SHO-FA: Robust compressive sensing with order-optimal complexity, measurements, and bits

Mayank Bakshi, Sidharth Jaggi, Sheng Cai, Minghua Chen

***Abstract*—Suppose $\mathbf{x}$ is any exactly k-sparse vector in $\mathbb{R}^n$. We present a class of "sparse" matrices A, and a corresponding algorithm that we call SHO-FA (for Short and Fast[1]) that, with high probability over A, can reconstruct $\mathbf{x}$ from $A\mathbf{x}$. The SHO-FA algorithm is related to the Invertible Bloom Lookup Tables (IBLTs) recently introduced by Goodrich *et al.*, with two important distinctions – SHO-FA relies on linear measurements, and is robust to noise and approximate sparsity. The SHO-FA algorithm is the first to simultaneously have the following properties: (a) it requires only $\mathcal{O}(k)$ measurements, (b) the bit-precision of each measurement and each arithmetic operation is $\mathcal{O}\left(\log(n)+P\right)$ (here 2^{-P} corresponds to the desired relative error in the reconstruction of $\mathbf{x}$), (c) the computational complexity of decoding is $\mathcal{O}(k)$ arithmetic operations, and (d) if the reconstruction goal is simply to recover a single component of $\mathbf{x}$ instead of all of $\mathbf{x}$, with high probability over A this can be done in constant time. All constants above are independent of all problem parameters other than the desired probability of success. For a wide range of parameters these properties are information-theoretically order-optimal. In addition, our SHO-FA algorithm is robust to random noise, and (random) approximate sparsity for a large range of k. In particular, suppose the measured vector equals $A(\mathbf{x}+\mathbf{z})+\mathbf{e}$, where $\mathbf{z}$ and $\mathbf{e}$ correspond respectively to the *source tail* and *measurement noise*. Under reasonable statistical assumptions on $\mathbf{z}$ and $\mathbf{e}$ our decoding algorithm reconstructs $\mathbf{x}$ with an estimation error of $\mathcal{O}(||\mathbf{z}||_1+(\log k)^2||\mathbf{e}||_1)$. The SHO-FA algorithm works with high probability over A, $\mathbf{z}$, and $\mathbf{e}$, and still requires only $\mathcal{O}(k)$ steps and $\mathcal{O}(k)$ measurements over $\mathcal{O}(\log(n))$-bit numbers. This is in contrast to most existing algorithms which focus on the "worst-case" $\mathbf{z}$ model, where it is known $\Omega(k\log(n/k))$ measurements over $\mathcal{O}(\log(n))$-bit numbers are necessary.[2]**

I. Introduction

In recent years, spurred by the seminal work on *compressive sensing* of [2], [3], much attention has focused on the problem of reconstructing a length-n "compressible" vector $\mathbf{x}$ over $\mathbb{R}$ with fewer than n linear measurements. In particular, it is known (*e.g.* [4], [5]) that with $m=\mathcal{O}(k\log(n/k))$ linear measurements one can computationally efficiently obtain a vector $\hat{\mathbf{x}}$ such that the *reconstruction error* $||\mathbf{x}-\hat{\mathbf{x}}||_1$ is $\mathcal{O}(||\mathbf{x}-\mathbf{x}_k{}^*||_1)$, where $\mathbf{x}_k{}^*$ is the best possible k-sparse approximation to $\mathbf{x}$. A number of algorithms give such performance, such as ℓ_1-optimization algorithms (*e.g.* [2], [3]), and iterative decoding algorithms (*e.g.* [6], [7]). Similar results, (with an additional additive term in the reconstruction error) hold even if the measurements also have noise added to them (*e.g.* [4], [5]). The fastest of these algorithms use ideas from the theory of expander graphs, and have running time $\mathcal{O}(n\log(n/k))$ [8]–[10]. These results are very strong – they hold for *all* $\mathbf{x}$ vectors, including those with "worst-case tails", *i.e.* even vectors where the components of $\mathbf{x}$ smaller than the k largest coefficients are chosen in a worst-case manner. In fact [11] prove that to obtain a reconstruction error that scales linearly with the ℓ_1-norm of the $\mathbf{z}$ (the tail of $\mathbf{x}$) requires $\Omega(k\log(n/k))$ linear measurements.

Number of measurements: In some applications such a lower bound for "worst-case $\mathbf{z}$" may be too pessimistic. For instance, it is known that if $\mathbf{x}$ is exactly k-sparse then based on Reed-Solomon codes [12] one can efficiently reconstruct $\mathbf{x}$ with $\mathcal{O}(k)$ noiseless measurements (*e.g.* [13]) via algorithms with decoding time-complexity $\mathcal{O}(n\log(n))$, or via codes such as in [14], [15] with $\mathcal{O}(k)$ noiseless measurements with decoding time-complexity $\mathcal{O}(n)$. In the regime where $k=\theta(n)$, [16] show that $\mathcal{O}(k)=\mathcal{O}(n)$ measurements suffice to reconstruct $\mathbf{x}$.

Noise/Approximate sparsity: If the length-n source vector is the sum of *any* exactly k-sparse vector $\mathbf{x}$ and a "random" source noise vector $\mathbf{z}$ (and possibly $\mathbf{y}=A(\mathbf{x}+\mathbf{z})$ also has a "random" noise vector $\mathbf{e}$ added to it), then as long as the noise variances are not "too much larger" than the signal power, the work of [17] demonstrates that $\mathcal{O}(k)$ measurements suffice (though the algorithms require time exponential in n). Indeed, even the work of [11], whose primary focus was to prove that $\Omega(k\log(n/k))$ linear measurements are necessary to reconstruct with the worst case $\mathbf{z}$, also notes as an aside that if $\mathbf{x}$ corresponds to an exactly sparse vector plus random noise, then in fact $\mathcal{O}(k)$ measurements suffice. The work in [18], [19] examines this phenomenon information-theoretically, and [20] show how

[1] Also, SHO-FA sho good! In fact, it's all $\mathcal{O}(k)$!

[2] A poster based on this work was presented at ISIT 2012 in Recent Results [1].

to computationally efficiently achieve this performance by exactly reconstructing $\mathbf{x}$ with $\mathcal{O}(\bar{d}(X)n)+o(n)$ samples in time $\mathcal{O}(n)$. Corresponding lower bounds showing $\Omega(k\log(n/k))$ samples are required in the higher noise regime are provided in [21], [22].

Number of measurement bits: However, most of the works above focus on minimizing the number of linear measurements in $A\mathbf{x}$, rather than the more information-theoretic view of trying to minimize the number of bits in $A\mathbf{x}$ over all measurements. Some recent work attempts to fill this gap [23], [24] uses "multi-layered non-linear measurements", and "one-bit compressive sensing" [25], [26] (the corresponding decoding complexity is somewhat high since it involves solving an LP).

Decoding time-complexity: The emphasis of the discussion thus far has been on the number of linear measurements/bits required to reconstruct $\mathbf{x}$. The decoding algorithms in most of the works above have decoding time-complexities[3] that scale at least linearly with n. In regimes where k is significantly smaller than n, it is natural to wonder whether one can do better. Indeed, algorithms based on iterative techniques answer this in the affirmative. These include Chaining Pursuit [27], group-testing based algorithms [28], and Sudocodes [29] – each of these have decoding time-complexity that can be sub-linear in n (but at least $\mathcal{O}(k\log(k)\log(n))$), but each requires at least $\mathcal{O}(k\log(n))$ linear measurements.

Database query: Finally, we consider a *database query* property that is not often of concern in the compressive sensing literature. Suppose, in addition to the properties above, one *also* wishes to reconstruct (with constant probability) just "a few" ($\mathcal{O}(1)$) *specific* components of $\mathbf{x}$ in $\mathcal{O}(1)$ time. If the matrix A is "dense" (most of its entries are non-zero) then one can directly see that this is impossible, but SHO-FA has this property. [4]

A. Our Contributions

Conceptually, the "iterative decoding" technique we use is not new (for *e.g.* [14], [30]–[32]), but we do not know of prior work has the same performance as our work – namely – information-theoretically order-optimal number of measurements, bits in those measurements, and time-complexity, for the problem of robust reconstructing a (approximately) sparse signal via (noisy) linear measurements (along with the database query property).[5] The key to this performance is our novel design of "sparse random" linear measurements, as described in Section II. To summarize, the desirable properties of SHO-FA are that with high probability:

Number of measurements: For every k-sparse $\mathbf{x}$, with high probability over A, $\mathcal{O}(k)$ linear measurements suffice to reconstruct $\mathbf{x}$. This is information-theoretically order-optimal.

