Learning from Untrusted Data

Moses Charikar
Stanford University
Stanford, CA 94305, USA
moses@cs.stanford.edu

Jacob Steinhardt
Stanford University
Stanford, CA 94305, USA
jsteinhardt@cs.stanford.edu

Gregory Valiant
Stanford University
Stanford, CA, USA 94305, USA
valiant@stanford.edu

ABSTRACT

The vast majority of theoretical results in machine learning and statistics assume that the training data is a reliable reflection of the phenomena to be learned. Similarly, most learning techniques used in practice are brittle to the presence of large amounts of biased or malicious data. Motivated by this, we consider two frameworks for studying estimation, learning, and optimization in the presence of significant fractions of arbitrary data.

The first framework, *list-decodable learning*, asks whether it is possible to return a list of answers such that at least one is accurate. For example, given a dataset of n points for which an unknown subset of αn points are drawn from a distribution of interest, and no assumptions are made about the remaining $(1-\alpha)n$ points, is it possible to return a list of $\mathrm{poly}(1/\alpha)$ answers? The second framework, which we term the *semi-verified* model, asks whether a small dataset of trusted data (drawn from the distribution in question) can be used to extract accurate information from a much larger but untrusted dataset (of which only an α-fraction is drawn from the distribution).

We show strong positive results in both settings, and provide an algorithm for robust learning in a very general stochastic optimization setting. This result has immediate implications for robustly estimating the mean of distributions with bounded second moments, robustly learning mixtures of such distributions, and robustly finding planted partitions in random graphs in which significant portions of the graph have been perturbed by an adversary.

CCS CONCEPTS

• **Security and privacy** → **Formal security models**; *Artificial immune systems*; • **Theory of computation** → *Sample complexity and generalization bounds*; **Models of learning**; Random projections and metric embeddings; Random network models;

KEYWORDS

outlier removal, robust learning, high-dimensional statistics

ACM Reference format:
Moses Charikar, Jacob Steinhardt, and Gregory Valiant. 2017. Learning from Untrusted Data. In *Proceedings of 49th Annual ACM SIGACT Symposium on the Theory of Computing, Montreal, Canada, June 2017 (STOC'17),* 14 pages.
DOI: 10.1145/3055399.3055491

STOC'17, Montreal, Canada

DOI: 10.1145/3055399.3055491

1 INTRODUCTION

What can be learned from data that is only partially trusted? In this paper, we study this question by considering the following setting: we observe n data points, of which αn are drawn independently from a distribution of interest, p^*, and we make no assumptions about the remaining $(1-\alpha)n$ points—they could be very biased, arbitrary, or chosen by an adversary who is trying to obscure p^*. Our goal is to accurately recover a parameter of interest of p^* (such as the mean), despite the presence of significant amounts of untrusted data. Perhaps surprisingly, we will show that in high dimensions, accurate estimation and learning is often possible, even when the fraction of real data is small (i.e., $\alpha \ll 1$). To do this, we consider two notions of successful learning–the *list decodable* model and the *semi-verified* model–and provide strong positive results for both notions. Our results have implications in a variety of domains, including building secure machine learning systems, performing robust statistics in the presence of outliers, and agnostically learning mixture models.

The goal of accurate robust estimation appears at first glance to be impossible if the fraction α of real data is less than one half. Indeed, if $\alpha = \frac{1}{2}$, it is possible that the real and fake data are distributed identically, except that the mean of the fake data is shifted by some large amount; in such a case, it is clearly impossible to differentiate which of these two distributions is "right". Perhaps, however, such symmetries are the *only* real problem that can occur. It might then be possible to output a short list of possible parameter sets–if $\alpha = \frac{1}{2}$, perhaps a list of two parameter sets–such that at least one is accurate. To this end, we consider a notion of successful learning called *list decodable learning*, first introduced by Balcan et al. (2008). In analogy with list decodable coding theory, the goal is for the learning algorithm to output a short list of possible hypotheses.

Definition 1.1 (List Decodable Learning). We say that a learning, estimation, or optimization problem is (m, ϵ) *list decodably solvable* if an efficient algorithm can output a set of at most m hypotheses/estimates/answers, with the guarantee that at least one is accurate to within error ϵ.

A central question in this paper concerns which learning problems can be robustly solved in the above sense:

> *To what extent are learning problems robustly solvable in the list decodable sense? If the dataset consists of only an α-fraction of real data, in what settings is it possible to efficiently output a list of at most $\frac{1}{\alpha}$ or $\mathrm{poly}(\frac{1}{\alpha})$ parameter sets or estimates with the guarantee that at least one closely approximates the solution that could be obtained if one were given only honest data?*

The intuition for why strong positive results are obtainable in the list decodable setting is the following. Given a dataset with an α

fraction of trusted data, the remaining data might do one of two things: either it can be fairly similar to the good data, in which case it can bias the overall answers by only a small amount, or the adversarial data may be very different from the trusted data. The key is that if a portion of the untrusted data tries too hard to bias the final result, then it will end up looking quite different, and can be clustered out.

Our investigation of robust learning has three motivations. First, from a theoretical perspective, it is natural to ask what guarantees are possible in the setting in which a majority of data is untrusted ($\alpha < \frac{1}{2}$). Is it the case that learning really becomes impossible (as is often stated), or can one at least narrow down the possible answers to a small set? Second, in many practical settings, there is a trade-off between the amount of data one can collect, and the quality of the data. For a fixed price, one might be able to collect either a small and accurate/trusted dataset, or a large but less trusted dataset. It is worth understanding how the quality of models derived from such datasets varies, across this entire range of dataset quality/quantity. Finally, robust learning with $\alpha \ll 1$ provides a new perspective on learning mixtures of distributions—by treating a single mixture component as the real data, and the remaining components as fake data, we can ask to what extent a mixture component can be learned, independently of the structure of the other components. While this perspective may seem to give up too much, we will show, somewhat surprisingly, that it is possible to learn mixtures almost as well under these adversarial assumptions as under stochastic assumptions.

Semi-Verified Learning. When $\alpha \leq \frac{1}{2}$, the list decodable model handles symmetries by allowing the learner to output multiple possible answers; an alternative is to break these symmetries with a small amount of side information. In particular, in many practical settings it is possible to obtain a (sometimes extremely small) verified set of data that has been carefully checked, which could be used to determine which of multiple alternative answers is correct. This motivates us to introduce the following new notion of learnability:

Definition 1.2 (The Semi-Verified Model). In the *semi-verified model*, we observe n data points, of which an unknown αn are "real" data reflecting an underlying distribution p^*, and the remaining $(1-\alpha)n$ points are arbitrary. Furthermore, we observe k "verified" data points that are guaranteed to be drawn from p^*.

The definition of the semi-verified model is inspired by the *semi-supervised* model of learning (see e.g. Chapelle et al. (2006)). In semi-supervised learning, one is concerned with a prediction/labeling task, and has access to a large amount of unlabeled data together with a small amount of labeled data; the central question is whether the presence of the unlabeled data can reduce the amount of labeled data required to learn. Analogously, in our robust learning setting, we are asking whether the presence of a large amount of untrusted data can reduce the amount of trusted data required for learning. Clearly the answer is "no" if we make no assumptions on the untrusted data. Nevertheless, the assumption that a significant fraction of that data is drawn from p^* seems plausible, and may be sufficient to achieve strong positive results. We therefore ask:

> *To what extent can the availability of a modest amount of "verified" data facilitate (either computationally or information theoretically) the extraction of the information contained in a larger but untrusted dataset? What learning tasks can be performed in the above semi-verified setting given $k \ll n$ verified data points? How does the amount k of verified data that is needed vary with the setting, the fraction α of honest data, etc.?*

The above definition and associated questions reflect challenges faced in a number of practical settings, particularly those involving large crowdsourced datasets, or datasets obtained from unreliable sensors or devices. In such settings, despite the unreliability of the data, it is often possible to obtain a small verified dataset that has been carefully checked. Given its pervasiveness, it is somewhat surprising that neither the theory nor the machine learning communities have formalized this model, and we think it is important to develop an understanding of the algorithmic possibilities in this domain. Obtaining theoretical guarantees in this setting seems especially important for designing *provably secure* learning systems that are guaranteed to perform well even if an adversary obtains control over some of the training data used by the algorithm.

Relationships between the models. The semi-verified and list decodable models can be reduced to each other. Informally, given m candidate outputs from a list decodable algorithm, we expect to be able to distinguish between them with $O(\log(m))$ verified data points. Conversely, if a model is learnable with k verified points then we can output $O((1/\alpha)^k)$ candidate parameters in the list decodable setting (since if we sample that many k-tuples from the untrusted data, at least one is likely to contain only honest data). For simplicity we state most results in the list decodable model.

Our contributions. We provide results on robust learnability in a general stochastic optimization setting, where we observe convex functions $f_1, \ldots, f_n$ of which αn are sampled from p^*, and we want to minimize the population mean $\bar{f} = \mathbb{E}_{p^*}[f]$.[1] Our results are given in terms of a spectral norm bound on the gradients ∇f_i. Therefore, we obtain robustness in any setting where we can establish a matrix concentration inequality on the good data — for instance, if the ∇f_i are sub-Gaussian and Lipschitz, or sometimes even with only bounded second moments.

From our general results (discussed in detail in the next section), we immediately obtain corollaries in specific settings, starting with mean estimation:

- **Robust mean estimation:** When $\alpha > \frac{1}{2}$ we can robustly estimate the mean of a distribution p^* to ℓ_2 error $O(\sigma)$, where σ^2 is a bound on the second moments of p^*. For α bounded away from 1, this improves upon existing work, which achieves error either $O(\sigma\sqrt{\log(d)})$ under a 4th moment bound on p^* (Lai et al., 2016) or matches our rate of $O(\sigma)$ but assumes p^* is sub-Gaussian (Diakonikolas et al., 2016). For $\alpha \leq \frac{1}{2}$, which was previously unexplored, we can estimate the mean to error $\tilde{O}(\sigma/\sqrt{\alpha})$.

[1] Typically, we observe data points x_i, and f_i is the loss function corresponding to x_i.

Since our results hold for any stochastic optimization problem, we can also study density estimation, by taking f_i to be the negative log-likelihood:

- **Robust density estimation:** Given an exponential family $p_\theta(x) \propto \exp(\theta^\top \phi(x))$, we can output θ with $KL(p_{\theta^*} \| p_\theta) \leq O(\frac{\sigma r}{\sqrt{\alpha}})$, where $\sigma = \lambda_{\max}(\mathrm{Cov}_{p^*}[\phi(x)])$ and $r = \|\theta^* \|_2$.

While density estimation could be reduced to mean estimation (via estimating the sufficient statistics), our analysis applies directly, to an algorithm that can be interpreted as approximately maximizing the log likelihood while removing outliers.

In the list decodable setting, our results also yield bounds for learning mixtures:

- **Learning mixture models:** Given a mixture of k distributions each with covariance bounded by σ^2, and with minimum mixture weight α, we can accurately cluster the points if the means are separated by a distance $\tilde{\Omega}(\sigma/\sqrt{\alpha})$, even in the presence of additional adversarial data. For comparison, even with few/no bad data points, the best efficient clustering algorithms require mean separation $\tilde{\Omega}(\sigma\sqrt{k})$ (Achlioptas and McSherry, 2005; Awasthi and Sheffet, 2012), which our rate matches if $\alpha = \Omega(\frac{1}{k})$.
- **Planted partition models:** In the planted partition model, we can approximately recover the planted partition if the average degree is $\tilde{\Omega}(1/\alpha^3)$, where αn is the size of the smallest piece of the partition. The best computationally efficient result (which assumes all the data is real) requires the degree to be $\Omega(1/\alpha^2)$ (Abbe and Sandon, 2015a;b).

It is fairly surprising that, despite making no assumptions on the structure of the data outside of a mixture component, we nearly match the best computationally efficient results that fully leverage this structure. This suggests that there may be a connection between robustness and computation: perhaps the *computational threshold* for recovering a planted structure in random data (such as a geometric cluster or a high-density subgraph) matches the *robustness threshold* for recovering that structure in the presence of an adversary.

Technical highlights. Beyond our main results, we develop certain technical machinery that may be of broader interest. Perhaps the most relevant is a novel matrix concentration inequality, based on ideas from spectral graph sparsification (Batson et al., 2012), which holds assuming only bounded second moments:

Proposition 1.3. *Suppose that p is a distribution on $\mathbb{R}^d$ with $\mathbb{E}_p[X] = \mu$ and $\mathrm{Cov}_p[X] \leq \sigma^2 I$ for some σ. Then, given $n \geq d$ samples from p, with probability $1 - \exp(-\frac{n}{64})$ there is a subset $I \subseteq [n]$ of size at least $\frac{n}{2}$ such that $\lambda_{\max}\left(\frac{1}{|I|}\sum_{i\in I}(x_i - \mu)(x_i - \mu)^\top\right) \leq 24\sigma^2$, where $\lambda_{\max}$ denotes the maximum eigenvalue.*

This result is strong in the following sense: if one instead uses all n samples x_i, the classical result of Rudelson (1999) only bounds $\lambda_{\max}$ by $\approx \sigma^2 \log(n)$, and even then only in expectation. Even under stronger assumptions, one often either needs at least $d\log(d)$ samples or incurs a $\log(d)$ factor in the bound on $\lambda_{\max}$. In the planted partition model, this log factor causes natural spectral approaches to fail on sparse graphs, and avoiding the log factor has been a topic of recent interest (Guédon and Vershynin, 2014; Le et al., 2015; Rebrova and Tikhomirov, 2015; Rebrova and Vershynin, 2016).

Proposition 1.3 says that the undesirable log factor only arises due to a manageable fraction of bad samples, which when removed give us sharper concentration. Our framework allows us to exploit this by defining the good data to be the (unknown) set I for which Proposition 1.3 holds. One consequence is that we are able to recover planted partitions in sparse graphs essentially "for free".

