Saturation Noise and Permutation Noise in Compressed Sensing

Bachelor's Thesis project

Sudhansh Peddabomma

under the supervision of Ajit Rajwade
Indian Institute of Technology, Bombay

April 2023

Contents

1 Introduction

The paradigm of Compressed Sensing involves the reconstruction of sparse signals from an under-determined system of linear equations. Typically, the measurements are attributed with Gaussian noise. These systems can be solved using an appropriately designed cost functions such as LASSO, which are generally convex, with a standard optimising procedure. Such reconstruction procedures are feasible under two conditions

1. The signal should be sparse enough
2. The number of measurements should be sufficient. This is typically ensured by choosing a measurements matrix that is 'incoherent' enough with the signal.

There has been a lot of theory developed that sets up the required conditions for a good reconstruction under Gaussian noise. In this report, we shall explore the area of Permutation noise in Compressed Sensing. That is, we try to set up the theory and demonstrate the performance of reconstruction algorithms when the order of measurements is permuted.

1.1 Compressed Sensing

Let us write down the problem of compressed sensing in terms of a mathematical model. Let $x \in \mathbb{R}^n$ be a sparse-vector with support S and sparsity s. Consider a sensing matrix $A \in \mathbb{R}^{m \times n}$ such that $m \ll n$. Then, the measurements of the signal are denoted by $y \in \mathbb{R}^m$ defined by

$$y = Ax \tag{1}$$

The measurment matrix A is typically chosen to follow the Restricted Isometry Property (RIP) [1] to ensure the incoherence between the measurements. This property ensures that an under-determined system of linear measurements is sufficient to reconstruct the true sparse signal, thereby allowing compression.
The RIP property for a matrix A is defined as follows. A matrix A is said to follow RIP of order s if for any s-sparse vector x, the following holds -

$$(1 - \delta_S)||x||^2 \leq ||Ax||^2 \leq (1 + \delta_S)||x||^2 \tag{2}$$

where, δ_S is the Restricted Isometry Constant (RIC) of the matrix A. RIP can be ensured by choosing a measurement matrix sampled from a Gaussian distribution or a Bernoulli distribution.

In general, when the signals are not sparse, the measurement matrix is associated with a transformation matrix $\Psi \in \mathbb{R}^{n \times n}$ such that x is sparse under Ψ. That is,

$$x = \Psi\theta \tag{3}$$

where $\theta \in \mathbb{R}^n$ is sparse. Then, the equation modeling this situation would be

$$y = (A\Psi)\theta \tag{4}$$

In this report, we will assume that the signal are sufficiently sparse so that we can do away without any transformation matrix.
The signal x can be recovered from the measurements y using the following cost function -

$$\min \|x\|_0 \text{ subject to } y = Ax \tag{5}$$

However, this optimisation problem using l_0 norm is NP-hard. Therefore, we resort to using a relaxed version utilisng l_1 norm known as Basis Pursuit.

$$\min \|x\|_1 \text{ subject to } y = Ax \tag{6}$$

This is a linear optimization problem that can be solved in polynomial time. The solution to the above optimization problem is exact with probability $1 - \delta$ if,

$$m \geq C \log\Big(\frac{n}{\delta}\Big) ||x||_0 \mu^2(A) \tag{7}$$

where C is a constant and μ is the incoherence function [1] defined by

$$\mu(A) = \sqrt{n} \max_{1 \leq j \leq m, 1 \leq i \leq n} \|\langle A^j, I_i \rangle\| \tag{8}$$

where A^j is the jth row of A and I_i is the ith column of $I^{n \times n}$.

In general, the signals obtained practically are associated with noise which is typically Gaussian. In this case, the problem model is given by

$$y = Ax + \eta \tag{9}$$

where $\eta \in \mathbb{R}^m$ represents the noise. Then the cost function gets modified as

$$\min \|x\|_1 \text{ such that } \|y - Ax\|_2^2 < \epsilon \tag{10}$$

where $\epsilon \in \mathbb{R}$ is chosen based on the upper bound of the magnitude of noise. This equation can be equivalently stated as

$$\min \|y - Ax\|_2^2 + \lambda \|x\|_1 \tag{11}$$

which is famously known as LASSO Optimization. Here, λ is the regularization parameter determined empirically.

If the matrix A follows RIP of the order $2s$ where s is the sparsity of x and $\delta_{2s} < \sqrt{2} - 1$, then the error in the reconstruction is given by

$$||x^* - x||_2 \leq \frac{C_0}{\sqrt{s}} ||x^* - x_s||_1 + C_1 \epsilon \tag{12}$$

where x^* is the true solution to $y = Ax$ and C_0, C_1 are constants independent of n [1].

This optimisation procedure works quite well when the signal has Gaussian independent noise, i.e, $\eta \sim \mathcal{N}(0, \sigma^2)$. This distribution of noise accommodates most of the noise models we see in signals generally. However, we shall see that there are scenarios where the noise has more components, such as a heavy tailed distribution. In such cases, the LASSO cost function does not fare out well and it needs to be modified to adopt to the noise model. The presence of Permutation noise in the signals highlights such a case.

1.2 Permutation Noise in Compressive Sensing

Permutation Noise refers to the scenarios where the measurements are mislabeled, i.e. the order of measurements is permuted. The motivation for handling such cases arises in many applications. For instance, a bit-rate reduction in a channel that does not preserve data order. It may also happen in an experimental setup with a large number of sensors – e.g., a microphone array, due to some handcrafting mistakes in the wiring between sensors and A/D converters.

An even more prominent example in the recent cases would be that of group testing for COVID19. Given n samples, each of which may be diseased or not, group testing aims to determine their health status indirectly by performing tests on $m < n$ 'groups' (also called 'pools'), where a group is obtained by mixing a subset of the n samples by testing of $m < n$ groups. Since the number of diseased samples is typically less, this problem can be modeled as a compressed sensing problem with a sparse vector.
The issue of mislabeling arises when the technician accidentally switches the labels of the group samples. This situation can occur in non-group testing too.

The methods that have been proposed to deal with this situation typically use brute-force where different permutations of measurements are explored [3]. We propose a hypothesis testing method using a LASSO debiasing procedure motivated from the work done in bit-flip noise in group testing.

1.3 Problem Model

Given an s-sparse vector $x \in \mathbb{R}^n$ and a measurement matrix $A \in \mathbb{R}^{m \times n}$, the permuted measurements are given by

$$y = PAx \tag{13}$$

where $P \in \mathbb{R}^{m \times m}$ is a permutation matrix represented by

$$P^i = I^j \tag{14}$$

if the measurement i is switched with measurement j where $i, j \in [m]$. The permutation matrix P can be modified to represent duplication noise too, i.e. the measurements are repeated.

We shall show that this model can be represented by the situation when the noise has a heavy-tailed distribution component. We shall use the theory derived for heavy-tailed distributions from the bit-flip noise model to develop a reconstruction algorithm for permuted measurements.

2 Theory for Permutation Noise

In this section, we present the extension of theory from bit-flip noise to permutation noise. We propose an algorithm to detect and even correct for the permutation errors.

2.1 Problem Formulation

Like before, let $b \in \mathbb{R}^n$ be the signal and $A \in \mathbb{R}^{m \times n}$ be a zero-centered Bernoulli matrix. The measurements $y \in \mathbb{R}^m$ are then given by

$$y = Ab + \eta \tag{15}$$

with a Gaussian noise $\eta \sim \mathcal{N}(0, \sigma^2)$. In the presence of permutation noise, the ith measurement is switched with jth measurement. That gives us

$$y = Ab + \delta + \eta \tag{16}$$

$$= \tilde{A}b + \eta = (A + \Delta A)b + \eta \tag{17}$$

where $\delta \in \mathbb{R}^m$ is the error vector for permutation noise. Like before, $\tilde{A} \in \mathbb{R}^{m \times n}$ is the observed sensing matrix and $\Delta A \in \mathbb{R}^{m \times n}$ represents the error in the sensing matrix.
Note that, we switch the measurements after the Gaussian noise is added. However, since we assume independent Gaussian noise for each measurement, the final equation can still be represented in such a manner. That is, permuting the final measurements is equivalent to saying permuting the measurements before adding the additive noise.

Let the true measurements Ab be represented by z. We then have,

$$\delta_i = \begin{cases} z_j - z_i & i\text{th measurement is switched with } j\text{th measurement} \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

This also gives us

$$\Delta A_i = \begin{cases} A_j - A_i & i\text{th measurement is switched with } j\text{th measurement} \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

2.2 Debiasing LASSO

The Huber loss estimator is modeled over the equation given by,

$$y = Ab + \delta + \eta \tag{20}$$

$$= [A|I] \begin{pmatrix} b \\ \delta \end{pmatrix} + \eta \tag{21}$$

$$= [A|I]x + \eta \tag{22}$$

where x is the concatenation of b and δ.

Let $x_\lambda \triangleq \arg\min_x \|y - (A|I_{m\times m})x\|^2 + \lambda\|x\|_1$. Note that, $x_\lambda^T = (b_\lambda^T, b_\lambda^T)$. Then the debiased estimate of x is given as follows:

$$x_d \triangleq x_\lambda + \frac{1}{m} M(A|I_{m\times m})^T (y - (A|I_{m\times m})x_\lambda), \tag{23}$$

where $M \in \mathbb{R}^{(m+n)\times(m+n)}$ is obtained from the approximate inverse of an empirical correlation matrix $\hat{\Sigma}_1 \triangleq A^T A/m$ given by M_1 obtained from the optimisation formulation given in eqn.(4) of [6]. To be more specific, M is defined as

$$M = \begin{bmatrix} M_1^{n\times n} & 0^{n\times m} \\ 0^{m\times n} & mI^{m\times m} \end{bmatrix} \tag{24}$$

Let $x^* = [b^*; \delta^*]$ represent the true value of x. Note that $\delta^* = 0$.

We shall now show that the distribution of δ follows the conditions required in Lemma 3. This involves showing that the expectation of permutation noise goes to 0 asymptotically. We can then use hypothesis testing to detect permutation noise as we did in bit-flip noise.

2.3 Expectation of the permutation noise

We make the following assumptions to derive the distribution of ΔA.

- **(A3)** There are $t \ll m$ mislabeled measurements in total
- **(A4)** Every measurement has an equal probability of being switched with any other measurement. The switching of measurements is independent of one another.

In such a case, the probability that the ith measurement is switched with the jth is given by $1/m$. The expected value of δ_i is then given by

$$E[\Delta A_i|A] = \frac{1}{m}\left(\frac{\sum_{j\neq i} A_j}{m-1} - A_i\right) \tag{25}$$

Now, when there are t permutations, we get

$$E[\Delta A_i|A] = \frac{t}{m}\left(\frac{\sum_{j\neq i} A_j}{m-1} - A_i\right) \tag{26}$$

Notice that the error goes to 0 asymptotically as $m \to \infty$ with the order of $o(1/m)$. Therefore, since permutation noise satisfies the required properties for Lemma 3, we can use the results from Lemma 4.

2.4 Detection of permuted measurements

As the permutation error satisfies the required properties, we can use Hypothesis testing on the debiased reconstruction to detect the permuted measurements. From the theorem in bit-flip noise we have

$$\sqrt{m}(\delta_d - \delta^*) - \sqrt{m}A(b_d - b_\lambda) \sim \mathcal{N}(0, \sigma^2\Sigma_\delta) \tag{27}$$

Furthermore, since we have no information about which measurements are permuted, the null hypothesis is taken as $\delta_i = 0$ for all $i \in [m]$. Since if there is no permutation, there should be no effect on the value of the measurements implying that $\delta_i^* = 0$. Based on the debiased lasso estimate of δ_i^*, the test statistic for testing $H_0 : \delta_i^* = 0$ vs $H_1 : \delta_i^* \neq 0$, is

$$D_i = \frac{[\sqrt{m}(\delta_d - \delta^*) - \Delta_{31}]_i}{\sigma\sqrt{[\Sigma_\delta]_{ii}}} \tag{28}$$

where

$$\Delta_{31} = \sqrt{m}A(b_d - b_\lambda) \tag{29}$$

$$\Sigma_\delta = \left(\sqrt{m}I + AM_1A^T/\sqrt{m}\right)\left(\sqrt{m}I + AM_1A^T/\sqrt{m}\right)^T \tag{30}$$

The expression of D_i follows the distribution $\mathcal{N}(0,1)$ under the null hypothesis.

2.5 Reconstruction of signal

After detecting the noisy measurements using the above test, we can simply choose to drop these measurements. We shall refer to this method as "drop" for lack of a better word.

However, one can also try to correct the permutations after detecting them. The following algorithm represents a greedy approach to correct the measurements -

Algorithm 1 Correcting for Permutation noise

Input: Measurement vector y, Sensing matrix A, LASSO estimate b_λ, debiased estimate b_d, $\alpha = 1.68$, the set P of detected permuted measurements
Output: Corrected measurement vector y_c

```
y_c ← y
for index i ∈ P do
    correct_index ← i
    min ← α
    for index j ∈ P do
        y_c[i] ← y[j]
        x'_λ ← arg min_x ‖y_c − (A|I_{m×m})x‖² + λ‖x‖₁
        D'_i ← [√m(δ'_d) − A(b'_d − b'_λ)]_i / σ√[Σ_δ]_ii
        confidence ← α − |D'_i|
        if confidence < min then
            correct_index ← j
            min ← confidence
        end if
    end for
    y_c[i] ← y[correct_index]
end for
```

In the above algorithm, we find the consider each index in the noisy set and switch them with another index in the set. We choose the measurement which gives the highest confidence for the switch. Then, we proceed switching every measurement in the noisy set greedily. We shall refer to this method as "corrected" moving further.

3 Experiments

3.1 Setup and Parameters

We check the performance of the proposed method by comparing the RMSE of the reconstructions obtained with the oracle result. The oracle result refers to the LASSO estimate of the signal when there is no permutation noise present. We compare the reconstructions while varying different parameters.

All the experiments are performed on signals of dimension $n = 128$ and $n = 256$. The non-zero elements in the signal are chosen from a uniform distribution between 0 and 1. The elements of the sensing matrix A are drawn i.i.d. from a Bernoulli distribution of -1 and 1 keeping in mind that we need a zero-centered matrix and also so that A would obey RIP with high probability. The additive noise is added to the measurements, followed by the permutation of measurements. For permuting the measurements, a set of measurements whose size is given by the parameter f_p is chosen and the measurements in this set are

permuted randomly.
Keeping all other parameters fixed, we analyse the variation in the curvature with regard to change in (A) number of measurements m; (B) signal sparsity s expressed as fraction $f_{sp} \in [0, 1]$ of signal dimension n; and (C) the fraction $f_p \in [0, 1]$ of the m measurements that are permuted.

When the value of $n = 128$,
For the measurements experiment (i.e. (A)), m is varied in $\{50, 65, 80, \ldots, 125\}$ with $f_{sp} = 0.05$, $f_p = 0.05$, $f_\sigma = 0.05$. For the sparsity experiment (i.e. (B)), f_{sp} is varied in $\{0.01, 0.04, 0.08, \ldots, 0.2\}$ with $m = 100$, $f_p = 0.05$, $f_\sigma = 0.05$. For the permutation noise experiment (i.e. (D)), f_p is varied in $\{0.02, 0.03, 0.05, 0.08, 0.1, 0.12\}$ with $m = 100$, $f_{sp} = 0.05$, $f_\sigma = 0.05$.

When the value of $n = 256$,
For the measurements experiment (i.e. (A)), m is varied in $\{100, 120, 140, \ldots, 200\}$ with $f_{sp} = 0.05$, $f_p = 0.05$, $f_\sigma = 0.05$. For the sparsity experiment (i.e. (B)), f_{sp} is varied in $\{0.01, 0.04, 0.08, \ldots, 0.2\}$ with $m = 200$, $f_p = 0.05$, $f_\sigma = 0.05$. For the permutation noise experiment (i.e. (D)), f_p is varied in $\{0.02, 0.03, 0.05, 0.08, 0.1, 0.12\}$ with $m = 200$, $f_{sp} = 0.05$, $f_\sigma = 0.05$. The experiments are repeated over 25 vectors and the RMSE is averaged.

The regularization parameters are chosen via cross validation with the unpermuted measurements with the validation set containing 20% of the measurements. The validation set is guaranteed to not have any permutations. This can be ensured in group testing by performing duplicate measurements over these samples. Cross validation is done with a probability of 40% for each signal sample. The algorithms are implemented via the `cvx` package available on `MATLAB`.

3.2 Results

The results obtained for $n = 128$ and $n = 256$ are displayed in figures 1 and 2 respectively.

We have also conducted experiments to see how Precision and Recall of the detection algorithm changes with respect to the number of measurements, sparsity of signal and the amount of permutation noise. The results are plotted as heatmaps and are displayed in figures 3 and 4

3.3 Observations

In case of permutation noise, the reconstruction method is very sensitive to the $m > (s \log n)^2$ condition. I have noticed that the cross validation yields poor parameters frequently if this does not hold. To fix this issue, we ensured that the indices in the validation set do not have any permutation.

Even after ensuring this, the reconstructed δ value has discrepancies which leads to poor recall values. It is also worth to notice that the precision values are quite high for the method in most scenarios. That means that the number of false positives is quite low. In contrast, the value of α in Hypothesis testing has a critical role in determining the number of false negatives. A very high value of α will lead to a large number of false negatives.

Other than this, we notice the expected trends in the RMSE for each set of experiments. The values of RMSE should decrease as the number of measurements m increase. They should increase when sparsity f_s or the permutation noise f_p increases.

