Sub-linear Time Compressed Sensing for Support Recovery using Sparse-Graph Codes

Xiao Li, Sameer Pawar and Kannan Ramchandran*
Department of Electrical Engineering and Computer Science (EECS)
University of California, Berkeley
{xiaoli, spawar, kannanr}@eecs.berkeley.edu

September 26, 2015

Abstract

We address the problem of robustly recovering the support of high-dimensional sparse signals[1] from linear measurements in a low-dimensional subspace. We introduce a new compressed sensing framework through carefully designed sparse measurement matrices associated with low measurement costs and low-complexity recovery algorithms. The measurement system in our framework captures observations of the signal through well-designed *measurement matrices sparsified* by capacity-approaching *sparse-graph codes*, and then recovers the signal by using a simple peeling decoder. As a result, we can simultaneously reduce both the measurement cost and the computational complexity. In this paper, we formally connect general sparse recovery problems in compressed sensing with sparse-graph decoding in packet-communication systems, and analyze our design in terms of the measurement cost, computational complexity and recovery performance.

Specifically, by structuring the measurements through sparse-graph codes, we propose two families of measurement matrices, the *Fourier family* and the *binary family* respectively, which lead to different measurement and computational costs. In the *noiseless* setting, our framework recovers the sparse support of any K-sparse signal in time[2] $O(K)$ with $2K$ measurements obtained by the *Fourier family*, or in time $O(K \log N)$ using $K \log_2 N + K$ measurements obtained by the *binary family*. In the presence of noise, *both* measurement and computational costs are reduced to $O(K \log^{1.3} N)$ in the case of the *Fourier family*. More importantly, the *binary family* achieves $O(K \log N)$ for both the measurement cost and the computational complexity in the presence of noise, which maintains the same measurement and computational scaling as the noiseless case. Therefore, when the signal sparsity K is sub-linear in the signal dimension N, our framework achieve *sub-linear time support recovery*. Further, our framework also admits a wide class of *random matrix family* that achieves $O(K \log N)$ measurements with near-linear run-time $O(N \log N)$. In terms of recovery performance, we show that our framework succeeds with probability one asymptotically under finite signal-to-noise ratios.

1 Introduction

A classic problem of interest is that of estimating an unknown vector $\mathbf{x}$ of length N from noisy observations

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w}, \tag{1}$$

where $\mathbf{A}$ is an $M \times N$ known matrix typically referred to as the *measurement matrix* and $\mathbf{w}$ is an additive noise vector. We refer to N as the *signal dimension*. In general, if $\mathbf{x}$ has no additional structure, it is impossible to recover $\mathbf{x}$ from fewer measurements than the signal dimension. However, if the signal is known to be sparse with

*This work was supported by grants NSF CCF EAGER 1439725, and NSF CCF 1116404 and MURI CHASE Grant No. 556016.

[1]The signal of interest can be sparse with respect to any known basis.

[2]The run-time is measured by the number of arithmetic operations needed for recovery after the measurements have been obtained.

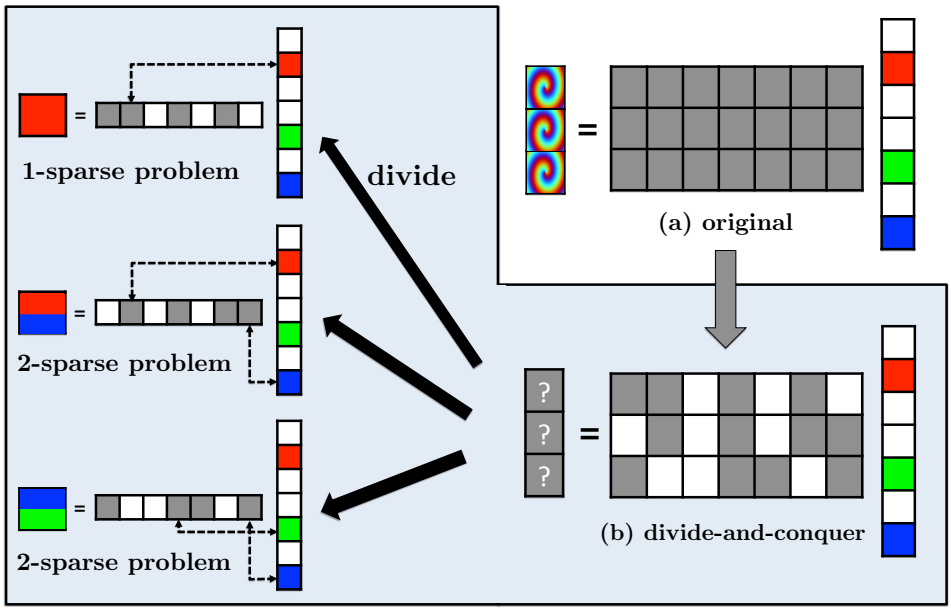


Figure 1: A conceptual cartoon diagram of the "divide-and-conquer" philosophy used in our design. In this diagram, zero entries are colored in *white* and the non-zero entries in the sparse vector are colored in *red*, *green* and *blue* respectively. We have a 3-sparse recovery problem in sub-figure (a), where the measurement matrix is colored in *grey* to indicate an arbitrary design. The resulting measurements are colored as *mixtures* because of the arbitrary mixing of different color components (red, green, blue). In sub-figure (b), we sparsify the measurement matrix by introducing three zeros in each row shown as the *white* spots. The resulting sparsified measurement matrix effectively divides the 3-sparse recovery problem into multiple sub-problems, where one of the sub-problems involves only one color that can be easily identified. In this example, the first measurement contains a single *red* color, whereas the second and third measurements contain a *mixture or red and blue* and a *mixture of blue and green* respectively. If the decoder knows that the first measurement contains a single *red* color, it can peel off its contribution from the *mixture of red and blue* in the second measurement, which forms a new measurement containing a single *blue* color. Similarly, the blue color can be peeled off from the third measurement such that the *green* color is decoded.

respect to some basis, wherein only K coefficients are non-zero or significant with $K \ll N$, it is possible to recover the signal from significantly fewer measurements. This has been studied extensively in the literature under the name of *compressed sensing* [1]. The compressed sensing problem of reconstructing high-dimensional signals from lower dimensional observations arises in diverse fields, such as medical imaging [2], optical imaging [3], speech and image processing [4], data streaming and sketching [5] etc. A large variety of design approaches and reconstruction algorithms have been proposed in the literature to exploit the inherent sparsity of signals to recover them from low-dimensional linear measurements (see Section 2.2 for a brief review of these methods).

In this work, we take a new "divide-and-conquer" approach to the problem by viewing compressed sensing through a new "sparse-graph coding" lens. Our design philosophy is depicted in Fig. 1 as a cartoon illustration, where we use different colors to distinguish the entries in the sparse vector, namely, we choose *red, green* and *blue* respectively for the non-zero entries, and *white* for zero entries. A conventional design in compressed sensing is to generate weighted linear measurements of the sparse vector through a carefully designed *measurement matrix* [6]. In this example, all the entries of the measurement matrix are colored in *grey* to indicate an arbitrary design and the corresponding measurements are some generic mixtures of *red, green* and *blue*, as shown in Fig. 1-(a).

We design the measurement matrix by sparsifying each row of the measurement matrix with zero patterns guided by sparse-graph codes, indicated by the *white* spots in Fig. 1-(b). This new measurement matrix leads to a different set of measurements, where some contain single colors and some contain their mixtures. Our design philosophy is to *disperse* the signal into multiple single color measurements (e.g., the red color in the first measurement) and *peel* them off from color mixtures (e.g., the red-blue mixture in the second measurement and the blue-green mixture in the third measurement) to decode other unknown colors in the spirit of "divide-and-conquer". By analogy, the use of sparse-graph codes essentially *divides* the general sparse recovery problem into multiple

sub-problems that can be easily *conquered* and synthesized for reconstructions. Furthermore, by viewing our design from a coding-theoretic lens, our design can further leverage the properties of sparse-graph codes in terms of both measurement cost (capacity-approaching) and computational complexity (fast peeling-based decoding). This leads to a new family of sparse measurement matrices simultaneously featuring low measurement costs and low computational costs.

1.1 Objective

In this paper, we focus on the recovery of the *exact support* of any K-sparse N-length signal. This so-called *support recovery* problem arises in an array of applications such as model selection [7], group testing [8], sparse approximation [9] and subset selection in regression problems [10]. Given $\widehat{\mathbf{x}}$ generated by some recovery method, there are various criteria for evaluating the recovery performance. A typical metric for *support recovery* is the probability $\mathbb{P}_F$ of failing to recover the *exact support* of the signal, defined as

$$\mathbb{P}_F := \mathsf{Pr}\left(\mathsf{supp}\left(\widehat{\mathbf{x}}\right) \neq \mathsf{supp}\left(\mathbf{x}\right)\right), \tag{2}$$

where $\mathsf{supp}\left(\cdot\right)$ represents the support of some vector

$$\mathsf{supp}\left(\mathbf{x}\right) := \{k:\ x[k] \neq 0,\ k \in [N]\}. \tag{3}$$

where $[N]$ is the set of integers $\{0, 1, \cdots, N-1\}$. The probability $\mathbb{P}_F$ is evaluated with respect to the randomness associated with the noise $\mathbf{w}$ *and* the measurement matrix $\mathbf{A}$. In other words, for any given K-sparse signal $\mathbf{x}$, our design generates a measurement matrix $\mathbf{A}$ (from a specific random ensemble[3]) and produces an estimate $\widehat{\mathbf{x}}$ whose support matches *exactly* that of $\mathbf{x}$ with probability $1 - \mathbb{P}_F$ approaching one asymptotically in K and N (i.e. $\lim_{K,N\to\infty} \mathbb{P}_F \to 0$).

1.2 Our Contributions

Our key contribution is the proposed new compressed sensing design framework for support recovery featuring *sub-linear time* complexity[4] and (near) optimal measurement costs. The sub-linear time feature can potentially enable real-time or near-real-time processing for massive datasets featuring sparsity, which are relevant to a multitude of practical applications. Here, using the big-Oh notation[5], we briefly summarize our technical results with respect to the *Fourier family* and the *binary family* in terms of measurement and computational costs in the following table.

	Measurement (Noiseless)	Complexity (Noiseless)	Measurement (Noisy)	Complexity (Noisy)
Fourier	$2(1+\epsilon)K$	$O(K)$	$O(K\log^{1.3} N)$	$O(K\log^{1.3} N)$
Binary	$(1+\epsilon)K(\log_2 N + 1)$	$O(K\log N)$	$O(K\log N)$	$O(K\log N)$

Table 1: Measurement Cost and Computational Complexity of Our Designs

We now provide some intuition about our sub-linear time results. Recall that the idea is to use sparse-graph codes to structure the measurement matrix in order to generate subsets of measurements containing isolated 1-sparse coefficients, as well as their mixtures. From Fig. 1, these 1-sparse coefficients (e.g., the red color in the first measurement) can be peeled off from their mixtures (e.g., the red and blue mixture in the second measurement), which forms new 1-sparse coefficients for further peeling. This divide-and-conquer approach allows us to tackle a K-sparse recovery problem by solving a series of 1-sparse problems instead. Therefore, the challenge is to keep

[3]Note that this is what is known as the "for-each" guarantee [5] in contrast to the "for-all" guarantee in some compressed sensing contributions, where a single measurement matrix is used for all signals once generated.

[4]The computational costs in this paper refer to the reconstruction time after *linear measurements* are available.

[5]Recall that a single variable function $f(x)$ is said to be $O(g(x))$, if for a sufficiently large x the function $|f(x)|$ is bounded above by $|g(x)|$, i.e., $\lim_{x\to\infty} |f(x)| < c|g(x)|$ for some constant c. Similarly, $f(x) = \Omega(g(x))$ if $\lim_{x\to\infty} |f(x)| > c|g(x)|$ and $f(x) = o(g(x))$ if the growth rate of $|f(x)|$ as $x \to \infty$, is negligible as compared to that of $|g(x)|$, i.e. $\lim_{x\to\infty} |f(x)|/|g(x)| = 0$.

this peeling process going until all 1-sparse components have been recovered. Hence we invoke sparse-graph codes principles to study this "turbo" peeling process theoretically to guarantee the success of decoding. As a result, we can focus on solving each 1-sparse problem. Clearly, depending on the specific measurement matrix used, there are many ways to solve these 1-sparse problems. In this paper, our sub-linear results are based on two designs for solving these 1-sparse problems:

- the Fourier family exploits the *Discrete Fourier Transform (DFT) matrix* by leveraging spectral estimation techniques [11]. Suppose that we choose the first two rows of the DFT matrix as the measurement matrix before being sparsified by sparse-graph codes. We have two measurements to estimate the unknown index and the unknown value of the 1-sparse coefficient, which is equivalent to estimating the frequency and amplitude of a complex discrete sinusoid from the DFT matrix. Therefore, in the noiseless setting, the frequency can be estimated by simply examining the relative phase between the two measurements, which only requires $O(1)$ measurements and computations. Then the unknown value of the coefficient can be obtained easily given the frequency. In the noisy setting, we further devise a successive spectral estimation scheme to obtain the unknown frequency and value using only $O(\log^{1.\dot{3}} N)$ measurements and $O(\log^{1.\dot{3}} N)$ operations.

- the binary family uses a simple $\log_2 N \times N$ *binary expansion matrix* where each column is a length-$\log_2 N$ vector containing the binary representation of the column index $k \in [N]$. Using this measurement matrix, there are $\log_2 N$ measurements in total. By taking the absolute values of the measurements, in the noiseless setting, we can directly obtain the signs of the measurements as the binary expansion index of the 1-sparse coefficient (assuming that the coefficient is positive). When the sign of the coefficient is unknown, we can use an extra row consisting of all one's to provide a reference sign. In fact, the signs of the measurements can be viewed as a length-$\log_2 N$ message bits received by the decoder for obtaining the unknown location of the 1-sparse coefficient. Therefore in the noisy setting, according to the channel coding theorem, we can encode the binary expansion matrix using good channel codes consisting of N codewords such that the index and the value of the 1-sparse coefficient can be decoded correctly in the presence of noise with high probability. If the channel codes are chosen with a block length of $O(\log N)$ and a decoding time that is *linear* in its block length $O(\log N)$, we can achieve a $O(\log N)$ costs for both measurements and computations for solving each 1-sparse problem.

Finally, since there are in total K sparse coefficients to estimate, the overall measurement and computational costs are further multiplied by a factor of K, which gives our sub-linear time results.

1.3 Notation and Organization

Throughout this paper, we use $\mathbb{R}$ and $\mathbb{C}$ to denote the real and complex fields. Any boldface lowercase letter such as $\mathbf{x} \in \mathbb{C}^N$ represents a vector containing the complex samples $\mathbf{x} = [x[0], \cdots, x[N-1]]^T$, and a boldface uppercase letter, such as $\mathbf{X} \in \mathbb{C}^{M \times N}$, represents a matrix with elements $X_{i,j}$ for $i \in [M]$ and $j \in [N]$. The calligraphic uppercase letter (i.e., $\mathcal{A}$) represents a set with cardinality denoted by $|\mathcal{A}|$, and the complement of set $\mathcal{A}$ is denoted by $\mathcal{A}^c$. The inner product between two vectors is defined as $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{t \in [N]} x[t](y[t])^*$ with arithmetic over $\mathbb{C}$.

This paper is organized as follows. In Section 2.2, we provide a brief overview of existing sparse recovery methods. Then we briefly summarize our main technical results in Section 2. In Section 3, for illustration purpose we provide a concrete example of our design framework using sparse-graph codes, followed by the analysis of the peeling decoder for sparse support recovery. Based on the example, we propose the principle and mathematical formulation of our measurement design in Section 4. Then, we proceed to discuss specific constructions for our noiseless recovery results in Section 6.1 and further the noisy recovery results in Section 7. Last but not least, we provide numerical results in Section 8 to corroborate our noiseless and noisy recovery performance.

2 Main Results and Related Work

In this paper, we consider the problem of recovering the sparse support of $\mathbf{x}$ from the measurements[6] obtained in (1). In particular, we are interested in support recovery for the noiseless and noisy settings. Our design is characterized by the triplet $(M, T, \mathbb{P}_F)$, where M is the measurement cost, T is the computational complexity in terms of arithmetic operations, and $\mathbb{P}_F$ is the failure probability defined in (2).

2.1 Main Results

To recover arbitrary K-sparse signals, we propose two designs with sub-linear time features as follows:

1. the *Fourier family* designs for noiseless and noisy recovery, called `FourierNoiseless` and `FourierNoisy`;
2. the *binary family* designs for noiseless and noisy recovery, called `BinaryNoiseless` and `BinaryNoisy`.

Theorem 1 (**Noiseless Recovery**). *In the absence of noise* $\mathbf{w} = \mathbf{0}$*, given any* K*-sparse signal* $\mathbf{x}$ *with* $x[k] \in \mathbb{C}$ *for* $k \in \mathsf{supp}(\mathbf{x})$ *and an arbitrary* $\epsilon > 0$*, our noiseless recovery schemes achieve a vanishing failure probability* $\mathbb{P}_F \to 0$ *asymptotically in* K *and* N *with*

	Measurement Cost M	Computational Complexity T
`FourierNoiseless`	$2(1+\epsilon)K$	$O(K)$
`BinaryNoiseless`	$(1+\epsilon)K(\log_2 N + 1)$	$O(K \log N)$

Table 2: Measurement Cost and Computational Complexity of Our Noiseless Recovery Results

Proof. We give the details and proofs of our design for the `FourierNoiseless` scheme in Section 6.1, and for the `BinaryNoiseless` scheme in Section 6.2. □

When it comes to the noisy settings, we further assume that all the non-zero coefficients belong to a set $\mathcal{X} = \{Ae^{\mathrm{i}\theta} : A \in \mathcal{A}, \theta \in \Theta\}$ where $\mathcal{A} := \{A_{\min} + \ell\rho\}_{\ell=0}^{L_1-1}$ for some arbitrarily large but finite $L_1 > 1$, $\rho > 0$ and $A_{\min} > 0$ while $\Theta := \{2\pi\ell/L_2\}_{\ell=0}^{L_2-1}$ for some arbitrarily large but finite $L_2 > 0$. This finite constellation assumption is imposed to simplify our analysis and its cardinality can be arbitrarily large but finite, which subsumes all practical digital signals that have been quantized with finite precision (essentially any signal processed by a digital computer).

Theorem 2 (**Sub-linear Time Noisy Recovery**). *In the presence of i.i.d. Gaussian noise with zero mean and variance* σ^2*, given any* K*-sparse signal* $\mathbf{x}$ *with* $x[k] \in \mathcal{X}$ *for* $k \in \mathsf{supp}(\mathbf{x})$*, our noisy recovery schemes achieve a vanishing failure probability* $\mathbb{P}_F \to 0$ *asymptotically in* K *and* N *with*

	Measurment Cost M	Computational Complexity T
`FourierNoisy`	$O(K \log^{1.3} N)$	$O(K \log^{1.3} N)$
`BinaryNoisy`	$O(K \log N)$	$O(K \log N)$

Table 3: Measurement Cost and Computational Complexity of Our Noisy Recovery Results

Proof. We give the details and proofs of our design for the `FourierNoisy` scheme in Section 7.3, and for the `BinaryNoisy` scheme in Section 7.4. □

Last but not least, we also have a *random matrix family*, called the `RandomNoisy` design, that encompasses a wide range of classic compressed sensing matrices with independent identically distributed (i.i.d.) entries such as Gaussian, Rademacher, Bernoulli, partial DFT matrix and so on. Although this design does not lead to sub-linear time recovery, we include it here for completeness and relevance to our sub-linear time results.

[6] More generally, we also allow the signal to be sparse in any linear transform domain. If the signal is sparse in the transform domain, one can pre-multiply the matrix $\mathbf{A}$ on right by the appropriate inverse transform.

Theorem 3 (**Near-linear Time Noisy Recovery**). *In the presence of i.i.d. Gaussian noise with zero mean and variance σ^2, given any K-sparse signal $\mathbf{x}$ with $x[k] \in \mathcal{X}$ for $k \in \texttt{supp}(\mathbf{x})$, the* `RandomNoisy` *scheme achieves a vanishing failure probability $\mathbb{P}_F \to 0$ asymptotically in K and N with a measurement cost of $M = O(K \log N)$ and a computational cost of $T = O(N \log N)$.*

Proof. See the design and proofs in Section 7.1. □

2.2 Related Work

In this section, we provide a summary of sparse recovery methods and point out a few differences that distinguish our design from existing results that can be roughly categorized into four classes. In the following we provide a brief overview of each class and place our design in context.

Convex Relaxation Approach: The classic formulation for sparse recovery from linear measurements is through an ℓ_0-norm minimization, which is a non-convex optimization problem. This problem has been known to be notoriously hard to solve. Convex optimization techniques relax the original combinatorial problem to a convex ℓ_1-norm minimization problem, where computationally efficient algorithms are designed to solve this relaxed problem. It has been shown that as long as the measurement matrices satisfy the Restricted Isometry Property (RIP) or mutual coherence (MC) conditions, the ℓ_1-relaxation of the original problem has exactly the same sparse solution as the original combinatorial problem. This class of methods is known to provide a high level of robustness against the measurement noise, and furthermore, do not depend on the structure of measurement matrices. Popular algorithms in this class include LASSO [12], Iterative Hard Thresholding (IHT) [13], fast iterative shrinkage-thresholding algorithm (FISTA) [14], message passing [15], Dantzig selector [16] and so on. Most of the existing results along this line measurement matrices that are characterized by a measurement cost of $O(K \log(N/K))$ and a computational complexity $O(\text{poly}(N))$.

Greedy Methods: Another class of methods, referred to as greedy iterative algorithms, attempts to solve the original ℓ_0-minimization problem directly using successive approximations of the sparse signal through various heuristics. Examples include Orthogonal Matching Pursuit (OMP) [17], CoSaMP [18], Regularized OMP (ROMP) [19], Stagewise OMP (StOMP) [20] and so on. Similar to convex relaxation approaches, this class also does not depend on the structure of the measurement. Although greedy algorithms are generally faster in practical implementations than the techniques based on convex relaxations, the common computational cost still scales as $O(\text{poly}(N))$ for both noiseless and noisy settings, with a few exceptions that incur near-linear run-time $O(N \log N)$ (e.g., StOMP algorithm [20]). Besides, the measurement matrix is typically stated in terms of MC conditions[7] which require $O(K^2)$ measurements. This phenomenon is commonly referred to as the square-root bottleneck, where the limit of sparsity for successful recovery is on the order of $K = O(\sqrt{N})$ even if measurement matrices achieving the MC lower bound are used (i.e. the Welch bound [23]).

Coding-theoretic Approach: This class of methods borrows the insights from modern coding theory to facilitate measurement designs and recovery algorithms. Compressed sensing measurement designs have been extensively studied from a coding-theoretic lens. For instance, [24, 25] exploits the algebraic properties of Reed-Muller codes and Delsarte Goethals codes, [26] uses a generalization of Reed-Solomon codes, and [27] establishes the connection between the channel decoding problem and the convex relaxation approach. Meanwhile, a multitude of work has emerged based on *expander graphs* [28,29], a popular design element in modern coding theory, which achieves near-linear time[8] recovery $O(N \log(N/K))$ using $O(K \log(N/K))$ measurements in the noiseless setting. Motivated by expander-based designs, researchers have proposed greedy approximation schemes that achieve similar costs, such as Expander Matching Pursuit (EMP) [31] and Sparse Matching Pursuit (SMP) [32]. Last but not least, there is a wide range of recovery algorithms using modern decoding principles such as list decoding [33, 34], efficient error-correcting codes via message passing [35–37]. Recently, [38] uses spatially-coupled LDPC codes in the measurement design and an approximate message passing decoding algorithm for recovery, which achieves the

[7] The measurement scaling of $O(K \log(N/K))$ for greedy pursuit methods exists under relaxed settings (e.g. bounded noise scenarios or probabilistic guarantees [21]). While there are some results on OMP based on the RIP, it is still ongoing work (see [22]).

[8] Using the same measurement design based on expanders, ℓ_1-minimziation can also be shown to achieve similar performance in polynomial time [30].

information-theoretically optimal measurement cost $O(K)$ given by [39] under a source coding setting. However, the decoding complexity remains polynomial time in N. Particularly relevant to our work are those based on fast verification-based decoding [35, 40, 41], where the sparse coefficients are solved by verifying and correcting each symbol iteratively. The Sudocodes design [35] introduces a noiseless scheme with $O(K \log N)$ measurements and sub-linear time computations $O(K \log K \log N)$ through a two-part verification decoding procedure. Further, [40] proposes a general high rate LDPC design with applications in compressed sensing, which provably provides guarantees for a broad class of measurement matrices under verification-based decoding, where the Sudocodes [35] is mentioned as a special case therein. Further, [42] proposed an algorithm that achieves a sample complexity of $O(K \log N \log \log N)$ and run-time $O(\text{poly}(K \log N))$ using a well-designed measurement matrix based on the proposed "summary-based" structure. Although our design shares certain elements in terms of the code properties being used, our approach differs significantly in designing the verification decoding schemes to achieve *sub-linear* time both *in the absence* and *presence* of noise, as well as the associated performance analysis.

Group Testing and Sketching: This class of methods exploit linear "sketches" of data for sparsity pattern recovery in *group testing* [8] and *data stream computing* [43]. In group testing, the common scenario is that we need to devise a collection of tests to find K anomalous items from N total items, where the typical goal is to recover the *support* of the underlying sparse vector and minimize the number of tests performed (measurements taken) [44]. In particular, [45] develops a compressed sensing design using group testing principle with $O(K \log^2 N)$ measurements and $O(K \log^2 N)$ operations. On the other hand, the goal of data stream computing is to maintain a short linear sketch of the network flows for approximating the sparse vector with some distortion measure. Examples include the count-min/count-sketch methods [46] and so on. Typical results in this bulk of literature require $O(K \log(N/K))$ measurements and near-linear time $O(N \log N)$ (see [5]). While there is a subset of sketching algorithms that achieve sub-linear time with $O(K \log(N/K))$ and $O(K \log^{O(1)} N)$ operations [47–49], these results typically provide *constant* failure probability guarantees for noiseless[9] measurements and sparse approximation instead of support recovery.

With a few exceptions, most of these sparse recovery results have been predominantly developed for sparse approximation under the ℓ_2/ℓ_1-norm or ℓ_1/ℓ_1-norm approximation error metrics[10], with a relatively much lower coverage of support recovery [5, 16, 22, 39, 50]. Meanwhile, necessary and sufficient conditions for support recovery have been studied in different regimes under various distortion measures using optimal decoders [51–55], ℓ_1-minimization methods [7, 56] and greedy methods [57]. For example, it is shown in [53] that $O(K \log(N/K))$ measurements are sufficient and necessary for support recovery when the measurement matrix consists of independent identically distributed (i.i.d.) Gaussian entries under Gaussian noise. Similar conditions under other signal and measurement models are also reported in [58–60]. Nonetheless, constructive recovery schemes that specifically target *support recovery* are relatively scarce [46, 59, 61], especially those that come with near optimal measurement costs and sub-linear complexity. In the following, we present the main idea of how to achieve our sub-linear time support recovery with near-optimal measurements in details.

3 Main Idea of Compressed Sensing using Sparse-Graph Codes

In this section, we present our design philosophy depicted in Fig. 1 with more details, and describe the main idea of our measurement design and recovery algorithm through a simple example.

3.1 Design Philosophy

As stated in the introduction, our design philosophy is depicted in Fig. 1, where we use different colors to distinguish different entries in the sparse vector, namely, we choose *red, green* and *blue* respectively for the non-zero

[9] Although sketching algorithms are not derived specifically to address noisy measurements, they could potentially be quite robust to various forms of noise.

[10] ℓ_p/ℓ_q-norm guarantees refer to the error metrics measured with respect to the best K-term approximation error $\|\mathbf{x}_K - \mathbf{x}\|$ (i.e., the vector $\mathbf{x}_K$ is the best K-term approximation containing the K most significant entries in the sparse vector $\mathbf{x}$), where the recovered sparse signal $\widehat{\mathbf{x}}$ satisfies $\|\widehat{\mathbf{x}} - \mathbf{x}\|_p \leq \kappa \|\mathbf{x}_K - \mathbf{x}\|_q$ for some absolute constant $\kappa > 0$.

entries, and *white* for zero entries. A conventional design in compressed sensing is to generate weighted linear measurements of the sparse vector through a carefully designed *measurement matrix* [6], e.g. popularly based on random independent identically distributed (i.i.d.) sub-Gaussian entries. In this example, all the entries of the measurement matrix are colored in *grey* to indicate some arbitrary design and the corresponding measurements are some generic mixtures of *red, green* and *blue*, as shown in Fig. 1-(a).