Separately, we introduce a novel regularizer based on minimum trace ellipsoids. This regularizer allows us to control the spectral properties of the model parameters at multiple scales simultaneously, and yields tighter bounds than standard trace norm regularization. We define the regularizer in Section 3, and prove a *local Hölder's inequality* (Lemma 5.1), which yields concentration bounds solely from deterministic spectral information.

We also employ *padded decompositions*, a space partitioning technique from the metric embedding literature (Fakcharoenphol et al., 2003). Their use is the following: when the loss functions are strongly convex, we can improve our bounds by identifying clusters in the data, and re-running our main algorithm on each cluster. Padded decompositions help us because they can identify clusters even if the remaining data has arbitrary structure. Our clustering scheme is described in Section 6.

Related work. The work closest to ours is Lai et al. (2016) and Diakonikolas et al. (2016), who study high-dimensional estimation in the presence of adversarial corruptions. They focus on the regime $\alpha \approx 1$, while our work focuses on $\alpha \ll 1$. In the overlap of these regimes (e.g. $\alpha = \frac{3}{4}$) our results improve upon these existing results. (The existing bounds are better as $\alpha \to 1$, but do not hold at all if $\alpha \leq \frac{1}{2}$.) The popular robust PCA algorithm (Candès et al., 2011; Chandrasekaran et al., 2011) allows for a constant fraction of the *entries* to be arbitrarily corrupted, but assumes the locations of these entries are sufficiently evenly distributed. However, Xu et al. (2010) give a version of PCA that is robust to arbitrary adversaries if $\alpha > \frac{1}{2}$. Bhatia et al. (2015) study linear regression in the presence of adversaries, and obtain bounds for sufficiently large α (say $\alpha \geq \frac{64}{65}$) when the design matrix is *subset strong convex*. Klivans et al. (2009) and Awasthi et al. (2014) provide strong bounds for robust classification in high dimensions for isotropic log-concave distributions.

The only works we are aware of that achieve general adversarial guarantees when $\alpha \leq \frac{1}{2}$ are Hardt and Moitra (2013), who study robust subspace recovery in the presence of a large fraction of outliers, and Steinhardt et al. (2016), which is an early version of this work that focuses on community detection.

Balcan et al. (2008) introduce the list-decodable learning model, which was later studied by others, e.g. Balcan et al. (2009) and Kushagra et al. (2016). That work provides bounds for clustering in the presence of some adversarial data, but has two limitations relative to our results (apart from being in a somewhat different setting): the fraction of adversaries tolerated is small ($O(\frac{1}{k})$), and the bounds are weak in high dimensions; e.g. Balcan et al. (2008) output a list of $k^{O(k/\gamma^2)}$ hypotheses, where γ can scale as $1/\sqrt{d}$.

Kumar and Kannan (2010) and the follow-up work of Awasthi and Sheffet (2012) find deterministic conditions under which efficient k-means clustering is possible, even in high dimensions. While the

goal is different from ours, there is some overlap in techniques. They also obtain bounds in the presence of adversaries, but only if the fraction of adversaries is smaller than $\frac{1}{k}$. Our corollaries for learning mixtures can be thought of as extending this line of work, by providing deterministic conditions under which clustering is possible even in the presence of a large fraction of adversarial data.

Separately, there has been considerable interest in *semi-random graph models* (Agarwal et al., 2015; Blum and Spencer, 1995; Chen et al., 2014b; Coja-Oghlan, 2004; 2007; Feige and Kilian, 2001; Feige and Krauthgamer, 2000; Guédon and Vershynin, 2014; Krivelevich and Vilenchik, 2006; Makarychev et al., 2012; Moitra et al., 2015) and *robust community detection* (Cai and Li, 2015; Kumar and Kannan, 2010; Makarychev et al., 2015; Moitra et al., 2015). In these models, a random graph is generated with a planted structure (such as a planted clique or partition) and adversaries are then allowed to modify some parts of this structure. Typically, the adversary is constrained to only modify $o(n)$ nodes or to only modify the graph in restricted ways, though some of the above work considers substantially stronger adversaries as well.

Robust learning is interesting from not just an information-theoretic but also a computational perspective. Guruswami and Raghavendra (2009) and Feldman et al. (2009) show that learning half-spaces is NP-hard for any $\alpha < 1$, while Hardt and Moitra (2013) show that learning k-dimensional subspaces in $\mathbb{R}^d$ is hard if $\alpha < \frac{k}{d}$. More generally, algorithms for list decodable learning imply algorithms for learning mixture models, e.g. planted partitions or mixtures of sub-Gaussian distributions, which is thought to be computationally hard in at least some regimes.

Finally, there is a large literature on learning with errors, spanning multiple communities including learning theory (Kearns and Li, 1993) and statistics (Tukey, 1960). We refer the reader to Huber and Ronchetti (2009) and Hampel et al. (2011) for recent surveys.

Comparison of techniques. We next explain how our techniques relate to those in recent robust learning work by Diakonikolas et al. (2016) and Lai et al. (2016). At a high level, our algorithm works by solving a convex optimization problem whose objective value will be low if all the data come from p^*; then, if the objective is high, by looking at the dual we can identify which points are responsible for the high objective value and remove them as outliers.

In contrast, Diakonikolas et al. (2016) solve a convex *feasibility* problem, where the feasible set depends on the true distribution p^* and hence is not observable. Nevertheless, they show that given a point that is far from feasible, it is possible to provide a separating hyperplane demonstrating infeasibility. Roughly speaking, then, we solve a "tainted" optimization problem and clean up errors after the fact, while they solve a "clean" (but unobserved) optimization problem and show that it is possible to make progress if one is far from the optimum. The construction of the separation oracle in Diakonikolas et al. (2016) is similar to the outlier removal step we present here, and it would be interesting to further understand the relationship between these approaches.

Diakonikolas et al. (2016) also propose another algorithm based on *filtering*. In the case of mean estimation, the basic idea is to compute the maximum eigenvector of the empirical covariance of the data — if this eigenvector is too large, then we can find a collection of points that are responsible for it being large, and remove them as outliers. Though it is not phrased this way, it can be thought of–similarly to our approach–as solving a tainted optimization problem (top eigenvalue on the noisy data) and then cleaning up outliers afterwards. Their outlier removal step seems tighter than ours, and it would be interesting to find an approach that obtains such tight bounds for a general class of optimization problems.

Finally, Lai et al. (2016) pursue an approach based on iteratively finding the top $n/2$ eigenvectors (rather than just the top) and projecting out the remaining directions of variation, as well as removing outliers if the eigenvalues are too large. This seems similar in spirit to the filtering approach described above.

Outline. Our paper is organized as follows. In Section 2 we present our main results and some of their implications in specific settings. In Section 3 we explain our algorithm and provide some intuition for why it should work. In Section 4 we provide a proof outline for our main results. In Sections 5 and 6, we sharpen our results, first showing how to obtain concentration inequalities on the errors, and then showing how to obtain tighter bounds and stronger guarantees for strongly convex losses. In Section 7 we present lower bounds showing that our results are optimal in some settings. Finally, in Section 8 we present some intuition for our bounds. Detailed proofs are deferred to the full version of the paper.

Acknowledgments. We thank the anonymous reviewers who made many helpful comments that improved this paper. MC was supported by NSF grants CCF-1565581, CCF-1617577, CCF-1302518 and a Simons Investigator Award. JS was supported by a Fannie & John Hertz Foundation Fellowship, a NSF Graduate Research Fellowship, and a Future of Life Institute grant. GV was supported by NSF CAREER award CCF-1351108, a Sloan Foundation Research Fellowship, and a research grant from the Okawa Foundation.

2 MAIN RESULTS AND IMPLICATIONS

We consider a general setting of stochastic optimization with adversaries. We observe convex functions $f_1, \ldots, f_n : \mathcal{H} \to \mathbb{R}$, where $\mathcal{H} \subseteq \mathbb{R}^d$ is a convex parameter space. For a "good" subset $I_g \subseteq [n]$ of size αn, $f_i \overset{i.i.d.}{\sim} p^*$ for $i \in I_g$, and the remaining f_i are chosen by an adversary whose strategy can depend on the f_i for $i \in I_g$.

Let $\bar{f}$ denote the mean of f under p^*, i.e. $\bar{f}(w) \overset{\text{def}}{=} \mathbb{E}_{f \sim p^*}[f(w)]$ for $w \in \mathcal{H}$; our goal is to find a parameter $\hat{w}$ such that $\bar{f}(\hat{w}) - \bar{f}(w^*)$ is small, where w^* is the minimizer of $\bar{f}$. We use r to denote the ℓ_2-radius of $\mathcal{H}$, i.e. $r \overset{\text{def}}{=} \max_{w \in \mathcal{H}} \|w\|_2$.

This stochastic optimization setting captures most concrete settings of interest — for instance, mean estimation corresponds to $f_i(w) = \|w - x_i\|_2^2$, linear regression to $f_i(w) = (y_i - \langle w, x_i \rangle)^2$, and logistic regression to $f_i(w) = \log(1 + \exp(-y_i \langle w, x_i \rangle))$.

A key player: spectral norm of gradients. To state our main results, we need to define the following key quantity, where $\|\cdot\|_{\text{op}}$ denotes the spectral or operator norm:

$$S \overset{\text{def}}{=} \max_{w \in \mathcal{H}} \frac{1}{\sqrt{|I_g|}} \left\| \left[\nabla f_i(w) - \nabla \bar{f}(w) \right]_{i \in I_g} \right\|_{\text{op}}. \tag{1}$$

In words, if we form the matrix of gradients $[\nabla f_{i_1}(w) \ \cdots \ \nabla f_{i_{\alpha n}}(w)]$, where $\{i_1, \ldots, i_{\alpha n}\} = I_g$, then S measures the difference between

this matrix and its expectation in operator norm, maximized over all $w \in \mathcal{H}$. This will turn out to be a key quantity for understanding learnability in the adversarial setting. It acts as an analog of uniform convergence in classical learning theory, where one would instead study the quantity $\max_{w\in\mathcal{H}} \|\frac{1}{|I_g|}\sum_{i\in I_g}(\nabla f_i(w) - \nabla \bar{f}(w))\|_2$. Note that this latter quantity is always bounded above by S.

The fact that $f_i \sim p^*$ is irrelevant to our results—all that matters is the quantity S, which exists even for a deterministic set of functions $f_1, \ldots, f_n$. Furthermore, S only depends on the good data and is independent of the adversary.

Scaling of S: examples. The definition (1) is a bit complex, so we go over some examples for intuition. We will see later that for the first two examples below (estimating means and product distributions), our implied error bounds are "good", while for the final example (linear classification), our bounds are "bad".

Mean estimation: Suppose that $f_i(w) = \frac{1}{2}\|w - x_i\|_2^2$, where $x_i \sim \mathcal{N}(\mu, \sigma^2 I)$. Then $\nabla f_i(w) - \nabla \bar{f}(w) = x_i - \mu$, and so S is simply the maximum singular value of $\frac{1}{\sqrt{|I_g|}}[x_i - \mu]_{i\in I_g}$. This is the square root of the maximum eigenvalue of $\frac{1}{|I_g|}\sum_{i\in I_g}(x_i - \mu)(x_i - \mu)^\top$, which converges to σ for large n.

Product distributions: Suppose that x_i is drawn from a product distribution on $\{0,1\}^d$, where the jth coordinate is 1 with probability p_j. Let $f_i(w) = \sum_{j=1}^d x_{ij}\log(w_j) + (1 - x_{ij})\log(1 - w_j)$. In this case $\bar{f}(w) = \sum_{j=1}^d p_j \log(w_j) + (1 - p_j)\log(1 - w_j)$, and $w_j^* = p_j$, so that $\bar{f}(w) - \bar{f}(w^*)$ is the KL divergence between p and w.

The jth coordinate of $\nabla f_i(w) - \nabla \bar{f}(w)$ is $(x_{ij} - p_j)(1/w_j + 1/(1 - w_j))$. In particular, the matrix in the definition of S can be written as $D(w)\cdot([x_i - p]_{i\in I_g})/\sqrt{|I_g|}$, where $D(w)$ is a diagonal matrix with entries $1/w_j + 1/(1 - w_j)$. Suppose that p is *balanced*, meaning that $p_j \in [1/4, 3/4]$, and that we restrict w_j to lie in $[1/4, 3/4]$ as well. Then $\|D(w)\|_{\text{op}} \le 16/3$, while the matrix $[x_i - p]/\sqrt{|I_g|}$ has maximum singular value converging to $\max_{j=1}^d p_j(1 - p_j) \le \frac{1}{4}$ for large enough n. Thus $S = O(1)$ in this setting.

Linear classification: Suppose that $x_i \sim \mathcal{N}(0, I)$ and that $y_i = \text{sign}(u^\top x_i)$ for some unknown vector u. Our loss function is the logistic loss $f_i(w) = \log(1 + \exp(-y_i\langle w, x_i\rangle))$. In this case $\nabla f_i(w) = \frac{-y_i x_i}{1+\exp(y_i\langle w, x_i\rangle)}$. It is less obvious how to compute S, but Lemma 2.1 below implies that it is $O(1)$.

Sub-gaussian gradients: A useful general bound on S can be obtained assuming that the f_i have sub-Gaussian gradients. Recall that a random variable X is *σ-sub-Gaussian* if $\mathbb{E}[\exp(u^\top(X - \mu))] \le \exp(\frac{1}{2}\sigma^2\|u\|_2^2)$, where $\mu = \mathbb{E}[X]$. If ∇f_i is sub-Gaussian, then $S = O(\sigma)$ if $\alpha n \ge \tilde{\Omega}(d)$:

Lemma 2.1. *Suppose that for each w, $\nabla f_i(w)$ is σ-sub-Gaussian and L-Lipschitz for $f_i \sim p^*$. If αn is at least $d\max(1, \log(\frac{rL}{\sigma})) + \log(1/\delta)$, then $S = O(\sigma)$ with probability $1 - \delta$.*

In most of our concrete settings, sub-Gaussianity of ∇f_i corresponds to sub-Gaussianity of the data points $x_i \in \mathbb{R}^d$.

