

Quantum Entropy Scoring for Fast Robust Mean Estimation and Improved Outlier Detection: Supplementary Material

May 25, 2019

Abstract

We study two problems in high-dimensional robust statistics: *robust mean estimation* and *outlier detection*. In robust mean estimation the goal is to estimate the mean μ of a distribution on \mathbb{R}^d given n independent samples, an ϵ -fraction of which have been corrupted by a malicious adversary. In outlier detection the goal is to assign an *outlier score* to each element of a data set such that elements more likely to be outliers are assigned higher scores. Our algorithms for both problems are based on a new outlier scoring method we call QUE-scoring based on *quantum entropy regularization*. For robust mean estimation, this yields the first algorithm with optimal error rates and nearly-linear running time $\tilde{O}(nd)$ in all parameters, improving on the previous fastest running time $\tilde{O}(\min(nd/\epsilon^6, nd^2))$. For outlier detection, we evaluate the performance of QUE-scoring via extensive experiments on synthetic and real data, and demonstrate that it often performs better than previously proposed algorithms.

1 Introduction

We study outlier-robust statistics in high dimensions, focusing on the question: *can theoretically sound outlier robust algorithms have practical running times for large, high-dimensional data sets?* We address two related problems: *robust mean estimation*, which is primarily theoretical, and an applied counterpart, *outlier detection*.

Robust mean estimation Our main theoretical contribution is the first nearly-linear time algorithm for robust mean estimation with nearly-optimal error. Here the goal is to estimate the mean $\mu \in \mathbb{R}^d$ of a d -dimensional distribution D given ϵ -corrupted samples X_1, \dots, X_n – that is, i.i.d. samples, an unknown ϵ -fraction of which have been maliciously corrupted. Under (for instance) the assumption that the covariance of D is bounded by Id , it has been long known to be possible in exponential time to estimate μ by $\hat{\mu}$ having $\|\mu - \hat{\mu}\|_2 \leq O(\sqrt{\epsilon})$. In particular, this rate of error is independent of d .

Polynomial-time algorithms provably achieving such d -independent error became known only recently, starting with the works [1, 2]. Until our work, the running time of algorithms with provably d -independent error remained suboptimal by polynomial factors in d or ϵ : the fastest running time achieved before this work was $\tilde{O}(\min(nd^2, nd/\epsilon^6))$ [3, 1, 2, 4]. (Here $\tilde{O}(\cdot)$ notation hides logarithmic factors in n and d). While these running times represent a dramatic improvement over previous exponential-time algorithms, there are still many interesting regimes where the additional runtime overheads these algorithms incur render them impractically slow. *We give the first algorithm for robust mean estimation with running time $\tilde{O}(nd)$ which achieves error $\|\mu - \hat{\mu}\|_2 \leq O(\sqrt{\epsilon})$.* Note that this running time is nearly-linear in the input size nd . Similar to prior works, our algorithm has information-theoretically optimal sample complexity and nearly-optimal error rates in both the bounded-covariance and sub-Gaussian regimes.

Outlier detection Our main applied contribution is a new algorithm for high-dimensional outlier detection, which we assess via experiments on both synthetic and real data. Our goal is to take a dataset $X_1, \dots, X_n \in \mathbb{R}^d$ and assign to each X_i an *outlier score* $\tau_i \geq 0$, so that higher scores τ_i are assigned to points X_i more likely to be outliers. Of course, what constitutes an outlier varies across applications, so no single algorithm for outlier detection is likely to be the best in all domains. We show that our method performs

well in settings where individual outliers are difficult to pick out on their own (by, say, their ℓ_2 norms or their distances to nearby points), but still collectively bias empirical statistics such as the mean and covariance.

We compare our method to baselines based on PCA and Euclidean distances, as well as more sophisticated algorithms from existing literature based on nearest-neighbor distances. Our algorithm has nearly-linear running time in theory, and simple implementations in practice incur minimal overhead beyond standard spectral methods, allowing us to run on 10^3 -dimensional data with no special optimizations. It can therefore be used in practice to complement existing approaches to outlier detection in exploratory data analysis.

1.1 What is an outlier and why are they hard to find?

For us, an outlier is an element of a data set which was generated according to a different process than the majority of the data. For instance, we may imagine that our samples X_1, \dots, X_n were sampled i.i.d. from a distribution $(1 - \epsilon)D + \epsilon N$ over \mathbb{R}^d , where D is the distribution of inliers, N is the distribution of outliers, and $\epsilon > 0$ is a small number – that is, we imagine that a *constant fraction* of our data may be outliers.

For this discussion, we also informally imagine that N is sufficiently distinct from D that the set of outliers could be approximately identified by brute-force search over subsets of $(1 - \epsilon)n$ samples, if given unlimited computational resources. Otherwise, outlier detection is not a meaningful problem, and robust mean estimation is easy (because the empirical mean will be a good estimator). Under these circumstances, what makes identifying outliers and estimating the mean in their presence difficult? Chiefly:

Outliers may not be identifiable in isolation. On its own, a typical outlier $X_i \sim N$ may look much like a typical inlier $X_j \sim D$. For instance, it could be $\|X_i\|_2 \approx \|X_j\|_2$, and X_i, X_j may have similar distance to the nearest few neighboring samples, especially in high dimensions where samples are far apart.

Outliers still introduce bias, collectively Even if individual outliers look innocuous, the collective effect a modified ϵ -fraction of samples X_i can still substantially change the empirical distribution of X_1, \dots, X_n . As a result, even simple statistical tasks like estimating the mean or covariance of D require sophisticated estimators: naively pruning individual outliers and then employing standard empirical estimators typically leads to far-suboptimal error rates. For example, an ϵ -fraction of X_1, \dots, X_n which are all slightly biased in a single direction may shift the empirical mean of X_1, \dots, X_n , but this bias will be difficult to detect by looking at small numbers of samples at once. This also demonstrates that successful outlier detection can require global geometric information about a high-dimensional dataset, such as whether or not a direction exists in which many (say, ϵn) samples are unusually biased.

Outliers may be inhomogeneous. Outliers need not exhibit unusual bias in only one direction, or all have the same norm, or lie in a single cluster. Rather, if a dataset exhibits several forms of corruption, there may be as many different-looking kinds of outliers. In the theoretical robust mean estimation setting, the adversary producing ϵ -corrupted samples may corrupt $\epsilon n/10$ samples by biasing them in some direction, another $\epsilon n/10$ samples by unusually enlarging their norms, and so forth.

Since robust mean estimation involves a malicious adversary, all of the above phenomena must be addressed by our robust mean estimation algorithm. In the empirical section of this paper, we focus on designing an outlier detection method suited to situations where at least one of them occurs – in other cases, existing methods (such as those based on Euclidean norms or local neighborhoods of individual samples [5]) may be more appropriate.

1.2 QUE: Quantum Entropy Scoring

Recent innovations in robust mean estimation [2, 1] rely on the following crucial observation about ϵ -corrupted samples X_1, \dots, X_n from a distribution D with covariance $\Sigma \preceq \text{Id}$. Namely: *any subset $S \subseteq \{X_1, \dots, X_n\}$ of samples which shift the empirical mean by distance more than $\sqrt{\epsilon}$ in some direction v also introduce an eigenvalue of magnitude greater than 1 to the empirical covariance.*

In robust mean estimation, this leads to (amongst others) the *filter* algorithm of [1, 4], one of the first to achieve dimension-independent error rates. Roughly speaking, the algorithm iterates the following until the empirical covariance $\bar{\Sigma}$ has small spectral norm: (1) compute the top eigenvector v of the empirical covariance of $\bar{\Sigma}$, then (2) throw out samples X_i whose projections $|\langle X_i - \bar{\mu}, v \rangle| \gg 1$ is unusually large, where

$\bar{\mu}$ is the empirical mean of the corrupted dataset. For outlier detection this suggests a natural scoring rule – let the outlier score τ_i of sample X_i be proportional to $|\langle X_i - \bar{\mu}, v \rangle|$.

The main drawback of these algorithms is that they do not adequately account for inhomogeneity of outliers. For the filter, this leads to a worst-case running time of $\tilde{O}(nd^2)$, because the filter operation (which can be implemented in $\tilde{O}(nd)$ time) may have to be repeated as many as d times if the adversary introduces outliers lying in d orthogonal directions. The rule $\tau_i = |\langle X_i - \bar{\mu}, v \rangle|$ may miss outliers causing a large eigenvalue of $\bar{\Sigma}$, but in a direction orthogonal to the top eigenvector v .

In the opposite extreme, if outliers are maximally inhomogeneous – no group of them is unusually biased in some shared direction v – then the only way they can bias the empirical mean is for the individual ℓ_2 norms $\|X_i - \bar{\mu}\|_2$ to be larger than typical. This suggests a different scoring rule: $\tau_i = \|X_i - \bar{\mu}\|_2$. But this approach breaks down when outliers are not identifiable in isolation.

Our main conceptual contribution is an approach to utilize information about outliers beyond what is available in the top eigenvector of the empirical covariance $\bar{\Sigma}$ and in individual ℓ_2 norms. Appropriately adapted to their respective settings, this leads to our algorithms for both robust mean estimation and outlier detection.

Our first observation is that *any eigenvalue/eigenvector λ, v of the empirical covariance with $\lambda \gg 1$ must be due to outliers.* We therefore consider the intermediate goal of finding a *distribution* over directions $v \in \mathbb{R}^d$ containing information about as many outlier directions as possible. We formalize this as the following entropy-regularized convex program over $d \times d$ positive semidefinite matrices:

$$\max_{U \in \mathbb{R}^{d \times d}} \alpha \cdot \langle U, \bar{\Sigma} \rangle + S(U) \text{ such that } U \succeq 0, \text{tr}(U) = 1, \quad (1)$$

where $\alpha \geq 0$ is some constant and $\langle A, B \rangle = \text{tr}(AB^\top)$ denotes the trace inner product of matrices. Here, $S(U) = -\langle U, \log U \rangle$ is the *quantum entropy* (also known as the *von Neumann entropy*) of the matrix U . If $U = \sum_{i=1}^d \mu_i v_i v_i^\top$ is the eigendecomposition of U , since it has $\text{tr}(U) = 1$ we may interpret it as a distribution over orthonormal vectors v_1, \dots, v_d with weights μ_1, \dots, μ_d and hence with entropy $S(U)$. Under this interpretation, $\langle U, \bar{\Sigma} \rangle = \mathbb{E}_{v_i \sim \mu} \langle v_i, \bar{\Sigma} v_i \rangle$. As α varies, (1) trades off optimizing for a distribution supported on many distinct directions for a distribution supported on eigenvectors of $\bar{\Sigma}$ with large eigenvalues. The optimizer of (1) takes the form $U = \exp(\alpha \cdot \bar{\Sigma}) / \text{tr} \exp(\alpha \cdot \bar{\Sigma})$ where $\exp(\cdot)$ is the matrix exponential function.

Definition 1.1. Let $U = \exp(\alpha \cdot \bar{\Sigma}) / \text{tr} \exp(\alpha \cdot \bar{\Sigma})$ be the optimizer of (1), for some data set $\mathcal{X} = X_1, \dots, X_n \in \mathbb{R}^d$ where $\bar{\Sigma}$ is the covariance of \mathcal{X} . The quantum entropy (QUE) scores with parameter α are given by $\tau_i = (X_i - \bar{\mu})^\top U (X_i - \bar{\mu})$, where $\bar{\mu}$ is the mean of \mathcal{X} .

Intuitively, the QUE scores will penalize any point which is causing a large eigenvalue in any direction, which should allow us to find more outliers than the naive spectral scores presented above. QUE scores also interpolate between two more naive scoring rules: when $\alpha = 0$ we have $U = \text{Id}/d$ and so $\tau_i = \frac{1}{d} \|X_i - \bar{\mu}\|_2^2$ is the ℓ_2 norm (up to a scaling), while when $\alpha \rightarrow \infty$ we have $U \rightarrow vv^\top$ where v is the top eigenvector of $\bar{\Sigma}$, recovering naive spectral scoring. *In both experiments and theory we find that choosing α strictly between 0 and ∞ outperforms either of the extreme choices.*

QUE scores are also appealing from a computational perspective: we show that a list of approximate QUE scores $\tau'_i = (1 \pm 0.01)\tau_i$ can be computed from X_1, \dots, X_n in nearly-linear time, by appropriate use of Johnson-Lindenstrauss sketching and efficient computation of the matrix exponential by series expansion. This is crucial to both the nearly-linear running time of our algorithm for robust mean estimation and to the scalability of our outlier detection method.

In Section 2 we describe refinements of QUE scoring which fit it into the matrix multiplicative weights framework [6], leading to our nearly-linear time algorithms for robust mean estimation. We give two very similar algorithms, one for when the distribution of inliers is only assumed to have bounded covariance, and one when the inliers are assumed to be subgaussian. The resulting algorithms are conceptually similar to the following modification of the filter mentioned above: until $\|\bar{\Sigma}\|_2 \leq O(1)$, compute QUE scores, throw out data points X_i with $\tau_i \gg 1$, and repeat. (To obtain provable guarantees, our final algorithms are somewhat more complex: in some iterations we use QUE scores based on certain reweightings of the data learned in previous iterations.)

In Section 3 we describe experiments validating the QUE scoring rule on both synthetic and real data sets. We show that it performs especially well by comparison to local-neighborhood methods and to scoring based on only the top eigenvector in data sets where the inliers are close to isotropic (or can be made so by applying data whitening procedures) and in which there are heterogeneous outliers.

1.3 Related work

Robust mean estimation: The study of robust statistics and in particular robust mean estimation began with major works by Anscombe, Huber, Tukey and others in the 1960s [7, 8, 9, 10]. The literature on polynomial-time algorithms for robust statistics has exploded in recent years, following works by Diakonikolas et al and Lai, Rao and Vempala giving the first polynomial-time algorithms for robust mean estimation with dimension-independent (or nearly dimension-independent) error [1, 2]. A full survey is beyond our scope here – see e.g. the recent theses [11, 12] for a thorough account. Particularly relevant to our work is the recent work of Cheng, Diakonikolas, and Ge who design an algorithm for robust mean estimation with running time $\tilde{O}(nd/\epsilon^6)$ – the first to achieve nearly linear time for constant ϵ – by appeal to nearly linear time solvers for packing and covering semidefinite programs [3]. Our algorithms carry two advantages over this prior work: first, our algorithm runs in nearly linear time for any choice of $\epsilon = \epsilon(n, d)$, and second, because we avoid the $1/\epsilon^6$ scaling and appeal to semidefinite programming, our theoretical ideas lead to a practical method for outlier detection. The techniques of Diakonikolas et al. were later extended to robust covariance estimation [13]; it remains an interesting direction to extend our techniques to covariance estimation.

Outlier detection Detection of outliers goes back nearly to the beginning of statistics itself [14]. Even restricting to the high dimensional case it has a literature too broad to survey here. Much recent work has focused on so-called *local outlier factor*-based methods, which assign outlier scores based on the local density of other samples near each X_i – see e.g. [15, 16] and further references in [5]. We find that QUE scoring compares favorably to such local methods in high-dimensional datasets like we describe in Section 1.1 – see Section 3 and supplementary material for details.

2 Robust mean estimation: results and algorithm

We turn to our algorithm for robust mean estimation, deferring details to supplemental material.

Definition 2.1 (ϵ -corrupted samples). Let D be a distribution on \mathbb{R}^d . We say that X_1, \dots, X_n are an ϵ -corrupted set of samples from D if they are first drawn i.i.d. from D , then modified by an adversary who may adaptively inspect all the samples, remove ϵn of them, and replace them with arbitrary vectors in \mathbb{R}^d .

Note that ϵ -corruption is a *stronger* outlier model than the $(1 - \epsilon)D + \epsilon N$ mixture model we described in Section 1; our algorithms also work in this milder mixture model. Our main theoretical result is:

Theorem 2.1. *For every $n, d \in \mathbb{N}$ and $\epsilon > 0$ there are algorithms QUESCOREFILTER, S.G.-QUESCOREFILTER with running time $\tilde{O}(nd)$, such that for every distribution D on \mathbb{R}^d with mean μ and covariance Σ , given n ϵ -corrupted samples from D , QUESCOREFILTER produces $\hat{\mu}$ such that $\|\hat{\mu} - \mu\|_2 \leq O(\sqrt{\epsilon}) + \tilde{O}(\sqrt{d/n})$ if $\Sigma \preceq \text{Id}$, and S.G.-QUESCOREFILTER produces $\hat{\mu}$ such that $\|\hat{\mu} - \mu\|_2 \leq O(\epsilon\sqrt{\log(1/\epsilon)} + \sqrt{d/n})$ if D is sub-Gaussian with $\Sigma = \text{Id}$, all with probability at least 0.99.*

For the bounded covariance case, the $O(\sqrt{\epsilon})$ term is information-theoretically optimal up to constant factors. The other term, $\tilde{O}(\sqrt{d/n})$, is information-theoretically optimal up to the logarithmic factors in the $\tilde{O}(\cdot)$ even without corruptions. For the sub-Gaussian case, the $O(\epsilon\sqrt{\log(1/\epsilon)})$ term is believed to be necessary for computationally efficient algorithms (see e.g. the statistical-query lower bound [17]), although that term can be made $O(\epsilon)$ by using computationally-intractable estimators such as Tukey median, and the latter is information-theoretically optimal [10]. The $\sqrt{d/n}$ term is information-theoretically optimal even without corruptions.

In this section we discuss our algorithm for the bounded-covariance case $\Sigma \preceq \text{Id}$ in the setting that the adversary may not remove samples, leaving technical details and the modifications necessary to handle removed samples and sub-Gaussian D to supplementary material.

Definition 2.2 (Simplified robust mean estimation). Let $S = \{X_1, \dots, X_n\} \subseteq \mathbb{R}^d$ be a dataset with the property that S partitions into $S = S_g \cup S_b$ with $|S_b| \leq \epsilon n$ and $\mathbb{E}_{i \sim S_g} (X_i - \mu_g)(X_i - \mu_g)^\top \preceq \text{Id}$, where $\mu_g = \mathbb{E}_{i \sim S_g} X_i$. Given S , the goal is to find a vector $\hat{\mu}$ with $\|\mu_g - \hat{\mu}\|_2 \leq O(\sqrt{\epsilon})$.

Like prior algorithms for robust mean estimation, ours maintains a *weight vector* $w_1, \dots, w_n \geq 0$ with $\sum w_i \leq 1$, initialized to $w_i = 1/n$. The algorithm iteratively decreases the weight of points suspected to be outliers that are causing $\|\mu(w) - \mu_g\|_2$ to be large.¹ A key insight of recent work on robust mean estimation is that it suffices to find weights w which place almost as much mass on S_g as does the uniform weighting and whose empirical covariance is small. This is formalized in the following lemma. For a weight vector w , let $|w| = \sum w_i$, $\mu(w) = \frac{1}{|w|} \sum w_i X_i$, and $M(w) = \frac{1}{|w|} \sum w_i (X_i - \mu(w))(X_i - \mu(w))^\top$. Let $\|M\|_2$ be the spectral norm of a matrix M .

Lemma 2.2 (Implicit in prior work). *Let $S = \{X_1, \dots, X_n\}$ be as in Definition 2.2. Suppose that w is a weight vector such that $\|M(w)\|_2 \leq O(1)$ and w is mostly good, by which we mean $|\frac{1}{n} \mathbf{1}_{S_g} - w_g| \leq |\frac{1}{n} \mathbf{1}_{S_b} - w_b|$, where $\mathbf{1}_{S_g}, \mathbf{1}_{S_b}$ are the indicators of S_g, S_b and w_g, w_b are w restricted to S_g, S_b respectively. (Intuitively, w is mostly good if it results by removing from the uniform weighting $\mathbf{1}_S/n$ more weight from S_b than from S_g .) Then $\|\mu(w) - \mu_g\|_2 \leq O(\sqrt{\epsilon})$.*

Lemma 2.2 captures the following geometric intuition: if the bad points S_b receive enough weight in w to cause $\|\mu(w) - \mu_g\|_2 \gg \sqrt{\epsilon}$, then an $O(\epsilon)$ -fraction of the mass of w is on X_i which are unusually correlated with the vector $\mu(w) - \mu_g$, which leads to a large maximum eigenvalue in $M(w)$. Prior works employ a variety of methods to find a mostly good weight vector w with $\|M(w)\|_2 \leq O(1)$. Perhaps the simplest is the *filter* of [1], which iterates: While $\|M(w)\|_2 \gg 1$, compute its top eigenvector v and *naive spectral scores* $\tau_i = \langle X_i - \mu(w), v \rangle^2$. Throw out X_i with large τ_i and repeat.

The filter ensures that the weight vector it maintains is mostly good because (in an averaged sense) τ_i can be large only for X_i which are corrupted. This is because the (weighted) sum of all scores $\sum w_i \tau_i = \langle M(w), vv^\top \rangle \gg 1$, while the contribution to this sum from S_g has $\sum_{i \in S_g} w_i \tau_i \approx \langle \frac{1}{n} \sum_{i \in S_g} (X_i - \mu_g)(X_i - \mu_g)^\top, vv^\top \rangle \leq 1$. (Here we ignore some details about centering X_i at μ_g rather than $\mu(w)$.) Thus, the τ_i from S_b must make up almost all of $\sum w_i \tau_i$. Simple approaches to removing or downweighting X_i with large τ_i then remove strictly more weight from S_b than from S_g .

However, filtering based on naive spectral scores alone faces a barrier to achieving nearly-linear running-time. If the corruptions S_b are split among many orthogonal directions, the naive spectral filter will have to find those directions one at a time. Thus, it may require $\Omega(d)$ iterations (leading to $\Omega(nd^2)$ running time) to arrive at w with $\|M(w)\|_2 \leq O(1)$.

Our main idea is that by replacing naive spectral scores with slightly modified QUE scores, each iteration of the filter can take into account projections of each sample onto many large eigenvectors of $M(w)$. We show that our modified QUE scores τ_i maintain the property that $\sum_{i \in S_b} w_i \tau_i \gg \sum_{i \in S_g} w_i \tau_i$, and so downweighting according to τ_i removes more mass from S_b than S_g . However, filtering with QUE scores makes faster progress than with naive spectral scores: roughly speaking, we show that only $O(\log d)^2$ rounds of filtering according to QUE scores are required to find a mostly-good weight vector w with $\|M(w)\|_2 \leq O(1)$.

The core of our algorithm is a subroutine, `DECREASESPECTRALNORM`, to take a mostly good weight vector w with $\|M(w)\|_2 \gg 1$ and in $O(\log d)$ rounds of QUE filtering produce another mostly good w' with $\|M(w')\|_2 \leq \frac{3}{4} \|M(w)\|_2$. Repeating this subroutine $O(\log d)$ times and then outputting the resulting $\mu(w)$ yields our main algorithm. An outline of this subroutine is presented as Algorithm 1. We first establish a rigorous sense in which downweighting according to outlier scores τ_i makes progress: *it decreases the weighted average of the scores while removing more weight from bad points than good.*

¹Some prior algorithms, e.g. the *filter* of [1] instead iteratively throw out points suspected to be outliers. However, since those algorithms are (necessarily) randomized, they can also be viewed as weighting points, where the weight of X_i is the probability it has not been thrown out. The algorithm we present here can also be implemented by throwing out points in a randomized fashion – we discuss further in supplementary material.

