High-Probability Bounds for Robust Stochastic Frank-Wolfe Algorithm

Tongyi Tang  
Krishnakumar Balasubramanian  
Thomas C. M. Lee

Abstract

We develop and analyze robust Stochastic Frank-Wolfe type algorithms for projection-free stochastic convex optimization problems with heavy-tailed stochastic gradients. Existing works on the oracle complexity of such algorithms require a uniformly bounded variance assumption, and hold only in expectation. We develop tight high-probability bounds for robust versions of Stochastic Frank-Wolfe type algorithm under heavy-tailed assumptions, including infinite variance, on the stochastic gradient. Our methodological construction of the robust Stochastic Frank-Wolfe type algorithms leverage techniques from the robust statistics literature. Our theoretical analysis highlights the need to utilize robust versions of Stochastic Frank-Wolfe type algorithm for dealing with heavy-tailed data arising in practice.

1 INTRODUCTION

In this paper, we consider constrained stochastic optimization problem of the form

$$\arg\min_{x \in X} \{ f(x) = \mathbb{E}_{\xi}[F(x, \xi)] = \int F(x, \xi) \, dP(\xi) \}, \quad (1)$$

where $f$ is a smooth convex function, $X$ is a closed convex subset of $\mathbb{R}^d$, $\xi$ is a random vector defined according to a distribution $P$ on the domain of $\xi$. We denote by $x^*$, the minimizer of $f(x)$ in $X$. Problems of the form in (1) arise frequently in modern machine learning, including matrix completion [Davenport and Romberg 2016], Freund et al. [2017], structured linear inverse problems [ Chandrasekaran et al. 2012], [Tewari et al. 2011], multi-task learning [ Sener and Koltun 2018], [Zhao et al. 2020] and particle filtering [Lacoste-Julien et al. 2015]. Stochastic Frank-Wolfe (SFW) and Stochastic Conditional Gradient Sliding (SCGS) algorithms are widely used iterative first-order algorithm for solving [1]. Each iteration of SFW/SCGS involves performing linear minimization over the constraint set $X$ based on stochastic first-order (or gradient) information, $\nabla F(\cdot, \xi) \in \mathbb{R}^d$. Depending on the geometry of the constraint set, SFW/SCGS is widely used in practice due to its projection-free nature [Besançon et al. 2021]. Despite its wide-spread usage, our understanding of the oracle complexity of SFW/SCGS algorithm is limited.

As each iteration of the SFW/SCGS algorithm requires access to a Stochastic First-order Oracle (SFO) and a Linear Minimization Oracle (LMO), the oracle complexity is measured by the number of calls to SFO and LMO to obtain an $\epsilon$-optimal solution – that is, a point $\tilde{x}$ such that $f(\tilde{x}) - f(x^*) \leq \epsilon$. A majority of the existing results on the oracle complexity of SFW/SCGS algorithm are available only in expectation, i.e., on the metric $\mathbb{E}_\xi[f(\tilde{x}) - f(x^*)] \leq \epsilon$. Furthermore, to obtain such oracle complexity results in expectation, it is assumed that stochastic gradient used has uniformly bounded variance (i.e., $\mathbb{E}_\xi[||\nabla F(\cdot, \xi) - \nabla f(x)||^2] \leq \sigma^2$), where $|| \cdot ||$ denotes the Euclidean norm on $\mathbb{R}^d$. This characterization of the oracle complexity provides very little understanding regarding the behaviour of the SFW/SCGS algorithms. In particular, the effect of the properties of the distribution $P$, on the heavy-tail nature of the stochastic gradient, and consequently on the oracle complexity is not characterized by the oracle complexity results in expectation. Furthermore, the oracle complexity of SFW/SCGS algorithm or its robust variants under infinite variance assumption is not known.

To provide a concrete motivating example, consider the problem of sparse linear regression: Given $(a, y) \in \mathbb{R}^d \times \mathbb{R}$, consider the linear regression model $y = \langle a, \tilde{x} \rangle + \epsilon$, where for two vectors $c, d \in \mathbb{R}^d$, $\langle c, d \rangle$ represents the Euclidean inner product. Here, we let $\epsilon \sim N(0, 1)$ and the true parameter $\tilde{x}$ is assumed to be $s$-sparse (i.e., it has only $s$ non-zero coordinates out of the $d$ coordinates). The $L_1$-constrained
least-squares estimator is then given by
\[ \arg \min_{x \in X_1(s)} \mathbb{E}[(y - \langle a, x \rangle)^2], \] (2)

where \( X_1(s) := \{ x \in \mathbb{R}^d : \sum_{j=1}^d |x_j| \leq s \} \) is the \( L_1 \) ball of radius \( s \). This problem fits in the setup of (1) with \( \xi := (a, y) \) and \( F(x, \xi) := (y - a^\top x)^2 \). Hence the stochastic gradient is given by
\[ G(x, \xi) := \nabla F(x, \xi) = 2(aa^\top x - ya) \in \mathbb{R}^d. \]

Note that as the iterates of SFW/SCGS algorithms (see, for example, Algorithm 2 for the description of SCGS) are in the set \( X_1(s) \), we have \( ||x|| \) is to always bounded for all \( x \) along the trajectory of the algorithm. Hence, the \((1 + \alpha)\)-th moment of the stochastic gradient, i.e., \( \mathbb{E}[||G(x, \xi)||^{(1+\alpha)}] \), is controlled by the order of \( \mathbb{E}[||a||^{(1+\alpha)}}] \). When the covariance \( a \) is a zero-mean multivariate \( t \)-distribution with degrees of freedom in the interval \([2, 4]\), or is a zero-mean multivariate Pareto distribution with parameter in the interval \([2, 4]\), the stochastic gradients have infinite variance. Hence, the existing oracle complexity results for SFW/SCGS are not applicable.