2.1 Main Results

We can now state our main results. Our first result is that, just using the untrusted data, we can output a small ellipse which contains a parameter attaining small error under $\bar{f}$. This meta-result underlies our results in the list decoding and semi-verified settings.

Theorem 2.2. *Given n data points containing a set I_g of αn data points with spectral norm bound S, we can obtain an ellipse $\mathcal{E}_Y = \{w \mid ww^\top \le Y\}$ such that $\text{tr}(Y) \le O\left(\frac{r^2}{\alpha}\right)$ and*

$$\min_{w\in\mathcal{E}_Y} \bar{f}(w) - \bar{f}(w^*) \le O\left(\frac{Sr}{\sqrt{\alpha}}\right). \tag{2}$$

Recall here that r is the ℓ_2-radius of the parameter space $\mathcal{H}$. Also note that when Y is invertible, $ww^\top \le Y$ is equivalent to $w^\top Y^{-1} w \le 1$, so Y really does define an ellipse. Theorem 2.2 shows that the unverified data is indeed helpful, by narrowing the space of possible parameters from all of $\mathcal{H}$ down to the small ellipse $\mathcal{E}_Y$.

To interpret the bound (2), consider the mean estimation example above, where $f_i(w) = \frac{1}{2}\|x_i - w\|_2^2$ with $x_i \sim \mathcal{N}(\mu, \sigma^2 I)$. Note that $\bar{f}(w) - \bar{f}(w^*) = \frac{1}{2}\|w - \mu\|_2^2$. Assuming that $\|\mu\|_2$ is known to within a constant factor, we can take $\mathcal{H}$ to be the ℓ_2-ball of radius $r = O(\|\mu\|_2)$. This leads to the bound $\|w - \mu\|_2^2 = O(\sigma\|\mu\|_2/\sqrt{\alpha})$, for some w in an ellipse of trace $\|\mu\|_2^2/\alpha$. Note that the ℓ_2-ball itself has trace $d\|\mu\|_2^2$, so the ellipse $\mathcal{E}_Y$ is much smaller than $\mathcal{H}$ if d is large. Moreover, a random $x_i \sim p^*$ will have $\|x_i - \mu\|_2^2 \approx d\sigma^2$, whereas the bound above implies that $\|w - \mu\|_2^2 \ll d\sigma^2$ if $\|\mu\|_2 \ll d\sigma$. Theorem 4.1 is thus doing real work, by finding a w that is much closer to μ than a randomly chosen x_i.

We note that by applying our algorithm multiple times we can improve the bound $\|w - \mu\|_2^2 = O(\sigma\|\mu\|_2/\sqrt{\alpha})$ to $\|w - \mu\|_2^2 = O(\sigma^2/\alpha)$, so that $\|w - \mu\|_2^2 \ll \|x_i - \mu\|_2^2$ independent of $\|\mu\|_2$. We discuss this in more detail in Section 6.

For an example where Theorem 2.2 is less meaningful, consider the linear classification example from before. In that case $S = O(1)$, and r is likely also $O(1)$, so we obtain the bound $\bar{f}(w) - \bar{f}(w^*) = O(1/\sqrt{\alpha})$. However, $\bar{f}(0) = \log(2)$ while $\bar{f}(w^*) \ge 0$, so $\bar{f}(0) - \bar{f}(w^*) \le \log(2) = O(1)$ and hence this bound is essentially vacuous.

List decodable learning. Using Theorem 2.2 as a starting point we can derive bounds for both models defined in Section 1, starting with the list decodable model. Here, we must make the further assumption that the f_i are κ-strongly convex, meaning that $f_i(w') - f_i(w) \ge (w' - w)^\top \nabla f_i(w) + \frac{\kappa}{2}\|w' - w\|_2^2$. The strong convexity allows us to show that for the good f_i, the parameters $\hat{w}_i = \arg\min_{w\in\mathcal{E}_Y} f_i(w)$ concentrate around w^*, with radius $r' \ll r$. By clustering the $\hat{w}_i$ and iteratively re-running our algorithm on each cluster, we can obtain bounds that do not depend on r, and output a single candidate parameter $\hat{w}_j$ for each cluster. We can thereby show:

Theorem 2.3. *Suppose the f_i are κ-strongly convex, and suppose there is a set I_g of size αn with spectral norm bound S. Then, for any $\varepsilon \le \frac{1}{2}$, it is possible to obtain a list of $m \le \lfloor\frac{1}{(1-\varepsilon)\alpha}\rfloor$ candidates $\hat{w}_1, \ldots, \hat{w}_m$, such that $\min_{j=1}^m \|\hat{w}_j - w^*\|_2 \le O\left(\frac{S}{\kappa}\sqrt{\frac{\log(\frac{2}{\alpha})}{\alpha\varepsilon}}\right)$.*

In Section 6 we state and prove a stronger version of this result. A key tool in establishing Theorem 2.3 is *padded decompositions* (Fakcharoenphol et al., 2003), which identify clusters in data while making minimal assumptions on the geometry of points outside of a cluster, and are thus useful in our adversarial setting.

Semi-verified learning. If the f_i are not strongly convex then we cannot employ the clustering ideas above. However, because we have reduced $\mathcal{H}$ to the much smaller set $\mathcal{E}_Y$, we can nevertheless often approximate w^* with only a small amount of verified data. In fact, in some settings we only need a single verified data point:

Lemma 2.4. *Suppose that $f(w) = \phi(w^\top x)$, where ϕ is 1-Lipschitz, and suppose that x has bounded qth moments in the sense that $\mathbb{E}_{p^*}[|\langle x - \mathbb{E}[x], u\rangle|^q]^{1/q} \le \sigma_q$ for all unit vectors u and some $q \ge 2$. Then given Y from Theorem 2.2 and a single verified $x \sim p^*$, we can obtain a $\hat{w}$ such that $\mathbb{P}_{x\sim p^*}\left[\bar{f}(\hat{w}) \ge \bar{f}(w^*) + C \cdot \frac{(S+t\sigma_q)r}{\sqrt{\alpha}}\right] \le t^{-q}$, for a universal constant C.*

The particular functional form for f_i was needed to obtain a concrete bound, but analogs of Lemma 2.4 should be possible in any setting where we can leverage the low complexity of $\mathcal{E}_Y$ into a bound on $f - \bar{f}$. Note that if we replace $\mathcal{E}_Y$ with $\mathcal{H}$ in Lemma 2.4, then the $r/\sqrt{\alpha}$ dependence becomes $r\sqrt{d}$, which is usually vacuous.

Optimality? The dependence on S, r and κ in the results above seems essentially necessary, though the optimal dependence on α is less clear. In Section 7 we show lower bounds for robust mean estimation even if p^* is known to be Gaussian. These bounds roughly translate to a lower bound of $\Omega\left(\frac{S}{\kappa}\sqrt{\log(1/\alpha)}\right)$ for strongly convex f_i, and $\Omega\left(Sr\sqrt{\log(1/\alpha)}\right)$ for linear f_i, and hold in both the list decodable and semi-verified settings. For general distributions, it is unclear whether the optimal dependence on α is $\sqrt{1/\alpha}$ or $\sqrt{\log(1/\alpha)}$ or somewhere in-between. We do note that any dependence better than $\sqrt{1/\alpha}$ would improve the best known results for efficiently solving k-means for well-separated clusters, which may suggest at least a computational barrier to achieving $\sqrt{\log(1/\alpha)}$.

2.2 Implications

We now go over some implications of our general results in some more specific settings. All the results below follow as corollaries of our main theorems, and are proved in the full version of the paper.

Robust mean estimation. Suppose we observe points $x_1, \dots, x_n \in \mathbb{R}^d$, of which αn are drawn from a distribution p^* with bounded covariance, and our goal is to recover the mean $\mu = \mathbb{E}_{x\sim p^*}[x]$. If we take $f_i(w) = \|w - x_i\|_2^2$, then Theorem 2.3, together with the matrix concentration bound Proposition 1.3, implies the following:

Corollary 2.5. *Suppose p^* has bounded covariance: $\mathrm{Cov}_{p^*}[x] \le \sigma^2 I$. Then, for $n \ge \frac{d}{\alpha}$, with probability $1 - \exp(-\Omega(\alpha n))$ it is possible to output $m \le O\left(\frac{1}{\alpha}\right)$ candidate means $\hat{\mu}_1, \dots, \hat{\mu}_m$ such that $\min_{j=1}^m \|\mu - \hat{\mu}_j\|_2 \le O\left(\sigma\sqrt{\frac{\log(2/\alpha)}{\alpha}}\right)$. Moreover, if $\alpha \ge 0.51$ then we can take $m = 1$.*

We can compare to the results of Lai et al. (2016) and Diakonikolas et al. (2016), who study mean estimation when $\alpha > \frac{1}{2}$ and one is required to output a single parameter (i.e., $m = 1$). For simplicity take $\alpha = \frac{3}{4}$. Roughly, Lai et al. (2016) obtain error $O(\sigma\sqrt{\log(d)})$ with sample complexity $n = O(d)$, while requiring a bound on the fourth moments of p^*; Diakonikolas et al. (2016) obtain error $O(\sigma)$ with sample complexity $n = O\left(d^3\right)$, and require p^* to be sub-Gaussian. Corollary 2.5 improves both of these by yielding error $O(\sigma)$ with sample complexity $n = O(d)$, and only requires p^* to have bounded second moments.[2] We note that in contrast to our results, these other results obtain error that vanishes as $\alpha \to 1$ (at a rate of $O(\sqrt{1-\alpha})$ in the first case and $\widetilde{O}(1-\alpha)$ in the second case). We thus appear to incur some looseness when $\alpha \approx 1$, in exchange for obtaining results in the previously unexplored setting $\alpha \le \frac{1}{2}$. It would be interesting to obtain a single algorithm that both applies when $\alpha \ll 1$ and achieves vanishing error as $\alpha \to 1$.

Learning mixture of distributions. In addition to robust mean estimation, we can use our results to efficiently learn mixtures of distributions, by thinking of a single mixture component as the good data and the remaining mixture components as bad data. Again applying Theorem 2.2 to $f_i(w) = \|w - x_i\|_2^2$, we obtain the following result, which says that we can successfully cluster samples from a mixture of distributions, even in the presence of arbitrary corruptions, provided the cluster means are separated in ℓ_2 distance by $\tilde{\Omega}(\sigma/\sqrt{\alpha})$.

Corollary 2.6. *Suppose we are given n samples, where each sample either comes from one of k distributions $p_1^*, \dots, p_k^*$ (with $\mathrm{Cov}_{p_i^*}[x] \le \sigma^2 I$ for all i), or is arbitrary. Let μ_i be the mean of p_i^*, let α_i be the fraction of points from p_i^*, and let $\alpha = \min_{i=1}^k \alpha_i$. Then if $n \ge \frac{d}{\alpha}$, with probability $1 - k\exp(-\Omega(\alpha\varepsilon^2 n))$ we can obtain a partition $T_1, \dots, T_m$ of $[n]$ and corresponding candidate means $\hat{\mu}_1, \dots, \hat{\mu}_m$ such that: for all but $\varepsilon\alpha_i n$ of the points drawn from p_i^*, the point lies in a set T_j with candidate mean $\hat{\mu}_j$ satisfying $\|\mu_i - \hat{\mu}_j\|_2 \le O\left(\frac{\sigma}{\varepsilon}\sqrt{\frac{\log(2/\alpha)}{\alpha}}\right)$. Moreover, $m \le O\left(\frac{1}{\alpha}\right)$.*

The $\frac{1}{\varepsilon}$ dependence can be replaced with $\sqrt{\log(n)/\varepsilon}$, or with $\sqrt{\log(2/\varepsilon)}$ if the x_i are sub-Gaussian. The only difference is in which matrix concentration bound we apply to the x_i. Corollary 2.6 says that we can partition the points into $O\left(\frac{1}{\alpha}\right)$ sets, such that two points from well-separated clusters are unlikely to end up in the same set. Note however that one cluster might be partitioned into multiple sets.

For comparison, the best known efficient algorithm for clustering a mixture of k distributions (with few/no corruptions) requires mean separation roughly $\tilde{O}(\sigma\sqrt{k})$ (Achlioptas and McSherry, 2005; Awasthi and Sheffet, 2012), which our result matches if $\alpha = \Omega(1/k)$.

Planted partitions. We next consider implications of our results in a version of the planted partition model (McSherry, 2001). In this model we observe a random directed graph, represented as a matrix $A \in \{0,1\}^{n\times n}$. For disjoint subsets $I_1, \dots, I_k$ of $[n]$, we generate edges as follows: (i) If $u, v \in I_i$, then $p(A_{uv} = 1) = \frac{a}{n}$. (ii) If $u \in I_i, v \notin I_i$, then $p(A_{uv} = 1) = \frac{b}{n}$. (iii) If $u \notin \cup_{i=1}^k I_i$, the edges emanating from u can be arbitrary. In contrast to the typical planted partition model, we allow some number of corrupted vertices not belonging to any of the I_i. In general a and b could depend on the partition indices i, j, but we omit this for simplicity.

[2] Some fine print: Diakonikolas et al. also estimate the covariance matrix, and their recovery results are stronger if $\Sigma = \mathrm{Cov}[x]$ is highly skewed; they roughly show $\|\hat{\mu} - \mu\|_{\Sigma^{-1}} = O(1)$. The adversary model considered in both of these other papers is also slightly more general than ours: first n points are drawn from p^* and then an adversary is allowed to corrupt $(1-\alpha)n$ of the points. However, it is straightforward to show (by monotonicity of the operator norm) that our bounds will be worse by at most a $1/\sqrt{\alpha}$ factor in this stricter setting, which is a constant if $\alpha = \frac{3}{4}$.

