Abstract—It was shown in [1] that a vector $x \in \mathbb{R}^n$ with at most $k < n$ nonzeros can be recovered from an expander sketch $Ax$ in $O(n \log n)$ operations via the Parallel-$\ell_0$ decoding algorithm, where $n \log n$ denotes the number of nonzero entries in $A \in \mathbb{R}^{m \times n}$. In this paper we present the Robust-$\ell_0$ decoding algorithm, which robustifies Parallel-$\ell_0$ when the sketch $Ax$ is corrupted by additive noise. This robustness is achieved by approximating the asymptotic posterior distribution of values in the sketch given its corrupted measurements. We provide analytic expressions that approximate these posteriors under the assumptions that the nonzero entries in the signal and the noise are drawn from continuous distributions. Numerical experiments presented show that Robust-$\ell_0$ is superior to existing greedy and combinatorial compressed sensing algorithms in the presence of small to moderate signal-to-noise ratios in the setting of Gaussian signals and Gaussian additive noise.

I. INTRODUCTION

COMPRESSED sensing is a well studied method by which a sparse or compressible vector can be acquired by a number of measurements proportional to the number of its (sub-)Gaussian entries or randomly subsampled Fourier matrices. For a more detailed introduction to compressed sensing see [15].

Here we extend an algorithm proposed in [1] which can be used to recover exactly a sparse signal from its expander sketch (see Section II for details). Specifically, [1] proposed Parallel-$\ell_0$ (Algorithm 2), for noiseless combinatorial compressed sensing which is guaranteed to converge in $O(n \log n)$ where the sensing matrix $A$ is an expander matrix (Definition 1.1) and the signal $x \in \chi_k^n$ is dissociated in the sense of Definition 1.2 or the signal is drawn independently of $A$. For alternative combinatorial compressed sensing algorithms see, for example, [10–14]. We borrow notation from combinatorics and use the shorthands $[n] := \{1, \ldots, n\}$, $[n]^k := \{S \subset [n] : |S| = k\}$ where $|S|$ denotes the cardinality of the set $S$, and $[n]^{\leq k} := \cup_{k \leq k}[n]^k$ for $n \in \mathbb{N}$ and $k < n$.

Definition 1.1 (Expander matrices [1]). The matrix $A \in \{0, 1\}^{m \times n}$ is a $(k, \varepsilon, d)$-expander matrix if $\sum_{i=1}^{m} \mathbb{I}_{|A_{i,j}|>0} = d$ for all $j \in [n]$ and

$$\left| \left\{ i \in [m] : \sum_{j \in S} \mathbb{I}_{|A_{i,j}|>0} \right\} \right| > (1 - \varepsilon)d|S|$$

for all $S \subset [n]^{\leq k}$. We denote by $\mathbb{E}^{m \times n}$ the set of $(k, \varepsilon, d)$-expander matrices of dimension $m \times n$.

Definition 1.2 (Dissociated signals [1]). A signal $x \in \mathbb{R}^n$ said to be dissociated if

$$\sum_{j \in S_1} x_j \neq \sum_{j \in S_2} x_j \quad \forall S_1, S_2 \subset \text{supp}(x) \text{ s.t. } S_1 \neq S_2.$$  

An example of (almost surely) dissociated signals are those drawn from a continuous distribution. It is shown in [1] that if $y = Ax$ is an expander sketch and $x \in \chi_k^n$ is dissociated, then there exists a subset $T \subset [n]$ such that, for each $j \in T$, $\{i \in [m] : y_i = y_j\}$ is bounded below by a positive constant depending on $d$ and $\varepsilon$. This guarantees that if $\{i \in \text{supp}(a_j) : y_i = y_j\}$ $> d/2$ then $x_j = y_j$. Parallel-$\ell_0$ (Algorithm 2) implements this observation by letting $\hat{x} = 0$ and estimating the decrease in $\|y\|_0$ when performing the update $\hat{x}_j \leftarrow \hat{x}_j + y_j$. We denote for $j \in [n]$ its neighbours by $N(j) := \{i \in [m] : |A_{i,j}| > 0\}$; to estimate the decrease in $\|y\|_0$. Parallel-$\ell_0$ computes

$$n_c \leftarrow |\{ \ell \in N(j) : y_\ell = y_j \}|,$$

$$n_z \leftarrow |\{ \ell \in N(j) : y_\ell = 0 \}|.$$
We extend their approach to the additive noise signal model of \( \hat{y} = y + \eta \) with \( y = Ax \) and \( \eta \in \mathbb{R}^m \) by replacing (1)-(2) with scores estimating the distribution of \( n_e \) and \( n_z \), e.g., (8)-(9).

That is, we follow a Bayesian approach to the computation of these scores and estimate:

1) The probability of \( y_i = 0 \) given that we observe \( \hat{y}_i \).
   \[
   p_z (\omega) := \mathbb{P} (y_i = 0 \mid \hat{y}_i = \omega) .
   \]

2) The probability of \( y_i = y_{i2} \) given that we observe \( \hat{y}_i - \hat{y}_{i2} \).
   \[
   p_e (\omega) := \mathbb{P} (y_i = y_{i2} \mid \hat{y}_i - \hat{y}_{i2} = \omega)
   \]

Among our contributions are series approximations for (3)-(4) when the signals and measurements are generated according to the generating model given in Definition I.3 and illustrated in Figure [I]. In what follows, we let \( \mathbb{D}(\mathbb{R}) \) be the set of distributions supported on \( \mathbb{R} \). If \( \mu \in \mathbb{D}(\mathbb{R}) \), we write \( z \sim \mu \) to denote that \( z \) was drawn according to the distribution \( \mu \).

We also use the notation \( v_i \sim \mu \) to denote that each \( v_i \) is drawn independently at random from \( \mu \). Finally, we use \( U(S) \) to denote the uniform distribution over a set \( S \).

