Robust Gradient Descent for Phase Retrieval

Alex Buna **Patrick Rebeschini**
University of Oxford

Abstract

Recent progress in robust statistical learning has mainly tackled convex problems, like mean estimation or linear regression, with non-convex challenges receiving less attention. Phase retrieval exemplifies such a non-convex problem, requiring the recovery of a signal from only the magnitudes of its linear measurements, without phase (sign) information. While several non-convex methods, especially those involving the Wirtinger Flow algorithm, have been proposed for noiseless or mild noise settings, developing solutions for heavy-tailed noise and adversarial corruption remains an open challenge. In this paper, we investigate an approach that leverages robust gradient descent techniques to improve the Wirtinger Flow algorithm's ability to simultaneously cope with fourth moment bounded noise and adversarial contamination in both the inputs (covariates) and outputs (responses). We address two scenarios: known zero-mean noise and completely unknown noise. For the latter, we propose a preprocessing step that alters the problem into a new format that does not fit traditional phase retrieval approaches but can still be resolved with a tailored version of the algorithm for the zero-mean noise context.

1 INTRODUCTION

While the foundations of asymptotic robust statistics have been laid in the previous century by the seminal works of Huber (1964) and Hampel (1974), advancements in modern computing naturally led to the surge of a rigorous non-asymptotic robust statistical theory. In recent years, significant breakthroughs have been made in developing methods that are statistically and computationally efficient for robust estimation and regression. These methods have been derived by both the statistics and computer science community, typically under different notions of robustness. Statisticians have focused on models with i.i.d. data that exhibit heavy tails, frequently employing M-estimators or the renowned median-of-means framework (Catoni, 2012; Minsker, 2015). Concurrently, computer scientists have looked into models of contamination where a portion of the data is altered by an adversary, and proposed dimension-halving and stability-based algorithms (Lai et al., 2016; Diakonikolas et al., 2019a). These efforts have produced optimal estimators for the fundamental problems of mean and covariance estimation (Tukey, 1975; Lugosi and Mendelson, 2019; Oliveira and Rico, 2022; Abdalla and Zhivotovskiy, 2024). Later, efficient techniques for mean estimation (Hopkins, 2020; Diakonikolas et al., 2020; Hopkins et al., 2020) and linear regression (Prasad et al., 2020; Pensia et al., 2020) were designed, the latter aiming to recover a signal from noisy linear measurements.

Moving away from linear models, there is limited literature on robustness results for noisy *quadratic* measurements, a challenge commonly referred to as phase retrieval. The goal of this non-convex and ill-posed problem is to recover an unknown n-dimensional signal $\boldsymbol{x}^* \in \mathbb{R}^n$ from m noisy measurements $\{(\boldsymbol{a}_j, y_j)\}_{j=1}^m$ generated as $y_j \approx (\boldsymbol{a}_j^\top \boldsymbol{x}^*)^2$, an ε fraction of the sample having been contaminated by an adversary. As this problem subsumes solving quadratic systems of equations, it has applications in engineering sciences, such as X-ray cristalography and optics, to name a few (Drenth, 2007; Shechtman et al., 2014), and it has been studied from different perspectives within the statistical learning community, e.g. in the context of multi-armed bandits (Lattimore and Hao, 2021) and deep learning (Hand et al., 2018; Chen et al., 2022). We formally describe the model in Section 2.

Phase retrieval has been approached by the statistical learning community using methods that can be broadly split into two categories: convex relaxations and Wirtinger Flow. We focus on the latter, as it

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

generally enjoys better statistical and computational guarantees, while also being more amenable to methods from robust statistics. A detailed exposition of convex relaxations or methods inspired by but significantly distinct from the Wirtinger Flow, such as (Wang et al., 2017), is beyond the scope of this paper and we redirect the reader to (Fannjiang and Strohmer, 2020). The Wirtinger Flow is a first-order method that aims at performing gradient descent with a spectral initialisatioin, or a modified version of it, on an empirical risk associated with the phase retrieval problem. The works of Chen and Candes (2015) and Zhang et al. (2018) describe a truncated Wirtinger Flow that is robust to outliers and bounded or deterministic noise in the responses (see the Supplementary Material for a more in-depth related work discussion and Table 1). However, no phase retrieval method proposed so far is resilient to heavy-tailed noise and adversarial contamination in both the covariates and the responses.

1.1 Our contributions

We design and analyse methodologies for the robust phase retrieval problem in two distinct situations: when the learner knows the noise has mean zero, and when the noise mean is unknown and potentially non-zero. Following a data preprocessing step (only needed in the latter situation), our methods are applicable in both scenarios and offer identical theoretical guarantees. The main framework consists of two separate procedures: first, a spectral initialisation that offers a good quality starting point for the second procedure, which is a descent-type algorithm that improves the quality of the estimate at each subsequent iteration.

For the spectral initialisation, we offer two alternatives. The first one uses computationally efficient (i.e. runtime nearly linear in input size) stable mean estimatiors and only assumes a bounded away from zero signal-to-noise ratio, but has sample size that scales as $m_0 = O(n^2 \log(n))$ and it restricts the contamination ε to be of order $1/n$ (Theorems 3.1 and B.1). Our second proposal employs covariance estimation techniques from the statistical literature (Oliveira and Rico, 2022; Abdalla and Zhivotovskiy, 2024). Consequently, it requires a fourth-moment bound on the noise and it operates with an improved $m_0 = O(n)$ samples and can accommodate a constant fraction ε of contamination (Theorems 3.2 and B.2).

Provided with a good initialisation point, we propose an iterative algorithm with constant stepsize that can accommodate constant contamination while requesting $\widetilde{m} = O(n \log(n))$ samples at each one of its T iterations (Theorems 3.3 and B.3, see also Table 1). Although the sample complexity (total number of sampled used) of the iterative procedure is $O(Tn \log(n))$, a tuning of $T = \widetilde{O}(1)$ (with hidden log factors independent of the dimension n) reduces the sample complexity to $O(n \log(n))$, c.f. Remark 3.3.2.

Focusing on the case when the noise mean is known to be zero, the high-level idea of our iterative algorithm is as follows: We start by considering the population risk under the squared loss $r(\boldsymbol{x}) = \mathbb{E}[((\boldsymbol{a}^\top \boldsymbol{x}^*)^2 - y)^2]/4$, associated to (1). In a ball $\mathcal{B}_\pm$ around the unknown minimisers $\pm\boldsymbol{x}^*$, the population risk is strongly-convex and smooth (Lemma 2.1). By standard results in convex optimisation, vanilla gradient descent on the population risk initialised inside $\mathcal{B}_\pm$ and run with an appropriately-chosen step size, is guaranteed to converge at a linear rate to one of $\pm\boldsymbol{x}^*$. However, gradients of the population risk are not available and we estimate them using robust gradient estimators (Definition 2.1.1). Three main challenges arise:

- The need for a robust spectral initialisation: both the center $\pm\boldsymbol{x}^*$ and the radius of $\mathcal{B}_\pm$ depend on $\pm\boldsymbol{x}^*$ and we need to guarantee that the first iterate falls within $\mathcal{B}_\pm$. Previous works (Candès and Li, 2014; Ma et al., 2019) achieve this by noting that $\pm\boldsymbol{x}^*$ is the principal eigenvector of $\mathbb{E}[y\boldsymbol{a}\boldsymbol{a}^\top]$. While this idea can be exploited in conjunction with mean estimators to give rise to a polynomial time algorithm, (see Procedure 1 with configuration `MeanEstStab` and Theorem 3.1), the resulting sample size and contamination level behave worse than what one would expect from the phase retrieval and robust statistics literatures. Instead, we identify a covariance matrix (of $y\boldsymbol{a}$) whose leading eigenvector is also $\pm\boldsymbol{x}^*$, and we propose using covariance estimators that, although not efficiently computable, gives better statistical guarantees (c.f. Procedure 1 with configuration `CovEst` and Theorem 3.2).

- Step-size tuning: it depends on the strong-convexity and smoothness parameters of the population risk in $\mathcal{B}_\pm$. However, these are again quantities that depend on the unknown $\|\boldsymbol{x}^*\|$. Instead, we use the norm of the initial iterate, which is guaranteed to be close to $\|\boldsymbol{x}^*\|$, to tune the learning rate.

- Ensuring iterates stay within $\mathcal{B}_\pm$: we use a generic *stable* mean estimator for the gradient estimation and the strict contraction property of vanilla gradient descent on strongly-convex and smooth functions. See Lemma 2.2 and Proposition 2.2.1.

None of the above-mentioned challenges were real concerns in previous works that use robust gradient descent, such as (Prasad et al., 2020; Liu et al., 2019), as they were applying it to functions that are *globally*

Table 1: Summary of the main results regarding the use of Wirtinger Flow (WF) for solving the phase retrieval (PR) problem. As detailed in Section 2, an adversary contaminates an ε fraction of samples from the noisy PR model $y = (\boldsymbol{a}^\top \boldsymbol{x}^*)^2 + z$, with noise variance σ^2. The goal of the learner is to find $\boldsymbol{x}$ that minimises the estimation error $\|\boldsymbol{x} \pm \boldsymbol{x}^*\| / \|\boldsymbol{x}^*\|$, decomposable into two errors of different types: the 'Statistical error' caused by the noise z and the contamination level ε; and the 'Optimisation error' of the iterative procedure used, in this case gradient descent (GD)/WF. All works use spectral initialisation (c.f. Procedure 1) followed by a GD in T steps (c.f. Algorithm 2) and we report in this table the statistical error only, as the optimisation error is $O(\exp(-T))$ in all works (including ours). In the third row, the dotted line (whenever present) reflects differences between the efficient mean-estimation based spectral initialisation we use (left of dotted line) and the robust GD we consider (right of dotted line).

Method	Contamination ε & noise z	Sample complexity	Statistical error
ℓ_2 loss, Vanilla GD with sprectral init. (Candes et al., 2015)	$\varepsilon = 0$ & $z = 0$	n	0 (no noise)
reshaped ℓ_2 loss *or* Poisson log-likelihood truncated WF with spectral init. (Zhang et al., 2018)	$\varepsilon \propto 1$ in y & bounded noise $\lvert z\rvert \le B \left\|\boldsymbol{x}^*\right\|^2$	$n \log(n)$	$\sqrt{B}(O(1) + \sqrt{\varepsilon})$ *or* $B(O(1) + \sqrt{\varepsilon})$
Spectral init based on mean estim. | ℓ_2 loss robust GD **Thms. 3.1 & 3.3**	$\varepsilon \propto n^{-1}$ | $\varepsilon \propto 1$ **in $\boldsymbol{a}$ and y & heavy-tailed z,** $\left\|\boldsymbol{x}^*\right\|^2 / \sigma > 0$	$n^2 \log(n)$ | $n \log(n)$	$\frac{\sigma}{\|\boldsymbol{x}^*\|^2}(O(1) + \sqrt{\varepsilon})$
ℓ_2 loss, robust GD with sprectral init. based on cov. estim. **Thms. 3.2 & 3.3**	$\varepsilon \propto 1$ **in $\boldsymbol{a}$ and y & heavy-tailed z,** $\mathbb{E}[z^4] < \infty$	$n \log(n)$	$\frac{\sigma}{\|\boldsymbol{x}^*\|^2}(O(1) + \sqrt{\varepsilon})$

strongly-convex and smooth and these parameters did not depend on the norm of the true signal.

To overcome the problem of not having information on the noise mean, we employ an extra data preprocessing step: we split the sample in two and 'subtract' from the points in the first half points from the second half. The new data, although coming from a model $\upsilon \approx \boldsymbol{b}^\top \boldsymbol{x}^* (\boldsymbol{x}^*)^\top \boldsymbol{c}$ that falls outside the scope of phase retrieval, has a zero mean noise. We note that in the rest of this paper, we present this preprocessing step as doubling the sample to avoid technicalities with odd-sized samples, but this does not change our results. More details can be found in Subsection 2.3.

1.2 Notation and organisation of the paper

For any positive integer k, we use $[k]$ to refer to the set $\{1, 2, \ldots, k\}$. We use lowercase bold-faced letters to denote vectors and uppercase letters to denote matrices. The vector with all entries equal to zero is denoted by $\mathbf{0}$. The vector $\boldsymbol{e}_i$ has 1 in the i'th coordinate and 0 everywhere else. For a vector $\boldsymbol{x}$, we denote by $\boldsymbol{x}_i$ its i'th entry and by $\|\boldsymbol{x}\|$ its Euclidean norm. We reserve the notation $A \succeq 0$ to mean that the square matrix A is positive semi-definite, $A \succeq B$ if $A - B \succeq 0$ and $A \preceq B$ if $B \succeq A$. The smallest and largest eigenvalue of a symmetric matrix A are denoted by $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$, respectively. The operator norm of a matrix A is denoted by $\|A\|_{\mathrm{op}}$, its Frobenius norm by $\|A\|_{\mathrm{F}}$. For a function $f : \mathbb{R}^d \to \mathbb{R}$, we write ∇f and $\nabla^2 f$ for its gradient and Hessian, respectively. $C, C_1, C_2, \ldots$ are absolute numerical constants. Finally, we use the big-O asymptotic notations: $f = O(g)$ if $f(x)/g(x)$ is upper-bounded by a constant for sufficiently large or small x (depending on the context), $f = \Omega(g)$ if $g = O(f)$, $f = \Theta(g)$ if both $f = O(g)$ and $g = O(g)$ hold, and $\widetilde{O}(\cdot)$ hides poly-logarithmic factors. For a scalar random variable X we will denote by $\mathrm{Var}(X)$ its variance. In case this is multidimensional, we will denote its covariance by $\mathrm{Cov}(X)$.

Section 2 introduces formally the model of robust phase retrieval and presents in more technical detail the main ideas employed for known mean zero noise. The concrete procedures and the main theorems for the known zero-mean noise can be found in Section 3. Finally, we present our conclusion in Section 4. All proofs, along with the unknown mean case main results are deferred to the Supplementary Material.

2 SETTING AND MAIN IDEAS

2.1 Robust phase retrieval

The setting of our problem and the made assumptions are as follows: For a fixed and unknown n-dimensional

vector $\boldsymbol{x}^* \in \mathbb{R}^n$, consider the statistical model

$$y = (\boldsymbol{a}^\top \boldsymbol{x}^*)^2 + z, \tag{1}$$

where the covariate $\boldsymbol{a} \in \mathbb{R}^n$ is distributed as a standard normal $\mathcal{N}(\mathbf{0}, I_n)$ and the noise has an *unknown* mean, *unknown* variance σ^2, unknown bounded central fourth moment $K_4^4 = \mathbb{E}[(z - \mathbb{E}[z])^4] < \infty$, and is independent of $\boldsymbol{a}$. A learning algorithm can request m samples. Then, data $S = \{(\boldsymbol{a}_j, y_j)\}_{j=1}^m$ is generated according to (1) and passed onto an adversary that inspects the sample S and returns to the learner a dataset $T \subset \mathbb{R}^n \times \mathbb{R}$ such that $|T| = m$ and $|T \cap S| \geq (1-\varepsilon)m$. The contamination level $\varepsilon \in (0, 1/2)$ is known to the learner.

The goal of a learning algorithm is to find, given access to the previously described process, some $\boldsymbol{x} \in \mathbb{R}^n$ that minimises the following notion of distance:

$$\operatorname{dist}(\boldsymbol{x}, \boldsymbol{x}^*) := \min\{\|\boldsymbol{x} - \boldsymbol{x}^*\|, \|\boldsymbol{x} + \boldsymbol{x}^*\|\}. \tag{2}$$

The distance is defined as in (2) because it is impossible to distinguish between $\boldsymbol{x}^*$ and $-\boldsymbol{x}^*$.

Assuming the covariates $\boldsymbol{a}$ are initially sampled from $\mathcal{N}(\mathbf{0}, I_n)$ is standard in the phase retrieval literature, see, for example, (Candes et al., 2015). Additionally, the assumption that the fourth moment of the noise is finite is, in fact, needed in only one of our results (Theorem 3.2) and has been made in many previous robust statistics works (for example, (Pensia et al., 2020; Oliveira and Rico, 2022)). The signal-to-noise ratio $\|\boldsymbol{x}^*\|^2/\sigma$ and its fourth-moment counterpart $\|\boldsymbol{x}^*\|^2/K_4$ are assumed to be a positive constant and in particular they do not depend on the dimension n. The model implicitly assumes that the algorithm can collect data at any moment. This is common across learning theory, for example in the definition of a PAC learning example oracle (Valiant, 1984).