We design the measurement matrix by sparsifying each row of the measurement matrix with zero patterns guided by sparse-graph codes, indicated by the *white* spots in Fig. 1-(b). This new measurement matrix leads to a different set of measurements, where some contain single colors and some contain their mixtures. In this example, the measurements become separately colored with *red*, a *mixture of red and blue* and a *mixture of blue and green* as shown on the left of Fig. 1-(b). If the decoder knows that the first measurement contains a single *red* color, it can peel off its contribution from the *mixture of red and blue* in the second measurement, which forms a new measurement containing a single *blue* color. Similarly, the blue color can be peeled off from the third measurement such that the *green* color is decoded. In other words, our design can efficiently isolate the measurements that contain a single color, iteratively peel them off from their mixtures in other measurements, and continue this process until all the colors in the sparse vector are recovered.

Next, we illustrate the principle of our recovery algorithm by connecting support recovery with sparse-graph decoding using an "oracle" (described below). Then, using the insights gathered from the oracle-based decoding algorithm, we explain how we can get rid of the "oracle" using the same example.

3.2 Oracle-based Sparse-Graph Decoding

Consider a simple illustration consisting of a sparse signal $\mathbf{x}$ of length $N = 20$ with $K = 5$ non-zero coefficients $x[1] = 1$, $x[3] = 4$, $x[5] = 2$, $x[10] = 3$ and $x[13] = 7$. To illustrate the principle of our recovery algorithm, we construct a bipartite graph with 20 left nodes and 9 right nodes. The left and right nodes are referred to as the *variable nodes* and *check nodes* respectively in the language of sparse-graph codes. The graph has the following properties:

- Each *variable node* (left) labeled with k is assigned a value $x[k]$ for $k \in [N]$;
- Each *variable node* (right) is connected to the *check nodes* according to the *sparse* bipartite graph[11] in Fig. 2;
- Each *check node* (right) labeled with r is assigned a value y_r equal to the complex sum of its left neighbors, similar to the parity-check constraints of the LDPC codes.

Now we briefly introduce how this bipartite graph helps us recover the 20-length sparse signal $\mathbf{x}$ on the variable nodes from the 9 measurements associated with the check nodes:

$$
\begin{aligned}
y_1 &= y_7 = y_9 = 0, \\
y_2 &= x[1] + x[5] + x[13], \\
y_3 &= x[10], \\
y_4 &= x[3], \\
y_5 &= x[5] + x[10], \\
y_6 &= x[1], \\
y_8 &= x[3] + x[13].
\end{aligned}
$$

Depending on the connectivity of the sparse bipartite graph, we categorize the measurements associated with the check nodes into the following types:

1. **Zero-ton**: a check node is a zero-ton if it does not involve any non-zero coefficient (e.g., the color *blue* in Fig. 2).

[11] Since the values of the check nodes are not affected by the variable nodes carrying zero coefficients, we show only the edges from the variable nodes with non-zero values $x[k] \neq 0$.

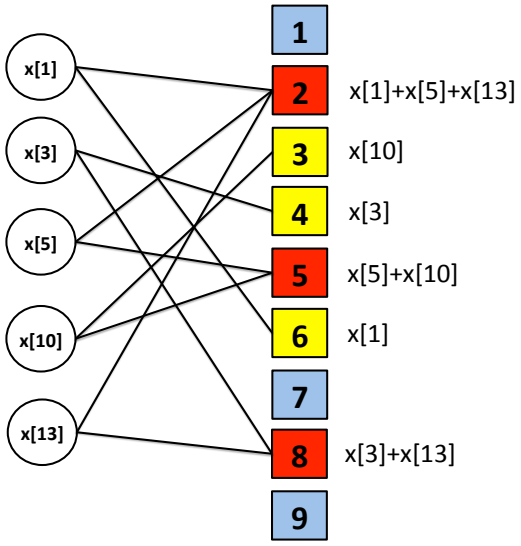


Figure 2: Example of a sparse bipartite graph consisting of 5 (non-zero) left nodes (variable nodes) with 2 edges randomly connected to the right nodes (check nodes). Blue color represents "zero-ton", yellow color represents "single-ton" and red color represents "multi-ton".

2. **Single-ton**: a check node is a single-ton if it involves only one non-zero coefficient (e.g., the color in *yellow* in Fig. 2). More specifically, we refer to the index k of the non-zero coefficient $x[k]$ and its associated value $x[k]$ as the **index-value pair** $(k, x[k])$ for that single-ton.

3. **Multi-ton**: a check node is a multi-ton if its value is the sum of more than one non-zero coefficient (e.g., the color *red* in Fig. 2).

To help illustrate our decoding algorithm, we assume that there exists an "oracle" that informs the decoder exactly which check nodes are *single-tons*. More importantly, the oracle further provides the index-value pair for that single-ton. In this example, the oracle informs the decoder that check nodes labeled 3, 4 and 6 are single-tons with index-value pairs $(10, x[10])$, $(3, x[3])$ and $(1, x[1])$ respectively. Then the decoder can subtract their contributions from other check nodes, forming new single-tons. Therefore generally speaking, with the oracle information, the peeling decoder repeats the following steps similar to [41, 62]:

Step (1) select all the edges in the bipartite graph with right degree 1 (identify single-ton bins);

Step (2) remove (peel off) these edges as well as the corresponding pair of variable and check nodes connected to these edges.

Step (3) remove (peel off) all other edges connected to the variable nodes that have been removed in **Step (2)**.

Step (4) subtract the contributions of the variable nodes from check nodes whose edges have been removed in **Step (3)**.

Finally, decoding is successful if all the edges are removed from the graph.

3.3 Getting Rid of the Oracle

Since the oracle information is critical in the peeling process, we proceed with our example and explain briefly how to obtain such information without an oracle. Clearly, we need more measurements to obtain such oracle information in its absence. Therefore, instead of simply assigning the simple *sum* to each check node, we assign a

vector-weighted sum to the check nodes, where each variable node (say k) is weighted by the k-th column of a **bin detection matrix** $\mathbf{S}$. For example, we can choose the bin detection matrix $\mathbf{S}$ as

$$\mathbf{S} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & W & W^2 & W^3 & W^4 & \cdots & W^{19} \end{bmatrix},$$

where $W = e^{\mathrm{i}\frac{2\pi}{N}}$ is the N-th root of unit with $N = 20$. Note that this is simply the first two rows of the 20×20 DFT matrix. In this way, each check node (say r) is assigned a 2-dimensional vector $\mathbf{y}_r = [y_r[0], y_r[1]]^T$ and we call each vector a **measurement bin**. For example, the measurements associated with check node 1, 2 and 3 become

$$\begin{aligned} \mathbf{y}_1 &= \mathbf{0}, \\ \mathbf{y}_2 &= x[1] \times \begin{bmatrix} 1 \\ W \end{bmatrix} + x[5] \times \begin{bmatrix} 1 \\ W^5 \end{bmatrix} + x[13] \times \begin{bmatrix} 1 \\ W^{13} \end{bmatrix}, \\ \mathbf{y}_3 &= x[10] \times \begin{bmatrix} 1 \\ W^{10} \end{bmatrix}. \end{aligned}$$

Now with these bin measurements, one can effectively determine if a check node is a zero-ton, a single-ton or a multi-ton. Although this procedure is formally stated in Section 6.1 in our noiseless recovery results, here as an illustration, we go through the procedures for check nodes 1, 2 and 3:

- **zero-ton bin**: consider the zero-ton check node 1. A zero-ton check node can be identified easily since the measurements are all zero

$$\mathbf{y}_1 = \mathbf{0}. \tag{4}$$

- **single-ton bin**: consider the single-ton check node 3. A single-ton can be verified by performing a simple "ratio test" of the two dimensional vector:

$$\begin{aligned} \widehat{k} &= \frac{\angle y_3[1]/y_3[0]}{2\pi/20} = 10, \\ \widehat{x}[\widehat{k}] &= y_3[0] = 3. \end{aligned}$$

  Another unique feature is that the measurements would have identical magnitudes $|y_3[0]| = |y_3[1]|$. Both the ratio test and the magnitude constraints are easy to verify for all check nodes such that the index-value pair is obtained for peeling.

- **multi-ton bin**: consider the multi-ton check node 2. A multi-ton can be easily identified by the ratio test

$$\widehat{k} = \frac{\angle y_2[1]/y_2[0]}{2\pi/20} = 12.59.$$

  Furthermore, the magnitudes are not identical $|y_2[0]| \neq |y_2[1]|$. Therefore, if the ratio test does not produce a non-zero integer and the magnitudes are not identical, we can conclude that this check node is a multi-ton.

This simple example shows how the problem of recovering the K-sparse signal $\mathbf{x}$ can be cast as an instance of sparse-graph decoding. Note that the sparse bipartite graph in this example only shows the idea of peeling decoding, but does not guarantee successful recovery for an arbitrary signal. Furthermore, this example also suggests that it is possible to obtain the index-value pair of any single-ton without the help of an "oracle" through a properly chosen bin detection matrix. We will address later how to construct sparse bipartite graphs to guarantee successful decoding (Section 5) and how to choose appropriate bin detection matrices for different schemes. In the following, we first present our general measurement design in Section 4, which is the cornerstone of our compressed sensing framework.

4 Measurement Design

Before delving into specifics, we define the *row-tensor* operator $\boxtimes$ to help explain our measurement design. Given a matrix $\mathbf{S} = [\mathbf{s}_0, \cdots, \mathbf{s}_{N-1}] \in \mathbb{C}^{M_2 \times N}$ and a matrix $\mathbf{H} = [\mathbf{h}_0, \cdots, \mathbf{h}_{N-1}] \in \mathbb{C}^{M_1 \times N}$, the row-tensor operation $\mathbf{H} \boxtimes \mathbf{S}$ is defined such that each row of $\mathbf{H}$ is augmented element-wise by performing a tensor product with each corresponding column in the matrix $\mathbf{S}$. Mathematically, the *row-tensor product* is a $M_1 M_2 \times N$ matrix given as

$$\mathbf{H} \boxtimes \mathbf{S} := \begin{bmatrix} \mathbf{h}_0 \otimes \mathbf{s}_0 & \cdots & \mathbf{h}_{N-1} \otimes \mathbf{s}_{N-1} \end{bmatrix},$$

where $\otimes$ is the standard Kronecker product. For example, let $\mathbf{H}$ be a sparse matrix with random coding patterns of $\{0, 1\}$ and $\mathbf{S}$ be chosen as the first two rows of a DFT matrix as in the simple example

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & W & W^2 & W^3 & W^4 & W^5 & W^6 \end{bmatrix} \tag{5}$$

with $W = e^{\mathrm{i}\frac{2\pi}{7}}$. Then the row-tensor product is given by

$$\mathbf{H} \boxtimes \mathbf{S} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & W & 0 & W^3 & 0 & W^5 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & W & 0 & W^3 & 0 & 0 & W^6 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & W^3 & W^4 & W^5 & W^6 \end{bmatrix}. \tag{6}$$

Since $\mathbf{H}$ has three rows of coding patterns, the product $\mathbf{H} \boxtimes \mathbf{S}$ contains three blocks of matrices, where each block is the corresponding sparsified version of $\mathbf{S}$ by the coding pattern in each row of $\mathbf{H}$.

Definition 1 (Measurement Matrix). *Let $M = RP$ for some positive integers R and P. Given a $R \times N$ coding matrix $\mathbf{H}$ and a $P \times N$ bin detection matrix $\mathbf{S}$, the $M \times N$ measurement matrix $\mathbf{A}$ is given by*

$$\mathbf{A} = \mathbf{H} \boxtimes \mathbf{S}, \tag{7}$$

where $\boxtimes$ is the row-tensor product, and the coding matrix and bin detection matrix are specified below.

- *The coding matrix $\mathbf{H} = [H_{r,n}]_{R \times N}$ is the $R \times N$ adjacency matrix of a bipartite graph $\mathcal{G}$ consisting of N left nodes $V_1 := [N]$ and R right nodes $V_2 := [R]$ with an edge set $\mathcal{E} := V_1 \times V_2$;*
- *The bin detection matrix $\mathbf{S} := [\mathbf{s}_0, \cdots, \mathbf{s}_{N-1}]$ is a $P \times N$ matrix whose explicit construction is given later for the Fourier family, binary family and random matrix family respectively.*

Corollary 1. *The measurement $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w}$ is divided into R measurement bins as $\mathbf{y} = [\mathbf{y}_1^T, \cdots, \mathbf{y}_R^T]^T$ with*

$$\mathbf{y}_r = \mathbf{S}\mathbf{z}_r + \mathbf{w}_r, \quad r = 1, \cdots, R \tag{8}$$

where $\mathbf{w}_r$ is the noise in the r-th measurement bin and $\mathbf{z}_r = [Z_r[0], \cdots, Z_r[N-1]]^T$ is a reduced sparse vector

$$Z_r[k] = \begin{cases} x[k], & k \in \mathcal{N}(r) \\ 0, & k \notin \mathcal{N}(r) \end{cases}, \tag{9}$$

and $\mathcal{N}(r)$ is the neighboring nodes of each right node $r = 1, \cdots, R$ in the bipartite graph $\mathcal{G}$.

Proof. The proof is straightforward and hence omitted. □

Since the vector $\mathbf{x}$ is by itself sparse on a support that may or may not overlap with the coding pattern given by the graph $\mathcal{G}$, the resulting equivalent sparse vector $\mathbf{z}_r$ in each bin r is even sparser with a reduced support $\mathsf{supp}\,(\mathbf{x}) \cap \mathcal{N}(r)$. If the coding pattern happens to make $\mathbf{z}_r$ a 1-sparse vector, we have a much easier problem to solve. Then we can use the recovered 1-sparse coefficient to recover other coefficients iteratively. Therefore, we need to distinguish the type of each bin in order to determine if $\mathbf{z}_r$ is 1-sparse, which can be regarded as a separate hypothesis in the presence of noise $\mathbf{w}_r$:

1. $\mathbf{y}_r$ is a **zero-ton** bin if $\mathsf{supp}\,(\mathbf{z}_r) = \varnothing$, denoted by $\mathbf{y}_r \sim \mathcal{H}_{\mathrm{Z}}$;
2. $\mathbf{y}_r$ is a **single-ton** bin with the index-value pair $(k, x[k])$ if $\mathsf{supp}\,(\mathbf{z}_r) = \{k\}$ for some $k \in [N]$ and $z[k] = x[k]$, denoted by $\mathbf{y}_r \sim \mathcal{H}_{\mathrm{S}}(k, x[k])$;
3. $\mathbf{y}_r$ is a **multi-ton** bin if $|\mathsf{supp}\,(\mathbf{z}_r)| \geq 2$, denoted by $\mathbf{y}_r \sim \mathcal{H}_{\mathrm{M}}$.

The spirit of divide-and-conquer is also manifested in this general design since the design of coding matrix ensures fast decoding by peeling, while the bin detection matrix ensures the correct detection of various bin hypotheses. These two designs are completely modular and can be designed independently depending on the applications. Now, given the above general measurement design, the following questions are of particular interests:

1. Given N left nodes and R right nodes, how to construct a bipartite graph that guarantees a "friendly" distribution of single-tons, zero-tons and multi-tons for successful peeling?
2. Given the sparsity K such that only K left nodes remain in the sparse bipartite graph, what is the minimum number of right nodes R to guarantee successful peeling?
3. How to choose the *bin detection matrix* $\mathbf{S}$ in general for providing the oracle information? Especially when the measurements are noisy?

In the following, we answer these questions in details and discuss the specific constructions for $\mathbf{H}$ and $\mathbf{S}$. In Section 5, we first present the peeling decoder analysis that guides the design of the bipartite graphs and the associated coding matrix $\mathbf{H}$, and then discuss the constructions of the bin detection matrix $\mathbf{S}$ for both noiseless and noisy scenarios in Section 6 and 7 respectively.

5 Sparse Graph Design and Peeling Decoder

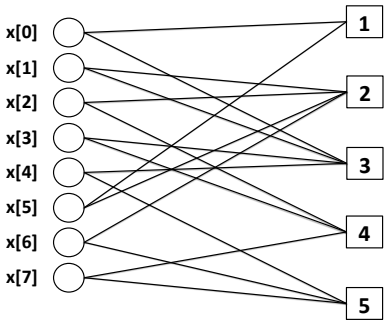
As mentioned above, the design of the coding matrix, or namely the sparse bipartite graph, is independent of the design of the bin detection matrix since they target different architectural objectives of the decoding algorithm. Simply put, the coding matrix (i.e. the sparse graph) can be designed assuming that there is an oracle present at decoding, while the bin detection matrix helps replace the oracle, which can be designed independently. Therefore, in this section we focus on the design of the coding matrix and study the sparse bipartite graphs that guarantee successful oracle-based decoding.

5.1 Sparse Graph Design for Compressed Sensing

The design of sparse bipartite graphs for peeling decoders has been studied extensively in the context of erasure-correcting sparse-graph codes [62, 63]. In this section, for simplicity we consider *the ensemble of left d-regular bipartite graphs* $\mathcal{G}_{\mathrm{reg}}^{N}(R, d)$ consisting of N left nodes (unknown coefficients $x[k]$ for $k \in [N]$) and R right nodes (compressed measurements $\mathbf{y}_r$ for $r = 1, \cdots, R$), where each left node $k \in [N]$ is connected to d right nodes $r = 1, \cdots, R$ uniformly at random and the number of right nodes is linear in the sparsity $R = \eta K$. We call η the *redundancy parameter*.

The coding matrix $\mathbf{H}$ constructed from the regular graph ensemble conforms with a random "balls-and-bins" model, where each row of $\mathbf{H}$ corresponds to a "bin" (i.e., right node) and each column of $\mathbf{H}$ corresponds to a "ball" (i.e., left node). If the (r, k)-th entry $H_{r,k} = 1$, then we say that the k-th ball is thrown into the r-th bin. In the

"balls-and-bins" model associated with the regular ensemble $\mathcal{G}_{\text{reg}}^{N}(R,d)$, each ball $k \in [N]$ is thrown uniformly at random to d bins. In the context of LDPC codes, the k-th coefficient $x[k]$ (variable node) appears in the parity check constraints in d right nodes (check nodes) chosen uniformly at random. For example, consider a smaller example with $N = 8$ left nodes and $R = 5$ nodes, where $\mathbf{x} = [x[0], \cdots, x[7]]^T$ is some generic signal vector. Then, an instance from the 2-regular ensemble $\mathcal{G}_{\text{reg}}^{8}(5,2)$ and the associated coding matrix $\mathbf{H}$ are shown in Fig. 3.



$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Figure 3: An example of the bipartite graph from the regular graph ensemble with $d = 2$ left degrees, consisting of $N = 8$ left nodes and $R = 5$ nodes, where the left nodes are labeled by the signal $\mathbf{x} = [x[0], \cdots, x[7]]^T$.

In our compressed sensing design, the sparse bipartite graph for peeling is the "pruned" graph after removing the left nodes with zero values. For example, if the signal is 4-sparse with non-zero coefficients $x[1]$, $x[4]$, $x[5]$ and $x[6]$, then the *"pruned" graph* is reduced to that in Fig. 4 on the right from the *full graph* on the left. Another example of a "pruned" graph has been shown in Fig. 2, which is associated with a 5-sparse signal and a left 2-regular graph with $N = 20$ left nodes and $R = 9$ right nodes.

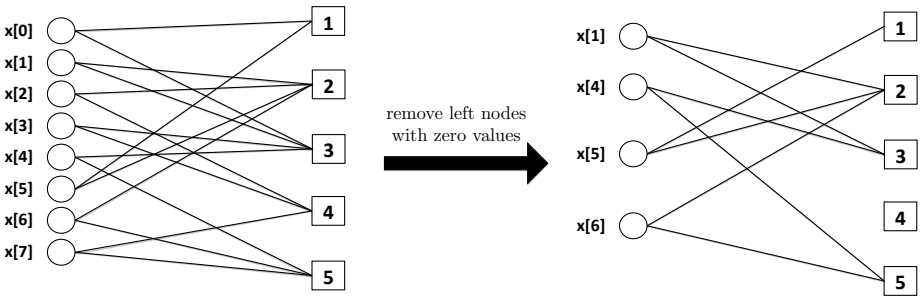


Figure 4: The "pruned" bipartite graph when the signal $\mathbf{x} = [x[0], \cdots, x[7]]^T$ is 4-sparse with non-zero coefficients $x[1]$, $x[4]$, $x[5]$ and $x[6]$.

Clearly, for compressed sensing of an arbitrary K-sparse signal $\mathbf{x}$, the *pruned* graph in Fig. 4, instead of the *full graph* in Fig. 3, determines the peeling decoder performance. However, the pruned graph depicted in Fig. 4 does not lead to successful decoding since the peeling is stuck with all multi-tons after removing the single-ton from right node $\#1$. The intuition is that there are 4 nodes on the left with degree 2 but only 5 nodes on the right, therefore there is a high probability for each right node to connect to more than one left node (i.e., in this case only one right node has degree 1). Therefore, in general, given the left degree d of the ensemble and the sparsity K, the graph needs to contain a sufficient number of right nodes to guarantee the success of the peeling decoder by choosing the redundancy parameter η properly. In the following, we study the peeling decoder performance over the pruned graphs from the regular ensemble $\mathcal{G}_{\text{reg}}^{N}(R,d)$ and shed light on how to specify the parameter η appropriately.

5.2 Oracle-based Peeling Decoder Analysis using the Regular Ensemble $\mathcal{G}_{\text{reg}}^{N}(R,d)$

In this section, we show that for the compressed sensing problem, if the redundancy parameter $\eta = R/K$ and the left regular degree d are chosen properly for the regular graph ensemble $\mathcal{G}_{\text{reg}}^{N}(R,d)$, then for an arbitrary K-sparse

signal $\mathbf{x}$, all the edges of the *pruned graph* can be peeled off in $O(K)$ peeling iterations *with high probability*. The formal statement is given in Theorem 4. In other words, we show that as long as the *full graph* is chosen properly, the *pruned graph* can lead to successful decoding with high probability for any given sparse signal. Our analysis is similar to the arguments in [62, 63] using the so-called *density evolution* analysis from modern coding theory, which tracks the average density[12] of the remaining edges in the pruned graph at each peeling iteration of the algorithm.

The proof techniques to analyze the peeling decoder in our framework are similar to those from [63] and [62], except that the graph we have is the "pruned" version with a sub-linear fraction K left nodes given adversarially by the input. Hence, this leads to some differences in the analysis from those in [62, 63], such as the degree distributions of the graphs (explained later) and the expansion properties of the graphs. As a result, we present an independent analysis here for our peeling decoder. In the following, we provide a brief outline of the proof elements highlighting the main technical components.

- **Density evolution**: We analyze the performance of our peeling decoder over a *typical graph* (i.e., cycle-free) of the ensemble $\mathcal{G}_{\text{reg}}^{N}(R, d)$ for a fixed number of peeling iterations i. We assume that a local neighborhood of every edge in the graph is cycle-free (tree-like) and derive a recursive equation that represents the average density of remaining edges in the pruned graph at iteration i.

- **Convergence to density evolution**: Using a Doob martingale argument as in [62] and [64], we show that the local neighborhood of most edges of a randomly chosen graph from the ensemble $\mathcal{G}_{\text{reg}}^{N}(R, d)$ is cycle-free with high probability. This proves that with high probability, our peeling decoder removes all but an arbitrarily small fraction of the edges in the pruned graph (i.e., the left nodes are removed at the same time after being decoded) in a constant number of iterations i.

- **Graph expansion property** for complete decoding: We show that if the sub-graph consisting of the remaining edges is an "expander" (as will be defined later in this section), and if our peeling decoder successfully removes all but a sufficiently small fraction of the left nodes from the pruned graph, then it removes all the remaining edges of the "pruned" graph successfully. This completes the decoding of all the non-zero coefficients in $\mathbf{x}$.

5.2.1 Density Evolution

Density evolution, a powerful tool in modern coding theory, tracks the average density of remaining edges that are not decoded after a fixed number of peeling iteration $i > 0$. We describe the concept of *directed neighborhood* of a certain edge in the pruned graph up to depth $\ell = 2i$. This concept is important in the density evolution analysis since the peeling of an edge in the i-th iteration depends solely on the removal of the edges from this neighborhood in the previous $i - 1$ iterations. The *directed neighborhood* $\mathcal{N}_{\mathsf{e}}^{\ell}$ at depth ℓ of a certain edge $e = (v, c)$ is defined as the induced sub-graph containing all the edges and nodes on paths $e_1, \cdots, e_\ell$ starting at a variable node v (left node) such that $e_1 \neq e$. An example of a directed neighborhood of depth $\ell = 2$ is given in Fig. 5.

To analyze the performance of the peeling decoder over the pruned graph, we need to understand the edge degree distributions on the left and right for the pruned graph. Let ρ_j be the fraction of edges in the pruned graph connecting to right nodes with degree j. Clearly, the total number of edges is Kd in the pruned graph since there are K left nodes in the pruned graph and each left node has degree d. Therefore, since the expected number of edges connected to right nodes with degree j can be obtained as $\mathsf{Pr}\left(\text{a right node has degree } j\right) Rj$, the fraction ρ_j can be obtained as

$$\rho_j = \frac{\mathsf{Pr}\left(\text{a right node has degree } j\right) Rj}{Kd} = \frac{j\eta}{d}\mathsf{Pr}\left(\text{a right node has degree } j\right), \tag{10}$$

[12]The density here refers to fraction of the remaining edges, or namely, the number of remaining edges divided by the total number of edges in the graph.

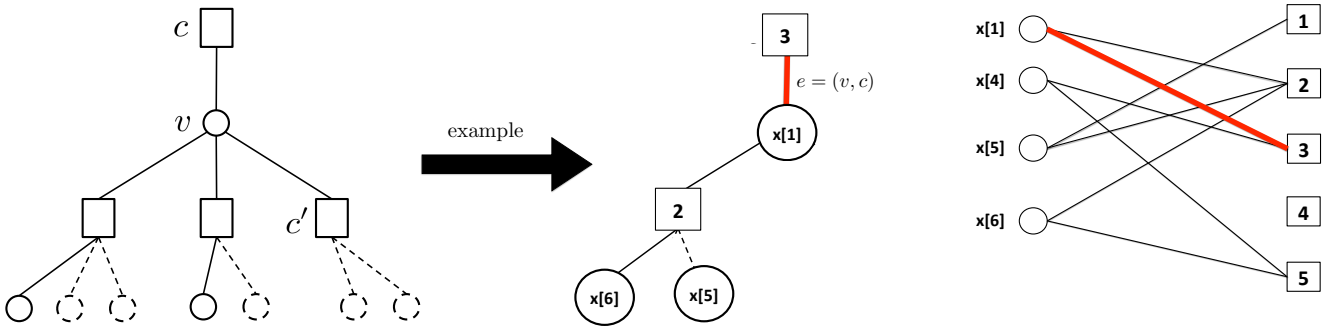


Figure 5: On the left sub-figure, we illustrate the directed neighborhood of depth 2 of an edge $e = (v, c)$, namely $\mathcal{N}_e^2$, while on the right we show this neighborhood for our example depicted in Fig. 4. The dashed lines on the left correspond to nodes/edges removed at the end of iteration $i-1$. The edge between v and c can be potentially removed at iteration i as one of the check nodes (right nodes) c' is a single-ton (it has no more variable nodes remaining at the end of iteration $i-1$). In our example, unlike the check node c' on the left, the edge $e = (x[1], 3)$ cannot be removed since the check node is still a multi-ton (i.e., $x[6]$ and $x[1]$ are still attached).

where we have used $R = \eta K$ and η is the redundancy parameter. According to the "balls-and-bins" model, the degree of a right node follows the binomial distribution $B(d/(\eta K), K)$ and can be well approximated by a Poisson variable as

$$\mathsf{Pr}\left(\text{a right node has degree } j\right) \approx \frac{(d/\eta)^j e^{-d/\eta}}{j!}. \tag{11}$$

As a result, the fraction ρ_j of edges connected to right nodes having degree j is

$$\rho_j = \frac{(d/\eta)^{j-1} e^{-d/\eta}}{(j-1)!}. \tag{12}$$

Now let us consider the local neighborhood $\mathcal{N}_e^{2i}$ of an arbitrary edge $e = (v, c)$ with a left regular degree d and right degree distribution given by $\{\rho_j\}_{j=1}^K$. If the sub-graph corresponding to the neighborhood $\mathcal{N}_e^{2i}$ of the edge $e = (v, c)$ is a *tree* or namely *cycle-free*, then the peeling procedures over different bins in the first i iterations (see Section 3.2) are independent, which can greatly simplify our analysis. Density evolution analysis is based on the assumption that this neighborhood is cycle-free (tree-like), and we will prove later (in the next subsection) that all graphs in the regular ensemble behave like a tree when N and K are large and hence the actual density evolution concentrates well around the density evolution result.