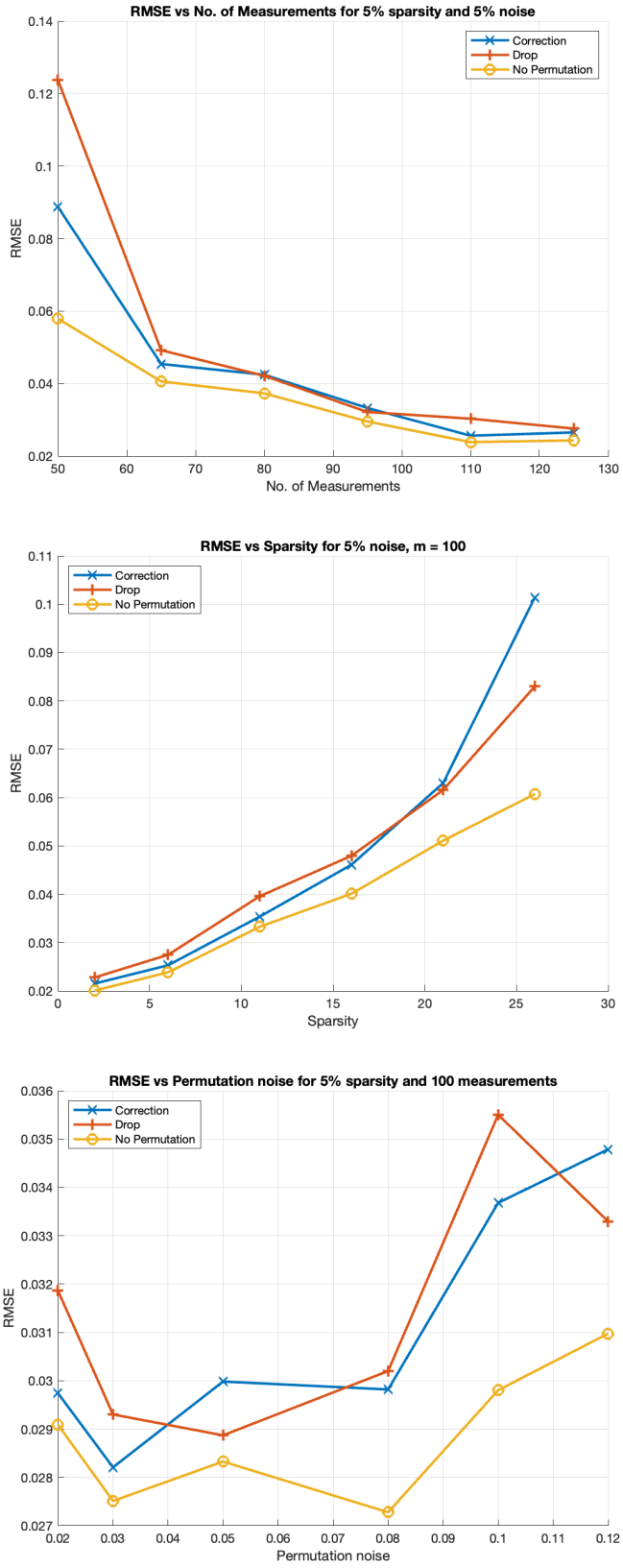


Figure 1: RMSE comparison between (i) corrected method (ii) drop method (iii) oracle results with respect to (A) Number of measurements m, (B) Signal sparsity s and (C) Permutation noise f_p for $n = 128$

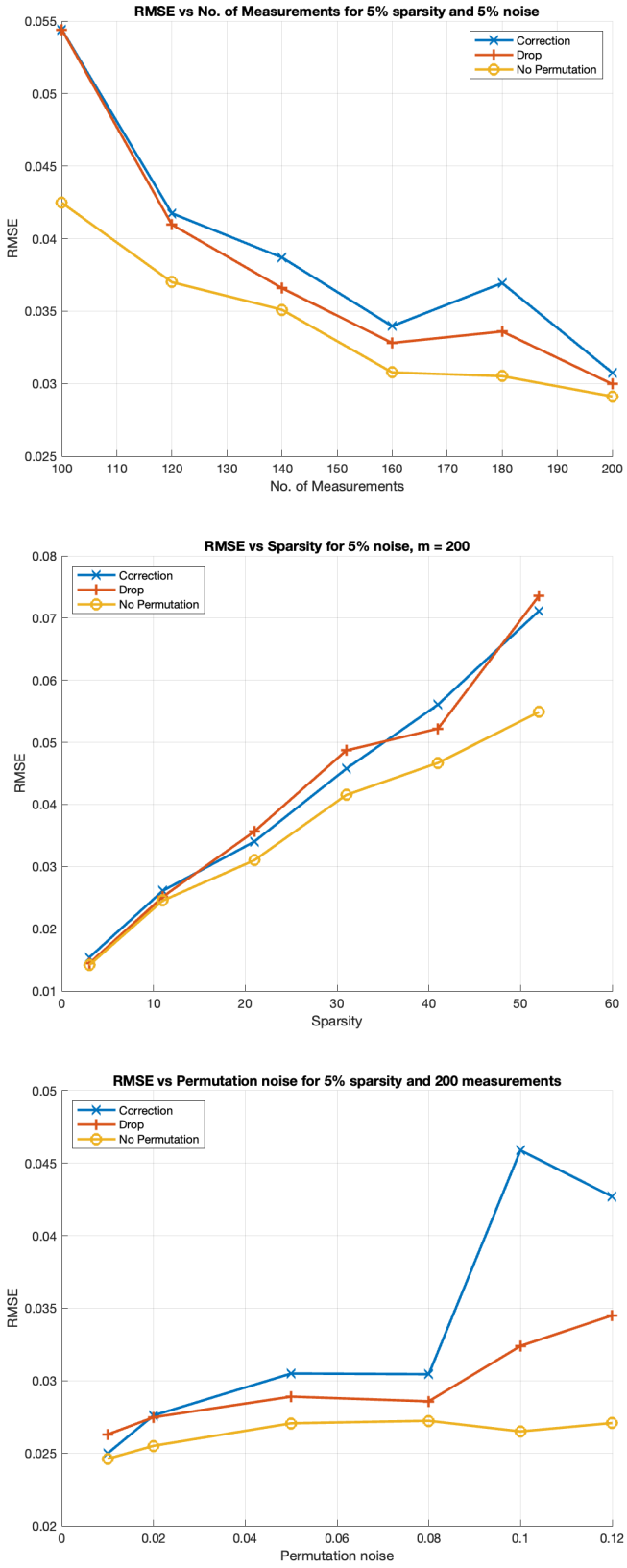


Figure 2: RMSE comparison between (i) corrected method (ii) drop method (iii) oracle results with respect to (A) Number of measurements m, (B) Signal sparsity s and (C) Permutation noise f_p for $n = 256$

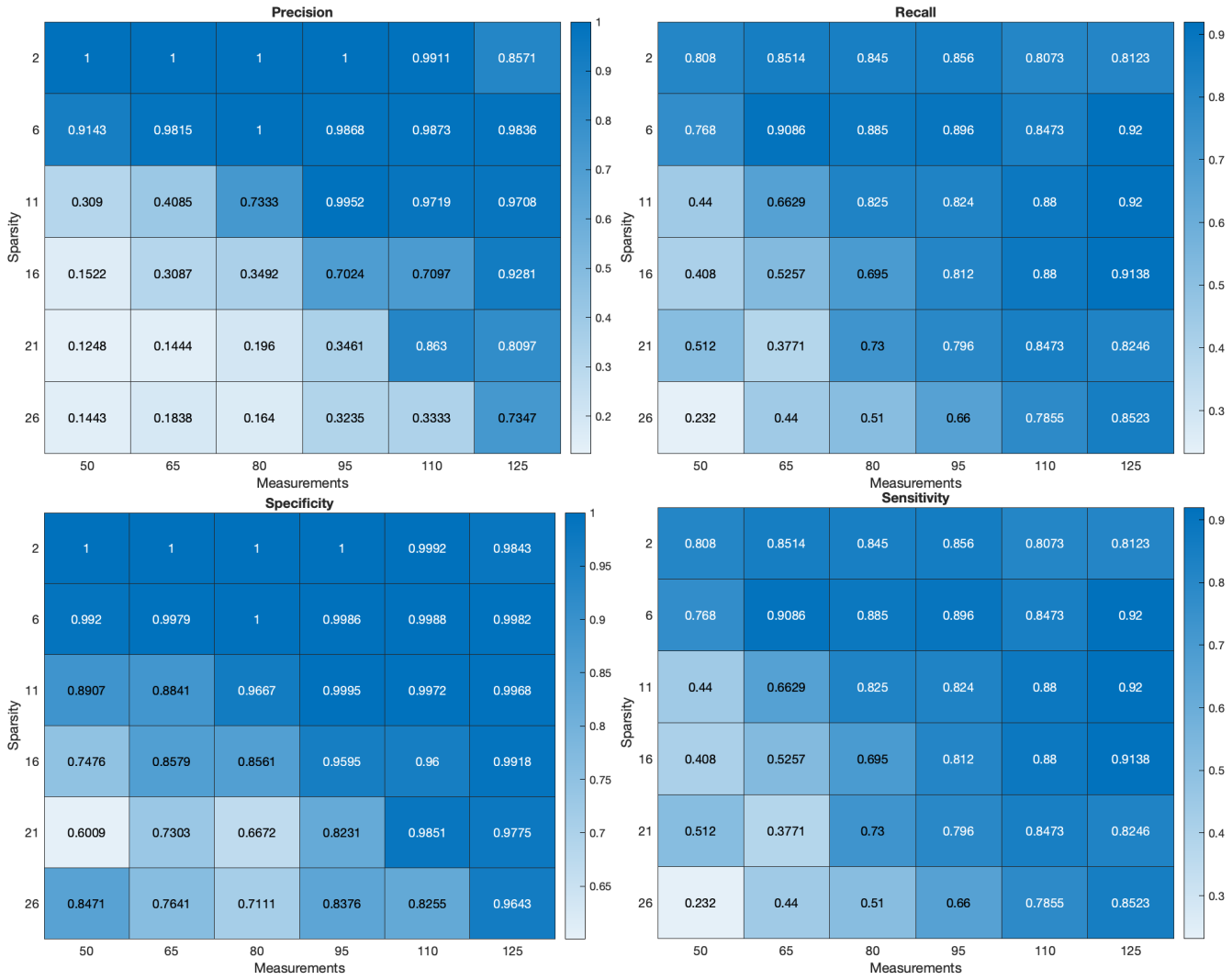


Figure 3: Heatmaps for recall when the $f_p = 0.1$

In almost all the experiments, we see excellent reconstructions with the RMSE rarely crossing 10%. However, in the experiments where the permutation noise is varied, we see that there are a lot of anomalies in the trend. The exact cause of this issue is unknown but the very low values of RMSE could be partially responsible.

Apart from this, we see that the correction algorithm closely matches with the performance of reconstructions where we drop the noisy measurements. This is partially due to the fact that the number of permuted measurements is much smaller than the total number of measurements. Also, in some cases dropping measurements also takes care of the additive Gaussian noise being too high. However, this situation is rare but this might result in poorer performance of the correction method.

In the heatmaps, the results are as expected. The values of recall, precision, specificity and sensitivity increase as the number of measurements increases and decrease as the sparsity increases.

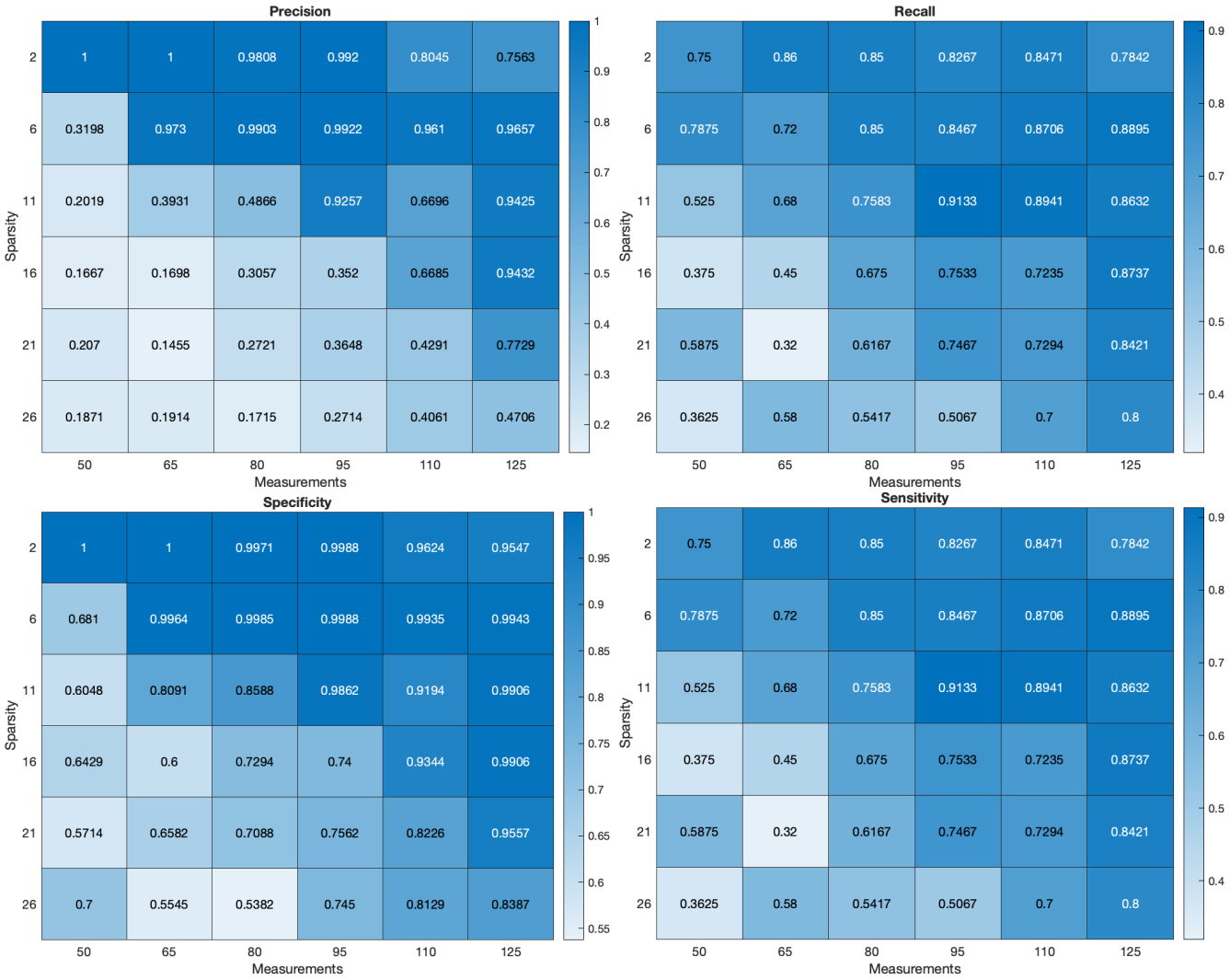


Figure 4: Heatmaps for recall when the $f_p = 0.15$

4 Double Debiasing for LASSO

4.1 Theory

The LASSO method, although a popular tool for solving compressive linear models, has an inherent bias that often leads to some error in statistical inference methods. To correct permutation noise in signal reconstruction, we have debiased the estimate of LASSO (23) to build confidence intervals on the debiased estimate. Since we use an augmented matrix construction as seen in the Huber Loss formulation (62), we modified the confidence intervals expression as seen in (28).

Recent work in [10] mentions that the debiased LASSO is not optimal when the number of strong coefficients dominate the number of weak coefficients. The bias in the strong coefficient increases in the presence of weak coefficients. This is quite important in our case as we have an augmented vector formulation.

$$y = [A|I] \begin{pmatrix} b \\ \delta \end{pmatrix} + \eta$$

Here, the coefficients in δ can be much larger than the coefficients in b. Although, we have a difference in magnitude among the coefficients, the number of strong coefficients need not necessarily dominate the number of weak coefficients. The approach put forward by [10] has the same performance as the debiased approach in such cases. The following sections will describe the technique and the experimental results with these methods.

4.2 Bootstrapping the debiased LASSO

Consider the model

$$y = Ab + \eta$$

where $y \in \mathbb{R}^m$, $x \in \mathbb{R}^n$, and $A \in \mathbb{R}^{m \times n}$ is a random matrix chosen with the covariance $\Sigma_n = A^T A/m$. We define a population gram matrix given by $\Sigma = \mathbb{E}[\Sigma_n]$ and also define $\Theta = \Sigma^{-1}$. We shall assume Θ is known in our case.

Let x_u be the initial LASSO estimate with the hyper-parameter λ. We define the correction score $W \in \mathbb{R}^{n \times m}$ as

$$\hat{W}^j = \frac{n P_{\Lambda_j} A \Theta_j}{X_j' * P_\Lambda A \Theta_j} \tag{31}$$

where P_{Λ_j} is a projection operator defined as follows. Consider the support estimated by Lasso given by $\hat{S} = \{j : x_u \neq 0\}$. Then, $\Lambda_j \triangleq \hat{S} \cap \{-j\}$. The expression for P_{Λ_j} is given by

$$P_{\Lambda_j} = I_n - A_{\Lambda_j} (A_{\Lambda_j}^T A_{\Lambda_j})^{-1} A_{\Lambda_j}^T \tag{32}$$

$A_{\Lambda_j}^T A_{\Lambda_j}$ is shown to be invertible with a high probability. We want the weights to behave such that $(e_j^T - w^T A/m)_S = 0$ and $(e_j^T - w^T A/m)_{S^c} \approx 0$. From Karush-Kuhn-Tucker conditions, we can see that $(e_j^T - w^T A/m)_{\hat{S}} = 0$.

A modified debiased LASSO estimate is calculated as

$$x_d = x_u + \frac{W * (y - A * x_u)}{m} \tag{33}$$

We compute x^*_λ as the LASSO estimate for the measurements $A * x_u$ and measurement matrix A. Then, the bias in x_d is estimated as

$$b = x^*_\lambda + \frac{WA * (x_u - x^*_\lambda)}{m} - x_u \tag{34}$$

The final bootstrapped-debiased estimate is given by

$$x_{dd} = x_d - b \tag{35}$$

Then, the confidence intervals on x_{dd} are constructed as

$$x_{dd} \pm \frac{q_{\alpha/2}\hat{\sigma}\|w_j\|_2}{m} \tag{36}$$

where $\hat{\sigma}$ is any consistent estimator of σ.

