Complexities in Projection-Free Stochastic Non-convex Minimization

Zebang Shen[1] **Cong Fang**[2] **Peilin Zhao**[3] **Junzhou Huang**[3] **Hui Qian**[*1]
[1]Zhejiang University [2]Peking University [3]Tencent AI Lab

Abstract

For constrained nonconvex minimization problems, we propose a meta stochastic projection-free optimization algorithm, named Normalized Frank Wolfe Updating, that can take any Gradient Estimator (GE) as input. For this algorithm, we prove its convergence rate, regardless of the choice of GE. Using a sophisticated GE, this algorithm can significantly improve the Stochastic First order Oracle (SFO) complexity. Further, a new second order GE strategy is proposed to incorporate curvature information, which enjoys theoretical advantage over the first order ones. Besides, this paper also provides a lower bound of Linear-optimization Oracle (LO) queried to achieve an approximate stationary point. Simulation studies validate our analysis under various parameter settings.

1 Introduction

We consider the constrained minimization problem

$$\min_{\mathbf{x}\in\mathcal{C}} f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}), \tag{1}$$

where each $f_i : \mathcal{C} \mapsto \mathbb{R}$ is an L-smooth but potentially *nonconvex* component function and $\mathcal{C} \subseteq \mathbb{R}^d$ is a compact convex feasible set. The goal is to find an ϵ-approximate first-order stationary point $\mathbf{x}$ such that

$$V_{\mathcal{C}}(\mathbf{x}; f) = \max_{\mathbf{u}\in\mathcal{C}} \langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{u}\rangle \leq \epsilon D, \tag{2}$$

where D is the diameter of $\mathcal{C}$.
Many important modern applications can be cast into the form of (1), for example robust lower rank matrix recovery (Qu et al., 2017), non-monotone submodular maximization (Bian et al., 2017), multiple sequence alignment (Alayrac et al., 2016), and multi-object tracking (Chari et al., 2015). These applications of interest have two important characteristics:
(i) The number of component functions n is large and hence stochastic methods are preferable to deterministic ones due to their low per-iteration cost. Note that the number of component function gradient evaluations $\nabla f_i(\cdot)$ is used as a cost measurement and is referred as the *Stochastic First-order Oracle (SFO)* complexity.
(ii) Solving linear optimization over the feasible set $\mathcal{C}$

$$L_{\mathcal{C}}(\mathbf{c}) := \operatorname*{argmin}_{\mathbf{x}\in\mathcal{C}} \mathbf{c}^\top \mathbf{x} \tag{3}$$

is more efficient than projecting on $\mathcal{C}$. Therefore the linear optimization based methods (also known as *projection-free*) have their natural advantage over the projection-based ones. Further, the amount of linear optimizations solved during the optimization procedure is used as another efficiency measurement and is referred as the *Linear-optimization Oracle (LO)* complexity.

The Frank-Wolfe (FW) algorithm and its stochastic variants are perhaps the most noteworthy methods that exploit the above problem structures (Frank and Wolfe, 1956; Clarkson, 2010; Jaggi, 2013; Hazan and Luo, 2016). These methods typically alternate between the following two steps in each iteration:
(i) computing $\mathbf{v}^t = L_{\mathcal{C}}(\mathbf{g}^t)$ by querying the linear optimization oracle, where $\mathbf{g}^t$ is some estimation of the full gradient $\nabla f(\mathbf{x}^t)$,
(ii) updating the iterate $\mathbf{x}^{t+1}$ as the convex combination of $\mathbf{x}^t$ and $\mathbf{v}^t$.
While originally designed for convex programming, a few attempts are made to generalize FW to the case with no convexity assumption. Lacoste-Julien (2016) shows that, by appropriately choosing parameters, the vanilla FW method solves nonconvex problem (1) at the cost of $\mathcal{O}(n/\epsilon^2)$ SFO queries and $\mathcal{O}(1/\epsilon^2)$ LO evaluations. Lafond et al. (2015) provide an online version FW in a full informational setting and obtain the similar SFO and LO complexity to achieve (2). Recently, Reddi et al. (2016) propose a variance reduced stochastic variant of FW, namely Stochastic Variance Reduced

[*]Corresponding author.

Proceedings of the 22[nd] International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

Frank Wolfe (SVRFW), to reduce the SFO complexity to $\mathcal{O}(n^{2/3}/\epsilon^2)$ at the same LO cost.
Despite the progress, the projection-free optimization for nonconvex problems remains largely unexplored:
(1) In the unconstrained setting, Fang et al. (2018) show that $\mathcal{O}(n^{1/2}/\epsilon^2)$ SFO queries are sufficient to obtain a point such that $\|\nabla f(\mathbf{x}^t)\| \leq \epsilon$. Can the same complexity be achieved for the projection-free optimization?
(2) While the current methods only use the gradient information, how can we incorporate second order information into the Frank-Wolfe method?
(3) Existing methods all have the consensus $\mathcal{O}(1/\epsilon^2)$ LO cost. Is it improvable or is it optimal already?

To answer these questions, we first design a meta-algorithm that takes the Gradient Estimation (GE) strategy as input and accepts existing methods are its instantiations. For this meta-algorithm, a unified convergence rate is obtained regardless of the GE choice. This constitutes the first contribution of this paper.

[C1] We formalize the aforementioned two-step update into a meta-algorithm named Normalized Frank Wolfe Update (NFWU), which takes the gradient estimator as an input. We show $\mathcal{O}(1/\epsilon^2)$ iterations of NFWU are sufficient to find an ϵ-approximate first-order stationary point if the gradient estimator meets some accuracy criterion. Plugging in a more sophisticated estimator named Stochastic Path-Integrated Differential EstimatoR (SPIDER)[1] (Fang et al., 2018), we derive an FW variant that achieves the $\mathcal{O}(n^{1/2}/\epsilon^2)$ SFO cost.

Fang et al. (2018) show that the $\mathcal{O}(n^{1/2}/\epsilon^2)$ SFO complexity is already optimal among the first-order methods. However, when second order information is available, we can further reduce the SFO complexity by designing a curvature aided gradient estimator, whose development is the second contribution of this paper:

[C2] We construct curvature aided gradient estimators that further improve the overall SFO complexity to $\mathcal{O}(\max\{n^{0.75}/\epsilon^{1.5}, 1/\epsilon^2\})$. For the robust low rank matrix recovery problem, we analyze the computation complexity of the newly proposed estimators and show that the only additional cost is a matrix-vector product.

While the former discussions are primarily about improving the efficiency of projection-free algorithms, this paper also analyzes the hardness of Problem (1) from the perspective of the inevitable LO complexity, which leads to our third contribution:

[C3] We show that the $\mathcal{O}(1/\epsilon^2)$ is the LO query lower bound to achieve (2) using only first-order oracle.

A simulation study on Robust Low Rank Matrix Recovery in conducted to validate our analysis and to show the advantage of curvature-aided estimators. The results are promising.

[1] We note that such estimator is also known as SARAH and appears in Nguyen et al. (2017).

2 Related Work and Preliminaries

2.1 Projection-free Strategies

For various important feasible sets, linear programming $L_{\mathcal{C}}(\cdot)$ is significantly more efficient than the projection operation $P_{\mathcal{C}}(\cdot)$, as the latter is in essence a convex quadratic program over the domain. For example:

- When $\mathcal{C} = \{\mathbf{X} \in \mathbb{R}^{M\times N} : \|\mathbf{X}\|_* \leq B\}$, the nuclear norm ball, the projection operation $P_{\mathcal{C}}(\mathbf{Y})$ computes the expensive singular value decomposition (SVD) of the $M \times N$ sized matrix $\mathbf{Y}$. On the other hand, the linear optimization $L_{\mathcal{C}}(\mathbf{Y}) := \operatorname{argmin}_{\mathbf{X}\in\mathcal{C}} \operatorname{tr}\{\mathbf{Y}^\top\mathbf{X}\}$ amounts to finding only the top singular vectors, which is much cheaper (Candès and Recht, 2009; Freund et al., 2017).
- When $\mathcal{C}$ is a structured polytope arising in combinatorial optimization, e.g. the unit flow polytope of a graph, the perfect matching polytope of a bipartite graph, and the base polyhedron of a matroid, highly efficient combinatorial algorithms exist for linear optimization (Schrijver, 2003). However, the projection operation would require general interior point solvers that neglect the specific structure of the feasible set and is hence inefficient.

We now give a concrete constrained nonconvex minimization problem with $\mathcal{C}$ being the nuclear norm ball.

Example (Low Rank Matrix Recovery (LRMR))**.** *LRMR plays a key role in solving many important learning tasks, such as collaborative filtering (Koren et al., 2009), dimensionality reduction (Weinberger and Saul, 2006), and multi-class learning (Xu et al., 2013). The loss of LRMR writes*

$$\begin{aligned} \min_{\mathbf{X}\in\mathbb{R}^{M\times N}} \quad & \sum_{(i,j)\in\Omega} \psi(\mathbf{X}_{ij} - \mathbf{Y}_{ij}) \\ s.t. \quad & \|\mathbf{X}\|_* \leq B, \end{aligned} \tag{4}$$

where $\psi : \mathbb{R} \to \mathbb{R}$ is the potentially nonconvex empirical loss function, $\mathbf{X}_{ij}$ is the i,j th element of matrix $\mathbf{X}$, and Ω is the set of observed indices in target matrix $\mathbf{Y} \in \mathbb{R}^{M\times N}$. Here we focus on a robust version of LRMR with the loss ψ being:

$$\psi(z;\sigma) = 1 - \exp(-z^2/2\sigma), \tag{5}$$

where σ is a tunable parameter. Loss (5) is less sensitive to the discrepancy $\mathbf{X}_{ij} - \mathbf{Y}_{ij}$ compared to the common least square loss $\psi(z) = z^2/2$, and hence is robust to adversarial outliers (Qu et al., 2017).

