Outlier-Robust Phase Retrieval in Nearly-Linear Time

Anonymous Author(s)
Affiliation
Address
email

Abstract

Phase retrieval is a fundamental problem in signal processing, where the goal is to recover a (complex-valued) signal from phaseless intensity measurements. In this paper, we propose and study the (real-valued) outlier-robust phase retrieval problem. Specifically, we seek to recover a vector $x \in \mathbb{R}^d$ from n intensity measurements $y_i = (a_i^\top x)^2$, where a small fraction of the (a_i, y_i) pairs are adversarially corrupted. Our main result is a near-sample-optimal and nearly-linear-time algorithm that provably recovers the ground-truth vector. Our algorithm first solves a lightweight convex program to find an initial point close to the ground truth, and then runs a robust version of gradient descent to achieve exact recovery. Our approach is conceptually simple and provides a framework for developing robust algorithms for other non-convex optimization problems.

1 Introduction

Phase retrieval is a fundamental problem in signal processing with applications in various fields, including electron microscopy [35], crystallography [36, 39], astronomy [13], and optical imaging [40]. In these applications, one often has access to only the magnitudes of the Fourier transforms of a complex signal. This is because measuring magnitude (e.g., by aggregating energy over time) is much easier than measuring phase (which requires detecting rapid changes). We refer the reader to the survey articles [40, 29] for more details about the theory and applications of phase retrieval.

In this paper, we focus on the (real-valued) generalized phase retrieval problem, where we are given intensity measurements of an arbitrary linear operator.

Definition 1.1 (Phase Retrieval). *Let $x \in \mathbb{R}^d$ be the ground-truth vector. Let $a_1, \ldots, a_n \in \mathbb{R}^d$ be n sampling vectors and let $y_i = \langle a_i, x \rangle^2 \in \mathbb{R}$ be the corresponding intensity measurements. Given $(a_i, y_i)_{i=1}^n$ as input, the task is to recover x.*

Note that it is impossible to distinguish between x and $-x$, thus recovering either is sufficient. This problem has been extensively studied by the machine learning community. Under certain assumptions on the distribution of the sampling vectors, such as when the a_i's are independent and Gaussian distributed, $\Theta(d)$ input pairs (a_i, y_i) are necessary and sufficient for exact recovery [8]. Additionally, it has been shown that the problem can be solved in linear time with respect to the size of the input [8]. This was first achieved using semidefinite programming (SDP) relaxations [6]. In practice, the problem is often solved by applying first-order optimization methods (e.g., gradient descent) to a suitable objective function such as

$$\min_{z \in \mathbb{R}^d} \quad f(z) = \sum_{i=1}^{n} (y_i - \langle a_i, z \rangle^2)^2 \, . \tag{1}$$

Even though many natural formulations of the phase retrieval objective are nonconvex, including the one in (1), prior work has shown that, depending on the input distribution, they may not have spurious local optima and thus can be solved using first-order optimization methods [37, 4, 43].

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

However, these analyses of the objective landscape rely on strong assumptions, such as the sampling vectors a_i being i.i.d. Gaussian. Our work is motivated by the following questions: Can we relax the assumptions used to prove landscape results for tractable nonconvex problems? In the context of phase retrieval, can we recover the ground-truth vector x when a small subset of the (a_i, y_i) pairs are adversarially corrupted?

Our focus on this adversarial setting, where an ϵ-fraction of the input data is corrupted, is inspired by recent advances in high-dimensional robust statistics. An example problem in robust statistics is to estimate the mean of a d dimensional spherical Gaussian when ϵ-fraction of the samples are arbitrarily corrupted. The goal of high-dimensional statistics is often to design efficient algorithms that can achieve dimension-independent error guarantees. Early work in robust statistics [45, 26, 28] provided sample-efficient estimators for various tasks, but with runtimes exponential in the dimension. A recent line of work initiated by [16] and [31] has developed computationally efficient robust algorithms for many fundamental statistical and learning tasks. Significant progress has been made in the algorithmic aspects of robust high-dimensional statistics (see, e.g., [15]).

We now formally define the ϵ-corruption model we study in this paper. For clarity, we define it directly in the context of phase retrieval.

Definition 1.2 (ϵ-Corrupted Set of Samples)**.** *Let $x \in \mathbb{R}^d$ be the ground-truth vector. Let $\epsilon > 0$. First, the algorithm specifies the sample complexity n. Then, n sampling vectors $(a_1, \ldots, a_n)$ are drawn from some known distribution D and the corresponding intensity measurements $y_i = \langle a_i, x \rangle^2$ are calculated. The adversary is allowed to replace ϵn pairs (a_i, y_i) with arbitrary data. We call a set of (a_i, y_i) pairs ϵ-corrupted if it is generated by this process.*

In this paper, we say *samples* to refer to (a_i, y_i) pairs. Note that we allow corruption in both the sampling vectors $a_i \in \mathbb{R}^d$ and the intensity measurements $y_i \in \mathbb{R}$, as long as the fraction of corruption is at most ϵ. We now formally define outlier-robust phase retrieval, the main problem we study in this paper.

Problem 1.3 (Outlier-Robust Phase Retrieval)**.** *Let $x \in \mathbb{R}^d$ be the ground-truth with $\|x\|_2 = 1$. Let $\epsilon > 0$. Let $(a_1, \ldots, a_n)$ be n sampling vectors drawn i.i.d. from $\mathcal{N}(0, I) \in \mathbb{R}^d$, and let $y_i = \langle a_i, x \rangle^2$ be the corresponding intensity measurements. An adversary arbitrarily corrupts an ϵ-fraction of these (a_i, y_i) pairs, and gives the ϵ-corrupted set of samples as input to the algorithm. The task is to find a vector $z \in \mathbb{R}^d$ such that $\min\{\|z - x\|_2, \|z + x\|_2\} \le \Delta$ for some precision parameter $\Delta > 0$.*

A simpler adversary setting where corruption is restricted to the intensity measurements y_i has been studied previously [27, 51, 19]. In our work, we study a more comprehensive setting where we also allow corruption in the measurement vectors. This models realistic scenarios where the measurement process is affected by hardware noise, miscalibration, or adversarial tampering, leading to perturbations in the sampling vectors a_i as well as in y_i. Concurrent work [3] studies this same problem. However, their algorithm uses a black-box subroutine for robust covariance estimation and thus requires at least $\Omega(d^2)$ time, making it impractical in high dimensions.

We are interested in designing a scalable and provably robust algorithm for Problem 1.3. We would like to resolve the following question:

> *Can we design a provably robust algorithm for the outlier-robust phase retrieval problem (Problem 1.3) that has near-optimal sample complexity and runs in nearly-linear time?*

1.1 Our Results

We answer the question outlined in the previous subsection affirmatively.

Theorem 1.4 (Main)**.** *Consider the outlier-robust phase retrieval problem as defined in Problem 1.3, where $x \in \mathbb{R}^d$ is the ground-truth vector. Let $0 < \epsilon \le \epsilon_0$ for sufficiently small universal constant ϵ_0. Let $\Delta > 0$ be the desired precision. Given an ϵ-corrupted set of $n = \widetilde{\Omega}(d \log^2(1/\Delta))$ samples, we can compute $z \in \mathbb{R}^d$ in time $\widetilde{O}(nd)$ such that $\min(\|z - x\|_2, \|z + x\|_2) \le \Delta$ with probability at least 0.95.* [1]

[1] Throughout the paper, we use $\widetilde{O}(\cdot)$ and $\widetilde{\Omega}(\cdot)$ to hide logarithmic factors in their parameters.

A few remarks are in order regarding Theorem 1.4. First, even without corruption, phase retrieval requires $\Omega(d)$ samples [8]. When $\Delta \geq \frac{1}{\mathrm{poly}(d)}$, Theorem 1.4 requires $\widetilde{O}(d)$ samples, runs in nearly linear time (since the input size is $O(nd)$), and achieves exact recovery. Therefore, our algorithm simultaneously achieves the best possible error, sample complexity, and runtime (up to logarithmic factors).

Second, the success probability in Theorem 1.4 can be boosted to $1 - \tau$ for any $\tau > 0$ by incurring an additional $O(\log(1/\tau))$ factor in the sample complexity and runtime. This can be achieved, for example, by partitioning the input, repeating the algorithm, letting candidate solutions vote for those within distance 2Δ, and finally selecting the solution with the most votes.

Lastly, ϵ_0 in Theorem 1.4 is an absolute constant that is independent of n or d, and our algorithm works for any corruption level $0 \leq \epsilon \leq \epsilon_0$. An important observation is that the optimal sample complexity is $\Theta(d)$, which is independent of ϵ. This follows from the fact that exact recovery is possible as long as the clean samples provide enough constraints to fully determine x, which has $d - 1$ degrees of freedom. [2] This is in contrast to problems in robust high-dimensional statistics, such as robust mean estimation, where exact recovery is impossible with a finite number of samples (even without corruption).

Since our algorithm guarantees exact recovery (to arbitrary precision Δ) for any corruption level ϵ as long as $\epsilon < \epsilon_0$, any input with corruption level $\epsilon < \epsilon_0$ can be treated as if it were corrupted at a fixed level ϵ_0. This explains why neither the sample complexity nor the runtime of our algorithm depends on ϵ. For simplicity, we refer to the corruption level ϵ as a sufficiently small universal constant throughout the remainder of the paper unless otherwise noted.

1.2 Our Approach and Techniques

When there are infinitely many samples and no corruption, the objective function $f(z)$ simplifies to

$$f(z) = \mathop{\mathbb{E}}_{a \sim \mathcal{N}(0, I_d)} \left[\left(\langle a_i, z \rangle^2 - y_i \right)^2 \right] = 3 \left\| x \right\|_2^4 + 3 \left\| z \right\|_2^4 - 2 \left\| x \right\|_2^2 \left\| z \right\|_2^2 - 4 \left\langle x, z \right\rangle^2 .$$

Despite being nonconvex, it is known that f has no spurious local optima [37, 4, 43].

Our approach follows a two-step structure used in many local convergence results for nonconvex problems (e.g., Candès et al. [4]), where the goal is to first initialize into a region free of saddle points and then perform gradient descent. Both steps are vulnerable to adversarial attacks and we develop provably robust and nearly-linear time algorithms for both steps. In the first step, we use spectral techniques to obtain an initial guess that is sufficiently close to the ground truth. In the second step, we run a robust gradient descent algorithm to refine this guess and converge to the final solution.

Step 1: Robust Spectral Initialization. Consider the following matrix $Y = \frac{1}{n} \sum_{i=1}^{n} y_i a_i a_i^\top$ where $y_i = \langle a_i, x \rangle^2$. When there is no corruption and the a_i's are drawn i.i.d. from $\mathcal{N}(0, I)$, we have $\mathbb{E}[Y] = I + 2xx^\top$. Hence, when there are enough samples and no corruption, we can obtain a good estimate of the ground truth x (or $-x$) by computing the largest eigenvector of Y. However, the corrupted (a_i, y_i) pairs can arbitrarily change the largest eigenvector of Y. One natural approach, which was explored in concurrent work [3], is to apply known robust covariance estimation algorithms [11, 1] to estimate Y. While this can recover the top eigenvector, the runtime is at least $\Omega(d^2)$, which is too slow for our goal of designing a nearly linear time algorithm.

One of the main technical insights of our work is that it is not necessary to robustly estimate the entire matrix Y, we only need to recover its few largest eigenspaces. We will assign a weight w_i to each sample such that the weighted intensity-based matrix $Y_w = \sum_{i=1}^{n} w_i y_i a_i a_i^\top$ is close to $I + 2xx^\top$. We show that this can be done in nearly-linear time, which is highly non-trivial: the ground-truth vector x is unknown, and explicitly computing Y_w via fast matrix multiplication takes $\Omega(d^2)$ time.

A key observation is that the corrupted samples can only add directions to Y_w but cannot remove directions, because each individual term $y_i a_i a_i^\top$ is a PSD matrix. (We assume w.l.o.g. that $y_i \geq 0$ for

[2] To build intuition why exact recovery is possible, consider the univariate case with an arbitrary positive ground-truth $x \in \mathbb{R}$. In this case, one can compute the multiset $\{\sqrt{y_i / a_i^2}\}$, and as long as $\epsilon < 1/2$, its most frequent element will be the correct x with probability 1.

all i, because any input with $y_i < 0$ must be corrupted.) Consequently, if we can compute a weight vector w that minimizes the sum of the top two eigenvalues of Y_w (which is a convex optimization problem), we can recover a matrix that is close to the unknown unbiased expectation $I + 2xx^\top$. We show that this optimization problem can be solved in $\widetilde{O}(nd)$ time by leveraring algorithmic techniques developed for list-decodable mean estimation [12].