Let p_i be the probability of this edge being present in the pruned graph after $i > 0$ peeling iterations. If the neighborhood is a tree as in Fig. 6, the probability p_i can be written with respect to the probability p_{i-1} recursively.

$$p_i = \left(1 - \sum_j \rho_j (1 - p_{i-1})^{j-1}\right)^{d-1}, \quad i = 1, 2, 3, \cdots. \tag{13}$$

The term $\sum_j \rho_j (1-p_{i-1})^{j-1}$ can be approximated using the right degree generating polynomial

$$\rho(x) := \sum_j \rho_j x^{j-1} = e^{-(1-x)\frac{d}{\eta}}, \tag{14}$$

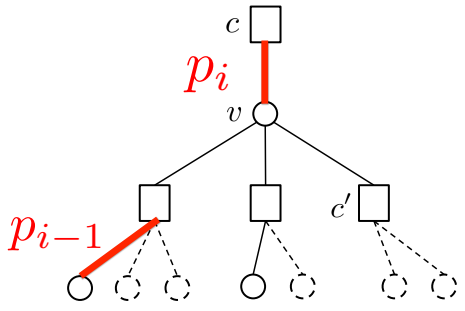


Figure 6: The schematic of density evolution in a local tree-like neighborhood.

where we have used (12) to derive the second expression. Therefore, the density evolution equation for our peeling decoder can be obtained as

$$p_i = f(p_{i-1}) = \left(1 - e^{-\frac{d}{\eta} p_{i-1}}\right)^{d-1}, \quad i = 1, 2, 3, \cdots. \tag{15}$$

An example of the density evolution with $d = 3$ and different values of η is given in Fig. 7. Clearly, the probability p_i can be made arbitrarily small for a sufficiently large but finite $i > 0$ as long as d and η are chosen properly. One can find the minimum value η for a given d to guarantee $p_i < p_{i-1}$, which is shown in Table 4. Due to lack of space we only show up to $d = 6$.

d	2	3	4	5	6
minimum η	2.0000	1.2219	1.2948	1.4250	1.5696

Table 4: Minimum value for η given the regular degree d according to density evolution.

Lemma 1 (Density evolution)**.** *Denote by $\mathcal{T}_i$ the event where the local $2i$-neighorhood $\mathcal{N}_e^{2i}$ of every edge in the graph is tree-like and let Z_i be the total number of edges that are not decoded after i (an arbitrarily large but fixed) peeling iterations. For any $\varepsilon > 0$, there exists a finite number of iteration $i > 0$ such that*

$$\mathbb{E}[Z_i|\mathcal{T}_i] = Kd\varepsilon/4, \tag{16}$$

where the expectation is taken with respect to the random graph ensemble $\mathcal{G}_{\text{reg}}^N(R, d)$ with the left regular degree d and the redundancy parameter $\eta = R/K$ chosen from Table 4.

Based on this lemma, we can see that if the pruned bipartite graph has a local neighborhood that is tree-like up to depth $2i$ for every edge, the peeling decoder on average peels off all but an arbitrarily small fraction of the edges in the graph. We prove this lemma below.

Proof. Let $Z_i^{(\text{e})} \in \{0, 1\}$ be the random variable denoting the presence of edge e after i iterations, thus

$$Z_i = \sum_{e=1}^{Kd} Z_i^{(\text{e})}. \tag{17}$$

The expected number of remaining edges over cycle-free graphs can be obtained as

$$\mathbb{E}\left[Z_i|\mathcal{T}_i\right] = \sum_{e=1}^{Kd} \mathbb{E}\left[Z_i^{(\text{e})}|\mathcal{T}_i\right] = Kdp_i, \tag{18}$$

where by definition $p_i = \mathsf{Pr}\left(Z_i^{(\text{e})} = 1|\mathcal{T}_i\right)$ is the *conditional probability* of an edge in the i-th peeling iteration conditioned on the event $\mathcal{T}_i$ studied in the density evolution equation (15). We are interested in the evolution of such probability p_i. In the following, we prove that for any given $\varepsilon > 0$, there exists a finite number of iterations $i > 0$ such that $p_i \leq \varepsilon/4$, which leads to our desired result in (16). □

5.2.2 Convergence to density evolution

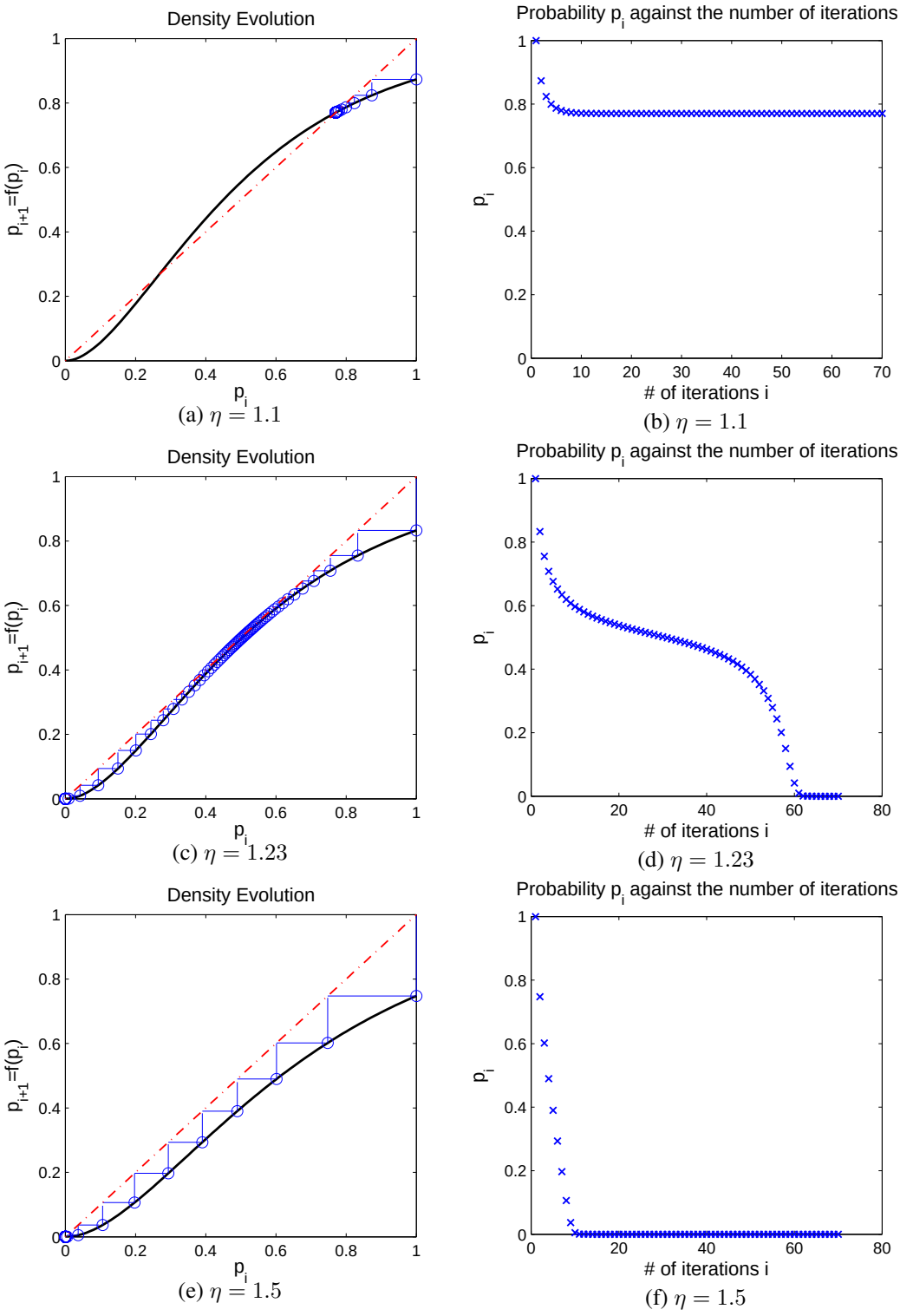
Given the mean performance analysis (in terms of the number of undecoded edges) over cycle-free graphs through density evolution, now we provide a *concentration analysis* on the number of the undecoded edges Z_i for *any graph from the regular ensemble* at the i-th iteration, by showing that Z_i converges to the density evolution result.

Lemma 2. *Over the probability space of all graphs from $\mathcal{G}_{\text{reg}}^N(R, d)$, let p_i be as given in the density evolution* (15)*. Given any $\varepsilon > 0$ and a sufficiently large K, there exists a constant $c_4 > 0$ such that*

$$\texttt{(i)} \qquad \mathbb{E}[Z_i] < Kd\varepsilon/2 \tag{19}$$

$$\texttt{(ii)} \qquad \mathsf{Pr}\left(|Z_i - \mathbb{E}[Z_i]| > Kd\varepsilon/2\right) \leq 2\exp\left(-c_4\varepsilon^2 K^{\frac{1}{4i+1}}\right) \tag{20}$$

$$\texttt{(iii)} \qquad \mathsf{Pr}\left(|Z_i - Kd\varepsilon/2| > Kd\varepsilon/2\right) \leq 2\exp\left(-c_4\varepsilon^2 K^{\frac{1}{4i+1}}\right) \tag{21}$$



(a) $\eta = 1.1$ (b) $\eta = 1.1$

(c) $\eta = 1.23$ (d) $\eta = 1.23$

(e) $\eta = 1.5$ (f) $\eta = 1.5$

Figure 7: The density evolution $f(p_i)$ and the probability p_i at each iteration i, where we have shown the case with $d = 3$ and $\eta = 1.1$, $\eta = 1.23$, $\eta = 1.5$. In the density evolution figures (a)-(c)-(e), the red line is the line $p_{i+1} = p_i$ while the black line is the actual density evolution recursion $f(p_i)$ against p_i. The blue circles that "zig-zag" between the red line and the black line are the specific p_i's that are achieved at each peeling iteration. It can be seen from (a) that when η is small (i.e. $\eta = 1.1$), the density evolution reaches a fixed point at around $p_i \approx 0.8$. On the other hand, when η is greater than the threshold 1.23 given by Table 4, the density p_i reaches 0 very quickly in (a) when $\eta = 1.5$. The values of p_i marked by the blue circles in (a)-(c)-(e) are further plotted against the peeling iterations i in (b)-(d)-(f), where in the case with $\eta = 1.5$ the density p_i approaches 0 after less than 10 iterations.

Proof. The details of the proof are given in Appendix A.1, but here we provide an outline of the proof. The *concentration analysis* is performed with respect to the number of the remaining edges for an *arbitrary graph from the ensemble* by showing that Z_i converges to the mean analysis result. This proof is done in two steps:

- **Mean analysis on general graphs from ensembles**: first, we use a counting argument similar to [64] to show that any random graph from the ensemble $\mathcal{G}^N_{\text{reg}}(R, d)$ behaves like a *tree* with high probability. Therefore, the expected number of remaining edges over all graphs can be made arbitrarily close to the mean analysis $|\mathbb{E}[Z_i] - \mathbb{E}[Z_i|\mathcal{T}_i]| < Kd\varepsilon/4$ such that

$$\mathbb{E}[Z_i] < Kd\varepsilon/2 \tag{22}$$

  as long as N and K are greater than some constants.

- **Concentration to mean by large deviation analysis**: then we use a Doob martingale argument as in [62] to show that the actual number of remaining edges Z_i concentrates well around its mean $\mathbb{E}[Z_i]$ with an exponential tail in K such that

$$\mathsf{Pr}\left(|Z_i - \mathbb{E}[Z_i]| > Kd\varepsilon/2\right) \leq 2\exp\left(-c_4\varepsilon^2 K^{\frac{1}{4i+1}}\right)$$

  for some constant $c_4 > 0$.

Then finally, it follows that $\mathsf{Pr}\left(|Z_i - Kd\varepsilon/2| > Kd\varepsilon/2\right) \leq 2\exp\left(-c_4\varepsilon^2 K^{\frac{1}{4i+1}}\right)$. □

5.2.3 Graph expansion property for complete decoding

From previous analyses, it has already been established that with high probability, our peeling decoder terminates with an arbitrarily small fraction of edges undecoded

$$Z_i < Kd\varepsilon, \quad \forall\varepsilon > 0, \tag{23}$$

where d is the left degree. In this section, we show that all the undecoded edges can be completely decoded if the sub-graph consisting of the remaining undecoded edges is a "good-expander". First, we introduce the concept of graph expanders.

Definition 2 (Graph Expander)**.** *A bipartite graph with K left nodes and regular left degree d is called a $(\varepsilon, 1/2)$-expander if for all subsets $\mathcal{S}$ of left nodes with $|\mathcal{S}| \leq \varepsilon K$, there exists a right neighborhood of $\mathcal{S}$ in the graph, denoted by $\mathcal{N}(\mathcal{S})$, that satisfies $|\mathcal{N}(\mathcal{S})| > d|\mathcal{S}|/2$.*

Lemma 3. *Consider the regular graph ensemble $\mathcal{G}^N_{\text{reg}}(R, d)$, then the pruned graph resulting from any given K-sparse signal $\mathbf{x}$ is an $(\varepsilon, 1/2)$-expander with probability at least $1 - O(1/K)$ for a sufficiently small constant $\varepsilon > 0$.*

Proof. See Appendix A.2. □

Without loss of generality, let the Z_i undecoded edges be connected to a set of left nodes $\mathcal{S}$. Since each left node has degree d, it is obvious from (23) that $|\mathcal{S}| \leq K\varepsilon$ with high probability. Note that our peeling decoder fails to decode the set $\mathcal{S}$ of left nodes if and only if there are no more single-ton right nodes in the neighborhood of $\mathcal{S}$. A sufficient condition for all the right nodes in $\mathcal{N}(\mathcal{S})$ to have at least one single-ton is that the average degree of the right nodes in the set $\mathcal{N}(\mathcal{S})$ is at most 2, which implies that $|\mathcal{S}|d/|\mathcal{N}(\mathcal{S})| \leq 2$ and hence $|\mathcal{N}(\mathcal{S})| \geq |\mathcal{S}|d/2$. Since we have shown in Lemma 3 that any pruned graph from the regular ensemble $\mathcal{G}^N_{\text{reg}}(R, d)$ is a $(\varepsilon, 1/2)$-expander with high probability such that $|\mathcal{N}(\mathcal{S})| \geq d|\mathcal{S}|/2$, there will be sufficient single-tons to peel off all the remaining edges.

Theorem 4. *Consider the ensemble $\mathcal{G}^N_{\text{reg}}(R, d)$ for our construction, the oracle-based peeling decoder peels off all the edges in the pruned graph in $O(K)$ iterations with probability at least $1 - O(1/K)$.*

Proof. The oracle-based peeling decoder fails when: 1) the number of remaining edges in the i-th iteration cannot be upper bounded as $Z_i < Kd\epsilon$ as in (21), or 2) the number of remaining edges can be upper bounded by $Z_i < Kd\epsilon$ as in (23) but the remaining sub-graph is not a $(\varepsilon, 1/2)$-expander. Event (1) occurs with an exponentially small probability so the total error probability is dominated by event (2). From Lemma 3, we have that event (2) occurs with probability $O(1/K)$, which approaches 0 asymptotically. Last but not least, since there are a total of $O(K)$ edges in the pruned graph, and there is at least one edge being peeled off in each iteration with high probability, the total number of iterations required to peel of the graph is $O(K)$. □

6 Noiseless Recovery: `FourierNoiseless` and `BinaryNoiseless`

In the noiseless setting, we consider a different graph ensemble to construct the coding matrix $\mathbf{H}$. If we use the regular graph ensemble $\mathcal{G}^N_{\text{reg}}(R, d)$ mentioned earlier to construct the coding matrix $\mathbf{H}$, the measurement cost is $M = RP$ with $R = \eta K$. Since each node has at least $P = 2$ measurements from the bin detection matrix $\mathbf{S}$, the measurement cost would be at least $2\eta K$. According to Table 4, given sufficiently large N and K, the minimum achievable η for successful decoding is $\eta = 1.23$ when $d = 3$, and hence the minimum measurement cost is at least $M \geq 2.46K$ if the regular ensemble is used. In order to achieve the minimum redundancy parameter $\eta \to 1$, bipartite graphs with *irregular* left degrees need to be considered. Hence, for the noiseless setting particularly, we construct the coding matrix $\mathbf{H}$ using an irregular graph ensemble rather than the regular graph ensemble $\mathcal{G}^N_{\text{reg}}(R, d)$ be better constants in our measurement costs.

In the irregular graph ensemble $\mathcal{G}^N_{\text{irreg}}(R, D)$, each left node has irregular left degrees $j = 2, \cdots, D+1$, where $D + 1$ is the maximum left degree. To describe the construction of the irregular graph ensemble, we use the left degree sequence $\{\lambda_j\}_{j=2}^{D+1}$, where λ_j is the fraction of edges[13] of degree j on the left[14]. For instance, the left degree sequence for the regular ensemble $\mathcal{G}^N_{\text{reg}}(R, d)$ is $\lambda_j = 1$ for $j = d$ and but 0 if $j \neq d$.

Definition 3 (Irregular Graph Ensemble $\mathcal{G}^N_{\text{irreg}}(R, D)$ for Noiseless Recovery)**.** *Given N left nodes and $R = (1 + \epsilon)K$ right nodes for an arbitrary $\epsilon > 0$, the edge set in the irregular graph ensemble $\mathcal{G}^N_{\text{irreg}}(R, D)$ is characterized by the degree sequence*

$$\lambda_j = \frac{1}{H(D)(j-1)}, \quad j = 2, \cdots, D+1 \tag{24}$$

where $D > 1/\epsilon$ and $H(D) = \sum_{j=1}^{D} 1/j$ is chosen such that $\sum_{j\geq 2} \lambda_j = 1$.

Theorem 5. *Consider the ensemble $\mathcal{G}^N_{\text{irreg}}(R, D)$ for our construction, the oracle-based peeling decoder peels off all the edges in the pruned graph in $O(K)$ iterations with probability at least $1 - O(1/K)$.*

Proof. See Appendix B. □

Given the coding matrix $\mathbf{H}$ constructed from the irregular ensemble, we now proceed to discuss the bin detection matrix constructions in the Fourier family and the binary family.

6.1 The `FourierNoiseless` Design

For the `FourierNoiseless` design, we choose the *bin detection matrix* $\mathbf{S}$ as

$$\mathbf{S} := \begin{bmatrix} 1 & \cdots & 1 & \cdots & 1 \\ 1 & \cdots & W^n & \cdots & W^{N-1} \end{bmatrix} \times \mathsf{diag}\left[F_0, F_1, \cdots, F_{N-1}\right], \tag{25}$$

where $W = e^{\mathrm{i}\frac{2\pi}{N}}$ is the N-th root of unity and F_k for $k \in [N]$ is a random variable drawn from some continuous distribution. The bin detection matrix is therefore the first 2 rows of the $N \times N$ DFT matrix with each column

[13] The graph is specified in terms of fractions of edges of each degree due to its notational convenience later on.

[14] An edge of degree j on the left (right) is an edge connecting to a left (right) node with degree j.

scaled by a random variable. This is similar to the example we used in Section 3.3, except for the random scaling on each column. We have briefly shown in Section 3.3 how to obtain the oracle information in the noiseless setting using a similar bin detection matrix. In the following, we restate the procedures more formally to be self-contained.

Using the two measurements in each bin $\mathbf{y}_r = [y_r[0], y_r[1]]^T$ for $r = 1, \cdots, R$, we perform the following tests to reliably identify the single-ton bins and obtain the correct index-value pair for any single-ton:

- **Zero-ton Test**: since there is no noise, it is clear that the bin is a zero-ton if $\|\mathbf{y}_r\|^2 = 0$.
- **Multi-ton Test**: The measurement bin is a multi-ton as long as $|y_r[1]| \neq |y_r[0]|$ and/or $\angle y_r[1]/y_r[0] \neq 0 \mod 2\pi/N$. The multi-ton test fails when the relative phase is a multiple of $2\pi/N$, which corresponds to the following condition according to the measurement model in (8)

$$\frac{y_r[1]}{y_r[0]} = \frac{\sum_{k\in[N]} H_{r,k}x[k]F_k e^{\mathrm{i}\frac{2\pi n}{N}}}{\sum_{k\in[N]} H_{r,k}x[k]F_k} = e^{\mathrm{i}\frac{2\pi\ell}{N}}, \quad \text{for some } \ell \in [N] \tag{26}$$

  where $H_{r,k}$ is the (r,k)-th entry in the coding matrix $\mathbf{H}$. Clearly, this event is measure zero under the continuous distribution of F_k for $k \in [N]$.
- **Single-ton Test**: After the zero-ton and multi-ton tests, if $|y_r[1]| = |y_r[0]|$ and $\angle y_r[1]/y_r[0] = 0 \mod 2\pi/N$, the measurement bin is detected as a single-ton with the index-value pair:

$$\widehat{k}_r = \frac{N}{2\pi}\angle\frac{y_r[1]}{y_r[0]}, \quad \widehat{x}[\widehat{k}_r] = y_r[0]/F_{\widehat{k}_r}. \tag{27}$$

  This gives us the index-value pair of the single-ton for peeling.

6.2 The `BinaryNoiseless` Design

Motivated by the Fourier family in the noiseless case, if the underlying bin is a single-ton, we encode the index-value pair information separately, where the value information is encoded with the first row (all-one vector) and the index information is encoded with the remaining rows. Recall that the `FourierNoiseless` design encodes the index information of the unknown coefficient as a single N-PSK symbol, and thus in the `BinaryNoiseless` design, we encode the index information as $\log_2 N$ BPSK symbols. More specifically, we define the length-$\log_2 N$ binary expansion vector $\mathbf{b}_k$ for some integer $k \in [N]$ and the *vector lifting* operation as

$$\mathbf{b}_k = \begin{bmatrix} b_k[1] \\ \vdots \\ b_k[p] \\ \vdots \\ b_k[\log_2 N] \end{bmatrix} \quad \text{and} \quad (-1)^{\mathbf{b}_k} = \begin{bmatrix} (-1)^{b_k[1]} \\ \vdots \\ (-1)^{b_k[p]} \\ \vdots \\ (-1)^{b_k[\log_2 N]} \end{bmatrix}, \quad k \in [N] \tag{28}$$

where $b_k[p]$ is the p-th bit of the $\log_2 N$-bit binary expansion of the column index k such that $k = b_k[p] \mod 2^p$.

Then we choose the *bin detection matrix* $\mathbf{S}$ as a concatenated matrix as

$$\mathbf{S} := \begin{bmatrix} 1 & 1 & \cdots & 1 \\ (-1)^{\mathbf{b}_0} & (-1)^{\mathbf{b}_1} & \cdots & (-1)^{\mathbf{b}_{N-1}} \end{bmatrix} \times \mathsf{diag}\left[F_0, F_1, \cdots, F_{N-1}\right], \tag{29}$$

where F_k for $k \in [N]$ is a random variable drawn from some continuous distribution. This choice can also be regarded as certain rows from the Hadamard matrix, which is the tensor products of 2-point DFT matrix. For example with $N = 8$, we can choose the matrix as

$$\mathbf{S} = \begin{bmatrix} F_0 & F_1 & F_2 & F_3 & F_4 & F_5 & F_6 & F_7 \\ -F_0 & -F_1 & -F_2 & -F_3 & F_4 & F_5 & F_6 & F_7 \\ -F_0 & -F_1 & F_2 & F_3 & -F_4 & -F_5 & F_6 & F_7 \\ -F_0 & F_1 & -F_2 & F_3 & -F_4 & F_5 & -F_6 & F_7 \end{bmatrix}.$$

In this way, the measurements at each bin is a $(\log_2 N+1)$-dimensional vector $\mathbf{y}_r = [y_r[0], \cdots, y_r[\log_2 N]]^T$. Then for each measurement bin, we perform the following tests to reliably identify the single-ton bins and obtain the correct index-value pair:

- **Zero-ton Test**: since there is no noise, it is clear that the bin is a zero-ton if $\|\mathbf{y}_r\|^2 = 0$.
- **Multi-ton Test**: The measurement bin is a multi-ton as long as the p-th measurement for $p = 1, \cdots, \log_2 N$ (normalized by the first measurement $p = 0$) is not either 0 or 1

$$\frac{y_r[p]}{y_r[0]} \notin \{-1, 1\}, \quad \forall p \in [\log_2 N].$$

The multi-ton test fails whenever

$$\frac{y_r[p]}{y_r[0]} = \frac{\sum_{k\in[N]} H_{r,k}F_k x[k](-1)^{b_k[p]}}{\sum_{k\in[N]} H_{r,k}F_k x[k]} \in \{-1, 1\}, \quad \text{for some } \ell \in [N] \tag{30}$$

where $H_{r,k}$ is the (r, k)-th entry in the coding matrix $\mathbf{H}$. Clearly, this event is also measure zero under the continuous distribution of F_k for $k \in [N]$.

- **Single-ton Test**: After the zero-ton and multi-ton tests, the measurement bin is detected as a single-ton if

$$\frac{y_r[p]}{y_r[0]} = (-1)^{b_k[p]} \in \{-1, 1\}, \quad \forall p \in [\log_2 N].$$

The index can be obtained by examining the sign of each measurement:

$$\mathsf{sgn}\left[y_r[p]\right] = \mathsf{sgn}\left[x[k]F_k(-1)^{b_k[p]}\right] = \mathsf{sgn}\left[F_k\right] \oplus \mathsf{sgn}\left[x[k]\right] \oplus b_k[p], \tag{31}$$

where the sign function is defined slightly different than the usual case:

$$\mathsf{sgn}\left[x\right] = \begin{cases} 1, & x < 0 \\ 0, & x \geq 0 \end{cases} \tag{32}$$

such that $x = |x|(-1)^{\mathsf{sgn}[x]}$. Note that we have $\mathsf{sgn}\left[y_r[0]\right] = \mathsf{sgn}\left[F_k\right] \oplus \mathsf{sgn}\left[x[k]\right]$, therefore the p-th bit $b_k[p]$ in the binary representation of the index k can be directly decoded as

$$\widehat{b}_k[p] = \mathsf{sgn}\left[y_r[p]\right] \oplus \mathsf{sgn}\left[y_r[0]\right]. \tag{33}$$

Finally, the index-value pair is obtained as follows for further peeling:

$$\widehat{k}_r = \sum_{p=1}^{\log_2 N} 2^{p-1} \times \widehat{b}_k[p], \quad \widehat{x}[\widehat{k}_r] = y_r[0]/F_{\widehat{k}_r}. \tag{34}$$

7 Noisy Recovery

In this section, we extend the Fourier and binary family designs for the noiseless setting in Section 6.1 to the noisy setting. We first discuss the construction of the coding matrix $\mathbf{H}$. Note that we can certainly use the irregular graph ensemble as in the noiseless case to design our coding matrix $\mathbf{H}$ for the noisy case as well, because it gives sharper measurement bound. However, since we are providing order-wise results for the measurement costs for noisy results, we consider the regular graph ensemble $\mathcal{G}^N_{\text{reg}}(R, d)$ for constructing $\mathbf{H}$ because of its simplicity. In the following, we discuss the constructions of the *bin detection matrix* $\mathbf{S}$ in the noisy setting.

In this setting, the role of the *bin detection matrix* $\mathbf{S}$ is critical in distinguishing the bin hypotheses (see Section 4) from one another. We propose three designs to perform bin detections that lead to different measurement costs and complexities, which are the `RandomNoisy` design, the `FourierNoisy` design and the `BinaryNoisy` design.

Proposition 1. *Given the bin detection matrix* $\mathbf{S}$ *in the* `RandomNoisy` *design (see Section 7.1),* `FourierNoisy` *design (see Section 7.3) and* `BinaryNoisy` *design (see Section 7.4), the failure probability* $\mathbb{P}_F$ *of the peeling decoder using the corresponding robust bin detection scheme is at most* $O(1/K)$.

Proof. See Appendix D. □

Since the procedures are the same for any measurement bin at any iteration, we drop the bin index r in (8) and use the italic font $\boldsymbol{y}$ to denote a generic bin measurement $\mathbf{y}_r$ using the following model

$$\boldsymbol{y} = \mathbf{S}\mathbf{z} + \mathbf{w} \tag{35}$$

for some bin detection matrix $\mathbf{S} = [\mathbf{s}_0, \cdots, \mathbf{s}_{N-1}]$ and some sparse vector $\mathbf{z}$. For example, in the first iteration at bin r, the sparse vector equals $\mathbf{z} = \mathbf{z}_r$ given in (8). As the peeling iterations proceed, the non-zero coefficients in $\mathbf{z}$ will be peeled off and potentially left with a 1-sparse coefficient. Therefore, at each iteration, we perform the bin detection routine to verify if $\mathbf{z}$ has become a 1-sparse signal (i.e. resolve the bin hypothesis) and obtain the associated index-value pair $(\widehat{k}, \widehat{x}[\widehat{k}])$.