2.2 Non-convex Projection-Free Methods

Here we briefly review the nonconvex projection-free methods. As far as we know, there are two lines of research: one that follows the Frank-Wolfe (FW) method, and the other that is based on the Conditional Gradient Sliding (CGS) methods Lan and Zhou (2016).

While mainly developed for convex programming, there are only two stochastic nonconvex FW variants, namely SFW and SVRFW (Reddi et al., 2016). They share the two-step update structure discussed in Section 1 and differ only in their construction of the gradient estimator $\mathbf{g}^t$: SFW uses the vanilla mini-batch stochastic gradient to approximate the full gradient; SVRFW utilizes the variance reduced gradient estimator SVRG proposed in (Johnson and Zhang, 2013). For a uniformed exposition, we will describe the detailed estimator construction in Section 3.1. In terms of the oracle complexities, $\mathcal{O}(1/\epsilon^4)$ and $\mathcal{O}(n^{2/3}/\epsilon^2)$ SFO queries are required by SFW and SVRFW to achieve the ϵ-approximate first-order stationarity respectively.

CGS type methods converge to a stationary point in a different sense from (2) and hence are not the focus of this paper. Specifically, they terminate the procedure when change between two consecutive iterations are small, $\mathbb{E}\|\mathbf{x}^{t+1}-\mathbf{x}^t\|^2=\mathcal{O}(\epsilon^2)$. Additionally, in their analysis, higher LO complexity ($\mathcal{O}(1/\epsilon^4)$ in general) is demanded to reach such goal, (Qu et al., 2017).

Assumptions and Notations

We now list the assumptions used in this paper.
(Finite function value) The objective function is bounded from below in $\mathcal{C}$: for all $\mathbf{x}\in\mathcal{C}$,

$$f(\mathbf{x})-\min_{\mathbf{y}\in\mathcal{C}} f(\mathbf{y})\le\Delta, \tag{6}$$

with Δ being some non-negative constant.
(First-order smoothness) We assume $f_i(\cdot)$ has L-Lipschitz continuous gradient in expectation:

$$\mathbb{E}_i\|\nabla f_i(\mathbf{x})-\nabla f_i(\mathbf{y})\|\le L\|\mathbf{x}-\mathbf{y}\|. \tag{7}$$

(Second-order smoothness) To analyze the curvature aided estimator, we assume the Hessian of each component function to be L_2-Lipschitz in expectation:

$$\mathbb{E}_i\|\nabla^2 f_i(\mathbf{x})-\nabla f_i^2(\mathbf{y})\|\le L_2\|\mathbf{x}-\mathbf{y}\|. \tag{8}$$

We now define the notations of expectations. $\mathbb{E}_{0:t}$ denotes the expectation w.r.t. all the randomness until iteration t. $\mathbb{E}_t$ denotes the expectation w.r.t. only the randomness in iteration t. $\mathbb{E}_{[i]}$ denotes the expectation w.r.t. the randomness in the output of Algorithm 1.

Algorithm 1 Normalized Frank-Wolfe Update

Input: initialization $\mathbf{x}^0$, domain size D, step size η^t, max iteration T, gradient estimator **GE**;
1: **for** $t=0$ **to** $T-1$ **do**
2: compute gradient estimator $\mathbf{g}^t$ using **GE**;
3: compute $\mathbf{v}^t=L_{\mathcal{C}}(\mathbf{g}^t)$ using LO
4:
$$\mathbf{x}^{t+1}=\mathbf{x}^t+\frac{\eta^t}{D}(\mathbf{v}^t-\mathbf{x}^t)$$
5: **end for**
Output: $\mathbf{x}^{t_0}$ with t_0 uniformly sampled from $[T]$;

3 Normalized Frank-Wolfe Update

We describe our meta-algorithm, namely the Normalized Frank Wolfe Update (NFWU), in Algorithm 1. NFWU accepts the gradient strategy as input and shares the two-step update structure: in line 3, it computes $\mathbf{v}^t$ by querying the LO and in line 4, it updates the variable $\mathbf{x}^{t+1}$ by a convex combination. A slight but important difference from the standard Frank-Wolfe mthod is that NFWU explicitly normalizes the update $\mathbf{v}^t-\mathbf{x}^t$ to ensure that the distance $\mathbf{x}^t$ travels per-iteration is precisely controlled by the step-size parameter η^t. Such normalization turns out to be critical to the convergence analysis and the amortized per-iteration SFO cost of efficient variance reduced estimators. The following theorem depicts the number of iterations NFWU takes to obtain an ϵ-approximate first-order stationary point, under the premise that $\mathbf{g}^t$ is sufficiently accurate.

Theorem 3.1 (Convergence in Expectation)**.** *Recall the definition of the expectations $\mathbb{E}_{0:t}$, $\mathbb{E}_t$, and $\mathbb{E}_{[i]}$. In Algorithm 1, by setting $\eta=\epsilon/L$ and assuming that*

$$\mathbb{E}_{0:t}[\|\nabla f(\mathbf{x}^t)-\mathbf{g}^t\|^2]\le\epsilon^2/4, \tag{9}$$

we have the convergence in expectation

$$\mathbb{E}_{[t_0]}[\mathbb{E}_{0:t_0-1}[V_{\mathcal{C}}(\mathbf{x}^{t_0};f)]]\le 5\epsilon, \tag{10}$$

where t_0 is uniformly sampled from $[T]=\{1,\dots,T\}$ with $T=L(f(\mathbf{x}^1)-f(\mathbf{x}^*))/\epsilon^2$.

Theorem 3.1 enables us to analyze the overall SFO complexity of any instantiation of NFWU in two steps: (1) calculating the per-iteration amortized SFO cost to compute a qualified gradient estimator $\mathbf{g}^t$ and (2) multiplying such amortized cost with $\mathcal{O}(1/\epsilon^2)$. We leave the task of analyzing the amortized SFO cost for specific $\mathbf{g}^t$ constructions to the following sections. Additionally, under Theorem 3.1, $\mathcal{O}(1/\epsilon^2)$ LO accesses are required to achieve (2). This matches existing results in DFW, SFW, and SVRFW, and is shown to be optimal among first-order methods in Section 5.

Proof of Theorem 3.1. From the smoothness of f

$$\begin{aligned}
f(\mathbf{x}^{t+1}) &\leq f(\mathbf{x}^t) + \langle \nabla f(\mathbf{x}^t), \mathbf{x}^{t+1} - \mathbf{x}^t \rangle + \frac{L}{2}\|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 \\
&= f(\mathbf{x}^t) + \langle \mathbf{g}^t, \mathbf{x}^{t+1} - \mathbf{x}^t \rangle + \frac{L}{2}\|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 \\
&\quad + \langle \nabla f(\mathbf{x}^t) - \mathbf{g}^t, \mathbf{x}^{t+1} - \mathbf{x}^t \rangle \\
&= f(\mathbf{x}^t) - \frac{\eta}{D}\langle \mathbf{g}^t, \mathbf{x}^t - \mathbf{v}^t \rangle + \frac{L\eta^2}{2D^2}\|\mathbf{u}^t - \mathbf{x}^t\|^2 \\
&\quad + \frac{\eta}{D}\langle \nabla f(\mathbf{x}^t) - \mathbf{g}^t, \mathbf{v}^t - \mathbf{x}^t \rangle \\
&\leq f(\mathbf{x}^t) - \frac{\eta}{D}\langle \mathbf{g}^t, \mathbf{x}^t - \mathbf{v}^t \rangle + \frac{L\eta^2}{2D^2}\|\mathbf{u}^t - \mathbf{x}^t\|^2 \\
&\quad + 2\eta\|\nabla f(\mathbf{x}^t) - \mathbf{g}^t\|.
\end{aligned}$$

Denoting $\mathbf{v}^+ = \operatorname{argmax}_{\mathbf{v}\in C}\langle \nabla f(\mathbf{x}^t), \mathbf{x}^t - \mathbf{v}\rangle$, we have

$$\begin{aligned}
V_{\mathcal{C}}(\mathbf{x}^t; f) &= \langle \nabla f(\mathbf{x}^t) - \mathbf{g}^t, \mathbf{x}^t - \mathbf{v}^+ \rangle + \langle \mathbf{g}^t, \mathbf{x}^t - \mathbf{v}^+ \rangle \\
&\leq \|\nabla f(\mathbf{x}^t) - \mathbf{g}^t\|\|\mathbf{x}^t - \mathbf{v}^+\| + \langle \mathbf{g}^t, \mathbf{x}^t - \mathbf{v}^t \rangle.
\end{aligned}$$

These two bound together gives

$$\frac{\eta}{D}V_{\mathcal{C}}(\mathbf{x}^t; f) \leq f(\mathbf{x}^t) - f(\mathbf{x}^{t+1}) + 4\eta\|\nabla f(\mathbf{x}^t) - \mathbf{g}^t\| + 2L\eta^2.$$

From the choice of step-size $\eta = \epsilon/L$ and accuracy requirement on the gradient estimator $\mathbb{E}_{0:t}[\|\nabla f(\mathbf{x}^t) - \mathbf{g}^t\|^2] \leq \epsilon^2/4$, we have for $t \geq 1$

$$\begin{aligned}
&\frac{\epsilon}{L}\mathbb{E}_{0:t-1}[V_{\mathcal{C}}(\mathbf{x}^t; f)] \\
\leq\ & \mathbb{E}_{0:t-1}[f(\mathbf{x}^t)] - \mathbb{E}_{0:t}[f(\mathbf{x}^{t+1})] + 2L\eta^2 \\
&+ 4\eta\mathbb{E}_{0:t}[\|\nabla f(\mathbf{x}^t) - \mathbf{g}^t\|] \\
\leq\ & \mathbb{E}_{0:t-1}[f(\mathbf{x}^t)] - \mathbb{E}_{0:t}[f(\mathbf{x}^{t+1})] + 4\epsilon^2/L.
\end{aligned}$$

