# $\ell_2/\ell_2$-Foreach Sparse Recovery with Low Risk

Anna C. Gilbert[1,⋆], Hung Q. Ngo[2], Ely Porat[3],
Atri Rudra[2], and Martin J. Strauss[1]

[1] University of Michigan
[2] University at Buffalo (SUNY)
[3] Bar-Ilan University

**Abstract.** In this paper, we consider the "foreach" sparse recovery problem with failure probability $p$. The goal of the problem is to design a distribution over $m \times N$ matrices $\Phi$ and a decoding algorithm $A$ such that for every $\mathbf{x} \in \mathbb{R}^N$, we have with probability at least $1 - p$

$$\|\mathbf{x} - A(\Phi\mathbf{x})\|_2 \leqslant C\|\mathbf{x} - \mathbf{x}_k\|_2,$$

where $\mathbf{x}_k$ is the best $k$-sparse approximation of $\mathbf{x}$.

Our two main results are: (1) We prove a lower bound on $m$, the number measurements, of $\Omega(k\log(n/k) + \log(1/p))$ for $2^{-\Theta(N)} \leqslant p < 1$. Cohen, Dahmen, and DeVore [4] prove that this bound is tight. (2) We prove nearly matching upper bounds that also admit *sub-linear* time decoding. Previous such results were obtained only when $p = \Omega(1)$. One corollary of our result is an an extension of Gilbert et al. [6] results for information-theoretically bounded adversaries.

## 1 Introduction

In a large number of modern scientific and computational applications, we have considerably more data than we can hope to process efficiently and more data than is essential for distilling useful information. Sparse signal recovery [7] is one method for both reducing the amount of data we collect or process initially and then, from the reduced collection of observations, recovering (an approximation to) the key pieces of information in the data. Sparse recovery assumes the following mathematical model: a data point is a vector $\mathbf{x} \in \mathbb{R}^N$, using a matrix $\Phi$ of size $m \times N$, where $m \ll N$, we collect "measurements" of $\mathbf{x}$ non-adaptively and linearly as $\Phi\mathbf{x}$; then, using a "recovery algorithm" $A$, we return a good approximation to $\mathbf{x}$. The error guarantee must satisfy

$$\|\mathbf{x} - A(\Phi\mathbf{x})\|_2 \leqslant C\|\mathbf{x} - \mathbf{x}_k\|_2, \tag{1}$$

where $C$ is a constant (ideally arbitrarily close to 1) and $\mathbf{x}_k$ is the best $k$-sparse approximation of $\mathbf{x}$. This is customarily called an $\ell_2/\ell_2$-*error guarantee* in the

⋆ A full version of this paper may be found at `http://arxiv.org/abs/1304.6232`. AG is supported in part by NSF CCF 1161233, HN by NSF grant CCF-1161196, AR by NSF CAREER grant CCF-0844796 and NSF grant CCF-1161196, and MS by NSF CCF 0743372 and NSF CCF 1161233.

F.V. Fomin et al. (Eds.): ICALP 2013, Part I, LNCS 7965, pp. 461–472, 2013.

literature. This paper considers the sparse recovery problem with failure probability $p$, the goal of which is to design a distribution over $m \times N$ matrices $\Phi$ and a decoding algorithm $A$ such that for every $\mathbf{x} \in \mathbb{R}^N$, the error guarantee holds with probability at least $1 - p$. The reader is referred to [7] and the references therein for a survey of sparse matrix techniques for sparse recovery, and to [1] for a collection of articles (and the references therein) that emphasize the applications of sparse recovery in signal and image processing.

There are many parameters of interest in the design problem: (i) number of measurements $m$; (ii) decoding time, i.e. runtime of algorithm $A$; (iii) approximation factor $C$ and (iv) failure probability $p$. We would like to minimize all the four parameters simultaneously. It turns out, however, that optimizing the failure probability $p$ can lead to wildly different recovery schemes. Much of the sparse recovery or compressive sensing literature has focused on the case of either $p = 0$ (which is called the "*forall*" model) or $p = \Omega(1)$ (the "*foreach*" model). Cohen, Dahmen, and DeVore [5] showed a lower bound of $m = \Omega(N)$ for the number of measurements when $p = 0$, rendering a sparse recovery system useless as one must collect (asymptotically) as many measurements as the length of the original signal[1]. Thus, algorithmically there is not much to do in this regime.

The case of $p \geqslant \Omega(1)$ has resulted in much more algorithmic success. Candés and Tao showed in [2] that $O(k \log(N/k))$ random measurements with a polynomial time recovery algorithm are sufficient for compressible vectors and Cohen et al. [5] show that $O(k \log(N/k))$ measurements are sufficient for any vector (but the recovery algorithm given is not polynomial time). In a subsequent paper, Cohen et al. [4] give a polynomial time algorithm with $O(k \log(N/k))$ measurements. The next goal was to match the $O(k \log(N/k))$ measurements but with *sub-linear* time decoding. This goal was achieved by Gilbert, Li, Porat, and Strauss [8] who showed that there is a distribution on $m \times N$ matrices with $m = O(k \log(N/k))$ and a decoding algorithm $A$ such that, for each $\mathbf{x} \in \mathbb{R}^N$ the $\ell_2/\ell_2$-error guarantee is satisfied with probability $p = \Theta(1)$. The next natural goal was to nail down the correct dependence on $C = 1+\varepsilon$. Gilbert et al.'s result actually needs $O(\frac{1}{\varepsilon} k \log(N/k))$ measurements. This was then shown to be tight by Price and Woodruff [21].