Lemma 2.3 (Progress in one round of downweighting, informal). *There is a downweighting algorithm which takes a density matrix U and a mostly good weight vector w and produces a mostly good weight vector w' by downweighting points with large score $\tau_i = \langle X_i - \mu(w), U(X_i - \mu(w)) \rangle$ such that $\sum w'_i \tau_i \leq \frac{1}{3} \sum w_i \tau_i$ so long as $\sum w_i \tau_i \gg 1$. Furthermore, $M(w') \preceq M(w)$.*

Let us give a geometric interpretation to Lemma 2.3: it establishes that if $\sum w_i \tau_i = \langle U, M(w) \rangle \gg 1$ then the quadratic form of $M(w')$ decreases in the directions defined by U , since

$$\langle M(w'), U \rangle \approx \sum w'_i \tau_i \leq \frac{1}{3} \sum w_i \tau_i = \frac{1}{3} \langle M(w), U \rangle. \quad (2)$$

This guarantee becomes more meaningful as the entropy $S(U)$ increases, because it suggests the quadratic form of $M(w)$ has decreased in more directions. To make this formal, we appeal to the matrix multiplicative weights framework. DECREASESPECTRALNORM applies downweighting iteratively using a sequence of entropy-maximizing density matrices U_1, \dots, U_T chosen according to the matrix multiplicative weights update rule, leading to a series of mostly good weight vectors w_1, \dots, w_T such that $\|M(w_T)\|_2 \leq \frac{3}{4} \|M(w_0)\|_2$. We choose

$$U_t = \exp \left(\frac{1}{\|M(w)\|_2} \sum_{k=0}^{t-1} M(w_k) \right) / \text{tr} \exp \left(\frac{1}{\|M(w)\|_2} \sum_{k=0}^{t-1} M(w_k) \right), \quad (3)$$

where $w_0 = w$ is the input weight vector, $U_0 = \text{Id}$, and w_t results from applying the downweighting of Lemma 2.3 to w_{t-1} using U_t (if $\langle M(w_{t-1}), U_t \rangle \gg 1$). The following lemma is a special case of the standard (local norm) regret bound for matrix multiplicative weights.

Lemma 2.4 (Special case of Theorem 3.1, [18]). *For any w_0, \dots, w_T , if $\alpha \leq 1/\|M(w_t)\|_2$ for all $t \leq T$, then*

$$\left\| \sum_{t=0}^{T-1} M(w_t) \right\|_2 \leq \sum_{t=0}^{T-1} \langle U_t, M(w_t) \rangle + \alpha \sum_{t=0}^{T-1} \langle U_t, M(w_t) \rangle \cdot \|M(w_t)\|_2 + \frac{\log d}{\alpha}. \quad (4)$$

Now we sketch the analysis of DECREASESPECTRALNORM.

Claim 2.5 (Informal). *If $w = w_0$ is mostly good, with $\|M(w_0)\|_2 \geq 100$, then DECREASESPECTRALNORM produces mostly good w_T with $\|M(w_T)\|_2 \leq \frac{3}{4} \|M(w_0)\|_2$.*

Proof sketch. Since $M(w_t) \preceq M(w_{t+1})$ by Lemma 2.3, we have $\|M(w_t)\|_2 \leq \|M(w_0)\|_2$ for all t , and hence $\alpha = 1/\|M(w_0)\|_2 \leq 1/\|M(w_t)\|_2$ for all t , so w_0, \dots, w_T and U_0, \dots, U_{T-1} satisfy the hypotheses of Lemma 2.4. By our choice of α and $M(w_T) \preceq M(w_t)$ for all t , (4) implies

$$T \cdot \|M(w_T)\|_2 \leq \left\| \sum_{t=0}^{T-1} M(w_t) \right\|_2 \leq 2 \sum_{t=0}^{T-1} \langle U_t, M(w_t) \rangle + \|M(w_0)\|_2 \cdot \log d.$$

If $\langle U_t, M(w_{t-1}) \rangle \geq \|M(w_0)\|_2/3 \gg 1$, then DECREASESPECTRALNORM performs downweighting, and by Lemma 2.3 and (2) (which we establish rigorously in supplemental material), $\langle M(w_t), U_t \rangle \leq \frac{1}{3} \langle M(w_{t-1}), U_t \rangle \leq \frac{1}{3} \|M(w_0)\|_2$. Otherwise, by hypothesis $\langle M(w_t), U_t \rangle = \langle M(w_{t-1}), U_t \rangle \leq \|M(w_0)\|_2/3$. Using this bound and dividing by T , we obtain $\|M(w_T)\|_2 \leq (\frac{2}{3} + \frac{\log d}{T}) \|M(w_0)\|_2$. Choosing $T \geq 20 \log d$ completes the proof sketch. \square

Running time: Our overall algorithm only requires $\log(nd)^{O(1)}$ iterations of DECREASESPECTRALNORM, and the latter only requires $O(\log(d))$ iterations of downweighting, so we just have to implement downweighting in nearly-linear time. We show in supplemental material that this can be done by avoiding representing any of the matrices U_t explicitly in memory: instead, we maintain only low-rank sketches of them. This leads to some approximation error in computing the QUE scores, but we show that approximations to the QUE scores suffice for all arguments above.

For remaining technical details and full proofs, see Sections 5-9 of supplemental materials.

Algorithm 1 DECREASESPECTRALNORM

- 1: **Input:** X_1, \dots, X_n as in Definition 2.2, mostly good weight vector w_0 .
 - 2: For iteration $t = 0, \dots, O(\log d)$, If $\|M(w_t)\|_2 \leq \frac{3}{4}\|M(w_0)\|_2$, output w_t and halt. Otherwise, let U_t as in (4). If $\langle U_t, M(w_{t-1}) \rangle \leq \frac{1}{3}\|M(w_0)\|_2$, let $w_{t+1} = w_t$. Else let w_{t+1} be the output of downweighting from Lemma 2.3 with U_t .
 - 3: Output w_T .
-

3 Outlier detection: algorithm and experimental results

In this section, we empirically evaluate outlier detection using QUE scoring. We must work with data containing well-defined and known inliers and outliers so that we can compare our results to ground-truth. We generate such data sets in three distinct ways, leading to three main experiments. (In supplemental material we also study some outlier-detection data sets appearing in prior work [5].)

Synthetic: We create synthetic data sets in 128 dimensions and $10^3 - 10^4$ samples with an ϵ -fraction of inhomogeneous outliers in k directions by sampling from a mixture of $k + 1$ Gaussians $(1 - \epsilon)\mathcal{N}(0, \text{Id}) + \sum_{i=1}^k \epsilon_i(\frac{1}{2}\mathcal{N}(C\sqrt{k/\epsilon} \cdot e_i, \sigma^2 \text{Id}) + \frac{1}{2}\mathcal{N}(-C\sqrt{k/\epsilon} \cdot e_i, \sigma^2 \text{Id}))$, where e_1, \dots, e_k are standard basis vectors, with $C \approx 1$ and $\sigma \ll 1$. The outliers are the samples from $\mathcal{N}(\pm C\sqrt{k/\epsilon} e_i, \sigma^2 \text{Id})$. By varying ϵ, k and the distribution $\epsilon_1, \dots, \epsilon_k$ of outlier weights, we demonstrate in this simplified model how max-entropy outlier scoring improves on baseline algorithms in the presence of inhomogeneous outliers. We choose the scaling $\sqrt{k/\epsilon} \cdot e_i$ because then standard calculations predict that if $\epsilon_i \approx \epsilon/k$ the outliers from $\mathcal{N}(\pm C\sqrt{k/\epsilon} e_i, \sigma^2 \text{Id})$ will contribute an eigenvalue greater than 1 to the overall empirical covariance.

Mixed – word embeddings: We create a data set consisting of word embeddings drawn from several sources. Inliers are the 100-dimensional GloVe embeddings ([19]) of the words in a random $\approx 10^3$ word long section of a novel (we use *Sherlock Holmes*) and outliers are embeddings of the first paragraphs of k featured Wikipedia articles from May 2019 [20].

Perturbed – images: We create a data set consisting of CIFAR10 images some of which have artificially-introduced *dead pixels*. Inliers are ≈ 4500 random CIFAR images $X \in \{1, \dots, 256\}^{1024}$. Outliers are ≈ 500 random CIFAR images, partitioned into groups S_1, \dots, S_k , such that for each group i a random coordinate $p_i \in \{1, \dots, 1024\}$ and a random value $c_i \in \{1, \dots, 256\}$ is chosen and for each $X \in S_i$ we set $X_{p_i} = c_i$.

Metric: All the methods we evaluate produce a vector of *scores* $\tau_1, \dots, \tau_n \in \mathbb{R}$. We use the standard *ROCAUC* metric to compare these scores to a ground-truth partition $S = S_g \cup S_b$ into inlier and outlier sets. $\text{ROCAUC}(\tau_1, \dots, \tau_n, S_b, S_g) = \Pr_{i \sim S_b, j \sim S_g}(\tau_i \geq \tau_j)$ is simply the probability that a randomly chosen outlier is scored higher than a random inlier.

Baselines: We compare QUE scoring to the following other scoring rules. ℓ_2 : $\tau_i = \|X_i - \bar{\mu}\|$ is the distance of X_i to the empirical mean; *top eigenvector naive spectral*: $\tau_i = \langle X_i - \bar{\mu}, v \rangle^2$ where v is the top eigenvector of the empirical covariance; *k-nearest neighbors (k-NN)* [21, 5] and *local outlier factor (LOF)* [22, 5] methods: τ_i is a function of the distances to its k nearest neighbors; *isolation forest and elliptic envelope*: standard outlier detection methods as implemented in scikit-learn [23, 24, 25].

Whitening: Scoring methods based on the projection of data points X_i onto large eigenvectors of the empirical covariance works best when those eigenvectors correspond to directions in which many outliers lie. In particular, if Σ_g , the covariance of S_g , itself has large eigenvalues then such spectral methods perform poorly. We assume access to a *whitening transformation* $W \in \mathbb{R}^{d \times d}$, which captures a small amount of prior knowledge about the distribution of inliers S_g . For best performance W should approximate $W^* = (\Sigma_g)^{-1/2}$ since W^*X_i form an isotropic set of vectors. Of course, to compute W^* exactly would require knowing which points are inliers, but we find that relatively naive approximations suffice. In particular, if a clean dataset Y_1, \dots, Y_m whose distribution is similar to the distribution of inliers is available, its empirical covariance can be used to find a good whitening transformation W . In our synthetic data we use $W = \text{Id}$. In our word embeddings experiment, we obtain W using the empirical covariance of the embedding of another random section of *Sherlock Holmes*. In our CIFAR-10 experiment, we obtain W from the empirical covariance of a fresh sample of ≈ 5000 randomly chosen images from CIFAR-10.

Algorithm 2 QUE-Scoring for Outlier Detection

- 1: **Input:** dataset $X_1, \dots, X_n \in \mathbb{R}^d$, optional whitening transformation $W \in \mathbb{R}^{d \times d}$, scalar $\alpha > 0$.
- 2: Let $X'_i = WX_i$ be whitened data, $\bar{\mu} = \frac{1}{n} \sum_{i=1}^n X'_i$ and $\bar{\Sigma} = \frac{1}{n} \sum_{i=1}^n (X'_i - \bar{\mu})(X'_i - \bar{\mu})^\top$.
- 3: For $i \leq n$, let $\tau_i = (X'_i{}^\top \exp(\alpha \bar{\Sigma} / \|\bar{\Sigma}\|_2) X'_i) / \text{Tr} \exp(\alpha \bar{\Sigma} / \|\bar{\Sigma}\|_2)$. Return τ_1, \dots, τ_n .

Note on α : in both synthetic and real data we find that $\alpha = 4$ is a good rule-of-thumb choice, consistently resulting in improved scores over baseline methods.

High-dimensional scaling: Implementing Algorithm 2 by explicitly forming the matrix $\bar{\Sigma}$ and performing a singular value decomposition (SVD) to compute $\exp(\alpha \bar{\Sigma})$ is feasible on relatively low-dimensional data ($d \approx 100$). See supplementary material for discussion and results of a nearly-linear time implementation.

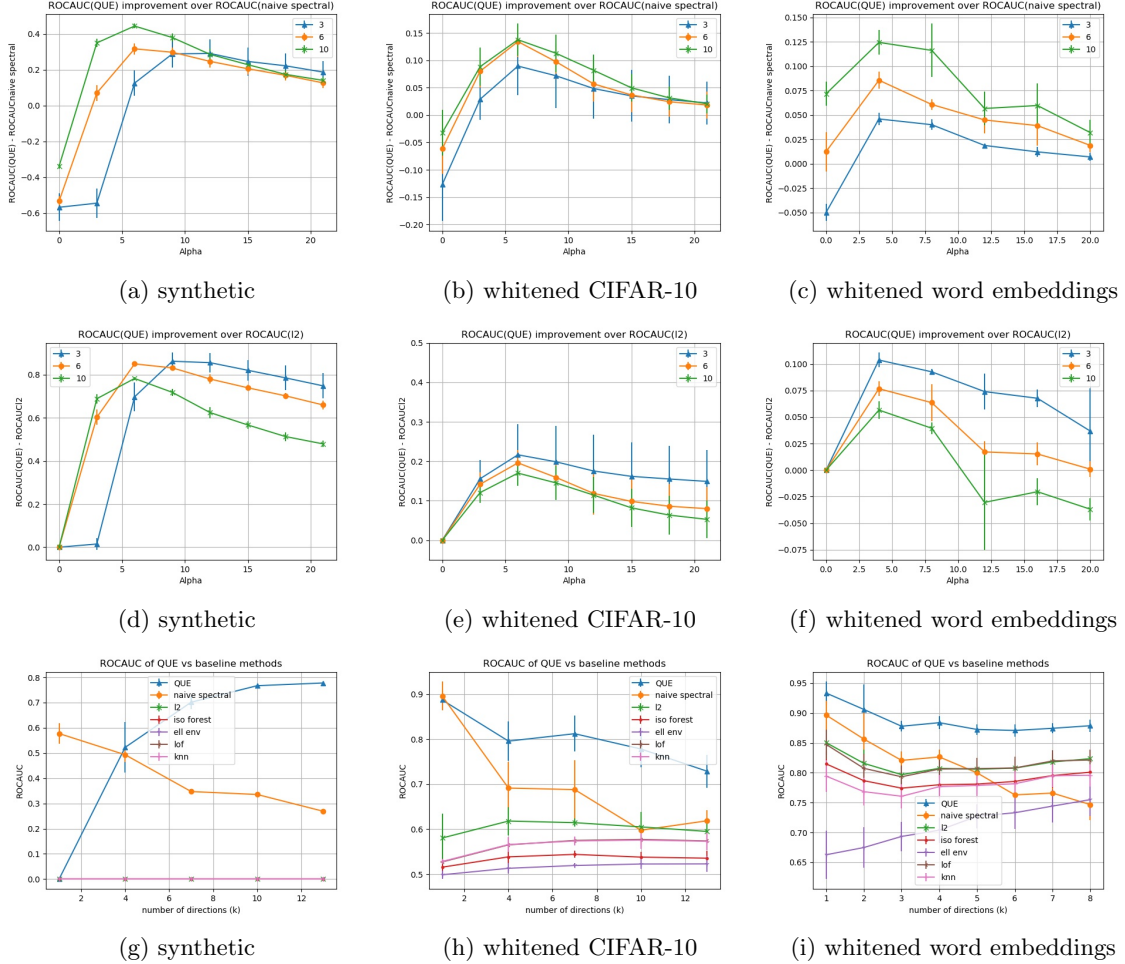


Figure 1: **(a-f)**: We plot the difference between ROCAUC performance of QUE and naive spectral (a-c), ℓ_2 scoring (d-f) on all three data sets, as α varies. Error bars represent one empirical standard deviation in 20 trials. Note that in all three cases the mean improvement in ROCAUC score given by QUE is at least one standard deviation above 0 for a wide range of α . Observe also that in synthetic data (which most closely parallels theory) the optimal α decreases with increasing number of outlier directions, in accord with the need to find a higher-entropy solution to (1). **(g-i)** We plot ROCAUC scores of QUE (with $\alpha = 4$) and a variety of other methods as the number of outlier directions increases. Error bars represent one standard deviation over 3 – 4 trials. Number of trials is small due to large running time requirements of Scikit-learn methods IsolationForest and EllipticEnvelope. The methods "lof" and "knn" are based on nearest-neighbor distances [5]. All except spectral methods perform poorly on synthetic data; as k increases the performance gap between QUE and naive spectral scoring grows. In all plots $\epsilon = 0.2$. Experiments were generated on a quad-core 2.6Ghz machine with 16GB RAM and an NVIDIA P100 GPU.

Roadmap of supplementary material

Here we provide a quick guide to the rest of the supplementary material. In Part I (Sections 4—9), we give omitted details to the description, and proof of correctness, of our nearly-linear time robust mean estimation algorithm. In Part II (Sections 10 and 11), we give additional experimental results and omitted details for the empirical evaluation of our outlier detection method based on QUE scoring.

Part I: Nearly-linear time robust mean estimation

4 Preliminaries

For two functions f, g , we say $f = \tilde{O}(g)$ if $f = O(g \log^c g)$ for some universal constant $c > 0$. We similarly define $\tilde{\Omega}$ and $\tilde{\Theta}$. For vectors $v \in \mathbb{R}^d$, we let $\|\cdot\|_2$ denote the usual ℓ_2 norm, and $\langle \cdot, \cdot \rangle$ denote the usual inner product between vectors. Let $\mathbf{1}_m \in \mathbb{R}^m$ denote the m -dimensional all-ones vector.

For matrices $A, M \in \mathbb{R}^{d \times d}$ we let $\|M\|_2$ denote its spectral norm, we let $\|M\|_F$ denote its Frobenius norm, and we let $\langle M, A \rangle = \text{tr}(M^\top A)$ denote the trace inner product between matrices. For any symmetric matrix $A \in \mathbb{R}^{d \times d}$, let $\exp(A)$ denote the usual matrix exponential of A . Finally, for scalars $x, y \in \mathbb{R}$, and any $\alpha > 0$, we say that $x \approx_\alpha y$ if $\frac{1}{1+\alpha}x \leq y \leq (1+\alpha)x$.

We say a distribution D over \mathbb{R}^d is isotropic if $\text{Cov}_{X \sim D}[X] = \text{Id}$. We say a univariate distribution D with mean μ is sub-gaussian with variance proxy s^2 if

$$\mathbb{E}_{X \sim D} \left[(X - \mu)^k \right] \leq \mathbb{E}_{X \sim \mathcal{N}(0, s^2)} [X^k]$$

for all k even. We say a distribution D over \mathbb{R}^d and mean μ is sub-gaussian with variance proxy $\Sigma \preceq I$, if for all unit vectors v , the distribution of $\langle v, X - \mu \rangle$ is sub-Gaussian with variance proxy $v^\top \Sigma v$. Intuitively, a sub-Gaussian distribution is simply any distribution which concentrates as well as a Gaussian.

4.1 Our results

With this terminology in place, we are now ready to state our main results on robust mean estimation. Our first result is for robust mean estimation under the assumption of bounded covariance:

Theorem 4.1. *Let D be a distribution with mean μ and covariance $\Sigma \preceq I$. Let $\varepsilon > 0$ be sufficiently small, and let $\delta > 0$. Let S be an ε -corrupted set of samples from D of size n . There is an algorithm $\text{QUESCOREFILTER}(S, \delta)$ which takes S and δ , and outputs $\hat{\mu}$ so that with probability $1 - \delta - \exp(-\varepsilon n)$, we have*

$$\|\mu - \hat{\mu}\|_2 \leq O \left(\sqrt{\varepsilon} + \sqrt{\frac{d}{n\delta}} + \sqrt{\frac{d(\log d + \log 1/\delta)}{n}} \right).$$

Moreover, the algorithm runs in time $\tilde{O}(nd \log 1/\delta)$.

We make two observations about this problem. First, it is well-understood (see e.g. [12, 11]) that $\Omega(\sqrt{\varepsilon})$ is unavoidable for this problem, no matter how many samples are given. Second, observe that the rate $O(\sqrt{d/n})$ is necessary for this problem even without corruptions. Thus, up to log factors, and the dependence on δ , this guarantee is information-theoretically optimal.

We also prove a strong statement for the case of robust mean estimation for sub-gaussian distributions:

Theorem 4.2. *Let D be an isotropic sub-gaussian distribution with variance proxy I and mean μ . Let $\varepsilon > 0$ be sufficiently small, and let $\delta > 0$. Let S be an ε -corrupted set of samples from D of size n . There is*

an algorithm $\text{S.G.-QUESCOREFILTER}(S, \delta, \varepsilon)$ which takes S, δ , and ε , and outputs $\hat{\mu}$ so that with probability $1 - \delta$, we have

$$\|\mu - \hat{\mu}\|_2 \leq O\left(\varepsilon\sqrt{\log 1/\varepsilon} + \sqrt{\frac{d + \log 1/\delta}{n}}\right).$$

Moreover, the algorithm runs in time $\tilde{O}(nd \log 1/\delta)$.

It is suspected, based on statistical-query lower bounds, that $\Omega(\varepsilon\sqrt{\log 1/\varepsilon})$ error is incurred by any computationally-efficient algorithm in this setting, although $\Theta(\varepsilon)$ is the minimax optimal dependence of the error rate on ε [17, 10]. Moreover, $\Omega\left(\sqrt{(d + \log 1/\delta)/n}\right)$ is the minimax rate for mean estimation for Gaussians without noise. Thus, the error guarantees of this algorithm are minimax optimal up to constants and the factor $\sqrt{\log(1/\varepsilon)}$.

This algorithm assumes that the distribution is isotropic. There is evidence that such an assumption is necessary to get error beyond $\sqrt{\varepsilon}$ using computationally efficient (i.e. poly-time) algorithms [26]. In our theorem statement above we also assume that the variance proxy is at most I . This is done for simplicity: it is easily verifiable that our algorithm works (with an appropriate scaling in front of the error guarantee) if the variance proxy is PSD upper bounded by $\sigma^2 I$ for any σ^2 .

5 Additional technical preliminaries

Before we describe our techniques, we require a few additional algorithmic tools, which we describe here.