Definition I.3 (Generating model GM\((n, m, k, d, \mu, \nu)\)). Let \( n, m, k, d \in \mathbb{N} \) be such that \( k \leq m < n \) and \( d \ll m \). Let \( \mu, \nu \in \mathbb{D}(\mathbb{R}) \). Then, the problem \((A, \hat{y})\) is drawn from the model GM\((n, m, k, d, \mu, \nu)\) if \( A \in \{0, 1\}^{m \times n} \) and \( \hat{y} \in \mathbb{R}^m \) are such that

1) each column of \( A \) has a support drawn uniformly at random from \([m]^{(d)}\);
2) \( \text{supp}(x) \) is drawn uniformly at random from \([n]^{(k)}\);
3) \( x_j \sim \mu \) for each \( j \in \text{supp}(x) \);
4) \( \eta_i \sim \nu \) for each \( i \in [m] \);
5) \( \eta \) is independent of \( x \); and
6) \( y = Ax \) and \( \hat{y} = y + \eta \).

We write \((A, \hat{y}) \sim \text{GM}(n, m, k, d, \mu, \nu)\) to denote problem instances drawn from this signal model.

Remark. It is important to note that a matrix \( A \in \mathbb{R}^{m \times n} \) generated under the model presented in Definition I.3 and Figure I is a \((k, \varepsilon, d)\)-expander matrix with high probability, see [22], [15] Theorem 13.6.

Moreover, the generating model in Definition I.3 also allows us to define robust estimates for (3)-(4) for general noise and signal distributions and to any degree of accuracy under the assumption that these probability measures are available.

From there we can define noisy analogues to the values \( n_e \) and \( n_z \) in (1)-(2) used in Parallel-\( \ell_0 \) but which are robust to additive noise. The contributions of this work are two-fold: (i) to present principled ways to compute (3)-(4) in the case where the nonzeros in \( \eta \) and \( x \) are drawn from continuous probability distributions; (ii) to provide a variation of Parallel-\( \ell_0 \) that is robust to noise. While other similar generating models can be considered using the techniques presented here, for ease of exposition and clarity, we restrict our description to this model.

Theorem I.4 (Probabilities for general signal and noise distributions). Let \( \delta, \rho \in (0, 1) \).

For each \( n > 1 \), let \( m = \delta n \),

\[ k = \rho m \]

and \( d < m \). If \( \mu, \nu \in \mathbb{D}(\mathbb{R}) \) and \((A, \hat{y}) \sim \text{GM}(n, m, k, d, \mu, \nu)\). Then as \( n \to \infty \),

\[
\begin{align*}
  p_z (\omega) \to & \frac{\nu(\omega)}{\sum_{q \geq 0} \frac{(d\rho)^q}{q!} (\nu * \mu_q)(\omega)}, \\
  p_e (\omega) \to & \frac{\bar{\nu}(\omega)}{\sum_{q \geq 0} \frac{(d\rho)^q}{q!} (\bar{\nu} * \bar{\mu}_q)(\omega)}.
\end{align*}
\]

Where \( \mu_q, \bar{\mu}_q, \bar{\nu} \) are probability measures constructed as in Definition I.1.

Equations (5) and (6) allows us to quantify the uncertainty associated with computing the score for Parallel-\( \ell_0 \) under the presence of additive noise. Note that equations (5) and (6) can be easily adapted to alternative generative models, such as where the expected density of nonzeros per row varies, but for expository clarity we restrict our discussion to this somewhat generic model. It will be discussed in Section III-C that equations (5) and (6) should not be used directly, but instead be scaled by considering normalised functions \( \bar{p}_z \) and \( \bar{p}_e \) defined as,

\[
\bar{p}_e (\omega) = \frac{p_e (\omega)}{\max_s p_e (s)}, \quad \bar{p}_z (\omega) = \frac{p_z (\omega)}{\max_s p_z (s)}.
\]

In the most general case \( n_e \) and \( n_z \) can be written as the sum of individual scores \( q_e \) and \( q_z \) as follows

\[
\begin{align*}
  n_e & \leftarrow \sum_{i \in N(j)} q_e (r_{i1} - r_{i2} \mid t) \quad (8) \\
  n_z & \leftarrow \sum_{i \in N(j)} q_z (r_i \mid t) \quad (9)
\end{align*}
\]
A confidence threshold \( t > 0 \) needs to be given for some variants of our algorithm, so to simplify the exposition we include this parameter in the scores \( q_e(\cdot \mid t) \) and \( q_z(\cdot \mid t) \) regardless of whether it is used or not. A summary of the score functions are given in Table I.

|                | Robust-\( \ell_0 \) | Robust-\( \ell_0 \) | Parallel-\( \ell_0 \) |
|----------------|---------------------|---------------------|----------------------|
| \( q_e(r_{i1} - r_{i2} \mid t) \) & \( \tilde{p}_e(r_{i1} - r_{i2}) \cdot \mathbb{1}_{\{\tilde{p}_e(r_{i1} - r_{i2}) \geq t\}} \) & \( \mathbb{1}_{\{r_{i1} = r_{i2}\}} \) |
| \( q_z(r_i \mid t) \) & \( \tilde{p}_z(r_i) \cdot \mathbb{1}_{\{\tilde{p}_z(r_i) \geq 1 - t\}} \) & \( \mathbb{1}_{\{r_i = 0\}} \)

**Table I:** Extensions of scores used in Expander \( \ell_0 \)-decoding to identify candidate updates to the sparse signal \( \hat{x} \).

A. Outline of the manuscript

The structure of this paper is as follows: Section II comprises a review of combinatorial compressed sensing and the Parallel-\( \ell_0 \) decoding algorithm extended here. Robust-\( \ell_0 \) decoding and the associated scores (8)-(9) are presented in Section III. In Section IV we present numerical experiments which demonstrate Robust-\( \ell_0 \) to perform superior to a number of leading greedy and combinatorial compressed sensing algorithms.

