An Accelerated Gradient Method for Convex Smooth Simple Bilevel Optimization

Jincheng Cao
ECE Department
UT Austin
jinchengcao@utexas.edu

Ruichen Jiang
ECE Department
UT Austin
rjiang@utexas.edu

Erfan Yazdandoost Hamedani
SIE Department
The University of Arizona
erfany@arizona.edu

Aryan Mokhtari
ECE Department
UT Austin
mokhtari@austin.utexas.edu

Abstract

In this paper, we focus on simple bilevel optimization problems, where we minimize a convex smooth objective function over the optimal solution set of another convex smooth constrained optimization problem. We present a novel bilevel optimization method that locally approximates the solution set of the lower-level problem using a cutting plane approach and employs an accelerated gradient-based update to reduce the upper-level objective function over the approximated solution set. We measure the performance of our method in terms of suboptimality and infeasibility errors and provide non-asymptotic convergence guarantees for both error criteria. Specifically, when the feasible set is compact, we show that our method requires at most $\mathcal{O}(\max\{1/\sqrt{\epsilon_f}, 1/\epsilon_g\})$ iterations to find a solution that is ϵ_f-suboptimal and ϵ_g-infeasible. Moreover, under the additional assumption that the lower-level objective satisfies the r-th Hölderian error bound, we show that our method achieves an iteration complexity of $\tilde{\mathcal{O}}(\max\{\epsilon_f^{-\frac{2r-1}{2r}}, \epsilon_g^{-\frac{2r-1}{2r}}\})$, which matches the optimal complexity of single-level convex constrained optimization when $r = 1$.

1 Introduction

In this paper, we investigate a class of bilevel optimization problems known as simple bilevel optimization, aiming to minimize an upper-level objective function over the solution set of a corresponding lower-level problem. This class has recently gained attention due to its broad applications in continual learning [1], hyper-parameter optimization [2, 3], meta-learning [4, 5], and over-parameterized machine learning [6–8]. Specifically, we focus on the following bilevel optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \; f(\mathbf{x}) \qquad \text{s.t.} \quad \mathbf{x} \in \operatorname*{argmin}_{\mathbf{z}\in\mathcal{Z}} \; g(\mathbf{z}), \tag{1}$$

where, $\mathcal{Z}$ is a convex set, and $f, g : \mathbb{R}^n \to \mathbb{R}$ are convex, continuously differentiable functions on an open set containing $\mathcal{Z}$. We assume that the lower-level objective function g is convex but not strongly convex, so the lower-level problem may have multiple optimal solutions. Throughout the paper, we use $\mathbf{x}^*$ to denote an optimal solution of problem (1). We define $f^* \triangleq f(\mathbf{x}^*)$ and $g^* \triangleq g(\mathbf{x}^*)$, representing the optimal value of problem (1) and the optimal value of the lower-level

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Table 1: Non-asymptotic results on simple bilevel optimization. (①: with a first-order Hölderian error bound assumption on g; ⓡ: with an rth-order ($r \geq 1$) Hölderian error bound assumption on g; Ⓐ: additional assumption implying that the projection onto the sublevel set of f is easy to compute.)

References	Upper level	Lower level		Convergence	
	Objective f	Objective g	Feasible set $\mathcal{Z}$	Upper level	Lower level
a-IRG [12]	Convex, Lipschitz	Convex, Lipschitz	Closed	$\mathcal{O}(1/\epsilon_f^4)$	$\mathcal{O}(1/\epsilon_g^4)$
Bi-SG [13]	Convex, Nonsmooth	Convex, Composite	Closed	$\mathcal{O}(1/\epsilon_f^{\frac{1}{1-\alpha}})$	$\mathcal{O}(1/\epsilon_g^{\frac{1}{\alpha}}), \alpha \in (0.5, 1)$
SEA [14]	Convex	Convex, Smooth	Compact	$\mathcal{O}(1/\epsilon_f^2)$	$\mathcal{O}(1/\epsilon_g^2)$
CG-BiO [6]	Convex, Smooth	Convex, Smooth	Compact	$\mathcal{O}(1/\epsilon_f)$	$\mathcal{O}(1/\epsilon_g)$
R-APM [8]	Convex, Smooth	Convex, Composite	Closed	$\mathcal{O}(1/\epsilon_f)$	$\mathcal{O}(1/\epsilon_g)$
AGM-BiO (Ours)	Convex, Smooth	Convex, Smooth	Compact	$\mathcal{O}(1/\epsilon_f^{0.5})$	$\mathcal{O}(1/\epsilon_g)$
R-APM ① [8]	Convex, Smooth	Convex, Composite	Closed	$\mathcal{O}(1/\epsilon_f^{0.5})$	$\mathcal{O}(1/\epsilon_g^{0.5})$
Bisec-BiO Ⓐ [15]	Convex, Composite	Convex, Composite	Closed	$\tilde{\mathcal{O}}(\max\{1/\epsilon_f^{0.5}, 1/\epsilon_g^{0.5}\})$	
AGM-BiO ① (Ours)	Convex, Smooth	Convex, Smooth	Closed	$\mathcal{O}(1/\epsilon_f^{0.5})$	$\tilde{\mathcal{O}}(1/\epsilon_g^{0.5})$
PB-APG ⓡ [16]	Convex, Composite	Convex, Composite	Compact	$\mathcal{O}(1/\epsilon_f^{0.5r}) + \mathcal{O}(1/\epsilon_g^{0.5})$	
AGM-BiO ⓡ (Ours)	Convex, Smooth	Convex, Smooth	Closed	$\tilde{\mathcal{O}}(1/\epsilon_f^{\frac{2r-1}{2r}})$	$\tilde{\mathcal{O}}(1/\epsilon_g^{\frac{2r-1}{2r}})$

objective g, respectively. This class of problems is referred to as the "simple bilevel problem" [9–11] to distinguish it from more general settings with parameterized lower-level problems.

The main challenge in solving problem (1) is that the feasible set, i.e., the optimal solution set of the lower-level problem, lacks a simple characterization and is not explicitly provided. This makes direct application of projection-based or projection-free methods infeasible, as projecting onto or solving a linear minimization problem over such an implicitly defined feasible set is intractable. Instead, our approach constructs an approximation set with specific properties, serving as a surrogate for the true feasible set. In Section 3, we detail how this set is constructed. Using this technique and building on the projected accelerated gradient method, we establish the best-known complexity bounds for solving problem (1).

