all the conclusions regarding $\mathbf{h}_p^{t+1}$ and $\mathbf{q}^t$ conditioning on $\mathfrak{S}_{t+1,t}^p$. We apply induction to prove Lemma 4, and follow the same flow of proof as in [58]: $\mathcal{B}_0 \to \mathcal{H}_1 \to \cdots \to \mathcal{B}_t \to \mathcal{H}_{t+1}$.

3.2.1 Important Results In Literature

The following theoretical results in literature are important for proving Lemma 4.

Lemma 5 *(Lemma 2 in [58]) For deterministic vectors* $\mathbf{u} \in \mathbb{R}^N$ *and* $\mathbf{v}_p \in \mathbb{R}^{n_p}$ *and a random matrix* $\widetilde{\mathbf{A}}_p \in \mathbb{R}^{n_p \times N}$ *consisting of i.i.d.* $\mathcal{N}(0, 1/n)$ *entries, the following arguments hold:*

i) $\mathbf{v}_p^T \widetilde{\mathbf{A}}_p \mathbf{u} \overset{d}{=} Z\|\mathbf{u}\|\|\mathbf{v}_p\|/n$, *where* $Z \sim \mathcal{N}(0, 1)$;

ii) $\|\widetilde{\mathbf{A}}_p \mathbf{u}\|^2 \overset{a.s.}{=} n_p\|\mathbf{u}\|^2/n = \omega_p\|\mathbf{u}\|^2$;

iii) For a full-column rank matrix $\mathbf{D}_p \in \mathbb{R}^{n_p \times d}$ *satisfying* $\mathbf{D}_p^T\mathbf{D}_p = n\mathbf{I}_d$, *we have* $\mathbf{P}_{\mathbf{D}_p}\widetilde{\mathbf{A}}_p\mathbf{u} \overset{d}{=} \|\mathbf{u}\|\mathbf{D}_p\mathbf{x}$ *with* $\mathbf{x} \to \mathbf{0} \in \mathbb{R}^d$ *almost surely.*

Lemma 6 *(Lemma 10 in [58]) Let* $\mathbf{A}$ *be a matrix consisting of i.i.d. zero-mean Gaussian entries. Given* $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{M}$, $\mathbf{Q}$, *and the linear constraints* $\mathbf{AQ} = \mathbf{Y}$ *and* $\mathbf{A}^T\mathbf{M} = \mathbf{X}$, *the conditional distribution of* $\mathbf{A}$ *satisfies*

$$\begin{aligned}\mathbf{A}|\{\mathbf{X}, \mathbf{Y}, \mathbf{M}, \mathbf{Q}\} &= (\mathbf{X}\mathbf{M}^\dagger)^T + \mathbf{P}_{\mathbf{M}}^\perp\mathbf{Y}\mathbf{Q}^\dagger + \mathbf{P}_{\mathbf{M}}^\perp\widetilde{\mathbf{A}}\mathbf{P}_{\mathbf{Q}}^\perp \\ &= \mathbf{Y}\mathbf{Q}^\dagger + (\mathbf{X}\mathbf{M}^\dagger)^T\mathbf{P}_{\mathbf{Q}}^\perp + \mathbf{P}_{\mathbf{M}}^\perp\widetilde{\mathbf{A}}\mathbf{P}_{\mathbf{Q}}^\perp,\end{aligned}$$

where $\widetilde{\mathbf{A}}$ *is an independent copy of* $\mathbf{A}$, *and* $\mathbf{P}_{\mathbf{M}}^\perp$ *and* $\mathbf{P}_{\mathbf{Q}}^\perp$ *are the orthogonal projector onto the complimentary column spaces of* $\mathbf{M}$ *and* $\mathbf{Q}$ *respectively.*

Theorem 8 *(Theorem 2.1 in [78], strong low of large numbers) Let* $\{X_{n,i}\}$ *(*$i = 1, \cdots, n$ *and* $n = 1, 2, \cdots$*) be a triangular array of zero-mean random variables with* $X_{n,i}$ *being mutually independent for same* n *and different* i, *and let* $\phi(t)$ *be a positive even function such that* $\phi(t)/|t|^2$ *is increasing and* $\phi(t)/|t|^3$ *is decreasing on* $(0, \infty)$. *If* $\{X_{n,i}\}$ *satisfies*

$$\sum_{n=1}^{\infty}\sum_{i=1}^{n}\frac{\mathbb{E}\phi(X_{n,i})}{\phi(n)} < \infty, \tag{3.62}$$

and

$$\sum_{n=1}^{\infty}\left[\sum_{i=1}^{n}\frac{\mathbb{E}(X_{n,i}^2)}{n^2}\right]^{2p}<\infty \tag{3.63}$$

for some positive integer p, then

$$\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}X_{n,i}\overset{a.s.}{=}0. \tag{3.64}$$

Lemma 7 *(Theorem 3 in [58]) If the triangular array $\{X_{n,i}\}$ in Theorem 8 satisfies*

$$\sum_{i=1}^{n}\mathbb{E}|X_{n,i}|^{2+\rho}\leq cn^{\frac{2+\rho}{2}} \tag{3.65}$$

for any $\rho\in[0,1)$ and some $c>0$ independent of n, then the strong law of large numbers (SLLN) holds.

It is straightforward to show the correctness of Lemma 7 when noticing that (3.65) implies (3.62) and (3.63).

Lemma 8 *(Stein's Lemma, [79]) For two zero-mean and jointly Gaussian random variables X and Y, let $f:\mathbb{R}\to\mathbb{R}$ be any function such that $\mathbb{E}\left[f'(Y)\right]$ and $\mathbb{E}[Xf(Y)]$ exist, then*

$$\mathbb{E}[Xf(Y)]=Cov(X,Y)\mathbb{E}\left[f'(Y)\right]. \tag{3.66}$$

Lemma 9 *(Equation 1.4.14-18 in [80], fundamental equation of linear estimation) For two jointly Gaussian random vectors $\mathbf{x}\in\mathbb{R}^m$ and $\mathbf{y}\in\mathbb{R}^n$, if the covariance matrix of their joint distribution is given by*

$$\mathbf{\Sigma}_{(\mathbf{x},\mathbf{y})}=\begin{bmatrix}\mathbf{P}_{xx} & \mathbf{P}_{xy}\\ \mathbf{P}_{yx} & \mathbf{P}_{yy}\end{bmatrix}, \tag{3.67}$$

where $\mathbf{P}_{xx}\in\mathbb{R}^{m\times m}$ and $\mathbf{P}_{yy}\in\mathbb{R}^{n\times n}$ are invertible, then the covariance matrix of the conditional distribution $p(\mathbf{x}|\mathbf{y})$ is

$$\mathbf{\Sigma}_{\mathbf{x}|\mathbf{y}}=\mathbf{P}_{xx}-\mathbf{P}_{xy}\mathbf{P}_{yy}^{-1}\mathbf{P}_{yx}. \tag{3.68}$$

Based on Lemma 9, we introduce the following definition.

Pseudo conditional auto-correlation: For two random vectors $\mathbf{x}$ and $\mathbf{y}$ with

$$\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right] = \mathbf{P}_{xx},\ \mathbb{E}\left[\mathbf{y}\mathbf{y}^T\right] = \mathbf{P}_{yy},\ \text{and}\ \mathbb{E}\left[\mathbf{x}\mathbf{y}^T\right] = \mathbf{P}_{xy},$$

we define the ***pseudo conditional auto-correlation*** (PCAC) of $\mathbf{x}$ given $\mathbf{y}$ as

$$\text{PCor}(\mathbf{x}|\mathbf{y}) = \mathbf{P}_{xx} - \mathbf{P}_{xy}\mathbf{P}_{yy}^{-1}\mathbf{P}_{yx}, \tag{3.69}$$

similarly, the PCAC of $\mathbf{y}$ given $\mathbf{x}$ is defined as

$$\text{PCor}(\mathbf{y}|\mathbf{x}) = \mathbf{P}_{yy} - \mathbf{P}_{yx}\mathbf{P}_{xx}^{-1}\mathbf{P}_{xy}. \tag{3.70}$$

Lemma 10 *(Lemma 7 in [58]) For a sequence of correlated $\mathcal{N}(0,1)$ random variables Z_1, Z_2, $\cdots$, Z_t, suppose that $Var(Z_i|Z_1,\cdots,Z_{i-1}) \geq c > 0$ for any $i = 1,\cdots,t$. Let Y be another random variable independent of Z_1, Z_2, $\cdots$, Z_t, and let $X_i = f(Z_i, Y)$, $i = 1,\cdots,t$, where $f : \mathbb{R}^2 \to \mathbb{R}$ is a Lipschitz continuous function, and the probability of $f(Z,Y)$ being non-constant with respect to Y is greater than 0, then there exists $c_t > 0$ independent of Z_1, Z_2, $\cdots$, Z_t such that*

$$PCor(X_t|X_1,\cdots,X_{t-1}) > c_t. \tag{3.71}$$

Proposition 2 *For any $\mathbf{x} \in \mathbb{R}^n$ and $q > p > 0$,*

$$\|\mathbf{x}\|_q \leq \|\mathbf{x}\|_p \leq n^{1/p-1/q}\|\mathbf{x}\|_q.$$

3.2.2 Induction to Prove Lemma 4

Denote $\mathcal{B}_t$ and $\mathcal{H}_{t+1}$ as all the conclusions in Lemma 4 given $\mathfrak{S}_{t,t}^p$ and $\mathfrak{S}_{t+1,t}^p$ respectively. The proof is done by induction, which has the same flow as the proof

of Lemma 1 in [58]. First, we prove the base cases $\mathcal{B}_0$ and $\mathcal{H}_1$. Then, we prove that $\mathcal{B}_t$ holds based on the induction hypothesis $\mathcal{B}_0, \cdots, \mathcal{B}_{t-1}, \mathcal{H}_1, \cdots$, and $\mathcal{H}_t$, and prove that $\mathcal{H}_{t+1}$ holds based on the induction hypothesis $\mathcal{B}_0, \cdots, \mathcal{B}_{t-1}, \mathcal{B}_t, \mathcal{H}_1, \cdots$, and $\mathcal{H}_t$. The proof literally is presented in the following four steps: $\mathcal{B}_0$, $\mathcal{H}_1$, $\mathcal{B}_t$, and $\mathcal{H}_{t+1}$.

Step I: $\mathcal{B}_0$

We prove $\mathcal{B}_0$ in the order of (a), (e), (c), (b), (d) and (h).

$\mathcal{B}_0$ a) Trivial since $\mathfrak{S}_{0,0}^p = \{\mathbf{q}^0, \mathbf{s}_0, \mathbf{w}_p\}$, according to (3.27), we have

$$\mathbf{b}_p^1 = \mathbf{A}_p \mathbf{q}^0. \tag{3.72}$$

$\mathcal{B}_0$ (e) Conditioning on $\mathfrak{S}_{0,0}^p$,

$$\begin{aligned} \frac{1}{N}\sum_{i=1}^{n_p}(\left[\mathbf{b}_p^0\right]_i)^{2\ell} &= \frac{1}{n_p}\sum_{i=1}^{n_p}\left\{\left[\mathbf{A}_p\mathbf{q}^0\right]_i^{2\ell}\right\} \\ &\overset{\text{a.s.}}{=} \frac{1}{n_p}\sum_{i=1}^{n_p}\left\{\left[\frac{Z_i\|\mathbf{q}^0\|}{\sqrt{N}}\right]^{2\ell}\right\} \\ &= \frac{1}{n_p}\sum_{i=1}^{n_p}\left\{Z_i^{2\ell}\left[\left\langle\mathbf{q}^0,\mathbf{q}^0\right\rangle\right]^{\ell}\right\} < \infty. \end{aligned} \tag{3.73}$$

$\mathcal{B}_0$ (c) Applying Lemma 5,

$$\left\langle\mathbf{b}_p^0,\mathbf{b}_p^0\right\rangle = \frac{\|\mathbf{b}_p^0\|^2}{n_p} = \frac{\|\mathbf{A}_p\mathbf{q}^0\|^2}{n_p} \overset{\text{a.s.}}{\to} \frac{\omega_p\|\mathbf{q}^0\|^2}{n_p} = \frac{\langle\mathbf{q}^0,\mathbf{q}^0\rangle}{\kappa}.$$

$\mathcal{B}_0$ (b) Define

$$X_{n_p,i} = \phi_b\left(b_{p,i}^0,\star\right) - \mathbb{E}_{\mathbf{A}_p}\left\{\phi_h\left(\left[\mathbf{A}_p\mathbf{q}^0\right]_i,\star\right)\right\},$$

where we use $\star$ to replace $w_{p,i}$.

We want to show that

$$\sum_{i=1}^{n_p} \mathbb{E}_{\mathbf{A}_p} |X_{n_p,i}|^{2+\rho} \leq c(n_p)^{\frac{2+\rho}{2}}$$

for any $\rho \in [0,)$ some c independent of $\mathbf{A}_p$, so that Lemma 7 can be applied to show that

$$\lim_{n\to\infty} \frac{1}{n_p} \sum_{i=1}^{n_p} X_{n_p,i} \overset{\text{a.s.}}{=} 0.$$

Using Lemma 5, we have

$$b_{p,i}^0 = \left[\mathbf{A}_p \mathbf{q}^0\right]_i \sim \text{i.i.d. } \frac{\|\mathbf{q}^0\|}{\sqrt{N}} Z_i, Z_i \sim \mathcal{N}(0,1),$$

and

$$\mathbb{E}_{\mathbf{A}^p} \left\{ \phi_b \left(\left[\mathbf{A}_p \mathbf{q}^0\right]_i, w_{p,i} \right) \right\} = \mathbb{E}_Z \left\{ \phi_b \left(\frac{Z_i}{\sqrt{N}} \|\mathbf{q}^0\|, w_{p,i} \right) \right\}.$$

Since $\phi_b(\cdot,\cdot)$ is pseudo Lipschitz continuous of order k, we have

$$\begin{aligned}
\mathbb{E}_{\mathbf{A}_p} |X_{n_p,i}|^{2+\rho} &= \mathbb{E}_{\widetilde{Z}_i} \left\{ \left| \phi_b \left(\frac{\widetilde{Z}_i}{\sqrt{N}} \|\mathbf{q}^0\|, \star \right) - \mathbb{E}_{Z_i} \phi_b \left(\frac{Z_i}{\sqrt{N}} \|\mathbf{q}^0\|, \star \right) \right|^{2+\rho} \right\} \\
&= \mathbb{E}_{\widetilde{Z}_i} \left| \int_{-\infty}^{+\infty} \left[\phi_b \left(\frac{\widetilde{z}_i \|\mathbf{q}^0\|}{\sqrt{N}}, \star \right) - \phi_b \left(\frac{z_i \|\mathbf{q}^0\|}{\sqrt{N}}, \star \right) \right] \frac{e^{-\frac{1}{2} z_i^2}}{\sqrt{2\pi}} dz_i \right|^{2+\rho} \\
&\leq L \mathbb{E}_{\widetilde{Z}_i} \left\{ \int_{-\infty}^{+\infty} |\widetilde{z}_i - z_i| \left[1 + \left(\frac{\|\mathbf{q}^0\|^2}{N} \widetilde{z}_i^2 + w_{p,i}^2 \right)^{\frac{k-1}{2}} \right. \right. \\
&\left. \left. + \left(\frac{\|\mathbf{q}^0\|^2}{N} z_i^2 + w_{p,i}^2 \right)^{\frac{k-1}{2}} \right] \frac{e^{-\frac{1}{2} z_i^2}}{\sqrt{2\pi}} dz_i \right\}^{2+\rho}.
\end{aligned}$$

Applying Proposition 2 for $\mathbf{x} = \left(\|\mathbf{q}^0\|^2 \widetilde{z}_i^2 / N, w_{p,i}^2 \right)$ or $\left(\|\mathbf{q}^0\|^2 z_i^2 / N, w_{p,i}^2 \right)$, $p = 1$

and $q = (k-1)/2$, we have

$$\begin{aligned}
&\mathbb{E}_{\mathbf{A}_p}|X_{n_p,i}|^{2+\rho} \\
&\leq L\mathbb{E}_{\widetilde{Z}_i}\Bigg\{\int_{-\infty}^{+\infty}|\widetilde{z}_i - z_i|\Big[1+ 2^{\frac{k-3}{2}}\Big(\left\langle \mathbf{q}^0, \mathbf{q}^0\right\rangle^{\frac{k-1}{2}}\left|\widetilde{z}_i^{k-1}\right| + \left|w_{p,i}^{k-1}\right|\Big) \\
&+2^{\frac{k-3}{2}}\Big(\left\langle \mathbf{q}^0, \mathbf{q}^0\right\rangle^{\frac{k-1}{2}}\left|z_i^{k-1}\right| + \left|w_{p,i}^{k-1}\right|\Big)\Big]\frac{e^{-\frac{1}{2}z_i^2}}{\sqrt{2\pi}}dz_i\Bigg\}^{2+\rho}.
\end{aligned}$$

After integration, the above inequality has the following form:

$$\begin{aligned}
&\mathbb{E}_{\mathbf{A}_p}|X_{n_p,i}|^{2+\rho} \leq \\
&\mathbb{E}_{\widetilde{Z}_i}\left\{\begin{bmatrix}1 & \left\langle \widetilde{\mathbf{q}}^0, \widetilde{\mathbf{q}}^0\right\rangle^{\frac{k-1}{2}} & |w_{p,i}|^{k-1}\end{bmatrix}\mathbf{C}_1\begin{bmatrix}1\\ |\widetilde{z}_i|\\ |\widetilde{z}_i|^{k-1}\\ |\widetilde{z}_i|^{k}\end{bmatrix}\right\}^{2+\rho}.
\end{aligned}$$

Applying Proposition 2 again for $p = 1$ and $q = 2 + \rho$, we can move the power $2 + \rho$ inside, i.e.,

$$\begin{aligned}
&\mathbb{E}_{\mathbf{A}_p}|X_{n_p,i}|^{2+\rho} \leq \\
&\mathbb{E}_{\widetilde{Z}_i}\left\{\left[1, \left\langle \mathbf{q}^0, \mathbf{q}^0\right\rangle^{\frac{(k-1)(2+\rho)}{2}}, |w_{p,i}|^{(k-1)(2+\rho)}\right]\mathbf{C}_2\begin{bmatrix}1\\ |\widetilde{z}_i|^{2+\rho}\\ |\widetilde{z}_i|^{(k-1)(2+\rho)}\\ |\widetilde{z}_i|^{k(2+\rho)}\end{bmatrix}\right\} \\
&= c_0 + c_1\left\langle \mathbf{q}^0, \mathbf{q}^0\right\rangle^{\frac{(k-1)(2+\rho)}{2}} + c_2|w_{p,i}|^{(k-1)(2+\rho)}.
\end{aligned}$$

Since

$$\sum_{i=1}^{n_p}|w_{p,i}|^{(k-1)(2+\rho)} = \sum_{i=1}^{n_p}\left\{|w_{p,i}|^{2(k-1)}\right\}^{\frac{2+\rho}{2}} \leq \left\{\sum_{i=1}^{n_p}|w_{p,i}|^{2(k-1)}\right\}^{\frac{2+\rho}{2}}, \tag{3.74}$$

we have

$$\sum_{i=1}^{n_p} \mathbb{E}_{\mathbf{A}_p} |X_{n_p,i}|^{2+\rho} \leq \left[c_0 + c_1 \left(\left\langle \mathbf{q}^0, \mathbf{q}^0 \right\rangle^{k-1} \right)^{\frac{2+\rho}{2}} \right] n_p + c_2 \left(\frac{\sum_{i=1}^{n_p} |w_{p,i}|^{2(k-1)}}{n_p} \right)^{\frac{2+\rho}{2}} (n_p)^{\frac{2+\rho}{2}} \tag{3.75}$$

By assumption of the empirical distribution of $\mathbf{q}^0$ and $\mathbf{w}_p$, we know that

$$\left(\left\langle \mathbf{q}^0, \mathbf{q}^0 \right\rangle^{k-1} \right)^{\frac{2+\rho}{2}} < \infty, \tag{3.76}$$

and

$$\left(\frac{\sum_{i=1}^{n_p} |w_{p,i}|^{2(k-1)}}{n_p} \right)^{\frac{2+\rho}{2}} \to \left(\mathbb{E} W^{2k-2} \right)^{\frac{2+\rho}{2}} < \infty, \tag{3.77}$$

therefore

$$\sum_{i=1}^{n_p} \mathbb{E}_{\mathbf{A}_p} |X_{n_p,i}|^{2+\rho} \leq c (n_p)^{\frac{2+\rho}{2}}$$

for any $\rho \in [0, 1)$ and some $c > 0$ independent of $\mathbf{A}_p$.

Applying Lemma 7, we have

$$\begin{aligned} &\lim_{n_p \to \infty} \frac{1}{n_p} \sum_{i=1}^{n_p} \phi_b \left(\left[\mathbf{b}_p^0 \right]_i, w_{p,i} \right) \\ &\overset{\text{a.s.}}{=} \lim_{n_p \to \infty} \frac{1}{n_p} \sum_{i=1}^{n_p} \mathbb{E}_Z \phi_b \left(\sqrt{\langle \mathbf{q}^0, \mathbf{q}^0 \rangle} Z, w_{p,i} \right). \end{aligned} \tag{3.78}$$

Let

$$\psi(w_{p,i}) = \mathbb{E}_Z \phi_b \left(\sqrt{\langle \mathbf{q}^0, \mathbf{q}^0 \rangle} Z, w_{p,i} \right).$$

Applying Lemma 4 in [58], we have

$$\lim_{n_p\to\infty}\frac{1}{n_p}\sum_{i=1}^{n_p}\psi(w_{p,i})\overset{\text{a.s.}}{=}\mathbb{E}_W\{\psi(W)\},\tag{3.79}$$

i.e.,

$$\begin{aligned}\lim_{n_p\to\infty}\frac{1}{n_p}\sum_{i=1}^{n_p}\phi_b\left(b_{p,i}^0,w_{p,i}\right)&\overset{\text{a.s.}}{=}\lim_{n_p\to\infty}\frac{1}{n_p}\sum_{i=1}^{n_p}\mathbb{E}_Z\phi_b\left(\sqrt{\langle\mathbf{q}^0,\mathbf{q}^0\rangle}\widehat{Z}_p^0,w_{p,i}\right)\\&\overset{\text{a.s.}}{=}\mathbb{E}_W\mathbb{E}_Z\phi_b\left(\tau_0\widehat{Z}_p^0,W\right).\end{aligned}\tag{3.80}$$

Note that $\widehat{Z}_p^0$ are mutually independent for different p, since the corresponding $\mathbf{A}_p$ are mutually independent.

Now we can show the following result, which is one intuitive indication of the Gaussianity of $\mathbf{b}_p^t$ and $\mathbf{h}_p^{t+1}$.

Corollary 1 *The empirical distributions of* $\mathbf{b}_p^t$ *and* $\mathbf{h}_p^{t+1}$*, defined as*

$$\widehat{F}_b^t(x)=\frac{1}{n_p}\sum_{i=1}^{n_p}\mathbb{I}_{(-\infty,x]}\left(b_{p,i}^t\right)$$

and

$$\widehat{F}_h^{t+1}(x)=\frac{1}{N}\sum_{i=1}^{N}\mathbb{I}_{(-\infty,x]}\left(h_{p,i}^{t+1}\right)$$

respectively, converges almost surely to

$$\Phi_b^t(x)=\mathbb{P}(\sigma_t Z\le x)$$

and

$$\Phi_h^{t+1}(x)=\mathbb{P}(\sqrt{\omega_p}\tau_t Z\le x)$$

as $N\to\infty$ and $n/N\to\kappa>0$.

Proof: Considering the base case $\mathcal{B}_0$, for each given x, $\mathbb{I}_{(-\infty,x]}\left(b_{p,i}^0\right)$ is not a continuous function of $b_{p,i}^0$. However, we can construct the following Lipschitz continuous function

series

$$\phi_x^k\left(b_{p,i}^0\right) = \begin{cases} 1 & \text{if } b_{p,i}^0 < x - \frac{1}{2k}, \\ \frac{1}{2} - k(b_{p,i}^0 - x) & \text{if } \left|b_{p,i}^0 - x\right| \leq \frac{1}{2k}, \\ 0 & \text{if } b_{p,i}^0 > x + \frac{1}{2k}. \end{cases} \tag{3.81}$$

It can be shown that $\lim_{k\to\infty} \phi_x^k\left(b_{p,i}^0\right) = \mathbb{I}_{(-\infty,x]}\left(b_{p,i}^0\right)$.

Now we apply $\mathcal{B}_0$ (b) for $\phi_x^k\left(b_{p,i}^0\right)$:

$$\lim_{n_p\to\infty} \frac{1}{n_p} \sum_{i=1}^{n_p} \phi_x^k\left(b_{p,i}^0\right) \overset{\text{a.s.}}{=} \mathbb{E}\phi_x^k\left(\tau_0 \widehat{Z}_p^0\right). \tag{3.82}$$

Let $k \to \infty$, the left hand side becomes

$$\frac{1}{n_p} \sum_{i=1}^{n_p} \mathbb{I}_{(-\infty,x]}\left(b_{p,i}^0\right) = \widehat{F}_b^0(x),$$

while the right hand side becomes

$$\mathbb{E}\left\{\mathbb{I}_{(-\infty,x]}\left(\sigma_0 Z\right)\right\} = \mathbb{P}(\sigma_0 Z \leq x) = \Phi_b^0(x).$$

Therefore, $\widehat{F}_b^0(x) \overset{\text{a.s.}}{\to} \Phi_b^0(x)$.

Similarly, on proving $\mathcal{B}_t$ and $\mathcal{H}_{t+1}$, we can apply the similar technique to obtain

$$\widehat{F}_b^t(x) \overset{\text{a.s.}}{\to} \Phi_b^t(x)$$

and

$$\widehat{F}_h^{t+1}(x) \overset{\text{a.s.}}{\to} \Phi_h^{t+1}(x).$$

$\mathcal{B}_0$ (d) Using $\mathcal{B}_0$ (b) for $\phi_b(b_{p,i}^0) = b_{p,i}^0 \phi(b_{p,i}^0, w_{p,i})$ and applying Lemma 8,

$$\lim_{n\to\infty} \left\langle \mathbf{b}_p^0, \mathbf{w}_p \right\rangle \overset{\text{a.s.}}{=} \mathbb{E}\{\sigma_0 \widehat{Z}_p \phi(\sigma_0 \widehat{Z}_p, W)\} = \sigma_0^2 \mathbb{E}\{\phi'(\sigma_0 \widehat{Z}_p, W)\}. \tag{3.83}$$

Note: $x\phi(x,\cdot) \in \mathrm{PL}(k)$.

By Corollary 1, empirical distribution of $(\mathbf{b}_p^0, \mathbf{w}_p)$ converges weakly to $(\sigma_0 \widehat{Z}, W)$,

applying Lemma 5 in [58], we have

$$\lim_{n\to\infty} \left\langle \boldsymbol{\phi}'(\mathbf{b}_p^0, \mathbf{w}_p) \right\rangle \overset{\text{a.s.}}{=} \mathbb{E}\{\phi'(\sigma_0 \widehat{Z}_p, W)\}. \tag{3.84}$$

Therefore,

$$\left\langle \mathbf{b}_p^0, \boldsymbol{\phi}(\mathbf{b}_p^0, \mathbf{w}_p) \right\rangle \overset{\text{a.s.}}{=} \tau_0^2 \left\langle \boldsymbol{\phi}'(\mathbf{b}_p^0, \mathbf{w}_p) \right\rangle \overset{\text{a.s.}}{=} \left\langle \mathbf{b}_p^0, \mathbf{b}_p^0 \right\rangle \left\langle \boldsymbol{\phi}'(\mathbf{b}_p^0, \mathbf{w}_p) \right\rangle.$$

$\mathcal{B}_0$ (h) Trivial since $\mathbf{m}_p^0 = \mathbf{m}_{p,\perp}^0$, $\lim_{n\to\infty} \left\langle \mathbf{m}_{p,\perp}^0, \mathbf{m}_{p,\perp}^0 \right\rangle \overset{\text{a.s.}}{=} \tau_0^2 > 0$.

Step II: $\mathcal{H}_1$

Before proving $\mathcal{H}_1$, we first show the following useful proposition.

Proposition 3 *Define $\xi_p^t = \left\langle \boldsymbol{g}_t'(\mathbf{b}_p^t, \mathbf{w}_p) \right\rangle$, then as $N \to \infty$, $n/N \to \kappa > 0$ with t, P and $\omega_1, \cdots, \omega_P$ fixed,*

$$\xi_p^t \overset{a.s.}{=} \xi_t, \ \forall p \in [P], \tag{3.85}$$

where ξ_t is defined in (3.13).

Proof: By definition, we know that

$$\xi_t = \sum_{p=1}^{P} \omega_p \xi_p^t. \tag{3.86}$$

For $t = 0$, apply $\mathcal{B}_0$ (b) for $\phi_b(x, w) = g_0'(x, w)$, we have

$$\xi_p^0 = \boldsymbol{g}_0'(\mathbf{b}_p^0, \mathbf{w}_p) \overset{\text{a.s.}}{=} \mathbb{E} g_0'(\sigma_0 Z, W), \ \forall p \in [P]. \tag{3.87}$$

Therefore,

$$\xi_0 = \sum_{p=1}^{P} \omega_p \xi_p^0 \overset{\text{a.s.}}{=} \mathbb{E} g_0'(\sigma_0 Z, W) \overset{\text{a.s.}}{=} \xi_p^0, \ \forall p \in [P]. \tag{3.88}$$

For $t > 0$, once proving $\mathcal{B}_t$ (b), we will have the same conclusion that $\xi_p^t \overset{\text{a.s.}}{=} \xi_t, \ \forall p \in [P]$.

Now we prove $\mathcal{H}_1$, in the order of (a), (c), (e), (f), (g), (b), (d), and (h).

$\mathcal{H}_1$ (a) Noting that $\mathfrak{S}_{1,0}^p = \{\mathbf{b}_p^0, \mathbf{m}_p^0, \mathbf{q}^0, \mathbf{s}^0, \mathbf{w}_p\}$, applying Lemma 6,

$$\mathbf{A}_p|\mathfrak{S}_{1,0}^p \overset{d}{=} \frac{\mathbf{b}_p^0(\mathbf{q}^0)^T}{\|\mathbf{q}^0\|_2^2} + \widetilde{\mathbf{A}}_p\mathbf{P}_{\mathbf{q}^0}^{\perp}, \tag{3.89}$$

$$\begin{aligned}\therefore \mathbf{h}_p^1|\mathfrak{S}_{1,0}^p &\overset{d}{=} \mathbf{P}_{\mathbf{q}^0}^{\perp}(\widetilde{\mathbf{A}}_p)^T\mathbf{m}_p^0 + \frac{\mathbf{q}^0(\mathbf{b}_p^0)^T\mathbf{m}_p^0}{\|\mathbf{q}^0\|_2^2} - \xi_0\omega_p\mathbf{q}^0 \\ &= \mathbf{P}_{\mathbf{q}^0}^{\perp}(\widetilde{\mathbf{A}}_p)^T\mathbf{m}_p^0 + \frac{n_p\left\langle\mathbf{b}_p^0, \mathbf{m}_p^0\right\rangle}{N\left\langle\mathbf{q}^0, \mathbf{q}^0\right\rangle}\mathbf{q}^0 - \xi_0\omega_p\mathbf{q}^0\end{aligned} \tag{3.90}$$

Apply $\mathcal{B}_0$ (d) for $\phi_b(x, w) = g_0(x, w)$, and then apply $\mathcal{B}_0$ (c) and Proposition 3,

$$\begin{aligned}\left\langle\mathbf{b}_p^0, \mathbf{m}_p^0\right\rangle &\overset{\text{a.s.}}{=} \left\langle\mathbf{b}_p^0, \mathbf{b}_p^0\right\rangle\left\langle\boldsymbol{g}_0'(\mathbf{b}_p^0, \mathbf{w}_p)\right\rangle \\ &\overset{\text{a.s.}}{=} \frac{1}{\kappa}\left\langle\mathbf{q}^0, \mathbf{q}^0\right\rangle\xi_p^0 \overset{\text{a.s.}}{=} \frac{1}{\kappa}\left\langle\mathbf{q}^0, \mathbf{q}^0\right\rangle\xi_0,\end{aligned} \tag{3.91}$$

i.e.,

$$\left\langle\mathbf{b}_p^0, \mathbf{m}_p^0\right\rangle = \frac{1}{\kappa}\left\langle\mathbf{q}^0, \mathbf{q}^0\right\rangle\xi_0 + \overrightarrow{o}_1(1). \tag{3.92}$$

Therefore,

$$\begin{aligned}\mathbf{h}_p^1|\mathfrak{S}_{1,0}^p &\overset{d}{=} \mathbf{P}_{\mathbf{q}^0}^{\perp}(\widetilde{\mathbf{A}}_p)^T\mathbf{m}_p^0 + \kappa\omega_p\frac{\frac{1}{\kappa}\left\langle\mathbf{q}^0, \mathbf{q}^0\right\rangle\xi_0 + \overrightarrow{o}_1(1)}{\left\langle\mathbf{q}^0, \mathbf{q}^0\right\rangle}\mathbf{q}^0 \\ -\xi_0\omega_p\mathbf{q}^0 &= \mathbf{P}_{\mathbf{q}^0}^{\perp}(\widetilde{\mathbf{A}}_p)^T\mathbf{m}_p^0 + \overrightarrow{o}_1(1)\mathbf{q}^0.\end{aligned} \tag{3.93}$$

Apply Lemma 5 iii),

$$\begin{aligned}\mathbf{P}_{\mathbf{q}^0}^{\perp}(\widetilde{\mathbf{A}}_p)^T\mathbf{m}_p^0 &= (\widetilde{\mathbf{A}}_p)^T\mathbf{m}_p^0 - \mathbf{P}_{\mathbf{q}^0}(\widetilde{\mathbf{A}}_p)^T\mathbf{m}_p^0 \\ &= (\widetilde{\mathbf{A}}_p)^T\mathbf{m}_p^0 + \overrightarrow{o}_1(1)\widetilde{\mathbf{q}}^0.\end{aligned} \tag{3.94}$$

Since $\|\mathbf{q}^0\|_2^2 \to N\kappa\sigma_0^2$ has the same order of $\|\widetilde{\mathbf{q}}^0\|_2^2 = N$, we finally show that

$$\mathbf{h}_p^1|\mathfrak{S}_{1,0}^p \overset{d}{=} \widetilde{\mathbf{A}}_p^T\mathbf{m}_p^0 + \overrightarrow{o}_1(1)\mathbf{q}^0 \overset{d}{=} \widetilde{\mathbf{A}}_p^T\mathbf{m}_p^0 + \overrightarrow{o}_1(1)\widetilde{\mathbf{q}}^0. \tag{3.95}$$

$\mathcal{H}_1$ (c) Apply $\mathcal{B}_0$ (b) for $\phi_b(x,w) = g_0(x,w)^2$,

$$\lim_{n\to\infty} \left\langle \mathbf{m}_p^0, \mathbf{m}_p^0 \right\rangle \overset{\text{a.s.}}{=} \mathbb{E}\left[g_0(\sigma_0 Z_p, W_p)\right] = \tau_0^2 < \infty, \tag{3.96}$$

then apply $\mathcal{H}_1$ (a) and Lemma 5 ii),

$$\begin{aligned} &\lim_{N\to\infty} \left\langle \mathbf{h}_p^1, \mathbf{h}_p^1 \right\rangle | \mathfrak{S}_{1,0}^p \overset{d}{=} \lim_{N\to\infty} \frac{\|(\widetilde{\mathbf{A}}_p)^T \mathbf{m}_p^0 + \overrightarrow{o}_1(1)\mathbf{q}^0\|^2}{N} \\ &= \lim_{N\to\infty} \frac{\|(\widetilde{\mathbf{A}}_p)^T \mathbf{m}_p^0\|^2}{N} \overset{\text{a.s.}}{=} \lim_{N\to\infty} \omega_p \left\langle \mathbf{m}_p^0, \mathbf{m}_p^0 \right\rangle = \omega_p \tau_0^2. \end{aligned} \tag{3.97}$$

$\mathcal{H}_1$ (e) Apply $\mathcal{H}_1$ (a), Proposition 2 and Lemma 5,

$$\begin{aligned} &\frac{1}{N}\sum_{i=1}^N (h_{p,i}^1)^{2\ell} | \mathfrak{S}_{1,0}^p \overset{d}{=} \frac{1}{N}\sum_{i=1}^N \left(\left[\widetilde{\mathbf{A}}_p^T \mathbf{m}_p^0 \right]_i + \overrightarrow{o}_1(1) q_i^0 \right)^{2\ell} \\ &\leq \frac{2^{2\ell}}{2} \frac{1}{N} \sum_{i=1}^N \left\{ \left[\widetilde{\mathbf{A}}_p^T \mathbf{m}_p^0 \right]_i^{2\ell} + \left[\overrightarrow{o}_1(1) q_i^0 \right]^{2\ell} \right\} \\ &\overset{\text{a.s.}}{=} \frac{2^{2\ell}}{2} \frac{1}{N} \sum_{i=1}^N \left\{ \left[\frac{Z_i \|\mathbf{m}_p^0\|}{\sqrt{n}} \right]^{2\ell} + \left[\overrightarrow{o}_1(1) q_i^0 \right]^{2\ell} \right\} \\ &= \frac{2^{2\ell}}{2} \frac{1}{N} \sum_{i=1}^N \left\{ Z_i^{2\ell} \left[\omega_p \left\langle \mathbf{m}_p^0, \mathbf{m}_p^0 \right\rangle \right]^\ell + \overrightarrow{o}_1(1) (q_i^0)^{2\ell} \right\}. \end{aligned}$$

Since $\sum_{i=1}^N Z_i^{2\ell}/N < \infty$, and according to (3.96) and the assumption of empirical distribution of $\mathbf{q}^0$, we know that

$$\frac{1}{N}\sum_{i=1}^N (h_{p,i}^1)^{2\ell} < \infty, \forall \ell \leq k. \tag{3.98}$$

$\mathcal{H}_1$ (f) Apply $\mathcal{H}_1$ (a) and Lemma 5,

$$\begin{aligned}
&\lim_{N\to\infty}\left\langle \mathbf{h}_p^1, \mathbf{q}^0\right\rangle \overset{d}{=} \lim_{N\to\infty}\left\langle \widetilde{\mathbf{A}}_p^T\mathbf{m}_p^0 + \overrightarrow{o}_1(1)\mathbf{q}^0, \mathbf{q}^0\right\rangle \\
&= \lim_{N\to\infty}\frac{(\mathbf{m}_p^0)^T\widetilde{\mathbf{A}}_p\mathbf{q}^0}{N} = \lim_{N\to\infty}\frac{Z\|\mathbf{m}_p^0\|\|\mathbf{q}^0\|}{N\sqrt{n}} \\
&= \lim_{N\to\infty}\frac{Z\sqrt{\omega_p\left\langle \mathbf{m}_p^0, \mathbf{m}_p^0\right\rangle\left\langle \mathbf{q}^0, \mathbf{q}^0\right\rangle}}{\sqrt{N}} \overset{\text{a.s.}}{=} 0.
\end{aligned} \tag{3.99}$$