II. BACKGROUND: COMBINATORIAL COMPRESSED SENSING AND \( \ell_0 \)-DECODING

As mentioned in the previous section, the branch of combinatorial compressed sensing measures \( x \in \mathbb{R}^n \) with the adjacency matrix of an expander graph. These matrices are of very low complexity in terms of generation and storage, and also promise faster encoding and decoding than their dense counterparts, see Theorem II.2. In this section we review the basic elements of expander graphs and combinatorial compressed sensing.

There have been various algorithms proposed to reconstruct a sparse vector \( x \) from measurements \( y = Ax \) when \( A \) is an expander matrix, see [17], [18], [20] and [19]; the work presented here starts with the Parallel-\( \ell_0 \) algorithm and to improve this algorithm by making it robust to noisy measurements, results in Robust-\( \ell_0 \) (Algorithm I). The key observation for the Parallel-\( \ell_0 \) algorithm is given by the following Lemma.

**Lemma II.1.** Let \( y = Ax \), \( x \) dissociated, \( A \in \mathbb{R}^{k \times n} \) with \( \varepsilon < \frac{1}{4} \). Then there exists a nonempty set \( T \subset [n] \times \mathbb{R} \) such that

\[
|\{i \in \mathcal{N}(j) : y_i = \omega\}| \geq (1 - 2\varepsilon)d \quad \forall (j, \omega) \in T
\]

and for every tuple in \( T \) that satisfies this property, we have \( w = x_j \).

This means that at each iteration, if the residual \( r \) is non-zero, i.e. if we have not yet found the correct \( x \), then there is a set of entries in \( x \) that we can change so that we reduce the number of non-zeros in \( r \) by at least \( |T|(1 - 2\varepsilon)d \).

**Theorem II.2 (Convergence of Algorithm II.1)**. Let \( A \in \mathbb{R}^{k \times n} \) and let \( \varepsilon \leq \frac{1}{4} \), and \( x \in \mathbb{R}^n \) be dissociated. Then, Parallel-\( \ell_0 \) with \( \alpha = (1 - 2\varepsilon)d \) can recover \( x \) from \( y = Ax \) in \( O(\log k) \) iterations of complexity \( O(dn) \).

To put this result into context and show its applicability, we recall the remark after Definition I.3 stating that random matrices as considered in this work are indeed expander matrices with high probability.
We furthermore emphasize the fact that the algorithm is designed in a way that allows for massively parallel implementations.

III. MAIN CONTRIBUTIONS: $\ell_0$-DECODING FOR NOISY MEASUREMENTS

We now consider the case where the measurements $y$ are subject to additive noise, i.e., instead of $y = Ax$, we measure $\hat{y} = y + \eta$ where $\eta$ is a realization of a random variable with $\eta_i \sim \nu$.

Parallel-$\ell_0$ is not able to cope with additive noise, as it needs to make decisions whether a value in $\hat{y}$ is zero and whether two values in $\hat{y}$ are equal to each other. While for very small noise levels we could consider two values as equal if they are within a certain number of standard deviations, for larger noise levels the decision becomes more challenging. A discussed in Section I we need to know $p_z(\hat{y}_i)$ and $p_z(\hat{y}_1 - \hat{y}_2)$ which correspond, respectively, to the probability of $y_i = 0$ given that we observe $\hat{y}_i$ and the probability of $y_i = y_2$ given that we observe $\hat{y}_1 - \hat{y}_2$. The functions $p_z$ and $p_n$ depend on the parameters fed into the generative model in Definition [1] and in particular on the distribution of $y$ and $\hat{y}$. Hence, since $y$ is a vector of sparse inner products we need to understand the limiting behaviour of sparse sums.

We include Definition [III.1] in order to remind the reader of some actions on measures used in this manuscript as so as to be relatively self contained.

Definition III.1 (Measures [23]). Let $\mathcal{B}(\mathbb{R})$ denote the Borel $\sigma$-algebra over $\mathbb{R}$. If $E \in \mathcal{B}(\mathbb{R})$ let $-E := \{-x : x \in E\}$. Let $\mu \in \mathcal{D}(\mathbb{R})$, we define the following measures.

1) The $q$-convolution,
$$
\nu_0(E) = 0, \quad \forall E \in \mathcal{B}(\mathbb{R})
$$
$$
\nu_1(E) = \mu(E), \quad \forall E \in \mathcal{B}(\mathbb{R})
$$
$$
\nu_{q+1}(E) = (\mu * \mu)(E), \quad \forall E \in \mathcal{B}(\mathbb{R}), q \in \mathbb{N}
$$

2) The negative measure,
$$
\nu^{-}(E) = \mu(-E), \quad \forall E \in \mathcal{B}(\mathbb{R})
$$

3) The symmetrized measure,
$$
\mu(E) = \frac{\mu(E) + \mu(-E)}{2}, \quad \forall E \in \mathcal{B}(\mathbb{R})
$$

4) The measure associated with the difference of two random variables,
$$
\nu(E) = (\mu * \mu)(E), \quad \forall E \in \mathcal{B}(\mathbb{R})
$$

Lemma III.2 (Limiting distribution for sparse sums of random variables). Let $p \in (0,1)$, let $\mu \in \mathcal{D}(\mathbb{R})$ and let $\mu_{q} \in \mathcal{D}(\mathbb{R})$ be its the $q$-fold convolution. For each $n \geq 1$, let
$$
s_n := \sum_{j=1}^{n} p_j x_j
$$
be such that,

1) $x_j \overset{i.i.d.}{\sim} \mu$ for each $j \in [n]$.
2) $b_j \overset{i.i.d.}{\sim} \text{Ber}(\frac{p_n}{n})$ for each $j \in [n]$ with $p_n \to p \in \mathbb{R}$ as $n \to \infty$.