To provide context, the best-known complexity bound for achieving an ϵ-accurate solution in single-level convex constrained optimization is $\mathcal{O}(\epsilon^{-0.5})$, as demonstrated in [17]. This optimal bound was achieved using the accelerated proximal method or FISTA (Fast Iterative Shrinkage-Thresholding Algorithm), which also influenced the development of our algorithm. While the literature on bilevel optimization is not as extensive as that for single-level optimization, there have been recent non-asymptotic results for solving this class of problems, which we summarize in Table 1.

Specifically, these results aim to establish convergence rates on the *infeasibility gap* $g(\mathbf{x}_k) - g^*$ and the *suboptimality gap* $f(\mathbf{x}_k) - f^*$ after k iterations. In [12], an iterative regularization-based method demonstrated a convergence rate of $\mathcal{O}(1/k^{0.5-b})$ in terms of suboptimality and a rate of $\mathcal{O}(1/k^b)$ in terms of infeasibility, where $b \in (0, 0.5)$ is a user-defined parameter. Setting $b = 0.25$ to balance these rates requires an iteration complexity of $\mathcal{O}(\max\{1/\epsilon_f^4, 1/\epsilon_g^4\})$ to find a solution that is ϵ_f-optimal and ϵ_g-infeasible. Later, the Bi-Sub-Gradient (Bi-SG) algorithm was proposed in [13] to address convex simple bilevel optimization problems with nonsmooth upper-level objective functions. It showed convergence rates of $\mathcal{O}(1/k^{1-\alpha})$ and $\mathcal{O}(1/k^{\alpha})$ in terms of suboptimality and infeasibility, respectively, where $\alpha \in (0.5, 1)$ serves as a hyper-parameter. Balancing the rates by setting $\alpha = 0.5$ results in an iteration complexity of $\mathcal{O}(\max\{1/\epsilon_f^2, 1/\epsilon_g^2\})$. Additionally, a structure-exploiting method introduced in [14] achieved an iteration complexity of $\mathcal{O}(\max\{1/\epsilon_f^2, 1/\epsilon_g^2\})$ when the upper-level objective is convex and the lower-level objective is convex and smooth. Imposing additional assumptions on the upper-level function, such as smoothness or strong convexity, does not result in faster rates for this method.

Recently, [6] presented a projection-free conditional gradient method (CG-BiO) that uses a cutting plane to approximate the solution set of the lower-level problem. Assuming both upper- and lower-level objective functions are convex and smooth, CG-BiO achieves a complexity of $\mathcal{O}(\max\{1/\epsilon_f, 1/\epsilon_g\})$. Since the suboptimality gap $f(\hat{\mathbf{x}}) - f^*$ may be negative for an infeasible point $\hat{\mathbf{x}}$, a more desirable metric is the *absolute suboptimality gap* $|f(\hat{\mathbf{x}}) - f^*|$. To ensure this, [6] introduced the Hölderian error bound condition on g. Specifically, under the r-th order Hölderian error bound condition, CG-BiO finds a solution $\hat{\mathbf{x}}$ with $|f(\hat{\mathbf{x}}) - f^*| \leq \epsilon_f$ and $g(\hat{\mathbf{x}}) - g^* \leq \epsilon_g$

after $\mathcal{O}(\max\{1/\epsilon_f^r, 1/\epsilon_g\})$ iterations. More recently, [8] introduced the regularized proximal accelerated method (R-APM), which runs the proximal accelerated gradient method on a weighted sum of the upper- and lower-level objective functions. Assuming both functions are convex and smooth, they established a complexity bound of $\mathcal{O}(\max\{1/\epsilon_f, 1/\epsilon_g\})$ to find an (ϵ_f, ϵ_g) solution. This bound is worse than the $\mathcal{O}(\max\{1/\sqrt{\epsilon_f}, 1/\epsilon_g\})$ complexity achieved by our proposed AGM-BiO method, assuming the feasible set $\mathcal{Z}$ is compact. Additionally, [8] showed that when the lower-level objective function g satisfies the weak sharpness property (equivalent to the Hölderian error bound condition with $r = 1$), R-APM finds an (ϵ_f, ϵ_g)-absolute optimal solution after at most $\mathcal{O}(\max\{1/\sqrt{\epsilon_f}, 1/\sqrt{\epsilon_g}\})$ iterations. This result is comparable to our convergence result for AGM-BiO, which considers a more general Hölderian error bound condition.

Contributions. In this paper, we present a novel accelerated gradient-based bilevel optimization method, AGM-BiO, which offers state-of-the-art non-asymptotic guarantees for both suboptimality and infeasibility. At each iteration, AGM-BiO uses a cutting plane to linearly approximate the solution set of the lower-level problem, followed by a variant of the projected accelerated gradient update on the upper-level objective function. Below, we summarize our theoretical guarantees:

- When the feasible set $\mathcal{Z}$ is compact, we show that AGM-BiO finds $\hat{\mathbf{x}}$ that satisfies $f(\hat{\mathbf{x}}) - f^* \leq \epsilon_f$ and $g(\hat{\mathbf{x}}) - g^* \leq \epsilon_g$ within $\mathcal{O}(\max\{1/\sqrt{\epsilon_f}, 1/\epsilon_g\})$ iterations, where f^* is the optimal value of problem (1) and g^* is the optimal value of the lower-level problem.
- With an additional r-th-order ($r \geq 1$) Hölderian error bound assumption on the lower-level problem, AGM-BiO finds $\hat{\mathbf{x}}$ satisfying $f(\hat{\mathbf{x}}) - f^* \leq \epsilon_f$ and $g(\hat{\mathbf{x}}) - g^* \leq \epsilon_g$ within $\tilde{\mathcal{O}}(\max\{\epsilon_f^{-\frac{2r-1}{2r}}, \epsilon_g^{-\frac{2r-1}{2r}}\})$ iterations. Moreover, it achieves the stronger guarantee that $|f(\hat{\mathbf{x}}) - f^*| \leq \epsilon_f$ and $g(\hat{\mathbf{x}}) - g^* \leq \epsilon_g$ within $\tilde{\mathcal{O}}(\max\{\epsilon_f^{-\frac{2r-1}{2}}, \epsilon_g^{-\frac{2r-1}{2r}}\})$ iterations.

These bounds all achieve the best-known complexity bounds in terms of both suboptimality and infeasibility guarantees for the considered settings. All the non-asymptotic results are summarized and compared in Table 1.