5.1 Soft selection of subsets of points

In our presentation of our filtering algorithm for robust mean estimation, it will be convenient for us to work with a “soft” version of the filter. Instead of wholly removing points that we deem suspicious, we will maintain a set of weights for each point, and downweight those that we find suspicious. In this section, we establish notation for dealing with such operations. However, we briefly remark that, as we will explain later in Appendix A.3, the same results (up to log factors) can be established using “hard” filtering more akin to the algorithms presented in prior work, e.g. [4].

Throughout this paper, we will let Δ_n denote the simplex in n dimensions, and we let

$$\Gamma_n = \{w \in \mathbb{R}^n : w_i \geq 0, \sum w_i \leq 1\}.$$

For $w \in \Gamma_n$, let $|w| = \sum w_i$ be its ℓ_1 norm. For $w, w' \in \Gamma_n$, we say $w' \leq w$ if $w'_i \leq w_i$ for all $i = 1, \dots, n$. Let $S = \{X_1, \dots, X_n\}$ be a (multi)-set of n points. For any $w \in \Gamma_n$, we let $\mu(w) = \mu(S, w) = \frac{1}{|w|} \sum_{i=1}^n w_i X_i$, and we let

$$M(w) = M(S, w) = \sum_{i=1}^n w_i (X_i - \mu(w))(X_i - \mu(w))^\top.$$

Typically when the set S is understood, we will omit the dependence on S in the notation. For any set $T \subseteq S$, we let $\mu(T) = \mu(w)$ where $w \in \Gamma_n$ is the vector $w_i = 1/n$ for $i \in T$ and $w_i = 0$ otherwise, and similarly we let $M(T) = M(w)$.

5.2 Naive pruning

One primitive we will require will be the ability to remove points which are “obviously” outliers. It is well-known that there exist randomized nearly-linear time algorithms for achieving this. For completeness we prove this lemma in Appendix A.

Lemma 5.1 (folklore). *Let $\varepsilon > 1/2$, and let $\delta > 0$. Let $S \subset \mathbb{R}^d$ be a set of n points so that there exists a ball B of radius r and a subset $S' \subseteq S$ so that $|S'| \geq (1 - \varepsilon)n$, and $S' \subset B$. Then there is an algorithm $\text{NAIVEPRUNE}(S, r, \delta)$ which runs in time $O(nd \log 1/\delta)$ and with probability $1 - \delta$ outputs a set of points $T \subseteq S$ so that $S' \subseteq T$, and T is contained in a ball of radius $4r$.*

In the case where the output of NAIVEPRUNE satisfies the conditions of the lemma, we say that NAIVEPRUNE *succeeds*.

5.3 The one-dimensional filter

An important algorithmic primitive for us will be an univariate soft outlier removal step. The sub-problem considered here is as follows: we are given a set of nonnegative scores τ_1, \dots, τ_m , with the guarantee that there is a small subset $S \subseteq [m]$ so that $\sum_{i \in S} \tau_i > \frac{1}{2} \sum_{i=1}^m \tau_i$, that is, they contribute a majority of the mass of the points. The goal is to then either downweight (or remove) the overall set of scores in such a way so that more mass from S is removed than from outside of S , or alternatively, more points are removed from S than from outside S . An algorithm for achieving this via downweighting has already been described in [12], and a randomized algorithm that achieves the same sorts of guarantees with high probability by removing points is implicit in the filtering algorithm of [1, 4] (e.g. in Algorithm 3 in Appendix A of [4]). In this paper, we will require a slight strengthening of these algorithms. We require that not only do we remove more weight from the bad points than the good points, but we also decrease the overall sum by a constant factor. We observe that while we will present a method for achieving this via downweighting, one can achieve the same guarantee (with high probability) by removing points. In the main text, we choose to present the soft downweighting method for robust mean estimation for simplicity. See Appendix A.3 for details.

Formally, we describe an algorithm 1DFILTER and prove the following guarantee for the algorithm. The algorithm and its analysis are fairly straightforward so we defer the formal descriptions and proofs to Appendix A.2.

Theorem 5.2. *Let $\eta \in (0, 1/2)$, let $b \geq 2\eta$, and let w_1, \dots, w_m and τ_1, \dots, τ_m be non-negative numbers so that $\sum_{i=1}^m w_i \leq 1$. Let $\tau_{\max} = \max_{i \in [m]} \tau_i$. Suppose there exist two disjoint sets S_g, S_b so that $S_g \cup S_b = [m]$, and moreover,*

$$\sum_{i \in S_g} w_i \tau_i \leq \eta \sigma, \text{ where } \sigma = \sum_{i=1}^m w_i \tau_i.$$

Then $\text{1DFILTER}(w, \tau, b)$ runs in time $O((1 + \log \frac{\tau_{\max}}{b\sigma})m)$ and outputs $0 \leq w' \leq w$ so that:

- *more weight is removed from S_b than S_g , i.e. $\sum_{i \in S_g} w_i - w'_i \leq \sum_{i \in S_b} w_i - w'_i$, and*
- *the weighted sum of the τ has decreased, i.e. w' satisfies*

$$\sum_{i=1}^m w'_i \tau_i \leq b\sigma. \tag{5}$$

In particular, note that if $b = \Omega(1)$ and $\tau_{\max}/\sigma \leq (nd)^{O(1)}$, then this algorithm runs in nearly linear time.

Randomized outlier removal As mentioned previously, there is also a randomized strategy that avoids downweighting and achieves the same guarantee with high probability (up to logarithmic factors in runtime). Our overall robust mean estimation algorithm (for both settings presented in the paper) can be instantiated using this algorithm rather than 1DFILTER . While as far as we know this yields no theoretical improvements for robust mean estimation (indeed, our analysis of it proves bound which are worse by logarithmic factors than our analysis of soft downweighting), it is much closer to the practical outlier detection method used in the experiments in Section 3 and also to prior algorithms presented in [4], and may be of instructive value. For this reason we describe this algorithm in Appendix A.3.

5.4 Matrix Multiplicative weights

We will use the following form of the MMW update, which is essentially the same as presented in [18]. In each iteration $t = 0, \dots, T$, the player chooses an action $U_k \in \Delta_{d \times d}$, receives a gain matrix $F_t \in \mathbb{R}^{d \times d}$, and receives reward $\langle F_k, U_k \rangle$. Then the player sees F_k . In [18], they demonstrate that if the player plays according to the entropy regularizer (or equivalently, matrix multiplicative weights), namely,

$$U_k = \exp \left(cI + \alpha \sum_{t=0}^{k-1} F_t \right), \quad (6)$$

where c is a constant ensuring that $\text{tr}(X_k) = 1$, and α satisfies $\alpha F_t \preceq I$ for all $0 = 1, \dots, T-1$, then we have that for any $U \in \Delta_{n \times n}$,

$$\sum_{t=0}^{T-1} \langle F_t, U - U_t \rangle \leq \alpha \sum_{t=0}^{T-1} \langle U_t, |F_t| \rangle \cdot \|F_t\|_2 + \frac{\log n}{\alpha}. \quad (7)$$

Here, for any symmetric matrix $A = \sum_{i=1}^d \lambda_i v_i v_i^\top$, we let $|A|$ denote $|A| = \sum_{i=1}^d |\lambda_i| v_i v_i^\top$. Equivalently, by rearranging terms, and taking a supremum over U of (8), we obtain that the update satisfies

$$\left\| \sum_{t=0}^{T-1} F_t \right\|_2 \leq \sum_{t=0}^{T-1} \langle U_t, F_t \rangle + \alpha \sum_{t=0}^{T-1} \langle U_t, |F_t| \rangle \cdot \|F_t\|_2 + \frac{\log n}{\alpha}. \quad (8)$$

6 An MMW algorithm for robust mean estimation with bounded covariance

In this section we describe the algorithm which achieves Theorem 4.1. We first identify a deterministic condition on the set of inliers under which our algorithm is guaranteed to be correct. It is a very mild condition: at a high level, it simply states that the empirical mean of the samples is converging to the true mean, and the empirical covariance is bounded.

Definition 6.1. We say a set of points S is (γ_1, γ_2) -good with respect to a distribution D with mean μ and covariance $\Sigma \preceq \text{Id}$ if the following two properties hold:

- $\|\mu(S) - \mu\|_2 \leq \gamma_1$, and
- $\|\text{Cov}(S)\|_2 \leq \gamma_2$.

The following is a generalization of Lemma A.18 in [4], which states that, with high probability, any set of i.i.d. points from a distribution with bounded covariance will contain a large set which is good with respect to that distribution. For completeness we prove this lemma in Appendix B.

Lemma 6.1. Let $\varepsilon \in [0, 1/2)$, and let n be a positive integer. Let D be a distribution with mean μ and covariance $\Sigma \preceq \text{Id}$, and let X_1, \dots, X_n be independent draws from D . Then, with probability $1 - \delta - \exp(-\varepsilon n)$, there exists a set $S \subseteq \{X_1, \dots, X_n\}$ so that the following two conditions are simultaneously satisfied:

- $|S| \geq (1 - \varepsilon)n$, and
- S is (γ_1, γ_2) -good with respect to D , where

$$\gamma_1 = \frac{1}{1 - \varepsilon} \cdot \left(\sqrt{\frac{2d}{n\delta}} + \sqrt{c\varepsilon} \right), \text{ and } \gamma_2 = \frac{1}{1 - \varepsilon} \cdot \frac{d(\log d + \log 2/\delta)}{c'\varepsilon n}, \quad (9)$$

for some universal constants $c, c' > 0$.

In particular, observe that for constant δ and for $n = \Omega(d \log d / \varepsilon)$, we have that $\gamma_1 = O(\varepsilon)$ and $\gamma_2 = O(1)$.

6.1 The good set

Throughout we will let $S = S_g \cup S_b \setminus S_r$, where S_g is (γ_1, γ_2) -good with respect to D , and $|S_b|, |S_r| \leq \varepsilon|S|$. For any $w \in \Gamma_n$, we let $w_g \in \Gamma_{|S_g|}$ denote the restriction of w to the indices in S_g , and similarly define $w_b \in \Gamma_{|S_b|}$.

Our set of weights of interest will be slightly different than those considered in prior papers, but morally captures the same concept, up to issues of reweighting. We will always guarantee that the weights we consider lie within the following set:

$$\mathfrak{S}_{n,\varepsilon} = \left\{ w \in \Gamma_n : w \leq \frac{1}{n} \mathbf{1}_n \text{ and } \left| \frac{1}{n} \mathbf{1}_{|S_g|} - w_g \right| \leq \left| \frac{1}{n} \mathbf{1}_{|S_b|} - w_b \right| \right\}.$$

Intuitively, weights in the set $\mathfrak{S}_{n,\varepsilon}$ are what happens when we remove points from our data set, but ensure that we always remove at least as much mass from the bad set as we do from the good set.

6.2 Geometric lemmata

We first prove the following sequence of structural lemmata. The first, which is implicit in the earlier work, and which is in some sense the fundamental geometric fact which guides our algorithmic design, gives an upper bound on the deviation between the weighted empirical mean of the data set and the true mean of the distribution in terms of the spectral norm of the weighted covariance of the dataset.

Lemma 6.2. *Let S, γ_1, γ_2 be as above, and let $w \in \mathfrak{S}_{n,\varepsilon}$. Then*

$$\|\mu(w) - \mu\|_2 \leq \frac{1}{1 - 2\varepsilon} \cdot \left(\sqrt{\varepsilon\gamma_2} + (1 + \varepsilon)\gamma_1 + 2\sqrt{\varepsilon \|\Sigma(w)\|_2} \right).$$

Proof. Let $\rho = \mu(w) - \mu$. We have the following sequence of identities:

$$\begin{aligned} \|\mu(w) - \mu\|_2^2 &= \langle \mu(w) - \mu, \mu(w) - \mu \rangle \\ &= \frac{1}{|w|} \sum_{i=1}^n w_i \langle X_i - \mu, \rho \rangle \\ &= \frac{1}{|w|} \left(\sum_{i \in S_g} w_i \langle X_i - \mu, \rho \rangle + \sum_{i \in S_b} w_i \langle X_i - \mu, \rho \rangle \right) \\ &= \frac{1}{|w|} \left(\underbrace{\sum_{i \in S_g} \frac{1}{n} \langle X_i - \mu, \rho \rangle}_{W_1} - \underbrace{\sum_{i \in S_g} \left(\frac{1}{n} - w_i \right) \langle X_i - \mu, \rho \rangle}_{W_2} + \underbrace{\sum_{i \in S_b} w_i \langle X_i - \mu, \rho \rangle}_{W_3} \right). \end{aligned}$$

We now upper bound each term separately. By Cauchy-Schwarz, we have

$$\begin{aligned} |W_1| &\leq (1 - \varepsilon) \left\| \frac{1}{(1 - \varepsilon)n} \sum_{i \in S_g} X_i - \mu \right\|_2 \|\rho\|_2 \\ &\leq \gamma_1 \|\rho\|_2, \end{aligned}$$

by the goodness of S_g .

We now turn our attention to W_2 and W_3 . Both bounds will follow from the following claim:

Claim 6.3. *Let $w', \alpha \in \Gamma_n$ be so that $|w'| \leq \varepsilon$, $w'_i \leq \frac{1}{n}$ for all $i \in [n]$, and $w' \leq \alpha$. Then, for any $v \in \mathbb{R}^d$, we have*

$$\left| \sum_{i=1}^n w'_i \langle X_i - \mu, v \rangle \right| \leq \sqrt{\varepsilon \|\Sigma(\alpha)\|_2} \|v\|_2 + \varepsilon \|\mu(\alpha) - \mu\|_2 \|v\|_2.$$

Proof. We have

$$\begin{aligned}
\left| \sum_{i=1}^n w'_i \langle X_i - \mu, v \rangle \right| &= \left| \sum_{i=1}^n w'_i \langle X_i - \mu(\alpha), v \rangle + |w'| \langle \mu(\alpha) - \mu, v \rangle \right| \\
&\leq \left| \sum_{i=1}^n w'_i \langle X_i - \mu(\alpha), v \rangle \right| + \varepsilon |\langle \mu(\alpha) - \mu, v \rangle| \\
&\leq \left| \sum_{i=1}^n w'_i \langle X_i - \mu(\alpha), v \rangle \right| + \varepsilon \|\mu(\alpha) - \mu\|_2 \|v\|_2 .
\end{aligned}$$

By Hölder's inequality, we have

$$\begin{aligned}
\left(\sum_{i=1}^n w'_i \langle X_i - \mu(\alpha), v \rangle \right)^2 &\leq \left(\sum_{i=1}^n \frac{(w'_i)^2}{\alpha_i} \right) \cdot \left(\sum_{i=1}^n \alpha_i \langle X_i - \mu(\alpha), v \rangle^2 \right) \\
&\stackrel{(a)}{\leq} |w'| \cdot \sum_{i=1}^n \alpha_i \langle X_i - \mu(\alpha), v \rangle^2 \\
&\stackrel{(b)}{\leq} \varepsilon \|\Sigma(\alpha)\|_2 \|v\|_2^2 .
\end{aligned}$$

Here (a) follows since $w'_i \leq \alpha_i$, and (b) follows from the definition of spectral norm, and the assumption on $|w'|$. Thus, by taking square roots and combining terms, we have

$$\left| \sum_{i=1}^n w'_i \langle X_i - \mu, v \rangle \right| \leq \sqrt{\varepsilon \|\Sigma(\alpha)\|_2} \|v\|_2 + \varepsilon \|\mu(\alpha) - \mu\|_2 \|v\|_2 ,$$

as claimed. \square

With this claim, we can now bound W_2 and W_3 . To bound W_2 , let $w'_i = \frac{1}{n} - w_i$ for $i \in S_g$ and $w_i = 0$ otherwise, and let $\alpha_i = \frac{1}{n}$ if $i \in S_g$, and 0 otherwise. Then, applying the claim with $v = \rho$ yields that

$$|W_2| \leq \sqrt{\varepsilon \|\Sigma(\alpha)\|_2} \|\rho\|_2 + \varepsilon \|\mu(S_g) - \mu\|_2 \|\rho\|_2 \leq \sqrt{\varepsilon \gamma_2} \cdot \|\rho\|_2 + \varepsilon \gamma_1 \|\rho\|_2 .$$

Similarly, to bound W_3 , let $w'_i = w_i$ if $i \in S_b$ and $w'_i = 0$ otherwise, and let $\alpha_i = w_i$ for all $i \in S$. Again, letting $v = \rho$, we get that

$$|W_3| \leq \sqrt{\varepsilon \|\Sigma(w)\|_2} \|\rho\|_2 + \varepsilon \|\rho\|_2^2 ,$$

as well. Combining these three bounds, and using the fact that $|w| \leq 1$, yields that

$$\|\rho\|_2^2 \leq \sqrt{\varepsilon \gamma_2} \|\rho\|_2 + (1 + \varepsilon) \gamma_1 \|\rho\|_2 + 2\sqrt{\varepsilon \|\Sigma(w)\|_2} \|\rho\|_2 + 2\varepsilon \|\rho\|_2^2 .$$

Simplifying this expression then yields the desired bound on $\|\rho\|_2$. \square

We also require the following linear algebraic fact:

Lemma 6.4. *Let $w', w \in \Gamma_n$ so that $w' \leq w$. Then $\Sigma(w') \preceq \Sigma(w)$.*

Proof. We first observe that if $w' \leq w$, then

$$\sum_{i=1}^n w'_i (X_i - \mu(w)) (X_i - \mu(w))^\top \preceq \sum_{i=1}^n w_i (X_i - \mu(w)) (X_i - \mu(w))^\top = \Sigma(w) .$$

To complete the argument, observe that

$$\begin{aligned} \sum_{i=1}^n w'_i (X_i - \mu(w)) (X_i - \mu(w))^\top &= \sum_{i=1}^n w'_i (X_i - \mu(w')) (X_i - \mu(w'))^\top + |w'| (\mu(w') - \mu(w)) (\mu(w') - \mu(w))^\top \\ &\succeq \sum_{i=1}^n w'_i (X_i - \mu(w')) (X_i - \mu(w'))^\top = \Sigma(w') , \end{aligned}$$

and so $\Sigma(w') \preceq \Sigma(w)$, as claimed. \square

6.3 General algorithm description, bounded second moment

In this section we describe the algorithm we would like to run via matrix multiplicative weights, and demonstrate that it will terminate in a small number of iterations, when given good approximations to the entropic scores.

The algorithm, which we call **QUESCOREFILTER**, proceeds in epochs, and takes as input a corrupted dataset S , and a *score oracle* \mathcal{O} . Initially, in epoch $s = 0$, we let $w^{(0)} = \frac{1}{n} \mathbf{1}_n$. Then, in epoch s , the algorithm proceeds iteratively as follows. First, approximately compute $\lambda^{(s)} \approx_{0.1} \|M(w^{(s)})\|_2$, and if $\lambda^{(s)} \leq 100\gamma_2$, then we terminate and output $\mu(w^{(s)})$.

Otherwise, we let $w_0^{(s)} = w^{(s)}$. Then, in iteration $t = 0, \dots, T_s$, we first approximately compute $\lambda_t^{(s)} \approx_{0.1} \|M(w_t^{(s)})\|_2$. If $\lambda_t^{(s)} \leq \frac{2}{3}\lambda_0^{(s)}$, we terminate the epoch and let $w^{(s+1)} = w_t^{(s)}$. Otherwise, we let $U_t^{(s)}$ be prescribed by the MMW update with parameter $\alpha^{(s)} = \frac{1}{1.1 \cdot \lambda_0^{(s)}}$, where the loss is given as follows. At time t , and for all $i \in [n]$, we let $\tilde{\tau}_{t,i}^{(s)} = \mathcal{O}(S, w_0^{(s)}, \dots, w_n^{(s)})$ be the set of scores that the oracle produces. We then compute $\sum_i \tilde{\tau}_{t,i}^{(s)}$. If $\sum_i \tilde{\tau}_{t,i}^{(s)} \leq \frac{1}{5}\lambda_0^{(s)}$, then let $w_{t+1}^{(s)} = w_t^{(s)}$. Otherwise, let $w_{t+1}^{(s)} = \text{1DFILTER}(w_t^{(s)}, \tilde{\tau}_t^{(s)}, 1/4)$. In either case, the algorithm receives the gain matrix $F_t = M(w_{t+1}^{(s)})$. The formal pseudocode for this algorithm is given in Algorithm 3.

The score oracles There are two important score oracles for our purposes. The first is the *exact* score oracle $\mathcal{O}_{\text{exact}}$, whose output is

$$\tau_{t,i}^{(s)} = \left(X_i - \mu(w_{t,i}^{(s)}) \right)^\top U_t^{(s)} \left(X_i - \mu(w_{t,i}^{(s)}) \right) , \quad (10)$$

where $U_t^{(s)}$ is as in Algorithm 3, namely,

$$U_t^{(s)} = \exp \left(c \text{Id} - \alpha \sum_{i=0}^{t-1} M(w_i^{(s)}) \right) , \quad (11)$$

where c is chosen so that $\text{tr}(U_t^{(s)}) = 1$.

Filtering using these scores corresponds to the intuition that we are using $U_t^{(s)}$ as a high entropy certificate. Ignoring runtime, this set of scores would be the most natural set of scores for the algorithm. However, computing this score is quite inefficient (polynomial but super-linear). Thus, we also require an *approximate* score oracle, which is an oracle $\mathcal{O}_{\text{approx}}$ whose output $\tilde{\tau}_{t,i}^{(s)}$ satisfies that

$$\tilde{\tau}_{t,i}^{(s)} \approx_{0.1} \tau_{t,i}^{(s)} \quad (12)$$

for all t, i, s , where $\tau_{t,i}^{(s)}$ is defined in (10) (the choice of 0.1 in the approximation here as well as for $\lambda_t^{(s)}$ is arbitrary; any constant sufficiently small will suffice). In Section 8 we will demonstrate how to construct such an approximate score oracle in nearly-linear time.