At this point, we completely understand the problem for the case of $p = 0$ or $p = \Omega(1)$. Somewhat surprisingly, there is *no* work that has explicitly considered the $\ell_2/\ell_2$ sparse recovery problem when $0 < p \leqslant o(1)$. The main goal of this paper is to close this gap in our understanding.

Given the importance of the sparse recovery problem, we believe that it is important to close the gap. Similar studies have been done extensively in a closely related field: coding theory. While the model of worst-case errors pioneered by Hamming (which corresponds to the forall model) and the oblivious/stochastic error model pioneered by Shannon (which corresponds to the foreach model) are most well-known, there is a rich set of results in trying to understand the

[1] For this reason, all of the forall sparse signal recovery results satisfy a different, weaker error guarantee. E.g. in the $\ell_1/\ell_1$ forall sparse recovery we replace the condition (1) by $\|\mathbf{x} - A(\Phi\mathbf{x})\|_1 \leqslant C\|\mathbf{x} - \mathbf{x}_k\|_1$.

power of intermediate channels, including the arbitrarily varying channel [14]. Another way to consider intermediate channels is to consider computationally bounded adversaries [16]. Gilbert et al. [6] considered a computationally bounded adversarial model for the sparse recovery problem in which signals are generated neither obliviously (as in the foreach model) nor adversarially (in the forall model) in order to interpolate between the forall and foreach signal models. Our results in this paper imply new results for the $\ell_1/\ell_1$ sparse recovery problem as well as the $\ell_2/\ell_2$ sparse recovery problem against bounded adversaries.

Our main contributions are as follows.

1. We prove that the number measurements has to be $\Omega(k\log(N/k)+\log(1/p))$ for $2^{-\Theta(N)} \leqslant p < 1$.
2. We prove nearly matching upper bounds that also admit *sub-linear* time decoding.
3. We present applications of our result to obtain
   (i) the best known number of measurements for $\ell_1/\ell_1$ *forall* sparse recovery with sublinear $(\mathrm{poly}(k, \log N))$ time decoding (in [9]), and
   (ii) nearly tight upper and lower bounds on the number of measurements needed to perform $\ell_2/\ell_2$-sparse recovery against information-theoretically bounded adversary.

As was mentioned earlier, there are many parameters one could optimize. We will not pay very close attention to the approximation factor $C$, other than to stipulate that $C \leqslant O(1)$. In most of our upper bounds, we can handle $C = 1+\varepsilon$ for an arbitrary constant $\varepsilon$, but optimizing the dependence on $\varepsilon$ is beyond the scope of this paper.

*Lower Bound Result.* We prove a lower bound of $\Omega(\log(1/p))$ on the number of measurements when the failure probability satisfies $2^{-\Theta(N)} \leqslant p < 1$. (When $p \leqslant 2^{-\Omega(N)}$, our results imply a tight bound of $m = \Omega(N)$.) The $\Omega(\log(1/p))$ lower bound along with the lower bound of $\Omega(k\log(N/k))$ from [21] implies the final form of the lower bound claimed above. The obvious follow-up question is whether this bound is tight. Indeed, an upper bound result Cohen, Dahmen, and DeVore [4] proves that this bound is tight if we only care about polynomial time decoding. Thus, the interesting algorithmic question is how close we can get to this bound with sub-linear time decoding.

*Upper Bound Results.* For the upper bounds, we provide several algorithms that span the trade-offs between number of measurements and failure probability. For completeness, we include the running times and the space requirements of the algorithms and measurement matrices in Table 1, which summarizes our main results and compares them with existing results.

We begin by first considering the most natural way to boost the failure probability of a given $\ell_2/\ell_2$ sparse recovery problem: we repeat the scheme $s$ times with independent randomness and pick the "best" answer– see the full version [9] for more details. This boosts the decoding error probability from the original $p$ to $p^{\Omega(s)}$– though the reduction does blow up the approximation factor by a multiplicative factor of $\sqrt{3}$.

**Table 1.** Summary of algorithmic results. The results in [20] are for $\ell_1/\ell_1$ forall sparse recovery but their results can be easily adapted to our setting with our proofs. $c$ is some constant $\geqslant 8$ and $\alpha > 0$ is any arbitrary constant and we ignore the constant factors in front of all the expressions.

