Faster Algorithms for High-Dimensional Robust Covariance Estimation

Yu Cheng[*] Ilias Diakonikolas[†] Rong Ge[‡] David P. Woodruff[§]

Abstract

We study the problem of estimating the covariance matrix of a high-dimensional distribution when a small constant fraction of the samples can be arbitrarily corrupted. Recent work gave the first polynomial time algorithms for this problem with near-optimal error guarantees for several natural structured distributions. Our main contribution is to develop faster algorithms for this problem whose running time nearly matches that of computing the empirical covariance.

Given $N = \widetilde{\Omega}(d^2/\epsilon^2)$ samples from a d-dimensional Gaussian distribution, an ϵ-fraction of which may be arbitrarily corrupted, our algorithm runs in time $\widetilde{O}(d^{3.26})/\operatorname{poly}(\epsilon)$ and approximates the unknown covariance matrix to optimal error up to a logarithmic factor. Previous robust algorithms with comparable error guarantees all have runtimes $\widetilde{\Omega}(d^{2\omega})$ when $\epsilon = \Omega(1)$, where ω is the exponent of matrix multiplication. We also provide evidence that improving the running time of our algorithm may require new algorithmic techniques.

1 Introduction

Estimating the covariance matrix of a high-dimensional distribution (covariance estimation) is one of the most fundamental statistical tasks (see, e.g., [BL08a, BL08b] and references therein). For a range of well-behaved distribution families, the empirical covariance matrix is known to converge to the true covariance matrix at an optimal statistical rate (with respect to various norms). For concreteness, suppose we are given N independent samples from a centered Gaussian $\mathcal{N}(\mathbf{0}, \Sigma)$ on $\mathbb{R}^d$, with unknown covariance Σ, and we want to estimate Σ with respect to the Frobenius norm. It is well-known (see, e.g., Section 4 of [CZZ10] for an explicit reference) that the empirical covariance matrix has expected Frobenius error at most $O(d/\sqrt{N}) \cdot \|\Sigma\|_2$ from Σ, where $\|\cdot\|_2$ denotes the spectral norm; and this bound is the best possible, within a constant factor, among all N-sample estimators. Equivalently, after $N = \Omega(d^2/\epsilon^2)$ samples, the empirical covariance will have Frobenius error at most $\epsilon \cdot \|\Sigma\|_2$ with high constant probability. This gives a computationally and statistically efficient covariance estimator for this fundamental setting. (By Lemma 2.2, the empirical covariance can be computed in time $O(d^{3.26}/\epsilon^2)$, which is the best known bound to date.)

In this paper, we study the outlier robust setting when a small constant fraction of our samples can be arbitrarily corrupted. We work in the following model of corruptions (see, e.g., [DKK$^+$16]) that generalizes Huber's contamination model ([Hub64]):

[*] Duke University. Email: yucheng@cs.duke.edu
[†] University of Southern California. Email: diakonik@usc.edu
[‡] Duke University. Email: rongge@cs.duke.edu
[§] Carnegie Mellon University. Email: dwoodruf@cs.cmu.edu

Definition 1.1 (ϵ-Corruption)**.** *Given $\epsilon > 0$ and a distribution family $\mathcal{D}$ on $\mathbb{R}^d$, the* adversary *operates as follows: The algorithm specifies some number of samples N, and N samples $X_1, X_2, \ldots, X_N$ are drawn from some (unknown) $D \in \mathcal{D}$. The adversary is allowed to inspect the samples, removes ϵN of them, and replaces them with arbitrary points. This set of N points is then given to the algorithm. We say that a set of samples is* ϵ-corrupted *if it is generated by the above process.*

More concretely, we study the following problem: Given an ϵ-corrupted set of N samples from an unknown $\mathcal{N}(\mathbf{0}, \Sigma)$ on $\mathbb{R}^d$, we want to compute an accurate estimate of Σ in Frobenius norm (or in a stronger, affine invariant version that guarantees small total variation distance). Note that even a single corrupted point can arbitrarily compromise the behavior of the empirical covariance matrix. Classical and more recent work in statistics has obtained minimax optimal robust covariance estimators. For example, [Rou85] proposed the minimum volume ellipsoid – a natural generalization of the interquartile range – and showed that it is provably robust in high-dimensions. More recently, [CGR18] proposes a (similarly robust) generalization of Tukey's median ([Tuk75]) for the covariance matrix. We note that the information-theoretically optimal error for robustly estimating the covariance of $\mathcal{N}(\mathbf{0}, \Sigma)$ in Frobenius norm is $O(\epsilon + d/\sqrt{N}) \cdot \|\Sigma\|_2$. That is, for $N = \Omega(d^2/\epsilon^2)$, one can estimate the covariance to accuracy $\Theta(\epsilon) \cdot \|\Sigma\|_2$, which is almost as well as in the non-contaminated setting. Unfortunately, these estimators are hard to compute in general, i.e., their runtime scales exponentially with the dimension.

Recent work in TCS ([DKK$^+$16, LRV16]) gave the first polynomial time robust estimators for a range of high-dimensional statistical tasks, including mean and covariance estimation. Since these initial papers ([DKK$^+$16, LRV16]), a growing body of subsequent works have obtained polynomial-time robust learning algorithms for a variety of unsupervised and supervised high-dimensional models. (See Section 1.3 for more related work.)

It should be noted that the aforementioned robust estimators have already been useful in exploratory data analysis. Specifically, [DKK$^+$17] evaluated the robust covariance estimators of [DKK$^+$16] and [LRV16] to detect patterns in a well-known genetic dataset ([NJB$^+$08]) in the presence of corruptions. Perhaps surprisingly, it was found that the robust algorithms developed for $\mathcal{N}(\mathbf{0}, \Sigma)$ outperformed all previous approaches on this real dataset, essentially matching the setting where there are no corruptions at all.

Once a polynomial-time algorithm for a computational problem has been discovered, the next step is to focus on designing asymptotically faster algorithms for the problem – with linear time as the ultimate goal. We note that the aforementioned robust estimators ([DKK$^+$16, LRV16]) are significantly slower than their non-robust counterparts (e.g., computing the empirical mean/covariance), hence may not be scalable when the dimension is very high. This raises the following natural question:

Can we design robust estimators that are as efficient as their non-robust analogues?

This direction was initiated in [CDG19] who gave a robust mean estimation algorithm with runtime $\widetilde{O}(Nd)/\operatorname{poly}(\epsilon)$,[1] nearly matching the runtime of computing the empirical mean (when ϵ is constant).

In this work, we continue this line of investigation. At a high-level, our main contribution is the first robust covariance estimator whose running time nearly matches that of computing the empirical covariance matrix. Moreover, we provide evidence that the runtime of our algorithm may not be improvable with current algorithmic techniques. In more detail, on input an ϵ-corrupted set of

[1] The $\widetilde{O}(\cdot)$ notation hides logarithmic factors in its argument.

$N = \widetilde{O}(d^2/\epsilon^2)$ samples from $\mathcal{N}(\mathbf{0}, \Sigma)$ on $\mathbb{R}^d$, our algorithm runs in time $\widetilde{O}(d^{3.26})/\operatorname{poly}(\epsilon)$, and outputs a covariance estimate with near-optimal error guarantee, matching the one in [DKK$^+$16] (see Theorem 1.2). Our algorithm uses the primal-dual framework of [CDG19] recently developed for robust mean estimation, with a number of crucial twists that are required for the more challenging task of covariance estimation (see Section 1.2).

For the sake of direct comparison, we note that the filtering-based robust covariance estimator of [DKK$^+$16] has runtime $\Omega(N^2 d) = \Omega(d^5)/\operatorname{poly}(\epsilon)$. On the other hand, the recursive dimension-halving estimator of [LRV16] requires $\Omega(\log d)$ SVD computations of a $d^2 \times d^2$ "covariance" matrix, hence has runtime $\Omega(d^{2\omega})$, where ω is the exponent of matrix multiplication. (Plugging in the best-known value for ω ([Gal14]) gives a runtime of $\Omega(d^{4.74})$.)

We note that the runtime of our algorithm, while being super-linear, essentially matches the best-known runtime to compute the empirical covariance matrix (see Section 6.1). Moreover, we provide evidence (Section 6.2) that this runtime may be a bottleneck even for the weaker task of obtaining an implicit representation to the output (by reweighing the input samples). It should be noted that all known computationally efficient robust estimators fit in this framework.

1.1 Our Results

Our first algorithmic result states that we can robustly estimate the covariance matrix of a high-dimensional Gaussian within multiplicative, dimension-independent error, with running time that almost matches that of computing the empirical covariance matrix.

Theorem 1.2 (Robust Covariance Estimation (Multiplicative))**.** *Let $D \sim \mathcal{N}(\mathbf{0}, \Sigma)$ be a zero-mean unknown covariance Gaussian on $\mathbb{R}^d$. Let κ denote the condition number of Σ. Let $0 < \epsilon < \epsilon_0$ for some universal constant ϵ_0. Given as input an ϵ-corrupted set of $N = \widetilde{\Omega}(d^2/\epsilon^2)$ samples drawn from D, there is an algorithm (Algorithm 1) that runs in time $\widetilde{O}(d^{3.26}\log(\kappa))/\operatorname{poly}(\epsilon)$ and outputs $\widehat{\Sigma} \in \mathbb{R}^{d\times d}$ such that with high probability it holds $\|\Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2} - I\|_F \leq O(\epsilon\log(1/\epsilon))$.*

We also develop a related robust covariance estimation algorithm (Algorithm 3) with additive error guarantee, whose running time does not depend on the condition number of Σ.

Theorem 1.3 (Robust Covariance Estimation (Additive))**.** *For the same setting as in Theorem 1.2, there is an algorithm (Algorithm 3) that runs in time $\widetilde{O}(d^{3.26})/\operatorname{poly}(\epsilon)$ and outputs $\widehat{\Sigma} \in \mathbb{R}^{d\times d}$ such that with high probability it holds $\|\widehat{\Sigma} - \Sigma\|_F \leq O(\epsilon\log(1/\epsilon))\,\|\Sigma\|_2$.*

We will prove Theorem 1.2 in Section 3.1 and Theorem 1.3 in Appendix B.1.

In Section 6, we provide evidence that the runtime of our algorithm may be difficult to improve. Specifically, we show that the best-known runtime for computing (or even approximating) the empirical covariance is $\Omega(d^{3.25})$. Moreover, even outputting a set of weights for the samples (such that the weighted empirical covariance works) seems to require $\Omega(d^{3.25})$ time with current methods.

1.2 Our Approach and Techniques

When $X \sim \mathcal{N}(0, \Sigma)$, we have $\mathbb{E}\big[XX^\top\big] = \Sigma$, so at a high level, we want to reduce the robust covariance estimation problem to the problem of robustly estimating the mean of the d^2-dimensional random variable $Z = X \otimes X$. However, there are two main difficulties in this reduction.

Faster Positive SDP Solvers for Tensor Input. The first difficulty is that the input of the mean estimation problem is now a set of d^2-dimensional vectors. Even just computing all of these vectors explicitly will take time $\Omega(Nd^2)$. Our algorithm needs to solve the robust mean estimation problem for $X \otimes X$ *without computing these vectors explicitly.* To achieve that, we adapt the approach in [CDG19]. Given data points $Z_i = X_i \otimes X_i$, the algorithm in [CDG19] starts with a guess $\nu \in \mathbb{R}^{d^2}$, and approximately solves the following two (dual) SDPs at every iteration:

$$\begin{aligned} &\text{minimize} && \lambda_{\max}\left(\textstyle\sum_{i=1}^N w_i (Z_i - \nu)(Z_i - \nu)^\top\right) \\ &\text{subject to} && \textstyle\sum_{i=1}^N w_i = 1, \quad \forall i, 0 \le w_i \le \frac{1}{(1-\epsilon)N} \end{aligned} \tag{1}$$

$$\begin{aligned} &\text{maximize} && \text{average of the smallest } (1-\epsilon)\text{-fraction of } \left((Z_i - \nu)^\top M (Z_i - \nu)\right)_{i=1}^N \\ &\text{subject to} && M \succeq 0, \operatorname{tr}(M) \le 1 \end{aligned} \tag{2}$$

[CDG19] showed the following win-win phenomenon: If the primal SDP (1) has a good solution, then it gives weights $w \in \mathbb{R}^N$ such that the weighted average $\sum_{i=1}^N w_i Z_i$ is close to the true mean. Otherwise, if (1) does not have a good solution, then the dual SDP (2) gives a direction of improvement that allows the algorithm to find ν' that is much closer to the true mean. In our setting of covariance estimation, writing out the matrix of $(Z_i)_{i=1}^N$ already takes $O(Nd^2)$ time. We circumvent this problem by "opening up" the fast positive SDP solver ([PTZ16]) they used. The slowest step of this SDP solver is to compute multiplicative approximations of the values $\exp(\Psi) \bullet Z_i Z_i^\top$ for all i, where $\Psi = \sum_{i=1}^N w_i Z_i Z_i^\top$. We exploit the structure of $Z_i = X_i \otimes X_i$ and design faster algorithms for getting such approximations. More precisely, let $\overline{Z}$ be a $N \times d$ matrix whose i-th column is $\sqrt{w_i} Z_i$, so that $\Psi = \overline{Z}\overline{Z}^\top$. We express $\exp(\Psi)$ as a low-degree polynomial over $\overline{Z}$ and $\overline{Z}^\top$, then use fast matrix multiplication (Lemma 2.2) and the transposition principle (Lemma 2.3) to show it is possible to right-multiply a vector with $\overline{Z}$ and $\overline{Z}^\top$ efficiently.

Iterative Refinement. The second difficulty is that existing robust mean estimation algorithms rely on the assumption that the distribution of the good data either has known covariance or unknown bounded covariance. By reducing covariance estimation to the mean estimation of $X \otimes X$, we run into the difficulty that the covariance of such vectors would correspond to the fourth order moments of the original variables X. So, directly applying the mean estimation algorithms does not give our desired strong guarantees.

We solve this problem using iterative refinement steps. Similar iterative refinement steps were discussed in [Kan18], which does not seem sufficient for our purposes, as it may require a linear number of iterations. We adapt this approach to obtain faster algorithms. More precisely, given an upper bound Σ_t on the true covariance matrix Σ, we can compute a more accurate upper bound $\Sigma_{t+1} \succeq \Sigma$. We use two different types of iterative refinement steps (see Lemmas 3.2 and 3.3). Both refinement steps first rotate the input samples $Y_i = \Sigma_t^{-1/2} X_i$ and compute the Kronecker products $Z_i = Y_i \otimes Y_i$. When $X \sim \mathcal{N}(0, \Sigma)$, the covariance matrix of Y is $\Sigma_t^{-1/2} \Sigma \Sigma_t^{-1/2}$, and the mean of Z is $\mathbb{E}[Z] = \left(\Sigma_t^{-1/2} \Sigma \Sigma_t^{-1/2}\right)^\flat$. Note that if we had $\mathbb{E}[Z]$, we could recover Σ from Σ_t immediately. We show that a good estimate of $\mathbb{E}[Z]$ will help us improve Σ_t and get a better upper bound Σ_{t+1}.

We establish two guarantees for the robust estimation of the mean of Z. In the first phase, we only use the fact that the covariance of Z is bounded. Repeating this will give a rough estimation of the covariance of X. In the second phase, we use the fact that the current estimate Σ_t is already very close to Σ, therefore $Y_i = \Sigma_t^{-1/2} X_i$ is very close to a standard Gaussian $N(0, I)$ for the good samples. In this case, we need to open up the algorithm in [CDG19] and prove stronger robust estimation guarantees tailored for the specific distribution of $Z = Y \otimes Y$. Our main algorithm (Algorithm 1) combines these two refinement steps to match the best-known robustness guarantees for covariance estimation.

Evidence of Hardness. It is natural to ask whether one can obtain faster running times than those achieved here. There is a sample complexity lower bound of $N = \Omega(d^2)$, so we assume we are given $X \in \mathbb{R}^{N \times d}$ as our sample matrix (here we will focus on constant ϵ). We note that even in the non-robust setting, it is not known how to output an approximate covariance matrix $\widehat{\Sigma}$ for which $\|\widehat{\Sigma} - \frac{1}{N} X^T X\|_F = O(1)$ in time faster than (d, d^2, d) matrix multiplication time, which is the time to multiply a $d \times d^2$ matrix by a $d^2 \times d$ matrix. Since the non-robust case is a special case of our setting, one cannot improve our running time without improving the running time of non-robust covariance estimation.

