PhaseCode: Fast and Efficient Compressive Phase Retrieval based on Sparse-Graph Codes

Ramtin Pedarsani, Dong Yin, Kangwook Lee, and Kannan Ramchandran

Abstract

We consider the problem of recovering a complex signal $\boldsymbol{x} \in \mathbb{C}^n$ from m intensity measurements of the form $|\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{x}|$, $1 \leq i \leq m$, where $\boldsymbol{a}_i^{\mathrm{H}}$ is the i-th row of measurement matrix $\boldsymbol{A} \in \mathbb{C}^{m\times n}$. Our main focus is on the case where the measurement vectors are unconstrained, and where $\boldsymbol{x}$ is exactly K-sparse, or the so-called general compressive phase retrieval problem. We introduce ***PhaseCode***, a novel family of fast and efficient algorithms that are based on a sparse-graph coding framework. We show that in the noiseless case, the PhaseCode algorithm can recover an arbitrarily-close-to-one fraction of the K non-zero signal components using only slightly more than $4K$ measurements when the support of the signal is uniformly random, with order-optimal time and memory complexity of $\Theta(K)$[1]. It is known that the fundamental limit for the number of measurements in compressive phase retrieval problem is $4K - o(K)$ for the more difficult problem of recovering the signal exactly and with no assumptions on its support distribution [1], [2]. This shows that under mild relaxation of the conditions, our algorithm is the first constructive *capacity-approaching* compressive phase retrieval algorithm: in fact, our algorithm is also order-optimal in complexity and memory. Further, we show that for any signal $\boldsymbol{x}$, PhaseCode can recover a random $(1-p)$-fraction of the non-zero components of $\boldsymbol{x}$ with high probability, where p can be made arbitrarily close to zero, with sample complexity $m = c(p)K$, where $c(p)$ is a small constant depending on p that can be precisely calculated, with optimal time and memory complexity. As a result, assuming that the non-zero components of $\boldsymbol{x}$ are lower bounded by $\Theta(1)$ and upper bounded by $\Theta(K^\gamma)$ for some positive constant $\gamma < 1$, we are able to provide a strong ℓ_1 guarantee for the estimated signal $\hat{\boldsymbol{x}}$ as follows: $\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_1 \leq p\|\boldsymbol{x}\|_1(1+o(1))$, where p can be made arbitrarily close to zero. As one instance, the PhaseCode algorithm can provably recover, with high probability, a random $1-10^{-7}$ fraction of the significant signal components, using at most $m = 14K$ measurements.

Next, motivated by some important practical classes of optical systems, we consider a "Fourier-friendly" constrained measurement setting, and show that its performance matches that of the unconstrained setting, when the signal is sparse in the Fourier domain with uniform support. In the Fourier-friendly setting that we consider, the measurement matrix is constrained to be a cascade of Fourier matrices (corresponding to optical lenses) and diagonal matrices (corresponding to diffraction mask patterns).

Finally, we tackle the compressive phase retrieval problem in the presence of noise, where measurements are in the form of $y_i = |\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{x}|^2 + w_i$, and w_i is the additive noise to the ith measurement. We assume that the signal is quantized, and each non-zero component can take L_m possible magnitudes and L_p possible phases. We consider the regime where $K = \beta n^\delta$, $\delta \in (0,1)$. We use the same architecture of *PhaseCode* for the noiseless case, and robustify it using two schemes: the almost-linear scheme and the sublinear scheme. We prove that with high probability, the almost-linear scheme recovers $\boldsymbol{x}$ with sample complexity $\Theta(K\log(n))$ and computational complexity $\Theta(L_m L_p n\log(n))$, and the sublinear scheme recovers $\boldsymbol{x}$ with sample complexity $\Theta(K\log^3(n))$ and computational complexity $\Theta(L_m L_p K\log^3(n))$.

Throughout, we provide extensive simulation results that validate the practical power of our proposed algorithms for the sparse unconstrained and Fourier-friendly measurement settings, for noiseless and noisy scenarios.

I. Introduction

A. Phase Retrieval Problem

Compressive sensing (CS) has recently emerged as a powerful framework for understanding the fundamental limits for signal acquisition and recovery [3], [4]. The basic premise of CS is that a high-dimensional signal that is sparse in some basis can be recovered from linear projections of the signal with respect to an appropriate lower-dimensional measurement system. A key attribute of CS is that the measurement system is linear and phase-preserving. That is, the acquired samples, complex-valued in general, contain both the magnitude and phase of the measurements.

In many applications of interest, e.g. related to optics [5], X-ray crystallography [6], [7], astronomy [8], ptychography [9], quantum optics [10], etc., the phase information in the measured samples is not available. For example, in optical systems, one can measure only the intensity of the measurements as they relate to the photon count on a detector. Thus, the phase of the measurements is lost. Indeed, the problem of recovering a signal from only the magnitude of its Fourier transform has been a well-studied problem in the signal processing literature for several decades under the umbrella of phase retrieval [11]. It has recently received renewed interest in the "post-compressed-sensing" era [12]–[14], allowing for the insights from compressive sensing to be incorporated into the phase retrieval problem when the signal of interest is sparse, and the measurement matrix is unconstrained.

Ramtin Pedarsani is with the ECE Department at UC Santa Barbara. email: ramtin@ece.ucsb.edu
Dong Yin and Kannan Ramchandran are with the EECS Department at UC Berkeley. email:{dongyin,kannanr}@eecs.berkeley.edu
Kangwook Lee is with the EE Department at KASIT. email: kw1jjang@kaist.ac.kr

[1] Here, we define the notation $\mathcal{O}(\cdot)$, $\Theta(\cdot)$, and $\Omega(\cdot)$. We have $f = \mathcal{O}(g)$ if and only if there exists a constant $C_1 > 0$ such that $|f/g| < C_1$; $f = \Theta(g)$ if and only if there exist two constants $C_1, C_2 > 0$ such that $C_1 < |f/g| < C_2$; and $f = \Omega(g)$ if and only if there exists a constant $C_1 > 0$ such that $|f/g| > C_1$.

Concretely, consider a signal $\boldsymbol{x} \in \mathbb{C}^n$ and a measurement matrix $\boldsymbol{A} \in \mathbb{C}^{m \times n}$. The phase retrieval problem is to recover $\boldsymbol{x}$ from the observations $\boldsymbol{y} = |\boldsymbol{A}\boldsymbol{x}|, \boldsymbol{x} \in \mathbb{C}^n$, where the magnitude is taken on each element of the vector $\boldsymbol{A}\boldsymbol{x}$. The compressive phase retrieval problem targets the case where $\boldsymbol{x}$ is K-sparse.

In this paper, we study the phase retrieval problem under the following settings:

(i) General compressive phase retrieval of sparse signals[2]; and

(ii) "Fourier-friendly" compressive phase retrieval of signals having a sparse spectrum.

We now summarize these settings:

(i) **General compressive phase retrieval of sparse signals**: In this setting, we are free to design the measurement matrix A without any constraints, and this represents the primary contribution of this paper. We consider it for three reasons.
(1) It is of broadest theoretical interest, being the most general compressive phase retrieval problem, for which we propose a sparse-graph coding framework that is a significant departure from currently popular approaches based on convex optimization, Semi-Definite Programming (SDP), alternating minimization, gradient descent, etc. [14]–[20].
(2) It provides the intellectual insights and the foundational framework needed to address more constrained problems, such as those studied under the Fourier-friendly setting of category **(ii)**.
(3) It is of independent interest in applications related to certain quantum optical systems. For example, compressive sensing has been used in recent work involving quantum optics [10] to measure the transverse wavefunction of a photon, where the design of the measurement matrix has no constraints.

(ii) **Fourier-friendly compressive phase retrieval of signals having a sparse spectrum**: In this category, motivated by applications related to Fourier optical systems, the measurement matrix A is constrained to be Fourier-friendly (see Section VI for a detailed treatment). Concretely, $\boldsymbol{A}$ is constrained to be the cascade of (up to a couple of) stages of a diagonal matrix (corresponding to a so-called optical mask or coded diffraction pattern) and a Fourier transform (corresponding to an optical lens). This constraint is motivated by practical optical systems [21], array imaging [22], etc., as also addressed recently by [23].

B. Main Contributions

A key contribution of this work is in the introduction of modern coding theory techniques such as density evolution and sparse-graph codes [24] for the compressive phase retrieval problem. Exploiting these techniques and a similar measurement system to [18], [25] allows us to come up with the provably efficient and fast PhaseCode algorithm that is order-optimal in terms of number of measurements needed, time-complexity, and memory-complexity, which are all $\mathcal{O}(K)$. Furthermore, we provide precise constants for the number of measurements needed to achieve a targeted reliability. To the best of our knowledge, *this is the first work that provides precise constants for the number of measurements*. More specifically, the main contribution of this paper are the following:

(i) For an arbitrary signal $\boldsymbol{x}$, the PhaseCode algorithm can provably recover a random fraction of at least $1 - 10^{-7}$ of the active signal components with $14K$ measurements, with optimal time and memory complexity $\Theta(K)$. This is one instance of an entire family of trade-offs between the number of measurements needed and the fraction of non-zero signal components that can be recovered using PhaseCode. More precisely, we show that for any signal $\boldsymbol{x}$, PhaseCode can recover a random $(1-p)$-fraction of the non-zero components of $\boldsymbol{x}$ with high probability, for arbitrarily-close-to-zero constant p with sample complexity $m = c(p)K$, where $c(p)$ is a small constant depending on p that can be precisely calculated. As a result, assuming that the non-zero components of $\boldsymbol{x}$ are lower bounded by $\Theta(1)$ and upper bounded by $\Theta(K^\gamma)$ for some positive constant $\gamma < 1$, we are able to provide a strong ℓ_1 guarantee for the estimated signal $\hat{\boldsymbol{x}}$ as follows: $\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_1 \leq p\|\boldsymbol{x}\|_1(1 + o(1))$, where p can be made arbitrarily close to zero.

(ii) The PhaseCode algorithm can recover an arbitrarily-close-to-one fraction of the non-zero components of $\boldsymbol{x}$ using $4K(1+\epsilon)$ measurements for an arbitrarily small constant $\epsilon > 0$, when the support of the non-zero components of $\boldsymbol{x}$ is uniformly random, with optimal time and memory complexity of $\Theta(K)$. It is well-known that $4K - o(K)$ measurements is the fundamental limit for unique recovery of K-sparse signals [1], [2] for the more difficult problem of recovering the signal exactly with no assumptions on the support of the signal. This shows that under mild relaxation of the conditions, the PhaseCode algorithm is *capacity-approaching*.

(iii) Another key contribution of this work is to adapt the PhaseCode algorithm to a more constrained Fourier-friendly setting that is useful in certain optical systems, when $\boldsymbol{x}$ has a sparse spectrum. Specifically, we show how it is possible to elegantly integrate the Chinese-Remainder-Theorem-centric framework of Pawar and Ramchandran [26] (that was used to find a fast sparse Discrete-Fourier-Transform) into our PhaseCode framework without any loss of system performance in terms of measurement cost or computational complexity. See Section VI for details.

(iv) We demonstrate that PhaseCode can be robustified in the presence of noise. We use the same architecture of *PhaseCode* for the noiseless case, and robustify it using two schemes: the almost-linear scheme and the sublinear scheme. We

[2] This is easily extended, as is well known, to the case where the signal $\boldsymbol{x}$ is sparse w.r.t. some other basis, such as a wavelet, but in the interests of conceptual clarity, we will not consider such extensions in this work.

assume that the signal is quantized, and each non-zero component can take L_m possible magnitudes and L_p possible phases. We prove that with high probability, the almost-linear scheme recovers $\boldsymbol{x}$ with sample complexity $\Theta(K\log(n))$ and computational complexity $\Theta(L_mL_pn\log(n))$, and the sublinear scheme recovers $\boldsymbol{x}$ with sample complexity $\Theta(K\log^3(n))$ and computational complexity $\Theta(L_mL_pK\log^3(n))$.

We provide pseudocode of our algorithms (in Appendix O) and an extensive set of simulation results for all of the above settings that validate our theoretical findings, and verify the close match between theory and practice.

C. Related Work

The phase retrieval problem has been studied extensively over several decades. We do not attempt to provide a comprehensive literature review here; instead, we highlight here only some of the pertinent and diverse approaches to this problem that we are aware of. A large body of literature is dedicated to the phase retrieval problem for the case where the signal to be recovered has no structure and is not sparse. "Phaselift" proposed by Candes *et al.* [15] and "PhaseCut" proposed by Waldspurger *et al.* [27] are examples of convex optimization methods to solve the problem using semi-definite programming with $\Theta(n\log(n))$ measurements. While algorithms based on SDP provide theoretical performance guarantees and are robust to noise, they suffer from a high computational complexity of $\mathcal{O}(n^3)$ rendering them unsuited for many practical applications that require n to scale.[3] In [16], the authors propose an algorithm based on alternating minimization that reconstructs the signal with $\Theta(n\log(n)^3)$ measurements. In [20], the authors propose a non-convex algorithm based on Wirtinger flow that reconstructs the signal with measurement and computational complexity of $\Theta(n\log n)$.

In [2], [28]–[30], several sets of authors investigate the fundamental limits of phase retrieval problem, with the goal of finding necessary or sufficient conditions on the minimum number of measurements needed to guarantee that the solution is unique. In summary, $4n-4$ measurements are shown to be sufficient [30], and $4n-o(n)$ measurements are necessary [2] to reconstruct any signal perfectly.

We now review some relevant literature on compressive phase retrieval. To the best of our knowledge, the first algorithm for compressive phase retrieval was proposed by Moravec *et al.* in [12]. This approach requires knowledge of the ℓ_1 norm of the signal, making it impractical in most scenarios. The authors in [1] showed that $4K-1$ measurements are theoretically sufficient to reconstruct the signal, but did not propose any low-complexity algorithm. This number was later improved to $4K-2$ in [31], [32]. The PhaseLift method is also proposed for the sparse case in [14] and [17], requiring $\Theta(K^2\log(n))$ intensity measurements, and having a computational complexity of $\mathcal{O}(n^3)$, making the method less practical for large-scale applications. In [33], the authors propose an efficient algorithm based on polarization method that is able to stably reconstruct any K-sparse vector from $\Theta(K\log(n))$ noisy intensity measurements with complexity polynomial in n. The alternating minimization method in [16] can also be adapted to the sparse case with $\Theta(K^2\log(n))$ measurements and a complexity of $\mathcal{O}(K^3n\log(n))$. Compressive phase retrieval via generalized approximate message passing (PR-GAMP) is proposed in [13], with good performance in both runtime and noise robustness shown via simulations without theoretical justification.

A common attribute of all of the above-mentioned compressive phase retrieval references is that they assume that the measurement matrix can be designed freely. This renders them inapplicable to many application-constrained settings such as Fourier-optical systems. In [23], Candes *et al.* consider measurement matrices that are Fourier-friendly as described in the previous subsection, but only for the non-sparse case. They show that PhaseLift is able to recover the signal with $\Theta(n\log(n)^4)$ measurements by using $\Theta(\log(n)^4)$ masks or coded diffraction patterns. For the sparse case, Jaganathan *et al.* consider the phase retrieval problem from Fourier measurements only [18], [19]. They propose an SDP-based algorithm, and show that the signal can be provably recovered with $\Theta(K^2\log(n))$ Fourier measurements [18]. They also propose a combinatorial algorithm for the case where the measurement matrix can be designed without constraints, and show that the signal can be recovered with $\Theta(K\log(n))$ measurements and time complexity of $\mathcal{O}(Kn\log(n))$ [18].

In the prior literature that we are aware of, the works which overlap the most in spirit with ours are (i) the recently proposed SUPER algorithm for compressive phase retrieval by Cai *et al.* in [25]; and (ii) the FFAST algorithm of Pawar and Ramchandran [26] which also features the use of coding-theoretic tools for efficiently computing a sparse Discrete Fourier Transform. With regard to the FFAST algorithm [26], despite the common use of coding-theoretic tools, our problem formulation, analysis, and resulting algorithm are significantly different, mainly because our problem involves the loss of measurement phase, unlike that of FFAST.

With regard to the SUPER algorithm of [25], again, while there are some similarities between the two approaches – mainly to do with the use of certain system subcomponents such as a similar (but not identical) trigonometric-modulation method to resolve phase ambiguities, and the common use of a giant-component-cluster in the initial phase of our proposed PhaseCode algorithm (see Section V-B for details), our works are significantly distinct at many levels. First, the SUPER algorithm targets only the general unconstrained compressive phase retrieval setting, whereas, as described earlier, we also target Fourier-friendly constrained settings that are applicable in optical systems. Secondly, even in the unconstrained phase retrieval setting, there are significant distinctions between the two works with respect to theory, algorithm, and performance

[3] This limits the use of SDP-based methods to small to moderate values of n in practice. In contrast, we show simulations in the paper where n can be very large, even as large as 10^{10}. See Figures 6 and 8.

guarantees. As a quick overview, the SUPER algorithm uses $\Theta(K)$ measurements and features $\Theta(K\log(K))$ complexity with a zero-error-floor asymptotically. In contrast, by trading off the zero-error-floor for an arbitrarily-small controllable error-floor, our solution features key advantages. Specifically, this allows us to design a capacity-approaching measurement system that is based on a new and novel sparse-graph coding framework. The use of a sparse-graph coding framework in PhaseCode allows for iterative message-passing operations between the left nodes (signal components) of the sparse-graph code and the right nodes or measurements (see Section IV). This contrasts the more inefficient strictly "one-way" procedure in SUPER [25] wherein measurements of different stages are processed sequentially rather than iteratively. Moreover, PhaseCode has an optimal $\Theta(K)$ decoding complexity with optimal $\Theta(K)$ memory requirements. We also demonstrate how PhaseCode can be robustified in the presence of noise, unlike the work of [25]. We note that SUPER can also achieve $O(K)$ results with error floor. However, their approach is unable to characterize and optimize this error floor when the number of measurements is cK for a specific constant c. Finally, we note that peeling-based algorithms and expander graphs have been used for compressive sensing [34].

D. Paper Organization

The rest of the paper is organized as follows. In Section II, we define the general compressive phase retrieval problem. In Section III, we explain the main idea of PhaseCode algorithm. We present PhaseCode algorithm in detail in Section IV. The main theoretical results of the paper are provided in Section V. Via extensive simulations, we evaluate PhaseCode algorithms, validating the theorem. In Section VI, we demonstrate how our proposed measurements can be adapted to a Fourier-friendly setting. In Section VII, we show that PhaseCode can be robustified to noise. Finally, we conclude the paper in Section VIII.

II. Problem Formulation and Overview of the Main Result

Consider a complex signal $\boldsymbol{x} \in \mathbb{C}^n$ of length n which is exactly K-sparse; that is, only K out of n components of vector $\boldsymbol{x}$ are non-zero. Let $\boldsymbol{A} \in \mathbb{C}^{m\times n}$ be the measurement matrix that needs to be designed. The phase retrieval problem is to recover the signal $\boldsymbol{x}$ from magnitude measurements $y_i = |\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{x}|$, where $\boldsymbol{a}_i^{\mathrm{H}}$ is the i-th row of measurement matrix $\boldsymbol{A} \in \mathbb{C}^{m\times n}$. Figure 1 illustrates the block diagram of our problem.

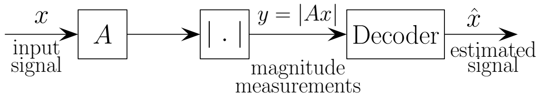


Fig. 1: Block diagram of general compressive phase retrieval problem. The measurements are $y_i = |\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{x}|$, where $\boldsymbol{a}_i^{\mathrm{H}}$ is the i-th row of measurement matrix $\boldsymbol{A}$. The objectives are to design measurement matrix $\boldsymbol{A}$ and the decoding algorithm to guarantee high reliability, while having small sample complexity as well as small time and memory complexity.

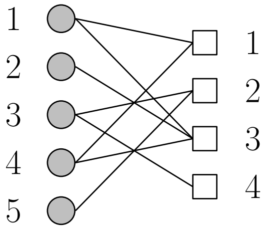
The main objectives of the general compressive phase retrieval problem is to design matrix $\boldsymbol{A}$, and the decoding algorithm to recover $\boldsymbol{x}$ such that

- The number of measurements m is as small as possible. Ideally, one wants m to be close to the fundamental limit of $4K - o(K)$ [1], [2].
- The decoding algorithm is fast with low computational complexity and memory requirements. Ideally, one wants the time complexity and the memory complexity of the algorithm to be $\mathcal{O}(K)$, which is optimal.
- The reliability of the recovery algorithm should be maximized. Ideally, one wants the probability of failure to be vanishing as the problem parameters K and m get large.

Remark In this work, we are interested in the asymptotic K regime. However, even when K is small, with proper modification of our algorithm, high reliability can be guaranteed when m gets large. It is worth mentioning that in this case, the number of

$$A = \begin{bmatrix} a_{11} & 0 & 0 & a_{14} & 0 \\ 0 & 0 & a_{23} & 0 & a_{25} \\ a_{31} & a_{32} & 0 & a_{34} & 0 \\ 0 & 0 & a_{43} & 0 & 0 \end{bmatrix}$$

(a) Measurement matrix A.



(b) Bipartite graph G.

Fig. 2: **Sparse graph codes.** The rows of A (the measurements) correspond to right nodes in the bipartite graph G, while the columns of A (the signal components) correspond to the left nodes of G.

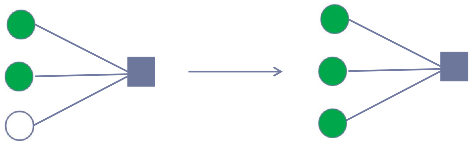


Fig. 3: **Coloring operation.** The figure illustrates when a right node is connected to exactly one uncolored active left node, and the other active left nodes connected to the right node are colored with the same color, then the uncolored active left node is colored with that color. In the graph, we have shown only the active left nodes.

measurements will be larger than the fundamental limit that is $4K(1+o(1))$. We do not discuss this any further in the interest of presentation clarity.

The main result of our paper is stated in the following (informal) theorem.

Theorem 1. *Consider a K-sparse signal $\boldsymbol{x} \in \mathbb{C}^n$, and the measurement matrix $\boldsymbol{A} \in \mathbb{C}^{m\times n}$ chosen by the PhaseCode algorithm.*

(i) *PhaseCode can recover a random $(1-p)$-fraction of the non-zero components of $\boldsymbol{x}$ with high probability, for arbitrarily-close-to-zero constant p. The measurement complexity of the algorithm is $m = c(p)K$, where $c(p)$ is a small constant depending on p that can be precisely calculated. The time and memory complexity of PhaseCode are also $\Theta(K)$. Further, for the estimated signal $\hat{\boldsymbol{x}}$, assuming that the non-zero components of $\boldsymbol{x}$ are lower bounded by $\Theta(1)$ and upper bounded by $\Theta(K^\gamma)$ for some positive constant $\gamma < 1$, we have*

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_1 \leq p\|\boldsymbol{x}\|_1 \left(1 + \Theta(\frac{1}{\log(K)})\right).$$

(ii) *Assuming that the support of $\boldsymbol{x}$ is distributed uniformly at random, with high probability, PhaseCode can recover an arbitrarily-close-to-one fraction of the non-zero components with $m = 4K(1+\epsilon)$ measurements for arbitrarily small constant $\epsilon > 0$.*

These results are more precisely stated in Theorems 2 and 3 in Section V. See Table II for some selected values of p and m.

III. Main Idea of the PhaseCode Algorithm

We now describe the main idea behind PhaseCode. As mentioned, the main novelty of our work is that we use sparse-graph codes, and the powerful tools of modern coding theory for design and analysis.

The design of an appropriate measurement matrix $\boldsymbol{A}$ for the compressive phase retrieval problem is equivalent to the design of an appropriate bipartite graph G, as for each measurement matrix, there exists a corresponding bipartite graph. Specifically, the rows of $\boldsymbol{A}$ (the measurements) are the right nodes in the bipartite graph G, while the columns of $\boldsymbol{A}$ (the signal components) are left nodes of G. We call the left nodes of G that correspond to an active (non-zero) signal component as active left nodes. Left node i is connected to right node j if a_{ji} is non-zero. The example shown in Figure 2 illustrates this connection.

As is well-known and also intuitive, in the phase-retrieval problem, the signal of interest can be recovered only to within an unknown global phase. The idea of our iterative reconstruction algorithm is to detect a non-zero signal component, give it global zero-phase, and align all other signal components with respect to it. This suggests the intuition of building up one or more clusters of non-zero components, where in our terminology, these clusters are identified by their colors; i.e. all the non-zero components belonging to a particular cluster have the same color. Two (or more) non-zero components (active left nodes) can be colored with the same color if their components are known in location, magnitude and phase relative to each other.