| Reference | $k$ | $m$ | $p$ | Decoding time | Space |
|---|---|---|---|---|---|
| [4] | Any $k$ | $k\log(N/k) + \log(1/p)$ | Any $p$ | $\text{poly}(N)$ | $\text{poly}(N)$ |
| [8] | Any $k$ | $k\log(N/k)$ | $p \geqslant \Omega(1)$ | $k \cdot \text{poly}\log N$ | $k \cdot \text{poly}\log N$ |
| [20] | $k \geqslant N^{\Omega(1)}$ | $k\log(N/k)$ | $p = (N/k)^{-k/\log^c k}$ | $k^{1+\alpha}\text{poly}\log N$ | $Nk^{0.2}$ |
| | Any $k$ | $k\log(N/k)\log_k^c N$ | $p = k^{-k/\log^c k}$ | $k^{1+\alpha}\text{poly}\log N$ | $Nk^{0.2}$ |
| [8] + weak/top conv. | Any $k$ | $k\log(N/k)$ | $p \geqslant 2^{-k/\log^c k}$ | $k \cdot \text{poly}\log N$ | $k \cdot \text{poly}\log N$ |
| This paper | $k \geqslant N^{\Omega(1)}$ | $k\log(N/k)$ | $p = (N/k)^{-k/\log^c k}$ | $k^{1+\alpha}\text{poly}\log N$ | $k \cdot \text{poly}\log N$ |
| | Any $k \geqslant \log(N/k)$ | $k\log(N/k)\log_k^\alpha N$ | $p = (N/k)^{-k/\log^c k}$ | $k^{2^{\alpha^{-1}}}\text{poly}\log N$ | $k \cdot \text{poly}\log N$ |

The above implies that if we can optimally solve the $\ell_2/\ell_2$ sparse recovery problem for $p \geqslant (N/k)^{-k}$ (i.e. with $O(k\log(N/k))$ measurements), then we can solve the problem optimally for smaller $p$. Hence, for the rest of the description we focus on the case $p \geqslant (N/k)^{-O(k)}$ (where the goal is to obtain $m = O(k\log(N/k))$). Note that in this case, the amplification does not help as even for $p = \Omega(1)$, previous results (e.g. [21]) imply that $m \geqslant \Omega(k\log(N/k))$. Thus, if the original decoding error probability is $p$ then to obtain the $(N/k)^{-k}$ decoding error probability implies that the number of measurements will be larger than the optimal value a factor of $k\log(N/k)/\log(1/p)$. As we will see shortly the best know upper bound can achieve $p = 2^{-k}$, which implies that amplification will be larger than the optimal value of $\Omega(k\log(N/k)$ by a factor of $\log(N/k)$. In this work, we show how to achieve the same goal with an asymptotically smaller blow-up.

For $p \geqslant (N/k)^{-k}$, there are two related works. The first is that of Porat and Strauss [20] who considered the sparse recovery problem under the $\ell_1/\ell_1$ forall guarantee. Despite the different error guarantee, our construction is closely related to that of [20] and our proofs imply the results for [20] listed in Table 1. Note that the results for polynomially large $k$ are pretty much the same except we have a better space complexity. For general $k$, our result also has better number of measurements and failure probability guarantee. The second work is that of Gilbert et al. [8]. Even though the results in that paper are cited for $p \geqslant \Omega(1)$, it can be shown that if one uses $O(k)$-wise independent random variables instead of the pair-wise independent random variables as used in [8], one can obtain a "weak system" with failure probability $2^{-k}$. Then our "weak system to top level system conversion" leads to the result claimed in the second to last row in Table 1. Our results have a better failure probability at the cost of larger number of measurements.

It is natural to ask whether decreasing the failure probability (the base changed from 2 to $(N/k)$) is worth giving up the optimality in the number of measurements (which is what [8] obtains). Note that achieving a failure probability of $(N/k)^{-k}$ is a very natural goal and our results are better than those in [8] when we anchor on the failure probability goal first.

*Bounded Adversary Results.* We also obtain some results for $\ell_2/\ell_2$-sparse recovery against information-theoretically-bounded adversaries as considered by Gilbert et al. [6]. (See Section 2 for a formal definition of such bounded adversaries.) Gilbert et al. show that $O(k \log(N/k))$ measurements is sufficient for such adversaries with $O(\log N)$ bits of information. Our results allow us to prove results for a general number of information bits bound of $s$. In particular, we observe that for such adversaries $O(k \log(N/k) + s)$ measurements suffice. Further, if one desires sublinear time decoding then our results in Table 1 allows for a similar conclusion but with extra $\operatorname{poly} \log k$ factors. We also observe that one needs $\tilde{\Omega}(\sqrt{s})$ many measurements against such an adversary (assuming the entries are polynomially large). In the final version of the paper, we will present a proof suggested to us by an anonymous reviewer that leads to the optimal $\Omega(s)$ lower bound.

*Lower Bound Techniques.* Our lower bound technique is inspired by the geometric approach of Cohen et al. [5] for the $p = 0$ case. Our bound holds for the entire range of failure probability $p$. Our technique also yields a simpler and more intuitive proof of Cohen et al. result. Both results hold even for sparsity $k = 1$.

The technical crux of the lower bound result in [5] for $p = 0$ is to show that any measurement matrix $\Phi$ with $O(N/C^2)$ rows has a null space vector $\mathbf{n}$ that is "non-flat," – i.e. $\mathbf{n}$ has one coordinate that has most of the mass of $\mathbf{n}$. On the other hand since $\Phi\mathbf{n} = \mathbf{0}$, the decoding algorithm $A$ has to output the same answer when $\mathbf{x} = \mathbf{n}$ and when $\mathbf{x} = \mathbf{0}$. It is easy to see that then $A$ does not satisfy (1) for at least one of these two cases (the output for $\mathbf{0}$ has to be $\mathbf{0}$ while the output for $\mathbf{n}$ has to be non-flat and in particular not $\mathbf{0}$).