A natural way of improving the running time of non-robust covariance estimation is to try to approximate the product $X^T X$ using oblivious sketching (see, e.g., [Woo14] for a survey) which works roughly as follows. One samples a random S from a certain family of random matrices, and computes $S \cdot X$ where S has much fewer than N rows. For structured random families of matrices, like fast JL matrices, $S \cdot X$ can be computed very quickly. Then one instead computes $X^T S^T S X$ with the guarantee that $\|X^T S^T S X - X^T X\|_F^2$ is small. Note that the matrix product $(X^T S^T) \cdot (SX)$ can be performed more quickly if S has a small number of rows. Unfortunately, for the guarantees we want, all known constructions of S require $\Omega(N)$ rows. In fact, we prove an information-theoretic result that any oblivious sketching matrix S must have $\Omega(N/\log N)$ rows in Lemma 6.1, thus ruling out this approach for achieving faster running time. Our proof uses arguments from communication complexity, arguing that such a family of sketching matrices would imply a better protocol for solving multiple copies of the Gap-Hamming communication problem.

Another way of trying to improve the runtime is to give an alternative definition of the problem: Instead of outputting a $d \times d$ matrix that is close to Σ, the algorithm outputs a set of nonnegative weights w such that $\|w\|_1 = 1$, $\|w\|_\infty \le \frac{1}{(1-\epsilon)N}$, and $\|\sum_{i=1}^N w_i X_i X_i^T - \Sigma\|_F = O(1)$. This bypasses the arguments above, since in the case of no corruption, we do not have to actually output $X^T X$ and can just set $w_i = 1/N$ for all i. However, even for this relaxed version of the problem, we show that unless one can solve a certain "column norm" distinguishing problem faster than rectangular matrix multiplication, one cannot solve this problem faster than (d, d^2, d) matrix multiplication time. This problem can be intuitively stated as follows: the good samples are drawn from $\mathcal{N}(0, I)$, and the corrupted samples are drawn from a mean-zero Gaussian distribution with a very slight and known perturbation to the identity covariance matrix. One needs to identify a large fraction of these corrupted samples. Even though the covariance of the perturbed Gaussians is known, it is so slight that the norms of the corrupted samples are very similar to the uncorrupted ones. Therefore, one needs to measure these norms along certain directions, which requires computing a matrix product of the samples with a worst-case covariance matrix. We show that outputting the weights described above requires solving this problem, which we conjecture to be hard.

1.3 Related and Prior Work

Learning in the presence of outliers is an important goal in statistics and has been studied in the robust statistics community since the 1960s ([Hub64]). After several decades of work, a number of sample-efficient and robust estimators have been discovered. The reader is referred to ([DKK+16, LRV16]) for a detailed summary of this line of work. Until recently, all known computationally efficient high-dimensional estimators could only tolerate a negligible fraction of outliers. Recent work ([DKK+16, LRV16]) gave the first efficient robust estimators for basic high-dimensional unsupervised tasks. Since these works, there has been a flurry of research activity on robust learning algorithms in both supervised and unsupervised settings ([BDLS17, CSV17, DKK+17, DKS17, DKK+18, SCV18, DKS18b, DKS18a, HL18, KSS18, PSBR18, DKK+19, KKM18, DKS19, LSLC18, CDKS18]).

The most relevant prior work is that of [CDG19], initiating the direction of obtaining fast algorithms for robust high-dimensional estimation. For the problem of robust mean estimation, [CDG19] proposed a primal-dual approach – building on the convex programming approach of [DKK+16] – yielding an algorithm with runtime $\widetilde{O}(Nd)/\operatorname{poly}(\epsilon)$. This improved on the $\widetilde{O}(Nd^2)$ runtime of the iterative filtering method in [DKK+16]. Our algorithm uses the same primal-dual framework; however, we emphasize that a standard application of their framework would only lead to a runtime of $\widetilde{O}(Nd^2)/\operatorname{poly}(\epsilon)$. To obtain our improved runtime, we need to overcome a number of technical obstacles, as we explained in Section 1.2.

2 Preliminaries

Basic Notations. For a positive integer n, we write $[n]$ for the set $\{1, \ldots, n\}$. We use e_i to denote the i-th standard basis vector, and I to denote the identity matrix. For a vector x, we use $\|x\|_1$, $\|x\|_2$, and $\|x\|_\infty$ to denote the ℓ_1, ℓ_2, and ℓ_∞ norm of x respectively. For a matrix A, we use $\|A\|_2$ and $\|A\|_F$ to denote the spectral norm and Frobenius norm of A respectively.

Let $\operatorname{tr}(A)$ be the trace of A, and $\kappa(A)$ be the condition number of A. A symmetric matrix $A \in \mathbb{R}^{n \times n}$ is said to be positive semidefinite (PSD) if $x^\top A x \geq 0$ for all $x \in \mathbb{R}^n$. For two symmetric matrices A and B, we write $A \preceq B$ iff the matrix $B - A$ is positive semidefinite.

For two vectors x and y of the same dimensions, let $\langle x, y \rangle = x^\top y = \sum_i x_i y_i$ be the inner product of x and y. For two matrices A and B of the same dimensions, let $A \bullet B = \langle A, B \rangle = \operatorname{tr}(A^\top B)$ be the entry-wise inner product of A and B. For a matrix $A \in \mathbb{R}^{n \times n}$, we use $A^\flat \in \mathbb{R}^{n^2}$ to denote its canonical flattening into a vector.

Throughout this paper, we use D to denote the ground-truth distribution. We use d for the dimension of D, N for the number of samples, and ϵ for the fraction of corrupted samples. We use $X \sim D$ to denote a sample (i.e., a vector random variable) drawn from D. Given N (possibly corrupted) samples $(X_i)_{i=1}^N$ drawn from D, we often abuse notation and again use X to denote the $N \times d$ matrix where the i-th column of X is the i-th sample $X_i \in \mathbb{R}^d$.

Connections Between the Second and Fourth Moments of a Gaussian. Let $A \otimes B$ denote the Kronecker product of A and B. In this paper, we frequently consider the Kronecker product of a sample $X \in \mathbb{R}^d$ with itself: $Z = X \otimes X \in \mathbb{R}^{d^2}$. Our algorithms crucially rely on the following lemma, which characterizes the connections between the second-moment and fourth-moment tensor of a Gaussian. Lemma 2.1 is proved in Appendix A.

Lemma 2.1. *Let $X \sim \mathcal{N}(0, \Sigma)$ and $Z = X \otimes X$. Let $\Sigma_Z \in \mathbb{R}^{d^2 \times d^2}$ be the covariance matrix of Z. We have (i) if $\Sigma \preceq I$, then $\mathrm{cov}(Z) \preceq 2I$; and (ii) if $0 < \tau < 1$ and $\|\Sigma - I\|_2 \le \tau$, then $\|\mathrm{cov}(Z) - 2I\|_2 \le 6\tau$.*

Fast Rectangular Matrix Multiplication. We will frequently use fast rectangular matrix multiplication in our algorithms. Let (a, b, c)-matrix multiplication time denote the time it takes to multiply an $a \times b$ matrix with a $b \times c$ matrix. It is forklore (see, e.g., [LR83]) that (a, a, b), (a, b, a), and (b, a, a) matrix multiplications require the same number of arithmetic operations. More specifically, we use the algorithm proposed in [Gal12]. They obtained new upper bounds on (n, n^α, n) matrix multiplication time. We use a special case of their result ($\alpha = 2$).

Lemma 2.2 (Fast Rectangular Matrix Multiplication ([Gal12]))**.** *We can compute (n, n^2, n)-matrix multiplication in time $O(n^{3.26})$. This implies that for $d > 0$ and $N = \widetilde{O}(d^2/\epsilon^2)$, we can multiply a $d \times N$ matrix with an $N \times d$ matrix in time $\widetilde{O}(d^{3.26}/\epsilon^2)$.*

To multiply a $d \times N$ matrix with an $N \times d$ matrix when $N = \widetilde{O}(d^2/\epsilon^2)$, we split the matrices into blocks of size $d \times d^2$ or $d^2 \times d$, multiply each pair of matrices and then add the results together. The total running time is $\frac{N}{d^2} \cdot O(d^{3.26} + d^2) = \widetilde{O}(d^{3.26}/\epsilon^2)$.

Transposition Principle for Matrix-Vector Multiplication. The transposition principle (see, e.g., [Bor57, Fid73]) plays a central role in our faster implementation of positive SDP solvers. It states that matrix-vector multiplication by $A^\top$ has almost exactly the same computational complexity as matrix-vector multiplication by A.

Lemma 2.3 (Transposition Principle ([Fid73]))**.** *Fix a matrix $A \in \mathbb{R}^{r \times c}$. Suppose there exists an arithmetic circuit of size s that can compute Ax for arbitrary $x \in \mathbb{R}^c$. Then, there exists an arithmetic circuit of size $O(s + m)$ that computes $A^\top y$ for arbitrary $y \in \mathbb{R}^r$.*

3 Estimating the Covariance of a Gaussian Distribution

In this section, we present our key structural and computational lemmas, and use them to prove our main algorithmic results (Theorems 1.2 and 1.3).

3.1 Robust Covariance Estimation: Multiplicative Approximation

We first present our algorithm (Algorithm 1) for robustly estimating the covariance of Gaussian distributions with multiplicative error guarantees. Algorithm 1 starts with an upper bound Σ_0 on the true covariance matrix Σ, and iteratively compute more and more accurate upper bounds $\Sigma_t \succeq \Sigma$.

First we need a reasonable starting point before we can run any iterative refinement steps.

Lemma 3.1. *Consider the same setting as in Theorem 1.2. We can compute a matrix Σ_0 in $\widetilde{O}(d^{3.26}/\epsilon^2)$ time such that, with high probability, $\Sigma \preceq \Sigma_0 \preceq (\kappa\, \mathrm{poly}(d))\Sigma$ and $\|\Sigma_0\|_2 \le \mathrm{poly}(d) \|\Sigma\|_2$.*

We use two different iterative refinement steps (Lemmas 3.2 and 3.3), which correspond to the two loops in Algorithm 1. In the first phase (Lemma 3.2), we only have a crude upper bound on Σ.

Algorithm 1: Robust Covariance Estimation for Gaussian Distributions (Multiplicative Error)

```
Input  : $0 < \epsilon < \epsilon_0$, and an $\epsilon$-corrupted set of $N = \widetilde{\Omega}(d^2/\epsilon^2)$ samples drawn from $\mathcal{N}(0, \Sigma)$.
Output: A matrix $\widehat{\Sigma} \in \mathbb{R}^{d \times d}$ such that with high prob.,
        $\left\| \widehat{\Sigma}^{-1/2} \Sigma \widehat{\Sigma}^{-1/2} - I \right\|_F \le O(\epsilon \log(1/\epsilon))$.
Compute an initial upper bound $\Sigma_0$ with $\Sigma \preceq \Sigma_0 \preceq \kappa \operatorname{poly}(d) \Sigma$ using Lemma 3.1.
Let $T_1 = O(\log \kappa + \log d)$ and $T_2 = T_1 + O(\log \log(1/\epsilon))$.
for $t = 0$ to $T_1 - 1$ do
    Compute $\Sigma_{t+1} \in \mathbb{R}^{d \times d}$ with $\Sigma \preceq \Sigma_{t+1} \preceq \Sigma + O(\sqrt{\epsilon}) \Sigma_t$ using Lemma 3.2 on $\Sigma_t$.
Let $\tau_{T_1} = O(\sqrt{\epsilon})$.
for $t = T_1$ to $T_2 - 1$ do
    Let $\tau_{t+1} = O(\sqrt{\epsilon \tau_t} + \epsilon \log(1/\epsilon))$.
    Compute $\Sigma_{t+1} \in \mathbb{R}^{d \times d}$ with $\Sigma \preceq \Sigma_{t+1} \preceq \Sigma + \tau_{t+1} \Sigma_t$ using Lemma 3.3 on $\Sigma_t$.
Compute $\widehat{\Sigma}$ by invoking Lemma 3.3 on $\Sigma_{T_2}$.
return $\widehat{\Sigma}$.
```

Lemma 3.2 (Iterative Refinement: Getting $\sqrt{\epsilon}$ Error)**.** *Consider the same setting as in Theorem 1.2. Given an upper bound $\Sigma_t \in \mathbb{R}^{d \times d}$ on the unknown covariance matrix Σ, i.e., $\Sigma \preceq \Sigma_t$, we can compute in time $\widetilde{O}(d^{3.26}/\epsilon^8)$ an upper bound matrix $\Sigma_{t+1} \in \mathbb{R}^{d \times d}$, and a hypothesis matrix $\widehat{\Sigma} \in \mathbb{R}^{d \times d}$ such that, with high probability,*

$$\Sigma \preceq \Sigma_{t+1} \preceq \Sigma + O(\sqrt{\epsilon}) \Sigma_t \ , \ \textit{and}$$

$$\| \widehat{\Sigma} - \Sigma \|_F \le O(\sqrt{\epsilon}) \left\| \Sigma_t \right\|_2 \ .$$

The first phase can only converge to a matrix Σ_{T_1} with $\Sigma \preceq \Sigma_{T_1} \preceq (1 + O(\sqrt{\epsilon}) \Sigma$. In the second phase (Lemma 3.3), we already have a fairly accurate estimate of Σ, so the refinement steps converge faster and eventually we can get to a matrix Σ_{T_2} with $\Sigma \preceq \Sigma_{T_2} \preceq (1 + O(\epsilon \log(1/\epsilon)) \Sigma$.

Lemma 3.3 (Iterative Refinement: Getting $\epsilon \log(1/\epsilon)$ Error)**.** *Consider the same setting as in Theorem 1.2. Let $0 < \tau_t < \tau_0$ for some universal constant τ_0. Given τ and Σ_t with $\Sigma \preceq \Sigma_t \preceq (1 + \tau_t) \Sigma$, we can compute in time $\widetilde{O}(d^{3.26}/\epsilon^8)$ an upper bound matrix Σ_{t+1} and a hypothesis matrix $\widehat{\Sigma}$ such that, with high probability, for $\tau_{t+1} = O(\sqrt{\epsilon \tau} + \epsilon \log(1/\epsilon))$,*

$$\Sigma \preceq \Sigma_{t+1} \preceq \Sigma + \tau_{t+1} \Sigma_t \ , \ \textit{and}$$

$$\| \Sigma^{-1/2} \widehat{\Sigma} \Sigma^{-1/2} - I \|_F \le \tau_{t+1} \ .$$

We defer the proof of Lemma 3.1 to Appendix B, and the proofs of Lemmas 3.2 and 3.3 to Section 3.2. We first use these three lemmas to prove Theorem 1.2 (correctness and runtime of Algorithm 1).

Proof of Theorem 1.2. We first use Lemma 3.1 to find an upper bound $\Sigma_0 \in \mathbb{R}^{d \times d}$ on the true covariance matrix Σ such that $\Sigma \preceq \Sigma_0 \preceq (\kappa \operatorname{poly}(d)) \Sigma$.

For any integer $t \geq 0$, given the upper bound matrix Σ_t, we can use Lemma 3.2 to obtain a better upper bound Σ_{t+1} such that $\Sigma_{t+1} \preceq \Sigma + O(\sqrt{\epsilon})\Sigma_t$. Since $\Sigma_0 \preceq \kappa \operatorname{poly}(d)\Sigma$, after $T_1 = O(\log \kappa + \log d)$ iterations, we have a matrix Σ_{T_1} with $\Sigma \preceq \Sigma_{T_1} \preceq (1 + O(\sqrt{\epsilon}))\, \Sigma$.

At this point we have a pretty accurate upper bound Σ_{T_1} with $\Sigma \preceq \Sigma_{T_1} \preceq (1 + \tau_{T_1})\Sigma$, where $\tau_{T_1} = O(\sqrt{\epsilon})$. For any integer $t \geq T_1$, given Σ_t and τ_t, we can use Lemma 3.3 to obtain a better upper bound matrix Σ_{t+1} such that $\Sigma \preceq \Sigma_{t+1} \preceq \Sigma + \tau_{t+1}\Sigma_t$, where $\tau_{t+1} = O(\sqrt{\epsilon\tau_t} + \epsilon \log(1/\epsilon))$. Similar to the previous step, after $O(\log\log(1/\epsilon))$ iterations, we have a matrix Σ_{T_2} such that $\Sigma \preceq \Sigma_{T_2} \preceq (1 + \tau_{T_2})\Sigma$, where $\tau_{T_2} = O(\epsilon \log(1/\epsilon))$.