Number of measurement bits: The total number of bits in $A\mathbf{x}$ required to reconstruct $\mathbf{x}$ to a relative error of 2^{-P} is $\mathcal{O}(k(\log(n)+P))$. This is information-theoretically order-optimal for any $k=\mathcal{O}(n^{1-\Delta})$ (for any $\Delta>0$).

Decoding time-complexity: The total number of arithmetic operations required is $\mathcal{O}(k)$. This is information-theoretically order-optimal.

Database queries: With constant probability $1-\epsilon$ any single database query can be answered in $\mathcal{O}(1)$ time. [6]

Noise: Suppose $\mathbf{z}$ and $\mathbf{e}$ have i.i.d. components drawn respectively from $\mathcal{N}(0,\sigma_z^2)$ and $\mathcal{N}(0,\sigma_e^2)$. For $k=\mathcal{O}(n^{1-\Delta})$ for any $\Delta>0$, a modified version of SHO-FA (mod-SHO-FA) that with high probability reconstructs $\mathbf{x}$ with an estimation error of $\mathcal{O}(||\mathbf{z}||_1+(\log k)^2||\mathbf{e}||_1)$.

Practicality: As validated by simulations (see [34, Appendix I]), most of the constant factors involved above are not large, and are in fact significantly smaller than the explicit constants that can be calculated via our analysis.

Different bases: As is common in the compressive sensing literature, our techniques generalize directly to the setting wherein $\mathbf{x}$ is sparse in an alternative basis (say, for example, in a wavelet basis).

Universality: While we present a specific ensemble of matrices over which SHO-FA operates, we argue that in fact similar algorithms work over fairly general ensembles of "sparse random matrices", and further that such matrices can occur in applications, for instance in wireless MAC problems [35].

B. Special acknowledgements

In particular, the bounds on the minimum number of measurements required for "worst-case" recovery and the corresponding discussion on recovery of signals with

[3] For ease of presentation, in accordance with common practice in the literature, in this discussion we assume that the time-complexity of performing a single arithmetic operation is constant. Explicitly taking the complexity of performing finite-precision arithmetic into account adds a multiplicative factor (corresponding to the precision with which arithmetic operations are performed) in the time-complexity of most of the works, including ours.

[4] Several compressive sensing algorithms (for instance [16]) are based on "sparse" matrices A, and it can be shown that in fact these algorithms do indeed have this property "for free" (as indeed does our algorithm), even though the authors do not analyze this. As can be inferred from the name, this database query property is more often considered in the database community, for instance in the work on IBLTs [30].

[5] While writing this paper, we became aware of a parallel work by Pawar and Ramchandran [33] that seems to achieve similar performance. However, at the time of submission, a preprint of this work was not available for us to compare the two works.

[6] The constant ϵ can be made arbitrarily close to zero, at the cost of a multiplicative factor $\mathcal{O}(1/\epsilon)$ in the number of measurements required. In fact, if we allow the number of measurements to scale as $\mathcal{O}(k\log(k))$, we can support any number of database queries, each in constant time, with probability of every one being answered correctly at with probability at least $1-\epsilon$.

"random tails" in [11] led us to consider this problem in the first place. Equally, the class of compressive sensing codes in [16], which in turn build upon the constructions of expander codes in [31], have been influential in leading us to this work. While the model in [32] differs from the one in this work, the techniques therein are of significant interest in our work. The analysis in [32] of the number of disjoint components in certain classes of random graphs, and also the analysis of how noise propagates in iterative decoding is potentially useful sharpening our results. The work that is conceptually the closest to SHO-FA is that of the Invertible Bloom Lookup Tables (IBLTs) introduced by Goodrich-Mitzenmacher [30] (though our results were derived independently, and hence much of our analysis follows a different line of reasoning). The data structures and iterative decoding procedure (called "peeling" in [30]) used are structurally very similar to the ones used in this work. However the "measurements" in IBLTs are fundamentally non-linear in nature – specifically, each measurement includes within it a "counter" variable – it is not obvious how to implement this in a linear manner. Therefore, though the underlying graphical structure of our algorithms is similar, the details of our implementation require new non-trivial ideas. Also, IBLTs as described are not robust to either signal tails or measurement noise.

II. Exactly k-sparse $\mathbf{x}$ and noiseless measurements

We first consider the simpler case when $\mathbf{y} = A\mathbf{x}$ (both $\mathbf{z}$ and $\mathbf{e}$ are zero). The intuition presented here carries over to the scenario wherein both $\mathbf{z}$ and $\mathbf{e}$ are non-zero, considered separately in Section III. For k-sparse input vectors $\mathbf{x} \in \mathbb{R}^n$ let the set $\mathcal{S}(\mathbf{x})$ denote its *support*. *i.e.*, its set of nonzero values. Recall that in our notation, for some m, a *measurement matrix* $A \in \mathbb{R}^{m \times n}$ is chosen probabilistically. This matrix operates on $\mathbf{x}$ to yield the *measurement vector* $\mathbf{y} \in \mathbb{R}^m$ as $\mathbf{y} = A\mathbf{x}$. The decoder takes the vector $\mathbf{y}$ as input and outputs the reconstruction $\hat{\mathbf{x}} \in \mathbb{R}^n$ – it is desired that $\hat{\mathbf{x}}$ equal $\mathbf{x}$ (with relative error at most 2^{-P}) with high probability (over A). We now describe a probabilistic construction of the measurement matrix A and a reconstruction algorithm SHO-FA that achieves the following guarantees (the proof of which is the focus of the remainder of this section).

Theorem 1. *Let $k \leq n$. There exists a reconstruction algorithm SHO-FA for $A \in \mathbb{R}^{m \times n}$ such that (i) for every $\mathbf{x} \in \mathbb{R}^n$, with probability $1 - \mathcal{O}(1/k)$ over the choice of A, SHO-FA produces a reconstruction $\hat{\mathbf{x}}$ such that $||\mathbf{x} - \hat{\mathbf{x}}||_1 / ||\mathbf{x}||_1 \leq 2^{-P}$, (ii) $m = ck$ for some $c > 0$, (iii) Expected number of steps required by SHO-FA is $\mathcal{O}(k)$, and (iv) Expected number of bitwise arithmetic operations required by SHO-FA is $\mathcal{O}(k(\log n + P))$*

High-level intuition: If $m = \Theta(n)$, the task of reconstructing $\mathbf{x}$ from $\mathbf{y} = A\mathbf{x}$ appears similar to that of *syndrome decoding* of a channel code of rate n/m [36]. It is well-known [37] that channel codes based on *bipartite expander graphs*, *i.e.*, bipartite graphs with good expansion guarantees for all sets of size less than or equal to k, allow for decoding in a number of steps that is linear in the size of $\mathbf{x}$. [7]

It is tempting to think that perhaps an optimized application of expander graphs could result in a design that require only $\mathcal{O}(k)$ number of measurements. However, as noted in [34, Lemma 2], in the compressive sensing setting, where, typically $k = o(n)$, it is not possible to satisfy the desired expansion properties. [8]Instead, one of our key ideas is that we do not really need "true" expansion. Instead, we rely on a notion of approximate expansion that guarantees expansion for most k-sized sets (and their subsets) of nodes on the left of our bipartite graph. We do so by showing that any set of size at most k, with high probability over suitably chosen measurement matrices, expands to the desired amount. Probabilistic constructions turn out to exist for our desired property.[9] Such a construction is shown in Lemma 1.

Our second key idea is that in order to be able to recover all the k non-zero components of $\mathbf{x}$ with at most $\mathcal{O}(k)$ steps in the decoding algorithm, it is necessary (and sufficient) that on average, the decoder reconstructs one previously undecoded non-zero component of $\mathbf{x}$, say x_j, in $\mathcal{O}(1)$ steps in the decoding algorithm. For $k = o(n)$ the algorithm does not even have enough time to write out all of $\mathbf{x}$, but only its non-zero values. To achieve such efficient identification of x_j, we go beyond the $0/1$ matrices used in almost all prior work on compressive sensing based on expander graphs. Instead, we use distinct values in each row for the non-zero values in A, so that if only one non-zero x_j is involved in the linear measurement involving a particular y_i (a situation that we demonstrate happens in a constant fraction of y_i), one can identify which x_j it must be in $\mathcal{O}(1)$ time. Our decoding then proceeds iteratively, by identifying such x_j and canceling their effects on y_i, and terminates after $\mathcal{O}(k)$ steps after all non-zero x_j

[7]Motivated by this [16] explore a measurement design that is derived from expander graphs and show that $\mathcal{O}(k \log(n/k))$ measurements suffice, and $\mathcal{O}(k)$ iterations with overall decoding complexity of $\mathcal{O}(n \log(n/k))$.