Our goal in designing the measurement matrix of the sparse graph is to create *iteratively decodable* right nodes (set of appropriately designed measurements). The key property of a right node that is conducive to our desired coloring operation is as follows. *If a right node is connected to one or more known components (colored active left nodes with the same color) and exactly one uncolored active left node (unresolved active signal component), then that component can be resolved, i.e. the uncolored active left node will be colored with the same color.* See Figure 3.

Our idea is to make this coloring "primitive operation" *iteratively trigger* more such coloring primitive operations in the system. Of course, the key is to design the graph efficiently to ensure that the domino-effect will continue till *all* the active left nodes are colored, while minimizing the number of right nodes needed to accomplish this (measurement cost).

This is the high-level connection between the compressive phase retrieval problem and sparse-graph code design. Our recovery process is conceptually similar to the "peeling" decoding of packets based on Low-Density-Parity-Check (LDPC) codes in packet-erasure communication systems, with the key distinction that *we cannot measure phase*. This makes our problem more challenging, therefore requiring a different analysis of the density evolution in the graph, as we will describe. But at a high level, our coloring primitive operation plays the analogous role of peeling in LDPC decoding.

Of course, a natural question is how our measurement system detects if a right node is indeed connected to one or more colored active left node and exactly one uncolored active left node. We can do so with a set of 4 cleverly designed "trigonometric" measurements that are part of each right node. We will explain the trigonometric measurements in detail in Section IV-A.

Notation	Description
$\boldsymbol{x}$	complex signal of length n
K	sparsity of the signal
n	length of the signal
m	number of measurements
M	number of the rows of the code matrix
$\boldsymbol{A}$	measurement matrix
$\boldsymbol{H}$	code matrix
$\boldsymbol{T}$	modulation matrix

TABLE I: Table of Notation.

IV. PhaseCode Algorithm

First we define $\boldsymbol{A} \in \mathbb{C}^{4M\times n}$ to be a "row tensor product"[4] of matrices $\boldsymbol{T}$ and $\boldsymbol{H}$, where $\boldsymbol{H} \in \{0,1\}^{M\times n}$ is a binary "code" matrix, to be shortly explained, and $\boldsymbol{T} \in 4\times n$ is the "trigonometric modulation" matrix that provides 4 measurements per each row of $\boldsymbol{H}$. We define a row tensor product of matrices $\boldsymbol{T}$ and $\boldsymbol{H}$, $\boldsymbol{T}\otimes\boldsymbol{H}$, as follows. Let $\boldsymbol{A} = \boldsymbol{T}\otimes\boldsymbol{H} = [\boldsymbol{A}_1^{\mathrm{H}}, \boldsymbol{A}_2^{\mathrm{H}}, \dots, \boldsymbol{A}_M^{\mathrm{H}}]^{\mathrm{H}}$ and $\boldsymbol{A}_i \in \mathbb{C}^{4\times n}$. Then, $A_i(jk) = T_{jk}H_{ik},\ 1 \le j \le 4,\ 1 \le k \le n$.

Example 1. Consider matrices

$$\boldsymbol{T} = \begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.4 & 0.5 & 0.6 \end{bmatrix} \text{ and } \boldsymbol{H} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Then, our measurement matrix $\boldsymbol{A}$ is designed from:

$$\boldsymbol{A} = \boldsymbol{T}\otimes\boldsymbol{H} = \begin{bmatrix} 0 & 0.2 & 0 \\ 0 & 0.5 & 0 \\ 0.1 & 0.2 & 0 \\ 0.4 & 0.5 & 0 \\ 0 & 0 & 0.3 \\ 0 & 0 & 0.6 \end{bmatrix}.$$

Matrix $\boldsymbol{H}$ is constructed using a carefully chosen random bipartite graph model with n left nodes and m right nodes. Each left node refers to a component of x, and each right node refers to a set of 4 measurements. There are K active left nodes corresponding to the K non-zero components of x. The bipartite graph is constructed as follows. $H_{ij} = 1$ if and only if left node j is connected to right node i, and $H_{ij} = 0$ otherwise.

While we provide the details of how to design matrix $\boldsymbol{T}$ in Section IV-A, for completeness of the description, we state it precisely here deferring explanation to Section IV-A. Let ω' be a uniformly random phase between 0 and 2π. We design $\boldsymbol{T} \in \mathbb{C}^{4\times n}$ to be

$$T = \begin{pmatrix} e^{\mathbf{i}\omega} & e^{\mathbf{i}2\omega} & \dots & e^{\mathbf{i}n\omega} \\ e^{-\mathbf{i}\omega} & e^{-\mathbf{i}2\omega} & \dots & e^{-\mathbf{i}n\omega} \\ \cos(\omega) & \cos(2\omega) & \dots & \cos(n\omega) \\ e^{\mathbf{i}\omega'} & e^{\mathbf{i}2\omega'} & \dots & e^{\mathbf{i}n\omega'} \end{pmatrix}. \tag{1}$$

As in [26], in the bipartite graph model, we use the following terminology extensively throughout the paper:

- *Singleton:* A right node is a singleton if it is connected to exactly one *active* left node.
- *Doubleton:* A right node is a doubleton if it is connected to exactly two active left nodes.
- *Multiton:* A right node is a multiton if it is connected to more than one active left node.[5]

We now describe PhaseCode algorithm, and analyze it in Section V. With the aid of the carefully designed matrix $\boldsymbol{T}$, our decoder is capable of performing the following functions:

- When an active left node is connected to a singleton right node, the active left node can be colored with a new color. That is, the non-zero component can be found in magnitude and location. However, the relative phase of the component with respect to other resolved components cannot be recovered. Figure 4 illustrates this operation.
  Note that in our terminology, each color refers to a local coordinate with a local phase, for example, the red coordinate, blue coordinate, etc. Then, the relative phase of two non-zero components that are colored as red is known. However, the relative phase of a blue component and a red component is not known.
- When a right node is connected to exactly one *uncolored* active left node, and the other non-empty set of active left nodes connted to the right node have all the same color (let's say green), then the uncolored active left node is colored with that color (i.e. it becomes green). Figure 3 illustrates this operation.

[4] Here, we apologize for not following popular convention for the notation for tensor product of matrices; instead, we define our own notation that is convenient for our purpose, which should hopefully not cause any confusion.

[5] In our terminology, a doubleton is also a multiton.

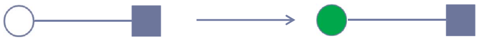


Fig. 4: **Singleton coloring operation.** The figure illustrates when a right node is a singleton, the corresponding active left node gets colored with a new color. In the graph, we have not showed the left nodes corresponding to 0 signal components.

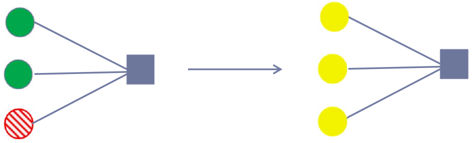


Fig. 5: **Combining colors.** The figure illustrates when a right node is connected to only colored active left nodes with two colors, then the colors can be combined.

- When *all* the active left nodes connected to a right node are colored, with exactly two colors, then those two colors can be combined into a single composite color. Figure 5 illustrates this operation.[6]

PhaseCode Algorithm In the first iteration of the algorithm, all the left nodes connected to singletons are colored. In the second iteration, all the doubletons that are connected to two colored active left nodes from the first iteration (strong doubletons), are detected, and their colors are combined. Then, the *largest* set of active left nodes having the same color[7] is selected, and *every other colored active left node gets uncolored*. At this point, there is only *one* color and no new colors are added to the system. In the following iterations, if a right node is connected to exactly one uncolored active left node and at least one colored active left node, then that uncolored active left node gets colored. (See Figure 3.) The algorithm continues until no more active left nodes can be colored.

We provide the pseudocode of the algorithm in Appendix O.

Remark PhaseCode has $\Theta(K)$ time and memory complexity.

Example 2. Let $K = 4$, $M = 5$ and $d = 2$. Without loss of generality, label the active left nodes by 1 to 4. Suppose that the bipartite graph is such that the right nodes are connected to $\{1\}$, $\{1, 2\}$, $\{3\}$, $\{1, 3\}$, and $\{2, 3, 4\}$. In the first iteration, 1 and 3 are colored, let us say by red and blue, respectively since these active left nodes are connected to singletons. In the second iteration, PhaseCode finds a strong doubleton, $\{1, 3\}$, that is connected to colored left nodes 1 and 3. Thus, their colors are combined to a composite color, let us say green, which will be the only color of the system after this iteration. In the third iteration, left node 2 is colored through the right node $\{1, 2\}$, since 2 is the only uncolored left node connected to this right node. Finally, in the forth iteration, left node 4 is colored through right node $\{2, 3, 4\}$. This completes the successful decoding of PhaseCode algorithm.

A. *Measurement Design: "Trig-Modulation"*

In this section, we will explain the choice of the measurement matrix $\boldsymbol{T}$. Our design of $\boldsymbol{T}$ draws heavily from the proposed trigonometric subsystem in [25] with proper modifications to better match our sparse-graph code subsystem, $\boldsymbol{H}$, that is distinct from [25]. We also show that one can decrease the number of these trig-based measurements from 5 per right node as proposed in [25] to 4 per right node as we describe that is crucial in designing a capacity-approaching scheme.

Define the length-4 vector $\boldsymbol{y}_i$ to be the measurement vector corresponding to the i-th row of matrix $\boldsymbol{H}$ for $1 \le i \le M$. Then $\boldsymbol{y} = [\boldsymbol{y}_1^T, \boldsymbol{y}_2^T, \dots, \boldsymbol{y}_M^T]^T$, where $\boldsymbol{y}_i = [y_{i,1}, y_{i,2}, y_{i,3}, y_{i,4}]^T$. Let $\omega = \frac{\pi}{2n}$. We design the measurement matrix $\boldsymbol{T} = [t_{j\ell}]$ as follows. For all $\ell,\ 1 \le \ell \le n$,

$$t_{1\ell} = e^{\mathbf{i}\omega\ell}, \tag{2}$$

$$t_{2\ell} = e^{-\mathbf{i}\omega\ell}, \tag{3}$$

$$t_{3\ell} = 2\cos(\omega\ell), \tag{4}$$

$$t_{4\ell} = e^{\mathbf{i}\omega'\ell}, \tag{5}$$

where as mentioned in Section IV, ω' is a random phase uniformly distributed between 0 and 2π.

As mentioned in Section IV, the measurement matrix should enable us to do the following operations: (1) Detect whether we have a singleton right node, and if yes, what the location index and magnitude of the corresponding active left node are (See Figure 4); (2) detect if a multiton right node is connected to colored active left nodes having exactly two unique colors, and if yes, what the relative phase of the colored components is. We call these as mergeable multitons (See Figure 5); (3)

[6] We use this operation only in the second iteration of PhaseCode.

[7] Whenever two active left nodes having colors C_1 and C_2 are combined, they get the same composite color C_{12}, and all other active left nodes with colors C_1 and C_2 are also recolored to C_{12}.

detect if a multiton right node is connected to colored active left nodes with the same color and only one uncolored active left node, the measurement system should be able to find the index, magnitude, and relative phase of the uncolored active left node. We call these right nodes resolvable multitons as in [25] (See Figure 3). In the following, we show how each of these detections can be accomplished using "guess and check" approach. We provide pseudocode of these detection procedures in Appendix O.

(i) **Singletons**: Suppose that we want to check the hypothesis that the i-th right node is a singleton. If the right node is a singleton, only one non-zero component of $\boldsymbol{x}$, let's say x_ℓ, is present in vector y_i, that is $y_{i,1} = |x_\ell e^{\mathbf{i}\omega\ell}|$, $y_{i,2} = |x_\ell e^{-\mathbf{i}\omega\ell}|$, and so on. Thus, the i-th right node is a singleton only if $y_{i,1} = y_{i,2} = y_{i,4}$. The event that i is not a singleton, and all these measurements are equal has measure 0 since ω' is a uniformly random phase.[8] In order to find the index ℓ, one uses $y_{i,3}$ to get

$$\ell = \frac{1}{\omega}\cos^{-1}\left(\cos(\omega\ell)\right) = \frac{1}{\omega}\cos^{-1}\left(\frac{y_{i,3}}{2y_{i,1}}\right).$$

Note that $\cos(\omega\ell)$ is positive if $0 \le \omega \le \frac{\pi}{2n}$ for all $\ell,\ 1 \le \ell \le n$.

(ii) **Mergeable multitons**: Consider a right node i as in Figure 5, which is already known to be connected to some (say, red) active left nodes (non-empty set $\mathcal{R}$) and some (say, blue) active left nodes (non-empty set $\mathcal{B}$). This means that the red (or blue) signal components are known in location, magnitude, and phase relative to each other. However, the relative phase of blue and red components' coordinate systems is not known. If there is no other active left node connected to i, we show that the relative phase can be found. Thus, the colors can be combined. (We again deploy a guess and check strategy.) First, we guess that right node i is connected to no other active left nodes. Then, we have access to measurement

$$y_{i,1} = |r + b|,$$

where $r = \sum_{\ell\in\mathcal{R}} x_j e^{\mathbf{i}\omega\ell}$ is the sum of complex numbers corresponding to the red components, and $b = \sum_{\ell\in\mathcal{B}} x_\ell e^{\mathbf{i}\omega\ell}$ is the sum of complex numbers corresponding to the blue components. Since red components are known up to a local phase, $|r|$ is known. Similarly, $|b|$ is also known. Without loss of generality, pick some $\ell_r \in \mathcal{R}$ and set the phase of x_{ℓ_r} to 0 to form the local coordinate for red components. Furthermore, pick some $\ell_b \in \mathcal{B}$ and set the phase of x_{ℓ_b} to 0 to form the local coordinate for blue components. Given the local coordinates, $r = |r|e^{\mathbf{i}\phi_r}$ and $b = |b|e^{\mathbf{i}\phi_b}$ are known. By the cosine law, the true relative phase between r and b can be found as

$$\theta = \cos^{-1}\left(\frac{|r|^2 + |b|^2 - y_{i,1}^2}{2|r||b|}\right), \tag{6}$$

up to a plus-minus sign. Assuming that the plus sign is true, we can merge these components as follows. Without loss of generality, we set the phase of x_{ℓ_r} to 0. Thus, $r = |r|e^{\mathbf{i}\phi_r}$ and $b = |b|e^{\mathbf{i}(\phi_r+\theta)}$. This shows that the local coordinate in $\mathcal{B}$ should be rotated by an angle $\theta + \phi_r - \phi_b$ to match with the new coordinate. Hence, we recover all the blue components with respect to the coordinate of red components, and the colors can be combined. A similar procedure can be done for the solution of θ with a minus sign. Now we again use the check equation to find whether one of these relative phases passes the check equation. If none of them passes, our guess is wrong, and right node i is not a mergeable multiton. Thus, we need to check whether

$$|\sum_{\ell\in\mathcal{R}\cup\mathcal{B}} x_\ell e^{\mathbf{i}\omega'\ell}| = y_{i,4}$$

is satisfied or not for the 2 values of θ derived in (6). If the guess is correct, the probability that the check fails is 0 since ω' is random. Moreover, if the guess is not correct, the probability that the check passes is 0.

(iii) **Resolvable multitons**: Consider a right node i, for which we know that it is connected to some known active left nodes that have the same color. We want to check if i is connected to exactly one other active left node; i.e. one unknown non-zero component of x, say x_ℓ, as in Figure 3. We now describe our guess and check strategy to check if right node i is indeed a resolvable multiton, and if so, to find ℓ and x_ℓ. If our guess is correct, we have access to measurements of the form:

$$y_{i,1} = |a + e^{\mathbf{i}\omega\ell}x_\ell| = |u|, \tag{7}$$

$$y_{i,2} = |b + e^{-\mathbf{i}\omega\ell}x_\ell| = |v|, \tag{8}$$

$$y_{i,3} = |c + 2\cos(\omega\ell)x_\ell| = |w|, \tag{9}$$

$$y_{i,4} = |d + e^{\mathbf{i}\omega'\ell}x_\ell|, \tag{10}$$

[8] In practice, every measurement system has a finite precision level. Moreover, practical systems suffer from the presence of noise. The measurement system introduced here is clearly not robust to noise and finite precision of the measurement matrix, but we will show in Section VII that PhaseCode can be robustified to noise while maintaining its iterative decoding architecture.

d	5	6	7	8	9	10
$m(p)$	$12.44K$	$12.72K$	$13.28K$	$\mathbf{13.92K}$	$14.64K$	$15.4K$
p	1.1×10^{-3}	8×10^{-5}	3.2×10^{-6}	$\mathbf{1 \times 10^{-7}}$	2.9×10^{-9}	7×10^{-11}

TABLE II: Family of trade-offs between error floor and number of measurements for Phasecode. The table shows that to achieve higher reliability, i.e. smaller error floor, the number of measurements m should be increased.

where complex numbers a, b, c and d are known values that depend on the values and locations of the known colored active left nodes. For the purpose of readability, we show the calculations of how to solve the system of equations (7)-(10) in Appendix A.

V. Main Result

In this section, we analyze the performance of PhaseCode and provide the main theoretical results of this paper.

A. Bipartite Graph Construction

As mentioned earlier, we design our code matrix based on a random bipartite graph model. Given a bipartite graph with n left nodes and M right nodes, define the pruned bipartite graph corresponding to $\boldsymbol{x}$ to be a bipartite graph with K left nodes corresponding to the non-zero components of $\boldsymbol{x}$ and M right nodes, such that all the left nodes corresponding to the zero components and their connected edges are deleted. From now on, we consider the pruned graph for analysis. Moreover, from now on, by a left node (of the pruned graph), we refer to an active left node.

We first define the left and right edge degree distribution of the random bipartite graph. Define ρ_i to be the probability that a randomly selected edge is connected to a right node of degree i, and λ_i to be the probability that a randomly selected edge is connected to a left node of degree i. Define the edge degree distributions or edge degree polynomials of right and left nodes as follows.

$$\rho(x) = \sum_{i \geq 1} \rho_i x^{i-1};$$
$$\lambda(x) = \sum_{i \geq 1} \lambda_i x^{i-1}.$$

We construct two random bipartite graph models as follows:

(i) Regular left degree: In this construction, each left node is connected to d right nodes randomly, where d is a constant to be chosen. Thus, the degree of all left nodes are d. More formally, let $\mathcal{C}^K(d, M)$ be the ensemble of regular left degree bipartite graphs with K left nodes, M right nodes, and left degree d. We pick a bipartite graph uniformly at random from this ensemble. When M and K get large, the degree of a random right node is Poisson distributed with parameter $\eta = \frac{Kd}{M}$. Note that the degree of a right node in the pruned graph is the number of active left nodes connected to it. Since ρ_i is the fraction of edges that are connected to a right node of degree i, we have

$$\begin{aligned} \rho_i &= \frac{iM}{Kd} \mathbb{P}(\text{random right node has degree } i) \\ &= \frac{i}{\eta} \frac{\eta^i e^{-\eta}}{i!} \\ &= \frac{\eta^{i-1} e^{-\eta}}{(i-1)!}. \end{aligned}$$

Then, the left edge and right edge degree distributions are

$$\lambda(x) = x^{d-1} \tag{11}$$
$$\rho(x) = e^{-\eta(1-x)}. \tag{12}$$

(ii) Irregular left degree: In this construction, we design the left degree distribution $\lambda(x)$ based on a truncated harmonic distribution as follows. Let $h(x) = \sum_{i=1}^{x} 1/i$. Then,

$$\lambda_i = \frac{1}{i-1} \times \frac{1}{h(D-1)}, \; 2 \leq i \leq D, \tag{13}$$

where D is a (large) constant to be determined. The harmonic distribution for irregular LDPC codes is well-known to be capacity-achieving for BEC channels [35].

The main theoretical results of this paper for the noiseless case are as follows.

Theorem 2. *Let $\boldsymbol{A} = \boldsymbol{T} \otimes \boldsymbol{H}$ be the measurement matrix, where $\boldsymbol{H}$ is chosen uniformly at random from the ensemble $\mathcal{C}^n(d, M)$ and $\boldsymbol{T}$ is the modulation matrix defined in* (1)*. Using the m measurements $\boldsymbol{y} = |\boldsymbol{A}\boldsymbol{x}|$, for any $p > 0$, Regular PhaseCode*

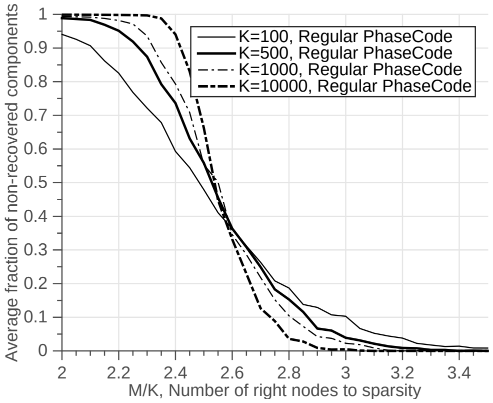


Fig. 6: **Performance of regular PhaseCode Algorithm.** We evaluate Regular PhaseCode algorithm via simulations. We chose the 3rd column of the table as an operating point, i.e., $(d, m, p^*(m)) = (7, 13.28K, 3.2 \times 10^{-6})$. PhaseCode algorithm successfully recovers almost all active signal components with high probability when $m = 4 \times 3.32K = 13.28K$.

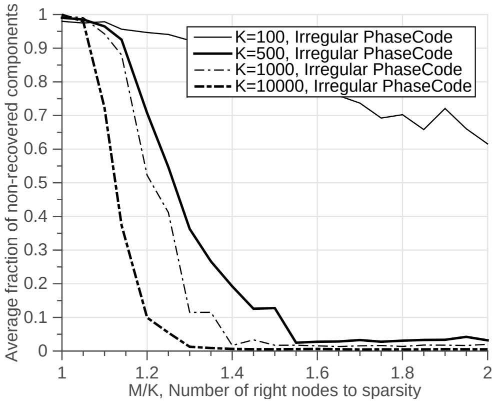


Fig. 7: **Performance of PhaseCode Algorithm.** We also evaluate Irregular PhaseCode, which demonstrates that it is capacity-approaching. We observe that for $K = 10000$ irregular PhaseCode can recover almost all the non-zero signal components with $m = 4 \times 1.3K$ measurements.

can recover at least a $1 - p$ fraction of the non-zero components of x chosen uniformly at random, where $m = c(p)K$ and tabulated in Table II for selected values. As a particular operating point, Regular PhaseCode is able to recover a random fraction $1 - 10^{-7}$ of non-zero components of $\boldsymbol{x}$ with $14K$ measurements with probability $1 - \mathcal{O}(1/m)$. Furthermore, the decoding complexity of the algorithm is $\Theta(K)$ which is order-optimal.

Theorem 3. *Let $\boldsymbol{A} = \boldsymbol{T} \otimes \boldsymbol{H}$ be the measurement matrix, where $\boldsymbol{H}$ is chosen according to the irregular construction in* (13)*, and $\boldsymbol{T}$ is the modulation matrix defined in* (1)*. Under the assumption that the support of the sparse signal is uniformly random, using $m = 4K(1 + \epsilon)$ measurements $\boldsymbol{y} = |\boldsymbol{Ax}|$ for arbitrarily small $\epsilon > 0$, Irregular PhaseCode is able to recover all but an arbitrarily small random fraction of the non-zero components of $\boldsymbol{x}$ with probability $1 - \mathcal{O}(1/m)$. Furthermore, the decoding complexity of the algorithm is $\Theta(K)$ which is order-optimal.*

We provide the proofs in Sections V-B and V-C.

Corollary 4. *Suppose that for a particular choice of parameters, PhaseCode has error floor p. For any signal $\boldsymbol{x} \in \mathbb{C}^n$, assuming that the non-zero components of $\boldsymbol{x}$ are lower bounded by $\Theta(1)$ and upper bounded by $\Theta(K^{\gamma})$ for some positive constant $\gamma < 1$, we have*

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_1 \leq p\|\boldsymbol{x}\|_1(1 + \Theta(\frac{1}{\log(K)})),$$

with probability $1 - \mathcal{O}(K^{\frac{1+\gamma}{2}} e^{-\frac{2K^{(1-\gamma)/2}}{\log^2(K)}})$ over the randomized choice of $\boldsymbol{A}$.

We provide the proof of Corollary 4 in Appendix B.