$\mathcal{H}_1$ (g)

$$\begin{aligned}
&\lim_{N\to\infty}\left\langle \mathbf{h}_p^1, \mathbf{h}_\ell^1\right\rangle \overset{d}{=} \\
&\lim_{N\to\infty}\left\langle \widetilde{\mathbf{A}}_p^T\mathbf{m}_p^0 + \overrightarrow{o}_1(1)\mathbf{q}^0, \widetilde{\mathbf{A}}_\ell^T\mathbf{m}_\ell^0 + \overrightarrow{o}_1(1)\mathbf{q}^0\right\rangle \\
&\overset{d}{=} \lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}\frac{Z_{p,i}Z_{\ell,i}\|\mathbf{m}_p^0\|\|\mathbf{m}_\ell^0\|}{n} \\
&= \lim_{N\to\infty}\frac{\omega_p}{N}\sum_{i=1}^{N}Z_{p,i}Z_{\ell,i}\sqrt{\left\langle \mathbf{m}_p^0, \mathbf{m}_p^0\right\rangle\left\langle \mathbf{m}_\ell^0, \mathbf{m}_\ell^0\right\rangle} \overset{\text{a.s.}}{=} 0.
\end{aligned} \tag{3.100}$$

$\mathcal{H}_1$ (b) Apply $\mathcal{H}_1$ (a), we know that

$$\begin{aligned}
&\phi_h\left(h_{1,i}^1, \cdots, h_{P,i}^1, s_{0,i}\right)|\mathfrak{S}_{1,0} \overset{d}{=} \\
&\phi_h\left(\left[\widetilde{\mathbf{A}}_1^T\mathbf{m}_1^0\right]_i + \overrightarrow{o}_1(1)q_i^0, \cdots, \left[\widetilde{\mathbf{A}}_P^T\mathbf{m}_P^0\right]_i + \overrightarrow{o}_1(1)q_i^0, s_{0,i}\right).
\end{aligned}$$

First, we want to show that

$$\begin{aligned}
&\lim_{N\to\infty}\sum_{i=1}^{N}\phi_h\left(\left[\widetilde{\mathbf{A}}_1^T\mathbf{m}_1^0\right]_i + \overrightarrow{o}_1(1)q_i^0,\right. \\
&\left.\cdots, \left[\widetilde{\mathbf{A}}_P^T\mathbf{m}_P^0\right]_i + \overrightarrow{o}_1(1)q_i^0, s_{0,i}\right) \overset{\text{a.s.}}{=} \\
&\lim_{N\to\infty}\phi_h\left(\left[\widetilde{\mathbf{A}}_1^T\mathbf{m}_1^0\right]_i, \cdots, \left[\widetilde{\mathbf{A}}_P^T\mathbf{m}_P^0\right]_i, s_{0,i}\right).
\end{aligned} \tag{3.101}$$

Let

$$\begin{aligned}
\mathbf{a}_i = &\left(\left[\widetilde{\mathbf{A}}_1^T\mathbf{m}_1^0\right]_i + \overrightarrow{o}_1(1)q_i^0,\right. \\
&\left.\cdots, \left[\widetilde{\mathbf{A}}_P^T\mathbf{m}_P^0\right]_i + \overrightarrow{o}_1(1)q_i^0, s_{0,i}\right)
\end{aligned}$$

and

$$\mathbf{c}_i = \left(\left[\widetilde{\mathbf{A}}_1^T \mathbf{m}_1^0 \right]_i, \cdots, \left[\widetilde{\mathbf{A}}_P^T \mathbf{m}_P^0 \right]_i, s_{0,i} \right),$$

by the assumption of ϕ_h, we have

$$|\phi_h(\mathbf{a}_i) - \phi_h(\mathbf{c}_i)| \leq L \left\{ 1 + \|\mathbf{a}_i\|^{k-1} + \|\mathbf{c}_i\|^{k-1} \right\} |q_i^0| \overrightarrow{o}_1(1).$$

Let $\mathbf{p} = [p_1, \cdots, p_N]$, where $p_i = 1 + \|\mathbf{a}_i\|^{k-1} + \|\mathbf{c}_i\|^{k-1}$, according to Cauchy-Schwartz inequality,

$$\begin{aligned} \mathbf{p}^T \mathbf{q}^0 &\leq \|\mathbf{p}\| \|\mathbf{q}^0\| = \sqrt{\sum_{i=1}^N (1 + \|\mathbf{a}_i\|^{k-1} + \|\mathbf{c}_i\|^{k-1})^2} \|\mathbf{q}^0\| \\ &\leq \sqrt{3 \sum_{i=1}^N (1 + \|\mathbf{a}_i\|^{2k-2} + \|\mathbf{c}_i\|^{2k-2})} \|\mathbf{q}^0\|. \end{aligned}$$

Therefore,

$$\begin{aligned} \frac{1}{N} \sum_{i=1}^N |\phi_h(\mathbf{a}_i) - \phi_h(\mathbf{c}_i)| &\leq \frac{L}{N} \mathbf{p}^T \mathbf{q}^0 \overrightarrow{o}_1(1) \\ &\leq L \sqrt{3 \sum_{i=1}^N \left(\frac{1 + \|\mathbf{a}_i\|^{2k-2} + \|\mathbf{c}_i\|^{2k-2}}{N} \right)} \sqrt{\langle \mathbf{q}^0, \mathbf{q}^0 \rangle} \overrightarrow{o}_1(1). \end{aligned}$$

In order to show that $\sum_{i=1}^N |\phi_h(\mathbf{a}_i) - \phi_h(\mathbf{c}_i)|/N \overset{\text{a.s.}}{=} 0$, we need to show that

$$\frac{1}{N} \sum_{i=1}^N \|\mathbf{a}_i\|^{2k-2} < \infty, \text{ and } \frac{1}{N} \sum_{i=1}^N \|\mathbf{c}_i\|^{2k-2} < \infty. \tag{3.102}$$

For the former, applying Proposition 2 and Lemma 5,

$$\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{a}_i\|^{2k-2} =$$
$$\frac{1}{N}\sum_{i=1}^{N}\left\{\sum_{p=1}^{P}\left[\left(\widetilde{\mathbf{A}}_p^T\mathbf{m}_p^0\right)_i+\overrightarrow{o}_1(1)q_i^0\right]^2+|s_{0,i}|^2\right\}^{k-1}$$
$$\leq\frac{2^{k-1}}{2}\frac{1}{N}\sum_{i=1}^{N}\left\{\sum_{p=1}^{P}\left[\widetilde{\mathbf{A}}_p^T\mathbf{m}_p^0\right]_i^{2k-2}+|s_{0,i}|^{2k-2}\right\}=$$
$$\frac{2^{k-1}}{2}\frac{1}{N}\sum_{i=1}^{N}\left\{\sum_{p=1}^{P}(\omega_p\left\langle\mathbf{m}_p^0,\mathbf{m}_p^0\right\rangle Z_{p,i}^2)^{k-1}+|s_{0,i}|^{2k-2}\right\}<\infty.$$

Similarly,

$$\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{c}_i\|^{2k-2}=\frac{1}{N}\sum_{i=1}^{N}\left\{\sum_{p=1}^{P}\left[\widetilde{\mathbf{A}}_p^T\mathbf{m}_p^0\right]_i^2+|s_{0,i}|^2\right\}^{k-1}$$
$$\leq\frac{2^{k-1}}{2}\frac{1}{N}\sum_{i=1}^{N}\left\{\sum_{p=1}^{P}\left[\widetilde{\mathbf{A}}_p^T\mathbf{m}_p^0\right]_i^{2k-2}+|s_{0,i}|^{2k-2}\right\}<\infty.$$

Therefore,

$$\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}\phi_h\left(h_{1,i}^1,\cdots,h_{P,i}^1,s_{0,i}\right)\overset{\text{a.s.}}{=}$$
$$\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}\phi_h\left(\left[\widetilde{\mathbf{A}}_1^T\mathbf{m}_1^0\right]_i,\cdots,\left[\widetilde{\mathbf{A}}_P^T\mathbf{m}_P^0\right]_i,s_{0,i}\right).$$

On the other hand, following the similar technique in proof of $\mathcal{B}_0$ (b), we can obtain

$$\begin{aligned}&\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}\phi_h\left(\left[\widetilde{\mathbf{A}}_1^T\mathbf{m}_1^0\right]_i,\cdots,\left[\widetilde{\mathbf{A}}_P^T\mathbf{m}_P^0\right]_i,s_{0,i}\right)\\&\overset{\text{a.s.}}{=}\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_Z\phi_h\left(\sqrt{\omega_1\left\langle\mathbf{m}_1^0,\mathbf{m}_1^0\right\rangle}Z_1^0,\right.\\&\left.\cdots,\sqrt{\omega_P\left\langle\mathbf{m}_P^0,\mathbf{m}_P^0\right\rangle}Z_P^0,s_{0,i}\right).\end{aligned}\tag{3.103}$$

According to (3.96), we further have

$$\begin{aligned}
&\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}\phi_h\left(\left[\widetilde{\mathbf{A}}_1^T\mathbf{m}_1^0\right]_i,\cdots,\left[\widetilde{\mathbf{A}}_P^T\mathbf{m}_P^0\right]_i,s_{0,i}\right)\\
&\overset{\text{a.s.}}{=}\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_Z\phi_h\left(\sqrt{\omega_1}\tau_0 Z_1^0,\cdots,\sqrt{\omega_P}\tau_0 Z_P^0,s_{0,i}\right).
\end{aligned}$$

Let $\psi(s_{0,i})=\mathbb{E}_Z\phi_h\left(\sqrt{\omega_1}\tau_0 Z_1^0,\cdots,\sqrt{\omega_P}\tau_0 Z_P^0,s_{0,i}\right)$, by the assumption of empirical distribution of $\mathbf{s}_0$, we can apply Lemma 4 in [58] to obtain

$$\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}\psi(s_{0,i})\overset{\text{a.s.}}{=}\mathbb{E}_{S_0}\{\psi(S_0)\},\tag{3.104}$$

i.e.,

$$\begin{aligned}
&\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}\phi_h\left(h_{1,i}^1,\cdots,h_{P,i}^1,s_{0,i}\right)\\
&\overset{\text{a.s.}}{=}\mathbb{E}_{S_0}\mathbb{E}_Z\phi_h\left(\sqrt{\omega_1}\tau_0 Z_1^0,\cdots,\sqrt{\omega_P}\tau_0 Z_P^0,S_0\right).
\end{aligned}\tag{3.105}$$

$\mathcal{H}_1$ (d) Using $\mathcal{H}_1$ (b) for $\phi_h\left(h_{1,i}^1,\cdots,h_{P,i}^1,s_{0,i}\right)=h_{p,i}^1\phi\left(\sum_{q=1}^{P}h_{q,i}^1,s_{0,i}\right)$ and applying Lemma 8, we have

$$\begin{aligned}
&\lim_{N\to\infty}\left\langle\mathbf{h}_p^1,\phi\left(\sum_{q=1}^{P}\mathbf{h}_q^1,\mathbf{s}_0\right)\right\rangle\\
&\overset{\text{a.s.}}{=}\mathbb{E}\left\{\sqrt{\omega_p}\tau_0 Z_p\phi\left(\sum_{q=1}^{P}\sqrt{\omega_q}\tau_0 Z_q,S_0\right)\right\}\\
&=\omega_p\tau_0^2\mathbb{E}\left\{\phi'\left(\sum_{q=1}^{P}\sqrt{\omega_q}\tau_0 Z_q,S_0\right)\right\}.
\end{aligned}\tag{3.106}$$

On the other hand, let $\phi_h(h_{1,i}^1,\cdots,h_{P,i}^1,s_{0,i})=h_{p,i}^1$, we have $\phi'(h_{p,i}^1,s_{0,i})=1$ everywhere, so

$$\lim_{N\to\infty}\left\langle\mathbf{h}_p^1,\mathbf{h}_p^1\right\rangle\overset{\text{a.s.}}{=}\omega_p\tau_0^2.\tag{3.107}$$

Applying Corollary 1, we know that the empirical distribution of $(\mathbf{h}_p^1,\mathbf{s}_0)\to$

$(\sqrt{\omega_p}\tau_0 Z, S_0)$, according to Lemma 5 of [58],

$$\left\langle \boldsymbol{\phi}' \left(\sum_{q=1}^{P} \mathbf{h}_q^1, \mathbf{s}_0 \right) \right\rangle \overset{\text{a.s.}}{\to} \mathbb{E} \left\{ \phi' \left(\sum_{q=1}^{P} \sqrt{\omega_q}\tau_0 Z_q, S_0 \right) \right\}. \tag{3.108}$$

Now, by (3.106), (3.107), and (3.108), we have

$$\lim_{N\to\infty} \left\langle \mathbf{h}_p^1, \boldsymbol{\phi} \left(\sum_{q=1}^{P} \mathbf{h}_q^1, \mathbf{s}_0 \right) \right\rangle \overset{\text{a.s.}}{=} \left\langle \mathbf{h}_p^1, \mathbf{h}_p^1 \right\rangle \left\langle \boldsymbol{\phi}' \left(\sum_{q=1}^{P} \mathbf{h}_q^1, \mathbf{s}_0 \right) \right\rangle.$$

$\mathcal{H}_1$ (h) holds since for $t = 0$, $\mathbf{q}^0 = \mathbf{q}_\perp^0$, and $\lim_{N\to\infty} \langle \mathbf{q}_\perp^0, \mathbf{q}_\perp^0 \rangle = \kappa\sigma_0^2 > 0$.

Step III: $\mathcal{B}_t$

In this step we prove in the order of (h), (a), (e), (c), (b), and (d). $\mathcal{B}_t$ (h) Using $\mathcal{B}_{t-1}$ (b) for

$$\phi_b(b_{p,i}^r, b_{p,i}^s, w_{p,i}) = g_r(b_{p,i}^r, w_{p,i}) g_s(b_{p,i}^s, w_{p,i}), \tag{3.109}$$

where $0 \le r, s \le t-1$, we have

$$\begin{aligned} \lim_{n\to\infty} \left\langle \mathbf{m}_p^r, \mathbf{m}_p^s \right\rangle &= \lim_{n\to\infty} \frac{1}{n_p} \sum_{i=1}^{n_p} g_r(b_{p,i}^r, w_{p,i}) g_s(b_{p,i}^s, w_{p,i}) \\ &\overset{\text{a.s.}}{=} \mathbb{E}\{ g_r(\sigma_r \widehat{Z}_p^r, W) g_s(\sigma_s \widehat{Z}_p^s, W) \}. \end{aligned} \tag{3.110}$$

Then, by definition of $\mathbf{m}_{p,\perp}^{t-1}$,

$$\begin{aligned} &\left\langle \mathbf{m}_{p,\perp}^{t-1}, \mathbf{m}_{p,\perp}^{t-1} \right\rangle = \frac{1}{n_p} (\mathbf{m}_{p,\perp}^{t-1})^T \mathbf{m}_{p,\perp}^{t-1} \\ &= \frac{1}{n_p} \left\{ \left[\mathbf{I} - \mathbf{P}_{\mathbf{M}_p^{t-1}} \right] \mathbf{m}_p^{t-1} \right\}^T \left\{ \left[\mathbf{I} - \mathbf{P}_{\mathbf{M}_p^{t-1}} \right] \mathbf{m}_p^{t-1} \right\} \\ &= \frac{1}{n_p} \left\{ (\mathbf{m}_p^{t-1})^T \mathbf{m}_p^{t-1} - (\mathbf{m}_p^{t-1})^T \mathbf{M}_p^{t-1} (\mathbf{M}_p^{t-1})^\dagger \mathbf{m}_p^{t-1} \right\} \\ &= \left\langle \mathbf{m}_p^{t-1}, \mathbf{m}_p^{t-1} \right\rangle - \\ &\frac{(\mathbf{m}_p^{t-1})^T \mathbf{M}_p^{t-1}}{n_p} \left[\frac{(\mathbf{M}_p^{t-1})^T \mathbf{M}_p^{t-1}}{n_p} \right]^{-1} \frac{(\mathbf{M}_p^{t-1})^T \mathbf{m}_p^{t-1}}{n_p}. \end{aligned} \tag{3.111}$$

By induction, $\forall s < t-1$,

$$\lim_{n\to\infty} \left\langle \mathbf{m}_{p,\perp}^{s}, \mathbf{m}_{p,\perp}^{s} \right\rangle > \zeta_s > 0. \tag{3.112}$$

Using Lemma 8 in [58], we have

$$\lambda_{\min}\left[\frac{(\mathbf{M}_p^{t-1})^T \mathbf{M}_p^{t-1}}{n_p}\right] > c' \tag{3.113}$$

independent of n. Then, applying Lemma 9 in [58], we know that $(\mathbf{M}_p^{t-1})^T \mathbf{M}_p^{t-1}/n_p$ is invertible as $n \to \infty$, and

$$\begin{aligned} &\lim_{n\to\infty}\left[\frac{(\mathbf{M}_p^{t-1})^T \mathbf{M}_p^{t-1}}{n_p}\right]_{r,s\le t-2} = \lim_{n\to\infty}\left\langle \mathbf{m}_p^r, \mathbf{m}_p^s \right\rangle \\ &\overset{\text{a.s.}}{=} \mathbb{E}\{g_r(\sigma_r \widehat{Z}_p^r, W) g_s(\sigma_s \widehat{Z}_p^s, W)\} \triangleq \left[\mathbf{C}^{t-1}\right]_{r,s}, \end{aligned} \tag{3.114}$$

and

$$\begin{aligned} &\left[\frac{(\mathbf{M}_p^{t-1})^T \mathbf{m}_p^{t-1}}{n_p}\right]_{0\le r\le t-2} = (\mathbf{m}_p^r)^T \mathbf{m}_p^{t-1}/n_p \\ &\overset{\text{a.s.}}{=} \mathbb{E}\{g_r(\sigma_r \widehat{Z}_p^r, W) g_{t-1}(\sigma_{t-1} \widehat{Z}_p^{t-1}, W)\} \triangleq \left[\mathbf{u}^{t-1}\right]_r, \end{aligned} \tag{3.115}$$

therefore

$$\begin{aligned} &\left\langle \mathbf{m}_{p,\perp}^{t-1}, \mathbf{m}_{p,\perp}^{t-1} \right\rangle \\ &\overset{\text{a.s.}}{=} \mathbb{E}\left\{\left[g_{t-1}(\sigma_{t-1}\widehat{Z}_p^{t-1}, W)\right]^2\right\} - (\mathbf{u}^{t-1})^T (\mathbf{C}^{t-1})^{-1} \mathbf{u}^{t-1}. \end{aligned} \tag{3.116}$$

Define $\widehat{X}_p^r = g_r(\widehat{Z}_p^r, W)$, noting that the correlation matrix of $(\widehat{X}_p^0, \widehat{X}_p^1, \cdots, \widehat{X}_p^{t-1})$ is exactly

$$\mathbf{C}^t = \begin{bmatrix} \mathbf{C}^{t-1} & \mathbf{u}^{t-1} \\ (\mathbf{u}^{t-1})^T & \mathbb{E}\left\{\left[g_{t-1}(\sigma_{t-1}\widehat{Z}_p^{t-1}, W)\right]^2\right\} \end{bmatrix}. \tag{3.117}$$

By definition of pseudo conditional auto-correlation (PCAC), we know

$$\left\langle \mathbf{m}_{p,\perp}^{t-1}, \mathbf{m}_{p,\perp}^{t-1} \right\rangle \overset{\text{a.s.}}{\to} \text{PCor}\left(\widehat{X}_p^{t-1} | \widehat{X}_p^0, \cdots, \widehat{X}_p^{t-2} \right). \tag{3.118}$$

In order to prove $\mathcal{B}_t$ (h), we need to show that

$$\text{PCor}\left(\widehat{X}_p^{t-1} | \widehat{X}_p^0, \cdots, \widehat{X}_p^{t-2} \right) > \zeta_{t-1} > 0, \tag{3.119}$$

which is a straightforward conclusion of Lemma 10, if the following holds:

$$\text{Var}\left[\sigma_r \widehat{Z}_p^r | \sigma_0 \widehat{Z}_p^0, \cdots, \sigma_{r-1} \widehat{Z}_p^{r-1} \right] > 0, \forall r \leq t-1. \tag{3.120}$$

Define $\mathbf{\Sigma}^{r+1} \in \mathbb{R}^{(r+1)\times(r+1)}$ whose elements are given by

$$\Sigma_{i,j}^{r+1} = \sigma_i \sigma_j \mathbb{E}(\widehat{Z}_p^i \widehat{Z}_p^j), \forall 0 \leq i,j \leq r. \tag{3.121}$$

Using $\mathcal{B}_{t-1}$ (b) for $\phi_b(b_{p,\ell}^0, \cdots, b_{p,\ell}^{t-1}) = b_{p,\ell}^i b_{p,\ell}^j$, then

$$\left\langle \mathbf{b}_p^i, \mathbf{b}_p^j \right\rangle = \frac{1}{n_p} \sum_{\ell=1}^{n_p} \mathbf{b}_{p,\ell}^i \mathbf{b}_{p,\ell}^j \overset{\text{a.s.}}{\to} \mathbb{E}(\sigma_i \sigma_j \widehat{Z}_p^i \widehat{Z}_p^j) = \Sigma_{i,j}^{r+1},$$

i.e.,

$$\begin{aligned} \mathbf{\Sigma}^{r+1} &= \begin{bmatrix} \mathbf{\Sigma}^r & \mathbf{\Sigma}_{0:(r-1),r}^{r+1} \\ \mathbf{\Sigma}_{r,0:(r-1)} & \sigma_r^2 \end{bmatrix} \\ &\overset{\text{a.s.}}{=} \begin{bmatrix} \frac{(\mathbf{B}_p^r)^T \mathbf{B}_p^r}{n_p} & \left\langle \mathbf{b}_p^{0\leq s\leq r-1}, \mathbf{b}_p^r \right\rangle \\ \left\langle \mathbf{b}_p^r, \mathbf{b}_p^{0\leq s\leq r-1} \right\rangle & \left\langle \mathbf{b}_p^r, \mathbf{b}_p^r \right\rangle \end{bmatrix} = \frac{(\mathbf{B}_p^{r+1})^T \mathbf{B}_p^{r+1}}{n_p}. \end{aligned} \tag{3.122}$$

Using induction $\mathcal{B}_s$ (c) for any $0 \leq s \leq r \leq t-1$,

$$\left[\frac{(\mathbf{B}_p^{r+1})^T \mathbf{B}_p^{r+1}}{n_p} \right]_{0\leq i,j\leq r} = \left\langle \mathbf{b}_p^i, \mathbf{b}_p^j \right\rangle \overset{\text{a.s.}}{=} \frac{1}{\kappa} \left\langle \mathbf{q}^i, \mathbf{q}^j \right\rangle, \tag{3.123}$$

i.e.,

$$\frac{(\mathbf{B}_p^{r+1})^T\mathbf{B}_p^{r+1}}{n_p} \overset{\text{a.s.}}{=} \frac{1}{\kappa}\frac{(\mathbf{Q}^{r+1})^T\mathbf{Q}^{r+1}}{N}, \forall r \leq t-1. \tag{3.124}$$

Therefore,

$$\frac{1}{\kappa}\frac{(\mathbf{Q}^{r+1})^T\mathbf{Q}^{r+1}}{N} \overset{\text{a.s.}}{\to} \mathbf{\Sigma}^{r+1}. \tag{3.125}$$

Applying Lemma 9, we get

$$\begin{aligned}
&\text{Var}\left[\sigma_r\widehat{Z}_p^r|\sigma_0\widehat{Z}_p^0,\cdots,\sigma_{r-1}\widehat{Z}_p^{r-1}\right] \\
&= \sigma_r^2 - \mathbf{\Sigma}_{r,0\leq s\leq r-1}^{r+1}(\mathbf{\Sigma}^r)^{-1}\mathbf{\Sigma}_{0\leq s\leq r-1,r} \\
&\overset{\text{a.s.}}{=} \frac{\langle\mathbf{q}^r,\mathbf{q}^r\rangle}{\kappa} - \frac{(\mathbf{q}^r)^T\mathbf{Q}^r}{\kappa N}\left[\frac{(\mathbf{Q}^r)^T\mathbf{Q}^r}{N}\right]^{-1}\frac{(\mathbf{Q}^r)^T\mathbf{q}^r}{N} \\
&= \frac{1}{\kappa}\left\langle\mathbf{q}^r,\mathbf{q}^r\right\rangle.
\end{aligned} \tag{3.126}$$

By induction $\mathcal{H}_{r+1}$ (h), $\langle\mathbf{q}_\perp^s,\mathbf{q}_\perp^s\rangle > \rho_s > 0$, $\forall s \leq r \leq t-1$.

Therefore, we can apply Lemma 10 to show that there exists $\zeta_{t-1} > 0$, such that

$$\left\langle\mathbf{m}_{p,\perp}^{t-1},\mathbf{m}_{p,\perp}^{t-1}\right\rangle > \zeta_{t-1}. \tag{3.127}$$

Furthermore, applying Lemma 8 and 9 in [58], we know that $\exists c > 0$ such that $\lambda_{\min}((\mathbf{M}_p^t)^T\mathbf{M}_p^t/n_p) > c$ and $\lambda_{\min}(\mathbf{C}^t) \geq c$, i.e., $\lim_{n_p\to\infty}(\mathbf{M}_p^t)^T\mathbf{M}_p^t/n_p \overset{\text{a.s.}}{=} \mathbf{C}^t$ is invertible. Based on these findings we have the following corollaries, which are useful for proving (a) and (e) of $\mathcal{B}_t$ and $\mathcal{H}_{t+1}$.

Corollary 2

$$\begin{aligned}
\boldsymbol{\alpha}_p^t &= (\alpha_p^0,\alpha_p^1,\cdots,\alpha_p^{t-1}) \\
&= \left[\frac{(\mathbf{M}_p^t)^T\mathbf{M}_p^t}{n_p}\right]^{-1}\frac{(\mathbf{M}_p^t)^T\mathbf{m}_p^t}{n_p} < \infty,
\end{aligned}$$

$$\boldsymbol{\beta}^t = (\beta^0, \beta^1, \cdots, \beta^{t-1})$$
$$= \left[\frac{(\mathbf{Q}^t)^T\mathbf{Q}^t}{N}\right]^{-1} \frac{(\mathbf{Q}^t)^T\mathbf{q}^t}{N} < \infty.$$

Corollary 3

$$\mathbf{M}_p^t \overrightarrow{o}_t(1) = \widetilde{\mathbf{M}}_p^t \overrightarrow{o}_t(1), \ \textit{and}$$
$$\mathbf{Q}^t \overrightarrow{o}_t(1) = \widetilde{\mathbf{Q}}^t \overrightarrow{o}_t(1).$$

Proof: By definition,

$$\widetilde{\mathbf{M}}_p^t = \mathbf{M}_p^t \left[\frac{(\mathbf{M}_p^t)^T\mathbf{M}_p^t}{n_p}\right]^{-\frac{1}{2}},$$

we need to show that

$$\left[\frac{(\mathbf{M}_p^t)^T\mathbf{M}_p^t}{n_p}\right]^{\frac{1}{2}} \overrightarrow{o}_t(1) = \overrightarrow{o}_t(1), \tag{3.128}$$

and

$$\left[\frac{(\mathbf{M}_p^t)^T\mathbf{M}_p^t}{n_p}\right]^{-\frac{1}{2}} \overrightarrow{o}_t(1) = \overrightarrow{o}_t(1), \tag{3.129}$$

which are equivalent to

$$\left\| \left[\frac{(\mathbf{M}_p^t)^T\mathbf{M}_p^t}{n_p}\right]^{\frac{1}{2}} \overrightarrow{o}_t(1) \right\| = o(1), \tag{3.130}$$

and

$$\left\| \left[\frac{(\mathbf{M}_p^t)^T\mathbf{M}_p^t}{n_p}\right]^{-\frac{1}{2}} \overrightarrow{o}_t(1) \right\| = o(1). \tag{3.131}$$

For (3.130), since $(\mathbf{M}_p^t)^T\mathbf{M}_p^t/N$ converges to $\mathbf{C}^t$ with each element having finite limit, while according to Gershgorin circle theorem [81], the largest eigenvalue of $\mathbf{C}^t$ satisfies

$$\lambda_{\max}(\mathbf{C}^t) \leq \sup_{i\in[t]} \sum_{j=1}^{t} |C_{i,j}^t| < \infty, \tag{3.132}$$

on the other hand, since $\mathbf{C}^t$ is symmetric and positive definite, it is orthogonally

similar to a diagonal matrix $\mathbf{D}^t$, i.e.,

$$\mathbf{C}^t = \mathbf{U}\mathbf{D}^t\mathbf{U}^T, \tag{3.133}$$

where $\mathbf{U} \in \mathbb{R}^{t\times t}$ is orthogonal, and the diagonal elements of $\mathbf{D}^t$ are all positive. Therefore

$$(\mathbf{C}^t)^{-\frac{1}{2}} = \mathbf{U}(\mathbf{D}^t)^{-\frac{1}{2}}\mathbf{U}^T, \tag{3.134}$$

which implies that

$$\lambda_{\max}\left[(\mathbf{C}^t)^{-\frac{1}{2}}\right] = \sqrt{\lambda_{\max}(\mathbf{C}^t)} < \infty. \tag{3.135}$$

Hence

$$\left\| \left[\frac{(\mathbf{M}_p^t)^T\mathbf{M}_p^t}{n_p}\right]^{\frac{1}{2}} \overrightarrow{o}_t(1) \right\| \leq \sqrt{\lambda_{\max}(\mathbf{C}^t)}\|\overrightarrow{o}_t(1)\| = o(1). \tag{3.136}$$

For (3.131), since $\lambda_{\min}(\mathbf{C}^t) > c$ independent of n_p, we have $\lambda_{\max}\left[(\mathbf{C}^t)^{-\frac{1}{2}}\right] < 1/\sqrt{c}$. Therefore

$$\left\| \left[\frac{(\mathbf{M}_p^t)^T\mathbf{M}_p^t}{n_p}\right]^{-\frac{1}{2}} \overrightarrow{o}_t(1) \right\| \leq \frac{1}{\sqrt{c}}\|\overrightarrow{o}_t(1)\| = o(1). \tag{3.137}$$

The same proof also applies for $\mathbf{Q}^t$.

$\mathcal{B}_t$ (a) $\mathfrak{S}_{t,t}^p = \{\mathbf{B}_p^t, \mathbf{M}_p^t, \mathbf{H}_p^t, \mathbf{Q}^{t+1}, \mathbf{v}^1, \cdots, \mathbf{v}^t, \mathbf{s}_0, \mathbf{w}_p\}$.

Applying Lemma 6, we have

$$\mathbf{A}_p|\mathfrak{S}_{t,t}^p \overset{d}{=} \mathbf{Y}_p^t(\mathbf{Q}^t)^\dagger + \left[\mathbf{X}_p^t(\mathbf{M}_p^t)^\dagger\right]^T \mathbf{P}_{\mathbf{Q}^t}^\perp + \mathbf{P}_{\mathbf{M}_p^t}^\perp \widetilde{\mathbf{A}}_p \mathbf{P}_{\mathbf{Q}^t}^\perp,$$

where $\mathbf{X}_p^t$ and $\mathbf{Y}_p^t$ are defined in (3.37).

Since $\mathbf{b}_p^t = \mathbf{A}_p\mathbf{q}^t - \lambda_t\mathbf{m}_p^{t-1}$,

$$\begin{aligned}
\mathbf{b}_p^t|\mathfrak{S}_{t,t}^p &\stackrel{d}{=} \left[\mathbf{X}_p^t(\mathbf{M}_p^t)^\dagger\right]^T \mathbf{q}_\perp^t + \mathbf{Y}_p^t(\mathbf{Q}^t)^\dagger\mathbf{q}_{\shortparallel}^t \\
&+ \mathbf{P}_{\mathbf{M}_p^t}^\perp \widetilde{\mathbf{A}}_p\mathbf{q}_\perp^t - \lambda_t\mathbf{m}_p^{t-1} \\
&= \left\{\left[\mathbf{H}_p^t + \mathbf{Q}^t\boldsymbol{\Xi}_p^t\right](\mathbf{M}_p^t)^\dagger\right\}^T \mathbf{q}_\perp^t \\
&+ \left(\mathbf{B}_p^t + \left[\mathbf{0}|\mathbf{M}_p^{t-1}\right]\boldsymbol{\Lambda}^t\right)(\mathbf{Q}^t)^\dagger\mathbf{q}_{\shortparallel}^t \\
&+ \mathbf{P}_{\mathbf{M}_p^t}^\perp(\widetilde{\mathbf{A}}_p)\mathbf{q}_\perp^t - \lambda_t\mathbf{m}_p^{t-1},
\end{aligned} \tag{3.138}$$

where Ξ_p^t and Λ^t are define in (3.38).

By definition of $\mathbf{q}_{\shortparallel}^t$, $\mathbf{q}_\perp^t$ and $\mathbf{Q}^t$, we have

$$(\mathbf{Q}^t)^\dagger\mathbf{q}_{\shortparallel}^t = \boldsymbol{\beta}^t,$$

and

$$\left\{\mathbf{Q}^t\boldsymbol{\Xi}_p^t(\mathbf{M}_p^t)^\dagger\right\}^T \mathbf{q}_\perp^t = \left[(\mathbf{M}_p^t)^\dagger\right]^T \boldsymbol{\Xi}_p^t\left[\mathbf{Q}^t\right]^T \mathbf{q}_\perp^t = 0.$$

Therefore,

$$\begin{aligned}
\mathbf{b}_p^t|\mathfrak{S}_{t,t}^p &= \left[\mathbf{H}_p^t(\mathbf{M}_p^t)^\dagger\right]^T \mathbf{q}_\perp^t + \mathbf{B}_p^t\boldsymbol{\beta}^t \\
&+ \mathbf{P}_{\mathbf{M}_p^t}^\perp \widetilde{\mathbf{A}}_p\mathbf{q}_\perp^t + \left[\mathbf{0}|\mathbf{M}_p^{t-1}\right]\boldsymbol{\Lambda}^t\boldsymbol{\beta}^t - \lambda_t\mathbf{m}_p^{t-1}.
\end{aligned} \tag{3.139}$$

Now we want to show that

$$\begin{aligned}
&\left[\mathbf{H}_p^t(\mathbf{M}_p^t)^\dagger\right]^T \mathbf{q}_\perp^t + \left[\mathbf{0}|\mathbf{M}_p^{t-1}\right]\boldsymbol{\Lambda}^t\boldsymbol{\beta}^t - \lambda_t\mathbf{m}_p^{t-1} \\
&= \mathbf{M}_p^t\overrightarrow{o}_t(1),
\end{aligned} \tag{3.140}$$

i.e.,

$$\begin{aligned}
\mathbf{b}_p^t|\mathfrak{S}_{t,t}^p &= \mathbf{B}_p^t\boldsymbol{\beta}^t + \mathbf{P}_{\mathbf{M}_p^t}^\perp \widetilde{\mathbf{A}}_p\mathbf{q}_\perp^t + \mathbf{M}_p^t\overrightarrow{o}_t(1) \\
&= \sum_{i=0}^{t-1}\beta_i\mathbf{b}_p^i + \mathbf{P}_{\mathbf{M}_p^t}^\perp \widetilde{\mathbf{A}}_p\mathbf{q}_\perp^t + \mathbf{M}_p^t\overrightarrow{o}_t(1),
\end{aligned} \tag{3.141}$$

Let $\mathbf{F} = (\mathbf{M}_p^t)^T \mathbf{M}_p^t / N$, noting that

$$
\begin{aligned}
&① \ \left[\mathbf{H}_p^t (\mathbf{M}_p^t)^\dagger\right]^T \mathbf{q}_\perp^t \\
&= \mathbf{M}_p^t \left[(\mathbf{M}_p^t)^T \mathbf{M}_p^t\right]^{-1} (\mathbf{H}_p^t)^T \mathbf{q}_\perp^t, \\
&= \left[\mathbf{m}_p^0 | \cdots | \mathbf{m}_p^{t-2} | \mathbf{m}_p^{t-1}\right] \mathbf{F}^{-1} \frac{(\mathbf{H}_p^t)^T \mathbf{q}_\perp^t}{N}, \text{ and} \\
&② \ \left[\mathbf{0} | \mathbf{M}_p^{t-1}\right] \mathbf{\Lambda}^t \boldsymbol{\beta}^t \\
&= \left[\mathbf{0} | \mathbf{m}_p^0 | \cdots | \mathbf{m}_p^{t-2}\right] \begin{bmatrix} \lambda_0 \beta_0 \\ \vdots \\ \lambda_{t-1} \beta_{t-1} \end{bmatrix}, \\
&③ - \lambda_t \mathbf{m}_p^{t-1}
\end{aligned}
\tag{3.142}
$$

are linear combinations of the t columns in $\mathbf{M}_p^t$. We will evaluate the coefficients of $\mathbf{m}_p^\ell$, $\forall 0 \leq \ell \leq t-1$ in each of them separately.

The coefficient of $\mathbf{m}_p^\ell$ in ② is $\lambda_{\ell+1}\beta_{\ell+1}$, $\forall 0 \leq \ell \leq t-2$ and 0 for $\ell = t-1$.

The coefficient of $\mathbf{m}_p^\ell$ in ① is $\left[\mathbf{F}^{-1}(\mathbf{H}_p^t)^T \mathbf{q}_\perp^t / N\right]_\ell$, where

$$
\begin{aligned}
&\left[\mathbf{F}^{-1} \frac{(\mathbf{H}_p^t)^T \mathbf{q}_\perp^t}{N}\right]_\ell = \sum_{r=0}^{t-1} \left[\mathbf{F}^{-1}\right]_{\ell,r} \frac{\left[(\mathbf{H}_p^t)^T \mathbf{q}_\perp^t\right]_r}{N} \\
&= \sum_{r=0}^{t-1} \left[\mathbf{F}^{-1}\right]_{\ell,r} \frac{(\mathbf{h}_p^r)^T \mathbf{q}_\perp^t}{N} = \sum_{r=0}^{t-1} \left[\mathbf{F}^{-1}\right]_{\ell,r} \left\langle \mathbf{h}_p^r, \mathbf{q}_\perp^t \right\rangle \\
&= \sum_{r=0}^{t-1} \left[\mathbf{F}^{-1}\right]_{\ell,r} \left\langle \mathbf{h}_p^r, \mathbf{q}^t - \mathbf{q}_\parallel^t \right\rangle \\
&= \sum_{r=0}^{t-1} \left[\mathbf{F}^{-1}\right]_{\ell,r} \left\langle \mathbf{h}_p^r, \mathbf{q}^t - \sum_{s=0}^{t-1} \beta_s \mathbf{q}^s \right\rangle \\
&= \sum_{r=0}^{t} \left\{ \left[\mathbf{F}^{-1}\right]_{\ell,r} \left\langle \mathbf{h}_p^r, \mathbf{q}^t \right\rangle - \sum_{s=0}^{t-1} \beta_s \left[\mathbf{F}^{-1}\right]_{\ell,r} \left\langle \mathbf{h}_p^r, \mathbf{q}^s \right\rangle \right\}.
\end{aligned}
\tag{3.143}
$$

Applying Lemma 8, we have

$$\begin{aligned}
\left\langle \mathbf{h}_p^r, \mathbf{q}^s \right\rangle &= \left\langle \mathbf{h}_p^r, f_s\left(\sum_{q=1}^{P} \mathbf{h}_q^s, \mathbf{v}^s, \mathbf{s}_0\right)\right\rangle \\
&\overset{\text{a.s.}}{=} \left\langle \mathbf{h}_p^r, \sum_{q=1}^{P} \mathbf{h}_q^s \right\rangle \left\langle f_s'\left(\sum_{q=1}^{P} \mathbf{h}_q^s, \mathbf{v}^s, \mathbf{s}_0\right)\right\rangle \\
&= \kappa\lambda_s \left\langle \mathbf{h}_p^r, \mathbf{h}_p^s \right\rangle \overset{\text{a.s.}}{=} \kappa\lambda_s\omega_p \left\langle \mathbf{m}_p^{r-1}, \mathbf{m}_p^{s-1} \right\rangle \\
&= \lambda_s \frac{(\mathbf{m}_p^{r-1})^T \mathbf{m}_p^{s-1}}{N} = \lambda_s F_{r,s}, \forall 0 \le s \le t-1,
\end{aligned} \tag{3.144}$$

and

$$\begin{aligned}
&\left[\mathbf{F}^{-1}\frac{(\mathbf{H}_p^t)^T\mathbf{q}_\perp^t}{N}\right]_\ell \overset{\text{a.s.}}{=} \sum_{r=0}^{t} \left[\mathbf{F}^{-1}\right]_{\ell,r} \left(\lambda_t F_{r,t} - \sum_{s=0}^{t-1} \beta_s\lambda_s F_{r,s}\right) \\
&= \lambda_t \left[\mathbf{F}^{-1}\mathbf{F}\right]_{\ell,t} - \sum_{s=0}^{t-1} \beta_s\lambda_s \left[\mathbf{F}^{-1}\mathbf{F}\right]_{\ell,s} = \lambda_t\delta_{\ell t} - \sum_{s=0}^{t-1} \beta_s\lambda_s\delta_{\ell s} \\
&= \begin{cases} -\lambda_{\ell+1}\beta_{\ell+1}, & 0 \le \ell \le t-2, \\ \lambda_t, & \ell = t-1. \end{cases}
\end{aligned} \tag{3.145}$$

The coefficient of $\mathbf{m}_p^\ell$ in ③ is 0, $\forall 0 \le \ell \le t-2$ and λ_t for $\ell = t-1$.