4.3 Experiments

The following results are an attempt to replicate the results in [10]. As mentioned in the paper, we choose $n = 300$, $m = 250$, and the design matrix A is chosen such that each row of A is *i.i.d* from a Gaussian distribution with mean 0 and covariance matrix $\Sigma = I_{n\times n}$. The signals are generated with a known noise level $\sigma = 1$. We consider 5 levels of sparsity $s = [4, 6, 8, 10, 12]$, and the support for x is chosen as $x_s = \{4, 2, 4, 2, \ldots, 4, 2, 0.2\}$. The signal is constructed so that the first $s - 1$ coefficients are strong and the last one is weak. This can also be verified The debiased estimate is computed using $\lambda = 2\sigma\sqrt{logn/m}$ and $w_j = X\Theta_j$. I have also conducted experiments using the correction score mentioned in (31), and used $m = 150$ for these experiments due to computational limitations. The experiments have been conducted on 100 independent iterations, and the bias was calculated using the median of 10 reconstructions in each iteration. The results are summarised in Figures 5, **??**, 7 and 8.

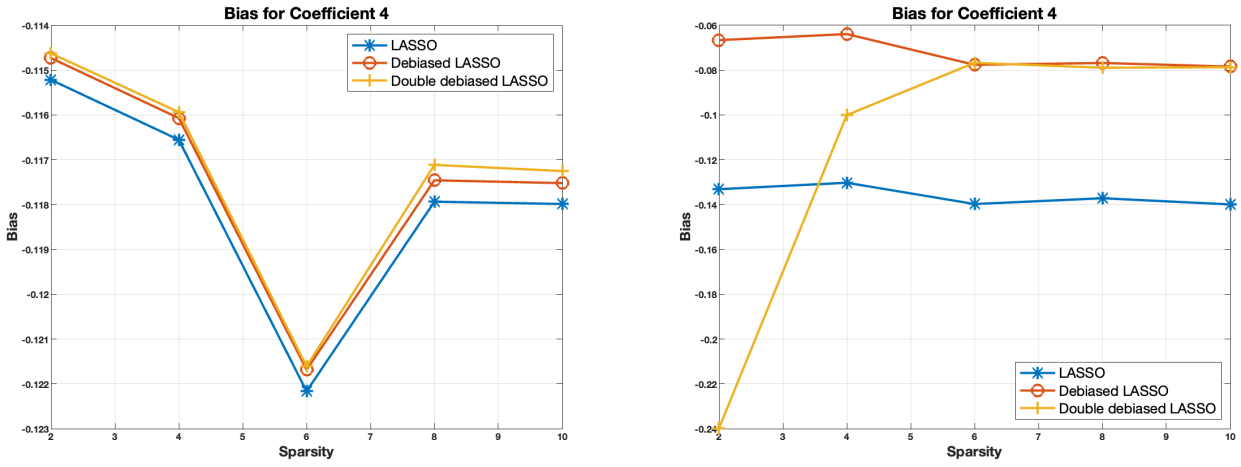


Figure 5: Bias in coefficient with value 4 with $w_j = X\Theta_j$ on the left, and w_j from (31) on the right

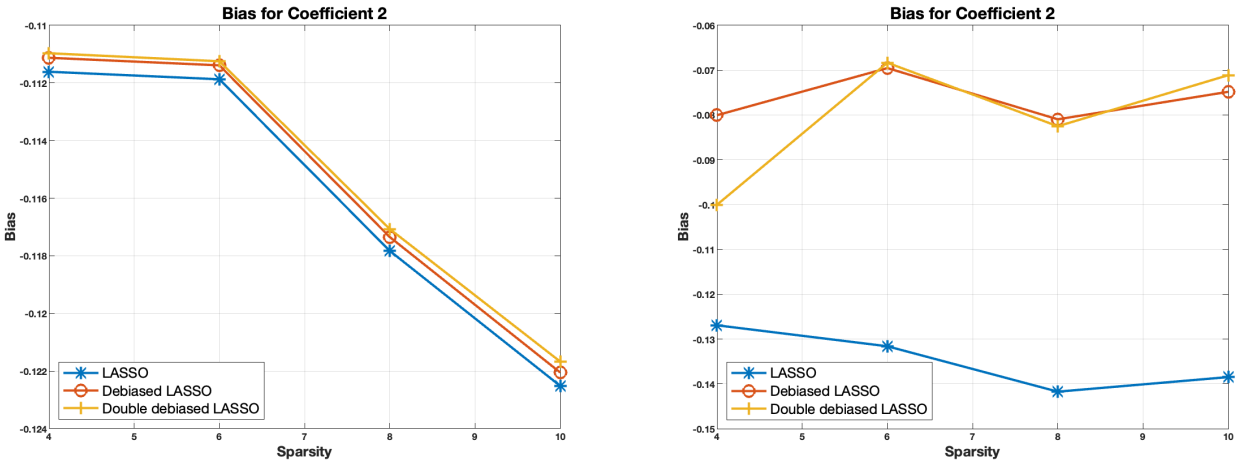


Figure 6: Bias in coefficient with value 2 with $w_j = X\Theta_j$ on the left, and w_j from (31) on the right

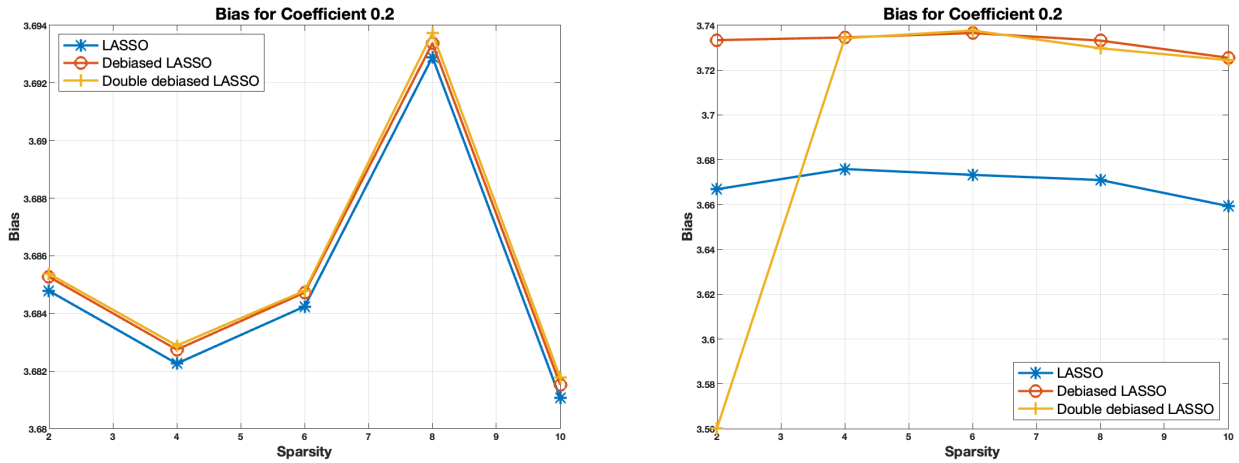


Figure 7: Bias in coefficient with value 0.2 with $w_j = X\Theta_j$ on the left, and w_j from (31) on the right

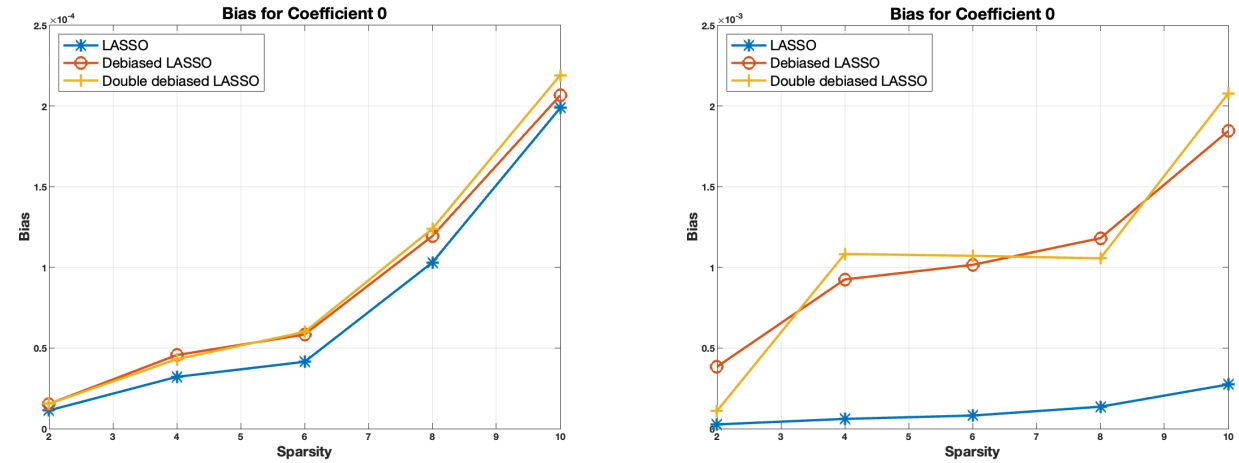


Figure 8: Bias in coefficient with value 0 with $w_j = X\Theta_j$ on the left, and w_j from (31) on the right

4.4 Observations

The debiasing experiments require m to be of the order of $(s \log n)^2$. However, the parameters used in the paper are $m = 100$, $n = 300$, and s varying between 2 and 16. With the given n and $s = 6$, the order of m should approximately be ~ 1100. There is some flexibility with the constant factor multiplied, but the conditions for debiasing are constrictive in this manner.

The results with correction scores calculated from (31) are very unstable, and the matrix $A_{\Lambda_j}^T A_{\Lambda_j}$ was displayed as non-invertible frequently. This could be due to m not satisfying the $\mathcal{O}((s \log n)^2)$ condition.

The decrease in bias with the modified debiased approach is very significant as compared to the subsequent decrease in bias due to bootstrapped method. The bias in unsupported regions of the signal actually increased after debiasing, and the bias in the 0.2 coefficient is very high with all three methods.

There is a uniform decrease in bias among the strong coefficients 2 and 4, but this change is miniscule.

4.5 Conclusion

The bootstrapped method correctly addresses the problem of increased bias among the strong coefficients in the support. This issue could improve the results in permutation noise correction. However, the improvement in the results as seen in the experiments in not too significant as compared to the debiased approach. In order to apply the bootstrapped method to permutation noise correction, the theory for bootstrap debiasing has to be modified to work for our augmented vector cost function formulation.

5 Theory for Saturation Noise

In practical scenarios, the measurements obtained from sensors may be corrupted by various sources of noise, including saturation noise caused by the limited dynamic range of the sensors. Saturation noise can lead to the clipping of measurements, resulting in loss of information and degradation of the quality of the recovered signal.

5.1 Problem Model

To model the presence of saturation noise, we introduce a clipping function $\mathcal{C}(\cdot)$ that maps the noisy measurements to a finite range. The clipped measurements, denoted as $\boldsymbol{y}_{\text{clip}}$, are given by:

$$\boldsymbol{y}_{\text{clip}} = \mathcal{C}(\boldsymbol{y}; a, b), \tag{37}$$

where $\mathcal{C}(\boldsymbol{y}, a, b)$ is a component-wise clipping operation that truncates the values of $\boldsymbol{y}$ to the interval $[a, b]$, where a and b represent the lower and upper saturation levels, respectively.

5.2 Previous Work

There exists a moderate-sized literature on the problem of compressed sensing (CS) recovery from saturated measurements, which we summarize here.
The work in [7] proposes two types of estimators for CS recovery from measurements with saturation effects and uniform quantization (i.e., bounded) noise:
(1) 'Saturation Rejection' (SR), which excludes saturated measurements and performs recovery only from the non-saturated measurements via the estimator:

$$\min \|\boldsymbol{x}\|_1 \text{ s.t. } \sum_{i \in Sns} \left(y_i - \boldsymbol{A^i x} \right)^2 \leq \epsilon_{ns}^2 \tag{38}$$

(2) 'Saturation Consistency' (SC), which imposes the added constraint in the SR estimator, that the recovered signal $\hat{\boldsymbol{x}}$ should obey the conditions:

$$\forall i \in S_-, \boldsymbol{A^i}\hat{\boldsymbol{x}} \leq -(\tau - \Delta) \qquad\qquad \forall i \in S_+, \boldsymbol{A^i}\hat{\boldsymbol{x}} \geq \tau - \Delta \tag{39}$$

where Δ denotes quantization width.
The SR method may potentially ignore valuable measurements, depending on the relationship between τ and $|x|_2$. In the worst case, the remaining part of the sensing matrix may not satisfy the Restricted Isometry Property (RIP) due to an insufficient number of measurements. On the other hand, the SC method is difficult to adapt to saturation effects with Gaussian noise, which is unbounded in nature.
In recent work by [8, 11], a cost function based on the assumption that set of saturated measurements is sparse is optimized:

$$J_{\text{ss}}(\boldsymbol{x}) = \lambda(\|\boldsymbol{x}\|_1 + \|\boldsymbol{r}\|_1) + \|\boldsymbol{y} - (\boldsymbol{A}\boldsymbol{x} + \boldsymbol{r})\|_2 \tag{40}$$

$$= \lambda\|\boldsymbol{x}; \boldsymbol{r}\|_1 + \|\boldsymbol{y} - [\boldsymbol{A}|\boldsymbol{I}](\boldsymbol{x}; \boldsymbol{r})\|_2 \tag{41}$$

where $\boldsymbol{r}$ refers to the error due to saturation effects, $(\boldsymbol{x}; \boldsymbol{r})$ is the concatenation of column vectors $\boldsymbol{x}$ and $\boldsymbol{r}$, $\boldsymbol{I}$ is the identity matrix, and the $|\boldsymbol{r}|_1$ term promotes sparsity of the vector $\boldsymbol{r}$. We shall refer to this approach as 'saturation sparsity' (SS). Although [8, 11] prove RIP of $[\boldsymbol{A}|\boldsymbol{I}]$, this property is only true in an asymptotic sense as $m \to \infty$ (with $n \to \infty$ and $m/n \to 0$). In the realistic regime when m is small, it

has been observed that this technique tends to estimate $\boldsymbol{r}$ as a vector of all zeroes due to the penalty on $|\boldsymbol{r}|_1$.
In recent work by [15], a greedy approximation algorithm is proposed to minimize the following cost function, which is designed to be resilient to measurement outliers:

$$J_\alpha(x) = \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_p + \lambda\|\boldsymbol{x}\|_0, \quad 0 < p < 1 \tag{42}$$

An approximation algorithm to minimize such a cost function is essential, as the $|x|_0$ pseudo-norm renders this problem NP-hard. Note that the approaches in [8, 11, 15] were designed for general impulse noise and not for saturation effects, and thus do not utilize knowledge of the saturation threshold τ.
Recent work in [4] provides theoretical bounds for the following interesting estimator, termed 'noise-cognizant ℓ_1-minimization (NCLM):

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{r}} \quad & |\boldsymbol{x}|_1 \text{ such that} \\ & (i) \quad C(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{r}; -\tau, \tau) = \boldsymbol{y}, \\ & (ii) \quad |\boldsymbol{r}|_2 \leq \gamma\epsilon, \\ & (iii) \quad |\boldsymbol{x}|_2 \leq \gamma'\mu\sqrt{m}. \end{aligned}$$

The parameters γ, γ', and μ need to be selected based on properties of the sensing matrix, ϵ is a bound on $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2$, and the vector $\boldsymbol{r}$ plays the same role as in Eqn. 3. Our method presented in this paper does not require the choice of so many parameters, nor does it require an upper bound on $\|\boldsymbol{x}\|_2$.

Recent work in [14] proposes a modification to the LASSO formulation to account for saturation noise. The data fidelity term, denoted as $\boldsymbol{J}_{csc}(\boldsymbol{y}; \boldsymbol{A}, \tau)$, is given by:

$$\boldsymbol{J}_{csc}(\boldsymbol{x}; \boldsymbol{A}, \boldsymbol{y}) = \frac{1}{2}\sum_{i \in S_{ns}} (\boldsymbol{y} - \boldsymbol{A}^i\boldsymbol{x})^2 + \sum_{i \in S_+} (\tau - \boldsymbol{A}^i\boldsymbol{x})_+^2 + \sum_{i \in S_-} (-\tau - \boldsymbol{A}^i\boldsymbol{x})_-^2 \tag{43}$$

where $(x)_+ = \max(0, x)$ and $(x)_- = -(-x)_+$. The cost function becomes zero if the estimated signal falls outside the saturation levels, as it forces the values in S_+ to be higher than τ and the values in S_- to be lower than $-\tau$. However, this also presents a limitation of the method, as it may not be suitable for accommodating Gaussian noise, as it restricts the signal from adjusting to more continuous and spread-out distributions of Gaussian noise.

5.3 Likelihood Maximization

Our cost function has been motivated from the likelihood function of the additive Gaussian noise. We have

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\eta} \tag{44}$$

The cost function is then given by

$$L(\boldsymbol{x}; \boldsymbol{A}, \boldsymbol{y}) = \lambda\|\boldsymbol{x}\|_1 + \frac{1}{2}\sum_{i \in S_{ns}} \left(\frac{y_i - \boldsymbol{A}^i\boldsymbol{x}}{\sigma}\right)^2 - \sum_{i \in S_+} \log\left(1 - \Phi\left(\frac{\tau - \boldsymbol{A}^i\boldsymbol{x}}{\sigma}\right)\right) - \sum_{i \in S_-} \log\left(1 - \Phi\left(\frac{-\tau - \boldsymbol{A}^i\boldsymbol{x}}{\sigma}\right)\right) \tag{45}$$

5.4 Theoretical Analysis

To derive the error bounds for our cost functions, we state and prove some important results.

Theorem 1. *The cost function $L(\boldsymbol{x}; \boldsymbol{A}, \boldsymbol{y})$ given by Equation (45) is convex.*

We provide the proof for the above theorem and subsequent results in Section B.2. To proceed further, we define the Restricted Strong Convexity property for the cost function. Denote $\hat{x}_\lambda$
A loss function L is said to obey the restricted strong convexity (RSC) property with curvature $\kappa_L > 0$ and tolerance function $\tau_L(x)$ if the Bregman divergence of the cost function given by

$$\delta_L(\Delta, x) = L(y; A\hat{x}_\lambda) - L(y; Ax) - \nabla L(y; Ax)^T(\Delta)$$

satisfies $\delta L(\Delta, x) \geq \kappa_L \|\Delta\|_2^2 - \tau_L^2(x)$ where $\Delta \triangleq \hat{x}_\lambda - x$ and for every vector $\Delta \in \mathcal{C}(S; x)$.

