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 \( \alpha n \) 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 \( \text{poly}(1/\alpha) \) 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 \( \alpha \)-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.

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 \( \alpha n \) 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 \( \alpha \) 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 \((m, \varepsilon)\) 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 \( \varepsilon \).

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 \( \alpha \)-fraction of real data, in what settings is it possible to efficiently output a list of at most \( \frac{1}{\alpha} \) or \( \text{poly}(\frac{1}{\alpha}) \) 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 \( \alpha \)
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 $an$ are "real" data reflecting an underlying distribution $p^*$, and the remaining $(1−a)n$ 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 $\alpha$ 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 $an$ are sampled from $p^*$, and we want to minimize the population mean $f = \mathbb{E}_p[f_i]$. Our results are given in terms of a spectral norm bound on the gradients $\nabla f_i$. Therefore, we obtain robustness in any setting where we can establish a matrix concentration inequality on the good data — for instance, if the $\nabla f_i$ 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 $\ell_2$ error $O(\sigma)$, where $\sigma^2$ is a bound on the second moments of $p^*$. For $\alpha$ bounded away from $\frac{1}{2}$, this improves upon existing work, which achieves error either $O(\sigma \sqrt{\log(\delta)})$ 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 $O(\sigma / \sqrt{n})$.

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

STOC’17, June 2017, Montreal, Canada Charikar, Steinhardt, and Valiant
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 $p_{\theta}(x) \propto \exp(\theta^T \phi(x))$, we can output $\theta$ with $\text{KL}(p_{\theta} \parallel p_{\theta'}) \leq O(\frac{\sigma}{\sqrt{n}})$, where $\sigma = \lambda_{\text{max}}(\text{Cov}_{p_{\theta}}[\phi(x)])$ 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 $\sigma^2$, and with minimum mixture weight $\alpha$, we can accurately cluster the points if the means are separated by a distance $\tilde{\Omega}(\sigma/\sqrt{n})$, 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 $\Omega(\sigma \sqrt{k})$ (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 $\tilde{\Omega}(1/\alpha^2)$, where $an$ 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 surprisingly 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 allows for a constant fraction of the partition to be recovered if the remaining data has arbitrary structure. Our clustering scheme is described in Section 6.

**Proposition 1.3.** Suppose that $p$ is a distribution on $\mathbb{R}^d$ with $\mathbb{E}_{p}[X] = \mu$ and $\text{Cov}_{p}[X] \preceq \sigma^2 I$ for some $\sigma$. Then, given $n \geq d$ samples from $p$, with probability $1 - \exp(-\frac{d^2}{2\sigma^2})$ there is a subset $I \subseteq \{1, \ldots, n\}$ of size at least $\frac{n}{2}$ such that $\lambda_{\text{max}}\left(\frac{1}{|I|} \sum_{i \in I} (x_i - \mu)(x_i - \mu)^T\right) \leq 2\sigma^2$, where $\lambda_{\text{max}}$ 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 $\lambda_{\text{max}}$ by $\sigma^2 \log(n)$, 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 $\lambda_{\text{max}}$. 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 $\alpha = 1$, while our work focuses on $\alpha \ll 1$. In the overlap of these regimes (e.g. $\alpha = \frac{1}{4}$) our results improve upon these existing results. (The existing bounds are better as $\alpha \rightarrow 1$, 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 $\alpha$ (say $\alpha \geq \frac{\log d}{d}$) 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 $\alpha \leq \frac{1}{4}$ 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 ($O(\frac{1}{d})$), and the bounds are weak in high dimensions; e.g. Balcan et al. (2008) output a list of $O(k/\gamma^2)$ hypotheses, where $\gamma$ 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 \( \frac{1}{2} \). 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 mixture 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 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 outliers 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 \( \alpha n \), \( f_i \sim p^* \) for \( i \in I_g \), and the remaining \( f_i \) are chosen by an adversary whose strategy can depend on the \( f_i \) for \( i \notin I_g \).

Let \( \bar{f} \) denote the mean of \( f \) under \( p^* \), i.e. \( \bar{f}(w) \defeq \mathbb{E}_{f \sim p^*}[f(w)] \) for \( w \in \mathcal{H} \); our goal is to find a parameter \( w^* \) such that \( \bar{f}(w) - \bar{f}(w^*) \) is small, where \( w^* \) is the minimizer of \( \bar{f} \). We use \( r \) to denote the \( \ell_2 \)-radius of \( \mathcal{H} \), i.e. \( r \defeq \max_{w \in \mathcal{H}} \|w\|_2 \).

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 \( f_i(w) = (y_i - (w, x_i))^2 \), and logistic regression to \( f_i(w) = \log(1 + \exp(-(y_i, x_i))) \).

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:

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

In words, if we form the matrix of gradients \( \nabla f_{i_1}(w) \cdots \nabla f_{i_{|I_g|}}(w) \), where \( \{i_1, \ldots, i_{|I_g|}\} = I_g \), 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 $\max_{w \in \mathcal{H}} \| \frac{1}{n} \sum_{i \in [n]} (\nabla f_i(w) - \nabla f(w)) \|_2$. 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," whereas 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 $x_i \sim N(\mu, \sigma^2 I)$. Then $\nabla f_i(w) = (x_i - \mu)/\sigma$, and so $S$ is simply the maximum singular value of $\frac{1}{\sigma \sqrt{n}} [x_i - \mu]_{i \in [n]}$. This is the square root of the maximum eigenvalue of $\frac{1}{n \sigma^2} [\sum_{i \in [n]} (x_i - \mu)(x_i - \mu)^	op]$, which converges to $\sigma$ for large $n$.