Sum the above inequalities from $t = 1$ to T to obtain

$$\sum_{t=1}^{T}\frac{\epsilon}{L}\mathbb{E}_{0:t-1}[V_{\mathcal{C}}(\mathbf{x}^t; f)] \leq f(\mathbf{x}^1) - f(\mathbf{x}^*) + T\cdot\frac{4\epsilon^2}{L}.$$

Hence, by sampling t_0 from $[T]$ uniformly, we have

$$\mathbb{E}_{[t_0]}[\mathbb{E}_{1:t_0-1}[V_{\mathcal{C}}(\mathbf{x}^t; f)]] \leq \frac{L(f(\mathbf{x}^1) - f(\mathbf{x}^*))}{T\epsilon} + 4\epsilon,$$

and thus when $T = L(f(\mathbf{x}^1) - f(\mathbf{x}^*))/\epsilon^2$, we have $\mathbb{E}_{[t_0]}[\mathbb{E}_{1:t_0-1}[V_{\mathcal{C}}(\mathbf{x}^t; f)]] \leq 5\epsilon$. □

Remark 3.1 (Convergence with High Probability)**.** *We derive the iteration complexity to obtain a high-probability convergence result. Denote $X = V_{\mathcal{C}}(\mathbf{x}^{t_0}; f)/10$, where $\mathbf{x}^{t_0}$ is the output of NFWU. We have $X \geq 0$ and $\mathbb{E}[X] \leq \epsilon/2$ under the setting of Theorem 3.1. Using the Markov's Inequality, we obtain*

$$Pr[X \geq 2\mathbb{E}[X]] \leq 0.5.$$

GE 1 SG($|\mathcal{S}^t|$)

1: sample $|\mathcal{S}^t|$ i.i.d. component functions uniformly;
2: $\mathbf{g}^t = \frac{1}{|\mathcal{S}^t|}\sum_{i\in\mathcal{S}^t}\nabla f_i(\mathbf{x}^t)$;

GE 2 SVRG(p, $|\mathcal{S}^t|$)

1: **if** $mod(t, p) = 0$
$\tilde{\mathbf{x}} = \mathbf{x}^t$, $\tilde{\mathbf{g}} = \nabla f(\tilde{\mathbf{x}})$;
2: sample $|\mathcal{S}^t|$ i.i.d. component functions uniformly;
3: $\mathbf{g}^t = \frac{1}{|\mathcal{S}^t|}\sum_{i\in\mathcal{S}^t}[\nabla f_i(\mathbf{x}^t) - \nabla f_i(\tilde{\mathbf{x}})] + \tilde{\mathbf{g}}$;

Let $\{X_i\}_{i=1}^k$ be a set of k i.i.d. samples of X. We have $Pr[\min_i X_i \geq 2\mathbb{E}[X]] \leq 0.5^k$. Thus, by setting $k = \mathcal{O}(\log(1/\delta))$, we derive

$$Pr[\min_i X_i \leq 2\mathbb{E}[X] \leq \epsilon] \geq 1 - \delta,$$

where δ is the failure probability. This means at least one of X_i satisfies (2).
Therefore, to find a point $\mathbf{x}_\delta$ such that with high probability $1-\delta$, $V_{\mathcal{C}}(\mathbf{x}_\delta; f) \leq \epsilon$, we take $\mathcal{O}(\log(1/\delta))$ independent NFWU trials, each of which consist $\mathcal{O}(1/\epsilon^2)$ iterations, and pick the output with minimum $V_{\mathcal{C}}$ value among all these trials. The overall iterations will be $\mathcal{O}(\log(1/\delta)/\epsilon^2)$ at the additional cost $\mathcal{O}(n\log(1/\delta))$.

3.1 SFW and SVRFW

SFW and SVRFW can be regarded as instantiations of NFWU that use the vanilla Stochastic Gradient (SG) and Stochastic Variance Reduced Gradient (SVRG) as gradient estimator respectively. We recover their SFO complexities using Theorem 3.1.

SFW = NFWU+SG:
With $\mathcal{S}^t$ being an index set uniformly drawn from $[n]$, the simplest mini-batch stochastic gradient writes

$$\mathbf{g}^t = \frac{1}{|\mathcal{S}^t|}\sum_{i\in\mathcal{S}^t}\nabla f_i(\mathbf{x}^t). \tag{11}$$

Using the smoothness assumption (7) and the boundedness of domain $\mathcal{C}$, we have $\mathbb{E}_i\|\nabla f_i(\mathbf{x})\|^2 \leq L^2D^2 \stackrel{\text{def}}{=} G^2$ for all $\mathbf{x} \in \mathbb{R}^d$, and, if we have $|\mathcal{S}^t| = G^2/\epsilon^2$, we obtain

$$\begin{aligned}
\mathbb{E}_{\mathcal{S}}^t\|\mathbf{g}^t - \nabla f(\mathbf{x}^t)\|^2 &= \frac{1}{|\mathcal{S}^t|}\mathbb{E}_i\|\nabla f_i(\mathbf{x}^t) - \nabla f(\mathbf{x}^t)\|^2 \\
&\leq \frac{1}{|\mathcal{S}^t|}\mathbb{E}_i\|\nabla f_i(\mathbf{x}^t)\|^2 \leq \frac{G^2}{|\mathcal{S}^t|} = \epsilon^2,
\end{aligned}$$

We conclude that the overall SFO complexity using the SG approximation is $\mathcal{O}(1/\epsilon^4)$.

SVRFW = NFWU+SVRG:
Following the strategy in (Johnson and Zhang, 2013), SG approximation can be improved by the variance

GE 3 SPIDERG(p, $|\mathcal{S}^t|$)

1: **if** $mod(t,p)=0$
  $\mathbf{g}^t=\nabla f(\mathbf{x}^t)$;
2: **else**
  sample $|\mathcal{S}^t|$ i.i.d. component functions uniformly;
  $\mathbf{g}^t=\frac{1}{|\mathcal{S}^t|}\sum_{i\in\mathcal{S}^t}\left[\nabla f_i(\mathbf{x}^t)-\nabla f_i(\mathbf{x}^{t-1})\right]+\mathbf{g}^{t-1}$;

GE 4 CASVRG(p, $|\mathcal{S}^t|$)

1: **if** $mod(t,p)=0$
  $\tilde{\mathbf{x}}=\mathbf{x}^t$, $\tilde{\mathbf{g}}=\nabla f(\tilde{\mathbf{x}})$, $\tilde{\mathbf{U}}=\nabla^2 f(\tilde{\mathbf{x}})$;
2: sample $|\mathcal{S}^t|$ i.i.d. component functions uniformly;
3: $\mathbf{c}^t=[\tilde{\mathbf{U}}-\frac{1}{|\mathcal{S}^t|}\sum_{i\in\mathcal{S}^t}\nabla^2 f_i(\tilde{\mathbf{x}})](\mathbf{x}^t-\tilde{\mathbf{x}})$;
4: $\mathbf{g}^t=\frac{1}{|\mathcal{S}^t|}\sum_{i\in\mathcal{S}^t}\left[\nabla f_i(\mathbf{x}^t)-\nabla f_i(\tilde{\mathbf{x}})\right)]+\tilde{\mathbf{g}}+\mathbf{c}^t$;

reduction technique. The SVRG strategy maintains a reference vector $\tilde{\mathbf{x}}$ and a fixed reference gradient $\tilde{\mathbf{g}}$ in an epoch-wise manner. Between each update to $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{g}}$, p iterations are conducted. The details are given in Algorithm 2. Under the assumption 7, we bound

$$\begin{aligned}&\mathbb{E}_{0:t}[\|\mathbf{g}^t-\nabla f(\mathbf{x}^t)\|^2]\\&=\mathbb{E}_{0:t-1}[\mathbb{E}_{\mathcal{S}^t}[\|\mathbf{g}^t-\nabla f(\mathbf{x}^t)\|^2|\mathbf{x}^t]]\\&=\mathbb{E}_{0:t-1}\left[\frac{1}{|\mathcal{S}^t|}\mathbb{E}_i[\|\nabla f_i(\mathbf{x}^t)-\nabla f_i(\tilde{\mathbf{x}}^s)\|^2|\mathbf{x}^t]\right]\\&\le\frac{L^2}{|\mathcal{S}^t|}\mathbb{E}_{0:t-1}[\|\mathbf{x}^t-\tilde{\mathbf{x}}^s\|^2]\le\frac{p^2\epsilon^2}{|\mathcal{S}^t|},\end{aligned}$$

where the last inequality is from the choice of $\eta=\epsilon/L$ in Theorem 3.1. By setting $|\mathcal{S}^t|=n^{2/3}$ and $p=n^{1/3}$, we have $\mathbb{E}_{0:t}[\|\mathbf{g}^t-\nabla f(\mathbf{x}^t)\|^2]\le\epsilon^2$ and the SFO complexity in the entire epoch is $n+|\mathcal{S}^t|\times p=2n$. Hence, we conclude the amortized per-iteration cost of SVRG is $2n/p=2n^{2/3}$, and the overall SFO complexity using the SVRG approximation in NFWU is $\mathcal{O}(n^{2/3}/\epsilon^2)$.

Remark 3.2. *We note a method named SAGAFW is proposed in Reddi et al. (2016) and is claimed to achieve the $\mathcal{O}(n^{1/3}/\epsilon^2)$ SFO complexity. We find some errors in their proof and exclude it from comparison. See a detailed discussion in the Appendix 7.2.*

3.2 Acceleration with SPIDERG

We now integrate the Stochastic Path-Integrated Differential EstimatoR Gradient (SPIDERG) into NFWU to accelerate the nonconvex projection-free optimization. SPIDERG is originally designed for unconstrained nonconvex problem in (Fang et al., 2018).