Then, as $n \to \infty$ it holds that $s_n \overset{(d)}{\rightarrow} s$ where
$$
s \sim \exp(-p) \sum_{q \geq 0} \frac{p^q}{q!} \mu_q.
$$

Proof. Let $\psi_{s_n}(t)$ be the characteristic function of $s_n$. Let $x \sim \mu$, then
$$
\psi_{s_n}(t) = \mathbb{E}[\exp(it s_n)]
$$
$$
= \prod_{j=1}^{n} \mathbb{E}[\exp(it b_j x_j)]
$$
$$
= \left( 1 - \frac{p_n}{n} + \frac{p_n}{n} \psi_x(t) \right)^n
$$
$$
= \left( 1 + \frac{p_n}{n} \psi_x(t) - 1 \right)^n.
$$

Taking the limit $n \to \infty$ we see that
$$
\lim_{n \to \infty} \psi_{s_n}(t) = \exp(-p) \exp(p \psi_x(t)).
$$

Letting $w_q = \sum_{j=1}^{q} x_j$ it holds by the independence of $\{x_1, \ldots, x_q\}$ that $w_q \sim \mu_q$ and
$$
[\psi_x(t)]^q = \psi_{w_q}(t).
$$

Now, consider a random variable $z$ distributed according to
$$
z \sim \exp(-p) \sum_{q \geq 0} \frac{p^q}{q!} \mu_q.
$$
The characteristic function of $z$ is given by

$$
\psi_z(t) = \mathbb{E} [\exp(itz)]
= \exp(-p) \sum_{q \geq 0} \frac{1}{q!} \psi_w(t)^q
= \exp(-p) \sum_{q \geq 0} \frac{1}{q!} (\psi_x(t))^q
= \exp(-p) \sum_{q \geq 0} \left( \frac{p \psi_x(t)}{2^q} \right)^q
= \exp(-p) \exp(p \psi_x(t)).
$$

Therefore (15) equals (12). By Lévy's continuity theorem, weak convergence of the characteristic functions implies pointwise convergence of the random variables (cf. [23] Theorem 15.23) and hence the statement follows.

**Theorem III.3** (Distribution of $\hat{y}_i$ and $\hat{y}_i - \hat{y}_{i_2}$). Fix $\delta, p \in (0,1)$ and for $n \in \mathbb{N}$ let $m = \delta n$, $k = \rho n$ and $d \ll m$. Furthermore, let $\mu$ and $\nu$ be measures and assume that $\hat{y} = y + \eta = Ax + \eta$ is drawn from the model $GM(n, m, k, d, \mu, \nu)$. Then as $n \to \infty$

$$
\hat{y}_i \xrightarrow{(d)} \hat{\mu}^* \quad \text{and} \quad \hat{y}_i - \hat{y}_{i_2} \xrightarrow{(d)} \hat{\nu}^* \quad \text{where}
$$

$$
\hat{\mu}^* \sim \exp(-d\rho) \sum_{q \geq 0} \frac{(dp)^q}{q!} \mu_q, \quad \hat{\nu}^* \sim \exp(-2d\rho) \sum_{q \geq 0} \frac{(2dp)^q}{q!} \nu_q.
$$

**Proof.** To show (16), let $i \in [m]$ and

$$
y_i = \sum_{j=1}^{n} A_{i,j} x_j.
$$

By our assumptions on $A$ and $x$,

$$
\mathbb{P}(A_{i,j} x_j \neq 0) = \mathbb{P}(A_{i,j} \neq 0 \land x_j \neq 0)
= \mathbb{P}(A_{i,j} \neq 0) \mathbb{P}(x_j \neq 0)
= \frac{d}{m} \frac{k}{n}
= \frac{dp}{n}
$$

where for two events $E_1$ and $E_2$ we let $E_1 \land E_2$ be the conjunction of the events. Note that if $A_{i,j} x_j \neq 0$, then $j \in \supp(x)$ so $A_{i,j} x_j = x_j \sim \mu$. Hence, letting $b_j \sim \text{Ber}\left(\frac{dp}{n}\right)$,

$$
y_i = \sum_{j=1}^{n} b_j x_j.
$$

Invoking Lemma III.2 with $p_n = p = dp$ we obtain that $y_i \to y_i^*$ as $n \to \infty$ where

$$
y_i^* \sim \exp(-d\rho) \sum_{q \geq 0} \frac{(dp)^q}{q!} \mu_q.
$$

By the independence of $y_i^*$ and $\eta_i$, since $\hat{y}_i = y_i + \eta_i$, the distribution of $\hat{y}_i^*$ is given by

$$
\hat{y}_i^* \sim \left( \exp(-d\rho) \sum_{q \geq 0} \frac{(dp)^q}{q!} \mu_q \right) \ast \nu.
$$

Equation (16) follows from (18) and the distributivity of the convolution operator.

To show (17), let $i_1, i_2 \in [m] \lor$ such that $i_1 \neq i_2$ and let

$$
y_{i_1} - y_{i_2} = \sum_{j=1}^{n} (A_{i_1,j} - A_{i_2,j}) x_j.
$$

Similarly to the previous case, we compute

$$
\mathbb{P}(\{A_{i_1,j} - A_{i_2,j}) x_j \neq 0\}) = \mathbb{P}(A_{i_1,j} - A_{i_2,j} \neq 0 \land x_j \neq 0)
= \mathbb{P}(A_{i_1,j} - A_{i_2,j} \neq 0) \mathbb{P}(x_j \neq 0)
= 2 \frac{d}{m} \frac{(m-1)-(d-1)k}{m-1} \frac{1}{n}
= \frac{2dp}{n} (1 - o(1))
$$

Note that if $(A_{i_1,j} - A_{i_2,j}) x_j \neq 0$, then $j \in \supp(x)$ and $(A_{i_1,j} - A_{i_2,j})$ is either $+1$ with probability $\frac{1}{2}$ or $-1$ with probability $\frac{1}{2}$. Then, Hence,

$$
(A_{i_1,j} - A_{i_2,j}) x_j \sim \begin{cases} 
\mu & \text{with probability } \frac{1}{2} \\
-\mu & \text{with probability } \frac{1}{2} 
\end{cases}
$$

then,

$$
(A_{i_1,j} - A_{i_2,j}) x_j \sim \mu.
$$

Letting $b_j \sim \text{Ber}\left(\frac{2dp}{n} (1 - o(1))\right)$,

$$
y_{i_1} - y_{i_2} \xrightarrow{(d)} \sum_{j=1}^{n} b_j x_j.
$$

Again, invoking Lemma III.2 with $p_n = 2dp(1 - o(1))$ we obtain that $p = 2dp$ and also that as $n \to \infty$, $y_{i_1} - y_{i_2} \to g^*$ with

$$
y_{i_1}^* - y_{i_2}^* \sim \exp(-2d\rho) \sum_{q \geq 0} \frac{(2dp)^q}{q!} \mu_q.
$$

Therefore, Given that $\eta_i - \eta_i \sim \nu \ast \nu^*$ and that

$$
\hat{y}_i - \hat{y}_{i_2} = (y_{i_1} - y_{i_2}) + (\eta_{i_1} - \eta_{i_2}),
$$

we convolve (19) with $\nu \ast \nu^*$ to recover (17).