The Bregman divergence term essentially is the error between the loss function value at $\hat{x}_\lambda$ and its first order Taylor series expansion about x.
Intuitively, a loss function that obeys RSC is sharply curved around x, so that any difference in the loss function $\|L(y; Ax) - L(y; A\hat{x}_\lambda)\|$ will imply a proportional estimation error $\|x - \hat{x}_\lambda\|$ for all error vectors $\hat{x}_\lambda - x \in \mathcal{C}(S; x)$.

Subsequently, we have the following lemma from [12].

Lemma 1. *Let $\hat{x}_c$ be the optimum of a general cost function $L(y; Ax) + \lambda\|x\|_1$ with a regularization parameter $\lambda \geq 2\|\nabla L(y; Ax)\|_\infty$. Then the error vector $\Delta = \hat{x}_\lambda - x$ belongs to the set $\mathcal{C}(S; x), \{\Delta | \|(x - \hat{x}_\lambda)\| \leq 3\|x - \hat{x}_\lambda\|$, where S is the set of indices of the s non-zero elements of x, and $\forall i \in S, x_{S(i)} = xi$; $\forall i \notin S, x_{S(i)} = 0$.*

Theorem 2 (Theorem 1 of [12])**.** *If L_g is convex, differentiable, and obeys the RSC property with curvature κ and tolerance $\tau^2(x)$, and if $\hat{x}_\lambda$ is as defined in Lemma 1 with $\lambda \geq 2|\nabla L(y; Ax)|_\infty$, and if x is an s-sparse vector, then we have -*

$$\|\hat{x}_\lambda - x\|_2 \leq \frac{9\lambda^2 s}{\kappa_L^2} + \frac{2\lambda\tau_L^2(x)}{\kappa_L} \tag{46}$$

We now state the following theorems pertaining to the cost function in (45) and prove them in B.2.

Theorem 3. *$L(y, Ax; \tau)$ from Eqn. 8 follows RSC with curvature $\kappa = \frac{\vartheta\gamma}{2}$ and tolerance function $\tau_L^2(x) = 0$, where γ is the restricted eigenvalue constant (REC) for A and ϑ is defined as*

$$\vartheta = \min(1/2, \{f''(\xi_i\}_{i \in S_+}, \{f''(\xi_i\}_{i \in S_-}) \tag{47}$$

The function $f : \mathbb{R} \to \mathbb{R}$ be defined as

$$f(x) = -\log(\Phi(x)) \tag{48}$$

Proving RSC for our cost function implies that we will reach the global minima.

Theorem 4. *For our noise model, the design matrix A and with additional constraints on the signal that $\forall i, X_1 \leq x_i \leq X_2$, we have the lower bound*

$$\|\nabla L\|_\infty \geq \sqrt{\frac{m_{ns}\varrho\log(n)}{m}} - \left(\frac{1}{\sigma}\sum_{i \in S_+} A^i T_+ + \frac{1}{\sigma}\sum_{i \in S_-} A^i T_-\right) \tag{49}$$

with probability $2\exp\{(-1/2(\rho-2)\log(n))\}$ *for constant* $\rho > 2$. *Here,* m_{ns} *denotes the number of unsaturated measurements and* T_+ *and* T_- *are defined as*

$$T_+ = \sqrt{\frac{2}{\pi}}\frac{e^{-v_2^2/2}}{2-\alpha e^{-\beta v_2^2/2}} \tag{50}$$

$$T_- = \sqrt{\frac{2}{\pi}}\frac{e^{-w_2^2/2}}{2-\alpha e^{-\beta w_2^2/2}} \tag{51}$$

we have $\forall i \in S_+, v_i^* \leq v_2$ *and* $\forall i \in S_-, w_i^* \leq w_2$ *where* $v_i^* \triangleq (A^i x - \tau)/\sigma$ *for* $i \in S_+$ *and* $w_i^* \triangleq (-\tau - A^i x)/\sigma$ *for* $i \in S_-$.

In the above expressions, we have the bounds corresponding to S_+ and S_- of the order $\mathcal{O}(\sqrt{m_+/m})$ and $\mathcal{O}(\sqrt{m_-/m})$ respectively.
We develop this lower bound for $\|\nabla L\|_\infty$ so that we can apply 2 to find the upper bound on the reconstruction error.

Theorem 5. *Let* $\hat{x}_\lambda$ *be the minimiser of the cost function in 45 with regularization parameter* $\lambda \geq 2\|\nabla L\|_\infty$ *and with the signal constraints from Theorem 4. Let* x *be the true* s*-sparse signal. Then we have the following upper bound with the same probability as in Theorem 4 -*

$$|x_c - x|_2 \leq \frac{144 s \log(n)\sigma^2\rho}{\gamma^2 m}(\sqrt{m_{ns}} + C\sqrt{m_+ + m_-}) \tag{52}$$

The upper bound is directly proportional to $s\log(n)$, which is equivalent to the upper bound in Lasso reconstruction. So, the tightness of the upper bound on the reconstruction error of our cost function is relatively close to that of Lasso reconstruction. The bound is directly proportional to σ^2 as well as $s = |x|_0$ and inversely proportional to $\gamma = REC(A; s)$ [13, 5], all of which is intuitive. The bound relaxes with an increase in the number of saturated measurements m_+ and m_-. If there are no saturated measurements, i.e., $m_- = m_+ = 0$, then the bound reduces to the normal LASSO bound [5], except that here we consider A with unit column norm as against column norm of m. The bound also increases with m_{ns}. However, it turns out that the constant factor C_1 for the $O(m_1 + m_2)$ term in the bounds is very large. This is because it contains other factors of the form $\varphi(z)$ or $\Phi(z)$, where z stands for either X_1 or X_2, which are both large in absolute value. Hence the $\mathcal{O}(m_+ + m_-)$ term dominates over the $O(m_{ns})$ term, which is intuitive.

6 Experiments

6.1 Signal Reconstruction

We conducted experiments to evaluate the performance of our algorithm for sparse signal reconstruction by comparing it with five other methods: (i) Saturation Rejection; (ii) Saturation Ignorance; (iii) Saturation Consistency; (iv) Noise Cognizant L1 Minimization; and (v) Consistent Sparse Coding.

To generate measurements, we randomly drew the elements of the sensing matrix A from $\mathcal{N}(0, \frac{1}{m})$ so that A would satisfy the Restricted Isometry Property (RIP) with high probability, and applied it on sparse signals of dimension $n = 256$. We then added additive noise independently to each patch and applied the saturation operator $\mathcal{C}$ to the patch. We conducted experiments to analyze the variation in the algorithm's performance with respect to three factors: (A) number of measurements m per patch, (B) noise standard deviation σ expressed as a fraction $f_\sigma \in [0, 1]$ of ζ, (C) the sparsity fraction $f_{sp} \in [0, 1]$ and (D) the fraction $f_{sat} \in [0, 1]$ of measurements that are saturated. Each experiment has been repeated over 50 independent signals and measurement matrices.

For the measurements experiment (A), we varied m in the range of $\{50, 80, 110, \ldots, 200\}$ with $sp = 25$, $f_{sat} = 0.15$ and $f_\sigma = 0.1$. For the Gaussian noise experiment (B), we varied f_σ in the range of $\{0.01, 0.04, 0.07, \ldots, 0.19\}$ with fixed $m = 180$, $sp = 25$ and $f_{sat} = 0.10$. For the sparsity experiment (C), we varied f_{sp} in the range of $\{0.05, 0.075, \ldots, 0.2\}$ with $m = 180$, $f_sigma = 0.05$, and $f_sat = 15\%$. For the saturation experiment (D), we varied f_{sat} in the range of $\{0, 0.1, 0.2, \ldots, 0.5\}$ with fixed $m = 180$ and $f_\sigma = 0.1$.

To choose the regularization parameters, we employed cross validation with the unsaturated measurements. The validation set consisted of 20% of the measurements, and we performed cross validation with a probability of 40% for each patch. The algorithms were implemented using the `cvx` package in `MATLAB`.

6.2 Image Reconstruction

We evaluated the performance of our algorithm for image reconstruction by comparing it with five other methods: (i) Saturation Rejection; (ii) Saturation Ignorance; (iii) Saturation Consistency; (iv) Noise Cognizant L1 Minimization; and (v) Consistent Sparse Coding. We used the Berkeley Image Segmentation Dataset and divided each image into non-overlapping patches of size 16×16, equivalent to a signal with $n = 256$. Each patch was reconstructed individually, as the images are sparse in the 2D-DCT (Discrete Cosine Transform) basis.

To generate measurements, we randomly drew the elements of the sensing matrix A from $\mathcal{N}(0, \frac{1}{m})$ so that A would satisfy the Restricted Isometry Property (RIP) with high probability. We then added additive noise independently to each patch and applied the saturation operator $\mathcal{C}$ to the patch. We conducted experiments to analyze the variation in the algorithm's performance with respect to three factors: (A) number of measurements m per patch, (B) noise standard deviation σ expressed as a fraction $f_\sigma \in [0, 1]$ of ζ, and (C) the fraction $f_{sat} \in [0, 1]$ of measurements that are saturated.

For the measurements experiment (A), we varied m in the range of $50, 80, 110, \ldots, 200$ with fixed $f_{sat} = 0.15$ and $f_\sigma = 0.1$. For the Gaussian noise experiment (B), we varied f_σ in the range of $0.01, 0.04, 0.07, \ldots, 0.19$ with fixed $m = 180$ and $f_{sat} = 0.10$. For the saturation experiment (C), we varied f_{sat} in the range of

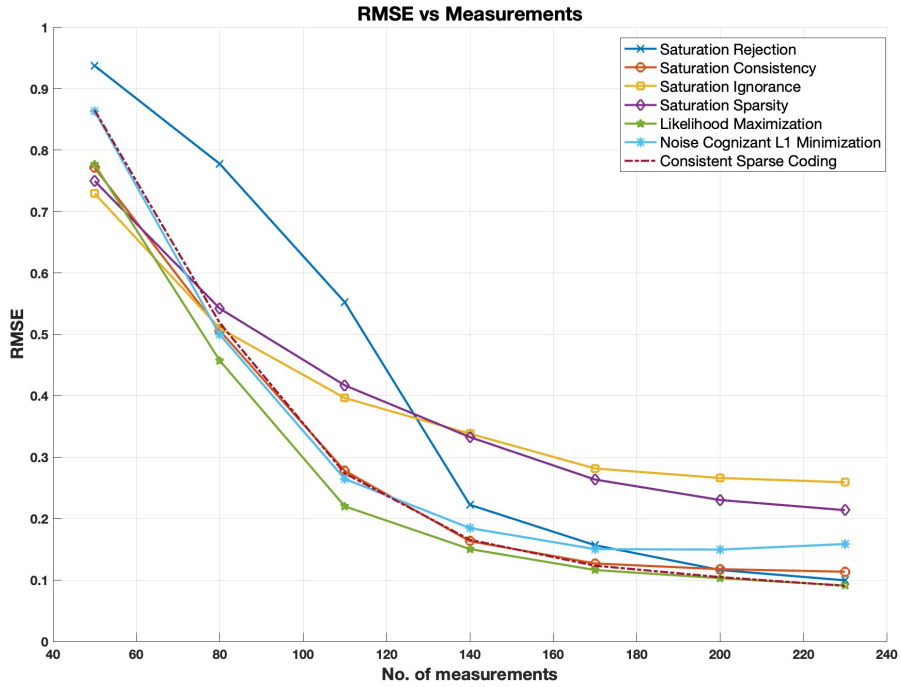


Figure 9: Signal Reconstructions Performance Summary with varying no. of measurements

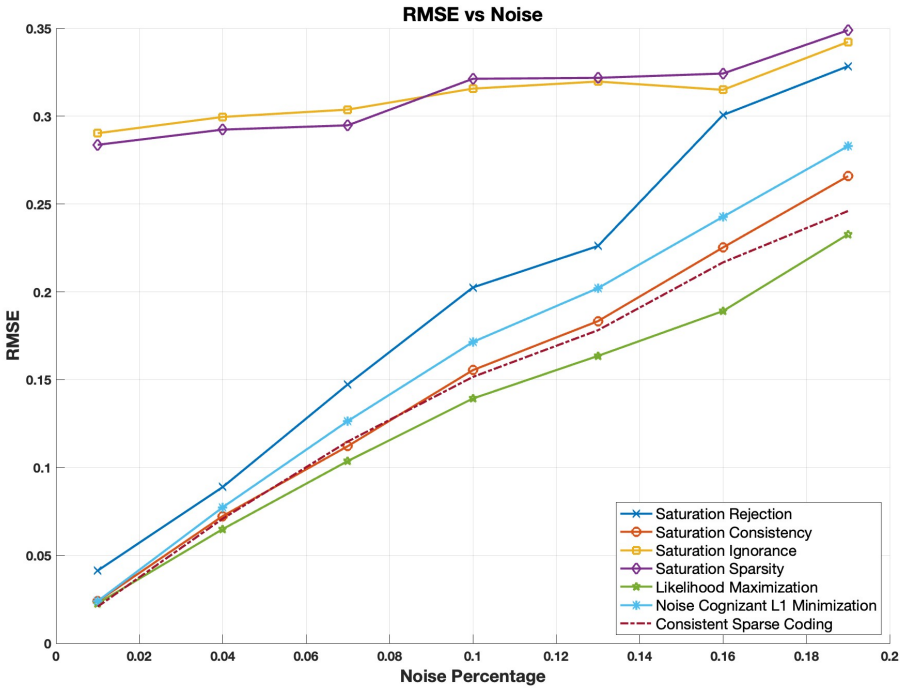


Figure 10: Signal Reconstructions Performance Summary with varying noise

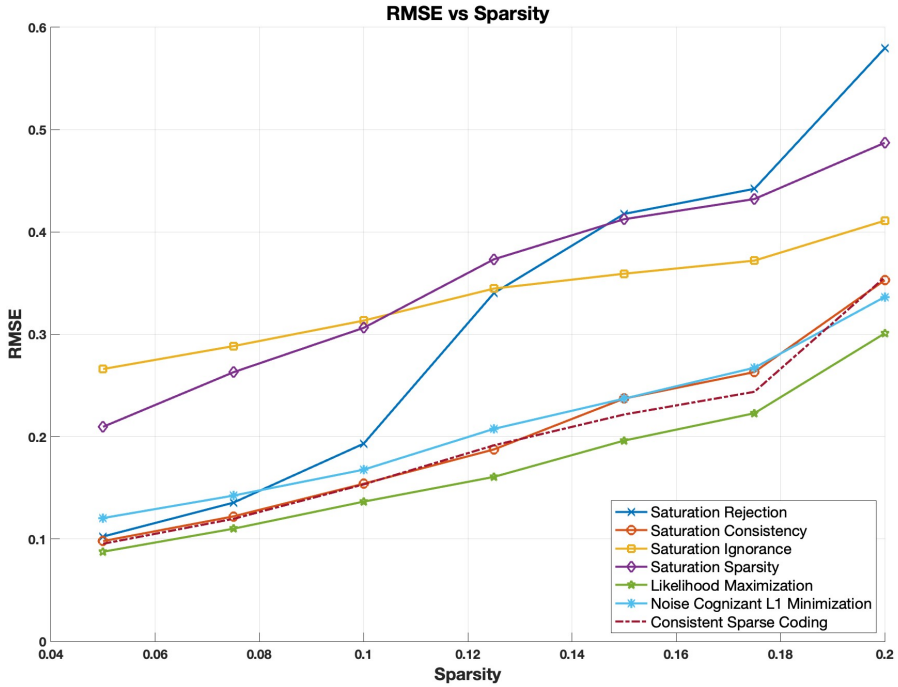


Figure 11: Signal Reconstructions Performance Summary with varying sparsity

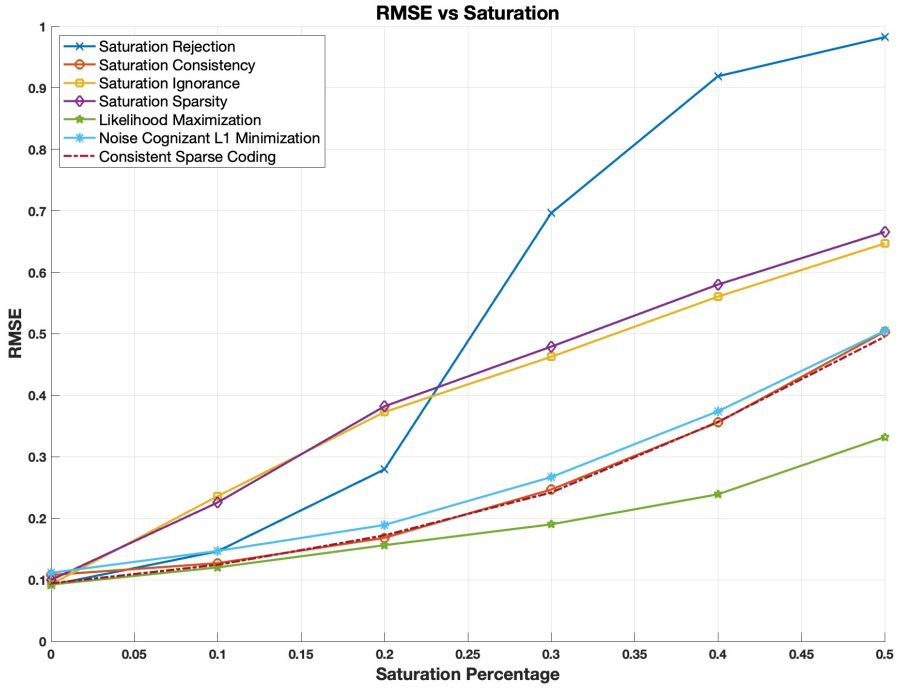


Figure 12: Signal Reconstructions Performance Summary with varying saturation

$0, 0.1, 0.2, \ldots, 0.5$ with fixed $m = 180$ and $f_\sigma = 0.1$.

