STOCHASTIC BILEVEL PROJECTION-FREE OPTIMIZATION*

Zeeshan Akhtar[1], **Amrit Singh Bedi**[2], **Ketan Rajawat**[1]

[1]Department of Electrical Engineering,
Indian Institute of Technology Kanpur, Kanpur 208016, India.
[2] Institute of Systems Research,
University of Maryland, College Park, MD, USA.
zeeshan@iitk.ac.in, amritbd@umd.edu, ketan@iitk.ac.in

Abstract

Bi-level optimization is a powerful framework to solve a rich class of problems such as hyper-parameter optimization, model-agnostic meta-learning, data distillation, and matrix completion. The existing first-order solutions to bi-level problems exhibit scalability limitations (for example, in matrix completion) because of the requirement of projecting solutions onto the feasible set. In this work, we propose a novel **S**tochastic **Bi**-level **F**rank-**W**olfe (SBFW) algorithm to solve the stochastic bi-level optimization problems in a projection-free manner. We utilize a momentum-based gradient tracker that results in a sample complexity of $\mathcal{O}(\epsilon^{-3})$ for convex outer objectives with strongly convex inner objectives. We formulate the matrix completion problem with denoising as a stochastic bilevel problem and show that SBFW outperforms the state-of-the-art methods for the problem of matrix completion with denoising and achieves improvements of up to 82% in terms of the wall-clock time required to achieve the same level of accuracy.

Introduction

We consider the two-level hierarchical optimization problem

$$\min_{\mathbf{x}\in\mathcal{X}\subset\mathbb{R}^m} \mathbb{E}_\theta[f(\mathbf{x},\mathbf{y}^\star(\mathbf{x});\theta)], \tag{1a}$$

$$\text{s.t.} \quad \mathbf{y}^\star(\mathbf{x}) \in \arg\min_{\mathbf{y}\in\mathbb{R}^n} \mathbb{E}_\xi[g(\mathbf{y},\mathbf{x};\xi)]. \tag{1b}$$

Here, the outer problem involves minimizing the objective function $F(\mathbf{x},\mathbf{y}^\star(\mathbf{x})) := \mathbb{E}_\theta[f(\mathbf{y}^\star(\mathbf{x});\theta)]$ with respect to $\mathbf{x}$ over the convex compact constraint set $\mathcal{X} \subset \mathbb{R}^m$. Here $\mathbf{y}^\star(\mathbf{x})$ is a unique solution of the inner optimization problem, which for a given $\mathbf{x}$, entails minimizing the strongly convex function $G(\mathbf{y},\mathbf{x}) := \mathbb{E}_\xi[g(\mathbf{y},\mathbf{x};\xi)]$ with respect to optimization variable $\mathbf{y}$. The function $F(\cdot)$ and $G(\cdot)$ are the expected values of continuous and proper closed functions $f : \mathbb{R}^m \to \mathbb{R}$ and $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ with respect to independent random variables $\theta \in \Theta$ and $\xi \in \Xi$, respectively. Observe that for bilevel problems of type (1), the inner and outer problems are interdependent and cannot be solved in isolation. Yet, these problems arise in a number of areas, such as meta-learning (Rajeswaran et al. 2019), continual learning (Borsos, Mutny, and Krause 2020), reinforcement learning (Zhang et al. 2020a), and hyper-parameter optimization (Franceschi et al. 2018). To solve such problems, the first-order stochastic approximation algorithms have been recently proposed (Yang, Ji, and Liang 2021; Chen, Sun, and Yin 2021). In some works, such as (Khanduri et al. 2021b), the constraint set $\mathcal{X}$ in the outer optimization problem is taken to be $\mathcal{X} = \mathbb{R}^m$, resulting in a simpler unconstrained outer optimization problem. However, in applications such as meta-learning, personalized federated learning, and corsets (Borsos, Mutny, and Krause 2020), the constraint set $\mathcal{X}$ is a strict subset of $\mathcal{X} \subset \mathbb{R}^m$. The standard approach to deal with such constraint sets is to project the updates of the outer optimization problem onto $\mathcal{X}$ 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 that use the Frank-Wolfe (FW) updates. These FW-based algorithms only require solving a linear program over $\mathcal{X}$, which could be significantly cheaper than solving a non-linear projection problem, as in the case of ℓ_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 $\mathcal{O}(\epsilon^{-2})$ (Xie et al. 2020; Akhtar and Rajawat 2022). 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 by 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 **S**tochastic **Bi**-level **F**rank-**W**olfe (SBFW) algorithm, which is the first projection-free algorithm for bi-level problems. Our main contributions are:

- We propose a novel projection-free SBFW algorithm, utilizing the idea of momentum-based gradient update (Cutkosky and Orabona 2019) to track the gradient of the outer objective function. The proposed algorithm is able

*A. S. Bedi acknowledges the support by Army Cooperative Agreement W911NF2120076.

Reference	Projection Free	Problem Type	SFO Complexity (Outer)	SFO Complexity (Inner)
SFW(Mokhtari, Hassani, and Karbasi 2020)	yes	Single-Level	$\mathcal{O}(\epsilon^{-3})$	-
ORGFW (Xie et al. 2020)	yes	Single-Level	$\mathcal{O}(\epsilon^{-2})$	-
SFW^{++}(Zhang et al. 2020b)	yes	Single-Level	$\mathcal{O}(\epsilon^{-3})$	-
BSA (Ghadimi and Wang 2018)	no	Bi-Level	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-3})$
stocBiO[1] , STABLE[2], MSTSA[3]	no	Bi-Level	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-2})$
TTSA (Hong et al. 2020)	no	Bi-Level	$\mathcal{O}(\epsilon^{-2.5})$	$\mathcal{O}(\epsilon^{-2.5})$
SUSTAIN (Khanduri et al. 2021b)	no	Bi-Level	$\mathcal{O}(\epsilon^{-1.5})$	$\mathcal{O}(\epsilon^{-1.5})$
SBFW (proposed)	yes	Bi-Level	$\mathcal{O}(\epsilon^{-3})$	$\mathcal{O}(\epsilon^{-1.5})$

Table 1: Comparision of the stochastic first-order complexity (SFO) for the outer and inner optimization problems.[1](Ji, Yang, and Liang 2020), [2] (Chen, Sun, and Yin 2021), [3] (Khanduri et al. 2021a)

to achieve a sample complexity of $\mathcal{O}(\epsilon^{-3})$.

- We test the proposed algorithm on matrix completion and establish the efficacy of the proposed techniques compared to state-of-the-art algorithms (cf. Sec. 4). We achieve an improvement of up to 82% in the computation time for the proposed algorithm as compared to state-of-the-art methods.

Related Works

A series of works proposed to solve the problem of the form (1) has appeared recently (Yang, Ji, and Liang 2021; Chen, Sun, and Yin 2021; Khanduri et al. 2021b; Huang and Huang 2021). The seminal works in (Ghadimi and Wang 2018; Yang, Ji, and Liang 2021) proposed a class of double-loop approximation algorithms to iteratively approximate the stochastic gradient of the outer objective and incurred a sample complexity of $\mathcal{O}(\epsilon^{-2})$ in order to achieve the ϵ-stationary point. The double loop structure of these approaches made them impractical for large-scale problems; (Ghadimi and Wang 2018) required solving an inner optimization problem to a predefined accuracy, while (Yang, Ji, and Liang 2021) required a large batch size of $\mathcal{O}(\epsilon^{-1})$ at each iteration. To address this issue, various single-loop methods involving simultaneous updates of inner and outer optimization variables have been developed (Chen, Sun, and Yin 2021; Khanduri et al. 2021b; Yang, Ji, and Liang 2021). A single-loop two-time scale stochastic algorithm proposed in (Hong et al. 2020) incurred a sub-optimal sample complexity of $\mathcal{O}(\epsilon^{-2.5})$. This is further improved recently in (Chen, Sun, and Yin 2021; Khanduri et al. 2021b; Yang, Ji, and Liang 2021), in which the authors have utilized the momentum-based variance reduction technique from (Cutkosky and Orabona 2019) to obtain optimal convergence rates. While all of the above-mentioned works seek to solve (1), they all require a projection onto $\mathcal{X}$ 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. A comprehensive list of all existing related works is provided in Table 1.

Motivating Example

In general, for noise-free data, the data matrix in the matrix completion problem can be modeled as a low-rank matrix motivating the use of the nuclear norm constraint. Low-rank matrix completion problem arises in various applications such as image processing, multi-task learning, and collaborative filtering. However, under noisy observations, directly solving the matrix completion problem with just the nuclear norm constraints can result in sub-optimal performance (McRae and Davenport 2021). Further, noise is present in many vision applications, and using only low-rank priors is insufficient to recover the underlying matrix. A common approach to tackle the noise is to apply a denoising algorithm as a pre-processing step. In general, however, it is necessary to apply some heuristics since denoising algorithms require access to the full matrix, which is not available in the pre-processing stage. Denoising is also impractical in online settings, where a random subset of the matrix entries is observed at every iteration. The bilevel optimization framework provides a way out, allowing the incorporation of denoising steps within the inner-level sub-problem. Mathematically, the bi-level matrix completion with denoising problem can be written as

$$\min_{\|\mathbf{X}\|_* \leq \alpha} \frac{1}{|\Omega_1|} \sum_{(i,j)\in\Omega_1} (\mathbf{X}_{i,j} - \mathbf{Y}_{i,j})^2, \tag{2}$$
$$\text{s. t.} \quad \mathbf{Y} \in \arg\min_{\mathbf{V}} \Big\{ \frac{1}{|\Omega_2|} \sum_{(i,j)\in\Omega_2} (\mathbf{V}_{i,j} - \mathbf{M}_{i,j})^2 + \lambda_1 \|\mathbf{V}\|_1 + \lambda_2 \|\mathbf{X} - \mathbf{V}\|_F^2 \Big\},$$

where $\mathbf{M} \in \mathbb{R}^{n\times m}$ is the given incomplete noisy matrix, $\|\mathbf{V}\|_1 := \sum_{i,j} |\mathbf{V}_{i,j}|$ is the sum-absolute-value (ℓ_1) norm, λ_1 and λ_2 are regularization parameters, and Ω_1 and Ω_2 represents the set of available entries at outer and inner level respectively. Note that the regularization over the discrepancy between $\mathbf{X}$ and denoised matrix $\mathbf{Y}$ results in bilevel formulation (2). A similar technique in deterministic settings is utilized in various other applications in machine learning and signal processing problems (Crockett and Fessler 2021). The problem in (2) is a special case of general formulation in (1) with $f(\mathbf{x}, \mathbf{y}^\star) := \sum_{i,j} (\mathbf{X}_{i,j} - \mathbf{Y}_{i,j})^2$ with $\mathbf{x} := \mathbf{X}$, $\mathbf{y}^\star := \mathbf{Y}$ and $g := \|\mathbf{X} - \mathbf{V}\|_F^2$ with $\mathbf{y} := \mathbf{V}$. However, when the entries are revealed in the form of randomly selected subsets $\Omega_1^t \subset \Omega_1$ and $\Omega_2^t \subset \Omega_2$ at every iteration, it becomes stochastic in nature. The main challenge here is due to the nuclear norm constraint, which makes it quite computationally expensive (sometimes even impractical) to solve (2) using projection-based bilevel algorithms. In Sec. 4, we will show experimentally that the proposed algorithm SBFW is best suited to address such challenges.