Discussions on two concurrent works. The authors in [15] proposed a bisection algorithm with a total operation complexity of $\tilde{\mathcal{O}}(\max\{\epsilon_f^{-0.5}, \epsilon_g^{-0.5}\})$ to find an (ϵ_f, ϵ_g)-optimal solution, assuming the upper-level objective f meets specific criteria. Specifically, Assumption 1(iv) in [15] implies the ability to compute the projection onto the sublevel set of the upper-level function f. However, this assumption may not hold for general functions, such as the mean squared loss function in our over-parameterized regression example in Section 5. In [16], the authors introduced the penalty-based accelerated proximal gradient method (PB-APG) for solving simple bilevel optimization problems with the r-th order Hölderian error bound assumption on the lower-level objective g. Their algorithm, similar to [8], runs the accelerated proximal gradient method on a weighted sum of the upper and lower-level objective functions. PB-APG achieves a complexity of $\mathcal{O}(\epsilon_f^{-0.5r}) + \mathcal{O}(\epsilon_g^{-0.5})$ to find an (ϵ_f, ϵ_g)-optimal solution. The term $\mathcal{O}(1/\epsilon_f^{0.5r})$ can become significantly large as the order of the Hölderian error bound r increases. In contrast, our algorithm, AGM-BiO, avoids this issue, requiring at most $\tilde{\mathcal{O}}(\max\{\epsilon_f^{-\frac{2r-1}{2r}}, \epsilon_g^{-\frac{2r-1}{2r}}\})$ iterations to achieve an (ϵ_f, ϵ_g)-optimal solution. Therefore, regardless of how large r is, the worst-case complexity for AGM-BiO is $\tilde{\mathcal{O}}(\max\{\epsilon_f^{-1}, \epsilon_g^{-1}\})$. Thus, our method achieves a better rate than PB-APG when $r > 1$.

2 Preliminaries

In this section, we state the assumptions and introduce the notions of optimality used in the paper.

2.1 Assumptions and Definitions

We focus on the case where both the upper and lower-level functions f and g are convex and smooth. Formally, we make the following assumptions.

Assumption 2.1. *Let $\|\cdot\|$ be an arbitrary norm on $\mathbb{R}^n$ and $\|\cdot\|_*$ be its dual norm. We assume these conditions hold:*

(i) $\mathcal{Z} \subset \mathbb{R}^n$ is convex and compact with diameter D, i.e., $\|\mathbf{x} - \mathbf{y}\| \leq D$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{Z}$.

(ii) g *is convex and continuously differentiable on an open set containing* $\mathcal{Z}$*, and its gradient is* L_g*-Lipschitz, i.e.,* $\|\nabla g(\mathbf{x}) - \nabla g(\mathbf{y})\|_* \leq L_g\|\mathbf{x} - \mathbf{y}\|$ *for all* $\mathbf{x}, \mathbf{y} \in \mathcal{Z}$*.*

(iii) f *is convex and continuously differentiable and its gradient is Lipschitz with constant* L_f*.*

In this paper, we denote the optimal value and the optimal solution set of the lower-level problem as $g^* \triangleq \min_{\mathbf{z}\in\mathcal{Z}} g(\mathbf{z})$ and $\mathcal{X}_g^* \triangleq \operatorname{argmin}_{\mathbf{z}\in\mathcal{Z}} g(\mathbf{z})$, respectively. By Assumption 2.1, the set $\mathcal{X}_g^*$ is nonempty, compact, and convex, but in general, not a singleton since g could have multiple optimal solutions on $\mathcal{Z}$, as g is only convex but not strongly convex. Moreover, we use f^* to denote the optimal value and $\mathbf{x}^*$ to denote an optimal solution of problem (1).

In the simple bilevel problem, the suboptimality of a solution $\hat{\mathbf{x}}$ is measured by $f(\hat{\mathbf{x}}) - f^*$. Similarly, its infeasibility is indicated by $g(\hat{\mathbf{x}}) - g^*$. To ensure minimal suboptimality and infeasibility, we formally define an (ϵ_f, ϵ_g)-optimal solution as follows.

Definition 2.1. $((\epsilon_f, \epsilon_g)$*-optimal solution). A point* $\hat{\mathbf{x}} \in \mathcal{Z}$ *is* (ϵ_f, ϵ_g)*-optimal for problem* (1) *if* $f(\hat{\mathbf{x}}) - f^* \leq \epsilon_f$ *and* $g(\hat{\mathbf{x}}) - g^* \leq \epsilon_g$*.*

This definition is commonly used in bilevel optimization literature [6–8, 13]. Due to the unique structure of bilevel optimization, it is not guaranteed that $f(\hat{\mathbf{x}}) - f^*$ will always be positive. To address this, we propose using $|f(\hat{\mathbf{x}}) - f^*|$ as the absolute optimal criterion.

Definition 2.2. $((\epsilon_f, \epsilon_g)$*-absolute optimal solution). A point* $\hat{\mathbf{x}} \in \mathcal{Z}$ *is* (ϵ_f, ϵ_g)*-absolute optimal for problem* (1) *if* $|f(\hat{\mathbf{x}}) - f^*| \leq \epsilon_f$ *and* $|g(\hat{\mathbf{x}}) - g^*| \leq \epsilon_g$*.*

3 Algorithm

Before presenting our method, we first introduce a conceptual accelerated gradient method for solving the simple bilevel problem in (1). The first step is to recast it as a constrained optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \; f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{x} \in \mathcal{X}_g^*, \tag{2}$$

where $\mathcal{X}_g^* \triangleq \operatorname{argmin}_{\mathbf{z}\in\mathcal{Z}} \; g(\mathbf{z})$ is the solution set of the lower-level objective. Conceptually, we apply Nesterov's accelerated gradient method (AGM) to achieve a rate of $\mathcal{O}(1/k^2)$ on the upper-level objective f. Several variants of AGM have been proposed; see, e.g., [18]. Here, we consider a variant proposed in [19]. It involves three intertwined sequences of iterates $\{\mathbf{x}_k\}_{k\geq 0}, \{\mathbf{y}_k\}_{k\geq 0}, \{\mathbf{z}_k\}_{k\geq 0}$ and the scalar variables $\{a_k\}_{k\geq 0}$ and $\{A_k\}_{k\geq 0}$. In the first step, we compute the auxiliary iterate $\mathbf{y}_k$ by $\mathbf{y}_k = \frac{A_k}{A_k+a_k}\mathbf{x}_k + \frac{a_k}{A_k+a_k}\mathbf{z}_k$. Then in the second step, we update $\mathbf{z}_{k+1}$ by

$$\mathbf{z}_{k+1} = \Pi_{\mathcal{X}_g^*}(\mathbf{z}_k - a_k\nabla f(\mathbf{y}_k)), \tag{3}$$