In a similar epoch-wise manner, SPIDERG utilizes $\mathbf{g}^{t-1}$ as a dynamic reference gradient rather than the fixed reference gradient $\tilde{\mathbf{g}}$ as SVRG. See the details in Algorithm 3. The variance of SPIDERG can be bounded by the following recursion, for $mod(t,p)\neq 0$:

$$\begin{aligned}&\mathbb{E}_{0:t}[\|\mathbf{g}^t-\nabla f(\mathbf{x}^t)\|^2]\\&=\mathbb{E}_{0:t-1}[\mathbb{E}_{\mathcal{S}^t}[\|\mathbf{g}^t-\nabla f(\mathbf{x}^t)\|^2|\mathbf{x}^t]]\\&=\mathbb{E}_{0:t-1}\left[\frac{1}{|\mathcal{S}^t|}\mathbb{E}_i[\|\nabla f_i(\mathbf{x}^t)-\nabla f_i(\mathbf{x}^{t-1})\|^2|\mathbf{x}^t]\right]\\&\quad+\mathbb{E}_{0:t-1}[\|\mathbf{g}^{t-1}-\nabla f(\mathbf{x}^{t-1})\|^2]\\&\le\frac{L^2}{|\mathcal{S}^t|}\cdot\frac{\epsilon^2}{L^2}+\mathbb{E}_{0:t-1}[\|\mathbf{g}^{t-1}-\nabla f(\mathbf{x}^{t-1})\|^2].\end{aligned}$$

Since we have $\mathbf{g}^{\bar{t}}=\nabla f(\mathbf{x}^{\bar{t}})$ for every $\bar{t}$ such that $mod(\bar{t},p)=0$, repeating the above recursion $mod(t,p)$ times gives the bound on $\mathbb{E}_{0:t}[\|\mathbf{g}^t-\nabla f(\mathbf{x}^t)\|^2]$ as

$$\mathbb{E}_{0:t}[\|\mathbf{g}^t-\nabla f(\mathbf{x}^t)\|^2]\le\frac{p\epsilon^2}{|\mathcal{S}^t|},$$

where the inequality follows since $mod(t,p)\le p$. By taking $|\mathcal{S}^t|=\sqrt{n}$ and $p=\sqrt{n}$, we have $\mathbb{E}_{0:t}[\|\mathbf{g}^t-\nabla f(\mathbf{x}^t)\|^2]\le\epsilon^2$. In the meantime, the SFO complexity in each epoch is still $2n$, and hence the amortized per-iteration cost is $2n/p=2\sqrt{n}$. We conclude that the overall SFO complexity using the SPIDERG approximation in NFWU is $\mathcal{O}(n^{1/2}/\epsilon^2)$.

4 New Curvature Aided Estimators

The availability to high order gradient, e.g. the Hessian, allows the algorithms to exploit extra curvature information, which is shown to accelerate the convergence for projection based method, e.g. (Nesterov and Polyak, 2006). In this section, we show that variance reduced gradient estimators like SVRG and SPIDERG also benefit from such high order information and hence the SFO complexity can be further reduced. The following analyses require the second-order smoothness assumption (8). Additionally, we define the Stochastic Second-order Oracle (SSO) complexity as the number of component Hessian evaluations.

Curvature Aided SVRG (CASVRG)

The intuition behind boosting the estimation accuracy of SVRG with curvature is the Mean Value Theorem

$$\nabla f_i(\mathbf{x}^t)-\nabla f_i(\tilde{\mathbf{x}})=\nabla^2 f_i(\bar{\mathbf{x}}^t),$$

with $\bar{\mathbf{x}}^t=\alpha^t\cdot\mathbf{x}^t+(1-\alpha^t)\cdot\tilde{\mathbf{x}}$ for some $\alpha^t\in[0,1]$, and we have under the Hessian smoothness assumption

$$\|\nabla f_i(\mathbf{x}^t)-\nabla f_i(\tilde{\mathbf{x}})-\nabla^2 f_i(\tilde{\mathbf{x}})(\mathbf{x}^t-\tilde{\mathbf{x}})\|=\mathcal{O}(\|\mathbf{x}^t-\tilde{\mathbf{x}}\|^2).$$

From the previous discussion, we see that the distance $\|\mathbf{x}^t-\tilde{\mathbf{x}}\|^2$ is well-controlled by the normalization in NFWU and the choice of step-size in Theorem 3.1. Therefore, the bound on the variance can be improved. Specifically, CASVRG adds a correction term

$$\mathbf{c}^t=-\frac{1}{|\mathcal{S}^t|}\sum_{i\in\mathcal{S}^t}\nabla^2 f_i(\tilde{\mathbf{x}})(\mathbf{x}^t-\tilde{\mathbf{x}})+\nabla^2 f(\tilde{\mathbf{x}})(\mathbf{x}^t-\tilde{\mathbf{x}})$$

Table 1: Per-iteration oracle complexities. The overall complexity can be obtained by multiplying $1/\epsilon^2$. We emphasize that, while CASVRG appears in (Gower et al., 2018), they only obtain local convergence in convex unconstrained setting. Instead, we prove the global convergence in nonconvex constrained case.

Gradient Estimator	SFO	Component Hessian	LO	From
SG	$\mathcal{O}(\epsilon^{-2})$	-	1	(Reddi et al., 2016)
SVRG	$\mathcal{O}(n^{2/3})$	-	1	(Reddi et al., 2016)
SPIDERG	$\mathcal{O}(n^{1/2})$	-	1	(Fang et al., 2018) (unconstrained)
CASVRG	$\mathcal{O}(\max\{n^{0.8}\epsilon^{0.4}, 1\})$	$\mathcal{O}(\max\{n^{0.8}\epsilon^{0.4}, 1\})$	1	(Gower et al., 2018) (convex)
CASPIDERG	$\mathcal{O}(\max\{n^{0.75}\epsilon^{0.5}, 1\})$	$\mathcal{O}(\max\{n^{0.75}\epsilon^{0.5}, 1\})$	1	this paper

to the original SVRG estimator, as detailed in Algorithm 4. Assuming Lipschitz continuity of the Hessian of f_i, we bound the variance of $\mathbf{g}^t$ conditioned on $\mathbf{x}^t$ by (the condition is omitted for notation simplicity)

$$\begin{aligned}
&\mathbb{E}_{\mathcal{S}^t}[\|\mathbf{g}^t - \nabla f(\mathbf{x}^t)\|^2] \\
\leq&\frac{1}{|\mathcal{S}^t|}\mathbb{E}_i[\|\nabla f_i(\mathbf{x}^t) - \nabla f_i(\tilde{\mathbf{x}}^s) - \nabla^2 f_i(\tilde{\mathbf{x}}^s)(\mathbf{x}^t - \tilde{\mathbf{x}}^s)\|^2] \\
=&\frac{1}{|\mathcal{S}^t|}\mathbb{E}_i[\|\nabla^2 f_i(\bar{\mathbf{x}}^t)(\mathbf{x}^t - \tilde{\mathbf{x}}^s) - \nabla^2 f_i(\tilde{\mathbf{x}}^s)(\mathbf{x}^t - \tilde{\mathbf{x}}^s)\|^2] \\
=&\frac{L_2^2}{|\mathcal{S}^t|}\|\bar{\mathbf{x}}^t - \tilde{\mathbf{x}}^s\|^2 \cdot \|\mathbf{x}^t - \tilde{\mathbf{x}}^s\|^2 \leq \frac{L_2^2}{|\mathcal{S}^t|} \cdot \frac{p^4\epsilon^4}{L^2},
\end{aligned}$$

where we use $\|\bar{\mathbf{x}}^t - \tilde{\mathbf{x}}^s\|^2 \leq p^2\epsilon^2$ in the last inequality. To achieve the best possible amortized SFO cost, we solve the following problem

$$\begin{aligned}
\min_{|\mathcal{S}^t|\in\mathbb{Z}_+,\ p\in\mathbb{Z}_+} \quad & |\mathcal{S}^t| + \frac{n}{p} \\
s.t. \quad & |\mathcal{S}^t| \geq L_2^2/L^2 \cdot p^4\epsilon^2.
\end{aligned}$$

Consequently, we set $p = L^{0.4}L_2^{-0.4}n^{0.2}\epsilon^{-0.4}$ and $|\mathcal{S}^t| = L_2^{0.4}L^{-0.4}n^{0.8}\epsilon^{0.4}$ for $\epsilon \geq n^{-2}L/L_2$. In this way, we have $p \times |\mathcal{S}^t| = n$ and $\mathbb{E}[\|\mathbf{g}^t - \nabla f(\mathbf{x}^t)\|^2] \leq \epsilon^2$ in the meantime, and the amortized SFO cost is $2n/p = 2L_2^{0.4}L^{-0.4}n^{0.8}\epsilon^{0.4}$.

In the other case when $\epsilon \leq L/(L_2n^2)$, we set $|\mathcal{S}^t| = 1$ since $L_2^{0.4}L^{-0.4}n^{0.8}\epsilon^{0.4} \leq 1$ and set $p = L^{0.5}L_2^{0.5}\epsilon^{-0.5}$ such that the requirement (9) is met. Therefore $p \times |\mathcal{S}^t| \geq n$ and hence the amortized SFO cost is no more than $2|\mathcal{S}^t| = 2$.

In conclusion, the overall SFO complexity of NFWU using the CASVRG approximation is $\mathcal{O}(\max\{n^{0.8}/\epsilon^{1.6}, 1/\epsilon^2\})$. Such complexity improves over the $\mathcal{O}(\sqrt{n}/\epsilon^2)$ result from SPIDERG when $\epsilon = \mathcal{O}(n^{-0.75})$. Surprisingly, it also indicates that when the target accuracy is high (ϵ is small) a single component gradient with Hessian correction is sufficient to approximate the full gradient ($|\mathcal{S}^t| = 1$).

However, such reduction to SFO comes at the cost of $\mathcal{O}(\max\{n^{0.8}/\epsilon^{1.6}, 1/\epsilon^2\})$ SSO which potentially does more harm than good to the optimization procedure. Interestingly, in the robust low rank matrix recovery, the computational cost to evaluate SSO and SFO are the same as discussed at the end of this section.