Algorithm Development

We note that solving the bi-level optimization problem in (1) is NP-hard in general, but we restrict our focus to problems where the inner objective is continuously twice differentiable in $(\mathbf{x}, \mathbf{y})$ and also strongly convex w.r.t $\mathbf{y}$ with parameter $\mu_g > 0$. Such an assumption is common in the related works (Ghadimi and Wang 2018; Chen, Sun, and Yin 2021; Yang, Ji, and Liang 2021) and ensures that $\mathbf{y}^\star(\mathbf{x})$ is unique for any $\mathbf{x} \in \mathcal{X}$. A stochastic projected gradient descent update to solve (1) can be written as

$$\mathbf{x}_{t+1} = \mathcal{P}_{\mathcal{X}}\left[\mathbf{x}_t - \alpha h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t)\right], \tag{3}$$

where we can write the biased estimate $h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t)$ as

$$\begin{aligned} &h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t) \\ &\quad = \nabla_{\mathbf{x}} f(\mathbf{x}_t, \mathbf{y}_t; \theta_t) - M(\mathbf{x}_t, \mathbf{y}_t; \tilde{\xi}_t) \cdot \nabla_{\mathbf{y}} f(\mathbf{x}_t, \mathbf{y}_t; \theta_t), \end{aligned} \tag{4}$$

where $\nabla_{\mathbf{x}} f(\mathbf{x}_t, \mathbf{y}_t; \theta_t)$ is an unbiased estimate of $\nabla_{\mathbf{x}} F(\mathbf{x}_t, \mathbf{y}_t)$, $\nabla_{\mathbf{y}} f(\mathbf{x}_t, \mathbf{y}_t; \theta_t)$ is an unbiased estimate of $\nabla_{\mathbf{y}} F(\mathbf{x}_t, \mathbf{y}_t)$, and $M(\mathbf{x}_t, \mathbf{y}_t; \tilde{\xi}_t)$ is a biased estimate of product $\nabla^2_{\mathbf{xy}} G(\mathbf{y}_t, \mathbf{x}_t) \cdot [\nabla^2_{\mathbf{yy}} G(\mathbf{y}_t, \mathbf{x}_t)]^{-1}$. Here, we have used ξ_t in LHS to highlight the fact that the hessian of the function $g(\cdot)$ is also random in nature. The term $M(\mathbf{x}_t, \mathbf{y}_t; \tilde{\xi}_t)$ is a biased estimation of $[\nabla^2_{\mathbf{yy}} G(\mathbf{y}_t, \mathbf{x}_t)]^{-1}$ with bounded variance. The explicit form of $M(\mathbf{x}_t, \mathbf{y}_t; \tilde{\xi}_t)$ is

$$\begin{aligned} M(\mathbf{x}_t, \mathbf{y}_t; \tilde{\xi}_t) =& \nabla^2_{\mathbf{xy}} g(\mathbf{y}_t, \mathbf{x}_t; \xi_{t,0}) \times \\ & \times \left[\frac{k}{L_g} \Pi_{i=1}^{l} \left(I - \frac{1}{L_g} \nabla^2_{\mathbf{yy}} g(\mathbf{y}_t, \mathbf{x}_t; \xi_{t,i})\right)\right], \end{aligned} \tag{5}$$

here, $\tilde{\xi}_t$ is a collection of $k+1$ i.i.d. samples i.e. $\tilde{\xi}_t := \{\xi_{t,i} : i \in \{0, 1, \cdots, k\}\}$, with $\xi_{t,0}$ being the sample of $\nabla^2_{\mathbf{xy}} g(\mathbf{y}_t, \mathbf{x}_t)$ and $\xi_{t,1}, \cdots, \xi_{t,k}$ being the i.i.d. samples of $\nabla^2_{\mathbf{yy}} g(\mathbf{y}_t, \mathbf{x}_t)$. Further, the parameter l is selected uniformly from $\{1, \cdots, k\}$ and for l=0, we use the convention $\Pi_{i=1}^{l}\left(I - \frac{1}{L_g}\nabla^2_{\mathbf{yy}} g(\mathbf{x}_t, \mathbf{y}_t; \xi_{t,i})\right) = I$. Here, I is the identity matrix, and L_g is the Lipschitz parameter for gradient $\nabla_{\mathbf{y}} g(\mathbf{x}, \mathbf{y})$.

Similar to the update in (3), a significant challenge that remains unaddressed to date in the literature for the bi-level problems is associated with the projection operator in (3). The projection is easy to evaluate when the constraint set is a simple convex set (onto which projection operation is computationally cheap such as probability simplex) or has a closed-form solution (set of unit-ball). However, the projection step is often computationally costly (e.g., nuclear norm constraint), and its complexity could be comparable to the problem at hand (Jaggi 2013). In this work, we alleviate this issue by proposing projection-free algorithms for bi-level optimization problems, which is the key novel aspect of our work.

Stochastic Projection-Free Bi-level Optimization

Before proceeding, we discuss the particular choice of $\mathbf{y}_t$ in (3). A popular choice (see (Chen, Sun, and Yin 2021; Hong et al. 2020; Ji, Yang, and Liang 2020)) for $\mathbf{y}_t$ is the stochastic gradient descent update for the inner optimization problem given by $\mathbf{y}_t = \mathbf{y}_{t-1} - \delta_t \nabla_{\mathbf{y}} g(\mathbf{y}_{t-1}, \mathbf{x}_{t-1}; \xi_t)$, where $\nabla_{\mathbf{y}} g(\mathbf{y}_{t-1}, \mathbf{x}_{t-1}; \xi_t)$ is the unbiased estimate of the gradient $\mathbb{E}_{\xi}\left[\nabla_{\mathbf{y}} g(\mathbf{y}_{t-1}, \mathbf{x}_{t-1}; \xi)\right]$, and $\delta_t > 0$ denotes the step size. Now we are ready to propose the first projection-free algorithm for bi-level stochastic optimization problems. We propose to use a conditional gradient method (CGM) based approach (Jaggi 2013; Hazan and Luo 2016) instead of calculating the projection in (3). That is, we solve a linear minimization problem to find a feasible direction $\mathbf{s}_t \in \mathcal{X}$ for a given stochastic gradient direction $h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t)$, given by, $\mathbf{s}_t := \arg\min_{\mathbf{s} \in \mathcal{X}} \langle \mathbf{s}, h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t) \rangle$. This reduces the optimization problem of evaluating the projection operator in (3) to solving a linear program which is easier to solve in practice. Hence, the iterate in (3) gets modified to

$$\mathbf{s}_t := \arg\min_{\mathbf{x} \in \mathcal{X}} < \mathbf{x}, h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t) > \tag{6}$$

$$\mathbf{x}_{t+1} = (1 - \eta_{t+1})\mathbf{x}_t + \eta_{t+1}\mathbf{s}_t, \tag{7}$$

where $\eta_t > 0$ is the step size. To this end, we would like to emphasize that naive use of $h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t)$ in (6) for the evaluation of $\mathbf{s}_t$ which is then used in (7) can result in the iterate divergence due to the non-vanishing variance of the gradient estimate (Mokhtari, Hassani, and Karbasi 2020). 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 (Hazan and Luo 2016; Reddi et al. 2016) with linearly increasing batch size with iteration index. Such an approach runs into memory issues when utilized in practice. To address the issue of memory and iterate divergence, we took motivation from the momentum-based approach in (Cutkosky and Orabona 2019) and propose to use the following gradient tracking scheme given by

$$\begin{aligned} \mathbf{d}_t =& (1 - \rho_t)(\mathbf{d}_{t-1} - h(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}; \theta_t, \xi_t)) \\ & + h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t). \end{aligned} \tag{8}$$

We remark that such a tracking technique is recently utilized in (Khanduri et al. 2021b) for projection-based bi-level op-

Algorithm 1: **S**tochastic **B**i-level **F**rank **W**olfe

Input: $\mathbf{x}_1 \in \mathcal{X}, \mathbf{y}_1 \in \mathbb{R}^m, \eta_t, \delta_t, \rho_t, \beta_t$, and $\mathbf{d}_1 = h_1(\theta_1; \xi_1)$ using (4)

for $t = 2$ ***to*** T **do**

  Update approximate inner optimization solution

$$\mathbf{y}_t = \mathbf{y}_{t-1} - \delta_t \nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \xi_t)$$

  Gradient tracking evaluate $h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t)$ and $h(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}; \theta_t, \xi_t)$ using (4) and compute

$$\begin{aligned} \mathbf{d}_t = (1 - \rho_t)(\mathbf{d}_{t-1} - h(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}; \theta_t, \xi_t) \\ + h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t) \end{aligned}$$

  Evaluate feasible direction $\mathbf{s}_t = \arg\min_{\mathbf{s} \in \mathcal{X}} \langle \mathbf{s}, \mathbf{d}_t \rangle$;

  Update solution $\mathbf{x}_{t+1} = (1 - \eta_t)\mathbf{x}_t + \eta_t \mathbf{s}_t$

Output: $\mathbf{x}_{T+1}$ or $\hat{\mathbf{x}}$ selected uniformly from $\{\mathbf{x}_i\}_{i=1}^{T}$

timization problems. However, in this work, our focus lies in developing projection-free algorithms, and hence analysis is significantly different from (Khanduri et al. 2021b). Our proposed algorithm is summarised in Algorithm 1.

