Projection-Free Algorithm for Stochastic Bi-level Optimization

Zeeshan Akhtar*, Amrit Singh Bedi†, Srujan Teja Thomdapu* and Ketan Rajawat*

Abstract

This work presents the first projection-free algorithm to solve stochastic bi-level optimization problems, where the objective function depends on the solution of another stochastic optimization problem. The proposed Stochastic Bi-level Frank-Wolfe (SBFW) algorithm can be applied to streaming settings and does not make use of large batches or checkpoints. The sample complexity of SBFW is shown to be $O(\epsilon^{-3})$ for convex objectives and $O(\epsilon^{-4})$ for non-convex objectives. Improved rates are derived for the stochastic compositional problem, which is a special case of the bi-level problem, and entails minimizing the composition of two expected-value functions. The proposed Stochastic Compositional Frank-Wolfe (SCFW) is shown to achieve a sample complexity of $O(\epsilon^{-2})$ for convex objectives and $O(\epsilon^{-3})$ for non-convex objectives, at par with the state-of-the-art sample complexities for projection-free algorithms solving single-level problems. We demonstrate the advantage of the proposed methods by solving the problem of matrix completion with denoising and the problem of policy value evaluation in reinforcement learning.

I. INTRODUCTION

We consider a class of two-level hierarchical optimization problems, given by

$$(P_1) \quad \min_{x \in X \subset \mathbb{R}^m} Q(x) := F(x, y^*(x)) \tag{outer}$$

$$y^*(x) \in \arg \min_y G(y, x) \tag{inner},$$

where $x$ is the optimization variable. Here, the outer problem involves minimizing the objective function $Q(x)$ with respect to $x$ over the convex compact constraint set $X \subset \mathbb{R}^m$. The objective function is of the form $Q(x) := F(x, y^*(x))$, where $y^*(x)$ is a solution of the inner stochastic optimization problem, which for a given $x$, entails minimizing the strongly convex function $G(y, x)$ with respect to optimization variable $y$. Observe that for bilevel problems of type $P_1$, the inner and outer problems are inter-dependent and cannot be solved in isolation. Yet, these problems arise in a number of areas, such as meta-learning [1], continual learning [2], reinforcement learning [3], and hyper-parameter optimization [4], [5]. Of particular interest are the large-scale or stochastic settings, where first-order stochastic approximation algorithms have been recently proposed [6]–[9]. The main idea behind these algorithms is to run one gradient descent step in order to solve the inner optimization problem, and subsequently, utilize the updated variable $y$ to run another gradient descent step on the outer minimization problem.

*Z. Akhtar, S. T. Thomdapu and K. Rajawat are with the Department of Electrical Engineering, Indian Institute of Technology Kanpur, Kanpur 208016, India (e-mail: zeeshan@iitk.ac.in, srujant@iitk.ac.in, ketan@iitk.ac.in). †A. S. Bedi is with the U.S. Army Research Laboratory, Adelphi, MD, USA. (e-mail: amrit0714@gmail.com).
In some works, such as [19], [21], the constraint set $X$ in the outer optimization problem is taken to be $X = \mathbb{R}^m$, resulting in a simpler unconstrained outer optimization problem. However, in applications such as meta learning [1], personalized federated learning [22], and corsets [2], the constraint set $X$ is a strict subset of $X \subset \mathbb{R}^m$. The standard approach to dealing with such constraint sets is to project the updates of the outer optimization problem onto $X$ at every iteration. Though popular and widely used, the projected gradient approaches may not necessarily be practical, for instance, in cases where the projection sub-problem is too expensive to be solved at every iteration. The difficulties surrounding projection-based methods have motivated the development of projection-free algorithms [23], that make use of the Frank-Wolfe (FW) updates [24]. These FW-based algorithms only require solving a linear program over $X$, which could be significantly cheaper than solving a non-linear projection problem, as in the case of $\ell_1$-norm or nuclear norm ball constraints.

Projection-free algorithms for single-level stochastic optimization algorithms are well-known and state-of-the-art algorithms achieve a sample complexity of $O(\epsilon^{-2})$ [11], [12]. These algorithms rely on a recursive gradient tracking approach that allows the samples to be processed sequentially and achieves variance reduction without the use of checkpoints or large batches. Motivated from these developments, we ask the following question:

"Is it possible to develop efficient projection-free algorithms for bi-level stochastic optimization problems?"

This work puts forth the Stochastic Bi-level Frank-Wolfe (SBFW) algorithm, which is the first projection-free algorithm for bi-level problems. Further, we also focus on the special class of problems called stochastic compositional problems where inner optimization problem in [1] exhibits a closed form expression for the optimal [19], [21]. Although the results developed for bi-level problems would apply to compositional problems, we have derived improved convergence rates for the stochastic compositional problems, hence studied them separately (cf. Sec. III-B). A comprehensive list of all existing related works is provided in Table I. For bi-level problems, the second column of Table I represents the objective type (convex/non-convex) of outer function only. The inner objective is assumed to be strongly convex for all the methods in Table I. Next, we summarize the contributions.

| Reference | Objective | Projection Free | Problem Type | SFO Complexity (Outer) | SFO Complexity (Inner) |
|-----------|-----------|----------------|--------------|------------------------|------------------------|
| SFW [10]  | Convex, Non-Convex | ✓ | Single-Level | $O(\epsilon^{-3}), O(\epsilon^{-4})$ | - |
| ORFW [11] | Convex | ✓ | Single-Level | $O(\epsilon^{-2})$ | - |
| SBFW++ [12] | Non-Convex | ✓ | Single-Level | $O(\epsilon^{-3})$ | - |
| SCGD [13] | Convex, Non-Convex | × | Compositional | $O(\epsilon^{-3}), O(\epsilon^{-4})$ | - |
| ASC [14] | Convex, Non-Convex | × | Compositional | $O(\epsilon^{-2}), O(\epsilon^{-2.25})$ | - |
| NASA [15], SCSC [16] | Convex, Non-Convex | × | Compositional | $O(\epsilon^{-2})$ | - |
| SCFW (This work) | Convex, Non-Convex | ✓ | Compositional | $O(\epsilon^{-3}), O(\epsilon^{-3})$ | - |
| SBA [7] | non-convex | × | Bi-Level | $O(\epsilon^{-2})$ | $O(\epsilon^{-3})$ |
| succBiO [17] | non-convex | × | Bi-Level | $O(\epsilon^{-2})$ | $O(\epsilon^{-3})$ |
| STABLE [18] | non-convex | × | Bi-Level | $O(\epsilon^{-2.5})$ | $O(\epsilon^{-2.5})$ |
| MSTSA [20] | non-convex | × | Bi-Level | $O(\epsilon^{-2.5})$ | $O(\epsilon^{-2.5})$ |
| TTSA [19] | non-convex | × | Bi-Level | $O(\epsilon^{-3})$ | $O(\epsilon^{-3})$ |
| SUSTAIN [19] | non-convex | × | Bi-Level | $O(\epsilon^{-3}), O(\epsilon^{-4})$, $O(\epsilon^{-1.5})$, $O(\epsilon^{-2})$ |
| SBFW (This work) | Convex, Non-convex | ✓ | Bi-Level | $O(\epsilon^{-2}), O(\epsilon^{-4})$, $O(\epsilon^{-1.5})$, $O(\epsilon^{-2})$ |

TABLE I: This table compares the different algorithms available in the literature to solve single-level, bi-level, and compositional problems. We present the stochastic first order complexity (SFO) for the outer as well as inner optimization problems, which denotes the number of minimum queries to the first order oracle made by the corresponding algorithm to achieve an $\epsilon$ optimal stationary point. Note that the proposed algorithms are able to achieve the best possible SFO complexities as compared to the existing projection-free algorithms.
(i) Firstly, we propose a novel projection-free SBFW algorithm, utilizing the idea of momentum-based gradient tracking \cite{25} in order to track the gradient of the outer objective function. The combination of this idea along with FW updates allows us to achieve the sample complexity of $O(\epsilon^{-3})$ and $O(\epsilon^{-4})$ for convex and non-convex cases, respectively (cf, Sec. IV).

(ii) Secondly, in contrast to the existing literature, we consider the compositional problems (which is a special case of bi-level problems) separately in this paper and propose a novel Stochastic Compositional Frank Wolfe (SCFW) algorithm. The SCFW algorithm is able to achieve better convergence rate than bi-level problems under fewer assumptions on the inner objective functions. SCFW achieves the sample complexity of $O(\epsilon^{-2})$ and $O(\epsilon^{-3})$ for convex and non-convex cases, respectively (cf, Sec. IV). These results are interesting because they match with the existing best possible convergence rates of projection-based methods (which are computationally expensive) for compositional problems \cite{14}, \cite{16} and projection-free methods for non-compositional (single-level) cases \cite{11}, \cite{26}.

(iii) Finally, we test the proposed algorithm on matrix completion and the problem of policy value evaluations in reinforcement learning and establish the efficacy of the proposed techniques as compared to state-of-the-art algorithms (cf. Sec. V).

A. Motivating Application

Matrix Completion with Denoising. The goal of matrix completion is to recover the missing entries of a partially observed matrix \cite{27}. In general, data matrix can be modeled as a low-rank matrix motivating the matrix completion problem with nuclear norm constraint. Such low-rank matrix completion problem arises in a wide range of applications such as image processing \cite{28}, multi-task learning \cite{29}, and collaborative filtering \cite{30}, etc. Under noisy observations, solving the matrix completion problem with nuclear norm constraints often results in suboptimal results. This issue can be tackled by introducing a preprocessing step called denoising before solving the matrix completion problem. However, this requires access to all the elements of the observed set $\Omega$. Also note that such denoising step is impractical in the online stochastic settings where we only observe a random subset of $\Omega$ at each iteration $t$. As a solution, we can formulate the matrix completion problem as a stochastic bi-level problem by incorporating the denoising step as the inner-problem. Mathematically, the bi-level matrix completion with denoising problem can be defined as

$$\min_{\|X\|_* \leq \alpha} \frac{1}{|\Omega_1|} \sum_{(i,j) \in \Omega_1} (X_{i,j} - Y_{i,j})^2,$$

s. t. $Y \in \text{arg min}_Y \left\{ \frac{1}{|\Omega_2|} \sum_{(i,j) \in \Omega_2} (Y_{i,j} - M_{i,j})^2 + \lambda_1 \|Y\|_1 + \lambda_2 \|X - Y\|_F^2 \right\},$

where $M \in \mathbb{R}^{n \times m}$ is the given incomplete noisy matrix, $\|Y\|_1 = \sum_{i,j} |Y_{i,j}|$ is the $\ell_1$ norm, and $\lambda_1$ and $\lambda_2$ are regularization parameters. Note that the regularization over the discrepancy between $X$ and denoised matrix $Y$ results in bilevel formulation \cite{3}. A similar technique in deterministic settings is utilized in various other applications in machine learning and signal processing problems \cite{31}, \cite{32}. Observe that \cite{3} is in the form of bilevel formulation \cite{1}; however, when the entries are revealed in the form of randomly selected subsets $\Omega_1 \subset \Omega_1$ and $\Omega_2 \subset \Omega_2$
at every iteration, it becomes stochastic in nature (cf. Sec. II). The main challenge here is due to the nuclear norm constraint, which makes it quite computationally expensive (sometimes even impractical) to solve (3) with projection-based bilevel algorithms. This challenge is accentuated in the case of large size matrices. In Sec. V, we will show experimentally that the proposed algorithm SBFW is best suited to address such challenges in bi-level stochastic optimization problems.

B. Related Works

Bi-level optimization has had a long history, with the earliest applications in economic game theory [33]. Bi-level optimization has recently received great attention from the machine learning (ML) community due to the number of applications in the area [34]. A series of works that proposed to solve the problem of the form (1) has appeared recently [7]–[9], [18], [19], [35], [36]. Of these, the seminal works in [7], [8] proposed a class of double-loop approximation algorithms to iteratively approximate the stochastic gradient of the outer objective and incurred a sample complexity of $O(\epsilon^{-2})$ in order to achieve the $\epsilon$-stationary point. The double loop structure of these approaches made them impractical for large-scale problems; [7] required solving an inner optimization problem to a predefined accuracy, while [8] required a large batch size of $O(\epsilon^{-1})$ at each iteration. To address this issue, various single-loop methods, involving simultaneous update of inner and outer optimization variables, have been developed [8], [9], [18], [19]. A single-loop two-time scale stochastic algorithm proposed in [18] incurred a sub-optimal sample complexity of $O(\epsilon^{-2.5})$. This is further improved recently in [8], [9], [19], in which the authors have utilized the momentum-based variance reduction technique from [25] to obtain optimal convergence rates. While all of the above mentioned works seek to solve (1), they are projection-based and require a projection on to $\mathcal{X}$ at every iteration. In this work, we are interested in developing projection-free stochastic optimization algorithms for bi-level problems which is still an open problem, and the subject of the work in this paper.

Compositional optimization problems have been recently studied and various algorithms have been proposed in [13]–[16], [37]–[39]. The seminal work in [13] proposed a quasi-gradient method called stochastic compositional gradient descent (SCGD) to solve the problem via a two time-scale approach. In [14], the authors proposed an accelerated SCGD method that achieved an improved sample complexity of $O(\epsilon^{-2})$ for convex objectives and $O(\epsilon^{-2.25})$ for non-convex objectives. Further, different variance-reduced SCGD variants have been proposed, such as SCVR [40], VRSC-PG [41], SARAH-Compositional [37], [42], and STORM-Compositional [38]. In the literature, we can also find some single time-scale algorithms to solve compositional problems [15], [16]. Work in [15] presented a nested averaged stochastic approximation (NASA) and proved a sample complexity of $O(\epsilon^{-2})$. Recently, another single time-scale algorithm called the stochastically corrected stochastic compositional gradient method (SCSC) is proposed in [16] that converges at the same rate as the SGD methods for non-compositional stochastic optimization. It further adopted the Adam-type adaptive gradient approach and achieved the optimal sample complexity of $O(\epsilon^{-2})$. Again, all the above-mentioned algorithms either solve an unconstrained problem or use projection operation at each iteration to deal with the constraints. In this work, we developed a projection-free algorithm for compositional problems as well. Note that even-though compositional problems are a special case
of bi-level problems, we have studied them separately in this work and proposed a novel projection-free algorithm specifically for compositional problems to achieve the optimal sample complexity.

**Projection-free algorithms** have been extensively studied to solve the single-level optimization problems of the form (1) in the literature [43]–[45]. A number of first-order projection-free algorithms have been developed for stochastic optimization problems as well [10], [11], [46], [47]. The stochastic FW method proposed in [46] achieves a sample complexity of $O(\epsilon^{-3})$ but requires an the batch size $b = O(t)$, where $t$ is the iteration index. The need for increasing batch sizes was dropped in [10], which worked with a standard mini-batch, but still achieved the same sample complexity. Finally, an improved stochastic recursive gradient estimator-based algorithm called ORGFW was proposed in [11] and achieved a sample complexity of $O(\epsilon^{-2})$. For non-convex problems, [47] proposed an approach where the batch-size depends on the total number of iterations, resulting in a sample complexity of $O(\epsilon^{-4})$. Later, work in [12] came up with a two-sample strategy and achieved $O(\epsilon^{-3})$ sample complexity. In summary, FW-based projection-free algorithms have been well-studied for single-level problems but not for bi-level problems.

**II. Problem Formulation**

Let us consider the optimization problem $\mathcal{P}_1$. For most of the applications in practice [1]–[3], [5], [18], the outer objective $F(x, y^*(x)) := \mathbb{E}_\theta[f(y^*(x); \theta)]$ and the inner objective $G(y, x) := \mathbb{E}_\xi[g(y, x; \xi)]$ are expected values of continuous and proper closed functions $f : \mathbb{R}^m \to \mathbb{R}$, and $g : \mathbb{R}^n \to \mathbb{R}^m$ with respect to the independent random variables $\theta$ and $\xi$, respectively. Hence, the equivalent bi-level stochastic optimization problem is given by

$$x^* := \arg\min_{x \in X \subset \mathbb{R}^m} Q(x) = \mathbb{E}_\theta[f(y^*(x); \theta)],$$  \hspace{1cm} (4a)

subject to

$$y^*(x) \in \arg\min_{y \in \mathbb{R}^n} \mathbb{E}_\xi[g(y, x; \xi)].$$  \hspace{1cm} (4b)

Besides the general form in (4), we are also interested in the special cases where it is possible to solve the inner optimization problem at a given $x$ and obtain a smooth closed form expression for the optimal solution $y^*(x)$. For instance, if the inner objective $g$ is quadratic in $y$, i.e. $g(x, y, \xi) = \|y - h(x, \xi)\|^2$, where $h$ is a smooth function over $x$, then we can write $y^*(x) = h(x, \xi)$. Hence, the problem in (4) boils down to a stochastic compositional optimization problem given by

$$\min_{x \in X \subset \mathbb{R}^m} C(x) := \mathbb{E}_\theta [f(\mathbb{E}_\xi[h(x, \xi)], \theta)],$$ \hspace{1cm} (5)

which involves the nested expectations in the objective. The problem in (5) has been independently considered in the literature and solved using two-time scale approaches [13], [48]. A single-time scale approach for the problem in (5) is also proposed in [15], [16] but all the existing approaches are projection-based. In this work, we are interested in developing first order methods to solve the problem in (4) in a projection-free manner. We remark here that while the algorithms developed for problem in (4) could be readily applied to solve the problems of the form in (5), but there is a scope to further propose faster algorithms for compositional problems in (5). Therefore, we will consider the compositional problems separately from bi-level problems and derive better convergence rates.
A. Notations

First, we defined the compact notations we use throughout the paper. We denote column vectors with lowercase boldface \( x \), its transpose as \( x^\top \), and its Euclidean norm by \( ||x|| \). We use \( E_t := E[\cdot | \mathcal{F}_t] \) to denote the conditional expectation with respect to given sigma field \( \mathcal{F}_t \) which contains all algorithm history (randomness) till step \( t \).