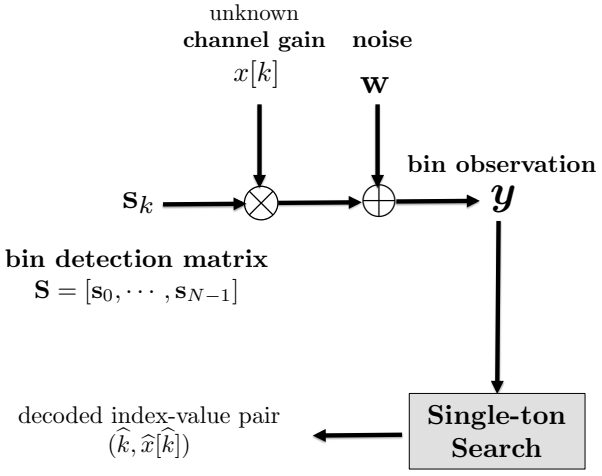


Figure 8: An illustration of single-ton search.

From the noiseless designs `FourierNoiseless` and `BinaryNoiseless`, we can see that the bin detection matrix $\mathbf{S}$ is in fact a properly chosen *codebook* for encoding the unknown value and location of the 1-sparse coefficient, where each column of $\mathbf{S}$ is a *codeword*. On one hand, the first row of both designs is an all-one vector, which captures directly the unknown value (but not the index). On the other hand, the `FourierNoiseless` design encodes the index information into a single N-PSK symbol (i.e. $W^k = e^{-\mathrm{i}\frac{2\pi k}{N}}$ for $k \in [N]$), while the `BinaryNoiseless` design encodes the index information into $\log_2 N$ BPSK symbols $\{\pm 1\}$.

To guarantee the success of peeling in the presence of noise, the decoder needs to figure out whether the bin measurement is contributed by *purely noise* (i.e. zero-ton), a single *codeword* (i.e. single-ton), or multiple *codewords* (i.e. multi-ton)? If the bin is indeed contributed by a single codeword, the decoder needs to decode the transmitted codeword received in noise (see Fig. 8). In the noiseless case, it is not difficult for the `FourierNoiseless` and `BinaryNoiseless` designs to deal with the aforementioned tasks. In the presence of noise, the codebook needs to be re-designed such that it can be robustly decoded.

7.1 The `RandomNoisy` Design

Instead of simple N-PSK or BPSK encoding in our noiseless designs, our `RandomNoisy` design follows the principle of using randomized linear codes to resolve different bin hypotheses and obtain the index-value pair.

Definition 4. *The* $P \times N$ *bin detection matrix* $\mathbf{S}$ *is constructed at random with* $P = O(\log N)$ *rows as an instance of the random matrix* $\mathbf{S}$ *from the following ensembles:*

- **Random Matrix Ensemble***: the ensemble of* $P \times N$ *matrices* $\mathbf{S} = [S_{i,j}]_{P\times N}$ *where* $S_{i,j}$*'s are i.i.d. sub-gaussian entries with zero mean and unit variance.*
- **Random DFT Ensemble***: the ensemble of* $P \times N$ *matrices where the* p*-th row is a random row* $\ell_p \in [N]$ *chosen from the* N*-point DFT matrix, and each column is weighted by i.i.d. Gaussian entries* $F_k \sim \mathcal{N}(0,1)$*:*

$$\mathbf{S} = \begin{bmatrix} 1 & W^{\ell_1} & \dots & W^{n\ell_1} & \dots \\ 1 & W^{\ell_2} & \dots & W^{n\ell_2} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & W^{\ell_P} & \dots & W^{n\ell_P} & \dots \end{bmatrix} \times \mathsf{diag}\left[F_0, F_1, \cdots, F_{N-1}\right]. \tag{36}$$

Step 1) zero-ton verification: *is it just noise?* Given the noisy measurements, the decoder needs to first detect whether there is indeed an existing signal component (if not, there is no decoding necessary). To do so, it is expected that the energy $\|\boldsymbol{y}\|^2$ to be small relative to the energy of a single-ton. Therefore, this idea is used to verify zero-tons:

$$\widehat{\mathcal{H}} = \mathcal{H}_{\mathrm{Z}}, \quad \text{if } \frac{1}{P}\|\boldsymbol{y}\|^2 \leq (1+\gamma)\sigma^2, \tag{37}$$

where $0 < \gamma < A_{\min}^2/\sigma^2$ is some chosen constant.

Step 2) single-ton search denoted by $\psi : \boldsymbol{y} \to (\widehat{k}, \widehat{x}[\widehat{k}])$: *if it is not purely noise, then by assuming that the underlying bin is a single-ton bin, which column of the bin detection matrix* $\mathbf{S}$ *is present in the measurement, and with what amplitude?* After verifying that the bin is not a zero-ton, this step estimates the index-value pair $(\widehat{k}, \widehat{x}[\widehat{k}])$ assuming that the underlying bin is a single-ton. Given the random codes, we employ a generalized likelihood test in the *single-ton search*, similar to the spirit of joint-typicality decoding. For each possible coefficient index k, we obtain the maximum likelihood (ML) of the coefficient as:

$$\alpha_k = \frac{\mathbf{s}_k^\dagger \boldsymbol{y}}{\|\mathbf{s}_k\|^2}. \tag{38}$$

Substituting the estimate of the coefficient a_k into the likelihood of the particular single-ton hypothesis in Proposition 1, we choose the index k that minimizes the residual energy:

$$\widehat{k} = \arg\min_k \ \|\boldsymbol{y} - \alpha_k \mathbf{s}_k\|^2. \tag{39}$$

The search is over the coding pattern in the r-th bin $k \in \mathcal{N}(r)$, which is known a priori. With the estimated index $\widehat{k}$, the coefficient is obtained by aligning it to the closest alphabet symbol in $\mathcal{X}$

$$\widehat{x}[\widehat{k}] = \min_{x \in \mathcal{X}} \left|\alpha_{\widehat{k}} - x\right|^2. \tag{40}$$

Step 3) single-ton verification: *does the single-ton search estimate explain sufficiently well the bin measurement?* This step confirms whether the bin is a single-ton with the estimated index-value pair via the residual test

$$\widehat{\mathcal{H}} = \mathcal{H}_{\mathrm{S}}(\widehat{k}, \widehat{x}[\widehat{k}]) \quad \text{if } \frac{1}{P}\left\|\boldsymbol{y} - \widehat{x}[\widehat{k}]\mathbf{s}_{\widehat{k}}\right\|^2 \leq (1+\gamma)\sigma^2. \tag{41}$$

Since there are a total of $R = \eta K$ bins and each bin has $P = O(\log N)$ measurements, the `RandomNoisy` design leads to a measurement cost of $M = \eta K P = O(K \log N)$. The measurement cost $O(K \log N)$ is near-optimal in the order sense with respect to the fundamental limits for support recovery studied in [53,58–60]. This is implied by information theory where random codes are oftentimes capacity-achieving, but the decoding complexity is high. More specifically, this scheme requires an exhaustive search over all elements in $\mathcal{X}$ and all indices $k \in \mathcal{N}(r)$ for some bin r. Since each left node k (i.e. "ball") out of N total possible left nodes is included in the set $\mathcal{N}(r)$ for the r-th bin independently with uniform probability $O(1/K)$ due to the "balls-and-bins" construction, this leads to an exhaustive search over $O(N/K)$ elements on average in each peeling iteration, where each element imposes a search complexity of $P = O(\log N)$ by the generalized likelihood ratio test. As a result, across all $O(K)$ peeling iterations, this results in a total complexity of $T = O(N/K) \times O(\log N) \times O(K) = O(N \log N)$.

7.2 Bin Detection Scheme Going below Linear Time?

Given the `RandomNoisy` design, the question to ask is: *is it possible to achieve the same performance with sub-linear time complexity*? We answer in the affirmative. Recall that the robust bin detection involves three steps:

1) zero-ton verification $\frac{1}{P}\|\boldsymbol{y}\|^2 \leq (1+\gamma)\sigma^2$;

2) single-ton search that estimates the index-value pair $(\widehat{k}, \widehat{x}[\widehat{k}])$;

3) single-ton verification $\frac{1}{P}\left\|\boldsymbol{y} - \widehat{x}[\widehat{k}]\mathbf{s}_{\widehat{k}}\right\|^2 \leq (1+\gamma)\sigma^2$.

The `RandomNoisy` design is a straightforward construction to guarantee high probability of success for all steps. However, it does not optimize its choice of codebook to facilitate step (2), which causes the high complexity.

To reduce the complexity without compromising the measurement cost, the spirit of divide-and-conquer also applies by using two codebooks, where one uses `RandomNoisy` design deals with step (1) and (2) for reliable verifications, while the other codebook (introduced next) deals with step (2) for fast and robust decoding. The design for this fast and robust decoding component is the key. We first present the `FourierNoisy` design that achieves a sub-linear complexity $T = O(K \log^{1.\dot{3}} N)$ with a measurement cost of $M = O(K \log^{1.\dot{3}} N)$. Then we bring the complexity down to $T = O(K \log N)$ without increasing the measurement cost $M = O(K \log N)$ using our `BinaryNoisy` design.

7.3 The `FourierNoisy` Design

Using the Fourier family, we show that if we impose an extra factor of $O(\log^{1/3} N)$ measurements to the random DFT ensemble of the `RandomNoisy` design, we can reduce the complexity from the near-linear regime $O(N \log N)$ to the sub-linear regime $O(K \log^{1.\dot{3}} N)$. Since the codebook for verification is chosen from the random DFT ensemble in Definition 4, to avoid repetition, we only discuss the codebook used for single-ton search here.

Definition 5 (Structured DFT Matrix). *Let $P = O(\log N)$ and $Q = O(\log^{1/3} N)$ where N is not a multiple*[15] *of 2. The bin detection matrix $\mathbf{S}$ consists of P sub-matrices $\{\mathbf{S}_p\}_{p\in[P]}$ of size $Q \times N$*

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_0 \\ \mathbf{S}_1 \\ \vdots \\ \mathbf{S}_{P-1} \end{bmatrix} \quad \textit{where} \quad \mathbf{S}_p = \begin{bmatrix} 1 & 1 & \cdots & 1 & \cdots \\ 1 & W^{2^p} & \cdots & W^{n2^p} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & W^{(Q-1)2^p} & \cdots & W^{n(Q-1)2^p} & \cdots \end{bmatrix}, \tag{42}$$

where $W = e^{\mathrm{i}\frac{2\pi}{N}}$ is the N-th root of unity and each sub-matrix $\mathbf{S}_p$ consists of Q consecutive 2^p-dyadically spaced rows[16] *from the $N \times N$ DFT matrix.*

With this bin detection matrix, the measurement at some single-ton bin can be obtained as $\boldsymbol{y} = \mathbf{S}\mathbf{z} + \mathbf{w}$ for some 1-sparse vector $\mathbf{z}$ with an index-value pair $(k, x[k])$. The measurements for single-ton search can then be divided into P clusters $\boldsymbol{y} = [\boldsymbol{y}_0^T, \cdots, \boldsymbol{y}_{P-1}^T]^T$,

$$\boldsymbol{y}_p = \mathbf{S}_p\mathbf{z} + \mathbf{w}_p, \quad p \in [P], \tag{43}$$

where each cluster $\boldsymbol{y}_p = [y_p[0], \cdots, y_p[Q-1]]$ corresponds to each sub-matrix $\mathbf{S}_p$. By definition, each cluster $\boldsymbol{y}_p$ contains the periodic samples $y_p[m]$ that are taken with a dyadic spacing 2^p for $q = 0, \cdots, Q-1$. Without loss of generality, let us consider the p-th block $\mathbf{S}_p$, which

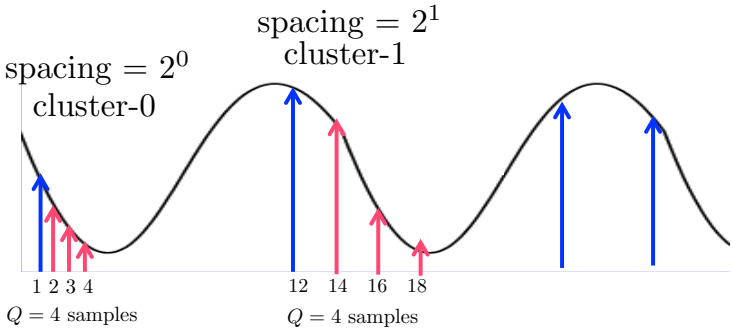


Figure 9: Example of the real/imaginary part of the observation obtained by the structured random DFT matrix in Definition 5, with $P = 4$ clusters and $Q = 4$ samples in each cluster.

[15] The base element 2 is chosen for simplicity, and can be generalized to any integer.

[16] The reason for choosing the dyadic spacing is that N is not a multiple of 2. Since N is not a multiple of 2, the dyadic nature does not lead to repetitive choices of the DFT rows due to the periodic wrap-around of the DFT matrix. Therefore, in general, the spacing can be chosen arbitrarily to be the powers of any number that does not factor N. This choice only changes the constants in our big-Oh statements, but not the scaling.

consists of Q consecutive rows (2^p-spaced) from the DFT matrix. If the underlying bin is in fact a single-ton with some index-value pair $(k, x[k])$, the measurements $\boldsymbol{y}_p$ correspond to the consecutive samples of a complex sinusoid

$$y_p[q] = x[k]e^{\mathrm{i}\frac{2\pi k\times 2^p}{N}q} + w_p[q], \quad q = 0, \cdots, Q-1 \tag{44}$$

with some frequency $\omega_p = 2\pi k/N \times 2^p$ and complex amplitude $x[k]$. Equivalently, the samples obtained by this structured DFT matrix can be visualized as clusters of samples from a complex sinusoid in Fig. 9.

Using the Fourier structure, estimating the unknown index k of a single-ton can be viewed as the estimation of a discrete frequency of a complex sinusoid given by each column of the DFT matrix

$$\omega = \frac{2\pi k}{N} \tag{45}$$

using spectral estimation methods that have been extensively studied in signal processing [65]. This is a systematic version of the ratio test for the noiseless case. A reliable way to estimate the frequency ω_p and coefficient is through an ML estimator, which however leads to high complexity. Therefore, we leverage the unbiased and efficient linear estimator developed in [65] and propose a successive spectral estimation scheme for single-ton search.

Lemma 4 (Frequency Estimation [65])**.** *Given the complex sinusoid in* (44) *with* $w[q] \sim \mathcal{CN}(0, \sigma^2)$ *and a finite signal-to-noise ratio* $|x[k]|^2/\sigma^2$*, the following estimator*

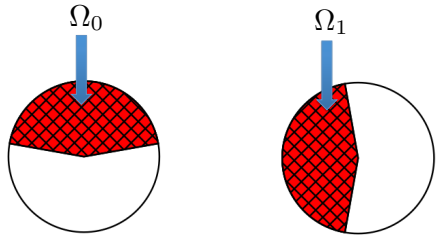
$$\widehat{\omega}_p = \sum_{q=0}^{Q-2} r_q \angle y_p^*[q]y_p[q+1] \tag{46}$$

with weights $r_q := \frac{3Q/2}{Q^2-1}\left(1 - \left[\frac{q-(Q/2-1)}{Q/2}\right]^2\right)$ *has an estimation error* $\Delta_p = \widehat{\omega}_p - \omega_p \sim \mathcal{N}\left(0, \frac{6}{Q(Q^2-1)}\frac{\sigma^2}{|x[k]|^2}\right)$.

Nonetheless, the index k can be estimated without ambiguity from the first cluster $\omega_0 = 2\pi k/N$, since there are 2^p possible choices of k that results in the same frequency $\omega_p = 2\pi k/N \times 2^p$. Therefore, the linear estimator in Lemma 4 would require using cluster-0 and require the resulting estimation error to be sufficiently small $\Delta_0 < 1/N$ with high probability, such that ω_0 can be obtained correctly within $[-\pi/N, \pi/N]$. From the distribution of the estimation error Δ_0, this implies a measurement scaling of $Q = O(N^{1/3})$, which is polynomial in N and undesirable for fast implementation. However, if we further exploit the multiple clusters and analyze the ambiguous pattern of each ω_p, we can improve the scaling dramatically by performing such estimation recursively over P clusters of $Q = O(\log^{1/3} N)$ measurements. This is what constitutes the sub-linear time single-ton search via successive estimation over multiple clusters, which is explained below.

Step 1: Cluster Estimation

According to Lemma 4, the measurement $\boldsymbol{y}_p$ in the p-th cluster provides an estimate $\widehat{\omega}_p$ of $2^p\omega$ modulo 2π. Let $\Omega_0 = (\omega_0 - \pi/2, \omega_0 + \pi/2)$ be the set of frequencies around the estimate ω_0 obtained by processing cluster-0, which we call the *certainty region*. It is known from Lemma 4 that with some probability $\lim_{Q\to\infty} \mathsf{Pr}\left(\omega \notin \Omega_0\right) \to 0$, the estimate will be outside of the certainty region. Similarly, for the second cluster, with some other probability $\lim_{Q\to\infty} \mathsf{Pr}\left(2\omega \notin \Omega_1\right) \to 0$, the estimate will be outside of the certainty region $\Omega_1 = (\omega_1 - \pi/2, \omega_1 - \pi/2)$. The certainty regions are illustrated for the first two clusters in Fig. 10, whereas the same holds for other clusters over $p \in [P]$.



(a) $\widehat{\omega}_0 = \omega + \Delta_0$ (b) $\widehat{\omega}_1 = 2\omega + \Delta_1$

Figure 10: Example of cluster estimations over the first two clusters, where the shape of the red pie indicates the certainty region Δ_i for the cluster estimate $\widehat{\omega}_i$ of the frequency $2^i\omega$.

Step 2: Unwrapping

We define the *unwrap* procedure to infer ω from each estimate of $2^p\omega$ in the p-th cluster. Since $\omega = 2\pi k/N$ for some $k \in [N]$, the factor of 2^p essentially maps 2^p frequencies to the same frequency $\omega_p = 2^p\omega$ (modulo 2π), one of which is the ground truth ω. Therefore, the certainty region Ω_p can be unwrapped into 2^p much smaller slices of *unwrapped certainty regions* of width $\pi/(2^p)$ over the unit circle. We denote the set of all *unwrapped certainty regions* as $\Omega_p/2^p$. Note that the certainty region Ω_p and the unwrapped certainty region $\Omega_p/2^p$ have the same cardinality of Ω_p, but their occupancies on the unit circle is different, as shown in Fig. 11.

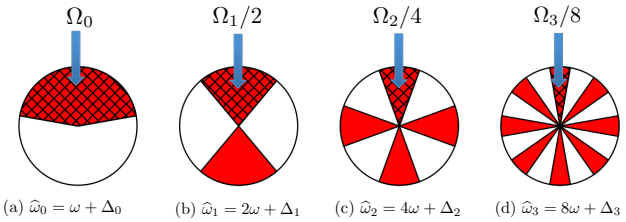


Figure 11: Frequency unwrapping over multiple clusters, from left (cluster 0) to right (cluster 3). Cluster-1 provides the unwrapped uncertainty region consisting of 2 slices due to the frequency estimate of 2ω, cluster-2 provides unwrapped certainty region consisting of 4 slices due to the frequency estimate of 4ω, cluster-3 provides unwrapped certainty region consisting of 8 slices due to the frequency estimate of 8ω.

Step 3: Estimate Fusion

Based on the unwrapped certainty regions over multiple clusters, the frequency estimate $\widehat{\omega} = 2\pi\widehat{k}/N$ is refined successively by intersecting the C unwrapped certainty regions

$$\Omega := \bigcap_{p=0}^{P-1} \Omega_p/2^p. \tag{47}$$

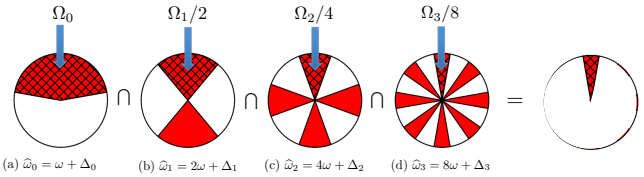


Figure 12: Estimate fusion over multiple clusters, from left (cluster 0) to right (cluster 3), which leads to the intersection of the reduced slice of pie on the right.

Since $|\Omega_0| = \pi$ and N is sufficiently large, the intersection of the first two clusters leads to a halved uncertainty interval $|\Omega_0 \bigcap \Omega_1/2| \approx \pi/2$ and thus intuitively, the size of the set Ω shrinks down exponentially with the number of clusters P, as shown in Fig. 12.

As mentioned in the estimate fusion step, the successive intersection over multiple clusters sharpens the certainty set Ω in terms of its cardinality, which shrinks exponentially with respect to the number of clusters. Clearly, as long as we have $P = O(\log N)$ clusters, the cardinality of the final certainty set Ω can be made as small as $|\Omega| \leq \pi/N$ such that it contains only $O(1)$ possible discrete frequencies ω with high probability. Furthermore, with some probability $\lim_{Q\to\infty} \mathsf{Pr}\left(\omega \notin \Omega_p/2^p\right) \to 0$ that vanishes exponentially Q according to Lemma 4, the estimate falls outside the unwrapped certainty region, the probability of the estimate lying outside at least one of the unwrapped certainty regions can be upper bounded by a union bound over P clusters, which remains vanishing to 0 asymptotically in Q irrespective of P. As shown in Appendix G.2, as long as each cluster has $Q = O(\log^{1/3} N)$ measurements, the ground truth will fall in the final certainty region Ω with probability at least $1 - O(1/N)$. As a result, we can easily find an estimate $\widehat{k}$ with $2\pi\widehat{k}/N \in \Omega$ over these few possible candidates via (39) and (40).

7.4 The `BinaryNoisy` Design

From our `FourierNoisy` design, we have seen a significant reduction of complexity from the `RandomNoisy` design. The key to our low complexity `FourierNoisy` design in estimating the unknown index k is through fast spectral estimation. This is enabled by introducing certain coding redundancy into the random DFT ensemble codebook from the `RandomNoisy` design, which appends extra structured rows (i.e. dyadically spaced consecutive rows) from the DFT matrix to each row of the random DFT matrix. However, this comes with an extra cost of $Q = O(\log^{1/3} N)$ measurements for each bin. In the case of the binary family, we can show that if the codebook redundancy is introduced to our `BinaryNoiseless` design according to error-correcting codes, the extra factor of $O(\log^{1/3} N)$ is avoidable and we can achieve $O(K \log N)$ for both measurements and complexity. Similar to the

FourierNoisy design, the codebook for verification is chosen from the random matrix ensemble in Definition 4, we only discuss the codebook used for single-ton search here to avoid repetition.

For the single-ton search in the BinaryNoisy design, we first estimate the unknown index k and then obtain the unknown value using (40). For notational convenience, we denote the binary bin detection matrix as

$$\mathbf{S} = (-1)^{\mathbf{C}}, \tag{48}$$

where $\mathbf{C} = [\mathbf{c}_0, \cdots, \mathbf{c}_{N-1}]$ is some binary matrix $\{0,1\}^{P\times N}$ and $(-1)^{\mathbf{C}}$ is the *matrix lifting* operation similar to the vector lifting operation in (28). Given a single-ton bin with the index-value pair $(k, x[k])$ in the presence of noise, the measurements obtained from $\mathbf{S}$ can be written as

$$\boldsymbol{y} = x[k](-1)^{\mathbf{c}_k} + \mathbf{w} \tag{49}$$

where $\mathbf{w}$ is the i.i.d. Gaussian noise vector with variance σ^2. Different from the noiseless case, the sign of the measurement $\mathsf{sgn}\,[\boldsymbol{y}]$ is randomly flipped with respect to the signs of $x[k]$, as stated below.

Proposition 2. *Given a single-ton bin with $(k, x[k])$, the sign of the bin measurement vector $\mathsf{sgn}\,[\boldsymbol{y}]$ satisfies*

$$\mathsf{sgn}\,[\boldsymbol{y}] = \mathbf{c}_k \oplus \mathsf{sgn}\,[x[k]] \oplus \mathbf{e}, \quad p \in [P], \tag{50}$$

where $\mathbf{e}$ is a vector containing P Bernoulli variables with probability upper bounded as $\mathbb{P}_{\mathsf{e}} = e^{-\frac{|x[k]|^2}{2\sigma^2}}$.

Proof. See Appendix C. □

From Proposition 2, it can be seen that the sign of the measurement can be viewed as corrupted bits received over a binary symmetric channel (BSC). Recall that in the BinaryNoiseless design, the matrix $\mathbf{C}$ is simply the binary expansion matrix

$$\mathbf{B} = \begin{bmatrix}\mathbf{b}_0 & \mathbf{b}_1 & \cdots & \mathbf{b}_{N-1}\end{bmatrix} \tag{51}$$

where $\mathbf{b}_k = [\cdots, b_k[p], \cdots]^T$ is given by the binary expansion in (28). Therefore, the bit pattern $\mathbf{b}_k$ can be obtained easily from the sign of the measurement vector $\mathsf{sgn}\,[\boldsymbol{y}]$. Now, the single-ton search problem becomes a decoding problem over a BSC. The goal is to decode the bit sequence $\mathbf{b}_k$ from $\{\mathsf{sgn}\,[y[p]]\}_{p\in[P]}$ with an unknown sign flip $\mathsf{sgn}\,[x[k]]$. In order to decode $\mathbf{b}_k$ reliably over the BSC, we consider using linear channel codes with a certain block length $P = O(\log N)$ to achieve vanishing error probability.

Definition 6 (Coded Binary Matrix)**.** *For any given N and $n = \log_2 N$, let $\mathbf{B}$ be the $n \times N$ binary expansion matrix in* (51)*. Then the bin detection matrix $\mathbf{S} = (-1)^{\mathbf{C}}$ is chosen as*

$$\mathbf{C} = \mathbf{G}\mathbf{B}, \tag{52}$$

where $\mathbf{G}$ is a $P \times n$ generator matrix for some rate-$R(\beta)$ linear code with a block length $P = n/R(\beta)$ and the rate $R(\beta)$ is determined by the achievable minimum distance βP such that $\beta > \mathbb{P}_{\mathsf{e}}$.

The sign of the resulting measurement vector of some single-ton bin $(k, x[k])$ becomes

$$\mathsf{sgn}\,[\boldsymbol{y}] = \mathbf{G}\mathbf{b}_k \oplus \mathsf{sgn}\,[x[k]] \oplus \mathbf{e}. \tag{53}$$

If $\mathsf{sgn}\,[x[k]]$ is known a priori, it is trivial that we have

$$\mathsf{sgn}\,[\boldsymbol{y}] \oplus \mathsf{sgn}\,[x[k]] = \mathbf{G}\mathbf{b}_k \oplus \mathbf{e}. \tag{54}$$

There are $P = O(n)$ coded bits of the n information bits in $\mathbf{b}_k$, and the BSC flips each bit with probability at most $\mathbb{P}_{\mathsf{e}}$. As a result, if the linear code $\mathbf{c}_k = \mathbf{G}\mathbf{b}_k$ has a minimum distance of βP, it is obvious that the information bits $\mathbf{b}_k$ can be decoded with exponentially decaying probability of error as long as $\beta > \mathbb{P}_{\mathsf{e}}$ (see Appendix G.3).

There exist many codes that satisfy the minimum distance properties, but the challenge is the decoding time. It is desirable to have decoding time linear in the block length $P = O(n)$ so that the sample complexity and computational complexity can be maintained at $O(n)$ for each bin, same as the noiseless case. Therefore, we can achieve high probability of success in decoding over all $O(K)$ bins using $O(K \log N)$ measurements and $O(K \log N)$ decoding time. Excellent examples include the class of *expander codes* or *LDPC codes* that allow for linear time decoding. It has been well established [66] that for a given minimum distance βP, one can construct such expander codes with high probability. Thus we can randomly generate the matrix $\mathbf{G}$ offline and verify its minimum distance, and then keep using it for all instances.

Now, if $\mathsf{sgn}\,[x[k]]$ is not known a priori, the decoding still succeeds if the sign $\mathsf{sgn}\,[x[k]]$ is obtained correctly with high probability. There are two options to obtain the sign of $\mathsf{sgn}\,[x[k]]$: (1) the most straightforward way is to spend another $P = O(n)$ measurements to resolve the sign $\mathsf{sgn}\,[x[k]]$ with high probability; (2) a more elegant way is to include the all-one codeword in the codebook generated by $\mathbf{G}$. Now, given that the all-one codeword is in the codebook $\mathbf{G}$, the decoded codeword can either be the correct codeword $\mathbf{G}\mathbf{b}_k$ or its complementary codeword $\mathbf{G}\mathbf{b}_k \oplus \mathbf{1}$. The true codeword can then be obtained by the single-ton verification in step (3) according to (41).