Algorithm 3 MMW-based filtering method for robust mean estimation with bounded second moments

```

1: Input: dataset  $S \subset \mathbb{R}^d$  of size  $n$ , parameters  $\gamma_1, \gamma_2$ , score oracle  $\mathcal{O}$ 
2: Let  $w^{(0)} = \frac{1}{n} \mathbf{1}_n$ .
3: for epoch  $s = 0, \dots, O(\log \kappa)$  do
4:   Let  $\lambda^{(s)} \approx_{0.1} \|M(w^{(s)})\|_2$ 
5:   if  $\lambda^{(s)} \leq 100\gamma_2$  then
6:     return  $\mu(w^{(s)})$ 
7:   end if
8:   Let  $w_0^{(s)} = w^{(s)}$ 
9:   Let  $\alpha^{(s)} = \frac{1}{1.1 \cdot \lambda_0^{(s)}}$ 
10:  for iteration  $t = 0, \dots, O(\log d)$  do
11:    Compute  $\lambda_t^{(s)} \approx_{0.1} \|M(w_t^{(s)})\|_2$ .
12:    if  $\lambda_t^{(s)} \leq \frac{2}{3} \lambda_0^{(s)}$  then
13:      terminate epoch
14:    end if
15:    Let  $U_t^{(s)}$  be given by MMW update with parameter  $\alpha^{(s)}$ 
16:    For  $i = 1, \dots, n$ , let  $\tilde{\tau}_{t,i}^{(s)} = \mathcal{O}(S, w_0^{(s)}, \dots, w_n^{(s)})$ 
17:    if  $\sum_i \tilde{\tau}_{t,i}^{(s)} \leq \frac{1}{5} \lambda_0^{(s)}$  then
18:      Let  $w_{t+1}^{(s)} = w_t^{(s)}$ .
19:    else
20:      Let  $w_{t+1}^{(s)} = \text{1DFILTER}(w_t^{(s)}, \tilde{\tau}_t^{(s)}, 1/4)$ .
21:    end if
22:    Let feedback matrix to MMW update be  $F_t = M(w_{t+1}^{(s)})$ 
23:  end for
24:  Let  $w^{(s+1)} = w_t^{(s)}$ 
25: end for

```

6.4 Correctness of QUEScoreFilter

The remainder of this section is dedicated to a proof of correctness of QUEScoreFilter when instantiated with $\mathcal{O}_{\text{approx}}$. Formally, we show:

Theorem 6.5. *Let D be a distribution with covariance $\Sigma \preceq I$ and mean μ . Let $\varepsilon < c$, where c is a universal constant, and let $\gamma_1, \gamma_2 > 0$. Let S be a dataset so that $S = S_g \cup S_b \setminus S_r$ so that S_g is (γ_1, γ_2) -good with respect to D , and $|S_b|, |S_r| \leq \varepsilon |S|$. Suppose moreover that $\|X_i\|_2 \leq \kappa$ for all $i \in S$. Then $\text{QUEScoreFilter}(S, \mathcal{O}_{\text{approx}})$ terminates after at most $S = O(\log \kappa)$ epochs, and outputs a $w \in \mathfrak{S}_{n,\varepsilon}$ so that*

$$\|\mu(w) - \mu\|_2 \leq O(\sqrt{\varepsilon \gamma_2} + \gamma_1).$$

Moreover, each epoch runs for at most $O(\log d)$ iterations, requires $O(\log d)$ calls to $\mathcal{O}_{\text{approx}}$, and requires $\tilde{O}(nd + n \log \kappa)$ additional computation.

We first prove correctness. The main lemma is the following per-epoch guarantee:

Lemma 6.6. *The following invariants always hold. For all epochs s , we have:*

- $w^s \in \mathfrak{S}_{n,\varepsilon}$, and
- If $\|M(w^{(s)})\|_2 > \frac{100}{1.1} \gamma_2$, then epoch s finishes after $O(\log d)$ iterations, and outputs $w^{(s+1)}$ so that $\|M(w^{(s+1)})\|_2 \leq \frac{2}{3} \|M(w^{(s)})\|_2$.

We first show how the lemma implies the theorem.

Proof of Theorem 6.5 given Lemma 6.6. We first prove correctness. The bound on the ℓ_2 norm of X_i for all $i \in S$ immediately also implies that $\|M(w)\|_2 \leq O(\kappa^2)$ for all $w \in \Gamma_n$, and in particular for $w = \frac{1}{n}\mathbf{1}_n$. By Lemma 6.6, after $S = O(\log \kappa)$ epochs the algorithm must terminate, and moreover every epoch can run for at most $O(\log d)$ iterations. Lemma 6.6 additionally implies that if w is the output of $\text{QUESCOREFILTER}(S)$, then $w \in \mathfrak{S}_{n,\varepsilon}$, and moreover, $\|\Sigma(w)\|_2 \leq 110\gamma_2$. Therefore, by Lemma 6.2, we have $\|\mu(w) - \mu\|_2 \leq O(\sqrt{\varepsilon\gamma_2} + \gamma_1)$, from which the desired conclusion immediately follows.

We now prove the runtime bound. In every iteration, besides the call to the oracle, the only costly operations are the approximate top eigenvalue computations and running 1DFILTER . However, the approximate top eigenvalue computations can be done in time $\tilde{O}(nd)$ via power method since we only ask for a constant multiplicative approximation, and the bound on $\|X_i\|_2$ implies that 1DFILTER runs in $O(n \log \kappa)$ time. \square

We now prove Lemma 6.6.

Proof. Clearly these invariants hold at the beginning of the algorithm. Since in the remainder of the proof we will only deal with a single epoch s , for conciseness we will omit the superscript. We will require the following claim:

Claim 6.7. *Suppose $w \in \mathfrak{S}_{n,\varepsilon}$ so that $\|M(w)\|_2 \geq 100\gamma_2$, and let $U \in \Delta_{d \times d}$. Let $\tau_i = (X_i - \mu(w))^\top U (X_i - \mu(w))$, and $\tilde{\tau}_i$ be so that $\tilde{\tau}_i \approx_{0.1} \tau_i$ for all i . Suppose that $\lambda \approx_{0.1} \|M(w)\|_2$, and $\sum_{i=1}^n \tilde{\tau}_i \geq \frac{1}{5}\lambda$. Then, if $w' = \text{1DFILTER}(w, \tilde{\tau}, 1/4)$, then $w' \in \mathfrak{S}_{n,\varepsilon}$, and $\langle M(w'), U \rangle \leq 0.31 \langle M(w), U \rangle$.*

Proof. We first show that $\sum_{i \in S_g} w_i \tau_i \leq c \sum_{i=1}^n w_i \tau_i$ for some universal constant $c \leq 0.11$. Let $\tilde{w}_i = \frac{1}{n}$ if $i \in S_g$ and $\tilde{w}_i = 0$ otherwise. Then, we have

$$\begin{aligned}
\sum_{i \in S_g} w_i \tau_i &= \left\langle \sum_{i \in S_g} w_i (X_i - \mu(w))(X_i - \mu(w))^\top, U \right\rangle \\
&\stackrel{(a)}{\leq} \left\langle \sum_{i=1}^n \tilde{w}_i (X_i - \mu(w))(X_i - \mu(w))^\top, U \right\rangle \\
&= \left\langle \sum_{i=1}^n \tilde{w}_i (X_i - \mu(\tilde{w}))(X_i - \mu(\tilde{w}))^\top, U \right\rangle + |\tilde{w}| \cdot (\mu(\tilde{w}) - \mu(w))^\top U (\mu(\tilde{w}) - \mu(w)) \\
&\stackrel{(b)}{\leq} (1 - \varepsilon) \langle M(\tilde{w}), U \rangle + \|\mu(\tilde{w}) - \mu(w)\|_2^2 \\
&\stackrel{(c)}{\leq} 2\gamma_2 + \|\mu(\tilde{w}) - \mu(w)\|_2^2 \\
&\leq 2\gamma_2 + 2\|\mu(\tilde{w}) - \mu\|_2^2 + 2\|\mu(w) - \mu\|_2^2 \\
&\stackrel{(d)}{\leq} 2\gamma_2 + 2\gamma_1^2 + 9(\gamma_2^2 + \varepsilon \|M(w)\|_2) < \frac{1}{60} \|M(w)\|_2 \\
&\leq \frac{1.1}{12} \lambda \leq \frac{1.1^2}{12} \sum_{i=1}^n w_i \tilde{\tau}_i,
\end{aligned} \tag{13}$$

for ε sufficiently small and $\|M(w)\|_2 > \frac{100}{1.1}\gamma_2$. Here (a) follows since $w_i \leq \tilde{w}_i$ for $i \in S_g$, (b) follows since $\|U\|_2 \leq 1$, (c) follows from ε -goodness of S_g , and (d) follows from ε -goodness of S_g and Lemma 6.2. Therefore overall we have $\sum_{i \in S_g} w_i \tau_i \leq \frac{1.1^3}{12} \sum_{i=1}^n w_i \tilde{\tau}_i$. Thus Theorem 5.2 applies, and w' satisfies $\sum_{i=1}^n w'_i \tilde{\tau}_i \leq \frac{1}{4} \sum_{i=1}^n w_i \tilde{\tau}_i$ as well as $w' \in \mathfrak{S}_{n,\varepsilon}$. Applying the guarantee that $\tilde{\tau}_i = (1 \pm 0.1)\tau_i$ again yields that

$$\langle M(w'), U \rangle = \sum_{i=1}^n w'_i \tau_i \leq 1.1 \sum_{i=1}^n w'_i \tilde{\tau}_i \leq \frac{1.1}{4} \sum_{i=1}^n w_i \tilde{\tau}_i \leq \frac{1.1^2}{4} \sum_{i=1}^n w_i \tau_i = \frac{1.1^2}{4} \langle M(w), U \rangle \leq 0.31 \langle M(w), U \rangle.$$

This completes the proof of the claim. \square

Notice that Claim 6.7 immediately implies that the first invariant $w \in \mathfrak{S}_{n,\varepsilon}$ always holds. We now turn to proving the second invariant. Let T be the number of iterations that the epoch runs for. Observe that for all $t = 0, \dots, T$, we have that $M(w_t) \preceq M(w_0) \preceq \frac{1}{\alpha} I$, and so we are indeed in the setting of the guarantee in Section 5.4. Thus, by (8), since $F_t = M(w_{t+1})$, we obtain the following regret bound:

$$\begin{aligned} \left\| \sum_{t=0}^{T-1} M(w_{t+1}) \right\|_2 &\leq \sum_{t=0}^{T-1} \langle M(w_{t+1}), U_t \rangle + \alpha \sum_{t=0}^{T-1} \langle U_t, M(w_{t+1}) \rangle \|M(w_{t+1})\|_2 + \frac{\log d}{\alpha} \\ &\leq 2 \sum_{t=0}^{T-1} \langle M(w_{t+1}), U_t \rangle + \|M(w_0)\|_2 \cdot \log d, \end{aligned} \quad (14)$$

where the second inequality follows by our choice of α . We claim that for all $t > 0$, we must have $\langle M(w_t), U_t \rangle \leq 0.31 \|M(w_0)\|_2$. There are two cases. If we enter the if statement in Line 18, then

$$\langle M_t, U_t \rangle = \sum_{i=1}^n w_i \tau_i \leq 1.1 \sum_{i=1}^n w_i \tilde{\tau}_i \leq \frac{1.1}{5} \lambda \leq 0.31 \|M(w_0)\|_2,$$

and $M(w_{t+1}) = M(w_t)$, so this is clearly satisfied. Otherwise, the desired bound follows by Claim 6.7. Thus overall, by (14) we have that

$$\left\| \sum_{t=0}^{T-1} M(w_t) \right\|_2 \leq T \cdot 0.62 \|M(w_0)\|_2 + \|M(w_0)\|_2 \cdot \log d. \quad (15)$$

By Lemma 6.4, we further have that $M(w_{t+1}) \preceq M(w_t)$ for all $t = 0, \dots, T-1$, and so this implies that

$$T \|M(w_T)\|_2 \leq 0.62 \|M(w_0)\|_2 + \|M(w_0)\|_2 \cdot \log d.$$

Simplifying both sides yields that if $T = C \log n$ for some sufficiently large constant C , then $\|M(w_T)\|_2 \leq \frac{2}{3} \|M(w_0)\|_2$. Thus after $O(\log d)$ iterations, we must terminate. \square

7 An MMW algorithm for robust mean estimation for sub-gaussian distributions

In this section we give an analog of the result in Section 6 but in the setting where the distribution D is subgaussian, and has identity covariance. We again first identify a deterministic condition for the inliers under which our algorithms will succeed. In this case, we need a stricter analog of ε -goodness. Specifically, we will require:

Definition 7.1 (Subgaussian goodness). Let D be a distribution with covariance Id and mean μ . We say a set of points S is $(\varepsilon, \gamma_1, \gamma_2)$ -subgaussian good (or $(\varepsilon, \gamma_1, \gamma_2)$ -s.g. good for short) with respect to D if there exist universal constants C_1, C_2 so that the following inequalities are satisfied:

- $\|\mu(S) - \mu\| \leq \gamma_1$ and $\left\| \frac{1}{|S|} \sum_{i \in S} (X_i - \mu(S)) (X_i - \mu(S))^\top - \text{Id} \right\|_2 \leq \gamma_2$, and
- For any subset $T \subset S$ so that $|T| = 2\varepsilon|S|$, we have

$$\left\| \frac{1}{|T|} \sum_{i \in T} X_i - \mu \right\| \leq C_1 \cdot \sqrt{\log 1/\varepsilon}, \text{ and } \left\| \frac{1}{|T|} \sum_{i \in T} (X_i - \mu(S)) (X_i - \mu(S))^\top - \text{Id} \right\|_2 \leq C_2 \cdot \log 1/\varepsilon.$$

We have the following concentration inequality:

Lemma 7.1 (see Lemmata 2.1.8 and 2.1.9 in [11]). *Let $X_1, \dots, X_n \sim D$, where D is subgaussian with variance proxy 1. Then, for any $\varepsilon > 0$ sufficiently small, we have that $S = \{X_1, \dots, X_n\}$ is $(\varepsilon, \gamma_1, \gamma_2)$ -s.g. good with probability $1 - \delta$, where*

$$\gamma_1 = O\left(\sqrt{\frac{d + \log 1/\delta}{n}}\right), \text{ and } \gamma_2 = O\left(\min \sqrt{\frac{d + \log 1/\delta}{n}}, \frac{d + \log 1/\delta}{n}\right). \quad (16)$$

In particular, note that when $n = \Omega(\frac{d + \log 1/\delta}{\varepsilon^2 \log 1/\varepsilon})$, then Lemma 7.1 implies that n i.i.d. samples from an isotropic sub-gaussian distribution is $(\varepsilon, O(\varepsilon \sqrt{\log 1/\varepsilon}), O(\varepsilon \sqrt{\log 1/\varepsilon}))$ -s.g. good with probability $1 - \delta$. We will also require the following simple consequences of subgaussian goodness.

Fact 7.2. *Let D be an isotropic distribution. Let S be $(\varepsilon, \gamma_1, \gamma_2)$ -s.g. good w.r.t. D . Then:*

- *for all $w' \in \Gamma_n$ with $w' \leq \frac{1}{n} \mathbf{1}_n$ and $|w'| \leq 2\varepsilon$, and for all unit vectors $v \in \mathbb{R}^d$, we have*

$$\sum_{i=1}^n w'_i \langle X_i - \mu, v \rangle^2 \leq O(\varepsilon \log 1/\varepsilon).$$

- *if $w \in \Gamma_n$ satisfies $w \leq \frac{1}{n} \mathbf{1}_n$ and $|\frac{1}{n} \mathbf{1}_n - w| \leq 2\varepsilon$, then*

$$\left\| \sum_{i=1}^n w_i (X_i - \mu) (X_i - \mu)^\top - \text{Id} \right\|_2 \leq \gamma_2 + \gamma_1^2 + O(\varepsilon \log 1/\varepsilon), \text{ and} \quad (17)$$

$$\left\| \sum_{i=1}^n w_i (X_i - \mu(w)) (X_i - \mu(w))^\top - \text{Id} \right\|_2 \leq \gamma_2 + 4\gamma_1^2 + O(\varepsilon \log 1/\varepsilon). \quad (18)$$

Proof. We first prove the first claim. Let $w'' \in \Gamma_n$ be anything so that $w' \leq w'' \leq \frac{1}{n} \mathbf{1}_n$ and $|w''| = \varepsilon$. Then since all quantities on the LHS of the expression are nonnegative, we have that

$$\sum_{i=1}^n w'_i \langle X_i - \mu, v \rangle^2 \leq \sum_{i=1}^n w''_i \langle X_i - \mu, v \rangle^2. \quad (19)$$

Now let $A = \{w'' \in \Gamma_n : |w''| = \varepsilon\}$. This set is clearly convex, and moreover, by inspection, the vertices of A are exactly given by $\frac{1}{n} \mathbf{1}_T$ where $|T| = 2\varepsilon n$. Thus, by convexity, the maximum of the RHS of (19) over $w'' \in A$ is obtained by $w'' = \frac{1}{n} \mathbf{1}_T$ for some T with $|T| = 2\varepsilon n$. But then we have

$$\begin{aligned} \frac{1}{n} \sum_{i \in T} \langle X_i - \mu, v \rangle^2 &= 2\varepsilon \cdot \frac{1}{|T|} \sum_{i \in T} (\langle X_i - \mu, v \rangle^2 - 1) + \varepsilon \\ &\leq 2\varepsilon \left\| \frac{1}{|T|} \sum_{i \in T} (X_i - \mu) (X_i - \mu)^\top - \text{Id} \right\|_2 + \varepsilon \\ &\leq O(\varepsilon \log 1/\varepsilon), \end{aligned}$$

by the s.g.-goodness of S . This completes the proof of the first bullet point.

We now turn our attention to the second claim. We have that

$$\begin{aligned} \left\| \sum_{i=1}^n w_i (X_i - \mu) (X_i - \mu)^\top - \text{Id} \right\|_2 &= \left\| \sum_{i=1}^n \frac{1}{n} (X_i - \mu) (X_i - \mu)^\top - \text{Id} + \sum_{i=1}^n \left(\frac{1}{n} - w_i \right) (X_i - \mu) (X_i - \mu)^\top \right\|_2 \\ &\leq \left\| \sum_{i=1}^n \frac{1}{n} (X_i - \mu) (X_i - \mu)^\top - \text{Id} \right\|_2 + O(\varepsilon \log 1/\varepsilon), \end{aligned}$$

by the first claim. Further expanding, we have

$$\begin{aligned}
\left\| \sum_{i=1}^n \frac{1}{n} (X_i - \mu) (X_i - \mu)^\top - \text{Id} \right\|_2 &= \left\| \sum_{i=1}^n \frac{1}{n} (X_i - \mu(S)) (X_i - \mu(S))^\top - \text{Id} + |w| (\mu(S) - \mu) (\mu(S) - \mu)^\top \right\|_2 \\
&\leq \left\| \sum_{i=1}^n \frac{1}{n} (X_i - \mu(S)) (X_i - \mu(S))^\top - \text{Id} \right\|_2 + \|\mu(S) - \mu\|_2^2 \\
&\leq \gamma_2 + \gamma_1^2.
\end{aligned}$$

Putting it all together yields (17). To prove (18), simply observe that

$$\begin{aligned}
\left\| \sum_{i=1}^n w_i (X_i - \mu) (X_i - \mu)^\top - \sum_{i=1}^n w_i (X_i - \mu(w)) (X_i - \mu(w))^\top \right\|_2 &= |w| \left\| (\mu(w) - \mu) (\mu(w) - \mu)^\top \right\|_2 \\
&\leq O(\gamma_1^2) + O(\epsilon \log(1/\epsilon)).
\end{aligned}$$

In the last step we have used the definition of $(\epsilon, \gamma_1, \gamma_2)$ -s.g. goodness and convexity. \square

As a result, we also have the following tail bound on mean shifts caused by small subsets of points:

Corollary 7.3. *Let D be an isotropic distribution, and let S be $(\epsilon, \gamma_1, \gamma_2)$ -s.g. good w.r.t. D . Then:*

- for all $w' \in \Gamma_n$ with $w' \leq \frac{1}{n} \mathbf{1}_n$ and $|w'| \leq 2\epsilon$, we have

$$\left\| \sum_{i=1}^n w'_i (X_i - \mu) \right\|_2 \leq O\left(\epsilon \sqrt{\log 1/\epsilon}\right), \text{ and}$$

- if $w \in \Gamma_n$ satisfies $w \leq \frac{1}{n} \mathbf{1}_n$ and $|\frac{1}{n} \mathbf{1}_n - w| \leq 2\epsilon$, then

$$\|\mu(w) - \mu\|_2 \leq \frac{1}{1 - \epsilon} \left(\gamma_1 + O\left(\epsilon \sqrt{\log 1/\epsilon}\right) \right).$$

Proof. We first prove the first claim. Fix any unit vector $v \in \mathbb{R}^d$. Then we have

$$\begin{aligned}
\left(\sum_{i=1}^n w'_i \langle X_i - \mu, v \rangle \right)^2 &\stackrel{(a)}{\leq} |w'| \sum_{i=1}^n w'_i \langle X_i - \mu, v \rangle^2 \\
&\stackrel{(b)}{\leq} O(\epsilon^2 \log 1/\epsilon),
\end{aligned}$$

where (a) follows from Cauchy-Schwarz, and (b) follows from Fact 7.2. By taking square roots and a supremum over all unit vectors v , we obtain the desired conclusion.

We now prove the second claim. We expand

$$\begin{aligned}
|w| \|\mu(w) - \mu\|_2 &= \left\| \sum_{i=1}^n w_i (X_i - \mu) \right\|_2 \\
&= \left\| \sum_{i=1}^n \frac{1}{n} (X_i - \mu) + \sum_{i=1}^n \left(\frac{1}{n} - w_i \right) (X_i - \mu) \right\|_2 \\
&\leq \gamma_1 + O(\epsilon \sqrt{\log 1/\epsilon}),
\end{aligned}$$

where the last line follows from subgaussian goodness, and applying the first claim with $w'_i = \frac{1}{n} - w_i$. \square

7.1 Mean deviations to moment bounds

As before, we will require a lemma which relates the mean shift caused by a small fraction of points to spectral deviations. However, because in this case we will assume that our data is subgaussian, we will be able to prove stronger statements, which will in turn allow us to achieve much better error. We first record the following simple fact, which states that if we have a set of weights that puts almost all of its mass on a good set, then the restriction of that set of weights to the good set satisfies the conditions of the lemmata proved in the above section.

Fact 7.4. *Let $\varepsilon < 1/2$, and suppose $S = S_g \cup S_b \setminus S_r$, where S_g is $(\varepsilon, \gamma_1, \gamma_2)$ -s.g. good, and $|S_b|, |S_r| \leq \varepsilon|S|$. Let $w \in \mathfrak{S}_{n,\varepsilon}$. Then $w_g \leq \frac{1}{|S_g|} \mathbf{1}_{S_g}$ and $\left| \frac{1}{|S_g|} \mathbf{1}_{S_g} - w_g \right| \leq 2\varepsilon$.*

We first show that, no matter what, the smallest eigenvalue of the empirical covariance we choose cannot be too small. Formally:

Lemma 7.5. *Let $\gamma_1, \gamma_2 > 0$. Suppose $S = S_g \cup S_b \setminus S_r$, where S_g is $(\varepsilon, \gamma_1, \gamma_2)$ -s.g. good, and $|S_b|, |S_r| \leq \varepsilon|S|$. Let $w \in \mathfrak{S}_{n,\varepsilon}$. Then*

$$\sum_{i \in S_g \cap S} w_i (X_i - \mu(w)) (X_i - \mu(w))^\top \succeq (1 - \xi) \text{Id},$$

where $\xi = \gamma_2 + 2\gamma_1^2 + O(\varepsilon \log 1/\varepsilon)$.