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

The $j$th coordinate of $\nabla f_i(w) - \nabla f(w)$ 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 $D(w) \cdot ([x_i - p_i]_{i \in [n]})/\sqrt{\|f_i(w)\|_2}$, 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 $\|D(w)\|_{\text{op}} \leq 16/3$, while the matrix $[x_i - p_i]/\sqrt{\|f_i(w)\|_2}$ has maximum singular value converging to $\max_{i,j} p_j (1 - p_j) \leq \frac{1}{4}$ for large enough $n$. Thus $S = O(1)$ in this setting.

Linear classification: Suppose that $x_j \sim N(0,1)$ and that $y_j = \text{sign}(u \cdot x_j)$ for some unknown vector $u$. Our loss function is the logistic loss $f_i(w) = \log(1 + e^{-(y_j - w \cdot x_j)})$. In this case $\nabla f_i(w) = \frac{-y_j}{1 + e^{-(y_j - w \cdot x_j)}} x_j$. 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 $\sigma$-sub-Gaussian if $\mathbb{E}[\exp(\mu(X - \mu))] \leq \exp(\frac{\sigma^2}{2} \| \mu \|_2^2)$, where $\mu = \mathbb{E}[X]$. If $\nabla f_i$ is sub-Gaussian, then $S = O(\sigma)$ if $\alpha \geq \Omega(\delta)$.

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

In most of our concrete settings, sub-Gaussianity of $\nabla f_i$ 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 $\hat{f}$. This meta-result underlies our results in the list decoding and semi-verifying settings.

**Theorem 2.2.** Given $n$ data points containing a set $I_0$ of an $\alpha$-data points with spectral norm bound $S$, we can obtain an ellipse $E_Y = \{w \mid w w^T \leq Y \}$ such that $tr(Y) \leq O(\frac{S}{\sqrt{\alpha}})$ and

$$\min_{w \in E_Y} \hat{f}(w) - \hat{f}(\tilde{w}) \leq O\left(\frac{S}{\sqrt{\alpha}}\right).$$

Recall here that $r$ is the $\ell_2$-radius of the parameter space $H$. Also note that when $Y$ is invertible, $w w^T \leq Y$ is equivalent to $w^T Y^{-1} w \leq 1$, 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 $H$ down to the small ellipse $E_Y$.

To interpret the bound (2), consider the mean estimation example above, where $f_i(w) = \frac{1}{2} \| x_i - w \|_2^2$ with $x_i \sim N(\mu, \sigma^2 I)$. Note that $\hat{f}(w) - \hat{f}(\tilde{w}) = \frac{1}{2} \| w - \mu \|_2^2$. Assuming that $\| \mu \|_2$ is known to within a constant factor, we can take $H$ to be the $\ell_2$-ball of radius $r = O(\| \mu \|_2)$. This leads to the bound $\| w - \mu \|_2^2 = O(r^2 \| \mu \|_2^2 / \sqrt{\alpha})$, for some $w$ in an ellipse of trace $\| \mu \|_2^2 / \alpha$. Note that the $\ell_2$-ball itself has trace $d \| \mu \|_2^2$, so the ellipse $E_Y$ is much smaller than $H$ 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 $\| \mu \|_2 \ll d$. Theorem 4.1 is thus doing real work, by finding a $w$ that is much closer to $\mu$ than a randomly chosen $x_i$.

We note that by applying our algorithm multiple times we can improve the bound $\| w - \mu \|_2^2 = O(\sigma^2 / \sqrt{\alpha})$ 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 $\hat{f}(w) - \hat{f}(\tilde{w}) = O(1 / \sqrt{\alpha})$. However, if $\tilde{f}(0) = \log(2)$ while $\hat{f}(0) \geq 0$, so $\hat{f}(0) - \tilde{f}(0) \leq O(1)$ 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 $k$-strongly convex, meaning that $f_i(w') - f_i(w) \geq (w' - w)^\top \nabla f_i(w) + k \| w' - w \|_2^2$. The strong convexity allows us to show that for the good $f_i$, the parameters $\tilde{w}_i = \arg \min_{w \in E_Y} f_i(w)$ concentrate around $w^*$, with radius $r' \ll r$. By clustering the $\tilde{w}_i$ 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 $\hat{w}$ for each cluster. We can thereby show:

**Theorem 2.3.** Suppose the $f_i$ are $k$-strongly convex, and suppose there is a set $I_0$ of size $\alpha$ with spectral norm bound $S$. Then, for any $\epsilon \leq \frac{1}{2}$, it is possible to obtain a list of $m = \lceil \frac{S}{k (1 - \epsilon) \alpha} \rceil$ candidates $\tilde{w}_1, \ldots, \tilde{w}_m$, such that $\min_{j=1}^m \| \tilde{w}_j - w^* \|_2 \leq O\left(\frac{S}{k \sqrt{\alpha}}\right)$.

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 $f(w) = \phi(w^T x)$, where $\phi$ is 1-Lipschitz, and suppose that $x$ has bounded qth moments in the sense that $\mathbb{E}_p[|x - \mathbb{E}[x]|^q]^{1/q} \leq \sigma_q$ for all unit vectors $u$ and some $q \geq 2$. Then given $Y$ from Theorem 2.2 and a single verified $x - p^*$, we can obtain a $\hat{w}$ such that $\mathbb{P}_{x - p^*} \left[ \hat{f}(\hat{w}) \geq \hat{f}(w^*) + C \cdot \frac{(S + \log n)}{\sqrt{d}} \right] \leq \epsilon^{-q}$, 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 $f - \hat{f}$. Note that if we replace $\mathcal{E}_Y$ with $\mathcal{H}$ in Lemma 2.4, then the $r/\sqrt{d}$ dependence becomes $r/\sqrt{d}$, which is usually vacuous.