To choose the regularization parameters, we employed cross validation with the unsaturated measurements. The validation set consisted of 20% of the measurements, and we performed cross validation with a probability of 40% for each patch. The algorithms were implemented using the `cvx` package in `MATLAB`.

6.3 Audio Declipping

We have also performed experiments to reconstruction audio signals from clipped measurements. Unlike the previous experiments, this application does not fall under the domain of compressed sensing. We have a clipped input signal which is compressive in DCT, and we try to reconstruct the unclipped signal. To measure the amount of saturation in the signal, we use the metric SDR defined by

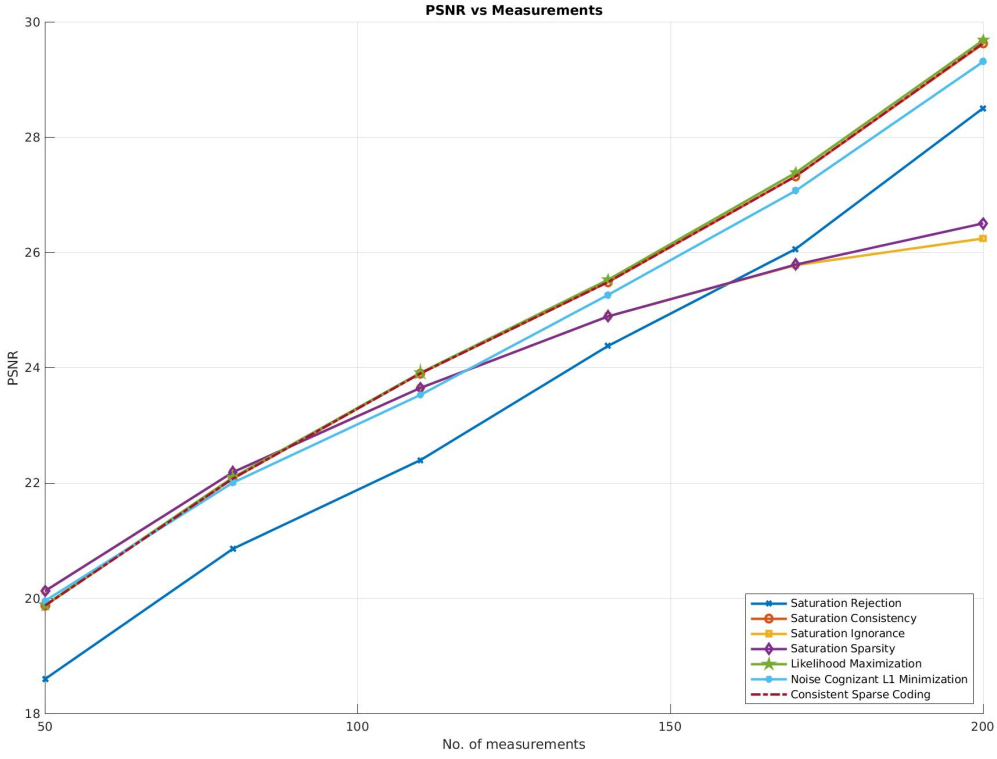
$$\mathrm{SDR}(\hat{\boldsymbol{x}}, \boldsymbol{x}) = 20 \log \frac{\|\boldsymbol{x}\|_2^2}{\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2} \tag{53}$$

where $\hat{x}$ is the noisy signal, and $\boldsymbol{x}$ is the true signal.

To reconstruct the signal $\hat{\boldsymbol{x}}$, we take the clipped signal of length T and divide it into n segments, each of length m, with overlaps between the adjacent segments, using a sliding window. For example, if the clipped signal has 65536 samples, we can form x as a matrix 256×511, using a sliding window of $m = 256$ samples with a hop size equal to 128. In the experiments, the clipping level τ is set using a routine from [9], which takes an input SDR and outputs the clipping level. The clean signal was normalized before Gaussian noise was added and subsequently being clipped at the specified level.

We compare our data-fidelity function with the CSC fidelity term using a music signal "mono-8000hz.mp3". This signal contain single channel recordings, sampled at 8 kHz. The input SNR is varied from $\{4, 6, 8, 10, 12, 15, 20\}$, and the results are displayed in Figure 16, 17.

Figures 18, 19 show how the waveform changes after clipping and after reconstruction with each method. I had also explored dictionary learning techniques to learn the dictionary for the audio data on the fly. The SAD approach as proposed by [9] uses a transfer learning framework with the CSC cost function. It is extremely efficient as the iterations use a closed-form solution for the sparse-coding step. I designed a dictionary learning procedure for both the data-fidelity terms but they did not yield as good results as expected.



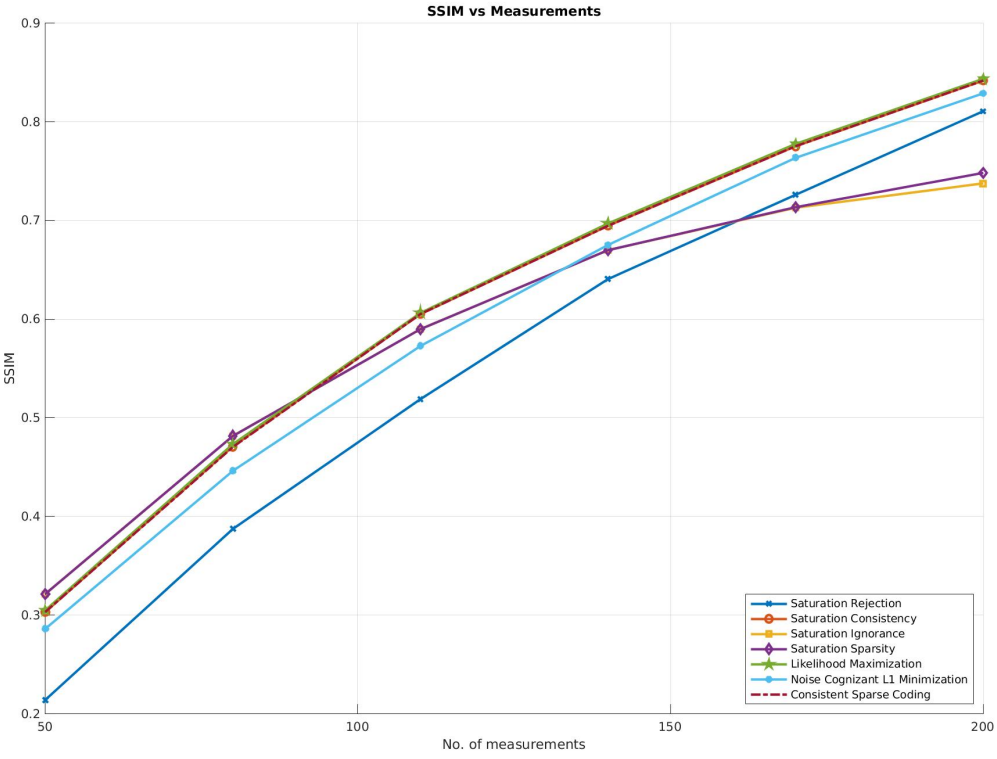
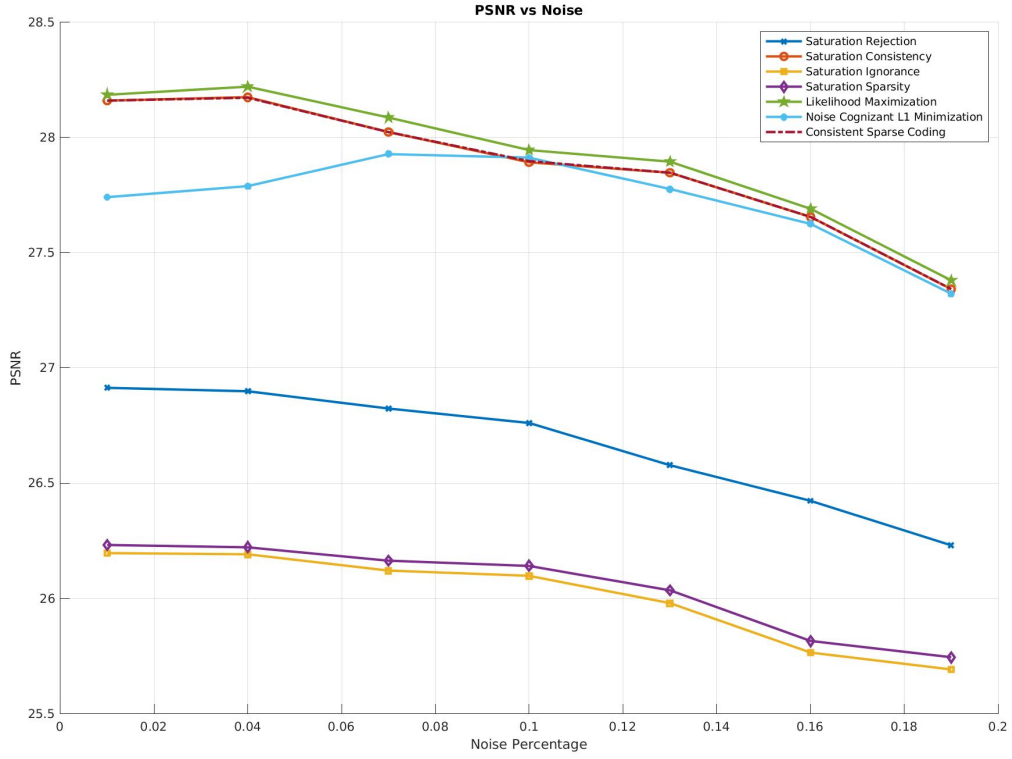


Figure 13: Image Reconstructions Performance Summary with varying number of measurements



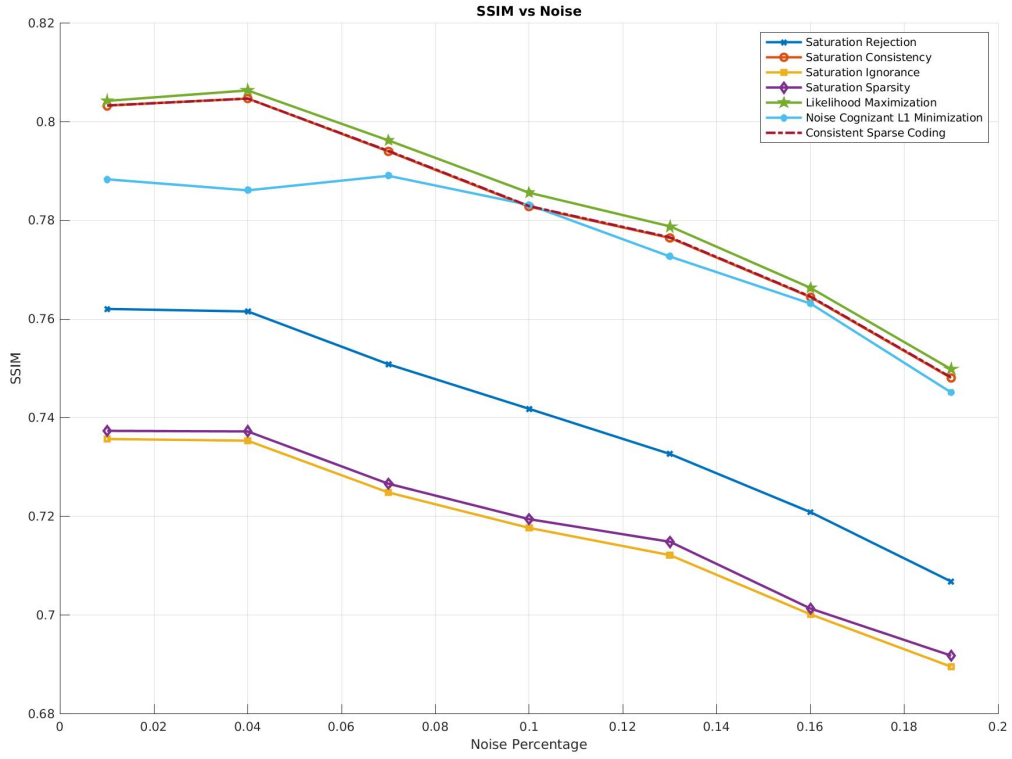
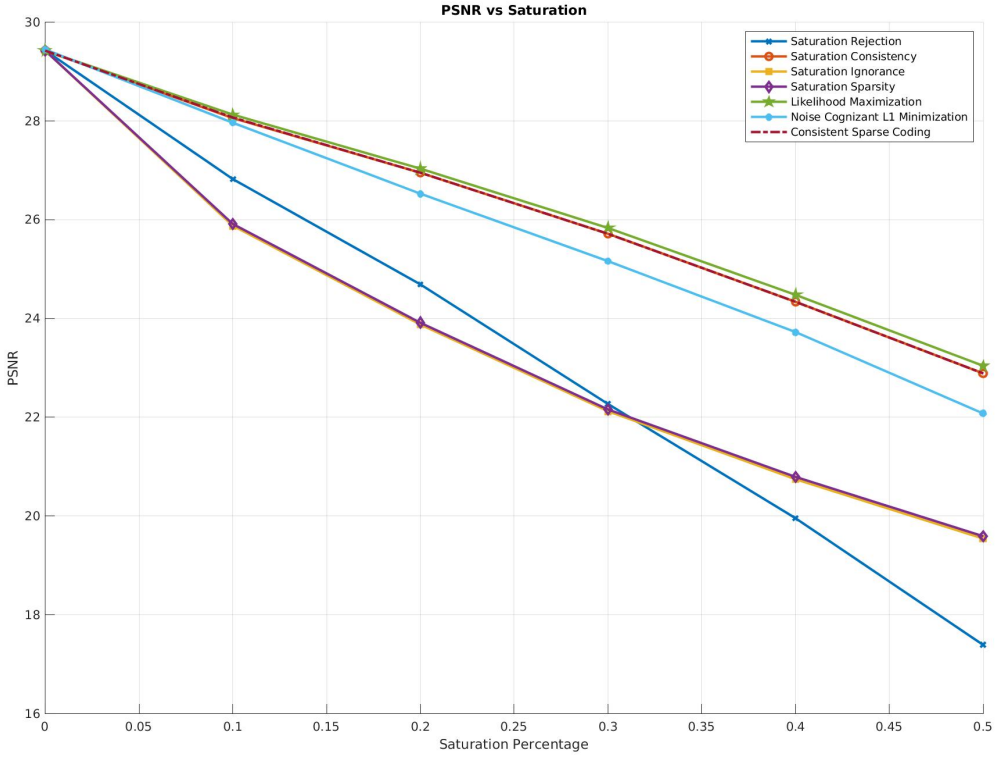