Proof. Let $w' \in \mathfrak{S}_{n,\varepsilon}$ be defined by $w'_i = w_i$ if $i \in S_g \cap S$ and $w'_i = 0$ otherwise. By Lemma 6.4, we know that

$$\begin{aligned} \sum_{i \in S_g \cap S} w_i (X_i - \mu(w)) (X_i - \mu(w))^\top &\succeq \sum_{i \in S_g \cap S} w_i (X_i - \mu(w')) (X_i - \mu(w'))^\top \\ &\succeq (1 - (\gamma_2 + 2\gamma_1^2 + O(\varepsilon \log 1/\varepsilon))) \text{Id}, \end{aligned}$$

by 18 of Fact 7.2. □

We now show the following:

Lemma 7.6. *Let $\gamma_1, \gamma_2 > 0$. Suppose $S = S_g \cup S_b \setminus S_r$, where S_g is $(\varepsilon, \gamma_1, \gamma_2)$ -s.g. good, and $|S_b|, |S_r| \leq \varepsilon|S|$. Let $w \in \mathfrak{S}_{n,\varepsilon}$, and let $\lambda = \|M(w_t) - \text{Id}\|_2$. Then*

$$\|\mu(w) - \mu\|_2 \leq \frac{1}{1 - \varepsilon} \cdot \left(2\gamma_1 + \sqrt{\varepsilon(\lambda + \gamma_2)} + O(\varepsilon \sqrt{\log 1/\varepsilon}) \right).$$

Before we prove this lemma, observe that if $\gamma = O(\varepsilon \sqrt{\log 1/\varepsilon})$ and $\gamma_2 = O(\varepsilon \log 1/\varepsilon)$ and $\varepsilon \leq 1/2$ then the RHS of the lemma simplifies to $O(\varepsilon \sqrt{\log 1/\varepsilon})$.

Proof of Lemma 7.6. Let $\rho = \mu(w) - \mu$. As before, we have the following sequence of identities:

$$\begin{aligned} |w| \cdot \|\mu(w) - \mu\|_2^2 &= |w| \cdot \langle \mu(w) - \mu, \rho \rangle \\ &= \sum_{i=1}^n w_i \langle X_i - \mu, \rho \rangle \\ &= \underbrace{\sum_{i \in S_g \cap S} w_i \langle X_i - \mu, \rho \rangle}_{W_0} + \underbrace{\sum_{i \in S_b} w_i \langle X_i - \mu, \rho \rangle}_{W_1}. \end{aligned}$$

We treat the two terms on the RHS separately. We first consider W_0 . We continue expanding, and observe:

$$\begin{aligned}
\left| \sum_{i \in S_g \cap S} w_i \langle X_i - \mu, \rho \rangle \right| &= \left| \sum_{i \in S_g \cap S} \frac{1}{n} \langle X_i - \mu, \rho \rangle + \sum_{i \in S_g \cap S} \left(\frac{1}{n} - w_i \right) \langle X_i - \mu, \rho \rangle \right| \\
&\leq \left| \sum_{i \in S_g} \frac{1}{n} \langle X_i - \mu, \rho \rangle \right| + \left| \sum_{i \in S_r} \frac{1}{n} \langle X_i - \mu, \rho \rangle \right| + \left| \sum_{i \in S_g \cap S} \left(\frac{1}{n} - w_i \right) \langle X_i - \mu, \rho \rangle \right| \\
&\stackrel{(a)}{\leq} \left| \sum_{i \in S_g} \frac{1}{n} \langle X_i - \mu, \rho \rangle \right| + O\left(\varepsilon \sqrt{\log 1/\varepsilon}\right) \|\rho\|_2 \\
&\stackrel{(b)}{\leq} \left(\gamma_1 + O\left(\varepsilon \sqrt{\log 1/\varepsilon}\right) \right) \|\rho\|_2,
\end{aligned} \tag{20}$$

where (a) follows from two applications of Corollary 7.3, and (b) follows from Cauchy-Schwarz and subgaussian goodness.

We now turn our attention to bounding W_1 . We have

$$\begin{aligned}
|W_1| &\leq \left| \sum_{i \in S_b} w_i \langle X_i - \mu(w), \rho \rangle \right| + \sum_{i \in S_b} w_i \|\rho\|_2^2 \\
&\leq \left| \sum_{i \in S_b} w_i \langle X_i - \mu(w), \rho \rangle \right| + \varepsilon \|\rho\|_2^2.
\end{aligned}$$

Focusing in on the first term in the RHS, we have

$$\begin{aligned}
\left(\sum_{i \in S_b} w_i \langle X_i - \mu(w), \rho \rangle \right)^2 &\stackrel{(a)}{\leq} \left(\sum_{i \in S_b} w_i \right) \sum_{i \in S_b} w_i \langle X_i - \mu(w), \rho \rangle^2 \\
&= \left(\sum_{i \in S_b} w_i \right) \sum_{i \in S_b} w_i \left(\langle X_i - \mu(w), \rho \rangle^2 - \|\rho\|_2^2 \right) + \left(\sum_{i \in S_b} w_i \right)^2 \|\rho\|_2^2 \\
&\stackrel{(b)}{\leq} \varepsilon \sum_{i \in S_b} w_i \left(\langle X_i - \mu(w), \rho \rangle^2 - 1 \right) + \varepsilon^2 \|\rho\|_2^2,
\end{aligned} \tag{21}$$

where (a) follows from Cauchy-Schwarz, and (b) follows since w places at most ε mass on S_b . Now observe that

$$\begin{aligned}
\left| \sum_{i \in S_b} w_i \left(\langle X_i - \mu(w), \rho \rangle^2 \right) \right| &= \left| \sum_{i=1}^n w_i \left(\langle X_i - \mu(w), \rho \rangle^2 \right) - \sum_{i \in S_g \cap S} w_i \left(\langle X_i - \mu(w), \rho \rangle^2 \right) \right| \\
&\leq \|M(w_t) - \text{Id}\|_2 \|\rho\|_2^2 + \left| \sum_{i \in S_g \cap S} w_i \left(\langle X_i - \mu(w), \rho \rangle^2 - \|\rho\|_2^2 \right) \right| \\
&\leq \|M(w_t) - \text{Id}\|_2 \|\rho\|_2^2 + \left\| \sum_{i \in S_g \setminus S_r} w_i (X_i - \mu(w)) (X_i - \mu(w))^\top - \text{Id} \right\|_2 \|\rho\|_2^2 \\
&\leq (\|M(w_t) - \text{Id}\|_2 + \gamma_2 + 2\gamma_1^2 + O(\varepsilon \log 1/\varepsilon)) \|\rho\|_2^2,
\end{aligned}$$

by Fact 7.2, where we take the convention that $w_i = 0$ for $i \in S_r$. Hence, combining terms, recalling the definition of $\lambda = \|M(w_t) - \text{Id}\|_2$, and taking square roots, we have

$$|W_1| \leq \left(\sqrt{\varepsilon(\lambda + \gamma_2)} + \gamma_1 \sqrt{2\varepsilon} + O\left(\varepsilon \sqrt{\log 1/\varepsilon}\right) \right) \cdot \|\rho\|_2 + \varepsilon \|\rho\|_2^2.$$

Combining this with (20) yields

$$\begin{aligned}\|\rho\|_2^2 &\leq |W_0| + |W_1| \\ &\leq \left(2\gamma_1 + \sqrt{\varepsilon(\lambda + \gamma_2)} + O(\varepsilon\sqrt{\log 1/\varepsilon})\right) \|\rho\|_2 + \varepsilon \|\rho\|_2^2.\end{aligned}$$

Solving for $\|\rho\|_2$ yields the desired claim. \square

7.2 Algorithm description

The algorithm is quite similar to the algorithm presented in Section 6 for the bounded covariance case. The formal pseudocode is presented in Algorithm 4. For any $w \in \Gamma_n$, let $\mu(w)$ and $M(w)$ be as in Section 6. However, we will require a slightly stronger notion of score oracle than before.

Recall that before, given a dataset S , and a sequence of weights w_0, \dots, w_{t-1} , the score oracle is asked to produce multiplicative approximations to $\tau_{t,i}$ where $\tau_{t,i}$ is defined as (10). One consequence of this is that this allows us to produce multiplicative approximations to $\langle M(w_t^{(s)}), U_t^{(s)} \rangle = \sum_{i=1}^n w_{t,i} \tau_{t,i}$. However, we will require multiplicative approximations to $\langle M(w_t^{(s)}) - \text{Id}, U_t^{(s)} \rangle$, which cannot be obtained black-box via multiplicative approximations to the original scores.

To rectify this, we say that an algorithm \mathcal{O}^* is an *augmented score oracle* if it takes a dataset S and a sequence of weights w_0, \dots, w_{t-1} , and outputs $\tilde{\tau}_{t,i}$ for all $i = 1, \dots, n$, but also an overall score q_t which is intended to approximate $\langle M(w_t^{(s)}) - \text{Id}, U_t^{(s)} \rangle$.

Given S and such an oracle \mathcal{O}^* , the algorithm again proceeds in epochs. Initially, we let $w^{(0)} = \frac{1}{n} \mathbf{1}_n$. In epoch $s = 0, \dots, L-1$, we proceed as follows. First, compute $\lambda^{(s)} \approx_{0.1} \|M(w^{(s)}) - \text{Id}\|_2$. If $\lambda^{(s)} \leq O(\gamma_2 + \gamma_1 + \gamma_1^2 + \varepsilon \log 1/\varepsilon)$, we terminate and output $\mu(w^{(s)})$.

Otherwise, we let $w_0^{(s)} = w^s$. Then, in iteration $t = 0, \dots, T_s - 1$, we first (approximately) compute $\lambda_t^{(s)} \approx_{0.1} \|M(w_t^{(s)}) - \text{Id}\|_2$. If $\lambda_t^{(s)} \leq \frac{1}{2} \lambda_0^{(s)}$, we terminate and let $w^{(s+1)} = w_t^{(s)}$. Otherwise, we let $U_t^{(s)}$ be prescribed by the MMW update with parameter $\alpha = 1/(1.1 \cdot \lambda^{(s)})$. Then, produce the gain matrix is given as follows.

At time t , run \mathcal{O}^* given S and the sequence of weights $w_0^{(s)}, \dots, w_{t-1}^{(s)}$ to obtain scores $\tilde{\tau}_{t,i}^{(s)}$ as well as an overall score $\tilde{q}_t^{(s)}$. Then, check if $\tilde{q}_t^{(s)} \leq \frac{1}{5} \lambda_0^{(s)}$. If so, then we let $w_{t+1}^{(s)} = w_t^{(s)}$.

Otherwise, sort the $\tilde{\tau}_{t,i}^{(s)}$ in descending order. WLOG assume that $\tilde{\tau}_{t,1}^{(s)} \geq \tilde{\tau}_{t,2}^{(s)} \geq \dots \geq \tilde{\tau}_{t,n}^{(s)}$. Let m be the smallest integer so that $\sum_{i \leq m} w_{t,i}^{(s)} \geq 2\varepsilon$. Then, run 1DFILTER on $\tilde{\tau}_{t,1}^{(s)}, \dots, \tilde{\tau}_{t,m}^{(s)}$ with corresponding weights $w_{t,1}^{(s)}, \dots, w_{t,m}^{(s)}$, to obtain a new set of weights w'_1, \dots, w'_m , and let $w_{t+1,i}^{(s)}$ be defined by

$$w_{t+1,i}^{(s)} = \begin{cases} w_{t,i}^{(s)} & \text{if } i > m; \\ w'_i & \text{if } i \leq m. \end{cases} \quad (22)$$

That is, we find the largest 2ε -percentile of the scores weighted by the current weights, and run the univariate filter on these set of weights, leaving the other weights unchanged. Finally, we output the gain matrix $F_t^{(s)} = M(w_{t+1}^{(s)}) - \text{Id}$. Notice that this matrix may not be PSD.

The score oracles The exact score oracle $\mathcal{O}_{\text{exact}}^*$ would, given S , and given $w_0^{(s)}, \dots, w_{t-1}^{(s)}$, would output scores $\tau_{t,i}^{(s)}$ given by (10) with $U_t^{(s)}$ as given by Algorithm 4. The exact score oracle would also output

$$q_t^{(s)} = \langle M(w_t^{(s)}) - \text{Id}, U_t^{(s)} \rangle. \quad (23)$$

Observe that the fact that the $F_t^{(s)}$ includes a negative identity term does not affect these scores at all, and indeed we can take $U_t^{(s)}$ to be as in (11), since the parameter c is chosen in any case to normalize $U_t^{(s)}$ to have trace 1.

Algorithm 4 MMW-based filtering method for robust mean estimation for subgaussian distributions

```

1: Input: dataset  $S \subset \mathbb{R}^d$  of size  $n$ , parameters  $\gamma_1, \gamma_2$ , augmented score oracle  $\mathcal{O}$ 
2: Let  $C > 0$  be a sufficiently large universal constant.
3: Let  $w^{(0)} = \frac{1}{n} \mathbf{1}_n$ .
4: for epoch  $s = 0, \dots, O(\log \kappa)$  do
5:   Let  $\lambda^{(s)} \approx_{0.1} \|M(w^{(s)}) - \text{Id}\|_2$ 
6:   if  $\lambda^{(s)} \leq C \cdot (\gamma_2 + \gamma_1 + \gamma_1^2 + \varepsilon \log 1/\varepsilon)$  then
7:     return  $\mu(w^{(s)})$ 
8:   end if
9:   Let  $w_0^{(s)} = w^{(s)}$ 
10:  Let  $\alpha^{(s)} = \frac{1}{1.1 \cdot \lambda_0^{(s)}}$ 
11:  for iteration  $t = 0, \dots, O(\log d)$  do
12:    Compute  $\lambda_t^{(s)} \approx_{0.1} \|M(w_t^{(s)}) - \text{Id}\|_2$ .
13:    if  $\lambda_t^{(s)} \leq \frac{1}{2} \lambda_0^{(s)}$  then
14:      terminate epoch
15:    end if
16:    Let  $U_t^{(s)}$  be given by MMW update with parameter  $\alpha^{(s)}$ 
17:    For  $i = 1, \dots, n$ , let  $\tilde{\tau}_{t,i}^{(s)}, \tilde{q}_t^{(s)} = \mathcal{O}(S, w_0^{(s)}, \dots, w_n^{(s)})$ 
18:    if  $\tilde{q}_t^{(s)} \leq \frac{1}{1.5} \lambda_0^{(s)}$  then
19:      Let  $w_{t+1}^{(s)} = w_t^{(s)}$ .
20:    else
21:      Sort the  $\tilde{\tau}_{t,i}^{(s)}$  in descending order.
22:      WLOG assume that  $\tilde{\tau}_{t,1}^{(s)} \geq \tilde{\tau}_{t,2}^{(s)} \geq \dots \geq \tilde{\tau}_{t,n}^{(s)}$ 
23:      Let  $m$  be the smallest integer so that  $\sum_{i \leq m} w_{t,i}^{(s)} \geq 2\varepsilon$ .
24:      Let  $w' = \text{1DFILTER}((w_{t,1}^{(s)}, \dots, w_{t,m}^{(s)}), (\tilde{\tau}_{t,1}^{(s)}, \dots, \tilde{\tau}_{t,m}^{(s)}), 1/4)$ .
25:      Let  $w_{t+1}^{(s)}$  be as defined in (22).
26:    end if
27:    Let feedback matrix to MMW update be  $F_t = M(w_{t+1}^{(s)})$ 
28:  end for
29:  Let  $w^{(s+1)} = w_t^{(s)}$ 
30: end for

```

As before, we cannot access these exact scores in nearly-linear time, so instead we ask for approximations. Specifically, we will assume an approximate augmented score oracle $\mathcal{O}_{\text{approx}}^*$, which given S and $w_0^{(s)}, \dots, w_{t-1}^{(s)}$, output scores $\tilde{\tau}_{t,i}^{(s)}$ so that $\tilde{\tau}_{t,i}^{(s)} \approx_{0.1} \tau_{t,i}^{(s)}$ for all $i = 1, \dots, n$, as well as $\tilde{q}_t^{(s)}$ satisfying

$$\left| \tilde{q}_t^{(s)} - q_t \right| \leq 0.1 \cdot q_t + 0.05 \cdot \|M(w_t) - \text{Id}\|_2.$$

As before, the choice of constants here is arbitrary, and any constants sufficiently small will work. In Section 8 we construct such an approximate augmented score oracle in nearly-linear time.

7.3 Correctness of s.g.-QUEScoreFilter

The rest of this section is dedicated to the proof of the following theorem:

Theorem 7.7. *Let D be a subgaussian isotropic distribution on \mathbb{R}^d with mean μ . Let $\varepsilon < c$, where c is a universal constant, and let $\gamma_1, \gamma_2 > 0$. Let S be a dataset, $|S| = n$, so that $S = S_g \cup S_b \setminus S_r$ so that S_g is $(\varepsilon, \gamma_1, \gamma_2)$ -s.g. good with respect to D , and $|S_b|, |S_r| \leq \varepsilon |S|$. Suppose that $\|X_i\|_2 \leq \kappa_1$ for all $i = 1, \dots, n$,*

and moreover $\|M(\frac{1}{n}\mathbf{1}_n) - \text{Id}\| \leq O(\varepsilon \log 1/\varepsilon + \varepsilon \kappa_2)$. Then $\text{S.G.-QUESCOREFILTER}(S, \varepsilon, \mathcal{O}_{\text{approx}})$ terminates after at most $O(\log \kappa_2)$ epochs, and outputs a $w \in \mathfrak{S}_{n,\varepsilon}$ so that

$$\|\mu(w) - \mu\|_2 \leq O\left(\gamma_1 + \varepsilon \sqrt{\log 1/\varepsilon} + \sqrt{\varepsilon(\gamma_1 + \gamma_2)}\right).$$

Moreover, each epoch runs for at most $O(\log d)$ iterations, requires $O(\log d)$ calls to $\mathcal{O}_{\text{approx}}$, and requires $\tilde{O}(nd + n \log \kappa_1)$ additional computation.

Our main lemma is the following:

Lemma 7.8. *The following invariants always hold. There exists some universal constant $C > 0$ so that for all epochs s , we have:*

- $w^{(s)} \in \mathfrak{S}_{n,\varepsilon}$, and
- If

$$\|M(w^{(s)}) - \text{Id}\|_2 > C \cdot (\gamma_2 + \gamma_1 + \gamma_1^2 + \varepsilon \log 1/\varepsilon),$$

then epoch s terminates after $O(\log d)$ iterations, and outputs $w^{(s+1)}$ so that $\|M(w^{(s+1)}) - \text{Id}\|_2 \leq \frac{3}{4} \|M(w^{(s)}) - \text{Id}\|_2$.

We first demonstrate how this lemma proves Theorem 7.7.

Proof of Theorem 7.7 given Lemma 7.8. By our condition on $M(\frac{1}{n}\mathbf{1}_n)$, after at most $s = O(\log \kappa_2)$ iterations, we must have that

$$\|M(w^{(s)}) - \text{Id}\|_2 \leq C \cdot (\gamma_2 + \gamma_1 + \gamma_1^2 + \varepsilon \log 1/\varepsilon).$$

Since $w^{(s)} \in \mathfrak{S}_{n,\varepsilon}$, Lemma 7.6 implies that for $\varepsilon < c$ sufficiently small, we have

$$\|\mu(w) - \mu\|_2 \leq O\left(\gamma_1 + \varepsilon \sqrt{\log 1/\varepsilon} + \sqrt{\varepsilon(\gamma_1 + \gamma_2)}\right),$$

as claimed.

We now turn to bounding the runtime. As in Theorem 6.5, it is clear that we make at most $\log d$ calls to the oracle every epoch, and 1DFILTER runs in time $O(n \log \kappa_1)$. Moreover, the approximate eigenvalue computations can still be done in $\tilde{O}(nd)$ time since we may run power method on $M(w_s^{(t)}) - \text{Id}$, as we can evaluate matrix-vector multiplications against this matrix in $O(nd)$ time. This completes the proof. \square

The remainder of the section is dedicated to the proof of this lemma. As in the previous section, for simplicity of notation, as we will only consider a fixed epoch s , we will drop the superscripts.

The proof of Lemma 7.8 breaks down into two parts. First, we will show that assuming we have not yet made sufficient progress, we remain in the regime where the filter is guaranteed to make progress, i.e., the majority of the mass of the τ_i are from bad points. This is captured in the following lemma:

Lemma 7.9. *At time t , suppose that $\langle M_t - \text{Id}, U_t \rangle > \frac{1}{1.15} \lambda_0$. Then $w_{t+1} \in \mathfrak{S}_{n,\varepsilon}$ and $\langle F_t, U_t \rangle \leq \frac{1}{4} \langle F_{t-1}, U_t \rangle$.*

Then, we will show that this implies that the regret bounds of MMW guarantee that we make constant progress in logarithmically many iterations:

Lemma 7.10. *Suppose for all $t = 0, \dots, T-1$, Lemma 7.9 holds, where $T = O(\log d)$. Then, $\|M(w_T) - \text{Id}\|_2 \leq 0.63 \cdot \|M(w_0) - \text{Id}\|$.*

Proof of Lemma 7.9. Recall that m was chosen to be the 2ε -percentile of the scores under the weighting given by w , and as before, for simplicity assume that the $\tilde{\tau}_{t,i}$ are in decending order. The main work in this proof will be to show that the scores $\tilde{\tau}_{t,1}, \dots, \tilde{\tau}_{t,m}$ and weights $w_{t,1}, \dots, w_{t,m}$ satisfy the conditions of Theorem 5.2.

The condition that $\langle M_t - \text{Id}, U_t \rangle > \frac{1}{1.15} \lambda_0$ implies that in this case, we will run the univariate filter. Let $S'_g = S_g \cap [m]$ and let $S'_b = S_b \cap [m]$, and let w'_g and w'_b the restriction of w_t to S'_g and S'_b , respectively. Observe that $\sum_{i \in S'_b} w_i \leq \sum_{i \in S_b} w_i \leq \varepsilon$, and therefore $\sum_{i \in S'_g} w_i \in [\varepsilon, 2\varepsilon]$. We then have

$$\begin{aligned}
\sum_{i \in S'_g} w_{t,i} \tau_{t,i} &= \sum_{i \in S'_g} w_{t,i} \left\langle (X_i - \mu(w_t)) (X_i - \mu(w_t))^\top, U_t \right\rangle \\
&\stackrel{(a)}{\leq} 2 \sum_{i \in S'_g} w_{t,i} \left\langle (X_i - \mu) (X_i - \mu)^\top, U_t \right\rangle + 2|w'_g| \left\langle (\mu - \mu(w_t))(\mu - \mu(w_t))^\top, U_t \right\rangle \\
&\stackrel{(b)}{\leq} O(\varepsilon \log 1/\varepsilon) + 2|w'_g| \left\langle (\mu - \mu(w_t))(\mu - \mu(w_t))^\top, U_t \right\rangle \\
&\leq O(\varepsilon \log 1/\varepsilon) + 4\varepsilon \|\mu - \mu(w_t)\|_2^2 \\
&\stackrel{(c)}{\leq} O(\varepsilon \log 1/\varepsilon) + 4\varepsilon \left(\gamma_1 + O(\varepsilon \sqrt{\log 1/\varepsilon}) + \sqrt{\varepsilon(\lambda_t + \gamma_2)} \right)^2 \\
&\leq O(\varepsilon \log 1/\varepsilon) + 8\varepsilon \gamma_1 + 8\varepsilon^2(\lambda_t + \gamma_2) \\
&\leq \frac{1}{30} \langle M(w_t) - \text{Id}, U_t \rangle. \tag{24}
\end{aligned}$$

where (a) follows since for any vectors x, y, z , we have $(x - y)(x - y)^\top \preceq 2(x - z)(x - z)^\top + 2(y - z)(y - z)^\top$, (b) follows from Fact 7.2, (c) follows from Lemma 7.6, and the last line follows from our assumption on $\langle M(w_t) - \text{Id}, U_t \rangle$. From this we conclude that $\sum_{i \in S'_g} w_{t,i} \tilde{\tau}_{t,i} \leq \frac{1}{29} \langle M(w_t) - \text{Id}, U_t \rangle$.