Optimality? The dependence on $S$, $r$, and $\kappa$ in the results above seems essentially necessary, though the optimal dependence on $\alpha$ 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 $\Omega \left( \frac{k}{\alpha \sqrt{\log(1/\alpha)}} \right)$ for strongly convex $f_i$, and $\Omega \left( r \sqrt{\log(1/\alpha)} \right)$ 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 $\alpha$ 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 $x_1, \ldots, x_n \in \mathbb{R}^d$, of which $\alpha n$ are drawn from a distribution $p^*$ with bounded covariance, and our goal is to recover the mean $\mu = \mathbb{E}_{x - p^*}[x]$. 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: $\text{Cov}_{p^*}[x] \leq \sigma^2 I$. Then, for $n \geq \frac{d}{\alpha}$, with probability $1 - \exp(-\Omega(\alpha n))$ it is possible to output $\hat{\mu} \leq O \left( \frac{1}{\alpha} \right)$ candidate means $\hat{\mu}_1, \ldots, \hat{\mu}_m$ such that $\min_{1 \leq j \leq m} \|\hat{\mu}_j\|_2 \leq O \left( \sqrt{\frac{\log(1/\alpha)}{\alpha}} \right)$. Moreover, if $\alpha \geq 0.51$ 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 $\alpha > 1/2$ and one is required to output a single parameter (i.e., $m = 1$). For simplicity take $\alpha = 1/2$. 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 $n = O(d^3)$, 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 results obtain error that vanishes as $\alpha \to 1$ (at a rate of $O(\sqrt{1 - \alpha})$ in the first case and $O(1 - \alpha)$ 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 $\alpha \leq 1$. 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 $\ell_2$ distance by $\Omega(\sigma / \sqrt{\alpha})$.

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

The $\frac{1}{\alpha}$ dependence can be replaced with $\sqrt{\log(n)/\epsilon}$, or with $\sqrt{\log(2/\epsilon)}$ 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 $O \left( \frac{\alpha}{\sigma} \right)$ 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 $O(\sigma \sqrt{k})$ (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 $I_1, \ldots, I_k$ of $[n]$, we generate edges as follows: (i) If $u, v \in I_i$, then $p(A_{uv} = 1) = \frac{\alpha}{n}$. (ii) If $u \in I_i, v \notin I_i$, then $p(A_{uv} = 1) = \frac{\alpha}{n}$. (iii) If $u \notin \bigcup_{i=1}^k I_i$, 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 $\alpha$ and $k$ 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 $\mathbb{E} = \text{Cov}[x]$ is highly skewed; they roughly show $\|\mu - \mu_{k-1}\|_2 = O(\delta)$. 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 $1/\sqrt{2}$ factor in this stricter setting, which is a constant if $\alpha = 1/4$. 

52
Note that the distribution over the row $A_{i\alpha}$ is the same for all $\alpha \in I_i$. By taking this distribution to be the distribution on $p^*$, Theorem 2.3 yields the following result:

**Corollary 2.7.** For the planted partition model above, let $a = \min_{\alpha \in I_i} |I_i|/|T_i|$. Then, with probability $1 - \exp(-\Omega(|\alpha| n))$, we can obtain sets $T_1, \ldots, T_m \subseteq [n]$, with $m \leq O\left(\frac{|T_i|}{a^2}\right)$, such that for all $i \in [k]$, there is a $j \in [m]$ with $|I_i \Delta T_j| \leq O\left(\frac{a \log \frac{3}{n}}{a^2(a-b)^2}\right) n$, where $\Delta$ denotes symmetric difference.

This shows that we can approximately recover the planted partition, even in the presence of arbitrary corruptions, provided $(a-b)^2 \gg \log(n/a)$ (since the bound on $|I_i \Delta T_j|$ needs to be less than $an$ to be meaningful). In contrast, the best efficient methods (assuming no corruptions) roughly require $(a-b)^2 \gg k$ 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 $a \gg \log((2/a)^2)$ in contrast to $a \gg \frac{1}{2}$.

**Summary.** For robust mean estimation, we match the best existing error bounds of $O(\sigma)$ when $a = \frac{3}{4}$, under weaker assumptions. For learning mixtures distributions, we match the best bound of $O(\sigma \sqrt{k})$ when $a = \Omega(1/k)$. 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 $f_j$ its own parameter vector $w_j$, and minimize $\sum_{i=1}^n f_i(w_j)$ 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 $\hat{f}(w)$ by standard concentration results.

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

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:

$$\text{minimize } \sum_{i=1}^n c_i f_i(w_i) + \lambda \text{tr}(Y)$$

subject to $w_i^\top Y w_i \leq Y$ for all $i = 1, \ldots, n$. (3)

6: if $\text{tr}(\hat{Y}) \leq \frac{\alpha^2}{\sigma}$ then

7: return $\hat{w}_{1:n}, \hat{Y}$

8: else

9: $c \leftarrow \text{UpdateWeights}(c, \hat{w}_{1:n}, \hat{Y})$ Re-weight points to down-weight outliers

10: end if

11: end while

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 $w_i^\top Y w_i \leq Y$ 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 $\text{tr}(Y)$ is related to the trace norm of $[w_1 \ldots w_n]$.

**Lemma 3.1.** For any points $w_1, \ldots, w_n \in \mathbb{R}^d$, suppose that $Y \succeq w_i^\top w_i^\top$ 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 $\text{tr}(Y) \geq \frac{\|W_T\|_F^2}{|T|^2}$ 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 $\text{tr}(Y)$ 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

---

1. It is convex because the constraint $w_i^\top w_i^\top \leq Y$ is equivalent to $[Y w_i^\top w_i^\top] \succeq 0$. 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.