Before presenting the proof of the main theorems, we illustrate the performance of regular and irregular PhaseCode via simulations in Figures 6 and 7. Theorem 2 guarantees that regular PhaseCode recovers a fraction $p^*(m)$ of $\boldsymbol{x}$ with m measurements with high probability, where $(d, m, p^*(m))$ can be chosen from Table II. We choose the 3-rd column of the table as an operating point, i.e., $(d, m, p^*(m)) = (7, 13.28K, 3.2 \times 10^{-6})$ for regular PhaseCode. We define the error probability

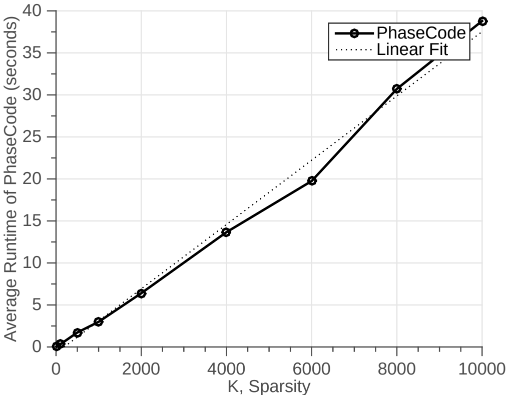


Fig. 8: **Time Complexity of PhaseCode.** We measure run-time of PhaseCode algorithm. We choose $n = 10^{10}$ and vary K.

Notation	Description
p_j	average fraction of non-recovered significant components at iteration j
η	average degree of right nodes
d	degree of left nodes in d-regular construction
D	truncation level for the harmonic distribution
$\lambda(x)$	left edge degree polynomial
$\rho(x)$	right edge degree polynomial
Z	number of uncolored edges after ℓ iterations

TABLE III: Table of Notation for Sections V-B and V-C.

to be the fraction of non-zero components of $\boldsymbol{x}$ that are not recovered. We measure the error probability while m is varied between $8K$ and $14K$ by averaging over 1000 simulation runs. We repeat the same procedure for several values of K. As expected, the PhaseCode algorithm successfully recovers essentially all the signal components when $m = 13.28K$. We also show simulation results for irregular PhaseCode in Fig. 7 that support Theorem 3. For example, when $K = 10000$, the coloring algorithm successfully recovers the signal with only $4 \times 1.3K = 5.2K$ measurements. From the simulations, it is clear that to operate close to capacity, one needs large asymptotics for K.

Theorems 2 and 3 also state that the decoding complexity of PhaseCode is $\Theta(K)$, which is order-optimal. In addition to that, its memory complexity is $\Theta(K)$, which is also order-optimal. In order to corroborate the claims, we measure the running time of the PhaseCode Algorithm. We choose the same operating point for regular PhaseCode as in the above simulations. We randomly generate signals of length $n = 10^{10}$, and increase the sparsity K up to 10^4 to see how the average runtime scales. The results are plotted in Figure 8; as K increases, the measured decoding time linearly increases. Indeed, PhaseCode successfully recovers $K = 10^4$ non-zero components in less then 40 seconds. The exact runtime can be further improved considering that the simulator is written in Python and is not fully optimized, and that the simulation is done on a normal laptop.[9]

B. Proof of Theorem 2

We first provide a brief outline of the proof elements, highlighting the main technical components needed to show that PhaseCode recovers an arbitrarily-close-to-one fraction of non-zero signal components with high probability.

- *Density evolution:* We analyze the performance of PhaseCode on a typical random bipartite graph (regular or irregular), for a fixed number of iterations, ℓ. First, we assume that a local neighborhood of depth 2ℓ of every edge in the graph is tree-like, i.e., cycle-free. Under this assumption, all the messages between right and left nodes, in the first j iterations of the algorithm, are independent. Using this independence assumption, we derive a recursive equation that represents the evolution of the expected number of unresolved components at each iteration.
- *Convergence to the cycle-free case:* : Using a Doob martingale argument as in [36], we show that the 2ℓ neighborhood of most of the edges of a randomly chosen graph from the ensemble is cycle-free with high probability. This proves that PhaseCode decodes all but a small fraction of the left nodes with high probability in a constant number of iterations. The main difference of our convergence analysis compared to [36] is that the right edge degree distribution in our graphs is Poisson distributed, while the right degree is regular in [36].

At each iteration of PhaseCode, we call the giant component as the largest set of signal components (left nodes) that have been resolved relative to each other. The algorithm follows 3 major steps to recover the active left nodes by coloring them.

[9] For the measurements, we used a laptop with 2GHz Intel Core i7 and 8GB memory.

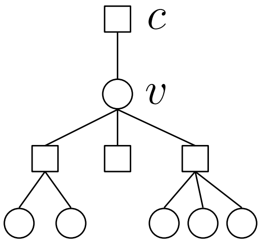


Fig. 9: Length-2 tree-like neighborhood of (v, c) for $d = 4$. The neighborhood is the subgraph of all the edges and nodes of paths having length less than or equal to 2, that start from v and the first edge of the path is not (v, c).

- *Step* 1*:* All the singleton right nodes and their corresponding left nodes are detected.
- *Step* 2*:* Strong doubletons are detected, and the color of the corresponding 2 left nodes get merged. We call the largest set of left nodes that *chain* hands together through these strong doubletons to be the giant component at this step.
- *Step* 3*:* After the initial giant component is formed, at each iteration of the algorithm, left nodes are colored one at a time through resolvable multitons, and become part of the giant component.

Now we analyze the message passing algorithm. A left node v passes a 0 message to neighbor right node c if it is not colored (i.e. it is not part of the giant component). Let p_j be the probability that a random message sent from a left node to a right node is 0, at iteration j of the algorithm. The density evolution equation is an equation relating p_j to p_{j+1}. Similarly, a right node c passes a message 0 to neighbor left node v if it can not get colored (become part of the giant component). Let q_j be the probability that a random message sent from a right node to a left node is 0, at iteration j of the algorithm. Under the tree-like assumption, and for $j \geq 2$ one has

$$p_{j+1} = (1 + e^{-\eta} - e^{-\eta p_j})^{d-1}. \tag{14}$$

Here is a proof of Equation (14). A left node v passes a 0 message to right node c at step $j+1$, if all of the other $d-1$ neighbor right nodes of v pass message 0 to v at step j. That is $p_{j+1} = q_j^{d-1}$. Note that for $j \geq 2$, if a right node is a singleton, it passes message 0 to neighbor left nodes, since in PhaseCode only resolvable multitons can color active left nodes after the second step of the algorithm.

We calculate q_j as follows. A right node c sends a message to a left node v that it is part of the giant component if c is connected to a non-empty set of left nodes other than v, and those left nodes are all in the giant component. Thus,

$$\begin{aligned} 1 - q_j = \sum_{i=2}^{\infty} \rho_i (1 - p_j)^{i-1} &= \rho(1 - p_j) - \rho_1 \\ &= e^{-\eta p_j} - e^{-\eta}. \end{aligned}$$

This proves (14). See Figure 9 for an illustration of the proof for the case $d = 4$.

Remark Note that if a right node is a singleton, it cannot recover the corresponding active left node in both phase and magnitude. This is a fundamental difference of our decoding process compared to that of conventional peeling-based decoders such as the LDPC decoder for erasure channel [24]. In LDPC decoding, since there is no phase ambiguity, as soon as a singleton is detected, the corresponding non-zero component is recovered and it is peeled from all other right nodes that are connected to that component. However, active left nodes in singletons cannot be peeled in our setting. Indeed, our problem has the peculiar attribute that singleton right nodes, while critical to initiating the growth of the giant component at the outset, are not useful once a giant component is formed, and too many singletons actually hurt the system performance by featuring useless isolated measurements. This is a significant departure from "phase-aware" measurement systems like LDPC codes. This is also the key reason to why our density evolution equation in (14) differs from that of linear measurement systems [24], [26], which is $p_{j+1} = (1 - e^{-\eta p_j})^{d-1}$.

An interesting but unfortunate fact is that $p_0 = 1$ is a fixed point of the density evolution equation. Thus, one cannot use (14) at the outset to follow the evolution of p_j, and to argue that it goes close to 0, since p_j can get stuck at 1. To use Equation (14), we need a more careful characterization of the first two steps of the algorithm that form the giant component. At the first iteration, all the (active) left nodes that are connected to at least one singleton right node are found. Since the relative phase of these signal components is not known, no giant component is formed yet; thus, $p_1 = 1$. At the second iteration, the giant component is formed by merging the colors of left nodes in strong doubletons. Recall that a strong doubleton right node is a right node that is connected to two colored left nodes. After the giant component is formed in the second iteration, the probability that a randomly chosen left node is not part of the giant component is p_2. If one can show that p_2 is small enough such that after a fixed number of iterations p_j gets close to 0, then concentration bounds can be used to show that the number

d	4	**5**	6	7	**8**	9	10
p^*	2.7×10^{-2}	1.1×10^{-3}	8×10^{-5}	3.2×10^{-6}	1×10^{-7}	2.9×10^{-9}	7×10^{-11}
c	3.31	**3.11**	3.18	3.32	3.48	3.66	3.85

TABLE IV: The table shows how the error floor, p^*, and $c = M/K$ (which indirectly determines the number of measurements) vary for different values of left degree, d. The minimum value of c is 3.11 that is achieved when $d = 5$. Moreover, one can see that p^* decreases as d increases.

of left nodes not being in the giant component is indeed highly concentrated around its mean after ℓ iterations, that is Kp_ℓ. In Lemma 7, we show that if $p_2 = 1 - \delta$ for some arbitrary constant $0 < \delta < 1$ independent of K, p_j gets close to 0 after a constant number of iterations. Clearly $p_2 = 1 - \delta$ if there exists a giant component of size linear in K after the second step. In Lemma 5, we find the condition for left-regular bipartite graph under which a linear size giant component will be formed after the second step of the algorithm.

Lemma 5. *There exist operating points $(d, M = cK)$ for which with probability $1 - \mathcal{O}(1/M)$, a giant component of size linear in K is formed after the second step of PhaseCode. In particular, $(d = 5, 3.11 \leq c \leq 19.24)$ and $(d = 8, 3.48 \leq c \leq 55.36)$ are two of these operation points.*

See Appendix C for the proof.

Remark Lemma 5 shows that for large enough m, a positive fraction of the signal components can get recovered after the second iteration of the algorithm. Thus, PhaseCode gets a proper *jump-start*, which is essential for proving that the algorithm terminates after a constant number of iterations, and successfully recovers an arbitrarily-close-to-one fraction of the signal components.

Remark As one observes in Lemma 5, if M is larger than some threshold (which corresponds to more measurements), the giant component will not get formed. At a first glance, this sounds counter-intuitive since having more right nodes seems to only help. However, one should keep in mind that in the statement of the lemma, the left degree d is kept fixed. Intuitively, when M is too large, for a fixed small d, the bipartite graph (with active left nodes) becomes so sparse that there are too few doubletons to form a giant component.

Corollary 6. *There exists a constant $0 < \delta < 1$ independent of K, such that $p_2 = 1 - \delta$.*

Due to the formation of a linear-size giant component in step 2 of the algorithm, we can revisit the density evolution equation (14):

$$p_{j+1} = (1 + e^{-\eta} - e^{-\eta p_j})^{d-1},$$

with the aid of Corollary 6, which guarantees that p_2 is strictly smaller than 1. Recall that $p_0 = 1$ is a fixed point of (14). But with the giant component formation, we can break away from the shackles of "being stuck" at $p_0 = 1$. With $p_2 < 1$, we hope to find a better fixed point of (14) to which our density evolution will converge.

Towards this end, ideally one wants Equation (6) to have the property

$$p_{j+1} = (1 + e^{-\eta} - e^{-\eta p_j})^{d-1} < p_j, \tag{15}$$

for all $p_j \in (0, 1)$. Let's take a closer look at the fixed point equation

$$t = f(t) = (1 + e^{-\eta} - e^{-\eta t})^{d-1}. \tag{16}$$

As mentioned, one solution is $t_1^* = 1$. As we can break away from t_1^*, fortunately there exists another solution approximately at $t_2^* \simeq e^{-\eta(d-1)}$ which is close to 0. To see this, consider the equation $y = (1 + e^{-\eta} - e^{-\eta x})^{d-1}$. Suppose that $0 < x = e^{-\eta(d-1)} \ll 1$. Then, $e^{-\eta x} \simeq 1$ and $1 + e^{-\eta} - e^{-\eta x} \simeq e^{-\eta}$. Thus, $y = x$ which shows that $e^{-\eta(d-1)}$ is approximately another fixed point of (14).[10] From now on, we will refer to this fixed point as the error floor p^*.

Lemma 7. *Let $d = 5$. If $2.33K \leq M \leq 13.99K$, then the fixed point equation (16) has exactly 2 solutions for $t \in [0, 1]$: $t_1^* = 1$ and $t_2^* \simeq e^{-\eta(d-1)}$ (See Figure 10). For $d = 8$, a similar result holds if $2.63K \leq M \leq 47.05K$.*

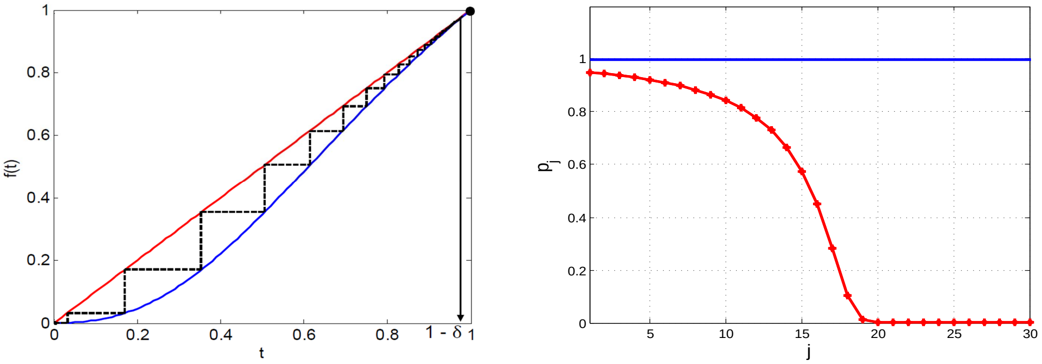
See Appendix D for the proof.

The following corollary is a direct result of Lemma 7.

Corollary 8. *For any $\epsilon > 0$, there exists a constant $\ell(\epsilon)$ such that $p_\ell \leq p^* + \epsilon$.*

Table IV illustrates how the error floor p^* and the minimum ratio of right nodes to active left nodes $c = M/K$ change for different values of d. If our reliability target allows the error floor to be set at 1.1×10^{-3}, then $d = 5$ minimizes the number of required right nodes. Recall that the total number of measurements is $m = 4M$ which matches the result of Table II. (See Section IV) If one wants to achieve smaller error floor, then d and c should be both increased.

[10] Of course, one can easily find the exact solution to (16), using numerical methods for given values of d and η.



(a) The density evolution curve for parameters $d = 5$ and $\eta = 2$.

(b) The evolution of p_j after each iteration for $d = 5$ and $\eta = 2$.

Fig. 10: Figure (a) illustrates the density evolution equation, $p_{j+1} = f(p_j)$, for Regular PhaseCode. In order to track the evolution of p_j, pictorially, one draws a vertical line from (p_j, p_j) to $(p_j, f(p_j))$, and then a horizontal line between $(p_j, f(p_j))$ and $(f(p_j), f(p_j))$. Since the two curves meet at $(1, 1)$ if $p_0 = 1$, then p_j gets stuck at 1. However, if $p_0 = 1 - \delta$, p_j decreases after each iteration, and it gets very close to 0. Figure (b) illustrates the same phenomenon by showing the evolution of p_j versus the iteration, j. Note that in this example, p_j gets very close to 0 after only 20 iterations.

In the density evolution analysis so far, we have shown that the *average* fraction of active signal components that cannot be recovered will be arbitrarily close to the error floor after a fixed number of iterations, provided that the tree-like assumption is valid. It remains to show that the actual fraction of left nodes that are not in the giant component after ℓ iterations is highly concentrated around p_ℓ. Towards this end, first in Lemma 9 we show that a neighborhood of depth ℓ of a typical edge is a tree with high probability for a constant ℓ. Second, in Lemma 10, we use the standard Doob's martingale argument [36], to show that the number of active signal components that are not recovered after ℓ iterations of the algorithm is highly concentrated around Kp_ℓ.

Consider a directed edge $\vec{e} = (v, c)$ from a left-node v to a right-node c. Define the directed neighborhood of depth ℓ of $(\vec{e})$ as $\mathcal{N}_{\vec{e}}^{\ell}$, that is the subgraph of all the edges and nodes on paths having length less than or equal to ℓ, that start from v and the first edge of the path is not $\vec{e}$. As an example, the directed neighborhood of depth 2 of $(\vec{e})$ is shown in Figure 9.

Lemma 9. *For a fixed ℓ^*, $\mathcal{N}_{\vec{e}}^{2\ell^*}$ is a tree-like neighborhood with probability at least* $1 - \mathcal{O}(\log(K)^{\ell^*}/K)$.

The proof is provided in Appendix E.

Lemma 10. *Over the probability space of the ensemble of d-left-regular graphs $\mathcal{C}_1^K(d, M)$, let Z be the number of uncolored edges*[11] *after ℓ iterations of the PhaseCode algorithm. Then, for any $\epsilon > 0$, there exist a large enough K and constants β and γ such that*

$$|\mathbb{E}[Z] - Kdp_\ell| < Kd\epsilon/2 \tag{17}$$

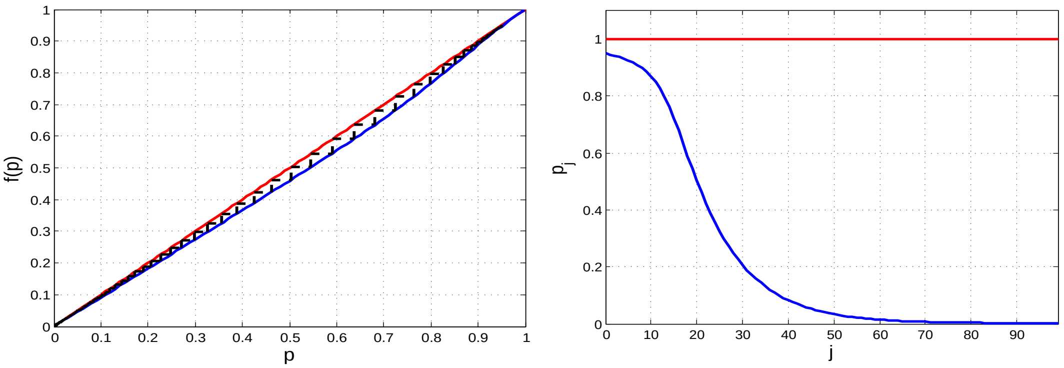
$$\mathbb{P}(|Z - Kdp_\ell| > Kd\epsilon) < 2e^{-\beta\epsilon^2 K^{1/(4\ell+1)}}, \tag{18}$$

where p_ℓ is derived from the density evolution equation (14).

The proof is provided in Appendix F.

Now gathering the results of Corollary 8 and Lemmas 5 and 10 completes the proof of Theorem 2. Note that since the construction of the bipartite graph is random, the fraction p of the non-zero components that can be missed are distributed uniformly at random among the K non-zero components. Indeed, the missed components are only a function of the graph structure that has a distribution which is oblivious to the indices of the left nodes by construction. Further, note that the dominant probability of error is due to the event that the giant component is not formed in the second iteration which happens with probability $\mathcal{O}(1/K)$. It is worth mentioning that Lemma 9 is used only to prove Lemma 10. Thus, the event that an edge does not have a tree-like neighborhood, which happens with probability $\mathcal{O}(\frac{\log(K)^{\ell^*}}{K})$, is not an error event of the algorithm. Given that a giant component has been formed after the second step of the algorithm, the error event of the algorithm is the event that more than a fraction p of the non-zero signal components are missed, and the probability of such event is upper bounded in (18).

[11] An edge is colored if its corresponding left node is colored.



(a) The density evolution curve for parameters $K = 10^5$, $\epsilon = 0.1$ and $D = 10^3$.

(b) The evolution of p_j after each iteration for parameters $K = 10^5$, $\epsilon = 0.1$ and $D = 10^3$.

Fig. 11: Figure (a) illustrates the density evolution equation for Irregular PhaseCode, which is similar to Figure 10a. Figure (b) illustrates the same phenomenon showing the evolution of p_j versus the iteration, j. Note that in this example, since $\epsilon = 0.1$ and we are operating very close to the capacity, p_j gets very close to 0 after around 90 iterations, which is much larger than around 20 iterations needed by Regular PhaseCode so that p_j gets very close to 0. The reason is that the gap between the two curves in (a) gets smaller once the number of measurements is close to the capacity.

C. *Proof of Theorem 3*

Recall that we design the left degree distribution $\lambda(x) = \sum_{i\geq 1} \lambda_i x^{i-1}$ of Irregular PhaseCode as follows: $\lambda_i = 0$ for $i \geq D+1$ and

$$\lambda_i = \frac{1}{i-1} \times \frac{1}{h(D-1)},\ 2 \leq i \leq D, \tag{19}$$

where D is a (large) constant and $h(x) = \sum_{i=1}^{x} 1/i$.

We design the number of right nodes to be $M = K/(1-\epsilon) \simeq K(1+\epsilon)$. How to choose constants D and ϵ will be shortly clarified in Lemma 12. The average degree of left nodes (of the pruned graph with K active left nodes) is $\bar{d} = \frac{1}{\sum_i \lambda_i/i}$. To see this, let E be the number of edges of the graph. Then, the number of left nodes of degree i is $E\lambda_i/i$ since λ_i is the fraction of edges with degree i on the left. Thus, the number of left nodes is $\sum_i E\lambda_i/i$. So the average left degree is

$$\bar{d} = \frac{E}{\sum_i E\lambda_i/i} = \frac{1}{\sum_i \lambda_i/i}.$$

Thus, with our design,

$$\bar{d} = (\sum_{i=2}^{D} \frac{\lambda_i}{i})^{-1} = h(D-1)\frac{D}{D-1}.$$

Consequently, the Poisson density parameter of the right-node degree distribution is:

$$\eta = \frac{K\bar{d}}{M} = h(D-1)\frac{D}{D-1}(1-\epsilon).$$

Lemma 11. *Let $f(x) = \lambda(1 + e^{-\eta} - e^{-\eta x})$. The fixed point equation $x = f(x)$ has exactly two solutions, $x_1^* = 1$ and $0 < x_2^* < 1$, in the interval $x \in [0,1]$. Furthermore, if $f'(1) > 1$, then $f(x) < x$ for $x \in (x_2^*, 1)$.*

See Appendix G for the proof.

As shown in Lemma 11, given that $f'(1) > 1$, the density evolution has a fixed point at 1, and the other fixed point of the equation is approximately $p^* \simeq \lambda(e^{-\eta})$, which corresponds to the error floor of the algorithm. In the following lemma, we show that for any arbitrarily small numbers p^* and ϵ, there exists a large enough constant $D(p^*, \epsilon)$ such that $f'(1) > 1$. This shows that with only $4M = 4K/(1-\epsilon) \simeq 4K(1+\epsilon)$ measurements, Irregular PhaseCode algorithm can recover an arbitrarily-close-to-one fraction of the non-zero signal components. So given that the coloring procedure starts (the density evolution equation can be started from $1-\delta$), Irregular PhaseCode is capacity-approaching.

Now we show that a linear size giant component of colored left nodes can be formed similar to Lemma 5 using a second stage of only $m' = \epsilon' K$ extra measurements. By assumption of Theorem 3, the support of the non-zero components of the signal is uniformly random. Now fix some arbitrarily small constant $\delta' > 0$. Let $\tilde{\boldsymbol{x}}$ be the vector of the first $\delta' n$ components of the signal. By the law of large numbers, the number of non-zero elements of $\tilde{\boldsymbol{x}}$ is $\delta' K + o(K)$. Consider the sub-problem of forming a giant component of size linear in K in $\tilde{\boldsymbol{x}}$. By Lemma 5, one can design $m' = 14\delta' K$ measurements to form the giant component. Thus, $\epsilon' = 14\delta'$. Since δ' can be made arbitrarily small, ϵ' can also be made arbitrarily small.

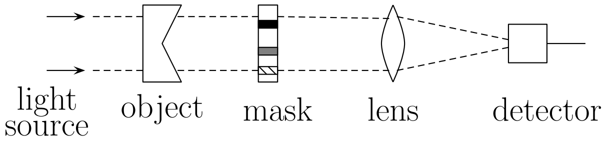


Fig. 12: A typical setup for many optical system where the object of interest is passed through a coded diffraction pattern or a mask , and then through a Fourier lens.