Step 2: Robust Gradient Descent. Starting with the initial guess z given by the robust spectral initialization, we want to run gradient descent to recover the ground truth x. Without corruption, if the initialization is close enough to x, each iteration will bring z closer to x by a constant factor. Intuitively, approximating the gradient at a specific point amounts to a robust mean estimation problem (for the underlying distribution of the gradients). When the input data is ϵ-corrupted, the gradients of the n samples can be viewed as an ϵ-corrupted set of vectors.

We can approximate the true gradient by running robust mean estimation on this ϵ-corrupted set of n gradients. To show convergence, the estimation of the gradient needs to be more accurate as we get closer to the optimal solution, and we show that this is possible because the variance of the gradient on clean samples decreases as the solution gets closer to the optimal solution.

1.3 Related and Prior Works

Phase Retrieval. Phase retrieval arises in various fields of science and engineering [13, 36]. Early research introduced error-reduction algorithms [25, 20, 21]. Convex and nonconvex optimization with various objective functions were later proposed and achieved exact recovery [47, 4–6, 48, 49, 41].

Outlier-Robust Phase Retrieval. Robust phase retrieval has been explored in the literature [51, 30, 8, 7, 34]. A simpler setting where corruption is restricted to the intensity measurements y_i has been studied previously [27, 51, 19]. In Appendix C, we show that methods developed for this setting do not work in ours. Concurrent work [3] is the only one that studies the general corruption model that we consider, allowing corruption in both the sampling vectors a_i and the intensity measurements y_i. The algorithm in [3] achieves near-optimal sample complexity, but relies on robust covariance estimation in a black-box manner, resulting in a slower runtime compared to ours. We emphasize that algorithm runs in nearly-linear time and achieves near-optimal sample complexity, which demonstrates that allowing corruption in both a_i and y_i incurs almost no penalty (asymptotically) in terms of statistical or computational complexity.

Nonconvex Optimization. Besides phase retrieval, it is known that all local optima are globally optimal for natural nonconvex formulations of various learning problems, such as matrix completion [24], matrix sensing [2], dictionary learning [42], and tensor decomposition [23] (we refer the interested reader to Chapter 7 of the book by [50]). A recent line of work explored the robustness of such landscape results: [33] studied matrix sensing in the ϵ-corruption model, [9] and [22] studied semi-random matrix completion and matrix sensing.

High-Dimensional Robust Statistics. Recent works developed nearly-linear time algorithms for robust mean estimation [10, 18, 32]. The robust gradient descent algorithm we use is closely related to algorithms proposed in previous works for finding *first-order* stationary points in robust stochastic optimization [38, 17].

2 Preliminary and Background

Notation. Let $[n] = \{1, 2, \dots, n\}$. For a vector x, we denote its i^{th} coordinate by x_i. We use $\|x\|_1$, $\|x\|_2$, and $\|x\|_\infty$ to denote the ℓ_1, ℓ_2, and ℓ_∞ norm of x, respectively. For two vectors x and y, we use $\langle x, y\rangle = x^\top y$ to denote their inner product.

We write I for the identity matrix. For a matrix A, we use $\|A\|_2$ to denote its spectral norm. A symmetric matrix A is positive semidefinite (PSD) if $x^\top Ax \geq 0$ for all vectors x. For two symmetric matrices A and B, we write $A \preceq B$ if $B - A$ is PSD. We write $\lambda_k(A)$ as the k^{th} largest eigenvalue of A, and $\overline{\lambda}_k(A)$ as the sum of the k largest eigenvalues of A. The Ky Fan k-norm of a matrix A is the sum of its k largest singular values, which is equal to $\overline{\lambda}_k(A)$ when A is PSD.

Ky Fan Norm Packing SDP. In our robust spectral initialization step, we solve a Ky Fan norm packing semidefinite program (SDP) of the following form:

$$\max_{w \in \mathbb{R}^n_{\geq 0}} \quad \|w\|_1 \quad \text{subject to} \quad \sum_{i=1}^{n} w_i A_i \preceq I, \quad \overline{\lambda}_k \left(\sum_{i=1}^{n} w_i B_i \right) \leq k \tag{**}$$

We use the nearly-linear time Ky Fan norm SDP solver from [12].

Lemma 2.1 (Ky Fan Norm SDP Solver [12])**.** *Given an SDP* (**) *with positive semidefinite matrices* $A_i \in \mathbb{R}^{d_1 \times d_1}$ *and* $B_i \in \mathbb{R}^{d_2 \times d_2}$ *with* $A_i = C_i C_i^\top$ *and* $B_i = D_i D_i^\top$ *for all* $i \in [n]$*, integer* $k > 0$*, error tolerance* $\epsilon_1 \geq 1/n^2$*, and failure probability* $\tau > 0$*, one can in time* $\widetilde{O}((t_C + t_D + d_1 + d_2)\,\mathrm{poly}(1/\epsilon_1, \log(1/\tau)))$ *output* $w' \in \mathbb{R}^n_{\geq 0}$ *such that* $\|w'\|_1 \geq (1 - \epsilon_1)\mathsf{OPT}$ *with probability at least* $1 - \tau$*. Here* OPT *is the optimal value of* (**)*,* t_{C_i} *and* t_{D_i} *are the time taken to perform a matrix-vector product with* C_i *and* D_i *respectively, and* $t_C = \sum_{i=1}^{n} t_{C_i}$ *and* $t_D = \sum_{i=1}^{n} t_{D_i}$*.*

Top Eigenvector Computation. We use the power method to compute an approximate top eigenvector. We refer to the analysis of the power method for PSD matrices by Trevisan [44].

Lemma 2.2 (Top Eigenvector via Power Method [44])**.** *Let* $A \in \mathbb{R}^{d \times d}$ *be a PSD matrix. Let* λ_1 *be the largest eigenvalue of* A*. For any* $\epsilon_2 > 0$*, one can compute a unit vector* $x \in \mathbb{R}^d$ *in time* $O((t_A + d)\log(d)/\epsilon_2)$ *such that* $x^\top A x \geq (1 - \epsilon_2)\lambda_1$ *with probability at least* 0.99*, where* t_A *is the time taken to perform a matrix-vector multiplication with* A*.*

Robust Mean Estimation. In the robust gradient descent step, we use nearly-linear time robust mean estimation algorithms for bounded-covariance distributions [10, 14, 18] to approximate the true gradient.

Lemma 2.3 (Robust Mean Estimation [18])**.** *Let* D *be a distribution on* $\mathbb{R}^d$ *with unknown mean* μ *and unknown covariance matrix* Σ *where* $\Sigma \preceq \sigma^2 I$*. Let* $\epsilon_3 > 0$ *be a sufficiently small universal constant. Let* $0 < \epsilon \leq \epsilon_3$ *and* $\tau > 0$*. Given an* ϵ*-corrupted set of* n *samples drawn from* D*, one can output a vector* $\widehat{\mu} \in \mathbb{R}^d$ *in time* $\widetilde{O}(nd\log(1/\tau))$ *such that, with probability at least* $1 - \tau - \exp(-n\epsilon)$*, we have* $\|\widehat{\mu} - \mu\|_2 = O\left(\sqrt{\epsilon} + \sqrt{\frac{d}{n\tau}} + \sqrt{\frac{d(\log d + \log(1/\tau))}{n}}\right)\sigma$*.*

3 Outlier-Robust Phase Retrieval

In this section, we present key technical lemmas for the two stages of our algorithm: robust spectral initialization (Lemma 3.1) and robust gradient descent (Lemma 3.2). We then use these lemmas to prove our main result (Theorem 1.4).

Lemma 3.1 shows that we can compute an initial guess close to the ground truth.

Lemma 3.1 (Robust Spectral Initialization)**.** *Consider the setting of Problem 1.3, where* $x \in \mathbb{R}^d$ *is the ground-truth. Let* ϵ *be a sufficiently small universal constant. Given an* ϵ*-corrupted set of* $n = \widetilde{\Omega}(d)$ *samples, we can compute* $z_1 \in \mathbb{R}^d$ *in time* $\widetilde{O}(nd)$ *such that* $\min\{\|z_1 - x\|_2, \|z_1 + x\|_2\} \leq \frac{1}{8}$ *with probability at least* 0.97*.*

Lemma 3.2 shows that after the initialization, a robust gradient descent algorithm can recover the ground-truth vector to arbitrary precision.

Lemma 3.2 (Robust Gradient Descent)**.** *Consider the setting of Problem 1.3, where* $x \in \mathbb{R}^d$ *is the ground-truth vector. Let* $\Delta > 0$ *be the desired precision. Let* ϵ *be a sufficiently small universal constant. Given an* ϵ*-corrupted set of* $n = \widetilde{\Omega}(d\log^2(1/\Delta))$ *samples and an initial guess* z_1 *with* $\|z_1 - x\|_2 \leq \frac{1}{8}$*, we can compute a vector* $z \in \mathbb{R}^d$ *in time* $\widetilde{O}(nd)$ *such that* $\|z - x\|_2 \leq \Delta$ *with probability at least* 0.98*.*

Lemma 3.1 is proved in Section 4. Lemma 3.2 is proved in Section 5. We first use these lemmas to prove Theorem 1.4.

Theorem 1.4. Let ϵ_0 be the minimum of the two universal constants of Lemma 3.1 and Lemma 3.2. Let $0 \leq \epsilon \leq \epsilon_0$. We assume that we have access to two separate ϵ-corrupted sets of samples, and use one set for Lemma 3.1 and the other for Lemma 3.2. Formally, one could randomly partition the input samples into two sets and apply Chernoff bounds to show that both sets are (2ϵ)-corrupted with high probability.

By Lemma 3.1, we can compute a vector $z_1 \in \mathbb{R}^d$ such that $\min\{\|z_1 - x\|_2, \|z_1 + x\|_2\} \le \frac{1}{8}$ for the ground-truth vector x. Given z_1, by Lemma 3.2, we can output a vector $z \in \mathbb{R}^d$ such that $\min\{\|z - x\|_2, \|z + x\|_2\} \le \Delta$ for the desired precision parameter $\Delta > 0$.

Let $n_1 = \widetilde{\Omega}(d)$ and $n_2 = \widetilde{\Omega}(d \log^2(1/\Delta))$ denote the number of samples used in Lemma 3.1 and Lemma 3.2, respectively. The overall sample complexity is therefore $n = n_1 + n_2 = \widetilde{\Omega}(d \log^2(1/\Delta))$. The overall runtime is $\widetilde{O}(n_1 d) + \widetilde{O}(n_2 d) = \widetilde{O}(nd)$. The overall success probability is at least 0.95 by a union bound over Lemma 3.1 and Lemma 3.2. □

4 Robust Spectral Initialization

In this section, we prove Lemma 3.1: given an ϵ-corrupted set of samples $\{(a_i, y_i)\}_{i \in [n]}$, we can compute an initial guess $z_1 \in \mathbb{R}^d$ that is close to the ground truth x or $-x$.

Consider the matrix $Y = \frac{1}{n} \sum_{i=1}^n y_i a_i a_i^\top$. When there is no corruption, where $a_i \sim \mathcal{N}(0, I)$ and $y_i = \langle a_i, x \rangle^2$, we have $\mathbb{E}[Y] = I + 2xx^\top$. However, the corrupted (a_i, y_i)'s can change Y arbitrarily. To address this, we propose a nearly-linear time initialization step (Algorithm 1) that computes a nonnegative weight vector $w \in \mathbb{R}^n$ such that the weighted sum $Y_w = \sum_{i=1}^n w_i y_i a_i a_i^\top$ is close to $I + 2xx^\top$. Consequently, we can show that the largest eigenvector of Y_w is close to $\pm x$.

Let $G \subset [n]$ be the set of indices of the remaining good samples. An ideal approach would be to assign weight $\frac{1}{|G|} = \frac{1}{(1-\epsilon)n}$ to every sample in G, and weight 0 to the corrupted samples. Formally, we consider weight vectors w in the set $\Delta_{n,\epsilon} = \left\{ w \in \mathbb{R}^n_{\ge 0} : \|w\|_1 = 1 \text{ and } \|w\|_\infty \le \frac{1}{(1-\epsilon)n} \right\}$. Algorithm 1 computes a near-optimal solution $\widehat{w}$ to the following optimization problem:

$$\min_{w \in \Delta_{n,\epsilon}} \lambda_1(Y_w) + \lambda_2(Y_w) ,$$

and then returns the largest eigenvector of $Y_{\widehat{w}}$.