Finally, using Lemma 3.3 one more time with Σ_{T_2} and τ_{T_2}, we can get a matrix $\widehat{\Sigma}$ with

$$\left\|\Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2} - I\right\|_F = O(\sqrt{\epsilon\tau_{T_2}} + \epsilon \log(1/\epsilon)) = O(\epsilon \log(1/\epsilon)) \ .$$

We note that both Lemmas 3.2 and 3.3 hold with high probability, so we can take a union bound over the failure probabilities and conclude that with high probability all iterative refinement steps are successful, and therefore, we can return $\widehat{\Sigma}$ as our final answer.

Now we analyze the running time of Algorithm 1. We call Lemma 3.1 once to compute Σ_0, which takes time $\widetilde{O}(d^{3.26}/\epsilon^2)$. After that, we use two iterative refinement steps. The total number of iterations is $O(\log \kappa + \log d + \log\log(1/\epsilon))$. In each iteration, we invoke either Lemma 3.2 or 3.3. Since both lemmas run in time $\widetilde{O}(d^{3.26}/\epsilon^8)$, the overall running time is $\widetilde{O}(d^{3.26}/\epsilon^2) + O(\log \kappa + \log d + \log\log(1/\epsilon)) \cdot \widetilde{O}(d^{3.26}/\epsilon^8) = \widetilde{O}(d^{3.26}/\epsilon^8)$. □

3.2 Implementing the Iterative Refinement Steps

In this section, we prove the iterative refinement lemmas (Lemmas 3.2 and 3.3). Both refinement steps use robust mean estimation algorithms as subroutines. More specifically, let $Y_i = \Sigma_t^{-1/2} X_i$ and $Z_i = Y_i \otimes Y_i$. When $X \sim \mathcal{N}(0, \Sigma)$, the covariance matrix of Y is $\Sigma_Y = \Sigma_t^{-1/2}\Sigma\Sigma_t^{-1/2}$, so the mean of Z is $\mathbb{E}[Z] = \left(\Sigma_t^{-1/2}\Sigma\Sigma_t^{-1/2}\right)^\flat$. If we can get a good estimate of $\mathbb{E}[Z]$, we can use this information to obtain a better upper bound Σ_{t+1}.

In the first phase (Lemma 3.2), we only have a crude upper bound on Σ. For any $\Sigma_t \succeq \Sigma$, we have $\Sigma_Y = \Sigma_t^{-1/2}\Sigma\Sigma_t^{-1/2} \preceq I$, which implies $\Sigma_Z \preceq 2I$ (Lemma 2.1). Because Z has bounded covariance, we can use the following robust mean estimation algorithm from [CDG19].

Lemma 3.4 (Robust Mean Estimation for Bounded-Covariance Distributions, [CDG19])**.** *Let D be a distribution on supported on $\mathbb{R}^d$ with unknown mean and unknown covariance matrix Σ such that $\Sigma \preceq \sigma^2 I$. Let $0 < \epsilon < \epsilon_0$ for some universal constant ϵ_0, and let $\delta = O(\sqrt{\epsilon})$. Given an ϵ-corrupted set of $N = \widetilde{\Omega}(d/\epsilon)$ samples drawn from D, Algorithm 5 outputs a hypothesis vector $\widehat{\mu}$ such that with high probability, $\|\mu - \mu^\star\|_2 \leq O(\sigma\delta) = O(\sigma\sqrt{\epsilon})$.*

In the second phase (Lemma 3.3), we use the fact that the current estimate Σ_t is already very close to Σ, therefore $Y = \Sigma_t^{-1/2} X$ is very close to $\mathcal{N}(0, I)$. In this case we need an algorithm with stronger robust estimation guarantees tailored for the specific distribution of $Z = Y \otimes Y$.

Lemma 3.5 (Robust Mean Estimation with Approximately Known Covariance)**.** *Let D be a distribution supported on $\mathbb{R}^d$ with unknown mean $\mu^\star$ and covariance Σ. Let $0 < \epsilon < \epsilon_0$ for some universal constant ϵ_0, $\tau \leq O(\sqrt{\epsilon})$, and $\delta = O(\sqrt{\tau\epsilon} + \epsilon \log(1/\epsilon))$. Suppose that D has exponentially decaying tails, and Σ is close to the identity matrix $\|\Sigma - I\|_2 \leq \tau$. Given an ϵ-corrupted set of*

$N = \widetilde{\Omega}(d/\delta^2)$ samples drawn from D, Algorithm 2 outputs a hypothesis vector $\widehat{\mu}$ such that with high probability, $\|\mu - \mu^\star\|_2 \le O(\delta)$.

It is worth noting that we cannot use these mean estimation algorithms in a black-box manner. This is because writing down the input explicitly takes $\Omega(Nd^2) = \widetilde{\Omega}(d^4/\epsilon^2)$ time, and these algorithms run in time $\widetilde{\Omega}(Nd^2)/\operatorname{poly}(\epsilon)$ in d^2 dimensions. One of our main contributions is to show that it is possible to open up these algorithms and take advantage of the additional structures of our inputs (they all have the form $Y_i \otimes Y_i$) to implement both algorithms to run in time $\widetilde{O}(d^{3.26})/\operatorname{poly}(\epsilon)$.

Proposition 3.6 (Robust Mean Estimation with Tensor Input)**.** *If all input samples $(Z_i)_{i=1}^N$ have the form $Z_i = Y_i \otimes Y_i$ for some $Y_i \in \mathbb{R}^d$, and they are given implicitly as the vectors $(X_i)_{i=1}^N$, then both Algorithms 5 and 2 can be implemented to run in $\widetilde{O}(d^{3.26}/\epsilon^8)$ time.*

We give a description of the algorithm for Lemma 3.4 in Appendix B.2 (Algorithm 5). We prove Lemma 3.5 and present the corresponding algorithm (Algorithm 2) in Section 4. In Section 5, we show that both algorithms can be implemented to run in time $\widetilde{O}(d^{3.26})/\operatorname{poly}(\epsilon)$ (Proposition 3.6). We first use Lemmas 3.4, 3.5, and Proposition 3.6 to prove the iterative refinement lemmas.

Proof of Lemma 3.2. Given an upper bound Σ_t on the true covariance matrix, we can rotate the input samples to compute $Y_i = \Sigma_t^{-1/2} X_i$. Let $Z_i = Y_i \otimes Y_i$. Note that when $X \sim \mathcal{N}(0, \Sigma)$, the random variable $Y = \Sigma_t^{-1/2} X$ is drawn from a Gaussian distribution with covariance $\Sigma_Y = \Sigma_t^{-1/2} \Sigma \Sigma_t^{-1/2} \preceq I$. Lemma 2.1 implies that, if $\Sigma_Y \preceq I$, then $\Sigma_Z \preceq 2I$. Therefore, $(Z_i)_{i=1}^N$ is an ϵ-corrupted set of samples drawn from a distribution with bounded covariance, so we can apply Algorithm 5 to robustly estimate its mean.

Let M be the output of Algorithm 5 reshaped into a $d \times d$ matrix. Because $\mathbb{E}[Z] = \left(\mathbb{E}\left[YY^\top\right]\right)^\flat = \left(\Sigma_t^{-1/2} \Sigma \Sigma_t^{-1/2}\right)^\flat$ and Algorithm 5 (Lemma 3.4) guarantees that $\left\|M^\flat - \mathbb{E}[Z]\right\|_2 \le O(\sqrt{\epsilon})$,

$$\left\|M - \Sigma_t^{-1/2} \Sigma \Sigma_t^{-1/2}\right\|_F \le O(\sqrt{\epsilon}) \; .$$

Let $\widehat{\Sigma} = \Sigma_t^{1/2} M \Sigma_t^{1/2}$. Using $\|AB\|_F \le \|A\|_2 \|B\|_F$, we can prove the first part of the lemma,

$$\left\|\widehat{\Sigma} - \Sigma\right\|_F = \left\|\Sigma_t^{1/2} M \Sigma_t^{1/2} - \Sigma\right\|_F \le O(\sqrt{\epsilon}) \left\|\Sigma_t\right\|_2 \; .$$

As a result, we have that $-O(\sqrt{\epsilon}) I \preceq \Sigma_t^{-1/2} (\widehat{\Sigma} - \Sigma) \Sigma_t^{-1/2} \preceq O(\sqrt{\epsilon}) I$, or equivalently

$$\widehat{\Sigma} - O(\sqrt{\epsilon}) \Sigma_t \preceq \Sigma \preceq \widehat{\Sigma} + O(\sqrt{\epsilon}) \Sigma_t \; .$$

Now $\Sigma_{t+1} = \widehat{\Sigma} + O(\sqrt{\epsilon}) \Sigma_t$ is a better upper bound, which satisfies $\Sigma \preceq \Sigma_{t+1} \preceq \Sigma + O(\sqrt{\epsilon}) \Sigma_t$.

For the running time, we can compute $\Sigma_t^{-1/2}$ explicitly in time $O(d^\omega)$ using SVD [DDH07]. Given the input sample matrix $X \in \mathbb{R}^{d \times N}$, we can apply $\Sigma_t^{-1/2}$ to all samples by computing $Y = \Sigma_t^{-1/2} X$ via fast rectangular matrix multiplication in time $\widetilde{O}(d^{3.26}/\epsilon^2)$ (Lemma 2.2).

Since all the Z_i's have the form $Y_i \otimes Y_i$, Proposition 3.6 shows that Algorithms 5 has running time $\widetilde{O}(d^{3.26}/\epsilon^8)$. Given the output of Algorithms 5, we can compute the new upper bound Σ_{t+1} in time $O(d^\omega)$ using a constant number of $d \times d$ matrix additions and multiplications. □

Proof of Lemma 3.3. Let $Y_i = \Sigma_t^{-1/2} X_i$ and $Z_i = Y_i \otimes Y_i$. We know that

$$\left\| \Sigma_Y - I \right\|_2 = \left\| \Sigma_t^{-1/2} \Sigma \Sigma_t^{-1/2} - I \right\|_2 \leq \tau \ .$$

By Lemma 2.1, we have $\left\| \Sigma_Z - 2I \right\|_2 \leq O(\tau)$. By standard concentration results, $Z = Y \otimes Y$ has exponential concentration about its mean in any direction. Therefore, $(Z_i)_{i=1}^N$ is an ϵ-corrupted set of samples drawn from a distribution that satisfies the conditions in Lemma 3.5, and we can apply Algorithm 2 to robustly learn the mean of Z. By Lemma 3.5, we can compute a matrix M such that

$$\left\| M - \Sigma_t^{-1/2} \Sigma \Sigma_t^{-1/2} \right\|_F \leq O(\sqrt{\epsilon\tau} + \epsilon \log(1/\epsilon)) \ .$$

Let $\widehat{\Sigma} = \Sigma_t^{1/2} M \Sigma_t^{1/2}$. Using $\left\| AB \right\|_F \leq \left\| A \right\|_2 \left\| B \right\|_F$, we can prove the first part of the lemma,

$$\begin{aligned}
\left\| \Sigma^{-1/2} \widehat{\Sigma} \Sigma^{-1/2} - I \right\|_F &= \left\| \Sigma^{-1/2} \Sigma_t^{1/2} M \Sigma_t^{1/2} \Sigma^{-1/2} - I \right\|_F \\
&\leq \left\| \Sigma^{-1/2} \Sigma_t^{1/2} \right\|_2 \left\| M - \Sigma_t^{-1/2} \Sigma \Sigma_t^{-1/2} \right\|_F \left\| \Sigma_t^{1/2} \Sigma^{-1/2} \right\|_2 \\
&\leq (1 + \tau) \cdot O(\sqrt{\epsilon\tau} + \epsilon \log(1/\epsilon)) \\
&= O(\sqrt{\epsilon\tau} + \epsilon \log(1/\epsilon)) \ .
\end{aligned}$$

This gives a better upper bound $\Sigma_{t+1} = \widehat{\Sigma} + O(\sqrt{\epsilon\tau} + \epsilon \log(1/\epsilon))\Sigma_t$ such that $\Sigma \preceq \Sigma_{t+1} \preceq \Sigma + O(\sqrt{\epsilon\tau} + \epsilon \log(1/\epsilon))\Sigma_t$.

We omit the running time analysis because it is identical to the one in the previous proof. □

4 Robust Mean Estimation Subroutines

We first present the robust mean estimation algorithm (Algorithm 2) that achieves Lemma 3.5.

Algorithm 2: Robust Mean Estimation with Approximately Known Covariance

```
Input  : An $\epsilon$-corrupted set of $N$ samples $\{Z_i\}_{i=1}^N$ on $\mathbb{R}^d$ with $N = \widetilde{\Omega}(d/\epsilon^2)$ and $\epsilon < \epsilon_0$.
Output: A vector $\widehat{\mu} \in \mathbb{R}^d$ such that, with high probability, $\left\| \widehat{\mu} - \mu^\star \right\|_2 \leq O(\sqrt{\epsilon\tau} + \epsilon \log(1/\epsilon))$.
Let $\nu \in \mathbb{R}^d$ be an initial guess with $\left\| \nu - \mu^\star \right\|_2 \leq \mathrm{poly}(d)$.
for $i = 1$ to $O(\log d)$ do
    Compute a near-optimal solution $w \in \mathbb{R}^N$ to the primal SDP (1) with parameters $\nu$ and $2\epsilon$.
    Compute a near-optimal solution $M \in \mathbb{R}^{d \times d}$ for the dual SDP (2) with parameters $\nu$ and $\epsilon$.
    if the value of $w$ in SDP (1) is at most $1 + c(\tau + \epsilon \log^2(1/\epsilon))$ for a universal constant $c$ then
        return the weighted empirical mean $\widehat{\mu}_w = \sum_{i=1}^N w_i Z_i$ (Lemma B.9) .
    else
        Move $\nu$ closer to $\mu^\star$ using the top eigenvector of $M$ (Lemma B.10).
```

We first recall the primal-dual approach recently developed in [CDG19]. Given data points $Z_i = X_i \otimes X_i$, their algorithm starts with a guess $\nu \in \mathbb{R}^{d^2}$, and then in each iteration solves the primal and dual SDPs (1) and (2). They showed that either a good primal solution gives weights $w \in \mathbb{R}^N$ such that the weighted average $\sum_{i=1}^N w_i Z_i$ is close to the true mean; or a good dual solution must identify a direction of improvement that allows the algorithm to move $\nu' \in \mathbb{R}^{d^2}$ much closer to the true mean of Z.

There are two obstacles for applying the algorithmic framework of [CDG19] to our setting. First, the input samples $Z_i = X_i \otimes X_i$'s are d^2-dimensional vectors. Writing down these vectors explicitly takes time $\Omega(Nd^2) = \Omega(d^4)$. Therefore, we want to solve the SDPs (1) and (2) on input Z_i without computing them explicitly. We resolve this issue in Section 5 (Proposition 3.6).

Second, their algorithms have error $O(\sqrt{\epsilon})$ for bounded-covariance distributions, and error $O(\epsilon\sqrt{\log(1/\epsilon)})$ for sub-gaussian distributions with identity covariance matrix. While we can directly use their result for bounded-covariance distributions for Lemma 3.4, we need to develop a new algorithm for Lemma 3.5. In Lemma 3.5, we have a distribution with exponential decaying tails, and we know its covariance is τ-close to the identity matrix. We want to robustly estimate its mean, with optimal error guarantees that depend on both ϵ and τ. We generalize the analysis of [CDG19] to handle this case. Lemma 3.5 is proved in Appendix B.3.

5 Faster Implementation of Robust Mean Estimation with Tensor Inputs

The bottleneck of both Algorithms 5 and 2 are solving SDPs (1) and (2). In this section, we prove Proposition 3.6, which states that when all input samples have the tensor-product form $Y \otimes Y$, we can solve these SDPs in time $\widetilde{O}(d^{3.26}) / \operatorname{poly}(\epsilon)$.