Focusing on the SCGS algorithm [Lan and Zhou, 2016], in this work we develop and analyze robust versions of it, and establish high-probability SFO and LMO complexity bounds. First, under a sub-Gaussian tail assumption on the stochastic gradient, we establish high-probability bounds for the standard SCGS algorithm. However, when the stochastic gradient is heavy-tailed or has infinite variance, the standard SCGS algorithm is sub-optimal. This is due to the fact that each iteration of SCGS algorithm requires a sample average of mini-batch stochastic gradients. When the stochastic gradients are heavy-tailed, it is well-known from classical robust statistics literature that sample averages are poor estimates of the true expectation. Hence, to deal with the case when the stochastic gradient is heavy-tailed or has infinite variance, we construct robust versions of mini-batch stochastic gradient estimates which are subsequently used in the SCGS algorithm. Our motivations for developing robust SCGS algorithms are based on the fact that data arising from various real-world problems (for example, finance, networks, linguistics) are modeled efficiently using heavy-tailed distribution [Resnick, 2007, Fan et al., 2014, Taleb, Roughgarden, 2020]. We establish high-probability bounds on the oracle complexity results on robust SCGS algorithm by developing tight concentration inequalities for heavy-tailed martingales, which might be of independent interest. The established high-probability bounds on the SFO and LMO complexity of SCGS, provide a fine-grained characterization of the effect of the moments/tails of the distribution \( \mathbb{P} \) on the performance of SCGS algorithms in terms of both the level of solution accuracy and the confidence specified by the practitioner.

1.1 RELATED WORKS

Robust statistics: Robust statistics is a classical topic with too large a literature to summarize completely. We refer the reader to [Huber, 2004] for an overview. We emphasize that a majority of the robust estimators developed in the statistics literature are invariably computationally intractable. The revival of robust statistics in modern mathematical statistics and learning theory communities has started with the work of [Catoni, 2012]. Since then, there has been intense work on robust mean and covariance estimation [Minsker, 2015, Cardot et al., 2017, Minsker, 2018, Lugosi and Mendelson, 2019a, b, Hopkins, 2020], and robust empirical risk minimization [Hsu and Sabato, 2016, Diakonikolas et al., 2019, Geoffrey et al., 2020, Lecue and Lerasle, 2020]. However, such results are mainly statistical in nature and are not directly applicable for stochastic optimization with heavy-tailed gradients. In this work, we leverage the classical trick of trimmed estimator from the robust statistics literature, and the recently proposed optimal mean-estimation technique by [Cherapanamjeri et al., 2022] in the context of projection-free stochastic optimization.

Robust stochastic optimization: Early works on robust stochastic optimization include [Krasulina, 1969, Martin andMasreliez, 1975, Price andVandendijke, 1979, Chen andGao, 1980, Nemirovskij and Yudin, 1983]. Limit theorems for random iterated maps and time-series with heavy-tails were proved in [Mirek, 2011, Buraczewski et al., 2012, Mikosch andWintenberger, 2016]. Such results make restrictive assumptions that are not satisfied by stochastic optimization algorithms. In modern stochastic optimization, several works consider the effect of heavy-tails on the performance of the algorithm [Oliveira and Thompson, 2017, Nazin et al., 2019, Wang et al., 2021, Davis et al., 2021, Bartl andMendelson, 2022, Anantharam andBorkar, 2012, Holland, 2021]. There has also been overwhelming theoretical and empirical evidence in the modern machine learning literature that shows the noise in stochastic gradient algorithm could easily turn out to have heavy-tails due to the composite or product nature of the random vectors/matrices in the stochastic gradient [Hodgkinson and Mahoney, 2021, Panigrahi et al., 2019, Simsekli et al., 2019, 2020a, Gurbuzbalaban andHu, 2021, Camuto et al., 2020, Scaman andMalherbe, 2020, Simsekli et al., 2020a]. Furthermore, recently robust mean estimation based and trimming based SGD algorithms were analyzed in [Prasad et al., 2020, and Bubeck et al., 2013, Gorbunov et al., 2020, Zhang et al., 2020a, Mai andJohansson, 2021, Zhang et al., 2020a] respectively. However the above works do not deal with projection-free robust stochastic optimization, which is the main focus of our work.

Frank-Wolfe Algorithms: The Frank-Wolfe method first proposed by [Frank and Wolfe, 1956, Levitin and Polyak, 1966], has had a renewed interest in the past decade. We refer the reader to [Jaggi, 2013, Harchaoui et al., 2015,}.
We now provide a list of theoretical contributions to the
(Sub-Gaussian) Assumption 1.2 with which we could obtain an
we first introduce the notion of optimality that we consider
1.3 Remark

\[ \nabla E \] 

\[ \alpha \]

\[ \beta \]

\[ \gamma \]

\[ \delta \]

\[ \epsilon \]

\[ \phi \]

\[ \psi \]

\[ \chi \]

\[ \theta \]

\[ \omega \]

\[ \nu \]

\[ \mu \]

\[ \lambda \]

\[ \kappa \]

\[ \tau \]

\[ \zeta \]

\[ \eta \]

\[ \ sprint \]

\[ \beta \]

\[ \gamma \]

\[ \delta \]

\[ \epsilon \]

\[ \phi \]

\[ \psi \]

\[ \chi \]

\[ \theta \]

\[ \omega \]

\[ \nu \]

\[ \mu \]

\[ \lambda \]

\[ \kappa \]

\[ \tau \]

\[ \zeta \]

\[ \eta \]

\[ \ sprint \]

\[ \beta \]

\[ \gamma \]

\[ \delta \]

\[ \epsilon \]

\[ \phi \]

\[ \psi \]

\[ \chi \]

\[ \theta \]

\[ \omega \]

\[ \nu \]

\[ \mu \]

\[ \lambda \]

\[ \kappa \]

\[ \tau \]

\[ \zeta \]

\[ \eta \]

\[ \ sprint \]

\[ \beta \]

\[ \gamma \]

\[ \delta \]

\[ \epsilon \]

\[ \phi \]

\[ \psi \]

\[ \chi \]

\[ \theta \]

\[ \omega \]

\[ \nu \]

\[ \mu \]

\[ \lambda \]

\[ \kappa \]

\[ \tau \]

\[ \zeta \]

\[ \eta \]

\[ \ sprint \]

\[ \beta \]