We are now ready to prove Theorem I.4

**Proof.** Theorem I.4 Using Bayes rule we write

$$
\mathbb{P}(y_i = 0|\hat{y}_i = \omega) = \frac{\mathbb{P}(\hat{y}_i = \omega \land y_i = 0)}{\mathbb{P}(\hat{y}_i = \omega)}.
$$

From (16) we can deduce that

$$
\mathbb{P}(\hat{y}_i = \omega \land y_i = 0) = \exp(-d\rho)\nu(\omega)
$$

and using equation (16) from Theorem III.3 we obtain that as $n \to \infty$,

$$
\mathbb{P}(\hat{y}_i = \omega) \to \exp(-d\rho) \sum_{q \geq 0} \frac{(dp)^q}{q!} (\nu \ast \mu_q)(\omega).
$$
Let, 
\[ R_{\ell}(\ell) := \exp(d\rho) - \sum_{q=0}^{\ell} \frac{(d\rho)^q}{q!} \]
and write
\[ S_{a}(\omega) := \sum_{q=0}^{\ell} \frac{(d\rho)^q}{q!} \varphi(\omega | \sigma_{a}^2), \]
\[ S_{b}(\omega) := \sum_{q=\ell+1}^{\infty} \frac{(d\rho)^q}{q!} \varphi(\omega | \sigma_{b}^2). \]
Note that \( S_{b}/R_{\ell} \) satisfies the conditions of Lemma III.4 so it corresponds to the density function of a centred random variable with variance

\[ \sigma_z^2 = \frac{1}{R_{\ell}} \sum_{q=0}^{\ell} \frac{(d\rho)^q}{q!} (q\sigma_a^2 + \sigma_n^2) \]
\[ = \sigma_z^2 (d\rho) R_{\ell}(\ell - 1) + \sigma_z^2 R_{\ell}(\ell) \]

Therefore,
\[ p_z(\omega) \approx \frac{\varphi(\omega | \sigma_z^2)}{\sum_{q=0}^{\ell} (\frac{d\rho)^q}{q!} \varphi(\omega | \sigma_q^2) + R_{\ell}(\ell) \varphi(\omega | \sigma_z^2)}. \]

A similar argument shows that
\[ p_e(\omega) \approx \frac{\varphi(\omega | 2\sigma_z^2)}{\sum_{q=0}^{\ell} (\frac{d\rho)^q}{q!} \varphi(\omega | \sigma_{q,e}^2) + R_{e}(\ell) \varphi(\omega | \sigma_z^2)}. \]

Where \( \sigma_{q,e}^2 = q\sigma_z^2 + \sigma_{n}^2 \), and
\[ R_{e}(\ell) = \exp(2d\rho) - \sum_{q=0}^{\ell} \left( \frac{(d\rho)^q}{q!} \right), \]
\[ \sigma_{e}^2 = \frac{\sigma_z^2 (2d\rho) R_{\ell}(\ell - 1) + 2\sigma_n^2 R_{e}(\ell)}{R_{e}(\ell)}. \]

B. Comparison with empirical probabilities

We test the approximations given in (28) and (29) by randomly generating \( \hat{y} \) and \( y \) according to the generating models \( GM(\alpha, \delta n, \rho\delta n, 7, N(0, 1), N(0, \sigma_z^2)) \), for \( \delta = 0.3, \rho \in \{0.1, 0.3\} \) and \( \sigma \in \{10^{-3}, 10^{-2}\} \). The results can be seen in Figure 4.

Overall the analytical expressions fit the empirical probabilities very well, indicating that the approximations we made in the calculations above are justified. However, we observe that as \( \rho \) and especially \( \sigma \) increase, both functions drop significantly. This means that for large values of these parameters, the noise eventually dominates and it is difficult to decide whether values are zero or equal.

C. Scaled probabilities \( \tilde{p}_z \) and \( \tilde{p}_e \)

We mentioned in Section 4 that our algorithms do not implement the functions \( p_z \) and \( p_e \) exactly, but a scaled version of these. As observed in Figure 4 the value of \( \max_z p_z(s) \) and \( \max_e p_e(s) \) varies significantly as \( \sigma \) and \( \rho \) change. Algorithm 1 evaluates whether a given score is large or not by implementing a sweeping parameter \( t \) that is set
we begin by describing the stopping conditions and measures used to denote successful recovery in the presence of noise in Section IV-A, along with how the parameter $c$ is varied in Section IV-B. We then present the main numerical results in Section IV-C where the algorithms phase transitions and runtime are presented, along with Sections IV-D and IV-E which show further details on Robust-$\ell_0$ decoding’s performance as a function of noise variance and for extreme subsampling respectively.