2.2 Zero mean noise

For the rest of this subsection, we assume that $\mathbb{E}[z] = 0$ and the learner knows this.

2.2.1 Population risk local geometry

Consider the ℓ_2 population risk associated to (1):

$$r(\boldsymbol{x}) = \mathbb{E}\left[((\boldsymbol{a}^\top \boldsymbol{x})^2 - y)^2\right]/4, \tag{3}$$

where the expectation is taken over $(\boldsymbol{a}, y)$. Given that $\mathbb{E}[z] = 0$, it is the case that $\pm\boldsymbol{x}^*$ are amongst the minimisers of the population risk. As mentioned in subsection 1.1, our algorithm aims at simulating gradient descent on the population risk (3) to recover one of these minimisers. In turn, this will be achievable as a consequence of the local geometry. We recall some basic definitions from the convex analysis literature. We primarily work with twice differentiable functions.

Definition 2.0.1. *Let $\mathcal{W} \subseteq \mathbb{R}^n$ be a convex set. A twice differentiable function $f : \mathcal{W} \to \mathbb{R}$ is said to be* α-strongly convex *for some $\alpha > 0$ if $\nabla^2 f \succeq \alpha I_n$, and* β-smooth *for some $\beta > 0$ if $\nabla^2 f \preceq \beta I_n$.*

Lemma 2.1. *The population risk (3) is $\alpha := 4\|\boldsymbol{x}^*\|^2$-strongly convex and $\beta := 73\|\boldsymbol{x}^*\|^2/9$-smooth in a ball around $\boldsymbol{x}^*$ of radius $R := \|\boldsymbol{x}^*\|/9$. That is:*

$$4\|\boldsymbol{x}^*\|^2 I_n \preceq \nabla^2 f(\boldsymbol{x}) \preceq (73\|\boldsymbol{x}^*\|^2/9) I_n \quad ,$$

for all $\boldsymbol{x} \in \mathbb{R}^n$ with $\|\boldsymbol{x} - \boldsymbol{x}^*\| \leq \|\boldsymbol{x}^*\|/9$.

In the rest of this subsection, we will use the specific values of α, β, and R as prescribed in Lemma 2.1. Of course, an analogous result holds for $-\boldsymbol{x}^*$ by symmetry. Given the favorable geometry of the population risk, consider running gradient descent initialised at some $\boldsymbol{x}_0$ that satisfies $\|\boldsymbol{x}_0 - \boldsymbol{x}^*\| \leq R$, with step-size $\eta = 2/(\alpha + \beta)$, whose iterate at step $t+1$ is

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta \nabla r(\boldsymbol{x}_t). \tag{4}$$

This will converge at a linear rate to $\boldsymbol{x}^*$, a well-known result in convex optimisation, see for example (Bubeck, 2015). The only subtlety is in making sure that each iterate stays in the region with this geometry, and this holds because the distance from iterates to $\boldsymbol{x}^*$ will strictly contract at each step.

2.2.2 Robust gradient descent

Leaving the question of initialisation aside, one generally does not have access to the population risk or its gradients. The insight from (Chen et al., 2017; Holland and Ikeda, 2019; Prasad et al., 2020) is that exchanging integration and differentiation, (4) becomes

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta \mathbb{E}\left[((\boldsymbol{a}^\top \boldsymbol{x}_t)^2 - y)(\boldsymbol{a}^\top \boldsymbol{x})\boldsymbol{a}\right],$$

and now the unknown object is the expectation of the random variable $((\boldsymbol{a}^\top \boldsymbol{x}_t)^2 - y)(\boldsymbol{a}^\top \boldsymbol{x})\boldsymbol{a}$. In turn, this can be replaced by accurate robust and heavy-tailed mean estimators:

Definition 2.1.1. *For a sample $T = \{(\boldsymbol{a}_i, y_i)\}_{i=1}^m$ of size m, we say that $\boldsymbol{g}(\,\cdot\,; T, \delta, \varepsilon)$ is a* gradient estimator *if there exists functions $A, B : \mathbb{N} \times [0,1]^2 \to \mathbb{R}$ such that, for any fixed $\boldsymbol{x} \in \mathbb{R}^n$, with probability at least $1 - \delta$,*

$$\|\boldsymbol{g}(\boldsymbol{x}; T, \delta, \varepsilon) - \nabla r(\boldsymbol{x})\| \leq A(m, \delta, \varepsilon)\|\boldsymbol{x} - \boldsymbol{x}^*\| + B(m, \delta, \varepsilon). \tag{5}$$

Instead of using $\nabla r(\boldsymbol{x}_t)$ as the descent direction, our algorithms will follow instead $\boldsymbol{g}_t := g(\boldsymbol{x}_t; T, \delta, \varepsilon)$,

where T is a sample of size m, δ is a fixed confidence parameter and ε is the known contamination level in the sample T. We will require that the sample T is fresh at each iteration to be able to control the deviation bound (5) for any fixed (and thus independent of T) point $\boldsymbol{x}$. Access to fresh samples at each iteration has appeared before in the robust gradient descent literature, see for example (Prasad et al., 2020; Merad and Gaïffas, 2023a). The new iterate is obtained as

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta \boldsymbol{g}_t. \tag{6}$$

The guarantees of new iterate $\boldsymbol{x}_{t+1}$ are described in the following Lemma:

Lemma 2.2. *Suppose $\boldsymbol{x}_t$ obeys $\|\boldsymbol{x}_t - \boldsymbol{x}^*\| \leq R$ and $\eta \leq 2/(\alpha+\beta)$. Then, with probability at least $1-\delta$, the new iterate $\boldsymbol{x}_{t+1}$ obtained according to (6) satisfies*

$$\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^*\| \leq \left(\sqrt{1 - \frac{2\eta\alpha\beta}{\alpha+\beta}} + \eta A(m,\delta,\varepsilon)\right)\|\boldsymbol{x}_t - \boldsymbol{x}^*\| + \eta B(m,\delta,\varepsilon). \tag{7}$$

Lemma 2.2 can be then applied inductively as long as the right-hand side of (7) is at most R. In turn, this will hold if $A(m,\delta,\varepsilon)$ and $B(m,\delta,\varepsilon)$ are small enough, for which an appropriate gradient estimator has to be chosen. We recall a result of Diakonikolas et al. (2020):

Proposition 2.2.1. *[Proposition 1.5 in (Diakonikolas et al., 2020)] Let T be an ε-corrupted set of m samples from a distribution in $\mathbb{R}^n$ with mean $\boldsymbol{\mu}$ and covariance Σ. Let $r(\Sigma) = \operatorname{tr}(\Sigma)/\|\Sigma\|_{\mathrm{op}}$ and $\varepsilon' = \Theta(\log(1/\delta)/m + \varepsilon) \leq c$ be given, for a constant $c > 0$. Then, any* stability-based algorithm *on input T and ε', efficiently computes $\widehat{\boldsymbol{\mu}}$ such that, with probability at least $1-\delta$,*

$$\|\widehat{\boldsymbol{\mu}} - \boldsymbol{\mu}\| = O\Big(\sqrt{\operatorname{tr}(\Sigma)\log r(\Sigma)/m} + \sqrt{\|\Sigma\|_{\mathrm{op}}\,\varepsilon} + \sqrt{\|\Sigma\|_{\mathrm{op}}\log(1/\delta)/m}\Big).$$

By now, there is an entire plethora of stability-based algorithms that can be used for mean estimation, the main difference between them being their runtime. We will simply use a generic stability algorithm `MeanEstStab`, and discuss a specific choice when talking about our algorithms' runtimes (c.f. Remark 3.1.2).

2.2.3 Spectral initialisation

We need to ensure that the initial iterate $\boldsymbol{x}_0$ satisfies $\|\boldsymbol{x}_0 - \boldsymbol{x}^*\| \leq R$ (or, symmetrically, $\|\boldsymbol{x}_0 + \boldsymbol{x}^*\| \leq R$). The idea used in (Candes et al., 2015) and (Ma et al., 2019) is the following: the leading eigenpair of $\mathbb{E}[y\boldsymbol{a}\boldsymbol{a}^\top] = \|\boldsymbol{x}^*\|^2 I_n + 2\boldsymbol{x}^*(\boldsymbol{x}^*)^\top$ is $(\pm\boldsymbol{x}^*, 3\|\boldsymbol{x}^*\|^2)$. So, to recover the direction and scale of $\boldsymbol{x}^*$, it suffices to compute the leading eigenpair for an estimate of the mean of the random matrix $y\boldsymbol{a}\boldsymbol{a}^\top$. However, we need to use a robust mean estimator instead of the matrix empirical mean, and such estimators were primarily designed for vectors. In order to overcome this issue, we propose two alternatives:

1. One can estimate the vectors $\mathbb{E}[y(\boldsymbol{a}^\top\boldsymbol{e}_i)\boldsymbol{a}] = \mathbb{E}[y\boldsymbol{a}\boldsymbol{a}^\top]\boldsymbol{e}_i$ for each $i \in [n]$ using `MeanEstStab` and stack these as the columns of the estimate for $\mathbb{E}[y\boldsymbol{a}\boldsymbol{a}^\top]$. While this will give a polynomial time algorithm with the right choise of `MeanEstStab`, aggregating the estimated columns would result in a suboptimal sample size and contamination level, scaling as $n^2\log(n)$ and $1/n$, respectively.
2. Instead, one can note that $\operatorname{Cov}(y\boldsymbol{a}) = (3\|\boldsymbol{x}^*\|^4 + \sigma^2)I_n + 12\|\boldsymbol{x}^*\|^2\boldsymbol{x}^*(\boldsymbol{x}^*)^\top$, whose leading eigenvector is again $\pm\boldsymbol{x}^*$. This allows for the use of covariance estimators for corrupt and heavy-tailed (fourth moment bounded) data, which will improve the sample complexity to $O(n)$ and allow for a constant (i.e. not dependent on the dimension n) contamination level. We generically call such estimators `CovEst` and present their statistical guarantees in the following:

Proposition 2.2.2. *[Theorem 1.3 in (Oliveira and Rico, 2022), see also Theorem 1 in (Abdalla and Zhivotovskiy, 2024)] Let T be an ε-corrupted set of m samples from a distribution in $\mathbb{R}^n$ of a random variable $\boldsymbol{X} \in \mathbb{R}^d$ with mean 0 and covariance matrix $\Sigma \in \mathbb{R}^{n\times n}$, satisfying $\mathbb{E}[\|\boldsymbol{X}\|^4] < \infty$. Denote by*

$$\kappa_4 = \sup_{\boldsymbol{v}\in\mathbb{R}^n, \boldsymbol{v}^\top\Sigma\boldsymbol{v}=1} \mathbb{E}[(\boldsymbol{v}^\top\boldsymbol{X})^4]^{1/4}.$$

Fix the confidence $1-\delta \in (0,1)$. Let $r(\Sigma) = \operatorname{tr}(\Sigma)/\|\Sigma\|_{\mathrm{op}}$ and suppose $\varepsilon = O(\kappa_4^{-4})$, $m = \Omega(r(\Sigma) + \log(1/\delta))$. There is an estimator `CovEst` depending on δ, m and ε that takes as input the corrupted sample T and outputs $\widehat{\Sigma} \in \mathbb{R}^{n\times n}$ such that, with probability at least $1-\delta$,

$$\left\|\widehat{\Sigma} - \Sigma\right\|_{\mathrm{op}} = O\Big(\kappa_4^2\|\Sigma\|_{\mathrm{op}}\big(\sqrt{r(\Sigma)/m} + \sqrt{\varepsilon} + \sqrt{\log(1/\delta)/m}\big)\Big).$$

Recovering the scale of $\boldsymbol{x}^*$ can be done by noting that $\mathbb{E}[y] = \|\boldsymbol{x}^*\|^2$, and thus a robust estimate for $\|\boldsymbol{x}^*\|$ can be obtained using `MeanEstStab` on the responses y.

Finally, recall that the step-size we use is $\eta \leq 2/(\alpha+\beta) = \Theta(\|\boldsymbol{x}^*\|^{-2})$. As a consequence of a spectral initialisation we also have $\|\boldsymbol{x}^*\| \approx \|\boldsymbol{x}_0\|$, so we can use $\|\boldsymbol{x}_0\|$ to find an appropriate value for η.

2.3 Unknown mean noise

For the case of noise with an unknown (and potentially non-zero) mean, the idea described in the pre-

vious subsection does not work anymore, as the posterior mean of the population risk (3) will depend on the unknown noise variance. We propose reducing this problem to one that falls outside the scope of phase retrieval, but in which the noise has again mean zero.

More specifically, suppose we have two samples $y = (\boldsymbol{a}^\top \boldsymbol{x}^*)^2 + z$ and $y' = ((\boldsymbol{a}')^\top \boldsymbol{x}^*)^2 + z'$. Subtracting one from the other and rescaling leads to

$$\underbrace{\frac{y - y'}{2}}_{:=\upsilon} = \underbrace{\left(\frac{\boldsymbol{a} + \boldsymbol{a}'}{\sqrt{2}}\right)^\top}_{:=\boldsymbol{b}} \boldsymbol{x}^* (\boldsymbol{x}^*)^\top \underbrace{\left(\frac{\boldsymbol{a} - \boldsymbol{a}'}{\sqrt{2}}\right)}_{:=\boldsymbol{c}} + \underbrace{\frac{z - z'}{2}}_{:=\zeta}.$$

Generally, from an (uncorrupted) sample $S = \{(\boldsymbol{a}_j, y_j)\}_{j=1}^{2m}$ of size $2m$ following (1) with unknown and potentially non-zero mean noise, we obtain a sample $S' = \{(\boldsymbol{b}_j, \boldsymbol{c}_j, \upsilon_j)\}_{j=1}^{m}$ from the model

$$\upsilon = \boldsymbol{b}^\top \boldsymbol{x}^* (\boldsymbol{x}^*)^\top \boldsymbol{c} + \zeta, \tag{8}$$

where now $\boldsymbol{b}, \boldsymbol{c} \sim \mathcal{N}(\boldsymbol{0}, I_d)$ are independent, and the noise ζ has (known) mean zero and variance $\sigma^2/2$ and is independent of $\boldsymbol{b}$ and $\boldsymbol{c}$. The new sample is obtained as $\boldsymbol{b}_j = (\boldsymbol{a}_j + \boldsymbol{a}_{m+j})/\sqrt{2}$, $\boldsymbol{c}_j = (\boldsymbol{a}_j - \boldsymbol{a}_{m+j})/\sqrt{2}$, and $\upsilon_j = (y_j - y_{m+j})/2$. Note that, when the initial sample is ε-corrupted, the new sample will be 2ε-corrupted.

A similar preprocessing step has been applied by Pensia et al. (2020) for linear regression. However, unlike in our case, the resulting data in their work is still following a linear regression. The new model (8) falls outside the scope of phase retrieval and can be seen as a restricted case of blind deconvolution (Ahmed et al., 2013). While the population risk associated with blind deconvolution does not exhibit local strong convexity and smoothness (Ma et al., 2019), in our case (8) does:

Lemma 2.3. *The population risk associated to (8),*

$$r_{\text{new}}(\boldsymbol{x}) = \mathbb{E}\left[(\boldsymbol{x}^\top \boldsymbol{b}\boldsymbol{c}^\top \boldsymbol{x} - \upsilon)^2\right]/2, \tag{9}$$

is $\|\boldsymbol{x}^*\|^2$-strongly convex and $49\|\boldsymbol{x}^*\|^2/12$-smooth in a ball around $\boldsymbol{x}^*$ of radius $\|\boldsymbol{x}^*\|/6$.

We defer this case to the Supplementary Material, where we show that the same type of procedures and analysis as in the case of known zero mean noise can be applied to this scenario after the preprocessing step, and the statistical and computational guarantees are identical up to universal constants.

3 MAIN RESULTS

In this section, we assume the learner knows that the noise in the underlying robust phase retrieval model has mean zero. We begin with Procedure 1, which outputs, using a spectral method, the initial iterate $\boldsymbol{x}_0 \in \mathbb{R}^d$ of the robust gradient descent. It takes as inputs a confidence level $\delta \in (0,1)$, the contamination parameter ε and it is allowed access to samples from the robust phase retrieval model. This is reminiscent of Algorithm 1 in (Ma et al., 2019), but now two configurations are possible: either the `MeanEstStab` algorithm or `CovEst` estimator is being used to estimate the mean of a random matrix or covariance of a random vector. As attested by Theorems 3.1 and 3.2, the property of Procedure 1 's output (in either configuration) is that with high probability it lies in a region of strong convexity and smoothness of Lemma 2.1.