Algorithm 1: Robust Spectral Initialization

```
Input: $\epsilon$-corrupted set of $n$ samples $\{(a_i, y_i)\}_{i \in [n]}$.
Output: An initial guess $z_1 \in \mathbb{R}^d$ of the ground-truth $x$ s.t. $\min\{\|z_1 - x\|_2, \|z_1 + x\|_2\} \le \frac{1}{8}$.
$\widehat{w} \leftarrow$ a near-feasible, near-optimal solution to: $\min_{w \in \Delta_{n,\epsilon}} [\lambda_1(Y_w) + \lambda_2(Y_w)]$ using Lemma A.2,
  where $Y_w = \sum_{i=1}^n w_i y_i a_i a_i^\top$;
$z_1 \leftarrow$ an approximate top eigenvector of $Y_{\widehat{w}}$ using the power method (Lemma 2.2);
return $z_1$;
```

To prove that the largest eigenvector of $Y_{\widehat{w}}$ is close to x, we will show that $x^\top Y_{\widehat{w}} x$ is large, and there is a gap between the first and second largest eigenvalues of $Y_{\widehat{w}}$.

Lemma 4.1. *Consider the setting of Problem 1.3, where $x \in \mathbb{R}^d$ is the ground-truth vector. Fix $\delta > 0$. There exists constants $\epsilon(\delta)$ and $c(\delta)$ such that if we are given an $\epsilon(\delta)$-corrupted set of $n = \widetilde{\Omega}(c(\delta) d)$ samples $(a_i, y_i)_{i \in [n]}$, Algorithm 1 outputs $\widehat{w} \in \mathbb{R}^n$ in time $\widetilde{O}(nd \operatorname{poly}(1/\epsilon(\delta)))$ such that, with probability at least 0.98, the following conditions hold:*

$$x^\top Y_{\widehat{w}} x \ge 3 - O(\delta), \quad |\lambda_1(Y_{\widehat{w}}) - 3| \le O(\delta), \text{ and } \quad |\lambda_2(Y_{\widehat{w}}) - 1| \le O(\delta).$$

where $Y_{\widehat{w}} = \sum_{i=1}^n \widehat{w}_i y_i a_i a_i^\top$.

We defer the proof of Lemma 4.1 to Appendix A.1 and first use it to prove the correctness and runtime of Algorithm 1 (Lemma 3.1).

Proof of Lemma 3.1. Let $Y_{\widehat{w}} = \sum_{i=1}^{d} \lambda_i v_i v_i^\top$ be the eigendecomposition of $Y_{\widehat{w}}$, where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$. Let $x = \sum_{i=1}^{d} \alpha_i v_i$. Note that $\sum_{i=1}^{d} \alpha_i^2 = \|x\|_2^2 = 1$ and $|\alpha_i| \leq 1$. If the event of Lemma 4.1 is true, with $\epsilon < \epsilon(\delta)$ and $m = \widetilde{\Omega}(c(\delta) d)$, then

$$\begin{aligned} 3 - O(\delta) \leq x^\top Y_{\widehat{w}} x &= \textstyle\sum_{i=1}^{d} \lambda_i \alpha_i^2 \leq \lambda_1 \alpha_1^2 + \lambda_2 (1 - \alpha_1^2) \\ &\leq (3 + O(\delta)) \alpha_1^2 + (1 + O(\delta))(1 - \alpha_1^2) \leq 1 + O(\delta) + 2\alpha_1^2 \ . \end{aligned}$$

This implies $|\alpha_1| \geq \alpha_1^2 \geq 1 - O(\delta)$. Consequently,

$$\begin{aligned} \min\{\|v_1 - x\|_2^2, \|v_1 + x\|_2^2\} &= \min\{(1 - \alpha_1)^2, (1 + \alpha_1)^2\} + \textstyle\sum_{i=2}^{d} \alpha_i^2 \\ &= \min\{2 - 2\alpha_1, 2 + 2\alpha_1\} = 2 - 2|\alpha_1| \leq O(\delta) \ . \end{aligned}$$

We choose δ as a sufficiently small constant so that $\min\{\|v_1 - x\|_2, \|v_1 + x\|_2\} \leq O(\sqrt{\delta}) \leq \frac{1}{16}$. Note that since it is sufficient to choose δ as a small universal constant, $c(\delta)$ and $\epsilon(\delta)$ can also be treated as universal constants.

We use the power method to approximate the largest eigenvector z_1 of $Y_{\widehat{w}}$. By Lemma 2.2, $\|z_1\|_2 = 1$ and $z_1^\top Y_{\widehat{w}} z_1 \geq (1 - \epsilon_2)\lambda_1$. Choosing $\epsilon_2 = O(\delta)$ and using Lemma 4.1, we have $z_1^\top Y_{\widehat{w}} z_1 \geq 3 - O(\delta)$. By the same arguments, we can show that $\min\{\|v_1 - z_1\|_2, \|v_1 + z_1\|_2\} \leq \frac{1}{16}$, and then by the triangle inequality, we conclude that $\min\{\|z_1 - x\|_2, \|z_1 + x\|_2\} \leq \frac{1}{8}$.

The success probability of Algorithm 1 is at least 0.97, as $\widehat{w}$ satisfies Lemma 4.1 with probability at least 0.98, and the power method succeeds with probability at least 0.99 by Lemma 2.2.

The required number of samples is $n = \widetilde{\Omega}(d)$ by Lemma 4.1. Algorithm 1 runs in time $\widetilde{O}(nd)$: It takes time $\widetilde{O}(nd \operatorname{poly}(1/\epsilon(\delta))) = \widetilde{O}(nd)$ to compute $\widehat{w}$ by Lemma 4.1. The power method can approximate the largest eigenvector of $Y_{\widehat{w}}$ in time $O(nd \log(d)/\delta) = \widetilde{O}(nd)$ by Lemma 2.2, since the matrix-vector product $Y_{\widehat{w}} v = \sum_{i=1}^{n} \widehat{w}_i y_i \langle a_i, v \rangle a_i$ can be computed in time $O(nd)$ for any $v \in \mathbb{R}^d$. □

5 Robust Gradient Descent

After the robust initialization in Section 4, we have an initial guess $z_1 \in \mathbb{R}^d$ that is close to the ground truth x or $-x$. We can assume without loss of generality that z_1 is closer to x than to $-x$.

In this section, we prove Lemma 3.2: Given an initial guess z_1 with $\|z_1 - x\|_2 \leq \frac{1}{8}$, we can use a robust gradient descent algorithm (Algorithm 2) to recover x to any desired precision $\Delta > 0$. We will show that Algorithm 2 converges geometrically even when the input is ϵ-corrupted.

Consider the following nonconvex optimization problem:

$$\min_{z \in \mathbb{R}^d} \sum_{i=1}^{n} f_i(z) \quad \text{where} \quad f_i(z) = \left(\langle a_i, z \rangle^2 - y_i \right)^2 \ .$$

Let g_i denote the gradient of f_i with respect to z. Let $\mathcal{D}_z$ denote the distribution of $g_i(z)$ when there is no corruption. Formally, $g(z) \sim \mathcal{D}_z$ is distributed as

$$g(z) = \frac{\partial}{\partial z} \left[\left(\langle a, z \rangle^2 - \langle a, x \rangle^2 \right)^2 \right] = 4 \left(\langle a, z \rangle^2 - \langle a, x \rangle^2 \right) \langle a, z \rangle \, a \ \text{ where } \ a \sim \mathcal{N}(0, I) \ . \quad (2)$$

To perform gradient descent, we want to approximate the *expected true gradient*

$$\mu_z = \mathop{\mathbb{E}}_{g(z) \sim \mathcal{D}_z} [g(z)] = \left(12 \|z\|_2^2 - 4 \|x\|_2^2 \right) z - 8 \langle x, z \rangle x \ . \quad (3)$$

However, the input $\{(a_i, y_i)\}_{i \in [n]}$ is ϵ-corrupted, so the corresponding gradients $\{g_i(z)\}_{i \in [n]}$ are an ϵ-corrupted set of vectors drawn from $\mathcal{D}_z$. We will run nearly-linear time robust mean estimation algorithms (e.g., [18]) on $\{g_i(z)\}_{i \in [n]}$ to approximate the true gradient μ_z.

The accuracy of robust mean estimation algorithms depends on the covariance matrix Σ_z of the distribution $\mathcal{D}_z$. The following lemma upper bounds the spectral norm of Σ_z.

Algorithm 2: Robust Gradient Descent

```
Input: An $\epsilon$-corrupted set of $n$ samples $\{(a_i, y_i)\}_{i \in [n]}$, an initial guess $z_1$ with $\|z_1 - x\| \le \frac{1}{8}$,
       and desired precision $\Delta > 0$.
Output: $z \in \mathbb{R}^d$ such that $\|z - x\|_2 \le \Delta$, where $x$ is the ground-truth vector.
$T \leftarrow O(\log(1/\Delta))$, $\eta \leftarrow \frac{1}{300}$;
$\{N_1, \ldots, N_T\} \leftarrow$ a random disjoint partition of $[n]$ such that $|N_t| = \frac{n}{T}$ for all $t \in [T]$;
for $t = 1, 2, \ldots, T$ do
    $g_i(z_t) \leftarrow 4\left(\langle a_i, z_t\rangle^2 - y_i\right)\langle a_i, z_t\rangle\, a_i$;
    $\widehat{\mu}_{z_t} \leftarrow$ robust mean estimation on input $\{g_i(z_t)\}_{i \in N_t}$ using Lemma 5.2;
    $z_{t+1} \leftarrow z_t - \eta\, \widehat{\mu}_{z_t}$;
end
return $z_{T+1}$;
```

Lemma 5.1. *Let $x \in \mathbb{R}^d$ be the ground-truth vector. Let $\mathcal{D}_z$ be the distribution of gradients at z as defined in Equation* (2)*. For any z with $\|z - x\|_2 \le 1$, the covariance matrix Σ_z of $\mathcal{D}_z$ satisfies*

$$\Sigma_z \preceq O\left(\|z - x\|_2^2\right) I \ .$$

We defer the proof of Lemma 5.1 to Appendix B. For technical reasons, we randomly partition the input $\{(a_i, y_i)\}_{i \in [n]}$ into T subsets and use one subset in each iteration. With high probability, each partition has at most (2ϵ)-fraction of corrupted samples. The next lemma shows that, given the covariance bound in Lemma 5.1, we can approximate the true gradient μ_z from a (2ϵ)-corrupted set of gradients with a small error.

Lemma 5.2. *Let $x \in \mathbb{R}^d$ be the ground-truth vector. Consider any $z \in \mathbb{R}^d$ with $\|z - x\|_2 \le 1$. Let $\mathcal{D}_z$ be the distribution defined in Equation* (2) *and let μ_z be the mean of $\mathcal{D}_z$. Let $c > 0$, and let $\tau \in (0, 1/4)$ be a small constant. There exists a constant $\epsilon(c)$ such that given a $2\epsilon(c)$-corrupted set of $m = \Omega(d \log(d)/(c^2\tau))$ vectors drawn from $\mathcal{D}_z$, we can compute $\widehat{\mu}_z \in \mathbb{R}^d$ in time $\widetilde{O}(md \log(1/\tau))$ such that $\|\widehat{\mu}_z - \mu_z\|_2 \le c\,\|z - x\|_2$ with probability at least $1 - 2\tau$.*

Proof of Lemma 5.2. We constraint $\epsilon(c) \le \epsilon_3/2$, where ϵ_3 is the universal constant in the robust mean estimation algorithm stated in Lemma 2.3. By Lemma 5.1, the covariance matrix Σ_z of $\mathcal{D}_z$ satisfies $\Sigma_z \preceq O\left(\|z - x\|_2^2\right) I$. Lemma 2.3 guarantees that robust mean estimation returns a vector $\widehat{\mu}_z \in \mathbb{R}^d$ such that

$$\|\widehat{\mu}_z - \mu_z\|_2 \le O\left(\sqrt{2\epsilon(c)} + \sqrt{\frac{d}{m\tau}} + \sqrt{\frac{d(\log d + \log(1/\tau))}{m}}\right)\sqrt{\|\Sigma_z\|_2} \le c\,\|z - x\|_2 \ .$$

The last inequality follows by properly choosing the constant $\epsilon(c) = O(c^2)$. By Lemma 2.3, the runtime is $\widetilde{O}(md \log(1/\tau))$ and the success probability is at least $1 - \tau - \exp(-\epsilon_0 m) \ge 1 - 2\tau$. □

The next lemma shows that the approximate gradient from Lemma 5.2 is sufficient for gradient descent to converge, reducing the distance to the ground truth x by a constant factor in each iteration. We provide a proof sketch for Lemma 5.3 and defer the full proof to Appendix B.

