Consistent Robust Regression

Kush Bhatia[*]
University of California, Berkeley
kushbhatia@berkeley.edu

Prateek Jain
Microsoft Research, India
prajain@microsoft.com

Parameswaran Kamalaruban[†]
EPFL, Switzerland
kamalaruban.parameswaran@epfl.ch

Purushottam Kar
Indian Institute of Technology, Kanpur
purushot@cse.iitk.ac.in

Abstract

We present the first efficient and provably consistent estimator for the robust regression problem. The area of robust learning and optimization has generated a significant amount of interest in the learning and statistics communities in recent years owing to its applicability in scenarios with corrupted data, as well as in handling model mis-specifications. In particular, special interest has been devoted to the fundamental problem of robust linear regression where estimators that can tolerate corruption in up to a constant fraction of the response variables are widely studied. Surprisingly however, to this date, we are not aware of a polynomial time estimator that offers a consistent estimate in the presence of dense, unbounded corruptions. In this work we present such an estimator, called CRR. This solves an open problem put forward in the work of [3]. Our consistency analysis requires a novel two-stage proof technique involving a careful analysis of the stability of ordered lists which may be of independent interest. We show that CRR not only offers consistent estimates, but is empirically far superior to several other recently proposed algorithms for the robust regression problem, including *extended Lasso* and the TORRENT algorithm. In comparison, CRR offers comparable or better model recovery but with runtimes that are faster by an order of magnitude.

1 Introduction

The problem of robust learning involves designing and analyzing learning algorithms that can extract the underlying model despite dense, possibly malicious, corruptions in the training data provided to the algorithm. The problem has been studied in a dizzying variety of models and settings ranging from regression [19], classification [11], dimensionality reduction [4] and matrix completion [8].

In this paper we are interested in the Robust Least Squares Regression (RLSR) problem that finds several applications to robust methods in face recognition and vision [22, 21], and economics [19]. In this problem, we are given a set of n covariates in d dimensions, arranged as a data matrix $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, and a response vector $\mathbf{y} \in \mathbb{R}^n$. However, it is known apriori that a certain number k of these responses cannot be trusted since they are corrupted. These may correspond to corrupted pixels in visual recognition tasks or untrustworthy measurements in general sensing tasks.

Using these corrupted data points in any standard least-squares solver, especially when $k = \mathcal{O}(n)$, is likely to yield a poor model with little predictive power. A solution to this is to exclude corrupted

[*]Work done in part while Kush was a Research Fellow at Microsoft Research India.
[†]Work done in part while Kamalaruban was interning at Microsoft Research India.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Table 1: A comparison of different RLSR algorithms and their properties. CRR is the first efficient RLSR algorithm to guarantee consistency in the presence of a constant fraction of corruptions.

Paper	Breakdown Point	Adversary	Consistent	Technique
Wright & Ma, 2010 [21]	$\alpha \to 1$	Oblivious	No	L_1 regularization
Chen & Dalalyan, 2010 [7]	$\alpha \geq \Omega(1)$	Adaptive	No	SOCP
Chen et al., 2013 [6]	$\alpha \geq \Omega\left(\frac{1}{\sqrt{d}}\right)$	Adaptive	No	Robust thresholding
Nguyen & Tran, 2013 [16]	$\alpha \to 1$	Oblivious	No	L_1 regularization
Nguyen & Tran, 2013b [17]	$\alpha \to 1$	Oblivious	No	L_1 regularization
McWilliams et al., 2014 [14]	$\alpha \geq \Omega\left(\frac{1}{\sqrt{d}}\right)$	Oblivious	No	Weighted subsampling
Bhatia et al., 2015 [3]	$\alpha \geq \Omega(1)$	Adaptive	No	Hard thresholding
This paper	$\alpha \geq \Omega(1)$	**Oblivious**	**Yes**	**Hard thresholding**

points from consideration. The RLSR problem formalizes this requirement as follows:

$$(\widehat{\mathbf{w}}, \widehat{S}) = \underset{\substack{\mathbf{w}\in\mathbb{R}^p, S\subset[n] \\ |S|=n-k}}{\arg\min} \sum_{i\in S} (y_i - \mathbf{x}_i^T \mathbf{w})^2, \tag{1}$$

This formulation seeks to simultaneously extract the set of uncorrupted points and estimate the least-squares solutions over those uncorrupted points. Due to the combinatorial nature of the RLSR formulation (1), solving it directly is challenging and in fact, NP-hard in general [3, 20].

Literature in robust statistics suggests several techniques to solve (1). The most common model assumes a realizable setting wherein there exists a gold model $\mathbf{w}^*$ that generates the non-corrupted responses. A vector of *corruptions* is then introduced to model the corrupted responses i.e.

$$\mathbf{y} = X^T \mathbf{w}^* + \mathbf{b}^*. \tag{2}$$

The goal of RLSR is to recover $\mathbf{w}^* \in \mathbb{R}^d$, the *true* model. The vector $\mathbf{b}^* \in \mathbb{R}^n$ is a k-sparse vector which takes non-zero values on at most k corrupted samples out of the n total samples, and a zero value elsewhere. A more useful, but challenging model is one in which (mostly heteroscedastic and i.i.d.) Gaussian noise is injected into the responses in addition to the corruptions.

$$\mathbf{y} = X^T \mathbf{w}^* + \mathbf{b}^* + \boldsymbol{\epsilon}. \tag{3}$$

Note that the Gaussian noise vector $\boldsymbol{\epsilon}$ is not sparse. In fact, we have $\|\boldsymbol{\epsilon}\|_0 = n$ almost surely.

2 Related Works

A string of recent works have looked at the RLSR problem in various settings. To facilitate a comparison among these, we set the following benchmarks for RLSR algorithms

1. (Breakdown Point) This is the number of corruptions k that an RLSR algorithm can tolerate is a direct measure of its robustness. This limit is formalized as the *breakdown point* of the algorithm in statistics literature. The breakdown point k is frequently represented as a fraction α of the total number of data points i.e. $k = \alpha \cdot n$.
2. (Adversary Model) RLSR algorithms frequently resort to an adversary model to specify how are the corruptions introduced into the regression problem. The strictest is the *adaptive adversarial* model wherein the adversary is able to view X and $\mathbf{w}^*$ (as well as $\boldsymbol{\epsilon}$ if Gaussian noise is present) before deciding upon $\mathbf{b}^*$. A weaker model is the *oblivious adversarial* model wherein the adversary generates a k-sparse vector in complete ignorance of X and $\mathbf{w}^*$ (and $\boldsymbol{\epsilon}$). However, the adversary is still free to make arbitrary choices for the location and values of corruptions.
3. (Consistency) RLSR algorithms that are able to operate in the *hybrid* noise model with sparse adversarial corruptions as well as dense Gaussian noise are more valuable. An RLSR algorithms is said to be consistent if, when invoked in the hybrid noise model on n data points sampled from a distribution with appropriate characteristics, the RLSR algorithm returns an estimate $\widehat{\mathbf{w}}_n$ such that $\lim_{n\to\infty} \mathbb{E}\left[\widehat{\mathbf{w}}_n - \mathbf{w}^*\right]_2 = 0$ (for simplicity, assume a fixed covariate design with the expectation being over random Gaussian noise in the responses).

In Table 1, we present a summarized view of existing RLSR techniques and their performance vis-a-vis the benchmarks discussed above. Past work has seen the application of a wide variety of algorithmic techniques to solve this problem, including more expensive methods involving L_1 regularization (for example $\min_{\mathbf{w},\mathbf{b}} \lambda_w \|\mathbf{w}\|_1 + \lambda_b \|\mathbf{b}\|_1 + \|X^\top \mathbf{w} + \mathbf{b} - \mathbf{y}\|_2^2$) and second-order cone programs such as [21, 7, 16, 17], as well as more scalable methods such as the robust thresholding and iterative hard thresholding [6, 3]. As the work of [3] shows, L_1 regularization and other expensive methods struggle to scale to even moderately sized problems.

The adversary models considered by these works is also quite diverse. Half of the works consider an oblivious adversary and the other half brace themselves against an adaptive adversary. The oblivious adversary model, although weaker, can model some important practical situations where there is systematic error in the sensing equipment being used, such as a few pixels in a camera becoming unresponsive. Such errors are surely not random, and hence cannot be modeled as Gaussian noise, but introduce corruptions the final measurement in a manner that is oblivious of the signal actually being sensed, in this case the image being photographed.

An important point of consideration is the breakdown point of these methods. Among those cited in Table 1, the works of [21] and [16] obtain the best breakdown points that allow a fraction of points to be corrupted that is arbitrarily close to 1. They require the data to be generated from either an isotropic Gaussian ensemble or be row-sampled from an incoherent orthogonal matrix. Most results mentioned in the table allow a constant fraction of points to be corrupted i.e. allow $k = \alpha \cdot n$ corruptions for some fixed constant $\alpha > 0$. This is still impressive since it allows a dense subset of data points to be corrupted and yet guarantees recovery. However, as we shall see below, these results cannot guarantee consistency while allowing $k = \alpha \cdot n$ corruptions.

We note that we use the term *dense* to refer to the corruptions in our model since they are a constant fraction of the total available data. Moreover, as we shall see, this constant shall be universal and independent of the ambient dimensionality d. This terminology is used to contrast against some other works which can tolerate only $o(n)$ corruptions which is arguably much sparser. For instance, as we shall see below, the work of [17] can tolerate only $o(n/\log n)$ corruptions if a consistent estimate is expected. The work of [6] also offers a weak guarantee wherein they are only able to tolerate a $1/\sqrt{d}$ fraction of corruptions. However, [6] allow corruptions in covariates as well.

However, we note that *none* of the algorithms listed here, and to the best of our knowledge elsewhere as well, are able to guarantee a consistent solution, irrespective of assumptions on the adversary model. More specifically, none of these methods are able to guarantee exact recovery of $\mathbf{w}^*$, even with $n \to \infty$ and constant fraction of corruptions $\alpha = \Omega(1)$ (i.e. $k = \Omega(n)$) . At best, they guarantee $\|\mathbf{w} - \mathbf{w}^*\|_2 \leq \mathcal{O}(\sigma)$ when $k = \Omega(n)$ where σ is the standard deviation of the white noise (see Equation 3). Thus, their estimation error is of the order of the white noise in the system, even if the algorithm is supplied with an infinite amount of data. This is quite unsatisfactory, given our deep understanding of the consistency guarantees for least squares models.

For example, consider the work of [17] which considers a corruption model similar to (3). The work makes deterministic assumptions on the data matrix and proposes the following convex program.

$$\min_{\mathbf{w},\mathbf{b}} \lambda_w \|\mathbf{w}\|_1 + \lambda_b \|\mathbf{b}\|_1 + \|X^\top \mathbf{w} + \mathbf{b} - \mathbf{y}\|_2^2. \tag{4}$$

For Gaussian designs, which we also consider, their results guarantee that for $n = \mathcal{O}(s \log d)$,

$$\|\widehat{\mathbf{w}} - \mathbf{w}^*\|_2 + \|\widehat{\mathbf{b}} - \mathbf{b}^*\|_2 \leq \mathcal{O}\left(\sqrt{\frac{\sigma^2 s \log d \log n}{n}} + \sqrt{\frac{\sigma^2 k \log n}{n}}\right)$$

where s is the sparsity index of the regressor $\mathbf{w}^*$. Note that for $k = \Theta(n)$, the right hand side behaves as $\Omega\left(\sigma\sqrt{\log n}\right)$. Thus, the result is unable to ensure $\lim_{n\to\infty} \mathbb{E}\left[\widehat{\mathbf{w}}_n - \mathbf{w}^*\right]_2 = 0$.

We have excluded some classical approaches to the RLSR problem from the table such as [18, 1, 2] which use the Least Median of Squares (LMS) and Least Trimmed Squares (LTS) methods that guaranteed consistency but may require an exponential running time. Our focus is on polynomial time algorithms, more specifically those that are efficient and scalable. We note a recent work [5] in robust stochastic optimization which is able to tolerate a constant fraction of corruptions $\alpha \to 1$. However, their algorithms operate in the *list-decoding* model wherein they output not one, but as many as $\mathcal{O}\left(\frac{1}{1-\alpha}\right)$ models, of which one (unknown) model is guaranteed to be correct.