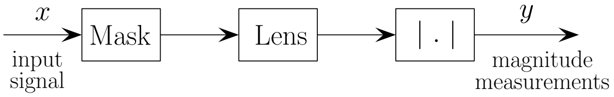


Fig. 13: The block diagram of an optical imaging system where signal x is passed through a mask (modulated by a diagonal matrix), and then passed through a lens (DFT matrix). The magnitude block, $|.|$, is showing that the phase information is not available in the measurements.

The main lemma for establishing the proof of Theorem 3 is as follows.

Lemma 12. *For any $p^* > 0$ and any $\epsilon > 0$, there exists a large enough constant $D(\epsilon, p^*)$ such that $M = K(1-\epsilon)^{-1} \simeq K(1+\epsilon)$ is the number of right nodes (bins), and p_j converges to p^* as j goes to infinity.*

See Appendix H for the proof.

Corollary 13. *Given that $p_2 = 1 - \delta$, for any $\epsilon_1 > 0$, there exists a constant $\ell(\epsilon_1)$ such that $p_\ell \leq p^* + \epsilon_1$.*

The rest of the proof is similar to Theorem 2. It remains to show that the actual fraction of active signal components that are not recovered after ℓ iterations is highly concentrated around p_ℓ. Since the maximum degree of left nodes is again a constant D, the exact procedure in Section V-B (Lemmas 9 and 10) can be used to get a similar concentration bound as in Lemma 10. Now the total number of measurements is $m = 4K(1+\epsilon) + m' = 4K(1+\epsilon+\epsilon')$. Since ϵ and ϵ' can be made arbitrarily small, the proof of Theorem 3 is complete.

VI. Fourier-Friendly PhaseCode

In some applications such as optical imaging [9], [21], the design of the measurement matrix cannot be arbitrary. In optical imaging, the object of interest, signal $\boldsymbol{x}$, can be passed through an optical diffraction pattern or a mask and an optical Fourier lens. A typical setup for optical imaging is shown in Figure 12. With a complex-valued mask, we can modulate each component of the signal x_i by some complex number d_i, while the lens takes the Fourier transform of the signal. For example, consider passing the signal through a mask and then Fourier lens which is common in optical imaging. The output of this transform is $\boldsymbol{FDx}$, where $\boldsymbol{F}$ is the DFT matrix of length n and $\boldsymbol{D} \in \mathbb{C}^{n\times n}$ is a diagonal mask matrix (Figure 13). In general, it is possible to have multiple stages of masks and lenses. While increasing the number of stages can make the system more complex, in many optical systems, having up to two stages is considered practical [37], [38]. In our proposed solution, we will have two masks for all measurements.

In this section, we show how one can have a Fourier-friendly implementation of the set of measurements described in previous sections. We first provide an overview of the result of [26] on constructing a sparse-graph code using "Chinese Remainder Theorem", in Subsection VI-A. In Subsection VI-B, we show how our proposed measurements can be obtained in a Fourier-friendly setup, with the aid of the result of [26].

A. Ensemble of Graphs Constructed by Chinese Remainder Theorem

In this subsection, we provide a brief overview of the result in [26] that uses the "Chinese Remainder Theorem" (CRT) to construct a deterministic and well-structured coding matrix that is also of practical interest. We use this construction to design a Fourier-friendly measurement matrix. For more details about the theory of the ensemble of graphs constructed by the CRT, we refer the readers to [26].

In Section V-B, we analyzed the performance of PhaseCode for the ensemble of graphs $\mathcal{C}_1^K(d, M)$. In this ensemble, each left node is connected to exactly d right nodes randomly. Now we consider another ensemble $\mathcal{C}_2^K(\mathcal{F}, m)$. Define the set $\mathcal{F}$ as $\mathcal{F} = \{f_1, f_2, \ldots, f_d\}$. Partition the right nodes into d sets. Let the number of right nodes in stage i be f_i; thus, $\sum_{i=1}^d f_i = m$. In this construction, each left node is connected to exactly one right node per stage randomly. Therefore, we again end up with having a bipartite graph with left regular degree d. Assuming that $f_i = F + \Theta(1)$ for all i and consequently $F = \Theta(K)$, the edge degree distribution of the right nodes does not change for large enough K and is given in (12). Therefore, the tree analysis and the density evolution equation stated in (14) remain the same, and one can essentially get all the previous results using this ensemble.

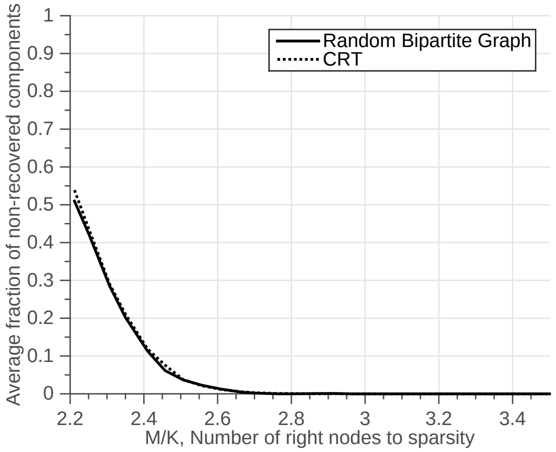


Fig. 14: **Comparison of random left-regular bipartite graph ensemble and CRT ensemble.** We choose the left degree $d = 7$, and construct an appropriate CRT ensemble based on $\mathcal{F} = \{47, 49, 50, 53, 57, 59, 61\}$. The number of right nodes is determined by $\mathcal{F}$, i.e., $M = \sum_{i=1}^{d} f_i = 376$. Each operating point is averaged over 10000 runs. We observe negligible difference in performance between the two ensembles.

Note that sampling a graph from $\mathcal{C}_2^K(\mathcal{F}, m)$ has no practical advantage over sampling from the ensemble $\mathcal{C}_1^K(d, M)$. However, we use the CRT to show that if the K non-zero components of the signal is chosen uniformly at random with replacement from the n components, and if K is in the sub-linear regime (more specifically, $K = n^{\delta}$ for some $\delta \in (0, 1)$), one can design a deterministic coding matrix which consists of d stages of sub-matrices with rows that are circularly-shifted versions of a deterministic *subsampling* pattern. The subsampling rate at stage i is f_i. In the following example, we demonstrate how the deterministic matrix is constructed.

Example 3. Suppose that the coding matrix has two stages with $f_1 = 2$ and $f_2 = 3$. Assume that $n = 6$. Then, the coding matrix is

$$\left(\begin{array}{cccccc} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right).$$

Now, we formally define the ensemble of graphs constructed by the CRT. First, assume $n = \prod_{i=1}^{d} f_i$ (i.e. $K = \Theta(n^{1/d})$). Partition the set of $m = \sum_{i=1}^{d} f_i$ right nodes to d stages in the trivial way. Suppose that the K non-zero components of the signal are chosen uniformly at random with replacement from the n components. Note that the "with replacement" assumption might lead to having a signal with less than K non-zero components, but this is only a technical assumption that we need to make, and via simulations we will show the good performance of the CRT-based code for exactly K-sparse signal. Let $\mathcal{I} = (i_1, i_2, \ldots, i_K)$ denote the non-zero components where $1 \le i_k \le n,\ 1 \le k \le K$. We associate the integers from 0 to $n-1$ to d numbers $(r_1, r_2, \ldots, r_d)$ using the CRT, where $0 \le r_i \le f_i - 1$; thus, i_k uniquely determines one right node per stage. The way this association is done will be explained shortly. Then, each active left node i_k is connected to the associated set of right nodes that are determined by $(r_1, r_2, \ldots, r_d)$. The ensemble $\mathcal{C}_3^K(\mathcal{F}, m)$ is the collection of all the graphs that are constructed as described. Furthermore, the uniformly at random selection of $\mathcal{I}$ makes sure that all these graphs occur with equal probability. See [26] for details.

To show how we associate $\mathcal{I}$ to $(r_1, r_2, \ldots, r_d)$, we need to review the Chinese Remainder Theorem. Let $n = \prod_{i=1}^{d} f_i$ and f_i's are pairwise co-prime positive integers. The theorem states that every integer n' between 0 and $n-1$ is uniquely represented by the sequence $(r_1, r_2, ..., r_d)$ of its remainders modulo $f_1, f_2, \ldots, f_d$ respectively and vice-versa. We use this unique CRT mapping to associate the active left nodes with d right nodes.

Lemma 14. *[26] The ensembles $\mathcal{C}_2^K(\mathcal{F}, m)$ and $\mathcal{C}_3^K(\mathcal{F}, m)$ are identical.*

Proof: Clearly, $\mathcal{C}_3^K(\mathcal{F}, m) \subset \mathcal{C}_2^K(\mathcal{F}, m)$. The reverse is also true by CRT since there is a unique integer between 0 to $n-1$ with remainders r_i modulo f_i for all i. ■

Figure 14 demonstrates the performance of PhaseCode with two ensembles: $\mathcal{C}_1^K(d, M)$ and $\mathcal{C}_3^K(\mathcal{F}, m)$. We choose $d = 7$ and $\mathcal{F} = \{47, 49, 50, 53, 57, 59, 61\}$. Thus, $M = \sum_{i=1}^{d} f_i = 376$. We varied the value of K ($107 \le K \le 170$) such that M/K varies between 2.2 and 3.5. Each point is averaged over 10000 runs to determine the error probability. One can observe negligible difference between the performance of the algorithm for the two ensembles.

In the following we provide remarks of how one can extend the above construction of CRT.

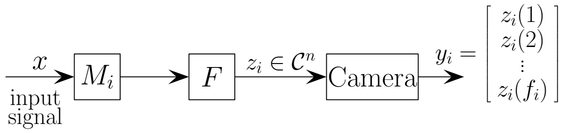

x
input signal
M_i
F
$z_i \in \mathcal{C}^n$
Camera
$y_i = \begin{bmatrix} z_i(1) \\ z_i(2) \\ \vdots \\ z_i(f_i) \end{bmatrix}$

Fig. 15: The block diagram of Fourier-friendly compressive phase retrieval using the CRT matrix. The figure shows stage i of the CRT matrix ($1 \leq i \leq d$). The signal of interest, x, is passed through a binary mask corresponding to stage i, and then the Fourier lens. The output of this experiment is signal z_i of length n. However, these n measurements are not unique; they are n/f_i replicas of f_i unique measurements. Thus, the camera only reads the first f_i components of z_i.

Remark In the above example of CRT construction, we implicitly assumed $K = \Theta(n^{1/d})$. The technique can be extended to cases where $K = \Theta(n^{\alpha/d})$ for $0 \leq \alpha < d$. Instead of using $\mathcal{F}$ as heights of the d stages of the bipartite graph, we use $\mathcal{F}' = \{f'_1, ..., f'_d\}$, where

$$f'_i = \prod_{j=0}^{\alpha-1} f_{((i+j) \bmod d)+1}.$$

For example, if $\alpha = 2$ and $d = 7$, one can convert a set of coprimes

$$\{f_1, f_2, f_3, f_4, f_5, f_6, f_7\}$$

to the set

$$\mathcal{F} = \{f_1 f_2, f_2 f_3, f_3 f_4, f_4 f_5, f_5 f_6, f_6 f_7, f_7 f_1\}.$$

Then, $M = \sum_{i=1}^{d} \prod_{j=0}^{\alpha-1} n_{((i+j) \bmod d)+1} = \Theta(n^{2/d})$, which is in the order of K. Because $\mathcal{F}$ can be chosen from a dense set of coprimes, one can always choose it carefully to induce a right number of measurements. For the most general case where $K = \Theta(n^{p/q})$ and $0 \leq p/q < 1$, one can use a similar extension and construction by finding q coprimes and stacking p of them in each stage. We omit details of the technique and refer interested readers to [26].

B. Fourier-Friendly Compressive Phase Retrieval

Without loss of generality, we consider only a 1-D case for $\boldsymbol{x}$ here, though our arguments extend in a straight-forward way to 2-D images as well. Suppose that the signal of interest $\boldsymbol{x}$ is sparse in the Fourier domain, which is of interest in many optical imaging settings. Let $\boldsymbol{X} = \boldsymbol{F}\boldsymbol{x}$ be the Fourier transform of the signal. In Subsection VI-A, we showed that the coding matrix $\boldsymbol{H}$ can be realized using d stages of circulant matrices without changing the performance of sparse-graph codes. To have a Fourier-friendly implementation of the CRT code matrix, we expand each stage of the $f_i \times n$ matrix to a circulant $n \times n$ matrix. Let $\boldsymbol{C}$ denote this circulant coding matrix for one stage. In the following, we show that how using our proposed CRT code matrix, one can have access to all the necessary measurements using *only diagonal masks and lenses*. Note that we are interested in measurements of the modulated signal by complex exponentials such as $e^{\mathbf{i}\omega\ell}$ or by magnitude modulators $\cos(\omega\ell)$. First let us see how the *plain* measurements without these modulations can be obtained if the coding matrix is circulant. The plain measurements are $|\sum_j C_{ij} X_j|$. Since $\boldsymbol{C}$ is circulant, the eigenvectors of $\boldsymbol{C}$ are the columns of a unitary Fourier matrix [39]. Thus, the eigenvalue decomposition of $\boldsymbol{C}$ is $\boldsymbol{C} = \boldsymbol{F}\boldsymbol{D}\boldsymbol{F}^{-1}$ for some diagonal matrix $\boldsymbol{D}$. Hence, we construct our measurements by modulating the signal $\boldsymbol{x}$ with the diagonal mask $\boldsymbol{D}$ and then taking a Fourier transform by using an optical lens:

$$\begin{aligned} |\boldsymbol{F}\boldsymbol{D}\boldsymbol{x}| &= |\boldsymbol{F}\boldsymbol{F}^{-1}\boldsymbol{C}\boldsymbol{F}\boldsymbol{x}| \\ &= |\boldsymbol{C}\boldsymbol{F}\boldsymbol{x}| \\ &= |\boldsymbol{C}\boldsymbol{X}|. \end{aligned}$$

For each stage of the CRT code matrix (there are d stages overall), we need one physical experiment. The physical experiment corresponding to the i-th stage, where $1 \leq i \leq d$, gives us n/f_i replicas of f_i unique measurements in one shot. As illustrated in Figure 15, for each experiment, the camera measures only one copy of the f_i measurements. Let $\boldsymbol{y}_i \in \mathcal{C}^{f_i}$ be the measurements corresponding to stage i. Then, the measurements of the different stages are gathered to form the measurement vector $\boldsymbol{y} \in \mathcal{C}^m$ as follows:

$$\boldsymbol{y} = [\boldsymbol{y}_1^T, \boldsymbol{y}_2^T, \ldots, \boldsymbol{y}_d^T]^T.$$

Thus, the actual sample complexity is still $m = \sum_{i=1}^{d} f_i = \Theta(K)$.

Now we explain how one can get access to all the necessary measurement $y_{1,i}$ to $y_{4,i}$. We explain the construction for $y_{1,i}$. Other measurements can be similarly realized. We use 3 blocks of Fourier transforms (lenses) and 2 masks as follows. Let $\tilde{\boldsymbol{D}}$

be a diagonal matrix such that $\tilde{d}_{\ell\ell} = e^{\mathrm{i}\omega\ell}$. We are interested in constructing the measurements of the form $|\boldsymbol{C}\tilde{\boldsymbol{D}}\boldsymbol{X}|$. This can be done by using two masks, $\tilde{\boldsymbol{D}}$ and $\boldsymbol{D}$, with three Fourier lenses as follows.

$$|\boldsymbol{F}\boldsymbol{D}\boldsymbol{F}\tilde{\boldsymbol{D}}\boldsymbol{F}\boldsymbol{x}| = |\boldsymbol{F}\boldsymbol{F}\boldsymbol{C}\boldsymbol{F}^{-1}\boldsymbol{F}\tilde{\boldsymbol{D}}\boldsymbol{X}| \tag{20}$$

$$= |\boldsymbol{F}^2\boldsymbol{C}\tilde{\boldsymbol{D}}\boldsymbol{X}|. \tag{21}$$

Note that $\boldsymbol{F}^2$ is just a permutation matrix so we can construct all the measurements $y_{1,i}$ using only two masks and Fourier lenses.

Remark Since each optical lens is equivalent to a Fourier transform, we can also implement a compressive Fourier-friendly phase retrieval algorithm, when $\boldsymbol{x}$ is sparse (and $\boldsymbol{X}$ is not sparse) by just adding an optical lens to the measurement system as follows. Suppose that $\boldsymbol{A}$ is a Fourier-friendly measurement system that is able to recover $\boldsymbol{x}$, when $\boldsymbol{X}$ is sparse. That is, one is measuring the sparse signal $\boldsymbol{X}$ with measurement matrix $\boldsymbol{A}\boldsymbol{F}^{-1}$. Then, $\boldsymbol{A}\boldsymbol{F}$ is a Fourier-friendly measurement matrix that is able to recover $\boldsymbol{x}$, when $\boldsymbol{x}$ is sparse since $\boldsymbol{A}\boldsymbol{F}\boldsymbol{x} = \boldsymbol{A}\boldsymbol{F}^{-1}\boldsymbol{F}^2\boldsymbol{x}$. Note that $\boldsymbol{F}^2$ is just a permutation matrix; thus, $\boldsymbol{F}^2\boldsymbol{x}$ is a sparse signal that is again measured by $\boldsymbol{A}\boldsymbol{F}^{-1}$.

VII. Robust PhaseCode

In this section, we consider the noisy compressive phase retrieval problem. The noisy compressive phase retrieval problem is to recover a K-sparse complex signal $\boldsymbol{x}$, from a set of quadratic measurements

$$y_i = \left|\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{x}\right|^2 + w_i, \quad i \in [m],$$

where $\boldsymbol{a}_i^{\mathrm{H}} \in \mathbb{C}^n$ are rows of the measurement matrix $\boldsymbol{A} \in \mathbb{C}^{m\times n}$, w_i's are noise, and $[m]$ denotes the set $\{1, 2, \ldots, m\}$. We consider the regime where there exist two constants β and δ such that $K = \beta n^{\delta}$, $\delta \in (0, 1)$. We assume that w_i's are independent, zero-mean, sub-exponential [40] random variables. This model is considered in many phase retrieval literatures [14], [15], [41].

We also assume that signal $\boldsymbol{x}$ is quantized, which means that the components of $\boldsymbol{x}$ lie in a finite set of complex numbers. More specifically, let L_m and L_p be the number of possible magnitudes and phases of the non-zero components, respectively. Then, each component of $\boldsymbol{x}$ is in the set

$$\mathbb{S} = \{u\varepsilon e^{\mathbf{i}\frac{2\pi(v-1)}{L_p}} | u \in [L_m], v \in [L_p]\} \cup \{0\} \subset \mathbb{C},$$

where $\varepsilon > 0$. Quantized signals can be good approximations of the real world signals and are natural for signal processing with computers [42], [43].

We propose two schemes to robustify PhaseCode in the presence of noise: almost-linear scheme and sublinear scheme. The main results of this section are the following theorems.

Theorem 15. *The almost-linear scheme can recover a random fraction $1-p$, for arbitrarily small p, of the non-zero elements of $\boldsymbol{x}$ with probability $1 - \mathcal{O}(1/K)$, with $\Theta(K\log(n))$ measurements. The computational complexity of the algorithm is $\Theta(L_m L_p n\log(n))$.*

Theorem 16. *The sublinear scheme can recover a random fraction $1-p$, for arbitrarily small p, of the non-zero elements of $\boldsymbol{x}$ with probability $1 - \mathcal{O}(1/K)$, with $\Theta(K\log^3(n))$ measurements. The computational complexity of the algorithm is $\Theta(L_m L_p K\log^3(n))$.*

See the proofs of Theorems 15 and 16 in Appendix I and L. Details of the measurement design and the decoding algorithm are shown in the following subsections.

A. Almost-linear Scheme

The idea of the almost-linear scheme is to encode the columns as different patterns. With the number of measurements in each right node being $\Theta(\log(n))$, the patterns are guaranteed to be different enough, so that we can successfully resolve singletons, mergeable multitons, and resolvable multitons.

1) Design of Measurements: Instead of using the 4-by-n trigonometric modulation matrix, we use a new random matrix $\boldsymbol{A}_0 = \{a_{ij}\}_{P\times n}$ whose entries are i.i.d. with the following distribution:

$$a_{ij} = \begin{cases} 0, & \text{with probability } 1/2 \\ e^{\mathbf{i}\theta_{ij}}, & \text{with probability } 1/2, \end{cases} \tag{22}$$

where θ_{ij}'s are i.i.d. and uniformly distributed in $[0, 2\pi)$. We call $\boldsymbol{A}_0$ the *test matrix*, and we can show that we need $P = \Theta(\log(n))$ for each right node to achieve successful recovery.

For the almost-linear algorithm, the measurement matrix of the lth right node is $\boldsymbol{A}_l = \boldsymbol{A}_0 \mathrm{diag}(\boldsymbol{h}_l)$. Without loss of generality, we omit index l, and simply use $\boldsymbol{h}$ to denote the coding pattern (the left nodes connected to the right node) of a right node. Then the measurements of this right node are

$$y_i = \left|\boldsymbol{a}_i^{\mathrm{H}} \mathrm{diag}(\boldsymbol{h})\boldsymbol{x}\right|^2 + w_i,\ i \in [P], \tag{23}$$

where $\boldsymbol{a}_i^{\mathrm{H}}$ is the ith row of $\boldsymbol{A}_0$, and the noise $w_i \in \mathbb{R}$, $i \in [n]$ satisfies the properties mentioned earlier. To simplify notation, we define a linear map $\mathcal{A}$ from $\mathbb{C}^{n\times n}$ to $\mathbb{R}^P$:

$$\mathcal{A}:\ \boldsymbol{Z} \mapsto \{\boldsymbol{a}_i^{\mathrm{H}} \boldsymbol{Z} \boldsymbol{a}_i\}_{i\in[P]}. \tag{24}$$

Now according to (23), by defining $\boldsymbol{z} = \mathrm{diag}(\boldsymbol{h})\boldsymbol{x}$, we have $\boldsymbol{y} = \mathcal{A}(\boldsymbol{z}\boldsymbol{z}^{\mathrm{H}}) + \boldsymbol{w}$, where $\boldsymbol{y} = \{y_i\}_{i\in[P]}$ and $\boldsymbol{w} = \{w_i\}_{i\in[P]}$ are the measurement vector and noise vector, respectively. We call $\boldsymbol{z}$ the *true signal* corresponding to this right node.

2) Decoding Algorithm: As mentioned earlier, the PhaseCode algorithm requires the measurements in each right node to enable three operations: detecting singletons, resolving strong doubletons, and detecting resolvable multitons. Using our new measurement system, these operations can be done reliably by a simple guess-and-check method: we guess all possible indices, magnitudes, and relative phases, and use an energy test to decide whether our guess is correct. For any of the three operations, we make a hypothesis on the unknown index, magnitude, and phase of the true signal $\boldsymbol{z}$ and construct the corresponding hypothesis signal $\hat{\boldsymbol{z}}$. For example, when we do singleton detecting, if our hypothesis is that the right node is a singleton, and that the location index of the active component is 5 with the magnitude being 3ε, we construct $\hat{\boldsymbol{z}} = 3\varepsilon \boldsymbol{e}_5$, where $\boldsymbol{e}_i$ denotes the ith vector of the canonical basis. Similarly, we can resolve strong doubletons. For instance, suppose that we know that a right node is connected to two active components which are located at positions 2 and 5, respectively, and we also know the magnitudes of the two components are 2ε and 3ε, respectively. Then, if we can make a hypothesis that the relative phase is $\frac{\pi}{4}$, we can construct $\hat{\boldsymbol{z}} = 2\varepsilon \boldsymbol{e}_2 + 3\varepsilon e^{\mathbf{i}\frac{\pi}{4}} \boldsymbol{e}_5$. Then, we need to check whether our hypothesis is correct. To do this, we perform an ℓ_1 norm energy test shown in (25):

$$\begin{aligned} &\hat{\boldsymbol{z}} \sim \boldsymbol{z},\ \text{if}\ \frac{1}{P}\left\|\boldsymbol{y} - \mathcal{A}(\hat{\boldsymbol{z}}\hat{\boldsymbol{z}}^{\mathrm{H}})\right\|_1 < t_0, \\ &\hat{\boldsymbol{z}} \nsim \boldsymbol{z},\ \text{otherwise}, \end{aligned} \tag{25}$$

where $\hat{\boldsymbol{z}} \sim \boldsymbol{z}$ means $\hat{\boldsymbol{z}}$ and $\boldsymbol{z}$ are equal up to a global phase, and t_0 is the threshold. The intuitive reason why we do this test is that when $\hat{\boldsymbol{z}} \sim \boldsymbol{z}$, $\mathcal{A}(\hat{\boldsymbol{z}}\hat{\boldsymbol{z}}^{\mathrm{H}}) = \mathcal{A}(\boldsymbol{z}\boldsymbol{z}^{\mathrm{H}})$, then $\boldsymbol{y} - \mathcal{A}(\hat{\boldsymbol{z}}\hat{\boldsymbol{z}}^{\mathrm{H}}) = \boldsymbol{w}$, whose energy should be small. Conversely, when $\hat{\boldsymbol{z}} \nsim \boldsymbol{z}$, the energy of $\boldsymbol{y} - \mathcal{A}(\hat{\boldsymbol{z}}\hat{\boldsymbol{z}}^{\mathrm{H}})$ should be large. Here, we give a result on the error probability of the energy test.