Remark 4.1. *We note that such curvature aiding strategy appears in (Gower et al., 2018). However, their work focuses on the convex unconstrained setting and only has local convergence. In contrast, our method converge globally for general nonconvex problems.*

Curvature Aided SPIDERG (CASPIDERG)

We now accelerate SPIDERG with the aforementioned curvature aiding technology. A slightly different correction term is used:

$$\mathbf{c}^t = [\nabla^2 f(\tilde{\mathbf{x}}) - \frac{1}{|\mathcal{S}^t|}\sum_{i\in\mathcal{S}^t}\nabla^2 f_i(\tilde{\mathbf{x}})](\mathbf{x}^t - \mathbf{x}^{t-1}). \quad (12)$$

The detailed estimator is given in Algorithm 5.

Assuming the Lipschitz continuity of the Hessian of f_i, we bound the variance of $\mathbf{g}^t$ conditioned on $\mathbf{x}^t$ by (the condition is omitted for notation simplicity)

$$\begin{aligned}
&\mathbb{E}_{\mathcal{S}^t}[\|\mathbf{g}^t - \nabla f(\mathbf{x}^t)\|^2|\mathbf{x}^t] \\
=&\frac{1}{|\mathcal{S}^t|}\mathbb{E}_i[\|\nabla f_i(\mathbf{x}^t) - \nabla f_i(\mathbf{x}^{t-1}) - \nabla^2 f_i(\tilde{\mathbf{x}}^s)(\mathbf{x}^t - \mathbf{x}^{t-1})\|^2 \\
&+ \|\mathbf{g}^{t-1} - \nabla f(\mathbf{x}^{t-1})\|^2 \\
=&\frac{1}{|\mathcal{S}^t|}\mathbb{E}_i[\|\nabla^2 f_i(\bar{\mathbf{x}}^t)(\mathbf{x}^t - \mathbf{x}^{t-1}) - \nabla^2 f_i(\tilde{\mathbf{x}}^s)(\mathbf{x}^t - \mathbf{x}^{t-1})\|^2] \\
&+ \|\mathbf{g}^{t-1} - \nabla f(\mathbf{x}^{t-1})\|^2 \\
=&\frac{L_2^2}{|\mathcal{S}^t|}\|\bar{\mathbf{x}}^t - \tilde{\mathbf{x}}^s\|^2 \cdot \|\mathbf{x}^t - \mathbf{x}^{t-1}\|^2 + \|\mathbf{g}^{t-1} - \nabla f(\mathbf{x}^{t-1})\|^2 \\
\leq&\frac{L_2^2}{|\mathcal{S}^t|} \cdot \frac{p^2\epsilon^4}{L^2} + \|\mathbf{g}^{t-1} - \nabla f(\mathbf{x}^{t-1})\|^2,
\end{aligned}$$

where $\bar{\mathbf{x}}^t = \alpha^t\mathbf{x}^t + (1-\alpha^t)\mathbf{x}^{t-1}$ for some $\alpha^t \in [0,1]$ is obtained from the Mean Value Theorem. Since we have $\mathbf{g}^{\bar{t}} = \nabla f(\mathbf{x}^{\bar{t}})$ for every $\bar{t}$ such that $mod(\bar{t}, p) = 0$, repeating the above recursion $mod(t, p)$ times gives the bound on $\mathbb{E}_{0:t}[\|\mathbf{g}^t - \nabla f(\mathbf{x}^t)\|^2]$ as

$$\mathbb{E}_{0:t}[\|\mathbf{g}^t - \nabla f(\mathbf{x}^t)\|^2] \leq \frac{L_2^2}{|\mathcal{S}^t|} \cdot \frac{p^3\epsilon^4}{L^2} \leq \epsilon^2. \quad (13)$$

Having such constraint on the choice of p and $|\mathcal{S}^t|$, we solve the following problem to obtain the best achiev-

GE 5 CASPIDERG(p, $|\mathcal{S}^t|$)

if $mod(t,p)=0$
  $\mathbf{g}^t=\nabla f(\mathbf{x}^t)$, $\tilde{\mathbf{x}}=\mathbf{x}^t$, $\tilde{\mathbf{U}}=\nabla^2 f(\tilde{\mathbf{x}})$;
else
  sample $|\mathcal{S}^t|$ i.i.d. component functions uniformly;
  $\mathbf{c}^t=[\tilde{\mathbf{U}}-\frac{1}{|\mathcal{S}^t|}\sum_{i\in\mathcal{S}^t}\nabla^2 f_i(\tilde{\mathbf{x}})](\mathbf{x}^t-\mathbf{x}^{t-1})$;
  $\mathbf{g}^t=\frac{1}{|\mathcal{S}^t|}\sum_{i\in\mathcal{S}^t}\left[\nabla f_i(\mathbf{x}^t)-\nabla f_i(\mathbf{x}^{t-1})\right]+\mathbf{g}^{t-1}+\mathbf{c}^t$;

able amortized per-iteration cost

$$\min_{|\mathcal{S}^t|\in\mathbb{Z}_+,\ p\in\mathbb{Z}_+} \quad |\mathcal{S}^t|+\frac{n}{p}$$
$$s.t. \quad |\mathcal{S}^t|\geq L_2^2/L^2\cdot p^3\epsilon^2.$$

For $\epsilon\geq n^{-1.5}L/L_2$, we set $|\mathcal{S}^t|=L^{-0.5}L_2^{0.5}\sqrt{\epsilon}\cdot n^{0.75}$ and $p=L^{0.5}L_2^{-0.5}n^{0.25}/\sqrt{\epsilon}$ to arrive at the $2|\mathcal{S}^t|=2L^{-0.5}L_2^{0.5}\sqrt{\epsilon}\cdot n^{0.75}$ average cost.
For $\epsilon\leq n^{-1.5}L/L_2$, we set $|\mathcal{S}^t|=1$ and $p=\epsilon^{-2/3}L_2^{-2/3}L^{2/3}$. Since $n/p\leq|\mathcal{S}^t|$, the average cost is no more than $2|\mathcal{S}^t|=2$.
In conclusion, the overall SFO complexity is $\mathcal{O}(\max\{n^{0.75}/\epsilon^{1.5},1/\epsilon^2\})$. Such complexity further improves the result of CASVRG and outperforms SPIDER when $\epsilon=\mathcal{O}(n^{-0.5})$. The additional SSO complexity is $\mathcal{O}(\max\{n^{0.75}/\epsilon^{1.5},1/\epsilon^2\})$ for CASPIDERG.

Remark 4.2. *While the curvature aided methods reduce the SFO complexity, the evaluation of SSO is usually expensive for general problem. However, these methods are particularly suited for the important Robust Low Rank Matrix Recovery problem (RLRMR), as both SFO and SSO can be evaluated in $\mathcal{O}(1)$. Hence the only additional cost using the curvature aided estimators is the matrix-vector product.*

5 Lower Bound of LO Complexity Among First-order Methods

For a convex objective $f(\cdot)$, Jaggi (2013) shows that $\mathcal{O}(1/\epsilon)$ LO queries are necessary to obtain an ϵ suboptimal point $\mathbf{x}\in\mathcal{C}$, i.e. $f(\mathbf{x})\leq\min_{\mathbf{x}\in\mathcal{C}}f(\mathbf{x})+\epsilon$. Such LO lower bound is absent in the nonconvex setting.

In this section, we establish the Linear Oracle (LO) lower bound for a broad class of first-order projection free methods to satisfy the optimal condition (2). We do this by exploiting the difficulty in $f(\cdot)$ itself, because we make the same convexity assumption on the feasible set. The single component ($n=1$) case of Problem (1) is considered and hence there is *no randomness* in component function query. Clearly, hard instance construction in this special case provides an LO lower bound for the general problem ($n\neq1$).

We specify the class of algorithms under consideration. Let $\mathbf{x}_k$ and $\mathbf{u}_k$ be the iterate and the output of the LO produced by the algorithm at iteration k. We put no restriction on the input to LO, but require

$$\mathbf{x}_k=\mathcal{A}_k(\mathbf{u}_0,\ldots,\mathbf{u}_{k-1}), \tag{14}$$

where $\mathcal{A}_k$ is a *deterministic* mapping from $\mathbb{R}^{k\times d}$ to $\mathbb{R}^d$. Such requirement is met by many projection-free methods like Frank-Wolfe, Block-Frank-Wolfe, Conditional Gradient Sliding, etc..

Theorem 5.1. *For any deterministic projection-free algorithm satisfying (14), there is an instance of Problem (1) with f being L-smooth and bounded from below $f(0)-\min_{\mathbf{x}\in\mathcal{C}}f(\mathbf{x})\leq\Delta$, such that it takes at least $\Omega(L\Delta/\epsilon^2)$ LO queries to obtain a point $\mathbf{x}$ such that $V_{\mathcal{C}}(\mathbf{x};f)\leq\epsilon D$, where D is the diameter of $\mathcal{C}$.*

The intuition behind our argument is that when the iterate $\mathbf{x}$ is close to the origin and hence distant from the boundary of $\mathcal{C}$, the suboptimality $V_{\mathcal{C}}(\mathbf{x};f)$ is $\Omega(\|\nabla f(\mathbf{x})\|)$ which is large due to the hardness from the non-convexity of $f(\cdot)$; when $\mathbf{x}$ is far from the origin and hence is possibly close to the boundary, our construction ensures that $\langle\nabla f(\mathbf{x}),\mathbf{x}\rangle$ is large, which is a nature lower bound of $V_{\mathcal{C}}(\mathbf{x};f)$ by taking $\mathbf{u}$ to be the origin. The detailed proof borrow ideas from (Carmon et al., 2017) and is deferred to the appendix 7.1.

6 Experiment

Under the framework of Normalized Frank Wolfe Update, we compare the efficiency of listed stochastic gradient estimators, namely SG, SVRG, SPIDERG, CASVRG, and CASPIDERG. We focus on the simulation studies of the Robust Low Rank Matrix Recovery problem discussed in Section 2.1. The parameter σ is fixed to 1 in all settings.