We first convert the SDPs (1) and (2) into packing/covering SDPs as follows.

$$\begin{array}{ll} \text{maximize} & \mathbf{1}^\top w \\ \text{subject to} & w_i \geq 0, \sum_{i=1}^N w_i A_i \preceq I \end{array} \tag{3}$$

$$\begin{array}{ll} \text{minimize} & \operatorname{tr}(M) \\ \text{subject to} & A_i \bullet M \geq 1, M \succeq 0 \ , \end{array} \tag{4}$$

where each $A_i \in \mathbb{R}^{(d^2+N)\times(d^2+N)}$ is a PSD matrix given by[2]

$$A_i = \left[\begin{array}{cc} \rho(Z_i - \nu)(Z_i - \nu)^\top & 0 \\ 0 & (1-\epsilon)N \cdot e_i e_i^\top \end{array} \right] . \tag{5}$$

Here ρ is a binary search parameter that is between $\frac{1}{d}$ and 1. At the core of nearly-linear time width-independent SDP solvers (e.g., [ALO16, PTZ16]) is an application of matrix multiplicative weight update, where the algorithm maintains a weighted sum Ψ of the matrices. In iteration t, we have $\Psi^t = \sum_{i=1}^n w_i A_i$, and we will update the weights based on the values of $A_i \bullet \frac{\exp(\Psi^t)}{\operatorname{tr}(\exp(\Psi^t))}$.

[2] Recall that $X_i \in \mathbb{R}^{d\times 1}$ is the i-th sample, and $e_i \in \mathbb{R}^{N\times 1}$ is the i-th standard basis vector.

Lemma 5.1 (Positive SDP Solver, [PTZ16])**.** *Let $A_1, \ldots, A_n$ be $m \times m$ PSD matrices given in factorized form $A_i = C_i C_i^\top$. Consider the following pair of packing and covering SDPs:*

$$\begin{aligned} &\max_{x \geq 0} \mathbf{1}^\top x \qquad \text{s.t. } \sum_{i=1}^{n} x_i A_i \preceq I \,. \\ &\max_{Y \succeq 0} \operatorname{tr}(Y) \qquad \text{s.t. } A_i \bullet Y \geq 1, \forall i \,. \end{aligned}$$

Fix $\epsilon > 0$. Given an oracle algorithm that, on input $\Psi = \sum_{i=1}^{n} w_i A_i$ with $\|\Psi\|_2 = O(\log(n)/\epsilon)$, runs in time $T_{\exp}$ and returns $(1 \pm \epsilon)$-multiplicative approximations to $\frac{\exp(\Psi)}{\operatorname{tr}(\exp(\Psi))} \bullet A_i$ for all i. Then, we can compute feasible primal and dual solutions x and Y, such that with high probability, $\mathbf{1}^\top x \geq (1 - O(\epsilon))\mathrm{OPT}$ and $\operatorname{tr}(Y) \leq (1 + O(\epsilon))\mathrm{OPT}$. Moreover, we can do so in time $\widetilde{O}((T_{\exp} + n) \log^2 n/\epsilon^3)$, where q is the total number of non-zero entries in the C_i's.

In the rest of this section, we will prove that when each Z_i has the form $Z_i = Y_i \otimes Y_i$, we can implement the oracle algorithm required by Lemma 5.1 in time $T_{\exp} = \widetilde{O}(d^{3.26}/\epsilon^5)$. It is worth pointing out that we need to implement this oracle without ever writing down $\Psi \in \mathbb{R}^{d^2 \times d^2}$ explicitly.

We will approximate each $\exp(\Psi) \bullet A_i$ and $\operatorname{tr}(\exp(\Psi))$ separately. Observe that $\Psi = \sum_{i=1}^{n} w_i A_i$ and $\exp(\Psi)$ have the same block structure as the A_i's. Due to the special structure of the bottom-right block, we can compute its contribution to $\operatorname{tr}(\exp(\Psi))$ and $\exp(\Psi) \bullet A_i$ exactly. Therefore, we can focus on the top-left block. Moreover, because the goal is to compute a multiplicative approximation of the top-left block's contribution to $\operatorname{tr}(\exp(\Psi))$ and $\exp(\Psi) \bullet A_i$, we can ignore the scalar ρ. We prove the following lemma.

Lemma 5.2. *Fix $d > 0$ and $N = \widetilde{\Omega}(d^2/\epsilon^2)$. Fix $Y \in \mathbb{R}^{d \times N}$, $w \in \mathbb{R}^N$, $\nu \in \mathbb{R}^{d^2}$, and $0 < \epsilon < 1$. Let $Z_i = Y_i \otimes Y_i$ and (abusing notation) let $\Psi = \sum_{i=1}^{N} w_i (Z_i - \nu)(Z_i - \nu)^\top$. Suppose $\|\Psi\|_2 = O(\log d/\epsilon)$. We can compute, in time $\widetilde{O}(d^{3.26}/\epsilon^5)$, $(1 \pm \epsilon)$-multiplicative approximations to $\operatorname{tr}(\exp(\Psi))$ and $\exp(\Psi) \bullet ((Z_i - \nu)(Z_i - \nu)^\top)$ for all $i \in [N]$.*

Proof. Let $Z \in \mathbb{R}^{d \times N}$ be the matrix whose i-th column is Z_i. Observe that $\Psi = (Z - \nu\mathbf{1}^\top) D_w (Z - \nu\mathbf{1}^\top)^\top \in \mathbb{R}^{d^2 \times d^2}$, where $D_w \in \mathbb{R}^{N \times N}$ is a diagonal matrix with w on the diagonal. First we show that, for any vector $s \in \mathbb{R}^{d^2}$, we can compute the matrix-vector multiplication $\Psi s = (Z - \nu\mathbf{1}^\top) D_w (Z - \nu\mathbf{1}^\top)^\top s$ in time $\widetilde{O}(d^{3.26}/\epsilon^2)$. This is because

(i) $(Z - \nu\mathbf{1}^\top)s = Zs - (\mathbf{1}^\top s)\nu$ and we can compute $Zs = \left(Y D_s Y^\top\right)^\flat$ via fast rectangular matrix multiplication (Lemma 2.2),

(ii) matrix-vector multiplication with $(Z - \nu\mathbf{1}^\top)$ or $(Z - \nu\mathbf{1}^\top)^\top$ has the same running time by the Transposition Principle (Lemma 2.3 in Section 2)), and

(iii) multiplication with a diagonal matrix D_w can be done in time $O(N)$.

We continue to show how to compute some η such that $\eta \approx_{\epsilon/2} \operatorname{tr}(\exp(\Psi))$. Since $\|\Psi\|_2 = \epsilon^{-1} \log d$, by Lemma 5.3, we can find a degree-$(\epsilon^{-1} \log d)$ matrix polynomial p such that $p(\Psi) \approx_{\epsilon/8} \exp(\Psi/2)$. Let $M = p(\Psi)$, we have $\operatorname{tr}(M^2) \approx_{\epsilon/4} \operatorname{tr}(\exp(\Psi))$. Thus, it is sufficient to compute some $\eta \approx_{\epsilon/4} \operatorname{tr}(M^2)$. We will write $\operatorname{tr}(M^2) = \sum_{i=1}^{d^2} (M^2)_{i,i}$ and approximate all $(M^2)_{i,i} = \|M e_i\|_2^2$ simultaneously. By the Johnson-Lindenstrauss lemma, there is a $O(\log d/\epsilon^2) \times d^2$ matrix Q such that with high probability, $\|M e_i\|_2^2 \approx_{\epsilon/4} \|Q M e_i\|_2$ for all $i \in [d^2]$. Note that Ψ is symmetric and so is M. We can compute $QM = (MQ^\top)^\top$ by multiplying each column of $Q^\top$ through M, and each $MQ_i = p(\Psi)Q_i$ can be evaluated using $\deg(M)$ matrix-vector multiplications Ψv for some $v \in \mathbb{R}^{d^2}$.

The overall running time to approximate $\operatorname{tr}(\exp(\Psi))$ is $O(\log d/\epsilon^2) \cdot O(\log d/\epsilon) \cdot \widetilde{O}(N + d^{3.26}/\epsilon^2) = \widetilde{O}(d^{3.26}/\epsilon^5)$.

We approximate $\exp(\Psi) \bullet (Z_i - \nu)(Z_i - \nu)^\top$ using a similar approach: $\exp(\Psi) \bullet (Z_i - \nu)(Z_i - \nu)^\top = \|\exp(\Psi/2)(Z_i - \nu)\|_2^2 \approx_{\epsilon/4} \|M(Z_i - \nu)\|_2^2 \approx_{\epsilon/4} \|QM(Z_i - \nu)\|_2^2$. Notice that the last line is precisely the squared norm of the i-th column of $QM(Z - \nu\mathbf{1}^\top)$. For the same reasons as in the previous case, we can compute this matrix in time $\widetilde{O}(d^{3.26}/\epsilon^5)$. □

We can approximate $\exp(A)$ with a matrix polynomial of A, whose degree depends on the spectral norm of A and the desired precision (see, e.g., [AK16]).

Lemma 5.3 (Taylor Expansion of Matrix Exponential). *Let A be PSD matrix with $\|A\|_2 \leq \ell$, then there exists a polynomial $p(A)$ of degree $O(\max(\ell, \log(2/\epsilon)))$ such that $p(A) \approx_\epsilon \exp(A)$.*

Proof of Proposition 3.6. By Lemma 5.1, we only need to show that the required oracle algorithm can be implemented in time $\widetilde{O}(d^{3.26}/\epsilon^5)$. We approximate $\operatorname{tr}(\exp(\Psi))$ and each $\exp(\Psi) \bullet A_i$ separately. Given $\Psi = \sum_{i=1}^N w_i A_i$, we will compute the contribution from bottom-right block explicitly, and use Lemma 5.2 for the top-left block. The bottom-right block adds $\sum_{i=1}^N \exp(w_i(1-\epsilon)N)$ to $\operatorname{tr}(\exp(\Psi))$, and for every i, it adds $w_i \exp(w_i(1-\epsilon)N)$ to $\exp(\Psi) \bullet A_i$. □

6 Evidence of Hardness

In this section, we provide some evidence which suggests that the running time of our algorithm has near-optimal dependence on d. We start by noting that our sample complexity $N = \widetilde{\Omega}(d^2/\epsilon^2)$ is tight up to polylogarithmic factors, and this holds even when there is no corruption. For the rest of this section, we will assume both ϵ and κ are constants, and focus on the dependence on d in the running time. Since the running time of our algorithm is dominated by (d, d^2, d)-matrix multiplication time, faster matrix multiplication algorithms time will improve our running time.

In Section 6.1, we show that even when there are no corrupted samples, it is not known how to compute the empirical covariance matrix faster than (d, d^2, d)-matrix multiplication time. We give a communication complexity lower bound that rules out all oblivious matrix sketching approaches.

In Section 6.2, to circumvent the difficulty raised in Section 6.1, we consider a weaker problem where the algorithm only need to find a set of good weights (instead of a $d \times d$ matrix). We give a reduction to show that this problem is still at least as hard as some basic matrix computation question, which we do not know how to solve faster than (d, d^2, d)-matrix multiplication time.

6.1 Approximating the Empirical Covariance Matrix

Our algorithm matches the running time of the best non-robust covariance estimation algorithm. When there are no corrupted samples and $N = \widetilde{\Omega}(d^2/\epsilon^2)$, with high probability, the empirical second-moment matrix $\frac{1}{N}\sum_{i=1}^N X_i X_i^\top$ is ϵ-close to the true covariance matrix in Frobenius norm. However, it is not known how to (approximately) compute this empirical second-moment matrix faster than (d, d^2, d) matrix multiplication time.

Problem 1 (Approximating Matrix Products). *Let $d > 0$ and $N = \Omega(d^2)$. Given $X \in \mathbb{R}^{N \times d}$ where each column of X is drawn from $\mathcal{N}(0, \Sigma)$ for some unknown $\Sigma \preceq I$, compute a matrix $\widehat{\Sigma}$ such that*

$$\left\| \widehat{\Sigma} - \frac{1}{N} X^\top X \right\|_F = O(1).$$

For approximate matrix product of an $N \times d$ matrix A with $\|A\|_2 = O(1)$, we want to choose a sketching matrix S so that $\left\|A^\top S^\top SA - A^\top A\right\|_F^2 = O(1)$. Known results for approximate matrix product state that if S has s rows, then $\left\|A^\top S^\top SA - A^\top A\right\|_F^2 = O\left(\frac{\|A\|_F^4}{s}\right)$ with probability at least 9/10, see, e.g., Section 2.2 of [Woo14] for a survey. In the context of Problem 1, letting $A = \frac{1}{\sqrt{N}}X$, we have $\|A\|_2 = O(1)$ and $\|A\|_F^4 = O(d^2)$. The error is then $O(d^2/s)$, and consequently S must have $s = \Omega(d^2)$ rows for the error to be at most $O(1)$.

We can show that the argument above is almost tight for all oblivious sketches.

Lemma 6.1. *Let $N = d^2$. There is no distribution over $t \times N$ matrices S, oblivious to the underlying input $N \times d$ matrix A, where $t = o(d^2/\log d)$, such that with probability at least $2/3$, it holds that $\|A^\top S^\top SA - A^\top A\|_F^2 \le C_1 \frac{\|A\|_F^4}{d^2}$, where $C_1 = 4 \cdot 25 \cdot 2000^2$ is a positive constant.*

Proof. Suppose, to the contrary, there were such a distribution on matrices S satisfying $t = o(d^2/\log d)$.

For $N = d^2$, consider a uniformly random $N \times d$ matrix $A \in \{-\frac{1}{d}, \frac{1}{d}\}^{N\times d}$. Then $\|A\|_F^2 = d$, and so for a random matrix S from our family and a random input A from this family of inputs, it holds that with probability at least $\frac{2}{3}$, $\|A^\top S^\top SA - A^\top A\|_F^2 \le C_1 \frac{\|A\|_F^4}{d^2} = C_1$. By anti-concentration of the binomial distribution, with probability at least $\frac{99}{100}$, at least a $\frac{99}{100}$-fraction of the off-diagonal entries of $A^\top A$ have absolute value at least $\frac{1}{1000d}$.

Consequently, for at least a $\frac{24}{25}$-fraction of the entries in the bottom left $\frac{d}{2} \times \frac{d}{2}$ submatrix of $A^\top A$, we have the property that the entry has the same sign as in $A^\top S^\top SA$, and also the entries in $A^\top S^\top SA$ are at least $\frac{1}{2000d}$. Indeed, otherwise we would have $\|A^\top S^\top SA - A^\top A\|_F^2 > (\frac{d}{2})^2 \cdot \frac{1}{25} \cdot (\frac{1}{1000d} - \frac{1}{2000d})^2 > C_1$ with probability at least $\frac{99}{100}$ over the choice of A and S, and in particular there exists a fixed A for which this holds with probability at least $\frac{99}{100}$ over the choice of S, contradicting our assumption on the family of matrices S.

Now consider the following two-player communication game with public shared randomness. Alice has the first $\frac{d}{2}$ columns of A, denoted $A_L \in \{-1,1\}^{N\times d/2}$ while Bob has the remaining $\frac{d}{2}$ columns of A, denoted $A_R \in \{-1,1\}^{N\times d/2}$. The entries in the in the bottom left $\frac{d}{2} \times \frac{d}{2}$ submatrix of $A^\top A$ are exactly the inner products between all columns of Alice and all columns of Bob. Suppose there were such a family of matrices S as described above. Alice and Bob use the public coin to agree upon S with no communication. Alice then computes $S \cdot A_L$, and rounds each entry to the nearest power of $(1 + \frac{1}{\mathrm{poly}(d)})$. Note all entries of $S \cdot A_L$ need to be at most $\mathrm{poly}(d)$ and rounding preserves $\|A^\top S^\top SA - A^\top A\|_F^2$ up to additive $\frac{1}{\mathrm{poly}(d)}$. Therefore, we maintain the property that, at least a $\frac{24}{25}$-fraction of the entries in the bottom left $\frac{d}{2} \times \frac{d}{2}$ submatrix of $A^\top A$ have the same signs in $A^\top A$ and $A^\top S^\top SA$. Alice sends each of the rounded entries of $S \cdot A_L$, which is $\Theta(td \log d) = o(d^3)$ bits. Bob then computes $S \cdot A_R$ and thus forms $S \cdot A$, from which he can compute $A^\top S^\top SA$. At this point, Bob can recover the sign of a uniformly random entry in the bottom left $d/2 \times d/2$ submatrix of $A^\top A$ with probability at least $2/3 - 1/25 - 1/100 > 3/5$.