1: procedure UpdateWeights($c$, $\hat{w}_{1:n}$, $\hat{Y}$)
2: for $i = 1, \ldots, n$ do
3: Let $\hat{w}_i$ be the solution to
4: \begin{equation}
5: \begin{array}{l}
6: \text{minimize } f_i(\hat{w}_i) \\
7: \text{subject to } \hat{w}_i = \sum_{j=1}^{n} a_{ij} \hat{w}_j, \\
8: 0 \leq a_{ij} \leq \frac{2}{an}, \sum_{j=1}^{n} a_{ij} = 1.
9: \end{array}
10: \end{equation}
11: end for
12: $z_{\text{max}} \leftarrow \max_{i} \left| z_i \right|$ if $c_i \neq 0$
13: $c_i \leftarrow c_i - \frac{\max_{i} \left| z_i \right| - 2a_i}{z_{\text{max}}} \text{ for } i = 1, \ldots, n$
14: return $c'$
15: end procedure

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

The optimization (6) roughly asks for a parameter $\hat{w}_i$ that minimizes $f_i$, subject to $\hat{w}_i$ being the average of at least $\frac{2}{\alpha n}$ distinct parameters $\hat{w}_j$. Given the solution to $\hat{w}_i$, we then downweight the influence of the $i$th data point based on the value of $f_i(\hat{w}_i) - f_i(\check{w}_i)$. In particular, we will multipy the weight $c_i$ by $1 - \frac{1}{\eta} \left( f_i(\hat{w}_i) - f_i(\check{w}_i) \right)$ for some appropriate $\eta$. 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 $\mathcal{D}$, which brings us into the realm of classical learning theory. The analogue to the optimization (3) is regularized empirical risk minimization:

$$\hat{w} = \min_{w \in \mathcal{H}} \sum_{i=1}^{n} f_i(w) + \lambda h(w),$$

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 $\sum_{i=1}^{n} f_i(\hat{w}) = \sum_{i=1}^{n} f_i(w^*) + \lambda h(w^*)$ (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 $\hat{f}(w)$ for all $w \in \mathcal{H}$. Therefore, $\hat{w}$ is nearly as good as $w^*$ in terms of the true risk $\hat{f}$.

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 $w \in \mathcal{E}_I$ with small error:

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

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

$$\min_{w_1, \ldots, w_n} \sum_{i=1}^{n} c_i f_i(w_i) + \lambda \text{tr}(Y)$$

subject to $w_i \hat{y}_i^T \leq Y$ for all $i = 1, \ldots, n$. (8)

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 $\sum_{i \in \mathcal{E}_I} c_i \left( f_i(\hat{w}_i) - f_i(w^*) \right)$, and the statistical error $\sum_{i \in \mathcal{E}_I} c_i \left( \hat{f}(\hat{w}_{\text{avg}}) - f_i(\hat{w}_i) \right)$. 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 $\hat{f}(\hat{w}_{\text{avg}})$.

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 $\text{tr}(Y)$ 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 $\hat{w}_i$ found in Algorithm 2 are bounded by an ellipse $\check{Y}$ with small trace, and that $f_i(\hat{w}_i) = f_i(\check{w}_i)$ for $i \in \mathcal{E}_I$. By the optimality of $(\hat{w}_{1:n}, \hat{Y})$ for (8), the only way that $\text{tr}(\hat{Y})$ can be much larger than $\text{tr}(\check{Y})$ is therefore if $f_i(\hat{w}_i) \ll f_i(\check{w}_i)$ for $i \notin \mathcal{E}_I$. 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

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

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

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

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

**Bounding the statistical error.** We next consider the statistical error. We cannot bound this error via standard uniform convergence techniques, because each $f_j$ has a different argument $w_j$. 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 $\hat{f}_i$. In particular, we have:

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

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

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

**Bounding the trace.** We then bound $\operatorname{tr}(\hat{Y})$ itself. We again exploit the optimality constraint (9), which implies that

$\operatorname{tr}(\hat{Y}) \leq \operatorname{tr}(\hat{Y}) + \frac{1}{\lambda} \left(\sum_{i=1}^{n} c_i (f_i(\hat{w}_i) - f_i(\tilde{w}_i))\right)$

for any feasible $(\hat{w}_{1:n}, \hat{Y})$. We will take $\hat{w}_i$ to be $\tilde{w}_i$ as defined in equation (5) of Algorithm 2; we then take $Y$ to be $\frac{2}{\alpha n} \hat{W}_{\text{avg}}^\top$, where $\hat{W} = [\hat{w}_1 \cdots \hat{w}_n]$. Lemma 4.4 asserts that $(\hat{w}_{1:n}, \hat{Y})$ is feasible, and uses this to bound $\operatorname{tr}(\hat{Y})$:

**Lemma 4.4.** For $\hat{w}_i$ as defined in (6), and $\hat{Y} \defeq \frac{2}{\alpha n} \hat{W}_{\text{avg}}^\top$, we have $\hat{w}_i \hat{w}_i^\top \leq \hat{Y}$ for all $i$, and also $\operatorname{tr}(\hat{Y}) \leq \frac{2n^2}{\alpha}$. In addition,

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

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

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

This “almost” bounds $\operatorname{tr}(\hat{Y})$ in the following sense: if instead of a bound on $\sum_{i \in I_g} c_i (f_i(\hat{w}_i) - f_i(\tilde{w}_i))$, (14) gave a bound on the full sum $\sum_{i=1}^{n} c_i (f_i(\tilde{w}_i) - f_i(\hat{w}_i))$, then we could plug in to (13) to obtain (e.g.) $\operatorname{tr}(\hat{Y}) \leq \frac{2\alpha^2}{\lambda} + \frac{\alpha}{\lambda} \sqrt{\operatorname{tr}(\hat{Y}) + r}$, 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 $\tilde{z}_i \defeq f_i(\tilde{w}_i) - f_i(\hat{w}_i)$. 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 $[n] \setminus I_g$ 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 $\frac{1}{n} \sum_{i=1}^{n} c_i (f_i(\tilde{w}_i) - f_i(\hat{w}_i))$ is at least $\frac{2\alpha}{\lambda} \sum_{i \in I_g} c_i (f_i(\tilde{w}_i) - f_i(\hat{w}_i))$. Then, the update step in Algorithm 2 satisfies

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

Moreover, the supposition (1) holds if $\lambda = \frac{\sqrt{\alpha} n S}{\alpha}$ and $\operatorname{tr}(\hat{Y}) > \frac{\alpha^2}{\lambda}$.