Note that the distribution over the row A_u is the same for all $u \in I_i$. By taking this distribution to be the distribution p^*, Theorem 2.3 yields the following result:

Corollary 2.7. *For the planted partition model above, let* $\alpha = \min_{i=1}^{k} \frac{|I_i|}{n}$. *Then, with probability* $1 - \exp(-\Omega(\alpha n))$, *we can obtain sets* $T_1, \ldots, T_m \subseteq [n]$, *with* $m \leq O\left(\frac{1}{\alpha}\right)$, *such that for all* $i \in [k]$, *there is a* $j \in [m]$ *with* $|I_i \triangle T_j| \leq O\left(\frac{a \log(\frac{2}{\alpha})}{\alpha^2 (a-b)^2}\right) n$, *where* $\triangle$ *denotes symmetric difference.*

This shows that we can approximately recover the planted partition, even in the presence of arbitrary corruptions, provided $\frac{(a-b)^2}{a} \gg \frac{\log(2/\alpha)}{\alpha^3}$ (since the bound on $|I_i \triangle T_j|$ needs to be less than αn to be meaningful). In contrast, the best efficient methods (assuming no corruptions) roughly require $\frac{(a-b)^2}{a+(k-1)b} \gg k$ in the case of k equal-sized communities (Abbe and Sandon, 2015a;b). In the simplifying setting where $b = \frac{1}{2}a$, our bounds require $a \gg k^3 \log(k)$ while existing bounds require $a \gg k^2$. The case of unequal size communities is more complex, but roughly, our bounds require $a \gg \frac{\log(2/\alpha)}{\alpha^3}$ in contrast to $a \gg \frac{1}{\alpha^2}$.

Summary. For robust mean estimation, we match the best existing error bounds of $O(\sigma)$ when $\alpha = \frac{3}{4}$, under weaker assumptions. For learning mixtures distributions, we match the best bound of $\tilde{O}(\sigma \sqrt{k})$ when $\alpha = \Omega(1/k)$. For recovering planted partitions, we require average degree $k^3 \log(k)$, in contrast to the best known bound of k^2. It is pleasing that a single meta-algorithm is capable of matching or nearly matching the best rate in these settings, despite allowing for arbitrary corruptions. We can also achieve bounds for robust density estimation; see the full paper for details.

3 ALGORITHM

In this section we present our algorithm, which consists of an SDP coupled with an outlier removal step. At a high level, our algorithm works as follows: first, we give *each function* f_i its own parameter vector w_i, and minimize $\sum_{i=1}^{n} f_i(w_i)$ subject to regularization which ensures the w_i remain close to each other; formally, we bound the w_i to lie within a small ellipse. The reason for doing this is that the different w_i are now only coupled via this regularization, and so the influence of adversarial data on the good parameters can only come from its effect on the shape of the ellipse. We will show that whenever the adversaries affect the shape of the ellipse more than a small amount, they are necessarily outliers that can be identified and removed. In the remainder of this section, we elaborate on these two steps of regularization and outlier removal, and provide pseudocode.

Per-function adaptivity. If the functions $f_1, \ldots, f_n$ were all drawn from p^* (i.e., there are no adversaries), then a natural approach would be to let $\hat{w}$ be the minimizer of $\sum_{i=1}^{n} f_i(w)$, which will approximately minimize $\bar{f}(w)$ by standard concentration results.

The problem with using this approach in the adversarial setting is that even a single adversarially chosen function f_i could substantially affect the value of $\hat{w}$. To minimize this influence, we give each f_i its own parameter w_i, and minimize $\sum_{i=1}^{n} f_i(w_i)$, subject to a regularizer which encourages the w_i to be close together. The adversary now has no influence on the good w_i except via the regularizer, so the key challenge is to find a regularizer which sufficiently controls statistical error while also bounding the influence of the adversary.

Algorithm 1 Algorithm for fitting p^*.

Input: $f_1, \ldots, f_n$
Initialize $c \leftarrow [1; \cdots; 1] \in \mathbb{R}^n$
Set $\lambda \leftarrow \frac{\sqrt{8\alpha} n S}{r}$
while true do
  Let $\hat{w}_{1:n}, \hat{Y}$ be the solution to

$$\underset{w_1,\ldots,w_n, Y}{\text{minimize}} \sum_{i=1}^{n} c_i f_i(w_i) + \lambda \operatorname{tr}(Y) \quad \text{subject to } w_i w_i^\top \preceq Y \text{ for all } i = 1, \ldots, n. \tag{3}$$

  if $\operatorname{tr}(\hat{Y}) \leq \frac{6r^2}{\alpha}$ **then** ▷ Check for outliers
    return $\hat{w}_{1:n}$, $\hat{Y}$ ▷ Not many outliers, can return
  else
    $c \leftarrow \textsc{UpdateWeights}(c, \hat{w}_{1:n}, \hat{Y})$ ▷ Re-weight points to down-weight outliers
  end if
end while

It turns out that the right regularizer in this case constrains the w_i to lie within an *ellipse with small trace*. Formally, the centerpiece of our algorithm is the following convex optimization problem:[3]

$$\underset{w_1,\ldots,w_n, Y}{\text{minimize}} \sum_{i=1}^{n} c_i f_i(w_i) + \lambda \operatorname{tr}(Y) \quad \text{subject to } w_i w_i^\top \preceq Y \text{ for all } i = 1, \ldots, n. \tag{4}$$

Here the coefficients c_i are non-negative weights which will eventually be used to downweight outliers (for now imagine that $c_i = 1$).

Note that the semidefinite constraint $w_i w_i^\top \preceq Y$ is equivalent to $w_i^\top Y^{-1} w_i \leq 1$, which says that w_i lies within the ellipse centered at 0 defined by Y. The regularizer is thus the trace of the minimum ellipse containing the w_i; penalizing this trace will tend to push the w_i closer together, but is there any intuition behind its geometry? The following lemma shows that $\operatorname{tr}(Y)$ is related to the trace norm of $[w_1 \ \cdots \ w_n]$:

Lemma 3.1. *For any points* $w_1, \ldots, w_n \in \mathbb{R}^d$, *suppose that* $Y \succeq w_i w_i^\top$ *for all* i. *Then, letting* $\|\cdot\|_*$ *denote the trace norm (i.e., sum of singular values) and* $W_T = [w_i]_{i \in T}$, *we have* $\operatorname{tr}(Y) \geq \frac{\|W_T\|_*^2}{|T|}$ *for all sets* $T \subseteq [n]$.

The appearance of the trace norm makes sense in light of the intuition that we should be clustering the functions f_i; indeed, trace norm regularization is a key ingredient in spectral algorithms for clustering (see e.g. Chen et al. (2014a;b); Zha et al. (2001)). Lemma 3.1 says that $\operatorname{tr}(Y)$ simultaneously bounds the trace norm on every subset T of $[n]$, which ends up yielding better results than are obtained by simply penalizing the overall trace norm; we believe that this *local trace norm regularization* likely leads to better results even in non-adversarial spectral learning settings. The most important

[3] It is convex because the constraint $w_i w_i^\top \preceq Y$ is equivalent to $[Y w_i; w_i^\top 1] \succeq 0$. Given an oracle for computing $\nabla f_i(w)$, it can be solved in $\text{poly}(n, d)$ time.

Algorithm 2 Algorithm for updating c to downweight outliers.

procedure UpdateWeights(c, $\hat{w}_{1:n}$, $\hat{Y}$)
  for $i = 1, \ldots, n$ **do**
    Let $\tilde{w}_i$ be the solution to

$$\begin{aligned} &\underset{\tilde{w}_i, a_i}{\text{minimize}} \;\; f_i(\tilde{w}_i) \\ &\text{subject to} \;\; \tilde{w}_i = \sum_{j=1}^{n} a_{ij} \hat{w}_j, \qquad (5) \\ &\qquad 0 \le a_{ij} \le \frac{2}{\alpha n}, \qquad \sum_{j=1}^{n} a_{ij} = 1. \end{aligned}$$

    Let $z_i \leftarrow f_i(\tilde{w}_i) - f_i(\hat{w}_i)$
  end for
  $z_{\max} \leftarrow \max\{z_i \mid c_i \neq 0\}$
  $c_i' \leftarrow c_i \cdot \frac{z_{\max} - z_i}{z_{\max}}$ for $i = 1, \ldots, n$
  return c'
end procedure

property of $\mathrm{tr}(Y)$ is that it admits a certain type of local Hölder's inequality which we will explain in Section 5.

Removing outliers. Solving (4) is not by itself sufficient to achieve robustness. The problem is that a single function f_i could strongly push w_i to a given value w_{target} (e.g. if $f_i(w) = 10^{100}\|w - w_{\text{target}}\|_2^2$) which allows the adversaries to arbitrarily expand the ellipse defined by Y. To combat this, we need some way of removing outlier functions f_i from our dataset. We will do this in a soft way, by assigning a weight c_i to each function f_i, and downweighting functions that seem likely to be outliers.

How can we tell that a function is an outlier? Intuitively, if a function f_i is really drawn from p^*, then there should be many other functions f_j, $j \neq i$, that are "similar" to f_i. We can quantify this by considering whether there are a large number of $j \neq i$ for which the parameter w_j for f_j does a good job of minimizing f_i. Formally, given a solution $(\hat{w}_1, \ldots, \hat{w}_n)$ to (4), we compare $\hat{w}_i$ to $\tilde{w}_i$, which is defined as the solution to the following optimization:

$$\begin{aligned} &\underset{\tilde{w}_i, a_i}{\text{minimize}} \;\; f_i(\tilde{w}_i) \qquad (6) \\ &\text{subject to} \;\; \tilde{w}_i = \sum_{j=1}^{n} a_{ij} \hat{w}_j, \quad 0 \le a_{ij} \le \frac{2}{\alpha n}, \qquad \sum_{j=1}^{n} a_{ij} = 1. \end{aligned}$$

The optimization (6) roughly asks for a parameter $\tilde{w}_i$ that minimizes f_i, subject to $\tilde{w}_i$ being the average of at least $\frac{\alpha n}{2}$ distinct parameters $\hat{w}_j$. Given the solution $\tilde{w}_i$ to (6), we then downweight the influence of the ith data point based on the value of $f_i(\tilde{w}_i) - f_i(\hat{w}_i)$. In particular, we will multiply the weight c_i by $1 - \eta\,(f_i(\tilde{w}_i) - f_i(\hat{w}_i))$ for some appropriate η. Hopefully, this will downweight any outliers by a large amount while only downweighting good points by a small amount (this hope is verified in Lemma 4.5 below).

Pseudocode for our algorithm is given in Algorithms 1 and 2.

4 APPROACH AND PROOF OUTLINE

We now provide an outline of the proof of Theorem 2.2, by analyzing the output of Algorithm 1. The structure of our proof has analogies to classical uniform convergence arguments, so we will start by reviewing that case.

Warm-up: Uniform Convergence. In uniform convergence arguments, we assume that all of $f_1, \ldots, f_n$ are drawn from p^*, which brings us into the realm of classical learning theory. The analogue to the optimization (3) is regularized empirical risk minimization:

$$\hat{w} = \underset{w \in \mathcal{H}}{\arg\min} \sum_{i=1}^{n} f_i(w) + \lambda h(w), \qquad (7)$$

where $h(w)$ is a non-negative regularizer. Uniform convergence arguments involve two parts:

(1) **Bound the optimization error**: Use the definition of $\hat{w}$ to conclude that $\sum_{i=1}^n f_i(\hat{w}) \le \sum_{i=1}^n f_i(w^*) + \lambda h(w^*)$ (since $\hat{w}$ minimizes (7)). This step shows that $\hat{w}$ does almost as well as w^* at minimizing the empirical risk $\sum_{i=1}^n f_i(w)$.
(2) **Bound the statistical error**: Show, via an appropriate concentration inequality, that $\frac{1}{n}\sum_{i=1}^n f_i(w)$ is close to $\bar{f}(w)$ for all $w \in \mathcal{H}$. Therefore, $\hat{w}$ is nearly as good as w^* in terms of the true risk $\bar{f}$.

We will see next that the proof of Theorem 2.2 contains steps similar to these, though bounding the statistical error in the presence of adversaries requires an additional step of removing outliers.

Proof Overview. We will establish a stronger version of Theorem 2.2, which exhibits an explicit $w \in \mathcal{E}_{\hat{Y}}$ with small error:

Theorem 4.1. *Let $\hat{w}_{1:n}$, $\hat{Y}$ be the output of Algorithm 1, and let $\hat{w}_{\text{avg}} = (\sum_{i \in I_g} c_i \hat{w}_i) / (\sum_{i \in I_g} c_i)$. Then, $\bar{f}(\hat{w}_{\text{avg}}) - \bar{f}(w^*) \le 18Sr/\sqrt{\alpha}$. Furthermore, $\hat{w}_{\text{avg}} \in \mathcal{E}_{\hat{Y}}$ and $\mathrm{tr}(\hat{Y}) \le 6r^2/\alpha$.*

To prove Theorem 4.1, recall that Algorithm 1 has at its core the following convex optimization problem:

$$\begin{aligned} &\underset{w_1, \ldots, w_n, Y}{\text{minimize}} \;\; \sum_{i=1}^{n} c_i f_i(w_i) + \lambda\, \mathrm{tr}(Y) \\ &\text{subject to } w_i w_i^\top \preceq Y \text{ for all } i = 1, \ldots, n. \qquad (8) \end{aligned}$$

This optimization asks to minimize $\sum_{i=1}^n c_i f_i(w_i)$ while constraining the w_i to lie within the ellipse defined by Y. As in the uniform convergence argument above, there are two sources of error that we need to bound: the *optimization error* $\sum_{i \in I_g} c_i\,(f_i(\hat{w}_i) - f_i(w^*))$, and the *statistical error* $\sum_{i \in I_g} c_i \left(\bar{f}(\hat{w}_{\text{avg}}) - f_i(\hat{w}_i)\right)$. Note that the statistical error now measures two quantities: the distance from $f_i(\hat{w}_i)$ to $f_i(\hat{w}_{\text{avg}})$, and from $f_i(\hat{w}_{\text{avg}})$ to $\bar{f}(\hat{w}_{\text{avg}})$.