\[ \gamma \]

\[ \delta \]

\[ \epsilon \]

\[ \phi \]

\[ \psi \]

\[ \chi \]

\[ \theta \]

\[ \omega \]

\[ \nu \]

\[ \mu \]

\[ \lambda \]

\[ \kappa \]

\[ \tau \]

\[ \zeta \]

\[ \eta \]

\[ \ sprint \]

\[ \beta \]

\[ \gamma \]

\[ \delta \]

\[ \epsilon \]

\[ \phi \]

\[ \psi \]

\[ \chi \]

\[ \theta \]

\[ \omega \]

\[ \nu \]

\[ \mu \]

\[ \lambda \]

\[ \kappa \]

\[ \tau \]

\[ \zeta \]

\[ \eta \]

\[ \ sprint \]

\[ \beta \]

\[ \gamma \]

\[ \delta \]

\[ \epsilon \]

\[ \phi \]

\[ \psi \]

\[ \chi \]

\[ \theta \]

\[ \omega \]

\[ \nu \]

\[ \mu \]

\[ \lambda \]

\[ \kappa \]

\[ \tau \]

\[ \zeta \]

\[ \eta \]

\[ \ sprint \]

\[ \beta \]

\[ \gamma \]

\[ \delta \]

\[ \epsilon \]

\[ \phi \]

\[ \psi \]

\[ \chi \]

\[ \theta \]

\[ \omega \]

\[ \nu \]

\[ \mu \]

\[ \lambda \]

\[ \kappa \]

\[ \tau \]

\[ \zeta \]

\[ \eta \]

\[ \ sprint \]

\[ \beta \]

\[ \gamma \]

\[ \delta \]

\[ \epsilon \]

\[ \phi \]

\[ \psi \]

\[ \chi \]

\[ \theta \]

\[ \omega \]

\[ \nu \]

\[ \mu \]

\[ \lambda \]

\[ \kappa \]

\[ \tau \]

\[ \zeta \]

\[ \eta \]

\[ \ sprint \]

\[ \beta \]

\[ \gamma \]

\[ \delta \]

\[ \epsilon \]

\[ \phi \]

\[ \psi \]

\[ \chi \]

\[ \theta \]

\[ \omega \]

\[ \nu \]

\[ \mu \]

\[ \lambda \]

\[ \kappa \]

\[ \tau \]

\[ \zeta \]

\[ \eta \]

\[ \ sprint \]

\[ \beta \]

\[ \gamma \]

\[ \delta \]

\[ \epsilon \]

\[ \phi \]

\[ \psi \]

\[ \chi \]

\[ \theta \]

\[ \omega \]

\[ \nu \]

\[ \mu \]

\[ \lambda \]

\[ \kappa \]

\[ \tau \]

\[ \zeta \]

\[ \eta \]

\[ \ sprint \]

\[ \beta \]

\[ \gamma \]

\[ \delta \]

\[ \epsilon \]

\[ \phi \]

\[ \psi \]

\[ \chi \]

\[ \theta \]

\[ \omega \]

\[ \nu \]

\[ \mu \]

\[ \lambda \]

\[ \kappa \]

\[ \tau \]

\[ \zeta \]

\[ \eta \]

\[ \ sprint \]

\[ \beta \]

\[ \gamma \]

\[ \delta \]

\[ \epsilon \]

\[ \phi \]

\[ \psi \]

\[ \chi \]

\[ \theta \]

\[ \omega \]

\[ \nu \]

\[ \mu \]

\[ \lambda \]

\[ \kappa \]

\[ \tau \]

\[ \zeta \]

\[ \eta \]

\[ \ sprint \]

\[ \beta \]

\[ \gamma \]

\[ \delta \]

\[ \epsilon \]

\[ \phi \]

\[ \psi \]

\[ \chi \]

\[ \theta \]

\[ \omega \]

\[ \nu \]

\[ \mu \]

\[ \lambda \]

\[ \kappa \]

\[ \tau \]

\[ \zeta \]

\[ \eta \]

\[ \ sprint \]

\[ \beta \]

\[ \gamma \]

\[ \delta \]

\[ \epsilon \]

\[ \phi \]

\[ \psi \]

\[ \chi \]

\[ \theta \]

\[ \omega \]

\[ \nu \]

\[ \mu \]

\[ \lambda \]

\[ \kappa \]

\[ \tau \]

\[ \zeta \]

\[ \eta \]

\[ \ sprint \]

\[ \beta \]

\[ \gamma \]

\[ \delta \]

\[ \epsilon \]

\[ \phi \]

\[ \psi \]

\[ \chi \]

\[ \theta \]

\[ \omega \]

\[ \nu \]

\[ \mu \]

\[ \lambda \]

\[ \kappa \]

\[ \tau \]

\[ \zeta \]

\[ \eta \]

\[ \ sprint \]

\[ \beta \]

\[ \gamma \]

\[ \delta \]

\[ \epsilon \]

\[ \phi \]

\[ \psi \]

\[ \chi \]

\[ \theta \]

\[ \omega \]

\[ \nu \]

\[ \mu \]

\[ \lambda \]

\[ \kappa \]

\[ \tau \]

\[ \zeta \]

\[ \eta \]

\[ \ sprint \]

\[ \beta \]

\[ \gamma \]

\[ \delta \]

\[ \epsilon \]

\[ \phi \]

\[ \psi \]

\[ \chi \]

\[ \theta \]

\[ \omega \]

\[ \nu \]

\[ \mu \]

\[ \lambda \]

\[ \kappa \]

\[ \tau \]

\[ \zeta \]

\[ \eta \]

\[ \ sprint \]

\[ \beta \]

\[ \gamma \]

\[ \delta \]

\[ \epsilon \]

\[ \phi \]

\[ \psi \]

\[ \chi \]

\[ \theta \]

\[ \omega \]

\[ \nu \]

\[ \mu \]

\[ \lambda \]

\[ \kappa \]

\[ \tau \]

\[ \zeta \]

\[ \eta \]

\[ \ sprint \]

\[ \beta \]

\[ \gamma \]

\[ \delta \]