To briefly introduce our technique, consider the case of $p = 2^{-N}$ (where we want a lower bound of $m = \Omega(N)$). The straightforward extension of Cohen, et al.'s argument is to define a distribution over, say, all the unit vectors in $\mathbb{R}^N$, argue that this gives a large measure of "bad vectors," and then apply Yao's minimax lemma to obtain our final result; i.e., that there are "a lot" of non-flat vectors in the null space of a given matrix $\Phi$. This argument fails because the distribution on the bad vectors must be independent of the measurement matrix $\Phi$ (and algorithm $A$) in order to apply Yao's lemma but null space vectors, of course, depend on $\Phi$. On the other hand, if we define the "hard" distribution to be the uniform distribution, then the measure of null space vectors for any $m \geqslant 1$ is zero, and thus this obvious generalization does not work.

We overcome this obstacle with a simple idea. The hard distribution is still the uniform distribution on the unit sphere $S^{N-1}$. We first show that there is a region $R$ on this sphere with large measure ($\geqslant p$) such that *all* vectors in $R$ have a *positive* "spike" (large mass) at one particular coordinate $j^* \in [N]$. (The region $R$ is simply a small *spherical cap* about the unit vector $\mathbf{e}_{j^*}$.) In particular, to recover an input vector $\mathbf{v} \in R$, the algorithm has to assign a large positive mass to the $j^*$th coordinate of $A(\Phi\mathbf{v})$. Next, by applying a certain invertible linear reflector to $R$, we can construct a region $R'$ (which is also a region on the sphere, and is just a reflection of $R$) with the same measure satisfying the following: for

each vector $\mathbf{v} \in R$, the reflection $\mathbf{v}'$ of $\mathbf{v}$ ($\mathbf{v}' \in R'$) has a *negative* spike at the same coordinate $j^*$; furthermore, $\Phi\mathbf{v} = \Phi\mathbf{v}'$, which forces the algorithm $A$ into a dichotomy. The algorithm can not recover both $\mathbf{v}$ and $\mathbf{v}'$ well at once. Roughly speaking, the algorithm will be wrong with probability at least half the total measure of $R$ and $R'$, which is $p$. Finally, Yao's lemma completes the lower bound proof. There are some additional technical obstacles that we need to overcome in this step—see [9] for more details.

*Upper Bound Techniques.* We believe that our main algorithmic contributions are the new techniques that we introduce in this paper, which should be useful in (similar) applications.

Our upper bounds follows the same outline used by Gilbert, Li, Porat and Strauss [8] and Porat and Strauss [20]. At a high level, the construction follows three steps. The first step is to design an "identification scheme," which in sub-linear time computes a set $S \subseteq [N]$ of size roughly $k$ that contains $\Omega(k)$ of the "heavy hitters." (Heavy hitters are the coordinates where if the output vector does not put in enough mass then (1) will not be satisfied.) In the second step, we develop a "weak level system" which essentially estimates the values of coordinates in $S$. Finally, using a loop invariant iterative scheme, we convert the weak system into a "top level system," which is the overall system that we want to design. (The way this iterative procedure works is that it makes sure that after iteration $i$, one is missing only $O(k/2^i)$ heavy hitters– so after $\log k$ steps we would have recovered all of them.) The last two steps are designed to run in time $|S| \cdot \operatorname{poly} \log N$, so if the first step runs in sub-linear time, then the overall procedure is sub-linear. [2]. Our main contribution is in the first step, so we will focus on the identification part here. The second step (taking median of measurements like Count-Sketch [3]) is standard [12].

In order to highlight and to summarize our technical contributions, we present an overview of the scheme in [20] (when adapted to the $\ell_2/\ell_2$ sparse recovery problem). We focus only on the identification step. For near-linear time identification, one uses a lossless bipartite expander where each edge in the adjacency matrix is replaced by a random $\pm 1$ value. The intuition is that because of the expansion property most heavy hitters will not collide with another heavy hitter in most of the measurements it participates in. Further, the expansion property implies that the $\ell_2^2$ noise in most of the neighboring measurements will be low. (The random $\pm 1$ is a standard trick to convert this to an low $\ell_2$ noise.) Thus, if we define the value of an index to be median value of all the measurements, then we should get very good estimates for most of the heavy hitters (and in particular, we can identify them by outputting the top $O(k)$ median values). Since this step implies computing $N$ medians overall we have a near linear time computation. However, note that if we had access to a subset $S' \subseteq [N]$ that had

[2] We would also like to point out that Gilbert et al.'s construction has a failure probability of $\Omega(1)$ in the very first iteration of the last step (weak to top level system conversion) and it seems unlikely that this can be made smaller without significantly changing their scheme.

most of the heavy hitters in it, we can get away with a run time nearly linear in $|S'|$ (by just computing the medians in $S'$).