Bounding the optimization error requires showing that the ellipse defined by $\hat{Y}$ is not too small (so that it contains w^*), while bounding the statistical error requires showing that the ellipse is not too large (so that we cannot overfit too much). The former turns out to be easy and is shown in Lemma 4.2. The latter is more involved and requires several steps. First, we show in Lemma 4.3 that the statistical error can be bounded in terms of $\mathrm{tr}(Y)$ and S, which verifies the intuition that bounding the statistical error reduces to bounding Y. Next, in Lemma 4.4, we show that the parameters $\tilde{w}_i$ found in Algorithm 2 are bounded by an ellipse $\tilde{Y}$ with small trace, and that $f_i(\tilde{w}_i) \approx f_i(\hat{w}_i)$ for $i \in I_g$. By the optimality of $(\hat{w}_{1:n}, \hat{Y})$ for (8), the only way that $\mathrm{tr}(\hat{Y})$ can be much larger than $\mathrm{tr}(\tilde{Y})$ is therefore if $f_i(\hat{w}_i) \ll f_i(\tilde{w}_i)$ for $i \notin I_g$. In this case, we can

identify outliers $i \notin I_g$ by considering the value of $f_i(\tilde{w}_i) - f_i(\hat{w}_i)$, and Lemma 4.5 verifies that we can use this to perform outlier removal. We expand on both the optimization error and statistical error bounds below.

Bounding optimization error on I_g. Throughout the argument, we will make use of the optimality of $(\hat{w}_{1:n}, \hat{Y})$ for (8), which implies

$$\sum_{i=1}^{n} c_i f_i(\hat{w}_i) + \lambda \operatorname{tr}(\hat{Y}) \le \sum_{i=1}^{n} c_i f_i(w_i) + \lambda \operatorname{tr}(Y) \tag{9}$$

for any feasible $(w_{1:n}, Y)$. We wish to bound $\sum_{i\in I_g} c_i f_i(\hat{w}_i)$, but the preceding bound involves all of $\sum_{i=1}^{n} c_i f_i(\hat{w}_i)$, not just the f_i for $i \in I_g$. However, because the $\hat{w}_i$ are free to vary independently, we can bound $\sum_{i\in I_g} c_i \left(f_i(\hat{w}_i) - f_i(w^*)\right)$ in terms of the amount that $\operatorname{tr}(\hat{Y})$ would need to increase before $w^*(w^*)^\top \preceq \hat{Y}$. In particular, by taking $Y = \hat{Y} + (w^*)(w^*)^\top$ in (9), we can obtain the following bound on the optimization error:

Lemma 4.2. *The solution $\hat{w}_{1:n}$ to (8) satisfies*

$$\sum_{i\in I_g} c_i \left(f_i(\hat{w}_i) - f_i(w^*)\right) \le \lambda \|w^*\|_2^2. \tag{10}$$

Bounding the statistical error. We next consider the statistical error. We cannot bound this error via standard uniform convergence techniques, because each f_i has a different argument $\hat{w}_i$. However, it turns out that the operator norm bound S, together with a bound on $\operatorname{tr}(\hat{Y})$, yield concentration of the f_i to $\bar{f}$. In particular, we have:

Lemma 4.3. *Let $\hat{w}_{\text{avg}} \stackrel{\text{def}}{=} \frac{\sum_{i\in I_g} c_i \hat{w}_i}{\sum_{i\in I_g} c_i}$. Then the solution $\hat{w}_{1:n}$, $\hat{Y}$ to (8) satisfies*

$$\sum_{i\in I_g} c_i \left(f_i(\hat{w}_{\text{avg}}) - f_i(\hat{w}_i)\right) \le \alpha n S \left(\sqrt{\operatorname{tr}(\hat{Y})} + r\right), \text{ and} \tag{11}$$

$$\sum_{i\in I_g} c_i \left(\bar{f}(\hat{w}_{\text{avg}}) - \bar{f}(w^*)\right) \le \sum_{i\in I_g} c_i \left(f_i(\hat{w}_{\text{avg}}) - f_i(w^*)\right) + 2\alpha n r S. \tag{12}$$

Lemma 4.3 relates $f_i(\hat{w}_i)$ to $f_i(\hat{w}_{\text{avg}})$ in (11), and then relates $f_i(\hat{w}_{\text{avg}})$ to $\bar{f}(\hat{w}_{\text{avg}})$ in (12). Together these allow us to bound the statistical error in terms of $\operatorname{tr}(\hat{Y})$ and S. The proof is an application of the matrix Hölder's inequality $|\operatorname{tr}(A^\top B)| \le \|A\|_* \|B\|_{\text{op}}$, with $A_i = \hat{w}_i - \hat{w}_{\text{avg}}$ and $B_i = \nabla f_i(\hat{w}_{\text{avg}}) - \nabla \bar{f}(\hat{w}_{\text{avg}})$.

Bounding the trace. We next bound $\operatorname{tr}(\hat{Y})$ itself. We again exploit the optimality constraint (9), which implies that $\operatorname{tr}(\hat{Y}) \le \operatorname{tr}(Y) + \frac{1}{\lambda}\left(\sum_{i=1}^{n} c_i \left(f_i(w_i) - f_i(\hat{w}_i)\right)\right)$ for any feasible $(w_{1:n}, Y)$. We will take $w_{1:n}$ to be $\tilde{w}_{1:n}$ as defined in equation (5) of Algorithm 2; we then take Y to be $\frac{2}{\alpha n}\hat{W}\hat{W}^\top$, where $\hat{W} = [\hat{w}_1 \ \cdots \ \hat{w}_n]$. Lemma 4.4 asserts that $(w_{1:n}, Y)$ is feasible, and uses this to bound $\operatorname{tr}(\hat{Y})$:

Lemma 4.4. *For $\tilde{w}_i$ as defined in (6), and $\tilde{Y} \stackrel{\text{def}}{=} \frac{2}{\alpha n}\hat{W}\hat{W}^\top$, we have $\tilde{w}_i\tilde{w}_i^\top \preceq \tilde{Y}$ for all i, and also $\operatorname{tr}(\tilde{Y}) \le \frac{2r^2}{\alpha}$. In addition,*

$$\operatorname{tr}(\hat{Y}) \le \frac{2r^2}{\alpha} + \frac{1}{\lambda}\left(\sum_{i=1}^{n} c_i \left(f_i(\tilde{w}_i) - f_i(\hat{w}_i)\right)\right), \tag{13}$$

and if $\sum_{i\in I_g} c_i \ge \frac{\alpha n}{2}$ then

$$\sum_{i\in I_g} c_i \left(f_i(\tilde{w}_i) - f_i(\hat{w}_i)\right) \le \alpha n \left(\sqrt{\operatorname{tr}(\hat{Y})} + r\right). \tag{14}$$

This "almost" bounds $\operatorname{tr}(\hat{Y})$ in the following sense: if instead of a bound on $\sum_{i\in I_g} c_i \left(f_i(\tilde{w}_i) - f_i(\hat{w}_i)\right)$, (14) gave a bound on the full sum $\sum_{i=1}^{n} c_i \left(f_i(\tilde{w}_i) - f_i(\hat{w}_i)\right)$, then we could plug in to (13) to obtain (e.g.) $\operatorname{tr}(\hat{Y}) \le \frac{2r^2}{\alpha} + \frac{n}{\lambda}\left(\sqrt{\operatorname{tr}(\hat{Y})} + r\right)$, after which solving the quadratic for $\operatorname{tr}(\hat{Y})$ would yield a bound. The issue is that $f_i(\tilde{w}_i)$, for $i \notin I_g$, could be arbitrarily large, so additional work is needed.

Outlier removal. This brings us to our final idea of *outlier removal*. The intuition is the following: let $z_i \stackrel{\text{def}}{=} f_i(\tilde{w}_i) - f_i(\hat{w}_i)$. Then either: (i) the average of z_i over all of $[n]$ is not much larger than over I_g (in which case the bound (14) extends from I_g to $[n]$), or (ii) the average of z_i is much larger on $[n]$ than on I_g, in which case it should be possible to downweight the points in $[n]\backslash I_g$ a substantial amount relative to the points in I_g. This is the role that the outlier removal step (Algorithm 2) plays, and Lemma 4.5 formalizes its effect on the weights c_i.

Lemma 4.5. *Suppose that $\frac{1}{n}\sum_{i=1}^{n} c_i \left(f_i(\tilde{w}_i) - f_i(\hat{w}_i)\right)$ is at least $\frac{2}{\alpha n}\sum_{i\in I_g} c_i \left(f_i(\tilde{w}_i) - f_i(\hat{w}_i)\right)$ (†). Then, the update step in Algorithm 2 satisfies*

$$\frac{1}{\alpha n}\sum_{i\in I_g} c_i - c_i' \le \frac{1}{2n}\sum_{i=1}^{n} c_i - c_i'. \tag{15}$$

Moreover, the supposition (†) holds if $\lambda = \frac{\sqrt{8\alpha} n S}{r}$ and $\operatorname{tr}(\hat{Y}) > \frac{6r^2}{\alpha}$.

Lemma 4.5 says that, if the average value of z_i is at least twice as large over $[n]$ as over I_g, then the weights c_i decrease at most half as quickly on I_g as on $[n]$. Moreover, this holds whenever $\operatorname{tr}(\hat{Y}) > \frac{6r^2}{\alpha}$.

Combining the results. Lemma 4.5 ensures that eventually we have $\operatorname{tr}(\hat{Y}) \le O\left(\frac{r^2}{\alpha}\right)$, which allows us to bound the overall statistical error (using Lemma 4.3) by $O\left(\sqrt{\alpha} n r S\right)$. In addition, since $\lambda = O\left(\sqrt{\alpha} n S / r\right)$, the optimization error is bounded (via Lemma 4.2) by $O\left(\sqrt{\alpha} n r S\right)$, as well. Combining the various bounds, we obtain

$$\left(\sum_{i\in I_g} c_i\right)\left(\bar{f}\left(\hat{w}_{\text{avg}}\right) - \bar{f}\left(w^*\right)\right) \le O\left(\sqrt{\alpha} n r S\right). \tag{16}$$

Then, since (15) ensures that the c_i decrease twice as quickly over $[n]$ as over I_g, we decrease $\sum_{i\in I_g} c_i$ by at most a factor of 2 over all iterations of the algorithm, so that $\sum_{i\in I_g} c_i \ge \frac{\alpha n}{2}$. Dividing (16) through by $\sum_{i\in I_g} c_i$ then yields Theorem 4.1.

The proofs of Lemmas 4.2 through 4.5, as well as the formal proof of Theorem 4.1, are given in the full version of the paper.

5 CONCENTRATION OF ERRORS: A LOCAL HÖLDER'S INEQUALITY

Most of the bounds in Section 4 are bounds on an average error: for instance, Lemma 4.2 bounds the average difference between $f_i(\hat{w}_i)$ and $f_i(w^*)$, and Lemma 4.3 bounds the average difference between $f_i(\hat{w}_{\text{avg}})$ and $f_i(\hat{w}_i)$. One might hope for a stronger bound, showing that the above quantities are close together for almost all i, rather than only close in expectation. This is relevant, for instance, in a clustering setting, where we would like to say that almost all points are assigned a parameter $\hat{w}_i$ that is close to the true cluster center. Even beyond this relevance, asking whether we

obtain concentration of errors in this adversarial setting seems like a conceptually natural question:

> *If the good data is sub-Gaussian, can we obtain sub-Gaussian concentration of the errors, or can the adversary force the error to have heavy tails? What properties of the good data affect concentration of errors in the presence of an adversary?*

In this section, we will show that we can indeed obtain sub-Gaussian concentration, at least for the statistical error. In particular, we will characterize the concentration behavior of the errors $f_i(\hat{w}_{\text{avg}}) - f_i(\hat{w}_i)$ using a *local Hölder's inequality*, which depends upon a refined notion of S that we denote by S_ε. Before defining S_ε, we will state the local Hölder's inequality:

Lemma 5.1. *Suppose that the weights $b_i \in [0,1]$ satisfy $\sum_{i\in I_g} b_i \geq \varepsilon\alpha n$, and that the parameters w_i satisfy $w_i w_i^\top \preceq Y$. Then, for any $w_0 \in \mathcal{H}$, we have*

$$\left|\sum_{i\in I_g} b_i \langle w_i, \nabla f_i(w_0) - \nabla \bar{f}(w_0)\rangle\right| \leq \left(\sum_{i\in I_g} b_i\right)\sqrt{\text{tr}(Y)}S_\varepsilon. \tag{17}$$

We call this a local Hölder's inequality because it is a sharpening of the following bound, which can be established via the matrix Hölder's inequality:

$$\left|\sum_{i\in I_g} c_i \langle w_i, \nabla f_i(w_0) - \nabla \bar{f}(w_0)\rangle\right| \leq \alpha n\sqrt{\text{tr}(Y)}S. \tag{18}$$

By taking $b_i = \mathbb{I}[\langle w_i, \nabla f_i(w_0) - \nabla \bar{f}(w_0)\rangle > \sqrt{\text{tr}(Y)}S_\varepsilon]$, Lemma 5.1 implies in particular that $\langle w_i, \nabla f_i(w_0) - \nabla \bar{f}(w_0)\rangle \leq \sqrt{\text{tr}(Y)}S_\varepsilon$ for all but $\varepsilon\alpha n$ values of $i \in I_g$.