Therefore, (3.140) holds, further, according to Lemma 5, we have

$$\mathbf{P}_{\mathbf{M}_p^t}(\widetilde{\mathbf{A}}_p)\mathbf{q}_\perp^t \overset{d}{=} \widetilde{\mathbf{M}}_p^t \overrightarrow{o}_t(1). \tag{3.146}$$

Therefore, applying Corollary 3, we get

$$\begin{aligned}
\mathbf{b}_p^t|\mathfrak{S}_{t,t}^p &\overset{d}{=} \sum_{i=0}^{t-1} \beta_i \mathbf{b}_p^i + \widetilde{\mathbf{A}}_p\mathbf{q}_\perp^t + \widetilde{\mathbf{M}}_p^t \overrightarrow{o}_t(1) + \mathbf{M}_p^t \overrightarrow{o}_t(1) \\
&= \sum_{i=0}^{t-1} \beta_i \mathbf{b}_p^i + \widetilde{\mathbf{A}}_p\mathbf{q}_\perp^t + \widetilde{\mathbf{M}}_p^t \overrightarrow{o}_t(1).
\end{aligned} \tag{3.147}$$

$\mathcal{B}_t$ (e) According to (3.141) and Proposition 2,

$$\begin{aligned}
&\frac{1}{n_p}\sum_{i=1}^{n_p}\left[\mathbf{b}_p^t\right]_i^{2\ell} = \frac{1}{n_p}\sum_{i=1}^{n_p}\left[\sum_{s=0}^{t-1}\beta_s\mathbf{b}_p^s + \mathbf{P}_{\mathbf{M}_p^t}^{\perp}\widetilde{\mathbf{A}}_p\mathbf{q}^0 + \mathbf{M}_p^t\overrightarrow{o}_t(1)\right]_i^{2\ell} \\
&\leq \frac{(t+2)^\ell}{n_p}\sum_{i=1}^{n_p}\left\{\sum_{s=0}^{t-1}\beta_s^{2\ell}\left[\mathbf{b}_p^s\right]_i^{2\ell}\right. \\
&\left.+\left[\mathbf{P}_{\mathbf{M}_p^t}^{\perp}\widetilde{\mathbf{A}}_p\mathbf{q}^0\right]_i^{2\ell} + \left[\mathbf{m}_p^s\right]_i^{2\ell}o(1)\right\},
\end{aligned} \tag{3.148}$$

According to Corollary 2 and induction $\mathcal{B}_r$ (e) for any $r \leq t-1$,

$$\frac{\sum_{s=0}^{t-1}\beta_s^{2\ell}\left[\mathbf{b}_p^s\right]_i^{2\ell}}{n_p} < \infty. \tag{3.149}$$

Considering that

$$\left[\mathbf{P}_{\mathbf{M}_p^t}^{\perp}\widetilde{\mathbf{A}}_p\mathbf{q}_{\perp}^t\right]_i = \mathbf{e}_i^T\mathbf{P}_{\mathbf{M}_p^t}^{\perp}\widetilde{A}^p\mathbf{q}_{\perp}^t \overset{d}{=} Z_i\frac{\|\mathbf{P}_{\mathbf{M}_p^t}^{\perp}\mathbf{e}_i\|\|\mathbf{q}_{\perp}^t\|}{\sqrt{n}}, \tag{3.150}$$

we have

$$\begin{aligned}
&\frac{\sum_{i=1}^{n_p}\left[\mathbf{P}_{\mathbf{M}_p^t}^{\perp}\widetilde{\mathbf{A}}_p\mathbf{q}_{\perp}^t\right]_i^{2\ell}}{n_p} \overset{d}{=} \sum_{i=1}^{n_p}\frac{Z_i^{2\ell}}{n_p}\left[\frac{\|\mathbf{P}_{\mathbf{M}_p^t}^{\perp}\mathbf{e}_i\|\|\mathbf{q}_{\perp}^t\|}{\sqrt{n}}\right]^{2\ell} \\
&\leq \sum_{i=1}^{n_p}\frac{Z_i^{2\ell}}{n_p}\left[\frac{\|\mathbf{q}^t\|}{\sqrt{n}}\right]^{2\ell} < \infty.
\end{aligned} \tag{3.151}$$

Further, since $g_t(\cdot,\cdot)$ is Lipschitz continuous, we know that $\exists L > 0$, such that

$$|m_{p,i}^s| = |g_t(b_{p,i}^s, w_{p,i}^s)| \leq |g_t(0,0)| + L\sqrt{|b_{p,i}^s|^2 + |w_{p,i}^s|^2}.$$

Therefore, we have

$$\frac{\sum_{i=1}^{n_p}\left[\mathbf{m}_p^s\right]_i^{2\ell}o(1)}{n_p} \leq \frac{L'}{n_p}\sum_{i=1}^{n_p}(1 + |b_{p,i}^s|^{2\ell} + |w_{p,i}|^{2\ell})o(1) < \infty,$$

and $\mathcal{B}_t$ (e) holds.

$\mathcal{B}_t$ (c)

We consider three cases: i) $r < t$ and $s < t$, ii) $s = t$ and $r < t$, and iii) $r = s = t$.

i) $\mathcal{B}_t$ (c) holds $\forall r, s < t$ by induction.

ii) $s = t, r < t$, according to (3.141),

$$\begin{aligned}\left\langle \mathbf{b}_p^t, \mathbf{b}_p^r \right\rangle | \mathfrak{S}_{t,t} &\stackrel{d}{=} \sum_{i=0}^{t-1} \beta_i \left\langle \mathbf{b}_p^i, \mathbf{b}_p^r \right\rangle \\ &+ \left\langle \mathbf{P}^{\perp}_{\mathbf{M}_p^t} \widetilde{\mathbf{A}}_p \mathbf{q}_{\perp}^t, \mathbf{b}_p^r \right\rangle + \sum_{i=0}^{t} o(1) \left\langle \mathbf{m}_p^i, \mathbf{b}_p^r \right\rangle .\end{aligned} \tag{3.152}$$

By induction $\mathcal{B}_{t-1}(d)$, $\left\langle \mathbf{m}_p^i, \mathbf{b}_p^r \right\rangle < \infty$, $\forall r \leq t-1$, $\forall i \leq t$.

$$\therefore \lim_{N\to\infty} \sum_{i=0}^{t} o(1) \left\langle \mathbf{m}_p^i, \mathbf{b}_p^r \right\rangle \stackrel{\text{a.s.}}{=} 0. \tag{3.153}$$

By Lemma 5,

$$\begin{aligned}&\left\langle \mathbf{P}^{\perp}_{\mathbf{M}_p^t} \widetilde{\mathbf{A}}_p \mathbf{q}_{\perp}^t, \mathbf{b}_p^r \right\rangle \stackrel{d}{=} \lim_{N\to\infty} \frac{Z}{n_p\sqrt{n}} \|\mathbf{q}_{\perp}^t\| \|\mathbf{P}^{\perp}_{\mathbf{M}_p^t} \mathbf{b}_p^r\| \\ &\leq \lim_{N\to\infty} \frac{Z}{n_p\sqrt{n}} \|\mathbf{q}^t\| \|\mathbf{b}_p^r\| = \lim_{N\to\infty} \sqrt{\left\langle \mathbf{q}^t, \mathbf{q}^t \right\rangle \left\langle \mathbf{b}_p^r, \mathbf{b}_p^r \right\rangle} \frac{Z}{\sqrt{\kappa n_p}}.\end{aligned}$$

By induction hypothesis $\mathcal{H}_t$ (b) and $\mathcal{B}_{t-1}$ (c), we know that

$$\sqrt{\left\langle \mathbf{q}^t, \mathbf{q}^t \right\rangle \left\langle \mathbf{b}_p^r, \mathbf{b}_p^r \right\rangle} < \infty. \tag{3.154}$$

Therefore,

$$\left\langle \mathbf{P}^{\perp}_{\mathbf{M}_p^t} \widetilde{\mathbf{A}}_p \mathbf{q}_{\perp}^t, \mathbf{b}_p^r \right\rangle \stackrel{\text{a.s.}}{\to} 0. \tag{3.155}$$

Hence

$$\begin{aligned}&\left\langle \mathbf{b}_p^t, \mathbf{b}_p^r \right\rangle \stackrel{\text{a.s.}}{=} \sum_{i=0}^{t-1} \beta_i \left\langle \mathbf{b}_p^i, \mathbf{b}_p^r \right\rangle \stackrel{\text{a.s.}}{=} \frac{1}{\kappa} \lim_{n\to\infty} \sum_{i=0}^{t-1} \beta_i \left\langle \mathbf{q}^i, \mathbf{q}^r \right\rangle \\ &= \frac{1}{\kappa} \lim_{n\to\infty} \left\langle \mathbf{q}_{\shortparallel}^t, \mathbf{q}^r \right\rangle = \frac{1}{\kappa} \lim_{n\to\infty} \left\langle \mathbf{q}^t, \mathbf{q}^r \right\rangle .\end{aligned}$$

iii) $r = s = t$,

$$
\begin{aligned}
&\left\langle \mathbf{b}_p^t, \mathbf{b}_p^t \right\rangle | \mathfrak{S}_{t,t}^p \overset{d}{=} \\
&\left\langle \sum_{i=0}^{t-1} \beta_i \mathbf{b}_p^i + \mathbf{P}_{\mathbf{M}_p^t}^{\perp} \widetilde{\mathbf{A}}_p \mathbf{q}_{\perp}^t + \sum_{i=0}^{t-1} \mathbf{m}_p^i o(1), \right. \\
&\left. \sum_{j=0}^{t-1} \beta_j \mathbf{b}_p^j + \mathbf{P}_{\mathbf{M}_p^t}^{\perp} \widetilde{\mathbf{A}}_p \mathbf{q}_{\perp}^t + \sum_{j=0}^{t-1} \mathbf{m}_p^j o(1) \right\rangle \\
&= \sum_{i,j=0}^{t-1} \beta_i \beta_j \left\langle \mathbf{b}_p^i, \mathbf{b}_p^j \right\rangle \ \textcircled{i} \\
&+ 2 \sum_{i=0}^{t-1} \beta_i \left\langle \mathbf{P}_{\mathbf{M}_p^t}^{\perp} \widetilde{\mathbf{A}}_p \mathbf{q}_{\perp}^t, \mathbf{b}_p^i \right\rangle \ \textcircled{ii} \\
&+ 2 \sum_{i=0}^{t-1} \sum_{j=0}^{t} \beta_i \left\langle \mathbf{b}_p^i, \mathbf{m}_p^j \right\rangle o(1) \ \textcircled{iii} \\
&+ \left\langle \mathbf{P}_{\mathbf{M}_p^t}^{\perp} \widetilde{\mathbf{A}}_p \mathbf{q}_{\perp}^t, \mathbf{P}_{\mathbf{M}_p^t}^{\perp} \widetilde{\mathbf{A}}_p \mathbf{q}_{\perp}^t \right\rangle \ \textcircled{iv} \\
&+ 2 \sum_{j=0}^{t} \left\langle \mathbf{P}_{\mathbf{M}_p^t}^{\perp} \widetilde{\mathbf{A}}_p \mathbf{q}_{\perp}^t, \mathbf{m}_p^j o(1) \right\rangle \textcircled{v} \\
&+ \sum_{i,j=0}^{t-1} \left\langle \mathbf{m}_p^i, \mathbf{m}_p^j \right\rangle o(1) \ \textcircled{vi}.
\end{aligned}
$$

By (3.155), $\mathcal{B}_{t-1}$ (d) and (b), we know that ⓘⓘ, ⓘⓘⓘ, and ⓥⓘ $\overset{\text{a.s.}}{\to} 0$.

Consider ⓥ,

$$
\begin{aligned}
&\left\langle \mathbf{P}_{\mathbf{M}_p^t}^{\perp} \widetilde{\mathbf{A}}_p \mathbf{q}_{\perp}^t, \mathbf{m}_p^j o(1) \right\rangle \\
&\overset{d}{=} Z \frac{\|\mathbf{q}_{\perp}^t\| \|\mathbf{P}_{\mathbf{M}_p^t}^{\perp} \mathbf{m}_p^j\| o(1)}{n_p \sqrt{n}} \overset{\text{a.s.}}{\to} 0,
\end{aligned}
\tag{3.156}
$$

ⓘ,

$$
\begin{aligned}
&\sum_{i,j=0}^{t-1} \beta_i \beta_j \left\langle \mathbf{b}_p^i, \mathbf{b}_p^j \right\rangle = \frac{1}{\kappa} \sum_{i,j=0}^{t-1} \beta_i \beta_j \left\langle \mathbf{q}^i, \mathbf{q}^j \right\rangle \\
&= \frac{1}{\kappa} \sum_{i=0}^{t-1} \beta_i \left\langle \mathbf{q}^i, \mathbf{q}_{\shortparallel}^t \right\rangle = \frac{1}{\kappa} \left\langle \mathbf{q}_{\shortparallel}^t, \mathbf{q}_{\shortparallel}^t \right\rangle,
\end{aligned}
\tag{3.157}
$$

and ⓘⓘ,

$$\begin{aligned}
&\left\langle \mathbf{P}^{\perp}_{\mathbf{M}_p^t} \widetilde{\mathbf{A}}_p \mathbf{q}_{\perp}^t, \mathbf{P}^{\perp}_{\mathbf{M}_p^t} \widetilde{\mathbf{A}}_p \mathbf{q}_{\perp}^t \right\rangle \\
&= \left\langle \widetilde{\mathbf{A}}_p \mathbf{q}_{\perp}^t, \widetilde{\mathbf{A}}_p \mathbf{q}_{\perp}^t \right\rangle - \left\langle \mathbf{P}_{\mathbf{M}_p^t} \widetilde{\mathbf{A}}_p \mathbf{q}_{\perp}^t, \mathbf{P}_{\mathbf{M}_p^t} \widetilde{\mathbf{A}}_p \mathbf{q}_{\perp}^t \right\rangle \\
&\overset{\text{a.s.}}{\to} \frac{\|\mathbf{q}_{\perp}^t\|^2}{n} - \frac{\|\widetilde{\mathbf{M}}_p^t \overrightarrow{o}_t(1)\|^2}{n_p} \overset{\text{a.s.}}{\to} \frac{1}{\kappa} \left\langle \mathbf{q}_{\perp}^t, \mathbf{q}_{\perp}^t \right\rangle .
\end{aligned} \tag{3.158}$$

Therefore,

$$\left\langle \mathbf{b}_p^t, \mathbf{b}_p^t \right\rangle \overset{\text{a.s.}}{\to} \frac{1}{\kappa} \left\langle \mathbf{q}_{\shortparallel}^t, \mathbf{q}_{\shortparallel}^t \right\rangle + \frac{1}{\kappa} \left\langle \mathbf{q}_{\perp}^t, \mathbf{q}_{\perp}^t \right\rangle = \frac{1}{\kappa} \left\langle \mathbf{q}^t, \mathbf{q}^t \right\rangle . \tag{3.159}$$

$\mathcal{B}_t$ (b) Apply $\mathcal{B}_t$ (a),

$$\begin{aligned}
&\phi_b \left(\left[\mathbf{b}_p^0\right]_i, \cdots, \left[\mathbf{b}_p^{t-1}\right]_i, \left[\mathbf{b}_p^t\right]_i, [\mathbf{w}_p]_i \right) | \mathfrak{S}_{t,t}^p \overset{d}{=} \\
&\phi_b \left(\left[\mathbf{b}_p^0\right]_i, \cdots, \left[\mathbf{b}_p^{t-1}\right]_i, \right. \\
&\left. \left[\sum_{r=0}^{t-1} \beta_r \mathbf{b}_p^r + \widetilde{\mathbf{A}}_p \mathbf{q}_{\perp}^t + \widetilde{\mathbf{M}}_p^t \overrightarrow{o}_t(1) \right]_i, [\mathbf{w}_p]_i \right) .
\end{aligned} \tag{3.160}$$

Let

$$\begin{aligned}
\mathbf{a}_i = &\left(\left[\mathbf{b}_p^0\right]_i, \cdots, \left[\mathbf{b}_p^{t-1}\right]_i, \right. \\
&\left. \left[\sum_{r=0}^{t-1} \beta_r \mathbf{b}_p^r + \widetilde{\mathbf{A}}_p \mathbf{q}_{\perp}^t + \widetilde{\mathbf{M}}_p^t \overrightarrow{o}_t(1) \right]_i, [\mathbf{w}_p]_i \right) ,
\end{aligned}$$

and

$$\mathbf{c}_i = \left(\left[\mathbf{b}_p^0\right]_i, \cdots, \left[\mathbf{b}_p^{t-1}\right]_i, \left[\sum_{r=0}^{t-1} \beta_r \mathbf{b}_p^r + \widetilde{\mathbf{A}}_p \mathbf{q}_{\perp}^t \right]_i, [\mathbf{w}_p]_i \right) .$$

Similar to $\mathcal{B}_0$ (b), we want to show that

$$\frac{1}{n_p} \left| \sum_{i=1}^{n_p} \left[\phi_b(\mathbf{a}_i) - \phi_b(\mathbf{c}_i) \right] \right| \overset{\text{a.s.}}{\to} 0. \tag{3.161}$$

Applying the property of pseudo Lipschitz continuous function of order k, we

have

$$\begin{aligned}
&\frac{1}{n_p}\left|\sum_{i=1}^{n_p}\left[\phi_b(\mathbf{a}_i)-\phi_b(\mathbf{c}_i)\right]\right| \le \frac{1}{n_p}\sum_{i=1}^{n_p}\left|\phi_b(\mathbf{a}_i)-\phi_b(\mathbf{c}_i)\right| \\
&\le \frac{L}{n_p}\sum_{i=1}^{n_p}(1+\|\mathbf{a}_i\|^{k-1}+\|\mathbf{c}_i\|^{k-1})\left|\left[\widetilde{\mathbf{M}}_p^t\overrightarrow{o}_t(1)\right]_i\right| \\
&\le \frac{L}{n_p}\sqrt{\sum_{i=1}^{n_p}(1+\|\mathbf{a}_i\|^{k-1}+\|\mathbf{c}_i\|^{k-1})^2}\sqrt{\frac{\|\widetilde{\mathbf{M}}_p^t\overrightarrow{o}_t(1)\|^2}{n_p}} \\
&\le L'\sqrt{\sum_{i=1}^{n_p}\frac{1+\|\mathbf{a}_i\|^{2k-2}+\|\mathbf{c}_i\|^{2k-2}}{n_p}o(1)},
\end{aligned} \tag{3.162}$$

where

$$\begin{aligned}
\frac{\sum_{i=1}^{n_p}\|\mathbf{a}_i\|^{2\ell}}{n_p} &= \frac{\sum_{i=1}^{n_p}\left\{\sum_{r=0}^{t}\left[\mathbf{b}_p^r\right]_i^2+[\mathbf{w}_p]_i^2\right\}^\ell}{N} \\
&\le (t+2)^\ell\frac{\sum_{i=1}^{n_p}\left\{\sum_{r=0}^{t}\left[\mathbf{b}_p^r\right]_i^{2\ell}+[\mathbf{w}_p]_i^{2\ell}\right\}}{n_p} \\
&< \infty, \text{ due to } \mathcal{B}_t \text{ (e)},
\end{aligned} \tag{3.163}$$

and

$$\mathbf{c}_i = \mathbf{a}_i - \left(0,\cdots,\left[\widetilde{\mathbf{M}}_p^t\overrightarrow{o}_t(1)\right]_i,0\right).$$

Applying Corollary 3, we know that

$$\mathbf{c}_i = \mathbf{a}_i - \left(0,\cdots,\left[\mathbf{M}_p^t\overrightarrow{o}_t(1)\right]_i,0\right),$$

which satisfies

$$\begin{aligned}
\frac{\sum_{i=1}^{n_p}\|\mathbf{c}_i\|^{2\ell}}{n_p} &\le C\frac{\sum_{i=1}^{n_p}\|\mathbf{a}_i\|^{2\ell}+\sum_{i=1}^{n_p}\left(\sum_{r=0}^{t-1}\left[\mathbf{m}_p^r\right]_i o(1)\right)^{2\ell}}{n_p} \\
&\le C\frac{\sum_{i=1}^{N}\|\mathbf{a}_i\|^{2\ell}}{n_p}+C'\frac{\sum_{r=0}^{t-1}\sum_{i=1}^{n_p}\left[\mathbf{m}_p^r\right]_i^{2\ell}o(1)}{n_p}<\infty.
\end{aligned} \tag{3.164}$$

Similar to the proof of $\mathcal{B}_0$ (b), using Lemma 5, we have

$$\begin{aligned}
&\frac{1}{n_p}\sum_{i=1}^{n_p}\phi_h\left(b_{p,i}^0,\cdots,b_{p,i}^{t-1},\left[\sum_{r=0}^{t-1}\beta_r\mathbf{b}_p^r+\widetilde{\mathbf{A}}_p\mathbf{q}_\perp^t\right]_i,w_{p,i}\right)\\
&\overset{\text{a.s.}}{\to}\frac{1}{n_p}\sum_{i=1}^{n_p}\mathbb{E}_Z\phi_b\left(b_{p,i}^0,\cdots,b_{p,i}^{t-1},\sum_{r=0}^{t-1}\beta_r b_{p,i}^r+\frac{\|\mathbf{q}_\perp^t\|Z}{\sqrt{n}},w_{p,i}\right).
\end{aligned}$$

The right hand side of the above equation is a function of $(b_{p,i}^0,\cdots,b_p^{t-1},i)$, so we can use induction $\mathcal{B}_{t-1}$ (b) to obtain

$$\begin{aligned}
&\frac{1}{n_p}\sum_{i=1}^{n_p}\mathbb{E}_Z\phi_b\left(b_{p,i}^0,\cdots,b_{p,i}^{t-1},\sum_{r=0}^{t-1}\beta_r b_{p,i}^r+\frac{\|\mathbf{q}_\perp^t\|Z}{\sqrt{n}},w_{p,i}\right)\\
&\overset{\text{a.s.}}{\to}\frac{1}{n_p}\sum_{i=1}^{n_p}\mathbb{E}_{Z,\widehat{Z}}\phi_b\left(\sigma_0\widehat{Z}_p^0,\cdots,\sigma_{t-1}\widehat{Z}_p^{t-1},\right.\\
&\left.\sum_{r=0}^{t-1}\beta_r\sigma_r\widehat{Z}_p^r+\frac{\|\mathbf{q}_\perp^t\|Z}{\sqrt{n}},w_{p,i}\right).
\end{aligned}$$

Now we need to show that

$$\operatorname{Var}\left\{\sum_{r=0}^{t-1}\beta_r\sigma_r\widehat{Z}_r\right\}+\frac{\|\mathbf{q}_\perp^t\|^2}{n}\overset{\text{a.s.}}{=}\sigma_t^2. \tag{3.165}$$

Using induction $\mathcal{H}_t$ (b) with

$$\begin{aligned}
&\phi_h(\left[\mathbf{h}_1^1\right]_i,\cdots,\left[\mathbf{h}_P^t\right]_i,\cdots,[\mathbf{s}_0]_i)\\
&=\left[f_t\left(\left[\sum_{p=1}^P\mathbf{h}_p^t\right]_i,[\mathbf{v}_t]_i,[\mathbf{s}_0]_i\right)\right]^2,
\end{aligned} \tag{3.166}$$

we have

$$\left\langle\mathbf{q}^t,\mathbf{q}^t\right\rangle=\frac{1}{N}\sum_{i=1}^N\left[f_t\left(\left[\sum_{p=1}^P\mathbf{h}_p^t\right]_i,[\mathbf{v}_t]_i,[\mathbf{s}_0]_i\right)\right]^2\overset{\text{a.s.}}{\to}\kappa\sigma_t^2. \tag{3.167}$$

On the other hand,

$$\left\langle\mathbf{q}^t,\mathbf{q}^t\right\rangle=\left\langle\mathbf{q}_{\shortparallel}^t,\mathbf{q}_{\shortparallel}^t\right\rangle+\left\langle\mathbf{q}_\perp^t,\mathbf{q}_\perp^t\right\rangle, \tag{3.168}$$

where

$$\begin{aligned}\left\langle \mathbf{q}_{\shortparallel}^{t}, \mathbf{q}_{\shortparallel}^{t} \right\rangle &= \left\langle \sum_{r=0}^{t-1} \beta^{r} \mathbf{q}^{r}, \sum_{s=0}^{t-1} \beta_{s} \mathbf{q}_{s} \right\rangle \\ &= \sum_{r=0}^{t-1} \sum_{s=0}^{t-1} \beta_{r} \beta_{s} \left\langle \mathbf{q}^{r}, \mathbf{q}^{s} \right\rangle .\end{aligned} \tag{3.169}$$

Applying $\mathcal{B}_t$ (c) and induction hypothesis $\mathcal{B}_{t-1}$ (b) for $\phi_b(\cdots) = b_{p,i}^r b_{p,i}^s$, $0 \leq r, s \leq t-1$, we have

$$\begin{aligned}\left\langle \mathbf{q}_{\shortparallel}^{t}, \mathbf{q}_{\shortparallel}^{t} \right\rangle &= \frac{1}{\kappa} \sum_{r=0}^{t-1} \sum_{s=0}^{t-1} \beta_{r} \beta_{s} \left\langle \mathbf{b}_{p}^{r}, \mathbf{b}_{p}^{s} \right\rangle \\ &\overset{\text{a.s.}}{=} \kappa \sum_{r=0}^{t-1} \sum_{s=0}^{t-1} \beta_{r} \beta_{s} \mathbb{E}(\sigma_{r} \widehat{Z}_{p}^{r} \sigma_{s} \widehat{Z}_{p}^{s}) = \kappa \mathrm{Var} \left\{ \sum_{r=0}^{t-1} \beta_{r} \sigma_{r} \widehat{Z}_{p}^{r} \right\} .\end{aligned}$$

Therefore, (3.165) holds. Applying Lemma 4 in [58], we finally have

$$\begin{aligned}&\lim_{n_p \to \infty} \sum_{i=1}^{n_p} \phi_b \left(\left[\mathbf{b}_{p}^{0} \right]_{i}, \cdots, \left[\mathbf{b}_{p}^{t} \right]_{i}, \left[\mathbf{w}_{p} \right]_{i} \right) \overset{\text{a.s.}}{=} \\ &\overset{\text{a.s.}}{=} \mathbb{E}_{\widehat{Z}, W} \phi_b \left(\sigma_0 \widehat{Z}_{p}^{0}, \cdots, \sigma_{t-1} \widehat{Z}_{p}^{t-1}, \sigma_t \widehat{Z}_{p}^{t}, W \right) .\end{aligned} \tag{3.170}$$

$\mathcal{B}_t$ (d) Using $\mathcal{B}_t$ (b) and Lemma 8,

$$\begin{aligned}&\lim_{n \to \infty} \left\langle \mathbf{b}_{p}^{t}, \phi(\mathbf{b}_{p}^{s}, \mathbf{w}_{p}) \right\rangle \overset{\text{a.s.}}{=} \mathbb{E}\{\sigma_t \widehat{Z}_{p}^{t} \phi(\sigma_s \widehat{Z}_{p}^{s}, W)\} \\ &\overset{\text{a.s.}}{=} \mathrm{Cov}(\sigma_t \widehat{Z}_{p}^{t}, \sigma_s \widehat{Z}_{p}^{s}) \mathbb{E}\{\phi'(\sigma_s \widehat{Z}_{p}^{s}, W)\} \\ &\overset{\text{a.s.}}{=} \left\langle \mathbf{b}_{p}^{t}, \mathbf{b}_{p}^{s} \right\rangle \left\langle \phi'(\mathbf{b}_{p}^{s}, \mathbf{w}_{p}) \right\rangle .\end{aligned} \tag{3.171}$$

Step IV: $\mathcal{H}_{t+1}$

The proofs of $\mathcal{H}_{t+1}$ (c), (e), and (h) are very similar to those of $\mathcal{B}_t$ and are skipped. For the remain parts, we prove them in the order of (a), (b), (g), (d), and (f).

$\mathcal{H}_{t+1}$ (a) $\mathfrak{S}_{t+1,t}^{p} = \{\mathbf{B}_{p}^{t+1}, \mathbf{M}_{p}^{t+1}, \mathbf{H}_{p}^{t}, \mathbf{v}^{1}, \cdots, \mathbf{v}^{t}, \mathbf{Q}^{t+1}, \mathbf{s}_0, \mathbf{w}_{p}\}$.

Applying Lemma 6, we have

$$\begin{aligned}\mathbf{A}_p|\mathfrak{S}_{t+1,t}^p &\overset{d}{=} \mathbf{P}_{\mathbf{M}_p^t}^{\perp}\mathbf{Y}_p^{t+1}(\mathbf{Q}^{t+1})^{\dagger} + \left[\mathbf{X}_p^t(\mathbf{M}_p^t)^{\dagger}\right]^T \\ &+ \mathbf{P}_{\mathbf{M}_p^t}^{\perp}\widetilde{\mathbf{A}}_p\mathbf{P}_{\mathbf{Q}^{t+1}}^{\perp}.\end{aligned} \tag{3.172}$$

Since $\mathbf{h}_p^{t+1} = \mathbf{A}_p^T\mathbf{m}_p^t - \omega_p\xi_t\mathbf{q}^t$,

$$\begin{aligned}\mathbf{h}_p^{t+1}|\mathfrak{S}_{t+1,t}^p &\overset{d}{=} \mathbf{X}_p^t(\mathbf{M}_p^t)^{\dagger}\mathbf{m}_{p,||}^t + \left[\mathbf{Y}_p^{t+1}(\mathbf{Q}^{t+1})^{\dagger}\right]^T\mathbf{m}_{p,\perp}^t \\ &+ \mathbf{P}_{\mathbf{Q}^{t+1}}^{\perp}(\widetilde{\mathbf{A}}_p)^T\mathbf{m}_{p,\perp}^t - \omega_p\xi_t\mathbf{q}^t \\ &= \left[\mathbf{H}_p^t + \mathbf{Q}^t\Xi_p^t\right]\boldsymbol{\alpha}_p^t + \left[\left(\mathbf{B}_p^{t+1} + \left[\mathbf{0}|\mathbf{M}_p^t\right]\right)(\mathbf{Q}^{t+1})^{\dagger}\right]^T\mathbf{m}_{p,\perp}^t \\ &+ \mathbf{P}_{\mathbf{Q}^{t+1}}^{\perp}\widetilde{\mathbf{A}}_p^T\mathbf{m}_{p,\perp}^t - \omega_p\xi_t\mathbf{q}^t \\ &= \mathbf{H}_p^t\boldsymbol{\alpha}_p^t + \mathbf{P}_{\mathbf{Q}^{t+1}}^{\perp}(\widetilde{\mathbf{A}}_p)^T\mathbf{m}_{p,\perp}^t \\ &+ \mathbf{Q}^t\Xi_p^t\boldsymbol{\alpha}_p^t + \left[\mathbf{B}_p^{t+1}(\mathbf{Q}^{t+1})^{\dagger}\right]^T\mathbf{m}_{p,\perp}^t - \omega_p\xi_t\mathbf{q}^t.\end{aligned}$$

Similar to the proof of $\mathcal{B}_t$ (a), we can show that

$$\begin{aligned}&\mathbf{Q}^t\Xi_p^t\boldsymbol{\alpha}_p^t + \left[\mathbf{B}_p^{t+1}(\mathbf{Q}^{t+1})^{\dagger}\right]^T\mathbf{m}_{p,\perp}^t - \omega_p\xi_t\mathbf{q}^t \\ &= \mathbf{Q}^{t+1}\overrightarrow{o}_{t+1}(1).\end{aligned} \tag{3.173}$$

Therefore,

$$\begin{aligned}\mathbf{h}_p^{t+1}|\mathfrak{S}_{t+1,t}^p &\overset{d}{=} \mathbf{H}_p^t\boldsymbol{\alpha}_p^t + \mathbf{P}_{\mathbf{Q}^{t+1}}^{\perp}\widetilde{\mathbf{A}}_p^T\mathbf{m}_{p,\perp}^t + \mathbf{Q}^{t+1}\overrightarrow{o}_{t+1}(1) \\ &= \sum_{i=0}^{t-1}\alpha_p^i\mathbf{h}_p^{i+1} + \mathbf{P}_{\mathbf{Q}^{t+1}}^{\perp}\widetilde{\mathbf{A}}_p^T\mathbf{m}_{p,\perp}^t + \mathbf{Q}^{t+1}\overrightarrow{o}_{t+1}(1).\end{aligned} \tag{3.174}$$

Applying Lemma 5, we have

$$\mathbf{P}_{\mathbf{Q}^{t+1}}(\widetilde{\mathbf{A}}_p)^T\mathbf{m}_{p,\perp}^t \overset{d}{=} \widetilde{\mathbf{Q}}^{t+1}\overrightarrow{o}_{t+1}(1). \tag{3.175}$$

Then applying Corollary 3, we have

$$\mathbf{h}_p^{t+1}|\mathfrak{S}_{t+1,t}^p \overset{d}{=} \sum_{i=0}^{t-1}\alpha_p^i\mathbf{h}_p^{i+1} + (\widetilde{\mathbf{A}}_p)^T\mathbf{m}_{p,\perp}^t + \widetilde{\mathbf{Q}}^{t+1}\overrightarrow{o}_{t+1}(1).$$

$\mathcal{H}_{t+1}$ (b) Applying $\mathcal{H}_{t+1}$ (a) and following a similar proof in $\mathcal{B}_t$ (b),

$$\begin{aligned}
&\frac{1}{N}\sum_{i=1}^{N}\phi_h\left(h_{1,i}^1,\cdots,h_{P,i}^1,\cdots,h_{1,i}^{t+1},\cdots,h_{P,i}^{t+1},v_i^1,\cdots,v_i^t,s_{0,i}\right)\\
&\overset{\text{a.s.}}{\to}\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_Z\phi_h\left(\tau_0\sqrt{\omega_1}Z_1^0,\cdots,\tau_0\sqrt{\omega_1}Z_P^0,\cdots,\tau_{t-1}\sqrt{\omega_1}Z_1^{t-1},\right.\\
&\left.\cdots,\tau_{t-1}\sqrt{\omega_1}Z_P^{t-1},\sum_{r=0}^{t-1}\sqrt{\omega_1}\alpha_1^r\tau_r Z_1^r+\frac{\|\mathbf{m}_{1,\perp}^t\|Z_1}{\sqrt{n}},\cdots,\right.\\
&\left.\sum_{r=0}^{t-1}\sqrt{\omega_P}\alpha_P^r\tau_r Z_P^r+\frac{\|\mathbf{m}_{P,\perp}^t\|Z_P}{\sqrt{n}},V_1,\cdots,V_{t-1},v_i^t,s_{0,i}\right).
\end{aligned}\tag{3.176}$$

Now we need to show that

$$\text{Var}\left\{\sum_{r=0}^{t-1}\sqrt{\omega_P}\alpha_P^r\tau_r Z_P^r\right\}+\frac{\|\mathbf{m}_{p,\perp}^t\|^2}{n}\overset{\text{a.s.}}{=}\omega_p\tau_t^2,\ \forall p.\tag{3.177}$$

Using induction $\mathcal{B}_t$ (b) with $\phi_b(\left[\mathbf{b}_p^0\right]_i,\cdots,\left[\mathbf{b}_p^t\right]_i,[\mathbf{w}_p]_i)=\left[g_t(\left[\mathbf{b}_p^t\right]_i,[\mathbf{w}_p]_i)\right]^2$, we have

$$\left\langle\mathbf{m}_p^t,\mathbf{m}_p^t\right\rangle=\left[g_t(\left[\mathbf{b}_p^t\right]_i,[\mathbf{w}_p]_i)\right]^2\overset{\text{a.s.}}{\to}\tau_t^2.\tag{3.178}$$

On the other hand,

$$\left\langle\mathbf{m}_p^t,\mathbf{m}_p^t\right\rangle=\left\langle\mathbf{m}_{p,\shortparallel}^t,\mathbf{m}_{p,\shortparallel}^t\right\rangle+\left\langle\mathbf{m}_{p,\perp}^t,\mathbf{m}_{p,\perp}^t\right\rangle,\tag{3.179}$$

where

$$\begin{aligned}
\left\langle\mathbf{m}_{p,\shortparallel}^t,\mathbf{m}_{p,\shortparallel}^t\right\rangle&=\left\langle\sum_{r=0}^{t-1}\alpha_p^r\mathbf{m}_p^r,\sum_{s=0}^{t-1}\alpha_p^s\mathbf{m}_p^s\right\rangle\\
&=\sum_{r=0}^{t-1}\sum_{s=0}^{t-1}\alpha_p^r\alpha_p^s\left\langle\mathbf{m}_p^r,\mathbf{m}_p^s\right\rangle,
\end{aligned}\tag{3.180}$$

applying $\mathcal{H}_{t+1}$ (c) and induction $\mathcal{H}_t$ (b) with $\phi_h([\mathbf{h}_1^1]_i,\cdots,[\mathbf{h}_P^t]_i,[\mathbf{s}_0]_i)=\left[\mathbf{h}_p^{r+1}\right]_i\left[\mathbf{h}_p^{s+1}\right]_i$,

we have

$$\begin{aligned}
\left\langle \mathbf{m}_{p,\shortparallel}^{t}, \mathbf{m}_{p,\shortparallel}^{t} \right\rangle &= \frac{1}{\omega_p} \sum_{r=0}^{t-1} \sum_{s=0}^{t-1} \alpha_p^r \alpha_p^s \left\langle \mathbf{h}_p^{r+1}, \mathbf{h}_p^{s+1} \right\rangle \\
&\overset{\text{a.s.}}{=} \frac{1}{\omega_p} \sum_{r=0}^{t-1} \sum_{s=0}^{t-1} \alpha_p^r \alpha_p^s \mathbb{E}(\sqrt{\omega_p}\tau_r Z_p^r \sqrt{\omega_p}\tau_s Z_p^s) \\
&= \mathrm{Var}\left\{ \sum_{r=0}^{t-1} \alpha_P^r \tau_r Z_P^r \right\}.
\end{aligned} \tag{3.181}$$

Therefore, (3.177) holds, and apply Lemma 4 in [58],

$$\begin{aligned}
&\frac{1}{N} \sum_{i=1}^{N} \phi_h \left(h_{1,i}^1, \cdots, h_{P,i}^1, \cdots, h_{1,i}^{t+1}, \cdots, h_{P,i}^{t+1}, v_i^1, \cdots, v_i^t, s_{0,i} \right) \\
&\overset{\text{a.s.}}{\to} \mathbb{E}_Z \phi_h \left(\tau_0 \sqrt{\omega_1} Z_1^0, \cdots, \tau_0 \sqrt{\omega_P} Z_P^0, \cdots, \tau_t \sqrt{\omega_1} Z_1^t, \cdots, \right. \\
&\left. \tau_t \sqrt{\omega_1} Z_P^t, V_1, \cdots, V_t, S_0 \right).
\end{aligned}$$

Also, since Z_p^t only depends on Z_p^s for any $s < t$, by induction $\mathcal{H}_t$ (b), we can see that it is independent of any Z_q^r for $q \neq p$.