\[ \epsilon \]

\[ \phi \]

\[ \psi \]

\[ \chi \]

\[ \theta \]

\[ \omega \]

\[ \nu \]

\[ \mu \]

\[ \lambda \]

\[ \kappa \]

\[ \tau \]

\[ \zeta \]

\[ \eta \]

\[ \ sprint \]

\[ \beta \]

\[ \gamma \]

\[ \delta \]

\[ \epsilon \]

\[ \phi \]

\[ \psi \]

\[ \chi \]

\[ \theta \]

\[ \omega \]

\[ \nu \]

\[ \mu \]

\[ \lambda \]

\[ \kappa \]

\[ \tau \]

\[ \zeta \]

\[ \eta \]
To our knowledge, our work provides the first comprehensive high-probability oracle complexity results for SCGS algorithm with light and heavy-tailed stochastic gradients (including infinite variance). We discuss the consequences of our theoretical results for practice by reporting simulations for heavy-tailed sparse linear regression (Section 4) and multi-class logistic regression (Appendix–Section 1).

2 ROBUST STOCHASTIC FRANK-WOLFE ALGORITHMS

Recall that our task is to solve constrained stochastic convex optimization problems of the form in \( \mathbb{X} \) using projection-free Frank-Wolfe type algorithm, which involves two main steps: the gradient evaluation step and the linear optimization step. The gradient evaluation step is typically based on averaging a mini-batch of gradients \( [\text{Hazan and Luo}, 2016, \text{Lan and Zhou}, 2016, \text{Balasubramanian and Ghadimi}, 2021] \). It is well-known from classical robust statistics that the sample average is not an accurate estimate of true expectation in the heavy-tailed setting. Hence, a natural strategy is to replace the sample average with appropriate robust versions of mean estimators, and incorporate such estimators in Frank-Wolfe type algorithms. Specifically, we consider the version of stochastic conditional gradient sliding algorithm in \( \text{Lan and Zhou} [2016] \), also analyzed in \( \text{Balasubramanian and Ghadimi} [2021] \) for the case of biased gradients. We first state a subroutine in Algorithm 1 that we subsequently use.

Note that Algorithm 1 is indeed the SFW algorithm for inexactly solving the following quadratic program

\[
P_{\mathbb{X}}(x, g, \gamma) = \arg \min_{u \in \mathcal{A}^n} \left\{ \langle g, u \rangle + \frac{\gamma}{2} \|u - x\|^2 \right\}, \tag{3}
\]

which is the standard subproblem of stochastic first-order methods applied to a minimization problem when \( g \) is an unbiased stochastic gradient of the objective function at \( x \). We now present Algorithm 2 which applies the Frank-Wolfe method to inexactly solve subproblems of the stochastic accelerated gradient method. It is well known that the above approach can significantly reduce the total number of calls to the stochastic oracle \( \text{[Lan and Zhou], 2016, Balasubramanian and Ghadimi, 2021]} \).

<table>
<thead>
<tr>
<th>Mean-Estimator</th>
<th>Tails</th>
<th>Theorem</th>
<th>SFO</th>
<th>LMO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>Asmp. 1.2</td>
<td>Thm. 3.3</td>
<td>( O\left( \frac{\log(1/\delta)}{\epsilon} \right)^2 )</td>
<td>( O\left( \frac{\log(1/\delta)}{\epsilon} \right)^2 )</td>
</tr>
<tr>
<td>Clipped Grad.</td>
<td>Asmp. 1.4</td>
<td>Thm. 3.6(a)</td>
<td>( O\left( \frac{\log(1/\delta)}{\epsilon} \right)^2 )</td>
<td>( O\left( \frac{\log(1/\delta)}{\epsilon} \right)^2 )</td>
</tr>
<tr>
<td>Clipped Grad.</td>
<td>Asmp. 1.5</td>
<td>Thm. 3.6(b)</td>
<td>( O\left( \frac{\sqrt{\log(1/\delta)}}{\epsilon} \right)^2 )</td>
<td>( O\left( \frac{\sqrt{\log(1/\delta)}}{\epsilon} \right)^2 )</td>
</tr>
<tr>
<td>CTBJ [Cherapanamjeri et al., 2022]</td>
<td>Asmp. 1.5</td>
<td>Thm. 3.8</td>
<td>( O\left( \frac{\log(1/\delta)}{\epsilon} \right)^2 )</td>
<td>( O\left( \frac{\log(1/\delta)}{\epsilon} \right)^2 )</td>
</tr>
<tr>
<td>BC Clipped Grad.</td>
<td>Asmp. 1.4</td>
<td>Thm. 3.13(a)</td>
<td>( O\left( C(d, \alpha, \delta) \right)^2 )</td>
<td>( O\left( C(d, \alpha, \delta) \right)^2 )</td>
</tr>
<tr>
<td>BC Clipped Grad.</td>
<td>Asmp. 1.5</td>
<td>Thm. 3.13(b)</td>
<td>( O\left( C(d, \beta, \delta) \right)^2 )</td>
<td>( O\left( C(d, \beta, \delta) \right)^2 )</td>
</tr>
</tbody>
</table>

Table 1: A summary of the obtained high-probability bounds. All results correspond to the notion of \((\epsilon, \delta)\)-optimal solution introduced in Definition 1.1. The parameters \( C(d, \alpha, \delta) \) and \( C(d, \beta, \delta) \) are defined in Theorem 3.13. BC stands for Bias-Corrected. The last two rows also require the symmetric condition described in Assumption 3.10.

Algorithm 1 Inexact Conditional Gradient (ICG) method

**Input:** \( (x, g, \gamma, \mu) \).

Set \( y_t = x \), \( t = 1 \), and \( \kappa = 0 \).

while \( \kappa = 0 \) do

\[ y_t = \arg\min_{u \in \mathcal{A}^n} \left\{ h_\gamma(u) := \langle g + \gamma(y_t - x), u - y_t \rangle \right\} \]

If \( h_\gamma(y_t) \geq -\mu \), set \( \kappa = 1 \). Else \( \tilde{y}_t = \frac{1}{\kappa + 1} y_t - \frac{2}{\kappa + 1} y_{t-1} \) and \( t = t + 1 \).

end while

Output \( \tilde{y}_{t-1} \).