8 Numerical Experiments

In this section, we provide the empirical performance of our design in the noiseless and noisy settings. Each data point in the simulation is generated by averaging over 200 experiments, where the signals $\mathbf{x}$ are generated once and kept fixed for all the subsequent experiments. In particular, the support of $\mathbf{x}$ are generated uniformly random from $[N]$ with values chosen from the set $\{\pm 1\}$. In the presence of noise, the signal-to-noise ratio (SNR) is defined as

$$\mathsf{SNR} = \frac{\mathbb{E}\left[\|\mathbf{A}\mathbf{x}\|^2\right]}{\mathbb{E}\left[\|\mathbf{w}\|^2\right]} = \frac{\|\mathbf{x}\|^2}{\sigma^2}\frac{\bar{d}}{R} \tag{55}$$

where $\bar{d}$ is the average left node degree of the bipartite graph, R is the number of right nodes in the graph, and the expectation is taken with respect to the noise, *random bipartite graph* and *bin detection matrix*. Then in noisy settings, we generate i.i.d. circularly complex Gaussian noise with variance σ^2 according to the specified SNR.

8.1 Density Evolution Threshold of the Regular Ensemble

We examine the density evolution result using the `FourierNoiseless` design in Section 6.1 in the absence of noise. We generate a sparse vector $\mathbf{x}$ with $K = 500$ and $N = 10^5$ for all the experiments. To understand the effects of the graph ensemble on density evolution, we numerically trace the probability of success $1 - \mathbb{P}_F$ against the redundancy parameter $\eta = R/K$ of the regular graph ensemble $\mathcal{G}^N_{\text{reg}}(R, d)$. For simplicity, we fix the left node degree $d = 3$ and vary the redundancy parameter $\eta = R/K$ from 1 to 1.5. It can be seen that the threshold for $R/K = \eta$ empirically matches with the density evolution analysis for regular graphs in Section 5.2, where the algorithm succeeds with some probability from $\eta = 1.2$ and reaches probability one after $\eta = 1.3$.

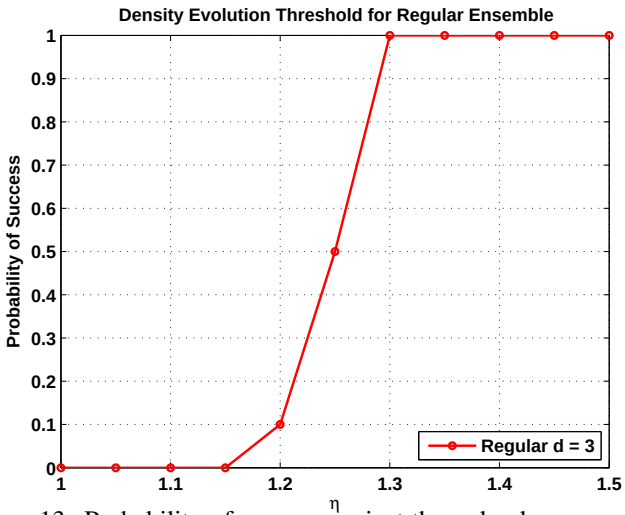
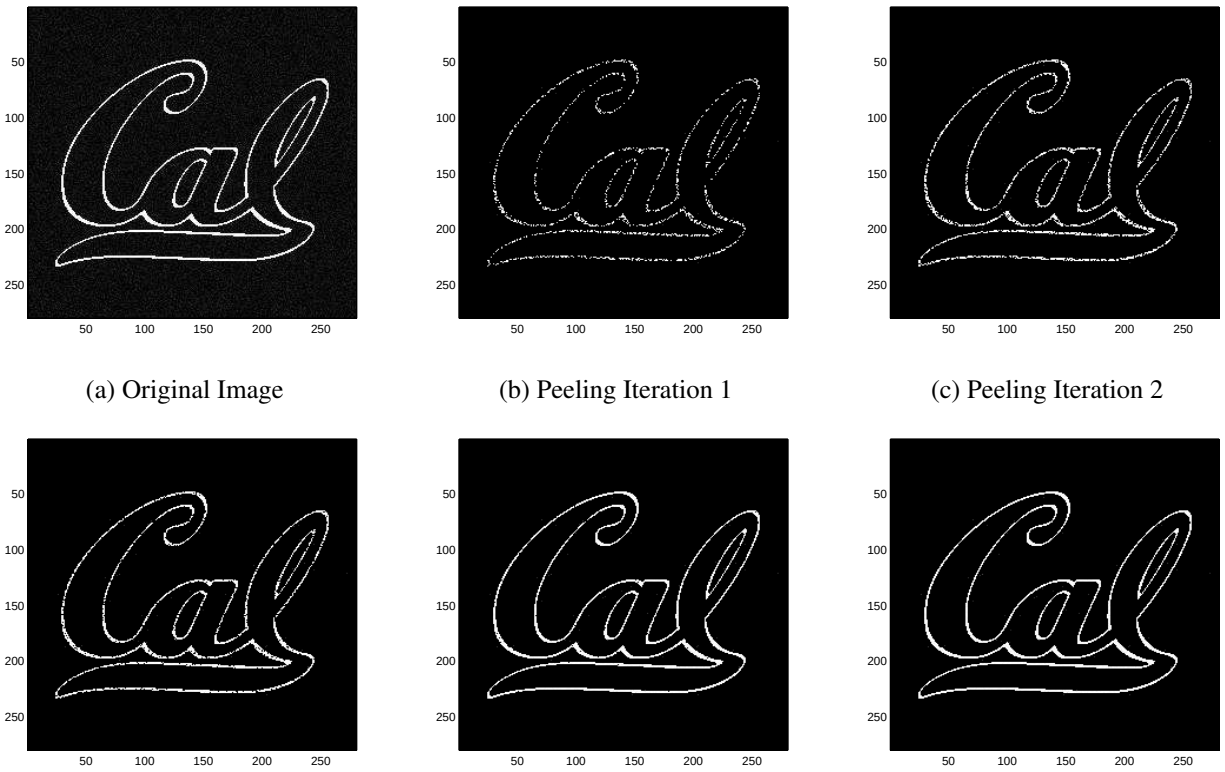


Figure 13: Probability of success against the redundancy parameter η for the regular ensemble $\mathcal{G}^N_{\text{reg}}(\eta K, 3)$ with $N = 0.1$ million.

8.2 Illustration of Density Evolution

We demonstrate the density evolution process by showing the peeling iterations of recovering a 280×280 noisy grayscale "Cal" image consisting of pixels taking values within $[0, 1]$. In this setting, we have the input dimension $N = 280 \times 280 = 78400$ and the sparsity $K = 3600$, and the image in Fig. 14a is free from noise. To recover this Cal image using our framework, we exploit the `FourierNoiseless` design in Section 6.1. In particular, the coding matrix $\mathbf{H}$ is constructed using the regular graph ensemble $\mathcal{G}_{\text{reg}}^{N}(R, d)$ with a regular degree $d = 3$ and a redundancy $R = 1.5K$, while the bin detection is the first two rows of an N-point DFT matrix such that $P = 2$. Therefore, the total measurement cost is $M = RP = 3K = 10800 \approx N \times 13.7\%$, which is around 10% of the input dimension. It can be seen from Fig. 14 that when the density evolution threshold is met $\eta = 1.5 > 1.23$, the image is quickly recovered from a few iterations, where the first 3 iterations almost capture most of the sparse coefficients while iteration 4 and 5 are cleaning up the very few guys that are left.



(a) Original Image (b) Peeling Iteration 1 (c) Peeling Iteration 2

(d) Peeling Iteration 3 (e) Peeling Iteration 4 (f) Peeling Iteration 5

Figure 14: Illustration of density evolution through peeling iterations over the recovery of the " Cal" image

8.3 Noise Robustness and Scalabilty

In this subsection, we showcase the robustness and scalability of the `FourierNoisy` and `BinaryNoisy` designs. The measurement matrix $\mathbf{A}$ is constructed as follows:

- the coding matrix $\mathbf{H}$ is constructed using the regular graph ensemble $\mathcal{G}_{\text{reg}}^{N}(R, d)$ with a regular degree $d = 3$ and a redundancy $R = 2K$;
- for the `FourierNoisy` design, we choose a $P_1 \times N$ random DFT matrix by Definition 4 for the zero-ton and single-ton verifications with $P_1 = \log N$, and a $P_2 Q \times N$ structured DFT matrix from Definition 5 for the single-ton search with $P_2 = \log N$ clusters and $Q = 3$ samples per cluster instead of $Q = O(\log^{1/3} N)$;

- for the `BinaryNoisy` design, we choose a $P_1 \times N$ random Rademacher matrix by Definition 4 for the zero-ton and single-ton verifications with $P_1 = \log N$, and a $P_2 \times N$ coded binary matrix by Definition 6 for the single-ton search with $P_2 = 2\log_2 N$. In particular, the coded binary matrix $\mathbf{C} = \mathbf{GB}$ is chosen based on the $P_2 \times \log_2 N$ generator matrix $\mathbf{G}$ associated with a $(3, 6)$-regular LDPC code, and the single-ton search utilizes the Gallager's bit flipping algorithm for decoding.

To demonstrate the noise robustness, the probability of success is plotted against a range of SNR from 0dB to 16dB for both designs. In each experiment, 50-sparse signals $\mathbf{x}$ (i.e., $K = 50$) with $N = 10^5$ are generated. It is seen in Fig. 15 that for a given measurement cost, there exists a threshold of SNR, above which our noisy recovery schemes succeed with probability 1. It is also observed that the thresholds increase gracefully when the measurement cost is reduced.

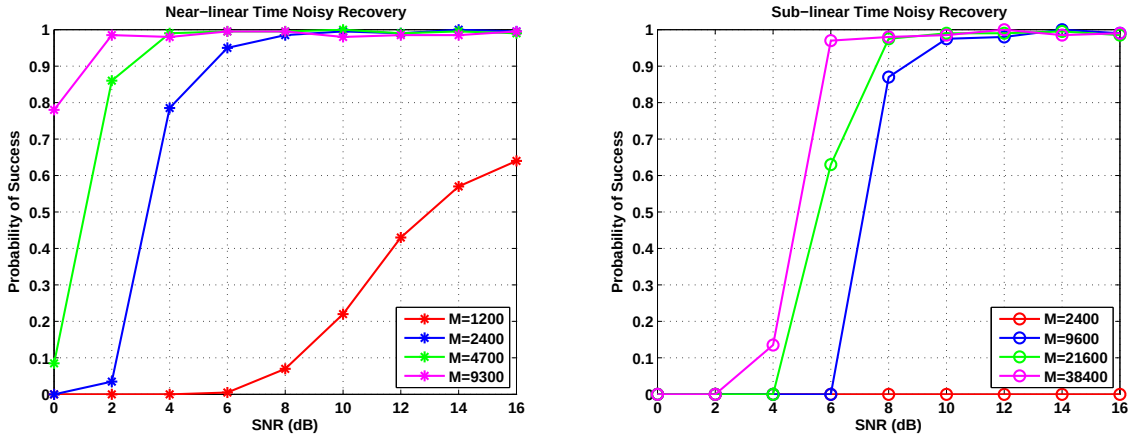


Figure 15: Probability of success of `FourierNoisy` and `BinaryNoisy` against SNR for $N = 0.1$ million and $K = 50$.

To showcase the scalability, we trace the average measurement cost and run-time for both designs. In each experiment, the sparsity of the K-sparse signals $\mathbf{x}$ is chosen as $K = N^\delta$ under different sparsity regimes $\delta = 1/6, 1/3$ and $1/2$, while the ambient dimensions of the signals for each sparsity regime ranges from $N = 10^2$ to $N = 10^7 \approx 10$ million. The measurements are obtained under SNR = 20dB.

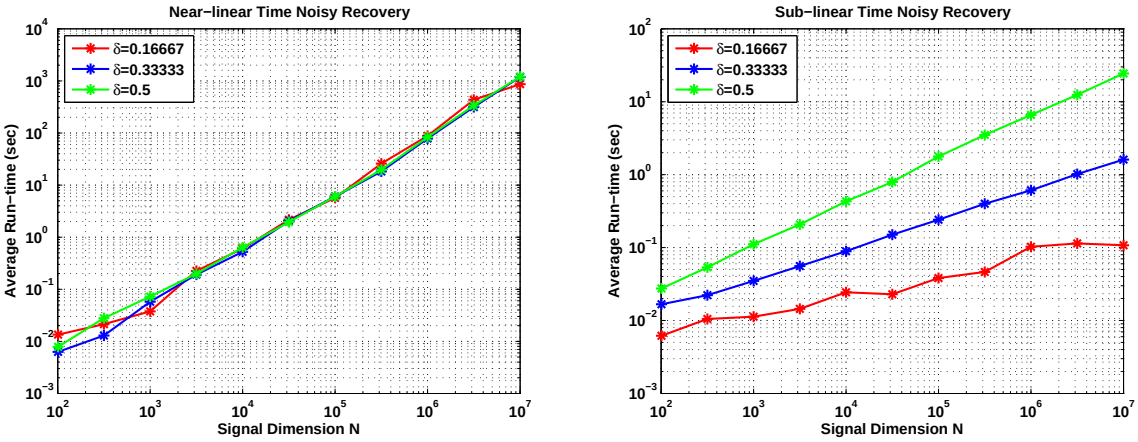


Figure 16: Measurement and computational costs as functions of the signal dimension N for noisy recovery. It can be seen that the measurement cost and the run-time of the `FourierNoisy` and `BinaryNoisy` designs scale sub-linearly with respect to N. For instance, when $N = 10$ million and $K = \sqrt{N}$ (the green curves on both plots), the measurement costs for both schemes are approximately 10^6 and the run-time is around 10 seconds for both.

9 Conclusions

In this paper, we addressed the support recovery problem for compressed sensing using sparse-graph codes. We proposed a compressed sensing design framework for sub-linear time support recovery, by introducing a new family of measurement matrices and fast recovery algorithms. We also provide simulation results to corroborate our theoretical findings.

References

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

[2] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse mri: The application of compressed sensing for rapid mr imaging," *Magnetic resonance in medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.

[3] E. J. Candes, Y. C. Eldar, T. Strohmer, and V. Voroninski, "Phase retrieval via matrix completion," *SIAM Journal on Imaging Sciences*, vol. 6, no. 1, pp. 199–225, 2013.

[4] M. Elad, *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer, 2010.

[5] A. Gilbert and P. Indyk, "Sparse recovery using sparse matrices." Institute of Electrical and Electronics Engineers, 2010.

[6] R. G. Baraniuk, "Compressive sensing," *IEEE signal processing magazine*, vol. 24, no. 4, 2007.

[7] E. J. Candès, Y. Plan *et al.*, "Near-ideal model selection by ?1 minimization," *The Annals of Statistics*, vol. 37, no. 5A, pp. 2145–2177, 2009.

[8] D.-Z. Du and F. K. Hwang, *Combinatorial group testing and its applications*. World Scientific, 1993.

[9] J.-J. Fuchs, "Recovery of exact sparse representations in the presence of bounded noise," *IEEE Trans. on Information Theory*, vol. 51, no. 10, pp. 3601–3608, 2005.

[10] E. Greenshtein *et al.*, "Best subset selection, persistence in high-dimensional statistical learning and optimization under l1 constraint," *The Annals of Statistics*, vol. 34, no. 5, pp. 2367–2386, 2006.

[11] S. Pawar and K. Ramchandran, "Computing a k-sparse n-length discrete fourier transform using at most 4k samples and o (k log k) complexity," in *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 464–468.

[12] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[13] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[14] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[15] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18 914–18 919, 2009.

[16] E. Candes and T. Tao, "The dantzig selector: Statistical estimation when p is much larger than n," *The Annals of Statistics*, pp. 2313–2351, 2007.

[17] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[18] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[19] D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *Foundations of computational mathematics*, vol. 9, no. 3, pp. 317–334, 2009.

[20] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Trans. on Information Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.

[21] J. Tropp and A. C. Gilbert, "Signal recovery from partial information via orthogonal matching pursuit," 2005.

[22] M. A. Davenport, M. F. Duarte, Y. C. Eldar, and G. Kutyniok, "Introduction to compressed sensing," *Preprint*, vol. 93, 2011.

[23] L. Welch, "Lower bounds on the maximum cross correlation of signals (corresp.)," *IEEE Transactions on Information theory*, pp. 397–399, 1974.

[24] S. D. Howard, A. R. Calderbank, and S. J. Searle, "A fast reconstruction algorithm for deterministic compressive sensing using second order reed-muller codes," in *Information Sciences and Systems, 2008. CISS 2008. 42nd Annual Conference on*. IEEE, 2008, pp. 11–15.

[25] L. Applebaum, S. D. Howard, S. Searle, and R. Calderbank, "Chirp sensing codes: Deterministic compressed sensing measurements for fast recovery," *Applied and Computational Harmonic Analysis*, vol. 26, no. 2, pp. 283–290, 2009.

[26] M. Akçakaya and V. Tarokh, "A frame construction and a universal distortion bound for sparse representations," *Signal Processing, IEEE Transactions on*, vol. 56, no. 6, pp. 2443–2450, 2008.

[27] A. G. Dimakis, R. Smarandache, and P. O. Vontobel, "Ldpc codes for compressed sensing," *Information Theory, IEEE Transactions on*, vol. 58, no. 5, pp. 3093–3114, 2012.

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

[29] S. Jafarpour, W. Xu, B. Hassibi, and R. Calderbank, "Efficient and robust compressed sensing using optimized expander graphs," *IEEE Trans. on Information Theory*, vol. 55, no. 9, pp. 4299–4308, 2009.

[30] R. Berinde, A. C. Gilbert, P. Indyk, H. Karloff, and M. J. Strauss, "Combining geometry and combinatorics: A unified approach to sparse signal recovery," in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*. IEEE, 2008, pp. 798–805.

[31] P. Indyk and M. Ruzic, "Near-optimal sparse recovery in the l1 norm," in *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*. IEEE, 2008, pp. 199–207.

[32] R. Berinde, P. Indyk, and M. Ruzic, "Practical near-optimal sparse recovery in the l1 norm," in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*. IEEE, 2008, pp. 198–205.

[33] F. Parvaresh and B. Hassibi, "Explicit measurements with almost optimal thresholds for compressed sensing," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 3853–3856.

[34] H. V. Pham, W. Dai, and O. Milenkovic, "Sublinear compressive sensing reconstruction via belief propagation decoding," in *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*. IEEE, 2009, pp. 674–678.

[35] S. Sarvotham, D. Baron, and R. G. Baraniuk, "Sudocodes - fast measurement and reconstruction of sparse signals," in *Information Theory, 2006 IEEE International Symposium on*. IEEE, 2006, pp. 2804–2808.

[36] M. Bakshi, S. Jaggi, S. Cai, and M. Chen, "Sho-fa: Robust compressive sensing with order-optimal complexity, measurements, and bits," in *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*. IEEE, 2012, pp. 786–793.

[37] F. Zhang and H. D. Pfister, "Compressed sensing and linear codes over real numbers," in *Information Theory and Applications Workshop, 2008*. IEEE, 2008, pp. 558–561.

[38] D. L. Donoho, A. Javanmard, and A. Montanari, "Information-theoretically optimal compressed sensing via spatial coupling and approximate message passing," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*. IEEE, 2012, pp. 1231–1235.

[39] Y. Wu and S. Verdú, "Optimal phase transitions in compressed sensing," *IEEE Trans. on Information Theory*, vol. 58, no. 10, pp. 6241–6263, 2012.

[40] F. Zhang and H. D. Pfister, "Verification decoding of high-rate ldpc codes with applications in compressed sensing," *Information Theory, IEEE Transactions on*, vol. 58, no. 8, pp. 5042–5058, 2012.

[41] M. Finiasz and K. Ramchandran, "Private stream search at the same communication cost as a regular search: Role of ldpc codes," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*. IEEE, 2012, pp. 2556–2560.

[42] M. A. Khajehnejad, J. Yoo, A. Anandkumar, and B. Hassibi, "Summary based structures with improved sublinear recovery for compressed sensing," in *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 1427–1431.

[43] M. Charikar, K. Chen, and M. Farach-Colton, "Finding frequent items in data streams," *Theoretical Computer Science*, vol. 312, no. 1, pp. 3–15, 2004.

[44] P. Indyk, H. Q. Ngo, and A. Rudra, "Efficiently decodable non-adaptive group testing," in *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2010, pp. 1126–1142.

[45] G. Cormode and S. Muthukrishnan, "Combinatorial algorithms for compressed sensing," in *Structural Information and Communication Complexity*. Springer, 2006, pp. 280–294.

[46] J. Haupt and R. Baraniuk, "Robust support recovery using sparse compressive sensing matrices," in *Information Sciences and Systems (CISS), 2011 45th Annual Conference on*. IEEE, 2011, pp. 1–6.

[47] A. C. Gilbert, M. J. Strauss, J. A. Tropp, and R. Vershynin, "One sketch for all: fast algorithms for compressed sensing," in *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*. ACM, 2007, pp. 237–246.

[48] A. C. Gilbert, Y. Li, E. Porat, and M. J. Strauss, "Approximate sparse recovery: optimizing time and measurements," *SIAM Journal on Computing*, vol. 41, no. 2, pp. 436–453, 2012.

[49] 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.

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

[51] G. Reeves and M. Gastpar, "Sampling bounds for sparse support recovery in the presence of noise," in *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*. IEEE, 2008, pp. 2187–2191.

[52] M. Gastpar and Y. Bresler, "On the necessary density for spectrum-blind nonuniform sampling subject to quantization," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, vol. 1. IEEE, 2000, pp. 348–351.

[53] M. J. Wainwright, "Information-theoretic limits on sparsity recovery in the high-dimensional and noisy setting," *IEEE Trans. on Information Theory*, vol. 55, no. 12, pp. 5728–5741, 2009.

[54] S. Aeron, V. Saligrama, and M. Zhao, "Information theoretic bounds for compressed sensing," *IEEE Trans. on Information Theory*, vol. 56, no. 10, pp. 5111–5130, 2010.

[55] M. Akçakaya and V. Tarokh, "Shannon-theoretic limits on noisy compressive sampling," *IEEE Trans. on Information Theory*, vol. 56, no. 1, pp. 492–504, 2010.

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

[57] T. T. Cai and L. Wang, "Orthogonal matching pursuit for sparse signal recovery with noise," *IEEE Trans. on Information Theory*, vol. 57, no. 7, pp. 4680–4688, 2011.

[58] A. K. Fletcher, S. Rangan, and V. K. Goyal, "Necessary and sufficient conditions for sparsity pattern recovery," *IEEE Trans. on Information Theory*, vol. 55, no. 12, pp. 5758–5772, 2009.

[59] W. Wang, M. J. Wainwright, and K. Ramchandran, "Information-theoretic limits on sparse signal recovery: Dense versus sparse measurement matrices," *IEEE Trans. on Information Theory*, vol. 56, no. 6, pp. 2967–2979, 2010.

[60] Y. Jin, Y.-H. Kim, and B. D. Rao, "Limits on support recovery of sparse signals via multiple-access communication techniques," *IEEE Trans. on Information Theory*, vol. 57, no. 12, pp. 7877–7892, 2011.

[61] A. Hormati, A. Karbasi, S. Mohajer, and M. Vetterli, "An estimation theoretic approach for sparsity pattern recovery in the noisy setting," *arXiv preprint arXiv:0911.4880*, 2009.

[62] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. on Information Theory*, vol. 47, no. 2, pp. 599–618, 2001.

[63] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Efficient erasure correcting codes," *IEEE Trans. on Information Theory*, vol. 47, no. 2, pp. 569–584, 2001.

[64] R. Pedarsani, K. Lee, and K. Ramchandran, "Phasecode: Fast and efficient compressive phase retrieval based on sparse-graph-codes," *arXiv preprint arXiv:1408.0034*, 2014.

[65] S. Kay, "A fast and accurate single frequency estimator," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 37, no. 12, pp. 1987–1990, 1989.

[66] T. Richardson and R. Urbanke, *Modern coding theory*. Cambridge University Press, 2008.

[67] L. Birgé, "An alternative point of view on lepski's method," *Lecture Notes-Monograph Series*, pp. 113–133, 2001.

A Oracle-based Peeling Decoder using the Regular Ensemble $\mathcal{G}^N_{\text{reg}}(R,d)$

A.1 Concentration Analysis

A.1.1 Proof of Mean Analysis on General Graphs

From (17), we have

$$\mathbb{E}\left[Z_i\right] = \sum_{e=1}^{Kd} \mathbb{E}\left[Z_i^{(\text{e})}\right] = Kd\mathbb{E}\left[Z_i^{(\text{e})}\right]. \tag{56}$$

From basic probability laws on conditional expectations

$$\mathbb{E}\left[Z_i^{(\text{e})}\right] = \mathbb{E}\left[Z_i^{(\text{e})}|\mathcal{T}_i\right]\mathsf{Pr}\left(\mathcal{T}_i\right) + \mathbb{E}\left[Z_i^{(\text{e})}|\mathcal{T}_i^c\right]\mathsf{Pr}\left(\mathcal{T}_i^c\right).$$

Recall from the density evolution analysis that $\mathbb{E}\left[Z_i^{(\text{e})}|\mathcal{T}_i\right] = p_i$, we have

$$\mathsf{Pr}\left(\mathcal{T}_i\right) \leq 1, \quad \mathbb{E}\left[Z_{\text{e}}|\mathcal{T}_i^c\right] \leq 1 \tag{57}$$

and therefore the following holds:

$$p_i - \mathsf{Pr}\left(\mathcal{T}_i^c\right) \leq \mathbb{E}\left[Z_i^{(\text{e})}\right] \leq p_i + \mathsf{Pr}\left(\mathcal{T}_i^c\right). \tag{58}$$

If the probability of a general graph not behaving like a tree can be made arbitrarily small for any $\varepsilon > 0$,

$$\mathsf{Pr}\left(\mathcal{T}_i^c\right) < \frac{\varepsilon}{4}, \tag{59}$$

then we can obtain the result in (22) by letting $p_i = \varepsilon/4$ in the density evolution analysis. Next, we show that (59) holds for sufficiently large K.

Lemma 5. *For any given constant $\varepsilon > 0$ and iteration $i > 0$, there exists some absolute constant $K_0 > 0$ such that*

$$\mathsf{Pr}\left(\mathcal{T}_i^c\right) < c_0\frac{\log^i K}{K} \tag{60}$$

for some constant $c_0 > 0$ as long as $K > K_0$.

From this lemma, we can see that for an arbitrary $\varepsilon > 0$, the result follows as long as $K > K_0$ where K_0 is the smallest constant that satisfies $K_0/\log^i K_0 > 4c_0/\varepsilon$ given ε and i. In the following we give the proof of the lemma.

Proof. Let C_j be the number of check nodes and V_j be the number of variable nodes in the neighborhood $\mathcal{N}_{\text{e}}^{2j}$. In [62], it has been shown that the directed neighborhood $\mathcal{N}_{\text{e}}^{2i}$ at depth i is not a tree with probability at most $O(1/K)$. However, the proof therein largely rests on the regular degrees for the left and right nodes in the graph. Now, because the graph ensemble $\mathcal{G}^N_{\text{reg}}(R,d)$ follows Poisson distributions on the right, the results in [62] are not immediately applicable here. In this setting, the key idea is to prove that the size of the directed neighborhood $\mathcal{N}_{\text{e}}^{2i}$ unfolded up to depth $2i$ is bounded by $O(\log^i K)$ with high probability, and this neighborhood is not a tree with probability at most $O(\log^i K/K)$.