A local spectral norm bound. We now define S_ε; it is the maximum operator norm over subsets of I_g of size at least $\varepsilon|I_g|$:

$$S_\varepsilon \stackrel{\text{def}}{=} \max_{w\in\mathcal{H}} \max_{T\subseteq I_g, |T|\geq\lfloor\varepsilon\alpha n\rfloor} \frac{1}{\sqrt{|T|}} \left\|\left[\nabla f_i(w) - \nabla \bar{f}(w)\right]_{i\in T}\right\|_{\text{op}}. \tag{19}$$

(As a special case note that $S_1 = S$.) The quantity S_ε bounds not just the operator norm of all of the points in I_g, but also the operator norm on any large subset of I_g. We will see later that it is often possible to obtain good bounds on S_ε.

Concentration of statistical error. Using S_ε, we can obtain an improved version of the bounds (11) and (12) from Lemma 4.3, showing that $f_i(\hat{w}_i)$ is close to a nominal value $f_i(\hat{w}^b_{\text{avg}})$ for almost all $i \in I_g$:

Lemma 5.2. *Let the weights $b_i \in [0,1]$ satisfy $\sum_{i\in I_g} b_i \geq \varepsilon\alpha n$, and define $\hat{w}^b_{\text{avg}} \stackrel{\text{def}}{=} \frac{\sum_{i\in I_g} b_i \hat{w}_i}{\sum_{i\in I_g} b_i}$. Then the solution $\hat{w}_{1:n}, \hat{Y}$ to (8) satisfies*

$$\begin{aligned}\sum_{i\in I_g} b_i\left(f_i(\hat{w}^b_{\text{avg}}) - f_i(\hat{w}_i)\right) &\leq \sum_{i\in I_g} b_i\langle\nabla f_i(\hat{w}^b_{\text{avg}}), \hat{w}^b_{\text{avg}} - \hat{w}_i\rangle \\ &\leq \left(\sum_{i\in I_g} b_i\right) S_\varepsilon\left(\sqrt{\text{tr}(\hat{Y})} + r\right).\end{aligned} \tag{20}$$

Moreover, for any $w, w' \in \mathcal{H}$, we have

$$\left|\sum_{i\in I_g} b_i(\bar{f}(w) - \bar{f}(w')) - \sum_{i\in I_g} b_i(f_i(w) - f_i(w'))\right| \leq 2\left(\sum_{i\in I_g} b_i\right) rS_\varepsilon. \tag{21}$$

Algorithm 3 Alternate algorithm for downweighting outliers.

```
procedure UpdateWeights(c, ŵ_{1:n}, Ŷ)
    τ ← S_ε(3√tr(Ŷ) + 9r)
    for i = 1, ..., n do
        Let w̃_i be the solution to (5) as in Algorithm 2.
        Let z_i ← max(f_i(w̃_i) − f_i(ŵ_i) − τ, 0)
    end for
    z_max ← max{z_i | c_i ≠ 0}
    c'_i ← c_i · (z_max − z_i)/z_max for i = 1, ..., n
    return c'
end procedure
```

Relative to Lemma 4.3, the main differences are: The bounds now hold for any weights b_i (with $\hat{w}_{\text{avg}}$ replaced by $\hat{w}^b_{\text{avg}}$), and both (20) and (21) have been strengthened in some minor ways relative to (11) and (12) — in (20) we are now bounding the linearization $\langle\nabla f_i(\hat{w}^b_{\text{avg}}), \hat{w}^b_{\text{avg}} - \hat{w}_i\rangle$, and (21) holds at all w, w' instead of just $\hat{w}_{\text{avg}}, w^*$. These latter strengthenings are trivial and also hold in Lemma 4.3, but were omitted earlier for simplicity. The important difference is that the inequalities hold for any b_i, rather than just for the original weights c_i.

It is perhaps unsatisfying that (20) holds relative to $\hat{w}^b_{\text{avg}}$, rather than $\hat{w}_{\text{avg}}$. Fortunately, by exploiting the fact that $\hat{w}_{\text{avg}}$ is nearly optimal for $\bar{f}$, we can replace $\hat{w}^b_{\text{avg}}$ with $\hat{w}_{\text{avg}}$ at the cost of a slightly weaker bound:

Corollary 5.3. *Let the weights $b_i \in [0,1]$ satisfy $\sum_{i\in I_g} b_i \geq \varepsilon\alpha n$, and suppose that $\sum_{i\in I_g} c_i \geq \frac{1}{2}\alpha n$. Then the solution $\hat{w}_{1:n}, \hat{Y}$ to (8) satisfies*

$$\sum_{i\in I_g} b_i(f_i(\hat{w}_{\text{avg}}) - f_i(\hat{w}_i)) \leq \left(\sum_{i\in I_g} b_i\right)\left(S_\varepsilon\left(3\sqrt{\text{tr}(\hat{Y})} + 9r\right) + \frac{2\lambda r^2}{\alpha n}\right). \tag{22}$$

In particular, if $\text{tr}(\hat{Y}) = O\left(\frac{r^2}{\alpha}\right)$ and $\lambda = O\left(\sqrt{\alpha}nS_\varepsilon/r\right)$, then for all but $\varepsilon\alpha n$ values of $i \in I_g$ we have $f_i(\hat{w}_{\text{avg}}) - f_i(\hat{w}_i) \leq O\left(S_\varepsilon r/\sqrt{\alpha}\right)$.

Corollary 5.3 shows that no matter what the adversary does, the function errors $f_i(\hat{w}_{\text{avg}}) - f_i(\hat{w}_i)$ will be relatively tightly concentrated (at least assuming S_ε is small; we will address the typical size of S_ε later). Looking ahead, we will also be able to show that, in the case that the f_i are strongly convex, $\|w_i - w^*\|_2^2$ is also small for almost all $i \in I_g$. We give this result as Lemma 6.3 in Section 6.

Preserving inliers. Our outlier removal step can be modified based on S_ε so that almost none of the good points are removed. This is not strictly necessary for any of our later results, but is an intuitively appealing property for our algorithm to have. That we can preserve the good points is unsurprising in light of Corollary 5.3, which says that the good points concentrate, and hence should be cleanly separable from any outliers. The modified outlier removal step is given as Algorithm 3.

Algorithm 3 is almost identical to Algorithm 2. The only difference from Algorithm 2 is that, instead of setting z_i to $f_i(\tilde{w}_i) - f_i(\hat{w}_i)$, we set z_i to $\max(f_i(\tilde{w}_i) - f_i(\hat{w}_i) - \tau, 0)$ for an appropriately chosen τ. This creates a buffer such that we do not start to downweight points until their function error passes the threshold τ, which helps

to make sure that very little mass is removed from the good points (because we do not start to take away mass until we are fairly sure that a point is bad). Formally, we have the following result for Algorithm 3, which is analogous to Lemma 4.5 for Algorithm 2:

LEMMA 5.4. *Suppose that* $\lambda = \frac{\sqrt{8\alpha}nS}{r}$ *and* $\mathrm{tr}(\hat{Y}) > \frac{35r^2}{\alpha}$. *Then, the update step in Algorithm 3 satisfies*

$$\sum_{i\in I_g} c_i - c_i' \le \frac{\varepsilon\alpha}{2}\sum_{i=1}^n c_i - c_i'. \tag{23}$$

This shows that the rate at which mass is removed from I_g is at most $\frac{\varepsilon}{2}$ the rate at which mass is removed overall.

Interpreting S_ε. We end this section by giving some intuition for the typical scale of S_ε. Recall that Lemma 2.1 shows that, when the gradients of f_i are sub-Gaussian with parameter σ, then $S \le O(\sigma)$ assuming $n \gg d/\alpha$. A similar bound holds for S_ε, with an additional factor of $\sqrt{\log(2/\varepsilon)}$:

LEMMA 5.5. *If* $\nabla f_i(w) - \nabla\bar{f}(w)$ *is* σ*-sub-Gaussian and L-Lipschitz for all* w*, then with probability* $1-\delta$ *we have*

$$S_\varepsilon = O\left(\sigma\left(\sqrt{\log(2/\varepsilon)} + \sqrt{\frac{d\max(1,\log(rL/\sigma)) + \log(1/\delta)}{\varepsilon\alpha n}}\right)\right). \tag{24}$$

In particular, if $n \ge \frac{1}{\varepsilon\alpha}(d\max(1,\log(rL/\sigma)) + \log(1/\delta))$, *then* $S_\varepsilon = O\left(\sigma\sqrt{\log(2/\varepsilon)}\right)$ *with probability* $1-\delta$, *where* $O(\cdot)$ *masks only absolute constants.*

Lemma 5.5 together with Corollary 5.3 show that, if the gradients of f_i are sub-Gaussian, then the errors between $f_i(\hat{w}_{\mathrm{avg}})$ and $f_i(\hat{w}_i)$ are also sub-Gaussian, in the sense that the fraction of i for which $f_i(\hat{w}_{\mathrm{avg}}) - f_i(\hat{w}_i) \ge \Omega(\sigma\sqrt{\log(2/\varepsilon)/\alpha})$ is at most ε. Inverting this, for sufficiently large t the fraction of i for which $f_i(\hat{w}_{\mathrm{avg}}) - f_i(\hat{w}_i) \ge t\sigma/\sqrt{\alpha}$ is at most $\exp(-\Omega(t^2))$. In other words, no matter what the adversary does, it cannot prevent the function errors from concentrating in a sub-Gaussian manner, provided the good data itself is sub-Gaussian.

A general Chebyshev bound. What happens if the function errors are not sub-Gaussian, but we still have a bound on $S = S_1$? We can then bound S_ε in terms of S by exploiting the monotonicity of the operator norm.

LEMMA 5.6. *For any* $\varepsilon_1 \le \varepsilon_2$, $S_{\varepsilon_1} \le \sqrt{\frac{\lfloor\varepsilon_2\alpha n\rfloor}{\lfloor\varepsilon_1\alpha n\rfloor}}S_{\varepsilon_2} \le 2\sqrt{\frac{\varepsilon_2}{\varepsilon_1}}S_{\varepsilon_2}$.

When coupled with Corollary 5.3, this shows that the function errors concentrate in a Chebyshev-like manner: The fraction of i for which $f_i(\hat{w}_{\mathrm{avg}}) - f_i(\hat{w}_i)$ exceeds $\Omega\left(\sigma/\sqrt{\alpha\varepsilon}\right)$ is at most ε, and so the fraction of i for which $f_i(\hat{w}_{\mathrm{avg}}) - f_i(\hat{w}_i) \ge t\sigma/\sqrt{\alpha}$ is $O\left(\frac{1}{t^2}\right)$. Note that this is already a strengthening of the naïve bound from Markov's inequality, which would only say that the fraction is $O\left(\frac{1}{t}\right)$. The local Hölder's inequality in Lemma 5.1 thus leads to a tighter analysis even without any further bounding of S_ε.

6 BOUNDS FOR STRONGLY CONVEX LOSSES

We now turn our attention to the special case that the functions f_i are strongly convex in w, in the sense that for all $w, w' \in \mathcal{H}$,

$$f_i(w') \ge \langle w' - w, \nabla f_i(w)\rangle + \frac{\kappa}{2}\|w' - w\|_2^2. \tag{25}$$

In this case, we will obtain stronger bounds by iteratively clustering the output $\hat{w}_{1:n}$ of Algorithm 1 and re-running the algorithm on each cluster. The main theorem in this section is a recovery result in the list decoding model, for an algorithm (Algorithm 4) that formalizes this clustering intuition:

THEOREM 6.1. *Suppose that* $\varepsilon \le \frac{1}{2}$ *and let* $\mathcal{U}$, $\hat{w}_{1:n}$ *be the output of Algorithm 4. Then* $\mathcal{U}$ *has size at most* $\lfloor\frac{1}{(1-\varepsilon)\alpha}\rfloor$, *and* $\min_{u\in\mathcal{U}}\|u - w^*\|_2 \le O\left(\frac{S_\varepsilon\sqrt{\log(2/\alpha)}}{\kappa\sqrt{\alpha}}\right)$. *Moreover,* $\|\hat{w}_i - w^*\|_2 \le O\left(\frac{S_\varepsilon\sqrt{\log(2/\alpha)}}{\kappa\sqrt{\alpha}}\right)$ *for all but* $\varepsilon\alpha n$ *values of* $i \in I_g$.

Note, interestingly, that the bound does not depend on the radius r. Since the list decoding model can be reduced to the semi-verified model, Theorem 6.1 also yields strengthened results in the semi-verified model when the functions are strongly convex (we omit these for brevity).

Algorithm and proof overview. Algorithm 4 works at a high level as follows: first, run Algorithm 1 to obtain $\hat{w}_i$ that are (as we will show in Proposition 6.2) relatively close to w^* for most $i \in I_g$. Supposing that the good $\hat{w}_i$ are within distance $r' \ll r$ of w^*, we can cluster $\hat{w}_{1:n}$ into balls of radius $\tilde{O}(r')$, and re-run Algorithm 1 on each cluster; Theorem 4.1 will now yield bounds in terms of r' instead of r. By repeating this enough times, we can shrink our hypothesis space to a small ball around w^*, thus obtaining substantially better bounds. A key piece of machinery which will allow us to obtain a satisfactory clustering is the notion of a *padded decomposition*, originally due to Fakcharoenphol et al. (2003), which we explain in more detail later in this section.

Pseudocode for Algorithm 4 is provided above: We keep track of an upper bound $r^{(t)}$ on the distance from the $\hat{w}_i$ to w^*, which is initially r and decreases by a factor of 2 each time. If this radius drops below a threshold, then we perform a final greedy clustering and exit. Otherwise we use padded decompositions to cluster the points, and run Algorithm 4 on each cluster to obtain new assignments for each $\hat{w}_i$ (since the padded decomposition is randomized, we repeat this several times to ensure correctness with high probability). We can show (Lemma 6.6) that these new assignments $\hat{w}_i$ will be within distance $\frac{1}{2}r^{(t)}$ to w^* for almost all $i \in I_g$, which is the key to proving correctness of the algorithm.