Note that as a consequence of this, we have that $\tilde{\tau}_{t,m} \leq O(\log 1/\varepsilon) + 8\varepsilon \gamma_1 + 8\varepsilon(\lambda_t + \gamma_2)$, since that is an upper bound on the average value of the $\tilde{\tau}_{t,i}$ for $i \in S'_g$, as $|w'_g| \geq \varepsilon$.

We now turn to lower bound the contribution from S'_b . First observe that

$$\begin{aligned}
\sum_{i \in S_b} w_i \tau_{t,i} &= \sum_{i \in S_b} w_{t,i} \left\langle (X_i - \mu(w_t)) (X_i - \mu(w_t))^\top, U_t \right\rangle \\
&= \sum_{i \in S_b} w_{t,i} \left\langle (X_i - \mu(w_t)) (X_i - \mu(w_t))^\top - \text{Id}, U_t \right\rangle - |w_b| \\
&= \langle M(w_t) - \text{Id}, U_t \rangle - \sum_{i \in S_g \cap S} w_{t,i} \left\langle (X_i - \mu(w_t)) (X_i - \mu(w_t))^\top - \text{Id}, U_t \right\rangle - |w_b| \\
&\stackrel{(a)}{\geq} \langle M(w_t) - \text{Id}, U_t \rangle - (\gamma_2 + 2\gamma_1^2 + O(\varepsilon \log 1/\varepsilon)) \\
&\geq \frac{99}{100} \langle M(w_t) - \text{Id}, U_t \rangle,
\end{aligned}$$

where (a) follows from Fact 7.2 and since $|w_b| \leq \varepsilon$, and the last inequality follows by our assumption on

$\langle M(w_t) - \text{Id}, U_t \rangle$. Therefore

$$\begin{aligned}
\sum_{i \in S'_b} w_i \tau_{t,i} &= \sum_{i \in S_b} w_i \tau_{t,i} - \sum_{i \in S_b \setminus S'_b} w_i \tau_{t,i} \\
&\geq \frac{99}{100} \langle M(w_t) - \text{Id}, U_t \rangle - \sum_{i \in S_b \setminus S'_b} w_i \tau_{t,i} \\
&\stackrel{(a)}{\geq} \frac{99}{100} \langle M(w_t) - \text{Id}, U_t \rangle - 1.21 \cdot |w_b| (O(\log 1/\varepsilon) + 8\varepsilon\gamma_1 + 8\varepsilon(\lambda_t + \gamma_2)) \\
&\stackrel{(b)}{\geq} \frac{49}{50} \langle M(w_t) - \text{Id}, U_t \rangle, \tag{25}
\end{aligned}$$

where (a) follows since $\tau_{t,i} \leq 1.1\tilde{\tau}_{t,i} \leq 1.1^2 \cdot \tau_{t,m}$ for all $i \in S_b \setminus S'_b$, and (b) follows from our assumption on $\langle M(w_t) - \text{Id}, U_t \rangle$.

Equations (24) and (25) together imply that our set of scores in this instance will be in the setting of Theorem 5.2. Thus Theorem 5.2 guarantees that the univariate filter outputs a set of weights w'_1, \dots, w'_m so that $\sum_{i=1}^m w'_i \tilde{\tau}_{t,i} \leq \frac{1}{5} \sum_{i=1}^m w_{t,i} \tilde{\tau}_{t,i}$. Notice that in particular this implies that $w_{t+1} \in \mathfrak{S}_{n,\varepsilon}$, which proves one of the claims in the Lemma. To complete the proof of the lemma, observe that

$$\begin{aligned}
\langle M(w_t) - \text{Id}, U_t \rangle - \langle F_t, U_t \rangle &= \sum_{i=1}^m (w_{t,i} - w'_i) (\tau_{t,i} - 1) \\
&\stackrel{(a)}{\geq} \sum_{i=1}^m (w_{t,i} - w'_i) \tau_{t,i} - 2\varepsilon \\
&\stackrel{(b)}{\geq} \frac{4}{1.21 \cdot 5} \sum_{i=1}^m w_{t,i} \tau_{t,i} - 2\varepsilon \\
&\stackrel{(c)}{\geq} \frac{4}{1.21 \cdot 5} \frac{49}{50} \langle M(w_t) - \text{Id}, U_t \rangle - 2\varepsilon \\
&\stackrel{(d)}{\geq} 0.63 \langle M(w_t) - \text{Id}, U_t \rangle,
\end{aligned}$$

where (a) follows since m is the 2ε -percentile, (b) follows from the guarantee of Theorem 5.2, (c) follows from (25), and (d) follows from our assumption on λ_t . Rearranging terms completes the proof. \square

We now show that this is enough to guarantee Lemma 7.10, which guarantees we make constant multiplicative progress in every epoch.

Proof of Lemma 7.10. Observe that if we terminate prematurely we clearly satisfy the lemma. Thus we may assume we do not terminate until timestep $T-1$. Lemma 7.9 then implies that no matter which update we do at time t for $t = 0, \dots, T-1$, we have the guarantee that

$$\langle F_t, U_t \rangle \leq 0.63 \langle F_{t-1}, U_t \rangle \leq 0.63 \langle F_0, U_t \rangle \leq 0.63 \|M(w_0) - \text{Id}\|_2. \tag{26}$$

Moreover, by Lemma 6.4, we have that $M(w_t) - \text{Id} \preceq M(w_0) - \text{Id}$, and hence $\frac{1}{\alpha} (M(w_t) - \text{Id}) \preceq I$ by our choice of α . Therefore by our regret bound, we have

$$\left\| \sum_{i=0}^{T-1} (M(w_t) - \text{Id}) \right\|_2 \leq \sum_{t=0}^{T-1} \langle U_t, M(w_t) - \text{Id} \rangle + \sum_{u=0}^{T-1} \langle U_t, |M(w_t) - \text{Id}| \rangle \frac{\|M(w_t) - \text{Id}\|_2}{\|M(w_0) - \text{Id}\|_2} + \log(n) \cdot \|M(w_0) - \text{Id}\|_2. \tag{27}$$

By Lemma 7.5, we know that for all $t = 0, \dots, T-1$, we must have

$$M(w_t) - \text{Id} \succeq -(\gamma_2 + 2\gamma_1^2 + O(\varepsilon \log 1/\varepsilon)) \text{Id}. \tag{28}$$

This will allow us to simplify a number of expressions. In particular, since $M(w_t) - \text{Id} \preceq M(w_0) - \text{Id}$, this implies that all positive eigenvalues of $M(w_t) - \text{Id}$ are smaller than λ_0 , and (28) implies that all negative eigenvalues are bounded in absolute value by λ_0 , by our assumption on λ_0 . Hence

$$\frac{\|M(w_t) - \text{Id}\|_2}{\|M(w_0) - \text{Id}\|_2} \leq 1. \quad (29)$$

Another implication is that

$$|M(w_t) - \text{Id}| \preceq M(w_t) - \text{Id} + 2(\gamma_2 + 2\gamma_1^2 + O(\varepsilon \log 1/\varepsilon)) \text{Id},$$

and hence

$$\langle U_t, |M(w_t) - \text{Id}| \rangle \leq \langle U_t, M(w_t) - \text{Id} \rangle + 2(\gamma_2 + 2\gamma_1^2 + O(\varepsilon \log 1/\varepsilon)). \quad (30)$$

Finally, we observe that (28) and Lemma 6.4 together imply that either

$$\|M(w_T) - \text{Id}\|_2 \leq \gamma_2 + 2\gamma_1^2 + O(\varepsilon \log 1/\varepsilon) \quad \text{or} \quad (31)$$

$$\left\| \sum_{i=0}^{T-1} (M(w_i) - \text{Id}) \right\|_2 \geq T \cdot \|M(w_{T-1}) - \text{Id}\|_2. \quad (32)$$

In the case of (31), we are clearly done, so we may assume that we are in the case of (32). Thus, plugging in (29), (30), and (32) into (27), and dividing by T , we obtain

$$\begin{aligned} \|M(w_{T-1}) - \text{Id}\|_2 &\leq \frac{2}{T} \sum_{i=0}^{T-1} (\langle U_t, M(w_i) - \text{Id} \rangle) + 2(\gamma_2 + 2\gamma_1^2 + O(\varepsilon \log 1/\varepsilon)) + \frac{\log n}{T} \|M(w_0) - \text{Id}\|_2 \\ &\stackrel{(a)}{\leq} \left(\frac{1}{2} + \frac{\log n}{T} \right) \|M(w_0) - \text{Id}\|_2 + 2 + 2(\gamma_2 + 2\gamma_1^2 + O(\varepsilon \log 1/\varepsilon)) \\ &\stackrel{(b)}{\leq} \frac{3}{4} \|M(w_0) - \text{Id}\|_2, \end{aligned}$$

where (a) follows from (26), and (b) follows since $T = \Theta(\log d)$, and since $\|M(w_0) - \text{Id}\|_2$ is a large constant factor larger than $\gamma_2 + 2\gamma_1^2 + O(\varepsilon \log 1/\varepsilon)$, by assumption. This completes the proof. \square

8 Fast approximate score oracles

In this section we describe how to implement the approximate score oracles and approximate augmented score oracles in nearly-linear time. Recall that an approximate score oracle takes as input a set of points $S \subset \mathbb{R}^d$ of size n , and a sequence of weights w_0, \dots, w_{t-1}, w_t , and computes $\tilde{\tau}_t \in \mathbb{R}^n$, where for all $i = 1, \dots, n$ we have $\tilde{\tau}_{t,i} \approx_{0.1} \tau_{t,i}$, where

$$\tau_{t,i} = (X_i - \mu(w_t))^\top U_t (X_i - \mu(w_t))^\top, \quad (33)$$

where

$$U_t = \exp \left(c \text{Id} - \alpha \sum_{i=0}^{t-1} M(w_i) \right) = \frac{\exp \left(-\alpha \sum_{i=0}^{t-1} M(w_i) \right)}{\text{tr} \exp \left(-\alpha \sum_{i=0}^{t-1} M(w_i) \right)},$$

and $\alpha > 0$ is a parameter. Additionally, recall that in Section 7 we additionally require that the score oracle be able to produce \tilde{q}_t so that

$$|\tilde{q}_t - q_t| \leq 0.1 \cdot q_t + 0.05 \cdot \|M(w_t) - \text{Id}\|_2, \text{ where } q_t = \langle M(w_t) - \text{Id}, U_t \rangle. \quad (34)$$

Our main result in this section is the following, which says that it is possible to achieve such approximations with high probability in nearly linear time:

Lemma 8.1. *Let $S = \{X_1, \dots, X_n\} \subseteq \mathbb{R}^d$, let $\delta > 0$, and let w_0, \dots, w_t be as above. Then there is an algorithm $\text{APPROXScores}(S, w_0, \dots, w_t, \delta)$ which outputs $\tilde{\tau}_{t,i}$ and \tilde{q}_t so that with probability $1 - \delta$, we have $\tilde{\tau}_{t,i} \approx_{0.1} \tau_{t,i}$ for all $i = 1, \dots, n$, where τ_t, \tilde{q}_t are defined in (33) and (34), respectively. Moreover, APPROXScores runs in time $\tilde{O}(tnd \log 1/\delta)$.*

If the output of APPROXScores satisfies the conditions of Lemma 8.1, we say that APPROXScores *succeeds*.

8.1 Tools from randomized numerical linear algebra

We need a few tools which are standard in the design of fast algorithms based on matrix multiplicative weights. The first is the standard Johnson-Lindenstrauss dimension reduction lemma:

Lemma 8.2 (Johnson-Lindenstrauss lemma [27]). *Let $\Phi \in \mathbb{R}^{r \times d}$ be a matrix whose entries are i.i.d. samples from $\mathcal{N}(0, 1/r)$. For every vector $u \in \mathbb{R}^d$ and every $\epsilon \in (0, 1)$,*

$$\Pr [(1 - \epsilon)\|u\|_2 \leq \|\Phi u\|_2 \leq (1 + \epsilon)\|u\|_2] \geq 1 - \exp(-\Omega(\epsilon^2 r)) .$$

We also require the following, slightly stronger version of the JL guarantee, which states that it preserves matrix inner products:

Lemma 8.3. *Let $A, U \in \mathbb{R}^{d \times d}$. Suppose $U = BB^\top$ for some symmetric B . Let $S \in \mathbb{R}^{r \times d}$ have i.i.d. entries from $\mathcal{N}(0, 1/r)$. There is a universal constant c such that for all $\epsilon > 0$,*

$$\Pr [|\langle A, BS^\top SB \rangle - \langle A, U \rangle| > \epsilon \|A\|_2 \cdot \text{tr}(U)] \leq 2 \exp(-cr \cdot \min(\epsilon, \epsilon^2)) .$$

Proof. Notice that $\mathbb{E} [\langle A, BS^\top SB \rangle] = \langle A, U \rangle$. By the Hanson-Wright inequality [28] together with standard arguments about averages of i.i.d. sub-exponential random variables, for every $t > 0$,

$$\Pr [|\langle A, BS^\top SB \rangle - \langle A, U \rangle| > t] \leq 2 \exp(-\ell \cdot \Omega(\min(t^2/\|BAB\|_F^2, t/\|BAB\|_2))) .$$

To finish the proof it will be enough to show that

$$\|BAB\|_F^2 \leq \|A\|_2^2 \cdot \text{tr}(U)^2 \quad \text{and} \quad \|BAB\|_2 \leq \|A\|_2 \cdot \text{tr}(U) .$$

For the first statement, note that

$$\|BAB\|_F^2 = \text{tr}(AU AU) = \langle AU A, U \rangle \leq \|AU A\|_2 \cdot \text{tr}(U) .$$

Since spectral norm is sub-multiplicative, $\|AU A\|_2 \leq \|A\|_2^2 \|U\|_2 \leq \|A\|_2^2 \cdot \text{tr}(U)$.

It remains to prove $\|BAB\|_2 \leq \|A\|_2 \cdot \text{tr}(U)$. Again we have

$$\|BAB\|_2 \leq \|B\|_2^2 \|A\|_2 = \|U\|_2 \|A\|_2 \leq \text{tr}(U) \cdot \|A\|_2 .$$

which finishes the proof. \square

We will also make use of Taylor series approximations to the matrix exponential function. The next lemma helps to control the errors incurred by such approximations.

Lemma 8.4 (folklore, see e.g. [29]). *For $\ell \in \mathbb{N}$, let $P_\ell(Y) = \sum_{j=0}^{\ell} \frac{1}{j!} (Y)^j$ be the degree- ℓ Taylor series approximation to $\exp(Y)$. For every $n \times n$ symmetric real matrix Y ,*

$$\|P_\ell(Y) - \exp(Y)\|_2 \leq \exp(\|Y\|_2) \cdot \exp(-\ell) .$$

8.2 Efficient approximate score oracles

With these tools in hand, we are now ready to describe APPROXScores. For some sufficiently large constant $C > 0$, let $\delta' = \delta/3$, let $r = C \log n/\delta'$, let $\ell = C \log d$, and let $S \in \mathbb{R}^{r \times d}$ be a matrix with i.i.d. entries from $\mathcal{N}(0, 1/r)$. The algorithm will form the $r \times d$ matrix

$$A_{r,\ell} = S \cdot P_\ell \left(-\frac{\alpha}{2} \sum_{t=0}^{t-1} M(w_t) \right). \quad (35)$$

The estimate for the scores will then be given by

$$\tilde{\tau}_{t,i} = \frac{1}{\text{tr}(A_{r,\ell} A_{r,\ell}^\top)} \|A_{r,\ell}(X_i - \mu(w_t))\|_2^2, \quad (36)$$

and the estimate for $q_{t,i}$ will be given by

$$\tilde{q}_{t,i} = \sum_{i=1}^n (\tilde{\tau}_{t,i} - 1). \quad (37)$$

The formal pseudocode is given in Algorithm 5. We first demonstrate that APPROXScores indeed runs in

Algorithm 5 Randomized nearly-linear time approximate score computation

- 1: **Input:** dataset $S \subset \mathbb{R}^d$ of size n , weight vectors w_0, \dots, w_t , failure probability $\delta > 0$
 - 2: Let $C > 0$ be a universal constant sufficiently large
 - 3: Let $\delta' = \delta/3$, let $r = C \log n/\delta'$, and let $\ell = C \log d$.
 - 4: Let $S \in \mathbb{R}^{r \times d}$ have entries drawn i.i.d. from $\mathcal{N}(0, 1/r)$.
 - 5: Compute $A_{r,\ell}$ as defined as in (35).
 - 6: For all $i = 1, \dots, n$, let $\tilde{\tau}_{t,i}$ be as in (36), and let \tilde{q}_t be as in (37).
 - 7: **return** $\tilde{\tau}_t, \tilde{q}_t$
-

the claimed runtime:

Lemma 8.5. $\text{APPROXScores}(S, w_0, \dots, w_t, \delta)$ runs in time $\tilde{O}(tnd \log 1/\delta)$.

Proof. The main algorithmic work being done in APPROXScores is to form the matrix $A_{r,\ell}$. We claim that this matrix can be formed in time $\tilde{O}(tnd \log 1/\delta)$. Afterwards, notice that $\text{tr}(A_{r,\ell} A_{r,\ell}^\top)$ can be computed in time $O(dr^2)$, and given that, each $\tilde{\tau}_{t,i}$ can be computed in time $O(nr)$. Given the $\tilde{\tau}_{t,i}$, we then observe that \tilde{q}_t can be computed in time $O(n)$. Since $r = O(\log n/\delta)$ and $\ell = O(\log d)$, we conclude that the overall runtime of APPROXScores is dominated by the time to form $A_{r,\ell}$, and so it runs in time $\tilde{O}(tnd \log 1/\delta)$.

We now demonstrate how to form $A_{r,\ell}$ efficiently. Observe that for any vector $v \in \mathbb{R}^d$, we can evaluate $v^\top \sum_{i=0}^{t-1} M(w_i)$ in time $O(tnd)$. By iterating this process, we can compute $v^\top \left(\sum_{i=0}^{t-1} M(w_i) \right)^j$ in time $O(jtnd)$, and thus we can compute $v^\top P_\ell \left(-\frac{\alpha}{2} \sum_{i=0}^{t-1} M(w_i) \right)$ in time $O(\ell^2 tnd)$. Since S has $r = O(\log n/\delta)$ rows, and $\ell = O(\log d)$, we can therefore form $A_{r,\ell}$ in time $\tilde{O}(tnd \log 1/\delta)$ by forming each row of $A_{r,\ell}$. \square

Note that APPROXScores is an approximate augmented score oracle, and so it is also clearly an approximate score oracle. In the remainder of this section, we show:

Lemma 8.6. *With probability $\geq 1 - \delta$, the output of APPROXScores satisfies $\tilde{\tau}_{t,i} \approx_{0.1} \tau_{t,i}$ for all $i = 1, \dots, n$, and $\tilde{q}_t \approx_{0.1} q_t$.*

Proof. Let $M = \sum_{i=0}^{t-1} M(w_i)$, and let $Y_t = \exp(M)$, so that $U_t = \frac{Y_t}{\text{tr}(Y_t)}$, and so that $A_{r,\ell} = S \cdot P_\ell(M)$. Then, by Lemma 8.4 and our choice of ℓ , we have $\|Y_t^{1/2} - P_\ell(M)\|_2 \leq \frac{0.01}{d} \cdot \|Y_t^{1/2}\|_2$, which immediately implies that $\|Y_t - P_\ell(M)^2\|_2 \leq \frac{0.03}{d} \cdot \|Y_t\|_2$. Notice that this implies that $\text{tr}(P_\ell(M)^2) \approx_{0.03} \text{tr}(Y_t)$.

We now condition on the event that the following three events hold simultaneously:

$$\|SP_\ell(M)(X_i - \mu(w_t))\|_2^2 \approx_{0.01} \|P_\ell(M)(X_i - \mu(w_t))\|_2^2 \text{ for all } i = 1, \dots, n, \quad (38)$$

$$\text{tr}(P_\ell(M)S^\top SP_\ell(M)) \approx_{0.01} \text{tr}(P_\ell(M)^2), \quad (39)$$

$$|\langle M(w_t) - \text{Id}, P_\ell(M)S^\top SP_\ell(M) \rangle - \langle M(w_t) - \text{Id}, P_\ell(M)^2 \rangle| \leq 0.01 \cdot \text{tr}(P_\ell(M)^2) \|M(w_t) - \text{Id}\|_2. \quad (40)$$

By our choice of δ' , Lemma 8.2, and a union bound, we know that (38) holds with probability at least $1 - \delta/3$. By instantiating Lemma 8.3 with $A = I$ and $A = M(w_t) - \text{Id}$ respectively, we also know that (39) and (40) each hold with probability at least $1 - \delta/3$. Thus, by a union bound, all three conditions hold simultaneously with probability at least $1 - \delta$. We claim that conditioned on these three events, the conditions of the lemma are satisfied. Indeed, we have

$$\begin{aligned} \tilde{\tau}_{t,i} &= \frac{1}{\text{tr}(P_\ell(M)S^\top SP_\ell(M))} \|S \cdot P_\ell(M)(X_i - \mu(w_t))\|_2^2 \\ &\approx_{0.0121} \frac{1}{\text{tr}(P_\ell(M)^2)} \|P_\ell(M)(X_i - \mu(w_t))\|_2^2 \\ &\approx_{0.0363} \frac{1}{\text{tr}(Y_t)} \|P_\ell(M)(X_i - \mu(w_t))\|_2^2 \\ &\approx_{0.01} \frac{1}{\text{tr}(Y_t)} (X_i - \mu(w_t))^\top Y_t (X_i - \mu(w_t)) \\ &= \tau_{t,i}. \end{aligned}$$

Here the second line follows from (38) and (39), the third line follows from our condition on the trace, and the final approximation follows since $P_\ell(M)$ approximates Y_t in spectral norm. This proves the claim about the $\tilde{\tau}_{t,i}$.