III. Algorithm Development

In this section, we develop the proposed algorithm to solve the problem in (4). We note that solving the bi-level optimization problem in (4) is NP hard in general but we restrict our focus to problems where the inner objective is continuously twice differentiable in \((x, y)\) and also strongly convex w.r.t \( y \) with parameter \( \mu_y > 0 \). Such an assumption is common in the related works \([7] - [9]\) and ensures that for any \( x \in \mathcal{X} \), \( y^*(x) \) is unique. Applying any first order method to solve (4) requires the evaluation of the gradient of outer objective \( \nabla Q(x) \) with respect to \( x \). Let us denote the iteration index by \( t \in \{1, 2, \cdots, T\} \), and then we can write the standard projected gradient descent update as

\[
x_{t+1} = \mathcal{P}_\mathcal{X} \left[ x_t - \alpha \nabla Q(x_t) \right],
\]

where \( \mathcal{P}_\mathcal{X} [\cdot] \) denotes the projection onto the constraint set \( \mathcal{X} \) and \( \alpha \) is the constant step size. Note that the gradient calculation in (6) is achieved by application of implicit function theorem to the optimality condition for inner optimization problem \( \nabla_y G(y, x) = 0 \), calculating total derivative followed by chain rule to obtain the expression

\[
\nabla Q(x_t) = \nabla_x F(x_t, y^*(x_t)) - \nabla^2_{yy} G(y^*(x_t), x_t) [\nabla^2_{xy} G(y^*(x_t), x_t)]^{-1} \nabla_x F(x_t, y^*(x_t)).
\] (7)

From (7), note that \( \nabla Q(x_t) \) requires the information about \( y^*(x_t) \) which is not available in general, unless the second level problem (2) has a closed form solution. Hence, it is not possible to utilize gradient based algorithms to solve the problem in (4). This challenge is addressed in the literature via the utilization of approximate gradients (7). Following a similar approach, we use a surrogate gradient defined as \( \nabla S(x_t, y_t) \) in place of original gradient \( \nabla Q(x_t) \). The surrogate gradient is obtained by replacing \( y^*(x_t) \) in (7) with some \( y_t \in \mathbb{R}^n \) (we will define the explicit value of \( y_t \) later) to write

\[
\nabla S(x_t, y_t) = \nabla_x F(x_t, y_t) - \nabla^2_{yy} G(y_t, x_t) [\nabla^2_{xy} G(y_t, x_t)]^{-1} \nabla_x F(x_t, y_t).
\] (8)

Even after replacing \( y^*(x_t) \) with some \( y_t \in \mathbb{R}^n \), there is an additional challenge of evaluating the individual terms in (8) for the stochastic bi-level problems mentioned in (4). For instance, the term \( \nabla_x F(x_t, y_t) = E_\theta [\nabla_x f(y_t; \theta)] \) in (8) involves the evaluation of the expectation operator, which is not possible in practice due to the unknown data distribution. One standard approach is to use unbiased stochastic gradient instead of the original gradient, but an unbiased estimate of \( \nabla S(x_t, y) \) would still require the computation of Hessian inverse. To avoid such complicated matrix computations, we follow the approach presented in [18, Sec. E.4] and compute a mini-batch approximation of Hessian inverse using the samples returned by the sampling oracle. We assume the availability of sampling oracle such that for a given \( x \in \mathcal{X} \) and \( y \in \mathbb{R}^n \), it returns unbiased samples \( \nabla_x f(x, y, \xi), \nabla_y f(x, y, \theta), \nabla_y g(x, y, \xi) \), \( \nabla^2_{yx} g(x, y, \xi) \), and \( \nabla^2_{yy} g(x, y, \xi) \) realized at random variables \( \xi \) and \( \theta \). Having access to such oracle, we can write
the biased estimate (denoted by $h(x_t, y_t; \theta_t, \xi_t)$) of surrogate gradient $\nabla S(x_t, y_t)$ in (8) for a given $x_t \in X$ and $y_t$ as

$$h(x_t, y_t; \theta_t, \xi_t) = \nabla_x f(x_t, y_t; \theta_t) - M(x_t, y_t; \tilde{\xi}_t) \cdot \nabla_y f(x_t, y_t; \theta_t),$$

(9)

where $\nabla_x f(x_t, y_t; \theta_t)$ is an unbiased estimate of $\nabla_x F(x_t, y_t)$, $\nabla_y f(x_t, y_t; \theta_t)$ is an unbiased estimate of $\nabla_y F(x_t, y_t)$, and $M(x_t, y_t; \tilde{\xi}_t)$ is a biased estimate of product $\nabla^2_{x,y} G(y_t, x_t) : [\nabla^2_{x,y} G(y_t, x_t)]^{-1}$. In the term $M(x_t, y_t; \tilde{\xi}_t)$, $\tilde{\xi}_t$ is defined as $\tilde{\xi}_t := \{\xi_{t,i} : i \in \{0, 1, \ldots, k\}\}$ which represents a collection of $(k+1)$ i.i.d. samples $\nabla^2_{x,y} g(y_t, x_t; \xi_{t,i})$.

The explicit form of $M(x_t, y_t; \tilde{\xi}_t)$ is given by

$$M(x_t, y_t; \tilde{\xi}_t) = \nabla^2_{x,y} g(y_t, x_t; \xi_{t,0}) \left[ \frac{k}{L_g} \prod_{i=1}^{k} \left(I - \frac{1}{L_g} \nabla^2_{x,y} g(y_t, x_t; \xi_{t,i})\right) \right],$$

(10)

where $l$ is selected uniformly from $\{0, 1, \ldots, k-1\}$.

Further, for $l=0$, we use the convention $\prod_{i=1}^{k} \left(I - \frac{1}{L_g} \nabla^2_{x,y} g(y_t, x_t; \xi_{t,i})\right) = I$. Hence, a stochastic version of the update in (6) would be given by

$$x_{t+1} = \mathcal{P}_X [x_t - \alpha h(x_t, y_t; \theta_t, \xi_t)].$$

(11)

Similar to the update in (11), different variants are proposed in the literature [7], [9], [19]. But a significant challenge which remains un-addressed till date in the literature for the bi-level problems is associated with the projection operator in (11). Projected subgradient methods require to perform a computationally expensive projection step at each iteration $t$. One projection step is called projection oracle call. For instance, to achieve $\epsilon$ suboptimality, the projected subgradient method [49] entails $O(\epsilon^{-2})$ projection oracle calls. The projection is easy to evaluate when the constraint set is a simple convex set or has closed form solution. However, the projection step is often computationally costly (e.g., nuclear norm constraint), and its complexity could be comparable to the problem at hand [23].

In the next subsection, we obviate the issue related to projections by proposing projection-free algorithms for both the bi-level and compositional stochastic optimization problems, which is the key novel aspect of work in this paper.

A. Stochastic Bi-level Frank Wolfe (SBFW) Algorithm:

Before proceeding towards the design of projection-free algorithm, a discussion regarding the particular choice of $y_t$ in (11) is due. A popular choice (see [9], [15], [17], [18]) for $y_t$ is the stochastic gradient descent update for the inner optimization problem given by

$$y_t = y_{t-1} - \delta_t \nabla_y g(y_{t-1}, x_{t-1}; \xi_t),$$

(12)

where $\nabla_y g(y_{t-1}, x_{t-1}; \xi_t)$ is the unbiased estimate of the gradient $\mathbb{E}_{\xi_t} [\nabla_y g(y_{t-1}, x_{t-1}; \xi_t)]$, and $\delta_t$ denotes the step size. Now we are ready to propose the first projection-free algorithm for the bi-level stochastic optimization problems. We propose to use a conditional gradient method (CGM) based approach instead of calculating the projection in (11). That is, we solve a linear minimization problem to find a feasible direction $s_t \in X$ for a given stochastic gradient direction $h(x_t, y_t; \theta_t, \xi_t)$, given by, $s_t := \arg \min_{s \in X} \langle s, h(x_t, y_t; \theta_t, \xi_t) \rangle$. This reduces the
Another line of work suggests the use of gradient tracking given by $d_{t} = (1 - \rho_{t})(d_{t-1} - h(x_{t-1}, y_{t-1}; \theta_{t}, \xi_{t})) + h(x_{t}, y_{t}; \theta_{t}, \xi_{t})$. Such an approach runs into memory issues when utilized in practice. For example, a mini-batch approximation is proposed in [46], [47] with linearly increasing batch size with iteration index. The standard approach to deal with this issue is to use a biased gradient estimate with low variance instead of an unbiased one. For example, a mini-batch approximation is proposed in [46], [47] with linearly increasing batch size with iteration index. Such an approach runs into memory issues when utilized in practice. Another line of work suggests the use of gradient tracking given by $d_{t} = (1 - \rho_{t})(d_{t-1} - h(x_{t-1}, y_{t-1}; \theta_{t}, \xi_{t})) + h(x_{t}, y_{t}; \theta_{t}, \xi_{t})$, where $\rho_{t}$ being the tracking parameter, which does not suffer from the problem of increasing batch size. But such gradient tracking schemes are shown to be suboptimal even for single-level optimization problems [10], so no scope for much harder bi-level problems is considered in this work. To address the issue of memory and iterate divergence, we took motivation from the momentum based approach in [25] and propose to use the following gradient tracking scheme given by

We remark that such a tracking technique is recently utilized in [19] for projection-based bi-level optimization problems. However, in this work, our focus lies in developing projection-free algorithms, and hence analysis is significantly different from [19]. We will show in Lemma 3 (supplementary material) that momentum based tracking technique such as in (17) results in a reduced variance for the gradient estimate, hence eventually results in

---

Algorithm 1 Stochastic Bi-level Frank Wolfe (SBFW)

Input: $x_{t} \in X$, $y_{1} \in \mathbb{R}^{m}$, $\eta_{t}$, $\delta_{t}$, $\rho_{t}$, $\beta_{t}$, and $d_{1} = h_{1}(\theta_{1}; \xi_{1})$ using (9)

for $t = 2$ to $T$ do

Update approximate inner optimization solution

$$y_{t} = y_{t-1} - \delta_{t} \nabla_{\theta_{t}} g(x_{t-1}, y_{t-1}, \xi_{t})$$

(13)

Gradient tracking evaluate $h(x_{t}, y_{t}; \theta_{t}, \xi_{t})$ and $h(x_{t-1}, y_{t-1}; \theta_{t}, \xi_{t})$ using (9) and compute

$$d_{t} = (1 - \rho_{t})(d_{t-1} - h(x_{t-1}, y_{t-1}; \theta_{t}, \xi_{t})) + h(x_{t}, y_{t}; \theta_{t}, \xi_{t}).$$

(14)

Evaluate feasible direction $s_{t} = \arg\min_{s \in X} \langle s, d_{t} \rangle$

Update solution $x_{t+1} = (1 - \eta_{t})x_{t} + \eta_{t}s_{t}$

end for

Output: $x_{T+1}$ or $\check{x}$, where $\check{x}$ chosen uniformly at random from $\{x_{1}, x_{2}, \cdots, x_{T}\}$
In this section, we propose a separate novel algorithm for problems of the form in (5) in a projection-free manner. The details about the specific choice of step sizes \( \delta_t, \rho_t, \) and \( \eta_t \) are discussed in Sec. [IV].

### B. Stochastic Compositional Frank Wolfe (SCFW) Algorithm

In this section, we propose a separate novel algorithm for problems of the form in (5) in a projection-free manner. To solve compositional problems in (5), the classical SGD methods require the unbiased samples of the actual gradient \( \nabla C(x_t) \) given by \( \nabla C(x_t, \theta_t, \xi_t) := \nabla h(x_t, \xi_t)^\top \nabla f(\mathbb{E}_\xi[h(x_t, \xi)], \theta_t) \). That is, for a given vector \( x_t \) and random sample \( \xi_t \) and \( \theta_t \), this gradient computation requires \( \mathbb{E}_\xi[h(x_t, \xi)] \) which is not available on a single query to sampling oracle. The existing projection-based stochastic compositional methods adopt some kind of tracking approach to approximate \( \mathbb{E}_\xi[h(x_t, \xi)] \) using the samples returned by the oracle. For instance, the seminal work SCGD in [13] approximated \( h(x_t) \) at each iteration as \( y_t = (1 - \delta_t)(y_{t-1} - h(x_{t-1}, \xi_t)) + h(x_t, \xi_t) \) where \( \delta_t \) is a diminishing step size but achieved suboptimal convergence rate of \( \mathcal{O}(1/\epsilon^4) \). Recently, work in [16] proposed Adam-SCSC that introduces a stochastic correction to the original SCGD [13] using Adam-type adaptive gradient approach and achieved the convergence rate of \( \mathcal{O}(1/\epsilon^2) \). However, all these existing algorithms require projection onto the set \( \mathcal{X} \) at each iteration. In the proposed algorithm SCFW, similar to SBFW, we proposed to use momentum-based tracking technique to estimate \( y_t \) as

\[
y_t = (1 - \delta_t)(y_{t-1} - h(x_{t-1}, \xi_t)) + h(x_t, \xi_t). 
\]  

---

**Algorithm 2** Stochastic Compositional Frank Wolfe (SCFW)

**Input**: \( x_0 \in \mathcal{X}, \eta_t, \delta_t, \rho_t, y_0 = h(x_0, \xi_0), \) \( d_0 = \nabla F(x_0, \theta_0, \xi_0). \)

for \( t = 1 \) to \( T \) do

   **Inner function tracking**
   \[
y_t = (1 - \delta_t)(y_{t-1} - h(x_{t-1}, \xi_t)) + h(x_t, \xi_t)
   \]

   **Gradient tracking**
   \[
d_t = (1 - \rho_t)(d_{t-1} - \nabla F(x_{t-1}, y_{t-1}, \theta_t, \xi_t)) + \nabla F(x_t, y_t, \theta_t, \xi_t)
   \]

   **Evaluate** feasible direction \( s_t = \arg \min_{s \in \mathcal{X}} \langle s, d_t \rangle \)

   **Update** solution \( x_{t+1} = (1 - \eta_t)x_t + \eta_ts_t \)

end for

**Output**: \( x_{T+1} \) or \( \hat{x} \), where \( \hat{x} \) chosen uniformly at random from \( \{x_1, x_2, \cdots, x_T\} \)
After utilizing $y_t$, we denote the stochastic gradient estimate of compositional objective as $\nabla C(x_t, y_t, \theta_t, \xi_t) = \nabla h(x_t, \xi_t)^\top \nabla f(y_t, \theta_t)$, and then we proposed to use the momentum based method to track the compositional gradient as

$$d_t = (1 - \rho_t)(d_{t-1} - \nabla C(x_{t-1}, y_{t-1}, \theta_t, \xi_t)) + \nabla C(x_t, y_t, \theta_t, \xi_t). \tag{21}$$

We will establish that using such momentum based tracking technique for both the inner function (20) as well as on the gradient of the objective (21) not only reduces the variance of the approximation noises with each iteration but also helps in establishing optimal convergence results. The rest of the steps essentially remains the same as SBFW; that is, we solve a linear minimization problem $s_t = \arg \min_{s \in X} \langle s, d_t \rangle$ and then update the iterate as $x_{t+1} = (1 - \eta_{t+1})x_t + \eta_{t+1}s_t$. The proposed SCFW algorithm is summarized in Algorithm 2. The details about the specific choice of step sizes $\delta_t$, $\rho_t$ and $\eta_t$ are provided in Sec. IV.

IV. CONVERGENCE ANALYSIS

This section presents the convergence rate analysis for the proposed Algorithm 1 and Algorithm 2. We provide the convergence rate results for both convex and non-convex objectives. Before proceeding towards the analysis, we first discuss the convergence criteria we utilize to evaluate the performance of the proposed algorithms.

Convergence Criteria: For the convex objective $Q(x)$, we use the expected suboptimality $E[Q(x_T) - Q(x^*)]$ after $T$ number of iterations where $x^*$ is as defined in (4). The expectation here is with respect to the randomness in the objective as well as iterate $x_T$. However, for non-convex $Q(x)$, we use the Frank-Wolfe gap as the performance metric defined as

$$E[G(x)] := \max_{v \in X} \langle v - x, -\nabla Q(x) \rangle. \tag{22}$$

The Frank-Wolfe gap is a standard performance metric for the constrained non-convex settings as mentioned in [12], [47], [50]. From the definition in (22), we note that $G(x) \geq 0$ for all $x \in X$ and if $\exists x' \in X$ such that $G(x') = 0$, then $x'$ is a first-order stationary point. Further, for comparison with other methods, we will use the stochastic first order oracle (SFO) complexity which is a commonly used metric to compare stochastic first order methods. SFO defines the total number of times an algorithm is required to call first order oracle (which provides stochastic gradients) to reach $\epsilon$-approximate (or stationary) solution. Another metric that is often used to evaluate stochastic projection-free algorithms is the linear minimization oracle (LMO) complexity, which is the total number of times an algorithm needs to solve the linear minimization problem to reach $\epsilon$-approximate (or stationary) solution. However, since both the proposed SBFW and SCFW algorithms and most of the existing state of the art projection-free methods require $O(1)$ LMO calls at each iteration, the LMO complexity is similar to the SFO complexity. We will discuss the SFO complexity of each algorithm after the main theorems.