$\mathcal{H}_{t+1}$ (g) Using $\mathcal{H}_{t+1}$ (b) with $\phi_h([\mathbf{h}_1^1]_i, \cdots, \mathbf{s}_0) = \left[\mathbf{h}_p^r\right]_i \left[\mathbf{h}_q^s\right]_i$ for $r, s \leq t+1$ and $p \neq q$, we have

$$\left\langle \mathbf{h}_p^r, \mathbf{h}_q^s \right\rangle \overset{\text{a.s.}}{\to} \mathbb{E} Z_p^{r-1} Z_q^{s-1} = 0. \tag{3.182}$$

$\mathcal{H}_{t+1}$ (d) Let

$$\begin{aligned}
&\phi_h(h_{1,i}^1, \cdots, h_{P,i}^{t+1}, v_i^1, \cdots, v_i^t, s_{0,i}) = \\
&h_{p,i}^{t+1} \phi \left(\sum_{q=1}^{P} h_{q,i}^{s+1}, v_i^s, s_{0,i} \right),
\end{aligned} \tag{3.183}$$

using $\mathcal{H}_{t+1}$ (b) and Lemma 8, we have

$$\begin{aligned}
&\lim_{N\to\infty}\left\langle \mathbf{h}_p^{t+1}, \boldsymbol{\phi}\left(\sum_{q=1}^{P}\mathbf{h}_q^{s+1}, \mathbf{v}^s, \mathbf{s}_0\right)\right\rangle \\
&\overset{\text{a.s.}}{=} \mathbb{E}\{\sqrt{\omega_p}\tau_t Z_p^t \phi(\sum_{q=1}^{P}\sqrt{\omega_q}\tau_q^s Z_q^s, V_s, S_0)\} \\
&\overset{\text{a.s.}}{=} \omega_p \mathrm{Cov}(\tau_t Z_p^t, \tau_s Z_q^s)\mathbb{E}\{\phi'(\tau_s Z_s, V_s, S_0)\}.
\end{aligned} \tag{3.184}$$

On the other hand, let

$$\phi_h(h_{1,i}^1, \cdots, h_{P,i}^{t+1}, v_i^1, \cdots, v_i^t, s_{0,i}) = h_{p,i}^{t+1} h_{p,i}^{s+1}, \tag{3.185}$$

we have

$$\begin{aligned}
&\lim_{N\to\infty}\left\langle \mathbf{h}_p^{t+1}, \mathbf{h}_p^{s+1}\right\rangle \overset{\text{a.s.}}{=} \mathbb{E}\{\sqrt{\omega_p}\tau_t Z_p^t \sqrt{\omega_p}\tau_s Z_p^s\} \\
&= \omega_p \mathrm{Cov}(\tau_t Z_p^t, \tau_s Z_p^s).
\end{aligned} \tag{3.186}$$

Since empirical distribution of $(\sum_{q=1}^{P}\mathbf{h}_q^{s+1}, \mathbf{v}^s, \mathbf{s}_0) \to (\tau_s Z_s, V_s, S_0)$, applying Lemma 5 in [58], we have

$$\phi'\left(\sum_{q=1}^{P}\mathbf{h}_q^{s+1}, \mathbf{v}^s, \mathbf{s}_0\right) \overset{\text{a.s.}}{=} \mathbb{E}\{\phi'(\tau_s Z_s, V_s, S_0)\}.$$

Therefore

$$\begin{aligned}
&\lim_{N\to\infty}\left\langle \mathbf{h}_p^{t+1}, \phi(\sum_{q=1}^{P}\mathbf{h}_q^{s+1}, \mathbf{v}^s, \mathbf{s}_0)\right\rangle \overset{\text{a.s.}}{=} \\
&\left\langle \mathbf{h}_p^{t+1}, \mathbf{h}_p^{s+1}\right\rangle \left\langle \boldsymbol{\phi}'\left(\sum_{q=1}^{P}\mathbf{h}_q^{s+1}, \mathbf{v}^s, \mathbf{s}_0\right)\right\rangle.
\end{aligned}$$

$\mathcal{H}_{t+1}$ (f) Using (3.174), $\mathcal{H}_r$ (f) for $r \leq t-1$ and Corollary 2,

$$\begin{aligned}
&\left\langle \mathbf{h}_p^{t+1}, \mathbf{q}^0 \right\rangle | \mathfrak{S}_{t+1,t}^p \overset{d}{=} \\
&\sum_{s=0}^{t-1} \alpha_p^s \left\langle \mathbf{h}_p^{s+1}, \mathbf{q}^0 \right\rangle + \left\langle \mathbf{P}_{\mathbf{Q}^{t+1}}^{\perp} (\widetilde{\mathbf{A}}_p)^T \mathbf{m}_{p,\perp}^t, \mathbf{q}^0 \right\rangle \\
&+ \left\langle \mathbf{Q}^{t+1} \overrightarrow{o}_{t+1}(1), \mathbf{q}^0 \right\rangle \\
&\overset{\text{a.s.}}{\to} \frac{Z \|\mathbf{P}_{\mathbf{Q}^{t+1}}^{\perp} \mathbf{q}^0\| \|\mathbf{m}_{p,\perp}^t\|}{N\sqrt{n}} + \sum_{s=0}^{t} \left\langle \mathbf{q}^s, \mathbf{q}^0 \right\rangle o(1).
\end{aligned} \tag{3.187}$$

Now

$$\begin{aligned}
\frac{\|\mathbf{P}_{\mathbf{Q}^{t+1}}^{\perp} \mathbf{q}^0\| \|\mathbf{m}_{p,\perp}^t\|}{N\sqrt{n}} &\leq \frac{\|\mathbf{q}^0\| \|\mathbf{m}_p^t\|}{N\sqrt{n}} \\
&= \frac{\sqrt{N \left\langle \mathbf{q}^0, \mathbf{q}^0 \right\rangle} \sqrt{n_p \left\langle \mathbf{m}_p^t, \mathbf{m}_p^t \right\rangle}}{N\sqrt{n}} \overset{\text{a.s.}}{\to} 0,
\end{aligned} \tag{3.188}$$

and,

$$\left\langle \mathbf{q}^s, \mathbf{q}^0 \right\rangle \leq \sqrt{\left\langle \mathbf{q}^s, \mathbf{q}^s \right\rangle \left\langle \mathbf{q}^0, \mathbf{q}^0 \right\rangle} < \infty, \tag{3.189}$$

we have

$$\left\langle \mathbf{h}_p^{t+1}, \mathbf{q}^0 \right\rangle \overset{\text{a.s.}}{\to} 0. \tag{3.190}$$

By now we finish the proof of Lemma 4.

3.3 Universality of SE

3.3.1 Families of Distributions Satisfying Lindeberg's Condition

In the previous section we prove that SE holds for DiAMP-G1 where $\mathbf{A}$ consists of i.i.d. $\mathcal{N}(0, 1/n)$ entries. In this section, we will show that the universality of SE can be extended to more general cases, even broader than the class of matrices composed of independent subguassian entries.

First, we introduce Lindeberg Central Limit Theorem (CLT) [82].

Theorem 9 *For n independent (not necessarily identically distributed) random variables $X_1, \cdots, X_n$, with each X_i of finite mean μ_i and σ_i^2, define $s_n = \sqrt{\sum_{i=1}^n \sigma_i^2}$. If Lindeberg's condition is satisfied:*

$$\lim_{n\to\infty} \frac{1}{s_n^2} \sum_{i=1}^{n} \mathbb{E}\left\{|X_i - \mu_i|^2 \mathbb{I}_{(\epsilon s_n, \infty)}(|X_i - \mu_i|)\right\} = 0 \tag{3.191}$$

for any $\epsilon > 0$, then as $n \to \infty$,

$$\frac{1}{s_n} \sum_{i=1}^{n} (X_i - \mu_i) \xrightarrow{d} \mathcal{N}(0, 1). \tag{3.192}$$

The Lindeberg's condition in (3.191) seems very stringent, it yet holds for a broad class of distributions. Here we list two families of distributions that satisfy Lindeberg's condition — one is the so-called subgaussian distribution [83], and another is sublaplacian distribution defined by us.

Definition 1: If the moment generating function (m.g.f.) of a random variable X satisfies

$$M_X(t) = \mathbb{E}(e^{tX}) \leq e^{b^2t^2/2}, \ \forall t, \ \text{for some } b > 0,$$

then X is called ***subgaussian*** or $b-$***subgaussian***.

Examples: The continuous uniform distribution

$$f_X(x) = \frac{1}{2a} \mathbb{I}_{[-a,a]}(x)$$

and the symmetric Bernoulli distribution

$$g_X(x) = \frac{1}{2}\delta(x + a) + \frac{1}{2}\delta(x - a)$$

are both a-subgaussian.

Subgaussian random variables have the following properties:

Proposition 4 *All b-subgaussian random variables have mean* 0 *and variance not greater than* b^2.

Proposition 5 *If random variables* X_1 *and* X_2 *are* b_1, b_2*-subgaussian random variables respectively, then* $c_1X_1 + c_2X_2$ *is* $(|c_1|b_1 + |c_2|b_2)$*-subgaussian. Moreover, if* X_1 *and* X_2 *are independent, then* $c_1X_1 + c_2X_2$ *is* $\sqrt{c_1^2b_1^2 + c_2^2b_2^2}$*-subgaussian.*

Proposition 6 *A random variable* X *is subgaussian if and only if there exists* $b > 0$ *such that*

$$\mathbb{P}(|X| \geq t) \leq 2e^{-\frac{t^2}{2b^2}} \text{ for any } t > 0.$$

Definition 2: If the probability density function (PDF) of a random variable X is a even function, and there exists constants α, $\beta > 0$ such that

$$\mathbb{P}(|X| \geq t) \leq \alpha^2 e^{-t/\beta}, \ \forall t > 0$$

then X is called ***sublaplacian*** or $(\alpha, \beta)-$***sublaplacian***.

Regarding all the sublaplacian random variables, we have the following results.

Proposition 7 *All* (α, β)*-sublapalacian random variables have mean* 0 *and variance not greater than* $2\alpha^2\beta^2$.

Proof: It is trivial to show that all the sublaplacian random variables are zero-mean since they have even PDFs by definition. For the variance of a (α, β)-sublaplacian random variable X, by definition, we have

$$\begin{aligned} &\text{Var}(X) = \mathbb{E}\left\{X^2\right\} = \mathbb{E}\left\{|X|^2\right\} = \\ &\int_0^\infty t^2 \frac{d\mathbb{P}(|X_i| \leq t)}{dt} dt = -\int_0^\infty t^2 \frac{d\mathbb{P}(|X_i| > t)}{dt} dt \\ &= t^2\mathbb{P}(|X_i| > t)|_\infty^0 + \int_0^\infty 2t\mathbb{P}(|X_i| > t) dt. \end{aligned} \tag{3.193}$$

By definition,

$$\mathbb{P}(|X| > t) \leq \alpha^2 e^{-t/\beta}, \tag{3.194}$$

we have

$$\lim_{t\to\infty} t^2 \mathbb{P}(|X| > t) = 0. \tag{3.195}$$

Therefore,

$$\begin{aligned} \mathrm{Var}(X) &= \int_0^\infty 2t\mathbb{P}(|X| > t)dt \\ &\leq \int_0^\infty 2t\alpha^2 e^{-t/\beta}dt = 2\alpha^2\beta^2. \end{aligned} \tag{3.196}$$

Proposition 8 *If X is (α, β)-sublaplacian, then cX is $(\alpha, |c|\beta)$-sublaplacian.*

Proof: Let $Y = cX$, then

$$\mathbb{P}(|Y| \geq t) = \mathbb{P}(|X| \geq t/|c|) \leq \alpha^2 e^{-t/(|c|\beta)}.$$

Therefore, Y is $(\alpha, |c|\beta)$-sublaplacian.

Proposition 9 *For a finite set of distributions $\{f_{X_1}(x), \cdots, f_{X_k}(x)\}$, where $f_{X_i}(x)$ is (α_i, β_i)-sublaplacian, $\forall i \in [k]$, then any mixture of the k distributions is $\left(\max\limits_{i\in[k]}\{\alpha_i\}, \max\limits_{i\in[k]}\{\beta_i\}\right)$-sublaplacian.*

Proof: Let $f_X(x) = \sum_{i=1}^k p_i f_{X_i}(x)$ be any mixture, where $p_i \geq 0$, $\forall i$ and $\sum_{i=1}^k p_i = 1$. Then we have

$$\begin{aligned} \mathbb{P}(|X| \geq t) &= \sum_{i=1}^k p_i \mathbb{P}(|X_i| \geq t) \\ &\leq \sum_{i=1}^k p_i \alpha_i^2 e^{-t/\beta_i} \leq \left[\max_{i\in[k]}\{\alpha_i\}\right]^2 e^{-t\Big/\left[\max\limits_{i\in[k]}\{\beta_i\}\right]}. \end{aligned} \tag{3.197}$$

Proposition 10 *If X has an even PDF $f_X(x)$, and there exists α, $\beta > 0$ such that*

$$f_X(x) \leq \frac{\alpha^2}{2\beta} e^{-t/\beta}, \ \forall x$$

then X is (α, β)-sublaplacian.

Proof: by the above assumption on $f_X(x)$,

$$\begin{aligned} \mathbb{P}(|X| > t) &= 2\int_t^\infty f_X(x)dx \\ &\leq 2\int_t^\infty \frac{\alpha^2}{2\beta} e^{-t/\beta} dx \leq \alpha^2 e^{-t/\beta}. \end{aligned} \tag{3.198}$$

Therefore, X is (α, β)-sublaplacian.

Examples: The Laplace distribution

$$f_X(x) = \frac{1}{2\beta} e^{-|x|/\beta} \tag{3.199}$$

is $(1, \beta)$-sublaplacian, but it is not subgaussian since

$$\mathbb{P}(|X| \geq t) = e^{-t/\beta},$$

and $\forall b > 0$, as long as $t > \max\{2b^2/\beta, \beta \ln 2\}$, we will have

$$\mathbb{P}(|X| \geq t) > 2e^{-\frac{t^2}{2b^2}},$$

which violates the condition in Proposition 6.

Another example is zero-mean Logistic distribution

$$g_X(x) = \frac{e^{x/\beta}}{\beta(1 + e^{x/\beta})^2}. \tag{3.200}$$

It is $(\sqrt{2}, \beta)$-sublaplacian, since

$$g_X(x) = g_X(|x|) = \frac{e^{|x|/\beta}}{\beta(1 + e^{|x|/\beta})^2} \leq \frac{1}{\beta} e^{-|x|/\beta}.$$

However, it is not subgaussian, since

$$g_X(x) = g_X(|x|) = \frac{e^{|x|/\beta}}{\beta(1 + e^{|x|/\beta})^2} \geq \frac{1}{4\beta} e^{-|x|/\beta},$$

which implies that

$$\mathbb{P}(|X| \geq t) \geq 2\int_t^\infty \frac{1}{4\beta}e^{-x/\beta}dx = \frac{1}{2}e^{-t/\beta},$$

and contradicts the condition in Proposition 6.

Furthermore, we can plug $\beta = 1/\sqrt{2}$ into $f_X(x)$ in (3.199) and $\beta = 3/\sqrt{\pi}$ into $g_X(x)$ in (3.200), which makes both distributions with unit variance. According to Proposition 9, their mixture $pf_X(x) + (1-p)g_X(x)$ for any $p \in [0,1]$ is $(\sqrt{2}, 3/\sqrt{\pi})$-sublaplacian with unit variance.

Lemma 11 *For n independent zero-mean random variables $X_1, \cdots, X_n$, with each X_i either being b-subgaussian, or (α,β)-sublaplacian, if there exists some $\rho \in (0,1]$ such that $Var(X_i) = \sigma_i^2 \geq \max\{b^2, 2\alpha^2\beta^2\}\rho$ for any i, then Lindeberg's condition holds.*

Proof of Lemma 11: Define $s_n = \sum_{i=1}^n \sigma_i^2$. We have

$$\begin{aligned}
&\mathbb{E}\left\{X_i^2 \mathbb{I}_{(\epsilon s_n,\infty)}(|X_i|)\right\} \\
&= \int_0^\infty t^2 \mathbb{I}_{(\epsilon s_n,\infty)}(t)\frac{d\mathbb{P}(|X_i| \leq t)}{dt}dt \\
&= -\int_{\epsilon s_n}^\infty t^2 \frac{d\mathbb{P}(|X_i| > t)}{dt}dt \\
&= t^2\mathbb{P}(|X_i| > t)|_\infty^{\epsilon s_n} + \int_{\epsilon s_n}^\infty 2t\mathbb{P}(|X_i| > t)dt.
\end{aligned} \tag{3.201}$$

Since X_i is either b-subgaussian or (α,β)-subgaussian,

$$\mathbb{P}(|X_i| > t) \leq \max\{2e^{-\frac{t^2}{2b^2}}, \alpha^2 e^{-t/\beta}\}, \tag{3.202}$$

we have

$$\lim_{t\to\infty} t\mathbb{P}(|X_i| > t) = 0, \tag{3.203}$$

$$t^2\mathbb{P}(|X_i| > t)|_{t=\epsilon s_n}^\infty \leq \epsilon^2 s_n^2 \max\{2e^{-\frac{\epsilon^2 s_n^2}{2b^2}}, \alpha^2 e^{-\epsilon s_n/\beta}\}, \tag{3.204}$$

and

$$\begin{aligned}
&\int_{\epsilon s_n}^{\infty} 2t\mathbb{P}(|X_i| > t)dt \leq \\
&\max\left\{\int_{\epsilon s_n}^{\infty} 4te^{-\frac{t^2}{2b^2}}dt, \int_{\epsilon s_n}^{\infty} 2t\alpha^2 e^{-t/\beta}dt\right\} \\
&= \max\left\{4b^2 e^{-\frac{\epsilon^2 s_n^2}{2b^2}}, 2\alpha^2\beta(\epsilon s_n + \beta)e^{-\epsilon s_n/\beta}\right\}.
\end{aligned} \tag{3.205}$$

Therefore,

$$\begin{aligned}
&\frac{1}{s_n^2}\sum_{i=1}^{n}\mathbb{E}\left\{X_i^2\mathbb{I}_{(\epsilon s_n,\infty)}(|X_i|)\right\} \\
&\leq n\epsilon^2\max\{2e^{-\frac{t^2}{2b^2}}, \alpha^2 e^{-t/\beta}\} \\
&+ \frac{n}{s_n^2}\max\left\{4b^2 e^{-\frac{\epsilon^2 s_n^2}{2b^2}}, 2\alpha^2\beta(\epsilon s_n + \beta)e^{-\epsilon s_n/\beta}\right\}.
\end{aligned} \tag{3.206}$$

Note that $\sigma_i^2 \geq \max\{b^2, 2\alpha^2\beta^2\}\rho$ implying that

$$\frac{b^2}{s_n^2} = \frac{b^2}{\sum_{i=1}^n \sigma_i^2} \leq \frac{1}{\rho n}, \tag{3.207}$$

and

$$\frac{2\alpha^2\beta^2}{s_n^2} = \frac{2\alpha^2\beta^2}{\sum_{i=1}^n \sigma_i^2} \leq \frac{1}{\rho n}, \tag{3.208}$$

i.e.,

$$s_n^2/b^2 = O(n), \text{ and } s_n/\beta = O(n^{1/2}). \tag{3.209}$$

Since either e^{-n} and $e^{-n^{1/2}}$ decays to 0 faster than any polynomial of n increases to ∞, we have

$$\frac{1}{s_n^2}\sum_{i=1}^{n}\mathbb{E}\left\{X_i^2\mathbb{I}_{(\epsilon s_n,\infty)}(|X_i|)\right\} \to 0 \text{ as } n \to \infty, \tag{3.210}$$

that is, Lindeberg's condition holds.

Remarks: The condition $\sigma_i^2 \geq b^2\rho$ or $\sigma_i^2 \geq 2\alpha^2\beta^2\rho$ is actually a natural property for many subgaussian and sublaplacian distributions. For example, the continuous

uniform distribution over $[-a, a]$ is a-subgaussian with variance $\sigma^2 = a^2/3$, where $\rho = 1/3$, and the symmetric Bernoulli distribution over $\{-a, a\}$ is also a-subgaussian with variance $\sigma^2 = a^2$, where $\rho = 1$. For the Laplace distribution in (3.199), $\sigma^2 = 2\beta^2$ while $2\alpha^2\beta^2 = 2\beta^2$, $\rho = 1$, and for the logistic distribution in (3.200), $\sigma^2 = \beta^2\pi^2/3$ while $2\alpha^2\beta^2 = 4\beta^2$, $\rho = \pi^2/12$.

3.3.2 Augmenting Technique to Prove SE's Universality

Now we can extend the theoretical results of SE for sensing matrices with i.i.d. Gaussian entries to more general cases. Specifically, we have the following lemma. This is done by a augmenting technique proposed by us.

Lemma 12 *In Lemma 4, if all the entries in sensing matrix* $\mathbf{A}$ *is changed to be independent (not necessarily identically distributed) random variables with mean* 0 *and variance* $1/n$*, and the elements in each row or column satisfy Lindeberg's condition, then all of the conclusions in Lemma 4 hold with the almost-sure convergence being replaced by convergence in distribution.*

Since the core of the proof of SE is the distribution of $\mathbf{A}_p$ conditioning on $\mathfrak{S}_{t_1,t_2}^p$, we need to first prove that $\mathbf{A}_p|\mathfrak{S}_{t_1,t_2}^p$ remains the same as that for $\mathbf{A}$ with i.i.d. Gaussian entries in the large system limit, which is the following lemma:

Lemma 13 *If* $\mathbf{A}$ *satisfies the condition in Lemma 12, then*

$$\begin{aligned}\mathbf{A}_p|\mathfrak{S}_{t_1,t_2}^p \xrightarrow{d} & \left[\mathbf{X}_p^{t_2}(\mathbf{M}_p^{t_2})^\dagger\right]^T + \mathbf{P}_{\mathbf{M}_p^{t_1}}^{\perp}\mathbf{Y}_p^{t_1}(\mathbf{Q}^{t_1})^\dagger \\ & + \mathbf{P}_{\mathbf{M}_p^{t_2}}^{\perp}\widetilde{\mathbf{A}}_p\mathbf{P}_{\mathbf{Q}^{t_1}}^{\perp} = \mathbf{Y}_p^{t_1}(\mathbf{Q}^{t_1})^\dagger \\ & + \left[\mathbf{X}_p^{t_2}(\mathbf{M}_p^{t_2})^\dagger\right]^T\mathbf{P}_{\mathbf{Q}^{t_1}}^{\perp} + \mathbf{P}_{\mathbf{M}_p^{t_2}}^{\perp}\widetilde{\mathbf{A}}_p\mathbf{P}_{\mathbf{Q}^{t_1}}^{\perp},\end{aligned}$$

where $\widetilde{\mathbf{A}}_p$ *consists of i.i.d.* $\mathcal{N}(0, 1/n)$ *entries independent of* $\mathbf{A}_p$*.*

Proof: Let us construct a matrix series $\{\mathbf{P}_k\}_{k\geq 0}$ as follows:

$$\mathbf{P}_k = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{P}_{k-1} & \mathbf{P}_{k-1} \\ \mathbf{P}_{k-1} & -\mathbf{P}_{k-1} \end{bmatrix}, \tag{3.211}$$

with initial value $\mathbf{P}_0 = [1]$. It can be easily verified that $\mathbf{P}_k$ is an orthogonal matrix of order 2^k, and all the elements in $\mathbf{P}_k$ have the same magnitude $2^{-k/2}$.

Consider the case where each row of $\mathbf{A}$ satisfies Lindeberg's condition. Now augment $\mathbf{A}_p$ into a $n_p \times 2^K$ matrix $\mathbf{A}_p^a$, where $2^K > N$:

$$\mathbf{A}_p^a = [\mathbf{A}_p | \mathbf{C}_p], \tag{3.212}$$

where $\mathbf{C}_p$ is independent of $\mathbf{A}_p$, and consists of i.i.d. $\mathcal{N}(0, 1/\sqrt{n})$. Let

$$\mathbf{S}_p = \mathbf{A}_p^a \mathbf{P}_K = \mathbf{A}_p \mathbf{P}_K^A + \mathbf{C}_p \mathbf{P}_K^C, \tag{3.213}$$

where $\mathbf{P}_K^A$ and $\mathbf{P}_K^C$ as the first N rows and the last $R = 2^K - N$ rows of $\mathbf{P}_K$. It is easy to show that

$$\mathbf{P}_K^A (\mathbf{P}_K^A)^T = \mathbf{I}_N, \ \mathbf{P}_K^C (\mathbf{P}_K^C)^T = \mathbf{I}_R, \tag{3.214}$$

$$\mathbf{P}_K^A (\mathbf{P}_K^C)^T = \mathbf{0} \in \mathbb{R}^{N \times R}, \tag{3.215}$$

and

$$\mathbf{A}_p^a = \mathbf{S}_p (\mathbf{P}_K)^T, \ \mathbf{A}_p = \mathbf{S}_p (\mathbf{P}_K^A)^T, \ \mathbf{C}_p = \mathbf{S}_p (\mathbf{P}_K^C)^T. \tag{3.216}$$

Using (3.216), then $\mathbf{h}_p^{t+1}$ and $\mathbf{b}_p^t$ in (3.216) become

$$\begin{aligned} \mathbf{h}_p^{t+1} &= \mathbf{P}_K^A \mathbf{S}_p^T \mathbf{m}_p^t - \xi_p^t \omega_p \mathbf{q}^t, \\ \mathbf{b}_p^t &= \mathbf{S}_p (\mathbf{P}_K^A)^T \mathbf{q}^t - \lambda_t \mathbf{m}_p^{t-1}, \end{aligned} \tag{3.217}$$

and the linear constraints in (3.39) can be written as

$$\mathbf{P}_K^A \mathbf{S}_p^T \mathbf{M}_p^{t_2} = \mathbf{X}_p^{t_2}, \mathbf{Y}_p^{t_1} = \mathbf{S}_p (\mathbf{P}_K^A)^T \mathbf{Q}^{t_1}. \tag{3.218}$$

The element in the i-th row and j-th column of $\mathbf{S}_p$ is

$$\begin{aligned} s_{i,j}^p &= \sum_{k=1}^{2^K} \left[\mathbf{A}_p^a\right]_{i,k} \left[\mathbf{P}_K\right]_{k,j} \\ &= \sum_{k=1}^{N} \left[\mathbf{A}_p\right]_{i,k} \left[\mathbf{P}_K\right]_{k,j} + \sum_{k=N+1}^{2^K} \left[\mathbf{C}_p\right]_{i,k-N} \left[\mathbf{P}_K\right]_{k,j}. \end{aligned} \tag{3.219}$$

Since the N random variables $\left[\mathbf{A}_p\right]_{i,1}, \cdots, \left[\mathbf{A}_p\right]_{i,N}$ satisfy Lindeberg's condition, we have

$$\lim_{N\to\infty} \frac{1}{N/n} \sum_{k=1}^{N} \mathbb{E}\left\{ \left[\mathbf{A}_p\right]_{i,k}^2 \mathbb{I}_{(\epsilon\sqrt{N/n},\infty)}(|\left[\mathbf{A}_p\right]_{i,k}|) \right\} = 0,$$

for any $\epsilon > 0$. Noting that $|\left[\mathbf{P}_K\right]_{k,j}| = 2^{-K/2}$ for all i and k, it is easy to verify that $\left[\mathbf{A}_p\right]_{i,k} \left[\mathbf{P}_K\right]_{k,j}$ for $k = 1, \cdots, N$ also satisfy Lindeberg's condition. Therefore, we can apply Lindeberg Central Limit Theorem (CLT) to obtain

$$\sum_{k=1}^{N} \left[\mathbf{A}_p\right]_{i,k} \left[\mathbf{P}_K\right]_{k,j} \xrightarrow{d} \mathcal{N}\left(0, \frac{N}{2^K}\frac{1}{n}\right). \tag{3.220}$$

Now consider the rest R random variables $\left[\mathbf{C}_p\right]_{i,k-N}$. Since they are i.i.d. $\mathcal{N}(0, 1/n)$, it is easy to show that

$$\sum_{k=N+1}^{2^K} \left[\mathbf{C}_p\right]_{i,k-N} \left[\mathbf{P}_K\right]_{k,j} \xrightarrow{d} \mathcal{N}\left(0, \frac{R}{2^K}\frac{1}{n}\right). \tag{3.221}$$

Therefore

$$s_{i,j}^p \xrightarrow{d} \mathcal{N}(0, 1/n), \text{ as } K \to \infty, \ \forall i, j. \tag{3.222}$$

Using the independence of entries in $\mathbf{A}_p^a$ and the orthogonality of $\mathbf{P}_K$, we can further show that

$$\mathrm{Cov}(s_{i_1,j_1}^p, s_{i_2,j_2}^p) = \frac{1}{n}\delta_{i_1,i_2}\delta_{j_1,j_2}, \ \forall i_1, i_2, j_1, j_2. \tag{3.223}$$

Therefore

$$\mathbf{S}_p \xrightarrow{d} \mathbf{S}_p^G, \text{ as } K \to \infty, \tag{3.224}$$

where $\mathbf{S}_p^G$ consists of i.i.d. $\mathcal{N}(0, 1/n)$ entries.

We can apply Lemma 11 in [58] to obtain

$$\mathbf{S}_p|\mathfrak{S}_{t_1,t_2}^p \xrightarrow{d} \mathbf{L}_{t_1,t_2}^p + \mathcal{P}_{t_1,t_2}(\widetilde{\mathbf{S}}_p^G), \tag{3.225}$$

where $\mathbf{L}_{t_1,t_2}^p$ is the least-square-solution of

$$\arg\min_{\mathbf{S}} \|\mathbf{S}\|_F^2, \text{ s.t. } \mathbf{P}_K^A \mathbf{S}^T \mathbf{M}_p^{t_2} = \mathbf{X}_p^{t_2}, \mathbf{Y}_p^{t_1} = \mathbf{S}(\mathbf{P}_K^A)^T \mathbf{Q}^{t_1},$$

$\widetilde{\mathbf{S}}_p^G$ consists of i.i.d. $\mathcal{N}(0, 1/n)$ entries independent of $\mathbf{S}_p$, and $\mathcal{P}_{t_1,t_2}$ is the orthogonal projector onto the following subspace

$$\mathcal{S} = \{\mathbf{S} : \mathbf{P}_K^A \mathbf{S}^T \mathbf{M}_p^{t_2} = \mathbf{0}, \mathbf{S}(\mathbf{P}_K^A)^T \mathbf{Q}^{t_1} = \mathbf{0}\}. \tag{3.226}$$

The way of obtaining $\mathbf{L}_{t_1,t_2}^p$ and $\mathcal{P}_{t_1,t_2}(\widetilde{\mathbf{S}}_p^G)$ is similar to that of conditional distribution of $\mathbf{A}$ in [58]. We will still present the process, yet in a way easier for readers to follow.

Obtain $\mathbf{L}_{t_1,t_2}^p$: Write the Lagrangian

$$\begin{aligned} J(\mathbf{S}, \mathbf{\Theta}, \mathbf{\Gamma}) &= \|\mathbf{S}\|_F^2 + tr\left\{\mathbf{\Theta}^T \left[\mathbf{Y}_p^{t_1} - \mathbf{S}(\mathbf{P}_K^A)^T \mathbf{Q}^{t_1}\right]\right\} \\ &+ tr\left\{\mathbf{\Gamma}^T \left[\mathbf{X}_p^{t_2} - \mathbf{P}_K^A \mathbf{S}^T \mathbf{M}_p^{t_2}\right]\right\}, \end{aligned}$$

where $\mathbf{\Theta}$ and $\mathbf{\Gamma}$ are Lagrangian multipliers to be determined, $tr\{\cdot\}$ denotes the trace of a matrix, and $tr\{\mathbf{A}^T\mathbf{B}\}$ is a well-defined inner product for vector spaces $\mathcal{V}$ where all the elements are matrices.

Note that

$$\begin{aligned} &tr\{\mathbf{\Theta}^T \mathbf{S}(\mathbf{P}_K^A)^T \mathbf{Q}^{t_1}\} = tr\{(\mathbf{P}_K^A)^T \mathbf{Q}^{t_1} \mathbf{\Theta}^T \mathbf{S}\} \\ &= tr\{\mathbf{S}^T \mathbf{\Theta} (\mathbf{Q}^{t_1})^T \mathbf{P}_K^A\}, \end{aligned} \tag{3.227}$$

and

$$tr\{\mathbf{\Gamma}^T\mathbf{P}_K^A\mathbf{S}^T\mathbf{M}_p^{t_2}\} = tr\{\mathbf{S}^T\mathbf{M}_p^{t_2}\mathbf{\Gamma}^T\mathbf{P}_K^A\} \tag{3.228}$$

Applying the KKT condition, we have

$$\nabla_{\mathbf{S}} J(\mathbf{S}, \mathbf{\Theta}, \mathbf{\Gamma}) = 2\mathbf{S} - \left[\mathbf{\Theta}(\mathbf{Q}^{t_1})^T + \mathbf{M}_p^{t_2}\mathbf{\Gamma}^T\right]\mathbf{P}_K^A = \mathbf{0}. \tag{3.229}$$

Post-multiplying the left hand side of (3.229) by $(\mathbf{P}_K^A)^T$, according to (3.214), we get

$$2\mathbf{S}(\mathbf{P}_K^A)^T = \mathbf{\Theta}(\mathbf{Q}^{t_1})^T + \mathbf{M}_p^{t_2}\mathbf{\Gamma}^T. \tag{3.230}$$

Post-multiplying both sides of (3.230) by $\mathbf{Q}^{t_1}$, plugging in the linear constraints in (3.218), we get

$$\mathbf{\Theta}(\mathbf{Q}^{t_1})^T\mathbf{Q}^{t_1} + \mathbf{M}_p^{t_2}\mathbf{\Gamma}^T\mathbf{Q}^{t_1} = 2\mathbf{Y}_p^{t_1}. \tag{3.231}$$

Take transposition and then post-multiply $\mathbf{M}_p^{t_2}$ on both sides of (3.230), plug in the linear constraints in (3.218) again, we have

$$\mathbf{Q}^{t_1}\mathbf{\Theta}^T\mathbf{M}_p^{t_2} + \mathbf{\Gamma}(\mathbf{M}_p^{t_2})^T\mathbf{M}_p^{t_2} = 2\mathbf{X}_p^{t_2}. \tag{3.232}$$

From (3.231) we can get

$$\mathbf{\Theta} = 2\mathbf{Y}_p^{t_1}\left[(\mathbf{Q}^{t_1})^T\mathbf{Q}^{t_1}\right]^{-1} - \mathbf{M}_p^{t_2}\left[(\mathbf{Q}^{t_1})^\dagger\mathbf{\Gamma}\right]^T. \tag{3.233}$$

Plugging it in (3.232), we have

$$\mathbf{P}_{\mathbf{Q}^{t_1}}^{\perp}\mathbf{\Gamma} = 2\left\{\mathbf{X}_p^{t_2} - \left[(\mathbf{M}_p^{t_2})^T\mathbf{Y}_p^{t_1}(\mathbf{Q}^{t_1})^\dagger\right]^T\right\}\left[(\mathbf{M}_p^{t_2})^T\mathbf{M}_p^{t_2}\right]^{-1}. \tag{3.234}$$

Plugging (3.40) in, the right hand side becomes

$$\begin{aligned}
&2\left\{\mathbf{X}_p^{t_2} - \left[(\mathbf{M}_p^{t_2})^T\mathbf{Y}_p^{t_1}(\mathbf{Q}^{t_1})^\dagger\right]^T\right\}\left[(\mathbf{M}_p^{t_2})^T\mathbf{M}_p^{t_2}\right]^{-1} \\
&= 2\left\{\mathbf{X}_p^{t_2} - \left[(\mathbf{X}_p^{t_2})^T\mathbf{Q}^{t_1}(\mathbf{Q}^{t_1})^\dagger\right]^T\right\}\left[(\mathbf{M}_p^{t_2})^T\mathbf{M}_p^{t_2}\right]^{-1} \\
&= 2\mathbf{P}^\perp_{\mathbf{Q}^{t_1}}\mathbf{X}_p^{t_2}\left[(\mathbf{M}_p^{t_2})^T\mathbf{M}_p^{t_2}\right]^{-1}.
\end{aligned} \tag{3.235}$$

$\mathbf{\Gamma}$ has infinitely many solutions since $\mathbf{P}^\perp_{\mathbf{Q}^{t_1}}$ is not full rank. Nevertheless, we can take the most straightforward one

$$\mathbf{\Gamma} = 2\mathbf{X}_p^{t_2}\left[(\mathbf{M}_p^{t_2})^T\mathbf{M}_p^{t_2}\right]^{-1}. \tag{3.236}$$

Plugging it back to (3.233), we obtain

$$\mathbf{\Theta} = 2\mathbf{P}^\perp_{\mathbf{M}_p^{t_2}}\mathbf{Y}_p^{t_1}\left[(\mathbf{Q}^{t_1})^T\mathbf{Q}^{t_1}\right]^{-1}. \tag{3.237}$$

Plugging (3.236) and (3.233) in (3.229), we finally obtain

$$\begin{aligned}
\mathbf{L}^p_{t_1,t_2} &= \left\{\mathbf{P}^\perp_{\mathbf{M}_p^{t_2}}\mathbf{Y}_p^{t_1}(\mathbf{Q}^{t_1})^\dagger + \left[\mathbf{X}_p^{t_2}(\mathbf{M}_p^{t_2})^\dagger\right]^T\right\}\mathbf{P}_K^A \\
&= \left\{\mathbf{Y}_p^{t_1}(\mathbf{Q}^{t_1})^\dagger + \left[\mathbf{X}_p^{t_2}(\mathbf{M}_p^{t_2})^\dagger\right]^T\mathbf{P}^\perp_{\mathbf{Q}^{t_1}}\right\}\mathbf{P}_K^A.
\end{aligned} \tag{3.238}$$

Obtaining $\mathcal{P}_{t_1,t_2}(\widetilde{\mathbf{S}}_p^G)$: First, we try to construct a linear operator $\mathcal{P}: \mathbb{R}^{n_p\times 2^K} \to \mathcal{S}$, where $\mathcal{S}$ is the subspace described in (3.226).