The rest of this section consists of three parts: First, we will show that if the f_i are strongly convex, and $\hat{w}_{1:n}$ is the output of Algorithm 1, then $\|\hat{w}_i - w^*\|_2$ is small for most $i \in I_g$ (this requires some work, since applying Theorem 4.1 directly would only imply that $\|\hat{w}_{\mathrm{avg}} - w^*\|_2$ is small). Next, we will introduce the notion of a padded decomposition, and show (following ideas in Fakcharoenphol et al. (2003)) that padded decompositions of small diameter exist in our setting. Finally, we will combine these two results to analyze Algorithm 4 and establish Theorem 6.1.

Establishing concentration of $\|\hat{w}_i - w^*\|_2$. We will first show that $\hat{w}_i$ is close to w^* for almost all $i \in I_g$:

PROPOSITION 6.2. *For some absolute constant* C *and for any* $\omega \ge 1$, *the output* $\hat{w}_{1:n}$ *of Algorithm 1 satisfies* $\|\hat{w}_i - w^*\|_2^2 \le C\omega\cdot\frac{rS_\varepsilon}{\kappa\sqrt{\alpha}}$ *for all but* $\frac{\varepsilon\alpha n}{\omega^2}$ *values of* $i \in I_g$.

Algorithm 4 Iterative clustering algorithm for approximating w^*

$B(u;s)$ denotes the ball of radius s centered at u
$A(u;s)$ denotes the output of Algorithm 1 with hypothesis space $\mathcal{H} \cap B(u;s)$, radius s, origin shifted to u
procedure FindClusters
  $\hat{w}_{1:n}^{(1)} \leftarrow A(0;r)$, $r^{(1)} \leftarrow r$ ▷ initialize $\hat{w}$, r
  for $t = 1, 2, \ldots$ **do**
    $\mathcal{W} \leftarrow \{\hat{w}_i^{(t)} \mid \hat{w}_i^{(t)} \text{ is assigned}\}$
    if $r^{(t)} < C_1 \cdot S_\varepsilon \log(2/\alpha)/(\kappa\sqrt{\alpha})$ **then** ▷ clean up and exit
      Let $r_{\text{final}} = C_2 \cdot S_\varepsilon \sqrt{\log(2/\alpha)}/(\kappa\sqrt{\alpha})$. Find a maximal set of points $u_1, \ldots, u_m$ such that: ▷ C_1, C_2 are absolute constants
        (i) $|B(u_j; 2r_{\text{final}}) \cap \mathcal{W}| \geq (1-\varepsilon)\alpha n$ for all j, (ii) $\|u_j - u_{j'}\|_2 > 4r_{\text{final}}$ for all $j \neq j'$.
      return $\mathcal{U} = \{u_1, \ldots, u_m\}$ as well as $\hat{w}_{1:n}^{(t)}$.
    end if
    for $h = 1, \ldots, 112\log(t(t+1)/\delta)$ **do**
      $\bar{w}_{1:n}(h) \leftarrow$ unassigned
      Let $\mathcal{P}_h$ be a $(\rho, 2r^{(t)}, \frac{7}{8})$-padded decomposition of $\mathcal{W}$ with $\rho = O\left(r^{(t)}\log(\frac{2}{\alpha})\right)$.
      for each $T \in \mathcal{P}_h$ **do** ▷ run Algorithm 1 on each piece of the decomposition
        Let u be such that $B(u,\rho) \supseteq T$. For i with $\hat{w}_i^{(t)} \in T$, assign $\bar{w}(h)_i$ based on the output of $A(u; \rho + r^{(t)})$.
      end for
    end for
    for $i = 1, \ldots, n$ **do** ▷ find assignment that most $\bar{w}_i(h)$ agree on
      Find a h_0 such that $\|\bar{w}_i(h_0) - \bar{w}_i(h)\|_2 \leq \frac{1}{3}r^{(t)}$ for at least half of the h's.
      Set $\hat{w}_i^{(t+1)} \leftarrow \bar{w}_i(h_0)$ ▷ leave unassigned if h_0 does not exist
    end for
    $r^{(t+1)} \leftarrow \frac{1}{2}r^{(t)}$
  end for
end procedure

The key to establishing Proposition 6.2 lies in leveraging the bound on the statistical error from Lemma 5.2, together with the strong convexity of f_i. Recall that Lemma 5.2 says that for any $b_i \in [0,1]$ satisfying $\sum_{i\in I_g} b_i \geq \varepsilon\alpha n$, we have

$$\sum_{i\in I_g} b_i \langle \nabla f_i(\hat{w}_{\text{avg}}^b), \hat{w}_{\text{avg}}^b - \hat{w}_i\rangle \leq \Big(\sum_{i\in I_g} b_i\Big) S_\varepsilon \Big(\sqrt{\text{tr}(\hat{Y})} + r\Big). \quad (26)$$

By strong convexity of f_i, we then have

$$0 \leq \sum_{i\in I_g} b_i \left(f_i(\hat{w}_{\text{avg}}^b) - f_i(\hat{w}_i)\right) \quad (27)$$

$$\leq \sum_{i\in I_g} b_i \left(\langle \nabla f_i(\hat{w}_{\text{avg}}^b), \hat{w}_{\text{avg}}^b - \hat{w}_i\rangle - \frac{\kappa}{2}\|\hat{w}_i - \hat{w}_{\text{avg}}^b\|_2^2\right) \quad (28)$$

$$\leq \Big(\sum_{i\in I_g} b_i\Big) S_\varepsilon \Big(\sqrt{\text{tr}(\hat{Y})} + r\Big) - \frac{\kappa}{2}\sum_{i\in I_g} b_i \|\hat{w}_i - \hat{w}_{\text{avg}}^b\|_2^2. \quad (29)$$

Therefore:

Lemma 6.3. *For any $b_i \in [0,1]$ satisfying $\sum_{i\in I_g} b_i \geq \varepsilon\alpha n$,*

$$\frac{\sum_{i\in I_g} b_i \|\hat{w}_i - \hat{w}_{\text{avg}}^b\|_2^2}{\sum_{i\in I_g} b_i} \leq \frac{2}{\kappa}\left(\sqrt{\text{tr}(\hat{Y})} + r\right) S_\varepsilon. \quad (30)$$

By applying Lemma 6.3 to $b_i' = \frac{1}{2}\left(b_i + \frac{\sum_j b_j}{\sum_j c_j} c_i\right)$, we obtain the following, which gives bounds in terms of $\hat{w}_{\text{avg}}$ rather than $\hat{w}_{\text{avg}}^b$:

Corollary 6.4. *For any $b_i \in [0,1]$ satisfying $\varepsilon\alpha n \leq \sum_{i\in I_g} b_i \leq \sum_{i\in I_g} c_i$, we have*

$$\frac{\sum_{i\in I_g} b_i \|\hat{w}_i - \hat{w}_{\text{avg}}\|_2^2}{\sum_{i\in I_g} b_i} \leq \frac{16}{\kappa}\left(\sqrt{\text{tr}(\hat{Y})} + r\right) S_\varepsilon. \quad (31)$$

In particular, $\|\hat{w}_i - \hat{w}_{\text{avg}}\|_2^2 \geq \frac{16}{\kappa}\Big(\sqrt{\text{tr}(\hat{Y})} + r\Big) S_\varepsilon$ for at most $\varepsilon\alpha n$ points $i \in I_g$.

Corollary 6.4 is crucial because it shows that all but an ε fraction of the $\hat{w}_i$, for $i \in I_g$, concentrate around $\hat{w}_{\text{avg}}$.

Note that we also have $\|\hat{w}_{\text{avg}} - w^*\|_2^2 \leq \frac{2}{\kappa}\left(\bar{f}(\hat{w}_{\text{avg}}) - \bar{f}(w^*)\right)$, which is bounded by Theorem 4.1; moreover, Theorem 4.1 also bounds $\text{tr}(\hat{Y})$. Finally, we have $S_{\varepsilon/\omega^2} \leq 2\omega S_\varepsilon$ by Lemma 5.6. Combining all of these inequalities, we can obtain Proposition 6.2.

Padded decompositions. Proposition 6.2 says that the output $\hat{w}_{1:n}$ of Algorithm 1 satisfies $\|\hat{w}_i - w^*\|_2 \leq s$ for almost all $i \in I_g$, for some $s \ll r$. We would ideally like to partition the $\hat{w}_i$ into sets of small diameter (say $2s$), such that all of I_g is in a single piece of the partition (so that we can then run Algorithm 4 on each piece of the partition, and be guaranteed that at least one piece has most of I_g).

In general, this may not be possible, but we can obtain a probabilistic version of the hoped for result: We will end up finding a partition into sets of diameter $O(s\log(2/\alpha))$ such that, with probability $\frac{7}{8}$, all of I_g is in a single piece of the partition. This leads us to the definition of a padded decomposition:

Definition 6.5. Let $x_1, \ldots, x_n$ be points in a metric space. A (ρ, τ, δ)-*padded decomposition* is a (random) partition $\mathcal{P}$ of the set $\{x_1, \ldots, x_n\}$ such that: (i) each element of $\mathcal{P}$ has diameter at most ρ, and (ii) for each x_i, with probability $1 - \delta$ all points within distance τ of x_i lie in a single element of $\mathcal{P}$.

Fakcharoenphol et al. (2003) show that for any τ and δ, a (ρ, τ, δ)-padded decomposition exists with $\rho = O\left(\frac{\tau \log(n)}{\delta}\right)$. Moreover, the same proof shows that, if every x_i is within distance τ of at least $\Omega(\alpha n)$ other x_i, then we can actually take $\rho = O\left(\frac{\tau \log(2/\alpha)}{\delta}\right)$. In particular, in our case we can obtain a $(O(s \log(2/\alpha)), 2s, 7/8)$-padded decomposition of the $\hat{w}_i$ output by Algorithm 1; see the full paper for details. This probabilistic notion of clustering turns out to be sufficient for our purposes.

Analyzing Algorithm 4. We are now prepared to analyze Algorithm 4. In each iteration, Algorithm 4 independently samples $l = 112 \log(t(t+1)/\delta)$ padded decompositions of the $\hat{w}_i$. For each decomposition $\mathcal{P}_h$ $(h = 1, \ldots, l)$, it then runs Algorithm 1 on each component of the resulting partition, and thereby obtains candidate values $\bar{w}_1(h), \ldots, \bar{w}_n(h)$. Finally, it updates $\hat{w}_i$ by finding a point close to at least $\frac{1}{2}$ of the candidate values $\bar{w}_i(h)$, across $h = 1, \ldots, l$.

The idea for why this works is the following: since $\frac{7}{8}$ of the time, the padded decomposition $\mathcal{P}_h$ succeeds in preserving I_g, it is also the case that roughly $\frac{7}{8}$ of the candidate assignments $\bar{w}_i(h)$ are "good" assignments close to w^*. Therefore, any point that is close to at least $\frac{1}{2}$ of the $\bar{w}_i(h)$ is, by pigeonhole, close to one of the good $\bar{w}_i(h)$, and therefore also close to w^*.

If we formalize this argument, we obtain Lemma 6.6, which controls the behavior of the update $\hat{w}_{1:n}^{(t)} \to \hat{w}_{1:n}^{(t+1)}$ on each iteration:

Lemma 6.6. *Algorithm 4 satisfies the following property: at the beginning of iteration t of the outer loop, let $I_g^{(t)}$ denote the set of points $i \in I_g$ for which $\|\hat{w}_i^{(t)} - w^*\|_2 \le r^{(t)}$. Also suppose that $|I_g^{(t)}| \ge (1-\varepsilon)\alpha n$ and $\varepsilon \le \frac{1}{2}$. Then, with probability $1 - \frac{\delta}{t(t+1)}$ over the randomness in the padded decompositions, $\|\hat{w}_i^{(t+1)} - w^*\|_2 \le \frac{1}{2} r^{(t)}$ for all but $C_0 \cdot \left(\frac{S_\varepsilon \log(2/\alpha)}{\kappa r^{(t)} \sqrt{\alpha}}\right)^2 \cdot \varepsilon\alpha n$ points in $I_g^{(t)}$, for some absolute constant C_0.*

Essentially, Lemma 6.6 shows that if almost all of the good points are within $r^{(t)}$ of w^* at the beginning of a loop iteration, then almost all of the good points are within $\frac{1}{2} r^{(t)}$ of w^* at the end of that loop iteration, provided $r^{(t)}$ is large enough. Using Lemma 6.6, we can establish Theorem 6.1; see the full paper for details.

7 LOWER BOUNDS

We now prove lower bounds showing that the dependence of our bounds on S is necessary even if p^* is a multivariate Gaussian. For α, we are only able to show a necessary dependence of $\sqrt{\log(\frac{1}{\alpha})}$, rather than the $\sqrt{1/\alpha}$ appearing in our bounds. Determining the true worst-case dependence on α is an interesting open problem.

One natural question is whether S, which typically depends on the maximum singular value of the covariance, is really the right dependence, or whether we could achieve bounds based on e.g. the average singular value instead. Lemma 7.1 rules this out, showing that it would require $\Omega(2^k)$ candidates in the list-decodable setting to achieve dependence on even the kth singular value.