Figure 14: Image Reconstructions Performance Summary with varying additive noise



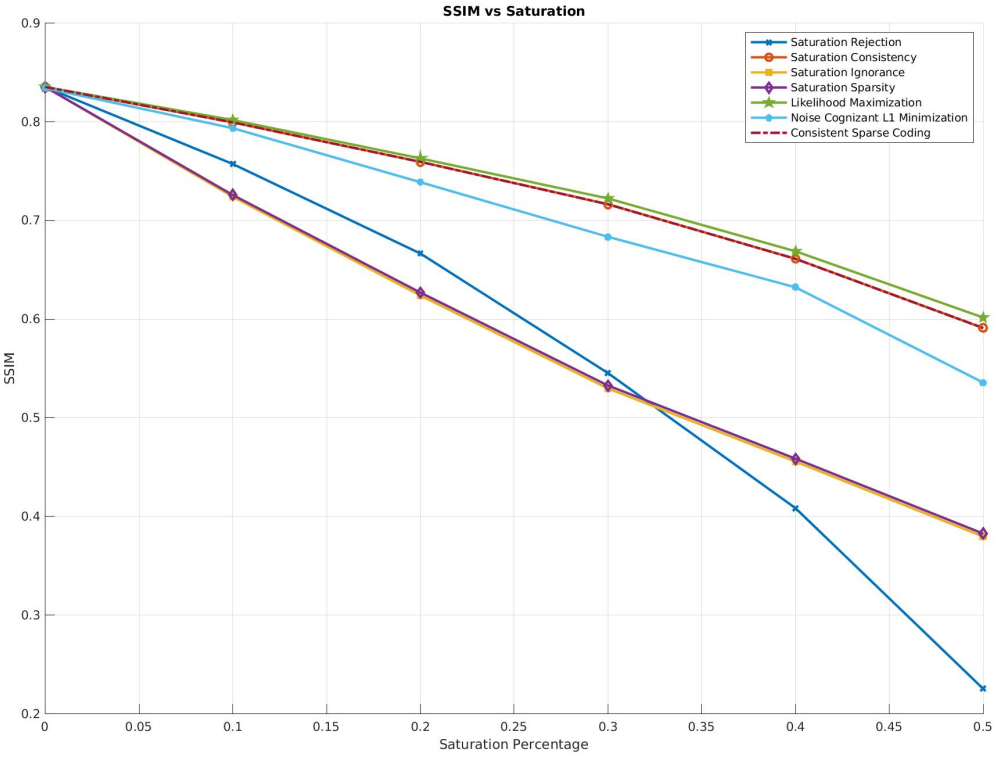


Figure 15: Image Reconstructions Performance Summary with varying saturation

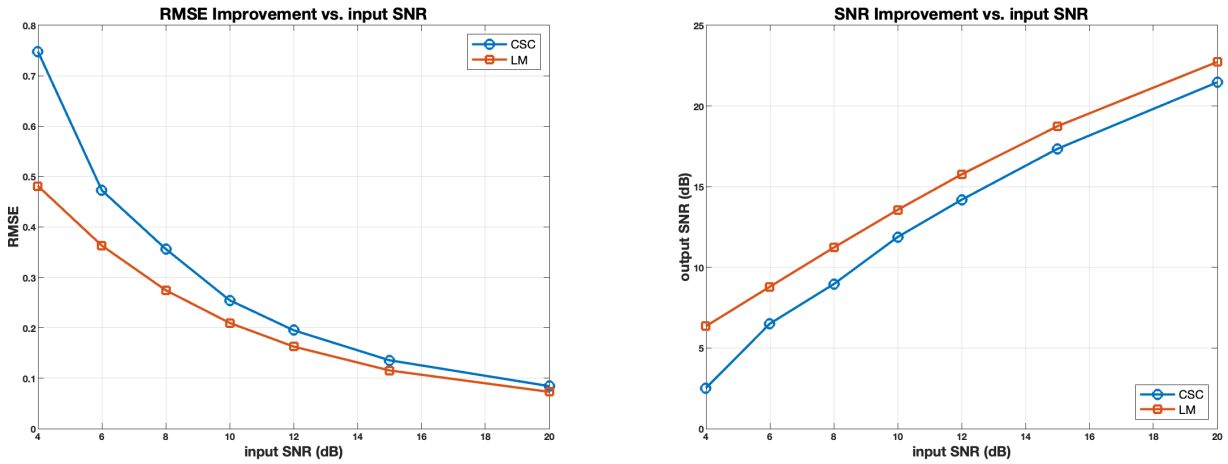


Figure 16: Audio Declipping Performance Summary with varying input SNR

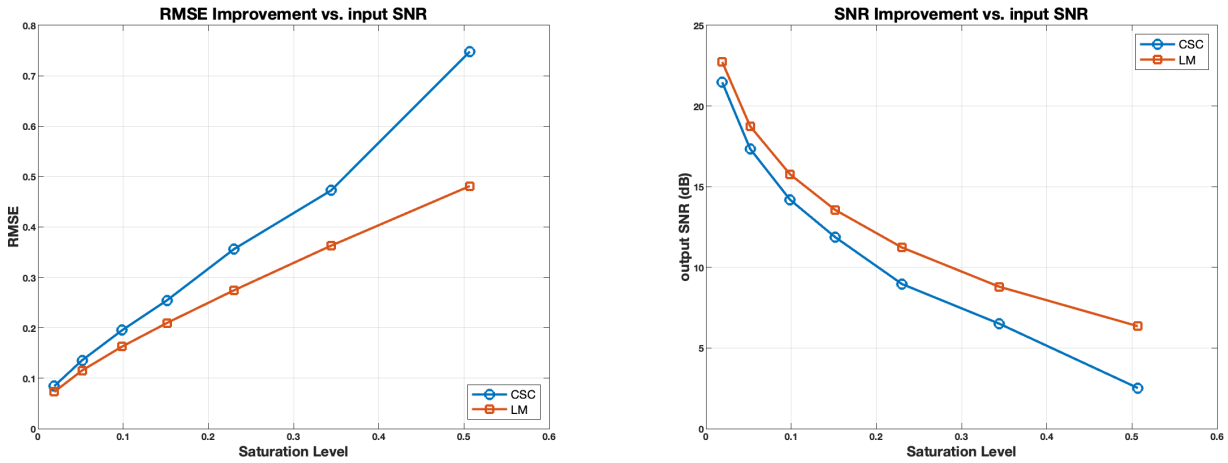


Figure 17: Audio Declipping Performance Summary with varying amount of Saturation

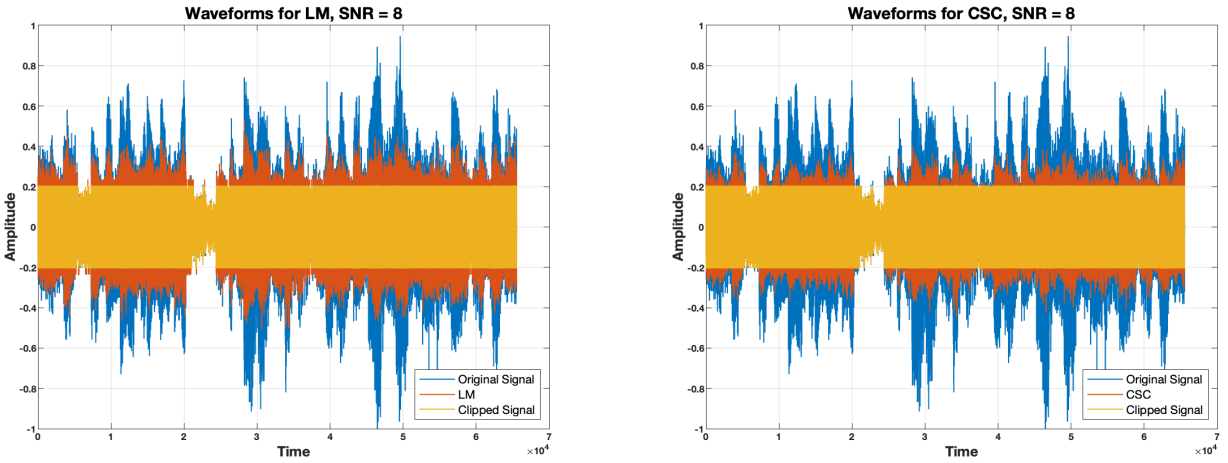


Figure 18: Complete Audio Waveform after clipping and reconstruction with both methods

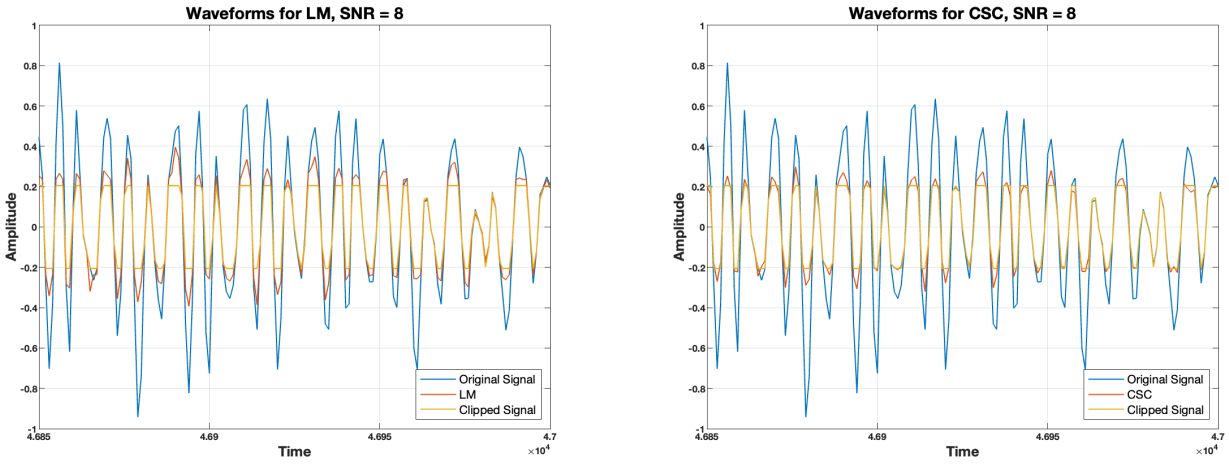


Figure 19: Audio Waveform from $t = 46850$ to $t = 47000$ after for both methods

A Details for bit-flip noise

A.1 Problem Formulation

The representation of the group-testing scenario is done as follows - Let $b \in \mathbb{R}^n$ be the vector of samples of n individuals. Let there be m groups. The presence of the ith sample in the jth group is represented by a binary vector - the ith element is set to 1 and the rest are set to 0 in a vector of length n.

Once the groups are made, all the samples in a group are mixed together and tested. This setup is given by a pooling matrix $B \in \mathbb{R}^{m \times n}$ whose ith row has the members in the ith group in binary representation.

$$B^{ij} \triangleq \begin{cases} 1 & \text{if } i\text{th sample belongs to } j\text{th group} \\ 0 & \text{otherwise} \end{cases} \tag{54}$$

That is, the pooling matrix B is a Bernoulli sampled matrix (note that, such matrices generally follow RIP). The grouped samples vector z along with the noise η' is then given by

$$z = Bb + \eta' \tag{55}$$

The noise associated with infection samples in COVID is modeled as a multiplicative Gaussian noise which is dependent on the value of the measurements. However, we will assume that it is independent Gaussian additive noise for simplicity.

Now, a bit-flip occurs when a member belonging to one group is switched with an other member who does not belong to the group. In such a case, the measurements we obtain are given by

$$\begin{aligned} z &= \tilde{B}b + \eta' \\ &= (B + \Delta B)b + \eta' \\ &= Bb + \delta' + \eta' \end{aligned}$$

where $\tilde{B}$ is the observed pooling matrix. Let $i, j \in [n]$ be such that i belongs to the kth group and j does not belong in the group. Then, a bit-flip is said to occur if the technician places j in the group instead of i. We have $B^{ik} = 1, B^{jk} = 0$ and $\tilde{B}^{ik} = 0, \tilde{B}^{jk} = 1$.

A.2 Centering the pooling matrix

The method we develop for correcting for bit-flip errors uses the Huber Loss function and debiasing of LASSO. These results require the pooling matrix to have zero mean. We obtain the zero mean matrix using the $A^i = 2B^i - 1^t$. Then,

$$\begin{aligned} y_i = 2(z_i - \bar{z}_i) &= 2(\tilde{B}^i b + \eta_i') - b \\ &= (2B^i - 1^t)b + 2\delta_i' + 2\eta_i' \\ &= A^i b + \delta_i + \eta_i \end{aligned}$$

where y_i is the ith zero-centered measurement. The elements of the matrix A are either 1 or -1.

A.3 Expectation of the bit-flip noise

The bit-flip noise is mathematically denoted as $\delta \triangleq (\tilde{A} - A)b = \Delta A b$ where $\tilde{A}$ is the nominal centered matrix after the bit-flips have occurred and A is the true centered matrix. We make the following assumptions to derive the distribution of δ.

- **(A1)** There are at most $t \ll m$ bit-flip errors in the pooling matrix
- **(A2)** The probability of a bit-flip occuring in each row is independent of the other rows and is uniform across all rows

Suppose there is a single bit-flip between the columns i and j on the kth row. Then, the probability of this happening is given by

$$P(\text{flip of } i \text{ and } j \text{ on } k\text{th row}) = \frac{1}{m \times \binom{n}{2}} \tag{56}$$

This is because every row and every pair of samples have an equal probability of having a flip. The structure of ΔA in this case would be

$$\Delta(A)_{i_1 j_1} = \begin{cases} 0 & i_1 j_1 \neq ik \text{ and } i_1 j_1 \neq jk \\ -2A_{i_1 j_1} & \text{otherwise} \end{cases} \tag{57}$$

We then get

$$E[\Delta A|A] = \frac{1}{m \times \binom{n}{2}} \times -2A \tag{58}$$

In the presence of t bit-flips, the expectation becomes

$$E[\Delta A|A] = \frac{-2t}{m \times \binom{n}{2}} \times A \tag{59}$$

We shall state the following result without proof -

Lemma 2. *Under the conditions that $t \ll n$, $\|b\|_2 \leq \varrho_2$ for some $\varrho_2 > 0$ and $0 \leq C_{min} \leq \lambda \min(E(A^T A)) \leq \max(E(A^T A)) \leq C_{max} < \infty$, we have $\|E(A^T(\delta + \eta))\|_2$ is $o(1/m)$*

A.4 Huber Loss formulation

The equation of centered measurements is given by

$$y = Ab + \delta + \eta \tag{60}$$

This equation can be modified as

$$y = [A|I_{m\times m}]x + \eta \tag{61}$$

where x is the concatenation of the vectors b and δ. The LASSO estimate for this equation is given by

$$\tilde{x} \triangleq \arg\min_x \|y - (A|I_{m\times m})x\|_2^2 + \|b\|_1 + \frac{1}{\alpha}\|\delta\|_1 \tag{62}$$

This can be rewritten as follows. We are trying to optimise

$$\begin{aligned} &\min_{b,\delta} \|y - (A|I_{m\times m})x\|_2^2 + \|b\|_1 + \frac{1}{\alpha}\|\delta\|_1 \\ =&\min_b \left\{ \|b\|_1 + \left\{ \min_\delta \|(y - Ax) - \delta\|_2^2 + \frac{1}{\alpha}\|\delta\|_1 \right\} \right\} \\ =&\min_b \left\{ \|b\|_1 + \frac{1}{m}\sum_{i=1}^{m} H_\alpha(y_i - A^i b) \right\} \end{aligned}$$

This gives us the Huber loss formulation. We then have the following result.

Lemma 3. *If Lemma 2 holds along with*

1. $E\left\{E(|\delta+\eta|^k|A)\right\} \leq M_k < \infty$, *for some* $k \geq 2$
2. $0 \leq C_{min} \leq \lambda \min(E(A^TA)) \leq \lambda \max(E(A^TA)) \leq C_{max} < \infty$
3. *For any* $v\mathbb{R}^p$, A^Tv *is sub-Gaussian with parameter atmost* κ_0^2,

then there exists a universal constant C *such that,*

$$\|b_\alpha^* - b\|_2 \leq \frac{2\alpha^{k-1}}{k-1}[\sqrt{M}k + C\kappa_0^k]\frac{\sqrt{C}_{max}}{C_{min}} + \frac{4s}{\binom{n}{2}m}\frac{C_{max}}{C_{min}}\|b\|_2 \tag{63}$$

where $b_\alpha^* = \arg\min_b E\{H_\alpha(y - Ab)\}$

A.5 Debiasing LASSO for δ

The LASSO estimator has a bias incurred due to the $l1$ penalty term. [6] provides a method to remove the bias and upon eliminating the bias the debiased lasso estimator has a normal distribution. We will use this property to perform Hypothesis testing to detect the bit-flip errors.

Let $x_\lambda \triangleq \arg\min_x \|y - (A|I_{m\times m})x\|^2 + \lambda\|x\|_1$. Also, $\hat{\Sigma} \triangleq (A|I_{m\times m})^T(A|I_{m\times m})/m$. We then have the following result -

Lemma 4. *For a chosen inverse* $M \in \mathbb{R}^{(m+n)\times(m+n)}$, *we have,*

$$\sqrt{m}(x_d - x^*) = Z + \Delta \tag{64}$$

where,

$$Z \sim \mathcal{N}(0, \sigma^2 M\Sigma M^T) \tag{65}$$

$$\Delta = \sqrt{m}(M\Sigma - I)(x^* - x_\lambda) \tag{66}$$

Z and Δ denote the bias in x_λ in the above equations. Upon using the structure of M stated in equation 24, we further get the following results assuming some conditions that are omitted here for simplicity -

Lemma 5. *Suppose* $m > (s\log n)^2$ *and* $t < \sqrt{n}$.
Define the following -

1. $Z_1 \triangleq M_1A^T\eta/\sqrt{m}$ *and* $Z_2 \triangleq \sqrt{m}\eta$ *such that* $Z = (Z_1^T Z_2^T)^T$
2. $\Delta_1 \triangleq \sqrt{m}(b_d - b^*) - Z_1$ *and* $\Delta_3 \triangleq \sqrt{m}(\delta_d - \delta^*) - Z_2 = \sqrt{m}A(b^* - b_\lambda)$ *such that* $\Delta = \Delta_1 + \Delta_3$.

then we have

1. Δ_1 *is asymptotically smaller than* Z_1 *as* Z_1 *has the order* $O_p(1)$ *whereas* Δ_1 *has the order* $o_p(1)$. *Under asymptotic conditions,*
$$\sqrt{m}(b_d - b^*) = Z_1 \tag{67}$$
We get these results using Lemma 3
2. *Define* $\Delta_{32} \triangleq \sqrt{m}A(b_d - b^*) = Z_1 + \Delta_1$. *Since* Δ_1 *is negligible wrt* Z_1, *we asymptotically have* $\Delta_{32} = Z_1$.
3. *As a result, we have*
$$Z_2 + \Delta_{32} \sim \mathcal{N}(0, \sigma^2\Sigma_\delta) \tag{68}$$
where $\Sigma_\delta = \left(\sqrt{m}I + AM_1A^T/\sqrt{m}\right)\left(\sqrt{m}I + AM_1A^T/\sqrt{m}\right)^T$

B Saturation Noise Bounds

B.1 Convexity

For the ease of proving results, we shall use the following notation for the cost function.

$$L(x; A, y) = L_1(x; A, y) + L_2(x; A, y) + L_3(x; A, y) \tag{69}$$

where

$$L_1(x; A, y) = \sum_{i \in S_{ns}} \frac{(y_i - A^i x)^2}{2\sigma^2} \tag{70}$$

$$L_2(x; A, y) = -\sum_{i \in S_+} \log\left(1 - \Phi\left(\frac{\tau - A^i x}{\sigma}\right)\right) \tag{71}$$

$$L_3(x; A, y) = -\sum_{i \in S_-} \log\left(\Phi\left(\frac{-\tau - A^i x}{\sigma}\right)\right) \tag{72}$$

To prove that the cost function is convex, we will show that each of these cost functions is convex by showing that the corresponding Hessian matrices are positive semi-definite. The cost function L_1 is convex because $\partial L_1/\partial x \partial x^T = \sum_{i \in S_{ns}} A^i (A^i)^T$, which is positive semi-definite. Denote $v_i = (A^i x - y_i)/\sigma$. Now, we have the following

$$\frac{\partial L_2(x)}{\partial x} = \frac{-1}{\sigma} \sum_{i \in S_+} \frac{A^i \phi(v_i)}{\Phi(v_i)} \tag{73}$$

This further gives -

$$\frac{\partial L_2(x)}{\partial x \partial x^T} = \frac{1}{\sigma^2} \sum_{i \in S_+} A^i (A^i)^T \phi(v_i) \left(\frac{\phi(v_i) + v_i \Phi(v_i)}{\Phi(v_i))^2}\right) \tag{74}$$

In the above expression, the terms $A^i (A^i)^T$, $\phi(v_i)$ and $\Phi(v_i))$ are positive. It will suffice to show that the numerator in the expression is positive. Denote $h(v) = \phi(v) + v\Phi(v)$. Then, $h'(v) = -v\phi(v) + \Phi(v) + v\phi(v) = \Phi(v) > 0$. Also, $\lim_{v \to -\infty} h(v) = 0$ as both $\phi(v)$ and $\Phi(v)$ go to 0 as v decreases and the rate of convergence of $\Phi(v) \to 0$ is faster than that of $v \to \infty$. Since $h(v)$ is an increasing function bounded below by 0, we have $h(v) \geq 0$ for all $v \in \mathbb{R}$. Consequently, this establishes that L_2 is convex.