Algorithm 2 Robust Stochastic Accelerated Gradient Method with Inexact Updates

**Input:** \( z_0 = x_0 \in \mathcal{X} \), positive integer sequence \( m_k \), and sequences \( \alpha_k, \gamma_k, \mu_k \) and iteration limit \( N \geq 1 \).

for \( k = 1, \ldots, N \) do

1. Set \( w_k = (1 - \alpha_k) z_{k-1} + \alpha_k x_{k-1} \)

2. Call the stochastic oracle \( m_k \) times, and compute (robust) stochastic gradients \( \tilde{G}_k \) as given by (4), (5), (6) or (10).

3. Set \( x_k = ICG(x_{k-1}, \tilde{G}_k, \gamma_k, \mu_k) \), where \( ICG(\cdot) \) is the output of Algorithm 1 with input \( (x_{k-1}, \tilde{G}_k, \gamma_k, \mu_k) \).

4. Set \( z_k = (1 - \alpha_k) z_{k-1} + \alpha_k x_k \).

end for

**Output:** \( z_N \)
2.1 ROBUST MEAN ESTIMATORS

Note that step 2 in Algorithm 2 requires the gradient estimator denoted by $\hat{G}_k$. We now elaborate the robust stochastic gradient estimators that we propose to use in this work. Recall that standard analysis of SCGS algorithm takes the sample average of the mini-batch of i.i.d. stochastic gradient (obtained by querying the SFO) in each iteration. In this case, the gradient estimator is given by

$$\hat{G}_k := \frac{1}{m_k} \sum_{j=1}^{m_k} G(w_k, \xi_{k,j}). \quad (4)$$

As we will see from our analysis, the above gradient estimator is not robust to heavy-tails, i.e., when the vectors $G(w_k, \xi_{k,j})$ are heavy-tailed. Our first robust stochastic gradient estimator is based on the idea of trimmed or clipped estimators, generalized to the multivariate setting [Tukey and McLaughlin 1963, Bickel 1965, Huber 2004, Stigler 1973]. More recently such ideas have been used in the context of bandit optimization [Bubeck et al. 2013] and stochastic gradient descent algorithm [Gorbunov et al. 2020, Zhang et al. 2020, Mai and Johansson 2021].

Formally, in our setting, given i.i.d. stochastic gradients $\{G(w_k, \xi_{k,j})\}_{j=1}^{m_k}$, the clipped gradient estimator is defined, for some $\delta \in (0,1)$,

$$\bar{G}_k := \frac{1}{m_k} \sum_{j=1}^{m_k} G(w_k, \xi_{k,j}) \mathbb{I}\{A_j\} \quad \text{(5)}$$

where $A_j := ||G(w_k, \xi_{k,j})|| \leq \left(\frac{j \sigma^{1+\alpha}}{\log(1/\delta)}\right)^{\frac{1}{1+\alpha}}$.

Here, $\mathbb{I}\{A\}$, for the event $A$ is defined as taking value 1 when the event $A$ is true and taking value 0 otherwise. While the above estimator handles robust stochastic gradients (including ones with potential infinite variance condition in Assumption 1.2), it turns out that the oracle complexities under the even weaker condition in Assumption 1.5 with the above clipped gradient estimator is sub-optimal. To improve the oracle complexity under Assumption 1.5 we leverage the recent optimal robust mean estimation procedure proposed in [Cherapanamjeri et al. 2022], which we call as the CTBJ procedure. Given i.i.d. stochastic gradients $\{G(w_k, \xi_{k,j})\}_{j=1}^{m_k}$, we use CTBJ estimator is given by (see Algorithm 1)

$$\bar{G}_k := \text{OPTIMALMEANEAST} \{\{G(w_k, \xi_{k,j})\}_{j=1}^{m_k}\}. \quad (6)$$

Roughly, the idea of [Cherapanamjeri et al. 2022] is to use filtering to remove outliers and then compute the median by gradient descent procedures. A full description of the procedure is provided in Appendix—Section 5 for the sake of completeness. We also emphasize that while the estimator for the light-tailed case in (4) is unbiased, the robust gradient estimators in (5) and (6) are biased. This is another challenge that we handle in our analysis. Finally, we also remark that in Section 3.4 we introduce a bias-corrected clipped gradient procedure which achieves improved oracle complexities under an additional symmetry assumption on the distribution on stochastic gradients.

3 HIGH-PROBABILITY BOUNDS

We now provide our main results on high-probability bounds on the oracle complexity of Algorithm 2. To do so, we also make the following standard smoothness assumption about the stochastic gradient, which is common in the literature of smooth convex optimization [Nesterov 2018, Lan and Zhou 2016, Balasubramanian and Ghadimi 2021].

Assumption 3.1. The objective function $f$ has Lipschitz continuous gradient with constant $L > 0$, i.e., $||\nabla f(y) - \nabla f(x)|| \leq L||y - x||$ for all $x, y \in \mathbb{R}^d$.

We first state a preliminary result about the iterates of Algorithm 2 under Assumption 3.1.