Lemma 17. *When $P = \Theta(\log(n))$ and ε is appropriately large, with proper threshold t_0, the error probability of the energy test shown in (25) is $\mathcal{O}(1/n^2)$.*

The proof of this lemma follows the similar idea which appears in Lemma 14 in [44]. We can also show that we need to perform $\Theta(n)$ energy tests before the algorithm stops. Then, using Lemma 17 and some basic principles in probability theory, we can show that the failure probability of the almost-linear scheme is $\mathcal{O}(1/K)$. As for the sample and computational complexity, since we have $\Theta(\log(n))$ measurements for each right node and $\Theta(K)$ right nodes, the sample complexity of the almost-linear scheme would be $\Theta(K\log(n))$; and since the computational cost of each test is $\Theta(L_m L_p \log(n))$ and there are $\Theta(n)$ tests, the computational complexity of the almost-linear scheme is $\Theta(L_m L_p n \log(n))$.

B. Sublinear Scheme

Although the $\mathcal{O}(n\log(n))$ computational complexity of almost-linear scheme is compelling, we can further improve the computational complexity. Recall that in the noiseless scenario, we get the location index of the active component in a singleton and the non-recovered active component in resolvable multitons by only decoding the measurements of a recoverable right node. Based on this idea, we propose the sublinear scheme for the noisy scenario, which can achieve much lower computational cost compared to the almost-linear scheme, at the cost of slightly larger sample complexity.

1) Design of Measurements: In the sublinear scheme, the measurement matrix for each right node is designed to be a concatenation of the test matrix $\boldsymbol{A}_0$ defined in the almost-linear scheme and R *index matrices* $\boldsymbol{F}_1, \ldots, \boldsymbol{F}_R$. The test matrix $\boldsymbol{A}_0$ is still used to perform the energy tests and the index matrices are used to find the location indices.

Now we show how to design the index matrices. The main idea is to encode each column as a binary code such that we can directly decode the column index of the component to get recovered from the measurements. A similar idea is also used in the Chaining Pursuit method [45]. First, we define a deterministic matrix $\boldsymbol{B} = \{b_{ij}\} \in \{0,1\}^{R\times n}$, where $R = \lceil \log n \rceil$, and the ith column of $\boldsymbol{B}$ is the binary representation of the integer $i-1$. For example, when $n = 4$, we have,

$$\boldsymbol{B} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}.$$

We use $\boldsymbol{b}_i$ and $\boldsymbol{B}_j$ to denote the ith row and jth column of $\boldsymbol{B}$, respectively. Let $\boldsymbol{F}_0 \in \mathbb{C}^{Q\times n}$ be a random matrix whose elements are i.i.d. and uniformly distributed on the unit circle, and $\boldsymbol{F} = \boldsymbol{F}_0 \otimes \boldsymbol{B} \in \mathbb{C}^{RQ\times n}$. This means we have $\boldsymbol{F} = [\boldsymbol{F}_1^{\mathrm{H}}\ \boldsymbol{F}_2^{\mathrm{H}}\ \cdots \boldsymbol{F}_R^{\mathrm{H}}]^{\mathrm{H}}$, where $\boldsymbol{F}_i = \boldsymbol{F}_0 \mathrm{diag}(\boldsymbol{b}_i) \in \mathbb{C}^{Q \times n}$. By concatenating with the test matrix, the measurement matrix of the lth right node is $\boldsymbol{A}_l = [\boldsymbol{A}_0^{\mathrm{H}} \ \boldsymbol{F}^{\mathrm{H}}]^{\mathrm{H}} \mathrm{diag}(\boldsymbol{h}_l) \in \mathbb{C}^{(P+QR) \times n}$. Here, we give a simple example of $\boldsymbol{A}_l$. Let $n = 4$ and thus $R = 2$. We have

$$\boldsymbol{A}_l = \left[\begin{array}{cccc} \boldsymbol{A}_{0,1} & \boldsymbol{A}_{0,2} & \boldsymbol{A}_{0,3} & \boldsymbol{A}_{0,4} \\ \hline \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{F}_{0,3} & \boldsymbol{F}_{0,4} \\ \boldsymbol{0} & \boldsymbol{F}_{0,2} & \boldsymbol{0} & \boldsymbol{F}_{0,4} \end{array} \right] \mathrm{diag}(\boldsymbol{h}_l), \tag{26}$$

where $\boldsymbol{A}_{0,i}$'s and $\boldsymbol{F}_{0,i}$'s are the columns of $\boldsymbol{A}_0$ and $\boldsymbol{F}_0$. We can show that we need $Q = \Theta(\log^2(n))$ to reliably find the correct location index and we also need $P = \Theta(\log(n))$ to perform energy tests.

Consequently, there are $R + 1$ sets of measurements. The first set $\boldsymbol{y}_0 = \{y_{0,i}\}_{i \in [P]}$ is the same as the measurements in almost-linear scheme and is called the *test measurements*:

$$y_{0,i} = \left| \boldsymbol{a}_i^{\mathrm{H}} \boldsymbol{z} \right|^2 + w_{0,i}, \ i \in [P],$$

where $\boldsymbol{z} = \mathrm{diag}(\boldsymbol{h})\boldsymbol{x}$ and is still called the true signal. The other R sets $\boldsymbol{y}_j = \{y_{j,i}\}_{i \in [Q]}$, $j \in [R]$ correspond to the index matrices and are called the *index measurements*. Each set is composed of Q measurements:

$$y_{j,i} = \left| \boldsymbol{f}_{j,i}^{\mathrm{H}} \boldsymbol{z} \right|^2 + w_{j,i}, \ i \in [Q], \ j \in [R],$$

where $\boldsymbol{f}_{j,i}^{\mathrm{H}}$ is the ith row of $\boldsymbol{F}_j$. We also let $\boldsymbol{w}_j$'s be the noise vectors, $j \in \{0\} \cup [R]$.

2) Decoding Algorithm: The sublinear scheme can find the location index by only looking at the measurements. For example, assume that a right node with measurement matrix in (26) is a singleton whose non-zero component is at position 2. Then, the decoder can see that the elements of the first set of index measurements $\boldsymbol{y}_1$ have small absolute value since these measurements only contain noise. Now the decoder knows that the non-zero element should be in the first half of the signal. Then, the decoder observes that the elements in $\boldsymbol{y}_2$ have large energy. The decoder knows that if the right node is indeed a singleton, the only possible index of the non-zero component would be 2. Actually this procedure is a binary search on all the n indices of the signal. After this indexing process, the decoder can use the same procedure as the almost-linear scheme to construct a signal $\hat{\boldsymbol{z}}$ as the hypothesis of the true signal of this right node, and then use the testing measurements to perform the same energy test.

Now we formally show the details of the fast index search. Assume that $|\mathrm{supp}(\boldsymbol{z})| = T$, and there are T_s non-recovered active components connected to the right node. More specifically, $\boldsymbol{z} = \boldsymbol{z}_c + \boldsymbol{z}_s$, $|\mathrm{supp}(\boldsymbol{z}_s)| = T_s$, $\mathrm{supp}(\boldsymbol{z}_c) \cap \mathrm{supp}(\boldsymbol{z}_s) = \emptyset$, and we know a vector $\hat{\boldsymbol{z}}_c \sim \boldsymbol{z}_c$. Note that when $T = T_s = 1$, we have $\hat{\boldsymbol{z}}_c = \boldsymbol{z}_c = 0$. Our goal is to find the index l_s of the non-zero element in $\boldsymbol{z}_s$ when $T_s = 1$ and $\mathrm{supp}(\boldsymbol{z}_s) = \{l_s\}$. When $T = 1$ and $T > 1$, we are looking for non-zero component in a singleton and non-recovered non-zero component in a resolvable multiton, respectively. We subtract the measurements contributed by the signal components which are already known as follows. Let $\hat{y}_{j,i} = |\boldsymbol{f}_{j,i}^{\mathrm{H}} \hat{\boldsymbol{z}}_c|^2$; then, $\tilde{y}_{j,i} = y_{j,i} - \hat{y}_{j,i}$. We perform the following index tests for $j \in [R]$ with threshold $t_1 > 0$ to get l_s:

$$\begin{aligned} &\tilde{b}_j = 0, \ \text{if} \ \left| \frac{1}{Q} \sum_{i=1}^{Q} \tilde{y}_{j,i} \right| < t_1, \\ &\tilde{b}_j = 1, \ \text{otherwise}. \end{aligned} \tag{27}$$

The index tests output a binary string $\tilde{\boldsymbol{b}} = \{\tilde{b}_j\}_{j \in [R]}$. Note that if $T_s > 1$, we still get an output after the index tests, but the energy test with the test measurements prevents us from making mistakes. Lemma 18 states that with high probability $\tilde{b}_j = b_{jl_s}$.

Lemma 18. *When $Q = \Theta(\log^2(n))$, with proper threshold t_1, if $\mathrm{supp}(\boldsymbol{x}_s) = \{l_s\}$, then $\mathbb{P}\{\tilde{b}_j \neq b_{jl_s}\} = \mathcal{O}(1/K^3)$.*

Similar to the almost-linear scheme, using Lemma 18, we can prove that the failure probability of the sublinear scheme is $\mathcal{O}(1/K)$. Since the total number of measurements per each right node is $P + RQ = \Theta(\log^3(n))$, the sample complexity of the sublinear scheme is $\Theta(K \log^3(n))$. In terms of the computational complexity, since there are $\Theta(K)$ right nodes and a constant number of iterations, the computational complexity of the sublinear algorithm is $\Theta(L_m L_p K \log^3(n))$.

C. Simulation Results

In this subsection, we show simulation results for the noisy case that validate our theoretical results. The simulations are conducted in Python. Since the sublinear scheme has much lower computational complexity than the almost-linear scheme, we only conduct simulations on the sublinear scheme here. We define the signal-to-noise ratio (SNR):

$$\mathrm{SNR} = 10 \log_{10} \frac{\sum_{j=0}^{R} \left\| \boldsymbol{y}_j - \boldsymbol{w}_j \right\|_2^2}{\sum_{j=0}^{R} \left\| \boldsymbol{w}_j \right\|_2^2},$$

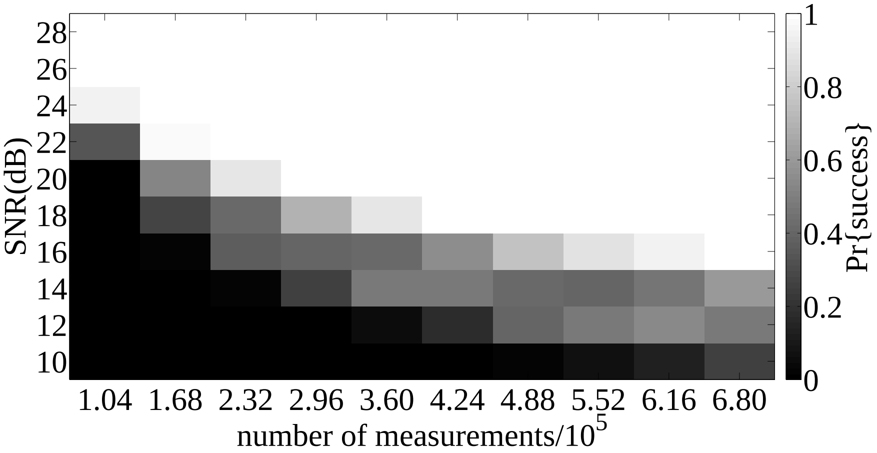


Fig. 16: **Probability of successful recovery.** We choose $n = 2^{20}$, $K = 50$, $L_m = 3$, and $L_p = 6$. Different values of SNR are tested, and for each set of parameters, 1000 experiments are conducted.

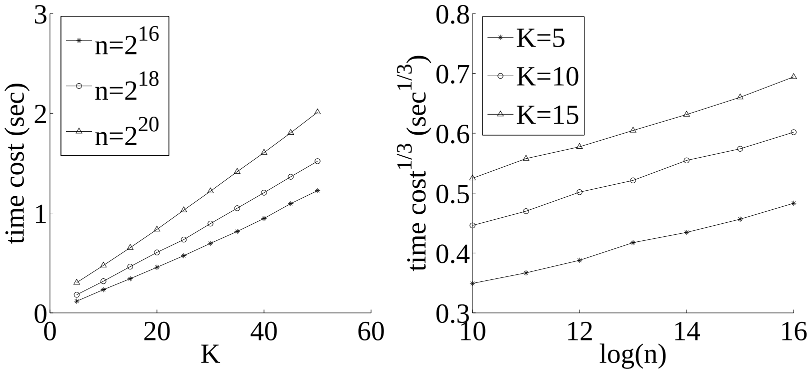


Fig. 17: **Decoding complexity.** We choose $Q = 2\log^2(n)$, SNR $= 20$dB, $L_m = 3$, and $L_p = 6$. Different values of n and K are tested, and for each set of parameters, 100 experiments are conducted and the average time cost is shown.

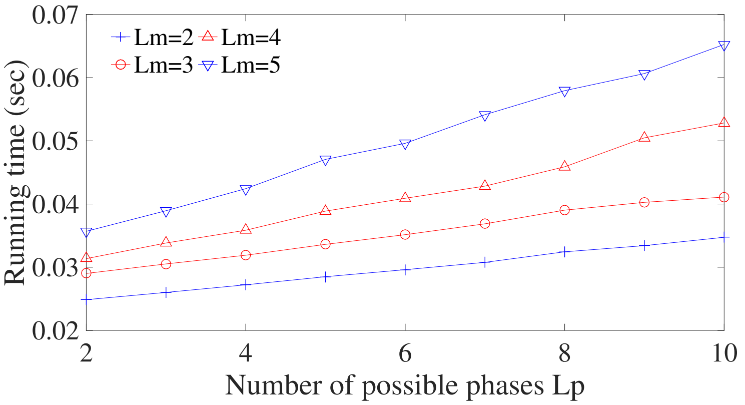


Fig. 18: **Decoding complexity vs. number of possible magnitudes and phases.** We choose $n = 4096$, $K = 10$, $Q = 5\log^2(n)$ and SNR $= 24$dB. Different values of L_m and L_p are tested, and for each set of parameters, 100 experiments are conducted and the average time cost is shown.

and use Gaussian noise. Since the fraction of non-recovered non-zero components p can be made arbitrarily small, in the simulations, we simply define a successful recovery as the cases when *all* the non-zero components are correctly recovered up to a global phase. In all the simulations, we set $P = 5\log(n)$, $d = 15$, $M = 8K$, and $\varepsilon = 1$.

In Figure 16, we show the simulation results on the probability of successful recovery as a function of the number of measurements and the SNR. In Figure 17, we show the simulation results on the decoding complexity of the sublinear scheme.[12] It can be seen that the time cost of sublinear scheme is indeed low and only linear in K and $\Theta(\log^3(n))$. In Figure 18, we show empirical results on the decoding complexity of the sublinear scheme as a function of the number of possible magnitudes and phases (L_m and L_p). One can observe that the time cost grows linearly in L_m and L_p.

[12] The simulations are conducted on a laptop with 2.8 GHz Intel Core i7 CPU and 16 GB memory.

VIII. Conclusion

We have considered the problem of recovering a K-sparse complex signal $\boldsymbol{x} \in \mathbb{C}^n$ from m intensity measurements of the form $|\boldsymbol{A}\boldsymbol{x}|$, where $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ is the measurement matrix. Our main focus was on the case where the measurement vectors are unconstrained and noiseless. We proposed the PhaseCode algorithm that is based on a sparse-graph codes framework. We showed that for any signal $\boldsymbol{x} \in \mathbb{C}^n$, using order-optimal sample and decoding complexity of $\Theta(K)$, PhaseCode can provably recover all but an arbitrarily small random fraction of the non-zero signal components with high probability. We also showed that PhaseCode can recover almost all the K non-zero signal components using only slightly more than $4K$ measurements if the support of the non-zero components of signal is uniformly random. To the best of our knowledge, our work is the first capacity-approaching low-complexity compressive phase retrieval algorithm. We furthermore showed that PhaseCode can be used for practical systems such as optical systems with proper modifications. Finally, we demonstrated how PhaseCode can be robustified in the presence of noise. Via extensive simulation results, we validated the performance of PhaseCode for various settings.

References

[1] M. Akcakaya and V. Tarokh, "New conditions for sparse phase retrieval," *arXiv preprint arXiv:1310.1351*, 2013.

[2] T. Heinosaari, L. Mazzarella, and M. M. Wolf, "Quantum tomography under prior information," *Communication in Mathematical Physics*, vol. 318, no. 2, pp. 355–374, 2013.

[3] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[4] D. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, 2006.

[5] A. Walther, "The question of phase retrieval in optics," *Optica Acta*, vol. 10, no. 1, pp. 41–49, 1963.

[6] R. P. Milane, "Phase retrieval in crystallography and optics," *J. Opt. Soc. Am. A*, vol. 7, pp. 394–411, 1990.

[7] R. W., "Harrison "phase problem in crystallography," *JOSA A*, vol. 10, pp. 1046–1055, 1993.

[8] J. C. Dainty and J. R. Fienup, "Phase retrieval and image reconstruction for astronomy," in *Image Recovery: Theory and Application*, pp. 231–275, Academic Press, 1987.

[9] J. M. Rodenburg, "Ptychography and related diffractive imaging methods," *Advances in Imaging and Electron Physics*, vol. 150, pp. 87–184, 2008.

[10] M. Mirhosseini, O. S. Magana-Loaiza, S. M. H. Rafsanjani, and R. W. Boyd, "Compressive direct measurement of the quantum wavefunction," *arXiv preprint arXiv:1404.2680*, 2014.

[11] M. H. Hayes, J. S. Lim, and A. V. Oppenheim, "Signal reconstruction from phase or magnitude," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 28, no. 6, pp. 672–680, 1980.

[12] M. L. Moravec, J. K. Romberg, and R. Baraniuk, "Compressive phase retrieval," *SPIE Conf. Series*, vol. 6701, 2007.

[13] P. Schniter and S. Rangan, "Compressive phase retrieval via generalized approximate message passing," in *Proceedings of Allerton Conference on Communication, Control, and Computing*, 2012.

[14] H. Ohlsson, A. Yang, R. Dong, and S. Sastry, "Compressive phase retrieval from squared output mea- surements via semidefinite programming," *arXiv preprint arXiv:1111.6323*, 2011.

[15] E. J. Candes, T. Strohmer, and V. Voroninski, "Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming," *Communications on Pure and Applied Mathematics*, vol. 66, no. 8, pp. 1241–1274, 2013.

[16] P. Netrapalli, P. Jain, and S. Sanghavi, "Phase retrieval using alternating minimization," *arXiv preprints arXiv:1306.0160*, 2013.

[17] X. Li and V. Voroninski, "Sparse signal recovery from quadratic measurements via convex programming," *arXiv preprints arXiv:1209.4785*, 2012.

[18] K. Jaganathan, S. Oymak, and B. Hassibi, "Sparse phase retrieval: Convex algorithms and limitations," pp. 1022–1026, 2013.

[19] K. Jaganathan, S. Oymak, and B. Hassibi, "Phase retrieval for sparse signals using rank minimization," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3449–3452, 2012.

[20] E. Candes, X. Li, and M. Soltanolkotabi, "Phase retrieval via wirtinger flow: Theory and algorithms," *arXiv preprint arXiv:1407.1065*, 2014.

[21] E. G. Loewen and E. Popov, *Diffraction gratings and applications*. CRC Press, 1997.

[22] O. Bunk, A. Diaz, F. Pfeiffer, C. David, B. Schmitt, D. K. Satapathy, and J. F. Veen, "Diffractive imaging for periodic samples: retrieving one-dimensional concentration profiles across microfluidic channels," *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 63, no. 4, pp. 306–314, 2007.

[23] E. J. Candes, X. Li, and M. Soltanolkotabi, "Phase retrieval from coded diffraction patterns," *arXiv preprint arXiv:1310.3240*, 2013.

[24] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2008.

[25] S. Cai, M. Bakshi, S. Jaggi, and M. Chen, "Super: Sparse signals with unknown phases efficiently recovered," *arXiv preprint arXiv:1401.4451*, 2014.

[26] S. Pawar and K. Ramchandran, "Computing a k-sparse n-length discrete fourier transform using at most $4k$ samples and $\mathcal{O}(k \log k)$ complexity," *arXiv preprint arXiv:1305.0870*, 2013.

[27] I. Waldspurger, A. d'Aspremont, and S. Mallat, "Phase recovery, maxcut and complex semidenite programming," *Mathematical Programming, pp.*, pp. 1–35, 2013.

[28] R. Balan, P. G. Casazza, and D. Edidin, "On signal reconstruction without phase," *Applied and Computational Harmonic Analysis*, vol. 20, May 2009.

[29] A. S. Bandeira, J. Cahill, D. G. Mixon, and A. A. Nelson, "Fundamental limits of phase retrieval," *Proc. 10th Intern. Conf. on Sampling Theory and Applications (SampTA)*, July 2013.

[30] B. G. Bodmann and N. Hammen, "Stable phase retrieval with low-redundancy frames," *arXiv preprint arXiv:1302.5487*, 2013.

[31] Y. Wang and Z. Xu, "Phase retrieval for sparse signals," *Applied and Computational Harmonic Analysis*, vol. 37, no. 3, pp. 531–544, 2014.

[32] M. Akçakaya and V. Tarokh, "Sparse signal recovery from a mixture of linear and magnitude-only measurements," *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1220–1223, 2015.

[33] A. S. Bandeira and D. G. Mixon, "Near-optimal phase retrieval of sparse vectors," in *SPIE Optical Engineering+ Applications*, pp. 88581O–88581O, International Society for Optics and Photonics, 2013.

[34] W. Xu and B. Hassibi, "Efficient compressive sensing with deterministic guarantees using expander graphs," in *Information Theory Workshop, 2007. ITW'07. IEEE*, pp. 414–419, IEEE, 2007.

[35] M. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. Spielman, "Improved low-density parity check codes using irregular graphs," *IEEE Trans. Info. Theory*, vol. 47, pp. 585–598, 2001.

[36] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Transactions on Information Theory*, vol. 47, pp. 599–618, February 2001.

[37] Z. Wang, L. Millet, M. Mir, H. Ding, S. Unarunotai, J. Rogers, M. U. Gillette, and G. Popescu, "Spatial light interference microscopy (slim)," *Opt. Express*, vol. 19, no. 2, pp. 1016–1026, 2011.

[38] S. R. P. Pavani and R. Piestun, “Three dimensional tracking of fluorescent microparticles using a photon-limited double-helix response system,” *Opt. Express*, vol. 16, pp. 22048–22057, 2008.

[39] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*. Prentice Hall, 1989.

[40] R. Vershynin, “Introduction to the non-asymptotic analysis of random matrices,” *arXiv preprint arXiv:1011.3027*, 2010.

[41] B. Alexeev, A. S. Bandeira, M. Fickus, and D. G. Mixon, “Phase retrieval with polarization,” *SIAM Journal on Imaging Sciences*, vol. 7, no. 1, pp. 35–66, 2014.

[42] D. J. Love, R. W. Heath, W. Santipach, and M. L. Honig, “What is the value of limited feedback for mimo channels?,” *IEEE Communications Magazine*, vol. 42, no. 10, pp. 54–59, 2004.

[43] J. Candy, “A use of limit cycle oscillations to obtain robust analog-to-digital converters,” *IEEE Transactions on Communications*, vol. 22, no. 3, pp. 298–305, 1974.

[44] Y. Chen, X. Yi, and C. Caramanis, “A convex formulation for mixed regression with two components: Minimax optimal rates.,” in *COLT*, pp. 560–604, 2014.

[45] A. C. Gilbert, M. J. Strauss, J. A. Tropp, and R. Vershynin, “Algorithmic linear dimension reduction in the l_1 norm for sparse vectors,” *arXiv preprint cs/0608079*, 2006.

[46] P. Erdos and A. Renyi, “On the evolution of random graphs,” *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, vol. 5, pp. 17–61, 1960.