$\forall\ \mathbf{S} \in \mathcal{S}$, we have $\mathbf{P}_K^A\mathbf{S}^T\mathbf{M}_p^{t_2} = \mathbf{0}$ and $\mathbf{S}(\mathbf{P}_K^A)^T\mathbf{Q}^{t_1} = \mathbf{0}$. Now we want to find such a $\mathbf{S}$.

Let us consider $\mathbf{P}_K^A\mathbf{S}^T\mathbf{M}_p^{t_2} = \mathbf{0}$ only. It is easy to show that for any $\mathbf{V}_1 \in \mathbb{R}^{n_p\times N}$, all the $\mathbf{S}$ satisfying $\mathbf{S}(\mathbf{P}_K^A)^T = \mathbf{P}^\perp_{\mathbf{M}_p^{t_2}}\mathbf{V}_1$ will suffice for $\mathbf{P}_K^A\mathbf{S}^T\mathbf{M}_p^{t_2} = \mathbf{0}$.

Let us consider $\mathbf{S}(\mathbf{P}_K^A)^T\mathbf{Q}^{t_1} = \mathbf{0}$ only, It is easy to show that for any $\mathbf{V}_2 \in \mathbb{R}^{n_p\times N}$, all the $\mathbf{S}$ satisfying $\mathbf{S}(\mathbf{P}_K^A)^T = \mathbf{V}_2\mathbf{P}^\perp_{\mathbf{Q}^{t_1}}$ will suffice for $\mathbf{S}(\mathbf{P}_K^A)^T\mathbf{Q}^{t_1} = \mathbf{0}$.

Now, if we want $\mathbf{S}$ to satisfy both, then $\mathbf{V}_1$ and $\mathbf{V}_2$ are coupled though $\mathbf{P}^\perp_{\mathbf{M}_p^{t_2}}\mathbf{V}_1 = \mathbf{V}_2\mathbf{P}^\perp_{\mathbf{Q}^{t_1}}$. To decouple, we pick up another arbitrary $\mathbf{V}_A \in \mathbb{R}^{n_p\times N}$, and let $\mathbf{V}_1 =$

$\mathbf{V}_A\mathbf{P}^{\perp}_{\mathbf{Q}^{t_1}}$ and $\mathbf{V}_2 = \mathbf{P}^{\perp}_{\mathbf{M}_p^{t_2}}\mathbf{V}_A$, then it is easy to show that $\mathbf{P}^{\perp}_{\mathbf{M}_p^{t_2}}\mathbf{V}_1 = \mathbf{V}_2\mathbf{P}^{\perp}_{\mathbf{Q}^{t_1}} = \mathbf{P}^{\perp}_{\mathbf{M}_p^{t_2}}\mathbf{V}_A\mathbf{P}^{\perp}_{\mathbf{Q}^{t_1}}$. In other words, $\forall\ \mathbf{V}_A \in \mathbb{R}^{n_p\times N}$, any $\mathbf{S}$ that satisfies

$$\mathbf{S}(\mathbf{P}_K^A)^T = \mathbf{P}^{\perp}_{\mathbf{M}_p^{t_2}}\mathbf{V}_A\mathbf{P}^{\perp}_{\mathbf{Q}^{t_1}}$$

will suffice for $\mathbf{S} \in \mathcal{S}$, and this condition is equivalent to

$$\begin{aligned}&\mathbf{S} = \mathbf{P}^{\perp}_{\mathbf{M}_p^{t_2}}\mathbf{V}_A\mathbf{P}^{\perp}_{\mathbf{Q}^{t_1}}\mathbf{P}_K^A + \mathbf{V}_c\mathbf{P}_K^C, \forall\ \mathbf{V}_A \in \mathbb{R}^{n_p\times N},\\ &\forall\ \mathbf{V}_c \in \mathbb{R}^{n_p\times R},\ \text{where } R = 2^K - N.\end{aligned} \tag{3.239}$$

Since $\mathbf{V}_A$ and $\mathbf{V}_C$ are arbitrary, they form an arbitrary $\mathbf{V} = [\mathbf{V}_A|\mathbf{V}_C]\,\mathbf{P}_K \in \mathbb{R}^{n_p\times 2^K}$ with $\mathbf{V}_A = \mathbf{V}(\mathbf{P}_K^A)^T$ and $\mathbf{V}_C = \mathbf{V}(\mathbf{P}_K^C)^T$. Plugging them back, we get the following operator $\mathcal{P}_{t_1,t_2}(\mathbf{V})$:

$$\mathcal{P}_{t_1,t_2}(\mathbf{V}) = \mathbf{P}^{\perp}_{\mathbf{M}_p^{t_2}}\mathbf{V}(\mathbf{P}_K^A)^T\mathbf{P}^{\perp}_{\mathbf{Q}^{t_1}}\mathbf{P}_K^A + \mathbf{V}(\mathbf{P}_K^C)^T\mathbf{P}_K^C. \tag{3.240}$$

To show that $\mathcal{P} \triangleq \mathcal{P}_{t_1,t_2}$ is indeed the projector onto $\mathcal{S}$, we need to verify that

a) $\forall\ \mathbf{V} \in \mathbb{R}^{n_p\times 2^K}$, $\mathcal{P}(\mathbf{V}) \in \mathcal{S}$, i.e., projection of any matrix should not go beyond $\mathcal{S}$.

b) $\mathcal{P}(\mathbf{S}) = \mathbf{S}$ for any $\mathbf{S} \in \mathcal{S}$.

c) $\forall\ \mathbf{S} \in \mathcal{S}$, $\exists \mathbf{V} \in \mathbb{R}^{n_p\times 2^K}$, s.t. $\mathcal{P}(\mathbf{V}) = \mathbf{S}$.

d) $\mathcal{P}\circ\mathcal{P} = \mathcal{P}$, i.e., projection of projection should remain the same.

e) $\forall\ \mathbf{V} \in \mathbb{R}^{n_p\times 2^K}$, $\forall\ \mathbf{S} \in \mathcal{S}$, $tr\left\{\mathbf{S}^T\left[\mathbf{V} - \mathcal{P}(\mathbf{V})\right]\right\} = 0$. Geometrically, this can be interpreted as $(\mathbf{V} - \mathcal{P}(\mathbf{V}))\perp\mathcal{S}$, i.e., projection should be orthogonal.

Verification of Properties (a) $\sim$ (e) of $\mathcal{P}$:

(a) $\forall\ \mathbf{V} \in \mathbb{R}^{n_p\times 2^K}$, we have

$$\begin{aligned}&\mathbf{P}_K^A\mathcal{P}(\mathbf{V})^T\mathbf{M}_p^{t_2} =\\ &\mathbf{P}_K^A\left[\mathbf{P}^{\perp}_{\mathbf{M}_p^{t_2}}\mathbf{V}(\mathbf{P}_K^A)^T\mathbf{P}^{\perp}_{\mathbf{Q}^{t_1}}\mathbf{P}_K^A + \mathbf{V}(\mathbf{P}_K^C)^T\mathbf{P}_K^C\right]^T\mathbf{M}_p^{t_2} = \mathbf{0},\end{aligned}$$

and

$$\mathcal{P}(\mathbf{V})(\mathbf{P}_K^A)^T\mathbf{Q}^{t_1} =$$
$$\left[\mathbf{P}^{\perp}_{\mathbf{M}_p^{t_2}}\mathbf{V}(\mathbf{P}_K^A)^T\mathbf{P}^{\perp}_{\mathbf{Q}^{t_1}}\mathbf{P}_K^A + \mathbf{V}(\mathbf{P}_K^C)^T\mathbf{P}_K^C\right](\mathbf{P}_K^A)^T\mathbf{Q}^{t_1} = \mathbf{0}.$$

So $\mathcal{P}(\mathbf{V}) \in \mathcal{S}$.

(b) $\forall \mathbf{S} \in \mathcal{S}$, we have $\mathbf{P}_K^A\mathbf{S}^T\mathbf{M}_p^{t_2} = \mathbf{0}, \mathbf{S}(\mathbf{P}_K^A)^T\mathbf{Q}^{t_1} = \mathbf{0}$,

$$\begin{aligned}
\mathcal{P}(\mathbf{S}) &= \mathbf{P}^{\perp}_{\mathbf{M}_p^{t_2}}\mathbf{S}(\mathbf{P}_K^A)^T\mathbf{P}^{\perp}_{\mathbf{Q}^{t_1}}\mathbf{P}_K^A + \mathbf{S}(\mathbf{P}_K^C)^T\mathbf{P}_K^C \\
&= \mathbf{P}^{\perp}_{\mathbf{M}_p^{t_2}}\mathbf{S}(\mathbf{P}_K^A)^T\left(\mathbf{I} - \mathbf{P}_{\mathbf{Q}^{t_1}}\right)\mathbf{P}_K^A + \mathbf{S}(\mathbf{P}_K^C)^T\mathbf{P}_K^C \\
&= \left(\mathbf{I} - \mathbf{P}_{\mathbf{M}_p^{t_2}}\right)\mathbf{S}(\mathbf{P}_K^A)^T\mathbf{P}_K^A + \mathbf{S}(\mathbf{P}_K^C)^T\mathbf{P}_K^C \\
&= \mathbf{S}(\mathbf{P}_K^A)^T\mathbf{P}_K^A + \mathbf{S}(\mathbf{P}_K^C)^T\mathbf{P}_K^C = \mathbf{S}.
\end{aligned}$$

(c) is a direct result of (b) by plugging $\mathbf{V} = \mathbf{S}$.

(d) $\forall\ \mathbf{V} \in \mathbb{R}^{n_p \times 2^K}$, applying (a), we have $\mathcal{P}(\mathbf{V}) \in \mathcal{S}$. Denote $\mathcal{P}(\mathbf{V}) = \mathbf{S}$, apply (b), we have $\mathcal{P} \circ \mathcal{P}(\mathbf{V}) = \mathcal{P}(\mathbf{S}) = \mathbf{S} = \mathcal{P}(\mathbf{V})$.

(e) $\forall\ \mathbf{V} \in \mathbb{R}^{n_p \times 2^K}$,

$$\begin{aligned}
&\mathbf{V} - \mathcal{P}(\mathbf{V}) = \\
&\mathbf{V} - \mathbf{P}^{\perp}_{\mathbf{M}_p^{t_2}}\mathbf{V}(\mathbf{P}_K^A)^T\mathbf{P}^{\perp}_{\mathbf{Q}^{t_1}}\mathbf{P}_K^A - \mathbf{V}(\mathbf{P}_K^C)^T\mathbf{P}_K^C \\
&= \mathbf{V}(\mathbf{P}_K^A)^T\mathbf{P}_K^A - \mathbf{P}^{\perp}_{\mathbf{M}_p^{t_2}}\mathbf{V}(\mathbf{P}_K^A)^T\mathbf{P}^{\perp}_{\mathbf{Q}^{t_1}}\mathbf{P}_K^A \qquad (3.241) \\
&= \mathbf{V}(\mathbf{P}_K^A)^T\mathbf{P}_K^A - (\mathbf{I} - \mathbf{P}_{\mathbf{M}_p^{t_2}})\mathbf{V}(\mathbf{P}_K^A)^T\mathbf{P}^{\perp}_{\mathbf{Q}^{t_1}}\mathbf{P}_K^A \\
&= \mathbf{V}(\mathbf{P}_K^A)^T\mathbf{P}_{\mathbf{Q}^{t_1}}\mathbf{P}_K^A + \mathbf{P}_{\mathbf{M}_p^{t_2}}\mathbf{V}(\mathbf{P}_K^A)^T\mathbf{P}^{\perp}_{\mathbf{Q}^{t_1}}\mathbf{P}_K^A.
\end{aligned}$$

Therefore, $\forall\ \mathbf{S} \in \mathcal{S}$, we have

$$\begin{aligned}
&tr\left\{\mathbf{S}^T\left[\mathbf{V} - \mathcal{P}(\mathbf{V})\right]\right\} \\
&= tr\left\{\mathbf{S}^T\mathbf{V}(\mathbf{P}_K^A)^T\mathbf{P}_{\mathbf{Q}^{t_1}}\mathbf{P}_K^A\right\} \qquad (3.242) \\
&+ tr\left\{\mathbf{S}^T\mathbf{P}_{\mathbf{M}_p^{t_2}}\mathbf{V}(\mathbf{P}_K^A)^T\mathbf{P}^{\perp}_{\mathbf{Q}^{t_1}}\mathbf{P}_K^A\right\}.
\end{aligned}$$

Since $\mathbf{S} \in \mathcal{S}$, $\mathbf{P}_K^A \mathbf{S}^T \mathbf{M}_p^{t_2} = \mathbf{0}, \mathbf{S}(\mathbf{P}_K^A)^T \mathbf{Q}^{t_1} = \mathbf{0}$, we have

$$\begin{aligned} & tr\left\{\mathbf{S}^T \mathbf{V}(\mathbf{P}_K^A)^T \mathbf{P}_{\mathbf{Q}^{t_1}} \mathbf{P}_K^A\right\} \\ & = tr\left\{\mathbf{V}(\mathbf{P}_K^A)^T \mathbf{P}_{\mathbf{Q}^{t_1}} \mathbf{P}_K^A \mathbf{S}^T\right\} \\ & = tr\left\{\mathbf{V}(\mathbf{P}_K^A)^T \left[\mathbf{S}(\mathbf{P}_K^A)^T \mathbf{Q}^{t_1}(\mathbf{Q}^{t_1})^\dagger\right]^T\right\} = 0, \end{aligned} \tag{3.243}$$

and

$$\begin{aligned} & tr\left\{\mathbf{S}^T \mathbf{P}_{\mathbf{M}_p^{t_2}} \mathbf{V}(\mathbf{P}_K^A)^T \mathbf{P}_{\mathbf{Q}^{t_1}}^\perp \mathbf{P}_K^A\right\} \\ & = tr\left\{\mathbf{P}_K^A \mathbf{S}^T \mathbf{M}_p^{t_2}(\mathbf{M}_p^{t_2})^\dagger \mathbf{V}(\mathbf{P}_K^A)^T \mathbf{P}_{\mathbf{Q}^{t_1}}^\perp\right\} = 0. \end{aligned} \tag{3.244}$$

So

$$tr\left\{\mathbf{S}^T\left[\mathbf{V} - \mathcal{P}(\mathbf{V})\right]\right\} = 0.$$

Now we have shown that

$$\begin{aligned} & \mathbf{S}_p | \mathfrak{S}_{t_1,t_2}^p \xrightarrow{d} \mathbf{L}_{t_1,t_2}^p + \mathcal{P}_{t_1,t_2}(\widetilde{\mathbf{S}}_p^G) = \\ & \left\{\mathbf{P}_{\mathbf{M}_p^{t_2}}^\perp \mathbf{Y}_p^{t_1}(\mathbf{Q}^{t_1})^\dagger + \left[\mathbf{X}_p^{t_2}(\mathbf{M}_p^{t_2})^\dagger\right]^T\right\} \mathbf{P}_K^A + \\ & \mathbf{P}_{\mathbf{M}_p^{t_2}}^\perp \widetilde{\mathbf{S}}_p^G (\mathbf{P}_K^A)^T \mathbf{P}_{\mathbf{Q}^{t_1}}^\perp \mathbf{P}_K^A + \widetilde{\mathbf{S}}_p^G (\mathbf{P}_K^C)^T \mathbf{P}_K^C. \end{aligned} \tag{3.245}$$

Since $\mathbf{A}_p = \mathbf{S}_p(\mathbf{P}_K^A)^T$, we have

$$\begin{aligned} & \mathbf{A}_p | \mathfrak{S}_{t_1,t_2}^p \xrightarrow{d} \mathbf{L}_{t_1,t_2}^p (\mathbf{P}_K^A)^T + \mathcal{P}_{t_1,t_2}(\widetilde{\mathbf{S}}_p^G)(\mathbf{P}_K^A)^T = \\ & \left\{\mathbf{P}_{\mathbf{M}_p^{t_2}}^\perp \mathbf{Y}_p^{t_1}(\mathbf{Q}^{t_1})^\dagger + \left[\mathbf{X}_p^{t_2}(\mathbf{M}_p^{t_2})^\dagger\right]^T\right\} \mathbf{P}_K^A (\mathbf{P}_K^A)^T + \\ & \mathbf{P}_{\mathbf{M}_p^{t_2}}^\perp \widetilde{\mathbf{S}}_p^G (\mathbf{P}_K^A)^T \mathbf{P}_{\mathbf{Q}^{t_1}}^\perp \mathbf{P}_K^A (\mathbf{P}_K^A)^T + \widetilde{\mathbf{S}}_p^G (\mathbf{P}_K^C)^T \mathbf{P}_K^C (\mathbf{P}_K^A)^T \\ & = \mathbf{P}_{\mathbf{M}_p^{t_2}}^\perp \mathbf{Y}_p^{t_1}(\mathbf{Q}^{t_1})^\dagger + \left[\mathbf{X}_p^{t_2}(\mathbf{M}_p^{t_2})^\dagger\right]^T + \mathbf{P}_{\mathbf{M}_p^{t_2}}^\perp \widetilde{\mathbf{S}}_p^G (\mathbf{P}_K^A)^T \mathbf{P}_{\mathbf{Q}^{t_1}}^\perp. \end{aligned} \tag{3.246}$$

Noting that $(\mathbf{P}_K^A)^T$ has orthogonal columns, it is easy to show that $\widetilde{\mathbf{A}}_p = \widetilde{\mathbf{S}}_p^G (\mathbf{P}_K^A)^T$ consists of i.i.d. $\mathcal{N}(0, 1/n)$ entries. Now we finish the proof of Lemma 13 for the case where each row of $\mathbf{A}$ satisfies Lindeberg's condition. For the case where each column of $\mathbf{A}$ satisfies Lindeberg's condition, we can augment $\mathbf{A}_p$ from the

row direction into a $2^K \times N$ matrix, where $2^K > n_p$, and apply the same proof to show that Lemma 13 holds.

Proof of Lemma 12: According to Lemma 13, $\mathbf{A}_p|\mathfrak{S}^p_{t_1,t_2}$ will converge in distribution to the same result as the case where $\mathbf{A}$ consists of i.i.d Gaussian entries. None of the assumptions used in the proof of Lemma 4 requires any type of convergences stronger than convergence in distribution, and for all the conclusions obtained during the proof of Lemma 4, there are no converge types weaker than convergence in distribution. Therefore, the same flow of proof works for Lemma 12, by replacing all the almost-sure convergences in the conclusions in Lemma 4 by convergences in distribution (weak convergence).

Corollary 4 *For DiAMP-G1, if all the entries in sensing matrix* $\mathbf{A}$ *are independent zero-mean random variables with variance* $1/n$*, with each one being either* $(b/\sqrt{n})$*-subgaussian, or* $(\alpha, \beta/\sqrt{n})$*-sublaplacian, then all of the conclusions in Lemma 4 hold with the almost-sure convergence being replaced by convergence in distribution.*

It is straightforward to prove Corollary 4, by applying Lemma 11 to show that each row or column of $\mathbf{A}$ satisfies Lindeberg's condition.

Lemma 12 provides a class of sensing matrices for DiAMP-G1 where SE still holds, which is even broader than the set of matrices consisting of independent subgussian random variables with variance $1/n$, as described in Section 3.3.1 and Corollary 4. This greatly extends the universality of SE on the theoretical level. Applying Lemma 4 and Lemma 12 (b) directly, we can show that Theorem 7 holds.

Remarks: The crucial part of proof of Lemma 13 is the augmenting. The orthogonal matrix $\mathbf{P}_K$ serves as a group of well-designed weights, to transform a non-Gaussian matrix into an asymptotic Gaussian matrix. In most cases, to apply Lindeberg's CLT for n independent random variables, a prerequisite is that their variances should be

in the same level, i.e., there should be no one's variance dominating the others so that when taking summations, the averaging effect "turns on" in terms of asymptotic Gaussianity. This implies that all the elements in $\mathbf{P}_K$ should be on the same level. However, there is no guarantee of the existence of such orthogonal matrices, unless it is of order 2^K.

3.4 Numerical Illustrations of Gaussianity in DiAMP

Having proved that SE holds in DiAMP, we now give some illustrations of Gaussianity in DiAMP, Q-Q Plot, hypothesis testing, and negentropy.

3.4.1 Q-Q Plot

Considering the DiAMP framework in (3.23), (3.24), and (3.21), on each Sensor p we further partition $\mathbf{A}_p$ equally by rows and obtain $\mathbf{A}_{p,1}$, $\mathbf{A}_{p,2} \in \mathbf{R}^{M_p/2\times N}$, and the corresponding $\mathbf{y}^{p,1}$, $\mathbf{y}^{p,2}$, $\mathbf{z}^t_{p,1}$, $\mathbf{z}^t_{p,2}$, etc. Denoting $\mathbf{u}^t_{p,i} = (\omega_p/2)\mathbf{x}^t + \mathbf{A}^T_{p,i}\mathbf{z}^t_{p,i}$ $(i = 1, 2)$, it can be shown that in the large system limit, the $2P$ random vectors $\mathbf{r}^t_{p,i} = \mathbf{u}^t_{p,i} - (\omega_p/2)\mathbf{s}_0$ behave like i.i.d. $\mathcal{N}(\mathbf{0}, (\omega_p/2)\sigma_t^2\mathbf{I}_N)$, where $\mathbf{I}_N$ is the $N \times N$ identity matrix. In Fig. 9 an example is provided illustrating the Gaussianity of $\mathbf{r}^t_{p,i}$ with soft thresholding function as the denoiser. As shown in the figure, all the Q-Q plots are close to straight lines, which is a good evidence of Gaussianity.

3.4.2 Hypothesis Test

Let us introduce an $N \times 2P$ matrix $\mathbf{G}$, with each column $\mathbf{g}_{2(p-1)+i} = \sqrt{2/\omega_p}\mathbf{r}^t_{p,i}$. If the Gaussianity assumption is valid, then all the elements g_{ij} in $\mathbf{G}$ follows i.i.d. $\mathcal{N}(0, \sigma_t^2)$. Therefore, we can design the following nonparametric hypothesis test:

H_0: g_{ij} follows i.i.d. $\mathcal{N}(0, \sigma_t^2)$;

H_1: g_{ij} does not follow i.i.d. $\mathcal{N}(0, \sigma_t^2)$,

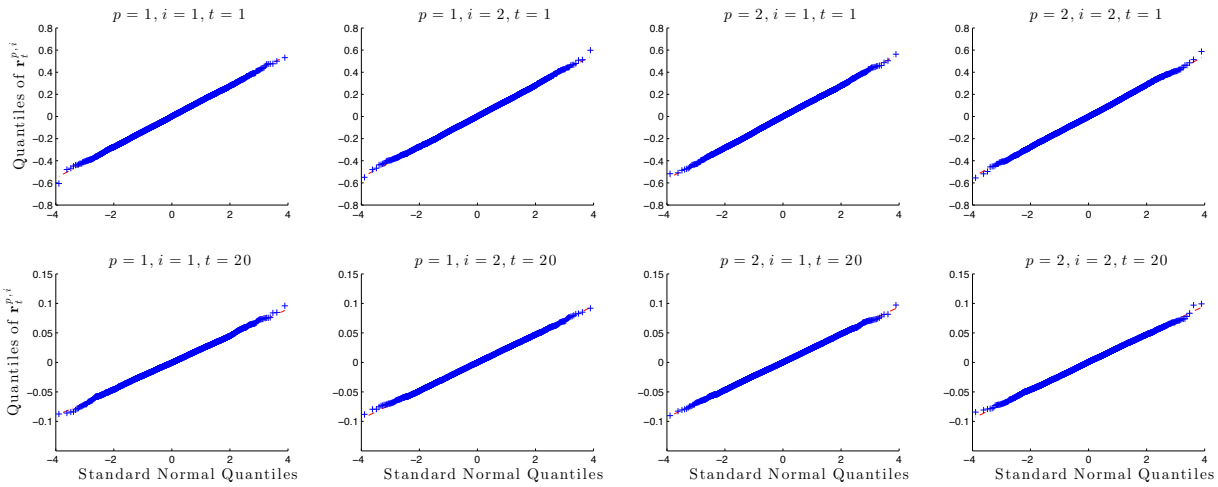

Fig. 9. QQ-plots of $\mathbf{r}_{p,i}^t$ ($p = 1, \cdots, P$ and $i = 1, 2$) at the 1-st and 20-th iterations of AMP with soft thresholding, with $P = 2$ and $\omega_1 = \omega_2 = 0.5$.

which can be performed by the Kolmogorov-Smirnov (K-S) test [84]. Note that K-S test for large sample is highly sensitive to outliers [85], whereas the sample size in our case is $2NP$ with the order of 10^5 or even larger, which may cause numerical instability if we directly run K-S on the sample. In the following we propose a hierarchical approach, which contains two layers of tests.

Layer 1: Randomly reorder the elements in $\mathbf{G}$, and then partition them equally into K_B blocks, with each block having $S_B = 2NP/K_B$ elements. For each block $i \in [K_B]$, run K-S test and obtain the corresponding p-value $p_{L_1}(i)$.

Layer 2: If the null hypothesis in Layer 1 is true, then all the p-values $p_{L_1}(i)$ should follow i.i.d. $\mathcal{U}(0,1)$ [86, 87], where $\mathcal{U}$ denotes uniform distribution; otherwise, most $p_{L_1}(i)$'s should concentrate near 0, which implies that the true CDF of $p_{L_1}(i)$'s, say $F_{L_1}(x)$, will soon increase to 1, i.e., $F(x) > x$. Therefore, we can build the following one-sided hypothesis test:

H_0: $F_{L_1}(x) = x$;

H_1: $F_{L_1}(x) > x$,

where we can perform K-S test again, and obtain the corresponding p-value p_{L_2}. Note that the test will be performed in each iteration of DiAMP, and there will be K_B p_{L_1}'s and 1 p_{L_2} per iteration. The larger p_{L_2}, the better, as is shown in the numerical results later on.

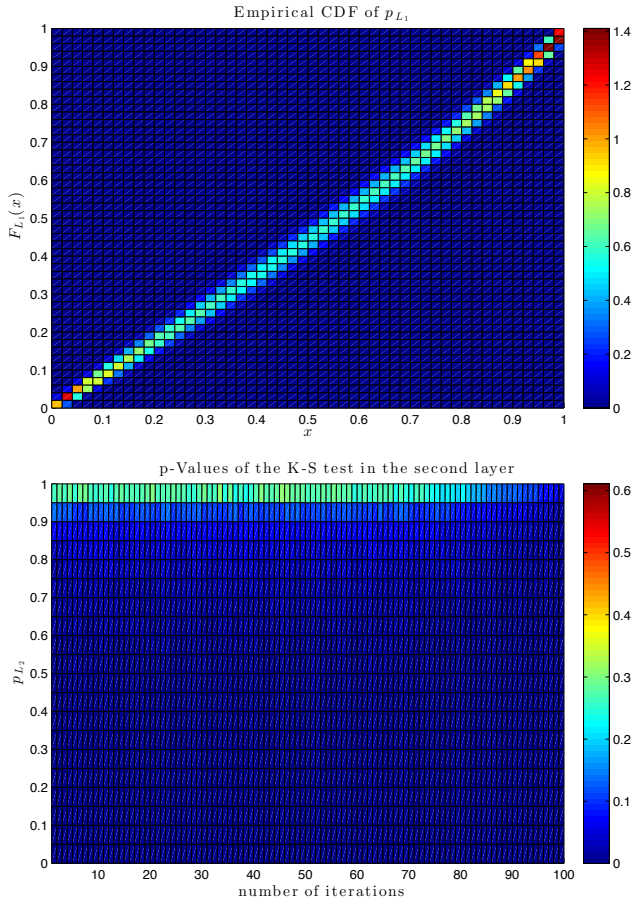

Fig. 10. p-Values of the proposed two-layer tests. In the first sub-figure, the x-axis corresponds to all the p-Values p_{L_1}'s of the first layer obtained in simulations, the $F_{L_1}(x)$-axis corresponds to their CDF values, and the color bar indicates the percentage of $(x, F_{L_1}(x))'s$ falling into each bin (in %). The second sub--figure shows all the p-Values p_{L_2}'s of the second layer in DiAMP iterations, where the color bar indicates the percentage of $(t, p_{L_2})'s$ falling into each bin (in %).

In Fig. 10 p-Values of the proposed two-layer tests are shown. As we can see, the distribution of the p-Values p_{L_1}'s of the first layer, that is, p-Values of Gaussianity

test of $\mathbf{r}_{p,i}^t$ in DiAMP, is very close to $\mathcal{U}(0,1)$; furthermore, the p-Values p_{L_2}'s of the second layer, that is, that of the uniformity test of p_{L_1}'s, are concentrated in the range $[0,1]$, which verifies the uniformity of p_{L_1}'s, thereby validating the Gaussianity of $\mathbf{r}_t^{p,i}$ in DiAMP.

3.4.3 Negentropy

We can also validate the Gaussianity in $\mathbf{r}_{p,i}^t$ by observing its negentropy, a non-Gaussianity measure [88] of a given distribution $Y \sim P_Y$ with zero mean and unit variance:

$$J_Y = \frac{\left[\mathbb{E}(Y^3)\right]^2}{12} + \frac{\left[\mathbb{E}(Y^4) - 3\right]^2}{48}, \tag{3.247}$$

where J_Y is a nonnegative number, the smaller it is, the more close it is to a standard normal distribution.

In Fig. 11 the negentropy values of $\mathbf{r}_{p,i}^t$ are shown in DiAMP, which are distributed within $[2 \times 10^{-7}, 2 \times 10^{-5}]$, very close to 0. This verifies the Gaussianity of $\mathbf{r}_{p,i}^t$ in DiAMP.

3.5 Applications: Lossy DiAMP

3.5.1 AMP with Bayesian MMSE Estimator

In previous sections on AMP, we assume no prior knowledge on $\mathbf{s}_0$, where the soft thresholding function is nearly a minimax risk denoiser. On the other hand, if we know that $\mathbf{s}_0$ follows some prior distribution, then the optimal denoiser in the mean-square-error (MSE) sense is the minimum MSE (MMSE) estimator:

$$\eta_t(F_t) = \mathbb{E}\left[S_0 \,|S_0 + \sigma_t Z = F_t\right]. \tag{3.248}$$

For simplicity of illustration, we assume that S_0 follows the Bernoulli Gaussian

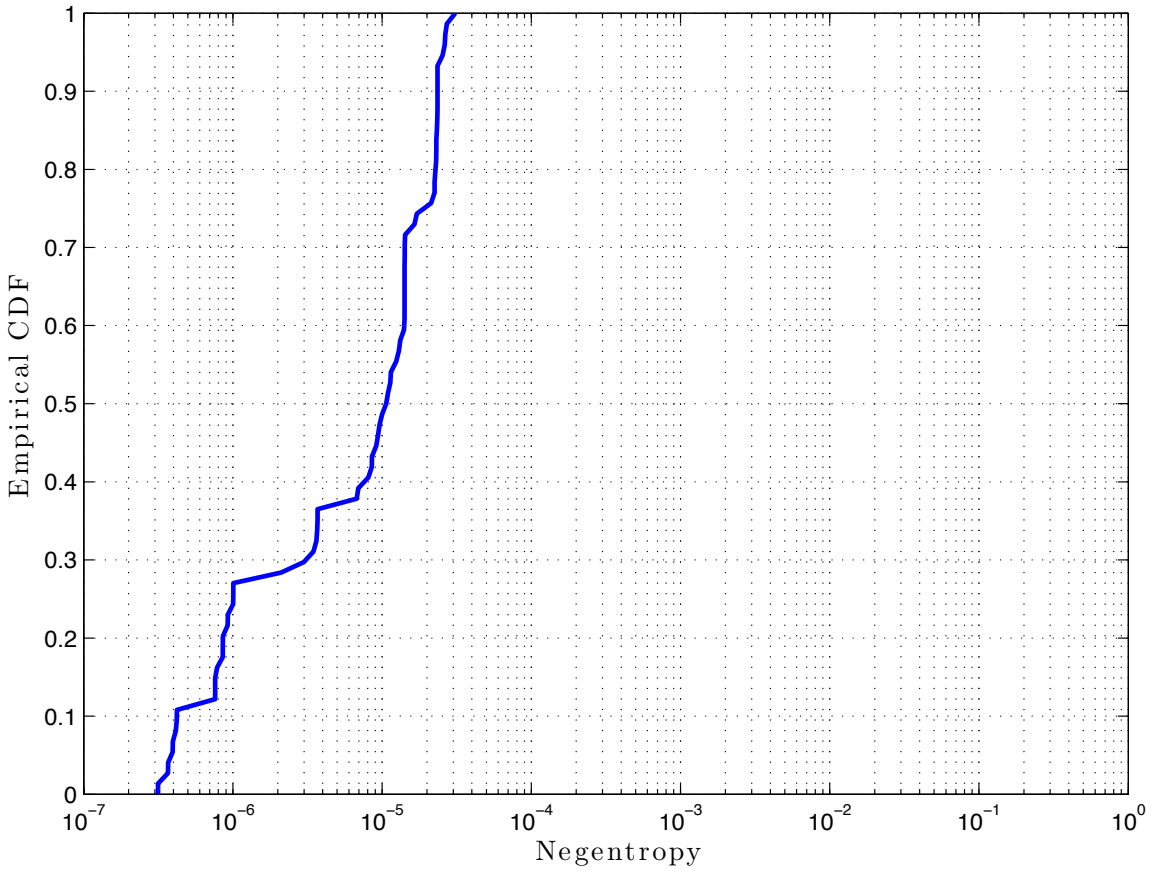

Fig. 11. Empirical CDF of negentropy of $\mathbf{r}_{p,i}^t$ in DiAMP.

distribution:

$$p_{S_0}(s) = \epsilon\mathcal{N}(s;\mu_s,\sigma_s^2) + (1-\epsilon)\delta(s), \tag{3.249}$$

where $\delta(s)$ denotes the Dirac delta function, the notation $\mathcal{N}(s;\mu_s,\sigma_s^2)$ denotes the value of Gaussian PDF $\mathcal{N}(\mu_s,\sigma_s^2)$ evaluated at s, and S_0 typically has mean $\mu_s = 0$. The denoiser is easily derived and given as follows:

$$\eta_t(F_t) = \frac{\epsilon\mathcal{N}(F_t;\mu_s,\sigma_s^2+\sigma_t^2)}{\epsilon\mathcal{N}(F_t;\mu_s,\sigma_s^2+\sigma_t^2)+(1-\epsilon)\mathcal{N}(F_t;0,\sigma_t^2)} \times \frac{F_t\sigma_s^2+\mu_s\sigma_t^2}{\sigma_s^2+\sigma_t^2}. \tag{3.250}$$

In this dissertation we set $\mu_s = 0$.

As a measure of the measurement noise level and recovery accuracy, we define

the signal-to-noise-ratio (SNR) as

$$\begin{aligned}\mathrm{SNR} &= 10\log_{10}\left(\mathbb{E}\left[\|\mathbf{A}\mathbf{s}_0\|^2\right]/\mathbb{E}\left[\|\mathbf{e}\|^2\right]\right)\\ &\approx 10\log_{10}\left(\mathbb{E}\left[\|\mathbf{s}_0\|^2\right]/\mathbb{E}\left[\|\mathbf{e}\|^2\right]\right) = 10\log_{10}\left(\rho/\sigma_e^2\right),\end{aligned}$$

where $\rho = \epsilon/\kappa$, and the signal-to-distortion-ratio (SDR) at iteration t as

$$\mathrm{SDR}(t) = 10\log_{10}\left(\mathbb{E}\left[\|\mathbf{s}_0\|^2\right]/\mathbb{E}\left[\|\mathbf{x}_t - \mathbf{s}_0\|^2\right]\right).$$

Using the SE equation in (2.5), we have

$$\mathrm{SDR}(t) = 10\log_{10}\left[\rho/\left(\sigma_t^2 - \sigma_e^2\right)\right].$$

Note that the Bernoulli Gaussian assumption in this paper is only for illustration, and our work is easily extended to other prior distributions p_{S_0}.

3.5.2 Multi-Processor AMP Framework

Consider a system with P processors and one fusion center. Each processor $p \in \{1, \cdots, P\}$ takes M/P rows of $\mathbf{A} \in \mathbb{R}^{M\times N}$, namely $\mathbf{A}^p$, and obtains $\mathbf{y}^p = \mathbf{A}^p\mathbf{s}_0 + \mathbf{e}^p$. The procedures in (2.1) — (2.3) can then be rewritten in a distributed manner:

Local Computation (LC) performed by each processor p:

$$\mathbf{z}_t^p = \mathbf{y}^p - \mathbf{A}^p\mathbf{x}_t + (1/\kappa)\overline{\eta_t'(\mathbf{f}_{t-1})}\mathbf{z}_{t-1}^p,$$

$$\mathbf{f}_t^p = \mathbf{x}_t/P + (\mathbf{A}^p)^T\mathbf{z}_t^p.$$

Global Computation (GC) performed by the fusion center:

$$\mathbf{f}_t = \sum_{p=1}^{P}\mathbf{f}_t^p,\ \overline{\eta_t'(\mathbf{f}_t)},\ \text{and}\ \mathbf{x}_{t+1} = \eta_t\left(\mathbf{f}_t\right).$$

It can be seen that in the GC step of MP-AMP, each processor p sends $\mathbf{f}_t^p$ to the fusion center, and the fusion center sums them to obtain $\mathbf{f}_t$ and $\mathbf{x}_{t+1}$, and sends $\mathbf{x}_{t+1}$

to each processor.[6] Our goal in this dissertation is to reduce these communication costs while barely impacting recovery performance.

Suppose that all the elements in $\mathbf{f}_t^p$ are computed as 32-bit single-precision floating-point numbers. As shown in previous sections, SE still holds in DiAMP even in the presence of quantization noises, we can compress $\mathbf{f}_t^p$ lossily up to some reasonable distortion level, and send the compressed output to the fusion center. By applying SE in DiAMP, we can link σ_t^2 to the quantization error D, while by applying rate-distortion theory [77], we can further connect D to the bit rates we use in lossy compression. In other words, we can precisely control the trade-off between recovery accuracy and communication cost, based on the one-to-one map from the bit rate per element R, and σ_t^2 in DiAMP, a measure of its accuracy.