Lemma 5.3. *Let $x \in \mathbb{R}^d$ be the ground-truth vector. Suppose at iteration t of Algorithm 2, the current solution z_t satisfies $\|z_t - x\|_2 \le \frac{1}{8}$. Let μ_{z_t} denote the expected true gradient at z_t defined in Equation* (3)*. Suppose the estimated gradient $\widehat{\mu}_{z_t} \in \mathbb{R}^d$ satisfies $\|\widehat{\mu}_{z_t} - \mu_{z_t}\|_2 \le c\,\|z_t - x\|_2$ for $c = 4$. Then, we have*

$$\|z_{t+1} - x\|_2^2 \le 0.99\,\|z_t - x\|_2^2 \ .$$

Proof Sketch of Lemma 5.3. Even though the objective function is nonconvex, it is known that gradient descent is well-behaved when initialized close enough to a global optimum [37]. More specifically, for any z with $\|z-x\|_2 \le \frac{1}{8}$, we can show that the expected true gradient at z aligns with the direction $(z-x)$:

$$\langle \mu_z, z-x\rangle \ge 7.5\,\|z-x\|_2^2 \quad \text{and} \quad \|\mu_z\|_2 \le 29\,\|z-x\|_2 \ ,$$

which is sufficient for proving geometric convergence.

Note that this analysis is robust to small error in μ_z. When $\|\widehat{\mu}_z - \mu_z\|_2 \le c\,\|z-x\|_2$, we have

$$\langle \widehat{\mu}_z, z-x\rangle \ge (7.5-c)\,\|z-x\|_2^2 \quad \text{and} \quad \|\widehat{\mu}_z\|_2 \le (29+c)\,\|z-x\|_2 \ .$$

When $c < 7.5$, we can choose an appropriate step size η such that the distance between z and x decreases by a constant factor in each iteration. □

We are now ready to prove Lemma 3.2, which states the correctness and runtime of Algorithm 2.

Lemma 3.2. First, we analyze the success probability of Algorithm 2. Algorithm 2 can fail in two ways: *(i)* some N_t has more than (2ϵ)-fraction of corrupted samples, or *(ii)* robust gradient estimation fails in some iteration t. The probability of event *(i)* is at most 0.01 for our choice of n, which follows from a standard application of Hoeffding's inequality and a union bound. For event *(ii)*, we choose $\tau \le 0.005/T$ in Lemma 5.2, so each robust gradient estimation fails with probability at most $2\tau = 0.01/T$. By a union bound over T iterations, the probability of event *(ii)* is at most 0.01. For the rest of the proof, we assume these bad events do not happen.

Next, we prove the correctness of Algorithm 2. Since $\|z_1 - x\|_2 \le \frac{1}{8}$, we can use Lemma 5.2 to obtain an approximation $\widehat{\mu}_{z_1}$ of the true gradient μ_{z_1} such that $\|\widehat{\mu}_{z_1} - \mu_{z_1}\|_2 \le c\,\|z_1 - x\|_2$ with $c = 4$. Then, by Lemma 5.3, we have $\|z_2 - x\|_2 \le 0.99\,\|z_1 - x\|_2$ after one iteration of gradient descent. Applying these two lemmas repeatedly, after $T = O(\log(1/\Delta))$ iterations, we have $\|z_{T+1} - x\|_2 \le \Delta$.

Finally, we analyze the sample complexity and runtime of Algorithm 2. The algorithm requires in total $n = mT = \Omega(d \log d \log^2(1/\Delta))$ samples. A random partition can be computed in $O(n)$ time via random shuffling. In each iteration, the m gradients in N_t can be computed in time $O(md)$ using Equation (2). By Lemma 5.2, the true gradient can be robustly estimated in time $\widetilde{O}(md \log(1/\tau)) = \widetilde{O}(md \log T) = \widetilde{O}(md \log\log(1/\Delta))$, and z_t can be updated in time $O(d)$. The overall runtime of Algorithm 2 is $\widetilde{O}(n + Tmd \log\log(1/\Delta)) = \widetilde{O}(nd)$. □

Remark. There are two technical details worth noting. First, robust mean estimation algorithms (Lemma 2.3) require a *known* upper bound σ^2 on the spectral norm of the covariance matrix of $\mathcal{D}_z$. By Lemma 5.1, a *known* upper bound on $\|z-x\|_2$ suffices. We can indeed maintain such an upper bound, which starts at $\frac{1}{8}$ and decreases geometrically, as shown in Lemma 5.3. Second, at each iteration t, to apply Lemma 5.2 to robustly estimate the gradient at z_t, we need a (2ϵ)-corrupted set of gradients drawn from $\mathcal{D}_{z_t}$. This is why we use a set of fresh samples N_t. By the principle of deferred decisions, we can view $(a_i, y_i)_{i \in N_t}$ as being generated and corrupted *after* z_t is chosen.

6 Conclusions and Future Directions

In this paper, we propose and study the problem of outlier-robust phase retrieval, where a small fraction of the input data is corrupted. Importantly, we allow adversarial corruption in both the sampling vectors $a_i \in \mathbb{R}^d$ and the intensity measurements $y_i \in \mathbb{R}$. We present a near-sample-optimal and nearly-linear-time algorithm for this problem with provable guarantees. One conceptual contributions of our work is that phase retrieval can be solved using a robust first-order methods even when the input is slightly misspecified or corrupted. Our algorithmic framework provides a general approach for developing robust algorithms for a wide range of tractable nonconvex problems, by first robustly initializing into a region free of saddle points and then using robust gradient descent to converge to a global optimum.

An immediate technical question is whether our sample complexity can be tightened by removing the $\log(1/\Delta)$ factors. One potential approach is to examine the stability conditions required by robust mean estimation algorithms and see if these conditions can be proved without using fresh samples in each iteration.

References

[1] P. Abdalla and N. Zhivotovskiy. Covariance estimation: Optimal dimension-free guarantees for adversarial corruption and heavy tails. *Journal of the European Mathematical Society*, 2024.

[2] S. Bhojanapalli, B. Neyshabur, and N. Srebro. Global optimality of local search for low rank matrix recovery. *Advances in Neural Information Processing Systems*, 29, 2016.

[3] A. Buna and P. Rebeschini. Robust gradient descent for phase retrieval. In *The 28th International Conference on Artificial Intelligence and Statistics*.

[4] E. J. Candès, X. Li, and M. Soltanolkotabi. Phase retrieval via wirtinger flow: Theory and algorithms. *IEEE Trans. Inf. Theory*, 61(4):1985–2007, 2015a.

[5] E. J. Candès, X. Li, and M. Soltanolkotabi. Phase retrieval from coded diffraction patterns. *Applied and Computational Harmonic Analysis*, 39(2):277–299, Sept. 2015b. ISSN 1063-5203.

[6] E. J. Candès, Y. C. Eldar, T. Strohmer, and V. Voroninski. Phase retrieval via matrix completion. *SIAM Rev.*, 57(2):225–251, 2015c.

[7] J. Chen, L. Wang, X. Zhang, and Q. Gu. Robust Wirtinger Flow for Phase Retrieval with Arbitrary Corruption, Jan. 2018.

[8] Y. Chen and E. Candes. Solving Random Quadratic Systems of Equations Is Nearly as Easy as Solving Linear Systems. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[9] Y. Cheng and R. Ge. Non-convex matrix completion against a semi-random adversary. In *Proc. 31st Conference on Learning Theory (COLT)*, volume 75, pages 1362–1394, 2018.

[10] Y. Cheng, I. Diakonikolas, and R. Ge. High-dimensional robust mean estimation in nearly-linear time. In *Proc. 30th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2755–2771, 2019.

[11] Y. Cheng, I. Diakonikolas, R. Ge, and D. P. Woodruff. Faster algorithms for high-dimensional robust covariance estimation. In A. Beygelzimer and D. Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 727–757. PMLR, 25–28 Jun 2019.

[12] Y. Cherapanamjeri, S. Mohanty, and M. Yau. List decodable mean estimation in nearly linear time. In *Proc. 61st IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 141–148, 2020.

[13] C. Dainty and J. R. Fienup. Phase retrieval and image reconstruction for astronomy. *Image recovery: theory and application*, 231:275, 1987.

[14] J. Depersin and G. Lecué. Robust sub-Gaussian estimation of a mean vector in nearly linear time. *The Annals of Statistics*, 50(1):511–536, 2022.

[15] I. Diakonikolas and D. M. Kane. *Algorithmic High-Dimensional Robust Statistics*. Cambridge University Press, 2023.

[16] I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, A. Moitra, and A. Stewart. Robust estimators in high dimensions without the computational intractability. In *57th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2016*, pages 655–664. IEEE Computer Soc., Los Alamitos, CA, 2016.

[17] I. Diakonikolas, G. Kamath, D. Kane, J. Li, J. Steinhardt, and A. Stewart. Sever: A robust meta-algorithm for stochastic optimization. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1596–1606. PMLR, 09–15 Jun 2019.

[18] Y. Dong, S. B. Hopkins, and J. Li. Quantum entropy scoring for fast robust mean estimation and improved outlier detection. In *Proc. 33rd Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[19] J. C. Duchi and F. Ruan. Solving (most) of a set of quadratic equalities: Composite optimization for robust phase retrieval. *Information and Inference: A Journal of the IMA*, 8(3):471–529, 2019.

[20] J. R. Fienup. Reconstruction of an object from the modulus of its Fourier transform. *Optics Letters*, 3(1):27–29, July 1978. ISSN 1539-4794.

[21] J. R. Fienup. Phase retrieval algorithms: A comparison. *Applied Optics*, 21(15):2758–2769, Aug. 1982. ISSN 2155-3165.

[22] X. Gao and Y. Cheng. Robust matrix sensing in the semi-random model. *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[23] R. Ge, F. Huang, C. Jin, and Y. Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, pages 797–842, 2015.

[24] R. Ge, J. D. Lee, and T. Ma. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems*, pages 2973–2981, 2016.

[25] R. W. Gerchberg. A practical algorithm for the determination of phase from image and diffraction plane pictures. *Optik*, Jan. 1972.

[26] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust statistics. The approach based on influence functions*. Wiley New York, 1986.

[27] P. Hand and V. Voroninski. Corruption robust phase retrieval via linear programming. *CoRR*, abs/1612.03547, 2016.

[28] P. J. Huber and E. M. Ronchetti. *Robust statistics*. Wiley New York, 2009.

[29] K. Jaganathan, Y. C. Eldar, and B. Hassibi. Phase retrieval: An overview of recent developments. *Optical Compressive Imaging*, pages 279–312, 2016.

[30] R. Kolte and A. Özgür. Phase Retrieval via Incremental Truncated Wirtinger Flow, June 2016.

[31] K. A. Lai, A. B. Rao, and S. Vempala. Agnostic estimation of mean and covariance. In *focs2016*, pages 665–674, 2016.

[32] G. Lecué, M. Lerasle, and T. Mathieu. Robust classification via MOM minimization. *Mach. Learn.*, 109(8):1635–1665, 2020.

[33] S. Li, Y. Cheng, I. Diakonikolas, J. Diakonikolas, R. Ge, and S. Wright. Robust second-order nonconvex optimization and its application to low rank matrix sensing. In *Proc. 37th Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[34] J.-W. Liu, Z.-J. Cao, J. Liu, X.-L. Luo, W.-M. Li, N. Ito, and L.-C. Guo. Phase Retrieval via Wirtinger Flow Algorithm and Its Variants. In *2019 International Conference on Machine Learning and Cybernetics (ICMLC)*, pages 1–9, July 2019.

[35] J. Miao, T. Ishikawa, B. Johnson, E. H. Anderson, B. Lai, and K. O. Hodgson. High resolution 3D X-ray diffraction microscopy. *Physical review letters*, 89(8):088303, 2002.

[36] R. P. Millane. Phase retrieval in crystallography and optics. *JOSA A*, 7(3):394–411, 1990.

[37] P. Netrapalli, P. Jain, and S. Sanghavi. Phase retrieval using alternating minimization. In *Proc. 27th Advances in Neural Information Processing Systems (NeurIPS)*, pages 2796–2804, 2013.