### A. Stopping conditions

We are interested in the signal model $y = Ax + \eta$ where $\eta \sim \mathcal{N}(0, \sigma^2 I_{m \times m})$. If $\hat{x}$ is an approximation to $x$ the residual is $r = y - A\hat{x}$. Note that if $\hat{x} = x$, then

$$\|r\|_1 = \|y - A\hat{x}\|_1$$

and we should not seek reductions in the residual below $\|\eta\|_1$, which would result in fitting to the additive noise. We further account for the variance of $\|\eta\|_1$ and denote the algorithm to have successfully recovered $x$ if $\hat{x}$ satisfies

$$\frac{\|x - \hat{x}\|_1}{\|x\|_1} \leq \frac{E[\|\eta\|_1] + C_1 \sqrt{\text{Var}[\|\eta\|_1]}}{\|x\|_1}$$

for some $C_1 \geq 0$. We should be aware that the right hand side of (30) might be greater than 1 for some choices of $k$, $m$ and $\sigma$. When this happens, the stopping condition (30) becomes invalid since we expect $\hat{x}$ to have captured a proportion of the $\ell_1$-energy of $\|x\|_1$. Hence, if the right hand side of (30) is greater than $\frac{1}{10}$, we clip the upper bound at this value and use the stopping condition

$$\frac{\|x - \hat{x}\|_1}{\|x\|_1} \leq \min \left( \frac{E[\|\eta\|_1] + C_1 \sqrt{\text{Var}[\|\eta\|_1]}}{\|x\|_1}, \frac{1}{10} \right).$$

For the numerical experiments conducted in this section we consider nonzero entries in $x$ drawn as $x_i \sim \mathcal{N}(0, 1)$, and noise $\eta_i \sim \mathcal{N}(0, \sigma^2)$ for which $E[\|x\|_1] = k\sqrt{\frac{2}{\pi}}$ and $E[\|\eta\|_1] = m\sigma\sqrt{\frac{2}{\pi}}$ and moreover $\text{Var}[\|\eta\|_1] = m\sigma^2 (1 - \frac{2}{\pi})$, see e.g. (24).

### B. Selection of parameter $c$ in Algorithm 2

The sweeping parameter $t$ in Algorithm 2 is initialised at 1 and updated by decreasing it by a constant value $c$. We observed in our experiments that the quality of the phase transitions are sensitive on the parameter $c$ especially for low $\delta$ and $\rho$. We do not provide a way to fine-tune $c$, but we run our phase transitions with the following choices:

1) If the algorithm is quantised,

$$c = \begin{cases} 
0.01 & \delta \leq 0.05 \\
0.05 & \delta > 0.05 \text{ and } \rho \in (0, 0.1] \\
0.075 & \delta > 0.05 \text{ and } \rho \in (0.1, 0.2] \\
0.1 & \delta > 0.05 \text{ and } \rho \in (0.2, 1]
\end{cases}.$$
2) If the algorithm is continuous,  

\[ c = \begin{cases} 
0.01 & \delta \leq 0.05 \\
0.025 & \delta > 0.05 
\end{cases} \]  

(33)

The values in (32) and (33) were chosen heuristically for \( \nu \) and \( \mu \) Gaussian.

### C. Phase transitions and runtime

We benchmark the variants of Robust-\( \ell_0 \) against other greedy algorithms via their phase-transitions and runtime. The user can supply two binary flags, adaptive_k and quantised which yield four different variants of Robust-\( \ell_0 \). We assigned a unique label to each of these variants as described in Table II.

| Algorithm                      | adaptive_k | quantised |
|-------------------------------|------------|-----------|
| Robust-\( \ell_0 \)           | No         | No        |
| Robust-\( \ell_0 \)-adaptive  | Yes        | No        |
| Robust-\( \ell_0 \)-quantised | No         | Yes       |
| Robust-\( \ell_0 \)-adaptive-quantised | Yes       | Yes |

**TABLE II: Variants of Robust-\( \ell_0 \)**

The phase transition of a compressed-sensing algorithm [24] is the largest value of \( k/n \) for which the algorithm is typically able recovery all \( k \) sparse vectors with sparsity less than \( k \) for a fixed \( m/n \). Hence, for a fixed value of \( \delta = m/n \) the phase transition of an algorithm is the largest value \( \rho^*(\delta) \) for which the algorithm converges for all \( \rho(\delta) < \rho^*(\delta) \). The value \( \rho^*(m/n) \) often converges to a fixed value as \( n \to \infty \), so phase transitions often partition the \( \delta \times \rho \) space into two regions: One in which the algorithm converges with high probability and another in which the algorithm doesn’t converge with high probability. We benchmark Robust-\( \ell_0 \) against the algorithms presented in [18], [19], [26]. Specifically, our tests include the following algorithms,

\{Robust-\( \ell_0 \), Robust-\( \ell_0 \)-adaptive, Robust-\( \ell_0 \)-quantised, Robust-\( \ell_0 \)-adaptive-quantised, SSMP, SMP, CGIHT\}.  

In the deterministic case Parallel-\( \ell_0 \) was compared against a range of combinatorial compressed sensing algorithms in [1]; out of those we have selected SSMP and SMP as these perform best and are similar in nature to Robust-\( \ell_0 \). We also compare with CGIHT, as this algorithm was shown to be the
fastest among the greedy algorithms compared in [13], [26]. Figures [3a] [3e] and [3i] show the phase transition curves for these algorithms with $\sigma = 10^{-3}, 10^{-2}, 10^{-1}$ respectively. The curves were computed by setting $n = 2^{18}, d = 7$, and using the stopping condition $\|\hat{\rho}\|_1 \leq E[\|\eta\|_1] = ma \sqrt{\frac{2}{\pi}}$ and a success condition $\mathbf{[30]}$ with $C_1 = 1$. The testing is done at $m = \delta_p n$ for

$$\delta_p \in \{0.02p : p \in [4]\} \cup \left\{0.1 + \frac{89}{1900}(p - 1) : p \in [20]\right\}.$$ 

For each $\delta_p$, we set $\rho = 0.01$ and generate 10 synthetic problems to apply the algorithms to, with a problem generated as in GM with the given parameters and $\mu$ and $\nu$ being normal Gaussian. If at least one such problem was recovered successfully, the sparsity ratio $\rho$ is increased by 0.01 and the experiment is repeated. Following the testing framework in [27], the recovery data is fitted using a logistic function and finally the 50% recovery transition function is computed and presented in the phase transition plots contained herein. Figures [3b] [3i] and [3l] show a selection map for these algorithms. Namely, these plots indicate which algorithm requires the least computational time[1] at each point in the $\delta \times \rho$ space where the algorithm converges. Finally, Figures [4a] [4e] and [4i] show the total time for convergence in milliseconds for the fastest algorithm at each point in the $\delta \times \rho$ space. The ratio of the time for the second fastest algorithm over the time for the fastest algorithm is given in Figures [5a] [5e] and [5i].