A. Convergence Analysis of SBFW (Algorithm 1)

We start with the assumptions required for the analysis of bi-level problem in (4).
Assumption 1 For the bi-level stochastic optimization problem in [4], we need the following statements to hold true for the analysis.

(i) The stochastic gradient estimates satisfies
\[ \mathbb{E}[||\nabla_x f(x, y) - \nabla_x f(x, y, \theta)||^2] \leq \sigma_x^2, \]
\[ \mathbb{E}[||\nabla_y f(x, y) - \nabla_y f(x, y, \theta)||^2] \leq \sigma_y^2, \]
\[ \mathbb{E}[||\nabla^2_{xy} g(x, y) - \nabla^2_{xy} g(x, y, \theta)||^2] \leq \sigma_{xy}^2, \]
\[ \mathbb{E}[||\nabla_y g(x, y) - \nabla_y g(x, y, \xi)||^2] \leq \sigma_y^2, \]
for some \( \sigma_x^2 > 0, \sigma_y^2 > 0, \sigma_{xy}^2 > 0, \) and \( \sigma_y^2 > 0. \)

(ii) For any given \( x \in \mathcal{X}, \) the terms \( \nabla_x f(x, y), \nabla_y f(x, y), \nabla_y g(x, y), \nabla^2_{xy} g(x, y) \) and \( \nabla^2_{yy} g(x, y) \) are Lipschitz continuous with respect to \( y \) with Lipschitz parameter \( L_{f_x}, L_{f_y}, L_g, L_{g_{xy}} \) and \( L_{g_{yy}}, \) respectively.

(iii) For any given \( y \in \mathbb{R}^n, \) the terms \( \nabla_y f(x, y), \nabla^2_{xy} g(x, y) \) and \( \nabla^2_{yy} g(x, y) \) are Lipschitz continuous with respect to \( x \) with positive constants \( L_{f_y}, L_{g_{xy}} \) and \( L_{g_{yy}}, \) respectively. Note that for the sake of simplicity, here we slightly abused the notation and used the same constants as in Assumption 2(iii).

(iv) For all \( x \in \mathcal{X} \) and \( y \in \mathbb{R}^n, \) it holds that \( \mathbb{E}[||\nabla_y f(x, y)||] \leq C_y \) and \( \mathbb{E}[||\nabla^2_{xy} g(x, y)||] \leq C_{xy} \) for some for constants \( C_y > 0 \) and \( C_{xy} > 0. \)

(v) The inner function \( g(x, y) \) is \( \mu_y \)-strongly convex in \( y \) for any \( x \in \mathcal{X}. \)

These assumptions are similar to the assumptions considered in the existing literature [18], [20].

Before proceeding towards the analysis, we provide a Lemma regarding the property of the bias induced by the gradient estimate [9] for bi-level problems.

Lemma 1 Under Assumption 2 consider the estimator defined in [9], then

(i) define bias \( B_t := \mathbb{E}[h(x_t, y_t; \theta, \xi_t)] - \nabla S(x_t, y_t), \) it holds that we have,
\[ ||B_t|| \leq (C_{xy} C_y / \mu_y) (1 - (\mu_y / L_g))^k, \] \( \tag{23} \)
\[ \mathbb{E}[h(x_t, y_t; \theta, \xi_t) - \nabla S(x_t, y_t) - B_t]^2 \leq \sigma_t^2, \] \( \tag{24} \)
where \( \sigma_t^2 = \sigma_x^2 + \frac{3}{\mu_y^2} \left[ (\sigma_y^2 + C_y^2)(\sigma_{xy}^2 + 2C_{xy}^2) + \sigma_y^2 C_{xy}^2 \right]. \)

(ii) For \( t \geq 0, \) it is possible select \( k \) (required to approximate the Hessian inverse in \( [10] \)) such that \( ||B_t|| \leq \beta_t \) where \( \beta_t \leq ct^a \) for some constant \( c \) and \( a > 0. \)

For proof of Lemma 1(i) see [Lemma 11, 18]. The proof Lemma 1(ii) is straight forward. From (23) we have \( \beta_t \leq \mathcal{O}(1 - \mu_y / L_g)^k. \) Now on setting \( k = \mathcal{O}(\log(t)) \) we can get the required condition as \( \beta_t \leq c t^a. \) It shows that with proper selection of \( k, \) we can make the bias to decay polynomially to zero. Next, we characterize the optimality gap of the lower level problem and the error of the momentum based gradient estimator.
Lemma 2 Consider the proposed Algorithm \textsuperscript{[7]} and \( x_t \) be the iterates generated by it, then for the algorithm parameter \( \delta_t \leq \frac{2}{\eta_t} \) and step size \( \eta_t \), the optimality gap of the lower level problem satisfies

\[
E_t[\|y_t - y^*(x_t)\|^2] \leq \left(1 - \frac{\delta_t \mu_y}{2}\right) E_t[\|y_{t-1} - y^*(x_{t-1})\|^2] + \frac{2\eta_t^2}{\delta_t \mu_y} \left(\frac{C_{xy}}{\mu_y}\right)^2 D^2 + 4\delta_t^2 \sigma_y^2.
\]  

The proof of Lemma \textsuperscript{2} is provided in section \textsuperscript{VIII-A} of the supplementary material. Lemma \textsuperscript{2} quantifies how close \( y_t \) is from the optimal solution of inner problem at \( x_t \) and establishes the progress of the inner-level update.

Lemma 3 Consider the proposed Algorithm \textsuperscript{[7]} and \( x_t \) be the iterates generated by it, then for the algorithm parameter \( \delta_t, \rho_k \) and \( \eta_t \), we have

\[
E_t[\|d_t - \nabla S(x_t, y_t) - B_t\|^2] \leq (1 - \rho_k)^2 E_t[\|d_{t-1} - \nabla S(x_{t-1}, y_{t-1}) - B_{t-1}\|^2] + 4L_k \delta_t^2 L_g^2 + 4L_k \eta_t^2 D^2 + 2\rho_k^2 \sigma_f^2.
\]  

The proof of Lemma \textsuperscript{3} is provided in section \textsuperscript{VIII-B} of the supplementary material. Lemma \textsuperscript{3} describes the tracking error in the gradient approximation \( \nabla S(x_t, y) \) at point \( x_t \) and \( y_t \). The presence of \( (1 - \rho_k)^2 \) term in RHS of (26) shows that the variance of the tracking error reduces with iteration.

Theorem 1 \textbf{(Convergence Rate of SBFW)} Consider the proposed Algorithm \textsuperscript{[7]} and suppose Assumption \textsuperscript{[7]} is satisfied. Then,

(i) \textbf{(Convex Bi-Level):} If \( Q \) is convex on \( \mathcal{X} \) and we set \( \delta_t = \frac{2a_0}{t^q} \), where \( a_0 = \min\{\frac{1}{3\mu_y}, \frac{\mu_y}{2(1+\sigma_q) L_g}\} \), \( \rho_k = \frac{2}{t^q} \), \( \beta_t \leq \frac{C_{xy} C_x}{\rho_y(t+1)^2} \) and \( \eta_t \leq \frac{2}{(t+1)^3 \eta_t} \) for \( 0 < q \leq 1 \), the gradient approximation error \( E[\|\nabla Q(x_t) - d_t\|^2] \) converges to zero at the following rate

\[
E[\|\nabla Q(x_t) - d_t\|^2] \leq \frac{C_1}{(t+1)^q},
\]  

where \( C_1 = 3(\max\{2^q \|y_1 - y^*(x_1)\|^2, (2(C_{xy}/\mu_y)^2 D^2 + 16a_0^2 \sigma_g^2)/(2a_0 - 1)\} + \frac{C_{xy} C_x}{\mu_y} + 8(2L_k L_g^2 + L_k D^2 + \sigma_f^2)) \).

The result in Corollary \textsuperscript{1} is presented in general form and indicates that for properly chosen parameters \( q \), the gradient approximation error in expectation decreases at each iteration and approaches zero asymptotically. We will use this upper bound to prove the convergence of the proposed algorithm SBFW for different type of objective functions in the following theorem. Note that in the analysis of Corollary \textsuperscript{1} we have set \( \beta_t \leq \frac{C_{xy} C_x}{\rho_y(t+1)^2} \). To satisfy this condition, the number of samples \( k \) at iteration \( t \) needed to approximate the Hessian inverse in \textsuperscript{(10)} is \( k = O(\log((1 + t)^q)) \).

Now we are ready to present the first main result of this work as Theorem \textsuperscript{1}.

Theorem 1 \textbf{(Convergence Rate of SBFW)} Consider the proposed Algorithm \textsuperscript{[7]} and suppose Assumption \textsuperscript{[7]} is satisfied. Then,

(i) \textbf{(Convex Bi-Level):} If \( Q \) is convex on \( \mathcal{X} \) and we set \( \delta_t = \frac{2a_0}{t^q} \), where \( a_0 = \min\{\frac{1}{3\mu_y}, \frac{\mu_y}{2(1+\sigma_q) L_g}\} \), \( \rho_k = \frac{2}{t^q} \), \( \eta_t = \frac{2}{t^q} \) and \( k = \frac{2\mu}{\eta_t}(\log(1 + t)) \), then the output is feasible \( x_{T+1} \in \mathcal{X} \) and satisfies

\[
E[Q(x_{T+1}) - Q(x^*)] \leq \frac{12D \sqrt{C_1}}{5(T+1)^{3/2}} + \frac{2L_Q D^2}{(T+1)}.
\]
(ii) (Non-Convex Bi-level): If $Q$ is non-convex and we set $\delta_t = \frac{3a_0}{\epsilon^3 t^2}$, where $a_0 = \min\{\frac{1}{3\mu_y}, \frac{\mu_y}{2(1+\sigma^2_y)D}\}$, $\eta_t = \frac{\sigma_y}{T + 1}^{\frac{3}{4}}$, and $k = \frac{L_y}{2\mu_y}(\log(1 + t))$, then the output is feasible $\hat{x} \in \mathcal{X}$ and satisfies

$$E[\mathcal{G}(\hat{x})] \leq \frac{Q(x_1) - Q(x^*)}{(T + 1)^{1/4}} + \frac{16D\sqrt{C_1}}{3(T + 1)^{1/4}} + \frac{L_QD^2}{(T + 1)^{3/4}}.$$ 

Here $L_Q = (L_{xy} + L_{xy})C_{xy} \mu_y^2 + L_x + C_y \left[ \frac{L_{xy}C_x}{\mu_y} + \frac{L_{xy}C_{xy}}{\mu_y^2} \right]$ and $C_1 = 3(\max\{2\|y_1 - y^*(x_1)\|^2, (2C_{xy}/\mu_y)^2D^2 + 16a_0^2\sigma_y^2)/(2a_0 - 1)\} + \frac{C_{xy}C_x}{\mu_y} + 8(2L_kL_y^2 + L_kD^2 + \sigma_y^2)).$

The proof of Theorem 1 is provided in section VIII-D of the supplementary material. Theorem 1 shows that the optimality gap for SBFW decays as $O(T^{-1/3})$ for general convex objectives and for non-convex case it establishes an upper bound on the expected Frank-Wolfe gap for the iterates generated by SBFW that converges to zero at least at the rate of $O(T^{-1/4})$, where $T$ is the total number of iterations. It must be noted that for at each iteration, SBFW requires $2k + 1$ gradient samples to obtain gradient estimate: $2k$ samples for outer gradient estimate (17) and one sample for inner variable update (12). Further, we have set $k \approx O(\log(t))$. Hence, the SFO complexity of SBFW for outer objective is $O(\log(\epsilon^{-1})\epsilon^{-3}) \approx O(\epsilon^{-3})$ and $O(\log(\epsilon^{-1})\epsilon^{-4}) \approx O(\epsilon^{-4})$ for convex and non-convex objective, respectively. Similarly, observe that $E[\|y_t - y^*(x_t)\|] \leq O((t + 1)^{-q})$ (see (71) in the supplementary material), where $q = 2/3$ (as $\delta_t = O(t^{-2/3})$) for convex objective and $q = 1/2$ (as $\delta_t = O(t^{-1/2})$) for non-convex objective. Hence, the SFO complexity of inner objective for SBFW is $O(\epsilon^{-1.5})$ and $O(\epsilon^{-2})$ for convex and non-convex function, respectively. It can be seen that complexity for inner level objective of the proposed algorithm SBFW is comparable to the projection-based state of the art methods [7], [9], [17]–[19], however, it shows slightly worse performance in terms of the outer level complexity. This is not surprising as we are tackling the outer level in a projection-free manner.

B. Convergence Analysis of SCFW (Algorithm 2)

In this section, we will discuss results for our second proposed algorithm SCFW. The basic assumptions for SCFW are essentially the same as SBFW; however, for the sake of clarity, we state them here explicitly. It must be noted that for compositional problems, the inner function $h(\cdot)$ is not necessarily required to be strongly convex.

Assumption 2 For the compositional stochastic optimization problem in (4), we need the following statements to hold true for the analysis.

(i) (Sampling oracle) For a given $x \in \mathcal{X}$ and $y \in \mathbb{R}^n$, the oracle returns samples $h(x, \xi)$, $\nabla h(x, \xi)$ and $\nabla f(y, \theta)$ for some random sample $\xi$ and $\theta$. These samples are unbiased that is $E_\xi[h(x, \xi)] = h(x)$ and $E_{\xi,\theta}[\nabla h(x, \xi)\trans \nabla f(y, \theta)] = \nabla h(x)\trans \nabla f(y)$.

(ii) (Bounded Variance) The inner function $h(\cdot)$ has bounded variance, i.e.,

$$E[\|h(x, \xi) - h(x)\|] \leq \sigma^2_h.$$ 

(29)

(iii) (Lipschitz continuous) The functions $f(\cdot)$ and $h(\cdot)$ are smooth, i.e., for any $y_1, y_2 \in \mathbb{R}^n$

$$\|\nabla f(y_1, \theta) - \nabla f(y_2, \theta)\| \leq L_f \|y_1 - y_2\|,$$

(30)
and for any \( x_1, x_2 \in \mathcal{X} \), it holds that
\[
\| \nabla h(x_1, \xi) - \nabla h(x_2, \xi) \| \leq L_h \| x_1 - x_2 \| .
\] (31)

(iv) (Bounded second moments) The stochastic gradients of functions \( f(\cdot) \) and \( h(\cdot) \) have bounded second order moments, i.e., for any \( x \in \mathcal{X} \), and \( y \in \mathbb{R}^n \)
\[
E[|\nabla f(y, \theta)|^2] \leq M_f; \quad E[\|\nabla h(x, \xi)\|^2] \leq M_h.
\] (32)

Note that for SCFW, we have assumed both the inner and outer functions to be smooth; the resulting composition function \( C(x) \) will also be smooth \cite{51}. Its smoothness parameter \( L_F \) can be easily obtained using Assumption \ref{assumption:smoothness} \( \iii \) \( \iv \) as \( L_F = M_h^2 L_f + M_f L_h \).

Now, in order to study the convergence rate of the proposed algorithm SCFW, we state some results on the error of the inner function approximation and gradient approximation in the form of following lemma.

**Lemma 4** Consider the proposed Algorithm \ref{algorithm:SCFW} and \( x_t \) be the iterates generated by it, then the sequence \( \|y_t - h(x_t)\|^2 \) converges to zero at the following rate
\[
E \|y_t - h(x_t)\|^2 \leq (1 - \delta t)^2 E \|(y_{t-1} - h(x_{t-1}))\|^2 + 2\delta^2 t^2 \sigma^2 h + 2(1 - \delta t)^2 \eta^2_{t-1} M_h D^2.
\] (33)

The proof of Lemma \ref{lemma:inner_convergence} is provided in section \[IX-A\] of the supplementary material. Lemma \ref{lemma:inner_convergence} shows that the distance between \( y_t \) and \( h(x_t) \) decreases with iteration in expectation. Intuitively, this means that our tracking variable \( y_t \) will converge to the unknown \( h(x_t) \). This result will be used to obtain the convergence rates of proposed SCFW algorithm for different kind of objective functions.

Next, we will find bound on the norm square difference \( E \|d_t - \hat{\nabla} C(x_t, y_t)\|^2 \) in the following Lemma.

**Lemma 5** For the iterates \( x_t \) generated by the proposed Algorithm \ref{algorithm:SCFW} the norm square difference \( E \|d_t - \hat{\nabla} C(x_t, y_t)\|^2 \) can be bounded as
\[
E \|d_t - \hat{\nabla} C(x_t, y_t)\|^2 \leq (1 - \rho t)^2 E \|d_{t-1} - \hat{\nabla} C(x_{t-1}, y_{t-1})\|^2 + 4\rho^2 t M_f M_h
\]
\[
+ 4(1 - \rho t)^2 [(M_f L_h + 3M_h^2 L_f)\eta^2_{t-1} D^2 + 3M_h L_f \delta^2 t E_t [\|y_{t-1} - h(x_{t-1})\|^2] + 3\delta^2 t M_h L_f \sigma^2 h],
\]
where \( \hat{\nabla} C(x_t, y_t) = E[\nabla C(x_t, y_t, \theta_t, \xi_t)] \).