[38] A. Prasad, A. S. Suggala, S. Balakrishnan, and P. Ravikumar. Robust estimation via robust gradient estimation. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, 82(3):601–627, 2020.

[39] W. H. Robert. Phase problem in crystallography. *JOSA a*, 10(5):1046–1055, 1993.

[40] Y. Shechtman, Y. C. Eldar, O. Cohen, H. N. Chapman, J. Miao, and M. Segev. Phase retrieval with application to optical imaging: a contemporary overview. *IEEE signal processing magazine*, 32(3):87–109, 2015.

[41] M. Soltanolkotabi. Structured signal recovery from quadratic measurements: Breaking sample complexity barriers via nonconvex optimization. *IEEE Trans. Inf. Theory*, 65(4):2374–2400, 2019.

[42] J. Sun, Q. Qu, and J. Wright. Complete dictionary recovery over the sphere i: Overview and the geometric picture. *IEEE Trans. Inf. Theor.*, 63(2):853–884, 2 2017.

[43] J. Sun, Q. Qu, and J. Wright. A geometric analysis of phase retrieval. *Found. Comput. Math.*, 18(5):1131–1198, 2018.

[44] L. Trevisan. Lecture notes on graph partitioning, expanders and spectral methods. *University of California, Berkeley, https://people. eecs. berkeley. edu/luca/books/expanders-2016. pdf*, 2017.

[45] J. Tukey. Mathematics and picturing of data. In *Proceedings of ICM*, volume 6, pages 523–531, 1975.

[46] R. Vershynin. *High-dimensional probability*, volume 47 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge, 2018.

[47] V. Voroninski. *PhaseLift: A Novel Methodology for Phase Retrieval*. PhD thesis, UC Berkeley, 2013.

[48] G. Wang, G. Giannakis, Y. Saad, and J. Chen. Solving most systems of random quadratic equations. *Advances in Neural Information Processing Systems*, 30, 2017.

[49] G. Wang, G. B. Giannakis, Y. Saad, and J. Chen. Phase retrieval via reweighted amplitude flow. *IEEE Transactions on Signal Processing*, 66(11):2818–2833, 2018.

[50] J. Wright and Y. Ma. *High-dimensional data analysis with low-dimensional models: Principles, computation, and applications*. Cambridge University Press, 2022.

[51] H. Zhang, Y. Chi, and Y. Liang. Provable non-convex phase retrieval with outliers: Median truncated wirtinger flow. In *International conference on machine learning*, pages 1022–1031. PMLR, 2016.

A Omitted Proofs in Section 4

A.1 Proof of Lemma 4.1

In this section, we prove Lemma 4.1: We can compute $\widehat{w} \in \mathbb{R}^n$ in nearly-linear time such that $x^\top Y_{\widehat{w}} x$ is large and the two largest eigenvalues of $Y_{\widehat{w}}$ are approximately 3 and 1.

The following lemma shows that, for any $w \in \Delta_{n,2\epsilon}$, the contribution of the remaining good samples to Y_w is close to $I + 2xx^\top$. This allows us to lower bound $\lambda_1(Y_{\widehat{w}})$, $\lambda_2(Y_{\widehat{w}})$, and $x^\top Y_{\widehat{w}} x$.

Lemma A.1. *Let $x \in \mathbb{R}^d$ be the ground-truth vector. Fix $\delta > 0$. There exists constants $\overline{\epsilon}(\delta)$ and $c(\delta)$ such that if we let $0 < \epsilon \le \overline{\epsilon}(\delta)$, and we are given an ϵ-corrupted set of $n = \widetilde{\Omega}(c(\delta)d)$ samples $(a_i, y_i)_{i\in[n]}$, with probability at least 0.99, for all $w \in \Delta_{n,2\epsilon}$,*

$$\left\| Y_{G,w} - (I + 2xx^\top) \right\|_2 \le \delta \ ,$$

where G is the set of indices of the remaining good samples and $Y_{G,w} = \sum_{i\in G} w_i y_i a_i a_i^\top$.

Intuitively, Lemma A.1 holds because the moments of Gaussian distributions are stable when a small fraction of the samples are removed. We defer the proof of Lemma A.1 to Appendix A.

The next lemma shows that, assuming Lemma A.1 holds, we can compute a near-optimal $\widehat{w}$ that minimizes $\lambda_1(Y_{\widehat{w}}) + \lambda_2(Y_{\widehat{w}})$ in nearly-linear time.

Lemma A.2. *Let $\delta \in (0, 1/2)$. There exists a constant $\epsilon(\delta)$ such that if we are given an $\epsilon(\delta)$-corrupted set of n samples $(a_i, y_i)_{i\in[n]}$ that satisfies Lemma A.1, we can compute $\widehat{w} \in \Delta_{n,2\epsilon(\delta)}$ in time $\widetilde{O}(nd \operatorname{poly}(1/\epsilon(\delta))$ such that with probability at least 0.99,*

$$\lambda_1(Y_{\widehat{w}}) + \lambda_2(Y_{\widehat{w}}) \le 4 + O(\delta) \ ,$$

where d is the dimension of a_i and $Y_{\widehat{w}} = \sum_{i=1}^n \widehat{w}_i y_i a_i a_i^\top$.

Proof. We reduce the optimization problem $\min_{w\in\Delta_{n,\epsilon(\delta)}} [\lambda_1(Y_{\widehat{w}}) + \lambda_2(Y_{\widehat{w}})]$ to the following Ky Fan norm packing semidefinite program (SDP):

$$\max_{w\in\mathbb{R}^n_{\ge 0}} \quad \|w\|_1 \quad \text{subject to} \quad \sum_{i=1}^n w_i A_i \preceq I, \quad \overline{\lambda}_2\left(\sum_{i=1}^n w_i B_i\right) \le 4 + O(\delta) \qquad (*)$$

in which $A_i = (1 - \epsilon(\delta))n e_i e_i^\top$ where $e_i \in \mathbb{R}^n$ is the i^{th} basis vector, $B_i = y_i a_i a_i^\top$, and $\overline{\lambda}_2$ is the sum of the two largest eigenvalues.

Let G be the set of indices of the remaining good samples. Consider the weight vector $w^\star \in \mathbb{R}^n$ that is uniform on G:

$$w_i^* = \begin{cases} \frac{1}{|G|} = \frac{1}{(1-\epsilon)n} & i \in G \ , \\ 0 & i \notin G \ . \end{cases}$$

Let $\epsilon(\delta) = \min\{\delta, \overline{\epsilon}(\delta)\}$, where $\overline{\epsilon}(\delta)$ is the constant of Lemma A.1. By Lemma A.1, $Y_{w^\star} \preceq (1+\delta)I + 2xx^\top$, which implies $\overline{\lambda}_2(Y_{w^\star}) \le 4 + O(\delta)$. Since $w^\star$ is feasible, the optimal value OPT of (*) must be at least $\|w^\star\|_1 = 1$.

We invoke Lemma 2.1 to solve (*) with failure probability $\tau = 0.01$ and error tolerance parameter $\epsilon_1 = \epsilon(\delta)$. The resulting solution $w' \in \mathbb{R}^n_{\ge 0}$ satisfies $\|w'\|_1 \ge (1 - \epsilon_1)\mathrm{OPT} \ge 1 - \epsilon(\delta)$. Define $\widehat{w} = \frac{w'}{\|w'\|_1}$ so that $\|\widehat{w}\|_1 = 1$. The constraint with A_i guarantees that $\|w'\|_\infty \le \frac{1}{(1-\epsilon(\delta))n}$, so $\|\widehat{w}\|_\infty \le \frac{\|w'\|_\infty}{1-\epsilon(\delta)} \le \frac{1}{(1-2\epsilon(\delta))n}$ and thus $w \in \Delta_{n,2\epsilon(\delta)}$. The constraint with B_i implies that $\overline{\lambda}_2(Y_{w'}) \le 4 + O(\delta)$, and after scaling we have $\overline{\lambda}_2(Y_{\widehat{w}}) \le \frac{4+O(\delta)}{1-\epsilon(\delta)} \le 4 + O(\delta)$ since $\epsilon(\delta) \le \delta < \frac{1}{2}$. The success probability is at least $1 - \tau = 0.99$.

We can write $A_i = C_i C_i^\top$ with $C_i = \sqrt{(1-\epsilon)n} e_i$, and $B_i = D_i D_i^\top$ with $D_i = \sqrt{y_i} a_i$. It takes $O(1)$ time to perform a matrix-vector product with C_i, and $O(d)$ time with D_i. Therefore, $t_C + t_D$ in Lemma 2.1 is $O(nd)$, and the runtime of Lemma 2.1 is $\widetilde{O}(nd \operatorname{poly}(1/\epsilon))$ for $\tau = 0.01$. □

We now proceed to prove Lemma 4.1.

Proof of Lemma 4.1. Without loss of generality, we can assume that $y_i \geq 0$ for all $i \in [n]$. Because y_i should be $\langle a_i, x\rangle^2$, any sample with $y_i < 0$ must be corrupted and can be discarded. By Lemma A.2, we can compute $\widehat{w} \in \Delta_{n,2\epsilon(\delta)}$ such that $\lambda_1(Y_{\widehat{w}}) + \lambda_2(Y_{\widehat{w}}) \leq 4 + O(\delta)$.

By Lemma A.1 and $y_i \geq 0$, we have

$$Y_{\widehat{w}} = \sum_{i \in S} \widehat{w}_i y_i a_i a_i^\top \succeq \sum_{i \in G} \widehat{w}_i y_i a_i a_i^\top = Y_{G,\widehat{w}} \succeq (1-\delta)I + 2xx^\top \ ,$$

which gives a lower bound $x^\top Y_{\widehat{w}} x \geq 3 - O(\delta)$, as well as lower bounds on the eigenvalues of $Y_{\widehat{w}}$:

$$\lambda_1(Y_{\widehat{w}}) \geq 3 - O(\delta) \quad \text{and} \quad \lambda_2(Y_{\widehat{w}}) \geq 1 - O(\delta) \ .$$

Putting the upper and lower bounds together, we obtain

$$\begin{aligned} \lambda_1(Y_{\widehat{w}}) &= \overline{\lambda}_2(Y_{\widehat{w}}) - \lambda_2(Y_{\widehat{w}}) \leq 4 + O(\delta) - (1 - O(\delta)) \leq 3 + O(\delta) \ , \text{ and} \\ \lambda_2(Y_{\widehat{w}}) &= \overline{\lambda}_2(Y_{\widehat{w}}) - \lambda_1(Y_{\widehat{w}}) \leq 4 + O(\delta) - (3 - O(\delta)) \leq 1 + O(\delta) \ . \end{aligned}$$

Lemma A.1 holds with probability at least 0.99. Assuming Lemma A.1 holds, Lemma 2.1 succeeds with probability at least 0.99, so the success probability of Lemma 4.1 is at least 0.98.

For the initialization step, it suffices to use Lemma A.2 to find a $(1 - \epsilon(\delta))$-optimal solution $\widehat{w} \in \Delta_{n,2\epsilon(\delta)}$ to the SDP (*), and the runtime to compute such $\widehat{w}$ is $\widetilde{O}(nd \,\mathrm{poly}(1/\epsilon(\delta)))$. □

A.2 Proof of Lemma A.1

This section is devoted to the proof of Lemma A.1. We will use the following concentration results.

Lemma A.3 ([4], Section A.4.2). *Let $x \in \mathbb{R}^d$. For any $\delta > 0$, there exists a constant $C(\delta) > 0$ such that when $n > C(\delta) \cdot d \log d$ and we are given a set of n samples $\{(a_i, y_i)\}_{i=1}^n$ with $a_i \sim \mathcal{N}(0, I)$ independently and $y_i = \langle a_i, x\rangle^2$ for all $i \in [n]$, then with probability at least 0.99, it holds*

$$\left\| \frac{1}{n} \sum_{i=1}^n y_i a_i a_i^\top - (I + 2xx^\top) \right\|_2 \leq \delta \ .$$

Proposition A.4. *Let $\alpha \in (0, 2/e)$. Let $X_1, \ldots, X_m$ be m random variables drawn i.i.d. from $\mathcal{N}(0,1)$. Define*

$$H = \{i \in [m] : |X_i| \geq 4 \ln^2(2/\alpha)\} \ .$$

With probability at least $1 - 10^{-3}$, the following are all true:

$$\begin{aligned} &(a) \quad |H| \leq O(m\alpha) \ , \\ &(b) \quad \sum_{i \in H} X_i^4 = O\big(m\alpha \log^8(2/\alpha)\big) \ , \\ &(c) \quad \max_{i \in H} X_i^2 = O(\log m) \ . \end{aligned}$$

Proof. For $X \sim \mathcal{N}(0,1)$ and $t > 0$, it holds that $\mathbf{Pr}[|X| \geq t] \leq 2\exp(-t^2/2)$. By setting $t = 4\ln^2(2/\alpha)$, we have $\mathbf{Pr}\big[X^4 \geq t\big] \leq \alpha$. Let $Y_i \in \{0,1\}$ be the indicator random variable for the event "$i \in H$", so that $|H| = \sum_{i \in H} Y_i$.