Notice that the sign of such an entry is the same as solving the Gap-Hamming communication problem under the uniform distribution: in this communication problem there are two players, Alice and Bob, who hold uniformly random vectors $x, y \in \{-1,1\}^N$, respectively, and wish to decide if $\langle x, y\rangle > 0$ or $\langle x, y\rangle < 0$. This problem requires $\Omega(N)$ randomized communication complexity [CR12]. Moreover, as shown by Braverman et al. [BGPW16], the information complexity of this problem is $\mathcal{I} = \Omega(N)$ bits. In our setting, we can think of Alice as having $\frac{d}{2}$ independent instances

$x^1, \ldots, x^{d/2}$, and Bob having an index $i \in \{1, 2, \ldots, \frac{d}{2}\}$ as well as a vector y and Bob wants to solve the Gap-Hamming problem on the pair (x^i, y). However, only Alice is allowed to speak, and she sends a single message to Bob, without knowing i. By standard direct sum arguments in communication complexity [BR11] (see also [PSW14] where Gap-Hamming composed with the Index problem was used), the randomized one-way communication complexity of this problem is $\Omega(d \cdot \mathcal{I}) = \Omega(d^3)$ bits. However, the communication cost of our protocol is $\Theta(td \log d) = o(d^3)$ bits, which is a contradiction. Consequently, we must have $t = \Omega(d^2/\log d)$, as desired. □

It is worth noting that this lower bound holds for any possible algorithm one can run on SA (i.e., the algorithm can do more than just computing $A^\top S^\top SA$), so it is a stronger information-theoretic statement.

6.2 Finding Good Weights

To circumvent the difficulty of Problem 1, we could redefine our problem so that the algorithm does not need to output a $d \times d$ matrix, instead it outputs a set of good weights w such that $\| \sum_{i=1}^N w_i X_i X_i^\top - \Sigma \|_F = O(\sqrt{\epsilon})$. We will show that, even for this weaker problem of finding good weights, one still need to come up with faster algorithms for a basic matrix problem.

Problem 2 (Identifying Columns with Larger Norms in a Product Matrix)**.** *Let $d > 0$ and $N = \Omega(d^2)$. Fix an arbitrary $U \in \mathbb{R}^{d \times \frac{d}{2}}$ with orthonormal columns. Let Σ' be defined as in Equation* (6)*. Let $X \in \mathbb{R}^{d \times N}$ be a matrix with $(1-\epsilon)N$ columns drawn from $D = \mathcal{N}(0, I)$ and ϵN columns drawn from $D' = \mathcal{N}(0, \Sigma')$. Let B be a set of ϵN columns from D'. Find a set $S \subset [N]$ such that $|S| \le 2\epsilon N$ and $|S \cap B| \ge \frac{|B|}{2}$.*

Consider the following instance. Let U be an arbitrary $d \times \frac{d}{2}$ matrix with orthonormal columns. Let the good distribution be $D = \mathcal{N}(0, \Sigma = I)$, and the noise distribution D' is defined as

$$D' = \mathcal{N}(0, \Sigma'), \qquad \text{where } \Sigma' = \frac{1}{1 + \frac{c}{\epsilon\sqrt{d}}} \left(I + \frac{2c}{\epsilon\sqrt{d}} UU^\top \right) \text{ for some } c = O(\log^{1/2} d) \ . \tag{6}$$

We draw $(1-\epsilon)N$ samples from D and ϵN samples from D'. The empirical covariance matrix of the mixed distribution is $\widehat{\Sigma} = (1-\epsilon)\Sigma + \epsilon\Sigma' = \left(1 - \frac{c}{\sqrt{d} + 1/\epsilon}\right) I + \left(\frac{2c}{\sqrt{d}+1/\epsilon}\right) UU^\top$. Observe that $\left\| \widehat{\Sigma} - \Sigma \right\|_F^2 = d \left(\frac{c}{\sqrt{d}+1/\epsilon} \right)^2 = \Omega(1)$, so the bad samples are distorting the empirical covariance matrix by more than we could tolerate.

Observe that the good and bad samples have similar ℓ_2-norm: $\mathbb{E}_{X' \sim D'}[\|X'\|_2] = \operatorname{tr}(\Sigma') = d = \mathbb{E}_{X \sim D}[\|X\|_2]$. However, the bad samples have slightly larger norm in the column space of U:

$$\mathbb{E}_{X \sim D}\left[\left\| U^\top X \right\|_2^2 \right] = \operatorname{tr}(UU^\top) = \frac{d}{2} \ , \text{ and}$$
$$\mathbb{E}_{X' \sim D'}\left[\left\| U^\top X' \right\|_2^2 \right] = \operatorname{tr}(UU^\top) = \frac{d}{2} \cdot \frac{1 + \frac{2c}{\epsilon\sqrt{d}}}{1 + \frac{c}{\epsilon\sqrt{d}}} \ge \frac{d}{2} \left(1 + \frac{0.9c}{\epsilon\sqrt{d}} \right) \ .$$

Therefore, a natural way of distinguishing them is to compute $U^\top X$, which requires (d, d^2, d)-matrix multiplication time. We could compute the column norms of $SU^\top A$, where S is a Johnson-Lindenstrauss matrix. However, S must have $\frac{1}{\epsilon^2}$ rows to obtain $(1+\epsilon)$-approximation, and therefore

S must have $\widetilde{\Omega}(d^2)$ rows. Even if one uses a sparse matrix S, one has that $SU^\top$ is a dense matrix, and it is unclear how to compute $SU^\top A$ quickly.

Finally, we show that for this specific instance, any algorithm that can find a set of good weights $w \in \mathbb{R}^N$ must solve Problem 2.

Lemma 6.2. *Consider the same setting as in Problem 2. Given a set of weights w such that $\|w\|_1 = 1$, $\|w\|_\infty \le \frac{1}{(1-\epsilon)N}$, and $\|\sum_{i=1}^N w_i X_i X_i^\top - I\|_F = O(1)$, we can solve Problem 2 in $O(N)$ time.*

Proof. Let G and B denote the set of good and bad samples respectively. We have shown that $\mathbb{E}_{i\in G}\left[\left\|U^\top X_i\right\|_2^2\right] = \frac{d}{2}$, and $\mathbb{E}_{i\in B}\left[\left\|U^\top X_i'\right\|_2^2\right] \ge \frac{d}{2}\left(1 + \frac{0.9c}{\epsilon\sqrt{d}}\right)$. By standard concentration result of Chi-squared distributions, we know that there exists $c = O(\log^{1/2} d)$ such that with high probability,

$$\forall i \in G, \quad \left\|U^\top X_i\right\|_2^2 \ge \frac{d}{2}\left(1 - \frac{0.1c}{\sqrt{d}}\right) \text{ , and}$$

$$\forall i \in B, \quad \left\|U^\top X_i\right\|_2^2 \ge \frac{d}{2}\left(1 + \frac{0.8c}{\epsilon\sqrt{d}}\right) \text{ .}$$

For the rest of proof we assume the samples meet these conditions.

Let $\Sigma_w = \sum_{i=1}^N w_i X_i X_i^\top$. Let w_G and w_B denote the total weights on G and B respectively. Since $\|\Sigma_w - I\|_F = O(1)$, by Cauchy-Schwarz,

$$U^\top(\Sigma_w - I)U = (UU^\top) \bullet (\Sigma_w - I) \le \left\|UU^\top\right\|_F \|\Sigma_w - I\|_F = O(\sqrt{d}) \le 0.05c \cdot \sqrt{d} \text{ .}$$

On the other hand,

$$\begin{aligned} U^\top(\Sigma_w - I)U &= \sum_{i=1}^N w_i \left\|U^\top X_i\right\|_2^2 - \frac{d}{2} \\ &\ge w_B\left(\frac{d}{2}\cdot\frac{0.8c}{\epsilon\sqrt{d}}\right) - w_G\left(\frac{d}{2}\cdot\frac{0.1c}{\sqrt{d}}\right) \\ &\ge \frac{w_B}{\epsilon}(0.4c\cdot\sqrt{d}) - (0.05c\cdot\sqrt{d}) \text{ .} \end{aligned}$$

Putting these two inequalities together, we get that $w_B \le \frac{\epsilon}{4}$. In other words, the average weight of a bad sample is $\frac{1}{4N}$.

Let $S = \{i \in [N] : w_i \le \frac{1}{2N}\}$. By Markov's inequality, we have $|S \cap B| \ge \frac{|B|}{2}$. Since $\|w\|_\infty \le \frac{1}{(1-\epsilon)N}$ and $w_G = \|w\|_1 - w_B \ge 1 - \frac{\epsilon}{4}$, again by Markov's inequality, we get that $|S \cap G| \le \epsilon N$ and hence $|S| \le |B| + \epsilon N = 2\epsilon N$. □

Acknowledgments

This work was done in part while some of the authors were visiting the Simons Institute for the Theory of Computing. Ilias Diakonikolas was supported by NSF Award CCF-1652862 (CAREER) and a Sloan Research Fellowship. Rong Ge is supported by NSF Award CCF-1704656, CCF-1845171 (CAREER), a Sloan Research Fellowship, and a Google Faculty Research Award. David Woodruff was supported in part by Office of Naval Research (ONR) grant N00014-18-1-2562.

References

[AK16] S. Arora and S. Kale. A combinatorial, primal-dual approach to semidefinite programs. *Journal of the ACM*, 63(2):12:1–12:35, 2016.

[ALO16] Z. Allen-Zhu, Y. Lee, and L. Orecchia. Using optimization to obtain a width-independent, parallel, simpler, and faster positive SDP solver. In *Proc. 27th Annual Symposium on Discrete Algorithms (SODA)*, pages 1824–1831, 2016.

[BDLS17] S. Balakrishnan, S. S. Du, J. Li, and A. Singh. Computationally efficient robust sparse estimation in high dimensions. In *Proc. 30th Annual Conference on Learning Theory (COLT)*, pages 169–212, 2017.

[BGPW16] M. Braverman, A. Garg, D. Pankratov, and O. Weinstein. Information lower bounds via self-reducibility. *Theory of Computing Systems*, 59(2):377–396, 2016.

[BL08a] P. J. Bickel and E. Levina. Covariance regularization by thresholding. *Ann. Statist.*, 36(6):2577–2604, 12 2008.

[BL08b] P. J. Bickel and E. Levina. Regularized estimation of large covariance matrices. *Ann. Statist.*, 36(1):199–227, 02 2008.

[Bor57] J. L. Bordewijk. Inter-reciprocity applied to electrical networks. *Applied Scientific Research, Section A*, 6(1):1–74, 1957.

[BR11] M. Braverman and A. Rao. Information equals amortized communication. In *Proc. 52nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 748–757, 2011.

[CDG19] Y. Cheng, I. Diakonikolas, and R. Ge. High-dimensional robust mean estimation in nearly-linear time. In *Proc. 30th Annual Symposium on Discrete Algorithms (SODA)*, pages 2755–2771, 2019.

[CDKS18] Y. Cheng, I. Diakonikolas, D. M. Kane, and A. Stewart. Robust learning of fixed-structure Bayesian networks. In *Proc. 33rd Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 10304–10316, 2018.

[CGR18] M. Chen, C. Gao, and Z. Ren. Robust covariance and scatter matrix estimation under Huber's contamination model. *Ann. Statist.*, 46(5):1932–1960, 10 2018.

[CR12] A. Chakrabarti and O. Regev. An optimal lower bound on the communication complexity of gap-hamming-distance. *SIAM J. on Comput.*, 41(5):1299–1317, 2012.

[CSV17] M. Charikar, J. Steinhardt, and G. Valiant. Learning from untrusted data. In *Proc. 49th Annual ACM Symposium on Theory of Computing (STOC)*, pages 47–60, 2017.

[CZZ10] T. T. Cai, C.-H. Zhang, and H. H. Zhou. Optimal rates of convergence for covariance matrix estimation. *Ann. Statist.*, 38(4):2118–2144, 08 2010.

[DDH07] J. Demmel, I. Dumitriu, and O. Holtz. Fast linear algebra is stable. *Numerische Mathematik*, 108(1):59–91, 2007.

[DKK$^+$16] I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, A. Moitra, and A. Stewart. Robust estimators in high dimensions without the computational intractability. In *Proc. 57th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 655–664, 2016.

[DKK$^+$17] I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, A. Moitra, and A. Stewart. Being robust (in high dimensions) can be practical. In *Proc. 34th International Conference on Machine Learning (ICML)*, pages 999–1008, 2017.

[DKK$^+$18] I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, A. Moitra, and A. Stewart. Robustly learning a Gaussian: Getting optimal error, efficiently. In *Proc. 29th Annual Symposium on Discrete Algorithms (SODA)*, pages 2683–2702, 2018.

[DKK$^+$19] I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, J. Steinhardt, and A. Stewart. Sever: A robust meta-algorithm for stochastic optimization. In *Proc. 36th International Conference on Machine Learning (ICML)*, 2019.

[DKS17] I. Diakonikolas, D. M. Kane, and A. Stewart. Statistical query lower bounds for robust estimation of high-dimensional Gaussians and Gaussian mixtures. In *Proc. 58th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 73–84, 2017.

[DKS18a] I. Diakonikolas, D. M. Kane, and A. Stewart. Learning geometric concepts with nasty noise. In *Proc. 50th Annual ACM Symposium on Theory of Computing (STOC)*, pages 1061–1073, 2018.

[DKS18b] I. Diakonikolas, D. M. Kane, and A. Stewart. List-decodable robust mean estimation and learning mixtures of spherical Gaussians. In *Proc. 50th Annual ACM Symposium on Theory of Computing (STOC)*, pages 1047–1060, 2018.

[DKS19] I. Diakonikolas, W. Kong, and A. Stewart. Efficient algorithms and lower bounds for robust linear regression. In *Proc. 30th Annual Symposium on Discrete Algorithms (SODA)*, pages 2745–2754, 2019.

[Fid73] C. M. Fiduccia. *On the algebraic complexity of matrix multiplication.* PhD thesis, Brown University, 1973.

[Gal12] F. L. Gall. Faster algorithms for rectangular matrix multiplication. In *Proc. 53rd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 514–523, 2012.

[Gal14] F. L. Gall. Powers of tensors and fast matrix multiplication. In *International Symposium on Symbolic and Algebraic Computation (ISSAC)*, pages 296–303, 2014.

[HL18] S. B. Hopkins and J. Li. Mixture models, robustness, and sum of squares proofs. In *Proc. 50th Annual ACM Symposium on Theory of Computing (STOC)*, pages 1021–1034, 2018.

[Hub64] P. J. Huber. Robust estimation of a location parameter. *Ann. Math. Statist.*, 35(1):73–101, 03 1964.

[Kan18] D. M. Kane. Robust covariance estimation. *Talk given at TTIC Workshop on Computational Efficiency and High-Dimensional Robust Statistics*, 2018. Available at http://www.iliasdiakonikolas.org/tti-robust/Kane-Covariance.pdf.

[KKM18] A. Klivans, P. Kothari, and R. Meka. Efficient algorithms for outlier-robust regression. In *Proc. 31st Annual Conference on Learning Theory (COLT)*, pages 1420–1430, 2018.

[KSS18] P. K. Kothari, J. Steinhardt, and D. Steurer. Robust moment estimation and improved clustering via sum of squares. In *Proc. 50th Annual ACM Symposium on Theory of Computing (STOC)*, pages 1035–1046, 2018.

[LR83] G. Lotti and F. Romani. On the asymptotic complexity of rectangular matrix multiplication. *Theor. Comp. Sci.*, 23:171–185, 1983.

[LRV16] K. A. Lai, A. B. Rao, and S. Vempala. Agnostic estimation of mean and covariance. In *Proc. 57th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 665–674, 2016.

[LSLC18] L. Liu, Y. Shen, T. Li, and C. Caramanis. High dimensional robust sparse regression. *arXiv preprint arXiv:1805.11643*, 2018.

[NJB$^+$08] J. Novembre, T. Johnson, K. Bryc, Z. Kutalik, A. R. Boyko, A. Auton, A. Indap, K. S. King, S. Bergmann, M. R. Nelson, et al. Genes mirror geography within europe. *Nature*, 456(7218):98–101, 2008.

[PSBR18] A. Prasad, A. S. Suggala, S. Balakrishnan, and P. Ravikumar. Robust estimation via robust gradient estimation. *arXiv preprint arXiv:1802.06485*, 2018.

[PSW14] R. Pagh, M. Stöckel, and D. P. Woodruff. Is min-wise hashing optimal for summarizing set intersection? In *Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, pages 109–120, 2014.

[PTZ16] R. Peng, K. Tangwongsan, and P. Zhang. Faster and simpler width-independent parallel algorithms for positive semidefinite programming. *arXiv preprint arXiv:1201.5135v3*, 2016.

[Rou85] P. Rousseeuw. Multivariate estimation with high breakdown point. *Mathematical Statistics and Applications*, pages 283–297, 1985.

[SCV18] J. Steinhardt, M. Charikar, and G. Valiant. Resilience: A criterion for learning in the presence of arbitrary outliers. In *Proc. 9th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 45:1–45:21, 2018.