This seems like a chicken and egg problem as the set $S'$ is what we were after to begin with! Porat and Strauss use recursion to compute $S'$ in sub-linear time. (The scheme was also subsequently used by Ngo, Porat and Rudra to design near optimal sub-linear time decodable $\ell_1/\ell_1$ forall sparse recovery schemes for non-negative signals [19].) To give the main intuition, consider the scheme that results in $\tilde{O}(\sqrt{N})$ identification time. We think of the domain $[N]$ as $L \times R$, where both $L$ and $R$ are isomorphic to $[\sqrt{N}]$. (Think of $L$ as the first $\frac{\log N}{2}$ bits in the $\log N$-bit representation of any index in $[N]$ and $R$ to be the remaining bits.) If one can obtain lists $S_L \subset L$ and $S_R \subset R$ that contain the projections of the heavy hitters in $L$ and $R$, respectively, then $S_L \times S_R$ will contain all the heavy hitters, i.e. $S' \subseteq S_L \times S_R$. (We can use the near linear time scheme to obtain $S_L$ and $S_R$ in $\tilde{O}(\sqrt{N})$ time in the base case. One also has to make sure that when going from $[N]$ to a domain of size $\sqrt{N}$, not too many heavy hitters collide. This can be done by, say, randomly permuting $[N]$ before applying the recursive scheme.) The simplest thing to do would be to set $S' = S_L \times S_R$. However, since both $|S_L|$ and $|S_R|$ can be $\Omega(k)$, this step itself will take $\Omega(k^2)$ time, which is too much if we are shooting for a decoding time of $k^{1+\alpha}\text{poly}\log N$ for $\alpha < 1$. The way Porat and Strauss solved this problem was to store the whole inversion map as a table. This allowed $k \cdot \text{poly}\log N$ decoding time but the scheme ended up needing $\Omega(N)$ space overall.

To get a running time of $k^{1+\alpha}\text{poly}\log N$ one needs to apply the recursive idea with more levels. One can think of the whole procedure as a recursion tree with $\mathcal{N} = O(\log_k N)$ nodes. Unfortunately, this process introduces another technical hurdle. At each node, the expander based scheme loses some, say $\zeta$, fraction of the heavy hitters. To bound the overall fraction of lost heavy hitters, Porat and Strauss use the naive union bound of $\zeta \cdot \mathcal{N}$. However, we need the overall fraction of lost heavy hitters to be $O(1)$. This in turn introduces extra factors of $\log_k N$ in the number of measurements (resulting in the ultimate number of measurements of $k \log(N/k) \log_k^8 N$ in [20]).

We are now ready to present the new ideas that improve upon Porat and Strauss' solutions to solve the two issues raised above. Instead of dividing $[N]$ into $[\sqrt{N}] \times [\sqrt{N}]$, we first apply a code $\mathcal{C} : [N] \to [\sqrt[b]{N}]^r$. (Note that the Porat Strauss construction corresponds to the case when $r = b = 2$ and $\mathcal{C}$ just "splits" the $\log N$ bits into two equal parts.) Thus, in our recursive algorithm at the root we will get $r$ subsets $S_1, \ldots, S_r \subseteq [\sqrt[b]{N}]$ with the guarantee that for (most) $i \in [r]$, $S_i$ contains $\mathcal{C}(j)_i$ for most heavy hitters $j$. Thus, we need to recover the $j$'s for which the condition in the last sentence is true. This is *exactly* the list recovery problem that has been studied in the coding theory literature. (See e.g. [22].) Thus, if we can design a code $\mathcal{C}$ that solves the list recovery problem very efficiently, we would solve the first problem above[3]. For the second problem,

[3] We would like to point out that [19] also uses list recoverable codes but those codes are used in a different context: they used it to replace expanders and further, the codes have the traditional parameters.

note that since we are using a code $\mathcal{C}$, even if we only have $\mathcal{C}(j)_i \in S_i$ for say $r/2$ positions $i \in [r]$, we can recover all such indices $j$. In other words, unlike in the Porat Strauss construction where we can lose a heavy hitter even if we lose it in any of the $\mathcal{N}$ recursive call, in our case we only lose a heavy hitter if it is lost in *multiple* recursive calls. This fact allows us to do a better union bound than the naive one used in [20].

The question then is whether there exists code $\mathcal{C}$ with the desired properties. The most crucial part is that the code needs to have a decoding algorithm whose running time is (near) linear in $\max_{i\in[r]} |S_i|$. Further, we need such codes with $r = O(1)$, i.e. of constant block length independent of $\max_{i\in[r]} |S_i|$. Unfortunately, the known results on list recovery, be it for Reed-Solomon codes [11] or folded Reed-Solomon codes [10] do not work well in this regime—these results need $r \geqslant \Omega(\max_{i\in[r]} |S_i|)$, which is way too expensive. For our setting, the best we can do with Reed-Solomon list recovery is to do the naive thing of going through all possibilities in $\times_{i\in[r]} S_i$. (These codes however can correct for optimal number of errors and lead to our result in the last row of Table 1.) Fortunately, a recent result of Ngo, Porat, Ré and Rudra [18] gave an algorithmic proof of the Loomis-Whitney inequality [17]. The (combinatorial) Loomis-Whitney inequality has found uses in theoretical computer science before [13,15]. In this work, we present the first application of the algorithmic Loomis Whitney inequality of [18] and show that it naturally defines a code $C$ with the required (algorithmic) list recoverability. This code leads to the result in the second to last row of Table 1. Interestingly, we get *optimal* weak level systems by this method. We lose in the final failure probability because of the weak level to top level system conversion.