where $\Pi_{\mathcal{X}_g^*}(\cdot)$ denotes the Euclidean projection onto the set $\mathcal{X}_g^*$. Finally, in the third step, we compute $\mathbf{x}_{k+1} = \frac{A_k}{A_k+a_k}\mathbf{x}_k + \frac{a_k}{A_k+a_k}\mathbf{z}_{k+1}$ and $A_{k+1} = A_k + a_k$. It can be shown that if the stepsize is selected as $a_k = \frac{k+1}{4L_f}$, then the suboptimality gap $f(\mathbf{x}_k) - f(\mathbf{x}^*)$ of the iterates generated by the method above converges to zero at the optimal rate of $\mathcal{O}(1/k^2)$. In this case, indeed all the iterates are feasible as it is possible to project onto the set $\mathcal{X}_g^*$. However, the conceptual method above is not directly implementable for the simple bilevel problem considered in this paper, as the constraint set $\mathcal{X}_g^*$ is not explicitly given. As a result projection onto the set $\mathcal{X}_g^*$ is not computationally tractable.

To address this issue, we replace the implicit set $\mathcal{X}_g^*$ in (3) with $\mathcal{X}_k$, which can be explicitly characterized, making the Euclidean projection onto $\mathcal{X}_k$ feasible. Additionally, $\mathcal{X}_k$ must encompass the optimal solution set $\mathcal{X}_g^*$. Inspired by the cutting plane approach in [6], we define $\mathcal{X}_k$ as the intersection of $\mathcal{Z}$ and a halfspace:

$$\mathcal{X}_k \triangleq \{\mathbf{z} \in \mathcal{Z} : g(\mathbf{y}_k) + \langle \nabla g(\mathbf{y}_k), \mathbf{z} - \mathbf{y}_k\rangle \leq g_k\}. \tag{4}$$

Here, the auxiliary sequence $\{g_k\}_{k\geq 0}$ should be selected such that $g_k \geq g^*$ and $g_k \to g^*$. One straightforward way to generate this sequence is by applying an accelerated projected gradient method to the lower-level objective g separately. The loss function of the iterates generated by this algorithm can be considered as $\{g_k\}$ for the above halfspace. Note that in this case, it is known that

$$0 \leq g_k - g^* \leq \frac{2L_g\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{(k+1)^2}, \quad \forall k \geq 0. \tag{5}$$

Algorithm 1 Accelerated Gradient Method for Bilevel Optimization (AGM-BiO)

Input: A sequence $\{g_k\}_{k=0}^K$, a scalar $\gamma \in (0, 1]$
Initialization: $A_0 = 0$, $\mathbf{x}_0 = \mathbf{z}_0 \in \mathcal{Z}$
for $k = 0, \ldots, K$ **do**
  Set $a_k = \gamma \frac{k+1}{4L_f}$
  Compute $\mathbf{y}_k = \frac{A_k}{A_k + a_k}\mathbf{x}_k + \frac{a_k}{A_k + a_k}\mathbf{z}_k$
  Compute $\mathbf{z}_{k+1} = \Pi_{\mathcal{X}_k}(\mathbf{z}_k - a_k \nabla f(\mathbf{y}_k))$, where

$$\mathcal{X}_k \triangleq \{\mathbf{z} \in \mathcal{Z} : g(\mathbf{y}_k) + \langle \nabla g(\mathbf{y}_k), \mathbf{z} - \mathbf{y}_k \rangle \leq g_k\}$$

  Compute $\mathbf{x}_{k+1} = \frac{A_k}{A_k + a_k}\mathbf{x}_k + \frac{a_k}{A_k + a_k}\mathbf{z}_{k+1}$
  Update $A_{k+1} = A_k + a_k$
end for
Return: $\mathbf{x}_K$

Hence, the above requirements on the sequence $\{g_k\}$ are satisfied. Two remarks on the set $\mathcal{X}_k$ are in order. First, the set $\mathcal{X}_k$ in (4) has an explicit form, making the Euclidean projection onto $\mathcal{X}_k$ tractable. It can also be verified that $\mathcal{X}_k$ always contains the lower-level problem solution set $\mathcal{X}_g^*$. To prove this, let $\hat{\mathbf{x}}^*$ be any point in $\mathcal{X}_g^*$. By using the convexity of g, we obtain $g(\mathbf{y}_k) + \langle \nabla g(\mathbf{y}_k), \hat{\mathbf{x}}^* - \mathbf{y}_k \rangle \leq g(\hat{\mathbf{x}}^*) = g^* \leq g_k$. Thus, $\hat{\mathbf{x}}^*$ satisfies both constraints in (4), so $\hat{\mathbf{x}}^* \in \mathcal{X}_k$.

Now that we have identified an appropriate replacement for the set $\mathcal{X}_g^*$, we can easily implement a variant of the projected accelerated gradient method for the bilevel problem using the surrogate set $\mathcal{X}_k$. We refer to our method as the Accelerated Gradient Method for Bilevel Optimization (AGM-BiO) and its steps are outlined in Algorithm 1. It is important to note that the iterates, when projected onto the set $\mathcal{X}_k$, may not belong to the set $\mathcal{X}_g^*$, as $\mathcal{X}_k$ is an approximation of the true solution set. Consequently, the iterates might be infeasible. However, the design of $\mathcal{X}_k$ allows us to control the infeasibility of the iterates, as we will demonstrate in the convergence analysis section.

Remark 3.1. The design of the halfspace as specified in (4) should be recognized as a nuanced task. Various alternative formulations of halfspaces could fulfill the same primary conditions, such as $\{\mathbf{z} \in \mathcal{Z} : g(\mathbf{x}_k) + \langle \nabla g(\mathbf{x}_k), \mathbf{z} - \mathbf{x}_k \rangle \leq g_k\}$ and $\{\mathbf{z} \in \mathcal{Z} : g(\mathbf{z}_k) + \langle \nabla g(\mathbf{z}_k), \mathbf{z} - \mathbf{z}_k \rangle \leq g_k\}$. However, the selection of the gradient at $\mathbf{y}_k$ for constructing the halfspace is not arbitrary but essential as we characterize in the convergence analysis of our method.

Remark 3.2. How to project onto the set $\mathcal{X}_k$? In some cases, such as our over-parameterized regression problem, $\mathcal{X}_k$ is the intersection of an L_2 ball and a half-space, for which a closed-form solution exists to find the projected iterates. In other cases, such as our linear inverse problem, we may not be able to find $\mathcal{X}_k$ directly. Instead, we can solve the projection subproblem using Dykstra's projection algorithm [20]. In this case, an additional loop is needed to solve the subproblem.