In each trial, we first generate an underlying matrix $\mathbf{M}$ of size 200×200 and rank $\gamma\in\mathbb{Z}_+$, where γ takes value ranging from 5 to 15. The singular values of $\mathbf{M}$ are set as $2^{[\gamma]}/2^{\gamma}\times50$ and hence $\|\mathbf{M}\|_*\leq C=100$, where $[\gamma]=\{1,\ldots,\gamma\}$. We then inject adversarial noise into $\mathbf{M}$ by (1) uniformly sampling 5% of the entries in $\mathbf{M}$ and (2) adding random noise uniformly sampled from $[-\rho,\rho]$ to each selected entry, where the noise level ρ ranges from 0 to 10. Denote $\hat{\mathbf{M}}$ as the matrix after noise injection. We uniformly sample 10% of the entries in $\hat{\mathbf{M}}$ to obtain the observations, i.e. $\mathbf{Y}_{ij}$. Hence $|\Omega|$, the number of observation is $M\times N\times10\%=4,000$.
In terms of algorithmic parameter setting, we vary the mini-batch size $|\mathcal{S}^t|$ from $|\Omega|/10$ to $|\Omega|/50$ and set $p=|\Omega|/|\mathcal{S}^t|$ correspondingly. The number of epoch T is set to 20 for all cases, and the convex combination parameter η^t is set to p/T in all cases for all methods. With underlying rank being 5 and noise scale being 10, we present the comparison of listed methods using

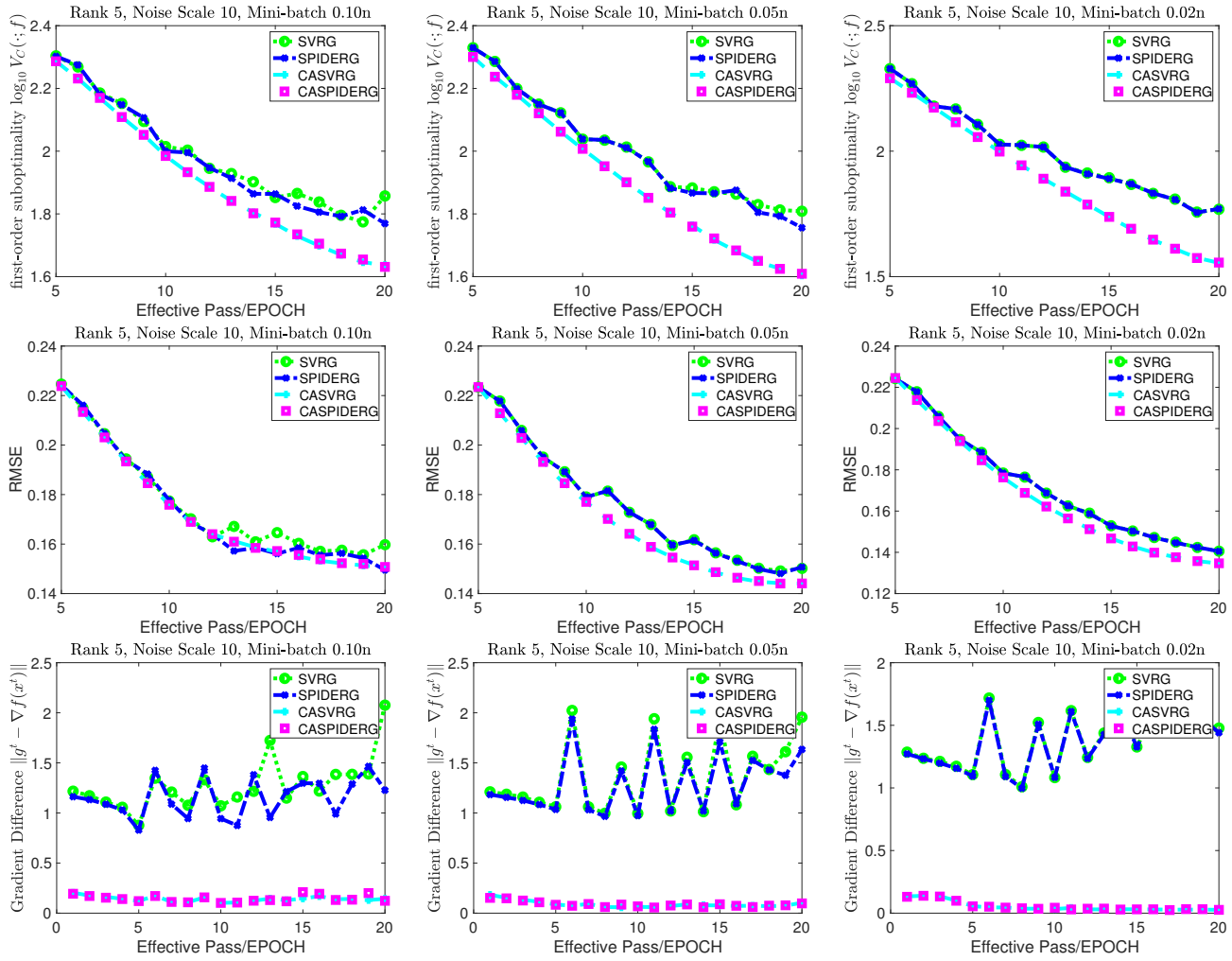

Figure 1: Each column reports the $V_{\mathcal{C}}(\cdot; f)$ value, the Root Mean Square Error (RMSE), and $\|\mathbf{g}^t - \nabla f(\mathbf{x}^t)$ value under a single parameter setting. We vary the mini-batch size in each column from $n/10$ to $n/50$ with the underlying matrix rank γ being 5 and adversarial noise level ρ being 10.

different minibatch sizes in Figure 1. More comparisons under different parameter settings are deferred to the appendix. We did not plot the result of SG as its performance is always significantly worse than other methods. In the first row, we compare the first-order suboptimality $V_{\mathcal{C}}(\cdot; f)$. We can see that the curvature aided methods always outperforms the first-order ones by a large amount. In the second row, we compare the testing error by measuring the Root Mean Square Error (RMSE) between the prediction matrix and the underlying matrix. The curvature aided methods show their advantages on these plots as well. In the third row, we compare accuracy of gradient estimation $\|\mathbf{g}^t - \nabla f(\mathbf{x}^t)\|$. CASVRG and CASPIDER has significantly less approximation error than SVRG and SPIDER in all settings. While the performance of SVRG and SPIDERG are close in the graph, SPIDERG is generally slightly better than SVRG in our experiments. All these results validate our analysis.

Conclusion

We design a projection-free meta-algorithm named NFWU that accepts gradient estimator (GE) as input and derive its unified convergence regardless of the choice of GE. By plugging SPIDERG into NFWU, we obtain the optimal SFO complexity for first-order methods. We then propose new curvature-aided estimators to further reduce the SFO cost. Finally, we provide a hard case to show that the $\mathcal{O}(1/\epsilon^2)$ LO complexity is optimal among first-order methods.

Acknowledgements

This work is supported by Zhejiang Provincial Natural Science Foundation of China under Grant No. LZ18F020002, and National Natural Science Foundation of China (No: 61472347, 61672376, 61751209).

References

Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4575–4583, 2016.

An Bian, Kfir Levy, Andreas Krause, and Joachim M Buhmann. Continuous dr-submodular maximization: Structure and algorithms. In *Advances in Neural Information Processing Systems*, pages 486–496, 2017.

Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points i. *arXiv preprint arXiv:1710.11606*, 2017.

Visesh Chari, Simon Lacoste-Julien, Ivan Laptev, and Josef Sivic. On pairwise costs for network flow multi-object tracking. In *CVPR*, volume 20, page 15, 2015.

Kenneth L Clarkson. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Transactions on Algorithms (TALG)*, 6(4):63, 2010.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path integrated differential estimator. *arXiv preprint arXiv:1807.01695*, 2018.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

Robert M Freund, Paul Grigas, and Rahul Mazumder. An extended frank–wolfe method with "in-face" directions, and its application to low-rank matrix completion. *SIAM Journal on Optimization*, 27(1):319–346, 2017.

Robert Gower, Nicolas Le Roux, and Francis Bach. Tracking the gradients using the hessian: A new look at variance reducing stochastic methods. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, 2018.

Elad Hazan and Haipeng Luo. Variance-reduced and projection-free stochastic optimization. In *International Conference on Machine Learning*, pages 1263–1271, 2016.

Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pages I–427–I–435. JMLR.org, 2013. URL `http://dl.acm.org/citation.cfm?id=3042817.3042867`.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

Simon Lacoste-Julien. Convergence rate of frank-wolfe for non-convex objectives. *arXiv preprint arXiv:1607.00345*, 2016.

Jean Lafond, Hoi-To Wai, and Eric Moulines. On the online frank-wolfe algorithms for convex and non-convex optimizations. *arXiv preprint arXiv:1510.01171*, 2015.

Guanghui Lan and Yi Zhou. Conditional gradient sliding for convex optimization. *SIAM Journal on Optimization*, 26(2):1379–1409, 2016. doi: 10.1137/140992382. URL `https://doi.org/10.1137/140992382`.

Arkadii Nemirovskii, David Borisovich Yudin, and Edgar Ronald Dawson. Problem complexity and method efficiency in optimization. 1983.

Yurii Nesterov and Boris T Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Stochastic recursive gradient algorithm for nonconvex optimization. *arXiv preprint arXiv:1705.07261*, 2017.

Chao Qu, Yan Li, and Huan Xu. Non-convex conditional gradient sliding. *arXiv preprint arXiv:1708.04783*, 2017.

Sashank J Reddi, Suvrit Sra, Barnabás Póczos, and Alex Smola. Stochastic frank-wolfe methods for non-convex optimization. In *Communication, Control, and Computing (Allerton), 2016 54th Annual Allerton Conference on*, pages 1244–1251. IEEE, 2016.

Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2003.