[8]In particular, if one tries to mimic the approach of [16], one would need bipartite expanders such that *all* sets of size k on one side of the graph "expand".

[9]In fact similar properties have been considered before in the literature – for instance [38] constructed "probabilistic expanders". Our contribution is the way we use this property for our needs.

and their locations have been identified (since we require our algorithm to work with high probability for all $\mathbf{x}$, we also add "verification" measurements – this only increases the total number of measurements by a constant factor). Our calculations are precise to $\mathcal{O}(\log(n)+P)$ bits – the first term in this comes from requirements necessary for computationally efficient identification of non-zero x_j, and the last term from the requirement that we require that the reconstructed vector be correct up to P-precision. Hence the total number of bits over all measurements is $\mathcal{O}(k((\log(n)+P))$. Note that this is information-theoretically order-optimal, since even specifying k locations in a length-n vector requires $\Omega(k(\log(n/k))$ bits, and specifying the value of the non-zero locations so that the relative reconstruction error is $\mathcal{O}(2^{-P})$ requires $\Omega(kP)$ bits.

We now present our SHO-FA algorithm in two stages. We first use our first key idea (of "approximate") expansion in Section II-A to describe some properties of bipartite expander graphs with certain parameters. We then show in Section II-B how these properties, via our second key idea (of efficient identification) can be used by SHO-FA to obtain desirable performance.

A. Description of graph properties

We first construct a bipartite graph $\mathcal{G}$ (see Example 1 in the Appendix) with some desirable properties outlined below that follow from Lemmas 1 and 2. In Section II-B we then use these graph properties in the SHO-FA algorithm. To simplify notation in what follows (unless otherwise specified) we omit rounding numbers resulting from taking ratios or logarithms, with the understanding that the corresponding inaccuracy introduced is negligible compared to the result of the computation. Also, for ease of exposition, we fix various internal parameters to "reasonable" values rather than optimizing them to obtain "slightly" better performance at the cost of obfuscating the explanations – whenever this happens we shall point it out parenthetically.

Properties of $\mathcal{G}$:

1. *Construction of a left-regular bipartite graph:* The graph $\mathcal{G}$ is chosen uniformly at random from the set of bipartite graphs with n nodes on the left and m' nodes on the right, such that each node on the left has degree $d \geq 7$. In particular, m' is chosen to equal ck for some design parameter c to be specified later as part of code design.

2. *Edge weights for "identifiability":* For each node on the right, the weights of the edges attached to it are required to be distinct. In particular, each edge weight is chosen as a complex number of unit magnitude, and phase between 0 and $\pi/2$. Since there are a total of dn edges in $\mathcal{G}$, choosing distinct phases for each edge attached to a node on the right requires at most $\log(dn)$ bits of precision (though on average there are about dn/m' edges attached to a node on the right, and hence on average one needs about $\log(dn/m')$ bits of precision).

3. $\mathcal{S}(\mathbf{x})$*-expansion:* In Lemma 1, we note that with a high probability over $\mathcal{G}$ defined in Property 1 above, every set of k nodes on the left and all its subsets "expand" by a factor at least $2d/3$. Daskalakis *et al.* [38, Lemma 4] prove this property in the context of error correcting codes for graphs that have slightly different parameters than ours. As such, we omit the proof here and refer the reader to [34, Lemma 1] for a detailed proof.

4. *"Many"* $\mathcal{S}(\mathbf{x})$*-leaf nodes:* For any set $\mathcal{S}(\mathbf{x})$ of at most k nodes on the left of $\mathcal{G}$, we call any node on the right of $\mathcal{G}$ an $\mathcal{S}(\mathbf{x})$*-leaf node* if it has exactly one neighbor in $\mathcal{S}(\mathbf{x})$, and we call it a $\mathcal{S}(\mathbf{x})$*-non-leaf node* if it has two or more neighbours in $\mathcal{S}(\mathbf{x})$. (If the node on the right has no neighbours in $\mathcal{S}(\mathbf{x})$, we call it a $\mathcal{S}(\mathbf{x})$*-zero node*.) Assuming $\mathcal{S}(\mathbf{x})$ satisfies the expansion condition in Property 3 above, it can be shown that at least a fraction $1/2$ of the nodes that are neighbours of any $\mathcal{S}'(\mathbf{x}) \subseteq \mathcal{S}(\mathbf{x})$ are $\mathcal{S}'(\mathbf{x})$-leaf nodes.This statement is the subject of Lemma 2 and follows from a counting argument similar to that used in expander codes [31]. For completeness, the reader is referred to [34, Lemma 3] for a proof.

Lemma 1. ($\mathcal{S}(\mathbf{x})$-expansion): *Let $k < n \in \mathbb{N}$ be arbitrary, and let $c \in \mathbb{N}$ be fixed. Let $\mathcal{G}$ be chosen uniformly at random from the set of all bipartite graphs with n nodes (each of degree d) on the left and $m' = ck$ nodes on the right. Then, for any $\mathcal{S}(\mathbf{x})$ of size at most k and any $\mathcal{S}'(\mathbf{x}) \subseteq \mathcal{S}(\mathbf{x})$, with probability $1 - o(1/k)$ (over the random choice $\mathcal{G}$) there are at least $2d/3$ times as many nodes neighbouring those in $\mathcal{S}'(\mathbf{x})$, as there are in $\mathcal{S}'(\mathbf{x})$.*

Lemma 2. *Let $\mathcal{S}(\mathbf{x})$ be a set of k nodes on the left of $\mathcal{G}$ such that the number of nodes neighbouring those in any $\mathcal{S}'(\mathbf{x}) \subseteq \mathcal{S}(\mathbf{x})$ is at least $2d/3$ times the size of $\mathcal{S}'(\mathbf{x})$. Then at least a fraction $1/2$ of the nodes that are neighbours of any $\mathcal{S}'(\mathbf{x}) \subseteq \mathcal{S}(\mathbf{x})$ are $\mathcal{S}'(\mathbf{x})$-leaf nodes.*

Note here that, in contrast to the "usual" definition of "vertex expansion" [37] (wherein the expansion property is desired *"for all"* subsets of left nodes up to a certain size) Lemma 1 above only gives a probabilistic expansion guarantee for any subset of $\mathcal{S}(\mathbf{x})$ of size k. In fact, for the parameters of interest to us, [34, Lemma 2] shows that "for all"-type expanders cannot exist.

B. Description of SHO-FA

Given a graph $\mathcal{G}$ satisfying properties 1- 4, we now describe our encoding and decoding procedure. We begin with a description of the measurement matrix A.

Matrix structure and entries: The encoder's *measurement matrix* A is chosen based on the structure of $\mathcal{G}$ (recall that $\mathcal{G}$ has n nodes on the left and m' nodes on the right). To begin with, the matrix A has $m = 2m'$ rows, and its non-zero values are unit-norm complex numbers. This choice of using complex numbers rather than real numbers in A is for notational convenience only. One can equally well choose a matrix A' with $m = 4m'$ rows, and replace each row of A with two consecutive rows in A' comprising respectively of the real and imaginary parts of rows of A. Since the components of $\mathbf{x}$ are real numbers, hence there is a bijection between $A\mathbf{x}$ and $A'\mathbf{x}$ – indeed, consecutive pairs of elements in $A'\mathbf{x}$ are respectively the real and imaginary parts of the complex components of $A\mathbf{x}$. Also, as we shall see, the choice of unit-norm complex numbers ensures that "noise" due to finite precision arithmetic does not get "amplified". In particular, corresponding to node i on the right-hand side of $\mathcal{G}$, the matrix A has two rows. The j^{th} entries of the $(2i-1)^{th}$ and $2i^{th}$ rows of A are respectively denoted $a^{(I)}_{i,j}$ and $a^{(V)}_{i,j}$ respectively. (The superscripts (I) and (V) respectively stand for *Identification* and *Verification*, for reasons that shall become clearer when we discuss the process to reconstruct $\mathbf{x}$.)