To show this, we unfold the neighborhood of an edge e up to level i. Fix some constant κ_1, then at each level $j \leq i$ we upper bound the probability of a tree having more than $O(\log^j K)$ left nodes $V_j > \kappa_1 \log^j K$ and right

nodes $C_j > \kappa_1 \log^j K$. Specifically, from the law of total probability, we upper bound the probability for some constant $\kappa_1 > 0$

$$\mathsf{Pr}\left(\mathcal{T}_i^c\right) \leq \mathsf{Pr}\left(V_j > \kappa_1 \log^j K\right) + \mathsf{Pr}\left(C_j > \kappa_1 \log^j K\right) \tag{61}$$

$$+ \mathsf{Pr}\left(\mathcal{T}_i^c | V_j < \kappa_1 \log^j K, C_j < \kappa_1 \log^j K\right). \tag{62}$$

Denoting the first term in (61) as $a_j = \mathsf{Pr}\left(V_j > \kappa_1 \log^j K\right)$, we bound a_j using the total law of probability as follows

$$a_j \leq a_{j-1} + \mathsf{Pr}\left(V_j > \kappa_1 \log^j K | V_{j-1} < \kappa_1 \log^{j-1} K\right). \tag{63}$$

Since the left degree from the regular and irregular ensembles is upper bounded by constants d and $(D+1)$ respectively, thus given $V_{j-1} < \kappa_1 \log^{j-1} K$ at depth $(j-1)$, the number of right neighbors is bounded by $C_{j-1} < \kappa_2 \log^{j-1} K$ for some $\kappa_2 > 0$. Therefore, the second term in (63) can be bounded as

$$\mathsf{Pr}\left(V_j > \kappa_1 \log^j K | V_{j-1} < \kappa_1 \log^{j-1} K\right) \leq \mathsf{Pr}\left(V_j > \kappa_1 \log^j K | C_{j-1} < \kappa_2 \log^{j-1} K\right). \tag{64}$$

Now let the number of check nodes at exactly depth $(j-1)$ be C'_{j-1} such that $C_{j-1} = C'_{j-1} + C_{j-2}$, and further let d_ℓ be the degree of each check node at this depth $\ell = 1, \cdots, C'_{j-1}$, then the right hand side can be evaluated as

$$\mathsf{Pr}\left(V_j > \kappa_1 \log^j K | C_{j-1} < \kappa_2 \log^{j-1} K\right) \leq \mathsf{Pr}\left(\sum_{\ell=1}^{C'_{j-1}} d_\ell \geq \kappa_3 \log^j K\right) \tag{65}$$

for some $\kappa_3 > 0$. Since each check node degree d_ℓ is an independent Poisson variable with rate $1/\eta$, the sum of d_ℓ over $\ell = 1, \cdots, C'_{j-1}$ remains a Poisson variable with rate C'_{j-1}/η. Since obviously $C'_{j-1} < C_{j-1} < \kappa_1 \log^{j-1} K$ such that the sum rate is $C'_{j-1}/\eta = O(\log^{j-1} K)$. With this sum rate, the probability in (65) can be upper bounded with the tail bound of a Possible variable X with rate λ as $\mathsf{Pr}\left(X \geq x\right) \leq (\lambda e/x)^x$:

$$\mathsf{Pr}\left(\sum_{\ell=1}^{C'_{j-1}} d_\ell \geq \kappa_3 \log^j K\right) \leq \left(\frac{eC'_{j-1}/\eta}{\kappa_3 \log^j K}\right)^{\kappa_3 \log^j K} = \left(\frac{e \times O(\log^{j-1} K)}{\kappa_3 \log^j K}\right)^{\kappa_3 \log^j K} \leq \left(\frac{\kappa_4}{\log K}\right)^{\kappa_3 \log^j K} \leq \frac{\kappa_5}{K}$$

for some sufficiently large constants $\kappa_4 > 0$ and $\kappa_5 > 0$. Therefore we have

$$\alpha_j \leq \alpha_{j-1} + \frac{\kappa_5}{K} \tag{66}$$

and thus the number of variable nodes exposed until the i-th iteration can be bounded by $\log^j K$ with high probability

$$\mathsf{Pr}\left(V_j > \kappa_1 \log^j K\right) = O\left(\frac{1}{K}\right). \tag{67}$$

Similar technique can be used to show that the tail bound for the check nodes is

$$\mathsf{Pr}\left(C_j > \kappa_1 \log^j K\right) = O\left(\frac{1}{K}\right). \tag{68}$$

Now that it has been shown that the number of nodes is well bounded by $O(\log^j K)$, we can proceed to bound the second term in (61) by induction. Assuming that the neighborhood $\mathcal{N}_{\mathrm{e}}^{2j}$ at the j-th iteration $(j < i)$ is tree-like, we prove that $\mathcal{N}_{\mathrm{e}}^{2(j+1)}$ is tree-like with high probability. First of all, we examine the neighborhood $\mathcal{N}_{\mathrm{e}}^{2j+1}$. The probability that a certain edge from a variable node does not create a cycle in $\mathcal{N}_{\mathrm{e}}^{2j+1}$ is the probability that it is

connected to one of the check nodes that are not already included in the tree in $\mathcal{N}_{\mathrm{e}}^{2j}$, which is lower bound by $1 - C_j/(\eta K)$. Therefore, given that $\mathcal{N}_{\mathrm{e}}^{2j}$ is tree-like, the probability that $\mathcal{N}_{\mathrm{e}}^{2j+1}$ is tree-like is lower bounded by

$$\left(1 - \frac{C_j}{\eta K}\right)^{C_{j+1}-C_j} > \left(1 - \frac{C_i}{\eta K}\right)^{C_{j+1}-C_j}. \tag{69}$$

Similarly, given that $\mathcal{N}_{\mathrm{e}}^{2j+1}$ is tree-like, the probability that $\mathcal{N}_{\mathrm{e}}^{2(j+1)}$ is tree-like is lower bounded by

$$\left(1 - \frac{V_j}{K}\right)^{V_{j+1}-V_j} > \left(1 - \frac{V_i}{K}\right)^{V_{j+1}-V_j}. \tag{70}$$

Therefore, the probability that $\mathcal{N}_{\mathrm{e}}^{2(j+1)}$ is tree-like is lower bounded by

$$\left(1 - \frac{C_i}{\eta K}\right)^{C_j}\left(1 - \frac{V_i}{K}\right)^{V_j} \geq \left(1 - \frac{C_i}{\eta K}\right)^{C_i}\left(1 - \frac{V_i}{K}\right)^{V_i} \geq 1 - \left(\frac{V_i^2}{K} + \frac{C_i^2}{\eta K}\right) \geq 1 - O\left(\frac{\log^i K}{K}\right).$$

Therefore the probability of not being tree-like is upper bounded by

$$\mathsf{Pr}\left(\mathcal{T}_i^c\right) < c_0 \frac{\log^i K}{K} \tag{71}$$

for some absolute constant $c_0 > 0$.

□

A.1.2 Proof of Concentration to Mean by Large Deviation Analysis

Now it remains to show the concentration of Z_i around its mean $\mathbb{E}[Z_i]$. According to (17), the number of remaining edges is a sum of random variables $Z_i = \sum_{e=1}^{Kd} Z_i^{\mathrm{e}}$ while summands Z_i^{e} are not independent with each other. Therefore, to show the concentration, we use a standard martingale argument and Azuma's inequality provided in [62] with some modifications to account for the irregular degrees of the right nodes.

Suppose that we expose the whole set of $E = Kd$ edges of the graph one at a time. We let

$$Y_\ell = \mathbb{E}\left[Z_i | Z_i^1, \cdots, Z_i^\ell\right], \quad \ell = 1, \cdots, Kd. \tag{72}$$

By definition, $Y_0, Y_1, \cdots, Y_{Kd}$ are a Doob's martingale process, where $Y_0 = \mathbb{E}[Z_i]$ and $Y_{Kd} = Z_i$. To use Azuma's inequality, it is required that $|Y_{\ell+1} - Y_\ell| \leq \Delta_\ell$ for some $\Delta_\ell > 0$. If the variable node has a regular degree d and the check node has a regular degree d_C, then [62] shows that $\Delta_\ell = 8(dd_C)^i$ with i being the number of peeling iterations. However, the check node degree is not regular with degree d_C and therefore requires further analysis.

Proof of Finite Difference Δ_ℓ

To prove that the difference Δ_ℓ is finite for check node degrees with Poisson distributions, we first prove that the degree of all the check nodes can be upper bounded by $d_C \leq O(K^{\frac{2}{4i+1}})$ with probability[17] at least

$$c_1 K \exp\left(-c_2 K^{\frac{2}{4i+1}}\right)$$

[17] Let X be a Poisson variable with parameter λ, then the following holds

$$\mathsf{Pr}\left(X > cK^{\frac{2}{4i+1}}\right) \leq \left(\frac{e\lambda}{cK^{\frac{2}{4i+1}}}\right)^{cK^{\frac{2}{4i+1}}} \leq c_1 \exp\left(-c_2 K^{\frac{2}{4i+1}}\right)$$

for some c_1 and c_2.

for some constants c_1 and c_2. Let $\mathcal{B}$ be the event that at least one check node has more than $O\left(K^{\frac{2}{4i+1}}\right)$ edges, then for some $c_3 > 0$ we have

$$\mathsf{Pr}\left(\mathcal{B}\right) < c_3 K \exp\left(-c_2 K^{\frac{2}{4i+1}}\right). \tag{73}$$

by applying a union bound on all the $R = \eta K$ check nodes of the graphs from $\mathcal{G}_{\text{reg}}^{N}(R, d)$. As a result, under the complement event $\mathcal{B}^c$, we have

$$\Delta_\ell^2 = O\left(K^{\frac{4i}{4i+1}}\right). \tag{74}$$

Large Deviation by Azuma's Inequality

For any given $\varepsilon > 0$, the tail probability of the event $Z_i > Kd\varepsilon$ can be computed as

$$\begin{aligned}
\mathsf{Pr}\left(|Z_i - \mathbb{E}[Z_i]| > \frac{Kd\varepsilon}{2}\right) &\leq \mathsf{Pr}\left(|Z_i - \mathbb{E}[Z_i]| > \frac{Kd\varepsilon}{2}\Big|\mathcal{B}^c\right) + \mathsf{Pr}\left(\mathcal{B}\right) \\
&\leq 2\exp\left(-\frac{K^2\bar{d}^2\varepsilon^2/4}{2\sum_{\ell=1}^{Kd}\Delta_\ell^2}\right) + c_3 K \exp\left(-c_2 K^{\frac{2}{4i+1}}\right) \\
&\leq 2\exp\left(-c_4\varepsilon^2 K^{\frac{1}{4i+1}}\right),
\end{aligned}$$

where c_4 is some constant depending on d, η and all the other constants c_1, c_2, c_3. This concludes our proof for (21).

A.2 Proof of Graph Expansion Properties in Lemma 3

Let $\mathcal{S}_v$ denote the event that a variable node subset of size v with at most $\bar{d}|\mathcal{S}_v|/2$ neighbors, whose probability can be obtained readily for any size $|\mathcal{S}_v| = v$ as

$$\mathsf{Pr}\left(\mathcal{S}_v\right) \leq \binom{K}{v}\binom{\eta K}{\bar{d}v/2}\left(\frac{v\bar{d}}{2\eta K}\right)^{\bar{d}v}, \tag{75}$$

where we have used the fact that the number of check nodes is ηK. Using the inequality $\binom{a}{b} \leq (ae/b)^b$, we have

$$\mathsf{Pr}\left(\mathcal{S}_v\right) \leq \left(\frac{v}{K}\right)^{(\bar{d}/2-1)v} c^v \leq \left(\frac{vc^2}{K}\right)^{v/2}, \tag{76}$$

where $c = e(\bar{d}/2\eta)^{\bar{d}/2}$ is some constant. Then a union bound is applied over all possible values v up to the remaining variable nodes $\varepsilon_\star K$. Choosing $\varepsilon_\star < 1/(2c^2)$ yields

$$\sum_{v=2}^{\varepsilon_\star K}\mathsf{Pr}\left(\mathcal{S}_v\right) \leq \sum_{v=2}^{\varepsilon_\star K}\left(\frac{vc^2}{K}\right)^{v/2} = O\left(\frac{1}{K}\right). \tag{77}$$

Therefore, asymptotically in K, the random graphs from both the regular and irregular ensembles are good expanders on small sets of variable nodes.

B Oracle-based Peeling Decoder using the Irregular Ensemble $\mathcal{G}_{\text{irreg}}^{N}(R, D)$

Based on the peeling decoder analysis in Section 5.2, it can be easily shown that the concentration analysis and graph expansion property carry over to the irregular graph ensemble. Hence, we focus on the density evolution for the oracle-based peeling decoder over irregular ensemble.

To study the probability p_i of an edge being present in the pruned graph from the irregular ensemble after i iterations, we need to first understand the right edge degree distributions ρ_j of the graph. Using the degree sequence λ_j of the irregular graph ensemble $\mathcal{G}^N_{\text{irreg}}(R, D)$ in Definition 3, it can be shown that the right degree sequence ρ_j follows a Poisson distribution similar to (12)

$$\rho_j \approx \frac{\left(\bar{d}/(1+\epsilon)\right)^{j-1} e^{-\bar{d}/(1+\epsilon)}}{(j-1)!},$$

where we have used $R = (1+\epsilon)K$ and $\bar{d}$ is the average degree of a left node in the irregular graph ensemble

$$\bar{d} = \frac{1}{\sum_{j=2}^{D+1} \lambda_j / j} = H(D)\left(1 + \frac{1}{D}\right). \tag{78}$$

Using the left and right degree sequence (λ_j, ρ_j), we can readily obtain the left and right degree generating polynomials $\lambda(x) = \sum_{d=1}^{\infty} \lambda_j x^{j-1}$ and $\rho(x) = \sum_{j=1}^{\infty} \rho_j x^{j-1}$

$$\lambda(x) = \frac{1}{H(D)} \sum_{j=2}^{D+1} \frac{1}{(j-1)} x^{j-1}, \quad \rho(x) = e^{-\frac{\bar{d}}{1+\epsilon}(1-x)}.$$

As a result, the associated density evolution equation can be written using the degree generating polynomials similar to that in (15)

$$p_i = f(p_{i-1}) = \lambda(1 - \rho(1 - p_{i-1})), \quad i = 1, 2, 3, \cdots. \tag{79}$$

The density evolution analysis suggests that if the fraction p_i in (79) can be made arbitrarily small if the density evolution recursion is contracting

$$\lambda(1 - \rho(1 - x)) < x, \quad \forall\, x \in [0, 1]. \tag{80}$$

Examples of this density evolution using different values of D and ϵ are given in Fig. 17. Clearly, when $\epsilon = 0.1$, the density evolution equation becomes a contraction mapping when $D = 100$ but not when $D = 10$. Now we study how to choose D for any given $\epsilon > 0$. Since $\lambda(x)$ is a non-decreasing function, we can apply $x = \lambda^{-1}(p_{i-1})$ on both sides of (80), then the contraction condition is equivalent to

$$\rho(1 - \lambda(x)) > 1 - x, \quad \forall x \in [0, 1]. \tag{81}$$

By substituting the right generating polynomial $\rho(x)$ into the above recursion, we have

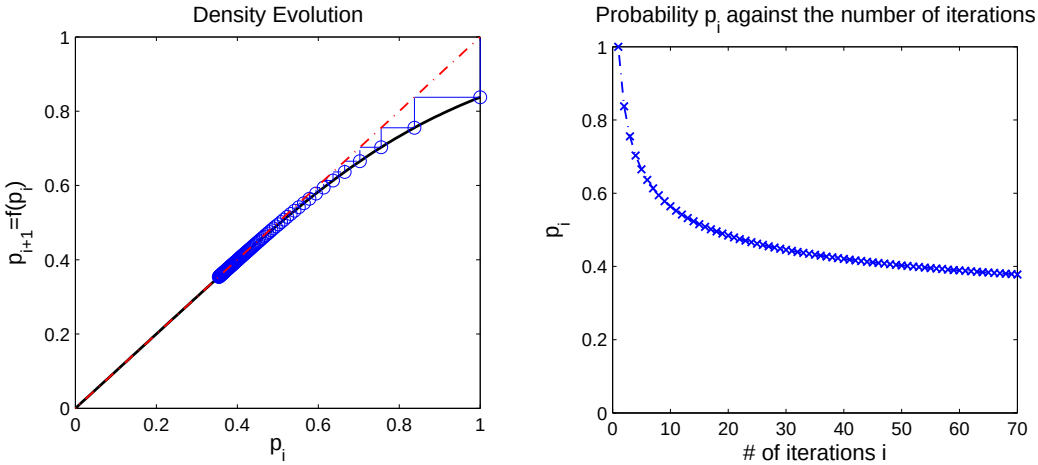
$$\rho(1 - \lambda(x)) = e^{-\frac{\bar{d}}{(1+\epsilon)}\lambda(x)}. \tag{82}$$

To simplify our expressions, we further bound $\lambda(x)$ for the irregular graph ensemble $\mathcal{G}^N_{\text{irreg}}(R, D)$ as $\lambda(x) > -\frac{1}{H(D)} \log(1-x)$. This is because $\lambda(x)$ is a D-term approximation of the Taylor expansion for $\log(1-x)$, scaled by the normalization constant $H(D)$. By substituting this bound into (82), we have

$$\rho(1 - \lambda(x)) > e^{\frac{\bar{d}}{(1+\epsilon)} \frac{1}{H(D)} \log(1-x)} = (1-x)^{\frac{\bar{d}}{(1+\epsilon)H(D)}}.$$

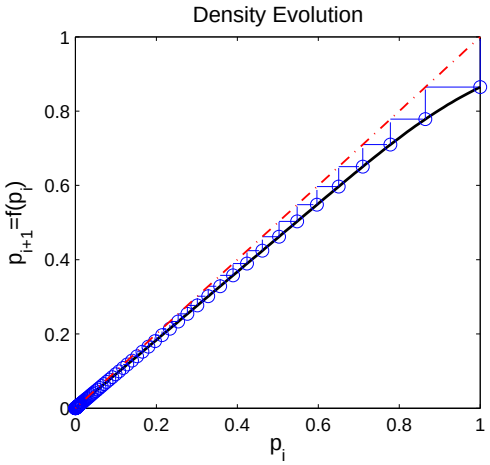
It can be seen that the right hand side is no less than $1 - x$ as long as $H(D) \geq \frac{\bar{d}}{(1+\epsilon)}$. Substituting the average degree $\bar{d}$ from (78) back to this condition, then for any $\epsilon > 0$, we can choose $D > 1/\epsilon$ as in Definition 3 to render the recursion a contracting mapping.

Finally, together with the concentration analysis and graph expansion properties of the irregular graphs, the oracle-based peeling decoder successfully decodes all the edges in the graph with probability at least $1 - O(1/K)$.

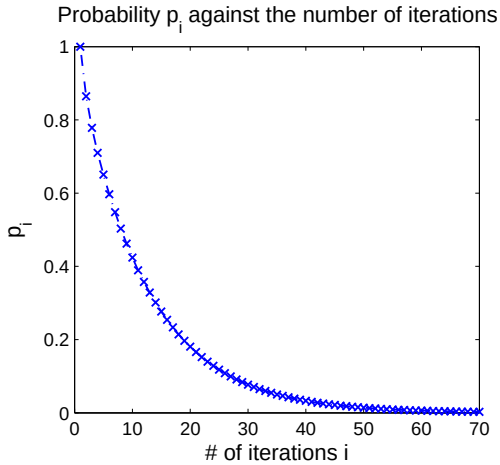


(a) $\epsilon = 0.1$ and $D = 10$

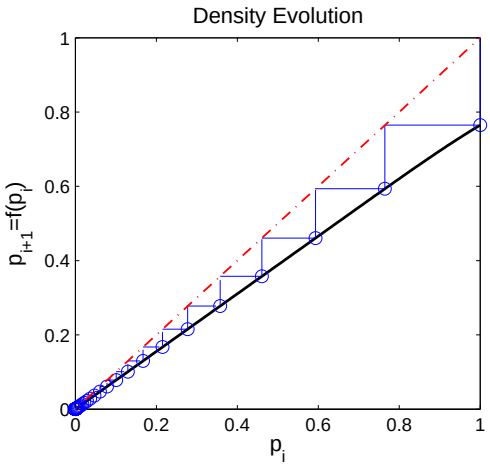
(b) $\epsilon = 0.1$ and $D = 10$



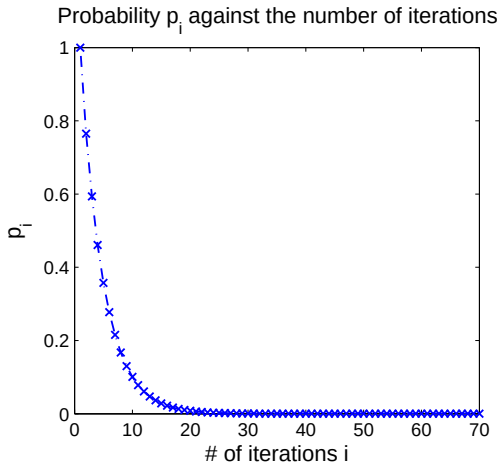
(c) $\epsilon = 0.1$ and $D = 100$



(d) $\epsilon = 0.1$ and $D = 100$



(e) $\epsilon = 0.3$ and $D = 100$



(f) $\epsilon = 0.3$ and $D = 100$

Figure 17: The density evolution $f(p_i)$ and the probability p_i at each iteration i, where we have shown cases with $\epsilon = 0.1$ and $D = 10$ and $D = 100$, as well as the case with $\epsilon = 0.3$ and $D = 100$. In the density evolution figures (a)-(c)-(e), the red line is the line $p_{i+1} = p_i$ while the black line is the density evolution $f(p_i)$ against p_i. The blue circles that "zig-zag" between the red line and the black line are the specific p_i's at each peeling iteration. It can be seen from (a) and (c) that when ϵ is small (i.e. $\epsilon = 0.1$), the density evolution requires a large maximum left degree D to reach density 0. On the other hand, when ϵ is large (i.e. $\epsilon = 0.3$), the density p_i reaches 0 very quickly in (e) with the same maximum left degree $D = 100$. The values of p_i marked by the blue circles in (a)-(c)-(e) are further plotted against the peeling iterations i in (b)-(d)-(f), where in the case with $\epsilon = 0.3$ and $D = 100$ the density p_i approaches 0 after less than 20 iterations.

C Proof of Proposition 2

Given a single-ton bin with an index-value pair $(k, x[k])$,

$$y[p] = |x[k]|(-1)^{b_k[p] \oplus \mathsf{sgn}[x[k]]} + w[p], \quad p \in [P], \tag{83}$$

it is clear that $\mathsf{sgn}\left[y[p]\right] = \mathsf{sgn}\left[x[k]\right] \oplus 1$ whenever the noise $w[p]$ is sufficiently large such that it crosses over $x[k](-1)^{b_k[p]}$. Clearly, this is a random event and we can model it with some Bernoulli variable $Z_p \in \{0, 1\}$ with some probability p_Z

$$\mathsf{sgn}\left[y[p]\right] = b_k[p] \oplus \mathsf{sgn}\left[x[k]\right] \oplus Z_p. \tag{84}$$

The exact parameter p_Z of the Bernoulli random variable Z_p can be found by studying the tail events that trigger the flipping, but here for simplicity we directly upper bound it as follows

$$p_Z \leq \mathbb{P}_{\mathsf{e}} := \mathsf{Pr}\left(|w[p]| > |x[k]|\right) \leq e^{-\frac{|x[k]|^2}{2N/B\sigma^2}} = e^{-\frac{\eta}{2}\mathrm{SNR}}. \tag{85}$$

D Peeling Decoder using a Robust Bin Detector

Let E_{bin} be the event where the robust bin detector makes a mistake. From the law of total probability, we have

$$\begin{aligned}\mathbb{P}_F &= \mathsf{Pr}\left(\mathsf{supp}\left(\widehat{\mathbf{x}}\right) \neq \mathsf{supp}\left(\mathbf{x}\right) \middle| E_{\mathrm{bin}}^c\right) \mathsf{Pr}\left(E_{\mathrm{bin}}^c\right) + \mathsf{Pr}\left(\mathsf{supp}\left(\widehat{\mathbf{x}}\right) \neq \mathsf{supp}\left(\mathbf{x}\right) \middle| E_{\mathrm{bin}}\right) \mathsf{Pr}\left(E_{\mathrm{bin}}\right) \\ &\leq \mathsf{Pr}\left(\mathsf{supp}\left(\widehat{\mathbf{x}}\right) \neq \mathsf{supp}\left(\mathbf{x}\right) \middle| E_{\mathrm{bin}}^c\right) + \mathsf{Pr}\left(E_{\mathrm{bin}}\right).\end{aligned}$$

Since it is known from Theorem 4 that $\mathsf{Pr}\left(\mathsf{supp}\left(\widehat{\mathbf{x}}\right) \neq \mathsf{supp}\left(\mathbf{x}\right) \middle| E_{\mathrm{bin}}^c\right) = O(1/K)$, then if further we have

$$\mathsf{Pr}\left(E_{\mathrm{bin}}\right) = O\left(\frac{1}{K}\right), \tag{86}$$

the overall failure probability can be upper bounded as

$$\mathbb{P}_F = O\left(\frac{1}{K}\right).$$

Now it remains to show that (86) holds. The main idea is to analyze the error probability of making at least an error on any bin measurement, followed by a union bound on all the $R = O(K)$ bins. Denote the error event in any bin j as E_j, then we have the following union bound across $R = \eta K$ measurement bins as well as $\bar{d}K$ iterations[18]

$$\mathsf{Pr}\left(E_{\mathrm{bin}}\right) \leq dK \bigcup_{j=1}^{\eta K} \mathsf{Pr}\left(E_j\right), \tag{87}$$

where $\bar{d}$ is the average left degree of the bipartite graph. Without loss of generality, we drop the bin index such that

$$\mathsf{Pr}\left(E_{\mathrm{bin}}\right) \leq \eta d K^2 \mathsf{Pr}\left(E\right), \tag{88}$$

where $\mathsf{Pr}\left(E\right)$ is the error probability for an arbitrary bin. It can be seen that due to the union bounds, it is required that $\mathsf{Pr}\left(E\right) = O(1/K^3)$ such that $\mathsf{Pr}\left(E_{\mathrm{bin}}\right) = O(1/K)$. In the following, we prove that $\mathsf{Pr}\left(E\right) = O(1/K^3)$ holds using the generic model in (35). Since there are different types of errors, thus in the following analysis $\mathbf{z}$ is fixed as a zero-ton, single-ton or multi-ton respectively for each class of errors.

[18]The number of iterations is taken to be the worst case where at each iteration only one edge is peeled off.

Definition 7. *The error probability* $\mathsf{Pr}\left(E\right)$ *for an arbitrary bin can be upper bounded as*

$$\mathsf{Pr}\left(E\right) \leq \sum_{\mathcal{F}\in\{\mathcal{H}_Z,\mathcal{H}_M\}} \mathsf{Pr}\left(\mathcal{F} \leftarrow \mathcal{H}_S(k,x[k])\right) + \sum_{\mathcal{F}\in\{\mathcal{H}_Z,\mathcal{H}_M\}} \mathsf{Pr}\left(\mathcal{H}_S(k,x[k]) \leftarrow \mathcal{F}\right) \tag{89}$$

$$+ \mathsf{Pr}\left(\mathcal{H}_S(\widehat{k},\widehat{x}[\widehat{k}]) \leftarrow \mathcal{H}_S(k,x[k])\right) \tag{90}$$

where $\mathcal{F}$ *is either a zero-ton* $\mathcal{H}_Z$ *or a multi-ton* $\mathcal{H}_M$ *and*

1. $\mathsf{Pr}\left(\mathcal{F} \leftarrow \mathcal{H}_S(k,x[k])\right)$ *is called the* **missed verification** *rate in which the single-ton verification fails when the ground truth is in fact a single-ton* $\mathcal{H} = \mathcal{H}_S(k,x[k])$ *for some* $k \in [N]$ *and* $x[k]$.
2. $\mathsf{Pr}\left(\mathcal{H}_S(k,x[k]) \leftarrow \mathcal{F}\right)$ *is called the* **false verification** *rate in which the single-ton verification is passed for some single-ton* $\mathcal{H} = \mathcal{H}_S(\widehat{k},\widehat{x}[\widehat{k}])$ *with an index-value pair* $(\widehat{k},\widehat{x}[\widehat{k}])$ *when the ground truth is* $\mathcal{F} \in \{\mathcal{H}_Z,\mathcal{H}_M\}$.
3. $\mathsf{Pr}\left(\mathcal{H}_S(\widehat{k},\widehat{x}[\widehat{k}]) \leftarrow \mathcal{H}_S(k,x[k])\right)$ *is called the* **crossed verification** *rate in which a single-ton with a wrong index-value pair* $\widehat{k} \neq k, \widehat{x}[\widehat{k}] \neq x[k]$ *passes the single-ton verification when the ground truth is a single-ton with an index-value pair* $\mathcal{H} = \mathcal{H}_S(k,x[k])$ *for some* $k \neq \widehat{k}$.

The false verification rate, missed verification rate and crossed verification rate are given in the following propositions. The proof of the following results hold for the bin detection matrix $\mathbf{S}$ given in the `RandomNoisy`, `FourierNoisy` and `BinaryNoisy` designs.

Proposition 3 (False Verification Rate)**.** *For any* $0 < \gamma < A_{\min}^2/2\sigma^2$*, the false verification rate for each bin hypothesis can be upper bounded as follows:*

$$\mathsf{Pr}\left(\mathcal{H}_S(\widehat{k},\widehat{x}[\widehat{k}]) \leftarrow \mathcal{H}_Z\right) < e^{-\frac{P}{4}\left(\sqrt{1+2\gamma}-1\right)^2}$$

$$\mathsf{Pr}\left(\mathcal{H}_S(\widehat{k},\widehat{x}[\widehat{k}]) \leftarrow \mathcal{H}_M\right) < \begin{cases} e^{-\frac{P}{4}\frac{\gamma^2}{1+4\gamma}} + 2e^{-c_6 P\left(1-\frac{2\gamma\sigma^2}{A_{\min}^2}\right)}, & \texttt{RandomNoisy} \\ e^{-\frac{P}{4}\frac{\gamma^2}{1+4\gamma}} + e^{-\frac{P}{4}\left(1-\frac{2\gamma\sigma^2}{A_{\min}^2}\right)}, & \texttt{FourierNoisy} \\ e^{-\frac{P}{4}\frac{\gamma^2}{1+4\gamma}} + 2e^{-c_6 P\left(1-\frac{2\gamma\sigma^2}{A_{\min}^2}\right)}, & \texttt{BinaryNoisy} \end{cases}$$

for some absolute constant $c_6 > 0$.