3.1 Algorithm for the Composite Setting

While our paper focuses on the smooth setting, our proposed method can be also extended to the composite setting. Let us consider the composite counterpart of Problem (1):

$$\min_{\mathbf{x} \in \mathbb{R}^n} \; f(\mathbf{x}) := f_1(\mathbf{x}) + f_2(\mathbf{x}) \qquad \text{s.t.} \quad \mathbf{x} \in \operatorname*{argmin}_{\mathbf{z} \in \mathbb{R}^n} \; g(\mathbf{z}) := g_1(\mathbf{z}) + g_2(\mathbf{z}), \tag{6}$$

where $f_1, g_1 : \mathbb{R}^n \to \mathbb{R}$ are smooth convex functions and $f_2, g_2 : \mathbb{R}^n \to \mathbb{R}$ are nonsmooth convex functions, respectively. To analyze and implement the proximal gradient-based methods, we need the following definition concerning the property of the proximal mapping.

Definition 3.1. *Given $h : \mathbb{R}^n \to (-\infty, +\infty]$ and $\eta > 0$, the proximal map of h is defined as*

$$\mathit{Prox}_{\eta h}(\mathbf{x}) \triangleq \operatorname*{argmin}_{\mathbf{u} \in \mathbb{R}^n} \{\frac{1}{2\eta}\|\mathbf{u} - \mathbf{x}\|^2 + h(\mathbf{u})\}. \tag{7}$$

To handle the upper-level nonsmooth part f_2, we change the projection step in Step 6 of Algorithm 1 to a proximal update, which is similar to the accelerated proximal gradient method for single-level problems in [17].

Algorithm 2 Proximal Accelerated Gradient Method for Bilevel Optimization (P-AGM-BiO)

Input: A sequence $\{g_k\}_{k=0}^K$, a scalar $\gamma \in (0, 1]$
Initialization: $A_0 = 0$, $\mathbf{x}_0 = \mathbf{z}_0 \in \mathbb{R}^n$
for $k = 0, \ldots, K$ **do**
  Set $a_k = \gamma \dfrac{k+1}{4L_f}$
  Compute $\mathbf{y}_k = \dfrac{A_k}{A_k + a_k}\mathbf{x}_k + \dfrac{a_k}{A_k + a_k}\mathbf{z}_k$
  Compute $\mathbf{z}_{k+1} = Prox_{a_k(f_2+\delta_{\mathcal{X}_k})}(\mathbf{z}_k - a_k \nabla f_1(\mathbf{y}_k))$, where

$$\mathcal{X}_k \triangleq \{\mathbf{z} \in \mathbb{R}^n : g_1(\mathbf{y}_k) + \langle \nabla g_1(\mathbf{y}_k), \mathbf{z} - \mathbf{y}_k \rangle + g_2(\mathbf{z}) \leq g_k\}$$

  Compute $\mathbf{x}_{k+1} = \dfrac{A_k}{A_k + a_k}\mathbf{x}_k + \dfrac{a_k}{A_k + a_k}\mathbf{z}_{k+1}$
  Update $A_{k+1} = A_k + a_k$
end for
Return: $\mathbf{x}_K$

On the other hand, to deal with the lower-level nonsmooth part g_2, it is necessary to modify the approximated lower-level solution set $\mathcal{X}_k$. Specifically, we keep the linear approximation of the smooth part of the lower-level objective function g_1 while adding the nonsmooth part g_2 as a lower bound of g_k to construct $\mathcal{X}_k$. Note that the constructed set $\mathcal{X}_k$ is no longer a halfspace in this setting due to the possibly non-linear nature of g_2. We refer to our method as the Proximal Accelerated Gradient Method for Bilevel Optimization (P-AGM-BiO) and its steps are outlined in Algorithm 2.

Different proximal-friendly assumptions are commonly used in the literature of composite single-level/bilevel optimization [8, 15–17]. The following proximal-friendly assumption is necessary for our method in the composite setting.

Assumption 3.1. *The function $f_2 + \delta_{\mathcal{X}_k}$ in the Step 6 of Algorithm 2 is proximal-friendly, i.e. the proximal mapping in Definition 3.1 is easy to compute, where $\delta_{\mathcal{X}_k}(\cdot)$ is the indicator function.*

This assumption implies that f_2 is proximal-friendly and that projecting onto the constructed set $\mathcal{X}_k$ can be done efficiently. Moreover, the function $f_2 + \delta_{\mathcal{X}_k}$ is the sum of two convex functions, and the study of proximal mappings for such sums is well-documented in the literature [21–24]. Under this assumption, all analysis for the smooth case can be extended to the composite setting. The details are provided in Section B of the Appendix.

4 Convergence Analysis

In this section, we analyze the convergence rate and iteration complexity of our proposed AGM-BiO method for convex simple bilevel optimization problems. We choose the stepsize $a_k = \frac{k+1}{4L_f}$, which is inspired from our theoretical analysis. The main theorem is as follows,

Theorem 4.1. *Suppose Assumption 2.1 holds. Let $\{\mathbf{x}_k\}_{k\geq 0}$ be the sequence of iterates generated by Algorithm 1 with stepsize $a_k = \frac{k+1}{4L_f}$ for $k \geq 0$ and suppose the sequence g_k used for generating the cutting plane satisfies* (5)*. Then, for any $k \geq 0$ we have,*

(i) The function suboptimality is bounded above by $f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{4L_f\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{k(k+1)}$.

(ii) The infeasibility term is bounded above by $g(\mathbf{x}_k) - g(\mathbf{x}^*) \leq \frac{4L_g\|\mathbf{x}_0 - \mathbf{x}^*\|^2 \ln(k+1)}{k(k+1)} + \frac{2L_g D^2}{k+1}$.

(iii) Furthermore, if the condition $f(\mathbf{x}_k) \geq f(\mathbf{x}^*)$ holds, then the infeasibility term is bounded above by $g(\mathbf{x}_k) - g(\mathbf{x}^*) \leq \frac{8L_g\|\mathbf{x}_0 - \mathbf{x}^*\|^2 \ln(k+1)}{k(k+1)}$.