Identification entries: If $\mathcal{G}$ has no edge connecting node j on the left with i on the right, then the *identification entry* $a^{(I)}_{i,j}$ is set to equal 0. Else, if there is indeed such an edge, $a^{(I)}_{i,j}$ is set to equal $e^{\iota j\pi/(2n)}$. (Here ι denotes the positive square root of -1.) This entry $a^{(I)}_{i,j}$ can also be thought of as the weight of the edge in $\mathcal{G}$ connecting j on the left with i on the right. In particular, the *phase* $j\pi/(2n)$ of $a^{(I)}_{i,j} = e^{\iota j\pi/(2n)}$ will be critical for our algorithm. As in Property 2 in Section II-A, our choice above guarantees distinct weights for all edges connected to a node i on the right

Verification entries: Whenever the identification entry $a^{(I)}_{i,j}$ equals 0, we choose to set the corresponding *verification entry* $a^{(V)}_{i,j}$ also to be zero. On the other hand, whenever $a^{(I)}_{i,j} \neq 0$, then we set $a^{(V)}_{i,j}$ to equal $e^{\iota\theta^{(V)}_{i,j}}$ for $\theta^{(V)}_{i,j}$ chosen uniformly at random from $[0, \pi/2]$ (with $\mathcal{O}(\log(k))$ bits of precision).[10]

Reconstruction: Since the measurement matrix A has interspersed identification and verification rows, this induces corresponding interspersed *identification observations* $y^{(I)}_i$ and verification *verifications observations* $y^{(V)}_i$ in the observation vector $\mathbf{y} = A\mathbf{x}$. Let $\mathbf{y}^{(I)} = \{y^{(I)}_i\}$ denote the length-m *identification vector* over $\mathbb{C}$, and $\mathbf{y}^{(V)} = \{y^{(V)}_i\}$ denote the length-m *verification vector* over $\mathbb{C}$.

Input: $(A,\mathbf{y})$
Output: $\hat{\mathbf{x}}$

$\hat{\mathbf{x}}(1) \leftarrow 0^n$, $\tilde{\mathbf{y}}^{(I)}(1) \leftarrow \mathbf{y}^{(I)}$, $\tilde{\mathbf{y}}^{(V)}(1) \leftarrow \mathbf{y}^{(V)}$;
$\mathcal{D}(1) \leftarrow \{i : y^{(I)}_i \neq 0\}$, $t \leftarrow 1$;
while $\mathcal{D}(t) \neq \phi$ **do**
  $\hat{\mathbf{x}}(t+1) \leftarrow \hat{\mathbf{x}}(t)$;
  Pick random $i(t) \in \mathcal{D}(t)$;
  $\theta^{(I)}(t) \leftarrow \angle\left(\tilde{y}^{(I)}_{i(t)}(t)\right)$, $\theta^{(V)}(t) \leftarrow \angle\left(\tilde{y}^{(V)}_{i(t)}(t)\right)$;
  if $\frac{2n}{\pi}\theta^{(I)}(t) \in \mathbb{N}$ **then**
    $j(t) \leftarrow \frac{2n}{\pi}\theta^{(I)}(t)$; // Identify
    if $\theta^{(V)}(t) = \theta^{(V)}_{i(t),j(t)}$; // Verify leaf
    then
      $\hat{x}_{j(t+1)} \leftarrow |\tilde{y}^{(I)}_{i(t)}(t)|$; // Decode
      $\Delta\mathbf{x} \leftarrow \hat{\mathbf{x}}(t+1) - \hat{\mathbf{x}}(t)$; // Update
      $\mathcal{D}(t+1) \leftarrow \mathcal{D}(t) \setminus \{i : (A\Delta\mathbf{x})_i \neq 0\}$;
      $\tilde{\mathbf{y}}^{(I)}(t+1) \leftarrow \tilde{\mathbf{y}}^{(I)}(t+1) - A\Delta\mathbf{x}$;
      $\tilde{\mathbf{y}}^{(V)}(t+1) \leftarrow \tilde{\mathbf{y}}^{(V)}(t) - A\Delta\mathbf{x}$;
  $t \leftarrow t+1$;
  // Terminate when $\mathcal{D}(t) = \phi$
end

Algorithm 1: SHO-FA Reconstruction Algorithm

Given the measurement matrix A and the observed $(\mathbf{y}^{(I)}, \mathbf{y}^{(V)})$ identification and verification vectors, the decoder's task is to find *any* k-sparse vector $\hat{\mathbf{x}}$ such that $A\hat{\mathbf{x}}$ results in the corresponding identification and observation vectors. We shall argue below that if we succeed, then with high probability over A (specifically, over the verification entries of A), this $\hat{\mathbf{x}}$ must equal $\mathbf{x}$.

To find such a $\hat{\mathbf{x}}$ we consider an iterative decoding scheme presented in Algorithm 1. In Step 1, we start by setting the initial *signal estimate vector* $\hat{\mathbf{x}}(1)$ to the all-zero vector, and the initial *residual measurement identification/verification vectors* $\tilde{\mathbf{y}}^{(I)}(1)$ and $\tilde{\mathbf{y}}^{(V)}(1)$ to $\mathbf{y}^{(I)}$ and $\mathbf{y}^{(V)}$ respectively. Next, we identify the set of non-zero indices of $\mathbf{y}^{(V)}$, and initializes the $\mathcal{D}(1)$, which we call the *neighborly set* as the set of non-zero *indices* of the *verification vector* $\mathbf{y}^{(V)}$ In the first iteration we then pick a uniformly random index i from the neighborly set. Next, the decoder attempts to recover the signal value at some index $j \in \mathcal{S}(\mathbf{x})$ by looking at $y^{(I)}_i$ and "estimating" which j on the left of $\mathcal{G}$ could have "caused the identification observation $y^{(I)}_i$". If index i is

[10]This choice of precision for the verification entries contributes one term to our expression for the precision of arithmetic required. As we argue later in Section II-D, this choice of precision guarantees that if a single identification step returns a value for x_j, this is indeed correct with probability $1-o(1/k)$. Taking a union bound over $\mathcal{O}(k)$ indices corresponding to non-zero x_j gives us an overall $1-o(1)$ probability of success.

not a $\mathcal{S}(\mathbf{x})$-leaf node, the decoder does not succeed in reconstructing x_j, it declares the iteration as a failure, and starts the second iteration by again choosing a new uniformly random index i from the neighborly set. On the other hand, if index i is a $\mathcal{S}(\mathbf{x})$-leaf node, the corresponding signal coordinate j will indeed be identified (Step 8) (and "verified" using the verification entry $a_{i,j}^{(V)}$ and the verification observation $y_i^{(V)}$ in Step 9)[11]; then the algorithm will decode the corresponding signal value in Step 11, and update the residual measurement vectors $\tilde{\mathbf{y}}^{(I)}$ and $\tilde{\mathbf{y}}^{(V)}$ by subtracting the "contribution" of the coordinate x_j to the measurements it influences (there are exactly d of them since the degree of the nodes on the left side of $\mathcal{G}$ is d) and remove i from the neighborly set (Steps 12- 15), and finally pick a new random index i from the neighbourly set for the next (second) iteration. The decoder performs the above operations repeatedly until $\hat{\mathbf{x}}$ has been completely recovered. We also show that (with high probability over A) in $\mathcal{O}(k)$ steps this process does indeed terminate. Our algorithm proceeds iteratively, and has $\mathcal{O}(k)$ overall (expected) number of iterations, with t being the variable indexing the iteration number.