Convergence Analysis: SBFW

We will start with the assumptions required to perform the analysis in this work that is similar to the assumptions considered in the existing literature (Hong et al. 2020; Khanduri et al. 2021a).

Assumption 1 *For some $\sigma_{\mathbf{x}}^2 > 0$, $\sigma_{\mathbf{y}}^2 > 0$, $\sigma_{\mathbf{xy}}^2 > 0$, and $\sigma_g^2 > 0$ we have $\mathbb{E}[\{\nabla_{\mathbf{x}} f(\mathbf{x},\mathbf{y}) - \nabla_{\mathbf{x}} f(\mathbf{x},\mathbf{y},\theta)\}^2] \leq \sigma_{\mathbf{x}}^2$, $\mathbb{E}[\{\nabla_{\mathbf{y}} f(\mathbf{x},\mathbf{y}) - \nabla_{\mathbf{y}} f(\mathbf{x},\mathbf{y},\theta)\}^2] \leq \sigma_{\mathbf{y}}^2$, $\mathbb{E}[\{\nabla_{\mathbf{xy}}^2 g(\mathbf{x},\mathbf{y}) - \nabla_{\mathbf{xy}}^2 g(\mathbf{x},\mathbf{y},\theta)\}^2] \leq \sigma_{\mathbf{xy}}^2$, $\mathbb{E}[\{\nabla_{\mathbf{y}} g(\mathbf{x},\mathbf{y}) - \nabla_{\mathbf{y}} g(\mathbf{x},\mathbf{y},\xi)\}^2] \leq \sigma_g^2$.*

Assumption 2 *For any given $\mathbf{x} \in \mathcal{X}$, the terms $\nabla_{\mathbf{x}} f(\mathbf{x},\mathbf{y})$, $\nabla_{\mathbf{y}} f(\mathbf{x},\mathbf{y})$, $\nabla_{\mathbf{y}} g(\mathbf{x},\mathbf{y})$, $\nabla_{\mathbf{xy}}^2 g(\mathbf{x},\mathbf{y})$ and $\nabla_{\mathbf{yy}}^2 g(\mathbf{x},\mathbf{y})$ are Lipschitz continuous with respect to $\mathbf{y}$ with Lipschitz parameter $L_{f_{\mathbf{x}}}$, $L_{f_{\mathbf{y}}}$, L_g, $L_{g_{\mathbf{xy}}}$ and $L_{g_{\mathbf{yy}}}$, respectively. Similarly, for any given $\mathbf{y} \in \mathbb{R}^n$, the terms $\nabla_{\mathbf{y}} f(\mathbf{x},\mathbf{y})$, $\nabla_{\mathbf{xy}}^2 g(\mathbf{x},\mathbf{y})$ and $\nabla_{\mathbf{yy}}^2 g(\mathbf{x},\mathbf{y})$ are Lipschitz continuous with respect to $\mathbf{x}$ with positive constants $L_{f_{\mathbf{y}}}$, $L_{g_{\mathbf{xy}}}$ and $L_{g_{\mathbf{yy}}}$, respectively. Note that for the sake of simplicity, here we slightly abused the notation and used the same constants.*

Assumption 3 *For all $\mathbf{x} \in \mathcal{X}$ and $\mathbf{y} \in \mathbb{R}^n$, it holds that $\mathbb{E}[\|\nabla_{\mathbf{y}} f(\mathbf{x},\mathbf{y})\|] \leq C_{\mathbf{y}}$ and $\mathbb{E}[\|\nabla_{\mathbf{xy}}^2 g(\mathbf{x},\mathbf{y})\|] \leq C_{\mathbf{xy}}$ for some for constants $C_{\mathbf{y}} > 0$ and $C_{\mathbf{xy}} > 0$.*

Assumption 4 *The inner function $g(\mathbf{x},\mathbf{y})$ is μ_g-strongly convex in $\mathbf{y}$ for any $\mathbf{x} \in \mathcal{X}$.*

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

Lemma 1 *Consider the proposed Algorithm 1 and $\mathbf{x}_t \in \mathbb{N}^+$ be the iterates generated by it, then for the algorithm parameter $\delta_t \leq \min\{\frac{2}{3\mu_g}, \frac{\mu_g}{2(1+\sigma_g^2)L_g^2}\}$ and step size η_t, it holds that*
(i) the optimality gap of the lower level problem satisfies

$$\mathbb{E}_t[\|\mathbf{y}_t - \mathbf{y}^\star(\mathbf{x}_t)\|^2] \leq \left(1 - \frac{\delta_t \mu_g}{2}\right) \tag{9}$$

$$\mathbb{E}_t[\|\mathbf{y}_{t-1} - \mathbf{y}^\star(\mathbf{x}_{t-1})\|^2] + \frac{2\eta_{t-1}^2}{\delta_t \mu_g}\left(\frac{C_{\mathbf{xy}}}{\mu_g}\right)^2 D^2 + 4\delta_t^2 \sigma_g^2.$$

(ii) Also, for the constant b_1 defined as $b_1 = \max\{2^q \|\mathbf{y}_1 - \mathbf{y}^\star(\mathbf{x}_1)\|^2, (2(C_{\mathbf{xy}}/\mu_g)^2 D^2 + 16a_0^2 \sigma_g^2)/(2a_0 - 1)\}$, it holds that

$$\mathbb{E}[\|\mathbf{y}_t - \mathbf{y}^\star(\mathbf{x}_t)\|] \leq \frac{b_1}{(t+1)^q}, \tag{10}$$

The proof Lemma 1 is provided in Appendix 4. Lemma 1 quantifies how close $\mathbf{y}_t$ is from the optimal solution of the inner problem at $\mathbf{x}_t$ and establishes the progress of the inner-level update.

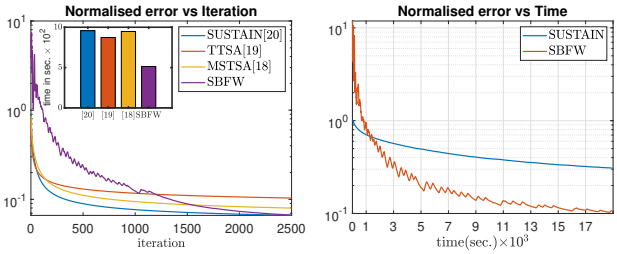

Figure 1: Left: Comparision between the proposed SBFW algorithm for matrix completion problem on MovieLens 100k dataset with TTSA (Hong et al. 2020), MSTSA (Khanduri et al. 2021a), and SUSTAIN (Khanduri et al. 2021b). Right: This figure compares the normalized error concerning computation time required for SBFW and SUSTAIN(Khanduri et al. 2021a) on MovieLens 1M dataset.

Lemma 2 *Consider the proposed Algorithm 1 and $\mathbf{x}_t \in \mathbb{N}^+$ be the iterates generated by it, then for the algorithm parameter δ_t, ρ_t, and η_t, we have*

$$\begin{aligned}&\mathbb{E}_t[\|\mathbf{d}_t - \nabla S(\mathbf{x}_t,\mathbf{y}_t) - B_t\|^2 \\ &\leq (1-\rho_t)^2 \mathbb{E}_t[\|(\mathbf{d}_{t-1} - \nabla S(\mathbf{x}_{t-1},\mathbf{y}_{t-1}) - B_{t-1})\|^2] \\ &\quad + 4L_k\delta_t^2 L_g^2 + 4L_k\eta_{t-1}^2 D^2 + 2\rho_t^2\sigma_f^2. \end{aligned} \tag{11}$$

where the bias B_t is defined as $B_t := \mathbb{E}[h(\mathbf{x}_t,\mathbf{y}_t;\theta_t,\xi_t)] - \nabla S(\mathbf{x}_t,\mathbf{y}_t)$.

The proof of Lemma 2 is provided in Appendix 4, which is based on the proof of Lemma C.3 of (Khanduri et al. 2021a). However, different from (Khanduri et al. 2021a), our proof bounds the term $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2$ using the update equation $\mathbf{x}_{t+1} = (1-\eta_t)\mathbf{x}_t + \eta_t \mathbf{s}_t$ from Algorithm 1 along with compactness assumption of the domain $\mathcal{X}$. Lemma 2 describe the tracking error in the gradient approximation $\nabla S(\mathbf{x},\mathbf{y})$ at point $\mathbf{x}_t$ and $\mathbf{y}_t$. The presence of $(1-\rho_t)^2$ term in RHS of (11) shows that the variance of the tracking error reduces with iteration. Next, we utilize Lemma 1-(2) to establish a bound on the gradient estimation error for ∇Q as Corollary 1.

Corollary 1 *For the proposed Algorithm 1, with $\delta_t = \frac{2a_0}{t^q}$, where $a_0 = \min\{\frac{1}{3\mu_g}, \frac{\mu_g}{2(1+\sigma_g^2)L_g^2}\}$, $\rho_t = \frac{2}{t^q}$, $\beta_t \leq \frac{C_{\mathbf{xy}} C_{\mathbf{y}}}{\mu_g (t+1)^q}$ and $\eta_t \leq \frac{2}{(t+1)^{3q/2}}$ for $0 < q \leq 1$, the gradient approximation error $\mathbb{E}\|\nabla Q(\mathbf{x}_t) - \mathbf{d}_t\|^2$ converges to zero at the following rate*

$$\mathbb{E}\|\nabla Q(\mathbf{x}_t) - \mathbf{d}_t\|^2 \leq \frac{C_1}{(t+1)^q}, \tag{12}$$

where $C_1 = 3(\max\{2^q \|\mathbf{y}_1 - \mathbf{y}^\star(\mathbf{x}_1)\|^2, (2(C_{\mathbf{xy}}/\mu_g)^2 D^2 + 16a_0^2\sigma_g^2)/(2a_0 - 1)\} + \frac{C_{\mathbf{xy}} C_{\mathbf{y}}}{\mu_g} + 8(2L_k L_g^2 + L_k D^2 + \sigma_f^2))$.