The proof of Lemma \ref{lemma:gradient_convergence} is provided in section \[IX-B\] of the supplementary material. Note that instead of directly using the gradient samples, we are approximating it using tracking technique. Hence, we will derive an upper bound on the gradient approximation error \( E \|\nabla F(x_t) - d_t\|^2 \) using Lemma \ref{lemma:inner_convergence} and \ref{lemma:gradient_convergence} in the form of following Corollary \ref{corollary:gradient_convergence} with proof in section \[IX-C\] of the supplementary material.

**Corollary 2** For the proposed Algorithm \ref{algorithm:SCFW} with \( \delta t = \frac{2}{p} \), \( \rho t = \frac{2}{p} \) and \( \eta t \leq \frac{2}{(t+1)^p} \) for \( 0 < p \leq 1 \), the gradient approximation error \( E \|\nabla C(x_t) - d_t\|^2 \) converges to zero at the following rate
\[
E \|\nabla C(x_t) - d_t\|^2 \leq \frac{A_1}{(t+1)^p}
\] (35)
where $A_1 := 32[M_h(M_f + 28L_f \sigma_h^2) + (M_f L_h + 28M_h^2 L_f)D^2]$.

The result in Corollary 2 indicates that for properly chosen parameters $p$, the gradient approximation error in expectation decreases at each iteration and approaches zero asymptotically. We will use this upper bound to prove the convergence of the proposed algorithm SCFW.

Now, we are ready to present the second main result of this work in Theorem 2 which characterizes the convergence rate for the SCFW algorithm.

**Theorem 2 (Convergence Rate of SCFW)** Consider the proposed Algorithm 2 and suppose that Assumption 2 is satisfied. Then,

(i) (Convex Compositional): If $C$ is convex on $X$ and we set $\rho_t = \frac{2}{t}$, $\delta_t = \frac{2}{t}$ and $\eta_t = \frac{2}{t+1}$, then the output is feasible $x_{T+1} \in X$ and satisfies

$$
\mathbb{E}[C(x_{T+1}) - C(x^*)] \leq \frac{8D\sqrt{A_1}}{3(T + 1)^{1/2}} + \frac{2L_F D^2}{(T + 1)^{1/2}},
$$

(ii) (Non-Convex Compositional): If $C$ is non-convex and we set $\rho_t = \frac{2}{t^{2/3}}$, $\delta_t = \frac{2}{t^{2/3}}$ and $\eta_t = \frac{2}{(T+1)^{2/3}}$, then the output is feasible $\hat{x} \in X$ and satisfies

$$
\mathbb{E}[G(\hat{x})] \leq \frac{C(x_1) - C(x^*)}{(T + 1)^{1/3}} + \frac{6D\sqrt{A_1}}{(T + 1)^{1/3}} + \frac{L_F D^2}{(T + 1)^{2/3}},
$$

where $L_F = M_f^2 L_f + M_f L_h$ and $A_1 := 32[M_h(M_f + 28L_f \sigma_h^2) + (M_f L_h + 28M_h^2 L_f)D^2]$.

The proof of Theorem 2 is provided in section IX-D of the supplementary material. Interestingly, the optimality gap decays as $O(T^{-1/2})$, which is the optimal rate even for projected stochastic compositional optimization problems with general convex objectives [14]. For the non-convex case, it establishes an upper bound on the expected Frank-Wolfe gap for the iterates generated by SCFW and shows that it converges to zero at least at the rate of $O(T^{-1/3})$, where $T$ is the total number of iterations. It must be noted that at each iteration, SCFW requires only two gradient samples to obtain gradient estimate. Hence, it has SFO complexity is $O(\epsilon^{-2})$ and $O(\epsilon^{-3})$ for convex and non-convex objectives, respectively. Interestingly, these results match with the state-of-the-art methods [11], [12] for projection-free single level (or non-compositional) stochastic optimization.

V. NUMERICAL EXPERIMENTS

A. Matrix Completion with Denoising

For the experiments, we consider the problem of low-rank matrix completion formulated in (3). Note that (3) is of the form (4) with $f(\cdot) = g(\cdot) = ||\cdot||_F^2$, and $(\theta_t, \xi_t)$ being the independent random subset of entries revealed at every iteration $t$. We emphasize that there are two challenges (C1 and C2) involved in solving the problem in (3): C1: the problem is bi-level stochastic in nature, and C2: the nuclear norm constraints are expensive to project onto. We will show that we can address both the challenges by solving the problem in (3) via the proposed SBFW algorithm.

We start with performing experiments to highlight the importance of denoising in matrix completion which lead to the bi-level formulation of the problem in (3). First, we solve the problem in (3) using the proposed SBFW
Fig. 1: This figure compares the performance of the proposed SBFW algorithm for matrix completion problem with the other state-of-the-art algorithms such as TTSA [18], MSTSA [20], and SUSTAIN [19]. Fig. 1a shows the benefit of bilevel formulation (solved via SBFW) for the matrix completion problem as compared to the standard single level problem (solved via SFW). We note that SBFW achieves a lower normalized error as compared to SWF. This advantage of SBFW is further confirmed in Fig. 1b, which plots the distance from the optimal value $\|\bar{e} - \bar{e}_0\|$ with respect to the noise factor in the matrix completion problem. The proposed algorithm is able to better handle the noise in the observed matrices by a significant margin. Next, in Fig. 1c and Fig. 1d, we compared the advantage of using the proposed Frank-Wolfe based method as compared to the other projection-based counterparts in the literature. Fig. 1c plots the convergence rates and Fig. 1d shows a bat plot of computation time of all the different methods, and presents that SBFW performs the best.

algorithm and compare it with the state-of-the-art conditional gradient method called SFW [10]. The evolution of the normalized error (defined in Appendix X-A in the supplementary material) is shown in Fig. 1a. Since SFW is a projection-free algorithm, it addresses the challenge C2 and converges fast but to a suboptimal point, while SBFW converges to better accuracy but slowly. Hence, the proposed algorithm solves the bi-level matrix completion problem efficiently (addressed challenge C1). Then, to further investigate the effectiveness of the proposed algorithm in addressing C1, we solve the problem for different noise factor $\hat{n} \in (0, 1)$ and compare the normalized error in the solution at each $\hat{n}$ with the normalized error obtained for zero noise case (i.e. $\hat{n} = 0$) denoted as $\bar{e}_0$ (shown in Fig. 1b). We note that the growth in the error difference is much slower for SBFW than SFW, which shows the effectiveness of the proposed algorithm in dealing with noise.

Next, we show the advantage of utilizing the proposed projection-free method to solve the problem in (3) as compared to other projection-based methods available in the literature such as SUSTAIN [19], TTSA [18], and MSTSA [20]. We plot the normalized error convergence performance for all the algorithms in Fig. 1c. The proposed
algorithm is not the fastest in terms of the convergence rate when compared to projection-based schemes, which is expected from the theoretical convergence analysis. However, in Fig. 1d we plot the computational time required to reach the normalized error of $\bar{e} \approx 10^{-2}$ for all the algorithms that shows the advantage of the proposed scheme. This performance gain comes from the fact that all the other methods require to perform projections over nuclear norm at each iteration which is computationally expensive due to the computation of full singular value decomposition. In contrast, SBFW requires to solve only a single linear program over nuclear norm constraint, which only requires the computation of the singular vectors corresponding to the highest singular value. The experimental details of the matrix completion is provided in section X-A of the supplementary material.

B. Sparse Policy Value Evaluation

![Fig. 2: Convergence results for ACSPG and SCFW algorithms](image)

We consider the problem of policy value evaluation in reinforcement learning as discussed in [14]. Consider a Markov Decision Process (MDP) with finite state space $S$ and action space $A$. For a fixed policy $\pi$ which maps the current state $s$ to action $a \in A$, the value function $V^\pi(s)$ for state $s$ is given by

$$V^\pi(s) = \mathbb{E}\{r_{s,\hat{s}} + \gamma V^\pi(\hat{s})|s, \pi\}$$

for all $s, \hat{s} \in S$, (37)

where $r_{s,\hat{s}}$ is the reward for transitioning from $s$ to $\hat{s}$, $\gamma \in (0, 1)$ is the discount factor, and the expectation is taken over all the possible future states $\hat{s}$ conditioned on current state $s$ and the policy $\pi$. Looking at the bellman equation in (37), it is clear that evaluating the value function at all the states is impractical when $|S|$ is moderately large. Hence, we consider a linear function approximation for the value function such that $V^\pi(s) = \phi_s^T w^*$, for some $w^* \in \mathbb{R}^m$. Here, $\phi_s \in \mathbb{R}^m$ denotes the $m$ dimensional state features for state $s$. The goal is to learn an optimal $w^*$ to obtain a suitable linear function approximation for the value function. The problem of finding $w^*$ can be formulated as

$$\min_{\|w\|_1 \leq \alpha} \sum_{s=1}^S \left( \phi_s^T w - q_{\pi,s}(w) \right)^2,$$

(38)

where

$$q_{\pi,s}(w) = \mathbb{E}\{r_{s,\hat{s}} + \gamma \phi_{\hat{s}}^T w|s, \pi\} = \sum_{\hat{s}} P_{s\hat{s}} (r_{s,\hat{s}} + \gamma \phi_{\hat{s}}^T w).$$

(39)
The constraint $\|w\|_1 \leq \alpha$ in (38) is useful in practice, where the sparsity needs to be ensured. For instance, when the number of states in $\mathcal{S}$ is large, the features of each state would become large, thus making the dimension of the optimization variable large. We propose to solve (38) in a projection-free manner using the proposed SCFW algorithm. In the current context, the complete knowledge of transition probabilities $\{P_{ss}^{\pi}\}$ is not known but revealed sequentially.

For the experiments, we consider the number of states $|\mathcal{S}| = 100$ with 3 actions available at every state. Given a pair of state and action, the agent can move any one of the next possible states. The transition probabilities and rewards for each transition are uniformly sampled in $[0, 1]$. For the purposeful behavior, we follow the same approach as in [52], where the agent favors a single action at each state. Out of 3, one action is randomly selected and is assigned with the probability 0.9, and the others are evenly assigned with probabilities. The feature vector of each state has dimension $m = 100$. However, the additional $\ell_1$ constraint ensures the sparsity in optimization variable $w$.

For comparison, we implemented the proximal ASC-PG algorithm [14], where $\lambda\|w\|_1$ is added as a regularizer to the objective function. The parameter $\lambda$ is tuned so that $\|w\|_1 \leq \alpha$ holds. In our experiment, we set $\alpha = 0.3$ and ran both algorithms for 1000 iterations. The results of the experiment is reported in terms of objective convergence $\|w_t - w^*\|_2$ in Fig. 2. Here, $w_t$ is the algorithmic solution at iteration $t$. The optimal solution $w^*$ is obtained by running the algorithm for $10^5$ iterations, given that the complete information of transition probabilities is known. As expected, both of the algorithms have managed to reduce the error with iteration. However, note that ASC-PG is solving a relaxed version of the constraint problem by including the $\ell_1$ norm constraint as regularizer, that requires tuning of parameter $\lambda$. In fact, it might be possible to attain a better convergence plot for ASC-PG by tuning $\lambda$ further, but that might violate the constraint. In contrast, the proposed algorithm is converging while strictly satisfying the constraint.

VI. CONCLUSION

This paper presents the first projection-free algorithm for stochastic bi-level optimization problems with a strongly convex inner objective function. We utilized the concept of momentum based tracking to track the stochastic gradient estimate and show the convergence rates of the proposed SBFW algorithm for the convex and non-convex outer objective functions. We also developed the first projection-free algorithm called SCFW for stochastic compositional problems, which is a special case of bi-level problems. We show that tracking both the inner function and the gradient of the objective function with momentum technique reduces approximation noise which eventually helps obtain the optimal convergence rates.

REFERENCES

[1] A. Rajeswaran, C. Finn, S. Kakade, and S. Levine, “Meta-learning with implicit gradients,” arXiv preprint arXiv:1909.04630, 2019.
[2] Z. Borsos, M. Mutný, and A. Krause, “Coresets via bilevel optimization for continual learning and streaming,” arXiv preprint arXiv:2006.03875, 2020.
[3] H. Zhang, W. Chen, Z. Huang, M. Li, Y. Yang, W. Zhang, and J. Wang, “Bi-level actor-critic for multi-agent coordination,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 05, 2020, pp. 7325–7332.
[4] R. Grazzi, L. Franceschi, M. Pontil, and S. Salzo, “On the iteration complexity of hypergradient computation,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 3748–3758.
[5] L. Franceschi, P. Frasconi, S. Salzo, R. Grazzi, and M. Pontil, “Bilevel programming for hyperparameter optimization and meta-learning,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 1568–1577.
[6] S. Dempe and A. Zemkoho, *Bilevel Optimization*. Springer, 2020.
[7] S. Ghadimi and M. Wang, “Approximation methods for bilevel programming,” *arXiv preprint arXiv:1802.02246*, 2018.
[8] J. Yang, K. Ji, and Y. Liang, “Provably faster algorithms for bilevel optimization,” *arXiv preprint arXiv:2106.04692*, 2021.
[9] T. Chen, Y. Sun, and W. Yin, “A single-timescale stochastic bilevel optimization method;”, *arXiv preprint arXiv:2102.04671*, 2021.
[10] A. Mokhtari, H. Hassani, and A. Karbasi, “Stochastic conditional gradient methods: From convex minimization to submodular maximization,” *Journal of Machine Learning Research*, vol. 21, no. 105, pp. 1–49, 2020.
[11] J. Xie, Z. Shen, C. Zhang, B. Wang, and H. Qian, “Efficient projection-free online methods with stochastic recursive gradient.” in *AAAI*, 2020, pp. 6446–6453.
[12] M. Zhang, Z. Shen, A. Mokhtari, H. Hassani, and A. Karbasi, “One sample stochastic frank-wolfe;,” in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 4012–4023.
[13] M. Wang, E. X. Fang, and H. Liu, “Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions,” *Mathematical Programming*, vol. 161, no. 1-2, pp. 419–449, 2017.
[14] M. Wang, J. Liu, and E. X. Fang. “Accelerating stochastic composition optimization,” *Journal of Machine Learning Research*, 2017.
[15] S. Ghadimi, A. Ruszczyński, and M. Wang, “A single timescale stochastic approximation method for nested stochastic optimization,” *SIAM Journal on Optimization*, vol. 30, no. 1, pp. 960–979, 2020.
[16] T. Chen, Y. Sun, and W. Yin, “Solving stochastic compositional optimization is nearly as easy as solving stochastic optimization,” *arXiv preprint arXiv:2008.10847*, 2020.
[17] K. Ji, J. Yang, and Y. Liang, “Bilevel optimization: Nonasymptotic analysis and faster algorithms,” *arXiv preprint arXiv:2010.07962*, 2020.
[18] M. Hong, H.-T. Wai, Z. Wang, and Z. Yang, “A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic;,” *arXiv preprint arXiv:2007.05170*, 2020.
[19] P. Khanduri, S. Zeng, M. Hong, H.-T. Wai, Z. Wang, and Z. Yang, “A near-optimal algorithm for stochastic bilevel optimization via double-momentum,” *arXiv preprint arXiv:2102.07367*, 2021.
[20] ——, “A momentum-assisted single-timescale stochastic approximation algorithm for bilevel optimization,” *arXiv e-prints*, pp. arXiv–2102, 2021.
[21] T. Chen, Y. Sun, and W. Yin, “Tighter analysis of alternating stochastic gradient method for stochastic nested problems,” *arXiv preprint arXiv:2106.13781*, 2021.
[22] A. Fallah, A. Mokhtari, and A. Ozdaglar, “Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 3557–3568, 2020.
[23] M. Jaggi, “Revisiting frank-wolfe: Projection-free sparse convex optimization,” in *International Conference on Machine Learning*. PMLR, 2013, pp. 427–435.
[24] M. Frank, P. Wolfe et al., “An algorithm for quadratic programming,” *Naval research logistics quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956.
[25] A. Cutkosky and F. Orabona, “Momentum-based variance reduction in non-convex sgd,” *arXiv preprint arXiv:1905.10018*, 2019.
[26] H. Hassani, A. Karbasi, A. Mokhtari, and Z. Shen, “Stochastic conditional gradient++:(non) convex minimization and continuous submodular maximization,” *SIAM Journal on Optimization*, vol. 30, no. 4, pp. 3315–3344, 2020.
[27] P. Jain, P. Netrapalli, and S. Sanghavi, “Low-rank matrix completion using alternating minimization,” in *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, 2013, pp. 665–674.
[28] Y. Chen, X. Xiao, and Y. Zhou, “Low-rank quaternion approximation for color image processing,” *IEEE Transactions on Image Processing*, vol. 29, pp. 1426–1439, 2019.
[29] A. Evgeniou and M. Pontil, “Multi-task feature learning,” *Advances in neural information processing systems*, vol. 19, p. 41, 2007.
[30] B. Li, Q. Yang, and X. Xue, “Transfer learning for collaborative filtering via a rating-matrix generative model,” in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 617–624.
[31] M. T. McCann and S. Ravishankar, “Supervised learning of sparsity-promoting regularizers for denoising,” *arXiv preprint arXiv:2006.0552J*, 2020.
[32] C. Crockett and J. A. Fessler, “Motivating bilevel approaches to filter learning: A case study,” in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 2803–2807.
[33] H. Von Stackelberg and S. H. Von, *The theory of the market economy*. Oxford University Press, 1952.
[34] K. P. Bennett, G. Kunapuli, J. Hu, and J.-S. Pang, “Bilevel optimization and machine learning,” in *IEEE World Congress on Computational Intelligence*. Springer, 2008, pp. 25–47.
[35] F. Huang and H. Huang, “Biam: Fast adaptive bilevel optimization methods,” *arXiv preprint arXiv:2106.11396*, 2021.
[36] ———, “Enhanced bilevel optimization via bregman distance,” *arXiv preprint arXiv:2107.12301*, 2021.
[37] H. Yuan, X. Lian, and J. Liu, “Stochastic recursive variance reduction for efficient smooth non-convex compositional optimization,” *arXiv preprint arXiv:1912.13515*, 2019.
[38] J. Yang and W. Hu, “Stochastic recursive momentum method for non-convex compositional optimization,” *arXiv preprint arXiv:2006.01688*, 2020.
[39] S. T. Thomadapu, K. Rajawat *et al.*, “Stochastic compositional gradient descent under compositional constraints,” *arXiv preprint arXiv:2012.09400*, 2020.
[40] L. Liu, J. Liu, and D. Tao, “Variance reduced methods for non-convex composition optimization,” *arXiv preprint arXiv:1711.04416*, 2017.
[41] Z. Huo, B. Gu, J. Liu, and H. Huang, “Accelerated method for stochastic composition optimization with nonsmooth regularization,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
[42] H. Yuan, X. Lian, C. J. Li, and J. Liu, “Efficient smooth non-convex stochastic compositional optimization via stochastic recursive gradient descent,” 2019.
[43] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, “Robust stochastic approximation approach to stochastic programming,” *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1574–1609, 2009.
[44] A. Nitanda, “Stochastic proximal gradient descent with acceleration techniques,” *Advances in Neural Information Processing Systems*, vol. 27, pp. 1574–1582, 2014.
[45] G. Lan, “An optimal method for stochastic composite optimization,” *Mathematical Programming*, vol. 133, no. 1, pp. 365–397, 2012.
[46] E. Hazan and H. Luo, “Variance-reduced and projection-free stochastic optimization,” in *International Conference on Machine Learning*. PMLR, 2016, pp. 1263–1271.
[47] S. J. Reddi, S. Sra, B. Póczos, and A. Smola, “Stochastic frank-wolfe methods for nonconvex optimization,” in *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2016, pp. 1244–1251.
[48] M. Wang and J. Liu, “A stochastic compositional gradient method using markov samples,” in *2016 Winter Simulation Conference (WSC)*. IEEE, 2016, pp. 702–713.
[49] A. Beck, *First-order methods in optimization*. SIAM, 2017.
[50] S. Lacoste-Julien, “Convergence rate of frank-wolfe for non-convex objectives,” *arXiv preprint arXiv:1607.00345*, 2016.
[51] J. Zhang and L. Xiao, “A stochastic composite gradient method with incremental variance reduction,” *arXiv preprint arXiv:1906.10186*, 2019.
[52] A. White and M. White, “Investigating practical linear temporal difference learning,” *arXiv preprint arXiv:1602.08771*, 2016.
A. Existing Results