Expected number of iterations: We first argue that, with a constant probability, each iteration result in recovering a new non-zero coordinate from $\mathbf{x}$. Towards this, for each $t = 1, 2, \ldots$, let $\mathcal{S}(t)$ be the support of $\mathbf{x} - \hat{\mathbf{x}}(t)$. Note that $\mathcal{D}(t) = N(\mathcal{S}(t))$ and $\mathcal{S}(\mathbf{x}) = \mathcal{S}(1) \supseteq \mathcal{S}(2) \supseteq \ldots$. Then, according to Lemma 2 and the way we generate the measurement matrix A, with a high probability, for each t, the probability that there exists a node $i(t)$ in $\tilde{\mathbf{y}}^{(I)}(t)$ so that it is an $\mathcal{S}(t)$-leaf node is lower bounded by $1/2$. Consequently, exactly one non-zero coordinate in $\mathcal{S}(t)$ completely determines $\tilde{\mathbf{y}}_{i(t)}^{(I)}(t)$ and $\tilde{\mathbf{y}}_{i(t)}^{(V)}(t)$. The algorithm identifies this coordinate as j for the t^{th} iteration and at the end of iteration, recovers $\hat{x}_j$. Thus, whenever $i(t)$ is an $\mathcal{S}(t)$-leaf node, the set of recovered coordinates increases by 1. When $i(t)$ is not an $\mathcal{S}(t)$-leaf node, our reconstruction process wastes one iteration and will start another iteration by picking another node from the neighborly set $\mathcal{D}(t)$ uniformly at random. Hence, the operations among different iterations are independent, and each iteration succeeds with probability $1/2$. Since there are at most k non-zero coordinates in $\mathbf{x}$, the number of iterations before the algorithm terminates follows a Pascal distribution with parameters $(k, 1/2)$. The expected number of iterations is then simply $2k$.

Correctness: Next, we show that $\hat{\mathbf{x}} = \mathbf{x}$ with a high probability. To show this, it suffices to show that each non-zero update to the estimate $\hat{\mathbf{x}}(t)$ sets a previously untouched coordinate to the correct value with a high probability.

Note that if $i(t)$ is a leaf node for $\mathcal{S}(t)$, and if all non-zero coordinates of $\hat{\mathbf{x}}(t)$ are equal to the corresponding coordinates in $\mathbf{x}$, then the decoder correctly identifies the parent node $j(t) \in \mathcal{S}(t)$ for the leaf node $i(t)$ as the unique coordinate that passes the phase identification and verification checks.

Thus, the t^{th} iteration ends with an erroneous update only if $\angle(\sum_{p \in N(\{i(t)\})} x_p e^{\iota \theta_{i(t),p}^{(I)}}) = \theta_{i(t),j(t)}^{(I)}$ for some j such that there are more than one non-zero terms in the summation on the left. Recall that $\angle(\sum_{p \in N(\{i(t)\})} x_p e^{\iota \theta_{i(t),p}^{(V)}}) = \theta_{i(t),j(t)}^{(V)}$. Since $V(i(t), j)$ is drawn uniformly at random from $\{1, 2, \ldots, \lceil 4n \rceil\}$, the probability that this equality holds with more than one non-zero term in the summation on the left is at most $1/(4n)$. The above analysis gives an upper bound on the probability of incorrect update for a single iteration to be $1/(4n)$. Finally, as the total number of updates is at most k, by applying a union bound over the updates, the probability of incorrect decoding is upper bounded by $k/4n$. Since $k = o(n)$ by assumption, it follows that the error probability vanishes as n and k grow without bound.

C. Database query

A useful property of our construction of the matrix A is that any desired signal component x_j can be reconstructed with a constant probability given the measurement vector $\mathbf{y} = A\mathbf{x}$ in a constant time. The following Lemma makes this precise. The proof is in Appendix A.

Lemma 3. *Let $\mathbf{x}$ be k-sparse. Let $j \in \{1, 2, \ldots, n\}$ and let $A \in \mathbb{C}^{ck \times n}$ be randomly drawn according to SHO-FA. Then, there exists an algorithm $\mathcal{A}$ such that given inputs $(j, \mathbf{y})$, $\mathcal{A}$ produces an output $\hat{x}_j$ with probability at least $(1 - (d/c)^d)$ such that $\hat{x}_j = x_j$ with probability $(1 - o(1/k))$.*

D. Information-theoretically optimal number of bits

We recall that the reconstruction goal for SHO-FA is to reconstruct $\mathbf{x}$ up to relative error 2^{-P}., that is, $||\mathbf{x} - \hat{\mathbf{x}}||_1 / ||\mathbf{x}||_1 \leq 2^{-P}$.

We first present a sketch of an information-theoretic lower bound of $\Omega\left(k\left(P + \log n\right)\right)$ bits holds for any algorithm that outputs a k-sparse vector that achieves this goal with high probability.

To see this is true, consider the case where the locations of k non-zero entries in $\mathbf{x}$ are chosen uniformly at random among all the n, entries and the value of each non-zero entry is chosen uniformly at random from the set $\{1, \ldots, 2^P\}$. Then recovering even

[11] As Ronald W. Reagan liked to remind us, "*doveryai, no proveryai*".

the support requires at least $\log\left(2^{kP}\binom{n}{k}\right)$ bits, which is $\Omega(k\log(n/k))$. Also, at least a constant fraction of the k non-zero entries of $\mathbf{x}$ must be be correctly estimated to guarantee the desired relative error. Hence $\Omega\left(k\left(P+\log n\right)\right)$ is a lower bound on the measurement bit-complexity.

The following arguments show that the total number of bits used in our algorithm is information-theoretically order-optimal for any $k=\mathcal{O}(n^{1-\Delta})$ (for any $\Delta>0$). First, to represent each non-zero entry of $\mathbf{x}$, we need to use arithmetic of $\Omega(P+\log(k))$ bit precision. Here the P term is so as to attain the required relative error of reconstruction, and the $\log(k)$ term is to take into account the error induced by finite-precision arithmetic (say, for instance, by floating point numbers) in $\mathcal{O}(k)$ iterations (each involving a constant number of finite-precision additions and unit-magnitidue multiplications). Second, for each identification step, we need to use $\Omega(\log(n)+\log(k))$ bit-precision arithmetic. Here the $\log(n)$ term is so that the identification measurements can uniquely specify the locations of non-zero entries of $\mathbf{x}$. The $\log(k)$ term is again to take into account the error induced in $\mathcal{O}(k)$ iterations. Third, for each verification step, the number of bits we use are $3\log(k)$. Here, by the Schwartz-Zippel Lemma [39], [40], $2\log(k)$ bit-precision arithmetic guarantees that each verification step is valid with probability at least $1-1/k^2$ – a union bound over all $\mathcal{O}(k)$ verification steps guarantees that all verification steps are correct with probability at least $1-\mathcal{O}(1/k)$. Therefore, the total number of bits needed by SHO-FA $\mathcal{O}(k(\log(n)+P))$. As claimed, this matches, up to a constant factor, the lower bound sketched above.

III. Approximate reconstruction in the presence of noise

The design presented above relies on exact determination of all the phases and magnitudes of the measurement vector $A\mathbf{x}$. However, we often desire that the measurements and reconstruction be robust to corruption both before and and during measurements. We now show that SHO-FA may be made robust to such "noise". We consider the following setup. Let $\mathbf{x}\in\mathbb{R}^n$ be a k-sparse signal with support $\mathcal{S}(\mathbf{x})=\{j:x_j\neq 0\}$. Let $\mathbf{z}\in\mathbb{R}^n$ have support $\{1,2,\ldots,n\}\setminus\mathcal{S}(\mathbf{x})$ with each z_j distributed according to a Gaussian distribution with mean 0 and variance σ_z^2. Denote the measurement matrix by $A\in\mathbb{C}^{m\times n}$ and the measurement vector by $\mathbf{y}\in\mathbb{C}^m$. Let $\mathbf{e}\in\mathbb{C}^m$ be the measurement noise with e_i distributed as a Complex Gaussian with mean 0 and variance σ_e^2 along each axis. $\mathbf{y}$ is related to the signal as $\mathbf{y}=A(\mathbf{x}+\mathbf{z})+\mathbf{e}$. We propose a design procedure for A satisfying the following properties.