We conclude the contribution overview by pointing out three technical aspects of our results.

- As was mentioned earlier, we first randomly permute the columns of the matrix to make the recursion work. To complete our identification algorithm, we need to perform the inverse operation on the indices to be output. The naive way would be to use a table lookup, which will require $O(N \log N)$ space, but would still be an improvement over [20]. However, we are able to exploit the specific nature of the recursive tree and the fact that our main results use the Reed-Solomon code and the code based on Loomis-Whitney inequality to have sub-linear space usage.
- In the weak level to top level system, both Gilbert et al., and Porat and Strauss decrease the parameters geometrically—however in our case, we need to use different decay functions to obtain our failure probability.
- Unlike the argument in [20] we explicitly use an expander while Porat and Strauss used a random graph. However, because of this, [20] need at least $N$-wise independence in their random variables to make their argument go through. Our use of expanders allows us to get away with using only $\tilde{O}(k)$-wise independence, which among others leads to our better space usage.

## 2 Preliminaries

We fix notations, terminology, and concepts that will be used throughout the paper. Let $[N]$ denote the set $\{1, \ldots, N\}$. Let $G : [N] \times [\ell] \to [M]$ be an $\ell$-regular bipartite graph, and $\mathcal{M}_G$ be its adjacency matrix. We will often switch back and forth between the graph $G$ and the matrix $\mathcal{M}_G$.For any subset $S \subseteq [N]$, let $\Gamma(S) \subseteq [M]$ denote the set of neighbor vertices of $S$ in $G$. Further, let $\mathcal{E}(S)$ denote the set of edges incident on $S$. A bipartite graph $G : [N] \times [\ell] \to [M]$ is a $(t, \varepsilon)$-expander if for every subset $S \subseteq [N]$ of $|S| \leqslant t$, we have $|\Gamma(S)| \geqslant |S|\ell(1-\varepsilon)$. Several expander properties used in our proofs are listed in the complete paper [9], along with some probability basics.

*Sparse Recovery Basics.* For a vector $\mathbf{x} = (x_i)_{i=1}^N \in \mathbb{R}^N$, the set of $k$ highest-magnitude coordinates of $\mathbf{x}$ is denoted by $H_k(\mathbf{x})$. Such elements are called *heavy hitters*. Every element $i \in [N] \setminus H_k(\mathbf{x})$ such that $|x_i| \geqslant \sqrt{\frac{\zeta^2\eta}{k}} \cdot \|\mathbf{z}\|_2$ will be called a *heavy tail* element. Here, $\zeta$ and $\eta$ are constants that will be clear from context. All the remaining indices will be called *light tail* elements; let $\mathcal{L}$ denote the set of light tail elements. A vector $\mathbf{w} = (w_i)_{i=1}^N \in \mathbb{R}^N$ is called a *flat tail* if $w_i = 1/|S|$ for every non-zero $w_i$, where $S = \mathrm{supp}(\mathbf{w})$.

**Definition 1.** *A probabilistic $m \times N$ matrix $\mathcal{M}$ is called an $(k, C)$-approximate sparse recovery system or* $(k, C)$-top level system *with failure probability $p$ if there exists a* decoding algorithm *$A$ such that for every $\mathbf{x} \in \mathbb{R}^N$, the following holds with probability at least $1 - p$:*

$$\|\mathbf{x} - A(\mathcal{M}\mathbf{x})\|_2 \leqslant C \cdot \|\mathbf{x} - \mathbf{x}_{H_k(\mathbf{x})}\|_2.$$

*The parameter $m$ is called the number of measurements of the system.*

**Definition 2.** *A probabilistic matrix $\mathcal{M}$ with $N$ columns is called a $(k, \zeta, \eta)$-*weak identification *matrix with $(\ell, p)$-*guarantee *if there is an algorithm that, given $\mathcal{M}\mathbf{x}$ and a subset $S \subseteq [N]$, with probability at least $1 - p$ outputs a subset $I \subseteq S$ such that (i) $|I| \leqslant \ell$ and (ii) at most $\zeta k$ of the elements of $H_k(\mathbf{x})$ are not present in $I$. The time taken to compute $I$ will be called* identification time.

**Definition 3.** *We will call a (random) $m \times N$ matrix $\mathcal{M}$ a $(k, \zeta, \eta)$ weak $\ell_2/\ell_2$ system if the following holds for any vector $\mathbf{x} = \mathbf{y} + \mathbf{z}$ such that $|\mathrm{supp}(\mathbf{y})| \leqslant k$. Given $\mathcal{M}\mathbf{x}$ one can compute $\hat{\mathbf{x}}$ such that there exist $\hat{\mathbf{y}}, \hat{\mathbf{z}}$ that satisfy the following properties: (1) $\mathbf{x} = \hat{\mathbf{x}} + \hat{\mathbf{y}} + \hat{\mathbf{z}}$; (2) $|\mathrm{supp}(\hat{\mathbf{x}})| \leqslant O(k/\eta)$;[4] (3) $|\mathrm{supp}(\hat{\mathbf{y}})| \leqslant \zeta k$; (4) $\|\hat{\mathbf{z}}\|_2 \leqslant (1 + O(\eta)) \cdot \|\mathbf{z}\|_2$*