Recovering Sparse High-dimensional Models: We note that several previous works extend their methods and analyses to handle the case of sparse robust recovery in high-dimensional settings as well, including [3, 7, 17]. A benefit of such extensions is the ability to work even in data starved settings $n \ll d$ if the true model $\mathbf{w}^*$ is s-sparse with $s \ll d$. However, previous works continue to require the number of corruptions to be of the order of $k = o(n)$ or else $k = \mathcal{O}(n/s)$ in order to ensure that $\lim_{n\to\infty} \mathbb{E}\left[\widehat{\mathbf{w}}_n - \mathbf{w}^*\right]_2 = 0$ and cannot ensure consistency if $k = \mathcal{O}(n)$. This is evident, for example from the recovery guarantee offered by [17] discussed above, which requires $k = o(n/\log n)$. We do believe our CRR estimator can be adapted to high dimensional settings as well. However, the details are tedious and we reserve them for an expanded version of the paper.

3 Our Contributions

In this paper, we remedy the above problem by using a simple and scalable iterative hard-thresholding algorithm called CRR along with a novel two-stage proof technique. Given n covariates that form a Gaussian ensemble, our method in time poly(n, d), outputs an estimate $\widehat{\mathbf{w}}_n$ s.t. $\|\widehat{\mathbf{w}}_n - \mathbf{w}^*\|_2 \to 0$ as $n \to \infty$ (see Theorem 4 for a precise statement). In fact, our method guarantees a nearly optimal error rate of $\|\widehat{\mathbf{w}}_n - \mathbf{w}^*\|_2 \leq \sigma\sqrt{\frac{d}{n}}$. It is noteworthy that CRR can tolerate a constant fraction of corruptions i.e. tolerate $k = \alpha \cdot n$ corruptions for some fixed $\alpha > 0$.

We note that although hard thresholding techniques have been applied to the RLSR problem earlier [3, 6], none of those methods are able to guarantee a consistent solution to the problem. Our results hold in the setting where a *constant fraction* of the responses are corrupted by an *oblivious adversary* (i.e. the one which corrupts observations without information of the data points themselves). Our algorithm runs in time $\widetilde{\mathcal{O}}\left(d^3 + nd\right)$, where d is the dimensionality of the data. Moreover, as we shall see, our technique makes more efficient use of data than previous hard thresholding methods such as TORRENT [3].

To the best of our knowledge, this is the *first* efficient and consistent estimator for the RLSR problem in the challenging setting where a *constant* fraction of the responses may be corrupted in the presence of dense noise. We would like to note that the problem of consistent robust regression is especially challenging because without the assumption of an *oblivious* adversary, consistent estimation with a constant fraction of corruptions (even for an arbitrarily small constant) may be impossible even when supplied with infinitely many data points.

However, by crucially using the restriction of obliviousness on the adversary along with a novel proof technique, we are able to provide a consistent estimator for RLSR with optimal (up to constants) statistical and computational complexity.

Discussion on Problem Setting: We clarify that our improvements come at a cost. Our results assume an oblivious adversary whereas several previous works allowed a fully adaptive adversary. Indeed there is no free-lunch: it seems unlikely that consistent estimators are even possible in the face of a fully adaptive adversary who can corrupt a constant fraction of responses since such an adversary can use his power to introduce biased noise into the model in order to defeat any estimator. An oblivious adversary is prohibited from looking at the responses before deciding the corruptions and is thus unable to do the above.

Paper Organization: We will begin our discussion by introducing the problem formulation, relevant notation, and tools in Section 4. This is followed by Section 5 where we develop CRR, a near-linear time algorithm that gives consistent estimates for the RLSR problem, which we analyze in Section 6. Finally in Section 7, we present rigorous experimental benchmarking of this algorithm. In Section 8 we offer some clarifications on how the manuscript was modified in response to reviewer comments.

4 Problem Formulation

We are given n data points $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$, where $\mathbf{x}_i \in \mathbb{R}^d$ are the *covariates* and, for some *true* model $\mathbf{w}^* \in \mathbb{R}^d$, the vector of *responses* $\mathbf{y} \in \mathbb{R}^n$ is generated

$$\mathbf{y} = X^\top \mathbf{w}^* + \mathbf{b}^* + \boldsymbol{\epsilon}. \tag{5}$$

The responses suffer two kinds of perturbations – *dense white noise* $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ that is chosen in an i.i.d. fashion independently of the data X and the model $\mathbf{w}^*$, and *adversarial corruptions*

Algorithm 1 CRR: Consistent Robust Regression

Input: Covariates $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, responses $\mathbf{y} = [y_1, \ldots, y_n]^\top$, corruption index k, tolerance ϵ

$\mathbf{b}^0 \leftarrow \mathbf{0}, t \leftarrow 0,$
$P_X \leftarrow X^\top (XX^\top)^{-1} X$
while $\left\|\mathbf{b}^t - \mathbf{b}^{t-1}\right\|_2 > \epsilon$ **do**
  $\mathbf{b}^{t+1} \leftarrow \mathrm{HT}_k(P_X \mathbf{b}^t + (I - P_X)\mathbf{y})$
  $t \leftarrow t + 1$
end while
return $\mathbf{w}^t \leftarrow (XX^\top)^{-1} X(\mathbf{y} - \mathbf{b}^t)$

in the form of $\mathbf{b}^*$. We assume that $\mathbf{b}^*$ is a k^*-sparse vector albeit one with potentially unbounded entries. The constant k^* will be called the *corruption index* of the problem. We assume the *oblivious adversary* model where $\mathbf{b}^*$ is chosen independently of $X, \mathbf{w}^*$ and $\boldsymbol{\epsilon}$.

Although there exist works that operate under a fully adaptive adversary [3, 7], none of these works are able to give a consistent estimate, whereas our algorithm CRR does provide a consistent estimate. We also note that existing works are unable to give consistent estimates even in the oblivious adversary model. Our result requires a significantly finer analysis; the standard ℓ_2-norm style analysis used by existing works [3, 7] seems incapable of offering a consistent estimation result in the robust regression setting.

We will require the notions of *Subset Strong Convexity* and *Subset Strong Smoothness* similar to [3] and reproduce the same below. For any set $S \subset [n]$, let $X_S := [\mathbf{x}_i]_{i \in S} \in \mathbb{R}^{d \times |S|}$ denote the matrix with columns in that set. We define $\mathbf{v}_S$ for a vector $\mathbf{v} \in \mathbb{R}^n$ similarly. $\lambda_{\min}(X)$ and $\lambda_{\max}(X)$ will denote, respectively, the smallest and largest eigenvalues of a square symmetric matrix X.

Definition 1 (SSC Property)**.** *A matrix $X \in \mathbb{R}^{d \times n}$ is said to satisfy the* Subset Strong Convexity *Property at level m with constant λ_m if the following holds:*

$$\lambda_m \leq \min_{|S|=m} \lambda_{\min}(X_S X_S^\top)$$

Definition 2 (SSS Property)**.** *A matrix $X \in \mathbb{R}^{d \times n}$ is said to satisfy the* Subset Strong Smoothness *Property at level m with constant Λ_m if the following holds:*

$$\max_{|S|=m} \lambda_{\max}(X_S X_S^\top) \leq \Lambda_m.$$

Intuitively speaking, the SSC and SSS properties ensure that the regression problem remains well conditioned, even if restricted to an arbitrary subset of the data points. This allows the estimator to recover the exact model no matter what portion of the data was left uncorrupted by the adversary. We refer the reader to the Appendix A.1 for SSC/SSS bounds for Gaussian ensembles.

5 CRR: A Hard Thresholding Approach to Consistent Robust Regression

We now present a consistent method CRR for the RLSR problem. CRR takes a significantly different approach to the problem than previous works. Instead of attempting to exclude data points deemed unclean (as done by the TORRENT algorithm proposed by [3]), CRR focuses on correcting the errors. This allows CRR to work with the entire dataset at all times, as opposed to TORRENT that works with a fraction of the data at any given point of time.

To motivate the CRR algorithm, we start with the RLSR formulation $\min_{\mathbf{w} \in \mathbb{R}^p, \|\mathbf{b}\|_0 \leq k^*} \frac{1}{2} \left\|X^\top \mathbf{w} - (\mathbf{y} - \mathbf{b})\right\|_2^2$, and realize that given any estimate $\widehat{\mathbf{b}}$ of the corruption vector, the optimal model with respect to this estimate is given by the expression $\widehat{\mathbf{w}} = (XX^\top)^{-1} X(\mathbf{y} - \widehat{\mathbf{b}})$. Plugging this expression for $\widehat{\mathbf{w}}$ into the formulation allows us to reformulate the RLSR problem.

$$\min_{\|\mathbf{b}\|_0 \leq k^*} f(\mathbf{b}) = \frac{1}{2} \|(I - P_X)(\mathbf{y} - \mathbf{b})\|_2^2 \tag{6}$$

where $P_X = X^\top (XX^\top)^{-1} X$. This greatly simplifies the problem by casting it as a *sparse parameter estimation* problem instead of a data subset selection problem (as done by TORRENT). CRR directly

optimizes (6) by using a form of iterative hard thresholding. Notice that this approach allows CRR to keep using the entire set of data points at all times, all the while using the current estimate of the parameter $\mathbf{b}$ to correct the errors in the observations. At each step, CRR performs the following update: $\mathbf{b}^{t+1} = \text{HT}_k(\mathbf{b}^t - \nabla f(\mathbf{b}^t))$, where k is a parameter for CRR. Any value $k \geq 2k^*$ suffices to ensure convergence and consistency, as will be clarified in the theoretical analysis. The hard thresholding operator $\text{HT}_k(\cdot)$ is defined below.

Definition 3 (Hard Thresholding)**.** *For any $\mathbf{v} \in \mathbb{R}^n$, let the permutation $\sigma_{\mathbf{v}} \in S_n$ order elements of $\mathbf{v}$ in descending order of their magnitudes. Then for any $k \leq n$, we define the hard thresholding operator as $\widehat{\mathbf{v}} = HT_k(\mathbf{v})$ where $\widehat{\mathbf{v}}_i = \mathbf{v}_i$ if $\sigma_{\mathbf{v}}^{-1}(i) \leq k$ and 0 otherwise.*

We note that CRR functions with a fixed, unit step length, which is convenient in practice as it avoids step length tuning, something most IHT algorithms [12, 13] require. For simplicity of exposition, we will consider only Gaussian ensembles for the RLSR problem i.e. $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \Sigma)$; our proof technique works for general sub-Gaussian ensembles with appropriate distribution dependent parameters. Since CRR interacts with the data only using the projection matrix P_X, for Gaussian ensembles, one can assume without loss of generality that the data points are generated from a spherical Gaussian i.e. $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, I_{d\times d})$. Our analysis will take care of the condition number of the data ensemble whenever it is apparent in the convergence rates.

Before moving to present the consistency and convergence guarantees for CRR, we note that Gaussian ensembles are known to satisfy the SSC/SSS properties with high probability. For instance, in the case of the standard Gaussian ensemble, we have SSC/SSS constants of the order of $\Lambda_m \leq \mathcal{O}\left(m\sqrt{\log \frac{n}{m}} + \sqrt{n}\right)$ and $\lambda_m \geq n - \mathcal{O}\left((n-m)\sqrt{\log \frac{n}{n-m}} + \sqrt{n}\right)$. These results are known from previous works [3, 10] and are reproduced in Appendix A.1.

6 Consistency Guarantees for CRR

Theorem 4. *Let $x_i \in \mathbb{R}^d, 1 \leq i \leq n$ be generated i.i.d. from a Gaussian distribution, let y_i's be generated using (5) for a fixed $\mathbf{w}^*$, and let σ^2 be the noise variance. Also let the number of corruptions k^* be s.t. $2k^* \leq k \leq n/10000$. Then for any $\epsilon, \delta > 0$, with probability at least $1-\delta$, after $\mathcal{O}\left(\log \frac{\|\mathbf{b}^*\|_2}{\sigma k + \epsilon} + \log \frac{n}{d}\right)$ steps,* CRR *ensures that $\|\mathbf{w}^t - \mathbf{w}^*\|_2 \leq \epsilon + \mathcal{O}\left(\frac{\sigma}{\sqrt{\lambda_{\min}(\Sigma)}}\sqrt{\frac{d}{n}\log\frac{nd}{\delta}}\right)$.*

The above result establishes consistency of the CRR method with an error rate of $\tilde{\mathcal{O}}(\sigma\sqrt{d/n})$ that is known to be statistically optimal. It is notable that this optimal rate is being ensured in the presence of gross and unbounded outliers. We reiterate that to the best of our knowledge, this is the first instance of a poly-time algorithm being shown to be consistent for the RLSR problem. It is also notable that the result allows the corruption index to be $k^* = \Omega(n)$, i.e. allows upto a *constant* factor of the total number of data points to be arbitrarily corrupted, while ensuring consistency, which existing results [3, 6, 16] do not ensure.