We start the discussion by mentioning the existing results in Lemma 6 and 7 which are useful for the analysis in this paper.

**Lemma 6** [19, Lemma 2.2] Under Assumption 1, following statements hold.

(a) For any \( x \in X \) and \( y \in \mathbb{R}^n \),

\[
\| \nabla S(x, y) - \nabla Q(x) \| \leq L \| y^*(x) - y \| ,
\]

where \( L := L_{f_x} + \frac{L_{yy} C_{sx}}{\mu_g} + C_y \left[ \frac{L_{yx}}{\mu_g} + \frac{L_{yy} C_{sx}}{\mu_g} \right] \) and all the constants are as defined in Assumption 1.

(b) The inner optimal solution \( y^*(x) \) is \( \frac{C_{xy}}{\mu_g} \)-Lipschitz continuous in \( x \), which implies that for any \( x_1, x_2 \in X \), it holds that \( \| y^*(x_1) - y^*(x_2) \| \leq \frac{C_{xy}}{\mu_g} \| x_1 - x_2 \| \).

(c) The gradient of outer objective \( \nabla Q \) is \( L_Q \)-Lipschitz continuous in \( x \), which implies that for any \( x_1, x_2 \in X \), it holds that \( \| Q(x_1) - Q(x_2) \| \leq L_Q \| x_2 - x_1 \| \) where \( L_Q := \frac{(L_{f_x} + L_{f_y}) C_{sx}}{\mu_g} + L_{f_x} + C_y \left[ \frac{L_{yx}}{\mu_g} + \frac{L_{yy} C_{sx}}{\mu_g} \right] \).

**Lemma 7** [19, Lemma 4.1] Suppose Assumption 1 holds, and the gradient estimate \( h(x, y; \theta; \xi) \) is constructed with \( k \) number of samples using (9), then

(a) for any \( x \in X \) and \( y_1, y_2 \in \mathbb{R}^n \), we have

\[
\mathbb{E}_t[\| h(x, y_1; \theta_t, \xi_t) - h(x, y_2; \theta_t, \xi_t) \|^2] \leq L_k \mathbb{E}_t[\| y_1 - y_2 \|^2],
\]

(b) for any \( y \in \mathbb{R}^n \) and \( x_1, x_2 \in X \), we have

\[
\mathbb{E}_t[\| h(x_1, y; \theta_t, \xi_t) - h(x_2, y; \theta_t, \xi_t) \|^2] \leq L_k \mathbb{E}_t[\| x_1 - x_2 \|^2],
\]

where

\[
L_k = 2L_{f_x}^2 + \frac{6k[(L_g - \mu_g)^2(C_{g_yy}^2 L_{f_y}^2 + C_{g_yy}^2 L_{g_yy}^2) + k^2 C_{g_yy}^2 C_{f_y}^2 L_{g_yy}^2]}{\mu_g(2L_g - \mu_g)}.
\]

B. General Inequalities

Before proceeding towards the main analysis of this work, we first present and establish a general mathematical inequality in the form of Lemma 8 which will be useful to the analysis in this work. Further, we will present a general upper bound in Lemma 9 on the expected estimation error when the momentum-based method is employed to track the function or gradient.

**Lemma 8** Let \( \psi_t \) be a sequence of real numbers which satisfy

\[
\psi_{t+1} = \left( 1 - \frac{c_1}{(t + t_0)^{r_1}} \right) \psi_t + \frac{c_2}{(t + t_0)^{r_2}}
\]
for some \( r_1 \in (0, 1) \) such that \( r_1 \leq r_2 \leq 2r_1, \ c_1 > 1, \) and \( c_2 \geq 0. \) Then, \( \psi_{t+1} \) would converge to zero at the following rate

\[
\psi_{t+1} \leq \frac{c}{(t + t_0 + 1)^{r_2-r_1}},
\]

where \( c = \max\{\psi_1(t_0 + 1)^{r_2-r_1}, \frac{c_2}{c_1-1}\}. \)

**Proof:**

We prove Lemma 9 by induction. The base step of induction holds (for \( t = 0 \)) from the definition of \( c \) which implies that \( c \geq \psi_1(t_0 + 1)^{r_2-r_1}. \) Next, we assume that (44) holds for \( t = k, \) which means

\[
\psi_{k+1} \leq \frac{c}{(k + t_0 + 1)^{r_2-r_1}}. \tag{45}
\]

Now it remains to show that (44) also holds for \( t = k + 1. \) To proceed, we set \( t = k + 1 \) in (43) to obtain

\[
\psi_{k+2} = \left(1 - \frac{c_1}{(k + t_0 + 1)^{r_1}}\right) \psi_{k+1} + \frac{c_2}{(k + t_0 + 1)^{r_2}}. \tag{46}
\]

From the definition of \( c, \) it holds that \( c_2 \leq c(c_1 - 1) \) and we utilize the upper bound on \( \psi_{k+1} \) from (45) into (46) to obtain

\[
\psi_{k+2} \leq \left(1 - \frac{c_1}{(k + t_0 + 1)^{r_1}}\right) \frac{c}{(k + t_0 + 1)^{r_2-r_1}} + \frac{c(c_1 - 1)}{(k + t_0 + 1)^{r_2}}. \tag{47}
\]

Simplify and rearrange the terms in (47) to write

\[
\psi_{k+2} \leq c \left(\frac{1}{(k + t_0 + 1)^{r_2-r_1}} - \frac{1}{(k + t_0 + 1)^{r_1}}\right). \tag{48}
\]

Next, note that for general \( p, q, \) it holds that \( \frac{1}{x^{p+q}} - \frac{1}{x^p} \leq \frac{1}{(x+1)^{p+q}} \), which allows us to write

\[
\psi_{k+2} \leq \frac{c}{(k + t_0 + 2)^{r_2-r_1}}. \tag{49}
\]

Thus, it holds that the statement in (43) holds for all \( t \geq 0. \)

**Lemma 9** Let us estimate function \( \Psi(x) = \mathbb{E}_t[\Psi(x, \xi_t)] \) by \( y_t \) using step size \( \delta_t \) as follows

\[
y_t = (1 - \delta_t)(y_{t-1} - \Psi(x_{t-1}, \xi_t)) + \Psi(x_t, \xi_t). \tag{50}
\]

Then the expected tracking error \( \mathbb{E}_t[\|y_t - \Psi(x_t)\|^2] \) satisfies

\[
\mathbb{E}_t[\|y_t - \Psi(x_t)\|^2] \leq (1 - \delta_t)^2 \|y_{t-1} - \Psi(x_{t-1})\|^2 + 2(1 - \delta_t)^2 \mathbb{E}_t[\|\Psi(x_t, \xi_t) - \Psi(x_{t-1}, \xi_t)\|^2]
\]

\[
+ 2\delta_t^2 \mathbb{E}_t[\|\Psi(x_t, \xi_t) - \Psi(x_t)\|^2]. \tag{51}
\]

**Proof:**

Consider the update equation in (50), add/subtract the term \( (1 - \delta_t)\Psi(x_{t-1}) \) in the right hand side of (50) to obtain

\[
y_t = (1 - \delta_t)(y_{t-1} - \Psi(x_{t-1}, \xi_t)) + \Psi(x_t, \xi_t) + (1 - \delta_t)\Psi(x_{t-1}) - (1 - \delta_t)\Psi(x_{t-1}). \tag{52}
\]

Subtract \( \Psi(x_t) \) from both sides in (52) and take norm square, we get

\[
\|y_t - \Psi(x_t)\|^2 = \|(1 - \delta_t)(y_{t-1} - \Psi(x_{t-1})) - (1 - \delta_t)(\Psi(x_{t-1}, \xi_t) - \Psi(x_{t-1})) + (\Psi(x_t, \xi_t) - \Psi(x_t))\|^2. \tag{53}
\]
Now, expand the square and calculate conditional expectation $\mathbb{E}_t = \mathbb{E}[(\cdot)|\mathcal{F}_t]$ to obtain

\[
\mathbb{E}_t[\|y_t - \Psi(x_t)\|^2] = (1 - \delta_t)^2 \|y_{t-1} - \Psi(x_{t-1})\|^2 \\
- 2((1 - \delta_t)(y_{t-1} - \Psi(x_{t-1})), (1 - \delta_t)(\mathbb{E}_t[\Psi(x_{t-1}) - \Psi(x_{t-1}, \xi_t)] + \mathbb{E}_t[\Psi(x_t) - \Psi(x_t, \xi_t)]) \\
+ \mathbb{E}_t[(1 - \delta_t)(\Psi(x_{t-1}, \xi_t) - \Psi(x_{t-1})) + \Psi(x_t) - \Psi(x_t, \xi_t)]^2].
\]

(54)

Note that $\mathbb{E}_t[\Psi(x_{t-1}) - \Psi(x_{t-1}, \xi_t)] = 0$ and $\mathbb{E}_t[\Psi(x_t) - \Psi(x_t, \xi_t)] = 0$, which implies that

\[
\mathbb{E}_t[\|y_t - \Psi(x_t)\|^2] = (1 - \delta_t)^2 \|y_{t-1} - \Psi(x_{t-1})\|^2 \\
+ \mathbb{E}_t[(1 - \delta_t)(\Psi(x_{t-1}, \xi_t) - \Psi(x_{t-1})) + \Psi(x_t) - \Psi(x_t, \xi_t)]^2].
\]

(55)

Consider the second term on the right hand side of (55), we write

\[
\mathbb{E}_t[\|(1 - \delta_t)(\Psi(x_{t-1}, \xi_t) - \Psi(x_{t-1})) + (1 - \delta_t + \delta_t)(\Psi(x_t) - \Psi(x_t, \xi_t))\|^2] \\
= \mathbb{E}_t[\|(1 - \delta_t)(\Psi(x_{t-1}) - \Psi(x_{t-1}, \xi_t)) + \Psi(x_t, \xi_t) - \Psi(x_t))\|^2] + \delta_t^2 \mathbb{E}_t[\|(\Psi(x_t, \xi_t) - \Psi(x_t))\|^2] \\
\leq 2(1 - \delta_t)^2 \mathbb{E}_t[\|(\Psi(x_t, \xi_t) - \Psi(x_{t-1}, \xi_t))\|^2] + 2\delta_t^2 \mathbb{E}_t[\|(\Psi(x_t, \xi_t) - \Psi(x_t))\|^2],
\]

(56)

where the last inequality holds due to the fact that $\mathbb{E} \|X - \mathbb{E}[X] + Y\|^2 \leq 2\mathbb{E} \|X\|^2 + 2\mathbb{E} \|Y\|^2$ for any two random variables $X$ and $Y$. Using the upper bound in (56) into (55) results in the desired statement of Lemma 9.

Note that for SCFW, we have assumed both the inner and outer functions to be smooth; the resulting composition function $C(x)$ will also be smooth [51]. Its smoothness parameter $L_F$ can be easily obtained using Assumption 2 (iii-iv) as $L_F = M_h^2 L_f + M_f L_h$.

VIII. PROOFS FOR SBFW

We start the analysis by presenting intermediate Lemmas [22] and Corollary 2 which eventually leads to the main result of this section presented in Theorem 1.