*Bounded Adversary Model.* We summarize the relevant definitions of computationally bounded adversaries from [6]. In this setting, Mallory is the name of the process that generates inputs $x$ to the sparse recovery problem. We recall two definitions for Mallory: (i) **Oblivious:** Mallory cannot see the matrix $\Phi$ and

[4] This part is different from the weak system in [20], where we have $|\mathrm{supp}(\hat{\mathbf{x}})| \leqslant O(k)$.

generates the signal $x$ independent from $\Phi$. For sparse signal recovery, this model is equivalent to the "foreach" signal model. (ii) **Information-Theoretic:** Mallory's output has bounded mutual information with the matrix. To cast this in a computational light, we say that an algorithm $M$ is *(s-)information-theoretically-bounded* if $M(x) = M_2(M_1(x))$, where the output of $M_1$ consists of at most $s$ bits. This model is similar to that of the "information bottleneck" [23].

Lemma 1 of [6] relates the information-theoretically bounded adversary to a bound on the success probability of an oblivious adversary. We re-state the lemma for completeness:

**Lemma 1.** *Pick $\ell = \ell(N)$, and fix $0 < \alpha < 1$. Let $A$ be any randomized algorithm which takes input $x \in \{0,1\}^N$, $r \in \{0,1\}^m$, and "succeeds" with probability $1 - \beta$. Then for any information theoretically bounded algorithm $M$ with space $\ell$, $A(M(r), r)$ succeeds with probability at least*

$$\min\{1 - \alpha,\ 1 - \ell/\log(\alpha/\beta)\}$$

*over the choices of $r$.*

## 3 Lower Bounds

*Lower Bound for $\ell_2/\ell_2$-Foreach Sparse Recovery with Low Risk.* For the sake of completeness we present a simplified version of the proof of the $\Omega(N)$ lower bound from [5] for the $\ell_2/\ell_2$ forall sparse recovery in [9]. Our main result is the following, whose proof can be found in [9].

**Theorem 4.** *Let $C \geqslant 1$ and $p$ be such that $\sqrt{12 + 16C^2} \cdot e^{-\frac{\ln(6+8C^2)}{2} \cdot N} \leqslant p < 1$. Then, any $\ell_2/\ell_2$ foreach sparse recovery scheme using $m \times N$ measurement matrices $\Phi$ with failure probability at most $p$ and approximation factor $C$ must have $m \geqslant \frac{1}{(6+8C^2)\ln(6+8C^2)} \ln\left(\frac{\sqrt{12+16C^2}}{p}\right) = \Omega(\log(1/p))$ measurements.*

*Lower Bound for Bounded Adversary Model.* In this section, we show the following result (proof is in [9]):

**Theorem 5.** *Any $\ell_2/\ell_2$ sparse recovery scheme that uses at most $b$ bits in each entry of $\Phi$ needs at least $\Omega\left(\sqrt{\frac{s}{b}}\right)$ number of measurements to be successful against an $s$-information-theoretically-bounded adversary.*

## 4 Sublinear Decoding

We present known results with polynomial time decoding on $\ell_2/\ell_2$ sparse recovery problem in [9].

Our strategy for designing sub-linear time decodable top level systems will be as follows: we will first design weak identification matrices that have sublinear identification time. Then we (in a black-box manner) convert such matrices to

sub-linear time decodable top level systems. We now present an outline of how we implement our strategy. In [9] we show how expanders can be used to construct various schemes that will be useful later. In [9], we show how to convert weak identification systems to top level systems. The rest of technical development is in designing weak identification system with good parameters.

Our first main result on sub-linear time decodable top levels systems will be:

**Theorem 6.** *For any $k \geqslant N^{\Omega(1)}$ and $\varepsilon, \alpha > 0$, there exists a $(k, 1+\varepsilon)$-top level system with $O(\varepsilon^{-11} k \log(N/k))$ measurements, failure probability $(N/k)^{-k/\log^{13+\alpha} k}$ and decoding time $\varepsilon^{-4} \cdot k^{1+\alpha} \cdot \log^{O(1)} N$.*[5] *This scheme uses $O_\varepsilon(k \cdot \log^{O(1)} N)$ bits of space.*

In fact, our results also work for $k = N^{o(1)}$ but we then do not get the optimal number of measurements. However, an increase in the decoding time leads to our second main result, which has near-optimal number of measurements.

**Theorem 7.** *For any $1 \leqslant k \leqslant N$ and $\varepsilon, \alpha > 0$, there exists a $(k, 1+\varepsilon)$-top level system with $O(\varepsilon^{-11} k \log(N/k) \log_k^\alpha N)$ measurements, failure probability $(N/k)^{-k/\log^{13+\alpha} k}$ and decoding time $(k/\varepsilon)^{\Theta(2^{-\alpha})} \cdot \log^{O(1)} N$.*[6] *This scheme uses $O_\varepsilon(k \cdot \log^{O(1)} N)$ bits of space.*

The proofs are deferred to [9].