[47] B. Bollobas, *Random graphs*. Cambridge University Press, 2001.

[48] S. A. Pawar, *Pulse: Peeling-based ultra-low complexity algorithms for sparse signal estimation*. PhD thesis, University of California, Berkeley, 2013.

[49] M. Rudelson, R. Vershynin, *et al.*, “Hanson-wright inequality and sub-gaussian concentration,” *Electron. Commun. Probab*, vol. 18, no. 82, pp. 1–9, 2013.

Appendix

A. Guess and Check Strategy for Resolvable Multitons

Recall the equations:

$$\begin{aligned}
y_{i,1} &= |a + e^{\mathbf{i}\omega\ell} x_\ell| = |u|, && (28)\\
y_{i,2} &= |b + e^{-\mathbf{i}\omega\ell} x_\ell| = |v|, && (29)\\
y_{i,3} &= |c + 2\cos(\omega\ell) x_\ell| = |w|, && (30)\\
y_{i,4} &= |d + e^{\mathbf{i}\omega'\ell} x_\ell|, && (31)
\end{aligned}$$

where complex numbers a, b, c and d are known values that depend on the values and locations of the known colored active left nodes. We want to solve the first 3 equations (28)-(30) to find ℓ and x_ℓ, and use (31) to check if our guess is correct. Since $e^{\mathbf{i}\omega\ell} + e^{-\mathbf{i}\omega\ell} = 2\cos(\omega\ell)$, we know that $u + v = w$. Let α be the angle between complex numbers u and v. Then,

$$|u + v|^2 = |u|^2 + |v|^2 + 2|u||v|\cos(\alpha).$$

Thus, one can find α up to a plus-minus sign as,

$$\begin{aligned}
\alpha &= \cos^{-1}\left(\frac{|u+v|^2 - |u|^2 - |v|^2}{2|u||v|}\right)\\
&= \cos^{-1}\left(\frac{y_{i,3}^2 - y_{i,1}^2 - y_{i,2}^2}{2y_{i,1}y_{i,2}}\right).
\end{aligned}$$

We find possible x_ℓ's for two different signs of α. If our guess is true, the check measurement $y_{i,4}$ will determine which solution is the right one. Define a known variable z as

$$z = u/v = \frac{|u|}{|v|} e^{\mathbf{i}\omega\alpha}.$$

Thus,

$$a + e^{\mathbf{i}\omega\ell} x = z(b + e^{-\mathbf{i}\omega\ell} x),$$

or

$$x = \frac{zb - a}{e^{\mathbf{i}\omega\ell} - ze^{-\mathbf{i}\omega\ell}}. \tag{32}$$

Replacing x from (30) in (32), we have

$$\begin{aligned}
y_{i,3} &= |c + 2\cos(\omega\ell)\frac{zb - a}{e^{\mathbf{i}\omega\ell} - ze^{-\mathbf{i}\omega\ell}}|\\
&= |c\frac{\cos(\omega\ell)(1 - z + \frac{2zb-2a}{c}) + \mathbf{i}\sin(\omega\ell)(1+z)}{\cos(\omega\ell)(1-z) + \mathbf{i}\sin(\omega\ell)(1+z)}|. \qquad (33)
\end{aligned}$$

Define the following known complex variables:

$$\begin{aligned} k_1 &= 1 - z + \frac{2zb - 2a}{c}; \\ k_2 &= 1 + z; \\ k_3 &= 1 - z; \\ k_4 &= y_{i,3}/|c|. \end{aligned}$$

Also let $k_1 = k_{1r} + \mathbf{i}k_{1i}$ and use similar notation for the real and imaginary parts of other variables. Then, one can square (33) to get

$$\begin{aligned} &(k_{1r}\cos(\omega\ell) - k_{2i}\sin(\omega\ell))^2 + (k_{1i}\cos(\omega\ell) + k_{2r}\sin(\omega\ell))^2 \\ &\quad = k_4^2[(k_{3r}\cos(\omega\ell) - k_{2i}\sin(\omega\ell))^2 \\ &\qquad + (k_{3i}\cos(\omega\ell) + k_{2r}\sin(\omega\ell))^2]. \end{aligned}$$

Now defining appropriate new known real variables k_5, k_6 and k_7, we get an equation of the form

$$k_5\cos^2(\omega\ell) + k_6\sin^2(\omega\ell) = k_7\sin(\omega\ell)\cos(\omega\ell).$$

Squaring the above equation and using $\sin^2(\omega\ell) = 1 - \cos^2(\omega\ell)$, we get a quadratic equation in $\cos^2(\omega\ell)$ that one can easily solve to find at most 2 possible values for ℓ. Note that $\cos(\omega\ell)$ is positive by construction. Now since there are two possible values of α, one can get at most 4 solutions for ℓ and x_ℓ. Those solutions can be checked by (31). If the guess is true, the probability that the check fails is 0; thus, one can recover the resolvable multiton with probability 1.

B. Proof of Corollary 4

Let $(|x_{(1)}|, |x_{(2)}|, \dots, |x_{(K)}|)$ be the magnitudes of the non-zero components that are ordered increasingly. We partition the K components to $g = \lfloor K^{(1+\gamma)/2} \rfloor$ subgroups as follows:

$$(|x_{(1)}|, \dots, |x_{(K/g)}|), (|x_{(K/g+1)}|, \dots, |x_{(2K/g)}|), \dots, (|x_{(K-K/g+1)}|, \dots, |x_{(K)}|).$$

Let b_i be the largest number in subgroup i. By Azuma-Hoeffding's inequality, the probability that more than $(p+\epsilon)K/g$ components are missed in a subgroup is upper bounded by $2e^{-2\epsilon^2 K/g}$. Taking $\epsilon = 1/\log(K)$ and using union bound, we have

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_1 \le (p + 1/\log(K))(\sum_{i=1}^{g} b_i)K/g, \tag{34}$$

with probability $\mathcal{O}(ge^{-\frac{2K}{g\log^2(K)}})$. Further,

$$\begin{aligned} (\sum_{i=1}^{g} b_i)K/g &\le (|x_{(1)}| + \sum_{i=1}^{g} b_i)K/g & (35) \\ &\le \|\boldsymbol{x}\|_1 + b_g K/g & (36) \\ &\le \|\boldsymbol{x}\|_1(1 + \Theta(\frac{K^\gamma}{g})) & (37) \\ &= \|\boldsymbol{x}\|_1(1 + \Theta(K^{-\frac{1-\gamma}{2}})). & (38) \end{aligned}$$

Gathering (34) and (38), we conclude that with probability $1 - \mathcal{O}(K^{\frac{1+\gamma}{2}} e^{-\frac{2K^{(1-\gamma)/2}}{\log^2(K)}})$,

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_1 \le p\|x\|_1(1 + \Theta(\frac{1}{\log(K)}) + \Theta(K^{-\frac{1-\gamma}{2}})) = p\|x\|_1(1 + \Theta(\frac{1}{\log(K)})). \tag{39}$$

C. Proof of Lemma 5

Proof: We form a graph with nodes that are active left nodes which are in singleton right nodes. We construct edges between these nodes if the corresponding active left nodes are connected to a strong doubleton, and we use an Erdos-Renyi random graph model [46] to find parameters d and M for which there is a giant component of size linear in K after the second step of the algorithm. The Erdos-Renyi random graph model is characterized by 2 parameters: n, the number of nodes in the graph and p which is the probability that each of the $\binom{n}{2}$ possible edges are connected. Note that each edge is connected in the graph with probability p independently from every other edge. There is another variant of Erdos-Renyi random graph model which is parametrized by (n, M), where M is the total number of edges. Then, the graph is chosen uniformly at random from the collection of all graphs with n nodes and M edges. By the law of large numbers, the two models are equivalent for

$M = \binom{n}{2}p$ as long as $n^2 p \to \infty$. It is well known that in an Erdos-Renyi model if $np \to c > 1$, as $n \to \infty$, where c is some constant, then the graph will have a unique giant component of size linear in n [46].

Define K_s to be the random variable representing the number of active left nodes that are connected to singletons. We form an Erdos-Renyi random graph model with parameters (K_s, p_s) or equivalently parameters (K_s, M_s) where p_s is the probability that an edge is connected, and M_s is the total number of edges. Thus, as K_s gets large, M_s approaches $\binom{K_s}{2}p_s$. Now we compute the parameters K_s and p_s as follows. The probability of an active left node being connected to a singleton right node is the probability that at least one of its d neighbors is a singleton, that is:

$$q_s = 1 - (1 - \rho_1)^d. \tag{40}$$

Thus, by the law of large numbers as K gets large, there are $Kq_s + o(K)$ distinct active left nodes in singleton right nodes. Let $M = cK$ for some constant c. As K gets large, the number of doubleton right nodes approaches $M\frac{\eta^2 e^{-\eta}}{2!} + o(K)$ since the degree of right nodes (on the pruned graph with active left nodes) is Poisson distributed with parameter $\eta = Kd/M = d/c$. However, we want to count only distinct doubleton right nodes. It is easy to see that as K gets large, essentially all but a vanishing fraction of the doubleton right nodes are distinct. To this end, fix a doubleton right node with neighbors (v_1, v_2). The probability that a randomly chosen doubleton right node is connected to (v_1, v_2) is $1/\binom{K}{2}$. Since the number of doubleton right nodes is linear in K, only a vanishing $\Theta(1/K)$ fraction of them are non-distinct.

Let M_s be the number of strong doubletons (for which both left nodes are also in other singletons). Thus, M_s is the number of edges in our constructed Erdos-Renyi graph. Consider a random left node i. Let D be the event that i is connected to a doubleton right node and S be the event that i is connected to a singleton right node. We compute the following 2 relevant conditional probabilities:

$$\begin{aligned} p_1 \triangleq \mathbb{P}(D|S) &= \frac{\mathbb{P}(D \cap S)}{\mathbb{P}(S)} \\ &= \frac{1 - \mathbb{P}(\bar{S}) - \mathbb{P}(\bar{D}) + \mathbb{P}(\bar{S} \cap \bar{D})}{1 - \mathbb{P}(\bar{S})} \\ &= \frac{1 - (1-\rho_1)^d - (1-\rho_2)^d + (1-\rho_1-\rho_2)^d}{1-(1-\rho_1)^d}. \end{aligned}$$

$$\begin{aligned} p_2 \triangleq \mathbb{P}(D|\bar{S}) &= 1 - \mathbb{P}(\bar{D}|\bar{S}) \\ &= 1 - \frac{\mathbb{P}(\bar{S} \cap \bar{D})}{\mathbb{P}(\bar{S})} \\ &= 1 - \frac{(1-\rho_1-\rho_2)^d}{(1-\rho_1)^d}. \end{aligned}$$

Now we use Bayes' rule to find that

$$\begin{aligned} q \triangleq \mathbb{P}(S|D) &= \frac{\mathbb{P}(D|S)\mathbb{P}(S)}{\mathbb{P}(D|S)\mathbb{P}(S) + \mathbb{P}(D|\bar{S})\mathbb{P}(\bar{S})} \\ &= \frac{p_1 q_s}{p_1 q_s + p_2(1-q_s)}. \end{aligned}$$

Thus,

$$M_s = M\frac{\eta^2 e^{-\eta}}{2!}q^2. \tag{41}$$

The random graph is constructed with $K_s = K(1-(1-\rho_1)^d)$ nodes and M_s edges chosen uniformly at random among $\binom{K_s}{2}$ possible edges. The probability of a randomly chosen edge being connected is thus:

$$p_s = \frac{M\frac{\eta^2 e^{-\eta}}{2!}q^2}{\binom{K_s}{2}}.$$

From the well-known Erdos-Renyi random graph result [46] (also see [47]), a linear size giant component exists if $K_s p_s > 1$ with probability $1 - \mathcal{O}(1/K_s)$. More precisely, let Z be the size of the giant component. Then, one has

$$\mathbb{P}\left(|\frac{Z}{K_s} - \zeta| < \varepsilon\right) = 1 - \mathcal{O}\left(\frac{1}{\varepsilon^2 K_s}\right),$$

where $\zeta \in (0,1)$ is the unique solution of $\zeta + e^{-2\zeta M_s/K_s} = 1$, if $2M_s/K_s > 1$ or equivalently $K_s p_s > 1$ [25], [47]. Thus, a linear-size giant component exists if

$$\frac{Kq_s M_s}{\binom{Kq_s}{2}} > 1.$$

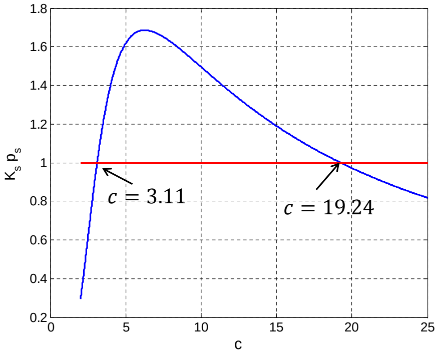
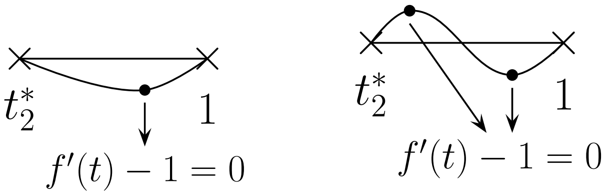


Fig. 19: The diagram shows the values of c for which the giant component is formed after step 2 of the algorithm. Note that $c = M/K$. In the random graph model the giant component is formed if $K_s p_s > 1$, where K_s is the number of nodes in the random graph, and p_s is the probability that an edge is connected. From the diagram, one can see that if $3.11 < c < 19.24$, the condition for having a giant component is satisfied.



(a) The good case.

(b) The bad case.

Fig. 20: Figure (a) illustrates the good case that there are no fixed points other than 1 and t_2^*. Figure (b) illustrates the bad case that there is another fixed point in the interval $(t_2^*, 1)$. In this case, $f'(t) = 1$ has two solutions for $t \in (t_2^*, 1)$, as shown in Figure (b).

We present two concrete examples to complete the proof of the lemma. Let $d = 5$. Replacing M_s and q_s by (41) and (40), one can check that the inequality holds if $3.11 \leq c \leq 19.24$ (See Figure 19). Similarly, one can set $d = 8$ and see that the inequality holds if $3.48 \leq c \leq 55.36$. ■

D. *Proof of Lemma 7*

First, let us consider a small neighborhood around $t_1^* = 1$. We want

$$f(t_1^* - h) < t_1^* - h = f(t_1^*) - h,$$

for some small $h > 0$. Equivalently, we want

$$\frac{f(t_1^*) - f(t_1^* - h)}{h} > 1.$$

Letting $h \to 0$, the condition becomes $f'(t)|_{t=1} > 1$. This is a necessary and sufficient condition for instability of point $t = 1$. In other words, this condition makes sure that (15) holds for p_j close to 1. Thus, in picking parameters d and η, one makes sure that

$$f'(t)|_{t=1} = (d-1)\eta e^{-\eta} > 1.$$

For $d = 5$, this leads to $0.3574 < \eta < 2.1533$ or $2.32K < M < 13.99K$. For $d = 8$, this leads to $0.17 < \eta < 3.06$ or $2.62K < M < 47.06K$. To complete the proof, we need to show that $f(t) - t < 0$ for $t \in (t_2^*, 1)$. Note that $f(t)$ is continuous and continuously differentiable. Thus to show that $f(t) - t < 0$ for $t \in (t_2^*, 1)$, it is enough to show that $f'(t) - 1 = 0$ has only one solution in that interval (the "good" case: See Figure 20a). To see this, suppose that $f(t) - t = 0$ for some t in the interval $(t_2^*, 1)$. Since 1 and t_2^* are also solutions of $f(t) - t = 0$, then $f'(t) - 1$ must change sign at least twice in the interval $(t_2^*, 1)$ (the "bad" case: See Figure 20b). Therefore, to ensure that $f(t) < t, \ \forall t \in (t_2^*, 1)$ it is sufficient to show that

$$f'(t) = \eta e^{-\eta t}(d-1)(1 + e^{-\eta} - e^{-\eta t})^{d-2} = 1,$$

has only one solution in the interval $t \in (t_2^*, 1)$. After some algebra, one can re-write the above equation as

$$(\eta(d-1))^{-\frac{1}{d-2}} e^{\eta t(1/(d-2)+1)} = e^{\eta t}(1 + e^{-\eta}) - 1.$$

Replacing $x = e^{\eta t}$, we get an equation of the form $x^a = bx - c$ for $a > 1$ and $b, c > 0$. This equation has clearly at most two solutions for $x \geq 0$. On the other hand, $f'(1) > 1$ and $f'(\infty) = 0$. Thus, $f'(1) = 1$ has a solution for $t > 1$, which shows that $f'(t) = 1$ has at most one solution in $t \in [0, 1]$.

E. Probability of Tree-like Neighborhood

In this section, we give a short proof of Lemma 9. Let C_ℓ be the number of right-nodes and V_ℓ be the number of left-nodes in $\mathcal{N}_{\vec{e}}^{2\ell}$. Since the ensemble of the graphs that we consider is only left-regular (and not right-regular), we cannot immediately use the result of [36]. Note that the degree distribution of right nodes is Poisson distribution with constant rate. The key idea is to show that the size of the tree is bounded by $\mathcal{O}(\log(K)^\ell)$ with high probability. This is intuitively clear since Poisson distribution has a tail decaying faster than exponential decay. To formally show this, we keep unfolding the tree up to level ℓ^*, and at each level ℓ we upper bound the probability that the size of the tree grows larger than $\mathcal{O}(\log(K)^\ell)$. Fix some constant c_1. We upper bound the probability of not having a tree as follows.

$$\begin{aligned}\mathbb{P}(\mathcal{N}_{\vec{e}}^{2\ell^*} \text{ is not a tree}) &\leq \mathbb{P}(V_{\ell^*} > c_1 \log(K)^{\ell^*}) \\ &\quad + \mathbb{P}(C_{\ell^*} > c_1 \log(K)^{\ell^*}) \\ &\quad + \mathbb{P}(\mathcal{N}_{\vec{e}}^{2\ell^*} \text{ is not a tree} | V_{\ell^*} < c_1 \log(K)^{\ell^*},\ C_{\ell^*} < c_1 \log(K)^{\ell^*}).\end{aligned}$$

Note that since the left degree is a constant, d, if V_{ℓ^*} is $\mathcal{O}(\log(K)^{\ell^*})$, C_{ℓ^*} is also $\mathcal{O}(\log(K)^{\ell^*})$. Let $\alpha_\ell = \mathbb{P}(V_\ell > c_1 \log(K)^\ell)$. Then,

$$\alpha_\ell \leq \alpha_{\ell-1} + \mathbb{P}(V_\ell > c_1 \log(K)^\ell | V_{\ell-1} < c_1 \log(K)^{\ell-1}) \tag{42}$$

$$\leq \alpha_{\ell-1} + \mathbb{P}(V_\ell > c_1 \log(K)^\ell | C_\ell < c_2 \log(K)^{\ell-1}), \tag{43}$$

where (43) is due to the fact that every left node has exactly d edges connected to right nodes so if $V_{\ell-1} < c_1 \log(K)^{\ell-1}$, there exists some constant c_2 such that $C_\ell < c_2 \log(K)^{\ell-1}$. To count the number of left nodes in depth ℓ, let $n_\ell < C_\ell$ be the number of right nodes exactly at depth ℓ after unfolding the tree. Let $X_i,\ 1 \leq i \leq n_\ell$ be the degree of these right nodes. Given that $V_{\ell-1} < c_1 \log(K)^{\ell-1}$, one has $V_\ell > c_1 \log(K)^\ell$, only if $X = \sum_{i=1}^{n_\ell} X_i > c_3 \log(K)^\ell$ for some constant c_3. The distribution of X is Poisson distribution with parameter $n_\ell \lambda$. We know that the tail probability of a Poisson random variable Y with parameter λ can be upper bounded as follows: $\mathbb{P}(Y \geq y) \leq \left(\frac{e\lambda}{y}\right)^y$. Thus,

$$\mathbb{P}(X > c_3 \log(K)^\ell) \leq \left(\frac{c_4}{\log(K)}\right)^{c_3 \log(K)^\ell} \leq \mathcal{O}(\frac{1}{K}).$$

Thus,

$$\alpha_\ell \leq \alpha_{\ell-1} + \frac{c_5}{K}, \tag{44}$$

for some constant c_5. Now since ℓ^* is a constant, summing up the inequalities in (44), we show that

$$\alpha_{\ell^*} = \mathbb{P}(V_{\ell^*} > c_1 \log(K)^{\ell^*}) \leq \mathcal{O}(\frac{1}{K}).$$

Similarly, one can show that

$$\mathbb{P}(C_{\ell^*} > c_1 \log(K)^{\ell^*}) \leq \mathcal{O}(\frac{1}{K}).$$

To complete the proof, we need to show that with high probability, we have a tree-like neighborhood, given that the number of nodes is bounded by $\mathcal{O}(\log(K)^{\ell^*})$. First, we find a lower bound on the probability that $\mathcal{N}_{\vec{e}}^{2\ell+1}$ is a tree-like neighborhood if $\mathcal{N}_{\vec{e}}^{2\ell}$ is a tree-like neighborhood, when $\ell < \ell^*$. Assume that t additional edges have been revealed at this stage without forming a cycle. The probability that the next edge from a left node does not create a cycle is the probability that it is connected to one of the right nodes that is not already in the subgraph which is lower bounded by $1 - \frac{C_{\ell*}}{m}$. Thus, the probability that $\mathcal{N}_{\vec{e}}^{2\ell+1}$ is a tree-like neighborhood if $\mathcal{N}_{\vec{e}}^{2\ell}$ is a tree-like neighborhood, is lower-bounded by $(1 - \frac{C_{\ell*}}{M})^{C_{\ell+1} - C_\ell}$. Similarly, the probability that $\mathcal{N}_{\vec{e}}^{2\ell+2}$ is a tree-like neighborhood if $\mathcal{N}_{\vec{e}}^{2\ell+1}$ is a tree-like neighborhood, is lower-bounded by $(1 - \frac{V_{\ell*}}{K})^{V_{\ell+1} - V_\ell}$. Therefore, the probability that $\mathcal{N}_{\vec{e}}^{2\ell^*}$ is a tree-like neighborhood is lower-bounded by

$$(1 - \frac{V_{\ell*}}{K})^{V_{\ell*}} (1 - \frac{C_{\ell*}}{M})^{C_{\ell*}}.$$

For large M and K, the above expression is approximately

$$e^{-(V_{\ell*}^2/K + C_{\ell*}^2/M)} \geq 1 - (V_{\ell*}^2/K + C_{\ell*}^2/M).$$

Now since $V_{\ell*}$ and $C_{\ell*}$ are upper-bounded by $\mathcal{O}(\log(K)^{\ell^*})$, the probability of having a tree-like neighborhood is at least $1 - \mathcal{O}(\log(K)^{\ell^*}/K)$.

F. Convergence to Cycle-free Case

In this section, we give a short proof of Lemma 10. The proof follows similar steps as in [36], with the difference that the right degree is irregular and Poisson-distributed.

First, we prove (17). Let $Z_i = 1_{\{\vec{e}_i \text{ is colored}\}},\ 1 \le i \le Kd$ be the indicator that $\vec{e}_i$ is colored after ℓ iterations of the algorithm. Let B be the event that $\mathcal{N}_{\vec{e}_1}^{2\ell}$ is tree-like. Then,

$$\begin{aligned}\mathbb{E}[Z_1] &= \mathbb{E}[Z_1|B]\mathbb{P}(B) + \mathbb{E}[Z_1|\bar{B}]\mathbb{P}(\bar{B})\\ &\le \mathbb{E}[Z_1|B] + \mathbb{P}(\bar{B})\\ &\le p_\ell + \frac{\gamma \log(K)^\ell}{K},\end{aligned}$$

for some constant γ, where the last inequality is by Lemma 9. Trivially, $|\mathbb{E}[Z_1|B]| \le 1$. Furthermore, $\mathbb{E}[Z] = Kd\mathbb{E}[Z_1]$. Hence,

$$Kd(1 - \frac{\gamma \log(K)^\ell}{K}) < \mathbb{E}[Z] < Kd(p_\ell + \frac{\gamma \log(K)^\ell}{K}).$$

Then, (17) follows from choosing K large enough such that $\frac{K}{\log(K)^\ell} > \frac{2\gamma}{\epsilon}$.