Lemma 7.1. *Suppose that p^* is known to be a multivariate Gaussian $\mathcal{N}(\mu, \Sigma)$ with covariance $\Sigma \le \Sigma_0$, where μ and Σ are otherwise unknown. Also let σ_k^2 denote the kth largest singular value of Σ_0. Then, given any amount of data, an α-fraction of which is drawn from p^*, any procedure for outputting at most $m = 2^{k-1}$ candidate means $\hat{\mu}_1, \ldots, \hat{\mu}_m$ must have, with probability at least $\frac{1}{2}$, $\min_{j=1}^m \|\hat{\mu}_j - \mu\|_2 \ge \frac{\sigma_k}{4} \cdot \frac{\log(\frac{1}{\alpha})}{\sqrt{1+\log(\frac{1}{\alpha})}}$.*

By the reduction from the semi-verified to the list-decodable model, we obtain a lower bound in the semi-verified setting as well; see the full paper for details. We remark that the same proofs show lower bounds for non-strongly convex losses as well.

Proof of Lemma 7.1. Let us suppose that the unverified data has distribution $\hat{p} = \mathcal{N}(0, \Sigma_0)$, which is possible iff $\hat{p} \ge \alpha p^*$. We start with a lemma characterizing when it is possible that the true distribution p^* has mean μ:

Lemma 7.2. *Let $\hat{p} = \mathcal{N}(0, \Sigma_0)$ and $p^* = \mathcal{N}(\mu, \Sigma_0 - \lambda\mu\mu^\top)$. Then $\hat{p} \ge \alpha p^*$ provided that $\mu^\top \Sigma_0^{-1} \mu \le \frac{\log^2(\frac{1}{\alpha})}{1+\log(\frac{1}{\alpha})}$ and $\lambda = \frac{1}{\log(\frac{1}{\alpha})}$.*

As a consequence, if $\mu^\top \Sigma_0^{-1} \mu \le t^2$ for $t = \frac{\log(\frac{1}{\alpha})}{\sqrt{1+\log(\frac{1}{\alpha})}}$, then p^* could have mean μ. Now, consider the space spanned by the k largest eigenvectors of Σ_0, and let B_k be the ball of radius $t\sigma_k$ in this space. Also let $\mathcal{P}$ be a maximal packing of B_k of radius $\frac{t}{4}\sigma_k$. A simple volume argument shows that $|B_k| \ge 2^k$. On the other hand, every element μ of B_k is a potential mean because it satisfies $\mu^\top \Sigma_0^{-1} \mu \le \frac{\|\mu\|_2^2}{\sigma_k^2} \le t^2$. Therefore, if the true mean μ is drawn uniformly from B_k, any 2^{k-1} candidate means $\hat{\mu}_j$ must miss at least half of the elements of B_k (in the sense of being distance at least $\frac{t}{4}\sigma_k$ away) and so with probability at least $\frac{1}{2}$, $\min_{j=1}^m \|\hat{\mu}_j - \mu\|_2$ is at least $\frac{t}{4}\sigma_k$, as was to be shown.

8 INTUITION: STABILITY UNDER SUBSETS

In this section, we establish a sort of "duality for robustness" that provides intuition underlying our results. Essentially, we will show the following:

> *If a statistic of a dataset is approximately preserved across every large subset of the data, then it can be robustly recovered even in the presence of a large amount of additional arbitrary/adversarial data.*

To be a bit more formal, suppose that for a set of points $\{x_1, \ldots, x_n\}$ lying in $\mathbb{R}^d$, there is a subset $I \subseteq [n]$ with the following property: For any subset $T \subseteq I$ of size at least $\frac{1}{2}\alpha^2 n$, the mean over T is ε-close to the mean over I. In symbols,

$$\|\mu_T - \mu_I\|_2 \le \varepsilon \text{ for all } T \subseteq I \text{ with } |T| \ge \frac{1}{2}\alpha^2 n, \tag{32}$$

where $\mu_T \stackrel{\text{def}}{=} \frac{1}{|T|}\sum_{i \in T} x_i$. In such a case, can we approximate the mean of I, even if I is unknown and the points in $[n] \backslash I$ are arbitrary?

If we do not care about computation, the answer is yes, via a simple exponential-time algorithm. Call a set of points J α-stable if it satisfies the following properties: $|J| \geq \alpha n$, and (32) holds with I replaced by J (i.e., the mean moves by at most ε when taking subsets of J of size at least $\frac{1}{2}\alpha^2 n$). Then, we can run the following (exponential-time) algorithm:

(1) Initialize $\mathcal{U} = []$.
(2) Find an α-stable set J which has overlap at most $\frac{1}{2}\alpha^2 n$ with all elements of $\mathcal{U}$.
(3) Append J to $\mathcal{U}$ and continue until no more such sets exist.

A simple inclusion-exclusion argument shows that $k \leq \frac{2}{\alpha}$. Therefore, the above algorithm terminates with $|\mathcal{U}| \leq \frac{2}{\alpha}$. On the other hand, by construction, I must have overlap at least $\frac{1}{2}\alpha^2 n$ with at least one element of $\mathcal{U}$ (as otherwise we could have also added I to $\mathcal{U}$). Therefore, for some $J \in \mathcal{U}$, $|I \cap J| \geq \frac{1}{2}\alpha^2 n$. But then, letting $T = I \cap J$, we have $\|\mu_I - \mu_J\|_2 \leq \|\mu_I - \mu_T\|_2 + \|\mu_T - \mu_J\|_2 \leq 2\varepsilon$. Therefore, the mean over I is within distance 2ε of at least one of the μ_J, for $J \in \mathcal{U}$.

We have therefore established our stated duality property: *if a statistic is stable under taking subsets of data, then it can also be recovered if the data is itself a subset of some larger set of data.*

Can we make the above algorithm computationally efficient? First, can we even *check* if a set J is α-stable? This involves checking the following constraint:

$$\Big\| \sum_{i \in J} c_i (x_i - \mu_J) \Big\|_2 \leq \varepsilon \|c\|_1 \text{ if } c_i \in \{0,1\} \text{ and } \sum_{i \in J} c_i \geq \frac{1}{2}\alpha^2 n. \quad (33)$$

It is already unclear how to check this constraint efficiently. However, defining the matrix $C_{ij} = c_i c_j \in [0,1]^{J \times J}$, we can take a semidefinite relaxation of C, resulting in the constraints $C_{ij} \in [0,1]$ and $\operatorname{tr}(C) \geq \frac{1}{2}\alpha^2 n$. Letting $A_{ij} = (x_i - \mu_J)^\top (x_j - \mu_J)$, this results in the following sufficient condition for α-stability:

$$\operatorname{tr}(A^\top C) \leq \varepsilon^2 \operatorname{tr}(C)^2 \text{ for all } C \text{ such that}$$
$$C \succeq 0, C_{ij} \in [0,1], \operatorname{tr}(C) \geq \frac{1}{2}\alpha^2 n. \quad (34)$$

This is still non-convex due to the presence of $\operatorname{tr}(C)^2$. However, a sufficient condition for (34) to hold is that $\operatorname{tr}(A^\top C) \leq \frac{1}{2}\varepsilon^2\alpha^2 n \operatorname{tr}(C)$ for all $C \succeq 0$, which is equivalent to $\|A\|_{\text{op}} \leq \frac{1}{2}\varepsilon^2\alpha^2 n$, or equivalently $\varepsilon \geq \sqrt{\frac{2}{\alpha}} \cdot \left(\frac{1}{\sqrt{\alpha n}}\|X\|_{\text{op}}\right)$, where X is the matrix with rows $x_i - \mu_J$. Noting that $S = \|X\|_{\text{op}}/\sqrt{\alpha n}$, we can see why $S/\sqrt{\alpha}$ appears in our bounds: it is a convex relaxation of the α-stable condition.

REFERENCES

E. Abbe and C. Sandon. Community detection in general stochastic block models: fundamental limits and efficient recovery algorithms. *arXiv*, 2015a.
E. Abbe and C. Sandon. Detection in the stochastic block model with multiple clusters: proof of the achievability conjectures, acyclic BP, and the information-computation gap. *arXiv*, 2015b.
D. Achlioptas and F. McSherry. On spectral learning of mixtures of distributions. In *COLT*, 2005.
N. Agarwal, A. S. Bandeira, K. Koiliaris, and A. Kolla. Multisection in the stochastic block model using semidefinite programming. *arXiv*, 2015.
P. Awasthi and O. Sheffet. Improved spectral-norm bounds for clustering. *Approximation, Randomization, and Combinatorial Optimization*, pages 37–49, 2012.
P. Awasthi, M. F. Balcan, and P. M. Long. The power of localization for efficiently learning linear separators with noise. In *STOC*, pages 449–458, 2014.
M. Balcan, A. Blum, and S. Vempala. A discriminative framework for clustering via similarity functions. In *STOC*, pages 671–680, 2008.
M. F. Balcan, H. Röglin, and S. Teng. Agnostic clustering. In *ALT*, pages 384–398, 2009.
J. Batson, D. A. Spielman, and N. Srivastava. Twice-Ramanujan sparsifiers. *SIAM J. Comput.*, 41(6):1704–1721, 2012.
K. Bhatia, P. Jain, and P. Kar. Robust regression via hard thresholding. In *NIPS*, pages 721–729, 2015.
A. Blum and J. Spencer. Coloring random and semi-random k-colorable graphs. *Journal of Algorithms*, 19(2):204–234, 1995.
T. T. Cai and X. Li. Robust and computationally feasible community detection in the presence of arbitrary outlier nodes. *The Ann. Stat.*, 43(3):1027–1059, 2015.
E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. ACM*, 58(3), 2011.
V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM J. Optimiz.*, 21(2):572–596, 2011.
O. Chapelle, A. Zien, and B. Scholkopf. *Semi-Supervised Learning*. MIT Press, 2006.
Y. Chen, A. Jalali, S. Sanghavi, and H. Xu. Clustering partially observed graphs via convex optimization. *JMLR*, 15:2213–2238, 2014a.
Y. Chen, S. Sanghavi, and H. Xu. Improved graph clustering. *IEEE Transactions on Information Theory*, 60(10):6440–6455, 2014b.
A. Coja-Oghlan. Coloring semirandom graphs optimally. *Automata, Languages and Programming*, pages 71–100, 2004.
A. Coja-Oghlan. Solving NP-hard semirandom graph problems in polynomial expected time. *Journal of Algorithms*, 62(1):19–46, 2007.
I. Diakonikolas, G. Kamath, D. Kane, J. Li, A. Moitra, and A. Stewart. Robust estimators in high dimensions without the computational intractability. In *FOCS*, 2016.
J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *STOC*, pages 448–455, 2003.
U. Feige and J. Kilian. Heuristics for semirandom graph problems. *Journal of Computer and System Sciences*, 63(4):639–671, 2001.
U. Feige and R. Krauthgamer. Finding and certifying a large hidden clique in a semirandom graph. *Random Structures and Algorithms*, 16(2):195–208, 2000.
V. Feldman, P. Gopalan, S. Khot, and A. K. Ponnuswami. On agnostic learning of parities, monomials, and halfspaces. *SIAM J. Comput.*, 39(2):606–645, 2009.
O. Guédon and R. Vershynin. Community detection in sparse networks via Grothendieck's inequality. *arXiv*, 2014.
V. Guruswami and P. Raghavendra. Hardness of learning halfspaces with noise. *SIAM J. Comput.*, 39(2):742–765, 2009.
F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. Wiley, 2011.
M. Hardt and A. Moitra. Algorithms and hardness for robust subspace recovery. In *COLT*, 2013.
P. J. Huber and E. M. Ronchetti. *Robust Statistics*. Wiley, 2009.
M. Kearns and M. Li. Learning in the presence of malicious errors. *SIAM J. Comput.*, 22(4):807–837, 1993.
A. R. Klivans, P. M. Long, and R. A. Servedio. Learning halfspaces with malicious noise. *JMLR*, 10:2715–2740, 2009.
M. Krivelevich and D. Vilenchik. Semirandom models as benchmarks for coloring algorithms. In *Meeting on Analytic Algorithmics and Combinatorics*, pages 211–221, 2006.
A. Kumar and R. Kannan. Clustering with spectral norm and the k-means algorithm. In *FOCS*, pages 299–308, 2010.
S. Kushagra, S. Samadi, and S. Ben-David. Finding meaningful cluster structure amidst background noise. In *ALT*, pages 339–354, 2016.
K. A. Lai, A. B. Rao, and S. Vempala. Agnostic estimation of mean and covariance. In *FOCS*, 2016.
C. M. Le, E. Levina, and R. Vershynin. Concentration and regularization of random graphs. *arXiv*, 2015.
K. Makarychev, Y. Makarychev, and A. Vijayaraghavan. Approximation algorithms for semi-random partitioning problems. In *STOC*, pages 367–384, 2012.
K. Makarychev, Y. Makarychev, and A. Vijayaraghavan. Learning communities in the presence of errors. *arXiv*, 2015.
F. McSherry. Spectral partitioning of random graphs. In *FOCS*, pages 529–537, 2001.
A. Moitra, W. Perry, and A. S. Wein. How robust are reconstruction thresholds for community detection? *arXiv*, 2015.
E. Rebrova and K. Tikhomirov. Coverings of random ellipsoids, and invertibility of matrices with iid heavy-tailed entries. *arXiv*, 2015.
E. Rebrova and R. Vershynin. Norms of random matrices: local and global problems. *arXiv*, 2016.
M. Rudelson. Random vectors in the isotropic position. *J. Funct. Anal.*, 164:60–72, 1999.
J. Steinhardt, G. Valiant, and M. Charikar. Avoiding imposters and delinquents: Adversarial crowdsourcing and peer prediction. In *NIPS*, 2016.
J. W. Tukey. A survey of sampling from contaminated distributions. *Contributions to probability and statistics*, 2:448–485, 1960.
H. Xu, C. Caramanis, and S. Mannor. Principal component analysis with contaminated data: The high dimensional case. *arXiv*, 2010.
H. Zha, X. He, C. Ding, H. Simon, and M. Gu. Spectral relaxation for k-means clustering. In *NIPS*, pages 1057–1064, 2001.