A. Proof of Lemma 2

Let us consider the term $\mathbb{E}_t[\|y_t - y^*(x_{t-1})\|^2]$ and from the update equation of Algorithm 1 in (13), we can write

\[
\mathbb{E}_t[\|y_t - y^*(x_{t-1})\|^2] = \mathbb{E}_t[\|y_{t-1} - \delta_t \nabla_y g(x_{t-1}, y_{t-1}, \xi_t) - y^*(x_{t-1})\|^2].
\]

(57)

By expanding the square and taking conditional expectation term inside the inner product terms, we obtain

\[
\mathbb{E}_t[\|y_t - y^*(x_{t-1})\|^2] = \mathbb{E}_t[\|y_{t-1} - y^*(x_{t-1})\|^2] + \delta_t^2 \mathbb{E}_t[\|\nabla_y g(x_{t-1}, y_{t-1}, \xi_t)\|^2] \\
- 2\mathbb{E}_t[\langle y_{t-1} - y^*(x_{t-1}), \nabla_y g(x_{t-1}, y_{t-1}, \xi_t) \rangle] \\
= \mathbb{E}_t[\|y_{t-1} - y^*(x_{t-1})\|^2] + \delta_t^2 \mathbb{E}_t[\|\nabla_y g(x_{t-1}, y_{t-1}, \xi_t)\|^2] \\
- 2\mathbb{E}_t[\langle y_{t-1} - y^*(x_{t-1}), \nabla_y g(x_{t-1}, y_{t-1}) \rangle],
\]

(58)
here \([58]\) comes from the fact that \(E_t[\nabla_y g(x_{t-1}, y_{t-1}, \xi_t)] = \nabla_y g(x_{t-1}, y_{t-1}).\) Next, using the strong convexity property of function \(g,\) we can upper bound the last inner product term on the right hand side of \((58)\) as

\[
E_t[\|y_t - y^*(x_{t-1})\|^2] \leq E_t[\|y_{t-1} - y^*(x_{t-1})\|^2 + \delta^2_t E_t[\|\nabla_y g(x_{t-1}, y_{t-1}, \xi_t)\|^2]
- 2\delta_t \mu_g E_t[\|y_{t-1} - y^*(x_{t-1})\|^2]
= (1 - 2\delta_t \mu_g) E_t[\|y_{t-1} - y^*(x_{t-1})\|^2] + \delta^2_t E_t[\|\nabla_y g(x_{t-1}, y_{t-1}, \xi_t)\|^2],
\]

(59)

Let us consider the last term \(E_t[\|\nabla_y g(x_{t-1}, y_{t-1}, \xi_t)\|^2]\) in the right hand side of \((60)\) as

\[
E_t[\|\nabla_y g(x_{t-1}, y_{t-1}, \xi_t)\|^2] \leq E_t[\|\nabla_y g(x_{t-1}, y_{t-1}, \xi_t) + \nabla_y g(x_{t-1}, y_{t-1}) - \nabla_y g(x_{t-1}, y_{t-1})\|^2]
\leq 2E_t[\|\nabla_y g(x_{t-1}, y_{t-1}, \xi_t) - \nabla_y g(x_{t-1}, y_{t-1})\|^2] + 2\|\nabla_y g(x_{t-1}, y_{t-1})\|^2,
\]

where we use the inequality \(|a + b|^2 \leq 2|a|^2 + 2|b|^2.\) From Assumption \([1]\), we can upper bound \((61)\) as

\[
E_t[\|\nabla_y g(x_{t-1}, y_{t-1}, \xi_t)\|^2] \leq 2\sigma_y^2 (1 + \|\nabla_y g(x_{t-1}, y_{t-1})\|^2) + 2\|\nabla_y g(x_{t-1}, y_{t-1})\|^2
= 2\sigma_y^2 + 2(1 + \sigma_y^2) \|\nabla_y g(x_{t-1}, y_{t-1})\|^2,
\]

(62)

Since \(\nabla_y g(x_{t-1}, y^*(x_{t-1})) = 0,\) we can write \((62)\) as

\[
E_t[\|\nabla_y g(x_{t-1}, y_{t-1}, \xi_t)\|^2] \leq 2\sigma_y^2 + 2(1 + \sigma_y^2) \|\nabla_y g(x_{t-1}, y_{t-1}) - \nabla_y g(x_{t-1}, y^*(x_{t-1}))\|^2
\leq 2\sigma_y^2 + 2(1 + \sigma_y^2) L_y^2 \|y_{t-1} - y^*(x_{t-1})\|^2,
\]

(63)

where we have used the Lipschitz continuity of gradients as stated in Assumption \([1]\). Substitute the upper bound in \((63)\) in place of the second term on the right hand side of \((60)\) to obtain

\[
E_t[\|y_t - y^*(x_{t-1})\|^2] \leq (1 - 2\delta_t \mu_g) E_t[\|y_{t-1} - y^*(x_{t-1})\|^2] + \delta^2_t [2\sigma_y^2 + 2(1 + \sigma_y^2) L_y^2 \|y_{t-1} - y^*(x_{t-1})\|^2]
\leq [(1 - 2\delta_t \mu_g) + 2\delta^2_t (1 + \sigma_y^2) L_y^2] E_t[\|y_{t-1} - y^*(x_{t-1})\|^2] + 2\delta^2_t \sigma_y^2
\leq (1 - \delta_t \mu_g) E_t[\|y_{t-1} - y^*(x_{t-1})\|^2] + 2\delta^2_t \sigma_y^2,
\]

(64)

The last inequality in \((64)\) is obtained by selecting \(\delta_t\) such that \(2\delta_t(1 + \sigma_y^2) L_y^2 \leq \mu_g.\) To proceed next, we use Young’s inequality to bound the term \(E_t[\|y_t - y^*(x_t)\|^2]\) in \((64)\) as

\[
E_t[\|y_t - y^*(x_t)\|^2] \leq \left(1 + \frac{1}{\alpha}\right) E_t[\|y_t - y^*(x_{t-1})\|^2] + (1 + \alpha) E_t[\|y^*(x_t) - y^*(x_{t-1})\|^2]
\leq \left(1 + \frac{1}{\alpha}\right) E_t[\|y_t - y^*(x_{t-1})\|^2] + (1 + \alpha) \left(\frac{C_{xy}}{\mu_g}\right)^2 E_t \|x_t - x_{t-1}\|^2
\leq \left(1 + \frac{1}{\alpha}\right) E_t[\|y_t - y^*(x_{t-1})\|^2] + (1 + \alpha) \left(\frac{C_{xy}}{\mu_g}\right)^2 \eta_{t-1}^2 D^2,
\]

(65)

where the second inequality in \((65)\) comes from Lemma \([6b]\), and the last inequality in \((65)\) while the last inequality comes from the update equation in \((19)\) and further utilizing the compactness of the domain \(\mathcal{X}\) (recall that \(D\) is the diameter of the set \(\mathcal{X}\)). Utilizing \((64)\) into \((65)\), we get

\[
E_t[\|y_t - y^*(x_t)\|^2] \leq \left(1 + \frac{1}{\alpha}\right) (1 - \delta_t \mu_g) E_t[\|y_{t-1} - y^*(x_{t-1})\|^2] + \left(1 + \frac{1}{\alpha}\right) 2\delta^2_t \sigma_y^2
+ (1 + \alpha) \left(\frac{C_{xy}}{\mu_g}\right)^2 \eta_{t-1}^2 D^2.
\]

(66)
To proceed next, we substitute $\alpha = \frac{2(1-\delta_t \mu_g)}{\delta_t \mu_g}$ which also implies that $(1 + \frac{1}{\alpha})(1 - \delta_t \mu_g) = 1 - \frac{\mu_g \delta_t}{2}$, we obtain from (66)

$$
\mathbb{E}_t[\|y_t - y^*(x_t)\|^2] \leq \left(1 - \frac{\delta_t \mu_g}{2}\right) \mathbb{E}_t[\|y_{t-1} - y^*(x_{t-1})\|^2] + 2 - \frac{\delta_t \mu_g}{\delta_t \mu_g} \left(\frac{C_{xy}}{\mu_g}\right)^2 \eta_{t-1}^2 D^2 + \left(1 + \frac{1}{\alpha}\right) 2\delta_t^2 \sigma_g^2
$$

$$
\leq \left(1 - \frac{\delta_t \mu_g}{2}\right) \mathbb{E}_t[\|y_{t-1} - y^*(x_{t-1})\|^2] + \frac{2\eta_{t-1}^2}{\delta_t \mu_g} \left(\frac{C_{xy}}{\mu_g}\right)^2 D^2 + \left(1 + \frac{1}{\alpha}\right) 2\delta_t^2 \sigma_g^2
$$

$$
\leq \left(1 - \frac{\delta_t \mu_g}{2}\right) \mathbb{E}_t[\|y_{t-1} - y^*(x_{t-1})\|^2] + \frac{2\eta_{t-1}^2}{\delta_t \mu_g} \left(\frac{C_{xy}}{\mu_g}\right)^2 D^2 + 4\delta_t^2 \sigma_g^2, \quad (67)
$$

where the second inequality comes from the fact that $\frac{2 - \delta_t \mu_g}{\delta_t \mu_g} < \frac{2}{\delta_t \mu_g}$ while in the last inequality, we have assumed that $\delta_t$ is chosen such that $\delta_t \leq \frac{2}{\delta_t \mu_g}$ giving $1 + \frac{1}{\alpha} \leq 2$. In Corollary 1 we will see that our choice of step sizes satisfies these conditions.

B. Proof of Lemma 3

Starting with update equation (17) and employing Lemma 9 we can write

$$
\mathbb{E}_t[\|d_t - \nabla S(x_t, y_t) - B_t\|^2 \leq (1 - \rho_t)^2 \mathbb{E}_t[\|d_{t-1} - \nabla S(x_{t-1}, y_{t-1}) - B_{t-1}\|]
$$

$$
+ 2(1 - \rho_t)^2 \mathbb{E}_t[\|h(x_t, y_t; \theta_t, \xi_t) - h(x_{t-1}, y_{t-1}; \theta_t, \xi_t)\|^2]
$$

$$
+ 2\rho_t^2 \mathbb{E}_t[\|h(x_t, y_t; \theta_t, \xi_t) - \nabla S(x_t, y_t) - B_t\|^2]
$$

$$
\leq (1 - \rho_t)^2 \mathbb{E}_t[\|d_{t-1} - \nabla S(x_{t-1}, y_{t-1}) - B_{t-1}\|]
$$

$$
+ 2\mathbb{E}_t[\|h(x_t, y_t; \theta_t, \xi_t) - h(x_{t-1}, y_{t-1}; \theta_t, \xi_t)\|^2 + 2\rho_t^2 \sigma_f^2], \quad (68)
$$

here the last inequality is obtained using (24) and the fact that $(1 - \rho_t)^2 \leq 1$. Now we introduce $h(x_t, y_{t-1}; \theta_t, \xi_t)$ and bound the second term of RHS of (68) as

$$
\mathbb{E}_t[\|h(x_t, y_t; \theta_t, \xi_t) - h(x_{t-1}, y_{t-1}; \theta_t, \xi_t)\|^2]
$$

$$
= \mathbb{E}_t[\|h(x_t, y_t; \theta_t, \xi_t) - h(x_t, y_t; \theta_t, \xi_t) + h(x_t, y_t; \theta_t, \xi_t) - h(x_{t-1}, y_{t-1}; \theta_t, \xi_t)\|^2]
$$

$$
\leq (a) 2\mathbb{E}_t[\|h(x_t, y_t; \theta_t, \xi_t) - h(x_t, y_t; \theta_t, \xi_t)\|^2 + 2\mathbb{E}_t[\|h(x_t, y_t; \theta_t, \xi_t) - h(x_{t-1}, y_{t-1}; \theta_t, \xi_t)\|^2]
$$

$$
\leq (b) 2L_k\delta_t^2 \mathbb{E}_t[\|y_t - y_{t-1}\|^2 + 2L_k \mathbb{E}_t[\|x_{t-1} - x_t\|^2]
$$

$$
\leq (c) 2L_k\delta_t^2 \mathbb{E}_t[\|
abla y g(x_{t-1}, y_{t-1}, \xi_t)\|^2 + 2L_k \eta_{t-1}^2 D^2
$$

$$
\leq (d) 2L_k\delta_t^2 L_g^2 + 2L_k \eta_{t-1}^2 D^2, \quad (69)
$$

here (a) comes from simple norm property, (b) comes from Lemma 7 (c) comes from update equation (12) while (d) comes from Lipschitz continuous Assumption 1(1) and from the compactness of the set. Using (69) in (68), we get the desired expression.
C. Proof of Corollary \[7\]

For the simplicity of analysis we start with writing Lemma \[2\] for \( t = t + 1 \) and set \( \delta_t = \frac{2a_0}{\sqrt{t}} \) where \( a_0 = \min\{\frac{1}{\nu_y}, \frac{\mu_\rho}{2(1+\sigma_g^2)L_g}\} \) and \( \eta_t = \frac{2}{(t+1)^{\frac{3}{2}}} \), which gives

\[
\mathbb{E}_t[\|y_{t+1} - y^*(x_{t+1})\|^2] \leq \left( 1 - \frac{2a_0}{(t+1)^{\frac{3}{2}}} \right) \mathbb{E}_t[\|y_t - y^*(x_t)\|^2] + \frac{2}{(t+1)^{\frac{3}{2}}} \left( \frac{C_{y_2}}{\mu_g} \right)^2 D^2 + \frac{16a_0^2}{(t+1)^{2q}} \sigma_g^2
\]

\[
= \left( 1 - \frac{2a_0}{(t+1)^{\frac{3}{2}}} \right) \mathbb{E}_t[\|y_t - y^*(x_t)\|^2] + \frac{2(C_{y_2}/\mu_g)^2 D^2 + 16a_0^2 \sigma_g^2}{(t+1)^{2q}}. \tag{70}
\]

Note such selection of \( \delta_t \) ensures that the conditions \( 2\delta_t(1 + \sigma_g^2)L_g^2 \leq \mu_g \) and \( \delta_t \leq \frac{2}{3\nu_y} \) required in Lemma \[2\] are satisfied. Now taking full expectation and using Lemma \[8\] we get

\[
\mathbb{E}[\|y_t - y^*(x_t)\|^2] \leq \frac{b_1}{(t+1)^{q}}, \tag{71}
\]

where \( b_1 = \max\{2^q \|y_1 - y^*(x_1)\|^2, (2(C_{y_2}/\mu_g)^2 D^2 + 16a_0^2 \sigma_g^2)/(2a_0 - 1)\} \). Similarly, in Lemma \[3\] setting \( \delta_t = \frac{2a_0}{(t)^{\frac{3}{2}}}, \eta_t = \frac{2}{(t+1)^{\frac{3}{2}}} \) and \( \rho_t = \frac{2}{(t)^{\frac{3}{2}}} \), we can write

\[
\mathbb{E}_t[\|d_{t+1} - \nabla S(x_{t+1}, y_{t+1}) - B_{t+1}\|^2] \leq \left( 1 - \frac{2}{(t+1)^{\frac{3}{2}}} \right) \mathbb{E}_t[\|d_t - \nabla S(x_t, y_t) - B_t\|^2] + \frac{16L_k L_g^2}{(t+1)^{2q}} + \frac{16L_k D^2}{(t+1)^{3q}} + \frac{8\sigma_j^2}{(t+1)^{2q}}
\]

\[
\leq \left( 1 - \frac{2}{(t+1)^{\frac{3}{2}}} \right) \mathbb{E}_t[\|d_t - \nabla S(x_t, y_t) - B_t\|^2] + \frac{16L_k L_g^2 + 16L_k D^2 + 8\sigma_j^2}{(t+1)^{2q}}, \tag{72}
\]

here the last inequality is obtained using the fact \( 1/(t+1)^{2q} \leq 1/(t+1)^{2p} \). Application of Lemma \[8\] gives

\[
\mathbb{E}_t[\|d_t - \nabla S(x_t, y_t) - B_t\|^2] \leq \frac{b_2}{(t+2)^{q}}, \tag{73}
\]

where \( b_2 = \max\{2^q \|d_1 - \nabla S(x_1, y_1) - B_1\|^2, 8(2L_k L_g^2 + L_k D^2 + \sigma_j^2)\} = 8(2L_k L_g^2 + L_k D^2 + \sigma_j^2) \). As, we have initialize \( d_1 = h(x_1, y_1; \theta_1, \xi_1) \) we can use the bound \( \|d_1 - \nabla S(x_1, y_1) - B_1\|^2 = \|h(x_1, y_1; \theta_1, \xi_1) - \nabla S(x_1, y_1) - B_1\|^2 \leq \sigma_j^2 \). Further using the fact that \( 0 < q \leq 1 \), we can simplify \( b_2 \) as \( b_2 = 8(2L_k L_g^2 + L_k D^2 + \sigma_j^2) \).

Now we can bound the term \( \mathbb{E}[\|\nabla Q(x_t) - d_t\|^2] \) as follows

\[
\mathbb{E}[\|\nabla Q(x_t) - d_t\|^2] = \mathbb{E}[\|\nabla Q(x_t) - d_t + B_t + \nabla S(x_t, y_t) - B_t - \nabla S(x_t, y_t)\|^2]
\]

\[
\leq 3\mathbb{E}[\|\nabla Q(x_t) - \nabla S(x_t, y_t)\|^2] + 3\|B_t\|^2 + 3\mathbb{E}[\|\nabla S(x_t, y_t) + B_t - d_t\|^2]
\]

\[
\leq 3\mathbb{E}[\|y^*(x_t) - y_t\|^2] + 3\beta_t^2 + \frac{3b_2}{(t+1)^q}
\]

\[
\leq \frac{3b_1}{(t+1)^q} + \frac{3b_2}{(t+1)^q} + \frac{3b_2}{(t+1)^q} := \frac{C_1}{(t+1)^q}, \tag{75}
\]

here second inequality comes from simple norm property, while third inequality is obtained using Lemma \[6\] on the first term, Lemma \[1\] on the second term and \( \|B_t\|^2 \) on the third term. The last inequality comes from \( \|B_t\|^2 \) and using \( \beta_t \leq \frac{C_{xy}C_y}{\mu_\rho(t+1)^q} := \frac{b_3}{(t+1)^q} \) and the constant \( C_1 = 3(b_1 + b_2 + b_3) \) is defined as

\[
C_1 = 3(\max\{2^q \|y_1 - y^*(x_1)\|^2, (2(C_{xy}/\mu_g)^2 D^2 + 16a_0^2 \sigma_g^2)/(2a_0 - 1)\} + 8(2L_k L_g^2 + L_k D^2 + \sigma_j^2) + \frac{C_{xy}C_y}{\mu_\rho}).
\]
D. Proof of Theorem 7

From the initialization of variable $x$, we have $x_1 \in X$. Also since we obtain $s_t$ solving a linear minimization problem over the set $X$, we have $s_t \in X$. Thus, $x_{t+1}$ which is a convex combination of $x_t$ and $s_t$, i.e. $x_{t+1} = (1 - \eta_t) x_t + \eta_t s_t$ will also lie in the set $X$. Hence $x_{T+1} \in X$ and $\hat{x} \in X$. Now, starting with definition of $Q(\cdot)$, we have $Q(x) = \mathbb{E}_\theta[f(x, y^*(x); \theta)]$. Also note that we have set $k = \frac{2L_x}{\mu_y} (\log(1 + t))$, this ensures that the condition $\beta_t \leq \frac{C_{xy} C_{y}}{\mu_y (t+1)}$ required in the analysis of Corollary I is satisfied. Hence, we can use results from Corollary I with $q = 2/3$ for convex case and $q = 1/2$ for non-convex case.