Because $\mathbb{E}\big[\sum_{i \in H} Y_i\big] \leq m\alpha$, by Markov's inequality, we have $|H| = O(m\alpha)$ with probability at least $1 - 10^{-3}/3$. We assume this holds for the rest of the proof.

By the principle of deferred decisions, an equivalent way to draw $X_1, \ldots, X_m$ from $\mathcal{N}(0,1)$ is to first draw Y_i, and then draw X_i conditioned on the value of Y_i. Note that H is fixed after drawing $Y_1, \ldots, Y_m$.

$$\mathbb{E}\left[\sum_{i \in H} X_i^4\right] = \sum_{i \in H} \mathbb{E}\big[X_i^4 \bigm| |X_i| \geq 4\ln^2(2/\alpha)\big] \leq O(m\alpha)\, \mathbb{E}\big[X_i^4 \bigm| |X_i| \geq 4\ln^2(2/\alpha)\big] \ . \quad (4)$$

For any $t \ge 4$, by the definition of conditional expectation, and using the fact that X_i is normally distributed:

$$\mathbb{E}\big[X_i^4 \bigm| |X_i| \ge t\big] = \frac{\frac{2}{\sqrt{2\pi}}\int_t^{\infty} x^4 e^{-x^2/2}dx}{\frac{2}{\sqrt{2\pi}}\int_t^{\infty} e^{-x^2/2}dx} = \frac{\int_t^{\infty} x^4 e^{-x^2/2}dx}{\int_t^{\infty} e^{-x^2/2}dx} \le 2t^4 \ . \tag{5}$$

We use Inequality (5) to upper bound (4):

$$\mathbb{E}\left[\sum_{i\in H} X_i^4\right] \le O(m\alpha)\left(2\cdot(4\ln^2(2/\alpha))^4\right) = O\big(m\alpha\log^8(2/\alpha)\big) \ .$$

By Markov's inequality, with probability at least $1 - 10^{-3}/3$, we have $\mathbb{E}\big[\sum_{i\in H} X_i^4\big] = O\big(m\alpha\log^8(2/\alpha)\big)$. Finally, since $X_1,\dots,X_m$ are drawn i.i.d. from $\mathcal{N}(0,1)$, we have that $\max_{i\in[m]}|X_i| = O(\sqrt{\log m})$ with probability at least $1-10^{-3}/3$. □

Proposition A.5. *Let $K_1 \ge 0$ and $K_2 \ge 0$. Let $a_1,\dots,a_m \ge 0$ such that $\max_{i\in[m]} a_i^2 \le K_1$ and $\sum_{i\in[m]} a_i^4 \le K_2$. Let $X_1,\dots,X_m$ be m random variables drawn i.i.d. from $\mathcal{N}(0,1)$. Then, with probability at least $1-10^{-3}12^{-d}$,*

$$\sum_{i\in[m]} a_i^2 X_i^2 = O\left(\sqrt{dK_2} + dK_1\right) \ .$$

Proof. Since $X_i \sim \mathcal{N}(0,1)$, the random variable X_i^2 is sub-exponential. Applying Bernstein's inequality for sub-exponential random variable [46, Theorem 2.8.2], for every $t \ge 0$,

$$\mathbf{Pr}\left[\sum_{i\in[m]} a_i X_i^2 \ge t\right] \le \exp\left(-c\min\left(\frac{t^2}{\sum_{i\in[m]} a_i^4}, \frac{t}{\max_i a_i^2}\right)\right) \ , \tag{6}$$

where $c > 0$ is a universal constant.

We need to choose a value of t such that the right-hand side of (6) is upper bounded by $10^{-3}12^{-d}$. Given our assumptions on $a_1,\dots,a_m$, it is sufficient to choose t such that $t = \Omega\left(\sqrt{dK_2}\right)$ and $t = \Omega\left(dK_1\right)$. □

Proposition A.6. *Let $\alpha \in (0,1)$. Let $X_1,\dots,X_m$ be m random variables drawn i.i.d. from $\mathcal{N}(0,1)$. With probability at least $1-10^{-3}12^{-d}$, it holds that:*

$$\max_{L\subseteq[m]:|L|=\alpha m} \sum_{i\in L} X_i^2 = O(m\alpha\log(1/\alpha) + d) \ .$$

Proof. We define the threshold function

$$h_r(z) = \begin{cases} 0, & z \le r \\ z, & z > r \end{cases}$$

with $r = 8\ln(1/\alpha)$. Since $z \le r + h_r(z)$ for all $z > 0$,

$$\max_{L\subseteq[m],|L|=\alpha m} \sum_{i\in L} X_i^2 \le \max_L \sum_{i\in L} r + \max_L \sum_{i\in L} h_r(X_i^2) \le m\alpha\cdot r + \sum_{i\in[m]} h_r(X_i^2) \ .$$

We will use Chernoff-bound like arguments to obtain a high-probability upper bound on $\sum_{i\in[m]} h_r(X_i^2)$. For any $c > 0$ and $t > 0$, we have:

$$\begin{aligned} \mathbf{Pr}\left[\sum_{i\in[m]} h_r(X_i^2) \ge t\right] &= \mathbf{Pr}\left[\exp\left(c\sum_{i\in[m]} h_r(X_i^2)\right) \ge \exp(c\cdot t)\right] \\ &\le e^{-ct}\,\mathbb{E}\left[\exp\left(c\sum_{i\in[m]} h_r(X_i^2)\right)\right] \\ &= e^{-ct}\prod_{i\in[m]} \mathbb{E}\big[\exp\big(c\cdot h_r(X_i^2)\big)\big] \ , \end{aligned} \tag{7}$$

where the inequality follows from Markov's inequality.

Thus, it is sufficient to upper bound $\mathbb{E}\big[\exp(c \cdot h_r(X_i^2))\big]$. For any $c < 1/2$, we have

$$\begin{aligned}
\mathbb{E}\big[\exp\big(c \cdot h_r(X_i^2)\big)\big] &= 1 \cdot \mathbf{Pr}\big[h_r(X_i^2) = 0\big] + \frac{1}{\sqrt{2\pi}} \int_{\sqrt{r}}^{\infty} e^{cx^2} e^{-x^2/2} dx \\
&\leq 1 + \frac{1}{\sqrt{2\pi}\sqrt{1-2c}} \int_{\sqrt{r(1-2c)}}^{\infty} e^{-y^2/2} dy \\
&= 1 + \frac{1}{\sqrt{1-2c}} \mathbf{Pr}\Big[X_i \geq \sqrt{r(1-2c)}\Big] \\
&\leq 1 + \frac{1}{\sqrt{1-2c}} \exp\big(-r(\tfrac{1}{2} - c)\big) \ ,
\end{aligned}$$

where the second inequality is obtained by substituting $x\sqrt{1-2c} = y$ in the integral, and the last inequality uses the Gaussian tail bound that $\mathbf{Pr}[X_i \geq z] \leq e^{-z^2/2}$ for all $z > 0$. Recall that $r = 8\ln(1/\alpha)$. We set $c = 1/4$, so that $\exp(-r/4) = \alpha^2$. Thus, we have that:

$$\mathbb{E}\left[\exp\left(\frac{1}{4} \cdot h_r(X_i^2)\right)\right] \leq 1 + \sqrt{2}\alpha^2 \leq e^{\sqrt{2}\alpha^2} \tag{8}$$

We substitute the upper bound (8) into (7) and obtain that

$$\mathbf{Pr}\left[\sum_{i \in [m]} h_r(X_i^2) \geq t\right] \leq \exp\left(\frac{-t}{4} + \sqrt{2}m\alpha^2\right) \ .$$

We can choose $t = 4\sqrt{2}m\alpha^2 + \Omega(d)$, and then conclude that with probability at least $1 - 10^{-3}12^{-d}$,

$$\max_{L \subseteq [m] : |L| = \alpha m} \leq m\alpha r + \sum_{i \in [m]} h_r(X_i^2) = O(m\alpha r + m\alpha^2 + d) = O(m\alpha \log(1/\alpha) + d) \ .$$

□

Lemma A.1. *Let $x \in \mathbb{R}^d$ be the ground-truth vector. Fix $\delta > 0$. There exists constants $\overline{\epsilon}(\delta)$ and $c(\delta)$ such that if we let $0 < \epsilon \leq \overline{\epsilon}(\delta)$, and we are given an ϵ-corrupted set of $n = \widetilde{\Omega}(c(\delta)d)$ samples $(a_i, y_i)_{i \in [n]}$, with probability at least 0.99, for all $w \in \Delta_{n,2\epsilon}$,*

$$\left\|Y_{G,w} - (I + 2xx^\top)\right\|_2 \leq \delta \ ,$$

where G is the set of indices of the remaining good samples and $Y_{G,w} = \sum_{i \in G} w_i y_i a_i a_i^\top$.

Proof of Lemma A.1. We recall the definition of $Y_{G,w} = \sum_{i \in G} w_i y_i a_i a_i^\top$. Let $\ell \leq \epsilon \cdot n$ and let $\{(a_{n+i}, y_{n+i})\}_{i=1}^{\ell}$ be the set of samples that were removed by the ϵ-corruption adversary. Let $G' = G \cup \{n+1, \ldots, n+\ell\}$, $n' = n + \ell$, and $\epsilon' = \epsilon/(1+\epsilon)$. Note that without loss of generality, we can assume that $|G| = (1-\epsilon)n$ and $|G'| = (1-\epsilon')n' = n$.

We define a mapping $\sigma : \Delta_{n,2\epsilon} \to \Delta_{n',3\epsilon'}$ such that

$$\sigma(w)_i = \begin{cases} w_i & i \in [n] \\ 0 & \text{otherwise} \end{cases} . \tag{9}$$

In other words, all the weights are the same for the samples with index in the set $[n]$, and are equal to 0 for the samples removed by the adversary. We can verify that $\sigma(w) \in \Delta_{n',3\epsilon'}$ for all $w \in \Delta_{n,2\epsilon}$ since $\sigma(w)_i \leq w_i \leq 1/(1-2\epsilon)n = 1/(1-3\epsilon')n'$ for all $i \in [n']$, and $\|\sigma(w)\|_1 = \|w\|_1 = 1$. Furthermore, we have $Y_{G,w} = Y_{G',\sigma(w)}$ for all $w \in \Delta_{n,2\epsilon}$. We denote with $w^* \in \Delta_{n',3\epsilon'}$ the desired uniform weighting of the samples with index in G', i.e., $w_i^* = \frac{1}{(1-\epsilon')n'} \mathbb{1}_{i \in G'}$.