The convexity of L_3 can be established in a similar manner. Define $w_i = -A^i x - \tau$. We have,

$$\frac{\partial L_3(x)}{\partial x \partial x^T} = \frac{1}{\sigma^2} \sum_{i \in S_-} A^i (A^i)^T \phi(w_i) \left(\frac{\phi(w_i) + w_i \Phi(w_i)}{\Phi(w_i))^2}\right) \tag{75}$$

As proved above, this function is positive as well, implying that L_2 is convex. Finally, since L_1, L_2, and L_3 are convex, the cost function L is convex. $\square$

B.2 Restricted Strong Convexity

We will now show the property of restricted strong convexity of the cost function. To do so, we state the definition of Bregman divergence of a cost function L with respect to a $\Delta \in \mathbb{R}^n$

$$\delta L(\boldsymbol{x}^*, \boldsymbol{\Delta}) = L(\boldsymbol{x}^* + \boldsymbol{\Delta}) - L(\boldsymbol{x}^*) - \langle \boldsymbol{\Delta}, \nabla L(\boldsymbol{x}^*) \rangle$$

The cost function L is said to follow restricted strong convexity if the following holds

$$\delta L(\boldsymbol{x}^*, \boldsymbol{\Delta}) \geq \kappa_L \|\boldsymbol{\Delta}\|_2^2 - \tau_L^2(\boldsymbol{x}^*) \tag{76}$$

for the curvature term $\kappa_L > 0$ and a positive tolerance function τ_L for $\boldsymbol{\Delta} \in \mathcal{C}$ such that
$\mathcal{C} \triangleq \{\boldsymbol{\Delta} : \|\boldsymbol{\Delta}_{S^c}\|_1 \leq 3\|\boldsymbol{\Delta}_S\|_2 + 4\|\boldsymbol{x}^*_{S^c}\|_1\}$. Here S stands for the set of indices of the s largest entries of the true signal $\boldsymbol{x}^*$, and S^c is the complement of S. Since we are dealing with purely sparse signals, S would correspond to the support of $\boldsymbol{x}^*$ and $\boldsymbol{x}^*_{S^c} = 0$. However, extensions to weakly sparse signals are straightforward.

B.3 Bregman divergence calculation

We have the following relation

$$\delta L(\boldsymbol{x}^*, \boldsymbol{\Delta}) = \delta L_1(\boldsymbol{x}^*, \boldsymbol{\Delta}) + \delta L_2(\boldsymbol{x}^*, \boldsymbol{\Delta}) + \delta L_3(\boldsymbol{x}^*, \boldsymbol{\Delta})$$

We shall derive expressions of each of the terms independently.

B.3.1 Bregman divergence of L_1

We have

$$L_1(x; A, y) = \sum_{i \in S_{ns}} \frac{(A^i x - y_i)^2}{2\sigma^2} \tag{77}$$

Therefore,

$$\nabla L_1(x; A, y) = \sum_{i \in S_{ns}} \frac{(A^i x - y_i)}{\sigma} A^i \tag{78}$$

We then get,

$$\begin{aligned} \delta L_1(x, \Delta) &= \sum_{i \in S_{ns}} \left[\frac{(A^i(x+\Delta) - y_i)^2}{2\sigma^2} - \frac{(A^i x - y_i)^2}{2\sigma^2} - \frac{A^i \Delta}{\sigma}(A^i x - y_i) \right] \\ &= \sum_{i \in S_{ns}} \frac{\|A^i \Delta\|^2}{2\sigma^2} \\ &= \sum_{i \in S_{ns}} \frac{k_i^2}{2} \end{aligned} \tag{79}$$

where $k_i = A^i \Delta / \sigma$ for $i \in S_{ns}$.

B.3.2 Bregman divergence of L_2

We have

$$\begin{aligned} L_2(x; A, y) &= -\sum_{i \in S_+} \log\left(1 - \Phi\left(\frac{\tau - A^i x}{\sigma}\right)\right) \\ &= -\sum_{i \in S_+} \log\left(\Phi\left(\frac{A^i x - \tau}{\sigma}\right)\right) \end{aligned} \tag{80}$$

Therefore,

$$\nabla L_2(x; A, y) = -\sum_{i \in S_+} \frac{\phi\left(\frac{A^i x - \tau}{\sigma}\right)}{\Phi\left(\frac{A^i x - \tau}{\sigma}\right)} A^i \tag{81}$$

The expression for $\delta L_2(x, \Delta)$ is given by,

$$\begin{aligned} \delta L_2(x, \Delta) = \sum_{i \in S_+} \Bigg[&- \log\left(\Phi\left(\frac{A^i(x+\Delta) - \tau}{\sigma}\right)\right) + \log\left(\Phi\left(\frac{A^i x - \tau}{\sigma}\right)\right) \\ &+ \frac{A^i \Delta}{\sigma} \frac{\phi\left(\frac{A^i x - \tau}{\sigma}\right)}{\Phi\left(\frac{A^i x - \tau}{\sigma}\right)} \Bigg] \end{aligned} \tag{82}$$

To simplify the notation, define $v_i = (A^i x - \tau)/\sigma$ and $k_i = A^i \Delta / \sigma$.
Let $v = [v_1, \ldots, v_m]$ and $k = [k_1, \ldots, k_m]$. We then have,

$$\delta L_2(v, k) = \sum_{i \in S_+} -\log(\Phi(v_i + k_i)) + \log(\Phi(v_i)) + k_i \frac{\phi(v_i)}{\Phi(v_i)} \tag{83}$$

We shall use Taylor's series expansion to simplify this expression. Let the function $f : \mathbb{R} \to \mathbb{R}$ be defined as follows

$$f(x) = -\log(\Phi(x)) \tag{84}$$

The Taylor series expansion of the function for $d \in \mathbb{R}$ is given by

$$\begin{aligned} f(x+d) &= f(x) + df'(x) + \frac{d^2}{2} f''(\xi) \\ -\log(\Phi(x+d)) &= -\log(\Phi(x)) - d\frac{\phi(x)}{\Phi(x)} + \frac{d^2}{2} \frac{\phi(\xi)}{\Phi(\xi)} \left(\frac{\phi(\xi)}{\Phi(\xi)} + \xi\right) \end{aligned} \tag{85}$$

for $\xi \in [x, x+d]$. Applying the expansion at v_i and $v_i + k_i$ for each $i \in S_+$, we get

$$\begin{aligned} -\log(\Phi(v_i + k_i)) &= -\log(\Phi(v_i)) - k_i \frac{\phi(v_i)}{\Phi(v_i)} + \frac{k_i^2}{2} \frac{\phi(\xi_i)}{\Phi(\xi_i)} \left(\frac{\phi(\xi_i)}{\Phi(\xi_i)} + \xi_i\right) \\ -\log(\Phi(v_i + k_i)) + \log(\Phi(v_i)) + k_i \frac{\phi(v_i)}{\Phi(v_i)} &= \frac{k_i^2}{2} \frac{\phi(\xi_i)}{\Phi(\xi_i)} \left(\frac{\phi(\xi_i)}{\Phi(\xi_i)} + \xi_i\right) \end{aligned} \tag{86}$$

Summing across all $i \in S_+$,

$$\begin{aligned}\sum_{i\in S_+}\left[-\log(\Phi(v_i+k_i))+\log(\Phi(v_i))+k_i\frac{\phi(v_i)}{\Phi(k_i)}\right] &= \sum_{i\in S_+}\left[\frac{k_i^2}{2}\frac{\phi(\xi_i)}{\Phi(\xi_i)}\left(\frac{\phi(\xi_i)}{\Phi(\xi_i)}+\xi_i\right)\right]\\ \delta L_2(v,k) &= \sum_{i\in S_+}\left[\frac{k_i^2}{2}\frac{\phi(\xi_i)}{\Phi(\xi_i)}\left(\frac{\phi(\xi_i)}{\Phi(\xi_i)}+\xi_i\right)\right]\\ &= \sum_{i\in S_+}\frac{k_i^2}{2}f''(\xi_i)\end{aligned} \tag{87}$$

for each $\xi_i \in [v_i, v_i + k_i]$. Note that k_i need not be positive.

B.3.3 Bregman divergence of L_3

Similar to the previous case, we have

$$L_3(x;A,y) = -\sum_{i\in S_+}\log\left(\Phi\left(\frac{-\tau - A^i x}{\sigma}\right)\right) \tag{88}$$

Therefore,

$$\nabla L_3(x;A,y) = \sum_{i\in S_+}\frac{\phi\left(\frac{-A^ix-\tau}{\sigma}\right)}{\Phi\left(\frac{-A^ix-\tau}{\sigma}\right)}A^i \tag{89}$$

The expression for $\delta L_3(x, \Delta)$ is given by,

$$\begin{aligned}\delta L_3(x,\Delta) = \sum_{i\in S_+}\Bigg[&-\log\left(\Phi\left(\frac{-A^i(x+\Delta)-\tau}{\sigma}\right)\right)+\log\left(\Phi\left(\frac{-A^ix-\tau}{\sigma}\right)\right)\\ &-\frac{A^i\Delta}{\sigma}\frac{\phi\left(\frac{-A^ix-\tau}{\sigma}\right)}{\Phi\left(\frac{-A^ix-\tau}{\sigma}\right)}\Bigg]\end{aligned} \tag{90}$$

To simplify the notation, define $w_i = (-A^i x - \tau)/\sigma$ and $k_i = A^i\Delta/\sigma$ as before.
Let $w = [w_1, \ldots, w_m]$ and $k = [k_1 \ldots k_m]$. We then have,

$$\delta L_3(v,k) = \sum_{i\in S_-} -\log(\Phi(w_i - k_i)) + \log(\Phi(w_i)) - k_i\frac{\phi(w_i)}{\Phi(w_i)} \tag{91}$$

Like before, the Taylor series expansion of f at x and $x + d$ is given by

$$\begin{aligned}f(x+d) &= f(x) + df'(x) + \frac{d^2}{2}f''(\xi)\\ -\log(\Phi(x+d)) &= -\log(\Phi(x)) - d\frac{\phi(x)}{\Phi(x)} + \frac{d^2}{2}\frac{\phi(\xi)}{\Phi(\xi)}\left(\frac{\phi(\xi)}{\Phi(\xi)}+\xi\right)\end{aligned} \tag{92}$$

for $\xi \in [x, x + d]$. We apply the expansion at w_i and $w_i - k_i$ for each $i \in S_-$, we get

$$\begin{aligned}-\log(\Phi(w_i-k_i)) &= -\log(\Phi(w_i)) + k_i\frac{\phi(w_i)}{\Phi(w_i)} + \frac{k_i^2}{2}\frac{\phi(\xi_i)}{\Phi(\xi_i)}\left(\frac{\phi(\xi_i)}{\Phi(\xi_i)}+\xi_i\right)\\ -\log(\Phi(w_i-k_i)) + \log(\Phi(w_i)) - k_i\frac{\phi(w_i)}{\Phi(w_i)} &= \frac{k_i^2}{2}\frac{\phi(\xi_i)}{\Phi(\xi_i)}\left(\frac{\phi(\xi_i)}{\Phi(\xi_i)}+\xi_i\right)\end{aligned} \tag{93}$$

Summing across all $i \in S_-$,

$$\sum_{i\in S_-}\left[-\log(\Phi(v_i+k_i))+\log(\Phi(v_i))+k_i\frac{\phi(v_i)}{\Phi(k_i)}\right]=\sum_{i\in S_-}\left[\frac{k_i^2}{2}\frac{\phi(\xi_i)}{\Phi(\xi_i)}\left(\frac{\phi(\xi_i)}{\Phi(\xi_i)}+\xi_i\right)\right]$$
$$\begin{aligned}\delta L_3(v,k)&=\sum_{i\in S_-}\left[\frac{k_i^2}{2}\frac{\phi(\xi_i)}{\Phi(\xi_i)}\left(\frac{\phi(\xi_i)}{\Phi(\xi_i)}+\xi_i\right)\right]\\&=\sum_{i\in S_-}\frac{k_i^2}{2}f''(\xi_i)\end{aligned}\tag{94}$$

for each $\xi_i \in [w_i, w_i - k_i]$.

B.4 Curvature Calculations

From the above derivations, we have

$$\begin{aligned}\delta L_1(x^*,\Delta)&=\sum_{i\in S_{ns}}\frac{k_i^2}{2}\\\delta L_2(x^*,\Delta)&=\sum_{i\in S_+}\frac{k_i^2}{2}f''(\xi_i)\\\delta L_3(x^*,\Delta)&=\sum_{i\in S_-}\frac{k_i^2}{2}f''(\xi_i)\end{aligned}\tag{95}$$

where $k = A\Delta/\sigma$, $v^* = (Ax^* - \tau)/\sigma$ and $w^* = (-Ax^* - \tau)/\sigma$. $\xi_i \in [v_i^*, v_i^* + k_i]$ for $i \in S_+$ and $\xi_i \in [w_i^*, w_i^* - k_i]$ for $i \in S_-$.

Using these expressions, the value of $\delta L(x^*, \Delta)$ is given by

$$\begin{aligned}\delta L(x^*,\Delta)&=\delta L_1(x^*,\Delta)+\delta L_2(x^*,\Delta)+\delta L_3(x^*,\Delta)\\&=\sum_{i\in S_{ns}}\frac{k_i^2}{2}+\sum_{i\in S_+}\frac{k_i^2}{2}f''(\xi_i)+\sum_{i\in S_-}\frac{k_i^2}{2}f''(\xi_i)\\&\geq\min(\frac{1}{2},\{f''(\xi_i)\}_{i\in S_+},\{f''(\xi_i)\}_{i\in S_-})\sum_{i\in[m]}\frac{k_i^2}{2}\end{aligned}\tag{96}$$

We shall denote $\vartheta = \min(1/2, \{f''(\xi_i\}_{i\in S_+}, \{f''(\xi_i\}_{i\in S_-})$. Then, using the REC property of the matrix A, we get

$$\begin{aligned}\delta L(x^*,\Delta)=&\geq\vartheta\sum_{i\in[m]}\frac{k_i^2}{2}\\&\geq\frac{\vartheta}{2}\|A\Delta\|_2^2=\frac{\vartheta\gamma}{2}\|\Delta\|_2^2\end{aligned}\tag{97}$$

Plugging this equation into (76), we get $\kappa = \vartheta\gamma/2$ and $\tau(x) = 0$ for $x \in \mathbb{R}$.

B.5 Comparison of Bregman divergence with Saturation Rejection

The cost function in Saturation Rejection is given by

$$\begin{aligned} L_{SR}(x; A, y) &= \sum_{i \in S_{ns}} \frac{1}{2} \frac{(A^i x - y_i)^2}{\sigma^2} \\ &= L_1(x; A, y) \end{aligned} \tag{98}$$

If we do the similar calculations for this cost function, we get the Bregman divergence expression as

$$\begin{aligned} \delta L_{SR}(x, \Delta) &= \sum_{i \in S_{ns}} \left[\frac{(A^i(x+\Delta) - y_i)^2}{2\sigma^2} - \frac{(A^i x - y_i)^2}{2\sigma^2} - \frac{A^i \Delta}{\sigma}(A^i x - y_i) \right] \\ &= \sum_{i \in S_{ns}} \frac{\|A^i \Delta\|^2}{2\sigma^2} \end{aligned} \tag{99}$$

We can see from 96 that this value is lower than that of our method. Therefore, the curvature values of Saturation Rejection will also be lower. We have verified this empirically in the following experiments.

B.5.1 Experimental Comparison

Here we present the empirical curvature values for both SR and LM and compare them for different parameters. The empirical calculation is done as follows -

$$\begin{aligned} \delta L(x^*; A, y) &= \kappa \|\Delta\|_2^2 - \tau_L^2(x^*) \\ \delta L(x^*; A, y) &\leq \kappa \|\Delta\|_2^2 \\ \kappa &\geq \delta L(x^*; A, y) / \|\Delta\|_2^2 \end{aligned} \tag{100}$$

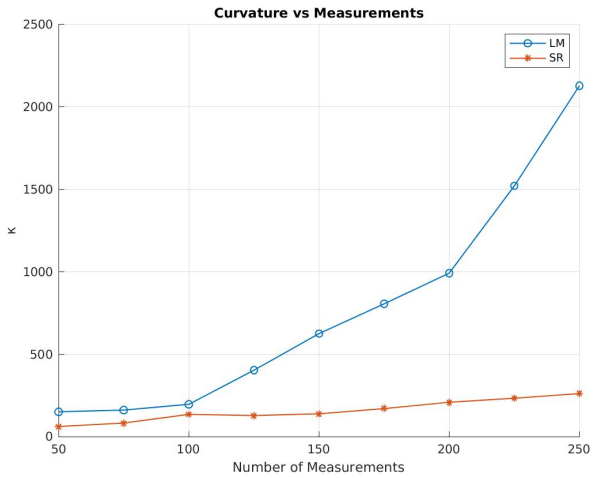
We do the same for Saturation rejection, and plot the values of empirical κ for both. The results are shown in Figure [20]

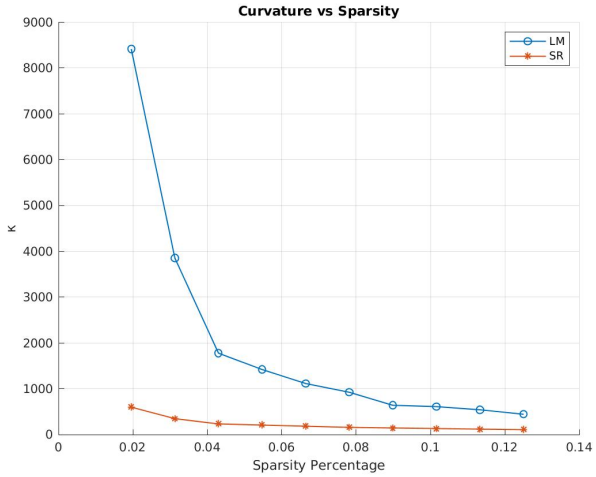
All the experiments are performed on signals of dimension $n = 256$. The elements of the sensing matrix A are drawn i.i.d. from $\mathcal{N}(0, \frac{1}{m})$ so that A would obey RIP with high probability. The additive noise is added to the measurements, followed by application of the saturation operator $\mathcal{C}$. Keeping all other parameters fixed, we analyse the variation in the curvature with regard to change in (A) number of measurements m; (B) signal sparsity s expressed as fraction $f_{sp} \in [0, 1]$ of signal dimension n; and (C) the fraction $f_{sat} \in [0, 1]$ of the m measurements that are saturated. For the measurements experiment (i.e. (A)), m is varied in $\{50, 75, 100, \ldots, 250\}$ with $s = 25$, $f_{sat} = 0.15$, $f_\sigma = 0.1$. For the sparsity experiment (i.e. (B)), f_{sp} is varied in $\{5, 8, 11, \ldots, 32\}/256$ with $m = 150$, $f_{sat} = 0.15$, $f_\sigma = 0.1$. For the saturation experiment (i.e. (D)), f_{sat} was varied in $\{0, 0.05, 0.1, \ldots, 0.5\}$ with $m = 150$, $f_{sp} = 25/256$, $f_\sigma = 0.1$. The curvature is computed over reconstructions from 100 signal samples.
The regularization parameters are chosen via cross validation with the unsaturated measurements with the validation set containing 20% of the measurements. Cross validation is done with a probability of 40% for each signal sample. The algorithms are implemented via the `cvx` package available on `MATLAB`.