We can see from Figure [3] that CGIHT [26] dominates the upper region of the phase transition space, while the Robust-$\ell_0$ algorithms only converge for $\rho \lesssim 0.3$ which is consistent with the observed phase transitions for Parallel-$\ell_0$ [1]. In terms of speed, Robust-$\ell_0$ seems to be the most competitive for $\sigma \in \{10^{-3}, 10^{-2}\}$ and $\rho \lesssim 0.2$. However, for large noise, $\sigma = 10^{-1}$, CGIHT becomes the fastest algorithm of all. We remark that while CGIHT performs very well in our numerical tests, the current theory developed for it does not hold in the setting considered here, as it requires zero-mean columns in $A$.

Figure [4] shows the widths for the Robust-$\ell_0$ algorithms. The widths measure how sharp the phase transition of an algorithm is; namely, how thin the boundary between the region of recovery with high-probability and the region of recovery where combinatorial search is needed. It has been shown that the widths of a compressed sensing algorithm tend to zero as $n \to \infty$ when decoding with linear programming [29], and we usually expect the same behaviour for other algorithms [26]. Figure [4] show that the widths for the Robust-$\ell_0$ algorithms indeed decrease with $n$ and with $\delta$. The observed smoothness of the phase transition widths signal also suggest that the stopping conditions of the algorithm are consistent for the problem under consideration.

D. Dependence on noise variance, $\sigma$

We now investigate the extent to which the phase transitions of the algorithm decrease as we increase the noise level $\sigma$. To do this, we consider $\delta \in \{0.01, 0.02, 0.1, 0.2\}$ and for each

1All the numerical results presented here were performed using a Linux machine with Intel Xeon E5-2643 CPUs 3.30 GHz, NVIDIA Tesla K10 GPUs and executed from Matlab R2016b. The code was added to the GAGA library available at http://www.gagals.org and described in [28], so as to facilitate large scale benchmarking against the other algorithms presented here which are also contained in the aforementioned library.
value of $\delta$ we compute we define the grid

$$
\sigma \in \{10^{-3+i} : i \in \{0\} \cup \{20\}\}.
$$

Then at each value of $\sigma$ we let $\rho = 0.01$ and draw ten problem instances from GM with signal distribution $\mathcal{N}(0, 1)$ and noise distribution $\mathcal{N}(0, \sigma^2)$. If at least one of the problems was recovered successfully, then we set $\rho \leftarrow \rho + 0.01$ and repeat the experiment. We do this procedure for each of the algorithms considered and record the largest $\rho$ having at least 50% success rate. The results are shown in Figure 5. In order to show where the clipping in (31) becomes active, the figures also show a TOL-curve which partitions the space into the region where the right hand side of (31) equals $\frac{1}{10}$ (bottom region) and the region where it equals the right hand side of (30) (top region). We can appreciate from Figure 5 that for small $\delta$ both Robust-$\ell_0$ and SSMP have the best recovery capabilities, with Robust-$\ell_0$ being preferable for noise levels $\sigma \lesssim 10^{-2}$ and SSMP being better suited for larger noise levels. For larger $\delta$, CGHT is preferable except for very low noise levels.

E. Phase transitions for extreme subsampling, $\delta \ll 1$

The numerical experiments of Parallel-$\ell_0$ in [1] showed flat phase transitions; that is, it was observed that $\rho^*(\delta)$ remained approximately 0.3 as $\delta \to 0$ provided $n$ was sufficiently large. While Robust-$\ell_0$ does not exhibit precisely the same behaviour in the presence of noise, we do observe that $\rho^*(\delta)$ remains nontrivial even for $\delta$ as small as $10^{-3}$, again provided $n$ is sufficiently large. We provide numerical evidence for this in Figure 6. For each $(\delta, \sigma) \in \{0.001, 0.01\} \times \{0.001, 0.01\}$ we let $\rho = 0.01$ and solve ten problems drawn from GM with nonzero distribution $\mathcal{N}(0, 1)$ and noise distribution $\mathcal{N}(0, \sigma^2)$. If at least one problem instance converges, we average the run-time of the problems that converged and repeat the process with $\rho \leftarrow \rho + 0.01$. We plot the timing at each $\rho$ for parameter values for which at which at least 50% of the problems were successfully recovered under the criteria [31].

It can be seen in Figure 6(a)–6(d) that the phase transition either remains nearly unchanged or increases as $n$ increases from $2^{22}$ to $2^{24}$. In particular, Figure 6(a) shows that for $\sigma = 0.001$ and $\delta = 0.001$, the phase transitions for all variants of the Robust-$\ell_0$ algorithms in fact increase in fact increase from $\approx 0.08$ to $\approx 0.1$ as $n$ increases from $2^{22}$ to $2^{24}$. Additionally, contrasting Figures 6(a) and 6(b) or 6(c) and 6(d) shows that a ten-fold increase in $\sigma$ has the expected effect of reducing the phase transition and increasing the computational time. Figures 6(a) and 6(b) show the phase transition for Robust-$\ell_0$ and Robust-$\ell_0$-adaptive remain significant even for $\delta$ as small as $10^{-3}$. Figures 6(c) and 6(d) show results for the same set of experiments, but for $\delta = 0.01$ which corresponds to a ten-fold increase in $\sigma$ over the value used in Figures 6(a) and 6(b) For $\sigma = 0.001$ the phase transition of Robust-$\ell_0$ reaches $\rho \approx 0.17$, while for $\sigma = 0.01$ the phase transitions drop to $\rho \approx 0.12$ and there is an increase in the computational time.