Kilian Q Weinberger and Lawrence K Saul. Unsupervised learning of image manifolds by semidefinite programming. *International journal of computer vision*, 70(1):77–90, 2006.

Miao Xu, Rong Jin, and Zhi-Hua Zhou. Speedup matrix completion with side information: Application to multi-label learning. In *Advances in neural information processing systems*, pages 2301–2309, 2013.

7 Appendix

7.1 Proof of Theorem 5.1

Proof of Theorem 5.1. Consider the problem

$$\min_{\mathbf{x}\in\mathcal{C}} f(\mathbf{x}) = \frac{1}{10}\|\mathbf{x}\|^2 + \Big\{\mathbf{G}(\mathbf{x}) \stackrel{\text{def}}{=} \mathbf{H}_T(\mathbf{V}^\top\mathbf{x})\Big\}$$

where we define $\mathbf{H}_T(\mathbf{y}) = \sum_{i=2}^{T}[\Psi(-y_{i-1})\Phi(-y_i) - \Psi(y_{i-1})\Phi(y_i)] - \Psi(1)\Phi(y_1)$ with

$$\begin{aligned}\Psi(y) &= \begin{cases} 0 & y \le 1/2 \\ \exp(1 - \frac{1}{(2y-1)^2}) & y > 1/2 \end{cases}, \\ \Phi(y) &= \sqrt{e}\int_{-\infty}^{y} e^{-\frac{1}{2}t^2} dt,\end{aligned} \tag{15}$$

$\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_T] \in \mathbb{R}^{d\times T}, \mathbf{V}^\top\mathbf{V} = \mathbf{I}_T$, and $\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| \le R\}$ with $R = 230\sqrt{T}$.

$\mathbf{H}_T(\mathbf{y})$ was originally constructed by Carmon et al. (2017) to show the lower bound of finding a first-order stationary point ($\|f(\mathbf{x})\| \le \epsilon$) for unconstrained smooth nonconvex minimization problem. In their paper, the following properties are established.

Lemma 7.1. *The function $\mathbf{H}_T(\mathbf{y})$ satisfies:*

1. *(large gradient) If $|y_i| < 1$ for any $i \le T$, then there exists $j \le i$ such that $|y_j| < 1$ and $|[\nabla\mathbf{H}_T(\mathbf{y})]_j| > 1$;*
2. *(bounded value) $\mathbf{H}_T(0) - \min_{\mathbf{y}\in\mathbb{R}^T}\mathbf{H}_T(\mathbf{y}) \le 12T$;*
3. *(bounded gradient) For all $\mathbf{y}$, $\|\nabla\mathbf{H}_T(\mathbf{y})\| \le 23\sqrt{T}$;*
4. *(smoothness) $\mathbf{H}_T$ is ℓ-Lipschitz smooth with $\ell = 3^{15}$ being a constant independent of T.*

We construct the columns of $\mathbf{V}$ based on the classical "resisting oracle" strategy (Nemirovskii et al., 1983): for every k, we let $\mathbf{v}_k$ be orthogonal to all $\{\mathbf{v}_r\}_{r=1}^{k-1}$ and $\{\mathbf{x}_r\}_{r=0}^{k}$. When the problem dimension d is sufficiently large, the adversary can make such construction by simulating the whole optimization procedure of the deterministic algorithm.

From such construction, we always have $\mathbf{x}_t^\top\mathbf{v}_T = 0$ for any $t \le T$. Using the "large gradient property" in Lemma 7.1, there exists $j \le T$ such that

$$|\mathbf{v}_j^\top\mathbf{x}_t| < 1 \text{ and } |\mathbf{v}_j^\top\nabla\mathbf{G}(\mathbf{x}_t)| > 1, \tag{16}$$

where the second inequality is from $|\mathbf{v}_j^\top\nabla\mathbf{G}(\mathbf{x}_t)| = |\mathbf{v}_j^\top\mathbf{V}\nabla\mathbf{H}_T(\mathbf{V}^\top\mathbf{x}_t)| = |[\nabla\mathbf{H}_T(\mathbf{V}^\top\mathbf{x}_t)]_j] > 1$.

We now show that $V_{\mathcal{C}}(\mathbf{x}_t; f)$ is large for all t.
For $\|\mathbf{x}_t\| \le \frac{2}{\sqrt{5}}R$, we have $V_{\mathcal{C}}(\mathbf{x}_t) \ge \|\nabla f(\mathbf{x}_t)\|(1 - \frac{2}{\sqrt{5}})R$, hence we only need to show $\|\nabla f(\mathbf{x}_t)\|$ is sufficiently large:

$$\begin{aligned}\|\nabla f(\mathbf{x}_t)\| &\ge |\mathbf{v}_j^\top\nabla f(\mathbf{x}_t)| = |\mathbf{v}_j^\top\nabla\mathbf{G}(\mathbf{x}_t) + \frac{1}{5}\mathbf{v}_j^\top\mathbf{x}_t| \\ &\ge |\mathbf{v}_j^\top\nabla\mathbf{G}(\mathbf{x}_t)| - |\frac{1}{5}\mathbf{v}_j^\top\mathbf{x}_t| > 4/5.\end{aligned}$$

Therefore we have $V_{\mathcal{C}}(\mathbf{x}_t; f) \ge (1 - \frac{2}{\sqrt{5}})R \cdot 4/5 \ge 0.08R$.
For $R \ge \|\mathbf{x}_t\| \ge \frac{2}{\sqrt{5}}R$, we have $V_{\mathcal{C}}(\mathbf{x}_t; f) \ge \mathbf{x}_t^\top\nabla f(\mathbf{x}_t)$,

$$\mathbf{x}_t^\top\nabla f(\mathbf{x}_t) = \frac{1}{5}\|\mathbf{x}_t\|^2 + \mathbf{x}_t^\top\nabla\mathbf{G}(\mathbf{x}_t) \ge \frac{1}{5}\|\mathbf{x}_t\|^2 - \|\mathbf{x}_t\|\|\nabla\mathbf{G}(\mathbf{x}_t)\| \ge \frac{3}{50}R^2 \ge 13.8R,$$

where we use $\|\nabla\mathbf{G}(\mathbf{x}_t)\| = \|\mathbf{V}\nabla\mathbf{H}_T(\mathbf{V}^\top\mathbf{x}_t)\| = \|\nabla\mathbf{H}_T(\mathbf{V}^\top\mathbf{x}_t)\| \le 23\sqrt{T} = R/10$.
In conclusion, we always have $V_{\mathcal{C}}(\mathbf{x}_t; f) = \Omega(R)$.

We rescaled f to obtain our hard instance. Define

$$\bar{f}(\mathbf{x}) = \frac{L\mu^2}{\ell} f(\mathbf{x}/\mu), \tag{17}$$

where $\mu \in \mathbb{R}$ is the scaling factor. We need to ensure the following properties.

1. $\bar{f}$ is L-Lipschitz smooth.
2. $\bar{f}$ is bounded from below: $\bar{f}(0) - \min_{\mathbf{x}\in\mathcal{C}} \bar{f}(\mathbf{x}) \leq \Delta$.
3. $V_{\mathcal{C}}(\mathbf{x}_t; \bar{f}) \geq \epsilon D$ for all $\mathbf{x}_t$, $t < T$.

The first requirement is met by the "smoothness" property in Lemma 7.1. For the second and third requirements, we set

$$\frac{L\mu^2}{\ell} c_1 T = \Delta, \frac{L\mu}{\ell} \cdot c_2 D = \epsilon D, \tag{18}$$

and hence we set $\mu = \frac{\ell\epsilon}{c_2 L}$ and $T = \frac{c_2^2}{\ell c_1} \cdot \frac{L\Delta}{\epsilon^2}$. In such case, $D = R \cdot \mu = c\sqrt{\Delta/L}$, with c being some constant. □

7.2 A Discussion about SAGAFW in Reddi et al. (2016)

We find the proof of their SAGAFW algorithm was wrong. When bounding R_T on the top of page 12, it is claimed that $\sum_{t=1}^{T} c_t \leq LD\gamma\sqrt{n}/(\rho\sqrt{b})$. However, this statement is incorrect: the right hand side is merely a bound on c_0, instead of an upper bound on the whole sum. Hence the $O(n^{1/3}/\epsilon^2)$ result is invalid. In fact it contradicts with a new lower bound on the unconstrained (and hence easier) non-convex infinite sum minimization problem which states that $O(\sqrt{n}/\epsilon^2)$ is necessary Fang et al. (2018).