Theorem 2. *Let $k=\mathcal{O}(n^{1-\Delta})$ for some $\Delta>0$. There exists a reconstruction algorithm SHO-FA for $A\in\mathbb{C}^{m\times n}$ such that (i) $m=ck$, (ii) SHO-FA consists of at most $4k$ iterations, each involving $\mathcal{O}(1)$ arithmetic operations with a precision of $\mathcal{O}(\log n)$ bits, and (iii) For some $C=C(\sigma_z,\sigma_e)>0$, with probability $1-o(1/k)$ over the design of A and randomness in $\mathbf{e}$ and $\mathbf{z}$,*

$$||\hat{\mathbf{x}}-\mathbf{x}||_1\leq C\left(||\mathbf{z}||_1+(\log k)^2||\mathbf{e}||_1\right).$$

Recall that in the exactly k-sparse case, each arithmetic operation must have low reconstruction error, as an error in an earlier iteration can propagate and cause potentially catastrophic errors. With noisy $\mathbf{z}$ and $\mathbf{e}$ this problem is accentuated. For instance, SHO-FA may be unable to identify whether y_i is a leaf node, or be unable to identify the j-coordinate of $\mathbf{x}$ it corresponds to. Even if it correctly estimates these quantities, the error due to components of $\mathbf{z}$ and $\mathbf{e}$ may propagate through the iterative decoding procedure and "amplify". To overcome these hurdles, our design takes the noise statistics into account to ensure that each iteration is resilient to noise with a high probability. This is achieved through several new ideas presented below. Key to this analysis is bounding the effect of propagation of estimation error as the decoder steps through the iterations.[12]

Key ideas:

Truncated reconstruction: In the presence of noise it is unlikely that x_j whose magnitudes are comparable to those of the noise values can be successfully recovered. Thus the decoder does not try to reconstruct these values as long as the overall penalty in ℓ_1-norm is not high. The following argument shows that this is indeed the case. Let $\mathcal{S}_\delta(\mathbf{x})\triangleq\{j:|x_j|<\delta/k\}$ and let $\mathbf{x}_{\mathcal{S}_\delta}\in\mathbb{R}^n$ be the vector such that, for each j, $(x_{\mathcal{S}_\delta})_j$ equals x_j if j lies in $\mathcal{S}_\delta$ and is 0 otherwise. Similarly, define $\mathbf{x}_{\mathcal{S}_\delta^c}$ which has non-zero entries only within the set $\mathcal{S}(\mathbf{x})\setminus\mathcal{S}_\delta(\mathbf{x})$. Then the total ℓ_1 norm of $\mathbf{x}_{\mathcal{S}_\delta}$ is "small", since $||\mathbf{x}_{\mathcal{S}_\delta}||_1=\sum_{j\in\mathcal{S}_\delta(\mathbf{x})}|x_j|\leq|\mathcal{S}_\delta(\mathbf{x})|\frac{\delta}{k}\leq\delta$.. Applying the triangle inequality, it then follows that $||\hat{\mathbf{x}}-\mathbf{x}||_1\leq||\hat{\mathbf{x}}-\mathbf{x}_{\mathcal{S}_\delta^c}||_1+||\mathbf{x}_{\mathcal{S}_\delta}||_1\leq||\hat{\mathbf{x}}-\mathbf{x}_{\mathcal{S}_\delta^c}||_1+\delta$. Our reconstruction $\hat{\mathbf{x}}$ can now be shown to satisfy the criterion that $||\hat{\mathbf{x}}-\mathbf{x}_{\mathcal{S}_\delta^c}||_1$ is at most $C_1(||\mathbf{z}||_1+(\log k)^2||\mathbf{e}||_1)$ with high probability, while simultaneously ensuring that

[12]For simplicity, the analysis presented here relies only on an upper bound on the length of the path through which the estimation error introduced in any iteration can propagate. This bound follows from known results on size of largest components in sparse hypergraphs [41]. We note, however, that a tighter analysis that relies on a finer characterization of the interaction between the size of these components and the contribution to total estimation error may lead to better bounds on the overall estimation error. Indeed, as shown in [32], such an analysis enables a tighter reconstruction guarantee of the form $||\mathbf{x}-\hat{\mathbf{x}}||_1=\mathcal{O}(||\mathbf{z}||_1+||\mathbf{e}||_1)$

our choice δ satisfies $\delta < C_2||\mathbf{z}||_1$ for some C_1, C_2.

Phase quantization: In the noisy setting, even when i is a leaf node for $\mathcal{S}(\mathbf{x})$, the phase of y_i may differ from the phase assigned by the measurement. To overcome this, we modify our decoding algorithm to work with "quantized" phases, rather than the actual received phases. The idea behind this is that if i is a leaf node for $\mathcal{S}(\mathbf{x})$, then quantizing the phase to one of the values allowed by the measurement identifies the correct phase with a high probability. The following lemma facilitates this simplification. For a desired error probability ϵ', it suffices to let $\alpha = (1/2)\log(1/2\epsilon')$. Appendix C examines this in more detail.

Lemma 4 (*Almost bounded phase noise*)**.** *Let $\mathbf{x}, \mathbf{z} \in \mathbb{R}^n$ with $|x_j| > \delta/k$ for each j. Let $A' \in \mathbb{C}^{m' \times n}$ be a complex valued measurement matrix with the underlying graph $\mathcal{G}$. Let i be a leaf node for $\mathcal{S}(\mathbf{x})$. Let $\Delta\theta_i = |\angle y_i - \angle (A'\mathbf{x})_i|$. Then, for every $\alpha > 0$, $E_{\mathbf{z},\mathbf{e}}(\Delta\theta_i) \leq \sqrt{\frac{2\pi k^2 (dn\sigma_z^2/ck + \sigma_e^2)}{\delta^2}}$ and $\Pr_{\mathbf{z},\mathbf{e}}\left(\Delta\theta_i > \alpha E_{\mathbf{z},\mathbf{e}}(\Delta\theta_i)\right) < \frac{1}{2}e^{-(\alpha^2/\pi)}$.*

Repeated measurements: Our algorithm works by performing a series of $\Gamma \geq 1$ identification and verification measurements in each iteration (instead of a single measurement of each type as done in the exactly k-sparse case). This is because, in the presence of noise, even though a single set of measurements cannot exactly identify j from a leaf y_i, it helps us narrows down the set of coordinates j that could have contributed to give the observed phase. Performing measurements repeatedly, each time with a different measurement, helps us identify the index x_j corresponding to a noisy leaf y_i (with high probability). Hence we first map each $j \in \{1, 2, \ldots, n\}$ to its Γ-digit representation in base $\mathbb{G} = \{0, 1, \ldots \lceil n^{1/\Gamma} - 1 \rceil\}$. For each $j \in \{1, 2, \ldots, n\}$, let $g(j) = (g_1(j), g_2(j), \ldots, g_\Gamma(j))$ be the Γ-digit representation of j. Next, we perform one pair of identification and verification measurements (and corresponding phase reconstructions), each of which is intended to distinguish exactly one of the digits. We show we only need a constant number of such phase measurements per iteration.

Description of measurements: As in the exactly k-sparse case, we start with a randomly drawn left regular bipartite graph $\mathcal{G}$ with n nodes on the left and m' nodes on the right.

Measurement matrix: The measurement matrix $A \in \mathbb{C}^{2m'\Gamma \times n}$ is chosen based on the graph $\mathcal{G}$. The rows of A are partitioned into m' groups, with each group consisting of 2Γ consecutive rows. The j-th entries of the rows $2(i-1)\Gamma+1, (i-1)\Gamma+2, \ldots, 2i\Gamma$ are denoted by $a_{i,j}^{(I,1)}, a_{i,j}^{(I,2)}, \ldots, a_{i,j}^{(I,\Gamma)}, a_{i,j}^{(V,1)}, a_{i,j}^{(V,2)}, \ldots, a_{i,j}^{(V,\Gamma)}$ respectively. In the above notation, I and V are used to refer to identification and verification measurements. For ease of notation, for each $\gamma = 1, 2, \ldots, \Gamma$, we use $A^{(I,\gamma)}$ (resp. $A^{(V,\gamma)}$) to denote the sub-matrix of A whose (i,j)-th entry is $a_{i,j}^{(I,\gamma)}$ (resp. $a_{i,j}^{(V,\gamma)}$). We define the γ-th identification matrix $A^{(I,\gamma)}$ as follows. For each (i,j), if the graph $\mathcal{G}$ does not have an edge connecting i on the right to j on the left, then $a_{i,j}^{(I,\gamma)} = 0$. Otherwise, we set $a_{i,j}^{(I,\gamma)}$ to be the unit-norm complex number $a_{i,j}^{(I,\gamma)} = e^{\iota g_\gamma(j)\pi/2(|\mathbb{G}|-1|)}$. Next, we define the γ-th verification matrix $A^{(V,\gamma)}$ in a way similar to how we defined the verification entries in the exactly k-sparse case. For each (i,j), if the graph $\mathcal{G}$ does not have an edge connecting i on the right to j on the left, then $a_{i,j}^{(V,\gamma)} = 0$. Otherwise, we set $a_{i,j}^{(V,\gamma)} = e^{\iota\theta_{ij}^{(V,\gamma)}}$, where $\theta_{i,j}^{(V,\gamma)}$ is drawn uniformly at random from $\{0, \pi/2(|\mathbb{G}|-1), \pi/(|\mathbb{G}|-1), 3\pi/2(|\mathbb{G}|-1) \ldots, \pi/2\}$. Given an signal vector $\mathbf{x}$, signal noise $\mathbf{z}$, and measurement noise $\mathbf{e}$, the measurement operation produces a measurement vector $\mathbf{y} = A(\mathbf{x} + \mathbf{e})$. Since A can be partitioned into Γ identification and Γ verification rows, we think of the measurement vector $\mathbf{y}$ as a collection of outcomes from Γ successive measurements such that $\mathbf{y}^{(I,\gamma)} = A^{(I,\gamma)}(\mathbf{x} + \mathbf{z}) + \mathbf{e}^{(I,\gamma)}$ and $\mathbf{y}^{(V,\gamma)} = A^{(V,\gamma)}(\mathbf{x} + \mathbf{z}) + \mathbf{e}^{(V,\gamma)}$ are the outcomes from the γ-th measurement and $\mathbf{y} = ((\mathbf{y}^{(I,\gamma)}, \mathbf{y}^{(V,\gamma)}) : 1 \leq \gamma \leq \Gamma)$.