We pause a bit to clarify some points regarding the result. Firstly we note that the upper bound on recovery error consists of two terms. The first term is ϵ which can be made arbitrarily small simply by executing the CRR algorithm for several iterations. The second term is more crucial and underscores the consistency properties of CRR. The second term is of the form $\mathcal{O}\left(\sigma\sqrt{d\log(nd)/n}\right)$ and is easily seen to vanish with $n \to \infty$ for any constant d, σ. Secondly we note that the result requires $k^* \leq n/20000$ i.e. $\alpha \leq 1/20000$. Although this constant might seem small, we stress that these constants are not the best possible since we preferred analyses that were more accessible. Indeed, in our experiments, we found CRR to be robust to much higher corruption levels than what the Theorem 4 guarantees. Thirdly, we notice that the result requires the CRR to be executed with the corruption index set to a value $k \geq 2k^*$. In practice the value of k can be easily tuned using a simple binary search because of the speed of execution that CRR offers (see Section 7).

For our analysis, we will divide CRR's execution into two phases – a *coarse convergence* phase and a *fine convergence* phase. CRR will enjoy a linear rate of convergence in both phases. However, the coarse convergence analysis will only ensure $\|\mathbf{w}^t - \mathbf{w}^*\|_2 = \mathcal{O}(\sigma)$. The fine convergence phase will then use a much more careful analysis of the algorithm to show that in at most $\mathcal{O}(\log n)$ more

iterations, CRR ensures $\|\mathbf{w}^t - \mathbf{w}^*\|_2 = \tilde{\mathcal{O}}(\sigma\sqrt{d/n})$, thus establishing consistency of the method. Existing methods, such as TORRENT, ensure an error level of $\mathcal{O}(\sigma)$, but no better.

As shorthand notation, let $\boldsymbol{\lambda}^t := (XX^\top)^{-1}X(\mathbf{b}^t - \mathbf{b}^*)$, $\mathbf{g} := (I - P_X)\boldsymbol{\epsilon}$, and $\mathbf{v}^t = X^\top\boldsymbol{\lambda}^t + \mathbf{g}$. Let $S^* := \text{supp}(\mathbf{b}^*)$ be the true locations of the corruptions and $I^t := \text{supp}(\mathbf{b}^t) \cup \text{supp}(\mathbf{b}^*)$.

Coarse convergence: Here we establish a result that guarantees that after a certain number of steps T_0, CRR identifies the corruption vector with a relatively high accuracy and consequently ensures that $\left\|\mathbf{w}^{T_0} - \mathbf{w}^*\right\|_2 \leq \mathcal{O}(\sigma)$.

Lemma 5. *For any data matrix X that satisfies the SSC and SSS properties such that $\frac{2\Lambda_{k+k^*}}{\lambda_n} < 1$, CRR, when executed with $k \geq k^*$, ensures for any $\epsilon, \delta > 0$, with probability at least $1 - \delta$ (over the random Gaussian noise $\boldsymbol{\epsilon}$ in the responses – see (3)) that after $T_0 = \mathcal{O}\left(\log \frac{\|\mathbf{b}^*\|_2}{e_0 + \epsilon}\right)$ steps, $\left\|\mathbf{b}^{T_0} - \mathbf{b}^*\right\|_2 \leq 3e_0 + \epsilon$, where $e_0 = \mathcal{O}\left(\sigma\sqrt{(k+k^*)\log\frac{n}{\delta(k+k^*)}}\right)$ for standard Gaussian designs.*

Using Lemma 12 (see the appendix), we can translate the above result to show that $\left\|\mathbf{w}^{T_0} - \mathbf{w}^*\right\|_2 \leq 0.95\sigma + \epsilon$, assuming $k^* \leq k \leq \frac{n}{150}$. However, Lemma 5 will be more useful in the following fine convergence analysis.

Fine convergence: We now show that CRR progresses further at a linear rate to achieve a consistent solution. In Lemma 6, we show that $\|X(\mathbf{b}^t - \mathbf{b}^*)\|_2$ has a linear decrease for every iteration $t > T_0$ along with a term which is $\tilde{\mathcal{O}}(\sqrt{dn})$. The proof proceeds by showing that for any fixed $\boldsymbol{\lambda}^t$ such that $\|\boldsymbol{\lambda}^t\|_2 \leq \frac{\sigma}{100}$, we obtain a linear decrease in $\|\boldsymbol{\lambda}^{t+1}\|_2 = \|(XX^T)^{-1}X(\mathbf{b}^t - \mathbf{b}^*)\|_2$. We then take a union bound over a fine ϵ-net over all possible values of $\boldsymbol{\lambda}^t$ to obtain the final result.

Lemma 6. *Let $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ be a data matrix consisting of i.i.d. standard normal vectors i.e $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, I_{d\times d})$, and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \cdot I_{n\times n})$ be a standard normal vector of white noise values drawn independently of X. For any $\boldsymbol{\lambda} \in \mathbb{R}^d$ such that $\|\boldsymbol{\lambda}\|_2 \leq \frac{\sigma}{100}$, define $\mathbf{b}^{new} = HT_k(X^\top\boldsymbol{\lambda} + \boldsymbol{\epsilon} + \mathbf{b}^*)$, $\mathbf{z}^{new} = \mathbf{b}^{new} - \mathbf{b}^*$ and $\boldsymbol{\lambda}^{new} = (XX^T)^{-1}X\mathbf{z}^{new}$, where $k \geq 2k^*$, $|supp(\mathbf{b}^*)| \leq k^*$, $k^* \leq n/1000$, and $d \leq n/1000$. Then, with probability at least $1 - 1/n^5$, for every $\boldsymbol{\lambda}$ s.t. $\|\boldsymbol{\lambda}\|_2 \leq \frac{\sigma}{100}$, we have*

$$\|X\mathbf{z}^{new}\|_2 \leq .9n\|\boldsymbol{\lambda}\|_2 + 100\sigma\sqrt{d\cdot n}\log^2 n,$$

$$\|\boldsymbol{\lambda}^{new}\|_2 \leq .91\|\boldsymbol{\lambda}\|_2 + 110\sigma\sqrt{\frac{d}{n}}\log^2 n.$$

Putting all these results together establishes Theorem 4. See Appendix A.2 for a detailed proof. Note that while both the coarse/fine stages offer a linear rate of convergence, it is the fine phase that ensures consistency. Indeed, the coarse phase only acts as a sort of good-enough initialization. Several results in non-convex optimization assume a nice initialization "close" to the optimum (alternating minimization, EM etc). In our case, we have a happy situation where the initialization and main algorithms are one and the same. Note that we could have actually used other algorithms e.g. TORRENT to perform initialization as well since TORRENT [3, Theorem 10] essentially offers the same (weak) guarantee as Lemma 5 offers.

7 Experiments

Experiments were carried out on synthetically generated linear regression datasets with corruptions. All implementations were done in Matlab and were run on a single core 2.4GHz machine with 8GB RAM. The experiments establish the following: 1) CRR gives consistent estimates of the regression model, especially in situations with a large number of corruptions where the ordinary least squares estimator fails catastrophically, 2) CRR scales better to large datasets than the TORRENT-FC algorithm of [3] (upto $5\times$ faster) and the Extended Lasso algorithm of [17] (upto $20\times$ faster). The main reason behind this speedup is that TORRENT keeps changing its mind on which active set of points it wishes to work with. Consequently, it expends a lot of effort processing each active set. CRR on the other hand does not face such issues since it always works with the entire set of points.

Data: The model $\mathbf{w}^* \in \mathbb{R}^d$ was chosen to be a random unit norm vector. The data was generated as $\mathbf{x}_i \sim \mathcal{N}(0, I_d)$. The k^* responses to be corrupted were chosen uniformly at random and the

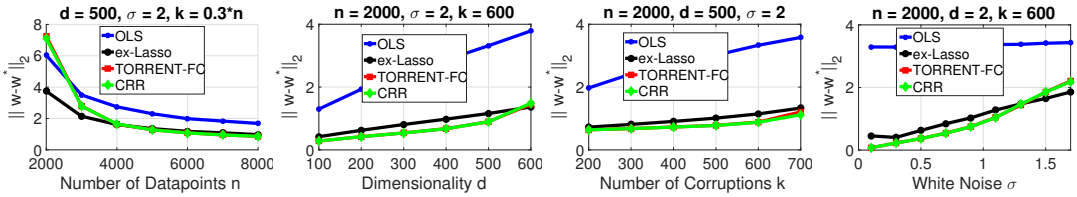

Figure 1: Variation of recovery error with varying number of data points n, dimensionality d, number of corruptions k^* and white noise variance σ. CRR and TORRENT show better recovery properties than the non-robust OLS on all experiments. Extended Lasso offers comparable or slightly worse recovery in most settings. Figure 1(a) ascertains the $\widetilde{\mathcal{O}}\left(\sqrt{1/n}\right)$-consistency of CRR as is shown in the theoretical analysis.

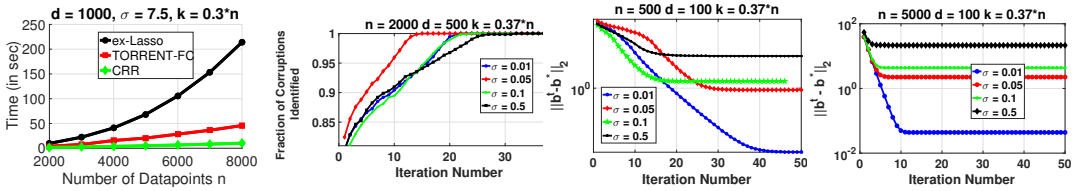

Figure 2: Figure 2(a) show the average CPU run times of CRR, TORRENT and Extended Lasso with varying sample sizes. CRR can be an order of magnitude faster than TORRENT and Extended Lasso on problems in 1000 dimensions while ensuring similar recovery properties.. Figure 2(b), 2(c) and 2(d) show that CRR eventually not only captures the total mass of corruptions, but also does support recovery of the corrupted points in an accurate manner. With every iteration, CRR improves upon its estimate of $\mathbf{b}^*$ and provides cleaner points for estimation of $\mathbf{w}$. CRR is also able to very effectively utilize larger data sets to offer much faster convergence. Notice the visibly faster convergence in Figure 2(d) which uses 10x more points than figure (c).

value of the corruptions was sets as $b_i^* \sim \text{Unif}\,(10, 20)$. Responses were then generated as $y_i = \langle \mathbf{x}_i, \mathbf{w}^* \rangle + \eta_i + b_i^*$ where $\eta_i \sim \mathcal{N}(0, \sigma^2)$. All reported results were averaged over 20 randomly trials.

Evaluation Metric: We measure the performance of various algorithms using the standard L_2 error: $r_{\widehat{\mathbf{w}}} = \|\widehat{\mathbf{w}} - \mathbf{w}^*\|_2$. For the timing experiments, we deemed an algorithm to converge on an instance if it obtained a model $\mathbf{w}^t$ such that $\|\mathbf{w}^t - \mathbf{w}^{t-1}\|_2 \leq 10^{-4}$.

Baseline Algorithms: CRR was compared to two baselines 1) the Ordinary Least Squares (OLS) estimator which is oblivious of the presence of any corruptions in the responses, 2) the TORRENT algorithm of [3] which is a recently proposed method for performing robust least squares regression, and 3) the Extended Lasso (ex-Lasso) approach of [15] for which we use the FISTA implementation of [23] and choose the regularization paramaters for our model data as mentioned by the authors.