Lemma 3.2. Let $\{z_k\}_{k \geq 1}$ be generated by Algorithm 2 and the function $f$ be convex. Let $\Delta_k := \hat{G}_k - \nabla f(w_k)$, $\hat{G}_k := \prod_{i=2}^{k}(1 - \alpha_i)$, $\hat{1} := 1$ and $D_0 = ||x_0 - x_*||^2$. Then under Assumption 3.1 we have

$$\frac{f(z_N) - f(x_*)}{\Gamma_N} \leq \frac{\gamma_0}{2} ||x_0 - x_*||^2 + \sum_{i=1}^{N} \frac{\alpha_k \bar{y}_k}{\Gamma_k} + \sum_{k=1}^{N} \frac{||\Delta_k||^2}{2L \Gamma_k}. \quad (7)$$

For our subsequent analysis, we set

$$\alpha_k = \frac{2}{k + 1}, \quad \gamma_k = \frac{4L}{k}, \quad \text{and} \quad \mu_k = \frac{LD_0}{kN}. \quad (8)$$

The proof of Lemma 3.2 is provided in Appendix—Section 4 and is an intermediate result in the proof of Theorem 3.1 in Balasubramanian and Ghadimi 2021 with minor change. Our high-probability bounds are now based on developing concentration inequalities for the various gradient estimators $\bar{G}_k$ and bounding the terms appearing in right hand side of (4) in high-probability. To do so, we prove novel user-friendly concentration inequalities for (scalar-valued) martingales with heavy-tails that are discussed in detail in Section 3. It is worth mentioning that Lesigne and Volny 2001 and Fan et al. 2017 also consider tail bounds for heavy-tailed martingales. However, they only provide deviation inequalities and their assumptions do not cover the regimes of heavy-tails that we are interested in.
3.1 ORACLE COMPLEXITY WITH SAMPLE AVERAGE ESTIMATOR

We now provide oracle complexity results for Algorithm 2 with the sample average gradient estimator in (5), that hold in high-probability.

**Theorem 3.3.** Let Algorithm 2 be run with $G_k$ as defined in (5), and with parameters $\alpha_k, \gamma_k, \mu_k$ as in (8). If the stochastic gradients $G(x, \xi)$ satisfy Assumptions 1.2 and 3.1 and $m_k = O(N^3)$, then

$$
P \left( f(z_N) - f(x_*) \leq \frac{D_0 \sigma^2 \log(1/\delta)}{N(N + 1)} \right) \geq 1 - \delta,
$$

and the SFO and LMO complexity are respectively bounded by

$$O \left( \left( \frac{\log(1/\delta)}{\epsilon} \right)^2 \right) \quad \text{and} \quad O \left( \frac{\log(1/\delta)}{\epsilon} \right).$$

**Remark 3.4.** Note that [Lan and Zhou 2016] shows that the SFO and LMO oracle complexity for Algorithm 2 with the sample average gradient estimator are of order $O(1/\epsilon^2)$ and $O(1/\epsilon)$ in expectation. Our results in Theorem 3.3 generalize this to the high-probability setting quantifying the effect of the allowed confidence level $\delta$ precisely.

3.2 ORACLE COMPLEXITY WITH CLIPPED GRADIENT ESTIMATOR

We first provide results on the bias, tail and moment bounds on the clipped gradient estimator $\tilde{G}_k$ defined in (5). Then, we provide oracle complexity results for Algorithm 2 with the clipped gradient estimator, that hold with high-probability.

**Lemma 3.5.** Let $\delta \in (0, 1)$ and $C$ be a positive universal constant. Let $\tilde{G}_k$ be as defined in (5).

(a) If the stochastic gradients $G(x, \xi)$ satisfy Assumption 1.4 then we have

$$\|E[\tilde{G}_k] - \nabla f(w_k)\| \leq \sigma \left( \frac{\log(1/\delta)}{m_k} \right)^{\frac{\alpha}{1 + \alpha}}$$

and

$$P \left( \|\Delta_k\| \geq 4\sigma \left( \frac{\log(1/\delta)}{m_k} \right)^{\frac{\alpha}{1 + \alpha}} \right) \leq \delta.$$

Consequently, we also have the following moment bound

$$E \left[ \exp \left\{ \left\| \frac{\Delta_k}{\epsilon} \right\|^{\frac{3\beta + 2}{2\beta}} m_k \right\} \right] \leq C.$$

(b) If the stochastic gradients $G(x, \xi)$ satisfy Assumption 1.5 then we have

$$\|E[\tilde{G}_k] - \nabla f(w_k)\| \leq \sqrt{d} \left( \frac{\log(1/\delta)}{m_k} \right)^{\frac{\beta}{1 + \beta}}$$

and

$$P \left( \|\Delta_k\| \geq 4\sqrt{d} \left( \frac{\log(1/\delta)}{m_k} \right)^{\frac{\beta}{1 + \beta}} \right) \leq \delta.$$

Consequently, we also have the following moment bound

$$E \left[ \exp \left\{ \left\| \frac{\Delta_k}{\epsilon} \right\|^{\frac{3\beta + 2}{2\beta}} m_k \right\} \right] \leq C.$$

**Theorem 3.6.** Let Algorithm 2 be run with $\tilde{G}_k$ as defined in (5), and with parameters $\alpha_k, \gamma_k, \mu_k$ as defined in (8).

(a) Define $a(\delta, \omega) = (\log(1/\omega))^{\frac{\omega}{1 + \omega}}$ and hence $a(\delta, \alpha) = (\log(1/\delta))^{\frac{\alpha}{1 + \alpha}}$. If the stochastic gradients $G(x, \xi)$ satisfy Assumptions 1.4 and 3.1 and $m_k = O(N^{2(\alpha + 1)/\alpha})$, then

$$P \left( f(z_N) - f(x_*) \leq \frac{D_0 \sigma^2 \log(1/\delta)}{N(N + 1)} \right) \geq 1 - \delta,$$

and the SFO and LMO complexity are respectively bounded by

$$O \left( \left( \frac{a(\delta, \alpha)}{\epsilon} \right)^{\frac{3\alpha + 2}{2\alpha}} \right) \quad \text{and} \quad O \left( \frac{a(\delta, \alpha)}{\epsilon} \right).$$

(b) If the stochastic gradients $G(x, \xi)$ satisfy Assumptions 1.3 and 3.1 and $m_k = O(N^{2(\alpha + 1)/\alpha})$, then

$$P \left( f(z_N) - f(x_*) \leq \frac{D_0 \sqrt{d} \log(1/\delta)}{N(N + 1)} \right) \geq 1 - \delta,$$

and the SFO and LMO complexity are, respectively, bounded by

$$O \left( \left( \frac{\sqrt{d} a(\delta, \beta)}{\epsilon} \right)^{\frac{3\beta + 2}{2\beta}} \right) \quad \text{and} \quad O \left( \frac{\sqrt{d} a(\delta, \beta)}{\epsilon} \right),$$

where $a(\delta, \beta) = (\log(1/\delta))^{\frac{\beta}{1 + \beta}}$.