3.5.3 Lossy Compression of $\mathbf{f}_t^p$

Due to the proof of SE in DiAMP in previous sections of this chapter, we know that elements of $\mathbf{f}_t^p - (1/P)\mathbf{s}_0$ are i.i.d. Gaussian with mean 0 and variance σ_t^2/P. Furthermore, $\mathbf{f}_t^p - (1/P)\mathbf{s}_0$ and $\mathbf{f}_t^q - (1/P)\mathbf{s}_0$ are independent for different processors p and q. In light of this property, $\mathbf{f}_t^p$ can be described as a scalar channel:

$$F_t^p = S_0/P + (\sigma_t/\sqrt{P})Z_p, \text{ where } Z_p \sim \mathcal{N}(0,1).$$

For the Bernoulli Gaussian distribution (3.249),

$$F_t^p \sim \epsilon\mathcal{N}\left(\mu_s/P, (\sigma_s^2 + P\sigma_t^2)/P^2\right) + (1-\epsilon)\mathcal{N}\left(0, \sigma_t^2/P\right).$$

Scalar Quantization: Next, we propose a uniform quantizer with entropy coding, also known as entropy coded scalar quantization (ECSQ) [89].

[6] In order to calculate each $\mathbf{z}_{t+1}^p$, the fusion center also needs to send $\overline{\eta_t'(\mathbf{f}_t)}$ to all the processors. This is a scalar, and the corresponding communication cost is negligible compared with that of transmitting a vector.

Let $\Psi(u)$ denote the characteristic function of F_t^p, it can be shown that

$$|\Psi(u)| \leq \epsilon \exp\left[-0.5\left(\sigma_s^2 + P\sigma_t^2\right)u^2/P^2\right]$$
$$+ (1-\epsilon)\exp\left(-0.5\sigma_t^2 u^2/P\right) \leq \exp\left(-0.5\sigma_t^2 u^2/P\right)$$

is nearly band-limited. Due to this property, it is possible to develop a uniform quantizer of $\mathbf{f}_t^p \sim$ i.i.d. F_t^p, where the quantization error $\mathbf{v}_t^p$ is approximately statistically equivalent to a uniformly distributed noise $V_t^p \sim \mathcal{U}\left[-0.5\Delta_Q, 0.5\Delta_Q\right]$ uncorrelated to F_t^p. Actually, a quantization bin size $\Delta_Q \leq 2\sigma_t/\sqrt{P}$ will suffice for validation of $\mathbf{v}_t^p \sim$ i.i.d. V_t^p [66].

The fusion center will receive the quantized data $\widetilde{\mathbf{f}}_t^p \sim$ i.i.d. $\widetilde{F}_t^p$, and calculate $\widetilde{\mathbf{f}}_t = \sum_{p=1}^P \widetilde{\mathbf{f}}_t^p \sim$ i.i.d. $\widetilde{F}_t$, where

$$\widetilde{F}_t = \sum_{p=1}^P \widetilde{F}_t^p = F_t + V_t, \text{ and } V_t = \sum_{p=1}^P V_t^p. \tag{3.251}$$

Applying the central limit theorem, V_t approximately follows $\mathcal{N}(0, P\sigma_Q^2)$ for large P, where $\sigma_Q^2 = \Delta_Q^2/12$.

Entropy Coding and Optimum Bit Rate: Let p_i be the probability that F_t^p falls into the i-th quantization bin. The entropy of quantized F_t^p, $\widetilde{F}_t^p$, is $H_Q = -\sum_i p_i \log_2(p_i)$[77], that is, the sensors need H_Q bits on average to represent each element in $\widetilde{\mathbf{f}}_t^p$ to the fusion center, which is achievable through entropy coding [77].

In rate distortion (RD) theory [77], we are given a length-n random sequence $Y_n = \{Y_{n,i}\}_{i=1}^n \sim$ i.i.d. F_Y, and our goal is to identify a reconstruction sequence $\widehat{Y}_n = \{\widehat{Y}_{n,i}\}_{i=1}^n$ that can be encoded at low rate while the distortion $d(Y_n, \widehat{Y}_n) = \frac{1}{n}\sum_i d(Y_{n,i}, \widehat{Y}_{n,i})$ (e.g., squared error distortion) between the input and the reconstruction sequence is small. RD theory has characterized the fundamental best-possible trade-off between the distortion $D = d(Y_n, \widehat{Y}_n)$ and coding rate $R(D)$, which is called the rate distortion function. The RD function $R(D)$ can be computed numerically (cf. Blahut [90] and Arimoto [91]). For the uniform quantizer that yields a quantiza-

tion MSE of σ_Q^2 with a coding rate H_Q bits per element, the RD function will give a bit rate $R(D = \sigma_Q^2) < H_Q$, which is achievable through vector quantization [89].

New SE Equation: For both ECSQ and RD-based vector quantization that lead to a quantization MSE of σ_Q^2, the fusion center will have $\widetilde{F}_t = S_0 + \sqrt{\sigma_t^2 + P\sigma_Q^2}\widetilde{Z}$, where $\widetilde{Z} \sim \mathcal{N}(0,1)$. The new denoiser and SE equation become

$$\eta_t^Q(\widetilde{F}_t) = \mathbb{E}\left[S_0 \middle| S_0 + \sqrt{\sigma_t^2 + P\sigma_Q^2}\widetilde{Z} = \widetilde{F}_t\right] \text{ and}$$

$$\sigma_{t+1}^2 = \sigma_e^2 + (1/\kappa)\mathbb{E}\left[\eta_t^Q\left(S_0 + \sqrt{\sigma_t^2 + P\sigma_Q^2}\widetilde{Z}\right) - S_0\right]^2. \tag{3.252}$$

Currently, we only consider compression of $\mathbf{f}_t^p$. When broadcast from the fusion center to the P processors is allowed in the network topology, the communication cost of sending $\mathbf{x}_t$ – even uncompressed – is smaller than that of communicating the P vectors $\mathbf{f}_t^p$. We are considering the case where broadcast is not allowed in our ongoing work.

3.5.4 Online Back-tracking (BT-MP-AMP)

Let $\sigma_{t,C}^2$ and $\sigma_{t,D}^2$ denote the σ_t^2 obtained by the centralized AMP (2.5) and MP-AMP (3.252), respectively. In order to reduce communication while maintaining high fidelity, we first constrain $\sigma_{t,D}^2$ so that it will not deviate much from $\sigma_{t,C}^2$, and then determine the minimum coding rate required in each iteration. This can be done through an online back-tracking algorithm, which we name BT-MP-AMP and present below.

In each iteration t, before quantizing $\mathbf{f}_t^p$, we first compute $\sigma_{t+1,C}^2$ for the next iteration. Then we find the maximum quantization MSE σ_Q^2 allowed so that the ratio $\sigma_{t+1,D}^2/\sigma_{t+1,C}^2$ does not exceed some constant, provided that the required bit rate does not exceed some threshold. Based on the obtained σ_Q^2 we construct the corresponding quantizer.

Note that the SE in (3.252) is only an approximation, and we do not know the true value of $\sigma_{t,D}^2$ in the current iteration. To better predict $\sigma_{t+1,D}^2$, we use $\widehat{\sigma}_{t,D}^2 = \|\mathbf{z}_t^p\|^2/M$, which is a good estimator for $\sigma_{t,D}^2$[57, 58], to compute $\sigma_{t+1,D}^2$. To obtain $\widehat{\sigma}_{t,D}^2$, each processor p sends the scalar $\|\mathbf{z}_t^p\|^2$ to the fusion center, which then sends the scalar $\hat{\sigma}_{t,D}^2 = \sum_{p=1}^P \|\mathbf{z}_t^p\|^2/M$ to all the processors. The corresponding communication cost is also negligible compared with that of communicating $\mathbf{f}_t^p$.

3.5.5 Dynamic Programming (DP-MP-AMP)

While back-tracking is a useful heuristic, it is possible for a given coding budget R per element, total number of AMP iterations T, and initial noise level σ_0^2 in the scalar channel to compute the coding rate allocations among the AMP iterations that minimize the final MSE, $\sigma_{T,D}^2$.

To do so, note that we can evaluate $\sigma_{t,C}^2$ offline and hence obtain the number of iterations required to reach the steady state, which would be a reasonable choice for T. Second, recalling the new SE equation in (3.252), $\sigma_{t,D}^2$ depends on $\sigma_{t-1,D}^2$ and σ_Q^2, which is a function of R_t, the coding rate allocated in the t-th iteration. Therefore, we can rewrite $\sigma_{t,D}^2$ as follows:

$$\begin{aligned}\sigma_{t,D}^2 &= f_1(\sigma_{t-1,D}^2, R_t) = f_2(\sigma_{t-2,D}^2, R_{t-1}, R_t) \\ &= \cdots = f_t(\sigma_0^2, R_1, \cdots, R_{t-1}, R_t),\end{aligned} \tag{3.253}$$

that is, given σ_0^2, $\sigma_{T,D}^2$ is only a function of R_t for $t \in \{1, 2, \cdots, T\}$. Denoting $\mathcal{F}_T(R) = \{R_1, \cdots, R_T \geq 0\colon \ \sum_{t=1}^T R_t = R\}$, minimizing $\sigma_{T,D}^2$ for a given R can be formulated as the following optimization problem:

$$\min_{\mathcal{F}_T(R)} \sigma_{T,D}^2 = \min_{\mathcal{F}_T(R)} f_T(\sigma_0^2, R_1, \cdots, R_T). \tag{3.254}$$

Since $\sigma_{t,D}^2$ is increasing with $\sigma_{t-1,D}^2$, it is easy to verify the following recursive rela-

tionship:

$$\min_{\mathcal{F}_T(R)} \sigma^2_{T,D} = \min_{0 \le R_T \le R} f_1\left(\min_{\mathcal{F}_{T-1}(R-R_T)} \sigma^2_{T-1,D}, R_T\right) = \cdots,$$

which makes the problem solvable through dynamic programming (DP).

To implement DP, we need to discretize $\mathcal{F}_T(R)$ into $\{R_1, \cdots, R_T \in \Omega : \sum_{t=1}^{T} R_t = R\}$, where $\Omega = \{R^{(1)}, \cdots, R^{(S)}\}$ with $R^{(s)} = R(s-1)/(S-1), \forall s \in \{1, \cdots, S\}$. In this dissertation, we set the bit rate resolution $\Delta R = R/(S-1) = 0.1$ bits per element. Then, we create an $S \times T$ array $\mathbf{\Sigma}$, with the element in the s-th row ($s \in \{1, \cdots, S\}$) and t-th column ($t \in \{1, \cdots, T\}$) denoted as $\sigma^2_D(s,t)$, storing the optimal value of $\sigma^2_{t,D}$ when a total of $R^{(s)}$ bits per element are used in the first t iterations. By the definition of $\sigma^2_D(s,t)$, we have

$$\sigma^2_D(s,t) = \min_{r \in \{1,2,\cdots,s\}} f_1\left(\sigma^2_D(r,t-1), R^{(s-r+1)}\right), \tag{3.255}$$

and the first column of elements in $\mathbf{\Sigma}$ is obtained by:

$$\sigma^2_D(s,1) = f_1\left(\sigma^2_0, R^{(s)}\right), \ \forall s \in \{1,2,\cdots,S\}. \tag{3.256}$$

After obtaining $\mathbf{\Sigma}$, the optimal value of $\sigma^2_{T,D}$, by definition, is $\sigma^2_D(S,T)$. Meanwhile, to obtain the optimal bit allocation strategy, we need another $S \times T$ array $\mathbf{R}$ to store the optimal bit rate $R_{DP}(s,t)$ that is allocated at iteration t when a total of $R^{(s)}$ bits per element are used in the first t iterations. Similar to BT-MP-AMP, we name the proposed MP-AMP approach combined with DP as DP-MP-AMP.

3.6 Numerical Results

We evaluate BT-MP-AMP and DP-MP-AMP in an MP system with $P = 30$ processors at SNR= 20 dB, where we set $N = 10{,}000$, $M = 3{,}000$, i.e., $\kappa = 0.3$, and generate Bernoulli-Gaussian sequences $\mathbf{s}_0$ with $\epsilon \in \{0.03, 0.05, 0.1\}$, $\mu_s = 0$, and $\sigma_s = 1$.

We first evaluate the SE equation (2.5) of centralized AMP for the three sparsity

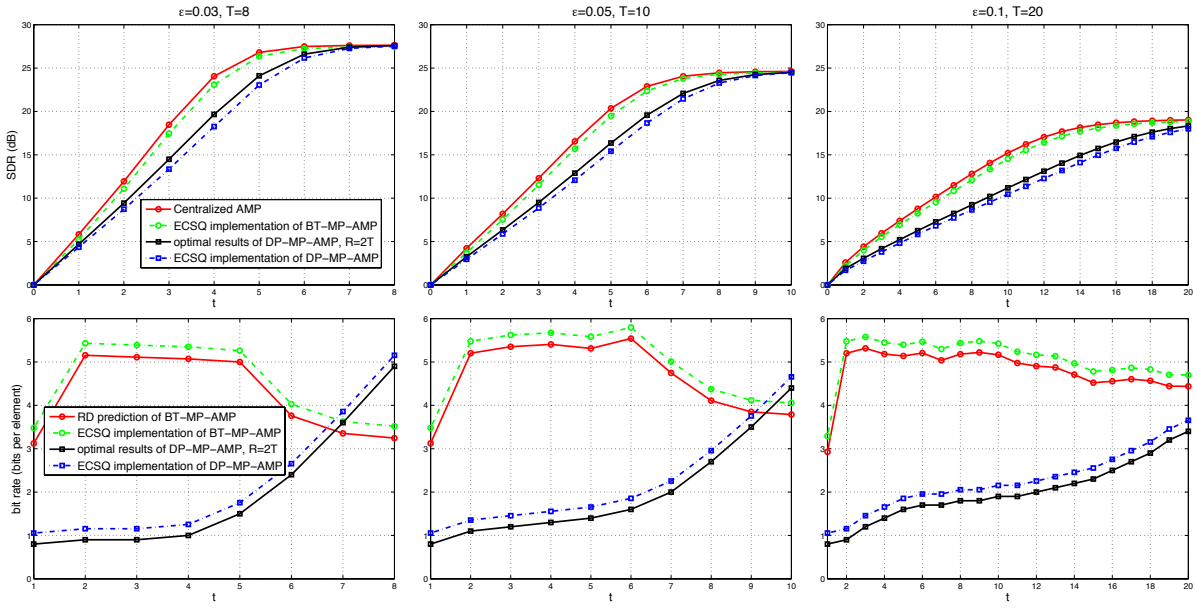

Fig. 12. SDR and bit rates as functions of iteration number t. ($N = 10{,}000$, $M = 3{,}000$, $\kappa = 0.3$, $\mu_s = 0, \sigma_s = 1$, SNR $= 20$ dB.)

levels. As shown in Fig. 12, they reach the steady state after $T = 8$, 10, and 20 iterations respectively. Then, we run BT-MP-AMP and DP-MP-AMP, where for the latter the total rates are $R = 2T$ bits per element and the RD-function models the relation between R_t and σ_Q^2.

According to RD theory, in the high rate limit, we should expect a gap of roughly 0.255 bits per element between the entropy and RD function for a given distortion level [89]. Therefore, in an implementation of DP-MP-AMP where we apply ECSQ, we add 0.255 bits per element to the results in each iteration obtained by DP. Note that the two solid curves in the top three panels are obtained through offline calculation and optimization, and the two dash-dotted curves are obtained through AMP simulations. As shown in Fig. 12, BT-MP-AMP uses fewer than 6 bits per element in each iteration, more than 80% communication savings compared with 32-bit single-precision floating-point transmission, while achieving almost the same SDR's as in centralized AMP. On the other hand, there are clear gaps between the SDR's of

Table 7. Total bits per element of MP-AMP

ϵ	0.03	0.05	0.10
T	8	10	20
BT-MP-AMP (RD prediction)	33.82	46.43	96.16
BT-MP-AMP (ECSQ simulation)	36.09	49.19	101.50
DP-MP-AMP (RD prediction)	16	20	40
DP-MP-AMP (ECSQ simulation)	18.04	22.55	45.10

DP-MP-AMP and centralized AMP during the first few iterations, but they vanish quickly as t approaches T, in return for over 50% communication reduction beyond that provided by BT-MP-AMP, as shown in Table 7.

Note also that the ECSQ implementation of DP-MP-AMP has lower SDR's than that predicted by DP results based on the RD function at the beginning. This is because the 0.255-bits gap only holds in the high rate limit. However, due to the robustness of SE to disturbances, and the increasingly high rates as t approaches T, the ECSQ implementation matches the predicted DP results at the last iteration.

3.7 Conclusion

In this chapter, we proved that SE still holds in a distributed AMP framework, where quantization noise is incorporated. Furthermore, compared to previous theoretical results which assume that the sensing matrix consists of independent subgaussian entries, our work further extends the validity of SE to more general sensing matrices, which strengths the theoretical support of AMP in general applications. Taking advantage of this theoretical progress, we proposed a multi-processor approximate message passing framework with lossy compression. We used a uniform quantizer with entropy coding to reduce communication costs, and reformulated the SE equation while accounting for quantization noise. Combining the quantizers and modified state evolution equation, an online back-tracking approach and another method based on dynamic programming were developed to determine the coding rate in each

iteration by controlling the induced error. The numerical results showed that our approaches can maintain a high signal-to-distortion-ratio despite a significant and often dramatic reduction in inter-processor communication costs.

CHAPTER 4

DCS BASED ON ITERATIVE HARD THRESHOLDING

4.1 Introduction

Different from AMP, iterative hard thresholding (IHT) is a deterministic greedy algorithm [44, 45]. Taking an input parameter K, the sparsity level of the sparse signal of $\mathbf{s}_0 \in \mathbb{R}^N$, IHT aims to find a local minimum of (1.2) close to $\mathbf{s}_0$; provided that the sensing matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ satisfies the RIP condition [16] well and there is no measurement noise, IHT hopefully converges to $\mathbf{s}_0$ [45]. The term "deterministic" is from the modeling perspective, meaning that there are no quantities in the CS measurement model in (1.1) that are viewed as random variables. The term "greedy" depicts its nature, since in each iteration it minimizes a surrogate function which is an upper bound on the objective function in (1.2). IHT has a linear convergence rate [44], which is good for most CS recovery algorithms, and compared with other iterative greedy algorithms like orthogonal matching pursuit (OMP), it has a lower complexity in each iteration, since no matrix inversion is performed.

Denote IHT with the input parameter K as IHT_K. It starts with $\mathbf{x}_0 = 0$ and $\mathbf{z}_0 = \mathbf{y}$, and repeats the following process:

$$\mathbf{f}_t = \mathbf{x}_t + \mu \mathbf{A}^T \mathbf{z}_t \tag{4.1}$$

$$\mathbf{x}_{t+1} = \eta(\mathbf{f}_t; K) \tag{4.2}$$

$$\mathbf{z}_{t+1} = \mathbf{y} - \mathbf{A}\mathbf{x}_{t+1} \tag{4.3}$$

where the step size μ can be any positive number within $(0, 1/\|\mathbf{A}\|_2)$, and $\mathbf{u} = \eta(\mathbf{v}; K)$

for $\mathbf{v} \in \mathbb{R}^n$ is called hard thresholding function, where $\mathbf{u} \in \mathbb{R}^n$ only keeps the K largest-in-magnitude components in $\mathbf{v}$ and has other components all zero, that is

$$\mathbf{u}(k) = \mathbf{v}(k)\mathbb{I}(|\mathbf{v}(k)| \geq \mathcal{T}_K(\mathbf{v})), \forall k \in [n] \tag{4.4}$$

Like DiAMP, we can introduce a similar intermediate matrix $\mathbf{W}_t = \left[\mathbf{w}_t^1, \cdots, \mathbf{w}_t^P\right]$ for distributed IHT (DIHT) [40]:

$$\mathbf{w}_t^p = \begin{cases} \mathbf{x}_t + \mu(\mathbf{A}^p)^T \mathbf{z}_t^p, & p = 1, \\ \mu(\mathbf{A}^p)^T \mathbf{z}_t^p, & \text{otherwise.} \end{cases} \tag{4.5}$$

It is easy to show that

$$\mathbf{f}_t = \sum_{p=1}^{P} \mathbf{w}_t^p \tag{4.6}$$

and

$$\mathbf{x}_{t+1} = \eta(\mathbf{f}_t; K) = \eta\left(\sum_{p=1}^{P} \mathbf{w}_t^p; K\right) \tag{4.7}$$

$$\mathbf{z}_{t+1}^p = \mathbf{y}^p - \mathbf{A}^p \mathbf{x}_{t+1}, \forall p = 1, \cdots, P \tag{4.8}$$

also hold for DIHT.

To save the communication cost in the GC step of DIHT, a modified TA (MTA) approach was proposed in [40], which has been introduced in Chapter 2. As shown in Table 8, in each iteration of the modified TA (MTA), there is an object being selected and the corresponding total score is computed; then an upper bound ν on magnitudes of the total scores that have not been computed yet is obtained. The algorithm terminates if the K-th largest magnitude of computed total scores is greater than ν.

Table 8. MTA Algorithm for DIHT

Input $\mathbf{w}_t^1, \cdots, \mathbf{w}_t^P$, K

Initialize $\mathbf{x}_{t+1} = \mathbf{0} \in \mathbb{R}^N$, `count`$= 0$, $\tau_T = +\infty$, $\tau_B = +\infty$,
$u_p = +\infty$, $\ell_p = -\infty$, $\forall p = 1, \cdots, P$;
Mark all the pairs $(n, \mathbf{w}_t^p(n))$ as "unsent", $\forall n$, p;
while `TRUE`
  for sensor $p = 1{:}P$
    obtain $R = \{n : (n, \mathbf{w}_t^p(n))$ is marked as "unsent"$\}$;
    if $\tau_T \geq \tau_B$
      set $n_s = \arg\max_{n \in R} \mathbf{w}_t^p(n)$;
      update $u_p = \mathbf{w}_t^p(n_s)$ and $\tau_T = \max\{0, \sum_{q=1}^P u_q\}$;
    else
      set $n_s = \arg\min_{n \in R} \mathbf{w}_t^p(n)$;
      update $\ell_p = \mathbf{w}_t^p(n_s)$ and $\tau_B = -\min\{0, \sum_{q=1}^P \ell_q\}$;
    endif
    $\star$ broadcast $(n_s, \mathbf{w}_t^p(n_s))$ and mark it as "sent";
    for sensor $q \neq p$
    $\star$ send $(n_s, \mathbf{w}_t^q(n_s))$ to sensor p and mark it as "sent";
      store $\mathbf{w}_t^p(n_s)$ as the new u_p or ℓ_p;
    endfor
    $\star$ compute $\mathbf{f}_t(n_s)$ and broadcast it to other sensors;
    `count`=`count`+1;
    let β be K-th largest element in $\{|\mathbf{f}_t(n)| : n \notin R \backslash \{n_s\}\}$;
    if $\max\{\tau_T, \tau_B\} < \beta$ or `count`$\geq N$
      update $\mathbf{x}_{t+1}(n) = \mathbf{f}_t(n)$ if $|\mathbf{f}_t(n)| > \beta$, $\forall n \notin R \backslash \{n_s\}$;
      set $N_s =$ `count`, the algorithm terminates;
    endif
  endfor
endwhile

Output $\mathbf{x}_{t+1}$

4.2 Proposed GC Algorithms for DIHT

4.2.1 GC.K Algorithm

For DIHT, according to (4.4), $\mathbf{x}_{t+1}(n) = 0$ if $|\mathbf{f}_t(n)| < \mathcal{T}_K(\mathbf{f}_t)$. Therefore, the GC in DIHT turns out to be a Top-K problem, that is, to find the K largest-in-magnitude total scores $\mathbf{f}_t(n) = \sum_{p=1}^P \mathbf{w}_t^p(n)$, as well as the indices n's of objects they correspond to. Similar to the derivation of GCAMP, what we are going to do is not using TPUT

[49] to solve the problem since entries in $\mathbf{W}_t$ can either be positive or negative, but following the essence of the TPUT algorithm to develop our own Top-K algorithm.

Before presenting GC.K, we first give the following straightforward propositions and lemmas, which shows the intuitive essence of GC.K.

Proposition 11 *Let $f(x)$ be a scalar-valued function with domain Ω_G, then for any $\Omega_L \subset \Omega_G$, we have*

$$\sup_{x\in\Omega_G} f(x) \geq \sup_{x\in\Omega_L} f(x) \tag{4.9}$$

Proposition 12 *Let $f(x)$ and $g(x)$ be two scalar-valued functions with a common domain Ω, if $f(x) \geq g(x)$ holds for any $x \in \Omega$, then we have*

$$\sup_{x\in\Omega} f(x) \geq \sup_{x\in\Omega} g(x) \tag{4.10}$$

With Propositions 11 and 12, we have the following Lemma.

Lemma 14 *Given $\mathbf{v} \in \mathbb{R}^N$, for arbitrary $\Omega \subset [N]$, let $\mathbf{u} = \mathbf{v}(\Omega)$ and $\mathbf{u}^c = \mathbf{v}([N] \setminus \Omega)$, we have*

$$\mathcal{T}_K(\mathbf{v}) \geq \mathcal{T}_K(\mathbf{u}), \forall K \leq |\Omega|. \tag{4.11}$$

The equality holds if $|\mathbf{u}^c(j)| < \mathcal{T}_K(v), \forall j \in [N - |\Omega|]$.

Furthermore, if $\mathbf{l} \in \mathbb{R}^{|\Omega|}$ satisfies $\mathbf{l}(j) \leq \mathbf{u}(j), \forall j \in [|\Omega|]$, then

$$\mathcal{T}_K(\mathbf{v}) \geq \mathcal{T}_K(\mathbf{l}), \forall K \leq |\Omega|. \tag{4.12}$$

Lemma 14 is straightforward after applying Propositions 11 and 12. With Lemma 14, we develop the following GC.K algorithm, of which the essence is to get lower bounds on $\mathcal{T}_K(|\mathbf{f}_t|)$ and upper bounds on $|\mathbf{f}_t(n)|$ for each $n \in [N]$.

Let Ω_p^1 be the set of indices of the largest-in-magnitude K partial scores $\mathbf{w}_t^p(n)$ on Sensor p. First, each Sensor $p \geq 2$ sends the $(n, \mathbf{w}_t^p(n))$ pairs for all $n \in \Omega_p^1$ to Sensor 1. Sensor 1 then computes a partial sum $P(n)$ for each $n \in \bigcup_{p=1}^P \Omega_p^1$, where a

partial sum for n is defined as the summation of partial scores Sensor 1 now collects for n (including $\mathbf{w}_t^1(n)$). Denote F^1 as the set of indices for the largest-in-magnitude K partial sums, then Sensor 1 broadcasts a request for all the partial scores for each $n \in F^1$ from other Sensors, computes the total score $\mathbf{f}_t(n)$ for each $n \in F^1$, and obtains $\nu_1 = \mathcal{T}_K(|\mathbf{f}_t(F^1)|)$.

The following process is very similar to GCAMP algorithm. Set $T = \theta\nu_1/(P-1)$, where $\theta \in (0,1)$. For $p \geq 2$, define $\Omega_p^2 = \{n \notin F^1 \bigcup \Omega_p^1 : |\mathbf{w}_t^p(n)| \geq T\}$. Each Sensor $p \geq 2$ sends the $(n,\ \mathbf{w}_t^p(n))$ pairs for all $n \in \Omega_p^2$ to Sensor 1. So far, Sensor 1 has obtained partial scores from Sensor p for all $n \in \Omega_p = \Omega_p^1 \bigcup \Omega_p^2 \bigcup F^1$, and the set

$$S_n = \{p \in [P] \setminus \{1\} : n \in \Omega_p\} \tag{4.13}$$

for each $n \notin F^1$. Sensor 1 then computes

$$L(n) = \max(|\mathbf{w}_t^1(n) + \sum_{p \in S_n} \mathbf{w}_t^p(n)| - (P - 1 - |S_n|)T, 0) \tag{4.14}$$

and

$$U(n) = |\mathbf{w}_t^1(n) + \sum_{p \in S_n} \mathbf{w}_t^p(n)| + (P - 1 - |S_n|)T \tag{4.15}$$

for each $n \notin F^1$. Denote ν_2 as the K-th largest lower bound $L(n)$ and set $\nu = \max(\nu_1, \nu_2)$, Sensor 1 obtains the set $F^2 = \{n \notin F^1 : U(n) \geq \nu\}$, requests all the partial scores for each $n \in F^2$ from other sensors, and computes the total score $\mathbf{f}_t(n)$. Up to now, we have obtained total scores for all $n \in F = F^1 \bigcup F^2$.

Finally, compute $\mathbf{x}_{t+1}$ as follows: for any $n \in F$,

$$\mathbf{x}_{t+1}(n) = \mathbf{f}_t(n)\mathbb{I}(\mathbf{f}_t(n) \geq \mathcal{T}_K(\mathbf{f}_t(F))) \tag{4.16}$$

and for any $n \notin F$, $\mathbf{x}_{t+1}(n) = 0$.

Lemma 15 *$L(n)$ in (4.14) is a lower bound on $|\mathbf{f}_t(n)|$, and $U(n)$ in (4.15) is an*

upper bound on $|\mathbf{f}_t(n)|$.

The proof of Lemma 15 is very similar to that of Lemma 1, by applying the triangular inequality and the fact that $|\mathbf{w}_t^p(n)| < T$ for all $p \notin S_n$.

Theorem 10 *In each iteration, GC.K algorithm gives exactly the same* $\mathbf{x}_{t+1}$ *as that of the centralized IHT algorithm computed by* (4.2).

Proof of Theorem 10: Let $\mathbf{x}_{t+1}^G$ and $\mathbf{x}_{t+1}^I$ denote the result obtained by the GC.K and the centralized IHT respectively. According to Lemma 14, ν_1 is a lower bound on $\mathcal{T}_K(|\mathbf{f}_t|)$. For all $n \notin F^1$, according to Lemma 15, $L(n) \leq |\mathbf{f}_t(n)|$; applying Lemma 14 again, ν_2 is another lower bound on $\mathcal{T}_K(|\mathbf{f}_t|)$. Therefore, $\mathcal{T}_K(\mathbf{f}_t) \geq \nu = \max(\nu_1, \nu_2)$.

Now, $\forall n \notin F$, we have $\mathbf{x}_{t+1}^G(n) = 0$; according to Lemma 15, we have $|\mathbf{f}_t(n)| \leq U(n) < \nu \leq \mathcal{T}_K(|\mathbf{f}_t|)$, so $\mathbf{x}_{t+1}^I(n) = 0$ and according to Lemma 14, $\mathcal{T}_K(|\mathbf{f}_t|) = \mathcal{T}_K(|\mathbf{f}_t(F)|)$. $\forall n \in F$, according to (4.2), (4.4), (4.16), and the fact that as we $\mathcal{T}_K(\mathbf{f}_t) = \mathcal{T}_K(\mathbf{f}_t(F))$ just showed, we have $\mathbf{x}_{t+1}^G(n) = \mathbf{x}_{t+1}^I(n)$. Therefore, $\mathbf{x}_{t+1}^G = \mathbf{x}_{t+1}^I$.

The pseudo code of the GC.K algorithm is shown in Table 9, which contains 8 steps. It can be shown that the total number of messages is $\sum_{p=1}^{P} |\Omega_p \bigcup F| + (|F|+1)$, where the first part is the number of data other sensors send to Sensor 1, and the second part is the number of broadcasting messages Sensor 1 sends to others.

Table 9.: GC.K algorithm

Input $\mathbf{w}_t^1, \cdots, \mathbf{w}_t^P$, K, θ

Step I

for sensor $p = 1{:}P$

obtain $\Omega_p^1 = \{n : |\mathbf{w}_t^p(n)| \geq \mathcal{T}_K(\mathbf{w}_t^p)\}$;

Sensors $p \geq 2$ send all $(n, \mathbf{w}_t^p(n))$ pairs for $n \in \Omega_p^1$ to Sensor 1

endfor

Step II define $\mathbf{x}_{t+1}^h \in \mathbb{R}^N$ and initialize $\mathbf{x}_{t+1}^h = 0$;

for $n \in \bigcup_{p=1}^P \Omega_p^1$

get $R_n := \{p : n \in \Omega_p^1\}$;

Compute a partial sum $P(n) = \mathbf{w}_t^1(n) + \sum_{p \in R_n \setminus \{1\}} \mathbf{w}_t^p(n)$;

endfor

Sort all the $(n,P(n))$ pairs in the descending order of $|P(n)|$;

Broadcast F^1, which is the set of n's in the first K pairs of the sorted list;

Step III Upon receiving the broadcast message from sensor 1

for sensor $p = 2{:}P$

send all $(n, \mathbf{w}_t^p(n))$ pairs for $n \in F^1 \backslash \Omega_p^1$ to sensor 1.

endfor

Step IV Sensor 1 assigns $\mathbf{x}_{t+1}^h(F^1) = \mathbf{f}_t(F^1)$, obtain $\nu_1 = \mathcal{T}_K(|\mathbf{x}_{t+1}^h(F_1))|$ and broadcast ν_1;

Step V Upon receiving ν_1 from sensor 1

for sensor $p = 2{:}P$

Set $T = \nu_1\theta/(P-1)$ and obtain $\Omega_p^2 = \{n : |\mathbf{w}_t^p(n)| > T\} \backslash (\Omega_p^1 \bigcup F_1)$;

$\Omega_p = \Omega_p^1 \bigcup \Omega_p^2 \bigcup F_1$;

send all $(n, \mathbf{w}_t^p(n))$ pairs for $n \in \Omega_p^2$ to sensor 1.

endfor

Step VI On sensor 1,

for $n \in [N] \setminus F_1$

get S_n as defined in (4.13), compute $L(n)$ and $U(n)$ according to (4.14) and (4.15);;

endfor

Define ν_2 as the K-th largest $L(n)$ and set $\nu = \max(\nu_1, \nu_2)$;

Broadcast $F^2 := \{n \in [N] \setminus F^1 : U(n) \geq \nu\}$;

Step VII

for sensor $p = 2{:}P$

send all $(n, \mathbf{w}_t^p(n))$ pairs for $n \in F^2 \setminus \Omega_p$ to sensor 1.

endfor

Step VIII for Sensor 1, assign $\mathbf{x}_{t+1}^h(F^2) = \mathbf{f}_t(F^2)$

obtain $F = F^1 \bigcup F^2$ and $\Gamma = \{n : |\mathbf{x}_{t+1}^h(n)| \geq \mathcal{T}_K(|\mathbf{x}_{t+1}^h(F)|)\}$;;

compute $\mathbf{x}_{t+1}(\Gamma) = \mathbf{x}_{t+1}^h(\Gamma)$ and set $\mathbf{x}_{t+1}([N] \setminus \Gamma) = 0$;

Output $\mathbf{x}_{t+1}$.

In Figs. 13 and 14 an example of GC.K was shown with the same input data as in Fig. 1. Suppose $K = 2$, in Step I, each Sensor p obtains the set of indicies Ω_p^1 for the largest-in-magnitude partial scores, as we can see, $\Omega_1^1 = \{6, 4\}$, $\Omega_2^1 = \{6, 2\}$, and $\Omega_3^1 = \{1, 7\}$. Sensors $p \geq 2$ send the $(n,\ \mathbf{w}_t^p(n))$ pairs for $n \in \Omega_p^1$ to Sensor 1. In Step II, Sensor 1 receives the data and computes a partial sum for each $n \in \bigcup_{p=1}^P \Omega_p^1 = \{1, 2, 4, 6, 7\}$. As we can see, the indicies of the 2 largest-in-magnitude partial sums are 6 and 7, so sensor 1 broadcasts $F^1 = \{6, 7\}$ to other sensors. In Step III, Sensor 2 and 3 send $\mathbf{w}_t^2(7)$ and $\mathbf{w}_t^3(6)$ to Sensor 1. In Step IV, Sensor 1 computes $\mathbf{f}_t(n) = \sum_{p=1}^P \mathbf{w}_t^p(n)$ for $n \in F^1$ by (3.251), gets $\nu_1 = 21$, which is the second largest-in-magnitude total score for $n \in F_1$, and broadcasts it to other sensors. Step V

to VIII are very similar to GCAMP except for the threshold for the upper bounds $U(n)$ obtained. Instead of directly applying ν_1 as the threshold, as in GCAMP, we compute a lower bound $L(n)$ for each $n \in [N] \setminus F^1$, and get the second largest lower bound $\nu_2 = 0$, and the threshold for $U(n)$ is $\nu = \max(\nu_1, \nu_2) = 21$. Finally, we obtain the non-zero components of $\mathbf{x}_{t+1}$, which are $\mathbf{x}_{t+1}(6) = 23$ and $\mathbf{x}_{t+1}(7) = -21$; meanwhile, we save all the total scores we calculated in $\mathbf{x}^h_{t+1}$, which are $\mathbf{x}^h_{t+1}(6) = 23$, $\mathbf{x}^h_{t+1}(7) = -21$ and $\mathbf{x}^h_{t+1}(4) = 7$. Overall, in this example, 10 data points are sent from the sensors to sensor 1, and the total number of messages is 14 (10 data points plus 4 broadcast requests).

4.2.1.1 The step size μ in DIHT

Theoretically, μ in AIHT can be any value within $(0, 1/\|\mathbf{A}\|_2)$, while from the convergence perspective, μ should be as large as possible. Therefore, in the centralized AIHT, we can just set μ to be $1/\|\mathbf{A}\|_2$ minus a very small positive number. However, this is intractable in DIHT, since the exact computation of $1/\|\mathbf{A}\|_2$ needs the access to the entire global sensing matrix, and this contradicts the basic assumption of our proposed DCS framework. An alternative was to get an upper bound on $\|\mathbf{A}\|_2$, as proposed in [43], in which each Sensor $p \geq 2$ gets the ℓ_2 norm of its own sensing matrix $\mathbf{A}^p$, and sends $\|\mathbf{A}^p\|_2$ to Sensor 1, then Sensor 1 can get $\sqrt{\sum_{p=1}^{P} \|\mathbf{A}^p\|_2^2}$, which is an upper bound on $\|\mathbf{A}\|_2$, sets $\mu = 1/\sqrt{\sum_{p=1}^{P} \|\mathbf{A}^p\|_2^2}$, and broadcasts μ to other sensors. However, this approach has a very conservative upper bound for $\|\mathbf{A}\|_2$ and leads to a much smaller μ than the centralized IHT can get. Here, we propose a new approach which gives a very good estimate of $\|\mathbf{A}\|_2$, by applying the random matrix theory (RMT).

Let $\mathbf{G} = \mathbf{A}\mathbf{A}^T$ be the Gram matrix of rows of $\mathbf{A}$, and let $L_1 > \cdots > L_M$ be the M eigenvalues of $\mathbf{G}$, then $\|\mathbf{A}\|_2 = \sqrt{L_1}$. In RMT, for $\mathbf{A} := [a_{ij}]_{M \times N}$ with $a_{ij} \sim$ i.i.d.