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 $\operatorname{tr}(\hat{Y}) > \frac{\alpha^2}{\lambda}$.

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

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

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 $\sum_{i \in I_g} c_i \geq \frac{\eta^2}{\alpha}$ 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 $f_i(\hat{w}_{avg}) - f_i(\hat{w}_i)$ using a local Hölder’s inequality, which depends upon a refined notion of $S$ that we denote by $S_r$. Before defining $S_r$, we will state the local Hölder’s inequality:

**Lemma 5.1.** Suppose that the weights $b_i \in [0, 1]$ satisfy $\sum_{i \in I_k} b_i \geq ean$, and that the parameters $w_i$ satisfy $w_i w_i^{\epsilon} \leq Y$. Then, for any $w_0 \in H$, we have

$$\sum_{i \in I_k} b_i(w_i, \nabla f_i(w_0) - \nabla \hat{f}(w_0)) \leq \Delta \sqrt{r} S_r. \quad (17)$$

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:

$$\sum_{i \in I_k} c_i(w_i, \nabla f_i(w_0) - \nabla \hat{f}(w_0)) \leq an \sqrt{\text{tr}(Y) S_r}. \quad (18)$$

By taking $b_i = \Delta[\sum_{i \in I_k} \nabla f_i(w_0) - \nabla \hat{f}(w_0)] > \Delta \sqrt{\text{tr}(Y) S_r}$, Lemma 5.1 implies in particular that $(\nabla f_i(w_0) - \nabla \hat{f}(w_0)) \leq \Delta \sqrt{\text{tr}(Y) S_r}$ for all but $ean$ values of $i \in I_k$.

**A local spectral norm bound.** We now define $S_r$; it is the maximum operator norm over subsets of $I_k$ of size at least $e an$:

$$S_r \overset{\text{def}}{=} \max_{w \in H} \max_{T \subseteq I_k \mid |T| \geq e an} \frac{1}{\sqrt{|T|}} \left\| \sum_{i \in T} f_i(w) - \nabla \hat{f}(w) \right\|_\infty.$$

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

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

**Lemma 5.2.** Let the weights $b_i \in [0, 1]$ satisfy $\sum_{i \in I_k} b_i \geq ean$, and define $\hat{w}^{b}_{avg} = \frac{1}{\sum_{i \in I_k} b_i w_i} \sum_{i \in I_k} b_i w_i$. Then the solution $\hat{w}_{1:n}, \hat{Y}$ to (8) satisfies

$$\sum_{i \in I_k} b_i \left( f_i(\hat{w}^{b}_{avg}) - f_i(\hat{w}_i) \right) \leq \sum_{i \in I_k} b_i \left\| f_i(\hat{w}^{b}_{avg}) - \hat{w}_i \right\| \leq \Delta \sqrt{r} S_r \left( \sqrt{\text{tr}(Y) + r} \right). \quad (20)$$

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

$$\sum_{i \in I_k} b_i(\hat{f}(w) - \hat{f}(w')) - \sum_{i \in I_k} b_i(f_i(w) - f_i(w')) \leq 2 \Delta \sqrt{r} S_r. \quad (21)$$

Relative to Lemma 4.3, the main differences are: The bounds now hold for any weights $b_i$ (with $\hat{w}^{b}_{avg}$ replaced by $\hat{w}^{b}_{avg}$) 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 $\nabla f_i(\hat{w}^{b}_{avg}), \hat{w}^{b}_{avg} - \hat{w}_i$, and (21) holds at all $w, w'$ instead of just $\hat{w}^{b}_{avg}, \hat{w}$. 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 $\hat{w}^{b}_{avg}$, rather than $\hat{w}_{avg}$. Fortunately, by exploiting the fact that $\hat{w}_{avg}$ is nearly optimal for $\hat{f}$, we can replace $\hat{w}^{b}_{avg}$ with $\hat{w}_{avg}$ at the cost of a slightly weaker bound:

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

$$\sum_{i \in I_k} b_i (f_i(\hat{w}_{avg}) - f_i(\hat{w}_i)) \leq \left( \sum_{i \in I_k} b_i \right) \left( \sqrt{\text{tr}(Y) + r} + \frac{2lr^2}{an} \right). \quad (22)$$

In particular, if $\text{tr}(\hat{Y}) = O\left(\frac{r}{\alpha}\right)$ and $\lambda = O\left(\sqrt{am} S_r / r\right)$, then for all but $ean$ values of $i \in I_k$ we have $f_i(\hat{w}_{avg}) - f_i(\hat{w}_i) \leq O\left(\frac{S_r \sqrt{r} / \sqrt{\alpha}}{r}\right)$.