The proof of Corollary 1 is provided in Appendix 4. The result in Corollary 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 types of objective functions in

Dataset	#users	#movies	#ratings	Time		
				SUSTAIN	SBFW	%imp.
Movielens 100k	1000	1700	10^5	959 sec.	433 sec.	55%
Movielens latest	600	9000	10^5	66.6 mins.	12.9 mins.	81%
Movielens 1M	6000	4000	10^6	10.16 hrs.	1.82 hrs.	82%

Table 2: Comparison of computation time of the proposed algorithm SBFW and the state-of-the-art projection-based algorithm SUSTAIN over the different sizes of real data sets.

the following theorem. Note that in the analysis of Corollary 1 we have set $\beta_t \leq \frac{C_{\mathbf{xy}} C_{\mathbf{y}}}{\mu_g (t+1)^q}$. To satisfy this condition, the number of samples k at iteration t needed to approximate the Hessian inverse in (5) is $k = \mathcal{O}(\log((1+t)^q))$.

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

Theorem 1 *Consider the proposed Algorithm 1 with* $\delta_t = \frac{a_0}{t^{2/3}}$, *where* $a_0 = \min\{\frac{2}{3\mu_g}, \frac{\mu_g}{2(1+\sigma_g^2)L_g^2}\}$, $\rho_t = \frac{2}{t^{2/3}}$, $\eta_t = \frac{2}{t+1}$ *and* $k = \frac{2L_g}{3\mu_g}(\log(1+t))$, *then the output is feasible* $\mathbf{x}_{T+1} \in \mathcal{X}$ *and satisfies*

$$\mathbb{E}[Q(\mathbf{x}_{T+1}) - Q(\mathbf{x}^\star)] \leq \frac{12D\sqrt{C_1}}{5(T+1)^{\frac{1}{3}}} + \frac{2L_Q D^2}{(T+1)}. \quad (13)$$

here $L_Q = \frac{(L_{f_y}+L)C_{\mathbf{xy}}}{\mu_g} + L_{f_x} + C_y\left[\frac{L_{g_{\mathbf{xy}}}C_y}{\mu_g} + \frac{L_{g_{yy}}C_x y}{\mu_g^2}\right]$ *and* $C_1 = 3(\max\{2\|\mathbf{y}_1 - \mathbf{y}^\star(\mathbf{x}_1)\|^2, (2(C_{\mathbf{xy}}/\mu_g)^2 D^2 + 16a_0^2\sigma_g^2)/(2a_0 - 1)\} + \frac{C_{\mathbf{xy}}C_{\mathbf{y}}}{\mu_g} + 8(2L_k L_g^2 + L_k D^2 + \sigma_f^2))$.

The proof of Theorem 1 is provided in Appendix 4. Theorem 1 shows that the optimality gap for SBFW decays as $\mathcal{O}(T^{-1/3})$ for general convex objectives, where T is the total number of iterations. We note that for at each iteration, SBFW requires $2k+1$ gradient samples to obtain gradient estimate: $2k$ samples for outer gradient estimate (8) and one sample for the inner variable update. Further, we have set $k \approx \mathcal{O}(\log(t))$. Hence, the SFO complexity of SBFW for the outer objective is $\mathcal{O}(\log(\epsilon^{-1})\epsilon^{-3}) \approx \mathcal{O}(\epsilon^{-3})$. Similarly, observe that $\mathbb{E}[\|\mathbf{y}_t - \mathbf{y}^\star(\mathbf{x}_t)\|] \leq \mathcal{O}((t+1)^{-q})$, where $q = 2/3$ (as $\delta_t = \mathcal{O}(t^{-2/3})$). Hence, the SFO complexity of the inner objective for SBFW is $\mathcal{O}(\epsilon^{-1.5})$. It can be seen that complexity for the inner level objective of the proposed algorithm SBFW is comparable to the projection-based state-of-the-art methods (Ji, Yang, and Liang 2020; Chen, Sun, and Yin 2021; Khanduri et al. 2021b); 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.

Numerical Experiments

In this section, first, we consider the problem of low-rank matrix completion formulated in (2) to illustrate the performance of our proposed SBFW algorithm. All the experiments are performed in MATLAB R2018a with Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz. To test the scalability of our proposed projection-free algorithm, we run an experiment over large-size matrices of MovieLens [1] datasets, which contain user ratings of movies ranging from 0 to 5. We start with Movielens 100k dataset of 10^5 ratings from 1000 users for 1700 movies. This dataset is denoted by observation matrix $\mathbf{M}$ of size 1000×1700. For the simulations, we define the set of observed entries Ω by sampling the matrix uniformly at random from $\mathbf{M}$ with a batch size of $b = 500$.

Note that as SBFW is a single loop algorithm, we compare the performance of SBFW with other state-of-the-art single loops projection-based bilevel algorithms such as SUSTAIN (Khanduri et al. 2021b), TTSA (Hong et al. 2020), and MSTSA (Khanduri et al. 2021a) . Fig. 1 plots the evolution of normalized error for 2500 iterations for all the algorithms. We note (Fig. 1(left)) that the proposed algorithm is not the best in terms of the convergence rate when compared to projection-based schemes, which is expected from the slower theoretical convergence rates. However, when compared in terms of the amount of clock time required to achieve the same level of normalized error (0.68×10^{-1}), the proposed scheme outperforms the other state-of-the-art methods as shown in the bar plot of Fig. 1(left).

To further highlight the importance of the projection-free bilevel algorithm in practice, we perform additional experiments on a larger dataset (of MovieLens 1M), which contains 1 million ratings from 6000 users and for 4000 movies. We plot the evolution of normalized error with time in Fig. 1(right), where we only compare SBFW against SUSTAIN, which is the state-of-the-art projection-based bilevel algorithm. It is interesting to note that even though SUSTAIN has a better theoretical convergence rate, it shows inferior performance in actual computation time (due to the projection operation) compared to SBFW, as evident from Fig. 1(right). In Table 2, we provide computation time comparisons (to complete 10^3 iteration) of both the algorithms (under same settings) over different real datasets. Observe that for large data sets, SBFW is approximately $10\times$ faster than the SUSTAIN and exhibits an improvement upto 82% in the computation time. This performance gain in terms of computation time comes from the fact that other methods require performing projections over the nuclear norm at each iteration which is computationally expensive due to the computation of full singular value decomposition. In contrast, SBFW solves only a single linear program at each iteration, which only requires the computation of singular vectors corresponding to the highest singular value.

Conclusion

This paper presents the first projection-free algorithm for stochastic bi-level optimization problems with a strongly convex inner objective function. We utilize the concept of momentum-based tracking to track the stochastic gradient estimate and establish the oracle complexities of the proposed SBFW algorithm for the convex outer objective functions. Numerical results show that the proposed projection-free variant has a significantly reduced wall-clock time as compared to its projection-based counterparts.

[1] https://grouplens.org/datasets/movielens/

Appendix

We will start with deriving an upper bound on the expected estimation error when the momentum-based method is employed to track the function or gradient.

Lemma 3 *Let us estimate function $\Psi(\mathbf{x}) = \mathbb{E}_\xi[\Psi(\mathbf{x},\xi)]$ by $\mathbf{y}_t$ using step size δ_t as follows*

$$\mathbf{y}_t = (1-\delta_t)(\mathbf{y}_{t-1} - \Psi(\mathbf{x}_{t-1},\xi_t)) + \Psi(\mathbf{x}_t,\xi_t). \quad (14)$$

Then the expected tracking error $\mathbb{E}_t[\|\mathbf{y}_t - \Psi(\mathbf{x}_t)\|^2]$ satisfies

$$\begin{aligned}\mathbb{E}_t[\|\mathbf{y}_t - \Psi(\mathbf{x}_t)\|^2] &\le (1-\delta_t)^2\|(\mathbf{y}_{t-1} - \Psi(\mathbf{x}_{t-1}))\|^2 \\ &+ 2(1-\delta_t)^2\mathbb{E}_t[\|\Psi(\mathbf{x}_t,\xi_t) - \Psi(\mathbf{x}_{t-1},\xi_t)\|^2] \\ &+ 2\delta_t^2\mathbb{E}_t[\|\Psi(\mathbf{x}_t,\xi_t) - \Psi(\mathbf{x}_t)\|^2]. \quad (15)\end{aligned}$$

Proof: Consider the update equation in (14), add/subtract the term $(1-\delta_t)\Psi(\mathbf{x}_{t-1})$ in the right hand side of (14) to obtain

$$\begin{aligned}\mathbf{y}_t = (1-\delta_t)(\mathbf{y}_{t-1} - \Psi(\mathbf{x}_{t-1},\xi)) + \Psi(\mathbf{x}_t,\xi_t) \\ + (1-\delta_t)\Psi(\mathbf{x}_{t-1}) - (1-\delta_t)\Psi(\mathbf{x}_{t-1}). \quad (16)\end{aligned}$$

Subtract $\Psi(\mathbf{x}_t)$ from both sides in (16) and take norm square:

$$\begin{aligned}\|\mathbf{y}_t - \Psi(\mathbf{x}_t)\|^2 = \|(1-\delta_t)(\mathbf{y}_{t-1} - \Psi(\mathbf{x}_{t-1})) \quad (17)\\ - (1-\delta_t)(\Psi(\mathbf{x}_{t-1},\xi_t) - \Psi(\mathbf{x}_{t-1})) + \Psi(\mathbf{x}_t,\xi_t) - \Psi(\mathbf{x}_t)\|^2.\end{aligned}$$