Theorem 4.1 shows the upper-level objective function gap is upper bounded by $\mathcal{O}(1/k^2)$, which matches the convergence rate of the accelerated gradient method for single-level optimization problems. On the other hand, the suboptimality of the lower-level objective which measures infeasibility

for the bilevel problem in the worst case is bounded above by $\mathcal{O}(1/k)$. In the case where $f(\mathbf{x}_k) \geq f^*$, this upper bound improves to $\mathcal{O}(1/k^2)$. As a corollary of the worst-case bounds, Algorithm 1 will return an (ϵ_f, ϵ_g)-optimal solution after at most the following number of iterations $\mathcal{O}(\max\{\frac{1}{\sqrt{\epsilon_f}}, \frac{1}{\epsilon_g}\})$. We should emphasize that, under the assumptions being considered, this complexity bound represents the best-known bound among all previous works summarized in Table 1.

Remark 4.1 (The necessity of compactness of $\mathcal{Z}$). For the lower-level objective, we show that $A_k(g(\mathbf{x}_k) - g(\mathbf{x}^*)) \leq \sum_{i=0}^{k-1} a_i(g_i - g^*) + \frac{L_g}{4L_f} \sum_{i=0}^{k-1} \|\mathbf{z}_{i+1} - \mathbf{z}_i\|^2$ ((24) in Section A). The main challenge in obtaining an accelerated rate of $\mathcal{O}(1/k^2)$ for g is controlling $\sum_{i=0}^{k-1} \|\mathbf{z}_{i+1} - \mathbf{z}_i\|^2$. Without a lower bound on f, this term cannot be bounded by the upper-level suboptimality alone. If $f(\mathbf{x}_k) \geq f(\mathbf{x}^*)$, we can achieve the rate of $\mathcal{O}(1/k^2)$ for g. Otherwise, we use the compactness of $\mathcal{Z}$ to achieve $\mathcal{O}(1/k)$ for g. Please refer to Section A for more details.

Remark 4.2 (Removable $\log$ terms). The $\log$ terms in all the complexity results can be removed by choosing the auxiliary sequence $g_k = g_K$ for all $0 \leq k \leq K$, which satisfies the condition (5). This eliminates the $\log$ term in (24) and all subsequent results. However, this choice of $\{g_k\}_{k\geq 0}$ requires predetermining the total number of iterations K.

Since the algorithm's output $\hat{\mathbf{x}}$ may fall outside the feasible set $\mathcal{X}_g^*$, the expression $f(\hat{\mathbf{x}}) - f^*$ may not necessarily be non-negative. On the other hand, under the considered assumptions, proving convergence in terms of $|f(\hat{\mathbf{x}}) - f^*|$ is known to be impossible due to a negative result presented by [25]. Specifically, for any first-order method and a given number of iterations k, they demonstrated the existence of an instance of Problem (1) where $|f(\mathbf{x}_k) - f^*| \geq 1$ for all $k \geq 0$. Thus, to provide any form of guarantee in terms of the absolute value of the suboptimality, i.e., $|f(\hat{\mathbf{x}}) - f^*|$, we need an additional assumption to obtain a lower bound on suboptimality and to provide a convergence bound for $|f(\hat{\mathbf{x}}) - f^*|$. We will address this point in the following section.

4.1 Convergence under Hölderian Error Bound

In this section, we introduce an additional regularity condition on g to establish a lower bound for $f(\hat{\mathbf{x}}) - f^*$. Specifically, we assume that the lower-level objective function g satisfies the Hölderian Error Bound condition, which governs how $g(\mathbf{x})$ grows as $\mathbf{x}$ moves away from the optimal solution set $\mathcal{X}_g^*$. Intuitively, since our method's output $\hat{\mathbf{x}}$ is ϵ_g-optimal for the lower-level problem, it should be close to $\mathcal{X}_g^*$ under this regularity condition. We can then use this proximity and the smoothness property of f to establish a lower bound for $f(\hat{\mathbf{x}}) - f^*$.

Assumption 4.1. *The function g satisfies the Hölderian error bound for some $\alpha > 0$ and $r \geq 1$, i.e,*

$$\frac{\alpha}{r} \operatorname{dist}\left(\mathbf{x}, \mathcal{X}_g^*\right)^r \leq g(\mathbf{x}) - g^*, \quad \forall \mathbf{x} \in \mathcal{Z}, \tag{8}$$

where $\operatorname{dist}\left(\mathbf{x}, \mathcal{X}_g^*\right) \triangleq \inf_{\mathbf{x}' \in \mathcal{X}_g^*} \|\mathbf{x} - \mathbf{x}'\|$.

We note that the Hölderian error bound condition in (8) is well-studied in the optimization literature [26–28] and is known to hold in general when function g is analytic and the set $\mathcal{Z}$ is bounded [29]. There are two important special cases of the Hölderian error bound condition: 1) g satisfies (8) with $r = 1$ known as the weak sharpness condition [30, 31]; 2) g satisfies (8) with $r = 2$ known as the quadratic functional growth condition [32]. By using the Hölderian error bound condition, [6] established a stronger relation between suboptimality and infeasibility, as shown next.

Proposition 4.2 ([6, Proposition 1])**.** *Assume that f is convex and g satisfies Assumption 4.1, and define $M = \max_{\mathbf{x} \in \mathcal{X}_g^*} \|\nabla f(\mathbf{x})\|_*$. Then $f(\hat{\mathbf{x}}) - f^* \geq -M(\frac{r(g(\hat{\mathbf{x}}) - g^*)}{\alpha})^{\frac{1}{r}}$ for any $\hat{\mathbf{x}} \in \mathcal{Z}$.*

Hence, under Assumption 4.1, Proposition 4.2 shows that the suboptimality $f(\hat{\mathbf{x}}) - f^*$ can also be bounded from below when $\hat{\mathbf{x}}$ is an approximate solution of the lower-level problem. As a result, we can establish a convergence bound on $|f(\mathbf{x}_k) - f^*|$ by combining Proposition 4.2 with the upper bounds in Theorem 4.1. Moreover, it also allows us to improve the convergence rate for the lower-level problem. To prove this claim, we first introduce the following lemma which establishes an upper bound on the weighted sum of upper and lower-level objectives.