1) Proof of Statement (i) (Convex case): Using the smoothness assumption of $Q$ we can write

$$Q(x_{t+1}) \leq Q(x_t) + \langle \nabla Q(x_t), x_{t+1} - x_t \rangle + \frac{L_Q}{2} \|x_{t+1} - x_t\|^2$$

$$= Q(x_t) + \eta_t \langle \nabla Q(x_t), s_t - x_t \rangle + \frac{L_Q \eta_t^2}{2} \|s_t - x_t\|^2,$$

where $L_Q = \frac{(L_{x} + L) C_{xy}}{\mu_y} + L_{f_x} + C_y \frac{L_{axy} C_{xy}}{\mu_y} + \frac{L_{axy} C_{xy}}{\mu_y}$ (see Lemma 3). Here, in the last expression we have replace term $x_{t+1} - x_t = \eta_t (s_t - x_t)$. Now adding and subtracting $\eta_t \langle d_t, s_t - x_t \rangle$ in (76) we get

$$Q(x_{t+1}) \leq Q(x_t) + \eta_t \langle \nabla Q(x_t) - d_t, s_t - x_t \rangle + \eta_t \langle d_t, s_t - x_t \rangle + \frac{L_Q \eta_t^2}{2} \|s_t - x_t\|^2$$

$$\leq Q(x_t) + \eta_t \langle \nabla Q(x_t) - d_t, s_t - x_t \rangle + \eta_t \langle d_t, x^* - x_t \rangle + \frac{L_Q \eta_t^2 D^2}{2},$$

Here in last the inequality we used optimum condition that is $s_t = \arg \min_{s_t \in X} \langle d_t, s \rangle$. Now introducing $\eta_t \langle \nabla Q(x_t), x^* - x_t \rangle$ in RHS of (77) and regrouping the terms we obtain

$$Q(x_{t+1}) \leq Q(x_t) + \eta_t D \|\nabla Q(x_t) - d_t\| + \frac{L_Q \eta_t^2 D^2}{2} + \eta_t \langle \nabla Q(x_t), x^* - x_t \rangle$$

$$\leq Q(x_t) + \eta_t D \|\nabla Q(x_t) - d_t\| + \frac{L_Q \eta_t^2 D^2}{2} - \eta_t \langle Q(x_t) - Q(x^*) \rangle,$$

here in the second inequality we use bound $\eta_t \langle \nabla Q(x_t) - d_t, s_t - x^* \rangle \leq \eta_t \|\nabla Q(x_t) - d_t\| \|s_t - x^*\| \leq \eta_t D \|\nabla Q(x_t) - d_t\|$ and in last inequality we used the bound $\langle \nabla Q(x_t), x^* - x_t \rangle \leq Q(x^*) - Q(x_t)$ from convexity assumption of $Q$. Now subtracting $Q(x^*)$ gives

$$Q(x_{t+1}) - Q(x^*) \leq (1 - \eta_t)(Q(x_t) - Q(x^*)) + \eta_t D \|\nabla Q(x_t) - d_t\| + \frac{L_Q \eta_t^2 D^2}{2}. (79)$$

Taking expectation on (79) and using the inequality $E \|X\| \leq \sqrt{E \|X\|^2}$ we can write

$$E[Q(x_{t+1}) - Q(x^*)] \leq (1 - \eta_t)E[Q(x_t) - Q(x^*)] + \eta_t D \sqrt{E \|\nabla Q(x_t) - d_t\|^2} + \frac{L_Q \eta_t^2 D^2}{2}.$$ (80)

Setting $q = 2/3$ hence, $\eta_t = \frac{2}{t+1}$ and using Corollary I we can bound the second term of (80) as

$$\eta_t D \sqrt{E \|\nabla Q(x_t) - d_t\|^2} \leq \frac{2D \sqrt{C_1}}{(t+1)^{4/3}}. (81)$$

Now using bound from (81) in (80) we obtain

$$E[Q(x_{t+1}) - Q(x^*)] \leq \left(1 - \frac{2}{t+1}\right)E[Q(x_t) - Q(x^*)] + \frac{2D \sqrt{C_1}}{(t+1)^{4/3}} + \frac{2L_Q D^2}{(t+1)^2}. (82)$$
Multiplying both side by \( t(t+1) \) we can write
\[
t(t+1)\mathbb{E}[Q(x_{t+1}) - Q(x^*)] \leq t(t-1)\mathbb{E}[Q(x_t) - Q(x^*)] + 2D\sqrt{C_1} \frac{t}{(t+1)^{3}} + 2L_QD^2 \frac{t}{t+1}
\]
\[
\leq t(t-1)\mathbb{E}[Q(x_t) - Q(x^*)] + 2D\sqrt{C_1}(t+1)^{\frac{3}{2}} + 2L_QD^2, \tag{83}
\]
here in last the inequality we used the fact that \( t < t+1 \). Summing for \( t = 1, 2, \ldots, T \) and rearranging we get
\[
\mathbb{E}[Q(x_{T+1}) - Q(x^*)] \leq \frac{1}{T(T+1)} \left( 2D\sqrt{C_1} \sum_{t=1}^{T} (t+1)^{\frac{3}{2}} + 2L_QD^2T \right)
\]
\[
\leq \frac{1}{T(T+1)} \left( \frac{6}{5} D\sqrt{C_1}(T+1)^{\frac{3}{2}} + 2L_QD^2T \right)
\]
\[
eq \frac{(T+1)}{T(T+1)^{2}} \left( \frac{6}{5} D\sqrt{C_1}(T+1)^{\frac{3}{2}} + 2L_QD^2T \right)
\]
\[
\leq \frac{12D\sqrt{C_1}}{5(T+1)^{\frac{3}{2}}} + \frac{2L_QD^2}{(T+1)}, \tag{84}
\]
here in the second inequality we use the fact that \( \sum_{t=1}^{T} (t+1)^{2/3} \leq \frac{3}{5}(T+1)^{5/3} \) and the last inequality is obtained using \( (T+1) = T(1 + \frac{1}{T}) \leq 2T \).

2) Proof of Statement (ii) (Non-convex case): Again starting with the smoothness assumption of \( Q \) we can write
\[
Q(x_{t+1}) \leq Q(x_t) + \langle \nabla Q(x_t), x_{t+1} - x_t \rangle + \frac{L_Q}{2} \|x_{t+1} - x_t\|^2
\]
\[
= Q(x_t) + \eta_t \langle \nabla Q(x_t), s_t - x_t \rangle + \frac{L_Q\eta_t^2}{2} \|s_t - x_t\|^2
\]
\[
\leq Q(x_t) + \eta_t \langle \nabla Q(x_t), s_t - x_t \rangle + \frac{L_Q\eta_t^2D^2}{2}
\]
\[
= Q(x_t) + \eta_t \langle \nabla Q(x_t) - d_t, s_t - x_t \rangle + \eta_t \langle d_t, s_t - x_t \rangle + \frac{L_Q\eta_t^2D^2}{2}, \tag{85}
\]
where in the second expression we have replace term \( x_{t+1} - x_t = \eta_t(s_t - x_t) \) and in third inequality we used compactness assumption of set while in the last expression we introduced \( \eta_t \langle d_t, s_t - x_t \rangle \). Next, we introduce the following quantity
\[
\hat{\nu}_t = \arg \max_{v \in X} \langle v - x_t, -\nabla Q(x_t) \rangle, \tag{86}
\]
and using the optimality of \( s_t \) that is \( s_t = \arg \min_{s \in X} \langle d_t, s \rangle \), we write inequality in (85) as
\[
Q(x_{t+1}) \leq Q(x_t) + \eta_t \langle \nabla Q(x_t) - d_t, s_t - x_t \rangle + \eta_t \langle d_t, \hat{\nu}_t - x_t \rangle + \frac{L_Q\eta_t^2D^2}{2}
\]
\[
= -\eta_t G(x_t) + Q(x_t) + \eta_t \langle \nabla Q(x_t) - d_t, s_t - x_t \rangle + \eta_t \langle d_t - \nabla Q(x_t), \hat{\nu}_t - x_t \rangle + \frac{L_Q\eta_t^2D^2}{2}
\]
\[
\leq -\eta_t G(x_t) + Q(x_t) + 2\eta_t D \|d_t - \nabla Q(x_t)\| + \frac{L_Q\eta_t^2D^2}{2}, \tag{87}
\]
where the second expression is obtained by adding and subtracting the term \( \eta_t \langle \hat{\nu}_t - x_t, -\nabla Q(x_t) \rangle \) while the last inequality is obtained using compactness assumption.
Rearranging (87), summing for \( t = 1, 2, \cdots, T \) and taking expectation, we get
\[
\sum_{t=1}^{T} \eta_t E[G(x_t)] \leq Q(x_1) - E[Q(x_{T+1})] + 2D \sum_{t=1}^{T} \eta_t E[\|d_t - \nabla Q(x_t)\|] + \frac{L_Q D^2}{2} \sum_{t=1}^{T} \eta_t^2
\]
\[
\leq Q(x_1) - Q(x^*) + 2D \sum_{t=1}^{T} \eta_t E[\|d_t - \nabla Q(x_t)\|] + \frac{L_Q D^2}{2} \sum_{t=1}^{T} \eta_t^2,
\]
here the last inequality follows from optimality of \( x^* \). Using bound from Corollary 1 for \( q = 1/2 \), and Jensen’s equality we can write
\[
E[\|d_t - \nabla Q(x_t)\|] \leq \sqrt{E[\|d_t - \nabla Q(x_t)\|^2]} \leq \frac{\sqrt{C_1}}{(t+1)^{\frac{1}{4}}},
\]
Now, using (89), setting \( \eta_t = \frac{2}{(T+1)^{3/4}} \) in (88) and rearranging we can write
\[
E[G(\hat{x})] \leq \frac{1}{T} \sum_{t=1}^{T} E[G(x_t)] \leq \frac{Q(x_1) - Q(x^*)}{2T(T+1)^{-3/4}} + \frac{2D\sqrt{C_1}}{T(T+1)^{1/4}} + \frac{L_Q D^2}{T(T+1)^{3/4}}
\]
\[
\leq \frac{Q(x_1) - Q(x^*)}{(T+1)^{1/4}} + \frac{8D\sqrt{C_1}}{3T(T+1)^{1/4}} + \frac{L_Q D^2}{(T+1)^{3/4}}
\]
\[
\leq \frac{Q(x_1) - Q(x^*)}{(T+1)^{1/4}} + \frac{16D\sqrt{C_1}}{3(T+1)^{1/4}} + \frac{L_Q D^2}{(T+1)^{3/4}} := O((T+1)^{-1/4}),
\]
here in the second inequality we use the fact that \( \sum_{t=1}^{T} (t+1)^{-1/4} \leq \frac{4}{3}(T+1)^{3/4} \) and the last inequality is obtained using \( (T+1) = T(1 + \frac{1}{T}) \leq 2T \).

IX. PROOFS FOR SCFW

Before proceeding we introduce some shorthand notations for simplicity. We will use \( f_{\theta_t}(y_t) := f(y_t, \theta_t) \), \( g_{\xi_t}(x_t) := g(x_t, \xi_t), f(y_t) = E_{\theta_t}[f(y_t, \theta_t)], \) and \( g(x_t) = E_{\xi_t}[g(x_t, \xi_t)] \). Also for brevity we will use \( \nabla C_{\phi_t}(x_t, y_t) \) to represent \( \nabla C(x_t, y_t, \xi_t, \theta_t) = \nabla h_{\xi_t}(x_t)^{\top}\nabla f_{\theta_t}(y_t) \) and \( \nabla C(x_t, y_t) = E[\nabla C_{\phi_t}(x_t, y_t)] \).

A. Proof of Lemma 4

Setting \( \Psi(\cdot) = h(\cdot) \) in Lemma (9), we can write
\[
E_t[\|y_t - h(x_t)\|^2]
\]
\[
\leq (1 - \delta_t)^2 \|y_{t-1} - h(x_{t-1})\|^2 + 2(1 - \delta_t)^2 E_t[\|h_{\xi_t}(x_t) - h_{\xi_t}(x_{t-1})\|^2] + 2\delta_t^2 E_t[\|h_{\xi_t}(x_t) - h(x_t)\|^2].
\]
Now, taking total expectation of (91) we get
\[
E[\|y_t - h(x_t)\|^2] \leq (1 - \delta_t)^2 E[\|y_{t-1} - h(x_{t-1})\|^2] + 2\delta_t^2 \sigma_h^2 + 2(1 - \delta_t)^2 E[\|h_{\xi_t}(x_t) - h_{\xi_t}(x_{t-1})\|^2]
\]
\[
\leq (1 - \delta_t)^2 E[\|y_{t-1} - h(x_{t-1})\|^2] + 2\delta_t^2 \sigma_h^2 + 2(1 - \delta_t)^2 M_h E[\|x_t - x_{t-1}\|^2]
\]
\[
= (1 - \delta_t)^2 E[\|y_{t-1} - h(x_{t-1})\|^2] + 2\delta_t^2 \sigma_h^2 + 2(1 - \delta_t)^2 M_h \eta_{t-1} E[\|s_{t-1} - x_{t-1}\|^2]
\]
\[
\leq (1 - \delta_t)^2 E[\|y_{t-1} - h(x_{t-1})\|^2] + 2\delta_t^2 \sigma_h^2 + 2(1 - \delta_t)^2 \eta_{t-1}^2 M_h D^2,
\]
here in the first inequality we have used Assumption 2(iii). In the second inequality we have used Assumption 2(iv)
while the next expression follows from the update step \( x_{t+1} = (1 - \eta_t)x_t + \eta_ts_t \). The last inequality obtained using compactness assumption of set \( \mathcal{X} \).

**B. Proof of Lemma 3**

Setting \( \Psi = \tilde{\nabla}C(x_t, y_t) \) in Lemma (9), we can write

\[
\mathbb{E}_t[||d_t - \tilde{\nabla}C(x_t, y_t)||^2]
= (1 - \rho_t)^2||d_{t-1} - \tilde{\nabla}C(x_{t-1}, y_{t-1})||^2
+ 2(1 - \rho_t)^2\mathbb{E}_t[||\nabla C_{\phi_t}(x_t, y_t) - \nabla C_{\phi_t}(x_{t-1}, y_{t-1})||^2] + 2\rho_t^2\mathbb{E}_t[||\nabla C_{\phi_t}(x_t, y_t) - \tilde{\nabla}C(x_t, y_t)||^2],
\]

We can bound the second term of (93) by taking full expectation and using Assumption 2 as follows

\[
2\rho_t^2\mathbb{E}_t[||\nabla C_{\phi_t}(x_t, y_t) - \nabla C_{\phi_t}(x_{t-1}, y_{t-1})||^2]
= 2\rho_t^2\mathbb{E}_t[||\nabla h_{\xi_t}(x_t)^\top \nabla f_{\theta}(y_t) - \nabla h_{\xi_t}(x_t)^\top \nabla f(y_t)||^2]
\leq 2\rho_t^2(2\mathbb{E}_t[||\nabla h_{\xi_t}(x_t)||^2] \mathbb{E}_t[||\nabla f_{\theta}(y_t)||^2] + 2\mathbb{E}_t[||\nabla h_{\xi_t}(x_t)||^2 \mathbb{E}_t[||\nabla f(y_t)||^2])
\leq 4\rho_t^2M_fM_h.
\]

We can bound the first term of (93) by introducing \( \nabla h_{\xi_t}(x_{t-1})^\top \nabla f_{\theta}(y_t) \) as follows

\[
\mathbb{E}_t[||\nabla C_{\phi_t}(x_t, y_t) - \nabla C_{\phi_t}(x_{t-1}, y_{t-1})||^2]
= \mathbb{E}_t[||\nabla h_{\xi_t}(x_t)^\top \nabla f_{\theta_t}(y_t) - \nabla h_{\xi_t}(x_t)^\top \nabla f_{\theta_t}(y_{t-1})||^2]
= \mathbb{E}_t[||\nabla h_{\xi_t}(x_t)^\top \nabla f_{\theta_t}(y_t) - \nabla h_{\xi_t}(x_{t-1})^\top \nabla f_{\theta_t}(y_{t-1})||^2]
+ \mathbb{E}_t[||\nabla h_{\xi_t}(x_{t-1})^\top \nabla f_{\theta_t}(y_{t-1})||^2]
= \mathbb{E}_t[||\nabla f_{\theta_t}(y_t)^\top (\nabla h_{\xi_t}(x_t) - \nabla h_{\xi_t}(x_{t-1})) + \nabla h_{\xi_t}(x_{t-1})^\top (\nabla f_{\theta_t}(y_t) - \nabla f_{\theta_t}(y_{t-1}))||^2]
\leq 2\mathbb{E}_t[||\nabla f_{\theta_t}(y_t)||^2] \mathbb{E}_t[||\nabla h_{\xi_t}(x_t) - \nabla h_{\xi_t}(x_{t-1})||^2]
+ 2\mathbb{E}_t[||\nabla h_{\xi_t}(x_{t-1})||^2] \mathbb{E}_t[||\nabla f_{\theta_t}(y_{t-1}) - \nabla f_{\theta_t}(y_{t-1})||^2]
\leq 2M_fL_h\mathbb{E}_t[||x_t - x_{t-1}||^2] + 2M_hL_f\mathbb{E}_t[||y_t - y_{t-1}||^2]
\leq 2M_fL_h\eta_{t-1}^2D^2 + 2M_hL_f\mathbb{E}_t[||y_t - y_{t-1}||^2].
\]