[Tuk75] J. W. Tukey. Mathematics and picturing of data. In *Proceedings of ICM*, volume 6, pages 523–531, 1975.

[Woo14] D. P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1-2):1–157, 2014.

A Omitted Proofs from Section 2

Lemma 2.1 *Let $X \sim \mathcal{N}(0, \Sigma)$ and $Z = X \otimes X$. Let $\Sigma_Z \in \mathbb{R}^{d^2 \times d^2}$ be the covariance matrix of Z. We have* *(i) If $\Sigma \preceq I$, then $\Sigma_Z \preceq 2I$.*

(ii) If $0 \le \tau < 1$ and $\|\Sigma - I\|_2 \le \tau$, then $\|\Sigma_Z - 2I\|_2 \le 6\tau$.

Proof. Let $a \in \mathbb{R}^{d^2}$ be any unit vector. Note that

$$\|\Sigma_Z\|_2 = \max_{a \in \mathbb{R}^{d^2}, \|a\|_2 = 1} a^\top \Sigma_Z a \,.$$

Let A be the unique matrix such that $A^\flat = v$. We have

$$v^\top \Sigma_Z v = \mathrm{Var}[v^\top Z] = \mathrm{Var}[X^\top A X] = \mathrm{tr}\left(A\Sigma(A + A^\top)\Sigma\right) \,.$$

Note that Σ is a covariance matrix, so it is always symmetric and PSD. We can write Σ as $\Sigma = \sum_{i=1}^d \lambda_i v_i v_i^\top$. Let $\lambda_{\max}$ and $\lambda_{\min}$ denote the maximum and minimum eigenvalues of Σ. Let $\widehat{A} = \frac{A + A^\top}{2}$. The right hand side is equal to

$$\begin{aligned} a^\top \Sigma_Z a = \mathrm{tr}\left(A\Sigma(A + A^\top)\Sigma\right) &= 2\,\mathrm{tr}\left(\widehat{A}\Sigma\widehat{A}\Sigma\right) \\ &= 2\sum_{i,j} \lambda_i \lambda_j \,\mathrm{tr}(\widehat{A} v_i v_i^\top \widehat{A} v_j v_j^\top) \\ &\le 2(\lambda_{\max})^2 \sum_{i,j} (v_i^\top \widehat{A} v_j)^2 \\ &\le 2(\lambda_{\max})^2 \,. \end{aligned}$$

The last step uses the fact that $\sum_{i,j}(v_i^\top \widehat{A} v_j)^2 = \left\|V\widehat{A}V^\top\right\|_F^2 = \|\widehat{A}\|_F^2 \le \|A\|_F^2 = \|a\|_2^2 = 1$. Since this holds for all unit vector $a \in \mathbb{R}^{d^2}$, we have $\Sigma_Z \preceq 2(\lambda_{\max})^2 I$. Similarly, we can prove that $\Sigma_Z \succeq 2(\lambda_{\min})^2 I$.

For (i), by assumption $\lambda_{\max} = \|\Sigma\|_2 \le 1$, so we have $\|\Sigma_Z\|_2 \le 2$.

For (ii), we know $1 - \tau \le \lambda_{\min} \le \lambda_{\max} \le 1 + \tau$ and $0 < \tau < 1$. It follows that $(1 - 2\tau)2I \preceq \Sigma \preceq (1 + 3\tau)2I$, and thus $\|\Sigma_Z - 2I\|_2 \le 6\tau$. □

B Omitted Proofs from Section 3

Lemma 3.1 *Let $D \sim \mathcal{N}(\mathbf{0}, \Sigma)$ be a zero-mean unknown covariance Gaussian on $\mathbb{R}^d$. Let κ denote the condition number of Σ. Let $0 < \epsilon < \epsilon_0$ for some universal constant ϵ_0. Given an ϵ-corrupted set of $N = \widetilde{\Omega}(d^2/\epsilon^2)$ samples drawn from D, we can compute a matrix Σ_0 in $\widetilde{O}(d^{3.26}/\epsilon^2)$ time such that, with high probability, $\Sigma \preceq \Sigma_0 \preceq (\kappa\,\mathrm{poly}(d))\Sigma$ and $\|\Sigma_0\|_2 \le \mathrm{poly}(d)\,\|\Sigma\|_2$.*

Proof. Let $(G_i)_{i=1}^N$ be the original set of good samples drawn from $\mathcal{N}(0, \Sigma)$, and let $(X_i)_{i=1}^N$ be the corrupted samples. Let S denote the set of $(1 - \epsilon)N$ samples with the smallest norm $\|X_i\|_2$. We define $\Sigma_0 = 2\left(\frac{1}{N}\sum_{i \in S} X_i X_i^\top\right)$.

We first show that $\Sigma_0 \succeq \Sigma$ with high probability. Since the adversary corrupts at most ϵN samples and we throw away ϵN samples, we are left with at least $(1 - 2\epsilon)N$ good samples in

S. We will use the fact that removing any (2ϵ)-fraction of the good samples will not change the empirical covariance too much. Let $Y_i = \Sigma^{-1/2} G_i$ so that if $G_i \sim \mathcal{N}(0, \Sigma)$ then $Y_i \sim \mathcal{N}(0, I)$. When $N = \widetilde{\Omega}(d/\epsilon^2)$, for any $T \subset [N]$ with $|T| = (1-2\epsilon)N$, we have that with high probability,

$$\left\| \frac{1}{N} \sum_{i \in T} Y_i Y_i^\top - I \right\|_2 \leq O(\epsilon \log(1/\epsilon)) .$$

We set $T \subseteq S$ to be a set of $(1-2\epsilon)N$ good samples in S, i.e., $G_i = X_i$ for all $i \in T$. Let $M = \frac{1}{N} \sum_{i \in T} Y_i Y_i^\top$. We know that $M = \frac{1}{N} \sum_{i \in T} \Sigma^{-1/2} G_i G_i^\top \Sigma^{-1/2}$ by definition, and $M \succeq (1 - O(\epsilon \log(1/\epsilon)))I \succeq \frac{I}{2}$ by the above concentration inequality. Therefore,

$$\Sigma_0 = \frac{2}{N} \sum_{i \in S} X_i X_i^\top \succeq \frac{2}{N} \sum_{i \in T} X_i X_i^\top = \frac{2}{N} \sum_{i \in T} G_i G_i^\top = 2 \left(\Sigma^{1/2} M \Sigma^{1/2} \right) \succeq \Sigma .$$

Next we show that $\|\Sigma_0\|_2 \leq \operatorname{poly}(d) \|\Sigma\|_2$ and $\Sigma_0 \preceq (\kappa \operatorname{poly}(d))\Sigma$. Let σ^2 denote the largest eigenvalue of Σ. Again let $Y_i = \Sigma^{-1/2} G_i$, we know that when $N = \widetilde{\Omega}(d/\epsilon^2)$, with high probability,

$$\forall i \in [N], \quad \|Y_i\|_2 \leq O(\sqrt{d \log d}) .$$

We assume this condition holds for the rest of the proof. As a result, $\|G_i\|_2 = \left\| \Sigma^{1/2} Y_i \right\|_2 \leq O(\sigma \sqrt{d \log d})$ for all i. Since only corrupted samples can have larger norm, and we remove the ϵN samples with the largest norm, all samples in S have norm at most $O(\sigma \sqrt{d \log d})$. This gives an upper bound on the spectral norm of Σ_0,

$$\|\Sigma_0\|_2 \leq \frac{2}{N} \sum_{i \in S} \left\| X_i X_i^\top \right\|_2 \leq \frac{2}{N} \cdot N \cdot \max_{i \in S} \|X_i\|_2^2 = O(\sigma^2 d^2 \log^2 d) .$$

This proves $\|\Sigma_0\|_2 \leq \operatorname{poly}(d) \|\Sigma\|_2$. Moreover, by the definition of condition number we know that $\Sigma \succeq \frac{\sigma^2}{\kappa} I$, which implies $\Sigma_0 \preceq (\sigma^2 \operatorname{poly}(d)) I \preceq (\kappa \operatorname{poly}(d))\Sigma$.

We conclude the proof by noting that Σ_0 can be computed by multiplying a $d \times |S|$ matrix with an $|S| \times d$ matrix. This can be done in time $\widetilde{O}(d^{3.26}/\epsilon^2)$ by fast rectangular matrix multiplication (Lemma 2.2). □

B.1 Robust Covariance Estimation: Additive Approximations

In this section, we prove Theorem 1.3. At a high level, we first use Lemma 3.2 to get a $O(\sqrt{\epsilon})$ additive approximation (Algorithm 4). We will then run this algorithm on subspaces that have much smaller eigenvalues in order to improve the guarantee (Algorithm 3). More precisely, we partition $\mathbb{R}^d$ into three disjoint subspaces S_1, S_2, and S_3, then we use Corollary B.2, Lemma B.4, and Lemma B.5 to learn the covariance in each component separately, and combine them together to get the final answer. The fact that we only need an additive approximation is crucial for this approach.

For a subspace S, we use Π_S to denote the projection matrix that maps $x \in \mathbb{R}^d$ onto S, and $S^\perp$ to denote the orthogonal complement of S. Given a matrix A and two subspaces S_1 and S_2, we use $\Sigma[S_1, S_2] = \Pi_{S_1} A \Pi_{S_2}$ to denote the projection of the rows and columns of A onto S_1 and S_2 respectively. We write $A[S]$ for $A[S, S]$.

Let us first prove the guarantee for the crude $O(\sqrt{\epsilon})$ additive estimation.

Algorithm 3: Robust Covariance Estimation for Gaussian Distributions (Additive Error)

```
Input  : $0 < \epsilon < \epsilon_0$, and an $\epsilon$-corrupted set of $N = \widetilde{\Omega}(d^2/\epsilon^2)$ samples $(X_i)_{i=1}^N$ drawn from
         $\mathcal{N}(0, \Sigma)$.
Output: A matrix $\widehat{\Sigma} \in \mathbb{R}^{d \times d}$ such that with high probability,
         $\left\|\widehat{\Sigma} - \Sigma\right\|_F \leq O(\epsilon \log(1/\epsilon)) \|\Sigma\|_2$.
Compute $M_0$ by running Algorithm 4 on input $(X_i)_{i=1}^N$.
Compute eigendecomposition of $M_0$, let $S_1$ be the subspace of all eigenvalues at least $C_1\sqrt{\epsilon}$.
Compute $M_1$ by running Algorithm 4 on input $(\Pi_{S_1^\perp} X_i)_{i=1}^N$.
Compute eigendecomposition of $M_1[S_1^\perp]$, let $S_2$ be the subspace of all eigenvalues at least
 $C_2\epsilon$.
Let $S_3$ be $(S_1 \oplus S_2)^\perp$.
Compute $M_2$ by calling Algorithm 1 on inputs $\{\Pi_{S_1 \oplus S_2} X_i\}$.
Compute $M_3$ by calling Algorithm 4 on inputs $\{(\sqrt{\epsilon}\Pi_{S_1} + \epsilon^{1/4}\Pi_{S_2} + \Pi_{(S_1 \oplus S_2)^\perp})X_i\}$.
Let $\widehat{\Sigma} = M_1 + M_2 - M_2[S_2] + \frac{1}{\sqrt{\epsilon}}(M_3[S_1, S_3] + M_3[S_3, S_1])$.
return $\widehat{\Sigma}$.
```

Algorithm 4: Crude Robust Covariance Estimation

```
Input  : $0 < \epsilon < \epsilon_0$, and an $\epsilon$-corrupted set of $N = \widetilde{\Omega}(d^2/\epsilon^2)$ samples drawn from $\mathcal{N}(0, \Sigma)$.
Output: A matrix $\widehat{\Sigma} \in \mathbb{R}^{d \times d}$ such that with high probability, $\left\|\widehat{\Sigma} - \Sigma\right\|_F \leq O(\sqrt{\epsilon}) \|\Sigma\|_2$.
Compute an initial upper bound $\Sigma_0$ with $\Sigma \preceq \Sigma_0$ and $\|\Sigma_0\|_2 \leq \mathrm{poly}(d) \|\Sigma\|_2$ using
 Lemma 3.1.
Let $T = O(\log d)$.
for $t = 0$ to $T - 1$ do
    Compute $\Sigma_{t+1} \in \mathbb{R}^{d \times d}$ with $\Sigma \preceq \Sigma_{t+1} \preceq \Sigma + O(\sqrt{\epsilon})\Sigma_t$ using Lemma 3.2 on $\Sigma_t$.
return $\widehat{\Sigma} = \Sigma_T$.
```

Lemma B.1. *Under the same setting as Theorem 1.3, there exists universal constant C_0 such that Algorithm 4 outputs an estimate $\widehat{\Sigma}$ that satisfies $\left\|\widehat{\Sigma} - \Sigma\right\|_F = O(\sqrt{\epsilon}) \|\Sigma\|_2$ with high probability in time $\widetilde{O}(d^{3.26})/\mathrm{poly}(\epsilon)$.*

Proof. By Lemma 3.1, in Algorithm 4, we can compute $\Sigma_0 \succeq \Sigma$ such that $\|\Sigma_0\|_2 \leq \mathrm{poly}(d) \|\Sigma\|_2$. Lemma 3.2 allows us to iteratively compute Σ_{t+1} such that $\Sigma \preceq \Sigma_{t+1} \preceq \Sigma + O(\sqrt{\epsilon})\Sigma_t$. It follows that $\|\Sigma_{t+1}\|_2 \leq \|\Sigma\|_2 + O(\sqrt{\epsilon}) \|\Sigma_t\|_2$, and thus after $O(\log d)$ iterations we have a matrix Σ_T with $\|\Sigma_T\|_2 \leq 2 \|\Sigma\|_2$. Using Lemma 3.2 with Σ_T, we can get a matrix $\widehat{\Sigma}$ with $\|\widehat{\Sigma} - \Sigma\|_F = O(\sqrt{\epsilon}) \|\Sigma_T\|_2 = O(\sqrt{\epsilon})\Sigma$. The running time follows from the running time of Lemmas 3.1 and 3.2. □

Suppose the constant hiding in the $O(\cdot)$ notation in Lemma B.1 is C_0, we have the following immediate corollary of Lemma B.1.

Corollary B.2. *In Algorithm 3, the matrix M_0 satisfies $\|M_0 - \Sigma\|_F \leq C_0\sqrt{\epsilon} \|\Sigma\|_2$.*

In Algorithm 3 we will choose $C_1 = 20C_r$ and define S_1 to be the subspace where the eigenvalues of M_0 are at least $C_1\sqrt{\epsilon}$. We can then show the following lemma:

Lemma B.3. *In Algorithm 3, with high probability, the matrix M_1 satisfies*

$$\left\|M_1 - \Sigma[S_1^\perp]\right\|_F \le (2C_1 + C_0)C_0\epsilon \left\|\Sigma\right\|_2 .$$

Proof. We assume the calls to compute M_0, M_1 are successful, which happens with high probability.

By Lemma B.1, we know

$$\left\|M_1 - \Sigma[S_1^\perp]\right\|_F \le C_0\sqrt{\epsilon} \left\|\Sigma[S_1^\perp]\right\|_2 .$$

We continue to bound $\|\Sigma[S_1^\perp]\|_2$. Notice that by Corollary B.2,

$$\begin{aligned}\left\|\Sigma[S_1^\perp]\right\|_2 &\le \left\|M_0[S_1^\perp]\right\|_2 + C_0\sqrt{\epsilon}\left\|\Sigma\right\|_2 \\ &\le C_1\sqrt{\epsilon}\left\|M_0\right\|_2 + C_0\sqrt{\epsilon}\left\|\Sigma\right\|_2 \\ &\le (2C_1 + C_0)\sqrt{\epsilon}\left\|\Sigma\right\|_2 .\end{aligned}$$

Here the last step uses the fact when ϵ_0 is small enough $\|M_0\|_2 \le 2\|\Sigma\|$. □

Let S_2 denote the subspace of $S_1^\perp$ where the eigenvalues of $M_1[S_1^\perp]$ is at least $C_2\epsilon$ where $C_2 = 20(2C_1 + C_0)C_0$. Let S_3 denote the orthogonal subspace of $S_1 \oplus S_2$ (which corresponds to the eigenvectors of $M_1[S_1^\perp]$ that are smaller than $C_2\epsilon$). Note that S_1, S_2, and S_3 form a disjoint partition of $\mathbb{R}^d$. We will learn Σ separately on the product of these subspaces, and combine them together to get the final answer $\widehat{\Sigma}$. We use S_{12} to denote the subspace $S_1 \oplus S_2$.