7.3 More Experiments

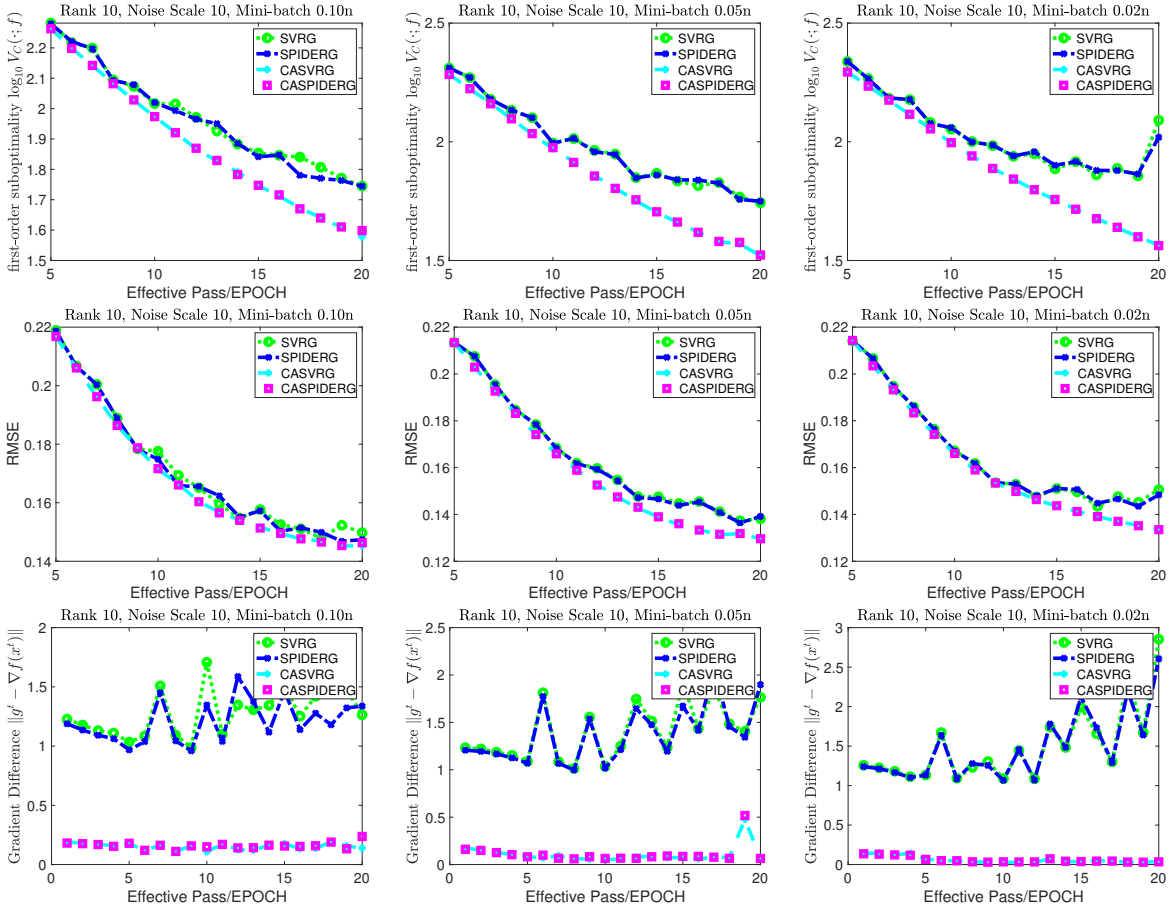

Figure 2: Each column reports the $V_{\mathcal{C}}(\cdot; f)$ value, the Root Mean Square Error (RMSE), and $\|\mathbf{g}^t - \nabla f(\mathbf{x}^t)$ value under a single parameter setting. We vary the mini-batch size in each column from $n/10$ to $n/50$ with the underlying matrix rank γ being 10 and adversarial noise level ρ being 10.

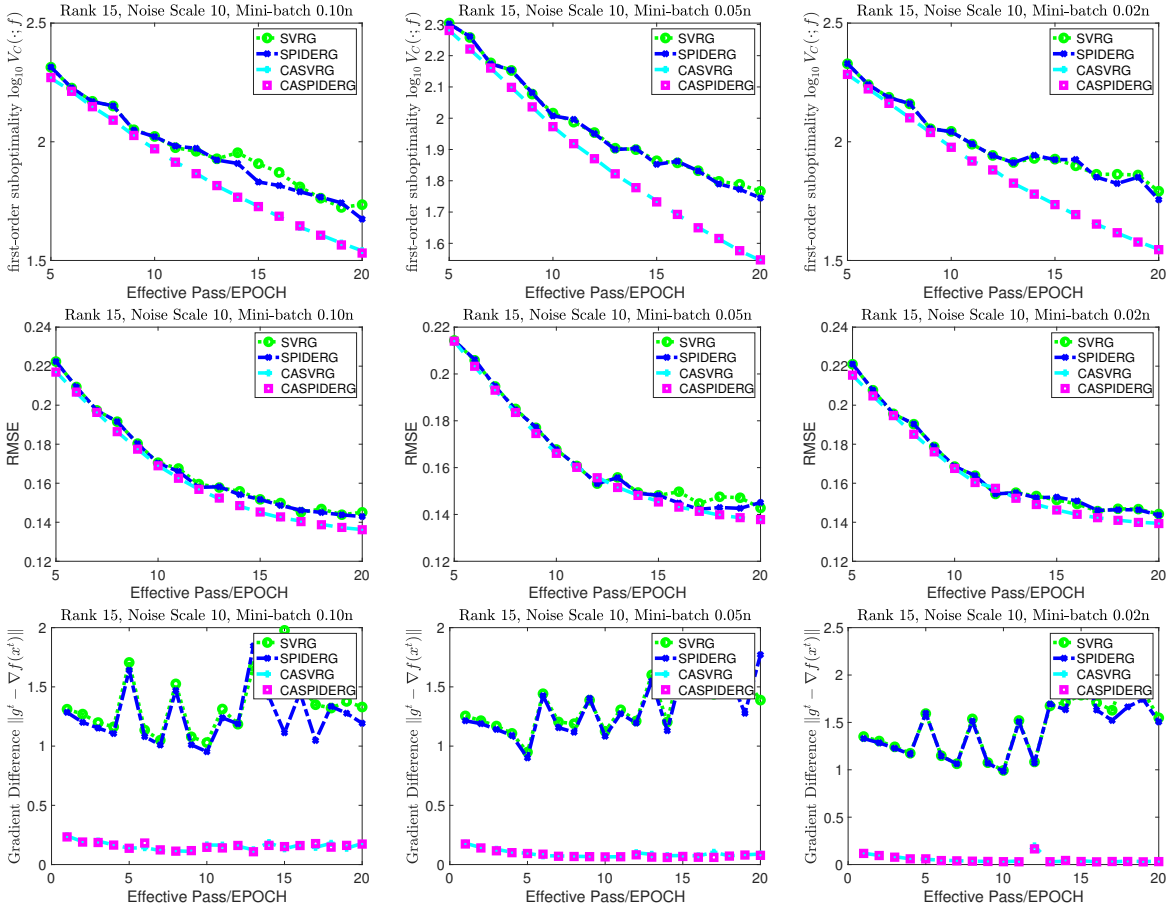

Figure 3: Each column reports the $V_C(\cdot; f)$ value, the Root Mean Square Error (RMSE), and $\|\mathbf{g}^t - \nabla f(\mathbf{x}^t)$ value under a single parameter setting. We vary the mini-batch size in each column from $n/10$ to $n/50$ with the underlying matrix rank γ being 15 and adversarial noise level ρ being 10.

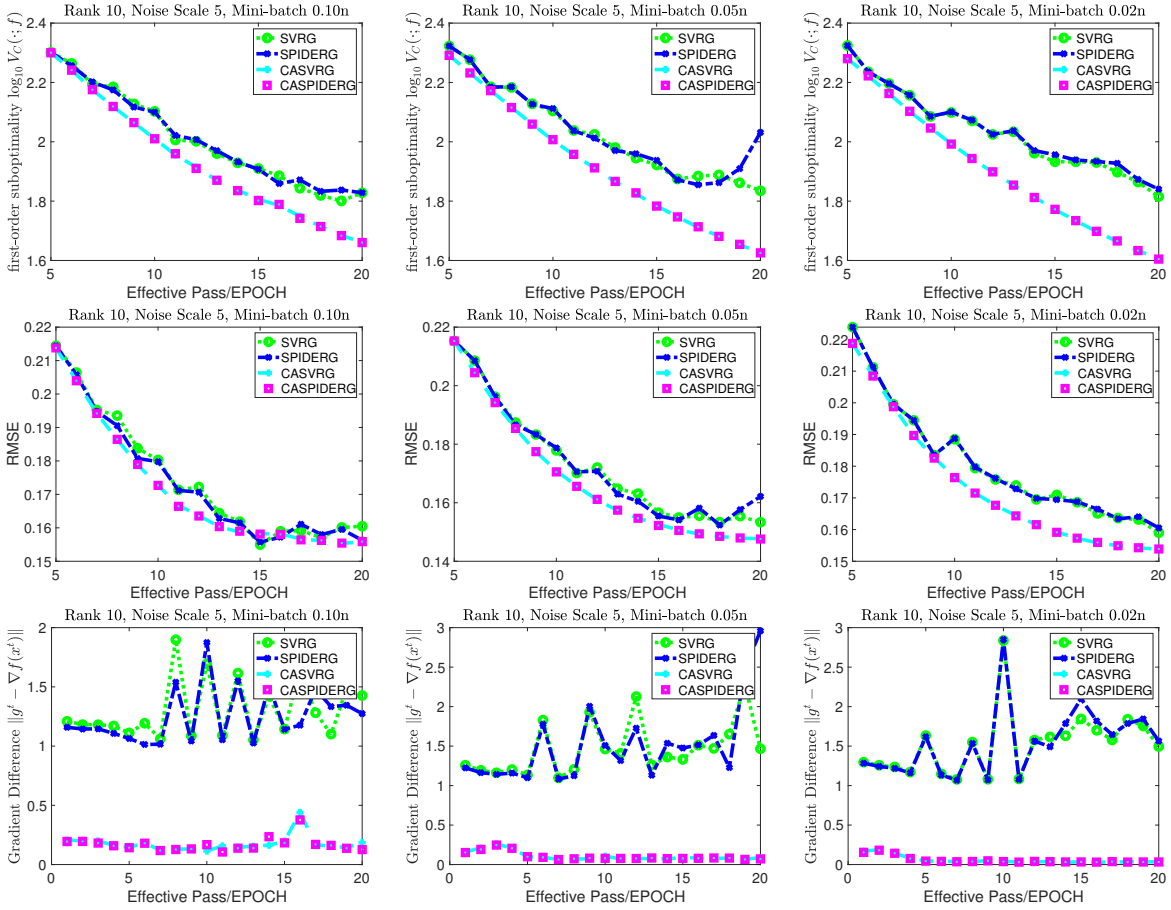

Figure 4: Each column reports the $V_{\mathcal{C}}(\cdot; f)$ value, the Root Mean Square Error (RMSE), and $\|\mathbf{g}^t - \nabla f(\mathbf{x}^t)$ value under a single parameter setting. We vary the mini-batch size in each column from $n/10$ to $n/50$ with the underlying matrix rank γ being 10 and adversarial noise level ρ being 5.