By triangle inequality, for any $w \in \Delta_{n,2\epsilon}$, it holds

$$\left\|Y_{G',\sigma(w)} - (I + 2xx^\top)\right\|_2 \leq \left\|Y_{G',w^*} - (I + 2xx^\top)\right\|_2 + \left\|Y_{G',\sigma(w)-w^*}\right\|_2 , \tag{10}$$

Thus, it suffices to show both $\left\|Y_{G',w^*} - (I + 2xx^\top)\right\|_2 \le \delta/2$ and $\left\|Y_{G',\sigma(w)-w^*}\right\|_2 \le \delta/2$. We upper bound the first term. By using the definition of w^*, note that

$$\begin{aligned}\left\|Y_{G',\sigma(w)} - (I + 2xx^\top)\right\|_2 &= \left\|\sum_{i\in G'} w_i^* y_i a_i a_i^\top - (I + 2xx^\top)\right\|_2 \\ &= \left\|\sum_{i\in G'} \frac{1}{|G'|} y_i a_i a_i^\top - (I + 2xx^\top)\right\|_2 .\end{aligned}$$

Since $\mathbb{E}\left[y_i a_i a_i^\top\right] = I + 2xx^\top$ for any $i \in G'$, we can use a concentration inequality to upper bound this term. By Lemma A.3, as long as $n \ge C(\delta/2) \cdot d \log d$, with probability at least 0.995, we have

$$\left\|Y_{G',w^*} - (I + 2xx^\top)\right\|_2 \le \delta/2 . \tag{11}$$

It remains to show a high-probability upper bound to the second term $\left\|Y_{G',w^*-\sigma(w)}\right\|_2 \le \delta/2$ that holds for any $w \in \Delta_{w,2\epsilon}$. To achieve this goal, we will provide a high-probability upper bound to the following quantity:

$$J = \sup_{w\in\Delta_{n,2\epsilon}} \left\|Y_{G',w^*-\sigma(w)}\right\|_2 = \sup_{w\in\Delta_{n,2\epsilon}} \left\|\sum_{i\in G'} (w_i^* - \sigma(w)_i) y_i a_i a_i^\top\right\|_2 .$$

Note that for every i, the matrix $y_i a_i a_i^\top$ is positive semidefinite since $y_i \ge 0$. Thus, it holds that:

$$J \le \sup_{w\in\Delta_{n,2\epsilon}} \left\|\sum_{i\in G'} |w_i^* - \sigma(w)_i| y_i a_i a_i^\top\right\|_2 .$$

For any $w \in \Delta_{n,2\epsilon}$ and any $i \in [n']$, it is easy to see that $0 \le |w_i^* - \sigma(w)_i| \le \frac{1}{(1-2\epsilon)n}$. Additionally the weighting w^* and $\sigma(w)$ cannot be too different. In particular, we can show the following upper bound:

$$\sum_{i\in G'} |w_i^* - \sigma(w)_i| \le \sum_{i=1}^{n'} |w_i^* - \sigma(w)_i| \le \sup_{w,w'\in\Delta_{n',3\epsilon'}} \sum_{i=1}^{n'} |w_i - w_i'| . \tag{12}$$

We observe that $\Delta_{n',3\epsilon'}$ can be seen as the convex combination of all possible uniform weighting over subsets of $n'(1 - 3\epsilon')$ samples. Thus, the maximum distance will be between two points of the convex hull, and we can upper bound (12) as:

$$\sum_{i\in G'} |w_i^* - \sigma(w)_i| \le \sup_{w,w'\in\Delta_{n',3\epsilon'}} \sum_{i=1}^{n'} |w_i - w_i'| \le \frac{6\epsilon' n}{n'(1 - 3\epsilon')} \le 6\epsilon . \tag{13}$$

Consider the family of weights defined as $\Gamma = \left\{\beta \in \mathbb{R}^{n'} : \sum_i \beta_i \le 6\epsilon \text{ and } 0 \le \beta_i \le \frac{1}{(1-2\epsilon)n}\right\}$. By the discussion above, we have that

$$J \le \sup_{w\in\Delta_{n,2\epsilon}} \left\|\sum_{i\in G'} |w_i^* - \sigma(w)_i| y_i a_i a_i^\top\right\|_2 \le \sup_{\beta\in\Gamma} \left\|\sum_{i\in G'} \beta_i y_i a_i a_i^\top\right\|_2 . \tag{14}$$

Since the map $\beta \mapsto \left\|\sum_{i\in G'} \beta_i y_i a_i a_i^\top\right\|_2$ is convex with respect to β, and Γ is a convex set, the supremum over $\beta \in \Gamma$ of (14) is achieved at one of the extreme points of Γ. Thus, it holds:

$$J \le \sup_{\beta\in\Gamma} \left\|\sum_{i\in G'} \beta_i y_i a_i a_i^\top\right\|_2 \le \frac{1}{(1 - 2\epsilon)n} \max_{L\subseteq G', |L|=6\epsilon n} \left\|\sum_{i\in L} y_i a_i a_i^\top\right\|_2 .$$

For any vector $v \in \mathbb{S}^{d-1}$ in the unit sphere, let

$$J(v) = \max_{L\subseteq G', |L|=6\epsilon n} \sum_{i\in L} y_i (v_i^\top a_i)^2 ,$$

and note that $J \le \frac{1}{(1-2\epsilon)n} \sup_{v\in\mathbb{S}^{d-1}} J(v)$.

Without loss of generality, assume that $x = e_1$, where $e_1 \in \mathbb{R}^d$ is the first canonical vector. Given any vector $u \in \mathbb{R}^d$, we will denote with u_1 the first coordinate, and with $\widetilde{u} \in \mathbb{R}^{d-1}$ the remaining $d-1$ coordinates, i.e., $u = (u_1, \widetilde{u})$. Assuming $x = e_1$, we have that $y_i = (x^\top a_i)^2 = a_{i,1}^2$ for any $i \in G'$. Let $H = \{i \in G' : |a_{i,1}| \geq 4\ln^2(2/\epsilon)\}$. We consider a set $L \subseteq G'$ that always contains H and then picks $6\epsilon n$ additional elements. In particular, it holds:

$$J(v) = \max_{L \subseteq G', |L| = 6\epsilon n} \sum_{i \in L} y_i (a_i^\top v_i)^2 \leq \left[\sum_{i \in H} y_i (a_i^\top v_i)^2 + \max_{L \subseteq G' \setminus H, |L| = 6\epsilon n} \sum_{i \in L} y_i (a_i^\top v_i)^2 \right] \ . \tag{15}$$

The first term of the right-hand side of (15) can be rewritten as follows:

$$\sum_{i \in H} y_i (a_i^\top v)^2 = \sum_{i \in H} a_{i,1}^2 \left(\sum_{j=1}^{d} a_{i,j} v_j \right)^2 \leq 2 \sum_{i \in H} \left[a_{i,1}^4 v_{i,1}^2 + a_{i,1}^2 (\widetilde{a}_i^\top \widetilde{v})^2 \right] \ . \tag{16}$$

For the second term of the right-hand side of (15), note that for any $i \in G' \setminus H$, it holds that $y_i = a_{i,1}^2 < 16\ln^4(2/\epsilon)$ due to the definition of H. Thus, we have that:

$$\begin{aligned} \max_{L \subseteq G' \setminus H, |L| = 6\epsilon n} \sum_{i \in L} y_i (a_i^\top v_i)^2 &\leq 16\ln^4(2/\epsilon) \max_{L \subseteq G' \setminus H, |L| = 6\epsilon n} \sum_{i \in L} (a_i^\top v_i)^2 && (17) \\ &\leq 32\ln^4(2/\epsilon) \max_{L \subseteq G' \setminus H, |L| = 6\epsilon n} \sum_{i \in L} \left[a_i^2 v_1^2 + (\widetilde{v}^\top \widetilde{a}_i)^2 \right] \ . && (18) \end{aligned}$$

Also, using the definition of H, note that for any $L \subseteq G' \setminus H$ with $|L| = 6\epsilon n$, we have that :

$$\sum_{i \in L} a_{i,1}^2 v_1^2 \leq \sum_{i \in L} a_{i,1}^2 = O\big(n\epsilon \ln^4(2/\epsilon)\big) \ . \tag{19}$$

By combining (16), (17), and (19) with (15), we obtain that:

$$J(v) = O\left(n\epsilon \log^8(1/\epsilon) + \sum_{i \in H} a_{i,1}^4 + \sum_{i \in H} a_{i,1}^2 (\widetilde{a}_i^\top \widetilde{v})^2 + \max_{L \subseteq G' \setminus H, |L| = 6\epsilon n} \sum_{i \in L} (\widetilde{v}^\top \widetilde{a}_i)^2 \right) \ .$$

Let E_1 be the event of Proposition A.4 with $\alpha = \epsilon$ and $n = m$. That is, with probability at least $1 - 10^{-3}$, we have that $|H| = O(\epsilon n)$, $\sum_{i \in H} a_{i,1}^4 = O\big(n\epsilon \log^8(1/\epsilon)\big)$ and $\max_i a_{i,1}^2 = O(\log n)$. For the remaining of this proof, assume that E_1 is true, and thus:

$$J(v) = O\left(n\epsilon \log^8(1/\epsilon) + \sum_{i \in H} a_{i,1}^2 (\widetilde{a}_i^\top \widetilde{v})^2 + \max_{L \subseteq G' \setminus H, |L| = 6\epsilon n} \sum_{i \in L} (\widetilde{v}^\top \widetilde{a}_i)^2 \right) \ . \tag{20}$$

Denote with $\overline{J}(v) = \sum_{i \in H} (\widetilde{a}_i^\top \widetilde{v})^2 + \max_{L \subseteq G' \setminus H, |L| = 6\epsilon n} \sum_{i \in L} (\widetilde{v}^\top \widetilde{a}_i)^2$ the last two terms of the right-hand side of (20). We will upper bound each term of $\overline{J}$ individually. First, observe that the random variables $Z_i = \widetilde{a}_i^\top \widetilde{v} / \|\widetilde{v}\|_2$ for $i \in G'$ are independent standard normal random variables. Let E_2^v be the event of Proposition A.5 for the random variables $\{Z_i : i \in H\}$ and weights $\{a_{i,1}^2 : i \in H\}$. That is, with probability at least $1 - 10^{-3} 12^{-d}$, it holds that $\sum_{i \in H} a_{i,1}^2 Z_i^2 = O\Big(\epsilon^{1/2} \sqrt{dn} \log^4(1/\epsilon) + d \log n\Big)$. For the second term of $\overline{J}$, we can invoke Proposition A.6 over the random variables $\{Z_i : i \in G' \setminus H\}$ with $\alpha = 12\epsilon$ and $m = |G' \setminus H| \geq n/2$ (if $\epsilon < 1/2$). Let E_3^v be the event of this proposition, that is, with probability at least $1 - 10^{-3} 12^{-d}$, it holds that:

$$\max_{L \subseteq G' \setminus H, |L| = 6\epsilon n} \sum_{i \in L} (\widetilde{v}^\top \widetilde{a}_i)^2 = O(\epsilon n \log(1/\epsilon) + d) \ .$$

By taking a union bound of the events E_2^v and E_3^v, we have that:

$$\begin{gathered} \text{given } v, \text{ with probability at least } 1 - 2 \cdot 10^{-3} 12^{-d}, \\ \overline{J}(v) = O\Big(\epsilon n \log(1/\epsilon) + \epsilon^{1/2} \sqrt{dn} \log^4(1/\epsilon) + d \log n \Big) \ . \end{gathered} \tag{21}$$

Consider a $1/4$-net $\mathcal{N}$ of $\mathbb{S}^{d-1}$, where $|\mathcal{N}| = O(12^d)$. Note that for any $v \in \mathbb{S}^{d-1}$, it holds $\sup_{v \in \mathbb{S}^{d-1}} \overline{J}(v) \le 2 \sup_{v \in \mathcal{N}} \overline{J}(v)$. Thus, by taking a union bound over the event described in (21) for all $v \in \mathcal{N}$, we obtain that with probability at least $1 - 2 \cdot 10^{-3}$, we have:

$$\sup_{v \in \mathcal{S}^{d-1}} \overline{J}(v) = O\Big(\epsilon n \log(1/\epsilon) + d \log n + \sqrt{dn\epsilon} \log^2(1/\epsilon)\Big) \tag{22}$$

We finally combine (22) and (20) to conclude that with probability at least $1 - 0.995$, it holds that

$$J \le O\left(\frac{1}{n}\left[d \log n + \epsilon^{1/2}\sqrt{dn} \log^4(1/\epsilon) + n\sqrt{\epsilon} \log^8(1/\epsilon)\right]\right) .$$

We can pick a sufficiently small ϵ (depending only on δ) and $n \ge c(\delta) d \log d$ for a sufficiently large constant $c(\delta)$ so that with probability at least $1 - 0.995$, it holds that $J \le \delta/2$. Utilizing this result along with (11) to upper bound (10) yields the desired statement. □

B Omitted Proofs in Section 5

Lemma 5.1. *Let $x \in \mathbb{R}^d$ be the ground-truth vector. Let $\mathcal{D}_z$ be the distribution of gradients at z as defined in Equation* (2)*. For any z with $\|z - x\|_2 \le 1$, the covariance matrix Σ_z of $\mathcal{D}_z$ satisfies*

$$\Sigma_z \preceq O\Big(\|z - x\|_2^2\Big) I .$$

Proof of Lemma 5.1. Recall that $g \sim \mathcal{D}_z$ is distributed as

$$g = \frac{\partial}{\partial z}\left[\Big(\langle a, z\rangle^2 - \langle a, x\rangle^2\Big)^2\right] = 4\left(\langle a, z\rangle^2 - \langle a, x\rangle^2\right)\langle a, z\rangle\, a \quad \text{where} \quad a \sim \mathcal{N}(0, I) .$$