Procedure 1: Spectral initialisation for robust phase retrieval with zero mean noise

Inputs: $\delta \in (0,1)$, $\varepsilon > 0$, access to robust phase retrieval data
Output: $\boldsymbol{x}_0 \in \mathbb{R}^n$
1. Receive a sample $T = \{(\boldsymbol{a}_j, y_j)\}_{j=1}^{m_0}$ of size m_0 from the robust phase retrieval model.

Either *(Algorithmic)* `MeanEstStab` *configuration:*
2. For $j \in [m_0]$ and $i \in [n]$, let $\boldsymbol{p}_{ij} = y_j(\boldsymbol{a}_j^\top \boldsymbol{e}_i)\boldsymbol{a}_j$. Let $\boldsymbol{\mu}^{(i)}$ be the `MeanEstStab` estimate for $\{\boldsymbol{p}_{ij}\}_{j=1}^{m_0}$, $\boldsymbol{Y} \in \mathbb{R}^{n\times n}$ the matrix whose i'th column is $\boldsymbol{\mu}^{(i)}$.

Or `CovEst` *configuration:*
2. Let $\boldsymbol{Y} \in \mathbb{R}^{n\times n}$ be the `CovEst` estimate for $\{y_j \boldsymbol{a}_j\}_{j=1}^{m_0}$.

3. Let $\widetilde{y}^2$ be the `MeanEstStab` output on $\{y_j\}_{j=1}^{m_0}$. If $\widetilde{y}^2 < 0$, return $\boldsymbol{0}$. Else, set $\boldsymbol{x}_0$, normalised to $\|\boldsymbol{x}_0\| = \sqrt{\widetilde{y}^2}$, to be an eigenvector corresponding to the largest eigenvalue of $(\boldsymbol{Y} + \boldsymbol{Y}^\top)/2$. Return $\boldsymbol{x}_0$.

Theorem 3.1. *(`MeanEstStab` Spectral initialisation) Let $\boldsymbol{x}^* \in \mathbb{R}^n$ be arbitrary and $\delta \in (0,1)$. There exist universal constants C_1, C_2 such that, when the contamination level satisfies $\varepsilon \le C_1(\|\boldsymbol{x}^*\|^4/\sigma^2)n^{-1}$, the sample size is $m_0 \ge C_2 \max\{n^2\log(n), n\log(n/\delta)\}\sigma^2/\|\boldsymbol{x}^*\|^4$, and Procedure 1 with configuration `MeanEstStab` is run on inputs δ, ε and requests a sample of size m_0 from the robust phase retrieval with mean zero noise model, it returns $\boldsymbol{x}_0 \in \mathbb{R}^n$ such that, with probability at least $1 - \delta$,*

$$\operatorname{dist}(\boldsymbol{x}_0, \boldsymbol{x}^*) \le \|\boldsymbol{x}^*\|/9.$$

Remark 3.1.1. (Sample size and contamination leve) *As prescribed by Theorem 3.1, Procedure 1 with configuration `MeanEstStab` requires $m_0 = O(n^2\log(n))$ samples and tolerates an amount of corruption $\varepsilon = O(1/n)$ in order to guarantee a 'good-quality' initial iterate $\boldsymbol{x}_0$. In particular, one would expect a better dependence on the ambient dimension n for both of these*

based on previous results in the respective literatures (i.e. $m_0 = O(n)$ as in (Candes et al., 2015; Ma et al., 2019) and $\varepsilon = O(1)$ (in n)). While these quantities can be improved in our setting by using a covariance estimator (c.f. Theorem 3.2), there are two advantage of using the `MeanEstStab`*. Firstly, this procedure configuration works even for noise that does not have a bounded fourth moment, but only a bounded away from zero signal-to-noise ratio. This primarily happens because, unlike in Proposition 2.2.2, there are no finite fourth moment assumptions in Proposition 2.2.1. The second advantage comes from a computational viewpoint, as outlined in the following Remark 3.1.2.*

Remark 3.1.2. (Computational complexity) *The heavier computation takes place in Step 2 of Procedure 1 with* `MeanEstStab` *configuration: n uses of* `MeanEstStab` *are being made, each time for m_0 vectors in $\mathbb{R}^n$. So Step 2's computational cost is n times larger than the runtime of* `MeanEstStab` *on an input of size nm_0 and thus it is entirely determined by the choice of a stable algorithm. For example, the one designed by Hopkins et al. (2020) runs in nearly linear time $\widetilde{O}(m_0 n)$. It is based on a matrix multiplicative update algorithm and, for brevity, we omit a detailed description.*[1] *According to Proposition 2.2.1, we need to ensure that $\varepsilon' = \Theta(\log(1/\delta)/m + \varepsilon) \leq c$ for a constant $c > 0$. This is the case when used to implement* `MeanEstStab` *in Procedure 1 (and, below, in Algorithm 2) because the signal-to-noise ratio $\|\boldsymbol{x}^*\|^2/\sigma$ is assumed to be constant. Furthermore, the reason why we use $(\boldsymbol{Y} + \boldsymbol{Y}^\top)/2$ in Step 3, instead of $\boldsymbol{Y}$, is that it is easier to compute eigenpairs for the former symmetric matrix than singular pairs for the latter. This step can be executed via the power method in time $\widetilde{O}(1)$.*

Theorem 3.2. *(*`CovEst` *Spectral initialisation) With the* `CovEst` *configuration of Procedure 1, but an improved sample size of $m_0 \geq C_2 \max\{n, \log(1/\delta)\} K_4^4 / \|\boldsymbol{x}^*\|^8$ and a contamination level independent of the ambient dimension $\varepsilon \leq C_1(\|\boldsymbol{x}^*\|^8 / K_4^4)$, the same conclusion as in Theorem 3.1 holds.*

Remark 3.2.1. *In comparison to the guarantees of Theorem 3.1, the spectral initialisation sample size is now of order $O(n)$, comparable to previous (but much less general, see the discussion in Subsection 1.1 and the Supplementary Material) works on phase retrieval, and the contamination is constant in n, as it is the case in the robust statistics literature. Also, while in Theorem 3.1 the sample size and contamination level were functions of the signal-to-noise ratio $\|\boldsymbol{x}^*\|^2/\sigma$, now they depend on $\|\boldsymbol{x}^*\|^2/K_4$, a fourth-moment analogue. Specifically, this new ratio is a result of computing the value κ_4 from Proposition 2.2.2 in our specific phase retrieval instance, whereas in Theorem 3.1 the signal-to-noise ratio dependence was a consequence of the factor $\|\Sigma\|_{\text{op}}^{1/2}$ in Proposition 2.2.1. Moreover, the assumption that the noise has a bounded fourth moment made in Subsection 2.1 is due to the data distribution requirement in Proposition 2.2.2.*

Remark 3.2.2. *Although, to the best of our knowledge, there are currently no efficient methods for covariance estimation, in fact one does not necessarily need to deploy covariance estimators for the spectral initialisation task. An efficient and statistically optimal algorithm that computes the leading eigenvector of a covariance matrix (i.e. combining Steps 2 and 3) would suffice. This latter task is a significant and independent topic of interest in robust statistics, with applications in other areas not directly related to phase retrieval, such as robust PCA. Enhancements in these tools would directly find applications within the setting we consider. At the level of generality we consider, fully addressing the simpler setting of linear regression has demanded considerable effort from the robust statistics community (c.f. Section 1).*

Next, this initial iterate $\boldsymbol{x}_0$ is then passed to the iterative Algorithm 2. It takes as inputs a confidence parameter $\delta \in (0,1)$, the contamination level ε, the number of iterations $T > 0$, and it is allowed access to robust phase retrieval data. Its theoretical guarantees are presented in Theorem 3.3.

Algorithm 2: Gradient descent for robust phase retrieval with zero-mean noise

Inputs: $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\delta \in (0,1)$, $\varepsilon > 0$, $T \in \mathbb{N}$.
Output: $\boldsymbol{x}_T \in \mathbb{R}^n$
1. Set $\eta = 128/(981\|\boldsymbol{x}_0\|^2)$. For $t = 0, \ldots, T-1$:

- Receive a sample $B_t = \{(\boldsymbol{a}_j, y_j)\}_{j=1}^{\widetilde{m}}$ of size $\widetilde{m}$ from the robust phase retrieval model.
- Gradient estimation: For each $(\boldsymbol{a}_j, y_j) \in B_t$, let $\boldsymbol{p}_j = ((\boldsymbol{a}_j^\top \boldsymbol{x}_j)^2 - y_j)(\boldsymbol{a}_j^\top \boldsymbol{x}_j)\boldsymbol{a}_j$ and $\boldsymbol{g}_t$ to be the `MeanEstStab` output for these $\widetilde{m}$ points.
- Update $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta \boldsymbol{g}_t$.

2. Return $\boldsymbol{x}_T$.

Theorem 3.3. *(Iterative scheme) Let $\boldsymbol{x}^* \in \mathbb{R}^n$ be an arbitrary vector and $\delta \in (0,1)$. There exist universal constants C_1, C_2, C_3, C_4 such that, when $\boldsymbol{x}_0 \in \mathbb{R}^n$ is in a ball around $\pm\boldsymbol{x}^*$ of radius $\|\boldsymbol{x}^*\|/9$, the contamination level is $\varepsilon \leq C_1\|\boldsymbol{x}^*\|^4/\sigma^2$, the sample size is $T\widetilde{m} \geq C_2 T \max\{n\log(n), \log(1/\delta)\}\sigma^2/\|\boldsymbol{x}^*\|^4$, and Al-*

[1] As noted after Lemma 3.2 in (Hopkins et al., 2020), the input ρ of the algorithm can be set as the squared diameter of the data.

gorithm 2 is run on inputs $\boldsymbol{x}_0$, δ, ε, T and requests a sample of size $\widetilde{m}$ from the robust phase retrieval with mean zero noise model at each iteration, it returns $\boldsymbol{x}_T \in \mathbb{R}^n$ such that, with probability at least $1 - T\delta$,

$$\frac{\operatorname{dist}(\boldsymbol{x}_T, \boldsymbol{x}^*)}{\|\boldsymbol{x}^*\|} \leq C_3 \exp\left(-C_4 \eta T \|\boldsymbol{x}^*\|^2 (1 - \sqrt{\varepsilon})\right) + C_3 \frac{\sigma}{\|\boldsymbol{x}^*\|^2} \left(\sqrt{\frac{n \log(n)}{\widetilde{m}}} + \sqrt{\frac{\log(1/\delta)}{\widetilde{m}}}\right) + C_3 \frac{\sigma}{\|\boldsymbol{x}^*\|^2} \sqrt{\varepsilon}.$$

Remark 3.3.1. *As in (Chen and Candes, 2015; Zhang et al., 2018), the step-size in Algorithm 2 does not depend on the ambient dimension, in contrast to (Candes et al., 2015; Ma et al., 2019). The estimation error is upper-bounded by a sum of an optimisation term and two statistical terms, one controlled by the sample size and the other by the contamination level:*

- *As $\eta = \Theta(\|\boldsymbol{x}^*\|^{-2})$, the optimisation error decays as $\exp(-O(T))$. As expected, when the corruption level ε increases, the convergence speed decreases.*
- *The first statistical error is the product between $\sigma / \|\boldsymbol{x}^*\|^2$, the inverse of the signal-to-noise ratio, and a slow rate term $\widetilde{m}^{-1/2}$. If there is no corruption ($\varepsilon = 0$), the smaller $\sigma / \|\boldsymbol{x}^*\|^2$ is, the easier to recover $\boldsymbol{x}^*$ one expects to be. Indeed, if we consider $\sigma \to 0$, the statistical error becomes 0. This is also the case in the infinite sample regime $\widetilde{m}/(n \log(n)) \to \infty$.*
- *Finally, the error induced by the adversary is the product between the inverse of the signal-to-noise ratio and $\sqrt{\varepsilon}$. The larger this product is, the more noise and contamination exist in the data, making the problem harder. We expect this term to be optimal based on the following information-theoretic lower-bound: estimating the mean of a distribution with variance σ^2 from an ε-corrupted sample incurs an error of at least $\Omega(\sigma\sqrt{\varepsilon})$.*

Remark 3.3.2. *(Dependence on T) The dependence on the number of iterations T in both sample complexity ($T\widetilde{m}$) and confidence level ($1 - T\delta$) is a common characteristic in works employing robust gradient descent techniques, e.g. (Prasad et al., 2020; Liu et al., 2019; Merad and Gaïffas, 2023b,a). This dependence arises from the use of fresh samples at each iteration, which in turn guarantees that the descent directions used by Algorithm 2 are gradient estimators (c.f. Definition 2.1.1). Note that the optimisation and statistical errors scale as $\exp(-O(T))$ and $O(\sigma / \|\boldsymbol{x}^*\|^2)$, respectively, when $\widetilde{m} = \Omega(n \log(n))$ samples are used per iteration. Setting $T = \log(C'\sigma / \|\boldsymbol{x}^*\|^2)$, for some suitable constant C', the these two errors become of the same order. Under the assumption that the signal-to-noise ratio $\|\boldsymbol{x}^*\|^2 / \sigma$ does not depend on the ambient dimension n, we have that $T = \widetilde{O}(1)$ leads to an overall sample complexity of order $n \log(n)$ for Algorithm 2. As a function of n, our iterative procedure has the same sample size as the ones proposed in (Chen and Candes, 2015; Zhang et al., 2018), while being robust to a more general contamination model, c.f. Table 1.*

Remark 3.3.3. *(Computational complexity) Recall that the sample size is $T\widetilde{m}$. One can implement* `MeanStabEst` *using the algorithm of Hopkins et al. (2020), which runs in nearly linear time (up to logarirthmic factors, see also Remark 3.1.2). This is then used T times on $\widetilde{m}$ points in Step 2 of Algorithm 2. So, Algorithm 2 essentially has the computational complexity required to read the data.*

4 CONCLUSION

To the best of our knowledge, our work represents the first results on robustness for phase retrieval with heavy-tailed noise and contamination in both the covariates and responses. This setting is *general* and imposes minimal assumptions on the data-generating mechanism. Fully addressing a simpler scenario in the context of linear regression has required considerable effort from the community (c.f. Section 1). We consider two settings: the learner either knows that the noise has mean zero, or has no information about the noise mean, which can potentially be non-zero. To tackle the latter case we propose in the Supplementary Material a data preprocessing step that reduces phase retrieval to a different statistical model that can be seen as a restriction of blind deconvolution. The population risk landscape associated with both models enjoys favorable geometry around the global minima. We use this property to run spectrally initialised robust gradient descent. The main appeal of this method is that it is as versatile as the estimator used: by choosing mean estimators resilient to both heavy tails and adversarial corruption, we obtain descent directions that are simultaneously robust to both types of outliers. The results we establish within our general robustness setting involve a contamination level that scales as $1/n$ for the computationally tractable initialisation step based on mean estimation, where n is the ambient dimension of the problem, and as a constant for the spectral initialisation based on covariance estimation, and as well a constant for the iterative procedure. Further improvements (either statistical or from an algorithmic perspective) in the initialisation scheme necessitate significant advances in leading eigenvector estimators for covariance matrices, which is a topic of independent interest with wider applications in other areas, such as robust PCA (c.f. Remark 3.2.2).

Acknowledgments

Patrick Rebeschini was funded by UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee [project reference EP/Y028333/1]. Alex Buna was supported by EPSRC through the StatML CDT programme [project reference EP/S023151/1]. For the purpose of Open Access, the authors have applied a CC BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission.

References

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

Ahmed, A., Recht, B., and Romberg, J. (2013). Blind deconvolution using convex programming. *IEEE Transactions on Information Theory*, 60(3):1711–1732.

Bubeck, S. (2015). Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357.

Candès, E. J. and Li, X. (2014). Solving quadratic equations via phaselift when there are about as many equations as unknowns. *Foundations of Computational Mathematics*, 14:1017–1026.

Candes, E. J., Li, X., and Soltanolkotabi, M. (2015). Phase retrieval via Wirtinger Flow: Theory and algorithms. *IEEE Transactions on Information Theory*, 61(4):1985–2007.

Catoni, O. (2012). Challenging the empirical mean and empirical variance: a deviation study. In *Annales de l'IHP Probabilités et statistiques*, volume 48, pages 1148–1185.