Second, we prove that

$$\mathbb{P}(|Z - Kdp_\ell| > Kd\epsilon/2) < 2e^{-\beta\epsilon^2 K^{1/(2\ell+1)}}. \tag{45}$$

Then, (18) follows from (17) and (45). To prove (45), we use the standard Martingale argument and Azuma's inequality provided in [36] with some modifications to account for the right irregular degree. Suppose that we expose the Kd edges of the graph one at a time. Let $Y_i = \mathbb{E}[Z|e_1^i]$. By definition, $Y_0, Y_1, \ldots, Y_{Kd}$ is a Doob's martingale process, where $Y_0 = \mathbb{E}[Z]$ and $Y_{Kd} = Z$. To use Azuma's inequality, we find the appropriate upper bound: $|Y_{i+1} - Y_i| \le \alpha_i$. If the right degree is regular and equal to d_c, it is shown in [36] that α_i can be chosen as $8(d_v d_c)^\ell$. We show that when the right degree has Poisson distribution with constant rate, the degree of all of the right nodes can be upper bounded by $\mathcal{O}(K^{\frac{1}{2\ell+0.5}})$ with probability at least $c_6K(e^{-\beta_1 K^{\frac{1}{2\ell+0.5}}})$ for some constants c_6 and β_1. To show this, let X be a Poisson random variable with parameter λ and c_7 be some constant. Then,

$$\mathbb{P}(X > c_7 K^{\frac{1}{2\ell+0.5}}) \le \left(\frac{e\lambda}{c_7 K^{\frac{1}{2\ell+0.5}}}\right)^{c_7 K^{\frac{1}{2\ell+0.5}}} \le c_6(e^{-\beta_1 K^{\frac{1}{2\ell+0.5}}}).$$

Now considering $M = \Theta(K)$ right nodes and using union bound, one can see that the probability that all the right nodes have degree less than $\mathcal{O}(K^{\frac{1}{2\ell+0.5}})$ is at least $1 - \mathcal{O}(K(e^{-\beta_1 K^{\frac{1}{2\ell+0.5}}}))$. Let E be the event that at least one right node has degree larger than $c_6K(e^{-\beta_1 K^{\frac{1}{2\ell+0.5}}})$. Given that E has not happened, one can upper bound α_i^2 by $\mathcal{O}(K^{\frac{2\ell}{2\ell+0.5}})$. Then,

$$\begin{aligned}\mathbb{P}(|Z - Kdp_\ell| > Kd\epsilon/2) &\le \mathbb{P}(|Z - Kdp_\ell| > Kd\epsilon/2|\bar{E}) + \mathbb{P}(E)\\ &\le 2e^{-\frac{K^2d^2\epsilon^2/4}{2\sum_i \alpha_i^2}} + c_6K(e^{-\beta_1 K^{\frac{1}{2\ell+0.5}}})\\ &\le 2e^{-\beta\epsilon^2 K^{1/(4\ell+1)}}.\end{aligned}$$

G. Proof of Lemma 11

First note that it is easy to prove the lemma for specific parameters by plotting the function. See for example Figure 10a. To formally show it, note that $f(1) = 1$ is one solution of the fixed point equation, since $\lambda(1) = 1$. Also $f(0) = \lambda(e^{-\eta}) > 0$. Thus, by continuity of $f(x)$ and using the assumption that $f'(1) > 1$, there is another fixed point x_2^*. Now since $f'(1) > 1$, $f(x) < x$ for x close to 1. In order to show that $f(x) < x$ for all $x \in (x_2^*, 1)$, it is enough to show that $f'(x) - 1 = 0$ has only one solution in $x \in (0, 1)$. To this end, see that

$$f'(x) = \eta e^{-\eta x}\lambda'(1 + e^{-\eta} - e^{-\eta x}).$$

For ease of notation, let $y = 1 + e^{-\eta} - e^{-\eta x}$ and $y \in (e^{-\eta}, 1)$. Equivalently, we want to show that

$$C(1 + e^{-\eta} - y)(1 + y + y^2 + \ldots + y^{D-2}) = 1$$

has only one solution where $C = \eta/h(D-1)$. This is easy to see since D is large so $y \simeq \frac{1-C-Ce^{-\eta}}{1-C}$.

H. Proof of Lemma 12

We show that if

$$D = \max\{(\frac{e}{1-\epsilon})^{2/\epsilon}, (1+\frac{1}{p^*})^{1/(1-\epsilon)}\}, \tag{46}$$

then,

$$f'(1) = \eta e^{-\eta} \sum_{i\geq 1} \lambda_i (i-1) > 1, \tag{47}$$

and the error floor which is approximately $\lambda(e^{-\eta})$ is at most p^*; that is,

$$\sum_{i\geq 1} \lambda_i e^{-\eta(i-1)} \leq p^*. \tag{48}$$

This shows that in the density evolution equation, p_j converges to p^* as j goes to infinity. This is illustrated in Figure 11.

Recall that

$$\bar{d} = (\sum_{i=2}^{D} \frac{\lambda_i}{i})^{-1} = h(D-1)\frac{D}{D-1}.$$

Thus, since $M = K/(1-\epsilon)$,

$$\eta = \frac{K\bar{d}}{M} = h(D-1)\frac{D}{D-1}(1-\epsilon).$$

First, we show (48) in the following.

$$\begin{aligned} \sum_{i=2}^{D} \lambda_i e^{-\eta(i-1)} &= \frac{1}{h(D-1)} \sum_{i=2}^{D} \frac{1}{i-1} e^{-\eta(i-1)} \\ &\leq \frac{1}{h(D-1)} \sum_{i=1}^{\infty} e^{-\eta i} \\ &= \frac{e^{-\eta}}{h(D-1)(1-e^{-\eta})}. \end{aligned}$$

It is enough to show that $h(D-1)(e^{\eta}-1) \geq \frac{1}{p^*}$. We have

$$\begin{aligned} h(D-1)(e^{\eta}-1) &\geq e^{\eta} - 1 \\ &\geq e^{\log(D)\cdot\frac{D}{D-1}(1-\epsilon)} - 1 \\ &\geq D^{1-\epsilon} - 1 \\ &\geq \frac{1}{p^*}, \end{aligned}$$

where the last inequality is due to (46).

Second, we show that (47) is satisfied in the following.

$$\begin{aligned} \eta e^{-\eta} \sum_{i=2}^{D} \mu_i (i-1) &= \eta e^{-\eta} \frac{D-1}{h(D-1)} && (49) \\ &= D(1-\epsilon) e^{-h(D-1)\frac{D}{D-1}(1-\epsilon)} && (50) \\ &\geq D(1-\epsilon) e^{-(1+\log(D))\frac{D}{D-1}(1-\epsilon)} && (51) \\ &= \frac{1-\epsilon}{e} D^{\frac{\epsilon D-1}{D-1}} && (52) \\ &\geq \frac{1-\epsilon}{e} D^{\epsilon/2} && (53) \\ &\geq 1, && (54) \end{aligned}$$

where (53) is due to (46) since $D \geq (\frac{e}{1-\epsilon})^{2/\epsilon} \geq \frac{2}{\epsilon}$ implies that $\frac{\epsilon D-1}{D-1} \geq \frac{\epsilon}{2}$, and (54) is due to (46). This shows that $p_j,\ j \geq 1$ is a strictly decreasing sequence which is lower bounded by p^*. Thus, $p_j \to p^*$ as $j \to \infty$. This completes the proof.

I. Proof of Theorem 15

We first introduce some notation. Here, $\|\cdot\|_F$ denotes the Frobenius norm of a matrix, $\|\cdot\|$ denotes the operator norm of a matrix. For a sub-exponential random variable, $\|\cdot\|_{\psi_1}$ denotes the sub-exponential norm of it; for a sub-gaussian random variable, $\|\cdot\|_{\psi_2}$ denotes the sub-gaussian norm of it [40]. The notations c, c_i, C, and C_i represent absolute constants with positive value.

In our model, we also assume that the noise w_i satisfies $\mathbb{E}[|w_i|] = \mu$, $\mathbb{E}[w_i^2] = \sigma^2$, and $\|w_i\|_{\psi_1} = \nu$. Since the entries in $\boldsymbol{A}_0$ and $\boldsymbol{F}_0$ are bounded and thus sub-gaussian, we let $\eta = \|\, |a_{ij}|\, \|_{\psi_2}$ and $\eta_0 = \|\, |f_{0,ij}|\, \|_{\psi_2}$, where a_{ij} and $f_{0,ij}$ are entries of $\boldsymbol{A}_0$ and $\boldsymbol{F}_0$.

In order to prove Theorem 15, we need to prove Lemma 17 first. Here, we restate Lemma 17 with more details.

Lemma 19. *There exists $\zeta > 0$, determined by η, ν, and σ, such that when $\phi > \mu/\zeta$, for any $t_0 \in (\mu, \zeta\phi)$,*

$$\mathbb{P}\left\{\frac{1}{P}\|\boldsymbol{w}\|_1 \geq t_0\right\} = \mathcal{O}(1/n^2), \tag{55}$$

and

$$\mathbb{P}\left\{\frac{1}{P}\left\|\boldsymbol{y} - \mathcal{A}(\hat{\boldsymbol{z}}\hat{\boldsymbol{z}}^{\mathrm{H}})\right\|_1 < t_0\right\} = \mathcal{O}(1/n^2), \tag{56}$$

when $\hat{\boldsymbol{z}} \nsim \boldsymbol{z}$.

See the proof of Lemma 17 in Appendix J. Now we can analyze the failure probability of the almost-linear scheme. Recall that the bipartite graph is d-left-regular; thus, there are dn edges in the graph. In the first iteration, we need to check every edge and detect the singletons. Therefore, we need to do $\Theta(n)$ tests in the first iteration. Similarly, in the following iterations, we need to do at most $\Theta(n)$ tests. Since the number of iterations is a constant, we need to do $N_t = \Theta(n)$ tests. Lemma 17 tells us that, for any energy test, if no error has been made in the previous tests, the error probability of the energy test is $\mathcal{O}(1/n^2)$. More specifically, let E_i be the event that there is an error in the ith test, while the tests $1, \ldots, i-1$ are all correct. The event E_{test} that there exists an error in at least one energy test can be decomposed as

$$E_{\text{test}} = \bigcup_{i=1}^{N_t} E_i.$$

By union bound, we have

$$\mathbb{P}\{E_{\text{test}}\} \leq N_t \sum_{i=1}^{N_t} \mathbb{P}\{E_i\} = \Theta(n)\mathcal{O}(1/n^2) = \mathcal{O}(1/n).$$

Another possibility of making an error lies in the coloring algorithm itself. When there is no error in energy tests, this probability is $\mathcal{O}(1/K)$ as analyzed in the noiseless case. Therefore the failure probability of the almost-linear scheme is

$$\begin{aligned}
\mathbb{P}\{E_a\} &= \mathbb{P}\{E_a|E_{\text{test}}\}\mathbb{P}\{E_{\text{test}}\} + \mathbb{P}\{E_a|E_{\text{test}}^{\mathsf{C}}\}\mathbb{P}\{E_{\text{test}}^{\mathsf{C}}\} \\
&\leq \mathbb{P}\{E_{\text{test}}\} + \mathbb{P}\{E_a|E_{\text{test}}^{\mathsf{C}}\} \\
&= \mathbb{P}\{E_{\text{test}}\} + \mathbb{P}\{E_{\text{coloring}}\} \\
&= \mathcal{O}(1/n) + \mathcal{O}(1/K) \\
&= \mathcal{O}(1/K)
\end{aligned}$$

The sample and computational complexity are already analyzed in Section VII-A. This completes the proof of Theorem 15.

J. Proof of Lemma 19

To prove Equation (55), we use the Bernstein's inequality in [40] as follows. For any $t > 0$,

$$\mathbb{P}\left\{\frac{1}{P}\sum_{i=1}^{P}(|w_i| - \mathbb{E}[|w_i|]) > t\right\} \leq \exp\left[-C_1 P \min\left\{\frac{t^2}{\nu^2}, \frac{t}{\nu}\right\}\right].$$

Therefore, by choosing $t_0 > \mathbb{E}[|w_i|] = \mu$ and $t = t_0 - \mu$, we have

$$\mathbb{P}\left\{\frac{1}{P}\|\boldsymbol{w}\|_1 \geq t_0\right\} \leq \exp[-\delta_1 P].$$

Since δ_1 is a constant and $P = \Theta(\log(n))$, (55) is proved.

Now we prove Equation (56). Before getting into the details of the proof, we give the definition of a new notation ϕ. For two vectors $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{S}^n$, it is easy to see that $\boldsymbol{p} \nsim \boldsymbol{q} \Leftrightarrow \boldsymbol{p}\boldsymbol{p}^{\mathrm{H}} - \boldsymbol{q}\boldsymbol{q}^{\mathrm{H}} \neq 0$. Since the entries of $\boldsymbol{p}$ and $\boldsymbol{q}$ lie in the quantized set $\mathbb{S}$, we know that there exists $\phi > 0$, such that $\|\boldsymbol{p}\boldsymbol{p}^{\mathrm{H}} - \boldsymbol{q}\boldsymbol{q}^{\mathrm{H}}\|_F > \phi$, when $\boldsymbol{p} \nsim \boldsymbol{q}$, where ϕ depends on ε, L_m, and L_p.

Lemma 20. *Given two vectors $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{C}^N$, let $\boldsymbol{X} = \boldsymbol{x}_1\boldsymbol{x}_1^{\mathrm{H}} - \boldsymbol{x}_2\boldsymbol{x}_2^{\mathrm{H}} \neq 0$. $\mathcal{A}$ is the linear function defined in (24), and $\boldsymbol{w}$ is the noise. Then, for any $s > 0$, we have,*

$$\mathbb{P}\left\{\frac{1}{P}\left\|\mathcal{A}(\boldsymbol{X}) + \boldsymbol{w}\right\|_1 < (\zeta - s\eta_d)\left\|\boldsymbol{X}\right\|_F - 2s\nu\right\} \leq \exp\left[-C_0 P \min\left\{s^2, s\right\}\right],$$

where $\zeta > 0$ depends on η, σ, and ν, $\eta_d > 0$ only depends on η.

See the proof of Lemma 20 in Appendix K. Note that $\boldsymbol{y} - \mathcal{A}(\hat{\boldsymbol{z}}\hat{\boldsymbol{z}}^{\mathrm{H}}) = \mathcal{A}(\boldsymbol{z}\boldsymbol{z}^{\mathrm{H}} - \hat{\boldsymbol{z}}\hat{\boldsymbol{z}}^{\mathrm{H}}) + \boldsymbol{w}$, and that $\|\boldsymbol{z}\boldsymbol{z}^{\mathrm{H}} - \hat{\boldsymbol{z}}\hat{\boldsymbol{z}}^{\mathrm{H}}\|_F > \phi$. Now using Lemma 20, conditioning on $\boldsymbol{h}$, we have for any $s > 0$,

$$\mathbb{P}\left\{\frac{1}{P}\left\|\boldsymbol{y} - \mathcal{A}(\hat{\boldsymbol{z}}\hat{\boldsymbol{z}}^{\mathrm{H}})\right\|_1 < \zeta\phi - (\eta_d\phi + 2\nu)s \mid \boldsymbol{h}\right\} \leq \exp\left[-C_0 P \min\left\{s^2, s\right\}\right]. \tag{57}$$

Since (57) holds for any $\boldsymbol{h}$, we know that it also holds without conditioning on $\boldsymbol{h}$. If $\zeta\phi > t_0$, we can choose $s = \frac{\zeta\phi - t_0}{\eta_d\phi + 2\nu}$. Then

$$\mathbb{P}\left\{\frac{1}{P}\left\|\boldsymbol{y} - \mathcal{A}(\hat{\boldsymbol{z}}\hat{\boldsymbol{z}}^{\mathrm{H}})\right\|_1 < t_0\right\} \leq \exp\left[-\delta_2 P\right].$$

Since δ_2 is a constant and $P = \Theta(\log(n))$, Equation (56) is proved.

We conclude that there exists ζ, determined by the statistics of noise, such that when $\phi > \mu/\zeta$, for any threshold $t_0 \in (\mu, \zeta\phi)$, the energy test fails with probability $\mathcal{O}(1/n^2)$. This completes the proof of Lemma 19.

K. Proof of Lemma 20

The proof of Lemma 20 is based on similar ideas in [44]. Let $\boldsymbol{\xi} = \mathcal{A}(\boldsymbol{X}) + \boldsymbol{w}$, then $\xi_i = \boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\boldsymbol{a}_i + w_i$. By the definition of matrix $\boldsymbol{A}$, we know that the Hanson-Wright inequality for complex random variables (shown in Appendix N) holds for $\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\boldsymbol{a}_i$. That is, for every $t > 0$,

$$\begin{aligned}\mathbb{P}\left\{\left|\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\boldsymbol{a}_i - \mathbb{E}\left[\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\boldsymbol{a}_i\right]\right| > t\right\} \leq & 6\exp\left[-c\min\left\{\frac{t^2}{\eta^4\left\|\boldsymbol{X}\right\|_F^2}, \frac{t}{\eta^2\left\|\boldsymbol{X}\right\|}\right\}\right] \\ \leq & 6\exp\left[-c\min\left\{\frac{t}{\eta^2\left\|\boldsymbol{X}\right\|_F} - \frac{1}{4}, \frac{t}{\eta^2\left\|\boldsymbol{X}\right\|_F}\right\}\right] \\ \leq & 6\exp\left[c\left(\frac{1}{4} - \frac{t}{\eta^2\left\|\boldsymbol{X}\right\|_F}\right)\right],\end{aligned}$$

where the second inequality is due to the fact that $(a - 1/2)^2 \geq 0$ and $\|\boldsymbol{X}\| \leq \|\boldsymbol{X}\|_F$. From [40], we know that $\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\boldsymbol{a}_i - \mathbb{E}\left[\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\boldsymbol{a}_i\right]$ is a sub-exponential random variable with sub-exponential norm

$$\left\|\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\boldsymbol{a}_i - \mathbb{E}\left[\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\boldsymbol{a}_i\right]\right\|_{\psi_1} \leq C\eta^2\left\|\boldsymbol{X}\right\|_F. \tag{58}$$

On the other hand,

$$\left|\mathbb{E}\left[\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\boldsymbol{a}_i\right]\right| = \frac{1}{2}\left|\left\|\boldsymbol{x}_1\right\|_2^2 - \left\|\boldsymbol{x}_2\right\|_2^2\right| \leq \frac{1}{2}\left\|\boldsymbol{X}\right\|_F.$$

Thus,

$$\begin{aligned}\left\|\xi_i\right\|_{\psi_1} = & \left\|\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\boldsymbol{a}_i - \mathbb{E}\left[\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\boldsymbol{a}_i\right] + \mathbb{E}\left[\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\boldsymbol{a}_i\right] + w_i\right\|_{\psi_1} \\ \leq & \left\|\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\boldsymbol{a}_i - \mathbb{E}\left[\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\boldsymbol{a}_i\right]\right\|_{\psi_1} + \left|\mathbb{E}\left[\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\boldsymbol{a}_i\right]\right| + \nu \\ \leq & (C\eta^2 + 1/2)\left\|\boldsymbol{X}\right\|_F + \nu,\end{aligned} \tag{59}$$

where the first inequality is due to the fact that $\mathbb{E}\left[\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\boldsymbol{a}_i\right]$ is a constant, $\|\mathbb{E}\left[\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\boldsymbol{a}_i\right]\|_{\psi_1} = |\mathbb{E}\left[\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\boldsymbol{a}_i\right]|$, and that $\|w_i\|_{\psi_1} = \nu$. Then,

$$\left\||\xi_i| - \mathbb{E}\left[|\xi_i|\right]\right\|_{\psi_1} \leq 2\left\|\xi_i\right\|_{\psi_1} \leq \eta_d\left\|\boldsymbol{X}\right\|_F + 2\nu, \tag{60}$$

where $\eta_d = 2C\eta^2 + 1$. Now by Bernstein's inequality in [40], for every $t > 0$,

$$\mathbb{P}\left\{\frac{1}{P}\sum_{i=1}^{P}(|\xi_i| - \mathbb{E}\left[|\xi_i|\right]) < -t\right\} \leq \exp\left[-C_0 P \min\left\{\frac{t^2}{(\eta_d\left\|\boldsymbol{X}\right\|_F + 2\nu)^2}, \frac{t}{\eta_d\left\|\boldsymbol{X}\right\|_F + 2\nu}\right\}\right].$$

Let $t = s(\eta_d\left\|\boldsymbol{X}\right\|_F + 2\nu)$. For any $s > 0$,

$$\mathbb{P}\left\{\frac{1}{P}\sum_{i=1}^{P}(|\xi_i| - \mathbb{E}\left[|\xi_i|\right]) < -s(\eta_d\left\|\boldsymbol{X}\right\|_F + 2\nu)\right\} \leq \exp\left[-C_0 P \min\left\{s^2, s\right\}\right]. \tag{61}$$

By Cauchy-Schwartz inequality, for any $i \in [P]$, we have

$$\left(\mathbb{E}\left[\xi_i^2\right]\right)^2 \leq \mathbb{E}\left[|\xi_i|\right]\mathbb{E}\left[|\xi_i|^3\right] \leq \mathbb{E}\left[|\xi_i|\right]\sqrt{\mathbb{E}\left[\xi_i^2\right]\mathbb{E}\left[\xi_i^4\right]},$$

which implies

$$\mathbb{E}\left[|\xi_i|\right] \geq \sqrt{\frac{(\mathbb{E}\left[\xi_i^2\right])^3}{\mathbb{E}\left[\xi_i^4\right]}}. \tag{62}$$

By the definition of sub-exponential norm and the fact that $\eta_d > 1$, we have

$$\mathbb{E}\left[\xi_i^4\right] \leq (4\left\|\xi_i\right\|_{\psi_1})^4 \leq (2\eta_d\left\|\boldsymbol{X}\right\|_F + 4\nu)^4 \leq (8\eta_d^2\left\|\boldsymbol{X}\right\|_F^2 + 32\nu^2)^2. \tag{63}$$

On the other hand, we have

$$\begin{aligned}
\mathbb{E}\left[\xi_i^2\right] &= \mathbb{E}\left[(\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\boldsymbol{a}_i)^2\right] + \mathbb{E}\left[w_i^2\right]\\
&= \mathbb{E}\left[(\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\boldsymbol{a}_i)\mathrm{tr}\left(\boldsymbol{a}_i\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\right)\right] + \sigma^2\\
&= \mathbb{E}\left[\mathrm{tr}\left((\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\boldsymbol{a}_i)\boldsymbol{a}_i\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\right)\right] + \sigma^2\\
&= \mathrm{tr}\left(\mathbb{E}\left[(\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\boldsymbol{a}_i)\boldsymbol{a}_i\boldsymbol{a}_i^{\mathrm{H}}\right]\boldsymbol{X}\right) + \sigma^2\\
&= \frac{1}{4}\mathrm{tr}\left((\boldsymbol{X} + \mathrm{tr}\left(\boldsymbol{X}\right)\boldsymbol{I})\boldsymbol{X}\right) + \sigma^2 \qquad (64)\\
&\geq \frac{1}{4}\left\|\boldsymbol{X}\right\|_F^2 + \sigma^2. \qquad (65)
\end{aligned}$$

Here we give an explanation of (64). Let $\boldsymbol{Y} = (\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{X}\boldsymbol{a}_i)\boldsymbol{a}_i\boldsymbol{a}_i^{\mathrm{H}}$. Then,

$$\begin{aligned}
\mathbb{E}\left[Y_{jk}\right] &= \mathbb{E}\left[\sum_{1\leq g,h\leq n} a_{ig}^* X_{gh} a_{ih} a_{ij} a_{ik}^*\right]\\
&= \sum_{g=1}^{n}\mathbb{E}\left[a_{ig}^* X_{gg} a_{ig} a_{ij} a_{ik}^*\right] + \sum_{g\neq h}\mathbb{E}\left[a_{ig}^* X_{gh} a_{ih} a_{ij} a_{ik}^*\right].
\end{aligned}$$

If $j = k$, we have

$$\begin{aligned}
\mathbb{E}[Y_{jj}] &= \sum_{g=1}^{n} X_{gg}\mathbb{E}[|a_{ig}|^2|a_{ij}|^2] + \sum_{g\neq h} X_{gh}\mathbb{E}[a_{ig}^* a_{ih}|a_{ij}|^2]\\
&= \frac{1}{4}(\mathrm{tr}\left(X\right) + X_{jj}).
\end{aligned}$$

If $j \neq k$, we have

$$\begin{aligned}
\mathbb{E}[Y_{jk}] &= \sum_{g=1}^{n} X_{gg}\mathbb{E}[|a_{ig}|^2 a_{ij} a_{ik}^*] + \sum_{g\neq h} X_{gh}\mathbb{E}[a_{ig}^* a_{ih} a_{ij} a_{ik}^*]\\
&= X_{jk}\mathbb{E}[|a_{ij}|^2|a_{ik}|^2]\\
&= \frac{1}{4}X_{jk}.
\end{aligned}$$

Therefore, $\mathbb{E}[\boldsymbol{Y}] = \frac{1}{4}(\boldsymbol{X} + \mathrm{tr}\left(\boldsymbol{X}\right)\boldsymbol{I})$.