Further, we can bound second term of (96) using the update equation for \( y_{t+1} \) from Algorithm:

\[
y_t - y_{t-1} = -\delta_t y_{t-1} - (1 - \delta_t)h_{\xi_t}(x_{t-1}) + h_{\xi_t}(x_t)
= \delta_t(h(x_{t-1}) - y_{t-1}) + \delta_t(h_{\xi_t}(x_{t-1}) - h(x_{t-1})) + h_{\xi_t}(x_t) - h_{\xi_t}(x_{t-1}).
\]

Taking norm square of (97) we get

\[
\mathbb{E}_t[||y_t - y_{t-1}||^2] \leq 3\delta_t^2\mathbb{E}_t[||y_{t-1} - h(x_{t-1})||^2] + 3\delta_t^2\sigma_h^2 + 3M_h\mathbb{E}_t[||x_t - x_{t-1}||^2]
\leq 3\delta_t^2\mathbb{E}_t[||y_{t-1} - h(x_{t-1})||^2] + 3\delta_t^2\sigma_h^2 + 3M_h\eta_{t-1}^2D^2.
\]

October 25, 2021 DRAFT
Substituting (98) in (96) we get
\[
\mathbb{E}_t\left[\left\|\nabla C_{\psi_t}(x_{t-1}, y_{t-1}) - \nabla C_{\psi_t}(x_{t-1}, y_t)\right\|^2\right]
\leq 2M_f L_f \eta_{t-1}^2 D^2 + 2M_h L_f \left(3 \delta_t^2 \mathbb{E}_t\left[\left\|y_{t-1} - h(x_{t-1})\right\|^2\right] + 3 \delta_t^2 \sigma_h^2 + 3M_h \eta_{t-1}^2 D^2\right)
= 2(M_f L_h + 3M_h^2 L_f) \eta_{t-1}^2 D^2 + 6M_h L_f \delta_t^2 \mathbb{E}_t\left[\left\|y_{t-1} - h(x_{t-1})\right\|^2\right] + 6 \delta_t^2 C M_h L_f \sigma_h^2.
\] (99)

Thus using bound from (95) and (99) into (93) and taking full expectation we obtain
\[
\mathbb{E}\left\|d_t - \nabla C(x_t, y_t)\right\|^2
\leq (1 - \rho_t)^2 \mathbb{E}\left\|d_{t-1} - \nabla C(x_{t-1}, y_{t-1})\right\|^2 + 4 \rho_t^2 M_f M_h
+ 4(1 - \rho_t)^2 \left[(M_f L_h + 3M_h^2 L_f) \eta_{t-1}^2 D^2 + 3M_h L_f \delta_t^2 \mathbb{E}_t\left[\left\|y_{t-1} - h(x_{t-1})\right\|^2\right] + 3 \delta_t^2 M_h L_f \sigma_h^2\right].
\] (100)

C. Proof of Corollary 2

Starting with Lemma 4 and using the fact that \((1 - \delta_t)^2 \leq (1 - \delta_t) \leq 1\), we can write
\[
\mathbb{E}\left\|y_{t+1} - h(x_{t+1})\right\|^2 \leq (1 - \delta_t) \mathbb{E}\left\|y_t - h(x_t)\right\|^2 + 2 \delta_t \sigma_h^2 + 2 \eta_t^2 M_h D^2.
\] (101)

Substituting \(\delta_t = \frac{2}{\eta^2}\) and \(\eta_t \leq \left(\frac{2}{t}\right)^{\frac{p}{2}}\), we get
\[
\mathbb{E}\left\|y_{t+1} - h(x_{t+1})\right\|^2 \leq \left(1 - \frac{2}{(t+1)^p}\right) \mathbb{E}\left\|y_t - h(x_t)\right\|^2 + \frac{8 \delta_t \sigma_h^2}{(t+1)^{2p}} + \frac{8 M_h D^2}{(t+1)^{2p}}.
\] (102)

Application of Lemma 8 in (102) gives
\[
\mathbb{E}\left\|y_t - h(x_t)\right\|^2 \leq \frac{a_1}{(t+1)^p},
\] (103)
where \(a_1 = \max\{2^p \mathbb{E}\left\|y_1 - h(x_1)\right\|^2, 8(\sigma_h^2 + M_h D^2)\}\). As we have initialized \(y_1 = g_{\zeta_1}(x_1)\) we can bound
\[
\mathbb{E}\left\|y_1 - h(x_1)\right\|^2 = \mathbb{E}\left\|g_{\zeta_1}(x_1) - h(x_1)\right\|^2 \leq \sigma_h^2.
\]
Further, using the fact that \(0 < p \leq 1\), we can simplify \(a_1\) as \(a_1 = 8(\sigma_h^2 + M_h D^2)\). Similarly, from Lemma 5 using the fact \((1 - \rho_t)^2 \leq 1\), we can write
\[
\mathbb{E}\left\|d_{t+1} - \nabla C(x_{t+1}, y_{t+1})\right\|^2
\leq (1 - \rho_{t+1})^2 \mathbb{E}\left\|d_t - \nabla C(x_t, y_t)\right\|^2 + 4 \rho_{t+1}^2 M_f M_h
+ 4 \left[(M_f L_h + 3M_h^2 L_f) \eta_{t+1}^2 D^2 + 3M_h L_f \delta_{t+1}^2 \mathbb{E}_t\left[\left\|y_t - h(x_t)\right\|^2\right] + 3 \delta_{t+1}^2 M_h L_f \sigma_h^2\right].
\] (104)

Substituting \(\delta_t = \frac{2}{\eta^2}\), \(\rho_t = \frac{2}{(t+1)^p}\), and \(\eta_t \leq \left(\frac{2}{t}\right)^{\frac{p}{2}}\), and using bound (103) we get
\[
\mathbb{E}\left\|d_{t+1} - \nabla C(x_{t+1}, y_{t+1})\right\|^2
\leq \left(1 - \frac{2}{(t+1)^p}\right) \mathbb{E}\left\|d_t - \nabla C(x_t, y_t)\right\|^2
+ 16 \left[\frac{M_f M_h}{(t+1)^{2p}} + \left(\frac{M_f L_h + 3M_h^2 L_f}{(t+1)^{2p}}\right) \left(\frac{a_1}{(t+1)^p}\right) + \frac{3M_h L_f \sigma_h^2}{(t+1)^{2p}}\right].
\] (105)

Application of Lemma 8 and the fact that \(\frac{1}{(t+1)^p} \leq \frac{1}{(t+1)^p}\) in (105) gives
\[
\mathbb{E}\left\|d_t - \nabla C(x_t, y_t)\right\|^2 \leq \frac{a_2}{(t+1)^p},
\] (106)
where \( a_2 := \max\{2p\|E\|d_1 - \nabla C(x_1, y_1)\|^2, 16(MfMh + (MfLh + 3M_f^2L_f)D^2 + 3M_hL_fa_1 + 3M_hL_f\sigma_h^2)\} \). As we have initialized \( d_1 = \nabla C_{\phi_1}(x_1, y_1) \), we can bound \( E\|d_1 - \nabla C(x_1, y_1)\|^2 = E\|\nabla C_{\phi_1}(x_1, y_1) - \nabla C(x_1, y_1)\|^2 \leq 2M_hM_f \) (see [23]). Also as \( 0 < p \leq 1 \), we can further simplify \( a_2 \) to get \( a_2 = 16(MfM_h + (MfL_h + 3M_f^2L_f)D^2 + 3M_hL_fa_1 + 3M_hL_f\sigma_h^2) \).

Now, introducing \( \nabla C(x_t, y_t) \) in \( E\|\nabla C(x_t) - d_t\|^2 \) we can write

\[
E\|\nabla C(x_t) - d_t\|^2 = E\|\nabla C(x_t) - d_t - \nabla C(x_t, y_t) + \nabla C(x_t, y_t)\|^2
\leq 2E\|d_t - \nabla C(x_t, y_t)\|^2 + 2E\|\nabla C(x_t, y_t) - \nabla C(x_t)\|^2
\leq 2E\|d_t - \nabla C(x_t, y_t)\|^2 + 2E\|\nabla h(x_t)\|^2 \nabla f(y_t) - \nabla h(x_t)\|^2
\leq 2E\|d_t - \nabla C(x_t, y_t)\|^2 + 2M_hL_fE\|y_t - h(x_t)\|^2
\leq \frac{2a_2}{(t+1)^p} + 2M_hL_f\sigma_h^2 := \frac{A_1}{(t+1)^p},
\]

(107)

where the first inequality comes from simple norm property while in the next expression to it we simply substitute the values of gradients. In the second inequality, we used the Assumption \( 1-2 \) while the last inequality is obtained using bounds from \( 103 \) and \( 106 \) and defined \( A_1 := 32[MfM_h + 28L_f\sigma_h^2] + (MfL_h + 28M_f^2L_f)D^2] \).

**D. Proof of Theorem 2**

1) **Proof of Statement (i) (convex case):** Starting with the smoothness of \( C \) with parameter \( L_F = M_h^2L_f + M_fL_h \) and proceeding the same way as (76)-(80) we can write

\[
E[C(x_{t+1}) - C(x*)] \leq (1 - \eta_t)E[C(x_t) - C(x*)] + \eta_tD\sqrt{E\|\nabla C(x_t) - d_t\|^2} + \frac{L_FT^2}{2}.
\]

(108)

Setting \( \eta_t = \frac{2}{t+1} \) and using Corollary 2 with \( p = 1 \), we can bound the second term of (108) as

\[
\eta_tD\sqrt{E\|\nabla C(x_t) - d_t\|^2} \leq \frac{2D\sqrt{A_1}}{(t+1)^{3/2}}.
\]

(109)

Now using bound from (109) in (108) we obtain

\[
E[C(x_{t+1}) - C(x*)] \leq \left(1 - \frac{2}{t+1}\right)E[C(x_t) - C(x*)] + \frac{2D\sqrt{A_1}}{(t+1)^{3/2}} + \frac{2L_FT^2}{(t+1)^2}.
\]

(110)

Multiplying both side by \( t(t+1) \) we can write

\[
t(t+1)E[C(x_{t+1}) - C(x*)] \leq t(t-1)E[C(x_t) - C(x*)] + 2D\sqrt{A_1} \frac{t}{(t+1)^{3/2}} + 2L_FT^2 \frac{t}{t+1}
\]

\[
\leq t(t-1)E[C(x_t) - C(x*)] + 2D\sqrt{A_1} (t+1)^{1/2} + 2L_FT^2,
\]

(111)

here in last the inequality we used the fact that \( t < t+1 \). Summing for \( t = 1, 2, \ldots, T \) and rearranging we get

\[
E[C(x_{T+1}) - C(x*)] \leq \frac{1}{T(T+1)} \left( 2D\sqrt{A_1} \sum_{t=1}^T (t+1)^{3/2} + 2L_FT^2 \right)
\]

\[
\leq \frac{1}{T(T+1)} \left( \frac{4}{3} D\sqrt{A_1}(T+1)^{3/2} + 2L_FT^2 \right)
\]

\[
= \frac{(T+1)}{T(T+1)^2} \left( \frac{4}{3} D\sqrt{A_1}(T+1)^{3/2} + 2L_FT^2 \right)
\]

\[
\leq \frac{8D\sqrt{A_1}}{3(T+1)^{1/2}} + \frac{2L_FT^2}{(T+1)},
\]

(112)
here in the second inequality we use the fact that $\sum_{t=1}^{T}(t+1)^{1/2} \leq \frac{2}{3}(T+1)^{3/2}$ and the last inequality is obtained using $(T+1) = T(1 + \frac{1}{T}) \leq 2T$.

2) Proof of Statement (ii) (Non-convex case): Again starting with the smoothness assumption of $F$ and proceeding the same way as (85)-(88), we can write

$$\sum_{t=1}^{T} \eta_t E[G(x_t)] \leq C(x_1) - C(x^*) + 2D \sum_{t=1}^{T} \eta_t \|d_t - \nabla C(x_t)\| + \frac{L_F D^2}{2} \sum_{t=1}^{T} \eta_t^2. \quad (113)$$

Using bound from Corollary 2 for $p = 2/3$, and Jensen’s equality we can write

$$E\|d_t - \nabla C(x_t)\| \leq \sqrt{E \|d_t - \nabla C(x_t)\|^2} \leq \frac{\sqrt{A_1}}{(t+1)^{1/3}}. \quad (114)$$

Now, using (114), setting $\eta_t = \frac{2}{(T+1)^{2/3}}$ in (113) and rearranging we can write

$$E[G(\hat{x})] \leq \frac{1}{T} \sum_{t=1}^{T} E[G(x_t)] \leq \frac{C(x_1) - C(x^*)}{2T(T+1)^{-2/3}} + \sum_{t=1}^{T} \frac{2D\sqrt{A_1}}{T(t+1)^{1/3}} + \sum_{t=1}^{T} \frac{L_F D^2}{T(T+1)^{2/3}}$$

$$\leq \frac{C(x_1) - C(x^*)}{2T(T+1)^{-2/3}} + \frac{3D\sqrt{A_1}}{T(T+1)^{2/3}} + \frac{L_F D^2}{(T+1)^{2/3}}$$

$$= \frac{(T+1)(C(x_1) - C(x^*))}{2T(T+1)^{1/3}} + \frac{3D\sqrt{A_1}}{T(T+1)^{1/3}} + \frac{L_F D^2}{(T+1)^{2/3}}$$

$$\leq \frac{C(x_1) - C(x^*)}{(T+1)^{1/3}} + \frac{6D\sqrt{A_1}}{(T+1)^{1/3}} + \frac{L_F D^2}{(T+1)^{2/3}}$$

$$:= O((T+1)^{-1/3}), \quad (115)$$

here in the second inequality we use the fact that $\sum_{t=1}^{T}(t+1)^{-1/3} \leq \frac{3}{2}(T+1)^{2/3}$ and the last inequality is obtained using $(T+1) = T(1 + \frac{1}{T}) \leq 2T$.

X. Experiment Details

A. Details of the Matrix Completion Problem

We follow the experimental setting of [10] and start with forming an observation matrix as $M = \hat{X} + E$. Here, $\hat{X} = WW^T$ with $W \in \mathbb{R}^{n \times r}$ containing normally distributed independent entries, and the noise matrix $E = \hat{n}(L + L^T)$ where $L \in \mathbb{R}^{n \times n}$ contains normally distributed independent entries and $\hat{n} \in (0, 1)$ is the noise factor. For this simulation, we set $n = 250$, $r = 10$, and $\alpha = \|\hat{X}\|_*$. Further, we define the set of observed entries $\Omega$ by sampling $M$ uniformly at random with probability 0.8.

We start with setting $\hat{n} = 0.5$ and solve the matrix completion problem with SBFW and SFW algorithms. For SBFW, we set the step sizes as dictated in theory and set $\lambda_1 = \lambda_2 = 0.05$, while for SFW, the step sizes are set as defined in [10]. We use a batch size of $b = 250$ for both the algorithms and run them for $10^4$ iterations. We analyze performance in terms of normalized error, defined as

$$e = \frac{\sum_{(i,j) \in \Omega}(X_{i,j} - \hat{X}_{i,j})^2}{\sum_{(i,j) \in \Omega}(\hat{X}_{i,j})^2} \quad (116)$$

where $X$ is the output generated by the algorithm.

The evolution of the normalized error is shown in Fig. 1a. It can be observed that SFW converges fast but gets saturated at some accuracy, while SBFW converges to better accuracy. This justifies the claim that additional
The denoising step improves the quality of matrix completion. To further understand the effect of noise, we solve the problem using both the algorithms with different noise factor $\hat{n} \in (0, 1)$ and compare the error at each $\hat{n}$ with the error obtained for zero noise case (i.e. $\hat{n} = 0$) denoted as $\bar{e}_0$. The plot of the variation in the error difference $\bar{e} - \bar{e}_0$ with noise factor $\hat{n}$ is shown in Fig.1b. It can be observed that the growth in the error difference is much slower for SBFW than the compared method. This shows that the proposed algorithm is able to deal with the effect of noise in a much better way as compared to SFW.

In order to highlight the advantage of projection-free algorithms, we will run the next experiment over a large size matrix ($n = 2000$ and $r = 100$) while keeping other settings as earlier. As SBFW is a single loop algorithm, we will compare its performance with state-of-the-art single loop projection-based bilevel algorithms (SUSTAIN, TTSA and, MSTSA). We keep the step sizes for SBFW as earlier, while for SUSTAIN, TTSA, and MSTSA, we set them as mentioned in [19], [18], and [20], respectively. We set the batch size of $b = 1000$ for all the algorithms, run them for $10^4$ iterations and plot the evolution of normalized error with iteration in Fig.1c. We also compare the computation time required by each algorithm to reach the normalized error of $\epsilon \approx 10^{-2}$ in Fig.1d. It can be observed from Fig.1c that SBFW shows comparable performance in terms of convergence. However, its main advantage comes in terms of computation time, as evident from Fig.1d. Such improvement is achieved due to the fact that all the compared methods require to perform projection over nuclear norm at each iteration that needs computation of full singular value decomposition (svd). In contrast, SBFW solves a single linear minimization problem over nuclear norm constraint that only requires the computation of the singular vectors corresponding to the highest singular values.