Now, expand the square and calculate conditional expectation $\mathbb{E}_t = \mathbb{E}[(\cdot)|\mathcal{F}_t]$ to obtain

$$\begin{aligned}\mathbb{E}_t[\|\mathbf{y}_t - \Psi(\mathbf{x}_t)\|^2] = (1-\delta_t)^2\|(\mathbf{y}_{t-1} - \Psi(\mathbf{x}_{t-1}))\|^2 \quad (18)\\ - 2\langle(1-\delta_t)(\mathbf{y}_{t-1} - \Psi(\mathbf{x}_{t-1})), (1-\delta_t)(\mathbb{E}_t[\Psi(\mathbf{x}_{t-1}) \\ - \Psi(\mathbf{x}_{t-1},\xi_t)]) + \mathbb{E}_t[\Psi(\mathbf{x}_t) - \Psi(\mathbf{x}_t,\xi_t)]\rangle \\ + \mathbb{E}_t\|(1-\delta_t)(\Psi(\mathbf{x}_{t-1},\xi_t) - \Psi(\mathbf{x}_{t-1})) + \Psi(\mathbf{x}_t) - \Psi(\mathbf{x}_t,\xi_t)\|^2.\end{aligned}$$

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

$$\begin{aligned}&\mathbb{E}_t[\|\mathbf{y}_t - \Psi(\mathbf{x}_t)\|^2] \\ &= (1-\delta_t)^2\|(\mathbf{y}_{t-1} - \Psi(\mathbf{x}_{t-1}))\|^2 \quad (19)\\ &+ \mathbb{E}_t\|(1-\delta_t)(\Psi(\mathbf{x}_{t-1},\xi_t) - \Psi(\mathbf{x}_{t-1})) + \Psi(\mathbf{x}_t) - \Psi(\mathbf{x}_t,\xi_t)\|^2 \\ &\le (1-\delta_t)^2\|(\mathbf{y}_{t-1} - \Psi(\mathbf{x}_{t-1}))\|^2 + \\ &+ 2(1-\delta_t)^2\mathbb{E}_t[\|\Psi(\mathbf{x}_t,\xi_t) - \Psi(\mathbf{x}_{t-1},\xi_t)\|^2] \\ &+ 2\delta_t^2\mathbb{E}_t[\|\Psi(\mathbf{x}_t,\xi_t) - \Psi(\mathbf{x}_t)\|^2] \quad (20)\end{aligned}$$

where the last inequality holds due to the fact that $\mathbb{E}\|X - \mathbb{E}[X] + Y\|^2 \le 2\mathbb{E}\|X\|^2 + 2\mathbb{E}\|Y\|^2$ for any two random variables X and Y.

Before proceeding, we will discuss some of the existing results which are useful for the analysis in this paper.

Lemma 4 *[(Ghadimi and Wang 2018), Lemma 2.2] Under Assumption1, the following statements hold.*

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

$$\|\nabla S(\mathbf{x},\mathbf{y}) - \nabla Q(\mathbf{x})\| \le L\|\mathbf{y}^\star(\mathbf{x}) - \mathbf{y}\|, \quad (21)$$

where $L := L_{f_\mathbf{x}} + \frac{L_{f_\mathbf{y}}C_{\mathbf{xy}}}{\mu_g} + C_\mathbf{y}\left[\frac{L_{g_{\mathbf{xy}}}}{\mu_g} + \frac{L_{g_{\mathbf{yy}}}C_{\mathbf{xy}}}{\mu_g^2}\right]$ and all the constants are as defined in Assumption1.

(b) The inner optimal solution $\mathbf{y}^\star(\mathbf{x})$ is $\frac{C_{\mathbf{xy}}}{\mu_g}$-Lipschitz continuous in $\mathbf{x}$, which implies that for any $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$, it holds that $\|\mathbf{y}^\star(\mathbf{x}_1) - \mathbf{y}^\star(\mathbf{x}_2)\| \le \frac{C_{\mathbf{xy}}}{\mu_g}\|\mathbf{x}_1 - \mathbf{x}_2\|$.

(c) The gradient of outer objective ∇Q is L_Q-Lipschitz continuous in $\mathbf{x}$, which implies that for any $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$, it holds that $\|Q(\mathbf{x}_1) - Q(\mathbf{x}_2)\| \le L_Q\|\mathbf{x}_2 - \mathbf{x}_1\|$ where $L_Q := \frac{(L_{f_y}+L)C_{xy}}{\mu_g} + L_{f_x} + C_y\left[\frac{L_{gxy}C_y}{\mu_g} + \frac{L_{gyy}C_xy}{\mu_g^2}\right]$.

Lemma 5 *[(Khanduri et al. 2021b), Lemma 4.1] Suppose Assumption1 holds, and the gradient estimate $h(\mathbf{x},\mathbf{y};\theta,\xi)$ is constructed with k number of samples using (4), then*
(a) for any $\mathbf{x} \in \mathcal{X}$ and $\mathbf{y}_1, \mathbf{y}_2 \in \mathbb{R}^n$, we have

$$\mathbb{E}_t\|h(\mathbf{x},\mathbf{y}_1;\theta_t,\xi_t) - h(\mathbf{x},\mathbf{y}_2;\theta_t,\xi_t)\|^2 \le L_k\mathbb{E}_t\|\mathbf{y}_1 - \mathbf{y}_2\|^2 \quad (22)$$

(b) for any $\mathbf{y} \in \mathbb{R}^n$ and $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$, we have

$$\mathbb{E}_t\|h(\mathbf{x}_1,\mathbf{y};\theta_t,\xi_t) - h(\mathbf{x}_2,\mathbf{y};\theta_t,\xi_t)\|^2 \le L_k\mathbb{E}_t\|\mathbf{x}_1 - \mathbf{x}_2\|^2;$$

$$L_k = 2L_{f_\mathbf{x}}^2 + \frac{6k[(L_g-\mu_g)^2(C_{g_\mathbf{xy}}^2L_{f_\mathbf{y}}^2 + C_{f_\mathbf{y}}^2L_{g_\mathbf{xy}}^2) + k^2C_{g_\mathbf{xy}}^2C_{f_\mathbf{y}}^2L_{g_\mathbf{yy}}^2]}{\mu_g(2L_g - \mu_g)}.$$

Lemma 6 *[Lemma 2 (Akhtar and Rajawat 2021)] Let ψ_t be a sequence of real numbers which satisfy*

$$\psi_{t+1} = \left(1 - \frac{c_1}{(t+t_0)^{r_1}}\right)\psi_t + \frac{c_2}{(t+t_0)^{r_2}} \quad (23)$$

for some $r_1 \in (0,1]$ such that $r_1 \le r_2 \le 2r_1$, $c_1 > 1$, and $c_2 \ge 0$. Then, for $c = \max\{\psi_1(t_0+1)^{r_2-r_1}, \frac{c_2}{c_1-1}\}$, ψ_{t+1} would converge to zero at the following rate

$$\psi_{t+1} \le \frac{c}{(t+t_0+1)^{r_2-r_1}}, \quad (24)$$

Lemma 7 *Under Assumption 1, consider the estimator defined in (4), then*

(i) define bias $B_t := \mathbb{E}[h(\mathbf{x}_t,\mathbf{y}_t;\theta_t,\xi_t)] - \nabla S(\mathbf{x}_t,\mathbf{y}_t)$, it holds that we have,

$$\|B_t\| \le (C_{\mathbf{xy}}C_\mathbf{y}/\mu_g)\left(1 - (\mu_g/L_g)\right)^k, \quad (25)$$

$$\mathbb{E}\|h(\mathbf{x}_t,\mathbf{y}_t;\theta_t,\xi_t) - \nabla S(\mathbf{x}_t,\mathbf{y}_t) - B_t\|^2 \le \sigma_f^2, \quad (26)$$

where $\sigma_f^2 = \sigma_\mathbf{x}^2 + \frac{3}{\mu_g^2}\left[(\sigma_\mathbf{y}^2 + C_\mathbf{y}^2)(\sigma_{\mathbf{xy}}^2 + 2C_{\mathbf{xy}}^2) + \sigma_\mathbf{y}^2C_{\mathbf{xy}}^2\right]$.

(ii) For $t \ge 0$, it is possible select k (required to approximate the Hessian inverse in (5)) such that $\|B_t\| \le \beta_t$ where $\beta_t \le ct^a$ for some constant c and $a > 0$.

Proof: For proof of Lemma (7)(i) see [Lemma 11, (Hong et al. 2020)]. The proof Lemma (7)(ii) is straightforward. From (25) we have $\beta_t = \left(\mathcal{O}(1-\mu_g/L_g)^k\right)$. Now on setting $k = \mathcal{O}(log(t))$ we can get the required condition as $\beta_t \le ct^a$. It shows that with proper selection of k, we can make the bias to decay polynomially to zero.