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

**Preserving inliers.** Our outlier removal step can be modified based on $S_r$ 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 $f_i(\hat{w}_i) - f_i(\hat{w}_i)$, we set $z_i$ to $\max(f_i(\hat{w}^b_i) - f_i(\hat{w}_i) - r, 0)$ for an appropriately chosen $r$. This creates a buffer such that we do not start to downweight points until their function error passes the threshold $r$, 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 \( \lambda = \frac{\log n}{R} \) and \( \text{tr}(Y) > 3\sigma^2/\alpha^2 \). Then, the update step in Algorithm 3 satisfies

\[
\sum_{i \in I_k} c_i - c_i' \leq \frac{\alpha}{2} \sum_{i=1}^{n} c_i - c_i'.
\]

This shows that the rate at which mass is removed from \( I_k \) is at most \( \frac{\alpha}{2} \) the rate at which mass is removed overall.

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

**Lemma 5.5.** If \( \nabla f_i(w) - \nabla \tilde{f}_i(w) \) is \( \sigma \)-sub-Gaussian and L-Lipschitz for all \( w \), then with probability \( 1 - \delta \) we have

\[
S_\epsilon = O\left( \sigma \left( \sqrt{\log(2/\epsilon)} + \sqrt{\max(1, \log(rL/\sigma)) + \log(1/\delta)} \right) \right).
\]

In particular, if \( n \geq \frac{1}{\alpha^2} \max(1, \log(rL/\sigma)) + \log(1/\delta) \), then \( S_\epsilon \in O\left( \sigma \sqrt{\log(2/\epsilon)} \right) \) 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 \( \tilde{f}_i(\tilde{w}_i) \) and \( f_i(w_i) \) are also sub-Gaussian, in the sense that the fraction of \( i \) for which \( f_i(\tilde{w}_i) - f_i(w_i) \geq \Omega(\sigma \sqrt{\log(2/\epsilon)} / \alpha) \) is at most \( \epsilon \). Inverting this, for sufficiently large \( \epsilon \), the fraction of \( i \) for which \( f_i(\tilde{w}_i) - f_i(w_i) \geq \epsilon \sigma / \sqrt{\alpha} \) is at most \( \exp(-\Omega(\epsilon^2)) \). 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 \( S_\epsilon \) in terms of \( S \) by exploiting the monotonicity of the operator norm.

**Lemma 5.6.** For any \( \epsilon \leq \epsilon_2 \), \( S_\epsilon \leq \sqrt{\max(\epsilon_2/\epsilon, \epsilon_2/\epsilon^\alpha)} \) \( S_1 \leq 2 \sqrt{\epsilon_2} S_\epsilon \).

When coupled with Corollary 5.3, this shows that the function errors concentrate in a Chebyshev-like manner; The fraction of \( i \) for which \( f_i(\tilde{w}_i) - f_i(w_i) \) exceeds \( \Omega(\sigma / \sqrt{\alpha}) \) is at most \( \epsilon \), and so the fraction of \( i \) for which \( f_i(\tilde{w}_i) - f_i(w_i) \geq \epsilon \sigma / \sqrt{\alpha} \) is \( O\left( \frac{1}{\epsilon} \right) \). Note that this is already a strengthening of the naïve bound from Markov’s inequality, which would only say that the further is \( O\left( \frac{1}{\epsilon} \right) \). The local Hölder’s inequality in Lemma 5.1 thus leads to a tighter analysis even without any further bounding of \( S_\epsilon \).

**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 \( w, w' \in \mathcal{H} \),

\[
f_i(w') \geq (w' - w, \nabla f_i(w)) + \frac{\Lambda}{2} ||w' - w||^2.
\]

In this case, we will obtain stronger bounds by iteratively clustering the output \( \tilde{w}_{1:n} \) 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 \( \epsilon \leq \frac{1}{2} \) and let \( \mathcal{U}, \tilde{w}_{1:n} \) be the output of Algorithm 4. Then \( \mathcal{U} \) has size at most \( \frac{1}{1 - \epsilon} \), and \( \min_{u \in \mathcal{U}} ||u - w||_2 \leq \frac{S}{\sqrt{\epsilon}} \sqrt{\log(2/\epsilon)} \) for all but \( \epsilon \) values of \( i \in I_k \).

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 \( \tilde{w}_i \) that are (as we will show in Proposition 6.2) relatively close to \( w^* \) for most \( i \in I_k \). Supposing that the good \( \tilde{w}_i \) are within distance \( r' \ll r \) of \( w^* \), we can cluster \( \tilde{w}_i \) 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 \( \tilde{w}_i \) 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 \( \tilde{w}_i \) (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 \( \tilde{w}_i \) will be within distance \( 2^{-t'} \) of \( w^* \) for almost all \( i \in I_k \), 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 \( \tilde{w}_{1:n} \) is the output of Algorithm 1, then \( ||\tilde{w}_i - w^*||_2 \) is small for most \( i \in I_k \) (this requires some work, since applying Theorem 4.1 directly would only imply that \( ||\tilde{w}_i - w^*||_2 \) 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 \( ||\tilde{w}_i - w^*||_2 \).** We will first show that \( \tilde{w}_i \) is close to \( w^* \) for almost all \( i \in I_k \):

**Proposition 6.2.** For some absolute constant \( C \) and for any \( \omega \geq 1 \), the output \( \tilde{w}_{1:n} \) of Algorithm 1 satisfies \( ||\tilde{w}_i - w^*||_2 \leq C \omega \cdot \frac{r(t')}{\sqrt{\epsilon}} \) for all but \( \frac{\epsilon}{\omega} \) values of \( i \).
20: Let \( r \) be a \( (\rho, 2r^2, \frac{\rho}{2}) \)-padded decomposition of \( I \) with \( \rho = O \left( \frac{r(t) \log(\frac{n}{t})}{t} \right) \).
21: for each \( T \in \mathcal{P}_k \) do  
22: \( \hat{w}_i(h) \) assigned \( \hat{w}_i(h) \) 
23: for \( h = 1, \ldots, n \) do  
24: Find a \( h_0 \) such that \( \| \hat{w}_i(h_0) - \hat{w}_i(h) \|_2 \leq \frac{\rho}{2} r(t) \) for at least half of the \( h \)'s. 
25: Set \( \hat{w}_i^{(t+1)} \) to \( \hat{w}_i(h_0) \). 
26: end for  
27: end for  
28: \( r(t+1) \sim \frac{1}{2} r(t) \)  
29: end for  
30: 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 \( \sum_{i \in I_k} b_i \geq \epsilon \), we have

\[
\sum_{i \in I_k} b_i (\nabla f_i \hat{w}_i^{b_i} - \nabla \hat{w}_i) \leq \left( \sum_{i \in I_k} b_i S_i \right) \left( \sqrt{\text{tr}(Y)} + r \right).
\] (26)