*Consequences for the Bounded Adversary Model.* Our first corollary is an upper bound for the information-theoretic bounded adversary and follows directly from Lemma 1 (by setting $\beta = \alpha 2^{-s/\alpha}$) and the result of [5].

**Corollary 8.** *Fix $0 < \alpha < 1$. There is a randomized sparse signal recovery algorithm that with $m = O(k \log(N/k) + s/\alpha)$ measurements will foil an $s$-information-theoretically bounded adversary; that is, the algorithm's output will meet the $\ell_2/\ell_2$ error guarantees with probability $1 - \alpha$.*

The algorithm in [5] does not have a sublinear running time. If the goal is to defeat such an adversary and to do so with a sublinear algorithm, we must adjust our measurements accordingly, using Table 1. We note that in [6], there was a single result for $O(\log N)$-information-theoretically bounded adversaries ($O(k \log(N/k)$ measurements are sufficient) and this corollary provides an upper bound for the entire range of parameter $s$.

## References

1. Baraniuk, R.G., Candes, E., Nowak, R., Vetterli, M.: Compressive sampling. IEEE Signal Processing Magazine 25(2) (2008)

[5] The $O(\cdot)$ notation here hides the dependence on $\alpha$.
[6] The $O(\cdot)$ notation here hides the dependence on $\alpha$.

2. Candès, E.J., Tao, T.: Near-Optimal Signal Recovery From Random Projections: Universal Encoding Strategies? IEEE Transactions on Information Theory 52(12), 5406–5425 (2006)
3. Charikar, M., Chen, K., Farach-Colton, M.: Finding frequent items in data streams. In: Widmayer, P., Triguero, F., Morales, R., Hennessy, M., Eidenbenz, S., Conejo, R. (eds.) ICALP 2002. LNCS, vol. 2380, pp. 693–703. Springer, Heidelberg (2002)
4. Cohen, A., Dahmen, W., De Vore, R.A.: Near Optimal Approximation of Arbitrary Vectors from Highly Incomplete Measurements. Bericht. Inst. für Geometrie und Praktische Mathematik (2007)
5. Cohen, A., Dahmen, W., DeVore, R.: Compressed sensing and best $k$-term approximation. J. Amer. Math. Soc. 22(1), 211–231 (2009)
6. Gilbert, A.C., Hemenway, B., Rudra, A., Strauss, M.J., Wootters, M.: Recovering simple signals. In: ITA, pp. 382–391 (2012)
7. Gilbert, A.C., Indyk, P.: Sparse recovery using sparse matrices. Proceedings of the IEEE 98(6), 937–947 (2010)
8. Gilbert, A.C., Li, Y., Porat, E., Strauss, M.J.: Approximate sparse recovery: Optimizing time and measurements. SIAM J. Comput. 41(2), 436–453 (2012)
9. Gilbert, A.C., Ngo, H., Porat, E., Rudra, A., Strauss, M.J.: L2/L2-foreach sparse recovery with low risk. ArXiv e-prints, arXiv:1304.6232 (April 2013)
10. Guruswami, V., Rudra, A.: Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. IEEE Transactions on Information Theory 54(1), 135–150 (2008)
11. Guruswami, V., Sudan, M.: Improved decoding of reed-solomon and algebraic-geometry codes. IEEE Transactions on Information Theory 45(6), 1757–1767 (1999)
12. Indyk, P., Ruzic, M.: Near-optimal sparse recovery in the l1 norm. In: FOCS, pp. 199–207 (2008)
13. Irony, D., Toledo, S., Tiskin, A.: Communication lower bounds for distributed-memory matrix multiplication. J. Parallel Distrib. Comput. 64(9), 1017–1026 (2004)
14. Lapidoth, A., Narayan, P.: Reliable communication under channel uncertainty. IEEE Transactions on Information Theory 44, 2148–2177 (1998)
15. Lehman, A.R., Lehman, E.: Network coding: does the model need tuning? In: SODA, pp. 499–504 (2005)
16. Lipton, R.J.: A new approach to information theory. In: Enjalbert, P., Mayr, E.W., Wagner, K.W. (eds.) STACS 1994. LNCS, vol. 775, pp. 699–708. Springer, Heidelberg (1994)
17. Loomis, L.H., Whitney, H.: An inequality related to the isoperimetric inequality. Bull. Amer. Math. Soc. 55, 961–962 (1949)
18. Ngo, H.Q., Porat, E., Ré, C., Rudra, A.: Worst-case optimal join algorithms. In: PODS, pp. 37–48 (2012)
19. Ngo, H.Q., Porat, E., Rudra, A.: Efficiently decodable compressed sensing by list-recoverable codes and recursion. In: STACS, pp. 230–241 (2012)
20. Porat, E., Strauss, M.J.: Sublinear time, measurement-optimal, sparse recovery for all. In: SODA, pp. 1215–1227 (2012)
21. Price, E., Woodruff, D.P.: $(1 + \varepsilon)$-approximate sparse recovery. In: FOCS, pp. 295–304 (2011)
22. Rudra, A.: List Decoding and Property Testing of Error Correcting Codes. PhD thesis, University of Washington (2007)
23. Tishby, N., Pereira, F.C., Bialek, W.: The information bottleneck method. In: The 37th Annual Allerton Conference on Communication, Control, and Computing, pp. 368–377 (1999)