Reconstruction for approximately k-sparse signals with noisy measurements: The decoding algorithm for this case extends the decoding algorithm presented earlier for the exactly k-sparse case by including the ideas presented above. The total number of iterations for our algorithm are upper bounded by $4k$. The decoding algorithm terminates after the T-th iteration, where $T = \min\{4k, \{t : \mathcal{D}(t+1) = \phi\}\}$.

REFERENCES

[1] M. Bakshi, S. Jaggi, S. Cai, and M. Chen, "Order-optimal Compressive Sensing for Approximately k-sparse signals: $\mathcal{O}(k)$ measurements, $\mathcal{O}(k)$ decoding steps ," 2012, poster: http://personal.ie.cuhk.edu.hk/~mayank/CS/poster.pdf, Video: http://youtu.be/UrTsZX7-fhI.

[2] E. J. Candès, J. K. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information." *IEEE Transactions on Information Theory*, pp. 489–509, 2006.

[3] D. L. Donoho, "Compressed sensing." *IEEE Transactions on Information Theory*, pp. 1289–1306, 2006.

[4] E. J. Candès, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, no. 9-10, pp. 589–592, 2008.

[5] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, vol. 28, no. 3, pp. 253–263, December 2008.

[6] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit." *IEEE Transactions on Information Theory*, pp. 4655–4666, 2007.

[7] 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 Transactions on Information Theory*, pp. 1094–1121, 2012.

[8] R. Berinde, P. Indyk, and M. Ruzic, "Practical near-optimal sparse recovery in the l1 norm," *Proceedings of the Annual Allerton conference*, 2008.

[9] R. Berinde and P. Indyk, "Sequential sparse matching pursuit," *Proceedings of the Annual Allerton conference*, 2009.

[10] A. Gilbert and P. Indyk, "Sparse recovery using sparse matrices," *Proceedings of IEEE*, vol. 98, no. 6, pp. 937–947, 2010.

[11] K. D. Ba, P. Indyk, E. Price, and D. Woodruff, "Lower bounds for sparse recovery," *Proceedings of the Symposium on Discrete Algorithms*, 2010.

[12] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300 – 304, Jun 1960.

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

[14] S. Kudekar and H. Pfister, "The effect of spatial coupling on compressive sensing," in *Proc. Annual Allerton Conf. on Commun., Control, and Comp*, 2010.

[15] M. Mitzenmacher and G. Varghese, "Biff (bloom filter) codes: Fast error correction for large data sets," *To appear in ISIT*, 2012.

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

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

[18] Y. Wu and S. Verdú, "Rényi information dimension: Fundamental limits of almost lossless analog compression," *IEEE Transaction on Information Theory*, vol. 56, no. 8, pp. 3721–3748, 2010.

[19] Y. Wu and S. Verdú, "Optimal phase transitions in compressed sensing," *ArXiv.org e-Print archive, arXiv:1111.6822 [cs.IT]*, 2011.

[20] D. L. Donoho, A. Javanmard, and A. Montanari, "Information-theoretically optimal compressed sensing via spatial coupling and approximate message passing," *ArXiv.org e-Print archive, arXiv:1112.0708 [cs.IT]*, 2011.

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

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

[23] Y. Lu, A. Montanari, B. Prabhakar, S. Dharmapurikar, and A. Kabbani, "Counter braids: a novel counter architecture for per-flow measurement," in *Proceedings of the 2008 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, ser. SIGMETRICS '08, 2008, pp. 121–132.

[24] Y. Lu, A. Montanari, and B. Prabhakar, "Counter braids: Asymptotic optimality of the message passing decoding algorithm," in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, Sept. 2008, pp. 209 –216.

[25] Y. Plan and R. Vershynin, "One-bit compressed sensing by linear programming," *ArXiv.org e-Print archive, arXiv:1109.4299 [cs.IT]*, 2011.

[26] L. Jacques, J. N. Laska, P. T. Boufounos, and R. G. Baraniuk, "Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors," *ArXiv.org e-Print archive, arXiv:1104.3160 [cs.IT]*, 2011.

[27] A. C. Gilbert, M. J. Strauss, J. A. Tropp, and R. Vershynin, "Algorithmic linear dimension reduction in the l1 norm for sparse vectors," in *Allerton 2006 (44th Annual Allerton Conference on Communication, Control, and Computing*, 2006.

[28] G. Cormode and S. Muthukrishnan, "Combinatorial algorithms for compressed sensing," in *Information Sciences and Systems, 2006 40th Annual Conference on*, March 2006, pp. 198 –201.

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

[30] M. T. Goodrich and M. Mitzenmacher, "Invertible bloom lookup tables," *ArXiv.org e-Print archive, arXiv:1101.2245 [cs.DS]*, 2011.

[31] D. A. Spielman, "Linear-time encodable and decodable error-correcting codes." in *STOC'95*, 1995, pp. 388–397.

[32] E. Price, "Efficient sketches for the set query problem," in *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '11, 2011, pp. 41–56.

[33] S. Pawar and K. Ramchandran, personal communication, 2012.

[34] M. Bakshi, S. Jaggi, S. Cai, and M. Chen, "SHO-FA: Robust compressive sensing with order-optimal complexity, measurements, and bits," *ArXiv.org eprint archive*, 2012, arXiv:1207.2335v1 [cs.IT].

[35] D. Guo, J. Luo, L. Zhang, and K. Shen, "Compressed neighbor discovery for wireless networks," *CoRR*, vol. abs/1012.1007, 2010.

[36] R. Roth, *Introduction to coding theory*. Cambridge University Press, 2006.

[37] S. Hoory, N. Linial, and A. Wigderson, "Expander graphs and their applications," *Bulletin (New series) of the American Mathematical Society*, vol. 43, pp. 439–561, 2006.

[38] C. Daskalakis, A. Dimakis, R. Karp, and M. Wainwright, "Probabilistic analysis of linear programming decoding," *Information Theory, IEEE Transactions on*, vol. 54, no. 8, pp. 3565 –3578, aug. 2008.

[39] J. T. Schwartz, "Fast probabilistic algorithms for verification of polynomial identities," *Journal of the ACM*, vol. 27, no. 4, p. 701–717, 1980.

[40] R. Zippel, "Probabilistic algorithms for sparse polynomials," *Proceedings of the International Symposiumon on Symbolic and Algebraic Computation*, pp. 216–226, 1979.

[41] M. Karoński and T. Łuczak, "The phase transition in a random hypergraph," *J. Comput. Appl. Math.*, vol. 142, no. 1, pp. 125–135, May 2002.

APPENDIX

A. Proof of Lemma 3

SHO-FA examines the set of all y_i that neighbour j in $\mathcal{G}$, and checks if any of them satisfies $y_i^{(I)} = y_i^{(V)} = 0$. If so, SHO-FA outputs $\hat{x}_j = 0$. Else it checks (using verification and identification observations as in the standard SHO-FA decoding) if there exists a $\mathcal{S}(\mathbf{x})$-leaf node among the neighbours of j. If there is a such a neighbouring leaf node, say i, then SHO-FA outputs $\hat{x}_j = |y_i|$. Else, the algorithm terminates without producing any output. To see this satisfies the database query property, consider the following two cases.