By combining (62), (63), and (65), we have

$$\begin{aligned}
\mathbb{E}\left[|\xi_i|\right] &\geq \sqrt{\left(\frac{\frac{1}{4}\left\|\boldsymbol{X}\right\|_F^2 + \sigma^2}{8\eta_d^2\left\|\boldsymbol{X}\right\|_F^2 + 32\nu^2}\right)^2\left(\frac{1}{4}\left\|\boldsymbol{X}\right\|_F^2 + \sigma^2\right)}\\
&\geq \zeta\left\|\boldsymbol{X}\right\|_F,
\end{aligned}$$

where $\zeta = \frac{1}{2}\min\left\{\frac{1}{32\eta_d^2}, \frac{\sigma^2}{32\nu^2}\right\}$ is a constant determined by the distribution of a_{ij} and w_i. Then, by (61), we have

$$\mathbb{P}\left\{\frac{1}{P}\sum_{i=1}^{P}|\xi_i| < \zeta\left\|\boldsymbol{X}\right\|_F - s(\eta_d\left\|\boldsymbol{X}\right\|_F + 2\nu)\right\} \leq \exp\left[-C_0 P\min\left\{s^2, s\right\}\right],$$

which completes the proof.

L. Proof of Theorem 16

To prove Theorem 16, we make essential use of Lemma 18. Here, we restate Lemma 18, providing more details.

Lemma 21. *If $T_s = 1$, $\operatorname{supp}(\boldsymbol{z}_s) = \{l_s\}$, and threshold $t_1 \in (0, \varepsilon^2/2)$, then for any $j \in [R]$,*

$$\mathbb{P}\left\{\tilde{b}_j \neq b_{jl_s}\right\} = \mathcal{O}(1/K^3).$$

See the proof of Lemma 21 in Appendix M. Then, by union bound, $\mathbb{P}\{\tilde{\boldsymbol{b}} \neq \boldsymbol{B}_{l_s}\} = \mathcal{O}(R/K^3) \leq \mathcal{O}(1/K^2)$, since $K = \beta n^\delta$. Thus, we can reliably find l_s from the measurements with probability $1 - \mathcal{O}(1/K^2)$. For a right node with $T_s = 1$, the probability of error in the index tests and the probability of error in the energy test are $\mathcal{O}(1/K^2)$ and $\mathcal{O}(1/n^2)$, respectively. Therefore, the error probability of the tests for a right node is $\mathcal{O}(1/K^2)$. For a bin with $T_s > 1$, only the energy test needs to be considered and its error probability is $\mathcal{O}(1/n^2)$. Then, we know the probability of error in the index and energy tests is $\mathcal{O}(1/K^2)$. Since there are $\Theta(K)$ right nodes and a constant number of iterations, using the same decomposition method as in the proof of Theorem 15, the error probability of all the tests is $\mathcal{O}(1/K)$. Similar to the almost-linear scheme, considering the $\mathcal{O}(1/K)$ probability of unsuccessful recovery in the coloring algorithm when there is no error in the index and energy tests, the failure probability of sublinear scheme is $\mathbb{P}\{E_s\} = \mathcal{O}(1/K)$. Since the sample and computational complexity of the algorithm are already analyzed in Section VII-B, the proof of Theorem 16 is now complete.

M. Proof of Lemma 21

First, we define an event E_h such that there are more than $C_3 \log K$ active left nodes connected to a right node. As shown in [48], we have $\mathbb{P}\{E_h\} = \mathcal{O}(1/K^3)$. Now we condition on the coding pattern $\boldsymbol{h}$ such that $E_h^{\complement}$ happens, and thus $|\operatorname{supp}(\boldsymbol{z})| = T \leq C_3 \log K$. Similar to the almost-linear algorithm, we define $R+1$ linear mappings, $\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_R$, where

$$\mathcal{A}_0 : \ \boldsymbol{Z} \mapsto \{\boldsymbol{a}_i^{\mathrm{H}} \boldsymbol{Z} \boldsymbol{a}_i\}_{i \in [P]},$$
$$\mathcal{A}_j : \ \boldsymbol{Z} \mapsto \{\boldsymbol{f}_{j,i}^{\mathrm{H}} \boldsymbol{Z} \boldsymbol{f}_{j,i}\}_{i \in [Q]}, \ \text{for } j \in [R].$$

Then, $\boldsymbol{y}_j = \mathcal{A}_j(\boldsymbol{z}\boldsymbol{z}^{\mathrm{H}}) + \boldsymbol{w}_j$, $j \in \{0\} \cup [R]$.

Define the matrix $\tilde{\boldsymbol{Z}} = \{\tilde{Z}_{ij}\}_{N \times N} := \boldsymbol{z}\boldsymbol{z}^{\mathrm{H}} - \tilde{\boldsymbol{z}}_c \tilde{\boldsymbol{z}}_c^{\mathrm{H}} = \boldsymbol{z}\boldsymbol{z}^{\mathrm{H}} - \boldsymbol{z}_c \boldsymbol{z}_c^{\mathrm{H}}$. Then, $\tilde{\boldsymbol{y}}_j = \mathcal{A}_j(\tilde{\boldsymbol{Z}}) + \boldsymbol{w}_j$ and $\tilde{y}_{j,i} = \boldsymbol{f}_{j,i}^{\mathrm{H}} \tilde{\boldsymbol{Z}} \boldsymbol{f}_{j,i} + w_{j,i}$. Let $f_{j,i,m}$ be the mth element of $\boldsymbol{f}_{j,i}$. Since for a fixed j, $f_{j,i,m}$'s are independent, using similar argument to the one in Appendix K, we have

$$\left\|\boldsymbol{f}_{j,i}^{\mathrm{H}} \tilde{\boldsymbol{Z}} \boldsymbol{f}_{j,i} - \mathbb{E}\left[\boldsymbol{f}_{j,i}^{\mathrm{H}} \tilde{\boldsymbol{Z}} \boldsymbol{f}_{j,i}\right]\right\|_{\psi_1} \leq C_2 \eta_0^2 \left\|\tilde{\boldsymbol{Z}}\right\|_F.$$

Thus, $\|\tilde{y}_{j,i} - \mathbb{E}[\tilde{y}_{j,i}]\|_{\psi_1} \leq C_2 \eta_0^2 \|\tilde{\boldsymbol{Z}}\|_F + \nu$. Since there are $2T - 1$ nonzero entries in $\tilde{\boldsymbol{Z}}$, we have $\|\tilde{\boldsymbol{Z}}\|_F \leq \sqrt{2T-1} L_m \varepsilon$. Moreover, $T \leq C_3 \log K$, which implies that $\|\tilde{y}_{j,i} - \mathbb{E}[\tilde{y}_{j,i}]\|_{\psi_1} \leq C_4 \eta_0^2 L_m \varepsilon \sqrt{\log K} + \nu \leq \zeta_0 \sqrt{\log K}$, where ζ_0 is determined by η_0, L_m, ε, and ν.

On the other hand, since $T_s = 1$ and $\operatorname{supp}(\boldsymbol{z}_s) = \{l_s\}$, $\tilde{\boldsymbol{Z}}$ has only one non-zero element on the diagonal, i.e., $\tilde{Z}_{l_s l_s} = |z_{l_s}|^2$. Note that $\mathbb{E}[\tilde{y}_{j,i}] = \mathbb{E}[|f_{j,i,l_s}|^2] |z_{l_s}|^2 = b_{jl_s} |z_{l_s}|^2$. Thus, by Bernstein's inequality, for every $t \geq 0$,

$$\begin{aligned}\mathbb{P}\left\{\left|\frac{1}{Q}\sum_{i=1}^{Q}(\tilde{y}_{j,i} - b_{jl_s}|z_{l_s}|^2)\right| > t \mid \boldsymbol{h}\right\} &\leq 2\exp\left[-C_5 Q \min\left\{\frac{t^2}{\zeta_0^2 \log K}, \frac{t}{\zeta_0 \sqrt{\log K}}\right\}\right] \\ &\leq 2\exp\left[-\frac{C_5}{\zeta_0^2}\sqrt{Q}\min\left\{t^2, t\right\}\right],\end{aligned}$$

where the last inequality is due to the fact that $Q = \Theta(\log^2 N)$. We choose $t_1 = t < \varepsilon^2/2$. When $b_{jl_s} = 0$, we have

$$\mathbb{P}\left\{\left|\frac{1}{Q}\sum_{i=1}^{Q}\tilde{y}_{j,i}\right| > t_1 \mid \boldsymbol{h}\right\} \leq 2\exp\left[-\frac{C_5}{\zeta_0^2}\sqrt{Q}\min\left\{t_1^2, t_1\right\}\right], \tag{66}$$

and when $b_{jl_s} = 1$, we have

$$\begin{aligned}\mathbb{P}\left\{\left|\frac{1}{Q}\sum_{i=1}^{Q}\tilde{y}_{j,i}\right| < t_1 \mid \boldsymbol{h}\right\} &\leq \mathbb{P}\left\{\frac{1}{Q}\sum_{i=1}^{Q}\tilde{y}_{j,i} < t_1 \mid \boldsymbol{h}\right\} \\ &\leq \mathbb{P}\left\{\frac{1}{Q}\sum_{i=1}^{Q}\tilde{y}_{j,i} < |z_{l_s}|^2 - t_1 \mid \boldsymbol{h}\right\} \qquad (67) \\ &\leq \mathbb{P}\left\{\left|\frac{1}{Q}\sum_{i=1}^{Q}\tilde{y}_{j,i} - |z_{l_s}|^2\right| > t_1 \mid \boldsymbol{h}\right\} \\ &\leq 2\exp\left[-\frac{C_5}{\zeta_0^2}\sqrt{Q}\min\left\{t_1^2, t_1\right\}\right], \qquad (68)\end{aligned}$$

where the inequality (67) is due to the fact that $t_1 < \varepsilon^2/2$ and $|z_{l_s}|^2 \geq \varepsilon^2$. Define the error events $E_{\text{index}} = \{|\frac{1}{Q}\sum_{i=1}^{Q} \tilde{y}_{j,i}| > t_1\}$, when $b_{jl_s} = 0$, and $E_{\text{index}} = \{|\frac{1}{Q}\sum_{i=1}^{Q} \tilde{y}_{j,i}| < t_1\}$, when $b_{jl_s} = 1$. Then, since $Q = \Theta(\log^2(n))$ and inequalities (66) and (68) hold for any $\boldsymbol{h} \in E_h^{\complement}$, we have,

$$\mathbb{P}\{E_{\text{index}} | E_h^{\complement}\} = \mathcal{O}(1/K^3).$$

Now we know that

$$\begin{aligned}
\mathbb{P}\{E_{\text{index}}\} &= \mathbb{P}\{E_{\text{index}} | E_h^{\complement}\}\mathbb{P}\{E_h^{\complement}\} + \mathbb{P}\{E_{\text{index}} | E_h\}\mathbb{P}\{E_h\} \\
&\leq \mathbb{P}\{E_{\text{index}} | E_h^{\complement}\} + \mathbb{P}\{E_h\} \\
&= \mathcal{O}(1/K^3) + \mathcal{O}(1/K^3) \\
&= \mathcal{O}(1/K^3),
\end{aligned}$$

which completes the proof.

N. Hanson-Wright Inequality for Complex Random Variables

Theorem 22. *Let $\boldsymbol{\gamma} = \{\gamma_i\}_{i\in[n]} \in \mathbb{C}^n$ be a random vector with independent entries γ_i, satisfying $\mathbb{E}[\gamma_i] = 0$, and $|\gamma_i|$ is sub-gaussian with $\||\gamma_i|\|_{\psi_2} \leq \eta$ for all $i \in [n]$. Let $\boldsymbol{U} \in \mathbb{C}^{n\times n}$ be a Hermitian matrix. Then, for every $t \geq 0$,*

$$\mathbb{P}\left\{\left|\boldsymbol{\gamma}^{\mathrm{H}}\boldsymbol{U}\boldsymbol{\gamma} - \mathbb{E}\left[\boldsymbol{\gamma}^{\mathrm{H}}\boldsymbol{U}\boldsymbol{\gamma}\right]\right| > t\right\} \leq 6\exp\left[-c_0 \min\left\{\frac{t^2}{\eta^4\|\boldsymbol{U}\|_F^2}, \frac{t}{\eta^2\|\boldsymbol{U}\|}\right\}\right].$$

Proof: Let $\boldsymbol{\alpha} = \{\alpha_i\}_{i\in[n]}$ and $\boldsymbol{\beta} = \{\beta_i\}_{i\in[n]}$ be the real and imaginary parts of $\boldsymbol{\gamma}$. Then, we know that α_i's and β_i's are sub-gaussian random variables with $\|\alpha_i\|_{\psi_2} \leq \eta$ and $\|\beta_i\|_{\psi_2} \leq \eta$ for all $i \in [n]$. Note that here, although γ_i's are independent, the real and imaginary parts of γ_i are not necessarily independent for a certain i. In other words, for any i, α_i and β_i may not be independent.

Let $\boldsymbol{V}$ and $\boldsymbol{W}$ be the real and imaginary parts of $\boldsymbol{U}$. Since $\boldsymbol{U}$ is a Hermitian matrix, we have $\boldsymbol{V} = \boldsymbol{V}^{\mathrm{T}}$ and $\boldsymbol{W} = -\boldsymbol{W}^{\mathrm{T}}$. We also know that $\boldsymbol{\gamma}^{\mathrm{H}}\boldsymbol{U}\boldsymbol{\gamma}$ is a real number. Then, we have

$$\boldsymbol{\gamma}^{\mathrm{H}}\boldsymbol{U}\boldsymbol{\gamma} = \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{\alpha} - 2\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{\beta} + \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{\beta}.$$

Therefore, $\mathbb{P}\left\{\left|\boldsymbol{\gamma}^{\mathrm{H}}\boldsymbol{U}\boldsymbol{\gamma} - \mathbb{E}\left[\boldsymbol{\gamma}^{\mathrm{H}}\boldsymbol{U}\boldsymbol{\gamma}\right]\right| > t\right\}$ is upper bounded by three terms,

$$\begin{aligned}
\mathbb{P}\left\{\left|\boldsymbol{\gamma}^{\mathrm{H}}\boldsymbol{U}\boldsymbol{\gamma} - \mathbb{E}\left[\boldsymbol{\gamma}^{\mathrm{H}}\boldsymbol{U}\boldsymbol{\gamma}\right]\right| > t\right\} \leq &\mathbb{P}\left\{\left|\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{\alpha} - \mathbb{E}\left[\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{\alpha}\right]\right| > t/4\right\} \\
&+\mathbb{P}\left\{\left|\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{\beta} - \mathbb{E}\left[\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{\beta}\right]\right| > t/4\right\} \\
&+\mathbb{P}\left\{\left|\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{\beta} - \mathbb{E}\left[\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{\beta}\right]\right| > t/4\right\}.
\end{aligned} \tag{69}$$

Since α_i's are independent and $\mathbb{E}[\alpha_i] = 0$, according to the Hanson-Wright inequality for real numbers [49], we have

$$\mathbb{P}\left\{\left|\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{\alpha} - \mathbb{E}\left[\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{\alpha}\right]\right| > t/4\right\} \leq 2\exp\left[-c_1 \min\left\{\frac{t^2}{\eta^4\|\boldsymbol{V}\|_F^2}, \frac{t}{\eta^2\|\boldsymbol{V}\|}\right\}\right].$$

Further, we have $\|\boldsymbol{V}\|_F \leq \|\boldsymbol{U}\|_F$, $\|\boldsymbol{V}\| \leq \|\boldsymbol{U}\|$. Therefore,

$$\mathbb{P}\left\{\left|\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{\alpha} - \mathbb{E}\left[\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{\alpha}\right]\right| > t/4\right\} \leq 2\exp\left[-c_1 \min\left\{\frac{t^2}{\eta^4\|\boldsymbol{U}\|_F^2}, \frac{t}{\eta^2\|\boldsymbol{U}\|}\right\}\right]. \tag{70}$$

And similarly,

$$\mathbb{P}\left\{\left|\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{\beta} - \mathbb{E}\left[\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{\beta}\right]\right| > t/4\right\} \leq 2\exp\left[-c_2 \min\left\{\frac{t^2}{\eta^4\|\boldsymbol{U}\|_F^2}, \frac{t}{\eta^2\|\boldsymbol{U}\|}\right\}\right]. \tag{71}$$

Now consider the cross term. Let W_{ij} be the entries of $\boldsymbol{W}$. Since $\boldsymbol{W} = -\boldsymbol{W}^{\mathrm{T}}$, $W_{ii} = 0$ for all $i \in [n]$. Then, $\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{\beta} = \sum_{i\neq j} W_{ij}\alpha_i\beta_j$, and $\mathbb{E}\left[\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{\beta}\right] = 0$. Then, we can bound $\mathbb{P}\left\{\left|\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{\beta}\right| > t/4\right\}$ in the same way as in [49] so that

$$\mathbb{P}\left\{\left|\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{\beta}\right| > t/4\right\} \leq 2\exp\left[-c_3 \min\left\{\frac{t^2}{\eta^4\|\boldsymbol{U}\|_F^2}, \frac{t}{\eta^2\|\boldsymbol{U}\|}\right\}\right]. \tag{72}$$

By combining (70), (71), and (72), Theorem 22 is proved.

■

O. Pseudocode

In this subsection, we provide the pseudocode of the PhaseCode algorithm. Moreover, we provide the pseudocodes of the right node processors: singleton processor, mergeable multiton processor, and resolvable multiton processor.

Pseudocode 1 PhaseCode Algorithm

```
$\mathcal{I} \leftarrow \emptyset$                                              ▷ No active component is found in the beginning
for each i in $\{1, 2, ..., M\}$ do                                                           ▷ Find all singletons
    Singleton Processor
for each i in $\{1, 2, ..., M\}$ do                                                ▷ Find all doubletons and merge
    Mergeable Multiton Processor
Color$_0$ ← Color of the largest colored component                              ▷ Find the largest colored component$^*$
for each $\ell$ in $\mathcal{I}$ do                                ▷ Uncolor all other left nodes and delete all values of them
    if Color$_\ell \neq$ Color$_0$ then
        $x_\ell \leftarrow$ None
        Color$_\ell$ ← None
        $\mathcal{I} \leftarrow \mathcal{I} - \{\ell\}$
while $|\mathcal{I}| < K$ and any changes are made in the previous loop do                    ▷ Keep resolving multitons
    Resolvable Multiton Processor
```

Pseudocode 2 Singleton Processor

```
if $y_{i,1} = y_{i,2} = y_{i,4}$ then                                ▷ Check whether this right node is a singleton or not
    $\ell \leftarrow \frac{1}{\omega} \cos^{-1}(\frac{y_{i,3}}{2y_{i,1}})$          ▷ Find the index of the active left node connected to this right node
    $x_\ell \leftarrow y_{i,1}$                                                    ▷ Assign a value to the active left node
    $\mathcal{I}_0 \leftarrow \mathcal{I}_0 \cup \{\ell\}$                                            ▷ Declare a new found active left node
    Color$_\ell$ ← new color                                      ▷ Color the new active left node with a new color
```

Pseudocode 3 Mergeable Multiton Processor

```
if Right node $i$ is connected to no colored active left node or the number of colors connected to the right node is not exactly
2 then
    Return                                                                        ▷ If this right node is not mergeable
Red, Blue ← Two colors of the active left nodes connected to the right node
$\mathcal{R} \leftarrow$ indices of the active left nodes that are colored with Red
$\mathcal{B} \leftarrow$ indices of the active left nodes that are colored with Blue
$r \leftarrow \sum_{\ell \in \mathcal{R}} x_j e^{\mathbf{i}\omega\ell}$
$b \leftarrow \sum_{\ell \in \mathcal{B}} x_\ell e^{\mathbf{i}\omega\ell}$
for each $z_1$ in $\{+1, -1\}$ do                                                            ▷ Consider two candidate
    $\phi \leftarrow z_1 \cos^{-1}\left(\frac{|r|^2+|b|^2-y_{i,1}^2}{2|r||b|}\right) + \angle r - \angle b$                          ▷ Find a candidate for phase offset
    if $\left|\sum_{\ell \in \mathcal{R}} x_\ell e^{\mathbf{i}\omega'\ell} + \exp(\mathbf{i}\phi) \times \sum_{\ell \in \mathcal{B}} x_\ell e^{\mathbf{i}\omega'\ell}\right| = y_{i,4}$ then                ▷ Check the candidate with $y_{i,4}$
        Color Red and Color Blue are combined to a new color
        for each $\ell$ in $\mathcal{B}$ do                ▷ Adjust phase of the components that are colored with Color Blue $^*$
            $x_\ell \leftarrow x_\ell \times \exp(\mathbf{i}\phi)$
        Return
```

Pseudocode 4 Resolvable Multiton Processor

```
if Right node $i$ is connected to no colored active left node or they are colored with more than 1 color then
    Return    ▷ If this right node is not resolvable
Color ← Common color of the connected active left nodes
$\mathcal{I}' \leftarrow \mathcal{I} \cap \{j | H_{i,j} = 1, 1 \leq j \leq n\}$    ▷ Colored active left nodes connected to this right node
$a \leftarrow \sum_{i \in \mathcal{I}'} x_i e^{\mathbf{i}\omega\ell}$
$b \leftarrow \sum_{i \in \mathcal{I}'} x_i e^{-\mathbf{i}\omega\ell}$
$c \leftarrow \sum_{i \in \mathcal{I}'} 2\cos(\omega\ell) x_i$
$d \leftarrow \sum_{i \in \mathcal{I}'} x_i e^{\mathbf{i}\omega'\ell}$
for each $z_1$ in $\{+1, -1\}$ do    ▷ Consider two signs of $\alpha$
    $\alpha \leftarrow z_1 \cos^{-1}(\frac{y_{i,3}^2 - y_{i,1}^2 - y_{i,2}^2}{2 y_{i,1} y_{i,2}})$
    $z \leftarrow \frac{y_{i,1}}{y_{i,2}} \exp(\alpha \mathbf{i})$
    $k_1 \leftarrow 1 - z + \frac{2(zb-a)}{c}$
    $k_2 \leftarrow 1 + z$
    $k_3 \leftarrow 1 - z$
    $k_4 \leftarrow \frac{y_{i,3}}{|c|}$
    $k_5 \leftarrow |k_1|^2 - k_4^2 |k_3|^2$
    $k_6 \leftarrow |k_2|^2 - k_4^2 |k_2|^2$
    $k_7 \leftarrow 2\,\mathrm{Re}(k_1)\,\mathrm{Im}(k_2) - 2\,\mathrm{Im}(k_1)\,\mathrm{Re}(k_2) + k_4^2(2\,\mathrm{Re}(k_2)\,\mathrm{Im}(k_3) - 2\,\mathrm{Re}(k_3)\,\mathrm{Im}(k_2))$
    $k_8 \leftarrow k_6^2 + k_7^2 - 2k_6k_7 + k_8^2$
    $k_9 \leftarrow 2k_6k_7 - k_8^2 - 2k_7^2$
    $k_{10} \leftarrow k_7^2$
    for each $z_2$ in $\{+1, -1\}$ do    ▷ Consider two solutions of a quadratic equation
        if $k_9^2 - 4k_8k_{10} < 0$ then
            Continue
        if $\frac{-k_9 + z_2\sqrt{k_9^2 - 4k_8k_{10}}}{2k_8} < 0$ then
            Continue
        $\ell' \leftarrow \cos^{-1}\left[\sqrt{\frac{-k_9 + z_2\sqrt{k_9^2 - 4k_8k_{10}}}{2k_8}}\right] / \omega$    ▷ Find a candidate of $\ell$
        $x' \leftarrow \frac{zb-a}{e^{\mathbf{i}\omega\ell} - ze^{-\mathbf{i}\omega\ell}}$    ▷ Find a candidate of $x_\ell$
        if $y_{i,4} = |d + e^{\mathbf{i}\omega'\ell'} x'|$ then    ▷ Check the validity of the candidates with $y_{i,4}$
            $x_{\ell'} \leftarrow x'$    ▷ Assign a value to the component
            $\mathcal{I}_0 \leftarrow \mathcal{I}_0 \cup \{\ell'\}$    ▷ Declare a new found component
            Color$'_\ell$ ← Color ▷ Color the new component with the color of the other components connected to the right node
            Return
```