**Remark 3.7.** First note that the oracle complexities under the weaker condition in Assumption 1.5 has an additional dimension factor $\sqrt{d}$. Hence, for a fixed value of $\delta$, for $\alpha = 1$ and $\beta = 1$ (i.e., finite variance case), we have the SFO complexity to be of order $O(e^{-5/2})$ and $O(d^{5/4}e^{-5/2})$ respectively. Furthermore, note that under our assumptions, only $(1 + \alpha)$ or $(1 + \beta)$ moment exists for the stochastic gradients. Consequently, as $\alpha \to 0$ or $\beta \to 0$, for a fixed value of $0 < \epsilon < 1$ and $\delta$, the SFO complexity tends to infinity, highlighting the difficulty of the problem.

3.3 ORACLE COMPLEXITY WITH CTBJ-BASED GRADIENT ESTIMATOR

In this section, we will use the mean estimator procedure proposed recently in [Cherapanamjeri et al. 2022], and
show that the dimension factor $\sqrt{d}$ appearing in the SFO complexity in part (b) of Theorem 3.6 could be removed, even under the weaker condition in Assumption 1.5.

**Theorem 3.8.** Let Algorithm 2 be run with $G_k$ as defined in (6), and with parameters $\alpha_k, \gamma_k$ and $\mu_k$ as defined in (8). If the stochastic gradients $G(x, \xi)$ satisfy Assumptions 1.5 and 3.1 and $m_k = \mathcal{O}(N(N + 1))$, then with a target confidence $\delta > 2^{-m_k}$ and $d \lesssim \log(1/\delta)$, we have
\[
\mathbb{P} \left( f(z_N) - f(x_*) \leq \frac{D_0 \max \{a(\delta, \beta), \frac{1}{\epsilon}a(\delta, \beta)^2\}}{N(N + 1)} \right) \geq 1 - \delta,
\]
and the SFO and LMO complexity are respectively bounded by
\[
\mathcal{O} \left( \frac{a(\delta, \beta)}{\epsilon} \right)^{\frac{3\beta + 2}{2\beta}} \quad \text{and} \quad \mathcal{O} \left( \frac{a(\delta, \beta)}{\epsilon} \right).
\]

**Remark 3.9.** The proof of the above theorem is based on a concentration result for the gradient estimator (6), established in [Cherapanamjeri et al., 2022]. In comparison to part (b) of Theorem 3.6, the SFO complexity in Theorem 3.8 under Assumption 1.5 does not have the additional dimensional factor $\sqrt{d}$, thereby demonstrating the benefits of using the mean-estimation procedure proposed in [Cherapanamjeri et al., 2022]. However, this improvement does not come for free, as the per-iteration complexity of using the robust mean-estimator (6) is significantly higher than that of the clipped gradient based robust mean-estimator in (6), although it has a polynomial dependency on the problem parameters. See Section 5 for details regarding the per-iteration computational complexity of (6).

### 3.4 IMPROVING THE ORACLE COMPLEXITY

The above oracle complexity results based on clipped gradient-based and CBJ-based robust gradient estimators have the following drawbacks. The $\epsilon$-dependency of the SFO complexity in part (a) of Theorem 3.6 under Assumption 1.4 or Theorem 3.8 under Assumption 1.5 is $\mathcal{O}(\epsilon^{-2.5})$ when $\alpha = 1$ and $\beta = 1$. In this section, we propose a bias-corrected clipped gradient based robust gradient estimation procedure with which SFO complexity of Algorithm 2 could be improved to $\mathcal{O}(d^2 \epsilon^{-2})$ under an additional symmetry assumption on the distribution of the stochastic gradient. Hence, when high-accuracy solutions in low-dimensional settings are required, the bias-corrected clipped gradient based robust SFW algorithm could be preferred. We now introduce the symmetry assumption and the clipped gradient procedure.

**Assumption 3.10.** Let the distribution of $G(x, \xi)$ be continuous and symmetric about $\mathbb{E}[G(x, \xi)] = \nabla f(x)$, for all $x \in \mathbb{R}^d$. Also, let the probability density function be a decreasing function with respect to $||G(x, \xi) - \nabla f(x)||$, for all $x \in \mathbb{R}^d$.

**Proposition 3.11.** For iteration $k$ the i.i.d. stochastic gradients $G(w_k, \xi_{k,j})$ are assumed to satisfy Assumptions 1.3 and 3.10. Let $\delta \in (0, 1)$. Consider the initial estimate defined as
\[
\hat{G}_k := \arg \min_{G(w_k, \xi_{k,j})} \min \left\{ r \geq 0 : \sum_{t = m_k}^{m_k} \mathbb{I} \{ ||G(w_k, \xi_{k,t}) - G(w_k, \xi_{k,j})|| \leq r \} \geq 0.3m_k \right\},
\]
and consider the bias-corrected clipped gradient estimator $\tilde{G}_k$ defined as
\[
\tilde{G}_k := \frac{2}{m_k} \sum_{t = 1}^{m_k/2} \min \left\{ \left( \frac{t}{\log(1/\delta)} \right)^{1/\beta} + 2 \sqrt{d} \right\} \times \mathbb{E}[\hat{G}_k] = \nabla f(w_k) \quad \text{and} \quad \mathbb{P} \left\{ ||\Delta_k|| \leq 8\pi \sqrt{d} \right\} \geq 1 - \delta.
\]

**Remark 3.12.** The initial estimate defined in (9) is the same as that in the CBJ estimator in (6). We show that this initial step, along with the clipped gradient procedure for a specific choice of clipping parameter (as defined in (10)) helps obtain an unbiased gradient estimator which is sufficiently concentrated.

We now leverage the result in Proposition 3.11 and show that one could obtain improved SFO complexity for certain ranges of $d$ and $\epsilon$ when running Algorithm 2 with the robust gradient estimator (10).

**Theorem 3.13.** Let Algorithm 2 be run with $G_k$ as defined in (10), and with parameters $\alpha_k, \gamma_k$ and $\mu_k$ as defined in (8).