Chen, J. and Ng, M. K. (2022). Error bound of empirical ℓ_2 risk minimization for noisy standard and generalized phase retrieval problems. *arXiv preprint arXiv:2205.13827*.

Chen, M., Lin, P., Quan, Y., Pang, T., and Ji, H. (2022). Unsupervised phase retrieval using deep approximate mmse estimation. *IEEE Transactions on Signal Processing*, 70:2239–2252.

Chen, Y. and Candes, E. (2015). Solving random quadratic systems of equations is nearly as easy as solving linear systems. *Advances in Neural Information Processing Systems*, 28.

Chen, Y., Su, L., and Xu, J. (2017). Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2):1–25.

Diakonikolas, I., Kamath, G., Kane, D., Li, J., Moitra, A., and Stewart, A. (2019a). Robust estimators in high-dimensions without the computational intractability. *SIAM Journal on Computing*, 48(2):742–864.

Diakonikolas, I., Kamath, G., Kane, D., Li, J., Steinhardt, J., and Stewart, A. (2019b). Sever: A robust meta-algorithm for stochastic optimization. In *International Conference on Machine Learning*, pages 1596–1606. PMLR.

Diakonikolas, I., Kane, D. M., and Pensia, A. (2020). Outlier robust mean estimation with subgaussian rates via stability. *Advances in Neural Information Processing Systems*, 33:1830–1840.

Drenth, J. (2007). Principles of protein X-ray crystallography. *Springer Nature*.

Duchi, J. C. and Ruan, F. (2019). 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.

Fannjiang, A. and Strohmer, T. (2020). The numerics of phase retrieval. *Acta Numerica*, 29:125–228.

Hampel, F. R. (1974). The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393.

Hand, P., Leong, O., and Voroninski, V. (2018). Phase retrieval under a generative prior. *Advances in Neural Information Processing Systems*, 31.

Holland, M. J. and Ikeda, K. (2019). Efficient learning with robust gradient descent. *Machine Learning*, 108:1523–1560.

Hopkins, S., Li, J., and Zhang, F. (2020). Robust and heavy-tailed mean estimation made simple, via regret minimization. *Advances in Neural Information Processing Systems*, 33:11902–11912.

Hopkins, S. B. (2020). Mean estimation with sub-Gaussian rates in polynomial time. *Annals of Statistics*, 48(2):1193–1213.

Huber, P. J. (1964). Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73 – 101.

Lai, K. A., Rao, A. B., and Vempala, S. (2016). Agnostic estimation of mean and covariance. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 665–674.

Lattimore, T. and Hao, B. (2021). Bandit phase retrieval. In *Advances in Neural Information Processing Systems*, volume 34, pages 18801–18811.

Liu, L., Li, T., and Caramanis, C. (2019). High dimensional robust M-estimation: Arbitrary corruption and heavy tails. *arXiv preprint arXiv:1901.08237*.

Lugosi, G. and Mendelson, S. (2019). Sub-Gaussian estimators of the mean of a random vector. *The Annals of Statistics*, 47(2):783 – 794.

Ma, C., Wang, K., Chi, Y., and Chen, Y. (2019). Implicit regularization in nonconvex statistical estimation: Gradient descent converges linearly for phase retrieval, matrix completion, and blind deconvolution. *Foundations of Computational Mathematics*, 20(3):451–632.

Merad, I. and Gaïffas, S. (2023a). Robust methods for high-dimensional linear learning. *Journal of Machine Learning Research*, 24(165):1–44.

Merad, I. and Gaïffas, S. (2023b). Robust supervised learning with coordinate gradient descent. *Statistics and Computing*, 33(5):116.

Minsker, S. (2015). Geometric median and robust estimation in Banach spaces. *Bernoulli*, 21(4):2308 – 2335.

Oliveira, R. I. and Rico, Z. F. (2022). Improved covariance estimation: optimal robustness and sub-gaussian guarantees under heavy tails. *arXiv preprint arXiv:2209.13485*.

Pensia, A., Jog, V., and Loh, P.-L. (2020). Robust regression with covariate filtering: Heavy tails and adversarial contamination. *arXiv preprint arXiv:2009.12976*.

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

Shechtman, Y., Eldar, Y. C., Cohen, O., Chapman, H. N., Miao, J., and Segev, M. (2014). Phase retrieval with application to optical imaging.

Tukey, J. W. (1975). Mathematics and the picturing of data. *Proceedings of ICM*, 6:523—531.

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142.

Vershynin, R. (2018). *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge University Press.

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

Wu, F. and Rebeschini, P. (2023). Nearly minimax-optimal rates for noisy sparse phase retrieval via early-stopped mirror descent. *Information and Inference: A Journal of the IMA*, 12(2):633–713.

Yin, D., Chen, Y., Kannan, R., and Bartlett, P. (2019). Defending against saddle point attack in byzantine-robust distributed learning. In *International Conference on Machine Learning*, pages 7074–7084.

Zhang, H., Chi, Y., and Liang, Y. (2018). Median-truncated nonconvex approach for phase retrieval with outliers. *IEEE Transactions on Information Theory*, 64(11):7287–7310.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes] The mathematical setting, assumptions and model are given in Subsection 2.1. The algorithms can be found in Section 3.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes] The proofs of all theorems and lemmas (along with sample size analysis, where applicable) can be found in the Supplementary Material. Time complexity discussions are given in Remarks 3.1.2 and 3.3.3.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes] All proofs are written formally.
   (b) Complete proofs of all theoretical results. [Yes] All proofs can be found in the Supplementary Materieal.
   (c) Clear explanations of any assumptions. [Yes] See Subsection 2.1.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Not Applicable]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Not Applicable]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Not Applicable]

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

   (a) Citations of the creator If your work uses existing assets. [Not Applicable]

   (b) The license information of the assets, if applicable. [Not Applicable]

   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]

   (d) Information about consent from data providers/curators. [Not Applicable]

   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

   (a) The full text of instructions given to participants and screenshots. [Not Applicable]

   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Material

A RELATED WORK

Robust gradient descent aims at performing gradient descent on the population risk associated with a learning problem. However, instead of using the inaccessible population risk gradients, it replaces them with robust mean estimates. Amongst the first uses of this method can be traced back to the works of Chen et al. (2017) and Holland and Ikeda (2019). The former uses smaller batches of data to construct gradients that are further aggregated using the geometric median-of-means, while the latter replaces the usual empirical risk gradient with a weighted version. (Yin et al., 2019) applies the robust gradient descent framework with different mean estimators (median, trimmed mean, iterative filtering) to non-convex problems, but in the Byzantine learning framework, which differs from ours, as we allow each fresh sample to contain outliers, rather than some samples to be clean and some to be completely corrupted. (Prasad et al., 2020) and (Diakonikolas et al., 2019b) employ different estimators for the robust gradient that can deal with large classes of convex problems, while (Liu et al., 2019) analyses the high-dimensional scenario under sparsity constraints. This latter work allows the contamination level to depend on the sparsity of the underlying solution, the number of samples and the dimension of the problem. By using robust coordinate descent, (Merad and Gaïffas, 2023b) designs algorithms for robust linear learning with almost the same run-time and sample complexity as the non-robust counterparts. Finally, (Merad and Gaïffas, 2023a) improves on the results of Liu et al. (2019) by using robust mirror descent, while also studying other linear learning problems.

Wirtinger Flow for phase retrieval performs gradient descent with spectral initialisation on empirical objectives with favorable geometry (i.e. strongly convex and smooth regions) around the true signal $\pm\boldsymbol{x}^*$. It was first proposed in (Candes et al., 2015) to study *exact* phase retrieval, i.e. when there is no noise or contamination in the responses. At a very high level, it considers the unregularised empirical risk under the ℓ_2 loss, which is used to direct the vanilla descent procedure with step size proportional to $1/n$ towards one of the global minima. The sample size and iteration complexity (i.e. number of steps) needed to guarantee an accuracy of τ are $n \log n$ and $n \log(1/\tau)$, respectively. The step size and iteration complexity for the same vanilla gradient descent procedure have been further improved in (Ma et al., 2019) to $1/\log(n)$ and $\log(n)\log(1/\tau)$, respectively, by noticing the implicit bias induced by the Wirtinger Flow to a so-called region of incoherence and contraction. Leaving the exact phase retrieval model aside, various truncation procedures have been proposed to make the Wirtinger Flow robust to bounded and possibly contamination in the responses (Chen and Candes, 2015; Zhang et al., 2018). While these works improve the sample complexity to n, the iteration complexity to $\log(1/\tau)$, and the step size to a constant, the trimming procedures applied to the gradient are rather ad-hoc. This makes the analyses of their theoretical guarantees hard to translate to more general types of noise, such as heavy-tailed, and more powerful adversaries that can alter the covariates as well. A detailed summary can be found in Table 1 in the Main Paper. Statistical guarantees for the ℓ_2 empirical risk minimiser of phase retrieval with heavy-tailed noise have previously been obtained in (Chen and Ng, 2022), but these do not assume any contamination in the data or offer a computational framework. Another approach to solving the noiseless phase retrieval problem when only the responses are contaminated can be found in (Duchi and Ruan, 2019) and it is based on composite optimisation. Rather than employing (a form of) gradient-descent, the authors resort to a prox-linear algorithm and convex programming. Finally, we mention that there are various extensions of Wirtinger Flow for sparse phase retrieval that replace the gradient descent procedure with mirror descent and an appropriately chosen mirror map, see, for example, the work of Wu and Rebeschini (2023) and the reference therein.

B UNKNOWN MEAN NOISE MAIN RESULTS

In this section, the assumption on the noise to have mean zero is dropped, yet the variance σ^2 is still bounded and unknown. The learner does not know the mean of the noise anymore but rather works with a modified dataset generated from the new statistical model (8) with zero mean noise.

We first present the spectral initialisation Procedure 3. As before, this has two configurations: either the `MeanEstStab` algorithm or `CovEst` estimator is being used. In the first case, it recovers the direction of $\boldsymbol{x}^*$ using an column-wise estimate of $\mathbb{E}[\upsilon \boldsymbol{b}\boldsymbol{c}^\top] = \boldsymbol{x}^*\boldsymbol{x}^{*\top}$, whose the top eigenvalue is aligned with $\boldsymbol{x}^*$. In the second

case, it estimates directly $\mathrm{Cov}(\nu b) = (\|\boldsymbol{x}^*\|^4 + \sigma^2/2)I_n + 2\|\boldsymbol{x}^*\|^2 \boldsymbol{x}^*(\boldsymbol{x}^*)^\top$, again whose leading eigenvector is a multiple of $\boldsymbol{x}^*$. Then, it does an appropriate rescaling by using `MeanEstStab` for $\mathbb{E}[\nu \boldsymbol{b}^\top \boldsymbol{c}] = \|\boldsymbol{x}^*\|^2$. Its theoretical guarantees are given in Theorems B.1 and B.2.

Procedure 3: Spectral initialisation for robust phase retrieval with unknown mean noise

Inputs: $\delta \in (0,1)$, $\varepsilon > 0$, access to robust phase retrieval data
Output: $\boldsymbol{x}_0 \in \mathbb{R}^n$
1. Receive a sample $T = \{(\boldsymbol{a}_j, y_j)\}_{j=1}^{2m_0}$ of size $2m_0$ from the robust phase retrieval model. Construct a new dataset: for each $j \in [m_0]$,

$$\boldsymbol{b}_j = (\boldsymbol{a}_j + \boldsymbol{a}_{m_0+j})/\sqrt{2}, \boldsymbol{c}_j = (\boldsymbol{a}_j - \boldsymbol{a}_{m_0+j})/\sqrt{2}, \quad \text{and} \quad \upsilon_j = (y_j - y_{m_0+j})/2.$$

Either (*Algorithmic*) `MeanEstStab` *configuration*:
2. For $j \in [m_0]$ and $i \in [n]$, let $\boldsymbol{p}_{ij} = \upsilon_j(\boldsymbol{c}_j^\top \boldsymbol{e}_i)\boldsymbol{b}_j$. Let $\boldsymbol{\mu}_i$ be the `MeanEstStab` estimate for $\{\boldsymbol{p}_{ij}\}_{j=1}^{m_0}$ and let $\boldsymbol{Y} \in \mathbb{R}^{n\times n}$ be the matrix whose i'th column is $\boldsymbol{\mu}_i$.

Or `CovEst` *configuration*:
2. Let $\boldsymbol{Y} \in \mathbb{R}^{n\times n}$ be the `CovEst` estimate for $\{\nu_j \boldsymbol{b}_j\}_{j=1}^{m_0}$.

---

3. Let $\widetilde{y}^2$ be the `MeanEastStab` output on $\{\nu \boldsymbol{b}^\top \boldsymbol{c}\}_{j=1}^{m_0}$. If $\widetilde{y}^2 < 0$, return $\mathbf{0}$. Else, set $\boldsymbol{x}_0$, normalised to $\|\boldsymbol{x}_0\| = \sqrt{\widetilde{y}^2}$, to be an eigenvector corresponding to the largest eigenvalue of $(\boldsymbol{Y} + \boldsymbol{Y}^\top)/2$. Return $\boldsymbol{x}_0$. Return $\boldsymbol{x}_0$.

Theorem B.1. *(`MeanEstStab` Spectral initialisation) Let $\boldsymbol{x}^* \in \mathbb{R}^n$ be an arbitrary vector and $\delta \in (0,1)$. There exist universal constants C_1, C_2 such that, when the contamination level satisfies $\varepsilon/2 \leq C_1(\|\boldsymbol{x}^*\|^4/\sigma^2)n^{-1}$, the sample size is $2m_0 \geq C_2 \max\{n^2\log(n), n\log(n/\delta)\}\sigma^2/\|\boldsymbol{x}^*\|^4$, and Procedure 3 with configuration `MeanEstStab` is run on inputs δ, ε and requests a sample of size $2m_0$ from the robust phase retrieval model (with unknown noise mean), it returns $\boldsymbol{x}_0 \in \mathbb{R}^n$ such that, with probability at least $1-\delta$,*

$$\mathrm{dist}(\boldsymbol{x}_0, \boldsymbol{x}^*) \leq \|\boldsymbol{x}^*\|/6.$$

Theorem B.2. *(`CovEst` Spectral initialisation) With the `CovEst` configuration of Procedure 3, but an improved sample size of $m_0 \geq C_2 \max\{n, \log(1/\delta)\}K_4^4/\|\boldsymbol{x}^*\|^8$ and a contamination level independent of the ambient dimension $\varepsilon \leq C_1(\|\boldsymbol{x}^*\|^8/K_4^4)$, the same conclusion as in Theorem B.1 holds.*

Note that Theorems B.1 and B.2 have essentially the same guarantees as Theorems 3.1 and 3.2, and the extra symmetrisation in Step 2 does increase the run-time.

These proofs of these two theorems are identical to the ones of Theorems 3.1 and 3.2, which can be found in Section D, and we omit them. The only difference is that now, instead of employing the expressions computed in Lemma E.3, the ones from Lemma E.4 should be used.

Next, we present Algorithm 4 that starts the iterative procedure from the output of Procedure 3. Its guarantees are given in Theorem B.3.

Algorithm 4: Gradient descent for robust phase retrieval with unknown mean noise

Inputs: $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\delta \in (0,1)$, $\varepsilon > 0$, $T \in \mathbb{N}$.
Output: $\boldsymbol{x}_T \in \mathbb{R}^n$
1. Set $\eta = 1024/(1647\left\|\boldsymbol{x}_0\right\|^2)$. For $t = 0, \ldots, T-1$:

- Receive a sample $B_t = \{(\boldsymbol{a}_j, y_j)\}_{j=1}^{2\widetilde{m}}$ of size $2\widetilde{m}$ from the robust phase retrieval model.
- Construct a new dataset: for each $j \in [\widetilde{m}]$,

$$\boldsymbol{b}_j = (\boldsymbol{a}_j + \boldsymbol{a}_{\widetilde{m}+j})/\sqrt{2}, \boldsymbol{c}_j = (\boldsymbol{a}_j - \boldsymbol{a}_{\widetilde{m}+j})/\sqrt{2} \quad \text{and} \quad \upsilon_j = (y_j - y_{\widetilde{m}+j})/2.$$

- Gradient estimation: For each $(\boldsymbol{a}_j, y_j) \in B_t$, let $\boldsymbol{p}_j = (\boldsymbol{x}_t^\top \boldsymbol{b}\boldsymbol{c}^\top \boldsymbol{x}_t - \upsilon_j)(\boldsymbol{b}\boldsymbol{c}^\top + \boldsymbol{c}\boldsymbol{b}^\top)\boldsymbol{x}_t$. Let $\boldsymbol{g}_t$ be the `MeanEstStab` estimate for these $\widetilde{m}$ points.
- Update $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta \boldsymbol{g}_t$.