To conclude, we observe that

$$\begin{aligned} \tilde{q}_t &= \frac{1}{\text{tr}(P_\ell(M)S^\top SP_\ell(M))} \sum_{i=1}^n \left(\|S \cdot P_\ell(M)(X_i - \mu(w_t))\|_2^2 - \text{tr}(P_\ell(M)S^\top SP_\ell(M)) \right) \\ &= \frac{1}{\text{tr}(P_\ell(M)S^\top SP_\ell(M))} \langle M(w_t) - \text{Id}, P_\ell(M)S^\top SP_\ell(M) \rangle \\ &= \frac{1}{\text{tr}(P_\ell(M)S^\top SP_\ell(M))} (\langle M(w_t) - \text{Id}, P_\ell(M)^2 \rangle + \eta), \end{aligned}$$

where $|\eta| \leq 0.01 \cdot \text{tr}(P_\ell(M)) \cdot \|M(w_t) - \text{Id}\|_2$ by (40). We further have

$$\begin{aligned} \frac{1}{\text{tr}(P_\ell(M)S^\top SP_\ell(M))} \langle M(w_t) - \text{Id}, P_\ell(M)^2 \rangle &\approx_{0.01} \frac{1}{\text{tr}(P_\ell(M)^2)} \langle M(w_t) - \text{Id}, P_\ell(M)^2 \rangle \\ &\approx_{0.1} \langle M(w_t) - \text{Id}, U_t \rangle \\ &= q_t. \end{aligned}$$

Therefore

$$|\tilde{q}_t - q_t| \leq 0.1q_t + \frac{|\eta|}{\text{tr}(P_\ell(M)S^\top SP_\ell(M))} \leq 0.1 \cdot q_t + 0.05 \frac{|\eta|}{\text{tr}(Y_t)} \leq 0.1 \cdot q_t + 0.05 \|M(w_t) - \text{Id}\|_2,$$

which was the desired bound for \tilde{q}_t . □

Lemmata 8.5 and 8.6 together immediately imply Lemma 8.1.

9 Putting it all together

In this section, we formally combine the guarantees derived in the previous sections to prove Theorems 4.1 and 4.2.

9.1 Proof of Theorem 4.1

Given the machinery we've developed, the algorithm is straightforward to describe. Given a corrupted dataset S and $\delta > 0$, run $\text{NAIVEPRUNE}(S, \sqrt{4dn/\delta}, \delta/4)$ to obtain a pruned dataset S' . Center all points in S' with the empirical mean of S' . Then, run $\text{QUESCOREFILTER}(S', \text{APPROXScores})$, with $\kappa = \sqrt{4dn/\delta}$, and the δ parameter in APPROXScores set to $O(\delta/(\log \kappa \log d))$. The formal pseudocode is presented in Algorithm 6.

Algorithm 6 Nearly-linear time robust mean estimation under bounded second moments

- 1: **Input:** dataset $S \subset \mathbb{R}^d$ of size n , failure probability $\delta > 0$
 - 2: Let $S' = \text{NAIVEPRUNE}(S, \sqrt{4dn/\delta}, \delta/4)$.
 - 3: Let $\kappa = \sqrt{4dn/\delta}$.
 - 4: Center all points in S' at the empirical mean of S' .
 - 5: Let \mathcal{O} be APPROXScores with failure probability $O(\delta/(\log \kappa \log d))$.
 - 6: Let $\hat{\mu} = \text{QUESCOREFILTER}(S', \mathcal{O})$.
 - 7: **return** $\hat{\mu}$.
-

We now prove correctness.

Proof of Theorem 4.1. Recall that by definition, we may assume that $S = T \cup S_b \setminus S_r$, where T is a set of n i.i.d. samples from D , and $|S_b|, |S_r| \leq \varepsilon n$. Let γ_1, γ_2 be as in (9). We condition on four events:

- $\|X_i - \mu\|_2 \leq \sqrt{\frac{4dn}{\delta}}$ for all $i \in T$,
- $\text{NAIVEPRUNE}(S', \sqrt{4dn/\delta}, \delta/4)$ succeeds,
- $T = S_g \cup T_b$, where S_g is (γ_1, γ_2) -good with respect to D , and $|T_b| \leq \varepsilon n$, and
- every time it is called, APPROXScores succeeds.

By Chebyshev's inequality, and an union bound over all n points in T , the first bullet point holds with probability at least $\delta/4$. By a further union bound and by adjusting constants in our choices of δ , all four of these conditions hold simultaneously with probability at least $1 - \delta - \exp(-\varepsilon n)$. We now claim that, conditional on these four events, we output a $\mu(w)$ so that $\|\mu - \mu(w)\|_2 = O(\sqrt{\varepsilon}) + \tilde{O}(\sqrt{d/(n\delta)})$. Indeed, the first two conditions imply that NAIVEPRUNE does not throw away any points in T , and moreover, all points X in the set S' satisfy $\|X_i\|_2 \leq \sqrt{4dn/\delta}$ after centering. Thus, since the scores output by APPROXScores satisfy the necessary conditions for Theorem 6.5, it follows that the final output satisfies $\|\mu - \mu(w)\|_2 = O(\sqrt{\varepsilon}) + \tilde{O}(\sqrt{d/(n\delta)})$, as claimed.

We now turn to runtime. Since each epoch runs for at most $O(\log d)$ iterations, and so we run for at most $O(\log \kappa \log d)$ iterations, the total time spent running APPROXScores is at most $\tilde{O}(nd \log 1/\delta)$. Thus overall the algorithm runs in time $\tilde{O}(nd \log 1/\delta)$, as claimed. \square

9.2 Proof of Theorem 4.2

Again, the algorithm is straightforward. Given a corrupted dataset S , parameters $\varepsilon > 0$ and $\delta > 0$, run $\text{NAIVEPRUNE}(S, \sqrt{4d \log(n/\delta)}, \delta/4)$ to obtain a pruned dataset S' . Center all points in S' with the empirical mean of S' . Then, as above, run $\text{S.G.-QUESCOREFILTER}(S', \varepsilon, \text{APPROXScores})$, with $\kappa = \sqrt{4d \log(n/\delta)}$,

and the δ parameter in APPROXScores set to $O(\delta/(\log \kappa/\varepsilon \log d))$. The formal pseudocode is presented in Algorithm 7.

Algorithm 7 Nearly-linear time robust mean estimation for isotropic subgaussian distributions

- 1: **Input:** dataset $S \subset \mathbb{R}^d$ of size n , failure probability $\delta > 0$, fraction of error ε
 - 2: Let $S' = \text{NAIVEPRUNE}(S, \sqrt{4d \log(n/\delta)}, \delta/4)$.
 - 3: Let $\kappa = \sqrt{4d \log(n/\delta)}$.
 - 4: Center all points in S' at the empirical mean of S' .
 - 5: Let \mathcal{O} be APPROXScores with failure probability $O(\delta/(\log \kappa/\varepsilon \log d))$.
 - 6: Let $\hat{\mu} = \text{s.g.-QUEScoreFILTER}(S', \mathcal{O}, \varepsilon)$.
 - 7: **return** $\hat{\mu}$.
-

We now prove correctness. The proof is very similar to the proof presented above.

Proof of Theorem 4.2. Recall that by definition, we may assume that $S = S_g \cup S_b \setminus S_r$, where T is a set of n i.i.d. samples from D , and $|S_b|, |S_r| \leq \varepsilon n$. Let γ_1, γ_2 be as in (16). We condition on four events:

- $\|X_i - \mu\|_2 \leq \sqrt{4d \log(n/\delta)}$ for all $i \in S_g$,
- $\text{NAIVEPRUNE}(S', \sqrt{4d \log(n/\delta)}, \delta/4)$ succeeds,
- S_g is $(\varepsilon, \gamma_1, \gamma_2)$ -s.g. good with respect to D ,
- every time it is called, APPROXScores succeeds.

By standard concentration inequalities for chi-squared random variables, and an union bound over all n points in T , the first bullet point holds with probability at least $\delta/4$. By a further union bound and by adjusting constants in our choices of δ , all four of these conditions hold simultaneously with probability at least $1 - \delta$. We now claim that, conditional on these four events, we output a $\mu(w)$ so that $\|\mu - \mu(w)\|_2 = O(\gamma_1 + \varepsilon \sqrt{\log 1/\varepsilon} + \sqrt{\varepsilon(\gamma_1 + \gamma_2)})$. The first two conditions imply that NAIVEPRUNE does not throw away any points in S_g , and moreover, all points X in the set S' satisfy $\|X_i\|_2 \leq \sqrt{4d \log(n/\delta)}$ after centering. By standard arguments, this implies that $\|M(\frac{1}{n}\mathbf{1}_n) - \text{Id}\|_2 \leq O(\varepsilon \log 1/\varepsilon + \varepsilon \kappa)$, for $\kappa = \sqrt{d \log(n/\delta)}$. Thus, since the scores output by APPROXScores satisfy the necessary conditions for Theorem 6.5, it follows that the final output satisfies $\|\mu - \mu(w)\|_2 = O(\gamma_1 + \varepsilon \sqrt{\log 1/\varepsilon} + \sqrt{\varepsilon(\gamma_1 + \gamma_2)})$, as claimed.

We now turn to runtime. Since each epoch runs for at most $O(\log d)$ iterations, and so we run for at most $O(\log \kappa/\varepsilon \log d)$ iterations, the total time spent running APPROXScores is at most $\tilde{O}(nd \log 1/\delta \log 1/\varepsilon)$. Thus overall the algorithm runs in time $\tilde{O}(nd \log 1/\delta \log 1/\varepsilon)$, as claimed. \square

Part II: Outlier detection in high-dimensional data sets

10 QUE scoring versus local methods

The work [5] compares a number of outlier detection methods (mainly those based on k -NN distances) on several datasets, both low and high dimensional. We evaluate QUE scoring on the InternetAds dataset from [5], with a 0.1-fraction of outliers. Unlike the experiments on our CIFAR-10 and text embedding data, to replicate the experimental setting of prior work as closely as possible we perform no whitening or other preprocessing.

We find that QUE scoring is outperformed by LOF/ k -NN-based methods on the InternetAds dataset. Choosing $\alpha = 4$ for QUE, we find the ROCAUC scores in the below table.

To elucidate the difference between the InternetAds setting where k -NN methods perform well and the other experimental settings in this paper, we offer the following histograms demonstrating that the

method	ROCAUC
naive spectral	0.539
QUE	0.626
ℓ_2	0.723
isolation forest	0.702
local outlier factor	0.853
k -NN	0.855

Figure 2: ROCAUC scores on InternetAds dataset from [5]. Note that QUE still improves on naive spectral methods, but because outliers are individually identifiable in the InternetAds data set, local and ℓ_2 methods improve on any spectral method. We use $\alpha = 4$ in QUE. See [21, 22, 5] for definitions of local outlier factor and k -NN methods.

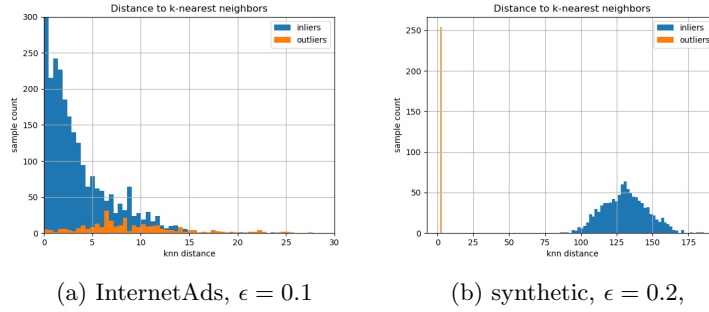


Figure 3: We plot histograms of the following collection $\{d_k(X_i, S) : X_i \in\}$ for $S = \text{InternetAds, synthetic}$, where $d_k(X_i, S)$ is the average squared- ℓ_2 distance of X_i and its k nearest neighbors, with $k = 10$. Observe that in the InternetAds dataset essentially only outliers have $d_k(X_i, S) > 15$, while for the synthetic dataset *all inliers* $d_k(X_i, S)$ is *much greater than that of every outlier*.

distribution of nearest-neighbor distances is markedly distinct for inliers and outliers in both data sets, but only in the InternetAds dataset do inliers have smaller k -NN distances.

11 Scaling up: a nearly-linear time implementation of QUE scoring

Most of the experiments we present involving QUE scores employ the following approach to compute them. Given $X_1, \dots, X_n \in \mathbb{R}^d$, explicitly form the empirical covariance $\bar{\Sigma}$ in memory. Use SciPy’s `expm` function to compute the matrix exponential $U = \exp(\alpha \bar{\Sigma})$, then compute $\tau_i = (X_i - \bar{\mu})^\top U (X_i - \bar{\mu})$. (This in turn uses the scaling and squaring algorithm for the matrix exponential of Al-Mohy and Higham [30].)

While we are already able to run experiments in 1000 or more dimensions using this approach, it requires at least d^2 memory to store the covariance, and somewhat more time to form and exponentiate it. We also implement an approximate method to compute QUE scores, whose running time is $\tilde{O}(nd)$. We demonstrate in this section that outlier detection from approximate QUE scores still improves over baseline methods on several data sets. The technique here is very similar to the one used in Section 8 to approximate the scores used in the fast robust mean estimation algorithm. At a high level, the idea is the same: approximate the exponential with a low-degree polynomial, and sketch this using Johnson-Lindenstrauss matrices. However, we make a couple of additional optimizations here.

QUE scores in nearly-linear time We use the following approximate method, inspired by our sketching approach to compute QUE scores from our nearly linear time algorithm for robust mean estimation.

Given X_1, \dots, X_n and $\alpha > 0$, our goal is to compute approximations $\tilde{\tau}_i$ to the QUE scores. At a high level, our approach employs two main tricks:

1. Approximate the matrix exponential $\exp(M)$ by Chebyshev polynomials of degree $O(\log d)$.
2. For $\bar{\Sigma}$ the empirical covariance of X_1, \dots, X_n , use fast versions of the Johnson-Lindenstrauss method to approximate $\langle X_i, \bar{\Sigma}^j X_i \rangle$ for all $i \leq n$ and $j \leq O(\log d)$.

Since for most applications only the *order* of the QUE scores matters, we will actually compute approximations to *non-normalized* scores, involving the matrix $U = \exp(\alpha \bar{\Sigma} / \|\bar{\Sigma}\|_2)$, rather than normalizing by $\text{tr}(U)$. For notational simplicity, let us assume $\bar{\mu} = 0$.

We first rewrite $\tau_i \propto \|\exp(\alpha \bar{\Sigma} / 2 \|\bar{\Sigma}\|_2) X_i\|_2^2$ where \propto hides the normalization $\text{tr} \exp(\alpha \bar{\Sigma})$. If we approximate the matrix exponential by a degree $O(\log d)$ Chebyshev approximation P , the goal is now to approximate $\|P(\alpha \bar{\Sigma} / 2 \|\bar{\Sigma}\|_2) X_i\|_2^2$ for all i . Suppose S is an $O(\log(d+n)) \times d$ sketching matrix. Then it will suffice to compute $M = SP(\alpha \bar{\Sigma} / 2 \|\bar{\Sigma}\|_2)$, since then in $\tilde{O}(nd)$ time we can compute all the matrix-vector products MX_i and their norms, and $\|SP(\alpha \bar{\Sigma} / 2 \|\bar{\Sigma}\|_2) X_i\|_2^2 \approx \|P(\alpha \bar{\Sigma} / 2 \|\bar{\Sigma}\|_2) X_i\|_2^2$.

Note that using the expansion into powers $\bar{\Sigma}^j$, right or left matrix-vector multiplication by $P(\alpha \bar{\Sigma} / 2 \|\bar{\Sigma}\|_2)$ can be accomplished in $O(nd \log d)$ time. As in the fast JL transform [31], we take $S = S' \cdot D \cdot H$ where S' is a sparse random matrix, D is a diagonal matrix with random ± 1 entries, and H is a Hadamard matrix. Fast fourier transform methods may be used to compute matrix-vector multiplications SX in $d(\log d)^{O(1)}$ time [32], leading to nearly-linear running time of this approach in theory. In practice, we use standard matrix multiplication; this still allows for experiments in thousands of dimensions.

Experiments show that the approximate QUE scores are consistent with the exact QUE scores, in that higher k leads to larger maximum improvement over naive spectral scores; furthermore approximate QUE consistently interpolates between ℓ_2 scores (when $\alpha = 0$) and naive spectral scores (when $\alpha = \infty$).

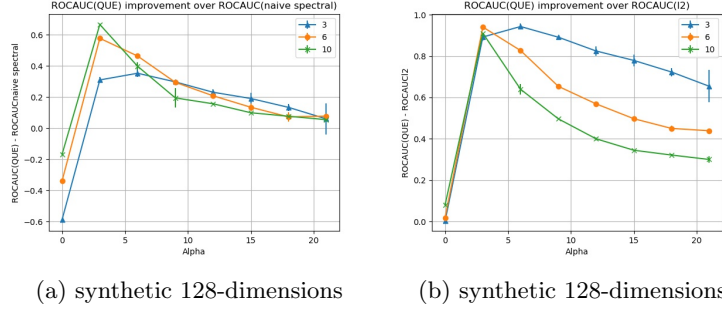


Figure 4: We plot the improvement of ROCAUC scores of our approximate QUE scoring implementation, over baseline methods ℓ_2 scoring as well as naive spectral scoring, on **128-dimensional** synthetic data, for 3, 6, and 10 directions of corruption. We use the degree-5 Chebyshev approximation to the matrix exponential function, along with scaling and squaring for approximating outside the interval $[-1, 1]$.

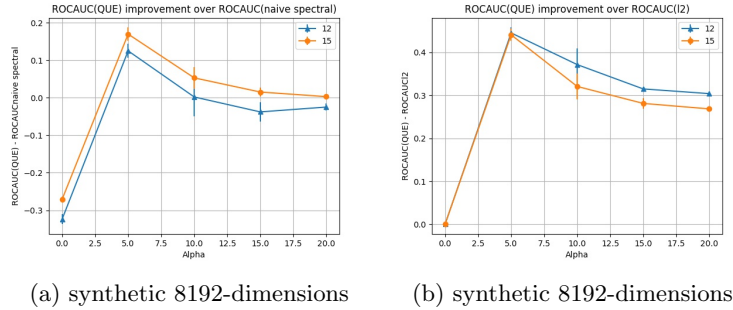


Figure 5: Similar to above, these show the improvement of ROCAUC scores of our approximate QUE scoring implementation, over baseline methods ℓ_2 scoring and naive spectral scoring, on **8192-dimensional** synthetic data, for 12 and 15 directions of corruption. These approximate QUE scores run in nearly-linear time and improve over the performance of naive spectral and ℓ_2 -based scoring.

References

- [1] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high-dimensions without the computational intractability. *SIAM Journal on Computing*, 48(2):742–864, 2019.
- [2] Kevin A Lai, Anup B Rao, and Santosh Vempala. Agnostic estimation of mean and covariance. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 665–674. IEEE, 2016.
- [3] Yu Cheng, Ilias Diakonikolas, and Rong Ge. High-dimensional robust mean estimation in nearly-linear time. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2755–2771. SIAM, 2019.
- [4] Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Being robust (in high dimensions) can be practical. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 999–1008. JMLR. org, 2017.
- [5] Guilherme O Campos, Arthur Zimek, Jörg Sander, Ricardo JGB Campello, Barbora Micenková, Erich Schubert, Ira Assent, and Michael E Houle. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, 30(4):891–927, 2016.

- [6] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.
- [7] Frank J Anscombe. Rejection of outliers. *Technometrics*, 2(2):123–146, 1960.
- [8] John W Tukey. A survey of sampling from contaminated distributions. *Contributions to probability and statistics*, pages 448–485, 1960.
- [9] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992.
- [10] John W Tukey. Mathematics and the picturing of data. In *Proceedings of the International Congress of Mathematicians, Vancouver, 1975*, volume 2, pages 523–531, 1975.
- [11] Jerry Zheng Li. *Principled approaches to robust machine learning and beyond*. PhD thesis, Massachusetts Institute of Technology, 2018.
- [12] Jacob Steinhardt. *Robust Learning: Information Theory and Algorithms*. PhD thesis, Stanford University, 2018.
- [13] Yu Cheng, Ilias Diakonikolas, Rong Ge, and David Woodruff. Faster algorithms for high-dimensional robust covariance estimation. In *Proceedings of the 32nd Annual Conference on Learning Theory (COLT 2019)*.
- [14] Douglas M Hawkins. *Identification of outliers*, volume 11. Springer, 1980.
- [15] Edwin M Knorr and Raymond T Ng. A unified notion of outliers: Properties and computation. In *KDD*, volume 97, pages 219–222, 1997.
- [16] Edwin M Knox and Raymond T Ng. Algorithms for mining distancebased outliers in large datasets. In *Proceedings of the international conference on very large data bases*, pages 392–403. Citeseer, 1998.
- [17] Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. Statistical query lower bounds for robust estimation of high-dimensional gaussians and gaussian mixtures. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 73–84. IEEE, 2017.
- [18] Zeyuan Allen-Zhu, Zhenyu Liao, and Lorenzo Orecchia. Spectral sparsification and regret minimization beyond matrix multiplicative updates. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 237–245. ACM, 2015.
- [19] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.
- [20] https://en.wikipedia.org/wiki/Wikipedia:Today's_featured_article/May_2019. May 2019. Wikimedia Foundation.
- [21] S Ramaswamy, R Rastogi, and K Shim. Efficient algorithms for mining outliers from large data sets. *Proceedings of the ACM international conference on management of data (SIGMOD)*, pages 427–438, 2000.
- [22] MM Breunig, HP Kriegel, R Ng, and J Sander. Efficient algorithms for mining outliers from large data sets. *Proceedings of the ACM international conference on management of data (SIGMOD)*, pages 93–104, 2000.
- [23] Peter J Rousseeuw and Katrien Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999.

- [24] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422. IEEE, 2008.
- [25] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.
- [26] Samuel B Hopkins and Jerry Li. How hard is robust mean estimation? *arXiv preprint arXiv:1903.07870*, 2019.
- [27] William B Johnson, Joram Lindenstrauss, and Gideon Schechtman. Extensions of lipschitz maps into banach spaces. *Israel Journal of Mathematics*, 54(2):129–138, 1986.
- [28] Mark Rudelson, Roman Vershynin, et al. Hanson-wright inequality and sub-gaussian concentration. *Electronic Communications in Probability*, 18, 2013.
- [29] David Steurer. Fast sdp algorithms for constraint satisfaction problems. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 684–697. Society for Industrial and Applied Mathematics, 2010.
- [30] Awad H Al-Mohy and Nicholas J Higham. A new scaling and squaring algorithm for the matrix exponential. *SIAM Journal on Matrix Analysis and Applications*, 31(3):970–989, 2009.
- [31] Nir Ailon and Bernard Chazelle. The fast johnson–lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on computing*, 39(1):302–322, 2009.
- [32] Alexandr Andoni, Piotr Indyk, Thijs Laarhoven, Ilya Razenshteyn, and Ludwig Schmidt. Practical and optimal lsh for angular distance. In *Advances in Neural Information Processing Systems*, pages 1225–1233, 2015.
- [33] Joel A Tropp et al. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

A Deferred details from Section 4

A.1 Proof of Lemma 5.1

The algorithm is straightforward: choose a random point in S , and check if strictly more than $n/2$ points lie within a ball of radius $2r$ around this point. If so, include all points with distance at most $4r$ from this point. If not, repeat, and run for $O(\log 1/\delta)$ iterations. We now prove correctness.