Lemma 4.3. *Suppose conditions (ii) and (iii) in Assumption 2.1 hold. Let $\{\mathbf{x}_k\}$ be the sequence of iterates generated by Algorithm 1 with stepsize $a_k = \gamma \frac{k+1}{4L_f}$, where $0 < \gamma \leq 1$. If the sequence g_k*

used for generating the cutting plane satisfies (5)*, then for any* $\lambda \geq \frac{L_g}{(2/\gamma-1)L_f}$ *and* $k \geq 0$ *we have*

$$\lambda(f(\mathbf{x}_k) - f(\mathbf{x}^*)) + g(\mathbf{x}_k) - g(\mathbf{x}^*) \leq \frac{4L_g\|\mathbf{x}_0 - \mathbf{x}^*\|^2 \ln(k+1)}{k(k+1)} + \frac{4\lambda L_f\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{\gamma k(k+1)}. \quad (9)$$

This result characterizes and upper bound of $\tilde{\mathcal{O}}(1/k^2)$ on the expression $\lambda(f(\mathbf{x}_k)-f(\mathbf{x}^*))+g(\mathbf{x}_k)-g(\mathbf{x}^*)$. That said, the first term in this expression, a.k.a., $\lambda(f(\mathbf{x}_k) - f(\mathbf{x}^*))$ may not be non-negative for a bilevel problem as discussed earlier. Hence, we cannot simply eliminate $\lambda(f(\mathbf{x}_k) - f(\mathbf{x}^*))$ to show an upper bound of $\mathcal{O}(1/k^2)$ on infeasibility, a.k.a., $g(\mathbf{x}_k) - g(\mathbf{x}^*)$. Instead, we leverage the Hölderian error bound on g and apply Proposition 4.2 to the first term. As a result, we can eliminate the dependence on f in (9). In this case, we can establish an upper bound on infeasibility.

Theorem 4.4. *Suppose conditions (ii) and (iii) in Assumption 2.1 hold and the lower-level function* g *satisfies the Hölderian error bound with* $r>1$*. Let* $\{\mathbf{x}_k\}$ *be the iterates generated by Algorithm 1 with stepsize* $a_k = \gamma\frac{k+1}{4L_f}$*, where* $\gamma = 1/(\frac{2L_g}{L_f}K^{\frac{2r-2}{2r-1}}+2)$ *and* K *is the total number of iterations. Moreover, suppose the sequence* g_k *used for generating the cutting plane satisfies* (5)*. If we define the constants* $C_f \triangleq 8L_f\|\mathbf{x}_0 - \mathbf{x}^*\|^2$*,* $C_g \triangleq 12L_g\|\mathbf{x}_0 - \mathbf{x}^*\|^2$ *and* $C \triangleq M(\frac{r}{\alpha})^{\frac{1}{r}}$*, where* $M \triangleq \max_{\mathbf{x}\in\mathcal{X}_g^*}\|\nabla f(\mathbf{x})\|$*,* α *and* r *are the parameters in Assumption 4.1, then the following results hold:*

(i) The function suboptimality is bounded above by

$$f(\mathbf{x}_K) - f(\mathbf{x}^*) \leq \frac{C_g(\ln K+1)}{K^{\frac{2r}{2r-1}}} + \frac{C_f}{K^2}.$$

(ii) The function suboptimality is bounded below by

$$f(\mathbf{x}_K) - f(\mathbf{x}^*) \geq -C\max\left\{\frac{(2C_g(\ln K+1))^{\frac{1}{r}}}{K^{\frac{2}{2r-1}}} + \frac{(2C_f)^{\frac{1}{r}}}{K^{\frac{2}{r}}}, \frac{(2C)^{\frac{1}{r-1}}}{K^{\frac{2}{2r-1}}}\right\}$$

(iii) The infeasibility term is bounded above by

$$g(\mathbf{x}_K) - g(\mathbf{x}^*) \leq \max\left\{\frac{2C_g(\ln K+1)}{K^{\frac{2r}{2r-1}}} + \frac{2C_f}{K^2}, \frac{(2C)^{\frac{r}{r-1}}}{K^{\frac{2r}{2r-1}}}\right\}$$

Before unfolding this result, we would like to highlight that unlike the result in Theorem 4.1, the above bounds in Theorem 4.4 do not require the feasible set to be compact. Since $r > 1$, the first result shows $f(\mathbf{x}_K) - f(\mathbf{x}^*)$ has an upper bound of $\tilde{\mathcal{O}}((\frac{1}{K})^{\frac{2r}{2r-1}})$ and the second result guarantees a lower bound of $-\tilde{\mathcal{O}}((\frac{1}{K})^{\frac{2}{2r-1}})$. These two bounds together lead to an upper bound of $\tilde{\mathcal{O}}((\frac{1}{K})^{\frac{2}{2r-1}})$ for the absolute error $|f(\mathbf{x}_K)-f(\mathbf{x}^*)|$. Moreover, the third result implies that the lower-level problem suboptimality which measures infeasibility is bounded above by $\tilde{\mathcal{O}}((\frac{1}{K})^{\frac{2r}{2r-1}})$.

The previous result presented in Theorem 4.4 is applicable when $r > 1$. However, for the case that 1st-order Hölderian error bound condition on g holds (i.e., weak sharpness condition), we require a distinct analysis and a different choice of γ to achieve the tightest bounds. In the subsequent theorem, we present our findings for this specific scenario.

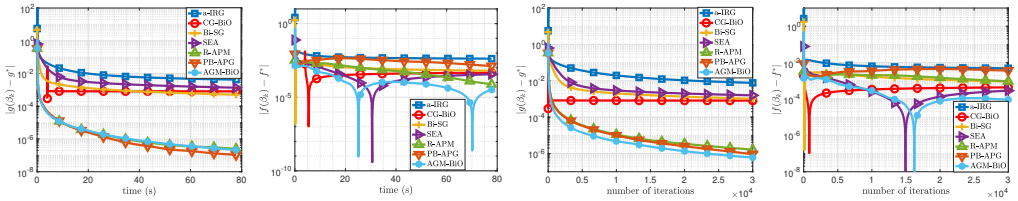
Theorem 4.5. *Suppose conditions (ii) and (iii) in Assumption 2.1 are met and that the lower-level objective function* g *satisfies the Hölderian error bound with* $r = 1$*. Let* $\{\mathbf{x}_k\}$ *be the sequence of iterates generated by Algorithm 1 with stepsize* $a_k = \gamma\frac{k+1}{4L_f}$*, where* $0 < \gamma \leq \min\{\frac{2\alpha L_f}{2ML_g+\alpha L_f}, 1\}$*. Moreover, suppose the sequence* g_k *used for generating the cutting plane satisfies* (5)*, and recall* $M \triangleq \max_{\mathbf{x}\in\mathcal{X}_g^*}\|\nabla f(\mathbf{x})\|$ *and* α *in Assumption 4.1. If we define the constants* $C_f \triangleq 4L_f\|\mathbf{x}_0 - \mathbf{x}^*\|^2$ *and* $C_g \triangleq 8L_g\|\mathbf{x}_0 - \mathbf{x}^*\|^2$*, then for any* $k \geq 0$*:*

(i) The function suboptimality is bounded above by $f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{C_f}{\gamma k(k+1)}$.