(a) If the stochastic gradients $G(x, \xi)$ satisfy Assumptions 1.4, 3.1 and 3.10 and $m_k = \mathcal{O}(N^{\frac{2(a+1)}{\alpha}})$, we have for
\[
\mathbb{P} \left( f(z_N) - f(x_*) \leq \frac{C(d, \alpha, \delta)D_0 \sigma}{N(N + 1)} \right) \leq \delta,
\]
where
\[
C(d, \alpha, \delta) = \max \left\{ \frac{1}{\sqrt{d}}a(\delta, \alpha)^{\frac{3}{2}}, a(\delta, \alpha)^2 \right\},
\]
and the SFO and LMO complexity are respectively bounded by
\[
\mathcal{O} \left( \frac{C(d, \alpha, \delta)}{\epsilon} \right) \quad \text{and} \quad \mathcal{O} \left( \frac{C(d, \alpha, \delta)}{\epsilon} \right).
\]
If the stochastic gradients $G(x, \xi)$ satisfy Assumptions 3.1 and 3.10 and $m_k = \mathcal{O}(N^{-\frac{d+1}{2}})$, the same conclusion in (11) holds with $\alpha$ replaced by $\beta$, with $C(d, \beta, \delta) := C(d, \alpha = \beta, \delta)$, for $(\log(1/\delta))^{\frac{1}{\beta}} \geq \left[\Gamma\left(\frac{\beta}{\beta + \delta}\right)\right]^{\frac{1}{\beta + \delta}}$.

Remark 3.14. Note that when $\alpha = 1$, for any fixed value of $\delta$, $C(d, \alpha, \delta)$ is $\mathcal{O}(d)$. Hence, the SFO complexity is of order $\mathcal{O}(d^2 \epsilon^{-2})$. Hence, when $d^2 < o(\epsilon^{-0.5})$ the SFO complexity of part (a) of Theorem 3.13 is better than part (a) of Theorem 3.6. A similar improvement holds for part (b). In Figure 3 (Appendix–Section 2), we compare Theorem 3.6, of Theorem 3.6. A similar improvement holds for part (b). In Figure 1, we report the performance of Algorithm 2 with the clipped gradient estimator $\mathcal{E}$ and mini-batch average estimator $\mathcal{E}$. We also compare against the 1-sample SFW method from Mokhtari et al. [2020]. From the top row, we see that the clipped gradient estimator has faster convergence, i.e., it achieves higher accuracy with lesser iterations compared to the standard mini-batch averaging or the 1-sample SFW method. Furthermore, from the histogram in the bottom row, we see that the distribution of the last iterate has more fluctuations for the mini-batch average estimator and the 1-sample SFW method, compared to the clipped gradient estimator. We quantify this statement by reporting the standard deviation and also the heavy-tailed index, a widely used metric to quantify fluctuations [Hoaglin et al. 2000], which is defined as

$$\tau(F) = \frac{F^{-1}(0.95) - F^{-1}(0.5)}{\Phi^{-1}(0.95) - \Phi^{-1}(0.5)}$$

where $\Phi$ is the distribution of a standard normal and $F$ is the empirical CDF obtained from the histogram. The results in Figure 1 confirm our theoretical results and highlight the benefits of using robust versions of SFW algorithms for dealing with heavy-tailed data arising practice.

Summary and Outlook: We proposed and analyzed robust versions of stochastic Frank-Wolfe type algorithms and established high-probability oracle complexity results. Our theoretical results are supported by numerical experiments on the problem of sparse linear regression and multi-class

### 4 CONSEQUENCES FOR PRACTICE

We now demonstrate the consequences of our theoretical results in practice. Among the robust gradient estimators in (5), (6) and (10), the most practical one (i.e., least per-iteration complexity) is the clipped gradient estimator in (5). Hence, we compare Algorithm 2 with the mini-batch average gradient in (4) and the clipped gradient estimator in (5) via experiments on sparse linear regression and multi-class logistic regression (presented in Appendix–Section 1 due to space limitations) with heavy-tailed covariates.

**Sparse Linear Regression:** We now provide simulation results for the regression problem described in (2). For our experiments, we consider the data vector $a \in \mathbb{R}^d$ to be a Pareto distribution with the exponent being 2.2 (which is asymptotically a $t$-distribution with degrees of freedom 2.2). We ran Algorithm 2 with parameters as defined in (8) for 100 trails. Here, $L$ could be calculated analytically for our problem. For the choice of batch size, while Theorem 3.6 suggests $m_k = \mathcal{O}(N^{-\frac{d+1}{2}})$, we found that in our experiments setting $m_k = 500$ works well already. In Figure 1, we report the performance of Algorithm 2 with the clipped gradient estimator $\mathcal{E}$ and mini-batch average estimator $\mathcal{E}$. We also compare against the 1-sample SFW method from Mokhtari et al. [2020]. From the top row, we see that the clipped gradient estimator has faster convergence, i.e., it achieves higher accuracy with lesser iterations compared to the standard mini-batch averaging or the 1-sample SFW method. Furthermore, from the histogram in the bottom row, we see that the distribution of the last iterate has more fluctuations for the mini-batch average estimator and the 1-sample SFW method, compared to the clipped gradient estimator. We quantify this statement by reporting the standard deviation and also the heavy-tailed index, a widely used metric to quantify fluctuations [Hoaglin et al. 2000], which is defined as

$$\tau(F) = \frac{F^{-1}(0.95) - F^{-1}(0.5)}{\Phi^{-1}(0.95) - \Phi^{-1}(0.5)}$$

where $\Phi$ is the distribution of a standard normal and $F$ is the empirical CDF obtained from the histogram. The results in Figure 1 confirm our theoretical results and highlight the benefits of using robust versions of SFW algorithms for dealing with heavy-tailed data arising practice.
logistic regression with heavy-tailed data. Developing oracle complexity results for robust projection-free algorithms under non-convexity, and developing more practical versions of robust projection-free algorithms are interesting problems that we plan to examine as future work.

**References**