V. CONCLUSIONS

We have shown that the decoding framework presented in [1] can be extended to the case where the measurements are corrupted by additive noise. This framework is extended by deriving the posterior distribution of an entry in the residual being zero or being equal to another residual entry given the corrupted measurements. This Bayesian approach to decoding was implemented in Robust-$\ell_0$ and its four variants. We show that the resulting algorithms inherit some desirable properties from Parallel-$\ell_0$ like high phase transitions for low $\delta$ and large $n$ and low-latency. However, these qualities are weakened by the corruption of the measurements. Our numerical experiments show that Robust-$\ell_0$ should be considered in cases of moderate noise and $\rho \lesssim 0.3$.

ACKNOWLEDGMENTS

The authors would like to thank William Carson for many helpful discussions on imaging applications of this work. This work was supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1.

Index Terms—compressed sensing, expander graphs, dissociated signals, robust algorithms.

REFERENCES

[1] R. Mendoza-Smith and J. Tanner, “Expander 10-decoding,” Applied and Computational Harmonic Analysis, pp. –, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1063520317300210

[2] E. J. Candès and T. Tao, “Decoding by linear programming,” IEEE Transactions on Information Theory, vol. 51, no. 12, pp. 4203–4215, Dec 2005.

[3] D. L. Donoho, “Compressed sensing,” IEEE Transactions on Information Theory, vol. 52, no. 4, pp. 1289–1306, April 2006.

[4] D. Donoho, V. Stodden, Y. Tsaiag et al., “Sparselab,” SparseLab: Seeking Sparse Solutions to Linear Systems of Equations, SparseLab toolbox shared online, http://parselab.stanford.edu, 24th August, 2007.

[5] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, “An interior-point method for large-scale $\ell_1$-regularized least squares,” IEEE journal of selected topics in signal processing, vol. 1, no. 4, pp. 606–617, 2007.
[6] M. A. Figueiredo, R. D. Nowak, and S. J. Wright, “Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems,” IEEE Journal of selected topics in signal processing, vol. 1, no. 4, pp. 586–597, 2007.

[7] E. Candes and J. Romberg, “l1-magic: Recovery of sparse signals via convex programming,” URL: www.acm.caltech.edu/l1magic/downloads/l1magic.pdf, vol. 4, p. 14, 2005.

[8] T. Blumensath and M. E. Davies, “Iterative hard thresholding for compressed sensing,” CoRR, vol. abs/0805.0510, 2008.

[9] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, “Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition,” pp. 40–44 vol.1, Nov 1993.

[10] D. Needell and J. A. Tropp, “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples,” Applied and Computational Harmonic Analysis, vol. 26, no. 3, pp. 301–321, 2009.

[11] T. Blumensath and M. E. Davies, “Normalized iterative hard thresholding: Guaranteed stability and performance,” IEEE Journal of selected topics in signal processing, vol. 4, no. 2, pp. 298–309, 2010.

[12] W. Dai and O. Milenkovic, “Subspace pursuit for compressive sensing signal reconstruction,” IEEE Trans. Inform. Theory, vol. 55, no. 5, pp. 2230–2249, 2009.

[13] J. D. Blanchard, J. Tanner, and K. Wei, “CGIHT: Conjugate gradient iterative hard thresholding for compressed sensing and matrix completion,” Information and Inference, vol. 4, no. 4, pp. 289–327, 2015.

[14] V. Cevher, “An ALPS view of sparse recovery,” in Acoustics Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, 2011, pp. 5808–5811.

[15] S. Foucart and H. Rauhut, A mathematical introduction to compressive sensing. Birkhäuser Basel, 2013, vol. 1, no. 3.

[16] S. Sarvotham, D. Baron, and R. G. Baraniuk, “Sudocodes — fast measurement and reconstruction of sparse signals,” in 2006 IEEE International Symposium on Information Theory, July 2006, pp. 2804–2808.

[17] W. Xu and B. Hassibi, “Efficient compressive sensing with deterministic guarantees using expander graphs,” pp. 414–419, Sept 2007.

[18] R. Berinde, P. Indyk, and M. Ruzic, “Practical near-optimal sparse recovery in the l1 norm,” pp. 36–43, Sept 2009.

[19] 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, Sept 2009.

[20] Y. Ma, D. Baron, and D. Needell, “Two-part reconstruction with noisy-sudocodes,” IEEE Trans. Signal Processing, vol. 62, no. 23, pp. 6323–6334, 2014. [Online]. Available: http://dx.doi.org/10.1109/TSP.2014.2362892

[21] B. Bah and J. Tanner, “Vanishingly sparse matrices and expander graphs, with application to compressed sensing,” IEEE transactions on information theory, vol. 59, no. 11, pp. 7491–7508, 2013.

[22] A. Klenke, Probability theory: a comprehensive course. Springer Science & Business Media, 2013.

[23] F. Leone, L. Nelson, and R. Nottingham, “The folded normal distribution,” Technometrics, vol. 3, no. 4, pp. 543–550, 1961.

[24] D. L. Donoho and J. Tanner, “Precise undersampling theorems,” Proceedings of the IEEE, vol. 98, no. 6, pp. 913–924, 2010.

[25] J. D. Blanchard, J. Tanner, and K. Wei, “Conjugate gradient iterative hard thresholding: Observed noise stability for compressed sensing,” IEEE Transactions on Signal Processing, vol. 63, no. 2, pp. 528–537, Jan 2015.

[26] J. D. Blanchard and J. Tanner, “Performance comparisons of greedy algorithms in compressed sensing.” Numerical Linear Algebra with Applications, vol. 22, no. 2, pp. 254–282, 2015.

[27] ——, “GPU accelerated greedy algorithms for compressed sensing.” Mathematical Programming Computation, vol. 5, no. 3, pp. 267–304, 2013.

[28] D. L. Donoho and J. Tanner, “Exponential bounds implying construction of compressed sensing matrices, error-correcting codes and neighborly polytopes by random sampling.” IEEE Transactions on Information Theory, vol. 56, no. 4, pp. 2002–2016, 2010.