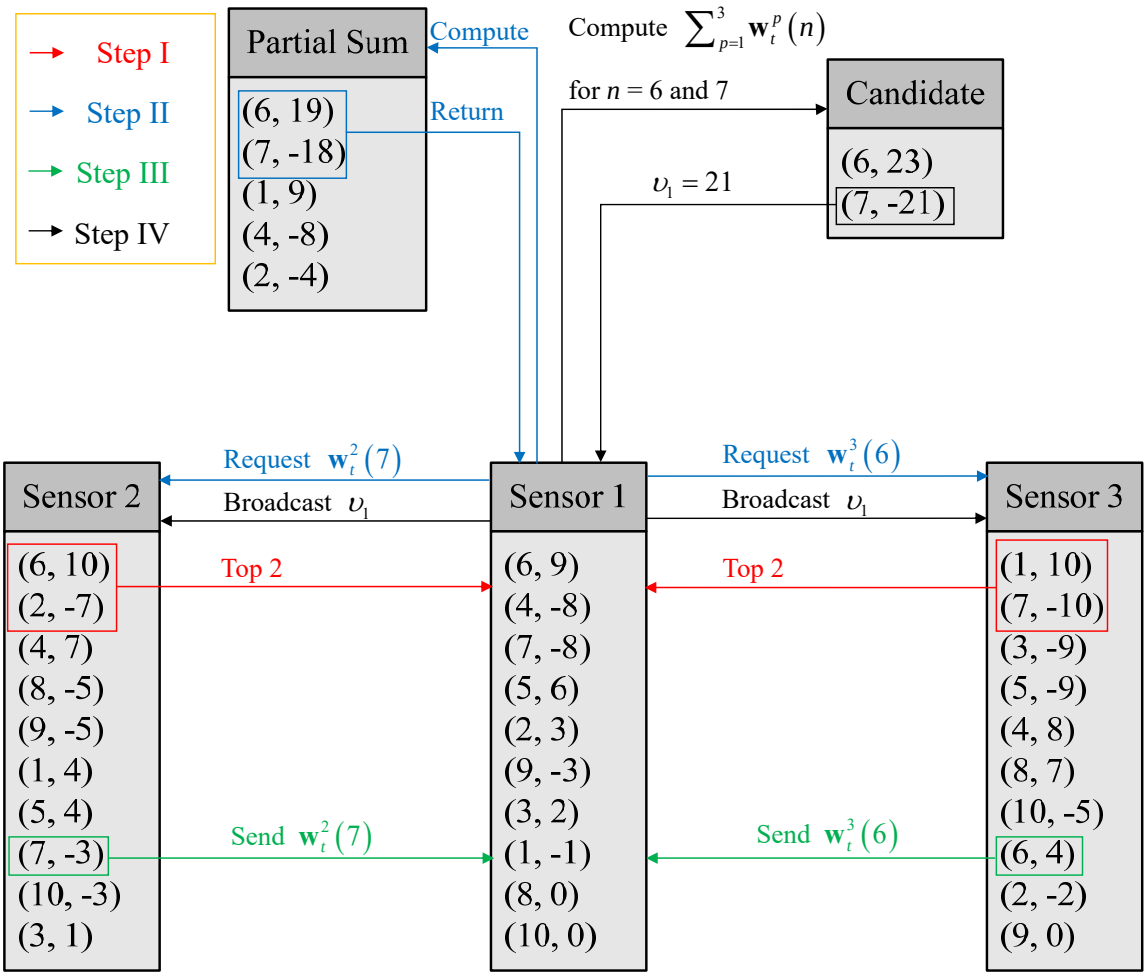

Fig. 13. Step I $\sim$ IV of GC.K algorithm

$\mathcal{N}(0, 1/M)$, the corresponding Gram matrix $\mathbf{G}$ is called Wishart matrix. According to [92], the largest eigenvalue L_1 of $\mathbf{G}$ has the following almost sure (a.s.) convergence in the large system limit:

$$L_1 \xrightarrow{\text{a.s.}} (1 + \sqrt{N/M})^2 \tag{4.17}$$

This implies that for large M and N, L_1 will become very close to a deterministic number, which only depends on the ratio between M and N, that is, the variability of L_1 vanishes in the large system limit. Further, the distribution of L_1 was studied

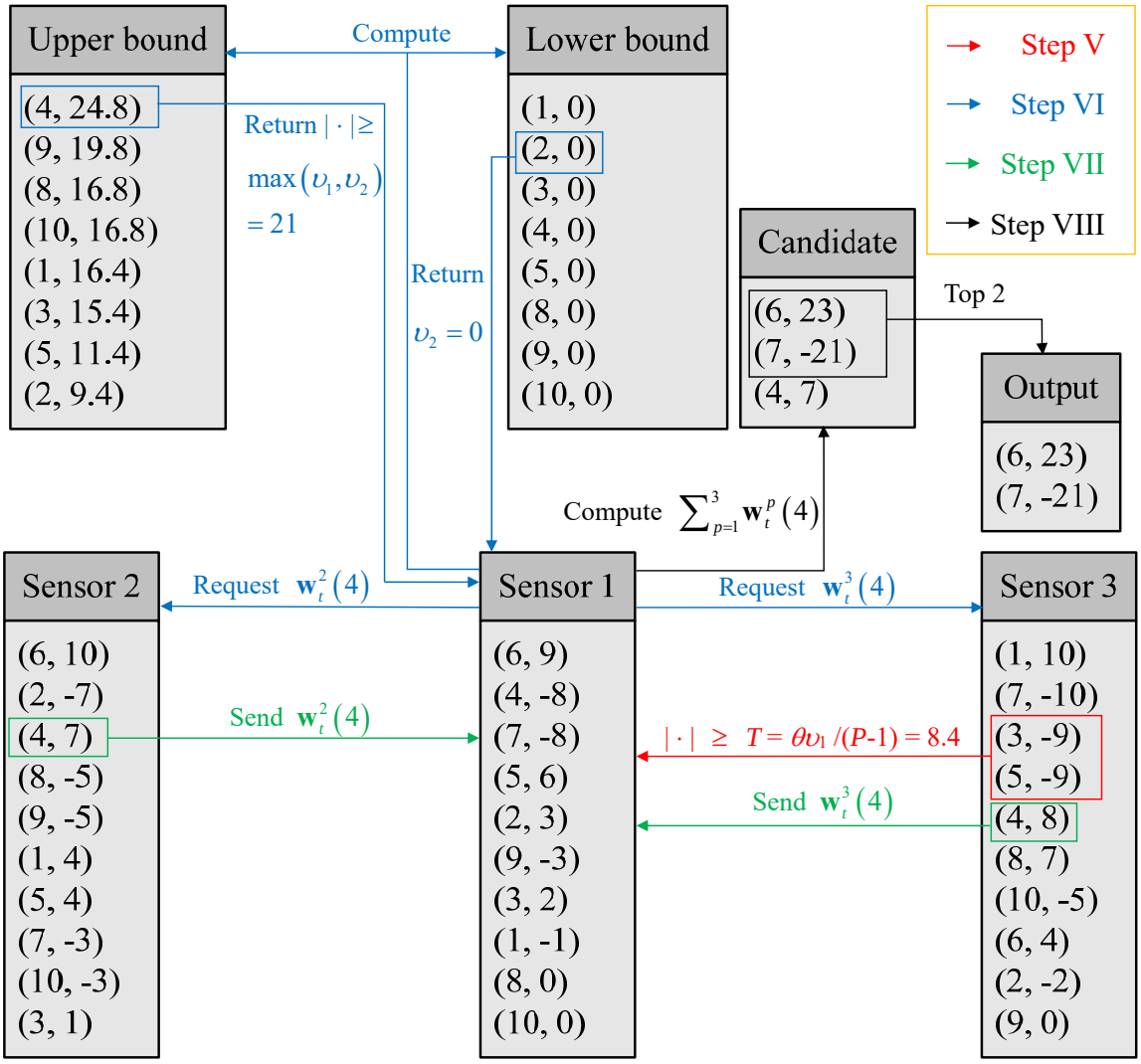

Fig. 14. Step V ∼ VIII of GC.K algorithm

in [93] and the following results were given: Let us define

$$\mu_{MN} = (1 + \sqrt{(N-1)/M})^2 \tag{4.18}$$

$$\sigma_{MN} = \frac{\sqrt{M} + \sqrt{N-1}}{M} \left(\frac{1}{\sqrt{M}} + \frac{1}{\sqrt{N-1}} \right)^{1/3} \tag{4.19}$$

then

$$L_1 \xrightarrow{\mathcal{D}} \mu_{MN} + \sigma_{MN} T_1 \text{ with } T_1 \sim F_1 \tag{4.20}$$

where F_1 is the cumulative distribution function (CDF) of Tracy-Widom (TW) law of order 1 [94]:

$$F_1(s) = \exp\left\{-\frac{1}{2}\int_s^\infty \left[q(x) + (x-s)q(x)^2\right] dx\right\} \tag{4.21}$$

where $q(s)$ is the solution to the Painlev type II equation

$$q''(s) = sq(s) + 2q(s)^3 \tag{4.22}$$

with the boundary condition

$$q(s) \sim \mathrm{Ai}(s), \text{ as } s \to +\infty \tag{4.23}$$

where $\mathrm{Ai}(s)$ is Airy function defined as

$$\mathrm{Ai}(s) = \frac{1}{3}\int_0^\infty \cos\left(\frac{t^3}{3} + st\right) dt \tag{4.24}$$

It has been shown that for $T_1 \sim F_1$, its mean is -1.21 and standard deviation is 1.27. In the large system limit, the standard deviation of L_1 approaches $1.27\sigma_{MN} \to 0$, which means that the distribution of L_1 will become more and more concentrated on its mean $\mu_{MN} - 1.21\sigma_{MN} \to (1+\sqrt{N/M})^2$ as the dimensionality of $\mathbf{A}$ increases. Taking advantage of the asymptotic deterministic property of L_1, we can get an approximate $q = 1-\alpha$ quantile for L_1 (α is a smaller number, e.g. 0.01), which is $L(\alpha) = \mu_{MN} + \sigma_{MN}F_1^{-1}(1-\alpha)$ and serves as a statistical upper bound on L_1. Due to the fact $\sigma_{MN} \to 0$, this bound will be very tight. We set the step length $\mu = 1/\sqrt{L(\alpha)}$. Note that each sensor can calculate μ which only depends on M and N, without any data transmission between sensors.

4.2.1.2 Numerical Results

We fix $N = 5,000$, set $M = N\kappa$ and $K = M\rho$, where $\kappa \in \{0.2, 0.3, 0.4, 0.5\}$ and $\rho \in \{0.1, 0.15, 0.2, 0.25\}$, and choose $P \in \{10, 15, \cdots, 50\}$. $\mathbf{s}_0$ is generated with random support and non-zero components drawn from $\mathcal{N}(0, 1)$. The noise $\mathbf{e} \sim \mathcal{N}(0, \sigma^2 I_M)$ with $\sigma \in \{0.01, 0.02, \cdots, 0.09\}$. IHT terminates if $\|\mathbf{x}_{t+1} - \mathbf{x}_t\|_2 \leq 0.001\|\mathbf{x}_t\|_2$ or if it runs up to 100 iterations. θ in GC.K is set to 0.8. We have the following setup: i) fix $(P, \sigma) = (10, 0.02)$, and change (κ, ρ); ii) fix $(\kappa, \rho, P) = (0.2, 0.1, 10)$, and change σ; iii) fix $(\kappa, \rho, \sigma) = (0.2, 0.1, 0.02)$, and change P. Under each parameter setting, we take $n_{\text{sim}} = 100$ Monte-Carlo runs.

For evaluating the communication cost, considering the approach sending all the data to Sensor 1, which has a total number of messages $N(P-1)$, we use the ratio between the number of messages of GC.K and $N(P-1)$, denoted as μ_M, to measure the efficiency of GC.K. After Sensor 1 obtains $\mathbf{x}_{t+1}$, it needs K messages to broadcast the non-zero components in $\mathbf{x}_{t+1}$ to other sensors. So we also define $T_M = \mu_M + K/[N(P-1)]$ to evaluate the performance of GC.K-based DIHT.

For MTA, as shown in Table 8, in each for-loop iteration inside the while-loop, the algorithm consumes $P+1$ messages, and there are totally N_s such iterations. So the number of messages in MTA is $N_s(P+1)$. It can be shown that if we run MTA on the data in Fig. 1, then we will get $N_s = 9$, which corresponds to $9 \times (3+1) = 36$ messages. After MTA terminates, each sensor has obtained the same $\mathbf{x}_{t+1}$, and there are no additional broadcast messages for the non-zero components of $\mathbf{x}_{t+1}$. Since the communication cost is proportional to $N_s \leq N$, we define μ_M for the MTA as $\mu_M = N_s/N$, and $T_M = N_s(P+1)/[N(P-1)]$. Note that the definitions of μ_M in GC.K and MTA are slightly different.

We first compare the GC.K-based DIHT.S and MTA-based DIHT.S. Since they

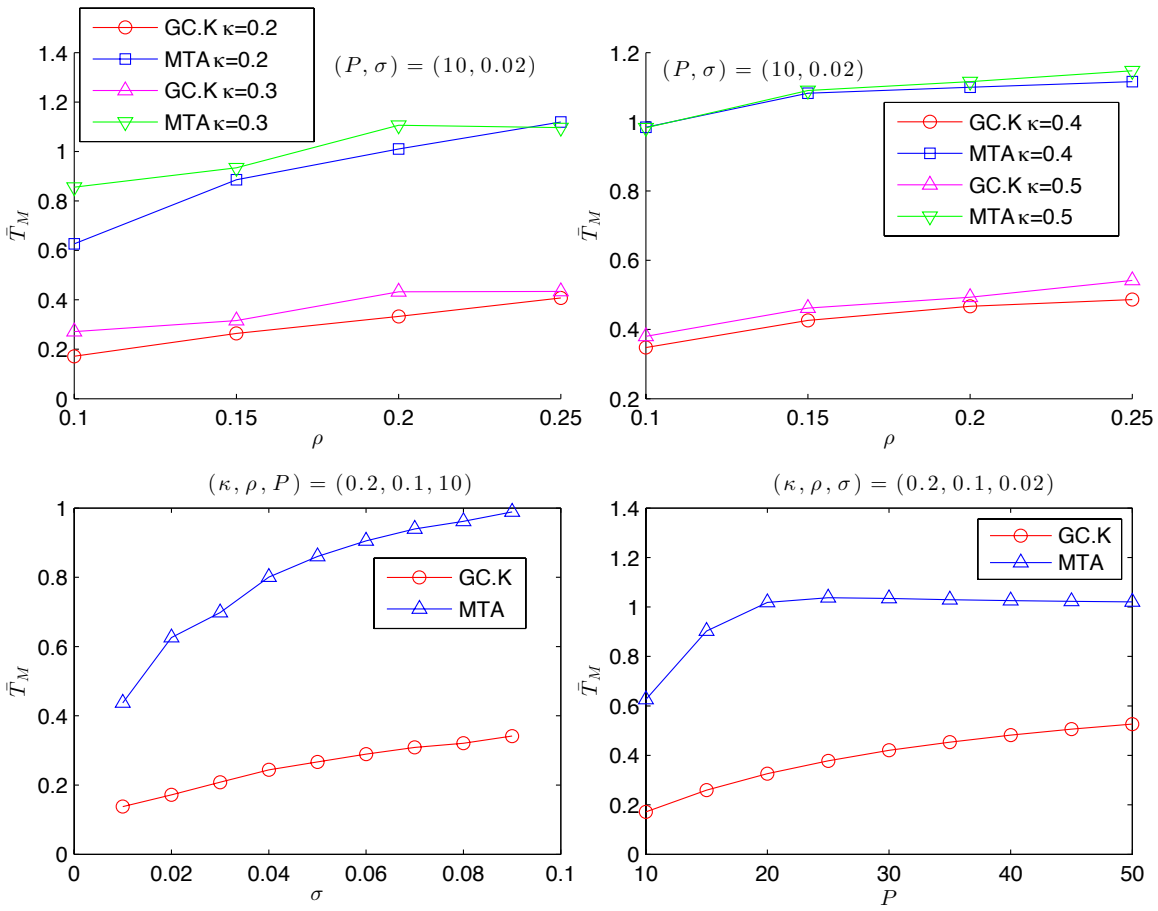

Fig. 15. Communication cost of GC.K and MTA.

have the same recovery results, we only compare their communication cost, i.e., μ_M and T_M defined above.

In Fig. 15 we show $\bar{T}_M$, the sample mean of T_M's, obtained by the two algorithms. As σ, P and K increase, the values of $\bar{T}_M$ in MTA become close to 1, which means that MTA hardly saves any communication cost, while GC.K can still work efficiently. In all the cases, GC.K outperforms MTA. In Fig. 16 the cumulative distributions of μ_M for GC.K and MTA are given under two extreme settings (large P and large K). In all iterations under these two settings, the number of messages in MTA are greater than $0.8N(P-1)$, while GC.K can save at least $0.35N(P-1)$ messages in 80% of the total iterations.

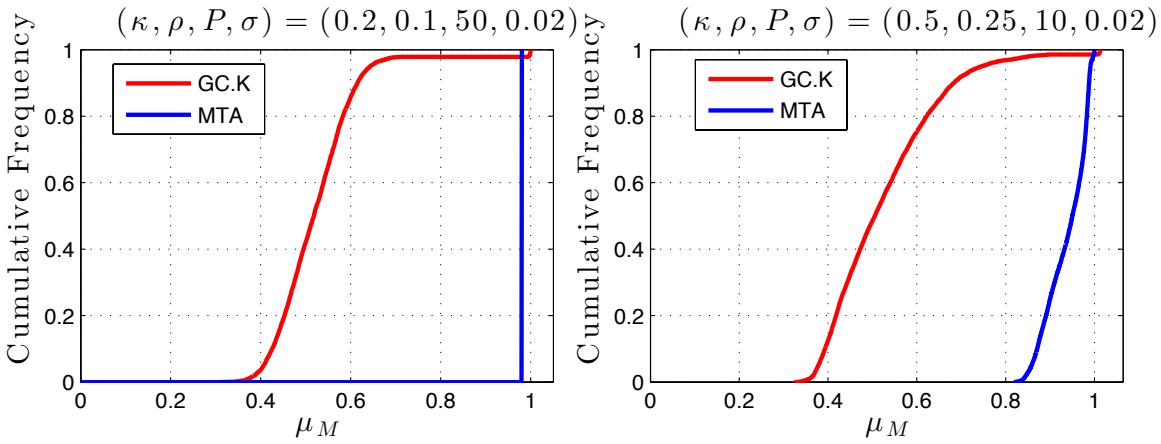

Fig. 16. Cumulative distributions of μ_M for GC.K and MTA.

Next, we compare GC.K-based DIHT.S with the oracle-aided approach GC.K-based DIHT.C, where $\|\mathbf{A}\|_2$ is known and $\mu = 0.99/\|\mathbf{A}\|_2$. The recovery accuracy is measured in terms of relative root mean squared error (RRMSE), which is defined as

$$\text{RRMSE} = \frac{\sqrt{\sum_{i=1}^{n_{\text{sim}}} ||(\mathbf{x}_i^* - \mathbf{s}_0)||_2^2 / n_{\text{sim}}}}{||\mathbf{s}_0||_2},$$

where $\mathbf{x}_i^*$ is the recovered signal in the i-th Monte-Carlo run. The convergence rate is evaluated in terms of $\bar{n}_{\text{iter}} := \sum_{i=1}^{n_{\text{sim}}} n_{\text{iter}}^i / n_{\text{sim}}$, where n_{iter}^i is the number of iterations in the i-th Monte-Carlo run. In Fig. 17 we show these quantities as well as the communication cost of DIHT.S and DIHT.C respectively, under all parameter settings, where $\bar{\mu}_M$ denotes the sample mean of μ_M's. As we can see, DIHT.S performs similarly to DIHT.C.

We also observe the ratios $\bar{\mu}_M / \bar{T}_M$ for GC.K under all parameter settings, and find that they are within the interval $[0.9771, 0.9989]$, which means that GC.K incurs most of the communication cost in the corresponding DIHT algorithms.

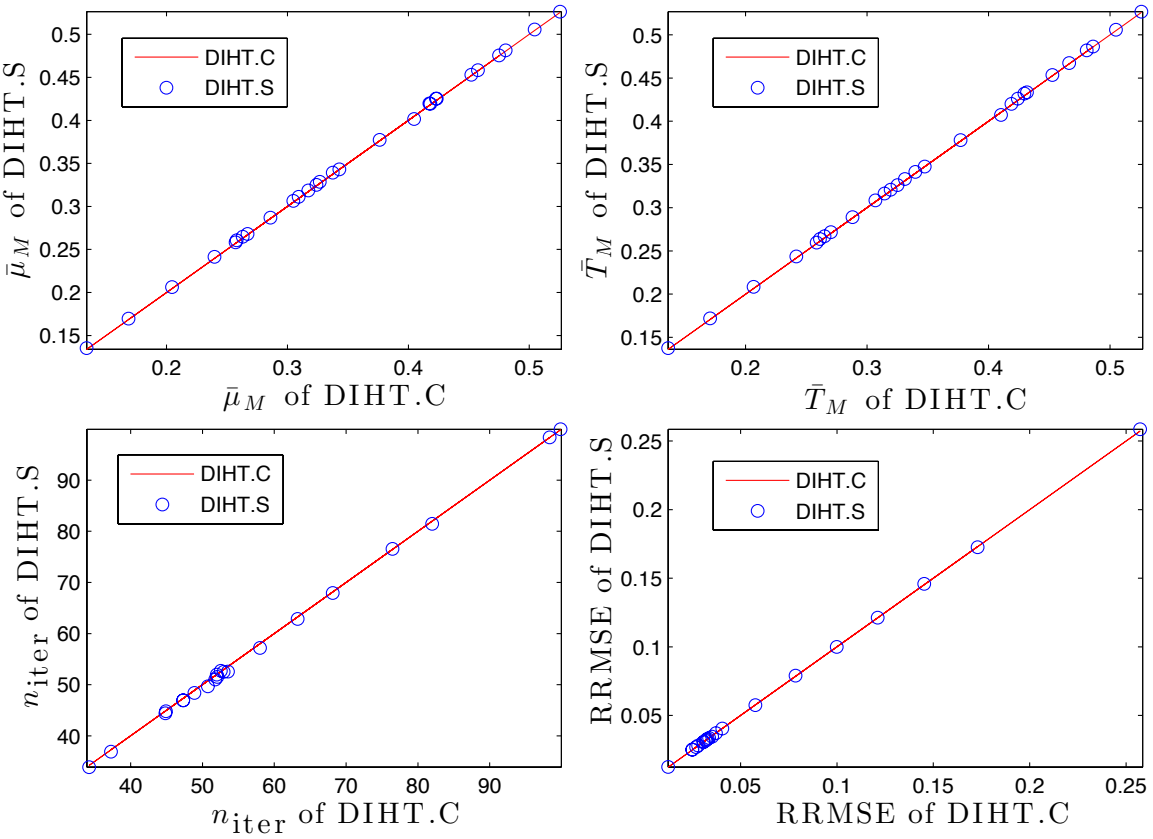

Fig. 17. Comparison of DIHT.S and DIHT.C.

4.2.2 Sign-Aware Data Querying

4.2.2.1 Improvement on GC.K

GC.K uses the number of communication messages (transmitted data points) as the metric of communication cost. Similar to SAGC, an improvement on GCAMP in Chapter 2, we notice that if we evaluate the communication cost in terms of the number of communication bits, which is more practical and informative than the number of communication messages, and dissertation taking advantage of the structure of transmitted data, then a further improvement on GC.K is achievable.

Here, we propose a new distributed data querying algorithm, sign-aware data querying (SADQ), which obtains an upper bound on $|\mathbf{f}_t(n)|$ and a lower bound on

$\mathcal{T}_K(\mathbf{f}_t)$, and decides whether to transmit all the $\mathbf{w}_t^p(n)$'s or not by comparing the two bounds. When calculating the number of bits needed in communication, we assume that all the entries in $\mathbf{w}_t^p$ are stored as 64-bits floating-pointing numbers, with 1-bit sign and 63-bits magnitudes. The SADQ algorithm contains the following 3 major steps:

Step I: The key idea of this step is to obtain a Top-K candidate in $\mathbf{f}_t$. First, each sensor $p \geq 2$ sends a package as shown in Fig. 18 to Sensor 1.

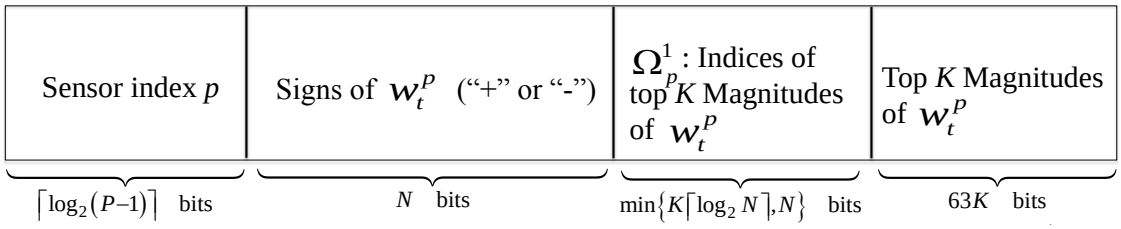

Fig. 18. The structure of the package Sensor p sends to Sensor 1 in Step I

Note that there are two ways for Sensor p to send Ω_p^1: It can either directly represent each index in Ω_p^1 using binary codes, or use a 1/0 flag to denote whether each index $n = 1, 2, \cdots, N$ is in Ω_p^1. The former requires $K\lceil \log_2 N \rceil$ bits, and the latter requires N bits. Sensor p will compare them and choose the one with fewer bits.

Let $O = \bigcup_{p=1}^P \Omega_p^1$ and define $C_n = \{p \geq 2 : n \in \Omega_p^1\}$ for each $n \in O$, Sensor 1 computes $P(n) = \mathbf{w}_t^1(n) + \sum_{p \in C_n} \mathbf{w}_t^p(n)$ for each $n \in O$. Denote F^1 as the set of indices for the largest-in-magnitude K $P(n)$'s. Sensor 1 broadcasts F^1 to other sensors, similar to encoding Ω_p^1, it takes $\min\{K\lceil \log_2 N \rceil, N\}$ bits to represent F^1.

Now Sensor 1 knows F^1, Ω_p^1 and $\mathbf{w}_t^p(\Omega_p^1)$, Sensor p only needs to send the sensor index p, and the magnitudes of $\mathbf{w}_t^p(F^1 \backslash \Omega_p^1)$ to Sensor 1, i.e., there is no need to send the indices set $F^1 \backslash \Omega_p^1$. So it takes $\lceil \log_2(P-1) \rceil + 63|F^1 \backslash \Omega_p^1|$ bits to send these data from Sensor p to 1. Finally, Sensor 1 computes $\mathbf{f}_t(F^1)$, obtains $\nu_1 = \mathcal{T}_K(\mathbf{f}_t(F^1))$, and

broadcasts it to other sensors, which requires 63 bits.

The number of required communication bits in this step is

$$
\begin{aligned}
B_1 = & \\
& \left[\lceil \log_2(P-1) \rceil + N + \min\{K\lceil \log_2 N \rceil, N\} + 63K\right](P-1) \text{ Sensor } p \text{ to } 1 \\
& + \min\{K\lceil \log_2 N \rceil, N\} \text{ Sensor 1 to } p \\
& + (P-1)\lceil \log_2(P-1) \rceil + 63 \sum_{p=2}^{P} |F^1 \backslash \Omega_p^1| \text{ Sensor } p \text{ to } 1 \\
& + 63 \text{ Sensor 1 to } p
\end{aligned}
$$

Step II: This step is to bound $|\mathbf{f}_t(n)|$ for each $n \notin F^1$.

Set $T = \theta\nu_1/(P-1)$, where θ is a parameter trading off the communication cost of the current step and the following step. Each sensor $p \geq 2$ sends the package as shown in Fig. 19 to Sensor 1.

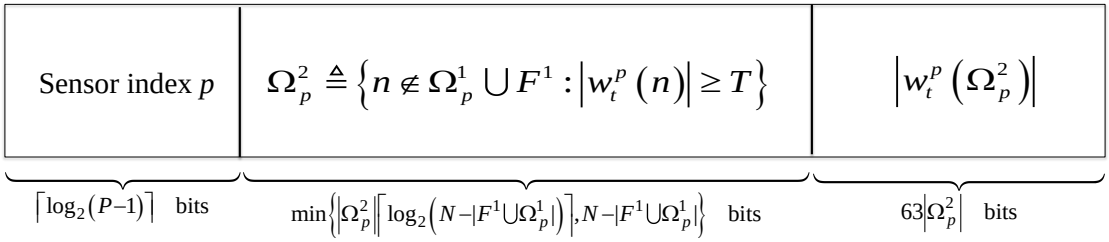

Fig. 19. The structure of the package Sensor p sends to Sensor 1 in Step II

Define $\Omega_p := \Omega_p^1 \bigcup \Omega_p^2 \bigcup F^1$ for each $p \geq 2$, which contains the indices of the entries Sensor p has sent to Sensor 1 by now, and $S_n := \{p \geq 2 : n \in \Omega_p\}$ for each $n \notin F^1$. Sensor 1 obtains a range $[L_t^p(n), U_t^p(n)]$ of $\mathbf{w}_t^p(n)$ for $n \notin \Omega_p$: $[0, T]$ if $\mathbf{w}_t^p(n) > 0$ and $[-T, 0]$ otherwise.

Next, Sensor 1 obtains a range $\left[B^L(n), B^U(n)\right]$ of $\mathbf{f}_t(n)$ as $B^L(n) = \mathbf{w}_t^1(n) + \sum_{p \in S_n} \mathbf{w}_t^p(n) + \sum_{p \notin S_n} L_t^p(n)$ and $B^U(n) = \mathbf{w}_t^1(n) + \sum_{p \in S_n} \mathbf{w}_t^p(n) + \sum_{p \notin S_n} U_t^p(n)$.

Therefore, we can compute an upper bound $U(n)$ and a lower bound $L(n)$ on $|\mathbf{f}_t(n)|$ for each $n \notin F^1$ based on the range. Denote ν_2 as the K-th largest $L(n)$ and set $\nu = \max(\nu_1, \nu_2)$, which is a lower bound on $\mathcal{T}_K(\mathbf{f}_t)$. Sensor 1 obtains the set $F^2 = \{n \notin F^1 : U(n) \geq \nu\}$, and sends F^2 to other sensors.

Similar to Step I, we can compute the number of bits in this step:

$$B_2 = (P-1)\lceil \log_2(P-1) \rceil + 63 \sum_{p=2}^{P} |\Omega_p^2| +$$

$$\sum_{p=2}^{P} \min(\lceil \log_2 \left(N - |F^1 \bigcup \Omega_p^1| \right) \rceil |\Omega_p^2|, N - |F^1 \bigcup \Omega_p^1|) \text{ Sensor } p \text{ to } 1$$

$$+ \min\{|F^2| \lceil \log_2(N-K) \rceil, N-K\}. \text{ Sensor 1 to } p$$

Step III: Each sensor $p \geq 2$ sends its sensor index p and $\mathbf{w}_t^p(F^2 \backslash \Omega_p)$ to Sensor 1. Sensor 1 computes $\mathbf{f}_t(n)$ for each $n \in F^2$. It can be shown that $F = F^1 \bigcup F^2$ contains all the indices of the $\mathbf{f}_t(n)$'s such that $|\mathbf{f}_t(n)| \geq \mathcal{T}_K(\mathbf{f}_t)$. Finally, Sensor 1 broadcasts the largest-in-magnitude K $\mathbf{f}_t(n)$'s as well as their indices to other sensors.

The number of required communication bits in this step is

$$B_3 = (P-1)\lceil \log_2(P-1) \rceil + 63 \sum_{p=2}^{P} |F^2 \backslash \Omega_p| \text{ Sensor } p \text{ to } 1$$

$$+ \min(K \lceil \log_2 |F| \rceil, |F|) + 64K. \text{ Sensor 1 to } p$$

In a naive approach in which other sensors directly send all the 64-bits numbers to Sensor 1, the communication cost can be easily derived and denoted as $B_{\max}$. We define the normalized number of bits required by SADQ as $n_B = \sum_{i=1}^{3} B_i / B_{\max}$.

4.2.2.2 Numerical Results

We set $N = 5,000$, $M = 1,000$, and $K = 100$. $\mathbf{s}_0$ is generated with random support and non-zero components $\sim$ i.i.d. $\mathcal{N}(0,1)$. The noise $\mathbf{e} \sim \mathcal{N}(0, \sigma^2 I)$ with

Table 10. Average communication costs of SADQ-based DIHT and MTA-based DIHT

P	5	10	15	20	25
θ^* in SADQ	0.8	1.1	1.2	1.3	1.4
$\bar{n}_B$ in SADQ	0.0984	0.1355	0.1704	0.2006	0.2288
$\bar{n}_B$ in MTA	0.2355	0.7116	1.0228	1.0962	1.0938
P	30	35	40	45	50
θ^* in SADQ	1.4	1.5	1.5	1.5	1.6
$\bar{n}_B$ in SADQ	0.2505	0.2708	0.2861	0.3026	0.3161
$\bar{n}_B$ in MTA	1.0801	1.0685	1.0597	1.0529	1.0475

$\sigma = 0.02$. We tune θ in SADQ, and obtain the optimal θ^*, as well as the corresponding mean value of n_B, ($\bar{n}_B$) based on 100 Monte-Carlo runs. We also implement the MTA, another data querying algorithm proposed in [40]. As shown in Table 10, the SADQ-based DIHT outperforms the MTA-based DIHT significantly in terms of communication savings.

4.3 Improvement on GC.K: Adaptive Approach

IHT needs the sparsity level K of $\mathbf{s}_0$ as an input parameter, which is unrealistic for most real-world applications. However, a good property is that IHT will always converge, regardless of whether K is the real sparsity level of $\mathbf{s}_0$. Due to this property, we can develop an adaptive IHT algorithm, which can tune K itself, i.e., can infer the sparsity level of $\mathbf{s}_0$ purely based on the measurement $\mathbf{y}$ and the sensing matrix $\mathbf{A}$.

4.3.1 Adaptive IHT Algorithm without Prior Knowledge of K

Since IHT needs the sparsity level K as an input, we denote IHT with a speacific K as IHT_K. We first introduce the following two theoretical results in [45] and [95]:

Theorem 11 *(Theorem 5 in [45]) Given a noisy observation* $\mathbf{y} = \mathbf{A}\mathbf{s}_0 + \mathbf{e}$*, where* $\mathbf{s}_0$ *is* K*-sparse. If* $\mathbf{A}$ *has the RIP with* $\delta_{3K} < \frac{1}{\sqrt{32}}$*, then* IHT_K *will eventually convergence*

to $\mathbf{x}^*$ *such that*

$$\|\mathbf{s}_0 - \mathbf{x}^*\|_2 \leq 6\|\mathbf{e}\|_2 \tag{4.25}$$

Lemma 16 *(Lemma 6.1 in [95]) Given a noisy observation* $\mathbf{y} = \mathbf{A}\mathbf{s}_0 + \mathbf{e}$*, where* $\mathbf{s}_0$ *is an arbitrary vector. Let* $\mathbf{s}_0^K$ *be the best* K*-sparse approximation to* $\mathbf{s}_0$ *and* $\mathbf{s}_r = \mathbf{s}_0 - \mathbf{s}_0^K$*. If* $\mathbf{A}$ *has the RIP with* (K, δ_K)*, then* $\mathbf{y}$ *can be rewritten in terms of* $\mathbf{s}_0^K$ *such that*

$$\mathbf{y} = \mathbf{A}\mathbf{s}_0^K + \mathbf{A}\mathbf{s}_r + \mathbf{e} = \mathbf{A}\mathbf{s}_0^K + \tilde{\mathbf{e}} \tag{4.26}$$

where $\tilde{\mathbf{e}}$ *is bounded by*

$$\|\tilde{\mathbf{e}}\|_2 \leq \sqrt{1+\delta_K}\|\mathbf{s}_0 - \mathbf{s}_0^K\|_2 + \sqrt{1+\delta_K}\frac{\|\mathbf{s}_0 - \mathbf{s}_0^K\|_1}{\sqrt{K}} + \|\mathbf{e}\|_2 \tag{4.27}$$

According to Theorem 11 and Lemma 16, we can directly obtain the following theoretical result:

Corollary 5 *Given a noisy observation* $\mathbf{y} = \mathbf{A}\mathbf{s}_0 + \mathbf{e}$*, where* $\|\mathbf{s}_0\|_0 = K_0$*. Let* $\mathbf{x}^K$ *be the final estimate given by IHT*$_K$*, where* $K < K_0$*. If* $\mathbf{A}$ *has the RIP for sparsities with* $\delta_{3K_0} < \dfrac{1}{\sqrt{32}}$*, then we have the following error bound:*

$$\|\mathbf{s}_0 - \mathbf{x}^K\|_2 \leq 6\left(\sqrt{1+\delta_K}\|\mathbf{s}_0 - \mathbf{s}_0^K\|_2 + \sqrt{1+\delta_K}\frac{\|\mathbf{s}_0 - \mathbf{s}_0^K\|_1}{\sqrt{K}} + \|\mathbf{e}\|_2\right). \tag{4.28}$$

Proposition 13 *A necessary condition for any CS recovery algorithm is* $M \geq 2K$*.[96]*

According to Corollary 5, even if $\|\mathbf{s}_0\|_0 > K$, IHT$_K$ will still recover $\mathbf{s}_0$ with a bounded error. Similar to the tuning procedure for the parameter τ in AMP, we derive the following adaptive IHT (AIHT) algorithm, which runs the original IHT as a subroutine for a list of candidate sparsity levels $\{K_\ell\}_{\ell=1}^L$, where $K_1 < K_2 < \cdots < K_L = M/2$.

Lemma 17 *(Theorem 4 in [44]) For any given* $K \leq N$*, the sequence* $\|\mathbf{y} - \mathbf{A}\mathbf{x}_t\|^2$ *in IHT*$_K$ *is non-increasing and converges to a local minimum of* (1.2)*.*

Table 11. Adaptive IHT algorithm

Input $\mathbf{y}$, $\mathbf{A}$, $K_1, \cdots, K_L$, T_1, T_2, ϵ_1, ϵ_2

Initialization $\mathbf{x}_0 = 0, \mathbf{z}_0 = \mathbf{y}$
for $\ell = 1{:}L$
  initialize $t_\ell = 0$;
  while $t_\ell < \max\{T_2, (T_1 - \sum_{i=1}^{\ell-1} t_i)/(\ell - 1)\}$
    $\mathbf{x}_{t_\ell+1} = \eta(\mathbf{x}_{t_\ell} + \mathbf{A}^T \mathbf{z}_{t_\ell}; K_\ell)$;
    $\mathbf{z}_{t_\ell+1} = \mathbf{y} - \mathbf{A}\mathbf{x}_{t_\ell+1}$;
    if $\|\mathbf{x}_{t_\ell+1} - \mathbf{x}_{t_\ell}\| \leq \epsilon_1 \|\mathbf{x}_{t_\ell}\|$
      $\mathbf{x}(K_\ell) = \mathbf{x}_{t_\ell+1}, \mathbf{z}(K_\ell) = \mathbf{z}_{t_\ell+1}$
      break
    else
      update $t_\ell \leftarrow t_\ell + 1$
      continue
    endif
  endwhile
  if $\|\mathbf{x}(K_\ell) - \mathbf{x}(K_{\ell-1})\| \leq \epsilon_2 \|\mathbf{x}(K_{\ell-1})\|$
    $\mathbf{x}^* = \mathbf{x}_{K_\ell}$, $K^* = K_\ell$
    break;
  else
    update $\mathbf{x}_0 = \mathbf{x}_{t_\ell+1}, \mathbf{z}_0 = \mathbf{z}_{t_\ell+1}$;
  endif
endfor

Output $\mathbf{x}^*$, K^*.

Regarding the performance of AIHT, we have the following theorem.