2. Return $\boldsymbol{x}_T$.

Theorem B.3. *(Iterative scheme) Let $\boldsymbol{x}^* \in \mathbb{R}^n$ be an arbitrary vector and $\delta \in (0,1)$. There exist universal constants C_1, C_2, C_3, C_4 such that, when $\boldsymbol{x}_0 \in \mathbb{R}^n$ is in a ball around $\pm\boldsymbol{x}^*$ of radius $\left\|\boldsymbol{x}^*\right\|/6$, the sample size satisfies $2T\widetilde{m} \geq C_2 T \max\{n\log(n), \log(1/\delta)\}\sigma^2/\left\|\boldsymbol{x}^*\right\|^4$, the contamination level satisfies $\varepsilon/2 \leq C_1 \left\|\boldsymbol{x}^*\right\|^4/\sigma^2$, and Algorithm 4 is run on inputs $\boldsymbol{x}_0$, δ, ε, T and requests a sample of size $2\widetilde{m}$ from the robust phase retrieval model (with unknown mean noise) at each iteration, it returns $\boldsymbol{x}_T \in \mathbb{R}^n$ such that, with probability at least $1 - T\delta$,*

$$\begin{aligned}\frac{\operatorname{dist}(\boldsymbol{x}_T, \boldsymbol{x}^*)}{\left\|\boldsymbol{x}^*\right\|} &\leq C_3 \exp\left(-C_4 \eta T \left\|\boldsymbol{x}^*\right\|^2 (1-\sqrt{\varepsilon})\right) \\ &\quad + C_3 \frac{\sigma}{\left\|\boldsymbol{x}^*\right\|^2}\left(\sqrt{\frac{n\log(n)}{\widetilde{m}}} + \sqrt{\frac{\log(1/\delta)}{\widetilde{m}}}\right) + C_3 \frac{\sigma}{\left\|\boldsymbol{x}^*\right\|^2}\sqrt{\varepsilon}.\end{aligned}$$

The proof of Theorem B.3 is very similar to the proof of Theorem 3.3, which can be found in Section D, and hence we omit it.

Once again, the algorithmic framework for robust phase retrieval with unknown mean noise is to first run Algorithm 3 and pass its output to Algorithm 4. Finally, Theorem B.3 has the same guarantees (up to absolute constants) as Theorem 3.3. In conclusion, the robust phase retrieval problem does not become harder if the learner does not know that the mean of the underlying noise is zero.

C MISSING PROOFS FROM SECTION 2

C.1 Proof of Lemma 2.1

Proof. Although similar bounds (modulo constants) have appeared in literature before (see, for example, (Ma et al., 2019) Subsection 2.2), we give a proof for completeness: Let $\boldsymbol{h} = \boldsymbol{x} - \boldsymbol{x}^*$, with $\left\|\boldsymbol{h}\right\| \leq R\left\|\boldsymbol{x}^*\right\|$. According to Lemma E.1, the Hessian becomes:

$$\nabla^2 r(\boldsymbol{x}) = 6\boldsymbol{h}\boldsymbol{h}^\top + 6\boldsymbol{h}(\boldsymbol{x}^*)^\top + 6\boldsymbol{x}^*\boldsymbol{h}^\top + 4\boldsymbol{x}^*(\boldsymbol{x}^*)^\top + 3\left\|\boldsymbol{h}\right\|^2 I_n + 6(\boldsymbol{h}^\top \boldsymbol{x}^*) I_n + 2\left\|\boldsymbol{x}^*\right\|^2 I_n.$$

The largest eigenvalue of the Hessian is bounded by its operator norm:

$$\begin{aligned}\lambda_{\max}\left(\nabla^2 r(\boldsymbol{x})\right) &\leq \left\|\nabla^2 r(\boldsymbol{x})\right\|_{\mathrm{op}} \leq 9\left\|\boldsymbol{h}\right\|^2 + 18\left\|\boldsymbol{x}^*\right\|\left\|\boldsymbol{h}\right\| + 6\left\|\boldsymbol{x}^*\right\|^2 \\ &\leq (9R^2 + 18R + 6)\left\|\boldsymbol{x}^*\right\|^2,\end{aligned}$$

where for the first inequality we have used the triangle and Cauchy-Schwarz inequalities and the fact that for rank one matrices, $\left\|\boldsymbol{u}\boldsymbol{v}^\top\right\|_{\mathrm{op}} = \left\|\boldsymbol{u}\right\|\left\|\boldsymbol{v}\right\|$.

Next, we lower-bound $\lambda_{\min}\left(\nabla^2 r(\boldsymbol{x})\right)$ in the region $\|\boldsymbol{h}\| \le R\,\|\boldsymbol{x}^*\|$. For this, we proceed as follows:

$$\begin{aligned}(\boldsymbol{x}^*)^\top \nabla^2 r(\boldsymbol{x})\boldsymbol{x}^* &= 6(\boldsymbol{h}^\top \boldsymbol{x}^*)^2 + 18(\boldsymbol{h}^\top \boldsymbol{x}^*)\,\|\boldsymbol{x}^*\|^2 + 3\,\|\boldsymbol{x}^*\|^2\,\|\boldsymbol{h}\|^2 + 6\,\|\boldsymbol{x}^*\|^4 \\ &\ge \|\boldsymbol{x}^*\|^2\left(6\,\|\boldsymbol{x}^*\|^2 - 18\,\|\boldsymbol{h}\|\,\|\boldsymbol{x}^*\|\right) \\ &\ge \|\boldsymbol{x}^*\|^4\,(6-18R).\end{aligned}$$

where for the first inequality we have dropped two positive terms and use Cauchy-Schwarz. Thus,

$$\lambda_{\min}\left(\nabla^2 r(\boldsymbol{x})\right) \ge (6-18R)\,\|\boldsymbol{x}^*\|^2\,.$$

The conclusion follows for $R = \frac{1}{9}$. □

C.2 Proof of Lemma 2.2

Proof. This is similar to the proof of Theorem 1 in (Prasad et al., 2020), which we reproduce with a shorter proof. Let $\mathbf{err}_t = \boldsymbol{g}_t - \nabla r(\boldsymbol{x}_t)$, so that $\|\mathbf{err}_t\| \le A(m,\delta)\,\|\boldsymbol{x}_t - \boldsymbol{x}^*\| + B(m,\delta)$ with probability at least $1-\delta$. On this event,

$$\begin{aligned}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^*\| &= \|\boldsymbol{x}_t - \eta\boldsymbol{g}_t - \boldsymbol{x}^*\| = \|\boldsymbol{x}_t - \boldsymbol{x}^* - \eta\left(\nabla r(\boldsymbol{x}_t) - \nabla r(\boldsymbol{x}^*)\right) - \eta\mathbf{err}_t\| && (10)\\ &\le \|\boldsymbol{x}_t - \boldsymbol{x}^* - \eta\left(\nabla r(\boldsymbol{x}_t) - \nabla r(\boldsymbol{x}^*)\right)\| + \eta\,\|\mathbf{err}_t\| && (11)\\ &\le \sqrt{1 - \frac{2\eta\alpha\beta}{\alpha+\beta}}\,\|\boldsymbol{x}_t - \boldsymbol{x}^*\| + \eta\left(A(m,\delta,\varepsilon)\,\|\boldsymbol{x}_t - \boldsymbol{x}^*\| + B(m,\delta,\varepsilon)\right) && (12)\\ &= \left(\sqrt{1 - \frac{2\eta\alpha\beta}{\alpha+\beta}} + \eta A(m,\delta)\right)\|\boldsymbol{x}_t - \boldsymbol{x}^*\| + \eta B(m,\delta).\end{aligned}$$

In (10) we have used the update (4), $\nabla r(\boldsymbol{x}^*) = 0$, and the definition of $\mathbf{err}_t$. In (11), we have used the triangle inequality. In (12) we have used the fact that $\|\boldsymbol{x}_t - \boldsymbol{x}^*\| \le R$, in which region the population risk is α-strongly convex and β-smooth, hence the descent step is contractive (see, for instance, Theorem 3.12 in (Bubeck, 2015)), and the definition of the gradient estimator. □

C.3 Proof of Lemma 2.3

Proof. The proof is identical to the one of Lemma 2.1, except the expressions for the Hessian of the risk, which is now given by Lemma E.2. □

D MISSING PROOFS FROM SECTION 3

D.1 Proof of Theorem 3.1.

Proof. Recall that the noise is has mean 0. We split the proof into three parts: recovering the direction (Step 2 of Procedure 1 with configuration `StabMeanEst`) and the scale (Step 3) of $\boldsymbol{x}^*$, and finally obtaining guarantees for the output of Procedure 1.

Recovering the direction of $\boldsymbol{x}^*$: Let $\Sigma_i := \mathrm{Cov}\left(y(\boldsymbol{a}^\top\boldsymbol{e}_i)\boldsymbol{a}\right)$. By the guarantees of Proposition 2.1.1, for each $i \in [n]$ with probability at least $1 - \delta/(2n)$,

$$\left\|\boldsymbol{\mu}_i - \mathbb{E}\left[y(\boldsymbol{a}^\top\boldsymbol{e}_i)\boldsymbol{a}\right]\right\| = O\left(\sqrt{\frac{\mathrm{tr}\left(\Sigma_i\right)\log(n)}{m_0}} + \sqrt{\|\Sigma_i\|_{\mathrm{op}}\,\varepsilon} + \sqrt{\frac{\log(n/\delta)}{m_0}}\right), \tag{13}$$

where we have used $n\,\|A\|_{\mathrm{op}} \ge \mathrm{tr}\,(A)$ for an $n \times n$ matrix. We compute upper-bounds on $\mathrm{tr}\,(\Sigma_i)$ and $\|\Sigma_i\|_{\mathrm{op}}$. Developing the expression for the covariance and using lemma E.3 (3.), we arrive at

$$\begin{aligned}\Sigma_i = &\left(12(\boldsymbol{x}_i^*)^2\,\|\boldsymbol{x}^*\|^2 + 3\,\|\boldsymbol{x}^*\|^4\right) I_n + 5\,\|\boldsymbol{x}^*\|^4\,\boldsymbol{e}_i\boldsymbol{e}_i^\top + 22\boldsymbol{x}_i^*\,\|\boldsymbol{x}^*\|^4\left(\boldsymbol{e}_i(\boldsymbol{x}^*)^\top + \boldsymbol{x}^*\boldsymbol{e}_i^\top\right)\\ &+ \left(20(\boldsymbol{x}_i^*)^2 + 12\,\|\boldsymbol{x}^*\|^2\right)\boldsymbol{x}^*(\boldsymbol{x}^*)^\top + \sigma^2\left(I_n + \boldsymbol{e}_i\boldsymbol{e}_i^\top\right).\end{aligned}$$

Using $\operatorname{tr}\left(\boldsymbol{u}\boldsymbol{v}^\top\right) = \boldsymbol{u}^\top\boldsymbol{v}$, $\left\|\boldsymbol{u}\boldsymbol{v}^\top\right\|_{\mathrm{op}} = \|\boldsymbol{u}\|\,\|\boldsymbol{v}\|$, the triangle and Cauchy-Schwarz inequalities, we have:

$$\operatorname{tr}(\Sigma_i) = O\left(n(\|\boldsymbol{x}^*\|^4 + \sigma^2)\right),$$
$$\|\Sigma_i\|_{\mathrm{op}} = O\left(\|\boldsymbol{x}^*\|^4 + \sigma^2\right).$$

Substituting these into (13), for each $i \in [n]$, with probability at least $1 - \delta/(2n)$:

$$\left\|\boldsymbol{\mu}_i - \mathbb{E}\left[y(\boldsymbol{a}^\top\boldsymbol{e}_i)\boldsymbol{a}\right]\right\| = O\left(\|\boldsymbol{x}^*\|^2\sqrt{1+\sigma^2/\|\boldsymbol{x}^*\|^4}\left(\sqrt{\frac{n\log(n)}{m_0}}+\sqrt{\varepsilon}+\sqrt{\frac{\log(n/\delta)}{m_0}}\right)\right).$$

By a union bound, with probability at least $1-\delta$:

$$\sqrt{\sum_{i=1}^n\left\|\boldsymbol{\mu}_i-\mathbb{E}\left[y\boldsymbol{a}\boldsymbol{a}^\top\right]\boldsymbol{e}_i\right\|^2} = O\left(\|\boldsymbol{x}^*\|^2\sqrt{1+\frac{\sigma^2}{\|\boldsymbol{x}^*\|^4}}\left(\sqrt{\frac{n^2\log(n)}{m_0}}+\sqrt{\varepsilon n}+\sqrt{\frac{n\log(n/\delta)}{m_0}}\right)\right).$$

As $\boldsymbol{\mu}_i$ and $\mathbb{E}[y\boldsymbol{a}\boldsymbol{a}^\top]\boldsymbol{e}_i$ are the i'th column of $\boldsymbol{Y}$ and $\mathbb{E}[y\boldsymbol{a}\boldsymbol{a}^\top]$, respectively, using the definition of the Frobenius norm, the previous inequality becomes

$$\left\|\boldsymbol{Y} - \mathbb{E}\left[y\boldsymbol{a}\boldsymbol{a}^\top\right]\right\| = O\left(\|\boldsymbol{x}^*\|^2\sqrt{1+\frac{\sigma^2}{\|\boldsymbol{x}^*\|^4}}\left(\sqrt{\frac{n^2\log(n)}{m_0}}+\sqrt{\varepsilon n}+\sqrt{\frac{n\log(n/\delta)}{m_0}}\right)\right).$$

By Lemma E.3 (2.) and the fact that the Frobenius norm dominates the operator norm, with probability at least $1-\delta$,

$$\left\|\boldsymbol{Y}-\left(\|\boldsymbol{x}^*\|^2 I_n+2\boldsymbol{x}^*(\boldsymbol{x}^*)^\top\right)\right\|_{\mathrm{op}} = O\left(\|\boldsymbol{x}^*\|^2\sqrt{1+\frac{\sigma^2}{\|\boldsymbol{x}^*\|^4}}\left(\sqrt{\frac{n^2\log(n)}{m_0}}+\sqrt{\varepsilon n}+\sqrt{\frac{n\log(2n/\delta)}{m_0}}\right)\right).$$

As the operator norms of a matrix and its transpose are equal, an application of the triangle inequality implies that, with probability at least $1-2\delta$, the same upper-bound holds for $\left\|(\boldsymbol{Y}+\boldsymbol{Y}^\top)/2 - (\|\boldsymbol{x}^*\|^2 I_n + 2\boldsymbol{x}^*(\boldsymbol{x}^*)^\top)\right\|_{\mathrm{op}}$.