C Gradient Bounds

The gradient of the cost function is given by

$$\nabla L(x; A, y) = \nabla L_1(x; A, y) + \nabla L_2(x; A, y) + \nabla L_3(x; A, y) \tag{101}$$





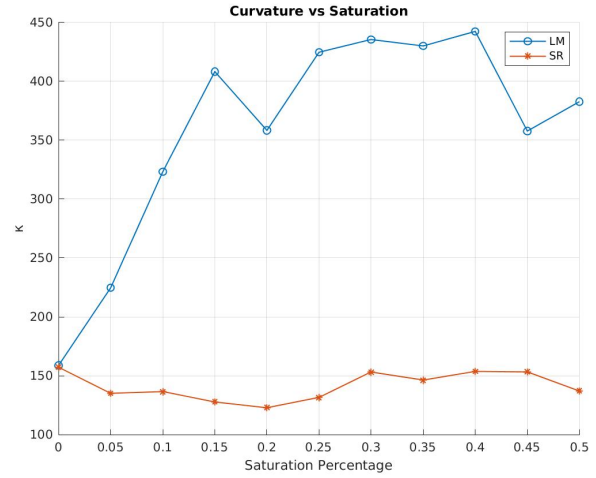


Figure 20: Empirical Curvature comparison of LM and SR wrt (A) Number of measurements m, (B) Signal sparsity s and (C) Saturation noise f_{sat}

$$\nabla L_1(x; A, y) = \frac{1}{\sigma} \sum_{i \in S_{ns}} A^i \frac{A^i x - y_i}{\sigma} \tag{102}$$

$$\nabla L_2(x; A, y) = -\frac{1}{\sigma} \sum_{i \in S_+} A^i \frac{\phi\left(\frac{A^i x - \tau}{\sigma}\right)}{\Phi\left(\frac{A^i x - \tau}{\sigma}\right)} \tag{103}$$

$$\nabla L_3(x; A, y) = \frac{1}{\sigma} \sum_{i \in S_-} A^i \frac{\phi\left(\frac{-\tau - A^i x}{\sigma}\right)}{\Phi\left(\frac{-\tau - A^i x}{\sigma}\right)} \tag{104}$$

We will derive the lower bounds for each of these terms and then use the triangle inequality to find the global lower bound for the gradient. To do so, we require the following lemmas.

Lemma 6. *The expression* $1 - \frac{\alpha}{2}\exp\left(-\beta u^2\right)$ *is non-negative when* $u \geq 0$, $\beta > 1$ *and* $0 < \alpha \leq \sqrt{\frac{2e}{\pi}}\frac{\sqrt{\beta-1}}{\beta}$.

Proof. From [2], we have that $erfc(x) \geq \alpha e^{-\beta x^2}$ when $x \geq 0$, $\beta > 1$ and $0 < \alpha \leq \sqrt{\frac{2e}{\pi}}\frac{\sqrt{\beta-1}}{\beta}$.
Using the definition of $erfc(u)$, for any $u \geq 0$ we have

$$\begin{aligned} 2(1 - \Phi(\sqrt{2}u)) &\geq \alpha \exp\left(-\beta u^2\right) \\ 1 - \Phi(\sqrt{2}u) &\geq \frac{\alpha}{2}\exp\left(-\beta u^2\right) \\ 0 \leq \Phi(\sqrt{2}u) &\leq 1 - \frac{\alpha}{2}\exp\left(-\beta u^2\right) \end{aligned}$$

□

Lemma 7. *The inverse Mill's ratio* ϕ/Φ *has the following bounds*

$$\frac{\phi(u)}{\Phi(u)} \geq \sqrt{\frac{2}{\pi}}\frac{e^{-u_2^2/2}}{2 - \alpha e^{-\beta u_2^2/2}}, \qquad 0 \leq u \leq u_1 \tag{105}$$

$$\frac{\phi(u)}{\Phi(u)} \geq \sqrt{\frac{2}{\pi}}\frac{e^{-u_3^2/2}}{2 - \alpha e^{-\beta u_3^2/2}}, \qquad u_2 \leq u \leq 0 \tag{106}$$

where $u_1, u_2 \in \mathbb{R}$

Proof. From the previous lemma, when $u \geq 0$, $\beta > 1$ and $0 < \alpha \leq \sqrt{\frac{2e}{\pi}}\frac{\sqrt{\beta-1}}{\beta}$, we have

$$\begin{aligned} 0 \leq \Phi(\sqrt{2}u) &\leq 1 - \frac{\alpha}{2}\exp\left(-\beta u^2\right) \\ \frac{1}{\Phi(u)} &\geq \frac{1}{1 - \frac{\alpha}{2}\exp(-\beta u^2/2)} \\ \frac{\phi(u)}{\Phi(u)} &\geq \frac{\frac{1}{\sqrt{2\pi}}\exp\left\{(-u^2/2)\right\}}{1 - \frac{\alpha}{2}\exp(-\beta u^2/2)} \end{aligned} \tag{107}$$

Now, let us consider the first case where $0 \leq u < u_1$ for some $u_1 \in \mathbb{R}$.

$$0 \leq u < u_1 \tag{108}$$
$$-u_1^2 \leq -u^2 \leq 0 \tag{109}$$
$$\sqrt{\frac{1}{2\pi}}\exp\{(-u_2^2)\} \leq \sqrt{\frac{1}{2\pi}}\exp\{(-u^2)\} \leq 1 \tag{110}$$
$$0 \leq 1 - \frac{\alpha}{2}\exp\{(-\beta u^2/2)\} \leq 1 - \frac{\alpha}{2}\exp\{(-\beta u_1^2/2)\} \qquad \text{Lemma 1} \tag{111}$$
$$0 \leq \frac{1}{1-\frac{\alpha}{2}\exp\{(-\beta u_1^2/2)\}} \leq \frac{1}{1-\frac{\alpha}{2}\exp\{(-\beta u^2/2)\}} \tag{112}$$

Using (107), (110) and (112) from the above set of steps, we obtain

$$\frac{\phi(u)}{\Phi(u)} \geq \sqrt{\frac{2}{\pi}}\frac{e^{-u_1^2/2}}{2-\alpha e^{-\beta u_1^2/2}}, \qquad 0 \leq u < u_1 \tag{113}$$

For $u_2 \leq u \leq 0$,

$$u_2 \leq u \leq 0 \tag{114}$$
$$-u_2^2 \leq -u^2 \leq 0 \tag{115}$$
$$\sqrt{\frac{1}{2\pi}}\exp\{(-u_2^2)\} \leq \sqrt{\frac{1}{2\pi}}\exp\{(-u^2)\} \leq 1 \tag{116}$$
$$0 \leq 1 - \frac{\alpha}{2}\exp\{(-\beta u^2/2)\} \leq 1 - \frac{\alpha}{2}\exp\{(-\beta u_2^2/2)\} \qquad \text{Lemma 1} \tag{117}$$
$$0 \leq \frac{1}{1-\frac{\alpha}{2}\exp\{(-\beta u_2^2/2)\}} \leq \frac{1}{1-\frac{\alpha}{2}\exp\{(-\beta u^2/2)\}} \tag{118}$$

Using (107), (116) and (118) from the above set of steps, we obtain

$$\frac{\phi(u)}{\Phi(u)} \geq \sqrt{\frac{2}{\pi}}\frac{e^{-u_2^2/2}}{2-\alpha e^{-\beta u_2^2/2}}, \qquad u \leq u_2 \leq 0$$

□

Moving forward, we shall use the same notation as before - $v_i \triangleq (A^i x - \tau)/\sigma$ for $i \in S_+$ and $w_i \triangleq (-\tau - A^i x)/\sigma$ for $i \in S_-$.

In order to bound the gradient of the cost function, we shall assume that the true signal x^* lies between two values X_1 and X_2. That is, $\forall i \in [n], X_1 \leq x \leq X_2$.

C.1 Bound for ∇L_1

We have

$$\nabla L_1(x; A, y) = \frac{1}{\sigma}\sum_{i \in S_{ns}} A^i \left(\frac{A^i x - y_i}{\sigma}\right)$$

Let A_1 be the sub-matrix of A consisting of only the rows with non-saturated entries, defined by $A_1 \triangleq [A^j | j \in S_{ns}]$. Let m_{ns} denote the number of non-saturated entries. The jth term of the gradient is

$-(A_1^j)^T z \sim \mathcal{N}(0, \|A_1^j\|^2)$ where $z_i \triangleq y_i - A_1^i x \sim \mathcal{N}(0, \sigma^2)$. This is due to the property of linear combination of normal variables. Also, assuming the entries of A are sampled from the distribution $\mathcal{N}(0, 1/m)$, we can write $m(A_1^{ij}) \sim \chi_1^2$ and

$$\left[m \sum_{i \in S_{ns}} (A_1^{ij})^2\right] \sim \chi_{m'}^2$$
$$\mathbb{E}\left[m \sum_{i \in S_{ns}} (A_1^{ij})^2\right] = m_{ns} \implies \|A_1^j\|_2^2 \approx \frac{m_{ns}}{m}$$

Hence, approximately

$$-(A_1^i)^T z \sim \mathcal{N}(0, \frac{m_{ns}\sigma^2}{m})$$

The Gaussian concentration bound for the gradient is given by,

$$\mathbb{P}\left[|(A_1^i)^T z| \geq u\right] \leq 2\exp\left(-\frac{u^2 m}{2m_{ns}\sigma^2}\right)$$

Then, using union bound,

$$\mathbb{P}\left[\|A_1^T z\|_\infty \geq u\right] \leq 2\exp\left(-\frac{u^2 m}{2m_{ns}\sigma^2} + \log(n)\right)$$

Define $\varrho \triangleq u^2 m / m_{ns}\sigma^2 \log(n)$. The above bound is useful when $\varrho > 2$. We then have,

$$\mathbb{P}\left[\|A_1^T z\|_\infty \geq \sqrt{\frac{m_{ns}\varrho\log(n)}{m}}\right] \leq 2\exp\left(-\frac{1}{2}(\varrho - 2)\log(n)\right)$$
$$\mathbb{P}\left[\|A_1^T z\|_\infty \leq \sqrt{\frac{m_{ns}\varrho\log(n)}{m}}\right] \geq 1 - 2\exp\left(-\frac{1}{2}(\varrho - 2)\log(n)\right)$$

Therefore,

$$\|\nabla L_1(y; A, x)\|_\infty \leq \sqrt{\frac{m_{ns}\varrho\log(n)}{m}}$$

with probability at least $1 - 2\exp\left(-\frac{1}{2}(\varrho - 2)\log(n)\right)$.

C.2 Bound for ∇L_2

We have

$$\nabla L_2(v) = -\frac{1}{\sigma}\sum_{i \in S_+} A^i \frac{\phi(v_i)}{\Phi(v_i)} \tag{119}$$

For the true signal x^*, we have

$$X_1 \leq x_j^* < X_2 \tag{120}$$

$$\sum_{j,A^{ij}>0} A^{ij}X_1 + \sum_{j,A^{ij}\leq 0} A^{ij}X_2 \leq \sum_{j=1}^{n} A^{ij}x_j^* < \sum_{j,A^{ij}>0} A^{ij}X_2 + \sum_{j,A^{ij}\leq 0} A^{ij}X_1 \tag{121}$$

$$(p_1)_i \leq A^i x < (p_2)_i \quad \text{renaming for convenience} \tag{122}$$

$$\frac{(p_1)_i - \tau}{\sigma} \leq \frac{A^i x^* - \tau}{\sigma} < \frac{(p_2)_i - \tau}{\sigma} \tag{123}$$

$$(v_1)_i \leq v_i^* < (v_2)_i \tag{124}$$

Also, $v_i^* \geq 0$. Let $v_2 = \max_i (v_2)_i$, then $\forall i \in S_+, v_i^* \leq v_2$. Using Lemma 7 on each individual term, we get for $i \in S_+$

$$\frac{\phi(v_i)}{\Phi(v_i)} \geq \sqrt{\frac{2}{\pi}} \frac{e^{-v_2^2/2}}{2 - \alpha e^{-\beta v_2^2/2}} = T_+ \tag{125}$$

Note that T_+ is a function of A. Then,

$$\nabla L_2(v) \geq -\frac{1}{\sigma} \sum_{i \in S_+} A^i T_+ \tag{126}$$

Also, v_2 is of the order $\mathcal{O}(\sqrt{m_+/m})$ since $(p_2)_i$ is summation of m_1 Gaussian variables, and its distribution is given by $\mathcal{N}(0, C * m_+/m)$ for some $C > 0$ where m_+ is the number of positively saturated measurements.

C.3 Bound for ∇L_3

We will similarly lower bound L_3 like above. We have

$$\nabla L_3(v) = \frac{1}{\sigma} \sum_{i \in S_-} A^i \frac{\phi(w_i)}{\Phi(w_i)} \tag{127}$$

For the true signal x^*, we have

$$X_1 \leq x_j^* < X_2 \tag{128}$$

$$\sum_{j,A^{ij}>0} A^{ij}X_1 + \sum_{j,A^{ij}\leq 0} A^{ij}X_2 \leq \sum_{j=1}^{n} A^{ij}x_j^* < \sum_{j,A^{ij}>0} A^{ij}X_2 + \sum_{j,A^{ij}\leq 0} A^{ij}X_1 \tag{129}$$

$$(p_3)_i \leq A^i x < (p_4)_i \quad \text{renaming for convenience} \tag{130}$$

$$\frac{-(p_4)_i - \tau}{\sigma} \leq \frac{-A^i x^* - \tau}{\sigma} < \frac{-(p_3)_i - \tau}{\sigma} \tag{131}$$

$$(w_1)_i \leq w_i^* < (w_2)_i \tag{132}$$

Also, $w_i^* \geq 0$. Let $w_2 = \max_i (w_2)_i$, then $\forall i \in S_-, w_i^* \leq w_2$. Then $\forall i \in S_-$,

$$\frac{\phi(w_i)}{\Phi(w_i)} \geq \sqrt{\frac{2}{\pi}} \frac{e^{-w_2^2/2}}{2 - \alpha e^{-\beta w_2^2/2}} = T_- \tag{133}$$

Note that T_- is a function of A. Then,

$$\nabla L_3(v) \geq \frac{1}{\sigma} \sum_{i \in S_-} A^i T_- \tag{134}$$

Also, w_2 is of the order $\mathcal{O}(\sqrt{m_-/m})$ similarly.

References

[1] E. Candes and M. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 2008.

[2] Seok-Ho Chang, Pamela C. Cosman, and Laurence B. Milstein. Chernoff-type bounds for the gaussian error function. *IEEE Transactions on Communications*, 59(11):2939–2944, 2011.

[3] Valentin Emiya, Antoine Bonnefoy, Laurent Daudet, and Remi Gribonval. Compressed sensing with unknown sensor permutation. pages 1040–1044, 05 2014.

[4] Simon Foucart and Jiangyuan Li. Sparse recovery from inaccurate saturated measurements. *Acta Applicandae Mathematicae*, 158, 12 2018.

[5] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman Hall/CRC, 2015.

[6] Adel Javanmard and Andrea Montanari. Hypothesis testing in high-dimensional regression under the gaussian random design model: Asymptotic theory, 2013.

[7] Jason Laska, Petros Boufounos, Mark Davenport, and Richard Baraniuk. Democracy in action: Quantization, saturation, and compressive sensing. *Applied and Computational Harmonic Analysis*, 31:429–443, 11 2011.

[8] Jason Laska, Mark Davenport, and Richard Baraniuk. Exact signal recovery from sparsely corrupted measurements through the pursuit of justice. pages 1556 – 1560, 12 2009.

[9] Bin Li, Lucas Rencker, Jing Dong, Yuhui Luo, Mark Plumbley, and Wenwu Wang. Sparse analysis model based dictionary learning for signal declipping. 01 2021.

[10] Sai Li. Debiasing the debiased lasso with bootstrap, 2020.

[11] Zhi Li, Feng Wu, and John Wright. On the systematic measurement matrix for compressed sensing in the presence of gross errors. *2010 Data Compression Conference*, pages 356–365, 2010.

[12] Sahand N. Negahban, Pradeep Ravikumar, Martin J. Wainwright, and Bin Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical Science*, 27(4), nov 2012.

[13] Garvesh Raskutti, Martin Wainwright, and B. Yu. Restricted eigenvalue properties for correlated gaussian designs. *Journal of Machine Learning Research*, 11:2241–2259, 08 2010.

[14] Lucas Rencker, Francis Bach, Wenwu Wang, and Mark D. Plumbley. Sparse recovery and dictionary learning from nonlinear compressive measurements. *IEEE Transactions on Signal Processing*, 67(21):5659–5670, 2019.

[15] George Tzagkarakis, John Nolan, and Panagiotis Tsakalides. Compressive sensing using symmetric alpha-stable distributions for robust sparse signal reconstruction. *IEEE Transactions on Signal Processing*, PP:1–1, 12 2018.