Theorem 12 *Given a noisy observation* $\mathbf{y} = \mathbf{A}\mathbf{s}_0 + \mathbf{e}$*, where* $\|\mathbf{s}_0\|_0 = K$*. The sequence* $\|\mathbf{y} - \mathbf{A}\mathbf{x}(K_\ell)\|^2$ *in AIHT is non-increasing. Furthermore, if AIHT returns* $\mathbf{x}^*$ *with* $\|\mathbf{x}^*\|_0 = K^*$*, and* $\mathbf{A}$ *has RIP with* $\delta_{3K} < 1/\sqrt{32}$ *and* $\delta_{3K^*} < 1/\sqrt{32}$*, then*

$$\|\mathbf{x}^* - \mathbf{s}_0\| \leq \begin{cases} 6.51\|\mathbf{s}_0 - \mathbf{s}_0^{K^*}\|_2 + 6.51\|\mathbf{s}_0 - \mathbf{s}_0^{K^*}\|_1/\sqrt{K^*} + 6\|\mathbf{e}\|_2 & \text{if } K^* \leq K \\ 7.1\|\mathbf{e}\|_2 & \text{if } K^* > K. \end{cases} \tag{4.29}$$

Proof of Theorem 12: The ℓ-th outer loop of AIHT runs IHT_{K_ℓ}, which starts from $\mathbf{x}(K_{\ell-1})$ with sparsity level $K_{\ell-1}$, which is a feasible solution of the optimization problem

$$\min_{\|\mathbf{x}\|_0 \le K_\ell} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2 \tag{4.30}$$

since $K_{\ell-1} < K_\ell$. By Lemma 17, the value of the objective function keeps non-increasing and IHT_{K_ℓ} will converge to a local optimum $\mathbf{x}(K_\ell)$. Therefore, we have $\|\mathbf{y} - \mathbf{A}\mathbf{x}(K_{\ell-1})\|^2 \ge \|\mathbf{y} - \mathbf{A}\mathbf{x}(K_\ell)\|^2$.

If $K^* \le K$, according to Lemma 5, we have

$$\|\mathbf{s}_0 - \mathbf{x}^*\| \le 6\|\widetilde{\mathbf{e}}\|_2 \le 6.51\|\mathbf{s}_0 - \mathbf{s}_0^{K^*}\|_2 + 6.51\|\mathbf{s}_0 - \mathbf{s}_0^{K^*}\|_1/\sqrt{K^*} + 6\|\mathbf{e}\|_2. \tag{4.31}$$

If $K^* > K$, letting $\mathbf{s}_e = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{e}$, we have $\mathbf{A}\mathbf{s}_e = \mathbf{e}$. Further, letting $\mathbf{s}_e^{K^*-K}$ be the best (K^*-K)-sparse approximation of $\mathbf{s}_e$, we have

$$\mathbf{y} = \mathbf{A}(\mathbf{s}_0 + \mathbf{s}_e) = \mathbf{A}(\mathbf{s}_0 + \mathbf{s}_e^{K^*-K}) + \mathbf{A}(\mathbf{s}_e - \mathbf{s}_e^{K^*-K}). \tag{4.32}$$

Since $\mathbf{s}_0$ is K-sparse and $\mathbf{s}_e^{K^*-K}$ is (K^*-K)-sparse, $\mathbf{s}_0+\mathbf{s}_e^{K^*-K}$ is K^*-sparse. Applying Lemma 17, we have

$$\|\mathbf{s}_0 - \mathbf{x}^* - \mathbf{s}_e^{K^*-K}\| \le 6\|\mathbf{A}(\mathbf{s}_e - \mathbf{s}_e^{K^*-K})\|_2 \le 6\|\mathbf{A}\mathbf{s}_e\|_2 = 6\|\mathbf{e}\|_2. \tag{4.33}$$

By the RIP property,

$$\|\mathbf{s}_e^{K^*-K}\|_2 \le \|\mathbf{A}\mathbf{s}_e^{K^*-K}\|_2/\sqrt{1-\delta_{K^*-K}} \le 1.1\|\mathbf{A}\mathbf{s}_\mathbf{e}\|_2 = 1.1\|\mathbf{e}\|_2. \tag{4.34}$$

Therefore,

$$\|\mathbf{s}_0 - \mathbf{x}^*\|_2 \le \|\mathbf{s}_0 - \mathbf{x}^* - \mathbf{s}_e^{K^*-K}\|_2 + \|\mathbf{s}_e^{K^*-K}\|_2 \le 7.1\|\mathbf{e}\|_2. \tag{4.35}$$

4.3.2 More Efficient GC Algorithms for DIHT

Similar to GCAMP in the previous chapter, GC.K and SADQ for DIHT are non-adaptive approaches. If we can propose an adaptive approach that takes advantage of the strong correlation between $\mathbf{W}_t$ and $\mathbf{W}_{t-1}$, then we may obtain significant communication savings.

Like GCAMP, GC.K and SADQ have the following outcomes:

i) A new sparse estimate $\mathbf{x}_{t+1}$ with support $\Gamma \subset F$.

ii) A group of total scores $\mathbf{f}_t(n)$ for $n \in F$.

iii) A lower bound ν for $\mathcal{T}_K(\mathbf{f}_t)$.

iv) A gap Δ_{t+1} between $\{|\mathbf{f}_t(n)| : n \in [N], |\mathbf{f}_t(n)| < \nu\}$ and ν.

The adaptive GC.K (A-GC.K) approach can use GC.K or SADQ as a subroutine, and utilize these outcomes as intermediate results.

4.3.3 A-GC.K Algorithm

The essence of the A-GC.K algorithm is quite similar to that of A-GCAMP. Suppose in the outer loop of AIHT, K is the current candidate sparsity, and at the inner loop iteration $t-1$, we obtained x_t as well as a group of total scores $\mathbf{f}_{t-1}(n)$ for $n \in I_t$ with magnitudes greater than a threshold $\gamma_{t-1} \leq \mathcal{T}_K(|\mathbf{f}_{t-1}|)$, and a gap Δ_t between $\{|\mathbf{f}_{t-1}(n)| : n \notin I_t\}$ and γ_{t-1}. Then at iteration t, we can first calculate the total scores for $n \in I_t$. Note that $|I_t| \geq K$, so we can obtain $\mathcal{T}_K(|\mathbf{f}_t(I_t)|)$, which is a tight lower bound on $\mathcal{T}_K(|\mathbf{f}_t|)$. Due to the linear convergence rate of AIHT, after a few iterations, $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ will be very close and may even have the same support. Therefore, $\mathcal{T}_K(|\mathbf{f}_t(I_t)|)$ is a tight lower bound. Then, we let $\gamma_t = \alpha_t \mathcal{T}_K(|\mathbf{f}_t(I_t)|)$, where $\alpha_t \in (0, 1)$, and run the A-GCAMP algorithm to get all the total scores for $n \notin I_t$ with magnitudes greater than γ_t.

Similar to the A-GCAMP algorithm, how to design the sequence $\{\alpha_t\}$ for $t \geq 1$ is also an important probelm in A-GC.K. First, we set up an upper bound $\alpha_{\max}$ for $\{\alpha_t\}$, where $\alpha_{\max}$ is close to 1, e.g., 0.99. Once α_{t-1} reaches $\alpha_{\max}$, we will reset $\alpha_t = \alpha_{\rm rst}$, where $\alpha_{\rm rst} \in (0,1)$ is a constant, and if $\alpha_{t-1} < \alpha_{\max}$, or if $t = 1$, we compute $\gamma_t^{'}$ as defined in (2.34), and obtain α_t as follows:

$$\alpha_t = \min\left(\frac{\gamma_t^{'}}{\mathcal{T}_K(|f_t(I_t)|)}, \alpha_{\max}\right). \tag{4.36}$$

Then, like in A-GCAMP, we can check whether (2.39) holds. If not, then we run the GCAMP to get all the total scores for $n \notin I_t$ with magnitudes greater than γ_t. If (2.39) holds, then we apply the A-GCAMP algorithms to get these total scores, which will save a lot of communication cost.

In Tables 12 and 13 we give the pseudo code of the A-GC.K algorithm and the corresponding DAIHT algorithm respectively.

Table 12.: A-GC.K Algorithm

Input w_t^p, Δw_t^p, x_t^h, `rst`, γ_{t-1}, Δ_t, K, $\alpha_{\rm rst}$, $\alpha_{\max}$, q, θ, ρ_w

---

Set $I_t = \{n : |\mathbf{x}_t^h(n)| > \gamma_{t-1}\}$, $R_t = \{n : 0 < |\mathbf{x}_t^h(n)| \leq \gamma_{t-1}\}$, and $C_t^1 = [N] \setminus I_t$;

Initialize $\mathbf{x}_{t+1}^h = 0$ and $\Delta\mathbf{x}_{t+1}^h = 0$;

Obtain $\mathbf{x}_{t+1}^h(I_t) = \sum_{p=1}^P \mathbf{w}_t^p(I_t)$ and $\beta_t = \mathcal{T}_K(|\mathbf{x}_{t+1}^h(I_t)|)$;

if `rst` $= 1$

  $\alpha_t = \alpha_{\rm rst}$;

else

  Compute α_t according to (4.36);

endif

$\gamma_t = \alpha_t \beta_t$;

Compute $\Delta\gamma_t$ as defined in (2.34);

if $\Delta\gamma_t > 0$

for sensor $p = 1 : P$

obtain $\Delta\Omega_t^p$ as defined in (2.35);

if $p \geq 2$

send the cardinality $|\Delta\Omega_t^p|$ to sensor 1;

endif

endfor

$N^g = \sum_{p=1}^{P} |\Delta\Omega_t^p|$;

else

$N^g = |C_t^1|P$;

endif

if $N^g \leq \rho_w |C_t^1| P$

$\left[\sim, \mathbf{x}_{t+1}^h(C_t^1), \Delta_{t+1}^1\right] = \text{A-GCAMP}(\mathbf{w}_t^1(C_t^1), \cdots, \mathbf{w}_t^P(C_t^1),$

$\Delta\mathbf{w}_t^1(C_t^1), \cdots, \Delta\mathbf{w}_t^P(C_t^1), \mathbf{x}_t^h(C_t^1), \gamma_{t-1}, \alpha_t, \beta_t, \Delta_t, \theta, \rho_w)$;

else

$\left[\sim, \mathbf{x}_{t+1}^h(C_t^1), \Delta_{t+1}^1\right] = \text{GCAMP}(\mathbf{w}_t^1(C_t^1), \cdots, \mathbf{w}_t^P(C_t^1), \gamma_t, \theta)$;

endif

$\Delta_{t+1}^2 = \gamma_t - \max(\{|\mathbf{x}_{t+1}^h(n)| : n \in I_t, |\mathbf{x}_{t+1}^h(n)| \leq \gamma_t\})$;

$\Delta_{t+1} = \min(\Delta_{t+1}^1, \Delta_{t+1}^2)$;

$\mathbf{x}_{t+1} = \eta^H(\mathbf{x}_{t+1}^h; K)$;

Output $\mathbf{x}_{t+1}, \mathbf{x}_{t+1}^h, \gamma_t, \Delta_{t+1}, \alpha_t$

Table 13.: DAIHT Algorithm Based on A-GC.K

Input $\{\mathbf{y}^p\}_{p=1}^P$, $\{\mathbf{A}^p\}_{p=1}^P$, $\{K_\ell\}_{\ell=1}^L$, μ, α_{rst}, $\alpha_{\max}$, q, θ, ρ_w, T_1, T_2

Initialize $\mathbf{x}_0 = 0$, $\mathbf{z}_0 = \mathbf{y}$, $K = K_1$;

for $\ell = 1{:}L$

initialize $t_\ell = 0$;

while $t_\ell < \min\{T_2, T_1 - \sum_{i=1}^{\ell-1} T_i/(\ell-1)\}$

set $t = \sum_{i=1}^{\ell-1} t_i + t_\ell$;

if $t \geq 1$

Compute $\mathbf{w}_t^p$ and $\Delta\mathbf{w}_t^p = \mathbf{w}_t^p - \mathbf{w}_{t-1}^p$ by (4.5);

if $t \geq 2$ and $\alpha_{t-1} = \alpha_{\max}$

`rst`= 1;

else

`rst`= 0;

endif

$\left[\mathbf{x}_{t+1}, \mathbf{x}_{t+1}^h, \gamma_t, \Delta_{t+1}, \alpha_t\right] = \text{A-GC.}K(\mathbf{w}_t^1, \cdots, \mathbf{w}_t^P, \Delta\mathbf{w}_t^1, \cdots,$

$\Delta\mathbf{w}_t^P, \mathbf{x}_t^h$,`rst`,$\gamma_{t-1}, \Delta_t, K, \alpha_{\text{rst}}, \alpha_{\max}, q, \theta, \rho_w)$;

else

$\left[\mathbf{x}_{t+1}, \mathbf{x}_{t+1}^h, \gamma_t, \Delta_{t+1}\right] = \text{GC.}K(\mathbf{w}_t^1, \cdots, \mathbf{w}_t^P, K, \theta)$;

endif

Compute $\mathbf{z}_{t+1}^p$ by (2.13) for each p

if $\|\mathbf{x}_{t+1} - \mathbf{x}_t\| \leq \epsilon_1 \|\mathbf{x}_t\|$

$\mathbf{x}(K_\ell) = \mathbf{x}_{t+1}, \mathbf{z}^p(K_\ell) = \mathbf{z}_{t+1}^p$ for $p = 1 \cdots P$

update $K = K_{\ell+1}$;

break;

else

$t_\ell \leftarrow t_\ell + 1$;

continue;

endif

endwhile

if $\|\mathbf{x}(K_\ell) - \mathbf{x}(K_{\ell-1})\| \leq \epsilon_2 \|\mathbf{x}(K_{\ell-1})\|$

set $\mathbf{x}^* = \mathbf{x}(K_\ell)$ and $K^* = K_\ell$;

break;

endif

endfor

Output $\mathbf{x}^*, \mathbf{z}^{p*}$

Regarding the accuracy of A-GC.K, we have the following theorem.

Theorem 13 *Given the same step size μ, A-GC.K algorithm obtains x_{t+1} which is exactly the same as that of the centralized AIHT algorithm computed by* (4.2).

The proof is very straightforward and is not presented in this dissertation due to the space limit.

Like DiAMP, we can also use the quantization procedure to further reduce the communication cost in GC.K and A-GC.K, namely, Q-GC.K and Q-A-GC.K.

4.4 Conclusion

In this chapter, we proposed an tuning algorithm for IHT which can tuning the sparsity level of $\mathbf{s}_0$ automatically. In the GC step of DIHT, we proposed a communication-efficient GC algorithm GC.K for DIHT. For the computation of the step size, we proposed a statistical approach DIHT.S which provides a very tight statistical upper bound on $\|\mathbf{A}\|_2$ that only depends on the dimensionality of $\mathbf{A}$. Like DiAMP, similar improvements can be made based on GC.K, by incorporating quantization, such as SADQ and Q-GC.K, and using adaptive approach, by first finding a lower bound on $\mathcal{T}_K(\mathbf{f}_t)$ and then performing the GC step of DIHT by A-GCAMP algorithm.

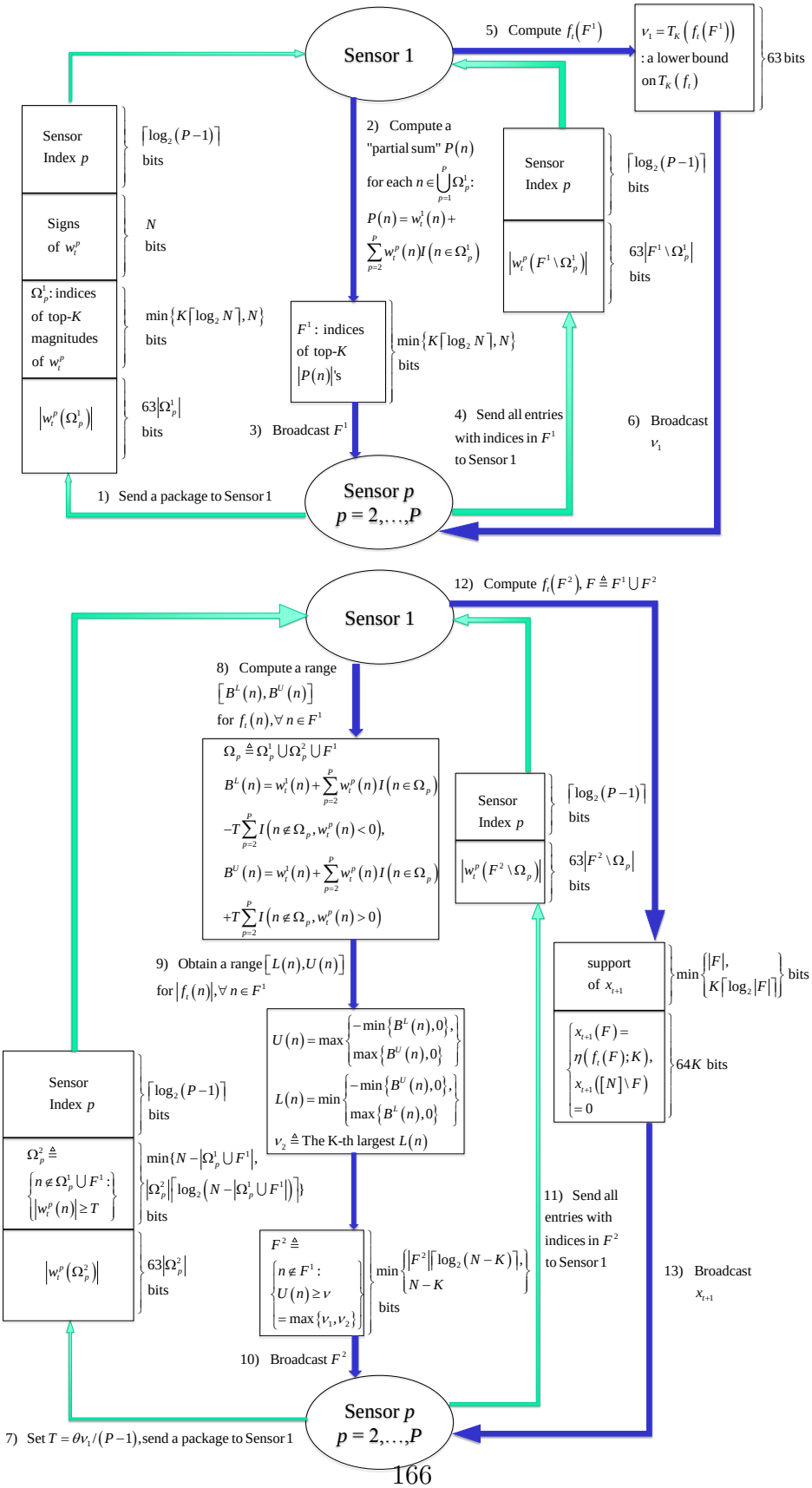

Fig. 20. SADQ Algorithm

CHAPTER 5

CONCLUSION

In this dissertation, two classes of distributed algorithms were developed for sparse signal recovery in large sensor networks. First, a distributed compressed sensing (DCS) framework was developed based on approximate message passing (AMP). The distributed AMP (DiAMP) framework does not need a prior knowledge of the sparsity of the original signal, and has exactly the same recovery result as the centralized AMP. Just like the centralized AMP, DiAMP is also a iterative approach, where each iteration contains a local computation (LC) step, and a global computation (GC) step, where the latter incurs communication among sensors. To reduce the communication cost in the global computation (GC) step, we developed GCAMP, which is a communication-efficient data-querying algorithm and significantly outperforms modified thresholding algorithm (MTA), another popular data query algorithm. By taking into consideration the correlation of data between adjacent iterations and incorporating quantization steps, a more sophisticated algorithm quantized-adaptive-GCAMP (Q-A-GCAMP) was developed, which comes close to requiring the minimum bit rates stipulated by the sparsity of the signal to be estimated.

Furthermore, we prove that state evolution (SE), a fundamental property of AMP that in high dimensionality limit, the output data are asymptotically Gaussian regardless of the distribution of input data, also holds for DiAMP, even in the presence of quantization noise. In addition, compared with the most recent theoretical results that SE holds for sensing matrices with independent subgaussian entries, we proved that the universality of SE can be extended to far more general sensing

matrices. These two theoretical results provided strong guarantee for AMP's performance, and greatly broaden its potential applications. As a direct application of this theoretical progress, we proposed a multi-processor approximate message passing framework with lossy compression. We used a uniform quantizer with entropy coding to reduce communication costs, and reformulated the state evolution formalism while accounting for quantization noise. Combining the quantizers and modified state evolution equation, an online back-tracking approach and another method based on dynamic programming were developed to determine the coding rate in each iteration by controlling the induced error. The numerical results suggest that our approaches can maintain a high signal-to-distortion-ratio despite a significant and often dramatic reduction in inter-processor communication costs.

Finally, in Chapter 4, another DCS approach was developed based on iterative hard thresholding (IHT). For distributed IHT (DIHT), there is a step size μ which depends on the ℓ_2 norm of the global sensing matrix $\mathbf{A}$. The exact computation of $\|\mathbf{A}\|_2$ is non-separable. We proposed a new method, based on the random matrix theory (RMT), to give a very tight statistical upper bound of $\|\mathbf{A}\|_2$, and the calculation of that upper bound is separable without any communication cost. Similar to DiAMP, DIHT also contains a LC step and a GC step in each iteration. In the GC step of DIHT, we developed another algorithm named GC.K, which is also communication-efficient and outperforms MTA. Then, by converting the GC step of DIHT into a GCAMP problem, we can apply the same improvement on GCAMP, such as the adaptive approach and quantization, to further improve the communication savings in DIHT.

REFERENCES

[1] D. L. Donoho. "Compressed sensing". In: *IEEE Trans. Inf. Theory* 52.4 (2006), pp. 1289–1306.

[2] E. J. Candes, J. K. Romberg, and T. Tao. "Stable signal recovery from incomplete and inaccurate measurements". In: *Comm. Pure Appl. Math.* 59.8 (2006), pp. 1207–1223.

[3] R. G. Baraniuk et al. "Model-based compressive sensing". In: *IEEE Trans. Inf. Theory* 56.4 (2010), pp. 1982–2001.

[4] M. F. Duarte and Y. C. Eldar. "Structured compressed sensing: From theory to applications". In: *IEEE Trans. Sig. Proc.* 59 (9 2011), pp. 4053–4085.

[5] B. Zhao, F. Li, and E. P. Xing. "Online detection of unusual events in videos via dynamic sparse coding". In: *IEEE Conf. Comput. Vis. Pattern Recogn. (CVPR)*. 2011, pp. 3313–3320.

[6] C. Li et al. "Abnormal behavior detection via sparse reconstruction analysis of trajectory". In: *IEEE Int. Conf. Image Graphics (ICIG)*. 2011, pp. 807–810.

[7] X. Mo et al. "Adaptive sparse representations for video anomaly detection". In: *IEEE Trans. Circuits Syst. Video Technol.* 24.4 (2014), pp. 631–645.

[8] Y. Liu, P. Ning, and M. K. Reiter. "False data injection attacks against state estimation in electric power grids". In: *ACM Trans. Inf. Syst. Security (TISSEC)* 14.1 (2011), p. 13.

[9] T. T. Kim and H. V. Poor. "Strategic protection against data injection attacks on power grids". In: *IEEE Trans. Smart Grid* 2.2 (2011), pp. 326–333.

[10] M. Ozay et al. “Sparse attack construction and state estimation in the smart grid: Centralized and distributed models”. In: *IEEE J. Sel. Areas Commun.* 31.7 (2013), pp. 1306–1318.

[11] J. Cao and Z. Lin. “Bayesian signal detection with compressed measurements”. In: *Inf. Sciences* 289 (2014), pp. 241–253.

[12] L. Anitori et al. “Compressive CFAR radar detection”. In: *IEEE Radar Conf. (RADAR)*. 2012, pp. 0320–0325.

[13] L. Anitori et al. “Design and analysis of compressed sensing radar detectors”. In: *IEEE Trans. Sig. Proc.* 61.4 (2013), pp. 813–827.

[14] R. G. Baraniuk and M. B. Wakin. “Random projections of smooth manifolds”. In: *Found. Comput. Math.* 9.1 (2009), pp. 51–77.

[15] M. Chen et al. “Compressive sensing on manifolds using a nonparametric mixture of factor analyzers: Algorithm and performance bounds”. In: *IEEE Trans. Sig. Proc.* 58.12 (2010), pp. 6140–6155.

[16] E. J. Candes. “Compressive sampling”. In: *Int. Cong. Math.* Vol. 3. Madrid, Spain, 2006, pp. 1433–1452.

[17] R. G. Baraniuk et al. “A simple proof of the restricted isometry property for random matrices”. In: *Constr. Approx.* 28 (3 2008), pp. 253–263.

[18] S. S. Chen, D. L. Donoho, and M. A. Saunders. “Atomic decomposition by basis pursuit”. In: *SIAM Rev.* 43 (2001), pp. 129–159.

[19] Lanchao Liu et al. “Detecting false data injection attacks on power grid by sparse optimization”. In: *IEEE Trans. Smart Grid* 5.2 (2014), pp. 612–621.

[20] O. Kosut et al. “Malicious data attacks on the smart grid”. In: *IEEE Trans. Smart Grid* 2.4 (2011), pp. 645–658.

[21] S. Boyd et al. "Distributed optimization and statistical learning via the alternating direction method of multipliers". In: *Foundations and Trends in Machine Learning* 3.1 (2011), pp. 1–122.

[22] X. Mo et al. "Low rank sparsity prior for robust video anomaly detection". In: *IEEE Int. Conf. Acoust., Speech, Sig. Proc. (ICASSP)*. 2014, pp. 1285–1289.

[23] Z. Ghahramani, G. E. Hinton, et al. *The EM algorithm for mixtures of factor analyzers*. Tech. rep. Technical Report CRG-TR-96-1, University of Toronto, 1996.

[24] M. Tipping and C. Bishop. "Mixtures of probabilistic principal component analyzers". In: *Neural computation* 11.2 (1999), pp. 443–482.

[25] J. Sethuraman. *A constructive definition of Dirichlet priors*. Tech. rep. DTIC Document, 1991.

[26] J. Paisley and L. Carin. "Nonparametric factor analysis with beta process priors". In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM. 2009, pp. 777–784.

[27] J. A. Bazerque and G. B. Giannakis. "Distributed spectrum sensing for cognitive radio networks by exploiting sparsity". In: *IEEE Trans. Sig. Proc.* 58.3 (2010), pp. 1847–1862.

[28] J. F. Mota et al. "Basis pursuit in sensor networks". In: *IEEE Int. Conf. Acoust., Speech, Sig. Proc. (ICASSP)*. 2011, pp. 2916–2919.

[29] C. Ravazzi, S. Fosson, and E. Magli. "Energy-saving gossip algorithm for compressed sensing in multi-agent systems". In: *IEEE Int. Conf. Acoust., Speech, Sig. Proc. (ICASSP)*. 2014, pp. 5060–5064.

[30] C. Lindberg, A. G. Amat, and H. Wymeersch. "Distributed compressed sensing for sensor networks with packet erasures". In: *IEEE Global Commun. Conf. (GLOBECOM),* 2014, pp. 13–19.

[31] G. Coluccia, C. Ravazzi, and E. Magli. *Compressed Sensing for Distributed Systems*. Springer, 2015.

[32] M. Masood. "Distribution Agnostic Structured Sparsity Recovery: Algorithms and Applications". PhD thesis. King Abdullah University of Science and Technology, 2015.

[33] D. Baron et al. "Distributed compressed sensing". In: 2005. URL: `http://dsp.rice.edu/sites/dsp.rice.edu/files/publications/report/2006/distributed-ece-2006.pdf`.

[34] D. Sundman, S. Chatterjee, and M. Skoglund. "A greedy pursuit algorithm for distributed compressed sensing". In: *IEEE Int. Conf. Acoust., Speech, Sig. Proc. (ICASSP)*. 2012, pp. 2729–2732.

[35] D. Sundman, S. Chatterjee, and M. Skoglund. "Parallel pursuit for distributed compressed sensing". In: *IEEE Global Conf. Sig. Inf. Proc. (GlobalSIP)*. IEEE. 2013, pp. 783–786.

[36] D. Sundman et al. "Distributed predictive subspace pursuit". In: *IEEE Int. Conf. Acoust., Speech, Sig. Proc. (ICASSP)*. 2013, pp. 4633–4637.

[37] J. Mota et al. "Distributed basis pursuit". In: *IEEE Trans. Sig. Proc.* 60.4 (2012), pp. 1942–1956.

[38] J. F. Mota et al. "D-ADMM: A communication-efficient distributed algorithm for separable optimization". In: *IEEE Trans. Sig. Proc.* 61.10 (2013), pp. 2718–2723.

[39] C. Ravazzi, S.M. Fosson, and E. Magli. "Distributed Iterative Thresholding for ℓ_0/ℓ_1 -Regularized Linear Inverse Problems". In: *IEEE Trans. Inf. Theory* 61.4 (2015), pp. 2081–2100.

[40] S. Patterson, Y. C. Eldar, and I. Keidar. "Distributed sparse signal recovery for sensor networks". In: *IEEE Int. Conf. Acoust., Speech, Sig. Proc. (ICASSP)*. 2013, pp. 4494–4498.

[41] P. Han et al. "Distributed approximate message passing for sparse signal recovery". In: *IEEE Global Conf. Sig. Inf. Proc. (GlobalSIP)*. 2014, pp. 497–501.

[42] P. Han, R. Niu, and Y. C. Eldar. "Modified distributed iterative hard thresholding". In: *IEEE Int. Conf. Acoust., Speech, Sig. Proc. (ICASSP)*. 2015, pp. 3766–3770.

[43] S. Patterson, Y. C. Eldar, and I. Keidar. "Distributed compressed sensing for static and time-varying networks". In: *IEEE Trans. Sig. Proc.* 62.19 (2014), pp. 4931–4946.

[44] T. Blumensath and M. E. Davies. "Iterative thresholding for sparse approximations". In: *J. Fourier Anal. Appl.* 14.5-6 (2008), pp. 629–654.

[45] T. Blumensath and M. E. Davies. "Iterative hard thresholding for compressed sensing". In: *Appl. Comput. Harmon. Anal.* 27 (3 2008), pp. 265–274.

[46] I. Daubechies, M. Defrise, and C. D. Mol. "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint". In: *Comm. Pure Appl. Math.* 57.11 (2004), pp. 1413–1457.

[47] D. L. Donoho, A. Maleki, and A. Montanari. "Message passing algorithms for compressed sensing". In: *Proc. Natl. Acad. Sci.* Vol. 106. Madrid, Spain, 2009, pp. 18914–18919.

[48] R. Fagin, A. Lotem, and M. Naor. "Optimal aggregation algorithms for middleware". In: *Symp. Princ. Database Syst.* 2001, pp. 614–656.

[49] P. Cao and Z. Wang. "Efficient top-k query calculation in distributed networks". In: *Intl. Symp. Princ. Distrib. Comput. (PODC)*. 2004, pp. 206–215.

[50] S. S. Chen, D. L. Donoho, and M. A. Saunders. "Atomic decomposition by basis pursuit". In: *SIAM J. Sci. Comput.* 20.1 (1998), pp. 33–61.

[51] P. L. Combettes and V. R. Wajs. "Signal recovery by proximal forward-backward splitting". In: *Multiscale Model. Sim.* 4.4 (2005), pp. 1168–1200.

[52] S. Chen, S. A. Billings, and W. Luo. "Orthogonal least squares methods and their application to non-linear system identification". In: *Int. J. control* 50.5 (1989), pp. 1873–1896.

[53] Y. C. Pati, R. Rezaiifar, and P. Krishnaprasad. "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition". In: *Asilomar Conf. Sig., Syst., Comput.* IEEE. 1993, pp. 40–44.

[54] W. Dai and O. Milenkovic. "Subspace pursuit for compressive sensing signal reconstruction". In: *IEEE Trans. Inf. Theory* 55.5 (2009), pp. 2230–2249.

[55] A. Beck and M. Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". In: *SIAM J. Imag. Sci.* 2.1 (2009), pp. 183–202.

[56] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[57] D. L. Donoho, A. Maleki, and A. Montanari. "The Noise-Sensitivity Phase Transition in Compressed Sensing". In: *IEEE Trans. Inf. Theory* 57 (10 2011), pp. 6920–6941.

[58] M. Bayati and A. Montanari. "The Dynamics of Message Passing on Dense Graphs, with Applications to Compressed Sensing". In: *IEEE Trans. Inf. Theory* 57 (2 2011), pp. 764–785.

[59] J. Tan, Y. Ma, and D. Baron. "Compressive imaging via approximate message passing with wavelet-based image denoising". In: *IEEE Global Conf. Sig. Inf. Proc. (GlobalSIP)*. 2014, pp. 424–428.

[60] J. Tan, Y. Ma, and D. Baron. "Compressive imaging via approximate message passing with image denoising". In: *IEEE Trans. Sig. Proc.* 63.8 (2015), pp. 2085–2092.

[61] Y. Ma, J. Zhu, and D. Baron. "Compressed sensing via universal denoising and approximate message passing". In: *arXiv preprint arXiv:1407.1944* (2014).

[62] L. Anitori et al. "Design and Analysis of Compressed Sensing Radar Detectors". In: *IEEE Trans. Signal Proc.* 61 (4 2013), pp. 813–827.

[63] E. T. Hale, W. Yin, and Y. Zhang. "Fixed-point continuation for ℓ_1-minimization: Methodology and convergence". In: *SIAM Journal on Optimization* 19.3 (2008), pp. 1107–1130.

[64] A. Maleki and R. G. Baraniuk. "Least favorable compressed sensing problems for the first order methods". In: *IEEE Int. Symp. Inf. Theory*. 2011, pp. 134–138.

[65] A. Maleki. "Approximate message passing algorithms for compressed sensing". PhD thesis. Stanford University, 2010.

[66] B. Widrow and I. Kollár. *Quantization Noise: Roundoff Error in Digital Computation, Signal Processing, Control, and Communications*. Cambridge University Press, 2008.

[67] "IEEE Standard for Floating-Point Arithmetic". In: *IEEE Std 754-2008* (2008), pp. 1–70. DOI: 10.1109/IEEESTD.2008.4610935.

[68] J. T. Parker, P. Schniter, and V. Cevher. "Bilinear generalized approximate message passing". In: *arXiv preprint arXiv:1310.2632* (2013).

[69] A. Montanari and E. Richard. "Non-negative principal component analysis: Message passing algorithms and sharp asymptotics". In: *IEEE Transactions on Information Theory* 62.3 (2016), pp. 1458–1484.

[70] T. Tanaka. "A statistical-mechanics approach to large-system analysis of CDMA multiuser detectors". In: *IEEE Transactions on Information Theory* 48.11 (2002), pp. 2888–2910.

[71] T. Tanaka. "Statistical physics of information processing in mobile communications". In: *Progress of Theoretical Physics Supplement* 157 (2005), pp. 176–183.

[72] M. Bayati, M. Lelarge, A. Montanari, et al. "Universality in polytope phase transitions and message passing algorithms". In: *The Annals of Applied Probability* 25.2 (2015), pp. 753–822.

[73] M. Talagrand. "On the high temperature phase of the Sherrington-Kirkpatrick model". In: *Annals of probability* (2002), pp. 364–381.

[74] D. L. Donoho, A. Javanmard, and A. Montanari. "Information-theoretically optimal compressed sensing via spatial coupling and approximate message passing". In: *Information Theory, IEEE Transactions on* 59.11 (2013), pp. 7434–7464.

[75] A. Javanmard and A. Montanari. "State evolution for general approximate message passing algorithms, with applications to spatial coupling". In: *Information and Inference* (2013), iat004.

[76] P. Han et al. "Multi-processor approximate message passing using lossy compression". In: *IEEE Int. Conf. Acoust., Speech, Sig. Proc. (ICASSP)*. 2016.

[77] T. M. Cover and J. A. Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[78] T. C. Hu and R. L. Taylor. "On the strong law for arrays and for the bootstrap mean and variance". In: *International Journal of Mathematics and Mathematical Sciences* 20.2 (1997), pp. 375–382.

[79] C. Stein. "A bound for the error in the normal approximation to the distribution of a sum of dependent random variables". In: *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Probability Theory*. Berkeley, Calif.: University of California Press, 1972, pp. 583–602.

[80] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. "Estimation with Applications To Tracking and Navigation". In: (2001).

[81] S. A. Gershgorin. "Uber die abgrenzung der eigenwerte einer matrix". In: . 6 (1931), pp. 749–754.

[82] B. M. Brown et al. "Martingale central limit theorems". In: *The Annals of Mathematical Statistics* 42.1 (1971), pp. 59–66.

[83] V. V. Buldygin and Y. V. Kozachenko. "Sub-Gaussian random variables". In: *Ukrainian Mathematical Journal* 32.6 (1980), pp. 483–489.

[84] F. J. Massey Jr. "The Kolmogorov-Smirnov test for goodness of fit". In: *Journal of the American statistical Association* 46.253 (1951), pp. 68–78.

[85] Z. Drezner, O. Turel, and D. Zerom. "A modified Kolmogorov–Smirnov test for normality". In: *Communications in Statistics Simulation and Computation* 39.4 (2010), pp. 693–704.

[86] J. Rice. *Mathematical statistics and data analysis*. Cengage Learning, 2006.

[87] D. J. Murdoch, Y. Tsai, and J. Adcock. "P-values are random variables". In: *The American Statistician* (2012).

[88] R. Sibson M. C. Jones. "What is Projection Pursuit?" In: *Journal of the Royal Statistical Society. Series A (General)* 150.1 (1987), pp. 1–37. ISSN: 00359238.

[89] A. Gersho and R. M. Gray. *Vector quantization and signal compression*. Vol. 159. Springer Science & Business Media, 2012.

[90] R. E. Blahut. "Computation of channel capacity and rate-distortion functions". In: *IEEE Trans. Inf. Theory* 18.4 (1972), pp. 460–473.

[91] S. Arimoto. "An algorithm for computing the capacity of arbitrary discrete memoryless channels". In: *IEEE Trans. Inf. Theory* 18.1 (1972), pp. 14–20.

[92] S. Geman. "A limit theorem for the norm of random matrices". In: *The Annals of Probability* (1980), pp. 252–261.

[93] I. M. Johnstone. "On the distribution of the largest eigenvalue in principal components analysis". In: *The Annals of Statistics* 29.2 (Apr. 2001), pp. 295–327.

[94] C. A. Tracy and H. Widom. "Level-spacing distributions and the Airy kernel". In: *Comm. Math. Phys.* 159.1 (1994), pp. 151–174.

[95] D. Needell and J. A. Tropp. "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples". In: *Applied and Computational Harmonic Analysis* 26.3 (2009), pp. 301–321.

[96] Yonina C Eldar and Gitta Kutyniok. *Compressed sensing: theory and applications*. Cambridge University Press, 2012.

VITA

Puxiao Han was born on July 5, 1985, in Luoyang, Henan Province, China and is a Chinese citizen. He received his Bachelor of Science from School of Electronics Engineering and Computer Science, Peking University (Beijing Campus), China, in 2008, and subsequently received his Master of Science from School of Electronics Engineering and Computer Science, Peking University (Shenzhen Campus), China, in 2011. After graduation from Master Program, he worked one year as an engineer in Laboratory of Communication and Information Security, Peking University (Shenzhen Campus). His research interests are in statistical and sparse signal processing, and distributed computing in large-scale sparse signal recovery in sensor networks. He has seven publications, and two journal papers near ready for submission to peer-reviewed journals.