Proof of Lemma 1

From the update step 2 of Algorithm 1, we can write

$$\begin{aligned}&\mathbb{E}_t[\|\mathbf{y}_t - \mathbf{y}^\star(\mathbf{x}_{t-1})\|^2] \quad (27)\\ &= \mathbb{E}_t[\|\mathbf{y}_{t-1} - \delta_t\nabla_\mathbf{y}g(\mathbf{x}_{t-1},\mathbf{y}_{t-1},\xi_t) - \mathbf{y}^\star(\mathbf{x}_{t-1})\|^2].\end{aligned}$$

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

$$\begin{aligned}
&\mathbb{E}_t[\|\mathbf{y}_t - \mathbf{y}^\star(\mathbf{x}_{t-1})\|^2] \\
&= \mathbb{E}_t[\|\mathbf{y}_{t-1} - \delta_t \nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \xi_t) - \mathbf{y}^\star(\mathbf{x}_{t-1})\|^2] \\
&= \mathbb{E}_t[\|\mathbf{y}_{t-1} - \mathbf{y}^\star(\mathbf{x}_{t-1})\|^2] \\
&\quad + \delta_t^2 \mathbb{E}_t[\|\nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \xi_t)\|^2] \\
&\quad - 2\delta_t \mathbb{E}_t[\langle \mathbf{y}_{t-1} - \mathbf{y}^\star(\mathbf{x}_{t-1}), \nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \xi_t)\rangle] \qquad (28) \\
&\leq \mathbb{E}_t[\|\mathbf{y}_{t-1} - \mathbf{y}^\star(\mathbf{x}_{t-1})\|^2] \\
&\quad + \delta_t^2 \mathbb{E}_t[\|\nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \xi_t)\|^2] \\
&\quad - 2\delta_t \mu_g \mathbb{E}_t[\|\mathbf{y}_{t-1} - \mathbf{y}^\star(\mathbf{x}_{t-1})\|^2] \qquad (29) \\
&= (1 - 2\delta_t \mu_g) \mathbb{E}_t[\|\mathbf{y}_{t-1} - \mathbf{y}^\star(\mathbf{x}_{t-1})\|^2] \\
&\quad + \delta_t^2 \mathbb{E}_t[\|\nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \xi_t)\|^2] \qquad (30)
\end{aligned}$$

here (28) comes from the fact that $\mathbb{E}_t[\nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \xi_t)] = \nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}_{t-1})$, while (29) comes from using the strong convexity property of function g. Now consider the last term $\mathbb{E}_t\|\nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \xi_t)\|^2$ of (30):

$$\begin{aligned}
&\mathbb{E}_t[\|\nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \xi_t)\|^2] \qquad (31) \\
&= \mathbb{E}_t\|\nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \xi_t) + \nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}) \\
&\quad - \nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}_{t-1})\|^2 \\
&\leq 2\mathbb{E}_t[\|\nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \xi_t) - \nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}_{t-1})\|^2] \\
&\quad + 2\|\nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}_{t-1})\|^2, \qquad (32)
\end{aligned}$$

where we use the inequality $\|\mathbf{a} + \mathbf{b}\|^2 \leq 2\|\mathbf{a}\|^2 + 2\|\mathbf{b}\|^2$. From Assumption 1, we can further upper bound (31) as

$$\begin{aligned}
&\mathbb{E}_t[\|\nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \xi_t)\|^2] \\
&\leq 2\sigma_g^2(1 + \|\nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}_{t-1})\|^2) + 2\|\nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}_{t-1})\|^2 \\
&= 2\sigma_g^2 + 2(1 + \sigma_g^2)\|\nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}_{t-1})\|^2 \qquad (33) \\
&\leq 2\sigma_g^2 + 2(1+\sigma_g^2)\|\nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}) - \nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}^\star(\mathbf{x}_{t-1}))\|^2 \\
&\leq 2\sigma_g^2 + 2(1 + \sigma_g^2) L_g^2 \|\mathbf{y}_{t-1} - \mathbf{y}^\star(\mathbf{x}_{t-1})\|^2 \qquad (34)
\end{aligned}$$

where we used the fact that $\nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}^\star(\mathbf{x}_{t-1})) = 0$. Substituting the upper bound in (33) in (30) we obtain

$$\begin{aligned}
\mathbb{E}_t[\|\mathbf{y}_t - \mathbf{y}^\star(\mathbf{x}_{t-1})\|^2] &\leq [(1 - 2\delta_t \mu_g) \\
&\quad + 2\delta_t^2(1 + \sigma_g^2) L_g^2] \mathbb{E}_t[\|\mathbf{y}_{t-1} - \mathbf{y}^\star(\mathbf{x}_{t-1})\|^2] + 2\delta_t^2 \sigma_g^2 \\
&\leq (1 - \delta_t \mu_g) \mathbb{E}_t[\|\mathbf{y}_{t-1} - \mathbf{y}^\star(\mathbf{x}_{t-1})\|^2] + 2\delta_t^2 \sigma_g^2. \qquad (35)
\end{aligned}$$

The last inequality in (35) is obtained by selecting δ_t such that $2\delta_t(1 + \sigma_g^2) L_g^2 \leq \mu_g$. To proceed next, we use Young's inequality to bound the term $\mathbb{E}_t[\|\mathbf{y}_t - \mathbf{y}^\star(\mathbf{x}_t)\|^2]$ in (35) as

$$\begin{aligned}
&\mathbb{E}_t[\|\mathbf{y}_t - \mathbf{y}^\star(\mathbf{x}_t)\|^2] \leq \left(1 + \tfrac{1}{\alpha}\right) \mathbb{E}_t[\|\mathbf{y}_t - \mathbf{y}^\star(\mathbf{x}_{t-1})\|^2] \qquad (36) \\
&\quad + (1 + \alpha) \mathbb{E}_t[\|\mathbf{y}^\star(\mathbf{x}_t) - \mathbf{y}^\star(\mathbf{x}_{t-1})\|^2] \\
&\leq \left(1 + \tfrac{1}{\alpha}\right) \mathbb{E}_t[\|\mathbf{y}_t - \mathbf{y}^\star(\mathbf{x}_{t-1})\|^2] \\
&\quad + (1 + \alpha) \left(\frac{C_{\mathbf{xy}}}{\mu_g}\right)^2 \mathbb{E}_t \|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2] \\
&\leq \left(1 + \tfrac{1}{\alpha}\right) \mathbb{E}_t[\|\mathbf{y}_t - \mathbf{y}^\star(\mathbf{x}_{t-1})\|^2] + (1+\alpha) \left(\frac{C_{\mathbf{xy}}}{\mu_g}\right)^2 \eta_{t-1}^2 D^2
\end{aligned}$$

where the second inequality comes from Lemma 4(b), and the last inequality comes from the update equation and the compactness of the domain $\mathcal{X}$. Utilizing (35) into (36), we get

$$\begin{aligned}
&\mathbb{E}_t[\|\mathbf{y}_t - \mathbf{y}^\star(\mathbf{x}_t)\|^2] \\
&\leq \left(1 + \frac{1}{\alpha}\right)(1 - \delta_t \mu_g) \mathbb{E}_t[\|\mathbf{y}_{t-1} - \mathbf{y}^\star(\mathbf{x}_{t-1})\|^2] \\
&\quad + \left(1 + \frac{1}{\alpha}\right) 2\delta_t^2 \sigma_g^2 + (1 + \alpha) \left(\frac{C_{\mathbf{xy}}}{\mu_g}\right)^2 \eta_{t-1}^2 D^2. \qquad (37)
\end{aligned}$$

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

$$\begin{aligned}
&\mathbb{E}_t[\|\mathbf{y}_t - \mathbf{y}^\star(\mathbf{x}_t)\|^2] \leq \left(1 - \frac{\delta_t \mu_g}{2}\right) \mathbb{E}_t \|\mathbf{y}_{t-1} - \mathbf{y}^\star(\mathbf{x}_{t-1})\|^2 \\
&\quad + \frac{2 - \delta_t \mu_g}{\delta_t \mu_g} \left(\frac{C_{\mathbf{xy}}}{\mu_g}\right)^2 \eta_{t-1}^2 D^2 + \left(1 + \frac{1}{\alpha}\right) 2\delta_t^2 \sigma_g^2 \\
&\leq \left(1 - \tfrac{\delta_t \mu_g}{2}\right) \mathbb{E}_t \|\mathbf{y}_{t-1} - \mathbf{y}^\star(\mathbf{x}_{t-1})\|^2 + \tfrac{2\eta_{t-1}^2}{\delta_t \mu_g} \left(\tfrac{C_{\mathbf{xy}}}{\mu_g}\right)^2 D^2 \\
&\quad + 4\delta_t^2 \sigma_g^2, \qquad (38)
\end{aligned}$$

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 δ_t is chosen such that $\delta_t \leq \frac{2}{3\mu_g}$ giving $1 + \frac{1}{\alpha} \leq 2$. In Corollary 1 we will see that our choice of step sizes satisfies these conditions.

To prove part (ii), we start with writing Lemma 1 (i) for $t = t + 1$ and set $\delta_t = \frac{2a_0}{t^q}$ where $a_0 = \min\{\frac{1}{3\mu_g}, \frac{\mu_g}{2(1+\sigma_g^2)L_g^2}\}$ and $\eta_t = \frac{2}{(t+1)^{\frac{3q}{2}}}$, which gives

$$\begin{aligned}
&\mathbb{E}_t \|\mathbf{y}_{t+1} - \mathbf{y}^\star(\mathbf{x}_{t+1})\|^2 \leq \left(1 - \tfrac{2a_0}{(t+1)^q}\right) \mathbb{E}_t \|\mathbf{y}_t - \mathbf{y}^\star(\mathbf{x}_t)\|^2 \\
&\quad + \frac{2}{(t+1)^{3q-q}} \left(\frac{C_{\mathbf{xy}}}{\mu_g}\right)^2 D^2 + \frac{16 a_0^2}{(t+1)^{2q}} \sigma_g^2 \qquad (39) \\
&= \left(1 - \tfrac{2a_0}{(t+1)^q}\right) \mathbb{E}_t[\|\mathbf{y}_t - \mathbf{y}^\star(\mathbf{x}_t)\|^2] + \tfrac{2(C_{\mathbf{xy}}/\mu_g)^2 D^2 + 16 a_0^2 \sigma_g^2}{(t+1)^{2q}}.
\end{aligned}$$

Note such selection of δ_t ensures that the conditions $2\delta_t(1 + \sigma_g^2)L_g^2 \leq \mu_g$ and $\delta_t \leq \frac{2}{3\mu_g}$ required in Lemma 1 are satisfied. Now taking full expectation and using Lemma 6 we get

$$\mathbb{E}[\|\mathbf{y}_t - \mathbf{y}^\star(\mathbf{x}_t)\|] \leq \frac{b_1}{(t+1)^q}, \qquad (40)$$

where $b_1 = \max\{2^q \left\|\mathbf{y}_1 - \mathbf{y}^\star(\mathbf{x}_1)\right\|^2, (2(C_{\mathbf{xy}}/\mu_g)^2 D^2 + 16a_0^2\sigma_g^2)/(2a_0 - 1)\}$.