Recovery Properties & Timing: CRR, TORRENT and ex-Lasso were found to be competitive, and offered much lower residual errors $\|\mathbf{w} - \mathbf{w}^*\|_2$ than the non-robust OLS method when varying dataset size Figure 1(a), dimensionality Figure 1(b), number of corrupted responses Figure 1(c), and magnitude of white noise Figure 1(d). In terms of scaling properties, CRR exhibited faster runtimes than TORRENT-FC as depicted in Figure 2(a). CRR can be upto $5\times$ faster than TORRENT and upto $20\times$ faster than ex-Lasso on problems of 1000 dimensions. Figure 2(a) suggests that executing both TORRENT and ex-Lasso becomes very expensive with an order of magnitude increase in the dimension parameter of the problem while CRR scales gracefully. Also, Figures 2(c) and 2(d) show the variation of $\|\mathbf{b}^t - \mathbf{b}^*\|_2$ for various values of the noise parameter σ. The plot depicts the fact that as $\sigma \to 0$, CRR is correctly able to identify all the corrupted points and estimate the level of corruption correctly, thereby returning the exact solution $\mathbf{w}^*$. Notice that in Figure 2(d) which utilizes more data points, CRR offers uniformly faster convergence across all white noise levels.

Choice of Potential Function: In Lemmata 5 and 6, we show that $\|\mathbf{b}^t - \mathbf{b}^*\|_2$ decreases with every iteration. Figures 2(c) and (d) back this theoretical statement by showing that CRR's estimate of $\mathbf{b}^*$ improves with every iteration. Along with estimating the magnitude of $\mathbf{b}^*$, Figure 2(b) shows that CRR is also able to correctly identify the support of the corrupted points with increasing iterations.

8 Response to Reviewer Comments

We are thankful to the reviewers for their comments aimed at improving the manuscript. Below we offer some clarifications regarding the same.

1. We have fixed all typographical errors pointed out in the reviews.
2. We have included additional references as pointed out in the reviews.
3. We have improved the presentation of the statement of the results to make them more crisp.
4. We note that CRR's reduction of the robust recovery problem to sparse recovery is not only novel, but also one that offers impressive speedups in practice over the existing TORRENT algorithm [3]. However, note that the reduction to sparse recovery actually hides a sort of "fully-corrective" step wherein the optimal model for a particular corruption estimate is used internally in the formulation. Thus, CRR is implicitly a fully corrective algorithm as well.
5. We agree with the reviewers that further efforts are needed to achieve results with sharper constants. For example, CRR offers robustness upto a breakdown fraction of 1/10000 which, although a constant, nevertheless leaves room for improvement. Having shown for the first time that tolerating a non-trivial, universally constant fraction of corruptions is possible in polynomial time, it is indeed encouraging to study how far can the breakdown point be pushed for various families of algorithms.
6. Our current efforts are aimed at solving the robust sparse recovery problems in high dimensional settings in a statistically consistent manner, as well as extending the consistency properties established in this paper for non-Gaussian, for example fixed, designs.

Acknowledgments

The authors thank the reviewers for useful comments. PKar is supported by the Deep Singh and Daljeet Kaur Faculty Fellowship and the Research-I foundation at IIT Kanpur, and thanks Microsoft Research India and Tower Research for research grants.

References

[1] J. Ámos Višek. The least trimmed squares. Part I: Consistency. *Kybernetika*, 42:1–36, 2006.

[2] J. Ámos Višek. The least trimmed squares. Part II: $\sqrt{n}$-consistency. *Kybernetika*, 42:181–202, 2006.

[3] K. Bhatia, P. Jain, and P. Kar. Robust Regression via Hard Thresholding. In *Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.

[4] E. J. Candès, X. Li, and J. Wright. Robust Principal Component Analysis? *Journal of the ACM*, 58(1):1–37, 2009.

[5] M. Charikar, J. Steinhardt, and G. Valiant. Learning from Untrusted Data. arXiv:1611.02315 [cs.LG], 2016.

[6] Y. Chen, C. Caramanis, and S. Mannor. Robust Sparse Regression under Adversarial Corruption. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013.

[7] Y. Chen and A. S. Dalalyan. Fused sparsity and robust estimation for linear models with unknown variance. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS)*, 2012.

[8] Y. Cherapanamjeri, K. Gupta, and P. Jain. Nearly-optimal Robust Matrix Completion. arXiv:1606.07315 [cs.LG], 2016.

[9] F. Cucker and S. Smale. On the Mathematical Foundations of Learning. *Bulleting of the American Mathematical Society*, 39(1):1–49, 2001.

[10] M. A. Davenport, J. N. Laska, P. T. Boufounos, and R. G. Baraniuk. A Simple Proof that Random Matrices are Democratic. Technical Report TREE0906, Rice University, Department of Electrical and Computer Engineering, 2009.

[11] J. Feng, H. Xu, S. Mannor, and S. Yan. Robust Logistic Regression and Classification. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems (NIPS)*, 2014.

[12] R. Garg and R. Khandekar. Gradient Descent with Sparsification: An Iterative Algorithm for Sparse Recovery with Restricted Isometry Property. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, 2009.

[13] P. Jain, A. Tewari, and P. Kar. On Iterative Hard Thresholding Methods for High-dimensional M-estimation. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems (NIPS)*, 2014.

[14] B. McWilliams, G. Krummenacher, M. Lucic, and J. M. Buhmann. Fast and Robust Least Squares Estimation in Corrupted Linear Models. In *28th Annual Conference on Neural Information Processing Systems (NIPS)*, 2014.

[15] N. M. Nasrabadi, T. D. Tran, and N. Nguyen. Robust Lasso with Missing and Grossly Corrupted Observations. In *Advances in Neural Information Processing Systems*, pages 1881–1889, 2011.

[16] N. H. Nguyen and T. D. Tran. Exact recoverability from dense corrupted observations via ℓ_1-minimization. *IEEE transactions on information theory*, 59(4):2017–2035, 2013.

[17] N. H. Nguyen and T. D. Tran. Robust Lasso With Missing and Grossly Corrupted Observations. *IEEE Transaction on Information Theory*, 59(4):2036–2058, 2013.

[18] P. J. Rousseeuw. Least Median of Squares Regression. *Journal of the American Statistical Association*, 79(388):871–880, 1984.

[19] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. John Wiley and Sons, 1987.

[20] C. Studer, P. Kuppinger, G. Pope, and H. Bölcskei. Recovery of Sparsely Corrupted Signals. *IEEE Transaction on Information Theory*, 58(5):3115–3130, 2012.

[21] J. Wright and Y. Ma. Dense Error Correction via ℓ^1 Minimization. *IEEE Transactions on Information Theory*, 56(7):3540–3560, 2010.

[22] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust Face Recognition via Sparse Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.

[23] A. Y. Yang, Z. Zhou, A. G. Balasubramanian, S. S. Sastry, and Y. Ma. Fast ℓ_1-minimization algorithms for robust face recognition. *IEEE Transactions on Image Processing*, 22(8):3234–3246, 2013.

A Supplementary Material for Consistent Robust Regression

A.1 SSC/SSS guarantees

In this section we restate some results from [3] which are required for the convergence analysis of the RLSR problem. Similar variants are known from other works, e.g. [10], as well.

Definition 7. *A random variable $x \in \mathbb{R}$ is called sub-Gaussian if the following quantity is finite*

$$\sup_{p \geq 1} p^{-1/2} \left(\mathbb{E} \left[|x|^p \right] \right)^{1/p}.$$

Moreover, the smallest upper bound on this quantity is referred to as the sub-Gaussian norm of x and denoted as $\|x\|_{\psi_2}$.

Definition 8. *A vector-valued random variable $\mathbf{x} \in \mathbb{R}^p$ is called sub-Gaussian if its unidimensional marginals $\langle \mathbf{x}, \mathbf{v} \rangle$ are sub-Gaussian for all $\mathbf{v} \in S^{p-1}$. Moreover, its sub-Gaussian norm is defined as follows*

$$\|X\|_{\psi_2} := \sup_{\mathbf{v} \in S^{p-1}} \|\langle \mathbf{x}, \mathbf{v} \rangle\|_{\psi_2}$$

Lemma 9. *Let $X \in \mathbb{R}^{p \times n}$ be a matrix whose columns are sampled i.i.d from a standard Gaussian distribution i.e. $\mathbf{x}_i \sim \mathcal{N}(0, I)$. Then for any $\epsilon > 0$, with probability at least $1 - \delta$, X satisfies*

$$\lambda_{\max}(XX^\top) \leq n + (1 - 2\epsilon)^{-1} \sqrt{cnp + c'n \log \frac{2}{\delta}}$$
$$\lambda_{\min}(XX^\top) \geq n - (1 - 2\epsilon)^{-1} \sqrt{cnp + c'n \log \frac{2}{\delta}},$$

where $c = 24e^2 \log \frac{3}{\epsilon}$ and $c' = 24e^2$.

Theorem 10. *Let $X \in \mathbb{R}^{p \times n}$ be a matrix whose columns are sampled i.i.d from a standard Gaussian distribution i.e. $\mathbf{x}_i \sim \mathcal{N}(0, I)$. Then for any $k > 0$, with probability at least $1 - \delta$, the matrix X satisfies the SSC and SSS properties with constants*

$$\Lambda_k \leq k \left(1 + 3e \sqrt{6 \log \frac{en}{k}} \right) + \mathcal{O}\left(\sqrt{np + n \log \frac{1}{\delta}} \right)$$
$$\lambda_k \geq n - (n - k) \left(1 + 3e \sqrt{6 \log \frac{en}{n-k}} \right) - \Omega\left(\sqrt{np + n \log \frac{1}{\delta}} \right).$$

Lemma 11. *Let $X \in \mathbb{R}^{p \times n}$ be a matrix with columns sampled from some sub-Gaussian distribution with sub-Gaussian norm K and covariance Σ. Then, for any $\delta > 0$, with probability at least $1 - \delta$, each of the following statements holds true:*

$$\lambda_{\max}(XX^\top) \leq \lambda_{\max}(\Sigma) \cdot n + C_K \cdot \sqrt{pn} + t\sqrt{n}$$
$$\lambda_{\min}(XX^\top) \geq \lambda_{\min}(\Sigma) \cdot n - C_K \cdot \sqrt{pn} - t\sqrt{n},$$

where $t = \sqrt{\frac{1}{c_K} \log \frac{2}{\delta}}$, and c_K, C_K are absolute constants that depend only on the sub-Gaussian norm K of the distribution.

A.2 Convergence Proofs for CRR

Theorem 4. *Let $x_i \in \mathbb{R}^d, 1 \leq i \leq n$ be generated i.i.d. from a Gaussian distribution, let y_i's be generated using (5) for a fixed $\mathbf{w}^*$, and let σ^2 be the noise variance. Also let the number of corruptions k^* be s.t. $2k^* \leq k \leq n/10000$. Then for any $\epsilon, \delta > 0$, with probability at least $1-\delta$, after $\mathcal{O}\left(\log \frac{\|\mathbf{b}^*\|_2}{\sigma k + \epsilon} + \log \frac{n}{d} \right)$ steps,* CRR *ensures that $\|\mathbf{w}^t - \mathbf{w}^*\|_2 \leq \epsilon + \mathcal{O}\left(\frac{\sigma}{\sqrt{\lambda_{\min}(\Sigma)}} \sqrt{\frac{d}{n} \log \frac{nd}{\delta}} \right)$.*

Proof. Using Lemma 5, after $\mathcal{O}\left(\log \frac{\|\mathbf{b}^*\|_2}{\sigma k + \epsilon} \right)$ steps, we get $\mathbf{b}^t$ s.t. $\|\boldsymbol{\lambda}^t\| = \|(X^T X)^{-1} X(\mathbf{b}^t - \mathbf{b}^*)\| \leq \frac{90(k^* + \sqrt{nd})}{n} \|\mathbf{b}^t - \mathbf{b}^*\| \leq \frac{90(k^* + \sqrt{nd})}{n} (6\sigma\sqrt{k^* \log n}) \leq \sigma/100$ as long as $n \geq k/1000$ and $n \geq d/1000$.