By strong convexity of \( f_i \), we then have

\[
0 \leq \sum_{i \in I_k} b_i \left( f_i \hat{w}_i^{b_i} - f_i \hat{w}_i \right) \leq \sum_{i \in I_k} b_i \left( \nabla f_i \hat{w}_i^{b_i} \right)
\] (27)

\[
\leq \sum_{i \in I_k} b_i \left( \nabla f_i \hat{w}_i^{b_i} \right), \hat{w}_i^{b_i} - \hat{w}_i \right) - \frac{\kappa}{2} \| \hat{w}_i - \hat{w}_i^{b_i} \|^2_2
\] (28)

\[
\leq \left( \sum_{i \in I_k} b_i \right) S_i \left( \sqrt{\text{tr}(Y)} + r \right) - \frac{\kappa}{2} \sum_{i \in I_k} b_i \| \hat{w}_i - \hat{w}_i^{b_i} \|^2_2.
\] (29)

Therefore:

**Lemma 6.3.** For any \( b_i \in [0, 1] \) satisfying \( \sum_{i \in I_k} b_i \geq \epsilon \), we have

\[
\frac{\sum_{i \in I_k} b_i \| \hat{w}_i - \hat{w}_i^{b_i} \|^2_2}{\sum_{i \in I_k} b_i} \leq \frac{2}{\kappa} \left( \sqrt{\text{tr}(Y)} + r \right) S_i.
\] (30)

By applying Lemma 6.3 to \( b_i \sim \frac{1}{2} \left( b_i + \frac{\sum_{i \in I_k} b_i}{S_i / \epsilon} \right) \), we obtain the following, which gives bounds in terms of \( \hat{w}_i^{b_i} \) rather than \( \hat{w}_i^{b_i} \):

**Corollary 6.4.** For any \( b_i \in [0, 1] \) satisfying \( \epsilon \), we have

\[
\frac{\sum_{i \in I_k} b_i \| \hat{w}_i - \hat{w}_i^{b_i} \|^2_2}{\sum_{i \in I_k} b_i} \leq \frac{16}{\kappa} \left( \sqrt{\text{tr}(Y)} + r \right) S_i.
\] (31)

In particular, \( \| \hat{w}_i - \hat{w}_i^{b_i} \|^2_2 \geq \frac{16}{\kappa} \left( \sqrt{\text{tr}(Y)} + r \right) S_i \) for at most \( \epsilon \) points \( i \in I_k \).

Corollary 6.4 is crucial because it shows that all but an \( \epsilon \) fraction of the \( \hat{w}_i \), for \( i \in I_k \), can concentrate around \( \hat{w}_i^{b_i} \).

Note that we also have \( \| \hat{w}_i^{b_i} - w^* \|^2_2 \leq \frac{\rho}{2} \left( f_i(\hat{w}_i^{b_i}) - f_i(w^*) \right) \), which is bounded by Theorem 4.1; moreover, Theorem 4.1 also bounds \( \text{tr}(Y) \). Finally, we have \( S_i / \epsilon \) 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 \( \| \hat{w}_i - w^* \|_2 \leq \sqrt{s} \) for almost all \( i \in I_k \), for some \( s \approx r \). We would ideally like to partition the \( \hat{w}_i \) into sets of small diameter (say 2s), such that all of \( I_k \) 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 \( I_k \)).

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 \( \frac{1}{2} \), all of \( I_k \) 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 \( (\rho, \tau, \delta)-\)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 \( \rho \), and (ii) for each \( x_i \), with probability \( 1 - \delta \) all points within distance \( \tau \) of \( x_i \) lie in a single element of \( \mathcal{P} \).

Fakharzadeh et al. (2003) show that for any \( \tau \) and \( \delta \), a \( (\rho, \tau, \delta) \)-padded decomposition exists with \( \rho = O \left( \frac{r \log(n)}{\delta} \right) \). Moreover, the same proof shows that, if every \( x_i \) is within distance \( \tau \) of at least \( \Omega(n\tau) \) other \( x_i \), then we can actually take \( \rho = O \left( \frac{r \log(2/\alpha)}{\delta} \right) \). In particular, in our case we can obtain a \( (\Omega(s \log(2/\alpha)),2s,7/8) \)-padded decomposition of the \( \bar{w}_i \) 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 \( l = 112 \log((t+1)/\delta) \) padded decompositions of the \( \bar{w}_i \). 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 \( \bar{w}_i \) 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 \( \frac{2}{3} \) of the time, the padded decomposition \( \mathcal{P}_h \) succeeds in preserving \( I_{\beta} \), it is also the case that roughly \( \frac{2}{3} \) 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 \( \bar{w}_{i:n}^{(t)} \rightarrow \hat{\bar{w}}_{i:n}^{(t+1)} \) on each iteration:

Lemma 6.6. Algorithm 4 satisfies the following property: at the beginning of iteration \( t \) of the outer loop, let \( I_{\beta}^{(t)} \) denote the set of points \( i \in I_{\beta} \) for which \( \|\bar{w}_i^{(t)} - w^*\|_2 \leq r^{(t)} \). Also suppose that \( |I_{\beta}^{(t)}| \geq (1-\epsilon)\alpha n \) and \( \delta \leq \frac{1}{2} \). Then, with probability \( 1 - \frac{r^{(t)} \delta}{r^{(t+1)}} \) over the randomness in the padded decompositions, \( \|\bar{w}_i^{(t)} - w^*\|_2 \leq \frac{1}{2} r^{(t)} \) for all but \( C_0 \left( \frac{S^2 \log(2/\alpha)}{\alpha n^2 \log(1/\epsilon)} \right)^2 \cdot \text{ran points in } I_{\beta}^{(t)} \), 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 \( p^\ast \) is a multivariate Gaussian. For \( \alpha \), we are only able to show a necessary dependence of \( \sqrt{\log \frac{1}{\alpha}} \), rather than the \( \sqrt{\log \frac{1}{\delta}} \) appearing in our bounds. Determining the true worst-case dependence on \( \alpha \) 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 4th singular value.

Lemma 7.1. Suppose that \( p^\ast \) is known to be a multivariate Gaussian \( \mathcal{N}(\mu, \Sigma) \) with covariance \( \Sigma \preceq \Sigma_0 \), where \( \mu \) and \( \Sigma \) are otherwise unknown. Also let \( \sigma_k \) denote the \( k \)th largest singular value of \( \Sigma_0 \). Then, given any amount of data, an \( \alpha \)-fraction of which is drawn from \( p^\ast \), any procedure for outputting at most \( m = 2^{k-1} \) candidate means \( \mu_1, \ldots, \mu_m \) must have, with probability at least \( \frac{1}{2} \),

\[
\min_{j=1}^m \|\hat{\mu}_j - \mu\|_2 \geq \frac{c_{\alpha}}{\sqrt{\log \frac{1}{\alpha}}} \sqrt{1 + \log \frac{1}{\alpha}}.
\]

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 if \( \hat{p} \geq p^\ast \). We start with a lemma characterizing when it is possible that the true distribution \( p^\ast \) has mean \( \mu \):

Lemma 7.2. Let \( \hat{p} = \mathcal{N}(0, \Sigma_0) \) and \( p^\ast = \mathcal{N}(\mu, \Sigma_0 - \lambda \mu \mu^\top) \). Then \( \hat{p} \geq p^\ast \) provided that \( \mu^\top \Sigma_0^{-1} \mu \leq \frac{\log \frac{1}{\alpha}}{1 + \log \frac{1}{\alpha}} \).

As a consequence, if \( \mu^\top \Sigma_0^{-1} \mu \leq t^2 \) for \( t = \frac{\log \frac{1}{\alpha}}{1 + \log \frac{1}{\alpha}} \), then \( p^\ast \) could have mean \( \mu \). Now, consider the space spanned by the \( k \) largest eigenvectors of \( \Sigma_0 \), and let \( B_k \) be the ball of radius \( \sigma_k \) in this space. Also let \( \mathcal{P} \) be a maximal packing of \( B_k \) of radius \( \frac{1}{4} \sigma_k \). A simple volume argument shows that \( |B_k| \geq 2^k \). On the other hand, every element \( \mu \) of \( B_k \) is a potential mean because it satisfies \( \mu^\top \Sigma_0^{-1} \mu \leq \frac{\|\mu\|_F^2}{\sigma_1^2} \leq t^2 \). Therefore, if the true mean \( \mu \) is drawn uniformly from \( B_k \), any \( 2^{k-1} \) candidate means \( \mu_j \) must miss at least half of the elements of \( B_k \) (in the sense of being distance at least \( \frac{1}{2} \sigma_k \) away) and so with probability at least \( \frac{1}{2} \), \( \min_{j=1}^m \|\hat{\mu}_j - \mu\|_2 \) is at least \( \frac{1}{2} \sigma_k \), 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/adviesarial 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 \( \frac{1}{4} \alpha^2 n \), the mean over \( T \) is \( \epsilon \)-close to the mean over \( I \). In symbols,

\[
\|\mu_T - \mu\|_2 \leq \epsilon \quad \text{for all } T \subseteq I \text{ with } |T| \geq \frac{1}{2} \alpha^2 n,
\]

where \( \mu_T \equiv \frac{1}{|T|} \sum_{i \in T} x_i \). In such a case, can we approximate the mean of \( I \), even if \( I \) is unknown and the points in \( [n] \setminus I \) are arbitrary?
If we do not care about computation, the answer is yes, via a simple exponential-time algorithm. Call a set of points $J$ $\alpha$-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 $r$ when taking subsets of $J$ of size at least $\frac{1}{2} \alpha^2 n^2$). Then, we can run the following (exponential-time) algorithm:

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

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

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 $\alpha$-stable? This involves checking the following constraint:

$$\left\| \sum_{i \in J} (x_i - \mu_J) \right\|_F \leq \epsilon \|c\|_2$$

if $c_i \in \{0, 1\}$ and $\sum_{i \in J} c_i \geq \frac{1}{2} \alpha^2 n$. (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 semi-definite relaxation of $C$, resulting in the constraints $C_{ij} \in \{0, 1\}$ and $\text{tr}(C) \geq \frac{1}{2} \alpha^2 n$. Letting $A_{ij} = (x_i - \mu_J)^\top (x_j - \mu_J)$, this results in the following sufficient condition for $\alpha$-stability:

$$\text{tr}(A_{ij}^\top C) \leq \epsilon^2 \text{tr}(C)^2$$

for all $C$ such that

$$C \geq 0, C_{ij} \in \{0, 1\}, \text{tr}(C) \geq \frac{1}{2} \alpha^2 n.$$ (34)

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

REFERENCES