Proof of Lemma 2

Starting with update equation (8) and employing Lemma 3 we can write

$$\begin{aligned}
&\mathbb{E}_t[\|\mathbf{d}_t - \nabla S(\mathbf{x}_t, \mathbf{y}_t) - B_t\|^2 \qquad (41)\\
&\leq (1-\rho_t)^2 \mathbb{E}_t[\|(\mathbf{d}_{t-1} - \nabla S(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}) - B_{t-1})\|]\\
&+ 2(1-\rho_t)^2 \mathbb{E}_t \|h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t) - h(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}; \theta_t, \xi_t)\|^2\\
&+ 2\rho_t^2 \mathbb{E}_t[\|h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t) - \nabla S(\mathbf{x}_t, \mathbf{y}_t) - B_t)\|^2]\\
&\leq (1-\rho_t)^2 \mathbb{E}_t[\|(\mathbf{d}_{t-1} - \nabla S(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}) - B_{t-1})\|]\\
&+ 2\mathbb{E}_t[\|h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t) - h(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}; \theta_t, \xi_t)\|^2 + 2\rho_t^2\sigma_f^2,
\end{aligned}$$

here the last inequality is obtained using (26) and the fact that $(1-\rho_t^2) \leq 1$. Now we introduce $h(\mathbf{x}_t, \mathbf{y}_{t-1}; \theta_t, \xi_t)$ and bound the second term of RHS of (41) as

$$\begin{aligned}
&\mathbb{E}_t[\|h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t) - h(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}; \theta_t, \xi_t)\|^2]\\
&= \mathbb{E}_t \|h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t) - h(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}; \theta_t, \xi_t)\\
&\quad + h(\mathbf{x}_t, \mathbf{y}_{t-1}; \theta_t, \xi_t) - h(\mathbf{x}_t, \mathbf{y}_{t-1}; \theta_t, \xi_t)\|^2\\
&\overset{(a)}{\leq} 2\mathbb{E}_t[\|h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t) - h(\mathbf{x}_t, \mathbf{y}_{t-1}; \theta_t, \xi_t)\|^2\\
&+ 2\mathbb{E}_t[\|h(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}; \theta_t, \xi_t) - h(\mathbf{x}_t, \mathbf{y}_{t-1}; \theta_t, \xi_t)\|^2\\
&\overset{(b)}{\leq} 2L_k \mathbb{E}_t \|\mathbf{y}_t - \mathbf{y}_{t-1}\|^2 + 2L_k \mathbb{E}_t \|\mathbf{x}_{t-1} - \mathbf{x}_t\|^2\\
&\overset{(c)}{\leq} 2L_k \delta_t^2 \mathbb{E}_t \left\|\nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \xi_t)\right\|^2 + 2L_k \eta_{t-1}^2 D^2\\
&\overset{(d)}{\leq} 2L_k \delta_t^2 L_g^2 + 2L_k \eta_{t-1}^2 D^2, \qquad (42)
\end{aligned}$$

here (a) comes from simple norm property, (b) comes from Lemma 5, (c) comes from the update equation while (d) comes from Assumption 1 and from the compactness of the set. Using (42) in (41), we get the desired expression.

Proof Corollary 1

We start with setting $\delta_t = \frac{2a_0}{(t)^q}$, $\eta_t = \frac{2}{(t+1)^{\frac{3q}{2}}}$ and $\rho_t = \frac{2}{(t)^q}$ in Lemma 2 to obtain

$$\begin{aligned}
&\mathbb{E}_t[\|\mathbf{d}_{t+1} - \nabla S(\mathbf{x}_{t+1}, \mathbf{y}_{t+1}) - B_{t+1}\|^2] \qquad (43)\\
&\leq \left(1 - \frac{2}{(t+1)^q}\right) \mathbb{E}_t[\|(\mathbf{d}_t - \nabla S(\mathbf{x}_t, \mathbf{y}_t) - B_t)\|^2]\\
&+ \frac{16L_k L_g^2 + 16L_k D^2 + 8\sigma_f^2}{(t+1)^{2q}}, \qquad (44)
\end{aligned}$$

here the last inequality is obtained using the fact $1/(t+1)^{3q} \leq 1/(t+1)^{2p}$. Application of Lemma 6 gives

$$\mathbb{E}_t[\|\mathbf{d}_t - \nabla S(\mathbf{x}_t, \mathbf{y}_t) - B_t\|^2] \leq \frac{b_2}{(t+2)^q}, \qquad (45)$$

where $b_2 = \max\{2^q \left\|\mathbf{d}_1 - \nabla S(\mathbf{x}_1, \mathbf{y}_1) - B_1\right\|^2, 8(2L_k L_g^2 + L_k D^2 + \sigma_f^2)\} = 8(2L_k L_g^2 + L_k D^2 + \sigma_f^2)$. As, we have initialize $\mathbf{d}_1 = h(\mathbf{x}_1, \mathbf{y}_1; \theta_1, \xi_1)$ we can use the bound $\left\|\mathbf{d}_1 - \nabla S(\mathbf{x}_1, \mathbf{y}_1) - B_1\right\|^2 = \|h(\mathbf{x}_1, \mathbf{y}_1; \theta_1, \xi_1) - \nabla S(\mathbf{x}_1, \mathbf{y}_1) - B_1\|^2 \leq \sigma_f^2$. Next, we can bound the term $\mathbb{E}\left\|\nabla Q(\mathbf{x}_t) - \mathbf{d}_t\right\|^2$ as follows

$$\begin{aligned}
&\mathbb{E}\left\|\nabla Q(\mathbf{x}_t) - \mathbf{d}_t\right\|^2 \qquad (46)\\
&= \mathbb{E}\left\|\nabla Q(\mathbf{x}_t) - \mathbf{d}_t + B_t + \nabla S(\mathbf{x}_t, \mathbf{y}_t) - B_t - \nabla S(\mathbf{x}_t, \mathbf{y}_t)\right\|^2\\
&\leq 3\mathbb{E}\left\|\nabla Q(\mathbf{x}_t) - \nabla S(\mathbf{x}_t, \mathbf{y}_t)\right\|^2 + 3\left\|B_t\right\|^2\\
&\quad + 3\mathbb{E}\left\|\nabla S(\mathbf{x}_t, \mathbf{y}_t) + B_t - \mathbf{d}_t\right\|^2\\
&\leq 3\mathbb{E}\left\|\mathbf{y}^\star(\mathbf{x}_t) - \mathbf{y}_t\right\|^2 + 3\beta_t^2 + \frac{3b_2}{(t+1)^q}\\
&\leq \frac{3b_1}{(t+1)^q} + \frac{3b_3}{(t+1)^q} + \frac{3b_2}{(t+1)^q} := \frac{C_1}{(t+1)^q}, \qquad (47)
\end{aligned}$$

here second inequality comes from simple norm property, while the third inequality is obtained using Lemma (4)(a) on the first term, Lemma 7 on the second term, and (45) on the third term. The last inequality comes from (40) and using $\beta_t \leq \frac{C_{\mathbf{xy}} C_{\mathbf{y}}}{\mu_g (t+1)^q} := \frac{b_3}{(t+1)^q}$ and the constant $C_1 = 3(b_1 + b_2 + b_3)$ is defined as $C_1 = 3(\max\{2^q \left\|\mathbf{y}_1 - \mathbf{y}^\star(\mathbf{x}_1)\right\|^2, (2(C_{\mathbf{xy}}/\mu_g)^2 D^2 + 16a_0^2\sigma_g^2)/(2a_0 - 1)\} + 8(2L_k L_g^2 + L_k D^2 + \sigma_f^2) + \frac{C_{\mathbf{xy}} C_{\mathbf{y}}}{\mu_g})$.

Proof of Theorem 1

From the initialization of variable $\mathbf{x}$, we have $\mathbf{x}_1 \in \mathcal{X}$. Also since we obtain $\mathbf{s}_t$ solving a linear minimization problem over the set $\mathcal{X}$, we have $\mathbf{s}_t \in \mathcal{X}$. Thus, $\mathbf{x}_{t+1}$ which is a convex combination of $\mathbf{x}_t$ and $\mathbf{s}_t$, i.e. $\mathbf{x}_{t+1} = (1 - \eta_{t+1})\mathbf{x}_t + \eta_{t+1}\mathbf{s}_t$ will also lie in the set $\mathcal{X}$. Hence $\mathbf{x}_{T+1} \in \mathcal{X}$ and $\hat{\mathbf{x}} \in \mathcal{X}$. Now, starting with definition of $Q(\cdot)$, we have $Q(\mathbf{x}) = \mathbb{E}_\theta[f(\mathbf{x}, \mathbf{y}^\star(\mathbf{x}); \theta)]$. Also note that we have set $k = \frac{qL_g}{\mu_g}(\log(1+t))$, this ensures that the condition $\beta_t \leq \frac{C_{\mathbf{xy}} C_{\mathbf{y}}}{\mu_g (t+1)^q}$ required in the analysis of Corollary (1) is satisfied. Hence, we can use results from Corollary (1) with $q = 2/3$ for convex case.