Proof of Lemma 5.1. By the triangle inequality, if we ever randomly select a point from S' , then we terminate, and in this case it is easy to see that the output satisfies the desired property. Thus, it is easy to see that the probability we have not terminated after t iterations is at most 2^{-t} . Suppose we have terminated. Then in that iteration, we selected a point $X \in S$ that has distance at most $2r$ to more than $n/2$ other points in S . This implies that it has distance at most $2r$ to some point in S' . By triangle inequality, this implies that all points in S' are at distance at most $4r$ from X , and so the output in this iteration must satisfy the claims of the Lemma. \square

We note that if one wishes to obtain a deterministic linear-time algorithm for this problem, it is also possible to do so, albeit using radius $O(r\sqrt{d})$. The algorithm is again simple: simply take the coordinate-wise median of all the data points, and take all points with distance at most $O(r\sqrt{d})$ from this point. It is not hard to see that the coordinate-wise median can differ in each coordinate from the points in S' by at most r , and so its distance to each point in S' can be at most $r\sqrt{d}$. While this is worse by a polynomial factor than the guarantee obtained above, since in the end our overall guarantees depend only logarithmically on r , this does not change our runtime guarantees by more than a logarithmic factor.

A.2 Omitted details from Section 5.3

The algorithm 1DFILTER is quite simple. For $i = 1, \dots, m$, and for any positive integer t , define

$$w_i^{(t)} = \left(1 - \frac{\tau_i}{\tau_{\max}}\right)^t w_i, \text{ and } F_t = \sum_{i=1}^n w_i^{(t)} \tau_i. \quad (41)$$

Observe that the F_t form a monotone decreasing sequence. The algorithm will simply find the smallest $t \in \{1, \dots, \frac{\tau_{\max}}{eb\sigma}\}$ so that $F_t \leq b\sigma$ via binary search, and outputs $w^{(t)}$. The formal pseudocode for the algorithm is given in Algorithm 8.

Algorithm 8 Improved univariate score downweighting

- 1: **Input:** nonnegative scores τ_1, \dots, τ_m , weights w_1, \dots, w_m , parameters b, η
 - 2: Let $\tau_{\max} = \max_{i \in [m]} \tau_i$.
 - 3: Let $\sigma = \sum_{i=1}^m w_i \tau_i$.
 - 4: By binary search, find the smallest $t \in \{1, \dots, \frac{\tau_{\max}}{eb\sigma}\}$ satisfying $F_t \leq b\sigma$, where F_t is defined in (41)
 - 5: **return** The weights $w_1^{(t)}, \dots, w_m^{(t)}$, where $w_i^{(t)}$ is defined as in (41).
-

We now prove that this algorithm satisfies Theorem 5.2. We say any set of weights w' satisfying $\sum_{i \in S_g} w_i - w'_i \leq \sum_{i \in S_b} w_i - w'_i$ is *admissible*. We first show that the sequence of weights we produce is always admissible, under some mild conditions:

Lemma A.1. *Let t be an integer so that $w^{(t)}$ is admissible, and $F_t > 2\eta\sigma$. Then $w^{(t+1)}$ is admissible.*

Proof of Lemma A.1. Because $w^{(t)} \leq w$, we have that $\sum_{i \in S_g} w_i^{(t)} \tau_i \leq \eta\sigma$. As a result, if $F_t \geq 2\eta\sigma$, we must have $\sum_{i \in S_b} w_i^{(t)} \tau_i > \sigma/2$. Therefore, we have the following two inequalities:

$$\begin{aligned} \sum_{i \in S_g} w_i^{(t)} - w_i^{(t+1)} &= \frac{1}{\tau_{\max}} \sum_{i \in S_g} w_i^{(t)} \tau_i \leq \frac{\sigma}{2\tau_{\max}} \\ \sum_{i \in S_b} w_i^{(t)} - w_i^{(t+1)} &= \frac{1}{\tau_{\max}} \sum_{i \in S_b} w_i^{(t)} \tau_i > \frac{\sigma}{2\tau_{\max}}. \end{aligned}$$

Consequently, we remove more mass from the weights in S_b than from S_g in going from $w^{(t)}$ to $w^{(t+1)}$. Since by assumption $w^{(t)}$ is admissible, this immediately implies that $w^{(t+1)}$ is admissible as well. \square

We are now ready to prove Theorem 5.2.

Proof of Theorem 5.2. By induction, Lemma A.1 guarantees that the output weights remain admissible, and the termination condition of the algorithm guarantees that the output satisfies (5). It suffices to bound the runtime of the algorithm.

First, observe that there must exist a valid T in the range we are searching. We first observe that

$$\begin{aligned} w_i^{(t)} \tau_i &= w_i \left(1 - \frac{\tau_i}{\tau_{\max}}\right)^t \tau_i \\ &\leq w_i \exp\left(-\frac{t \cdot \tau_i}{\tau_{\max}}\right) \tau_i. \end{aligned}$$

For any constant $A > 0$, the maximizer of the function $g(x) = x \exp(-Ax)$ in the range $x \in [0, \infty)$ is achieved by $x = \frac{1}{A}$, so $g(x) \leq \frac{1}{eA}$ for all $x \in [0, \infty)$. Setting $A = t/\tau_{\max}$, and letting $t = \frac{\tau_{\max}}{eb\sigma}$, we conclude that

$$\sum_{i=1}^m w_i^{(t)} \tau_i \leq \sum_{i=1}^m w_i \cdot \frac{\tau_{\max}}{et} \leq b\sigma.$$

Thus, there exists some t within our specified range which satisfies the conclusion. Finally, to bound the runtime, observe that every iteration runs in $O(n)$ time,² and we can run for at most $O(\log \tau_{\max}/(b\sigma))$ iterations, which completes the proof. \square

A.3 The randomized hard filter

In this section we show that a randomized outlier removal method, rather than soft downweighting, can achieve the more or less the same guarantees as 1DFILTER. Formally, we show:

Theorem A.2. *Let $\eta \in (0, 1/2)$, let $b \geq 2\eta$, and let $s, \delta > 0$. Let m satisfy*

$$m = \tilde{\Omega} \left(\frac{s \log^2(1/\delta) \log \frac{\tau_{\max}}{b\sigma}}{\varepsilon} \right).$$

Let τ_1, \dots, τ_m be non-negative scalars, and let $\tau_{\max} = \max_{i \in [m]} \tau_i$. Suppose there exist two disjoint sets S_g, S_b so that $S_g \cup S_b = [m]$, and moreover,

$$\sum_{i \in S_g} \tau_i \leq \eta \sigma, \text{ where } \sigma = \sum_{i=1}^n \tau_i.$$

Then $\text{RANDOMFILTER}(w, \tau)$ runs in time $O((1 + \log \frac{\tau_{\max}}{b\sigma})m)$ and outputs $S' \subseteq S$ so that with probability $1 - \delta$, we have:

- *not too many more points from S_g are removed than from S_b i.e.*

$$|(S \setminus S') \cap S_g| \leq |(S \setminus S') \cap S_b| + \frac{\varepsilon m}{s}, \text{ and}$$

- *the sum of the τ has decreased, i.e. S' satisfies*

$$\sum_{i \in S'} \tau_i \leq b\sigma. \tag{42}$$

The algorithm itself is very easy to describe. First, let $T = [m]$. Then, while $\sum_{i \in T} \tau_i > b\sigma$, throw away each point from T with probability $\tau_i/\tau_{\max}(T)$, where $\tau_{\max}(T) = \max_{i \in T} \tau_i$, and let T be the set of remaining points. At termination, we simply output the set $S' = T$. The formal pseudocode of this algorithm is given in Algorithm 9.

Algorithm 9 Improved randomized univariate filtering

- 1: **Input:** nonnegative scores τ_1, \dots, τ_m , parameters b, η
 - 2: Let $T = [m]$
 - 3: Let $\sigma = \sum_{i=1}^n \tau_i$.
 - 4: **while** $\sum_{i \in T} \tau_i \geq b\sigma$ **do**
 - 5: Let $\tau_{\max} = \max_{i \in T} \tau_i$
 - 6: Remove each point $i \in T$ from T with probability τ_i/τ_{\max}
 - 7: **end while**
 - 8: **return** The set T
-

As a brief aside, we note that this differs slightly from the algorithm presented in [4], as there the algorithm randomly selects a threshold, and throws away all points above this threshold. However, the key

²This is true in the real RAM model; in practice we can run any iteration in $O(m \log(\tau_{\max}/\sigma))$ time by exponentiation via repeated doubling to compute all the $w_i^{(t)}$, so we pay at most an additional log factor

property which the previous algorithm used of this random threshold was that for all $i \in T$, we had that $\Pr[i \text{ is thrown out}] = \tau_i / \tau_{\max}(T)$. Therefore a very similar analysis can be adapted for either case. However, the algorithm in [4] only succeeds with constant probability, and a martingale-style argument (as in [4]) is needed to ensure that it works.

The remainder of this section is dedicated to the proof of Theorem A.2. Our first lemma is similar to Lemma A.1.

Lemma A.3. *Suppose that $\sum_{i \in T} \tau_i > b\sigma$. Then, if we let T' be the random set obtained by throwing away each point from T with probability $\tau_i / \tau_{\max}(T)$, then:*

$$\mathbb{E}[|(T \setminus T') \cap S_g|] < \mathbb{E}[|(T \setminus T') \cap S_b|], \quad (43)$$

and moreover, for any $s > 0$, we have

$$\Pr\left[|(T \setminus T') \cap S_g| > |(T \setminus T') \cap S_b| \text{ and } |T \setminus T'| > \frac{\varepsilon m}{s}\right] \leq \exp\left(-\Omega\left(\frac{\varepsilon m}{s} \cdot \min((b - 2\eta)^2, 1)\right)\right). \quad (44)$$

Proof. For $i \in T$, let Y_i be the random variable which is 1 if we throw out i in T' and 0 otherwise. Then

$$\begin{aligned} \mathbb{E}[|(T \setminus T') \cap S_g|] &= \mathbb{E}\left[\sum_{i \in S_g \cap T} Y_i\right] = \sum_{i \in S_g \cap T} \frac{\tau_i}{\tau_{\max}(T)} \leq \sum_{i \in S_g} \frac{\tau_i}{\tau_{\max}(T)} \leq \frac{1}{\tau_{\max}(T)} \eta \sigma, \\ \mathbb{E}[|(T \setminus T') \cap S_b|] &= \mathbb{E}\left[\sum_{i \in S_b \cap T} Y_i\right] = \sum_{i \in S_b \cap T} \frac{\tau_i}{\tau_{\max}(T)} \geq \frac{1}{\tau_{\max}(T)} (\beta - \eta) \sigma, \end{aligned}$$

which proves (43), since $\beta > 2\eta$.

To prove (44), we break into two cases depending on σ / τ_{\max} . Suppose that $\sigma / \tau_{\max} \geq \varepsilon m / s$. Then by Bernstein's inequality, we have

$$\begin{aligned} \Pr\left[|(T \setminus T') \cap S_g| > |(T \setminus T') \cap S_b| \text{ and } |T \setminus T'| > \frac{2\varepsilon m}{s}\right] &\leq \Pr[|(T \setminus T') \cap S_g| > |(T \setminus T') \cap S_b|] \\ &= \Pr\left[\sum_{i \in S_g \cap T} Y_i - \sum_{i \in S_b \cap T} Y_i > 0\right] \\ &\leq \exp\left(-\Omega\left(\frac{\sigma}{\tau_{\max}} \cdot \min((b - 2\eta)^2, b - 2\eta)\right)\right) \\ &\leq \exp\left(-\Omega\left(\frac{\varepsilon m}{s} \cdot \min((b - 2\eta)^2, b - 2\eta)\right)\right) \end{aligned}$$

Otherwise, we observe that

$$\begin{aligned} \Pr\left[|(T \setminus T') \cap S_g| > |(T \setminus T') \cap S_b| \text{ and } |T \setminus T'| > \frac{2\varepsilon m}{s}\right] &\leq \Pr\left[|T \setminus T'| > \frac{\varepsilon m}{s}\right] \\ &= \Pr\left[\sum_{i \in T} Y_i > \frac{2\varepsilon m}{s}\right] \\ &\leq \exp\left(-\Omega\left(\frac{\varepsilon m}{s}\right)\right). \end{aligned}$$

Combining these two cases, and simplifying yields the desired claim. \square

We now show that with high probability, we do not need to repeat this procedure too many times before the sum decreases by a constant factor.

Lemma A.4. *Let $\delta > 0$, and let $t = \tilde{\Omega}(\log(\frac{\tau_{\max} m}{b\sigma}) \log(1/\delta))$. Then, the probability that Algorithm 9 runs for more than t iterations is at most δ .*

Proof. Let $J = \log \frac{\tau_{\max} m}{b\sigma}$. For $j = 1, \dots, J$, let

$$A_j = \{i \in T : \tau_i \in (\tau_{\max} \cdot 2^{-j}, \tau_{\max} \cdot 2^{-j-1}]\} .$$

For all $j = 1, \dots, J$, we claim that conditional on the event that the algorithm has not terminated yet, and all points from $A_{j'}$ have been removed, for $j' < j$, then after t' iterations, all points from A_j have been removed with probability at least $1 - m2^{-t'}$. Indeed, for all $i \in A_j$, in every iteration, if it has not been already removed, then it is removed with probability at least $1/2$. Thus after t' iterations, the probability that any point from A_j remains is at most $n2^{-t'}$. Therefore, by a union bound, after Jt' iterations, conditioned on the event that the algorithm hasn't terminated yet, the probability that any point from A_j for any j is at most $Jm2^{-t'}$. However, if all points from A_j are removed, for all j , then if T' is the remaining set, we have $\sum_{i \in T'} \tau_i \leq b\sigma$, so if all such points are removed, then the algorithm must either terminate or have already terminated. This proves the claim by setting $t' = \log(Jm/\delta)$. \square

Theorem A.2 follows from Lemma A.3 and Lemma A.4, by appropriately adjusting parameters.

The full algorithm using the randomized filter It is straightforward for the full matrix multiplicative weights algorithm to use the randomized filter rather than the downweighting-based method. Given an ε -corrupted dataset of size n initially, we do the following. At every instance where we pass to the filter, simply run the randomized filter instead of the downweighting-based method, and output the set of weights which is $1/n$ for every point that remains after running the randomized filter, and 0 otherwise.

The guarantee of the randomized filter is slightly weaker than the guarantee of the downweighting-based method, so we cannot use black-box use the analysis presented beforehand to also analyze the algorithm instantiated with the weights given by the randomized filter. This is because our guarantee allows for slightly more good points than bad points to be removed per run of the algorithm. However, since the matrix multiplicative weights routine runs for at most polylogarithmically many iterations, by setting $s = \text{poly log}(nd)$, we can guarantee that at the end of all of the runs, we have removed at most $2\varepsilon n$ data points from S_g . It is straightforward to verify that (up to a factor of 2), the same analysis for matrix multiplicative works with this slightly weaker guarantee, for ε sufficiently small. For conciseness, we omit the proof.

B Omitted Proofs from Section 6

B.1 Proof of Lemma 6.1

Before we prove this lemma, we require the following matrix Chernoff bound:

Fact B.1 (Theorem 5.1.1 in [33]). *Let $M_1, \dots, M_n \in \mathbb{R}^{d \times d}$ be a sequence of independent, random, PSD matrices. Assume that $\|M_i\|_2 \leq L$ for all $i = 1, \dots, n$, and suppose $\|\mathbb{E}[\sum_{i=1}^n M_i]\|_2 \leq n$. Then, there is some universal constant $c \leq 2 \log 2 - 1$ so that for all $t \geq 2$, we have*

$$\Pr \left[\left\| \sum_{i=1}^n M_i \right\|_2 \geq tn \right] \leq d \exp(-ctn/L) .$$

Proof. Observe that if $X \sim D$, then $\mathbb{E}[\|X - \mu\|_2^2] = \text{tr}(\Sigma) \leq d$. Let $c > 0$ be a constant to specify later. By Markov's inequality, we have

$$\Pr \left[\|X - \mu\|_2 \geq \sqrt{\frac{d}{c\varepsilon}} \right] \leq c\varepsilon . \quad (45)$$

Let E be the event $E = \left\{ X : \|X - \mu\|_2 < \sqrt{\frac{d}{c\varepsilon}} \right\}$, and let $S = \{X_i : X_i \in E\}$. We claim this set satisfies the properties claimed.

We first demonstrate that with probability $1 - \exp(-\varepsilon n)$, we have $|S| \geq (1 - \varepsilon)n$. Letting $Z_i = \mathbb{I}\{X_i \in E^c\}$, we have $|S| = n - \sum_{i=1}^n Z_i$, where $\mathbb{E}[Z_i] \leq c\varepsilon$, and hence, by Chernoff bounds,

$$\begin{aligned} \Pr[|S| < (1 - \varepsilon)n] &= \Pr\left[\sum Z_i > \varepsilon n\right] \\ &\leq \left(\frac{e^{1/c-1}}{(1/c)^{1/c}}\right)^{c\varepsilon n} \\ &= \exp(\varepsilon n - c\varepsilon n - \varepsilon n \log 1/c) . \end{aligned}$$

In particular, we let $c = e^{-2}$, by simplifying we obtain that $\Pr[|S| < (1 - \varepsilon)n] < \exp(-\varepsilon n)$. Let E_1 be the event that $|S| \geq (1 - \varepsilon)n$.

We now demonstrate that with probability $1 - \delta$, our set S is (γ_1, γ_2) -good with respect to D , where γ_1, γ_2 are given as in (9). We first prove concentration of the empirical mean. For $i = 1, \dots, n$, let $Y_i = (X_i - \mu) \cdot \mathbb{I}\{X_i \in E\}$. Since the X_i are independent, so too are the Y_i . Moreover, since multiplying by an indicator variable can only decrease variance, we have that $\text{Cov}[Y_i] \preceq I$. Letting $\mu' = \mathbb{E}[Y_i]$, we thus have

$$\mathbb{E}\left[\left\|\frac{1}{n} \sum_{i=1}^n Y_i - \mu'\right\|_2^2\right] \leq \frac{\text{tr}(\Sigma)}{n} \leq \frac{d}{n} .$$

Thus, if we let E_2 be the event

$$E_2 = \left\{ \left\|\frac{1}{n} \sum_{i=1}^n Y_i - \mu'\right\|_2 \leq \sqrt{\frac{2d}{n\delta}} \right\} ,$$

by Markov's inequality, we have $\Pr[E_2] \geq 1 - \delta/2$. We additionally have that for any unit vector v ,

$$\begin{aligned} |\langle v, \mu' \rangle| &= |\mathbb{E}[\langle v, X_i - \mu \rangle \cdot \mathbb{I}\{X_i \in E\}]| \\ &= |\mathbb{E}[\langle v, X_i - \mu \rangle \cdot \mathbb{I}\{X_i \in E^c\}]| \\ &\leq \mathbb{E}[\langle v, X_i - \mu \rangle^2]^{1/2} \Pr_{X \sim D}[X \in E^c]^{1/2} \\ &= (v^T \Sigma v)^{1/2} \cdot \Pr_{X \sim D}[X \in E^c]^{1/2} \leq \sqrt{c\varepsilon} , \end{aligned}$$

where the third line follows from Cauchy-Schwartz, the last line follows from (45), and since $\Sigma \preceq \text{Id}$. Taking a supremum over all unit vectors v yields that $\|\mu'\|_2 \leq \sqrt{c\varepsilon}$. Therefore, conditioned on both E_1 and E_2 , we have

$$\begin{aligned} \left\|\frac{1}{|S|} \sum_{i \in S} X_i - \mu\right\|_2 &= \frac{n}{|S|} \left\|\frac{1}{n} \sum_{i=1}^n Y_i\right\|_2 \\ &\leq \frac{n}{|S|} \left(\left\|\frac{1}{n} \sum_{i=1}^n Y_i - \mu'\right\|_2 + \|\mu'\|_2 \right) \\ &\leq \frac{n}{|S|} \cdot \left(\sqrt{\frac{2d}{n\delta}} + \sqrt{c\varepsilon} \right) \\ &\leq \frac{1}{1 - \varepsilon} \cdot \left(\sqrt{\frac{2d}{n\delta}} + \sqrt{c\varepsilon} \right) = \gamma_1 . \end{aligned} \tag{46}$$

We now turn our attention to the claimed bound on the covariance. Let Y_i be as above. Then, by assumption we have $\|Y_i Y_i^\top\|_2 = \|Y_i\|_2^2 \leq \frac{d}{c\varepsilon}$, and moreover we have

$$\mathbb{E} \left[\sum_{i=1}^n Y_i Y_i^\top \right] \preceq n \text{Cov}[Y_i] \leq n \cdot \text{Id} .$$

Hence, by Fact B.1, we have that

$$\Pr \left[\left\| \frac{1}{n} \sum_{i=1}^n Y_i Y_i^\top \right\|_2 > \frac{d(\log d + \log 2/\delta)}{c'\varepsilon n} \right] \leq \delta/2 , \quad (47)$$

for some universal constant $c' > 0$. Let E_3 be the event that (47) holds. Then, conditioned on both E_1 and E_3 holding, we have

$$\begin{aligned} \|\text{Cov}(S)\|_2 &= \left\| \frac{1}{|S|} \sum_{i=1}^n (Y_i - \mu(S))(Y_i - \mu(S))^\top \right\|_2 \\ &\stackrel{(a)}{\leq} \left\| \frac{1}{|S|} \sum_{i=1}^n Y_i Y_i^\top \right\|_2 \\ &\leq \frac{1}{1-\varepsilon} \cdot \frac{d(\log d + \log 2/\delta)}{c'\varepsilon n} = \gamma_2 , \end{aligned} \quad (48)$$

as claimed, where (a) follows since centering the second moment matrix can only decrease its top eigenvalue. Thus, (46) and (48) imply that, conditioned on events E_1, E_2, E_3 simultaneously, the set S is (γ_1, γ_2) -good with respect to D . By a union bound, these three events happen simultaneously with probability at least $1 - \delta - \exp(-\varepsilon n)$, as claimed. \square