Proof. See Appendix E. □

Proposition 4 (Missed Verification Rate)**.** *For any* $0 < \gamma < \min\{\rho^2/\sigma^2, A_{\min}^2/\sigma^2\}$*, the missed verification rate for each bin hypothesis can be upper bounded as follows:*

$$\mathsf{Pr}\left(\mathcal{H}_Z \leftarrow \mathcal{H}_S(k,x[k])\right) < \begin{cases} e^{-\frac{P}{4}\frac{\gamma^2}{1+4\gamma}} + 2e^{-c_6 P\left(1-\frac{2\gamma\sigma^2}{A_{\min}^2}\right)}, & \texttt{RandomNoisy} \\ e^{-\frac{P}{4}\frac{\gamma^2}{1+4\gamma}} + Qe^{-\frac{P}{4}\left(1-\frac{2\gamma\sigma^2}{A_{\min}^2}\right)}, & \texttt{FourierNoisy} \\ e^{-\frac{P}{4}\frac{\gamma^2}{1+4\gamma}} + 2e^{-c_6 P\left(1-\frac{2\gamma\sigma^2}{A_{\min}^2}\right)}, & \texttt{BinaryNoisy} \end{cases}$$

$$\mathsf{Pr}\left(\mathcal{H}_M \leftarrow \mathcal{H}_S(k,x[k])\right) < \begin{cases} e^{-\frac{P}{4}\left(\sqrt{1+2\gamma}-1\right)^2} + \frac{3}{2}e^{-P\gamma^2} + 6e^{-c_6 P\left(1-\frac{\gamma\sigma^2}{A_{\min}^2}\right)}, & \texttt{RandomNoisy} \\ e^{-\frac{P}{4}\left(\sqrt{1+2\gamma}-1\right)^2} + 8Pe^{-\frac{A_{\min}^2}{4\sigma^2}Q^3} + 4e^{-\frac{P}{4}\left(1-\frac{\gamma\sigma^2}{\rho^2}\right)}, & \texttt{FourierNoisy} \\ e^{-\frac{P}{4}\left(\sqrt{1+2\gamma}-1\right)^2} + 2e^{-\frac{(\beta-\mathbb{P}_\mathsf{e})^2}{2\mathbb{P}_\mathsf{e}(1-\mathbb{P}_\mathsf{e})}P} + 4e^{-\frac{P}{4}\left(1-\frac{\gamma\sigma^2}{\rho^2}\right)}, & \texttt{BinaryNoisy} \end{cases}$$

for some constant $c_6, c_7, c_8 > 0$.

Proof. See Appendix F. □

Proposition 5 (Crossed Verification Rate)**.** *For any $0 < \gamma < A_{\min}^2/\sigma^2$, the false verification rate for each bin hypothesis can be upper bounded as follows:*

$$\mathsf{Pr}\left(\mathcal{H}_S(\widehat{k},\widehat{x}[\widehat{k}]) \leftarrow \mathcal{H}_S(k,x[k])\right) < \begin{cases} e^{-\frac{P}{4}\frac{\gamma^2}{1+4\gamma}} + 2e^{-c_6 P\left(1-\frac{\gamma\sigma^2}{A_{\min}^2}\right)}, & \texttt{RandomNoisy} \\ e^{-\frac{P}{4}\frac{\gamma^2}{1+4\gamma}} + 2e^{-\frac{P}{4}\left(1-\frac{\gamma\sigma^2}{A_{\min}^2}\right)}, & \texttt{FourierNoisy} \\ e^{-\frac{P}{4}\frac{\gamma^2}{1+4\gamma}} + 2e^{-c_6 P\left(1-\frac{\gamma\sigma^2}{A_{\min}^2}\right)}, & \texttt{BinaryNoisy} \end{cases}$$

Proof. See Appendix E. □

Since all the error probabilities decay exponentially with respect to P in the `RandomNoisy` design, with respect to P and Q in the `FourierNoisy` design, all the verification error rates will vanish to zero at a rate $O(1/N^3) < O(1/K^3)$ as long as they are chosen as $O(\log N)$. Thus the result $\mathsf{Pr}\,(E) = O(1/K^3)$ holds.

E Proof of False Verification Rates in Proposition 3

The false verification events occur if the zero-ton or single-ton verifications fail when the ground truth is either a zero-ton or a multi-ton

$$\boldsymbol{y} = \mathbf{S}\mathbf{z} + \mathbf{w} \tag{91}$$

with $\mathbf{z}$ being a zero-ton $\mathbf{z} = \mathbf{0}$ or a multi-ton $|\mathsf{supp}\,(\mathbf{z})| > 1$.

E.1 Detecting a Zero-ton as a Single-ton

By definition, the probability of this event can be obtained under the zero-ton model

$$\boldsymbol{y} = \mathbf{w} \tag{92}$$

with the following two conditions:

- the zero-ton verification fails

$$\frac{1}{P}\left\|\mathbf{w}\right\|^2 \geq (1+\gamma)\sigma^2 \tag{93}$$

- and the single-ton verification step for some index-value pair $(\widehat{k},\widehat{x}[\widehat{k}])$ is passed:

$$\frac{1}{P}\left\|\mathbf{w} - \widehat{x}[\widehat{k}]\mathbf{s}_{\widehat{k}}\right\|^2 \leq (1+\gamma)\sigma^2 \tag{94}$$

As a result, the error probability can be bounded by either of the probabilities above:

$$\mathsf{Pr}\left(\mathcal{H}_{\mathrm{S}}(\widehat{k},\widehat{x}[\widehat{k}]) \leftarrow \mathcal{H}_{\mathrm{Z}}\right) \leq \mathsf{Pr}\left(\frac{1}{P}\left\|\mathbf{w}\right\|^2 \geq (1+\gamma)\sigma^2\right).$$

This tail bound can be readily obtained from Lemma 11 with the threshold $\tau_1 = (1+\gamma)\sigma^2$ and the non-centrality parameter $\nu_0 = 0$. This gives

$$\mathsf{Pr}\left(\mathcal{H}_{\mathrm{S}}(\widehat{k},\widehat{x}[\widehat{k}]) \leftarrow \mathcal{H}_{\mathrm{Z}}\right) \leq \exp\left(-\frac{P}{4}\left(\sqrt{1+2\gamma}-1\right)^2\right).$$

E.2 Detecting a Multi-ton as a Single-ton

By definition, the error probability can be evaluated under the multi-ton model

$$\boldsymbol{y} = \mathbf{S}\mathbf{z} + \mathbf{w} \tag{95}$$

when it passes the single-ton verification step for some index-value pair $(\widehat{k}, \widehat{x}[\widehat{k}])$

$$\mathsf{Pr}\left(\mathcal{H}_{\mathrm{S}}(\widehat{k}, \widehat{x}[\widehat{k}]) \leftarrow \mathcal{H}_{\mathrm{M}}\right) = \mathsf{Pr}\left(\frac{1}{P}\left\|\boldsymbol{y} - \widehat{x}[\widehat{k}]\mathbf{s}_{\widehat{k}}\right\|^2 \leq (1+\gamma)\sigma^2\right)$$

for some $\widehat{k}$ and $\widehat{x}[\widehat{k}]$. Then by letting $\mathbf{u} = \mathbf{S}(\mathbf{z} - \widehat{x}[\widehat{k}]\mathbf{e}_{\widehat{k}})$, we can write the error probability as

$$\mathsf{Pr}\left(\frac{1}{P}\left\|\mathbf{u} + \mathbf{w}\right\|^2 \leq (1+\gamma)\sigma^2\right) \tag{96}$$

Since both $\mathbf{u}$ and $\mathbf{w}$ are random, we compute this probability according to the total probability law as follows

$$\begin{aligned}
&\mathsf{Pr}\left(\frac{1}{P}\left\|\mathbf{u} + \mathbf{w}\right\|^2 \leq (1+\gamma)\sigma^2\right) \qquad (97)\\
&= \mathsf{Pr}\left(\frac{1}{P}\left\|\mathbf{u} + \mathbf{w}\right\|^2 \leq (1+\gamma)\sigma^2 \Big| \frac{\|\mathbf{u}\|^2}{P} \geq 2\gamma\sigma^2\right) \times \mathsf{Pr}\left(\frac{\|\mathbf{u}\|^2}{P} \geq 2\gamma\sigma^2\right)\\
&\quad + \mathsf{Pr}\left(\frac{1}{P}\left\|\mathbf{u} + \mathbf{w}\right\|^2 \leq (1+\gamma)\sigma^2 \Big| \frac{\|\mathbf{u}\|^2}{P} \leq 2\gamma\sigma^2\right) \times \mathsf{Pr}\left(\frac{\|\mathbf{u}\|^2}{P} \leq 2\gamma\sigma^2\right)\\
&\leq \mathsf{Pr}\left(\frac{1}{P}\left\|\mathbf{u} + \mathbf{w}\right\|^2 \leq (1+\gamma)\sigma^2 \Big| \frac{\|\mathbf{u}\|^2}{P} \geq 2\gamma\sigma^2\right) + \mathsf{Pr}\left(\frac{\|\mathbf{u}\|^2}{P} \leq 2\gamma\sigma^2\right),
\end{aligned}$$

where the first term is basically the single-ton verification error rate when the multi-ton has sufficiently large energy while the second term is the probability of any multi-ton not having sufficiently large energy. In the following, we bound these two probabilities separately with exponential tails.

Lemma 11 can be directly used to bound the first term in (97), by letting $\tau_2 = (1+\gamma)\sigma^2$. Note that the first term is conditioned on the event where $\|\mathbf{u}\|^2/P \geq 2\gamma\sigma^2$, therefore the minimum normalized non-centrality parameter can be obtained as $\nu_{\min} = \min_{\mathbf{u}} \|\mathbf{u}\|^2/P\sigma^2 = 2\gamma$. Clearly, the condition for the threshold in (147) holds for Corollary 3, and thus the first term can be bounded accordingly as

$$\mathsf{Pr}\left(\frac{1}{P}\left\|\mathbf{u} + \mathbf{w}\right\|^2 \leq (1+\gamma)\sigma^2 \Big| \frac{\|\mathbf{u}\|^2}{P} \geq 2\gamma\sigma^2\right) \leq \exp\left(-\frac{P}{4}\frac{\gamma^2}{1+4\gamma}\right). \tag{98}$$

Now we examine the probability of a multi-ton not having sufficiently large energy, or namely

$$\mathsf{Pr}\left(\frac{\|\mathbf{u}\|^2}{P} \leq 2\gamma\sigma^2\right) = \mathsf{Pr}\left(\frac{1}{P}\sum_{p\in[P]}|U[p]|^2 \leq 2\gamma\sigma^2\right). \tag{99}$$

Recall that

$$\mathbf{u} = \mathbf{S}(\mathbf{z} - \widehat{x}[\widehat{k}]\mathbf{e}_{\widehat{k}}) = \mathbf{S}\tilde{\mathbf{z}} \tag{100}$$

where $\tilde{\mathbf{z}} = [\tilde{Z}[0], \cdots, \tilde{Z}[N-1]]^T = \mathbf{z} - \widehat{x}[\widehat{k}]\mathbf{e}_{\widehat{k}}$ is at least 1-sparse. Since this probability depends on the specific matrix $\mathbf{S}$ used in the bin detector, we prove that this probability satisfies similar exponential tail bounds if we use the matrix $\mathbf{S}$ given by the `RandomNoisy` design, the `FourierNoisy` design and the `BinaryNoisy` design.

E.2.1 The `RandomNoisy` Design: the Random Matrix Ensemble

Lemma 6. *Given $\boldsymbol{\phi}_p := \mathbf{S}^T_{(p,:)}$ and $\tilde{\mathbf{z}}$, the variable $\xi_p = |U[p]|^2 = |\boldsymbol{\phi}_p^T \tilde{\mathbf{z}}|^2$ is sub-exponential with mean $\bar{\xi} = \|\tilde{\mathbf{z}}\|^2$ and an Orlicz-norm (i.e. the ψ_1-norm of sub-exponential variables) for some absolute constant $c_5 > 0$*

$$\xi_{\psi_1} = c_5 \bar{\xi}. \tag{101}$$

Proof. Note that one can re-write the variable as $\xi_p = \boldsymbol{\phi}_p^H \mathbf{Q} \boldsymbol{\phi}_p$ with $\mathbf{Q} = \tilde{\mathbf{z}}^* \tilde{\mathbf{z}}^T$. It is clear that ξ_p is bounded and hence it is sub-exponential with mean

$$\bar{\xi} = \mathbb{E}\left[\boldsymbol{\phi}_p^H \mathbf{Q} \boldsymbol{\phi}_p\right] = \mathrm{Tr}(\mathbf{Q}) = \|\tilde{\mathbf{z}}\|^2. \tag{102}$$

To compute its Orlicz-norm, we only need to find the constant ξ_{ψ_1} such that the following holds:

$$\mathsf{Pr}\left(|\xi_p - \bar{\xi}| > t\right) < 2\exp\left(-\frac{t}{\xi_{\psi_1}}\right).$$

Since $\boldsymbol{\phi}_p$ contains i.i.d. sub-gaussian variables, we can apply the Hanson-Wright inequality to obtain

$$\mathsf{Pr}\left(|\xi_p - \bar{\xi}| > t\right) = \mathsf{Pr}\left(\left|\boldsymbol{\phi}_p^H \mathbf{Q} \boldsymbol{\phi}_p - \mathbb{E}\left[\boldsymbol{\phi}_p^H \mathbf{Q} \boldsymbol{\phi}_p\right]\right| > t\right) \leq 2\exp\left(-\frac{t}{c_5 \|\mathbf{Q}\|_F}\right)$$

for some $c_5 > 0$. Since $\|\mathbf{Q}\|_F = \|\tilde{\mathbf{z}}\|^2$, we can readily obtain the Orlicz-norm of the variable $\xi_{\psi_1} = c_5 \bar{\xi}$. □

By Lemma 6, the variable $\xi_p = |U[p]|^2$ is sub-exponential with mean $\bar{\xi} = \|\tilde{\mathbf{z}}\|^2$ and an Orlicz-norm $\xi_{\psi_1} = c_5 \bar{\xi}$. Using the Bernstein-type inequality, then for any $t > 0$ we have

$$\mathsf{Pr}\left(\left|\frac{1}{P}\sum_{p\in[P]} (\xi_p - \bar{\xi})\right| \geq t\right) \leq 2\exp\left(-c_6 \frac{Pt}{\bar{\xi}}\right)$$

for some constant c_6. By taking $t = \bar{\xi} - 2\gamma\sigma^2$ with $\gamma < \bar{\xi}/2\sigma^2$, we have

$$\mathsf{Pr}\left(\frac{1}{P}\sum_{p\in[P]} (\xi_p - \bar{\xi}) \leq -(\bar{\xi} - 2\gamma\sigma^2)\right) \leq 2\exp\left(-c_6 P \frac{(\bar{\xi} - 2\gamma\sigma^2)}{\bar{\xi}}\right) \tag{103}$$

$$= 2\exp\left[-c_6 P\left(1 - \frac{2\gamma\sigma^2}{\bar{\xi}}\right)\right]. \tag{104}$$

Since the probability is monotonically decreasing with respect to $\bar{\xi}$, we can substitute the minimum $\bar{\xi} = \|\tilde{\mathbf{z}}\|^2 \geq A_{\min}^2$ for any multi-ton into the above tail bound and obtain

$$\mathsf{Pr}\left(\frac{1}{P}\sum_{p\in[P]} \xi_p \leq 2\gamma\sigma^2\right) \leq 2\exp\left[-c_6 P\left(1 - \frac{2\gamma\sigma^2}{A_{\min}^2}\right)\right].$$

for some c_6, which holds as long as $\gamma < A_{\min}^2/2\sigma^2$.

E.2.2 The `RandomNoisy` Design: the Random DFT Ensemble

If the bin detection matrix $\mathbf{S}$ is chosen from the partially random DFT matrix ensemble in Definition 4, the random variable $U[p]$ can be shown to be a Gaussian random variable so that the probability in (99) can be obtained readily

according to Lemma 11 and Corollary 3. In the following, we show that $U[p]$ is a Gaussian random variable. Recall that

$$U[p] = \sum_{k\in[N]} F_k W^{n\ell_p} \tilde{Z}[n], \tag{105}$$

where $W = e^{\mathrm{i}\frac{2\pi}{N}}$ is the N-th root of unity given in the matrix $\mathbf{S}$ in (36). Since F_k's are i.i.d. Gaussian variables and $W^{n\ell_p}$ is a uniformly random phase, it is obvious that $U[p]$ is also Gaussian with zero mean. Therefore, the covariance of the variables $U[p]$ across $p \in [P]$ can be readily obtained as

$$\mathbb{E}\left[U^*[p]U[p']\right] = \sum_{n\in[N]}\sum_{n'\in[N]} \mathbb{E}\left[F_n^* W^{-n\ell_p}\tilde{Z}[n]F_{n'}W^{n'\ell_{p'}}\tilde{Z}[n']\right] \tag{106}$$

$$= \sum_{n\in[N]}\sum_{n'\in[N]} \tilde{Z}^*[n]\tilde{Z}[n']\mathbb{E}\left[F_n^* F_{n'}\right]\mathbb{E}\left[W^{n'\ell_{p'}-n\ell_p}\right] \tag{107}$$

$$= \|\tilde{\mathbf{z}}\|^2 \delta[p-p']. \tag{108}$$

Therefore, the covariance of $U[p]$ for $p \in [P]$ is a scaled identity matrix and thus they are uncorrelated. The Gaussianity further proves that $U[p]$'s are independent across $p \in [P]$. Lemma 11 and Corollary 3 can be directly used here by letting $\tau_2 = 2\gamma\sigma^2$ and $\mathbf{u} = \mathbf{0}$. The condition for the threshold in (147) holds as long as $\tau_2 \le \theta^2 = \|\tilde{\mathbf{z}}\|^2$, which requires the threshold parameter γ to be chosen as $0 < \gamma \le \frac{\|\tilde{\mathbf{z}}\|^2}{2\sigma^2}$. Since $\tilde{\mathbf{z}}$ is at least 1-sparse such that $\|\tilde{\mathbf{z}}\|^2 \ge A_{\min}^2$, we have

$$0 < \gamma \le \frac{A_{\min}^2}{2\sigma^2}. \tag{109}$$

Therefore the tail bound can be obtained as

$$\mathsf{Pr}\left(\frac{1}{P}\sum_{p\in[P]}|U[p]|^2 \le 2\gamma\sigma^2\right) \le \exp\left(-\frac{P}{4}\left(1-\frac{2\gamma\sigma^2}{A_{\min}^2}\right)\right).$$

E.2.3 The `FourierNoisy` Design

In the `FourierNoisy` design, the measurements used for zero-ton and single-ton verifications are obtained from the partially random DFT matrix in the `RandomNoisy` design. Hence the same tail bound derived in Appendix E.2.2 for the `RandomNoisy` design holds.

E.2.4 The `BinaryNoisy` Design

In the `BinaryNoisy` design, the measurements used for zero-ton and single-ton verifications are obtained from an i.i.d. random matrix consisting of Bernoulli random variables $\{\pm 1\}$ with probability $1/2$. This is clearly included in the random matrix ensemble in the `RandomNoisy` design and hence the same tail bound derived in Appendix E.2.1 for the `RandomNoisy` design holds.

F Proof of Missed Verification Rates in Proposition 4

The missed verification events occur if the zero-ton or single-ton verifications pass when the ground truth is a single-ton $\mathcal{H}_{\mathsf{S}}(k, x[k])$ for some $k \in [N]$:

$$\boldsymbol{y} = \mathbf{S}\mathbf{z} + \mathbf{w} = \mathbf{s}_k x[k] + \mathbf{w}. \tag{110}$$

F.1 Detecting a Single-ton as a Zero-ton

This event occurs when the ground truth is a single-ton $\mathcal{H}_{\mathrm{S}}(k, x[k])$ with an index-value pair $(k, x[k])$, but the zero-ton test is passed such that it is regarded as a zero-ton. Therefore,

$$\mathsf{Pr}\left(\mathcal{H}_{\mathrm{Z}} \leftarrow \mathcal{H}_{\mathrm{S}}(k, x[k])\right) = \mathsf{Pr}\left(\frac{1}{P}\left\|x[k]\mathbf{s}_k + \mathbf{w}\right\|^2 \leq (1+\gamma)\sigma^2\right).$$

Similarly, by letting $\mathbf{u} = x[k]\mathbf{s}_k$, we have

$$\begin{aligned}
&\mathsf{Pr}\left(\frac{1}{P}\left\|\mathbf{u}+\mathbf{w}\right\|^2 \leq (1+\gamma)\sigma^2\right) \qquad (111)\\
&\leq \mathsf{Pr}\left(\frac{1}{P}\left\|\mathbf{u}+\mathbf{w}\right\|^2 \leq (1+\gamma)\sigma^2 \Big| \frac{\left\|\mathbf{u}\right\|^2}{P} \geq 2\gamma\sigma^2\right) + \mathsf{Pr}\left(\frac{\left\|\mathbf{u}\right\|^2}{P} \leq 2\gamma\sigma^2\right),
\end{aligned}$$

For the first term, Lemma 11 and Corollary 3 hold with the threshold $\tau_2 = (1+\gamma)\sigma^2$ and the minimum normalized non-centrality parameter $\nu_{\min} = 2\gamma$. This gives the following bound:

$$\mathsf{Pr}\left(\frac{1}{P}\left\|\mathbf{u}+\mathbf{w}\right\|^2 \leq (1+\gamma)\sigma^2 \Big| \frac{\left\|\mathbf{u}\right\|^2}{P} \geq 2\gamma\sigma^2\right) \leq \exp\left(-\frac{P}{4}\frac{\gamma^2}{1+4\gamma}\right). \qquad (112)$$

The second term $\mathsf{Pr}\left(\frac{\left\|\mathbf{u}\right\|^2}{P} \leq 2\gamma\sigma^2\right)$ has been bounded for different designs respectively for the `RandomNoisy` design in Appendix E.2.1 and E.2.2, the `FourierNoisy` design in Appendix E.2.3 and the `BinaryNoisy` design in Appendix E.2.4.

F.2 Detecting a Single-ton as a Multi-ton

This event occurs when the ground truth is a single-ton $\mathcal{H}_{\mathrm{S}}(k, x[k])$ with an index-value pair $(k, x[k])$, but the single-ton verification fails for some index-value pair $(\widehat{k}, \widehat{x}[\widehat{k}])$ obtained from the single-ton search:

$$\mathsf{Pr}\left(\mathcal{H}_{\mathrm{M}} \leftarrow \mathcal{H}_{\mathrm{S}}(k, x[k])\right) = \mathsf{Pr}\left(\frac{1}{P}\left\|\boldsymbol{y} - \widehat{x}[\widehat{k}]\mathbf{s}_{\widehat{k}}\right\|^2 \geq (1+\gamma)\sigma^2\right).$$

Since the single-ton search may or may not return the correct index-value pair, this probability is obtained as

$$\begin{aligned}
&\mathsf{Pr}\left(\frac{1}{P}\left\|\boldsymbol{y} - \widehat{x}[\widehat{k}]\mathbf{s}_{\widehat{k}}\right\|^2 \geq (1+\gamma)\sigma^2\right)\\
&= \mathsf{Pr}\left(\frac{1}{P}\left\|\boldsymbol{y} - \widehat{x}[\widehat{k}]\mathbf{s}_{\widehat{k}}\right\|^2 \geq (1+\gamma)\sigma^2 \Big| \widehat{k} \neq k \text{ or } \widehat{x}[\widehat{k}] \neq x[k]\right) \times \mathsf{Pr}\left(\widehat{k} \neq k \text{ or } \widehat{x}[\widehat{k}] \neq x[k]\right)\\
&\quad + \mathsf{Pr}\left(\frac{1}{P}\left\|\boldsymbol{y} - \widehat{x}[\widehat{k}]\mathbf{s}_{\widehat{k}}\right\|^2 \geq (1+\gamma)\sigma^2 \Big| \widehat{k} = k \text{ and } \widehat{x}[\widehat{k}] = x[k]\right) \times \mathsf{Pr}\left(\widehat{k} = k \text{ and } \widehat{x}[\widehat{k}] = x[k]\right)\\
&\leq \mathsf{Pr}\left(\widehat{k} \neq k \text{ or } \widehat{x}[\widehat{k}] \neq x[k]\right) + \mathsf{Pr}\left(\frac{1}{P}\left\|\boldsymbol{y} - \widehat{x}[\widehat{k}]\mathbf{s}_{\widehat{k}}\right\|^2 \geq (1+\gamma)\sigma^2 \Big| \widehat{k} = k \text{ and } \widehat{x}[\widehat{k}] = x[k]\right),
\end{aligned}$$

where the first term is basically the single-ton search error rate while the second term is the single-ton verification error rate conditioned on the fact that the single-ton search is partially correct. Now we bound the two error rates with exponential tails in the following.

F.2.1 Bounding $\mathsf{Pr}\left(\widehat{k} \neq k \text{ or } \widehat{x}[\widehat{k}] \neq x[k]\right)$

The first term is essentially the single-ton search error probability. Since different sensing matrices $\mathbf{S}$ are used in the single-ton tests, we have the following lemmas for this error probability.

Lemma 7 (**Single-ton Search Error Probability of the** `RandomNoisy` **Design).** *Using the* `RandomNoisy` *design, the probability of wrongly detecting the index-value pair is upper bounded as*

$$\mathsf{Pr}\left(\widehat{k} \neq k \text{ or } \widehat{x}[\widehat{k}] \neq x[k]\right) < \frac{3}{2}e^{-P\gamma} + 6e^{-c_6 P\left(1-\frac{\gamma\sigma^2}{\rho^2}\right)} \tag{113}$$

for some $c_6 > 0$ *and* $0 < \gamma < A_{\min}^2/\sigma^2$.

Proof. See Appendix G.1. □

Lemma 8 (**Single-ton Search Error Probability of the** `FourierNoisy` **Design).** *Using the* `FourierNoisy` *design, the probability of wrongly detecting the index-value pair is upper bounded as*

$$\mathsf{Pr}\left(\widehat{k} \neq k \text{ or } \widehat{x}[\widehat{k}] \neq x[k]\right) \leq 8Pe^{-\frac{A_{\min}^2}{4\sigma^2}Q^3} + 2e^{-\frac{P}{4}\left(1-\frac{\gamma\sigma^2}{\rho^2}\right)} \tag{114}$$

for some $0 < \gamma < \rho^2/\sigma^2$.

Proof. See Appendix G.2. □

Lemma 9 (**Single-ton Search Error Probability of the** `BinaryNoisy` **Design).** *Using the* `BinaryNoisy` *design, the probability of wrongly detecting the index-value pair is upper bounded as*

$$\mathsf{Pr}\left(\widehat{k} \neq k \text{ or } \widehat{x}[\widehat{k}] \neq x[k]\right) \leq 2e^{-\frac{(\beta-\mathbb{P}_{\mathsf{e}})^2}{2\mathbb{P}_{\mathsf{e}}(1-\mathbb{P}_{\mathsf{e}})}P} + 6e^{-c_6 P\left(1-\frac{\gamma\sigma^2}{\rho^2}\right)} \tag{115}$$

for some $c_6, c_7, c_8 > 0$ *and* $0 < \gamma < A_{\min}^2/\sigma^2$.