Now, recall that $\mathbf{b}_{t+1} = HT_k(\mathbf{b}^* + X^T(\boldsymbol{\lambda}^t - P_X\epsilon) + \epsilon)$. Now, using Lemma 13 with $\boldsymbol{\lambda} = \boldsymbol{\lambda}^t - P_X\epsilon$, we get:

$$\left\|\boldsymbol{\lambda}^{t+1}\right\|_2 \leq 0.91 \left\|\boldsymbol{\lambda}^t\right\|_2 + 110\sigma\sqrt{\frac{d}{n}}\log^2 n,$$

which ensures a linear convergence of the terms $\left\|\boldsymbol{\lambda}^t\right\|_2$ to a value $\epsilon + \mathcal{O}\left(\sigma\sqrt{\frac{d}{n}\log\frac{nd}{\delta}}\right)$. □

Lemma 5. *For any data matrix X that satisfies the SSC and SSS properties such that $\frac{2\Lambda_{k+k^*}}{\lambda_n} < 1$, CRR, when executed with $k \geq k^*$, ensures for any $\epsilon, \delta > 0$, with probability at least $1-\delta$ (over the random Gaussian noise $\boldsymbol{\epsilon}$ in the responses – see (3)) that after $T_0 = \mathcal{O}\left(\log\frac{\|\mathbf{b}^*\|_2}{e_0+\epsilon}\right)$ steps, $\left\|\mathbf{b}^{T_0} - \mathbf{b}^*\right\|_2 \leq 3e_0 + \epsilon$, where $e_0 = \mathcal{O}\left(\sigma\sqrt{(k+k^*)\log\frac{n}{\delta(k+k^*)}}\right)$ for standard Gaussian designs.*

Proof. We start with the update step in CRR, and use the fact that $\mathbf{y} = X^\top\mathbf{w}^* + \mathbf{b}^* + \boldsymbol{\epsilon}$ to rewrite the update as

$$\mathbf{b}^{t+1} \leftarrow \text{HT}_k(P_X\mathbf{b}^t + (I - P_X)(X^\top\mathbf{w}^* + \mathbf{b}^* + \boldsymbol{\epsilon})).$$

Since $X^\top = P_X X^\top$, we get, using the notation set up before,

$$\mathbf{b}^{t+1} \leftarrow \text{HT}_k(\mathbf{b}^* + X^\top\boldsymbol{\lambda}^t + \mathbf{g}).$$

Since $k \geq k^*$, using the properties of the hard thresholding step gives us

$$\left\|\mathbf{b}^{t+1}_{I^{t+1}} - (\mathbf{b}^*_{I^{t+1}} + X^\top_{I^{t+1}}\boldsymbol{\lambda}^t + \mathbf{g}_{I^{t+1}})\right\|_2 \leq \left\|\mathbf{b}^*_{I^{t+1}} - (\mathbf{b}^*_{I^{t+1}} + X^\top_{I^{t+1}}\boldsymbol{\lambda}^t + \mathbf{g}_{I^{t+1}})\right\|_2 = \left\|X^\top_{I^{t+1}}\boldsymbol{\lambda}^t + \mathbf{g}_{I^{t+1}}\right\|_2.$$

This, upon applying the triangle inequality, gives us

$$\left\|\mathbf{b}^{t+1} - \mathbf{b}^*\right\|_2 \leq 2\left\|X^\top_{I^{t+1}}\boldsymbol{\lambda}^t + \mathbf{g}_{I^{t+1}}\right\|_2.$$

Now, using the SSC and SSS properties of X, we can show that $\left\|X^\top_{I^{t+1}}\boldsymbol{\lambda}^t\right\|_2 = \left\|X^\top_{I^{t+1}}(XX^\top)^{-1}X_{I^t}(\mathbf{b}^t - \mathbf{b}^*)\right\|_2 \leq \frac{\Lambda_{k+k^*}}{\lambda_n}\left\|\mathbf{b}^t - \mathbf{b}^*\right\|_2$.

Since $\boldsymbol{\epsilon}$ is a Gaussian vector, using tail bounds for Chi-squared random variables (for example, see Lemma 20 in [3]), for any set S of size $k+k^*$, we have with probability at least $1-\delta$,$\|\boldsymbol{\epsilon}_S\|_2^2 \leq \sigma^2(k+k^*) + 2e\sigma^2\sqrt{6(k+k^*)\log\frac{1}{\delta}}$. Taking a union bound over all sets of size $(k+k^*)$ and $\binom{n}{k} \leq \left(\frac{en}{k}\right)^k$ gives us, with probability at least $1-\delta$, for all sets S of size at most $(k+k^*)$,

$$\|\boldsymbol{\epsilon}_S\|_2 \leq \sigma\sqrt{(k+k^*)}\sqrt{1 + 2e\sqrt{6\log\frac{en}{\delta(k+k^*)}}}$$

Using tail bounds on Gaussian random variables[3], we can also show that for every i, with probability at least $1-\delta$, we have $\|(X\epsilon)_i\|_2 \leq \sigma\left\|(X^\top)_i\right\|_2\sqrt{2\log\frac{1}{\delta}}$. Taking a union bound gives us, with the same confidence, $\|X\boldsymbol{\epsilon}\|_2^2 \leq 2\sigma^2\|X\|_F^2\log\frac{d}{\delta} \leq 2\sigma^2 d\Lambda_n\log\frac{d}{\delta}$. This allows us to bound $\|\mathbf{g}_{I^{t+1}}\|_2$

$$\begin{aligned}
\|\mathbf{g}_{I^{t+1}}\|_2 &= \left\|\boldsymbol{\epsilon}_{I^{t+1}} - X^\top_{I^{t+1}}(XX^\top)^{-1}X\boldsymbol{\epsilon}\right\|_2 \\
&\leq \sigma\sqrt{(k+k^*)}\sqrt{1 + 2e\sqrt{6\log\frac{en}{\delta(k+k^*)}}} + \sigma\frac{\sqrt{\Lambda_{k+k^*}\Lambda_n}}{\lambda_n}\sqrt{2d\log\frac{d}{\delta}} \\
&\leq \underbrace{\sigma\sqrt{(k+k^*)}\sqrt{1 + 2e\sqrt{6\log\frac{en}{\delta(k+k^*)}}}}_{e_0}\left(1 + \sqrt{\frac{2d}{n}\log\frac{d}{\delta}}\right) \\
&= 1.0003e_0,
\end{aligned}$$

[3] $\frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-t^2/2}dt \leq \frac{1}{\sqrt{2\pi}}\int_x^\infty \frac{t}{x}e^{-t^2/2}dt = \frac{1}{x\sqrt{2\pi}}e^{-x^2/2}$

where the second last step is true for Gaussian designs and sufficiently large enough n. Note that e_0 does not depend on the iterates and is thus, a constant. This gives us

$$\left\|\mathbf{b}^{t+1}-\mathbf{b}^*\right\|_2 \leq \frac{2\Lambda_{k+k^*}}{\lambda_n}\left\|\mathbf{b}^t-\mathbf{b}^*\right\|_2 + 2.0006e_0.$$

For data matrices sampled from Gaussian ensembles, whose SSC and SSS properties will be established later, assuming $n \geq d\log d$, we have $e_0 = \mathcal{O}\left(\sigma\sqrt{(k+k^*)\log\frac{n}{\delta(k+k^*)}}\right)$. Thus, if $\frac{2\Lambda_{k+k^*}}{\lambda_n} < 1$, then in $T_0 = \mathcal{O}\left(\log\frac{\|\mathbf{b}^*\|_2}{e_0+\epsilon}\right)$ steps, CRR ensures that $\left\|\mathbf{b}^{T_0}-\mathbf{b}^*\right\|_2 \leq 2.0009e_0+\epsilon$. □

Lemma 12. *Let $\lambda_{\min}(\Sigma)$ be the smallest eigenvalue of the covariance matrix of the distribution $\mathcal{N}(\mathbf{0},\Sigma)$ that generates the data points. Then at any time instant t, we have $\|\mathbf{w}^t-\mathbf{w}^*\|_2 \leq \frac{2}{\sqrt{\lambda_{\min}(\Sigma)}}\left(2\sigma\sqrt{\frac{d}{n}\log\frac{d}{\delta}}+\left\|\boldsymbol{\lambda}^t\right\|_2\right)$.*

Proof. As described in Algorithm 1, $\mathbf{w}^t = (XX^\top)^{-1}X(\mathbf{y}-\mathbf{b}^t) = \mathbf{w}^* + (XX^\top)^{-1}X(\boldsymbol{\epsilon}+\mathbf{b}^*-\mathbf{b}^t)$. If we let $\overline{X} = \Sigma^{-1/2}X$, we get:

$$\begin{aligned}
\left\|\mathbf{w}^t-\mathbf{w}^*\right\|_2 &\leq \frac{1}{\sqrt{\lambda_{\min}(XX^\top)}}\left\|X^\top(\mathbf{w}^t-\mathbf{w}^*)\right\|_2 \\
&\leq \frac{1}{\sqrt{n\lambda_{\min}(\Sigma)-C_\Sigma\sqrt{n}}}\left\|X^\top(\mathbf{w}^t-\mathbf{w}^*)\right\|_2 \\
&\leq \frac{1}{\sqrt{n\lambda_{\min}(\Sigma)-C_\Sigma\sqrt{n}}}\left\|X^\top(XX^\top)^{-1}X(\boldsymbol{\epsilon}+\mathbf{b}^*-\mathbf{b}^t)\right\|_2 \\
&\leq \frac{1}{\sqrt{n\lambda_{\min}(\Sigma)-C_\Sigma\sqrt{n}}}\left\|\overline{X}^\top(\overline{XX}^\top)^{-1}\overline{X}(\boldsymbol{\epsilon}+\mathbf{b}^*-\mathbf{b}^t)\right\|_2 \\
&\leq \frac{\sqrt{\Lambda_n}}{\sqrt{n\lambda_{\min}(\Sigma)-C_\Sigma\sqrt{n}}}\left\|(\overline{XX}^\top)^{-1}\overline{X}(\boldsymbol{\epsilon}+\mathbf{b}^*-\mathbf{b}^t)\right\|_2 \\
&\leq \frac{2}{\sqrt{\lambda_{\min}(\Sigma)}}\left(2\sigma\sqrt{\frac{d}{n}\log\frac{d}{\delta}}+\left\|\boldsymbol{\lambda}^t\right\|_2\right),
\end{aligned}$$

where the second step follows from results on eigenvalue bounds for data matrices drawn from non-spherical Gaussians, where C_Σ is a constant dependent on the subGaussian norm of the distribution. The fourth step is for sake of convenience alone and can be bypassed. The step uses the fact that even though X may be sampled from a non-spherical Gaussian $\mathcal{N}(\mathbf{0},\Sigma)$, the quantity $X^\top(XX^\top)^{-1}X$ is distributed as $\overline{X}^\top(\overline{XX}^\top)^{-1}\overline{X}$ where $\overline{X}$ is sampled from a spherical Gaussian $\mathcal{N}(\mathbf{0},I)$. The last step assumes $n \geq \frac{2C_\Sigma}{\lambda_{\min}(\Sigma)}$ and uses the proof technique used in Lemma 5 to get

$$\left\|(\overline{XX}^\top)^{-1}\overline{X}\boldsymbol{\epsilon}\right\|_2 \leq \sigma\frac{\sqrt{\Lambda_n}}{\lambda_n}\sqrt{2d\log\frac{d}{\delta}} \leq 2\sigma\sqrt{\frac{d}{n}\log\frac{d}{\delta}}.$$ □

Lemma 6. *Let $X=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n]$ be a data matrix consisting of i.i.d. standard normal vectors i.e $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0},I_{d\times d})$, and $\boldsymbol{\epsilon} \sim N(0,\sigma^2\cdot I_{n\times n})$ be standard normal vector drawn independently of X. For any $\boldsymbol{\lambda}\in\mathbb{R}^d$ such that $\|\boldsymbol{\lambda}\|_2 \leq \frac{\sigma}{100}$, define $\mathbf{z} = HT_k(X^\top\boldsymbol{\lambda}+\boldsymbol{\epsilon}+\mathbf{b}^*)-\mathbf{b}^*$, where $k=2k^*$ and $|supp(\mathbf{b}^*)| \leq k^*$. Also let $k^* \leq n/1000$ and $d \leq n/1000$. Then, the following holds (w.p. $\geq 1-1/n^5$):*

$$\|X\mathbf{z}\| \leq .9n\|\boldsymbol{\lambda}\| + 300\sigma\sqrt{d\cdot n}\log^2 n$$

Proof. We first decompose $\|X\mathbf{z}\|$ as:

$$\|X\mathbf{z}\|_2^2 = \frac{1}{\|\boldsymbol{\lambda}\|_2^2}(\boldsymbol{\lambda}^TX\mathbf{z})^2 + \max_{\mathbf{v},\|\mathbf{v}\|_2=1,\mathbf{v}^T\boldsymbol{\lambda}=0}(\mathbf{v}^TX\mathbf{z})^2. \tag{7}$$

We now consider the first term above. Let $\tau_k > 0$ be such that the k largest elements (in magnitude) of $\mathbf{b} = X^\top\boldsymbol{\lambda}+\boldsymbol{\epsilon}+\mathbf{b}^*$ are all greater than τ_k and the $(n-k)$ smallest elements (in magnitude) of $\mathbf{b}$ are all less than τ_k.