Let $\mu_z = \mathbb{E}_{g \sim \mathcal{D}_z}[g]$. We have

$$0 \preceq \Sigma_z = \mathop{\mathbb{E}}_{g \sim \mathcal{D}_z}\left[g g^\top\right] - \mu_z \mu_z^\top \preceq \mathop{\mathbb{E}}_{g \sim \mathcal{D}_z}\left[g g^\top\right] .$$

Consequently, it suffices to upper bound the spectral norm of $\mathbb{E}_{g \sim \mathcal{D}_z}\left[g g^\top\right]$. Let $h = z - x$.

$$\begin{aligned}
\|\Sigma_z\|_2 &\le \left\| \mathop{\mathbb{E}}_{g \sim \mathcal{D}_z}\left[g g^\top\right] \right\|_2 \\
&= \max_{\|v\|_2 = 1} v^\top \mathop{\mathbb{E}}_{g \sim \mathcal{D}_z}\left[g g^\top\right] v \\
&= \max_{\|v\|_2 = 1} \mathop{\mathbb{E}}_{g \sim \mathcal{D}_z}\left[\langle g, v\rangle^2\right] \\
&= 16 \max_{\|v\|_2 = 1} \mathop{\mathbb{E}}_{a \sim \mathcal{N}(0, I)}\left[\Big(\left(\langle a, z\rangle^2 - \langle a, x\rangle^2\right)\langle a, z\rangle \langle a, v\rangle\Big)^2\right] \\
&= 16 \max_{\|v\|_2 = 1} \mathop{\mathbb{E}}_{a \sim \mathcal{N}(0, I)}\left[\Big(\left(\langle a, h\rangle^2 + 2\langle a, x\rangle \langle a, h\rangle\right)\langle a, x + h\rangle \langle a, v\rangle\Big)^2\right] \\
&= 16 \max_{\|v\|_2 = 1} \mathop{\mathbb{E}}_{a \sim \mathcal{N}(0, I)}\left[\langle a, h\rangle^2 \langle a, 2x + h\rangle^2 \langle a, x + h\rangle^2 \langle a, v\rangle^2\right] \\
&\le 16 \max_{\|v\|_2 = 1} \left(\mathop{\mathbb{E}}_{a}\Big[\langle a, h\rangle^8\Big] \mathop{\mathbb{E}}_{a}\Big[\langle a, 2x + h\rangle^8\Big] \mathop{\mathbb{E}}_{a}\Big[\langle a, x + h\rangle^8\Big] \mathop{\mathbb{E}}_{a}\Big[\langle a, v\rangle^8\Big]\right)^{1/4} \\
&= 16 \max_{\|v\|_2 = 1} \left(105^4 \|h\|_2^8 \|2x + h\|_2^8 \|x + h\|_2^8 \|v\|_2^8\right)^{1/4} \\
&= (16 \cdot 105) \|h\|_2^2 \|2x + h\|_2^2 \|x + h\|_2^2 \\
&= O(\|h\|_2^2) .
\end{aligned}$$

The last inequality follows from the Cauchy-Schwarz inequality. The last step uses the fact that $\|2x + h\|_2 = O(1)$ and $\|x + h\|_2 = O(1)$, which follows from $\|x\|_2 = 1$ and $\|h\|_2 \le 1$. □

Lemma 5.3. *Let $x \in \mathbb{R}^d$ be the ground-truth vector. Suppose at iteration t of Algorithm 2, the current solution z_t satisfies $\|z_t - x\|_2 \le \frac{1}{8}$. Let μ_{z_t} denote the expected true gradient at z_t defined in Equation* (3). *Suppose the estimated gradient $\widehat{\mu}_{z_t} \in \mathbb{R}^d$ satisfies $\|\widehat{\mu}_{z_t} - \mu_{z_t}\|_2 \le c\,\|z_t - x\|_2$ for $c = 4$. Then, we have*

$$\|z_{t+1} - x\|_2^2 \le 0.99\,\|z_t - x\|_2^2 \ .$$

Proof of Lemma 5.3. Recall that $g \sim \mathcal{D}_z$ is distributed as

$$g = \frac{\partial}{\partial z}\left[\left(\langle a, z\rangle^2 - \langle a, x\rangle^2\right)^2\right] = 4\left(\langle a, z\rangle^2 - \langle a, x\rangle^2\right)\langle a, z\rangle\, a \quad \text{ where } \quad a \sim \mathcal{N}(0, I) \ .$$

We can compute the mean μ_z of $\mathcal{D}_z$ using moments of Gaussian:

$$\mu_z = \mathop{\mathbb{E}}_{g \sim \mathcal{D}_z}[g] = \left(12\,\|z\|_2^2 - 4\,\|x\|_2^2\right) z - 8\,\langle x, z\rangle\, x \ .$$

Consider one step of gradient descent in Algorithm 2: $z_{t+1} = z_t - \eta\widehat{\mu}_{z_t}$, where $\widehat{\mu}_{z_t}$ is close to μ_z. We have

$$\|z_{t+1} - x\|_2^2 = \|z_t - \eta\widehat{\mu}_{z_t} - x\|_2^2 = \|z_t - x\|_2^2 - 2\eta\,\langle\widehat{\mu}_{z_t}, z_t - x\rangle + \eta^2\,\langle\widehat{\mu}_{z_t}, \widehat{\mu}_{z_t}\rangle$$

To prove convergence, we need to lower bound $\langle\widehat{\mu}_{z_t}, z_t - x\rangle$ and upper bound $\langle\widehat{\mu}_{z_t}, \widehat{\mu}_{z_t}\rangle$.

Let $z = z_t$ and $h = z - x$. Substituting $z = x + h$ in the expression for μ_z, we get:

$$\begin{aligned}\mu_z &= \left(12\,\|x + h\|_2^2 - 4\,\|x\|_2^2\right)(x + h) - 8\,\langle x, x + h\rangle\, x \\ &= \left(16\,\langle x, h\rangle + 12\,\|h\|_2^2\right) x + \left(8\,\|x\|_2^2 + 24\,\langle x, h\rangle + 12\,\|h\|_2^2\right) h \ .\end{aligned}$$

We will be using the assumptions of this lemma: $\|x\|_2 = 1$, $\|h\|_2 \le \frac{1}{8}$, and $\|\widehat{\mu}_z - \mu_z\|_2 \le c\,\|h\|_2$.

First we lower bound $\langle\widehat{\mu}_z, h\rangle$.

$$\begin{aligned}\langle\widehat{\mu}_z, h\rangle &= \langle\mu_z, h\rangle + \langle\widehat{\mu}_z - \mu_z, h\rangle \\ &= 16\,\langle x, h\rangle^2 + 36\,\langle x, h\rangle\,\|h\|_2^2 + 8\,\|x\|_2^2\,\|h\|_2^2 + 12\,\|h\|_2^4 + \langle\widehat{\mu}_z - \mu_z, h\rangle \\ &\ge -\tfrac{81}{4}\,\|h\|_2^4 + 8\,\|x\|_2^2\,\|h\|_2^2 + 12\,\|h\|_2^4 - c\,\|h\|_2^2 \\ &\ge (-\tfrac{81}{256} + 8 + \tfrac{12}{64} - c)\,\|h\|_2^2 \\ &\ge (7.5 - c)\,\|h\|_2^2 \ .\end{aligned}$$

The first inequality uses the fact that $16\,\langle x, h\rangle^2 + 36\,\langle x, h\rangle\,\|h\|_2^2$ is a quadratic function of $\langle x, h\rangle$, which has minimum value $-\frac{81}{4}\,\|h\|_2^4$ for all $\langle x, h\rangle \in \mathbb{R}$.

Next we upper bound $\|\widehat{\mu}_z\|_2$ using the triangle inequality.

$$\begin{aligned}\|\widehat{\mu}_z\|_2 &\le \|\mu_z\|_2 + \|\widehat{\mu}_z - \mu_z\|_2 \\ &\le \left(16\,\langle x, h\rangle + 12\,\|h\|_2^2\right)\|x\|_2 + \left(8\,\|x\|_2^2 + 24\,\langle x, h\rangle + 12\,\|h\|_2^2\right)\|h\|_2 + c\,\|h\|_2 \\ &\le \left(16 + \tfrac{12}{8} + 8 + \tfrac{24}{8} + \tfrac{12}{64} + c\right)\|h\|_2 \\ &\le (29 + c)\,\|h\|_2 \ .\end{aligned}$$

Putting everything together, we have

$$\begin{aligned}\|z_{t+1} - x\|_2^2 &= \|z_t - x\|_2^2 - 2\eta\,\langle\widehat{\mu}_{z_t}, z_t - x\rangle + \eta^2\,\langle\widehat{\mu}_{z_t}, \widehat{\mu}_{z_t}\rangle \\ &\le \left[1 - 2(7.5 - c)\eta + (29 + c)^2\eta^2\right]\|z_t - x\|_2^2 \ .\end{aligned}$$

Choosing $c = 4$ and $\eta = 1/300$ gives that $\|z_{t+1} - x\|_2^2 \le 0.99\,\|z_t - x\|_2^2$. □

C A Counter-Example for Previous Algorithms

In this work, we consider a general setting that allows adversarial corruption in both the measurement vectors a_i's and the intensity measurements y_i's. Prior work in robust phase retrieval has addressed a special case where the adversarial corruption is restricted to the y_i's, still assuming that the measuring vectors a_i's are independently sampled from the Gaussian distribution [27, 51]. In this section, we construct a counter-example demonstrating the failure of algorithms developed for the restricted corruption setting when applied to the more general setting considered in our paper.

The Median Truncated Wirtinger Flow Algorithm [51] is an algorithm to address the robust phase retrieval problem with adversarial corruption limited to the y_i's. The algorithm first initializes $z^{(0)}$ using the spectral method. Let $\alpha \geq 3$. In particular, $z^{(0)}$ is computed as the top eigenvector of the empirical matrix $Y := \frac{1}{m}\sum_{i=1}^{m} y_i a_i a_i^\top \mathbb{1}_{|y_i| \leq \alpha^2 \operatorname{med}(\{y_i\}_{i=1}^m)}$ that only uses a truncated set of samples, where the threshold is determined by $\operatorname{med}(\{y_i\}_{i=1}^m)$, the median over all y_i's. The analysis of the algorithm relies on the fact that as long as the fraction of outliers is not too large and the sample complexity is large enough, the initialization is guaranteed to be within a small neighborhood of the ground truth.

We show that this initialization can fail to remove the distortion introduced by the adversarial if corruption is allowed for both a_i's and y_i's. Let $x \in \mathbb{S}^{d-1}$ be the ground truth unit vector. We construct an ϵ-corruption adversary that can manipulate the top eigenvector of the empirical covariance matrix $Y = \sum_{i=1}^{n} y_i a_i a_i^\top$, even when all y_i's are accurately calculated as $y_i = (a_i^\top x)^2$.

Let $u \in \mathbb{S}^{d-1}$ be a unit vector such that $x^\top u = 0$. Suppose the adversary changes 1% of the a_i's to $a_i = \sqrt{d - 1/25} \cdot u + (1/5) \cdot x$, and suppose that all the y_i's are accurate. In particular, the length of the corrupted a_i's is comparable to the length of a random Gaussian vector, and the corresponding intensity measurements satisfy $y_i = (a_i^\top x)^2 = 1/25$. Let $z = (a^\top x)^2$ for a random vector $a \sim \mathcal{N}(0, I)$. By direct computation, note that $\mathbf{Pr}[z \geq 0.2] \geq 0.6$. Thus, with high-constant probability, the median-truncated initialization in [51] is not able to filter out any of those samples. However, after the adversarial corruption, the top eigenvector of $\mathbb{E}\big[\sum_{i=1}^{n} y_i a_i a_i^\top\big] \approx O(d)uu^\top + O(\sqrt{d})(ux^\top + xu^\top) + O(1)(I + 2xx^\top)$ will be manipulated to u, which is far from the ground truth x.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

   Answer: [Yes]

   Justification: We provide formal statements on the guarantee of our algorithm, and state our main theorem. The rest of the paper is devoted in proving this theorem.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: Limitations and questions for future work are discussed in the conclusion.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Sections 3-5 and the first two sections of the appendix are devoted to the proof of our main result.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [NA]

   Justification: The paper does not contain experimental results.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [NA]

   Justification: The paper does not include experiments requiring code.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [NA]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [NA]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research follows the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper is theoretical and poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: The paper is theoretical and does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing experiments.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.