Using the smoothness assumption of Q we can write

$$\begin{aligned}
&Q(\mathbf{x}_{t+1}) - Q(\mathbf{x}_t)\\
&\leq \langle \nabla Q(\mathbf{x}_t), \mathbf{x}_{t+1} - \mathbf{x}_t \rangle + \frac{L_Q}{2}\left\|\mathbf{x}_{t+1} - \mathbf{x}_t\right\|^2\\
&= \eta_t \langle \nabla Q(\mathbf{x}_t), \mathbf{s}_t - \mathbf{x}_t \rangle + \frac{L_Q \eta_t^2}{2}\left\|\mathbf{s}_t - \mathbf{x}_t\right\|^2, \qquad (48)
\end{aligned}$$

where $L_Q = \frac{(L_{f_{\mathbf{y}}} + L) C_{\mathbf{xy}}}{\mu_g} + L_{f_{\mathbf{x}}} + C_{\mathbf{y}}\left[\frac{L_{g_{\mathbf{xy}}} C_{\mathbf{y}}}{\mu_g} + \frac{L_{g_{\mathbf{yy}}} C_{\mathbf{xy}}}{\mu_g^2}\right]$ (see Lemma 4). Here, in the last expression we have replace term $x_{t+1} - \mathbf{x}_t = \eta_t(\mathbf{s}_t - \mathbf{x}_t)$. Now adding and subtracting $\eta_t \langle \mathbf{d}_t, \mathbf{s}_t - \mathbf{x}_t \rangle$ in (48) we get

$$\begin{aligned}
Q(\mathbf{x}_{t+1}) &\leq Q(\mathbf{x}_t) + \eta_t \langle \nabla Q(\mathbf{x}_t) - \mathbf{d}_t, \mathbf{s}_t - \mathbf{x}_t \rangle\\
&\quad + \eta_t \langle \mathbf{d}_t, \mathbf{x}^\star - \mathbf{x}_t \rangle + \frac{L_Q \eta_t^2 D^2}{2}, \qquad (49)
\end{aligned}$$

here in last the inequality is obtained using optimality of s_t. Now introducing $\eta_t \langle \nabla Q(\mathbf{x}_t), \mathbf{x}^\star - \mathbf{x}_t \rangle$ in RHS of (49) and

regrouping the terms we obtain

$$
\begin{aligned}
&Q(\mathbf{x}_{t+1}) - \frac{L_Q \eta_t^2 D^2}{2} \\
&\leq Q(\mathbf{x}_t) + \eta_t \langle \nabla Q(\mathbf{x}_t) - \mathbf{d}_t, \mathbf{s}_t - \mathbf{x}^\star \rangle \\
&+ \eta_t \langle \nabla Q(\mathbf{x}_t), \mathbf{x}^\star - \mathbf{x}_t \rangle \\
&\leq Q(\mathbf{x}_t) + \eta_t D \left\| \nabla Q(\mathbf{x}_t) - \mathbf{d}_t \right\| + \eta_t \langle \nabla Q(\mathbf{x}_t), \mathbf{x}^\star - \mathbf{x}_t \rangle \\
&\leq Q(\mathbf{x}_t) + \eta_t D \left\| \nabla Q(\mathbf{x}_t) - \mathbf{d}_t \right\| - \eta_t (Q(\mathbf{x}_t) - Q(\mathbf{x}^\star)),
\end{aligned} \tag{50}
$$

here in the second inequality we use bound $\eta_t \langle \nabla Q(\mathbf{x}_t) - \mathbf{d}_t, \mathbf{s}_t - \mathbf{x}^\star \rangle \leq \eta_t \left\| \nabla Q(\mathbf{x}_t) - \mathbf{d}_t \right\| \left\| \mathbf{s}_t - \mathbf{x}^\star \right\| \leq \eta_t D \left\| \nabla Q(\mathbf{x}_t) - \mathbf{d}_t \right\|$ and in last inequality we used the bound $\langle \nabla Q(\mathbf{x}_t), \mathbf{x}^\star - \mathbf{x}_t \rangle \leq Q(\mathbf{x}^\star) - Q(\mathbf{x}_t)$. Subtracting $Q(\mathbf{x}^\star)$, taking expectation and using $\mathbb{E} \left\| X \right\| \leq \sqrt{\mathbb{E} \left\| X \right\|^2}$ we get

$$
\begin{aligned}
\mathbb{E}[Q(\mathbf{x}_{t+1}) - Q(\mathbf{x}^\star)] &\leq (1 - \eta_t) \mathbb{E}[Q(\mathbf{x}_t) - Q(\mathbf{x}^\star)] \\
&+ \eta_t D \sqrt{\mathbb{E} \left\| \nabla Q(\mathbf{x}_t) - \mathbf{d}_t \right\|^2} + \frac{L_Q \eta_t^2 D^2}{2}.
\end{aligned} \tag{51}
$$

Further, setting $q = 2/3$ hence, $\eta_t = \frac{2}{t+1}$ and using Corollary 1, we can bound the second term of (51) $\eta_t D \sqrt{\mathbb{E} \left\| \nabla Q(\mathbf{x}_t) - \mathbf{d}_t \right\|^2} \leq \frac{2D\sqrt{C_1}}{(t+1)^{4/3}}$. which gives

$$
\begin{aligned}
\mathbb{E}[Q(\mathbf{x}_{t+1}) - Q(\mathbf{x}^\star)] &\leq \left(1 - \frac{2}{t+1}\right) \mathbb{E}[Q(\mathbf{x}_t) - Q(\mathbf{x}^\star)] \\
&+ \frac{2D\sqrt{C_1}}{(t+1)^{\frac{4}{3}}} + \frac{2L_Q D^2}{(t+1)^2}.
\end{aligned} \tag{52}
$$

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

$$
\begin{aligned}
&t(t+1)\mathbb{E}[Q(\mathbf{x}_{t+1}) - Q(\mathbf{x}^\star)] \\
&\leq t(t-1)\mathbb{E}[Q(\mathbf{x}_t) - Q(\mathbf{x}^\star)] + \frac{2tD\sqrt{C_1}}{(t+1)^{\frac{1}{3}}} + \frac{2tL_Q D^2}{t+1} \\
&\leq t(t-1)\mathbb{E}[Q(\mathbf{x}_t) - Q(\mathbf{x}^\star)] + 2D\sqrt{C_1}(t+1)^{\frac{2}{3}} + 2L_Q D^2,
\end{aligned} \tag{53}
$$

Summing for $t = 1, 2, \cdots, T$ and rearranging we get

$$
\begin{aligned}
&\mathbb{E}[Q(\mathbf{x}_{T+1}) - Q(\mathbf{x}^\star)] \\
&\leq \frac{1}{T(T+1)} \left(\frac{6}{5} D\sqrt{C_1}(T+1)^{\frac{5}{3}} + 2L_Q D^2 T \right) \\
&\leq \frac{12D\sqrt{C_1}}{5(T+1)^{\frac{1}{3}}} + \frac{2L_Q D^2}{(T+1)},
\end{aligned} \tag{54}
$$

here we use the fact that $\sum_{t=1}^{T} (t+1)^{2/3} \leq \frac{3}{5}(T+1)^{5/3}$.

References

Akhtar, Z.; and Rajawat, K. 2021. Momentum based projection free stochastic optimization under affine constraints. In *2021 American Control Conference (ACC)*, 2619–2624. IEEE.

Akhtar, Z.; and Rajawat, K. 2022. Zeroth and First Order Stochastic Frank-Wolfe Algorithms for Constrained Optimization. *IEEE Transactions on Signal Processing*.

Borsos, Z.; Mutny, M.; and Krause, A. 2020. Coresets via bilevel optimization for continual learning and streaming. *Advances in Neural Information Processing Systems*, 33: 14879–14890.

Chen, T.; Sun, Y.; and Yin, W. 2021. A Single-Timescale Stochastic Bilevel Optimization Method. *arXiv preprint arXiv:2102.04671*.

Crockett, C.; and Fessler, J. A. 2021. Motivating Bilevel Approaches To Filter Learning: A Case Study. In *2021 IEEE International Conference on Image Processing (ICIP)*, 2803–2807. IEEE.

Cutkosky, A.; and Orabona, F. 2019. Momentum-based variance reduction in non-convex sgd. *arXiv preprint arXiv:1905.10018*.

Franceschi, L.; Frasconi, P.; Salzo, S.; Grazzi, R.; and Pontil, M. 2018. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*, 1568–1577. PMLR.

Ghadimi, S.; and Wang, M. 2018. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*.

Hazan, E.; and Luo, H. 2016. Variance-reduced and projection-free stochastic optimization. In *International Conference on Machine Learning*, 1263–1271. PMLR.

Hong, M.; Wai, H.-T.; Wang, Z.; and Yang, Z. 2020. A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic. *arXiv preprint arXiv:2007.05170*.

Huang, F.; and Huang, H. 2021. Enhanced Bilevel Optimization via Bregman Distance. *arXiv preprint arXiv:2107.12301*.

Jaggi, M. 2013. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *International Conference on Machine Learning*, 427–435. PMLR.

Ji, K.; Yang, J.; and Liang, Y. 2020. Bilevel Optimization: Nonasymptotic Analysis and Faster Algorithms. *arXiv preprint arXiv:2010.07962*.

Khanduri, P.; Zeng, S.; Hong, M.; Wai, H.-T.; Wang, Z.; and Yang, Z. 2021a. A Momentum-Assisted Single-Timescale Stochastic Approximation Algorithm for Bilevel Optimization. *arXiv e-prints*, arXiv–2102.

Khanduri, P.; Zeng, S.; Hong, M.; Wai, H.-T.; Wang, Z.; and Yang, Z. 2021b. A Near-Optimal Algorithm for Stochastic Bilevel Optimization via Double-Momentum. *arXiv preprint arXiv:2102.07367*.

McRae, A. D.; and Davenport, M. A. 2021. Low-rank matrix completion and denoising under Poisson noise. *Information and Inference: A Journal of the IMA*, 10(2): 697–720.

Mokhtari, A.; Hassani, H.; and Karbasi, A. 2020. Stochastic conditional gradient methods: From convex minimization to submodular maximization. *Journal of Machine Learning Research*, 21(105): 1–49.

Rajeswaran, A.; Finn, C.; Kakade, S.; and Levine, S. 2019. Meta-learning with implicit gradients. *arXiv preprint arXiv:1909.04630*.

Reddi, S. J.; Sra, S.; Póczos, B.; and Smola, A. 2016. Stochastic frank-wolfe methods for nonconvex optimization. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 1244–1251. IEEE.

Xie, J.; Shen, Z.; Zhang, C.; Wang, B.; and Qian, H. 2020. Efficient Projection-Free Online Methods with Stochastic Recursive Gradient. In *AAAI*, 6446–6453.

Yang, J.; Ji, K.; and Liang, Y. 2021. Provably Faster Algorithms for Bilevel Optimization. *arXiv preprint arXiv:2106.04692*.

Zhang, H.; Chen, W.; Huang, Z.; Li, M.; Yang, Y.; Zhang, W.; and Wang, J. 2020a. Bi-level actor-critic for multi-agent coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 7325–7332.

Zhang, M.; Shen, Z.; Mokhtari, A.; Hassani, H.; and Karbasi, A. 2020b. One sample stochastic frank-wolfe. In *International Conference on Artificial Intelligence and Statistics*, 4012–4023. PMLR.