That is,

$$\begin{aligned}\boldsymbol{\lambda}^\top X\mathbf{z} &= \sum_j (\boldsymbol{\lambda}^\top \mathbf{x}_j)(\mathbb{I}\left\{|\mathbf{x}_j^\top \boldsymbol{\lambda} + b_j^* + \epsilon_j| > \tau_k\right\}(b_j^* + \epsilon_j + \mathbf{x}_j^\top \boldsymbol{\lambda}) - b_j^*) \\ &= \sum_j (\boldsymbol{\lambda}^\top \mathbf{x}_j)^2 \mathbb{I}\left\{|\mathbf{x}_j^\top \boldsymbol{\lambda} + b_j^* + \epsilon_j| > \tau_k\right\} + \sum_j (\boldsymbol{\lambda}^\top \mathbf{x}_j)(\mathbb{I}\left\{|\mathbf{x}_j^\top \boldsymbol{\lambda} + b_j^* + \epsilon_j| > \tau_k\right\}(b_j^* + \epsilon_j) - b_j^*) \\ &\leq 20(k\log n/k + \sqrt{dn\log n})\|\boldsymbol{\lambda}\|^2 + \sum_j (\boldsymbol{\lambda}^\top \mathbf{x}_j)(\mathbb{I}\left\{|\mathbf{x}_j^\top \boldsymbol{\lambda} + b_j^* + \epsilon_j| > \tau_k\right\}(b_j^* + \epsilon_j) - b_j^*),\end{aligned} \tag{8}$$

where the last inequality follows from Theorem 10 and hold with probability $\geq 1 - 1/n^d$.

Now to bound the second term above, our approach is to show that each of the j-th term is small in expectation and then use tail bounds to obtain the final bound. Unfortunately, as $\boldsymbol{\lambda}$ and τ_k can depend on random variables X and ϵ, we cannot bound expectation as well as apply Hoeffding style tail bounds directly.

Instead, we take the standard approach of using ϵ-nets and union bounds. That is, we form $\gamma_{\boldsymbol{\lambda}}$-net over the set $B := \mathcal{B}_2(\mathbf{0}, \frac{\sigma}{100})$ (i.e. the ball of radius $\frac{\sigma}{100} \geq \|\boldsymbol{\lambda}\|$ centered at the origin in d-dimensions) and similarly a unit dimensional γ_τ-net over $[\sigma/2\sqrt{\log(n/k)},\ 30\sigma\sqrt{\log(n/k)}]$; we will provide the justification for selecting the above given range for τ below.

Recall that we hard-threshold $k = 2k^*$ entries of $X^\top\boldsymbol{\lambda} + \boldsymbol{\epsilon} + \mathbf{b}^*$. Moreover, $|supp(\mathbf{b}^*)| \leq k^*$. Now, let $S = \{j\ s.t.\ |x_j^\top \boldsymbol{\lambda} + \epsilon_j| \geq \tau_k$. Note that $|S| \geq k - k^*$ as τ_k ensures that top-k elements of $X^\top\boldsymbol{\lambda} + \boldsymbol{\epsilon} + \mathbf{b}^*$ are selected. Similarly, define $\widehat{S} = \{j\ s.t.\ |x_j^\top \boldsymbol{\lambda} + \epsilon_j| < \tau_k$. Again, note that $|\widehat{S}| \leq n - k + k^*$ as τ_k ensures that *only* k elements of $X^\top\boldsymbol{\lambda} + \boldsymbol{\epsilon} + \mathbf{b}^*$ are selected. Hence,

$$\frac{1}{|\widehat{S}|}\sum_{j\in\widehat{S}}(x_j^\top\boldsymbol{\lambda} + \epsilon_j)^2 \leq \tau_k^2 \leq \frac{1}{|S|}\sum_{j\in S}(x_j^\top\boldsymbol{\lambda} + \epsilon_j)^2. \tag{9}$$

Using the fact that $k \leq n/1000$ and using SSC/SSS bounds with $d < n/1000$ (apply Theorem 10 part 1 with $k - k^*$ and part 2 with $n - k + k^*$), we get with probability at least $1 - \frac{1}{n^6}$, $\tau_k \in [.5\sigma\sqrt{\log n/k}, 30\sigma\sqrt{\log n/k}]$.

Now, let $\widetilde{\boldsymbol{\lambda}}$ and $\widetilde{\tau}$ be the closest point from the $\gamma_{\boldsymbol{\lambda}}$-net to $\boldsymbol{\lambda}$ and τ_k, respectively. Now,

$$\left|\sum_j \mathbb{I}\left\{|\mathbf{x}_j^\top\boldsymbol{\lambda} + \epsilon_j + b_j^*| \geq \tau_k\right\} - \mathbb{I}\left\{|\mathbf{x}_j^\top\widetilde{\boldsymbol{\lambda}} + \epsilon_j + b_j^*| \geq \widetilde{\tau}\right\}\right| \leq \sum_j \mathbb{I}\left\{|\mathbf{x}_j^\top\widetilde{\boldsymbol{\lambda}} + \epsilon_j + b_j^* - \widetilde{\tau}| \leq \gamma_\tau + \|\mathbf{x}_j\|\gamma_{\boldsymbol{\lambda}}\right\}. \tag{10}$$

Now note that $\nu_j = \mathbf{x}_j^\top\widetilde{\boldsymbol{\lambda}} + \epsilon_j + b_j^* \sim \mathcal{N}(b_j^*, \|\boldsymbol{\lambda}\|^2 + \sigma^2)$ and each ν_j are independent of each other (recall that $\mathbf{b}^*$ is generated independently of $\mathbf{x}_i$ and $\boldsymbol{\lambda}$ is a fixed vector). Also, let $\gamma = \gamma_\tau + \|\mathbf{x}_j\|\gamma_{\boldsymbol{\lambda}}$. Recall that since the Gaussian distribution has a density function that always takes values less than unity at every point, for $g \sim \mathcal{N}(\mu, 1)$, $\Pr(|g - \tau| \leq \zeta) \leq 2\zeta$ for all τ, μ and for all $\zeta > 0$. Hence, for a fixed $\widetilde{\boldsymbol{\lambda}}$ and $\widetilde{\tau}$, we have:

$$\Pr\left(\sum_j \mathbb{I}\left\{|\nu_j - \widetilde{\tau}| \leq \gamma\right\} \geq 4d\right) = \Pr\left(\sum_j \mathbb{I}\left\{\left|\frac{\nu_j}{\widetilde{\sigma}} - \frac{\widetilde{\tau}}{\widetilde{\sigma}}\right| \leq \frac{\gamma}{\widetilde{\sigma}}\right\} \geq 4d\right) \leq \binom{n}{4d}\left(\frac{2\gamma}{\widetilde{\sigma}}\right)^{4d} \leq \binom{n}{4d}\left(\frac{2\gamma}{\sigma}\right)^{4d}. \tag{11}$$

In the above $\widetilde{\sigma} = \sqrt{\sigma^2 + \|\boldsymbol{\lambda}\|_2^2} \geq \sigma$. At this point, we recall that using standard results on covering numbers [9], we know that the net on λ values needs at most $(\frac{4\sigma}{100\gamma_{\boldsymbol{\lambda}}})^d$ elements and the net on τ values needs at most $\frac{2.5\sigma\sqrt{\log(n/k)}}{\gamma_\tau}$ elements. Thus, using a union bound, we have *for all* $\widetilde{\boldsymbol{\lambda}}$ in the $\gamma_{\boldsymbol{\lambda}}$-net of B and $\widetilde{\tau} \in \gamma_\tau$-net of $[.5\sigma\sqrt{\log(n/k)}, 30\sigma\sqrt{\log(n/k)}]$:

$$\Pr\left(\sum_j \mathbb{I}\left\{|\nu_j - \widetilde{\tau}| \leq \gamma\right\} \geq 4d\right) \leq \binom{n}{4d}\left(\frac{2\gamma}{\sigma}\right)^{4d} \cdot \left(\frac{4\sigma}{100\gamma_{\boldsymbol{\lambda}}}\right)^d \cdot \left(\frac{2.5\sigma\sqrt{\log(n/k)}}{\gamma_\tau}\right) \leq \frac{2\sqrt{\log n}}{n^{5d}}, \tag{12}$$

where the last inequality follows by setting $\gamma_\tau = \mathcal{O}\left(\frac{\sigma}{n^3}\right)$, $\gamma_{\boldsymbol{\lambda}} = \frac{\sigma}{2000dn^3 \log n}$. These bounds were set as such since $\|\boldsymbol{\lambda}\|_2 \leq \sigma/100$, and by using standard tail bounds on Chi-squared random variables, with probability at least $1 - 1/n^d$, we have $\max_i \|\mathbf{x}_i\|_2 \leq 20d \log n$. Using (10) with the above bound, we get that $w.p. \geq 1 - \frac{1}{n^d}$:

$$\left| \sum_j \mathbb{I}\left\{|\mathbf{x}_j^\top \boldsymbol{\lambda} + \epsilon_j + b_j^*| \geq \tau_k\right\} - \mathbb{I}\left\{|\mathbf{x}_j^\top \widetilde{\boldsymbol{\lambda}} + \epsilon_j + b_j^*| \geq \widetilde{\tau}\right\} \right| \leq 4d. \tag{13}$$

Also let $R = \left\{j, \text{ s.t., } \mathbb{I}\left\{|\mathbf{x}_j^\top \boldsymbol{\lambda} + \epsilon_j + b_j^*| \geq \tau_k\right\} \neq \mathbb{I}\left\{|\mathbf{x}_j^\top \widetilde{\boldsymbol{\lambda}} + \epsilon_j + b_j^*| \geq \widetilde{\tau}\right\}\right\}$. Above bound shows that $|R| \leq 4d$ with high probability.

Now,

$$\begin{aligned}\sum_{j \in R} |\boldsymbol{\lambda}^\top \mathbf{x}_j||b_j^* + \epsilon_j| &\leq \sum_{j \in R} |\boldsymbol{\lambda}^\top \mathbf{x}_j|(|\boldsymbol{\lambda}^\top \mathbf{x}_j| + \widetilde{\tau} + \gamma) \overset{\zeta_1}{\leq} \sum_{j \in R} (\boldsymbol{\lambda}^\top \mathbf{x}_j)^2 + \sqrt{|R|}(\widetilde{\tau} + \gamma)\sqrt{\sum_{j \in R} (\boldsymbol{\lambda}^\top \mathbf{x}_j)^2} \\ &\overset{\zeta_2}{\leq} 20(d \log n/d + \sqrt{dn \log n})\|\boldsymbol{\lambda}\|^2 + 160\sigma(d + \sqrt{dn}) \log n \cdot \|\boldsymbol{\lambda}\|, \end{aligned} \tag{14}$$

where ζ_1 follows from the fact that for all $j \in R$, $|b_j^* + \epsilon_j + \boldsymbol{\lambda}^\top \mathbf{x}_j - \widetilde{\tau}| \leq \gamma$ and by using Cauchy-Schwarz inequality. ζ_2 follows from SSS condition (Theorem 10), as well as bound on τ and γ (given above).