Next, we proceed as in the proof of Lemma 5 in (Ma et al., 2019). Let $\widetilde{\boldsymbol{x}}_0$ be an eigenvector of $\widetilde{\boldsymbol{Y}}$ for the eigenvalue $\lambda := \lambda_{\max}(\boldsymbol{Y})$ and with norm $\|\widetilde{\boldsymbol{x}}_0\| = 1$. Note that $-\widetilde{\boldsymbol{x}}_0$ has the same property. The eigenvalues of $\|\boldsymbol{x}^*\|^2 I_n + 2\boldsymbol{x}^*(\boldsymbol{x}^*)^\top$ are either $3\|\boldsymbol{x}^*\|^2$ or $\|\boldsymbol{x}^*\|^2$, corresponding to $\boldsymbol{x}^*/\|\boldsymbol{x}^*\|$ and a normal vector orthogonal to $\boldsymbol{x}^*$, respectively. As the trace of this matrix is $(n+2)\|\boldsymbol{x}^*\|^2$, it follows that exactly one is $3\|\boldsymbol{x}^*\|^2$ and the rest are $\|\boldsymbol{x}^*\|^2$. Applying Davis-Kahan's Theorem (see, for example, Theorem 4.5.5 in (Vershynin, 2018)), we obtain:

$$\begin{aligned}\operatorname{dist}\left(\|\boldsymbol{x}^*\|\,\widetilde{\boldsymbol{x}}, \boldsymbol{x}^*\right) &\le 2\sqrt{2}\,\|\boldsymbol{x}^*\|\,\frac{\left\|\frac{1}{2}\left(\boldsymbol{Y}+\boldsymbol{Y}^\top\right)-\left(\|\boldsymbol{x}^*\|^2 I_n+2\boldsymbol{x}^*(\boldsymbol{x}^*)^\top\right)\right\|_{\mathrm{op}}}{2\|\boldsymbol{x}^*\|^2}\\ &= O\left(\|\boldsymbol{x}^*\|\sqrt{1+\frac{\sigma^2}{\|\boldsymbol{x}^*\|^4}}\left(\sqrt{\frac{n^2\log(n)}{m_0}}+\sqrt{\varepsilon n}+\sqrt{\frac{n\log(n/\delta)}{m_0}}\right)\right).\end{aligned}$$

As $m_0 = \Omega((1+\sigma^2/\|\boldsymbol{x}^*\|^4)\max\{n^2\log(n), n\log(n/\delta)\})$ and $\varepsilon = O((\|\boldsymbol{x}^*\|^4/\sigma^2)n^{-1})$, the hidden constants can be chosen in such a way that

$$\operatorname{dist}\left(\|\boldsymbol{x}^*\|\,\widetilde{\boldsymbol{x}}, \boldsymbol{x}^*\right) \le \frac{1}{18}\|\boldsymbol{x}^*\|. \tag{14}$$

Recovering the scale of $\boldsymbol{x}^*$: Firstly, similar computations as in Lemma E.3 show that $\mathbb{E}[y] = \mathbb{E}\left[(\boldsymbol{a}^\top\boldsymbol{x}^*)^2\right] =$

$\|\boldsymbol{x}^*\|^2$ and $\mathrm{Var}(y) = 2\|\boldsymbol{x}^*\|^4 + \sigma^2$. By the guarantees of Proposition 2.2.1, with probability at least $1-\delta$:

$$\begin{aligned}\left|\widetilde{y}^2 - \|\boldsymbol{x}^*\|^2\right| &= O\left(\sqrt{\mathrm{Var}(y)}\left(\sqrt{\varepsilon} + \sqrt{\frac{\log(1/\delta)}{m_0}}\right)\right) \\ &= O\left(\|\boldsymbol{x}^*\|^2\sqrt{1+\frac{\sigma^2}{\|\boldsymbol{x}^*\|^4}}\left(\sqrt{\varepsilon} + \sqrt{\frac{\log(1/\delta)}{m_0}}\right)\right).\end{aligned}$$

The value of ε and the choice of m_0 guarantee that, in the previous line, the factor of $\|\boldsymbol{x}^*\|^2$ is less than 1. Then, an application of the reversed triangle inequality guarantees that $\widetilde{y}^2 > 0$, so that

$$\left|\|\boldsymbol{x}_0\|^2 - \|\boldsymbol{x}^*\|^2\right| = O\left(\|\boldsymbol{x}^*\|^2\sqrt{1+\frac{\sigma^2}{\|\boldsymbol{x}^*\|^4}}\left(\sqrt{\varepsilon} + \sqrt{\frac{\log(1/\delta)}{m_0}}\right)\right).$$

Next, as for any $a, b, c > 0$ with $b^2 > c$, $|a^2 - b^2| \le c$ implies $|a-b| \le b - \sqrt{b^2-c}$, we get to

$$|\|\boldsymbol{x}_0\| - \|\boldsymbol{x}^*\|| = O\left(\|\boldsymbol{x}^*\|\left(1 - \sqrt{1 - \sqrt{1+\frac{\sigma^2}{\|\boldsymbol{x}^*\|^4}}\left(\sqrt{\varepsilon} + \sqrt{\frac{\log(1/\delta)}{m_0}}\right)}\right)\right).$$

Once again, the value of ε and the choice of m_0 guarantee that

$$|\|\boldsymbol{x}_0\| - \|\boldsymbol{x}^*\|| \le \frac{1}{18}\|\boldsymbol{x}^*\|. \tag{15}$$

Combining the direction and the scale: Using the triangle inequality, together with (14) and (15), we have that with probability at least $1-2\delta$:

$$\begin{aligned}\mathrm{dist}(\boldsymbol{x}_0, \boldsymbol{x}^*) &\le \|\boldsymbol{x}_0 - \|\boldsymbol{x}^*\|\,\widetilde{\boldsymbol{x}}_0\| + \mathrm{dist}(\|\boldsymbol{x}^*\|\,\widetilde{\boldsymbol{x}}_0, \boldsymbol{x}^*) = |\|\boldsymbol{x}_0\| - \|\boldsymbol{x}^*\|| + \mathrm{dist}(\|\boldsymbol{x}^*\|\,\widetilde{\boldsymbol{x}}_0, \boldsymbol{x}^*) \\ &\le \frac{1}{9}\|\boldsymbol{x}^*\|.\end{aligned}$$

□

D.2 Proof of Theorem 3.2

Proof. We aim to apply Proposition 2.2.1. First of all, note that indeed the random variable $y\boldsymbol{a}$ has mean 0 and, under the assumption that the noise ε has bounded fourth moment, it holds that $\mathbb{E}[\|y\boldsymbol{a}\|^4] < \infty$.

Next, we give an upper-bound on κ_4. Recall its definition:

$$\kappa_4 = \sup_{\boldsymbol{v}\in\mathbb{R}^n, \boldsymbol{v}^\top\Sigma\boldsymbol{v}=1} \mathbb{E}[(\boldsymbol{v}^\top(y\boldsymbol{a}))^4]^{1/4},$$

where $\Sigma = \mathrm{Cov}(y\boldsymbol{a}) = (3\|\boldsymbol{x}^*\|^4 + \sigma^2)I_n + 12\|\boldsymbol{x}^*\|^2\boldsymbol{x}^*(\boldsymbol{x}^*)^\top$, using similar calculations as the ones in Lemma E.3. We begin by upper-bounding the expectation in the supremum. Using the inequality $(a+b)^4 \le 8(a^4+b^4)$,

$$\begin{aligned}\mathbb{E}[(\boldsymbol{v}^\top(y\boldsymbol{a}))^4] &= \mathbb{E}[(y^4\boldsymbol{v}^\top\boldsymbol{a})^4] = \mathbb{E}[((\boldsymbol{a}^\top\boldsymbol{x}^*)^2 + z)^4(\boldsymbol{v}^\top\boldsymbol{a})^4] \le 8\left(\mathbb{E}[(a^\top\boldsymbol{x}^*)^8(\boldsymbol{v}^\top\boldsymbol{a})^4] + K_4^4\mathbb{E}[(\boldsymbol{v}^\top\boldsymbol{a})^4]\right) \\ &= 35360\|\boldsymbol{x}^*\|^4(\boldsymbol{v}^\top\boldsymbol{x}^*)^4 + 47800\|\boldsymbol{x}^*\|^8\|v\|^4 + 24K_4^4\|v\|^4 \le (83160\|\boldsymbol{x}^*\|^8 + 24K_4^4)\|\boldsymbol{v}\|^4,\end{aligned}$$

where the quantities in the last equality can be obtained by performin similar computations to Lemma E.3, while the last inequality is due to Cauchy-Schwarz. Thus,

$$\kappa_4 \le (83160\|\boldsymbol{x}^*\|^8 + 24K_4^4)^{1/4} \sup_{\boldsymbol{v}\in\mathbb{R}^n, \boldsymbol{v}^\top\Sigma\boldsymbol{v}=1} \|v\|.$$

As $1 = \boldsymbol{v}^\top \Sigma \boldsymbol{v} = (3\|\boldsymbol{x}^*\|^4 + \sigma^2)\|\boldsymbol{v}\|^2 + 12\|\boldsymbol{x}^*\|^2 (\boldsymbol{v}^\top \boldsymbol{x}^*)^2$, we have, using also $(a+b)^{1/4} \le a^{1/4} + b^{1/4}$ for $a, b > 0$,

$$\kappa_4 \le \frac{16\|\boldsymbol{x}^*\|^2 + 3K_4}{\sqrt{3\|\boldsymbol{x}^*\|^4 + \sigma^2}} \le 16 + 3\frac{K_4}{\|\boldsymbol{x}^*\|^2}.$$

In particular, this imposes $\varepsilon = O(\|\boldsymbol{x}^*\|^8 / K_4^4)$, which is verified by the conditions of this Theorem. Using $r(\Sigma) \le n$, the same is true for the condition on m. Thus, applying Proposition 2.2.2 and $\sigma^2 \le K_4^2$, we have with probability $1-\delta$,

$$\|\boldsymbol{Y} - \mathrm{Cov}\,(y\boldsymbol{a})\|_{\mathrm{op}} = O\left(\|\boldsymbol{x}^*\|^4 \left(1 + \frac{K_4^4}{\|\boldsymbol{x}^*\|^8}\right)\left(\sqrt{\frac{n}{m}} + \sqrt{\varepsilon} + \sqrt{\frac{\log(1/\delta}{m}}\right)\right).$$

From this point, the proof is identical to the Proof of Theorem 3.1. In particular, although the eigenvalues of $\mathrm{Cov}\,(y\boldsymbol{a})$ depend on the unknown σ^2, the difference between them does not and hence Davis-Kahan's Theorem can be applied without issues. □

D.3 Proof of Theorem 3.3

Proof. We prove by induction that all iterates $(\boldsymbol{x}_t)_{t=0}^{T-1}$ remain in the ball centred at $\boldsymbol{x}^*$ with radius $\|\boldsymbol{x}^*\|/9$, which will allow us to derive guarantees for the last iterate.

The induction: We assume without loss of generaty that $\mathrm{dist}\,(\boldsymbol{x}_0, \boldsymbol{x}^*) = \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|$ (otherwise, the proof follows identically by replacing $\boldsymbol{x}^*$ with $-\boldsymbol{x}^*$). The base case follows from the conditions of the theorem. Also, by the reversed triangle inequality,

$$\|\boldsymbol{x}^*\| - \|\boldsymbol{x}_0\| \le \frac{1}{9}\|\boldsymbol{x}^*\| \quad \Longrightarrow \quad \|\boldsymbol{x}^*\|^2 \le \frac{81}{64}\|\boldsymbol{x}_0\|^2.$$

This guarantees that the step size satisfies $\eta = 128/(981\|\boldsymbol{x}_0\|^2) \le 18/(109\|\boldsymbol{x}^*\|^2) = 2/(\alpha+\beta)$.

Next, assume that for some $t \in \{0, 1, \dots, T-1\}$, $\|\boldsymbol{x}_t - \boldsymbol{x}^*\| \le \|\boldsymbol{x}^*\|/9$. According to Lemma 2.2, the next iterate will satisfy the inequality (7), in which we will now determine $A(\widetilde{m}, \delta, \varepsilon)$ and $B(\widetilde{m}, \delta, \varepsilon)$. Towards this objective, let $\Sigma = \mathrm{Var}\left(((\boldsymbol{a}^\top \boldsymbol{x}_t)^2 - y)(\boldsymbol{a}^\top \boldsymbol{x}_t)\boldsymbol{a}\right)$. According to Proposition 2.2.1, with probability at least $1-\delta$,

$$\|\boldsymbol{g}_t - \nabla r(\boldsymbol{x}_t)\| = O\left(\sqrt{\frac{\mathrm{tr}\,(\Sigma)\log(n)}{\widetilde{m}}} + \sqrt{\|\Sigma\|_{\mathrm{op}}\,\varepsilon} + \sqrt{\frac{\|\Sigma\|_{\mathrm{op}}\log(1/\delta)}{\widetilde{m}}}\right), \tag{16}$$

where we once again use $r(\Sigma) \le n$. This holds conditionally on $\boldsymbol{x}_t$, as a fresh sample was used for computing $\boldsymbol{g}_t$. By an application of the tower law, (16) holds in general with probability at least $1-\delta$. Using the bounds in Lemma E.3 (5.) and the inequality $\sqrt{a+b} \le \sqrt{a} + \sqrt{b}$, we have that, with probability at least $1-\delta$,

$$\|\boldsymbol{g}_t - \nabla r(\boldsymbol{x}_t)\| \le A(\widetilde{m}, \delta, \varepsilon)\|\boldsymbol{x}_t - \boldsymbol{x}^*\| + B(\widetilde{m}, \delta, \varepsilon),$$

where $A(\widetilde{m}, \delta, \varepsilon)$ and $B(\widetilde{m}, \delta, \varepsilon)$ are defined as

$$A(\widetilde{m}, \delta, \varepsilon) = O\left(\|\boldsymbol{x}^*\|^2\left(\sqrt{\frac{n\log(n)}{\widetilde{m}}} + \sqrt{\varepsilon} + \sqrt{\frac{\log(1/\delta)}{\widetilde{m}}}\right)\right),$$

$$B(\widetilde{m}, \delta, \varepsilon) = O\left(\sigma\|\boldsymbol{x}^*\|\left(\sqrt{\frac{n\log(n)}{\widetilde{m}}} + \sqrt{\varepsilon} + \sqrt{\frac{\log(1/\delta)}{\widetilde{m}}}\right)\right).$$

Thus, according to Lemma 2.2, with probability at least $1-\delta$,

$$\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^*\| \le \left(\sqrt{1 - \frac{2\eta\alpha\beta}{\alpha+\beta}} + \eta A(\widetilde{m}, \delta, \varepsilon)\right)\|\boldsymbol{x}_t - \boldsymbol{x}^*\| + \eta B(\widetilde{m}, \delta, \varepsilon). \tag{17}$$

Next, we show $\sqrt{1 - 2\eta\alpha\beta/(\alpha+\beta)} \le 87/100$, $\eta A(\widetilde{m}, \delta, \varepsilon) \le 3/100$, and $\eta B(\widetilde{m}, \delta, \varepsilon) \le \|\boldsymbol{x}^*\|/90$ for the chosen values of $\widetilde{m}$ and ε. Together with $\|\boldsymbol{x}_t - \boldsymbol{x}^*\| \le \|\boldsymbol{x}^*\|/9$ and (17), these will imply that $\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^*\| \le \|\boldsymbol{x}^*\|/9$ (with probability at least $1-\delta$).

- $\sqrt{1-2\eta\alpha\beta/(\alpha+\beta)} \leq 87/100$: this is a consequence of $\|\boldsymbol{x}_0\|^2 \leq 100\|\boldsymbol{x}^*\|^2/81$, which is obtained from $\|\boldsymbol{x}_t - \boldsymbol{x}^*\| \leq \|\boldsymbol{x}^*\|/9$ by an application of the reversed triangle inequality.
- $\eta A(\widetilde{m},\delta,\varepsilon) \leq 3/100$: This follows by noting that

$$\eta A(\widetilde{m},\delta,\varepsilon) \leq \frac{2}{\alpha+\beta}A(\widetilde{m},\delta,\varepsilon) = O\left(\sqrt{\frac{n\log(n)}{\widetilde{m}}} + \sqrt{\varepsilon} + \sqrt{\frac{\log(1/\delta)}{\widetilde{m}}}\right). \tag{18}$$

  Clearly, for $\widetilde{m} = \Omega(\max\{n\log(n), \log(1/\delta)\})$ and ε a small enough constant, which is satisfied by the initial constraint on ε, the right-hand side of (18) is at most $3/100$.
- $\eta B(\widetilde{m},\delta,\varepsilon) \leq \|\boldsymbol{x}^*\|/90$: As in the previous bullet point, this holds for $\varepsilon = O(\|\boldsymbol{x}^*\|^4/\sigma^2)$ and $\widetilde{m} = \Omega(\max\{n\log(n), \log(1/\delta)\}\sigma^2/\|\boldsymbol{x}^*\|^4)$.

This concludes the inductive part of the proof.