Case 1: $x_j = 0$. In this case, $\hat{x}_j = 0$ is output if at

least one neighbour of j lies outside $N(\mathcal{S}(\mathbf{x}))$. Since $N(\mathcal{S}(\mathbf{x}))$ has at most dk elements, and there are ck y_i's, the probability that none of the neighbours of j lie outside $N(\mathcal{S}(\mathbf{x}))$ is at least $(1-(d/c)^d)$. An error may also occur if all neighbours of j lie within $N(\mathcal{S}(\mathbf{x}))$ and SHO-FA incorrectly identifies one of these as a leaf node – by the analysis of SHO-FA, this event occurs with probability $o(1/k)$.

Case 2: $x_j \neq 0$. For SHO-FA to produce the correct output, it has to identify one of the neighbours of j as a leaf. The probability that there exists a leaf among the neighbours of j is at least $(1-(d/c)^d)$ by an argument similar to the previous case. Similarly, the proabability of erroneous identification is $o(1/k)$. ∎

B. Proof of Lemma 4

Let $\Delta\theta_i$ be the difference in phase between the "noiseless" output $(A'\mathbf{x})_i$ and the actual output $y_i = (A'(\mathbf{x}+\mathbf{z})+\mathbf{e})_i$. By a geometric argument, for fixed $\mathbf{z}$ and $\mathbf{e}$, the phase displacement $\Delta\theta_i$ is upper bounded by $\pi|(A'\mathbf{z})_i + e_i|/|(A'\mathbf{x})_i|$. Since i is a leaf node for $\mathcal{S}(\mathbf{x})$, $|(A'\mathbf{x})_i| \geq |\delta/k|$. Therefore, $\Pr_{\mathbf{z},\mathbf{e}}(\Delta\theta_i > \alpha) \leq Pr_{\mathbf{z},\mathbf{e}}(|(A'\mathbf{z})_i + e_i| > \alpha\delta/\pi k)$. Next, note that $(A'\mathbf{z})_i$ is a Complex Gaussian with zero mean and variance at most $n\sigma_z^2$. Further, each row of A' has at most dn/ck non-zero entries. Therefore, $(A'\mathbf{z})_i + e_i$ is a zero mean complex Gaussian with variance at most $(dn/ck)\sigma_z^2+\sigma_e^2$. Finally, to complete the proof, we apply standard analysis for computing the expectation of the absolute value and bounds on tail probabilities of Gaussian random variables. ∎

C. Probability of error

An error occurs only if one of the following take place:
1) $\mathcal{G}$ is not an $\mathcal{S}(\mathbf{x})$-expander: This occurs with probability $o(1/k)$.
2) Noise in $\tilde{y}_{i(t)}(t)$ leads to an incorrect decoding of $\hat{\theta}_t^{(I,\gamma)}$ or $\hat{\theta}_t^{(V,\gamma)}$ for some γ and t: The phase noise in $\tilde{y}_{i(t)}(t)$ consists of the effect of due to noise vectors $\mathbf{z}$ and $\mathbf{e}$, and the contribution due to the noise propagated while computing each $\tilde{y}_{i(t)}(\tau)$ from $\tilde{y}_{i(t)}(\tau-1)$ for $\tau \leq t$. The contribution due to the first term is bounded by Lemma 4. Thus for a target error probability ϵ', we choose $\alpha = (1/2)\log 1/2\epsilon'$, giving a contribution to the phase noise of at most $\frac{\log(1/2\epsilon')}{2}\sqrt{\frac{2\pi k^2(dn\sigma_z^2/ck+\sigma_e^2)}{\delta^2}}$. To bound the contribution due to the second term, note that at each iteration t, any error in reconstruction of $\hat{x}_{j(t)}$ potentially adds to reconstruction error in all future iterations t' for which there is a path from $j(t)$ to $j(t')$. Since the restriction of $\mathcal{G}$ to $\mathcal{S}(\mathbf{x})$ and its neighbours is a sparse graph, it follows from [41] that, with probability $1-o(1/k)$, it consists only of disjoint components of size $\mathcal{O}(\log k)$ (see [32] for such an analysis). Thus, the magnitude error in reconstruction of $\hat{x}_{j(t)}$ due to noisy reconstructions in previous iterations is

$$\mathcal{O}\left((\log k)^2 \log(1/\epsilon')\sqrt{(n\sigma_z^2/k+\sigma_e^2)}\right). \quad (1)$$

Thus, the order of the phase displacement in each $y_i^{(I,\gamma)}$ and $y_i^{(V,\gamma)}$ is at most $(\log k)^2\log(1/\epsilon')\sqrt{\frac{k^2(n\sigma_z^2/k+\sigma_e^2)}{\delta^2}}$. Therefore, as long as $(\log k)^2\log(1/\epsilon')\sqrt{\frac{k^2(n\sigma_z^2/k+\sigma_e^2)}{\delta^2}} = o\left(n^{-1/\Gamma}\right)$, the probability of any single phase being incorrectly detected is upper bounded by ϵ'. Since we there are a total of $8\Gamma k$ possible phase measurements, we choose $\epsilon' = \mathcal{O}(1/\Gamma k^2)$ to achieve a target error probability $\mathcal{O}(1/k)$.
3) The verification step passes for each measurement in the t-th measurement, even though $i(t)$ is not a leaf node for $\mathcal{S}_\delta^c(\mathbf{x})$. The probability of this even is again $\mathcal{O}(1/k)$ by our choice of ϵ'.
4) $\mathcal{D}(T) \neq \phi$, i.e., the algorithm terminates without recovering all x_j's. By Lemma 2, the probability that $i(t)$ is a leaf node for $\mathcal{S}_\delta(\mathbf{x}-\hat{\mathbf{x}}(t))$ is at least $1/2$. The probability that a leaf node $i(t)$ does not pass the verification tests is $\mathcal{O}(1/k)$. Thus, in expectation, the number of iterations required by the algorithm is $2k/(1-\mathcal{O}(1/k))$. By concentration arguments, it follows that the probability that the algorithm does not terminate in $4k$ iterations is $o(1/k)$ as k grows without bound.

D. Estimation error

Next, we bound the error in estimating $\hat{\mathbf{x}}$. We first find an upper bound on $||\hat{\mathbf{x}} - \mathbf{x}_{\mathcal{S}_\delta^c}||_1$ that holds with a high probability. Applying the bound in (1), for each $t = 1,2,\ldots,T$, $|x_{j(t)} - \hat{x}_{j(t)}| = \mathcal{O}\left((\log k)^2\log(1/\epsilon')\sqrt{(n\sigma_z^2/k+\sigma_e^2)}\right)$ with probability $1-o(1/k)$. Therefore, $||\hat{\mathbf{x}} - \mathbf{x}_{\mathcal{S}_\delta^c}||_1$ is of order $k(\log k)^2\log(1/\epsilon')\left(\sqrt{n\sigma_z^2/k}+\sigma_e\right)+\delta$. Next, note that $||\mathbf{z}||_1 = \sum_{j=1}^n |z_j|$ and $||\mathbf{e}||_1 = \sum_{i=1}^m |e_i|$. Since each z_j is a Gaussian random variable with variance σ_z^2, The expected value of $|z_j|$ is $\sigma_z\sqrt{2/\pi}$. By concentrating $||\mathbf{z}||_1$ and $||\mathbf{e}||_1$ around their respective means, and applying the bound in (III), we conclude that, with a high probability, $||\hat{\mathbf{x}}-\mathbf{x}||_1$ is of order $\sqrt{\frac{k}{n}}(\log k)^2\log(1/\epsilon')||\mathbf{z}||_1 + (\log k)^2\log(1/\epsilon')||\mathbf{e}||_1 + 2\delta$.

E. Proof of Theorem 2

Finally, to complete the proof of Theorem 2, we let $\delta = \min\{\mathcal{O}(n\sigma_z), o(1)\}$. By the argument in the previous subsection, with a high probability, $\delta = \mathcal{O}(||\mathbf{z}||)$. Finally, recall the assumption that $k = \mathcal{O}(n^{1-\Delta})$. Applying these to the bound obtained in the previous subsection, we get $||\hat{\mathbf{x}} - \mathbf{x}||_1 \leq C\left(||\mathbf{z}||_1 + (\log k)^2||\mathbf{e}||_1\right)$.