Using (8) and (14), we get (w.p. $\geq 1 - 1/n^d$):

$$\begin{aligned}\boldsymbol{\lambda}^\top X \mathbf{z} \leq 20((k+d) \log n/d + 2\sqrt{dn \log n})\|\boldsymbol{\lambda}\|^2 + 160\sigma(d + \sqrt{dn}) \cdot \log n \cdot \|\boldsymbol{\lambda}\| \\ + \sum_j (\boldsymbol{\lambda}^\top \mathbf{x}_j)\left(\mathbb{I}\left\{|\mathbf{x}_j^\top \widetilde{\boldsymbol{\lambda}} + b_j^* + \epsilon_j| > \widetilde{\tau}\right\}(b_j^* + \epsilon_j) - b_j^*\right). \end{aligned} \tag{15}$$

We now analyze the last term in the above expression:

$$\begin{aligned}\sum_j (\boldsymbol{\lambda}^\top \mathbf{x}_j)\left(\mathbb{I}\left\{|\mathbf{x}_j^\top \widetilde{\boldsymbol{\lambda}} + b_j^* + \epsilon_j| > \widetilde{\tau}\right\}(b_j^* + \epsilon_j) - b_j^*\right) &\overset{\zeta_1}{\leq} \sum_j (\widetilde{\boldsymbol{\lambda}}^\top \mathbf{x}_j)\left(\mathbb{I}\left\{|\mathbf{x}_j^\top \widetilde{\boldsymbol{\lambda}} + b_j^* + \epsilon_j| > \widetilde{\tau}\right\}(b_j^* + \epsilon_j) - b_j^*\right) \\ &\quad + \gamma_{\boldsymbol{\lambda}}\sqrt{d}\sqrt{\log n}\left(\sum_j \left|\left(\mathbb{I}\left\{|\mathbf{x}_j^\top \widetilde{\boldsymbol{\lambda}} + b_j^* + \epsilon_j| > \widetilde{\tau}\right\} - 1\right) b_j^*\right| + |\epsilon_j|\right), \\ &\overset{\zeta_2}{\leq} \sum_j (\widetilde{\boldsymbol{\lambda}}^\top \mathbf{x}_j)\left(\mathbb{I}\left\{|\mathbf{x}_j^\top \widetilde{\boldsymbol{\lambda}} + b_j^* + \epsilon_j| > \widetilde{\tau}\right\}(b_j^* + \epsilon_j) - b_j^*\right) + \gamma_{\boldsymbol{\lambda}}\sqrt{d}\sqrt{\log n} \cdot n \cdot \sqrt{d}\sigma\sqrt{\log n} \\ &\overset{\zeta_3}{\leq} \sum_j (\widetilde{\boldsymbol{\lambda}}^\top \mathbf{x}_j)\left(\mathbb{I}\left\{|\mathbf{x}_j^\top \widetilde{\boldsymbol{\lambda}} + b_j^* + \epsilon_j| > \widetilde{\tau}\right\}(b_j^* + \epsilon_j) - b_j^*\right) + \frac{\sigma}{2000n^2}, \end{aligned} \tag{16}$$

where (ζ_1) follows by using the fact that $\left\|\boldsymbol{\lambda} - \tilde{\boldsymbol{\lambda}}\right\|_2 \leq \gamma_{\boldsymbol{\lambda}}$ by definition, (ζ_2) follows from bounds on ϵ_j, and (ζ_3) follows from the setting of $\gamma_{\boldsymbol{\lambda}} = \frac{\sigma}{2000dn^3 \log n}$. Now, using a union bound over all the $(80dn^3 \log n)^d$ elements of the net over $\boldsymbol{\lambda}$ values and all $2.5n^3\sqrt{\log n}$ elements of the net over τ values on top of the result in Lemma 13, we have *for all* $\widetilde{\boldsymbol{\lambda}} \in \gamma_{\boldsymbol{\lambda}}$-net of B and for all $\widetilde{\tau} \in \gamma_\tau$-net of $[.5\sigma\sqrt{\log n/k}, 30\sigma\sqrt{\log n/k}]$ (w.p. $1 - 1/n^d$):

$$\begin{aligned}\sum_j (\widetilde{\boldsymbol{\lambda}}^\top \mathbf{x}_j)\left(\mathbb{I}\left\{|\mathbf{x}_j^\top \widetilde{\boldsymbol{\lambda}} + b_j^* + \epsilon_j| > \widetilde{\tau}\right\}(b_j^* + \epsilon_j) - b_j^*\right) \leq \left(0.4n\sqrt{\log \alpha}\exp(-\frac{\log \alpha}{33}) + 1.62\frac{n}{\alpha}\log(\alpha)\right)\|\boldsymbol{\lambda}\|_2^2 \\ + 8\sigma\|\boldsymbol{\lambda}\| \log n \cdot \sqrt{100nd \log n}, \end{aligned} \tag{17}$$

where $\alpha = n/k$.

Finally, using (15), (16), and (17), and the fact that $\alpha = n/(2k^*) \geq 1000$ and $d \leq n/1000$, we get with probability at least $\geq 1 - 1/n^{10}$:

$$\boldsymbol{\lambda}^\top X\mathbf{z} \leq .8n\|\boldsymbol{\lambda}\|^2 + 200\sigma\sqrt{n \cdot d} \cdot \log^2 n \cdot \|\boldsymbol{\lambda}\|. \tag{18}$$

We now consider the second term from (7). Let $\mathbf{v}$ be any unit vector such that $\mathbf{v}^\top \boldsymbol{\lambda} = 0$. Note that,

$$\mathbf{v}^T X\mathbf{z} = \widetilde{\mathbf{v}}^\top X\mathbf{z} - (\widetilde{\mathbf{v}} - \mathbf{v})^\top X\mathbf{z}, \tag{19}$$

where $\widetilde{\mathbf{v}}$ is the closest point to $\mathbf{v}$ in an ν-net over S^{d-1} such that each point over the net is orthogonal to $\widetilde{\boldsymbol{\lambda}}$; recall that $\widetilde{\boldsymbol{\lambda}}$ is the closest point to $\boldsymbol{\lambda}$ over $\gamma_{\boldsymbol{\lambda}}$ net over B.

Now,

$$\widetilde{\mathbf{v}}^\top X\mathbf{z} = \sum_j (\widetilde{\mathbf{v}}^\top \mathbf{x}_j)(\mathbb{I}\left\{|\boldsymbol{\lambda}^\top \mathbf{x}_j + b_j^* + \epsilon_j| \geq \tau_k\right\}(b_j^* + \epsilon_j) - b_j^*), \tag{20}$$

$$\begin{aligned} &= \sum_j (\widetilde{\mathbf{v}}^\top \mathbf{x}_j)(\mathbb{I}\left\{|\widetilde{\boldsymbol{\lambda}}^\top \mathbf{x}_j + b_j^* + \epsilon_j| \geq \widetilde{\tau}\right\}(b_j^* + \epsilon_j) - b_j^*) \\ &\qquad + \sum_j (\widetilde{\mathbf{v}}^\top \mathbf{x}_j)(\mathbb{I}\left\{|\boldsymbol{\lambda}^\top \mathbf{x}_j + b_j^* + \epsilon_j| \geq \tau_k\right\} - \mathbb{I}\left\{|\widetilde{\boldsymbol{\lambda}}^\top \mathbf{x}_j + b_j^* + \epsilon_j| \geq \widetilde{\tau}\right\})(b_j^* + \epsilon_j). \end{aligned} \tag{21}$$

Now, using the fact that,

$$\left|\mathbb{I}\left\{|\boldsymbol{\lambda}^\top \mathbf{x}_j + b_j^* + \epsilon_j| \geq \tau_k\right\} - \mathbb{I}\left\{|\widetilde{\boldsymbol{\lambda}}^\top \mathbf{x}_j + b_j^* + \epsilon_j| \geq \widetilde{\tau}\right\}\right| \leq \mathbb{I}\left\{\widetilde{\boldsymbol{\lambda}}^\top \mathbf{x}_j + b_j^* + \epsilon_j - \widetilde{\tau}| \leq \gamma_\tau + \|\mathbf{x}_j\|\gamma_{\boldsymbol{\lambda}}\right\},$$

as well as using (13), we get (w.p. $\geq 1 - 1/n^d$):

$$\begin{aligned} \sum_j (\widetilde{\mathbf{v}}^\top \mathbf{x}_j)(\mathbb{I}\left\{|\boldsymbol{\lambda}^\top \mathbf{x}_j + b_j^* + \epsilon_j| \geq \tau_k\right\} - \mathbb{I}\left\{|\widetilde{\boldsymbol{\lambda}}^\top \mathbf{x}_j + b_j^* + \epsilon_j| \geq \widetilde{\tau}\right\})(b_j^* + \epsilon_j) &\leq \sum_{j \in R} |\widetilde{\mathbf{v}}^\top \mathbf{x}_j|(|\widetilde{\boldsymbol{\lambda}}^\top \mathbf{x}_j| + \widetilde{\tau} + \gamma_\tau + \|\mathbf{x}_j\|\gamma_{\boldsymbol{\lambda}}|), \\ &\overset{\zeta_1}{\leq} 2\|\boldsymbol{\lambda}\| \sum_{j \in R} (\widetilde{\mathbf{v}}^\top \mathbf{x}_j)^2 + 2\frac{1}{\|\boldsymbol{\lambda}\|} \sum_{j \in R} (\widetilde{\boldsymbol{\lambda}}^\top \mathbf{x}_j)^2 + 30\sigma \log n \sum_{j \in R} |\widetilde{\mathbf{v}}^\top \mathbf{x}_j|, \\ &\overset{\zeta_2}{\leq} (d + \sqrt{dn})\sigma + 100\sigma d \log n, \end{aligned} \tag{22}$$

where $R = \left\{j, \text{ s.t., } \sum_j \mathbb{I}\left\{|\mathbf{x}_j^\top \boldsymbol{\lambda} + \epsilon_j + b_j^*| \geq \tau_k\right\} \neq \mathbb{I}\left\{|\mathbf{x}_j^\top \widetilde{\boldsymbol{\lambda}} + \epsilon_j + b_j^*| \geq \widetilde{\tau}\right\}\right\}$. ζ_1 follows from $\widetilde{\tau} \leq 30\sigma\sqrt{\log n}$. ζ_2 follows from Theorem 10 along with bound on $|R| \leq 4d$ obtained using (13).

Now consider the first term of (21). Note that

$$\mathbb{E}\left[\sum_j (\widetilde{\mathbf{v}}^\top \mathbf{x}_j)(\mathbb{I}\left\{|\widetilde{\boldsymbol{\lambda}}^\top \mathbf{x}_j + b_j^* + \epsilon_j| \geq \widetilde{\tau}\right\}(b_j^* + \epsilon_j) - b_j^*)\right] = 0.$$

Moreover, using argument similar to one used in Lemma 13, we get (w.p. $\geq 1 - \exp(-n/10)$):

$$|(\widetilde{\mathbf{v}}^\top \mathbf{x}_j)(\mathbb{I}\left\{|\widetilde{\boldsymbol{\lambda}}^\top \mathbf{x}_j + b_j^* + \epsilon_j| \geq \widetilde{\tau}\right\}(b_j^* + \epsilon_j) - b_j^*)| \leq 8\sigma \log n. \tag{23}$$

Using Hoeffding's inequality with union bound as well (19), (21), and (23), we get *for all* $\boldsymbol{\lambda}$ and *for all* $\mathbf{v} \in S^{d-1}$ s.t. $\mathbf{v} \perp \boldsymbol{\lambda}$ (w.p. $\geq 1 - 1/n^{d-1}$):

$$|\mathbf{v}^T X\mathbf{z}| \leq 100\sigma(d + \sqrt{d \cdot n}) \log^2 n. \tag{24}$$

Lemma now follows by combining (7), (18), and (24).