Guarantees for $\boldsymbol{x}_t$: We have proved that on an event with probability at least $1 - T\delta$, equation (17) holds for every $t \in \{0, 1, \ldots, T-1\}$. Using the inequality $\sqrt{1-a} \leq 1 - a/2$, valid for any $a \in [0,1]$, it follows that for every $t \in \{0, 1, \ldots, t-1\}$:

$$\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^*\| \leq \left(1 - \eta\left(\frac{\alpha\beta}{\alpha+\beta} - A(\widetilde{m},\delta,\varepsilon)\right)\right)\|\boldsymbol{x}_t - \boldsymbol{x}^*\| + \eta B(\widetilde{m},\delta,\varepsilon).$$

Iterating this over $t \in \{0, 1, \ldots, T-1\}$ and using $\|\boldsymbol{x}_0 - \boldsymbol{x}^*\| \leq \|\boldsymbol{x}^*\|/9$, we have

$$\begin{aligned}\|\boldsymbol{x}_T - \boldsymbol{x}^*\| &\leq \left(1-\eta\left(\frac{\alpha\beta}{\alpha+\beta} - A(\widetilde{m},\delta,\varepsilon)\right)\right)^T\|\boldsymbol{x}_0 - \boldsymbol{x}^*\| + \frac{\eta B(\widetilde{m},\delta,\varepsilon)}{1-\left(1-\eta\left(\frac{\alpha\beta}{\alpha+\beta} - A(\widetilde{m},\delta,\varepsilon)\right)\right)} \\ &\leq \frac{1}{9}\exp\left(-\eta T\left(\frac{\alpha\beta}{\alpha+\beta} - A(\widetilde{m},\delta,\varepsilon)\right)\right)\|\boldsymbol{x}^*\| + \frac{B(\widetilde{m},\delta,\varepsilon)}{\frac{\alpha\beta}{\alpha+\beta} - A(\widetilde{m},\delta,\varepsilon)}\end{aligned} \tag{19}$$

We bound each term of (19) separately. As $\alpha\beta/(\alpha+\beta) = \Theta(\|\boldsymbol{x}^*\|^2)$ and $A(\widetilde{m},\delta,\varepsilon) = O(\|\boldsymbol{x}^*\|^2\sqrt{\varepsilon})$ for $\widetilde{m} = O(\max\{n\log(n), \log(1/\delta)\})$, it holds that $\alpha\beta/(\alpha+\beta) - A(\widetilde{m},\delta,\varepsilon) = \Omega(\|\boldsymbol{x}^*\|^2(1-\varepsilon))$ and this gives that for some constants C_3' and C_4

$$\frac{1}{9}\exp\left(-\eta T\left(\frac{\alpha\beta}{\alpha+\beta} - A(\widetilde{m},\delta,\varepsilon)\right)\right)\|\boldsymbol{x}^*\| \leq C_3'\exp\left(-C_4\eta T\|\boldsymbol{x}^*\|^2(1-\sqrt{\varepsilon})\}\right)\|\boldsymbol{x}^*\|. \tag{20}$$

Further, when ε is a small enough constant, which is covered by the assumptions of our theorems, $A(\widetilde{m},\delta,\varepsilon) = O(\|\boldsymbol{x}^*\|^2)$, so that $\alpha\beta/(\alpha+\beta) - A(\widetilde{m},\delta,\varepsilon) = \Omega(\|\boldsymbol{x}^*\|^2)$. This leads to

$$\frac{B(\widetilde{m},\delta,\varepsilon)}{\frac{\alpha\beta}{\alpha+\beta} - A(\widetilde{m},\delta,\varepsilon)} \leq C_3''\frac{\sigma}{\|\boldsymbol{x}^*\|^2}\left(\sqrt{\frac{n\log(n)}{\widetilde{m}}} + \sqrt{\frac{\log(1/\delta)}{\widetilde{m}}} + \sqrt{\varepsilon}\right)\|\boldsymbol{x}^*\| \tag{21}$$

for some appropriate constant C_3''. Plugging (20) and (21) into (19), we arrive to the conclusion

$$\frac{\|\boldsymbol{x}_T - \boldsymbol{x}^*\|}{\|\boldsymbol{x}^*\|} \leq C_3\left(\exp\left(-C_4\eta T\|\boldsymbol{x}^*\|^2(1-\sqrt{\varepsilon})\right) + \frac{\sigma}{\|\boldsymbol{x}^*\|^2}\left(\sqrt{\frac{n\log(n)}{\widetilde{m}}} + \sqrt{\frac{\log(1/\delta)}{\widetilde{m}}} + \sqrt{\varepsilon}\right)\right).$$

□

E TECHNICAL LEMMAS

E.1 The gradient and Hessian of the population risk

Lemma E.1. *The expressions for the gradient and Hessian of the population risk (3) are given by*

$$\nabla r(\boldsymbol{x}) = 3\|\boldsymbol{x}\|^2\boldsymbol{x} - \|\boldsymbol{x}^*\|^2\boldsymbol{x} - 2((\boldsymbol{x}^*)^\top\boldsymbol{x})\boldsymbol{x}^*, \tag{22}$$

$$\nabla^2 r(\boldsymbol{x}) = 3\left(2\boldsymbol{x}\boldsymbol{x}^\top + \|\boldsymbol{x}\|^2 I_n\right) - \left(\|\boldsymbol{x}^*\|^2 I_n + 2\boldsymbol{x}^*(\boldsymbol{x}^*)^\top\right). \tag{23}$$

Proof. Fix $\boldsymbol{x}$ and $\boldsymbol{x}^*$. Exchanging differentiation and expectation, we have

$$\begin{aligned}\nabla r(\boldsymbol{x}) &= \mathbb{E}\left[\left((\boldsymbol{a}^\top\boldsymbol{x})^2 - y\right)(\boldsymbol{a}^\top\boldsymbol{x})\boldsymbol{a}\right] = \mathbb{E}\left[\left(\boldsymbol{a}^\top\boldsymbol{x})^2 - (\boldsymbol{a}^\top\boldsymbol{x}^*)^2 - z\right)(\boldsymbol{a}^\top\boldsymbol{x})\boldsymbol{a}\right] \\ &= \mathbb{E}\left[\left(\boldsymbol{a}^\top\boldsymbol{x})^2 - (\boldsymbol{a}^\top\boldsymbol{x}^*)^2\right)(\boldsymbol{a}^\top\boldsymbol{x})\boldsymbol{a}\right],\end{aligned}$$

where for the last line we have used the fact that $\mathbb{E}[z] = 0$ and z is independent of $\boldsymbol{a}$.

Using Lemma E.3 (1.), $\mathbb{E}\left[(\boldsymbol{a}^\top\boldsymbol{x}^*)^2(\boldsymbol{a}^\top\boldsymbol{x})\boldsymbol{a}\right] = 2((\boldsymbol{x}^*)^\top\boldsymbol{x})\boldsymbol{x}^* + \|\boldsymbol{x}^*\|^2\boldsymbol{x}$. Letting $\boldsymbol{x}^* := \boldsymbol{x}$, we also get $\mathbb{E}\left[(\boldsymbol{a}^\top\boldsymbol{x})^3\boldsymbol{a}\right] = 3\|\boldsymbol{x}\|^2\boldsymbol{x}$. Putting these two expectation together, we arrive at (22). Differentiating again, we obtain the expression of the Hessian (23). □

Lemma E.2. *The expressions for the gradient and Hessian of the population risk 8 are given by*

$$\begin{aligned}\nabla r_{\text{new}}(\boldsymbol{x}) &= 2\|\boldsymbol{x}\|^2\boldsymbol{x} - 2(\boldsymbol{x}^\top\boldsymbol{x}^*)\boldsymbol{x}, \\ \nabla^2 r_{\text{new}}(\boldsymbol{x}) &= 2\|\boldsymbol{x}^*\|^2 I_n + 4\boldsymbol{x}\boldsymbol{x}^\top - 2\boldsymbol{x}^*(\boldsymbol{x}^*)^\top.\end{aligned}$$

E.2 Quantities appearing in the proofs of Theorems 3.1, 3.2, and 3.3

Lemma E.3. *Let $\boldsymbol{x}$ and $\boldsymbol{x}^*$ be fixed vectors in $\mathbb{R}^n$, $\boldsymbol{a} \sim \mathcal{N}(\boldsymbol{0}, I_n)$, z a random variable with mean 0 and variance σ^2, and $y = (\boldsymbol{a}^\top\boldsymbol{x}^*)^2 + z$. Then:*

1. $\mathbb{E}\left[(\boldsymbol{a}^\top\boldsymbol{x}^*)^2(\boldsymbol{a}^\top\boldsymbol{x})\boldsymbol{a}\right] = 2((\boldsymbol{x}^*)^\top\boldsymbol{x})\boldsymbol{x}^* + \|\boldsymbol{x}^*\|^2\boldsymbol{x}$;
2. $\mathbb{E}\left[y\boldsymbol{a}\boldsymbol{a}^\top\right] = \|\boldsymbol{x}^*\|^2 I_n + 2\boldsymbol{x}^*(\boldsymbol{x}^*)^\top$;
3. $\mathbb{E}\left[(\boldsymbol{a}^\top\boldsymbol{x})^2(\boldsymbol{a}^\top\boldsymbol{x}^*)^4\boldsymbol{a}\boldsymbol{a}^\top\right] = v_1 I_n + v_2\boldsymbol{x}\boldsymbol{x}^\top + v_3\boldsymbol{x}^*\boldsymbol{x}^\top + v_4\boldsymbol{x}(\boldsymbol{x}^*)^\top + v_5\boldsymbol{x}^*(\boldsymbol{x}^*)^\top$, *where*
$$\begin{aligned}v_1 &= 12(\boldsymbol{x}^\top\boldsymbol{x}^*)^2\|\boldsymbol{x}^*\|^2 + 3\|\boldsymbol{x}^*\|^4\|\boldsymbol{x}\|^2, \\ v_2 &= 6\|\boldsymbol{x}^*\|^4, \\ v_3 &= 24(\boldsymbol{x}^\top\boldsymbol{x}^*)\|\boldsymbol{x}^*\|^2, \\ v_4 &= 24(\boldsymbol{x}^\top\boldsymbol{x}^*)\|\boldsymbol{x}^*\|^2, \\ v_5 &= 24(\boldsymbol{x}^\top\boldsymbol{x}^*)^2 + 12\|\boldsymbol{x}^*\|^2\|\boldsymbol{x}\|^2;\end{aligned}$$
4. *if $\Sigma = \mathrm{Var}\left(((\boldsymbol{a}^\top\boldsymbol{x})^2 - y)(\boldsymbol{a}^\top\boldsymbol{x})\boldsymbol{a})\right)$ is the variance of the loss gradient and $\|\boldsymbol{x} - \boldsymbol{x}^*\| \le 1/9$,*
$$\begin{aligned}\mathrm{tr}(\Sigma) &\le 525n\|\boldsymbol{x} - \boldsymbol{x}^*\|^2\|\boldsymbol{x}^*\|^4 + 6n\sigma^2\|\boldsymbol{x}^*\|^2, \\ \|\Sigma\|_{\text{op}} &\le 525\|\boldsymbol{x} - \boldsymbol{x}^*\|^2\|\boldsymbol{x}^*\|^4 + 6\sigma^2\|\boldsymbol{x}^*\|^2.\end{aligned}$$

Proof. 1. Let $A \in \mathbb{R}^{n\times n}$ be an orthogonal matrix ($AA^\top = A^\top A = I_n$) such that $A\boldsymbol{x}^* = \|\boldsymbol{x}^*\|\boldsymbol{e}_1$ and $A\boldsymbol{x} = \|\boldsymbol{x}\|(s_1\boldsymbol{e}_1 + s_2\boldsymbol{e}_2)$, where $s_1, s_2 \in \mathbb{R}$ satisfy $s_1^2 + s_2^2 = 1$. Also, note that $\boldsymbol{a}$ and $A\boldsymbol{a}$ are both distributed as $\mathcal{N}(\boldsymbol{0}, I_n)$. Then:

$$\begin{aligned}\mathbb{E}\left[(\boldsymbol{a}^\top\boldsymbol{x}^*)^2(\boldsymbol{a}^\top\boldsymbol{x})\boldsymbol{a}\right] &= \mathbb{E}\left[(\boldsymbol{a}^\top A^\top A\boldsymbol{x}^*)^2(\boldsymbol{a}^\top A^\top A\boldsymbol{x})A^\top A\boldsymbol{a}\right] \\ &= \|\boldsymbol{x}^*\|^2\|\boldsymbol{x}\|A^\top\mathbb{E}\left[(\boldsymbol{a}^\top\boldsymbol{e}_1)^2(\boldsymbol{a}^\top(s_1\boldsymbol{e}_1 + s_2\boldsymbol{e}_2))\boldsymbol{a}\right] \\ &= \|\boldsymbol{x}^*\|^2\|\boldsymbol{x}\|A^\top\mathbb{E}\left[a_1^2(s_1a_1 + s_2a_2)\boldsymbol{a}\right] \\ &= \|\boldsymbol{x}^*\|^2\|\boldsymbol{x}\|A^\top\mathbb{E}\begin{pmatrix} s_1a_1^4 + s_2a_1^3a_2 \\ s_1a_1^3a_2 + s_2a_1^2a_2^2 \\ s_1a_1^3a_3 + s_2a_1^2a_2a_3 \\ \cdots \\ s_1a_1^3a_n + s_2a_1^2a_2a_n\end{pmatrix} \\ &= \|\boldsymbol{x}^*\|^2\|\boldsymbol{x}\|A^\top(3s_1\boldsymbol{e}_1 + s_2\boldsymbol{e}_2) \\ &= \|\boldsymbol{x}^*\|^2\|\boldsymbol{x}\|A^\top(2s_1\boldsymbol{e}_1 + s_1\boldsymbol{e}_1 + s_2\boldsymbol{e}_2) \\ &= 2((\boldsymbol{x}^*)^\top\boldsymbol{x})\boldsymbol{x}^* + \|\boldsymbol{x}^*\|^2\boldsymbol{x},\end{aligned}$$

where for the last line we used $\boldsymbol{x}^* = \|\boldsymbol{x}^*\|A^\top\boldsymbol{e}_1$, $\boldsymbol{x} = \|\boldsymbol{x}\|A^\top(s_1\boldsymbol{e}_1 + s_2\boldsymbol{e}_2)$.

2. We apply the same technique as in the previous part. In particular, A is an orthonormal matrix with $A\boldsymbol{x}^* = \|\boldsymbol{x}^*\| \boldsymbol{e}_1$.

$$\begin{aligned}\mathbb{E}\left[y\boldsymbol{a}\boldsymbol{a}^\top\right] &= \mathbb{E}\left[\left((\boldsymbol{a}^\top\boldsymbol{x}^*)^2 + z\right)\boldsymbol{a}\boldsymbol{a}^\top\right] = \mathbb{E}\left[(\boldsymbol{a}^\top\boldsymbol{x}^*)^2\boldsymbol{a}\boldsymbol{a}^\top\right] = \|\boldsymbol{x}^*\|^2 A^\top \mathbb{E}\left[(\boldsymbol{a}^\top\boldsymbol{e}_1)^2\boldsymbol{a}\boldsymbol{a}^\top\right] A \\ &= \|\boldsymbol{x}^*\|^2 A^\top(I_n + 2\boldsymbol{e}_1\boldsymbol{e}_1^\top)A = \|\boldsymbol{x}^*\|^2 I_n + 2\boldsymbol{x}^*(\boldsymbol{x}^*)^\top.\end{aligned}$$

3. Again, consider an orthonormal matrix $A \in \mathbb{R}^{n\times n}$ such that $A\boldsymbol{x}^* = \|\boldsymbol{x}^*\| \boldsymbol{e}_1$ and $A\boldsymbol{x} = \|\boldsymbol{x}\| (s_1\boldsymbol{e}_1 + s_2\boldsymbol{e}_2)$, with $s_1, s_2 \in \mathbb{R}$, $s_1^2 + s_2^2 = 1$. Then, note that

$$\begin{aligned}&\mathbb{E}\left[(\boldsymbol{a}^\top\boldsymbol{x}^*)^2(\boldsymbol{a}^\top\boldsymbol{x}^*)^4\boldsymbol{a}\boldsymbol{a}^\top\right] \\ &= \|\boldsymbol{x}^*\|^4 \|\boldsymbol{x}\|^2 \mathbb{E}\left[(\boldsymbol{a}^\top(s_1\boldsymbol{e}_1 + s_2\boldsymbol{e}_2))^2(\boldsymbol{a}^\top\boldsymbol{e}_1)^4\boldsymbol{a}\boldsymbol{a}^\top\right] \\ &= \|\boldsymbol{x}^*\|^4 \|\boldsymbol{x}\|^2 \mathbb{E}\left[(s_1a_1 + s_2a_2)^2a_1^4\boldsymbol{a}\boldsymbol{a}^\top\right] \\ &= \|\boldsymbol{x}^*\|^4 \|\boldsymbol{x}\|^2 \begin{pmatrix} 105s_1^2 + 15s_2^2 & 30s_1s_2 & 0 & \cdots & 0 \\ 30s_1s_2 & 15s_1^2 + 9s_2^2 & 0 & \cdots & 0 \\ 0 & 0 & 15s_1^2 + 9s_2^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 15s_1^2 + 3s_2^2 \end{pmatrix} \\ &= \|\boldsymbol{x}^*\|^4 \|\boldsymbol{x}\|^2 ((15s_1^2 + 3s_2^2)I_n + 6(s_1\boldsymbol{e}_1 + s_2\boldsymbol{e}_2)(s_1\boldsymbol{e}_1 + s_2\boldsymbol{e}_2)^\top \\ &+ 24s_1\boldsymbol{e}_1(s_1\boldsymbol{e}_1 + s_2\boldsymbol{e}_2)^\top + 24s_1(s_1\boldsymbol{e}_1 + s_2\boldsymbol{e}_2)\boldsymbol{e}_1^\top + (36s_1^2 + 12s_2^2)\boldsymbol{e}_1\boldsymbol{e}_1^\top).\end{aligned}$$

The conclusion will follow by using $s_1^2 + s_2^2 = 1$, $s_1 = \boldsymbol{e}_1^\top(s_1\boldsymbol{e}_1 + s_2\boldsymbol{e}_2)$ and the inverse transformations $\boldsymbol{x}^* = \|\boldsymbol{x}^*\| A^\top\boldsymbol{e}_1$, $\boldsymbol{x} = \|\boldsymbol{x}\| A^\top(s_1\boldsymbol{e}_1 + s_2\boldsymbol{e}_2)$.