We will now show that M_2 computed by Algorithm 3 has low additive error in the subspace S_{12}.

Lemma B.4. *In Algorithm 3, with high probability, the matrix M_2 satisfies*

$$\|M_2 - \Sigma[S_{12}]\|_F \le O(\epsilon\log(1/\epsilon)\left\|\Sigma\right\|_2 .$$

Moreover, if we consider $\Pi_{S_{12}}X_i$ as vectors of dimension equal to the dimension of S_{12}, the algorithm runs in time $\widetilde{O}(d^{3.26})/\operatorname{poly}(\epsilon)$.

Proof. We assume the calls for computing matrices M_0, M_1, M_2 are all successful, which happens with high probability.

Let $Y_i = \Pi_{S_{12}}X_i$, we know the covariance of these samples are exactly equal to $\Sigma[S_{12}]$.

In this case, by the guarantee of Theorem 1.2 we know

$$\left\|M_2^{-1/2}\Sigma[S_{12}]M_2^{-1/2} - I\right\|_F \le O(\epsilon\log(1/\epsilon)).$$

We can left and right multiply by $M_2^{1/2}$ and get

$$\|\Sigma[S_{12}] - M_2\|_F \le O(\epsilon\log(1/\epsilon))\left\|M_2\right\|_2 .$$

When ϵ is small enough this implies $\|M_2\|_2 \le 2\|\Sigma[S_{12}]\|_2 \le 2\|\Sigma\|_2$, therefore as desired we have

$$\|\Sigma[S_{12}] - M_2\|_F \le O(\epsilon\log(1/\epsilon))\left\|\Sigma\right\|_2 .$$

The only thing left to establish is the running time. To bound the running time we will show $\kappa(\Sigma[S_{12}]) = O(1/\epsilon)$. Here we restrict the attention to the subspace S_{12}, so if S_{12} as dimension k, κ is the ratio of the largest eigenvalue and the k-th eigenvalue.

Let $S^\star$ be the subspace of eigenvectors of Σ with eigenvalue at most $\epsilon\|\Sigma\|_2$. We first show the following claim:

Claim. *For any unit vector $v \in S^\star$, $\|\Pi_{S_{12}} v\|_2^2 \le 1/5$.*

Proof. Since $\|\Pi_{S_{12}} v\|_2^2 = \|\Pi_{S_1} v\|_2^2 + \|\Pi_{S_2} v\|_2^2$, we will bound the contributions separately. Notice that for any $v \in S^\star$, we have $v^\top \Sigma v \le \epsilon$, therefore by Corollary B.2,

$$v^\top M_0 v \le v^\top \Sigma v + \|\Sigma - M_0\|_2 \le (C_0\sqrt{\epsilon} + \epsilon)\|\Sigma\|_2 \le 2C_0\sqrt{\epsilon}\|\Sigma\|_2 .$$

On the other hand, $v^\top M_0 v \ge C_1\sqrt{\epsilon}\|\Sigma\|_2 \|\Pi_{S_1} v\|_2^2$. Combining the two equations we get $\|\Pi_{S_1} v\|_2^2 \le 2C_r/C_1 = 1/10$. The proof for $\|\Pi_{S_2} v\|_2^2$ is exactly the same except we use Lemma B.3. □

For any two subspaces U and V, one can check that

$$\sup_{u \in U, \|u\|_2 = 1} \|\Pi_V u\|_2^2 = \sup_{v \in V, \|v\|_2 = 1} \|\Pi_U v\|_2^2 = \cos^2 \inf_{u \in U, v \in V, \|u\| = \|v\| = 1} \angle(u, v),$$

where $\angle(u, v)$ is the angle between u, v. Therefore we know for any vector $v \in S_{12}$, $\|\Pi_{S^\star} v\|_2^2 \le 1/5$. This implies for every $v \in S_{12}$,

$$v^\top \Sigma v \ge \epsilon \|\Sigma\|_2 \|\Pi_{Z^\perp} v\|_2^2 \ge \frac{4\epsilon}{5}.$$

This shows $\lambda_k(\Sigma_{S_{12}}) \ge \frac{4\epsilon}{5}\|\Sigma\|_2$, so $\kappa \le \frac{5}{4\epsilon} = O(1/\epsilon)$. □

Finally we give the guarantee for M_3.

Lemma B.5. *In Algorithm 3, with high probability, the matrix M_3 satisfies*

$$\left\| \frac{1}{\sqrt{\epsilon}} M_3[S_1, S_3] - \Sigma[S_1, S_3] \right\|_F \le O(\epsilon)\|\Sigma\|_2 \ .$$

Proof. We assume the calls to compute M_0, M_1, M_3 are successful, which happens with high probability.

Set $Y_i = (\epsilon^{1/2}\Pi_{S_1} + \epsilon^{1/4}\Pi_{S_2} + \Pi_{S_3})X_i$. Let Σ_Y denote the covariance of Y. It is easy to check that $\Sigma_Y \preceq 3(\epsilon\Sigma[S_1] + \sqrt{\epsilon}\Sigma[S_2] + \Sigma[S_3])$. By Lemma B.2,

$$\|\Sigma[S_2]\|_2 \le \left\|\Sigma[S_1^\perp]\right\|_2 \le O(\sqrt{\epsilon})\|\Sigma\|_2 + \left\|M_0[S_1^\perp]\right\|_2 = O(\sqrt{\epsilon})\|\Sigma\|_2 .$$

Similarly by Lemma B.3,

$$\|\Sigma[S_3]\|_2 \le O(\epsilon)\|\Sigma\|_2 + \|M_1[S_3]\|_2 = O(\epsilon)\|\Sigma\|_2 .$$

Combining these we have $\|\Sigma_Y\|_2 \le O(\epsilon)\|\Sigma\|_2$. Therefore by Lemma B.1 we know the estimation M_3 satisfies $\|M_3 - \Sigma_Y\|_F \le O(\epsilon^{1.5})\|\Sigma\|_2$.

On the other hand, it is easy to check that $\Sigma_Y[S_1, S_3] = \frac{1}{\sqrt{\epsilon}}\Sigma[S_1, S_3]$, therefore we have

$$\begin{aligned} \left\| \frac{1}{\sqrt{\epsilon}} M_3[S_1, S_3] - \Sigma[S_1, S_3] \right\|_F &= \frac{1}{\sqrt{\epsilon}} \|M_3[S_1, S_3] - \Sigma_Y[S_1, S_3]\|_F \\ &\le \frac{1}{\sqrt{\epsilon}} \|M_3 - \Sigma_Y\|_F = O(\epsilon)\|\Sigma\|_2 . \end{aligned}$$

□

Finally we are ready to combine all the steps.

Proof of Theorem 1.3. We will assume the four calls to Algorithm 1 and Algorithm 4 are all successful, which happens with high probability. The running time follows from Lemma B.1 and Lemma B.4. Now the resulting matrix looks like

$$\begin{bmatrix} M_2[S_1] & M_2[S_1, S_2] & \frac{1}{\sqrt{\epsilon}} M_3[S_1, S_3] \\ M_2[S_2, S_1] & M_1[S_2] & M_1[S_2, S_3] \\ \frac{1}{\sqrt{\epsilon}} M_3[S_3, S_1] & M_1[S_3, S_2] & M_1[S_3] \end{bmatrix}$$

By Lemmas B.3, B.4, B.5 we know for each one of these nine blocks the error is bounded by $O(\epsilon \log(1/\epsilon)) \|\Sigma\|_2$. Therefore, the entire matrix also has error at most $O(\epsilon \log(1/\epsilon)) \|\Sigma\|_2$. □

B.2 Robust Mean Estimation for Bounded-Covariance Distributions

We use the robust mean estimation algorithm for bounded-covariance distributions from [CDG19] to achieve Lemma 3.4.

We state this algorithm (Algorithm 5) to be self-contained.

```
Algorithm 5: Robust Mean Estimation for Bounded Covariance Distributions
Input  : $0 < \epsilon < \epsilon_0$, and an $\epsilon$-corrupted set of $N = \widetilde{\Omega}(d/\epsilon)$ samples $(Z_i)_{i=1}^N$ drawn from $D$.
         $D$ is the ground-truth distribution supported on $\mathbb{R}^d$ with mean $\mu^\star$ and covariance
$\Sigma \preceq I$.
Output: A vector $\widehat{\mu} \in \mathbb{R}^d$ such that, with high probability, $\|\widehat{\mu} - \mu^\star\|_2 \le O(\sqrt{\epsilon})$.
Let $\nu \in \mathbb{R}^d$ be an initial guess with $\|\nu - \mu^\star\|_2 \le \mathrm{poly}(d)$.
for $i = 1$ to $O(\log d)$ do
    Compute a near-optimal solution $w \in \mathbb{R}^N$ to the primal SDP (1) with parameters $\nu$ and
     $2\epsilon$.
    Compute a near-optimal solution $M \in \mathbb{R}^{d \times d}$ for the dual SDP (2) with parameters $\nu$ and
     $\epsilon$.
    if the objective value of $w$ in SDP (1) is at most $c$ ($c$ is a universal constant) then
        return the weighted empirical mean $\widehat{\mu}_w = \sum_{i=1}^N w_i Z_i$.
    else
        Move $\nu$ closer to $\mu^\star$ using the top eigenvector of $M$.
```

Notice that Algorithm 5 is almost identical to Algorithm 2, except the stopping criteria in the "if" statement. Therefore, we can speed up Algorithm 5 using Proposition 3.6, as we do for Algorithm 2.

B.3 Robust Mean Estimation with Approximately Known Covariance

In this section, we prove the error guarantee part of Lemma 3.5, i.e., correctness of Algorithm 2. Note that we will not worry about running time here, so we can use the naive implementation of Algorithm 2 which runs in time $\widetilde{O}(d^4)/\mathrm{poly}(\epsilon)$. For the same reason, we ignore the additional structure in our input and focus on the mean estimation problem. For the rest of this section, we use d to denote the dimensionality of the problem, and $N = \widetilde{\Omega}(d/\epsilon^2)$ to denote the number of samples. (We have $d = (d')^2$ if we are trying to estimate the covariance matrix of a (d')-dimensional Gaussian.)

We use $(X_i)_{i=1}^N$ to denote the input, which is a set of d-dimensional ϵ-corrupted samples drawn from some ground-truth distribution D. We know D has covariance matrix Σ with $\|\Sigma - I\|_2 \le \tau$, and the goal is to estimate the unknown mean $\mu^\star$ of D. We first restate Lemma 3.5.

Lemma 3.5 *Let D be a distribution supported on $\mathbb{R}^d$ with unknown mean $\mu^\star$ and covariance Σ. Let $0 < \epsilon < \epsilon_0$ for some universal constant ϵ_0, $\tau \le O(\sqrt{\epsilon})$, and $\delta = O(\sqrt{\tau\epsilon} + \epsilon \log(1/\epsilon))$. Suppose that D has exponentially decaying tails, and Σ is close to the identity matrix $\|\Sigma - I\|_2 \le \tau$. Given an ϵ-corrupted set of $N = \widetilde{\Omega}(d/\delta^2)$ samples drawn from D, Algorithm 2 outputs a hypothesis vector $\widehat{\mu}$ such that, with high probability, $\|\mu - \mu^\star\|_2 \le O(\delta)$.*

We use $G^\star$ for the original set of N good samples drawn from D. After ϵ-fraction of the samples are corrupted, we use $G \subseteq G^\star$ for the remaining good samples and B for the corrupted samples. The input to the algorithm is $G \cup B$. We have $|G| \ge (1-\epsilon)N$ and $|B| \le \epsilon N$. Let $\Delta_{N,\epsilon}$ denote the convex hull of all uniform distributions over subsets $S \subseteq [N]$ of size $|S| = (1-\epsilon)N$:

$$\Delta_{N,\epsilon} = \left\{ w \in \mathbb{R}^N : \sum_{i=1}^N w_i = 1 \text{ and } 0 \le w_i \le \frac{1}{(1-\epsilon)N} \text{ for all } i \right\} .$$

Every weight vector $w \in \Delta_{N,\epsilon}$ correspond to a fractional set of $(1-\epsilon)N$ samples.

By standard concentration results, we know that degree-2 polynomials of Gaussian random variables are exponentially concentrated around their mean.

Definition B.6 (Exponentially Decaying Tails)**.** *We say a distribution D supported on $\mathbb{R}^d$ has exponentially decaying tails iff, for any unit vector $v \in \mathbb{R}^d$, we have $\Pr_{Z\sim D}[\langle v, Z - \mu^\star \rangle \ge t] \le \exp(-\Omega(t))$.*

To avoid dealing with the randomness of the good samples, we require the following deterministic conditions on the original set of N good samples $G^\star$ (which hold with high probability when $N = \widetilde{\Omega}(d/\epsilon^2)$ when D satisfies Definition B.6). For all $w \in \Delta_{N,2\epsilon}$, we require the following conditions to hold for $\delta_1 = O(\epsilon \log(1/\epsilon))$ and $\delta_2 = O(\tau + \epsilon \log^2(1/\epsilon))$:

$$\left\| \sum_{i \in G^\star} w_i (X_i - \mu^\star) \right\|_2 \le \delta_1 , \quad \left\| \sum_{i \in G^\star} w_i (X_i - \mu^\star)(X_i - \mu^\star)^\top - I \right\|_2 \le \delta_2 , \tag{7}$$

$$\forall i \in G^\star,\ \|X_i - \mu^\star\|_2 \le O(\sqrt{d \log(d)}) . \tag{8}$$

At a high level, they state that with high probability, the good samples are never too far from $\mu^\star$, and the empirical first and second moments of the good samples behave as we expect them to. More specifically, δ_1 upper bounds the change in the mean when we remove *any* ϵ-fraction of the samples, and δ_2 upper bounds the change in the second-moment matrix. The second-order condition follows from the fact that $\|\Sigma - I\|_2 \le \tau$, the triangle inequality for the spectral norm, and with high probability for our choice of N,

$$\left\| \sum_{i \in G^\star} w_i (X_i - \mu^\star)(X_i - \mu^\star)^\top - \Sigma \right\|_2 \le O(\epsilon \log^2(1/\epsilon)) \, \|\Sigma\|_2 = O(\epsilon \log^2(1/\epsilon)) .$$

We adapt the proof of [CDG19] to prove the following lemma, which holds for general distributions that satisfy the concentration bounds above.

Lemma B.7. *Assume the concentration bounds (Conditions* (7) *and* (8)*) hold for the good samples with parameters δ_1 and δ_2 where $\delta_2 \geq \delta_1^2$. Let $\delta = \sqrt{\epsilon\delta_2}$. Then Algorithm 2, with threshold $(1 + O(\delta_2))$ in the "if" statement, will output a weight vector w such that the weighted empirical mean $\widehat{\mu}_w = \sum_{i=1}^N w_i X_i$ satisfies $\|\mu - \widehat{\mu}\|_2 \leq O(\delta)$ for $\delta = O(\delta)$.*

Lemma 3.5 follows immediately from Lemma B.7, because the output of Algorithm 2 has error $\delta = O(\sqrt{\epsilon\delta_2}) = O(\sqrt{\epsilon(\tau + \epsilon \log^2(1/\epsilon))}) = O(\sqrt{\epsilon\tau} + \epsilon\log(1/\epsilon))$ as needed.

Algorithm 2 is based on the primal-dual approach proposed by [CDG19] for robust mean estimation. Their algorithm starts with a guess $\nu \in \mathbb{R}^d$, and then in each iteration solves the primal and dual SDPs (1) and (2). They gave a win-win analysis: either a good primal solution gives weights $w \in \mathbb{R}^N$ such that $\widehat{\mu}_w$ is close to the true mean; or a good dual solution must identify a direction of improvement that allows the algorithm to move ν much closer to the true mean.

$$\begin{array}{ll} \text{minimize} & \lambda_{\max}\left(\sum_{i=1}^N w_i (X_i - \nu)(X_i - \nu)^\top\right) \\ \text{subject to} & w \in \Delta_{N,\epsilon} \end{array} \tag{1}$$

$$\begin{array}{ll} \text{maximize} & \text{average of the smallest } (1-\epsilon)\text{-fraction of } \left((X_i - \nu)^\top M (X_i - \nu)\right)_{i=1}^N \\ \text{subject to} & M \succeq 0, \operatorname{tr}(M) \leq 1 \end{array} \tag{2}$$

To prove Lemma B.7, we will show that the win-win analysis still holds in our setting by proving two structural lemmas. Lemma B.9 proves that a good primal solution for *any* guess ν will give an accurate weighted empirical mean. Lemma B.10 shows that we can use the top eigenvector of a near-optimal dual solution to move ν closer to $\mu^\star$ by a constant factor.