□

Lemma 13. *Let $\mathbf{x}_j \sim \mathcal{N}(0, I) \in \mathbb{R}^d$ for all $1 \leq j \leq n$ and $\epsilon_j \sim \mathcal{N}(0, \sigma^2)$. Let b^* be k^* sparse and $k = 2k^*$. Let $d < n/1000$ and $k < n/1000$. Let $\boldsymbol{\lambda} \in \mathbb{R}^d$ be a* fixed *vector with $\|\boldsymbol{\lambda}\| \leq \sigma/100$ and $\tau \in [.5\sigma \log(n/k), 2\sigma \log(n/k)]$ be a fixed constant. Then, the following holds (w.p. $\geq 1 - \delta - \exp(-n/10)$):*

$$\sum_j (\boldsymbol{\lambda}^\top \mathbf{x}_j)\left(\mathbb{I}\left\{|\mathbf{x}_j^\top \boldsymbol{\lambda} + b_j^* + \epsilon_j| > \tau\right\}(b_j^* + \epsilon_j) - b_j^*\right) \leq \left(0.4n\sqrt{\log\alpha}\exp(-\frac{\log\alpha}{33}) + 1.62\frac{n}{\alpha}\log(\alpha)\right)\|\boldsymbol{\lambda}\|_2^2 + 8\sigma\|\boldsymbol{\lambda}\|\log n\sqrt{n\log\frac{n}{\delta}}.$$

Proof. Let us define $\mathbf{x}_j^\top \boldsymbol{\lambda} := h_j, a_j = b_j^* + \epsilon_j$. We note that $h_j \sim N(0, \|\boldsymbol{\lambda}\|_2^2)$ and $a_j \sim N(b_j^*, \sigma^2)$. Then, we are interested in:

$$\sum_j (\boldsymbol{\lambda}^\top \mathbf{x}_j)\left(\mathbb{I}\left\{|\mathbf{x}_j^\top \boldsymbol{\lambda} + b_j^* + \epsilon_j| > \tau\right\}(b_j^* + \epsilon_j) - b_j^*\right) = \sum_j r_j, \tag{25}$$

where

$$r_j = \mathbb{I}\left\{|h_j + a_j| > \tau\right\} h_j a_j - h_j b_j^*.$$

Note that $\mathbb{E}\left[\sum_j h_j b_j^*\right] = 0$ since b_j^* is independent of h_j. Hence,

$$\mathbb{E}\left[\sum_j r_j\right] = \mathbb{E}\left[\sum_j \mathbb{I}\left\{|h_j + a_j| > \tau\right\} h_j a_j\right]. \tag{26}$$

Now, using distribution of h_j and a_j, we have:

$$\begin{aligned}\mathbb{E}\left[\mathbb{I}\left\{|h_j + a_j| > \tau\right\} h_j a_j | a_j\right] &= \frac{1}{\sqrt{2\pi}\|\boldsymbol{\lambda}\|_2}\int_{-\infty}^{-\tau - a_j} a_j h_j \exp\left(-\frac{h_j^2}{2\|\boldsymbol{\lambda}\|_2^2}\right)dh_j + \frac{1}{\sqrt{2\pi}\|\boldsymbol{\lambda}\|_2}\int_{\tau - a_j}^{\infty} a_j h_j \exp\left(-\frac{h_j^2}{2\|\boldsymbol{\lambda}\|_2^2}\right)dh_j \\ &= \frac{\|\boldsymbol{\lambda}\|_2}{\sqrt{2\pi}} a_j\left(\exp\left(-\frac{(\tau - a_j)^2}{2\|\boldsymbol{\lambda}\|_2^2}\right) - \exp\left(-\frac{(\tau + a_j)^2}{2\|\boldsymbol{\lambda}\|_2^2}\right)\right).\end{aligned}$$

Since h_j and a_j are independent $\mathbb{E}\left[\mathbb{I}\left\{|h_j + a_j| > \tau\right\} h_j a_j\right] = \mathbb{E}_{a_j}\left[\mathbb{E}_{h_j}\left[\mathbb{I}\left\{|h_j + a_j| > \tau\right\} h_j a_j | a_j\right]\right]$. Therefore,

$$\begin{aligned}\mathbb{E}\left[\mathbb{I}\left\{|h_j + a_j| > \tau\right\} h_j a_j\right] &= \frac{\|\boldsymbol{\lambda}\|_2^2}{\sqrt{2\pi}\|\boldsymbol{\lambda}\|_2}\frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty} a_j \exp\left(-\frac{(\tau - a_j)^2}{2\|\boldsymbol{\lambda}\|_2^2}\right)\exp\left(-\frac{(a_j - b_j^*)^2}{2\sigma^2}\right)da_j \\ &\quad - \frac{\|\boldsymbol{\lambda}\|_2^2}{\sqrt{2\pi}\|\boldsymbol{\lambda}\|_2}\frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty} a_j \exp\left(-\frac{(\tau + a_j)^2}{2\|\boldsymbol{\lambda}\|_2^2}\right)\exp\left(-\frac{(a_j - b_j^*)^2}{2\sigma^2}\right)da_j \\ &= \frac{1}{\sqrt{2\pi}}\frac{\|\boldsymbol{\lambda}\|_2^2}{\sqrt{\|\boldsymbol{\lambda}\|_2^2 + \sigma^2}}\exp\left(-\frac{1}{2}\left(\frac{\tau^2}{\|\boldsymbol{\lambda}\|_2^2} + \frac{b_j^{*2}}{\sigma^2}\right)\right)\frac{\frac{\tau}{\|\boldsymbol{\lambda}\|_2^2} + \frac{b_j^*}{\sigma^2}}{\frac{1}{\|\boldsymbol{\lambda}\|_2^2} + \frac{1}{\sigma^2}}\exp\left(\frac{1}{2}\frac{\left(\frac{\tau}{\|\boldsymbol{\lambda}\|_2^2} + \frac{b_j^*}{\sigma^2}\right)^2}{\frac{1}{\|\boldsymbol{\lambda}\|_2^2} + \frac{1}{\sigma^2}}\right) \\ &\quad + \frac{1}{\sqrt{2\pi}}\frac{\|\boldsymbol{\lambda}\|_2^2}{\sqrt{\|\boldsymbol{\lambda}\|_2^2 + \sigma^2}}\exp\left(-\frac{1}{2}\left(\frac{\tau^2}{\|\boldsymbol{\lambda}\|_2^2} + \frac{b_j^{*2}}{\sigma^2}\right)\right)\frac{\frac{\tau}{\|\boldsymbol{\lambda}\|_2^2} - \frac{b_j^*}{\sigma^2}}{\frac{1}{\|\boldsymbol{\lambda}\|_2^2} + \frac{1}{\sigma^2}}\exp\left(\frac{1}{2}\frac{\left(\frac{-\tau}{\|\boldsymbol{\lambda}\|_2^2} + \frac{b_j^*}{\sigma^2}\right)^2}{\frac{1}{\|\boldsymbol{\lambda}\|_2^2} + \frac{1}{\sigma^2}}\right) \\ &\leq 2\frac{\|\boldsymbol{\lambda}\|_2^2}{\sqrt{2\pi}}\frac{\tau + \frac{|b_j^*|\|\boldsymbol{\lambda}\|_2^2}{\sigma^2}}{\sigma}\exp\left(\frac{-1}{2}\frac{(|b_j^*| - \tau)^2}{\sigma^2 + \|\boldsymbol{\lambda}\|_2^2}\right),\end{aligned} \tag{27}$$

where the above inequalities follow from straightforward calculations.

Note that $\|\boldsymbol{\lambda}\|_2^2 \leq \frac{\sigma^2}{10^4}$. Let $\alpha := \frac{n}{k}$. Now, consider the following three cases for a fixed $\tau \in (\frac{1}{2}\sqrt{\log(\alpha)}, 2\sqrt{\log(\alpha)})$:

- $|b_j^*| \leq \tau/2$

  $$\mathbb{E}\left[r_j\right] \leq \|\boldsymbol{\lambda}\|_2^2\frac{2}{\sqrt{2\pi}}\frac{\tau + \frac{\tau}{2\cdot 10^4}}{\sigma}\exp\left(\frac{-1}{2.02}\frac{(|b_j^*| - \tau)^2}{\sigma^2}\right) \overset{\zeta_1}{\leq} \|\boldsymbol{\lambda}\|_2^2\, 0.8\frac{\tau}{\sigma}\exp\left(\frac{-1}{8.08}\frac{\tau^2}{\sigma^2}\right)$$

  where ζ_1 follows from using the fact that $|b_j^*| \leq \tau/2$.

- $\tau/2 \leq |b_j^*| \leq 2\tau$

$$\mathbb{E}\left[r_j\right] \leq \|\boldsymbol{\lambda}\|_2^2 \frac{2}{\sqrt{2\pi}} \frac{\tau + \frac{2\tau}{10^4}}{\sigma} \exp\left(\frac{-1}{2.02}\frac{(|b_j^*| - \tau)^2}{\sigma^2}\right) \overset{\zeta_1}{\leq} \|\boldsymbol{\lambda}\|_2^2 0.8\frac{\tau}{\sigma}$$

where ζ_1 follows from using $\exp\left(\frac{-1}{2.02}\frac{(|b_j^*|-\tau)^2}{\sigma^2}\right) \leq 1$.

- $|b_j^*| \geq 2\tau$

$$\mathbb{E}\left[r_j\right] \leq \|\boldsymbol{\lambda}\|_2^2 \frac{2}{\sqrt{2\pi}} \frac{0.5001|b_j^*|}{\sigma} \exp\left(\frac{-1}{2.02}\frac{(|b_j^*| - \tau)^2}{\sigma^2}\right) \overset{\zeta_1}{\leq} \|\boldsymbol{\lambda}\|_2^2 0.4\frac{|b_j^*|}{\sigma} \exp\left(\frac{-1}{8.08}\frac{b_j^{*2}}{\sigma^2}\right)$$

where ζ_1 follows from using $|b_j^*| \geq 2\tau$.

Recall that $\mathbf{b}^*$ has at most k^* non-zeros, hence at most k^* elements can belong to case 2 and 3 described above. Combining the above given cases, we have:

$$\begin{aligned}
\mathbb{E}\left[\sum_j r_j\right] &\leq \left((n-k^*)\max_{\beta \geq \frac{2\tau}{\sigma}}\left[0.4\beta\exp\left(\frac{-\beta^2}{8.08}\right), 0.8\frac{\tau}{\sigma}\exp\left(\frac{-1}{8.08}\frac{\tau^2}{\sigma^2}\right)\right] + 0.8\frac{\tau}{\sigma}\cdot k^*\right)\|\boldsymbol{\lambda}\|_2^2 \\
&\overset{\zeta_1}{\leq} \left((n-k^*)\max_{\beta \geq \frac{2\tau}{\sigma}}\left[0.4\beta\exp\left(\frac{-\beta^2}{8.08}\right), 0.8\frac{\tau}{\sigma}\exp\left(\frac{-1}{8.08}\frac{\tau^2}{\sigma^2}\right)\right] + 1.62\frac{n}{\alpha}\log(\alpha)\right)\|\boldsymbol{\lambda}\|_2^2 \\
&\overset{\zeta_2}{\leq} \left(0.4n\sqrt{\log\alpha}\exp(-\frac{\log\alpha}{33}) + 1.62\frac{n}{\alpha}\log(\alpha)\right)\|\boldsymbol{\lambda}\|_2^2, \qquad (28)
\end{aligned}$$

where ζ_1 follows from the fact that $\tau \leq 30\sigma\sqrt{\log(\alpha)}$. ζ_2 follows from maximizing $x\exp\left(-\frac{x^2}{8.08}\right)$ in the interval $(0.5\sqrt{\log(\alpha)}, 2\sqrt{\log(\alpha)})$ for τ and $(\sqrt{\log(\alpha)}, \infty)$ for β.

Above given equation bounds the expected values of $\sum_j r_j$. We now bound the deviation of $\sum_j r_j$ from its expected value. For this, we first consider $r_j = \mathbb{I}\left\{|h_j + a_j| > \tau\right\} h_j a_j - h_j b_j^*$. Here, we consider two cases:

1. $|h_j + a_j| > \tau$: in this case, $r_j = h_j\epsilon_j$. Moreover, w.p. $\geq 1 - \exp(-n/10)$, $|h_j| \leq 2\|\boldsymbol{\lambda}\|\sqrt{\log n}$, and $|\epsilon_j| \leq 2\sigma\sqrt{\log n}$. Hence, $|r_j| \leq 4\|\boldsymbol{\lambda}\|\sigma\log n$.
2. $|h_j + a_j| \leq \tau$: in this case, $|b_j^*| \leq \tau + |h_j| + |\epsilon_j| \leq \tau + 2(\|\boldsymbol{\lambda}\| + \sigma)\sqrt{\log n}$. Moreover, $r_j = -h_j b_j^*$, i.e., $|r_j| \leq 8\sigma\|\boldsymbol{\lambda}\|\log n$.

Hence, using Hoeffding's bound, we get (w.p. $\geq 1 - \delta - \exp(-n/10)$):

$$\left|\sum_j r_j - \mathbb{E}\left[r_j\right]\right| \leq 8\sigma\|\boldsymbol{\lambda}\|\log n\sqrt{n\log\left(\frac{2}{\delta}\right)}.$$

Lemma now follows by combining the above observation with (28). □