4. We start by computing Σ. Recall that the fact that the noise is symmetric with variance σ^2 and independent of the covariates:

$$\begin{aligned}\Sigma &= \mathrm{Var}\left(\left((\boldsymbol{a}^\top\boldsymbol{x})^2 - y\right)(\boldsymbol{a}^\top\boldsymbol{x})\boldsymbol{a}\right) \\ &= \mathbb{E}\left[\left((\boldsymbol{a}^\top\boldsymbol{x}_t)^2 - (\boldsymbol{a}^\top\boldsymbol{x}^*)^2 - z\right)^2(\boldsymbol{a}^\top\boldsymbol{x})^2\boldsymbol{a}\boldsymbol{a}^\top\right] - \nabla r(\boldsymbol{x})\nabla r(\boldsymbol{x})^\top \\ &= \mathbb{E}\left[(a^\top\boldsymbol{x})^6\boldsymbol{a}\boldsymbol{a}^\top\right] + \mathbb{E}\left[(\boldsymbol{a}^\top\boldsymbol{x}^*)^4(\boldsymbol{a}^\top\boldsymbol{x})^2\boldsymbol{a}\boldsymbol{a}^\top\right] + \sigma^2\mathbb{E}\left[(\boldsymbol{a}^\top\boldsymbol{x})^2\boldsymbol{a}\boldsymbol{a}^\top\right] \\ &- 2\mathbb{E}\left[(\boldsymbol{a}^\top\boldsymbol{x})^4(\boldsymbol{a}^\top\boldsymbol{x}^*)^2)\boldsymbol{a}\boldsymbol{a}^\top\right] - \nabla r(\boldsymbol{x})\nabla r(\boldsymbol{x})^\top.\end{aligned}$$

We can use the expression derived in part (3.) to arrive at expressions for the first, second and fourth term on the previous line. The last term has already been calculated in Lemma E.1, and the same computations as in the proof of (2.) give $\mathbb{E}\left[(\boldsymbol{a}^\top\boldsymbol{x}^*)^2\boldsymbol{a}\boldsymbol{a}^\top\right] = \|\boldsymbol{x}\|^2 I_n + 2\boldsymbol{x}\boldsymbol{x}^\top$. Putting all of these together,

$$\begin{aligned}\Sigma &= (15\|\boldsymbol{x}\|^6 + 12(\boldsymbol{x}^\top\boldsymbol{x}^*)^2\|\boldsymbol{x}^*\|^2 + 3\|\boldsymbol{x}^*\|^4\|\boldsymbol{x}\|^2 - 24(\boldsymbol{x}^\top\boldsymbol{x}^*)^2\|\boldsymbol{x}\|^2 - 6\|\boldsymbol{x}\|^4\|\boldsymbol{x}^*\|^2)I_n \\ &+ (91\|\boldsymbol{x}\|^4 + 5\|\boldsymbol{x}^*\|^4 - 48(\boldsymbol{x}^\top\boldsymbol{x}^*)^2 - 18\|\boldsymbol{x}\|^2\|\boldsymbol{x}^*\|^2)\boldsymbol{x}\boldsymbol{x}^\top \\ &+ (20(\boldsymbol{x}^\top\boldsymbol{x}^*)^2 + 12\|\boldsymbol{x}^*\|^2\|\boldsymbol{x}\|^2 - 12\|\boldsymbol{x}\|^4)\boldsymbol{x}^*(\boldsymbol{x}^*)^\top \\ &+ (22(\boldsymbol{x}^\top\boldsymbol{x}^*)\|\boldsymbol{x}^*\|^2 - 42(\boldsymbol{x}^\top\boldsymbol{x}^*)\|\boldsymbol{x}\|^2)(\boldsymbol{x}^*\boldsymbol{x}^\top + \boldsymbol{x}(\boldsymbol{x}^*)^\top) \\ &+ \sigma^2(\|\boldsymbol{x}\|^2 I_n + 2\boldsymbol{x}\boldsymbol{x}^\top).\end{aligned}$$

Next, the expression of Σ in terms of $\boldsymbol{x}^*$ and $\boldsymbol{h} := \boldsymbol{x} - \boldsymbol{x}^*$:

$$\Sigma = w_1I_1 + w_2\boldsymbol{x}^*(\boldsymbol{x}^*)^\top + w_3\boldsymbol{h}\boldsymbol{h}^\top + w_4(\boldsymbol{x}^*\boldsymbol{h}^\top + \boldsymbol{h}(\boldsymbol{x}^*)^\top) + \sigma^2A,$$

where the expressions of $w_1, w_2, w_3, w_4 \in \mathbb{R}$ and $A \in \mathbb{R}^{n\times n}$ are

$$\begin{aligned}
w_1 &= 15\|\boldsymbol{h}\|^6 + 72(\boldsymbol{h}^\top\boldsymbol{x}^*)^3 + 90\|\boldsymbol{h}\|^4(\boldsymbol{h}^\top\boldsymbol{x}^*) + 156\|\boldsymbol{h}\|^2(\boldsymbol{h}^\top\boldsymbol{x}^*)^2 + 39\|\boldsymbol{h}\|^4\|\boldsymbol{x}^*\|^2 \\
&\quad + 12\|\boldsymbol{h}\|^2\|\boldsymbol{x}^*\|^4 + 48\|\boldsymbol{x}^*\|^2(\boldsymbol{h}^\top\boldsymbol{x}^*)^2 + 108\|\boldsymbol{h}\|^2\|\boldsymbol{x}^*\|^2(\boldsymbol{h}^\top\boldsymbol{x}^*), \\
w_2 &= 69\|\boldsymbol{h}\|^4 + 80(\boldsymbol{h}^\top\boldsymbol{x}^*)^2 + 192(\boldsymbol{h}^\top\boldsymbol{x}^*)\|\boldsymbol{h}\|^2 + 48\|\boldsymbol{x}^*\|^2\|\boldsymbol{h}\|^2, \\
w_3 &= 81\|\boldsymbol{h}\|^4 + 276(\boldsymbol{h}^\top\boldsymbol{x}^*)^2 + 20\|\boldsymbol{x}^*\|^4 + 324(\boldsymbol{h}^\top\boldsymbol{x}^*)\|\boldsymbol{h}\|^2 + 192(\boldsymbol{h}^\top\boldsymbol{x}^*)\|\boldsymbol{x}^*\|^2 \\
&\quad + 144\|\boldsymbol{h}\|^2\|\boldsymbol{x}^*\|^2, \\
w_4 &= 81\|\boldsymbol{h}\|^4 + 192(\boldsymbol{h}^\top\boldsymbol{x}^*)^2 + 88(\boldsymbol{h}^\top\boldsymbol{x}^*)\|\boldsymbol{x}^*\|^2 + 282(\boldsymbol{h}^\top\boldsymbol{x}^*)\|\boldsymbol{h}\|^2 + 102\|\boldsymbol{h}\|^2\|\boldsymbol{x}^*\|^2, \\
A &= (\|\boldsymbol{x}^*\|^2 + 2(\boldsymbol{h}^\top\boldsymbol{x}^*) + \|\boldsymbol{h}\|^2)I_n + 2\boldsymbol{x}^*(\boldsymbol{x}^*)^\top + 2\boldsymbol{h}\boldsymbol{h}^\top + 2(\boldsymbol{x}^*\boldsymbol{h}^\top + \boldsymbol{h}(\boldsymbol{x}^*)^\top).
\end{aligned}$$

To compute an upper-bound on $\operatorname{tr}(\Sigma)$ we use the property $\operatorname{tr}\left(\boldsymbol{u}\boldsymbol{v}^\top\right) = \boldsymbol{u}^\top\boldsymbol{v}$, the Cauchy-Schwarz inequality and the assumption $\|\boldsymbol{h}\| \le \|\boldsymbol{x}^*\|/9$:

$$\begin{aligned}
\operatorname{tr}(\Sigma) &= (81+15n)\|\boldsymbol{h}\|^6 + (384+72n)(\boldsymbol{h}^\top\boldsymbol{x}^*)^3 + (486+90n)\|\boldsymbol{h}\|^4(\boldsymbol{h}^\top\boldsymbol{x}^*) \\
&\quad + (840+156n)\|\boldsymbol{h}\|^2(\boldsymbol{h}^\top\boldsymbol{x}^*)^2 + (213+39n)\|\boldsymbol{h}\|^4\|\boldsymbol{x}^*\|^2 + (68+12n)\|\boldsymbol{h}\|^2\|\boldsymbol{x}^*\|^4 \\
&\quad + (256+48n)(\boldsymbol{h}^\top\boldsymbol{x}^*)^2\|\boldsymbol{x}^*\|^2 + (588+108n)(\boldsymbol{h}^\top\boldsymbol{x}^*)\|\boldsymbol{h}\|^2\|\boldsymbol{x}^*\|^2 \\
&\quad + (n+2)\sigma^2(\|\boldsymbol{h}\|^2 + 2\boldsymbol{h}^\top\boldsymbol{x}^* + \|\boldsymbol{x}^*\|^2) \\
&\le (81+15n)\|\boldsymbol{h}\|^6 + (486+90n)\|\boldsymbol{h}\|^5\|\boldsymbol{x}^*\| + (1053+195n)\|\boldsymbol{h}\|^4\|\boldsymbol{x}^*\|^2 \\
&\quad + (972+180n)\|\boldsymbol{h}\|^3\|\boldsymbol{x}^*\|^3 + (324+60n)\|\boldsymbol{h}\|^2\|\boldsymbol{x}^*\|^4 \\
&\quad + (n+2)\sigma^2(\|\boldsymbol{h}\|^2 + 2\|\boldsymbol{h}\|\|\boldsymbol{x}^*\| + \|\boldsymbol{x}^*\|^2) \\
&\le \left(\frac{81+15n}{9^4} + \frac{486+90n}{9^3} + \frac{1053+195n}{9^2} + \frac{972+180n}{9} + 324 + 60n\right)\|\boldsymbol{h}\|^2\|\boldsymbol{x}^*\|^4 \\
&\quad + (n+2)\sigma^2\left(\frac{1}{81} + \frac{2}{9} + 1\right)\|\boldsymbol{x}^*\|^2 \\
&\le (442+83n)\|\boldsymbol{h}\|^2\|\boldsymbol{x}^*\|^4 + 2(n+2)\sigma^2\|\boldsymbol{x}^*\|^2 \\
&\le 525n\|\boldsymbol{h}\|^2\|\boldsymbol{x}^*\|^4 + 6n\sigma^2\|\boldsymbol{x}^*\|^2.
\end{aligned}$$

Taking the operator norm ($\left\|\boldsymbol{u}\boldsymbol{v}^\top\right\|_{\mathrm{op}} = \|\boldsymbol{u}\|_{\mathrm{op}}\|\boldsymbol{v}\|_{\mathrm{op}}$), using once again the Cauchy-Schwarz inequality and the assumption $\|\boldsymbol{h}\| \le \|\boldsymbol{x}^*\|/9$, we similarly arrive at

$$\|\Sigma\|_{\mathrm{op}} \le 525\|\boldsymbol{h}\|^2\|\boldsymbol{x}^*\|^4 + 6\sigma^2\|\boldsymbol{x}^*\|^2.$$

□

Lemma E.4. *Let $\boldsymbol{x}$ and $\boldsymbol{x}^*$ be fixed vectors in $\mathbb{R}^n$, $\boldsymbol{b}, \boldsymbol{c} \sim \mathcal{N}(\boldsymbol{0}, I_n)$ independently, ζ a random variable with mean 0 and variance $\sigma^2/2$, and $\upsilon = (\boldsymbol{x}^*)^\top\boldsymbol{b}\boldsymbol{c}^\top\boldsymbol{x}^* + \zeta$. Then:*

1. $\mathbb{E}\left[\upsilon\boldsymbol{b}\boldsymbol{c}^\top\right] = \boldsymbol{x}^*(\boldsymbol{x}^*)^\top$;
2. $\operatorname{Cov}(\upsilon(\boldsymbol{c}^\top\boldsymbol{e}_i)\boldsymbol{b}) = (\|\boldsymbol{x}^*\|^4 + 2(\boldsymbol{x}_1^*)^2\|\boldsymbol{x}^*\|^2)I_n + (2\|\boldsymbol{x}^*\|^2 + 3(\boldsymbol{x}_1^*)^2)\boldsymbol{x}^*(\boldsymbol{x}^*)^\top + \sigma^2 I_n/2$;
3. *if* $\Sigma = \operatorname{Var}\left((\boldsymbol{x}^\top\boldsymbol{b}\boldsymbol{c}^\top\boldsymbol{x} - \upsilon)(\boldsymbol{b}\boldsymbol{c}^\top + \boldsymbol{c}\boldsymbol{b}^\top)\boldsymbol{x}\right)$ *is the variance of the loss gradient, then*

$$\begin{aligned}
\Sigma &= (6\|\boldsymbol{x}\|^6 + 2\|\boldsymbol{x}^*\|^4\|\boldsymbol{x}\|^2 + 4(\boldsymbol{x}^\top\boldsymbol{x}^*)^2\|\boldsymbol{x}^*\|^2 - 12(\boldsymbol{x}^\top\boldsymbol{x}^*)^2\|\boldsymbol{x}\|^2)I_n \\
&\quad + (26\|\boldsymbol{x}\|^4 + 2\|\boldsymbol{x}^*\|^4 - 16(\boldsymbol{x}^\top\boldsymbol{x}^*)^2)\boldsymbol{x}\boldsymbol{x}^\top \\
&\quad + (12(\boldsymbol{x}^\top\boldsymbol{x}^*)^2 + 4\|\boldsymbol{x}^*\|^2\|\boldsymbol{x}\|^2 - 4\|\boldsymbol{x}\|^4)\boldsymbol{x}^*(\boldsymbol{x}^*)^\top \\
&\quad + (4(\boldsymbol{x}^\top\boldsymbol{x}^*)\|\boldsymbol{x}^*\|^2 - 16(\boldsymbol{x}^\top\boldsymbol{x}^*)\|\boldsymbol{x}\|^2)(\boldsymbol{x}^*\boldsymbol{x}^\top + \boldsymbol{x}(\boldsymbol{x}^*)^\top) \\
&\quad + \sigma^2(\|\boldsymbol{x}\|^2 I_n + \boldsymbol{x}\boldsymbol{x}^\top).
\end{aligned}$$

In particular, if $\|\boldsymbol{x} - \boldsymbol{x}^*\| \leq 1/6$,

$$\operatorname{tr}(\Sigma) \leq 242n \|\boldsymbol{x} - \boldsymbol{x}^*\|^2 \|\boldsymbol{x}^*\|^4 + 3n\sigma^2 \|\boldsymbol{x}^*\|^2,$$
$$\|\Sigma\|_{\mathrm{op}} \leq 242 \|\boldsymbol{x} - \boldsymbol{x}^*\|^2 \|\boldsymbol{x}^*\|^4 + 3\sigma^2 \|\boldsymbol{x}^*\|^2.$$

Proof. The proof of these results is identical to the one of Lemma E.3. □