Proof. See Appendix G.3. □

F.2.2 Bounding $\mathsf{Pr}\left(\frac{1}{P}\left\|\boldsymbol{y} - \widehat{x}[\widehat{k}]\mathbf{s}_{\widehat{k}}\right\|^2 \geq (1+\gamma)\sigma^2 \middle| \widehat{k} = k \text{ and } \widehat{x}[\widehat{k}] = x[k]\right)$

When the bin is a single-ton, it is clear that this probability is equivalent to

$$\mathsf{Pr}\left(\frac{1}{P}\left\|\boldsymbol{y} - \widehat{x}[\widehat{k}]\mathbf{s}_{\widehat{k}}\right\|^2 \geq (1+\gamma)\sigma^2 \middle| \widehat{k} = k \text{ and } \widehat{x}[\widehat{k}] = x[k]\right) = \mathsf{Pr}\left(\frac{1}{P}\left\|\mathbf{w}\right\|^2 \geq (1+\gamma)\sigma^2\right),$$

which can be directly obtained from Lemma 11 by letting $\tau_1 = (1+\gamma)\sigma^2$, $\mathbf{u} = 0$ and $\theta^2 = \sigma^2$. Therefore, the tail bound can be readily obtained as

$$\mathsf{Pr}\left(\frac{1}{P}\left\|\boldsymbol{y} - \widehat{x}[\widehat{k}]\mathbf{s}_{\widehat{k}}\right\|^2 \geq (1+\gamma)\sigma^2 \middle| \widehat{k} = k \text{ and } \widehat{x}[\widehat{k}] = x[k]\right) < \exp\left(-\frac{P}{4}\left(\sqrt{1+2\gamma}-1\right)^2\right).$$

G Proof of Single-ton Search Error Probability in Lemma 7, 8 and 9

G.1 Single-ton Search using the `RandomNoisy` Design

The exhaustive search fails when the estimate $\widehat{k} \neq k$ leads to a smaller residual such that the single-ton verification passes. Thus, the probability of error for this search can be obtained by a union bound as

$$\mathsf{Pr}\left(\widehat{k} \neq k \text{ or } \widehat{x}[\widehat{k}] \neq x[k]\right) \tag{116}$$

$$\leq \mathsf{Pr}\left(\widehat{k} \neq k \text{ and } \widehat{x}[\widehat{k}] \neq x[k]\right) + \mathsf{Pr}\left(\widehat{k} \neq k \text{ and } \widehat{x}[\widehat{k}] = x[k]\right) + \mathsf{Pr}\left(\widehat{k} = k \text{ and } \widehat{x}[\widehat{k}] \neq x[k]\right). \tag{117}$$

These error event occur when the generalized likelihood ratio test fails. Given an estimated pair $(\widehat{k}, \widehat{x}[\widehat{k}])$, by substituting the single-ton model for $\boldsymbol{y} = \mathbf{s}_k x[k] + \mathbf{w} \sim \mathcal{H}_{\mathrm{S}}(k, x[k])$, the error probability can be obtained as

$$\mathsf{Pr}\left(\left\|\boldsymbol{y} - \widehat{x}[\widehat{k}]\mathbf{s}_{\widehat{k}}\right\|^2 < \|\boldsymbol{y} - x[k]\mathbf{s}_k\|^2\right) = \mathsf{Pr}\left(\Re\left[(x^*[k]\mathbf{s}_k^\dagger - \widehat{x}^*[\widehat{k}]\mathbf{s}_{\widehat{k}}^\dagger)\mathbf{w}\right] > \frac{\left\|x[k]\mathbf{s}_k - \widehat{x}[\widehat{k}]\mathbf{s}_{\widehat{k}}\right\|^2}{2}\right) \tag{118}$$

under different scenarios given by (116). Let $\mathbf{u} = \mathbf{S}\tilde{\mathbf{z}}$ with $\tilde{\mathbf{z}} := x[k]\mathbf{e}_k - \widehat{x}[\widehat{k}]\mathbf{e}_{\widehat{k}}$, then this can be further bounded using the total probability law as

$$\begin{aligned}
&\mathsf{Pr}\left(\Re\left[(x^*[k]\mathbf{s}_k^\dagger - \widehat{x}^*[\widehat{k}]\mathbf{s}_{\widehat{k}}^\dagger)\mathbf{w}\right] > \frac{\left\|x[k]\mathbf{s}_k - \widehat{x}[\widehat{k}]\mathbf{s}_{\widehat{k}}\right\|^2}{2}\right) \\
&= \mathsf{Pr}\left(\Re\left[\frac{\mathbf{u}^\dagger\mathbf{w}}{P}\right] > \frac{\|\mathbf{u}\|^2}{2P}\middle|\frac{\|\mathbf{u}\|^2}{P} \geq 2\gamma\sigma^2\right)\mathsf{Pr}\left(\frac{\|\mathbf{u}\|^2}{P} \geq 2\gamma\sigma^2\right) \\
&\quad + \mathsf{Pr}\left(\Re\left[\frac{\mathbf{u}^\dagger\mathbf{w}}{P}\right] > \frac{\|\mathbf{u}\|^2}{2P}\middle|\frac{\|\mathbf{u}\|^2}{P} \leq 2\gamma\sigma^2\right)\mathsf{Pr}\left(\frac{\|\mathbf{u}\|^2}{P} \leq 2\gamma\sigma^2\right) \\
&\leq \mathsf{Pr}\left(\Re\left[\frac{\mathbf{u}^\dagger\mathbf{w}}{P}\right] > \frac{\|\mathbf{u}\|^2}{2P}\middle|\frac{\|\mathbf{u}\|^2}{P} \geq 2\gamma\sigma^2\right) + \mathsf{Pr}\left(\frac{\|\mathbf{u}\|^2}{P} \leq 2\gamma\sigma^2\right).
\end{aligned} \tag{119}$$

Since $\mathbf{w}$ is Gaussian with zero mean and variance σ^2, the variable $\Re\left[\mathbf{u}^\dagger\mathbf{w}/P\right]$ is also Gaussian with zero mean and variance $\|\mathbf{u}\|^2\sigma^2/P^2$. Therefore, by normalizing the variance on both sides of the first term and applying the Chernoff bound, we have

$$\begin{aligned}
\mathsf{Pr}\left(\Re\left[\frac{\mathbf{u}^\dagger\mathbf{w}}{P}\right] > \frac{\|\mathbf{u}\|^2}{2P}\middle|\frac{\|\mathbf{u}\|^2}{P} \geq 2\gamma\sigma^2\right) &= \mathsf{Pr}\left(\mathcal{N}(0,1) > \frac{\|\mathbf{u}\|}{2\sigma}\middle|\frac{\|\mathbf{u}\|^2}{P} \geq 2\gamma\sigma^2\right) \\
&\leq \mathsf{Pr}\left(\mathcal{N}(0,1) > \sqrt{2P\gamma}\right) \\
&\leq \frac{1}{2}\exp\left(-P\gamma\right).
\end{aligned}$$

Further, the second term $\mathsf{Pr}\left(\|\mathbf{u}\|^2/P \leq 2\gamma\sigma^2\right)$ in (119) is characterized by $\mathbf{u} = \mathbf{S}\tilde{\mathbf{z}}$ with $\tilde{\mathbf{z}} := x[k]\mathbf{e}_k - \widehat{x}[\widehat{k}]\mathbf{e}_{\widehat{k}}$, which depends on the conditions given by (116). Under these conditions, the probabilities can be bounded in a similar fashion to that in Appendix E.2.1 andE.2.2. Since we have $\mathbf{u} = \mathbf{S}\tilde{\mathbf{z}}$ and in each case

$$\|\tilde{\mathbf{z}}\|^2 \geq \begin{cases} 2A_{\min}^2, & \widehat{k} \neq k \text{ and } \widehat{x}[\widehat{k}] \neq x[k] \\ 2A_{\min}^2, & \widehat{k} \neq k \text{ and } \widehat{x}[\widehat{k}] = x[k] \\ \rho^2, & \widehat{k} = k \text{ and } \widehat{x}[\widehat{k}] \neq x[k] \end{cases} \tag{120}$$

The probability is always upper bounded by

$$\mathsf{Pr}\left(\|\mathbf{u}\|^2/P \leq 2\gamma\sigma^2\right) \leq 2\exp\left(-c_6 P\left(1 - \frac{2\gamma\sigma^2}{\rho^2}\right)\right) \tag{121}$$

for some $c_6 > 0$ and therefore, the total probability in (116) can be bounded by

$$\mathsf{Pr}\left(\widehat{k} \neq k \text{ or } \widehat{x}[\widehat{k}] \neq x[k]\right) \leq \frac{3}{2}\exp\left(-P\gamma\right) + 6\exp\left(-c_6 P\left(1 - \frac{2\gamma\sigma^2}{\rho^2}\right)\right). \tag{122}$$

G.2 Single-ton Search using the `FourierNoisy` Design

In this case, the single-ton search is done in two steps, where the first step is to estimate the index in terms of the frequency while the second step is to estimate the associated coefficient. Therefore, the probability of error for this search is

$$\begin{aligned}\mathsf{Pr}\left(\widehat{k} \neq k \text{ or } \widehat{x}[\widehat{k}] \neq x[k]\right) &\leq \mathsf{Pr}\left(\widehat{k} \neq k \text{ and } \widehat{x}[\widehat{k}] \neq x[k]\right) + \mathsf{Pr}\left(\widehat{k} \neq k \text{ and } \widehat{x}[\widehat{k}] = x[k]\right) + \mathsf{Pr}\left(\widehat{k} = k \text{ and } \widehat{x}[\widehat{k}] \neq x[k]\right) \\ &\leq 2\mathsf{Pr}\left(\widehat{k} \neq k\right) + \mathsf{Pr}\left(\widehat{k} = k \text{ and } \widehat{x}[\widehat{k}] \neq x[k]\right). \end{aligned} \tag{123}$$

In the following, we first bound the first term $\mathsf{Pr}\left(\widehat{k} \neq k\right)$ and then bound the second term.

G.2.1 Bounding the first term $\mathsf{Pr}\left(\widehat{k} \neq k\right)$

The estimate $\widehat{\omega}_p$ in cluster p is $2^p\omega$ with an error $\Delta_p = \widehat{\omega}_p - 2^p\omega$ characterized by Lemma 4, modulo 2π.

Corollary 2. *The estimation error Δ_p satisfies*

$$\mathsf{Pr}\left(|\Delta_p| > \frac{\pi}{2}\right) \leq 4\exp\left(-\frac{A_{\min}^2}{4\sigma^2}Q^3\right). \tag{124}$$

Proof. From Lemma 4, for any given amplitude $|x[k]|$ and frequency ω in the presence of noise with variance σ^2, the estimation error Δ from the linear frequency estimator satisfies

$$\mathsf{Pr}\left(|\Delta| > \frac{\pi}{2}\right) \leq 4\exp\left(-\frac{\pi}{6}\frac{|x[k]|^2}{\sigma^2}Q(Q^2-1)\right). \tag{125}$$

Since $|x[k]| \geq A_{\min}$, we have

$$\frac{|x[k]|^2}{\sigma^2} \geq \frac{A_{\min}^2}{\sigma^2}. \tag{126}$$

Note that $\pi/3 \geq 1$ and $Q^2 - 1 \geq Q^2/2$ for any $Q \geq 2$, the tail bound can be simplified as in (124). □

It is clear that the modulo 2π operation leads to ambiguities

$$2^p\omega = \widehat{\omega}_p + 2\pi\ell + \Delta_p \tag{127}$$

for some $\ell = 0, 1, \cdots, 2^p - 1$. Therefore, the estimated frequency $\widehat{\omega}_p$ needs to be unwrapped by a factor of 2^p as

$$\omega = \frac{\widehat{\omega}_p}{2^p} + \frac{\pi}{2^{p-1}}\ell + \frac{\Delta_p}{2^p}. \tag{128}$$

The estimate $\widehat{\omega}_p$ reduces the uncertainty Ω_p to $\omega \in \Omega_p/2^p$ by unwrapping:

$$\frac{\Omega_p}{2^p} := \bigcup_{\ell=0}^{2^p-1}\left[\frac{\pi}{2^{p-1}}\ell + \frac{\widehat{\omega}_p}{2^p} - \frac{\pi}{2^p}, \frac{\pi}{2^{p-1}}\ell + \frac{\widehat{\omega}_p}{2^p} + \frac{\pi}{2^p}\right].$$

Furthermore, it can be seen that although the ambiguities $\frac{\pi}{2^{p-1}}\ell$ increase as the unwrapping factor 2^p grows, the certainty improves exponentially $1/2^p$. This means that the size of the intersection Ω in (47) satisfies

$$|\Omega| \leq \frac{\pi}{2^{P-1}}. \tag{129}$$

According to Lemma 4 and the tail bound of the estimation error Δ_p in (124), the probability of the true frequency ω lying outside the unwrapped certainty region $\Omega_p/2^p$ can be bounded by

$$\mathsf{Pr}\left(\omega \notin \frac{\Omega_p}{2^p}\right) \leq 4e^{-\frac{A_{\min}^2}{4\sigma^2}Q^3}. \tag{130}$$

Applying a union bound on the intersection Ω of all P clusters, we have

$$\mathsf{Pr}\left(\omega \notin \Omega\right) \leq 4P \exp\left(-\frac{A_{\min}^2}{4\sigma^2}Q^3\right). \tag{131}$$

Note that the frequency in the single-ton search is of the form $2\pi k/N$ for some integer valued $k \in [N]$. Therefore, as long as $|\Omega| \leq \frac{\pi}{2^{P-1}} = O(1/N)$, there are only $O(1)$ possible locations k in that region with high probability. From the tail bound in (130), as long as $P = O(\log_2 N)$ and $Q = O(\log^{1/3} N)$, the reduced certainty region satisfies $|\Omega| \leq \frac{\pi}{2^{P-1}} = O(1/N)$ with probability at least $1 - O(1/N)$. Therefore,

$$\mathsf{Pr}\left(\widehat{k} \neq k\right) \leq 4P \exp\left(-\frac{A_{\min}^2}{4\sigma^2}Q^3\right). \tag{132}$$

G.2.2 Bounding the second term $\mathsf{Pr}\left(\widehat{k} = k \text{ and } \widehat{x}[\widehat{k}] \neq x[k]\right)$

By letting $\widehat{k} = k$ in (116), the error event in this case can be easily obtained as

$$\mathsf{Pr}\left(\widehat{k} = k \text{ and } \widehat{x}[\widehat{k}] \neq x[k]\right) = \mathsf{Pr}\left(\Re\left[(x^*[k] - \widehat{x}^*[k])\mathbf{s}_k^\dagger \mathbf{w}\right] > \frac{|x[k] - \widehat{x}[k]|^2 \left\|\mathbf{s}_k\right\|^2}{2}\right).$$

Let $\mathbf{u} = \mathbf{S}\tilde{\mathbf{z}}$ with $\tilde{\mathbf{z}} = (x[k] - \widehat{x}[k])\mathbf{e}_k$. Similar to (119), we have

$$\mathsf{Pr}\left(\widehat{x}[k] \neq x[k] \middle| \widehat{k} = k\right) \leq \mathsf{Pr}\left(\frac{\|\mathbf{u}\|^2}{2P} \middle| \frac{\|\mathbf{u}\|^2}{P} \geq 2\gamma\sigma^2\right) + \mathsf{Pr}\left(\frac{\|\mathbf{u}\|^2}{P} \leq 2\gamma\sigma^2\right).$$

Then the first term follows directly from Appendix G.1 as

$$\mathsf{Pr}\left(\Re\left[\frac{\mathbf{u}^\dagger \mathbf{w}}{P}\right] > \frac{\|\mathbf{u}\|^2}{2P} \middle| \frac{\|\mathbf{u}\|^2}{P} \geq 2\gamma\sigma^2\right) \leq \frac{1}{2}\exp\left(-P\gamma^2\right) \tag{133}$$

and the second term $\mathsf{Pr}\left(\frac{\|\mathbf{u}\|^2}{P} \leq 2\gamma\sigma^2\right)$ in (119) has been bounded for the `RandomNoisy` design in Appendix E.2.3 by using $\|\tilde{\mathbf{z}}\|^2 \geq \rho^2$ as

$$\mathsf{Pr}\left(\frac{\|\mathbf{u}\|^2}{P} \leq 2\gamma\sigma^2\right) \leq 2\exp\left(-\frac{P}{4}\left(1 - \frac{\gamma\sigma^2}{\rho^2}\right)\right) \tag{134}$$

for some $0 < \gamma < \rho^2/\sigma^2$.

G.3 Single-ton Search using the `BinaryNoisy` Design

Similar to the `FourierNoisy` design, the first step in our `BinaryNoisy` design is to estimate the index in terms of the frequency while the second step is to estimate the associated coefficient. Therefore, from (123) we have

$$\mathsf{Pr}\left(\widehat{k} \neq k \text{ or } \widehat{x}[\widehat{k}] \neq x[k]\right) \leq 2\mathsf{Pr}\left(\widehat{k} \neq k\right) + \mathsf{Pr}\left(\widehat{k} = k \text{ and } \widehat{x}[\widehat{k}] \neq x[k]\right),$$

where the second term can be upper bounded as in (134). Therefore we focus on bounding the first term $\mathsf{Pr}\left(\widehat{k} \neq k\right)$ for the `BinaryNoisy` design. This probability is essentially the probability of wrongly decoding the information bits $\mathbf{b}_k$, which is a standard channel coding problem.

There are $P = O(n)$ coded bits of the n information bits in $\mathbf{b}_k$. As a result, if the linear code $\mathbf{c}_k = \mathbf{G}\mathbf{b}_k$ has a minimum distance of βP, the probability of decoding error corresponds to the probability of the BSC flipping more than βP bits:

$$\mathsf{Pr}\left(\widehat{k} \neq k\right) = \mathsf{Pr}\left(\mathbf{1}^T\mathbf{e} > \beta P\right). \tag{135}$$

Since the BSC flips each bit independently with probability at most $\mathbb{P}_{\mathsf{e}}$, we can approximate $\mathbf{1}^T\mathbf{e}/P$ as a Gaussian variable $\mathcal{N}(\mathbb{P}_{\mathsf{e}}, \mathbb{P}_{\mathsf{e}}(1-\mathbb{P}_{\mathsf{e}})/P)$ and therefore

$$\mathsf{Pr}\left(\widehat{k} \neq k\right) = \mathsf{Pr}\left(\frac{\mathbf{1}^T\mathbf{e}}{P} > \beta\right) \approx \mathsf{Pr}\left(\mathcal{N}(0,1) > \frac{(\beta - \mathbb{P}_{\mathsf{e}})}{\sqrt{\mathbb{P}_{\mathsf{e}}(1-\mathbb{P}_{\mathsf{e}})/P}}\right) \leq 2e^{-\frac{(\beta-\mathbb{P}_{\mathsf{e}})^2}{2\mathbb{P}_{\mathsf{e}}(1-\mathbb{P}_{\mathsf{e}})}P} \tag{136}$$

as long as $\beta > \mathbb{P}_{\mathsf{e}}$. For a given minimum distance βP, there exist many codes that satisfy the minimum distance properties. Excellent examples include the class of *expander codes* or *LDPC codes* that also allow for linear time decoding. It has been well established [66] that for a given minimum distance βP, one can construct such expander codes with high probability. Thus we can randomly generate the matrix $\mathbf{G}$ offline and verify its minimum distance, and then keep using it for all instances.

H Proof of Crossed Verification Rates in Proposition 5

Crossed verification events occur when the ground truth is a single-ton $\mathcal{H}_{\mathrm{S}}(k, x[k])$ for some $k \in [N]$, and therefore, the probabilities can be obtained from the equivalent model in (35) under the single-ton hypothesis

$$\boldsymbol{y} = \mathbf{s}_k x[k] + \mathbf{w}. \tag{137}$$

A crossed verification implies that the single-ton search returns a wrong index-value pair, and yet the single-ton verification is passed with the wrong estimates.

$$\begin{aligned}\mathsf{Pr}\left(\mathcal{H}_{\mathrm{S}}(\widehat{k}, \widehat{x}[\widehat{k}]) \leftarrow \mathcal{H}_{\mathrm{S}}(k, x[k])\right) &= \mathsf{Pr}\left(\frac{1}{P}\left\|\boldsymbol{y} - \widehat{x}[\widehat{k}]\mathbf{s}_{\widehat{k}}\right\|^2 \leq (1+\gamma)\sigma^2 \middle| \widehat{k} \neq k, \widehat{x}[\widehat{k}] \neq x[k]\right) \\ &= \mathsf{Pr}\left(\frac{1}{P}\left\|x[k]\mathbf{s}_k - \widehat{x}[\widehat{k}]\mathbf{s}_{\widehat{k}} + \mathbf{w}\right\|^2 \leq (1+\gamma)\sigma^2\right)\end{aligned}$$

for some $\widehat{k} \neq k$ and $\widehat{x}[\widehat{k}] \neq x[k]$. This can be re-written as

$$\mathsf{Pr}\left(\frac{1}{P}\left\|\mathbf{u} + \mathbf{w}\right\|^2 \leq (1+\gamma)\sigma^2\right)$$

where $\mathbf{u} = \mathbf{S}\tilde{\mathbf{z}}$ and $\tilde{\mathbf{z}} := x[k]\mathbf{e}_k - \widehat{x}[\widehat{k}]\mathbf{e}_{\widehat{k}}$ is a 2-sparse vector. Similar to (97), we have

$$\mathsf{Pr}\left(\frac{1}{P}\left\|\mathbf{u} + \mathbf{w}\right\|^2 \leq (1+\gamma)\sigma^2\right) \leq \mathsf{Pr}\left(\frac{1}{P}\left\|\mathbf{u} + \mathbf{w}\right\|^2 \leq (1+\gamma)\sigma^2 \middle| \frac{\|\mathbf{u}\|^2}{P} \geq 2\gamma\sigma^2\right) + \mathsf{Pr}\left(\frac{\|\mathbf{u}\|^2}{P} \leq 2\gamma\sigma^2\right).$$

Similar to (98), the first term can be bounded as

$$\mathsf{Pr}\left(\frac{1}{P}\left\|\mathbf{u} + \mathbf{w}\right\|^2 \leq (1+\gamma)\sigma^2 \middle| \frac{\|\mathbf{u}\|^2}{P} \geq 2\gamma\sigma^2\right) \leq \exp\left(-\frac{P}{4}\frac{\gamma^2}{1+4\gamma}\right). \tag{138}$$

The second term $\mathsf{Pr}\left(\frac{\|\mathbf{u}\|^2}{P} \leq 2\gamma\sigma^2\right)$ has been bounded for different designs respectively for the `RandomNoisy` design in Appendix E.2.1 and E.2.2, the `FourierNoisy` design in Appendix E.2.3 and the `BinaryNoisy` design in Appendix E.2.4.

I Tail Bounds

Here we derive some tail bounds that are useful in our analysis.

Lemma 10 (Non-central Chi-Square Tail Bounds in [67])**.** *Let $Z \sim \chi^2_D$ be a non-central chi square variable with D degrees of freedom and non-centrality parameter $\nu \geq 0$. Then for all $z \geq 0$, the following tail bounds hold:*

$$\mathsf{Pr}\left(Z \geq (D+\nu) + 2\sqrt{(D+2\nu)z} + 2z\right) \leq \exp(-z)$$
$$\mathsf{Pr}\left(Z \leq (D+\nu) - 2\sqrt{(D+2\nu)z}\right) \leq \exp(-z)$$

Lemma 11. *Given $\mathbf{u} = [u[0], \cdots, u[P-1]]^T$ and a vector $\mathbf{w} = [w[0], \cdots, w[P-1]]^T$ with i.i.d. Gaussian variates $w[p] \sim \mathcal{N}(0, \theta^2)$ for all $p \in [P]$, the following tail bound holds:*

$$\mathsf{Pr}\left(\frac{1}{P}\left\|\mathbf{u}+\mathbf{w}\right\|^2 \geq \tau_1\right) \leq e^{-\frac{P}{4}\left(\sqrt{2\tau_1/\theta^2 - 1} - \sqrt{1+2\nu_0}\right)^2} \tag{139}$$

$$\mathsf{Pr}\left(\frac{1}{P}\left\|\mathbf{u}+\mathbf{w}\right\|^2 \leq \tau_2\right) \leq e^{-\frac{P}{4}\frac{\left(1+\nu_0-\tau_2/\theta^2\right)^2}{1+2\nu_0}} \tag{140}$$

for any τ_1 and τ_2 that satisfy

$$\tau_1 \geq \theta^2(1+\nu_0), \quad \tau_2 \leq \theta^2(1+\nu_0), \tag{141}$$

where ν_0 is the normalized non-centrality parameter given by

$$\nu_0 := \frac{\left\|\mathbf{u}\right\|^2}{P\theta^2}. \tag{142}$$

Proof. The quantity $\left\|\mathbf{u}+\mathbf{w}\right\|^2$ can be written element-wise as

$$\left\|\mathbf{u}+\mathbf{w}\right\|^2 = \sum_{p=0}^{P-1}\left(u[p]+w[p]\right)^2 \tag{143}$$

where each summand is a normal random variable with mean $u[p]$ and variance θ^2. Therefore, according to the definition of non-central chi-square variables, the quantity

$$\frac{\left\|\mathbf{u}+\mathbf{w}\right\|^2}{\theta^2} \sim \chi^2_P \tag{144}$$

is a non-central χ^2 random variable of P degrees of freedom with a non-centrality parameter

$$\nu = \sum_{p=0}^{P-1}\frac{|u[p]|^2}{\theta^2} = \frac{\left\|\mathbf{u}\right\|^2}{\theta^2}. \tag{145}$$

For notational convenience, we use the normalized non-centrality parameter ν_0 in (142) such that $\nu = P\nu_0$. Without loss of generality, let the thresholds τ_1 and τ_2 take the following form with respect to z_1 and z_2:

$$\tau_1 = \frac{\theta^2}{P}\left[(P+P\nu_0) + 2\sqrt{(P+2P\nu_0)z_1} + 2z_1\right]$$
$$\tau_2 = \frac{\theta^2}{P}\left[(P+P\nu_0) - 2\sqrt{(P+2P\nu_0)z_2}\right],$$

then the tail bounds in Lemma 10 can be obtained easily with respect to z_1 and z_2. Using (145), the corresponding z_1 and z_2 can be solved as

$$z_1 = \frac{P}{4}\left(\sqrt{2\tau_1/\theta^2 - 1} - \sqrt{1+2\nu_0}\right)^2$$
$$z_2 = \frac{P}{4}\frac{\left(1+\nu_0-\tau_2/\theta^2\right)^2}{1+2\nu_0}$$

as long as the thresholds τ_1 and τ_2 satisfy (141). Thus according to Lemma 10, we have the tail bounds in (139). □

Corollary 3. *Suppose that the normalized non-centrality parameter ν_0 in Lemma 11 is bounded between*

$$0 \le \nu_{\min} \le \nu_0 \le \nu_{\max}, \tag{146}$$

then the following worst case tail bounds hold:

$$\mathsf{Pr}\left(\frac{1}{P}\left\|\mathbf{u}+\mathbf{w}\right\|^2 \ge \tau_1\right) \le e^{-\frac{P}{4}\left(\sqrt{2\tau_1/\theta^2-1}-\sqrt{1+2\nu_{\max}}\right)^2}$$
$$\mathsf{Pr}\left(\frac{1}{P}\left\|\mathbf{u}+\mathbf{w}\right\|^2 \le \tau_2\right) \le e^{-\frac{P}{4}\frac{\left(1+\nu_{\min}-\tau_2/\theta^2\right)^2}{1+2\nu_{\min}}}$$

for any τ_1 and τ_2 that satisfy

$$\tau_1 \ge \theta^2(1+\nu_{\max}), \quad \tau_2 \le \theta^2(1+\nu_{\min}). \tag{147}$$

Proof. The first tail bound can be easily obtained since $\tau_1 \ge \theta^2(1+\nu_{\max})$, the exponent is monotonically decreasing with respect to ν_0, and therefore substituting it with $\nu_{\max}$ leads to an upper bound.

The second tail bound depends on the monotonicity with respect to ν_0. The tail bound is monotonic with respect to the exponent, so in the following we examine the monotonicity of the exponent with respect to ν_0. The exponent can be re-written as a form of the $x + 1/x$ function:

$$\frac{\left(1+\nu_{\min}-\tau_2/\theta^2\right)^2}{1+2\nu_{\min}} = \left(\nu_0+\frac{1}{2}\right) + \frac{\left(\frac{1}{2}-\frac{\tau_2}{\theta^2}\right)^2}{\left(\nu_0+\frac{1}{2}\right)} + 2\left(\frac{1}{2}-\frac{\tau_2}{\theta^2}\right), \tag{148}$$

which has a minimum at

$$\nu_0^\star = \left|\frac{1}{2}-\frac{\tau_2}{\theta^2}\right| - \frac{1}{2}, \tag{149}$$

and monotonically increasing for any $\nu_0 > \nu_0^\star$. Now it remains to see whether $\nu_0^\star$ is within the interval $[\nu_{\min}, \nu_{\max}]$, which needs to be discussed separately depending on the choice of τ_2:

1. $\theta^2/2 \le \tau_2 \le \theta^2(1+\nu_{\min})$: in this case, we have

$$\nu_0^\star = \frac{\tau_2}{\theta^2} - 1 \le \nu_{\min}. \tag{150}$$

2. $0 < \tau_2 < \theta^2/2$: in this case, we have

$$\nu_0^\star = -\frac{\tau_2}{\theta^2} \le 0 \le \nu_{\min}. \tag{151}$$

Therefore, it has been shown that as long as τ_2 satisfies (147), the exponent is monotonically increasing with respect to $\nu_0 \in [\nu_{\min}, \nu_{\max}]$ and therefore the minimum exponent is achieved by substituting ν_0 with $\nu_{\min}$. □