First we prove a helper lemma. Lemma B.8 gives upper and lower bounds on the optimal value of the SDPs (1) and (2). For example, Lemma B.8 allows us to estimate how far ν is from $\mu^\star$ from the optimal value of the SDPs.

Lemma B.8 (Optimal Value of the SDPs)**.** *Fix $0 < \epsilon < \epsilon_0$, δ_1, $\delta_2 \geq \delta_1^2$, and $\nu \in \mathbb{R}^d$. Let $\delta = \sqrt{\epsilon\delta_2}$. Let $\{X_i\}_{i=1}^N$ be an ϵ-corrupted set of N samples that satisfy Condition* (7)*. Let $\mathrm{OPT}_{\nu,\epsilon}$ denote the optimal value of the SDPs* (1) *and* (2) *with parameters ν and ϵ. Let $r = \|\nu - \mu^\star\|_2$. Then, we have:*

$$(1 - 2\epsilon)\left((1 - \delta_2) + r^2 - 2\delta_1 r\right) \leq \mathrm{OPT}_{\nu,\epsilon} \leq (1 + \delta_2) + r^2 + 2\delta_1 r\ .$$

In particular, when $\epsilon_0 < 1/20$ and $r = \Omega(\sqrt{\delta_2})$, we can simplify the above as

$$1 + 0.9r^2 \leq \mathrm{OPT}_{\nu,\epsilon} \leq 1 + 1.1r^2\ .$$

Proof. Let $\mathrm{OPT} = \mathrm{OPT}_{\nu,\epsilon}$.

One feasible primal solution is to set $w_i = \frac{1}{|G|}$ for all $i \in G$ (and $w_i = 0$ for all $i \in B$):

$$\begin{aligned} \mathrm{OPT} \leq \lambda_{\max}\left(\sum_{i=1}^N w_i (X_i - \nu)(X_i - \nu)^\top\right) &= \max_{y \in \mathbb{R}^d, \|y\|_2 = 1} \sum_{i \in G} w_i \langle X_i - \nu, y\rangle^2 \\ &= \max_{y \in \mathbb{R}^d, \|y\|_2 = 1} \left(\sum_{i \in G} w_i \langle X_i - \mu^\star, y\rangle^2 + \langle \mu^\star - \nu, y\rangle^2 + 2\langle \sum_{i \in G} w_i (X_i - \mu^\star), y\rangle \langle \mu^\star - \nu, y\rangle\right) \\ &\leq \max_{y \in \mathbb{R}^d, \|y\|_2 = 1} \left((1 + \delta_2) + \langle \mu^\star - \nu, y\rangle^2 + 2\delta_1 \langle \mu^\star - \nu, y\rangle\right) \\ &= (1 + \delta_2) + \|\mu^\star - \nu\|_2^2 + 2\delta_1 \|\mu^\star - \nu\|_2\ . \end{aligned}$$

We used Condition (7), since w can be viewed as a weight vector on $G^\star$ where $w \in \Delta_{N,\epsilon}$.

One feasible dual solution is $M = yy^\top$ where $y = \frac{\mu^\star - \nu}{\|\mu^\star - \nu\|_2}$. The dual objective value is the mean of the smallest $(1-\epsilon)$-fraction of $\left((X_i - \nu)^\top M (X_i - \nu)\right)_{i=1}^N$, which is at least

$$\frac{1}{(1-\epsilon)N} \min_{S \subset G, |S| = (1-2\epsilon)N} \sum_{i \in S} (X_i - \nu)^\top M (X_i - \nu) \; .$$

This is because the smallest $(1-\epsilon)N$ entries in G must include the smallest $(1-2\epsilon)N$ entries. Let $w'_i = \frac{1}{|S|}$ for all $i \in S$ and $w'_i = 0$ otherwise. Note that $S \subset G$ and $|S| = (1-2\epsilon)N$, so w' can be viewed as a weight vector on $G^\star$ with $w' \in \Delta_{N,2\epsilon}$. Therefore we have

$$\begin{aligned}
\mathrm{OPT} &\geq \sum_{i \in S} \frac{1}{(1-\epsilon)N} (X_i - \nu)^\top M (X_i - \nu) = \frac{|S|}{(1-\epsilon)N} \sum_{i \in G} w'_i \langle X_i - \nu, y \rangle^2 \\
&= \frac{1-2\epsilon}{1-\epsilon} \left(\sum_{i \in G} w'_i \langle X_i - \mu^\star, y \rangle^2 + w'_G \left\|\mu^\star - \nu\right\|_2^2 + 2 \sum_{i \in G} w'_i \langle X_i - \mu^\star, y \rangle \left\|\mu^\star - \nu\right\|_2 \right) \\
&\geq (1-2\epsilon) \left((1 - \delta_2) + \left\|\mu^\star - \nu\right\|_2^2 - 2\delta_1 \left\|\mu^\star - \nu\right\|_2 \right) \; .
\end{aligned}$$

To obtain the simpler upper and lower bounds, we note when $r = \Omega(\sqrt{\delta_2})$, the error term is $\delta_2 + 2\delta_1 r = O(r^2)$. Therefore, by increasing the constant in $r = \Omega(\sqrt{\delta_2})$, we can get $1 + 0.9r^2 \leq \mathrm{OPT} \leq 1 + 1.1r^2$. □

Next we show that a good primal solution w for any guess ν will give an accurate estimate $\widehat{\mu}_w$. Lemma B.9 proves the contrapositive statement: if the weighted empirical mean $\widehat{\mu}_w$ is far from $\mu^\star$, then no matter what our current guess ν is, w cannot be a good solution to the primal SDP.

Lemma B.9 (Good Primal Solution ⇒ Correct Mean)**.** *Fix $0 < \epsilon < \epsilon_0$, δ_1, and $\delta_2 \geq \delta_1^2$. Let $\delta = \sqrt{\epsilon \delta_2}$. Let $\{X_i\}_{i=1}^N$ be an ϵ-corrupted set of N samples that satisfy Condition (7). For all $w \in \Delta_{N,2\epsilon}$, if $\left\|\widehat{\mu}_w - \mu^\star\right\|_2 = \Omega(\delta)$ where $\widehat{\mu}_w = \sum_{i=1}^N w_i X_i$, then for all $\nu \in \mathbb{R}^d$,*

$$\lambda_{\max} \left(\sum_{i=1}^N w_i (X_i - \nu)(X_i - \nu)^\top \right) \geq 1 + \Omega(\delta_2) \; .$$

Proof. Fix any $w \in \Delta_{N,2\epsilon}$. If $\left\|\mu^\star - \nu\right\|_2 = \Omega(\sqrt{\delta_2})$, then because w is feasible and by Lemma B.8,

$$\lambda_{\max} \left(\sum_{i=1}^N w_i (X_i - \nu)(X_i - \nu)^\top \right) \geq \mathrm{OPT}_{\nu,2\epsilon} \geq 1 + 0.9 \left\|\mu^\star - \nu\right\|_2^2 \geq 1 + \Omega(\delta_2) \; .$$

Therefore, for the rest of this proof, we can assume $\left\|\mu^\star - \nu\right\|_2 = O(\sqrt{\delta_2})$.

We project the samples along the direction of $(\widehat{\mu}_w - \mu^\star)$. Consider the unit vector $y = (\widehat{\mu}_w - \mu^\star)/\left\|\widehat{\mu}_w - \mu^\star\right\|_2$. To bound from below the maximum eigenvalue, it is sufficient to show that

$$y^\top \left(\sum_{i=1}^N w_i (X_i - \nu)(X_i - \nu)^\top \right) y = \sum_{i=1}^N w_i \langle X_i - \nu, y \rangle^2 \geq 1 + \Omega(\delta_2) \; .$$

We first bound from below the contribution of the bad samples by $\Omega(\delta_2)$. By triangle inequality,

$$\begin{aligned}\left|\sum_{i\in B} w_i\langle X_i-\nu, y\rangle\right| &\geq \left|\sum_{i\in B} w_i\langle X_i-\mu^\star, y\rangle\right| - w_B\left|\langle \mu^\star-\nu, y\rangle\right| \\ &\geq \left|\sum_{i=1}^{N} w_i\langle X_i-\mu^\star, y\rangle\right| - \left|\sum_{i\in G} w_i\langle X_i-\mu^\star, y\rangle\right| - 2\epsilon\left\|\mu^\star-\nu\right\|_2 \\ &\geq \left\|\widehat{\mu}_w-\mu^\star\right\|_2 - \delta_1 - 2\epsilon\cdot O(\sqrt{\delta_2}) = \Omega(\delta)\ .\end{aligned}$$

The last line follows from our choice of y, $\delta \geq \max(\delta_1, \epsilon\sqrt{\delta_2})$, and the good samples satisfy Condition (7). By Cauchy-Schwarz,

$$\left(\sum_{i\in B} w_i\langle X_i-\nu, y\rangle^2\right)\left(\sum_{i\in B} w_i\right) \geq \left(\sum_{i\in B} w_i\langle X_i-\nu, y\rangle\right)^2 = \Omega(\delta^2)\ .$$

Since $w_B \leq 2\epsilon$, we have $\sum_{i\in B} w_i\langle X_i-\nu, y\rangle^2 = \Omega(\delta^2/\epsilon) = \Omega(\delta_2)$.

We continue to lower bound the contribution of the good samples to the quadratic form by $1-O(\delta_2)$. By Condition (7),

$$\begin{aligned}\sum_{i\in G} w_i\langle X_i-\nu, y\rangle^2 &= \sum_{i\in G} w_i\left(\langle X_i-\mu^\star, y\rangle^2 + \langle \mu^\star-\nu, y\rangle^2 + 2\langle X_i-\mu^\star, y\rangle\langle \mu^\star-\nu, y\rangle\right) \\ &\geq \sum_{i\in G} w_i\langle X_i-\mu^\star, y\rangle^2 + 2\langle \mu^\star-\nu, y\rangle\langle\sum_{i\in G} w_i(X_i-\mu^\star), y\rangle \\ &\geq (1-\delta_2) - 2\delta_1\left\|\mu^\star-\nu\right\|_2 = 1-O(\delta_2)\ .\end{aligned}$$

In the last step, we used $\delta_1\left\|\mu^\star-\nu\right\|_2 \leq 2\delta_1\sqrt{\delta_2} = O(\delta_2)$. Putting together the contribution of good and bad samples, we have $\sum_{i=1}^{N} w_i\langle X_i-\nu, y\rangle^2 \geq 1-O(\delta_2)+\Omega(\delta_2) = 1+\Omega(\delta_2)$. □

Lemma B.9 guarantees that, any solution to the primal SDP whose objective value is at most $1+O(\delta_2)$ will give good weights, and this is independent of our current guess ν.

We now deal with the other possibility: the primal SDP has no good solution. Lemma B.10 shows that in this case, we can solve the dual SDP (2) and move ν closer to $\mu^\star$ by a constant factor. We simplify the proof by assuming that we can solve the dual SDP exactly. This assumption is wlog as shown in [CDG19].

Lemma B.10 (Good Dual Solution ⇒ Better ν)**.** *Fix $0<\epsilon<\epsilon_0$, δ_1, and $\delta_2 \geq \delta_1^2$. Let $\delta = \sqrt{\epsilon\delta_2}$. Let $\{X_i\}_{i=1}^N$ be an ϵ-corrupted set of N samples that satisfy Condition (7). If the optimal solution $M \in \mathbb{R}^{d\times d}$ to the dual SDP (2) (with parameters ν and ϵ) has objective value at least $1+\Omega(\delta_2)$, then we can efficiently find a vector $\nu' \in \mathbb{R}^d$, such that $\left\|\nu'-\mu^\star\right\|_2 \leq \frac{3}{4}\left\|\nu-\mu^\star\right\|_2$.*

Proof. Because M is a feasible solution to the dual SDP (2) with parameters ν and ϵ, we know that $\mathrm{OPT}_{\nu,\epsilon} \geq 1+\Omega(\delta_2)$. When $\mathrm{OPT}_{\nu,\epsilon} \geq 1+\Omega(\delta_2)$, Lemma B.8 implies that $\left\|\mu^\star-\nu\right\|_2 \geq \Omega(\sqrt{\delta_2})$ and $\mathrm{OPT}_{\nu,\epsilon} \geq 1+0.9\left\|\mu^\star-\nu\right\|_2^2$.

We know $M \succeq 0$ and $\mathrm{tr}(M)=1$. Without loss of generality, we can assume M is symmetric.

Since the objective value is the average of the smallest $(1-\epsilon)N$ entries of $(X_i-\nu)^\top M(X_i-\nu)$ and one way to choose $(1-\epsilon)N$ entries is to focus on the good samples, using Condition (7),

$$\begin{aligned}
1+0.9\left\|\mu^\star-\nu\right\|_2^2 &\le \mathrm{OPT}_{\nu,\epsilon} \\
&\le \frac{1}{|G|}\sum_{i\in G}(X_i-\nu)^\top M(X_i-\nu) \\
&= \frac{1}{|G|}\sum_{i\in G}\langle M,(X_i-\mu^\star)(X_i-\mu^\star)^\top+2(X_i-\mu^\star)(\mu^\star-\nu)^\top+(\mu^\star-\nu)(\mu^\star-\nu)^\top\rangle \\
&\le 1+\delta_2+2\delta_1\left\|\mu^\star-\nu\right\|_2+\langle M,(\mu^\star-\nu)(\mu^\star-\nu)^\top\rangle \\
&\le 1+0.1\left\|\mu^\star-\nu\right\|_2^2+\langle M,(\mu^\star-\nu)(\mu^\star-\nu)^\top\rangle\ .
\end{aligned}$$

Therefore, we have $\langle M,(\mu^\star-\nu)(\mu^\star-\nu)^\top\rangle\ge\frac{4}{5}\left\|\mu^\star-\nu\right\|_2^2$.

We will continue to show that the top eigenvector of M aligns with $(\nu-\mu^\star)$. Let $\lambda_1\ge\lambda_2\ge\ldots\ge\lambda_d\ge0$ denote the eigenvalues of M, and let $v_1,\ldots,v_d$ denote the corresponding eigenvectors. The conditions on M implies that $\sum_{i=1}^d\lambda_d=1$. We decompose $(\mu^\star-\nu)$ and write it as $\mu^\star-\nu=\sum_{i=1}^d\alpha_iv_i$ where $\sum_{i=1}^d\alpha_i^2=\left\|\mu^\star-\nu\right\|_2^2$. Using these decompositions, we can rewrite $\langle M,(\mu^\star-\nu)(\mu^\star-\nu)^\top\rangle=\sum_{i=1}^d\lambda_i\alpha_i^2$.

First observe that $\lambda_1\ge\frac{4}{5}$, because $\lambda_1\sum_i\alpha_i^2\ge\sum_i\lambda_i\alpha_i^2\ge\frac{4}{5}\left\|\mu^\star-\nu\right\|_2^2=\frac{4}{5}\sum_i\alpha_i^2$. Moreover, because $\frac{4}{5}\sum_i\alpha_i^2\le\sum_i\lambda_i\alpha_i^2\le\lambda_1\alpha_1^2+(1-\lambda_1)(1-\alpha_1^2)\le\frac{4}{5}\alpha_1^2+\frac{1}{5}\sum_i\alpha_i^2$, we know that $\langle v_1v_1^\top,(\mu^\star-\nu)(\mu^\star-\nu)^\top\rangle=\alpha_1^2\ge\frac{3}{4}\sum_i\alpha_i^2$. Thus, we have a unit vector $v_1\in\mathbb{R}^d$ with $\langle v_1,\mu^\star-\nu\rangle=\alpha_1\ge\frac{\sqrt{3}}{2}\left\|\mu^\star-\nu\right\|_2$, so the angle between v_1 and $\mu^\star-\nu$ is at most $\theta\le\cos^{-1}(\frac{\sqrt{3}}{2})$.

Finally, we can estimate r from the value of $\mathrm{OPT}_{\nu,\epsilon}$ using Lemma B.8, and move ν in a direction almost aligned with $\mu^\star-\nu$, to obtain a new point ν' that is on a circle of radius $r'\approx r$ centered at ν. A basic geometric analysis shows that $\left\|\nu'-\mu^\star\right\|_2\le\frac{3}{4}\left\|\nu-\mu^\star\right\|_2$. □