(ii) The function suboptimality is bounded below by $f(\mathbf{x}_k) - f(\mathbf{x}^*) \geq -\frac{C_gM(\ln k+1)}{\alpha k(k+1)} - \frac{C_f}{\gamma k(k+1)}$.

(iii) The infeasibility term is bounded above by $g(\mathbf{x}_k) - g(\mathbf{x}^*) \leq \frac{C_g(\ln k+1)}{k(k+1)} + \frac{\alpha C_f}{\gamma Mk(k+1)}$.

Theorem 4.5 shows that under the Hölderian error bound with $r = 1$, also known as weak sharpness condition, the absolute value of the function suboptimality $|f(\mathbf{x}_k) - f(\mathbf{x}^*)|$ approaches zero at a rate of $\mathcal{O}(1/k^2)$ – ignoring the log term. The lower-level error $g(\mathbf{x}_k) - g(\mathbf{x}^*)$, capturing the infeasibility of the iterates, also approaches zero at a rate of $\mathcal{O}(1/k^2)$. As a corollary, Algorithm 1 returns an (ϵ_f, ϵ_g)-absolute optimal solution after $\tilde{\mathcal{O}}(\max\{\frac{1}{\sqrt{\epsilon_f}}, \frac{1}{\sqrt{\epsilon_g}}\})$ iterations.



(a) Infeas. vs. time (b) Subopt. vs. time (c) Infeas. vs. iterations (d) Subopt. vs. iterations

Figure 1: Comparison of a-IRG, CG-BiO, Bi-SG, SEA, R-APM, PB-APG, and AGM-BiO for solving the over-parameterized regression problem.

5 Numerical Experiments

In this section, we evaluate our AGM-BiO method on two different bilevel problems using real and synthetic datasets. We compare its runtime and iteration count with other methods, including a-IRG [12], CG-BiO [6], Bi-SG [13], SEA [14], R-APM [8], PB-APG [16], and Bisec-BiO [15].

Over-parameterized regression. We examine problem (1) where the lower-level problem corresponds to training loss, and the upper-level pertains to validation loss. The objective is to minimize the validation loss by selecting an optimal training loss solution. This method is also referred to as lexicographic optimization [33]. A common example of that is the constrained regression problem, where we aim to find an optimal parameter vector $\boldsymbol{\beta} \in \mathbb{R}^d$ for the validation loss that minimizes the loss $\ell_{\text{tr}}(\boldsymbol{\beta})$ over the training dataset $\mathcal{D}_{\text{tr}}$. To represent some prior knowledge, we constrain $\boldsymbol{\beta}$ to be in some subset $\mathcal{Z} \subseteq \mathbb{R}^d$, e.g., $\mathcal{Z} = \{\boldsymbol{\beta} \mid \beta_1 \leq \cdots \leq \beta_d\}$ in isotonic regression and $\mathcal{Z} = \{\boldsymbol{\beta} \mid \|\boldsymbol{\beta}\|_p \leq \lambda\}$ in L_p constrained regression. Without explicit regularization, an over-parameterized regression over the training dataset has multiple global minima, but not all these optimal regression coefficients perform equally on validation or testing datasets. Thus, the upper-level objective serves as a secondary criterion to ensure a smaller error on the validation dataset $\mathcal{D}_{\text{val}}$. The problem can be cast as

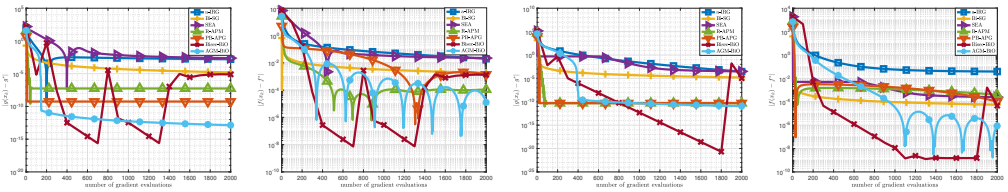
$$\min_{\boldsymbol{\beta} \in \mathbb{R}^d} f(\boldsymbol{\beta}) \triangleq \ell_{\text{val}}(\boldsymbol{\beta}) \quad \text{s.t.} \quad \boldsymbol{\beta} \in \operatorname*{argmin}_{\mathbf{z} \in \mathcal{Z}} g(\mathbf{z}) \triangleq \ell_{\text{tr}}(\mathbf{z})$$

In this case, both upper-level and lower-level objectives are convex and smooth if the loss ℓ is smooth and convex. Since projections onto the sublevel set of f are difficult to compute, Bisec-BiO is excluded from this experiment.

We apply the Wikipedia Math Essential dataset [34] which is composed of a data matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ with $n = 1068$ samples and $d = 730$ features and an output vector $\mathbf{b} \in \mathbb{R}^n$. We use 75% of the dataset as the training set $(\mathbf{A}_{\text{tr}}, \mathbf{b}_{\text{tr}})$ and 25% as the validation set $(\mathbf{A}_{\text{val}}, \mathbf{b}_{\text{val}})$. For both upper- and lower-level loss functions, we use the least squared loss. Then the lower-level objective is $g(\boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{A}_{tr}\boldsymbol{\beta} - \mathbf{b}_{tr}\|_2^2$, the upper-level objective is $f(\boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{A}_{val}\boldsymbol{\beta} - \mathbf{b}_{val}\|_2^2$, and the constraint set is chosen as the unit L_2-ball $\mathcal{Z} = \{\boldsymbol{\beta} \mid \|\boldsymbol{\beta}\|_2 \leq 1\}$. Note that this regression problem is over-parameterized since the number of features d is larger than the number of data points in both the training set and validation set.

In Figures 1(a) and 1(c), we observe that the three accelerated gradient-based methods (R-APM, PB-APG, and AGM-BiO) converge faster in reducing infeasibility, both in terms of runtime and number of iterations. In terms of absolute suboptimality, shown in Figures 1(b) and 1(d), AGM-BiO achieves the smallest absolute suboptimality gap among all algorithms. Unlike the infeasibility plots, R-APM and PB-APG underperform compared to AGM-BiO. Note that the lower-level objective in this problem does not satisfy the weak sharpness condition, so the regularization parameter η in R-APM is set as $1/(K+1)$. Consequently, the suboptimality for R-APM converges slower than AGM-BiO, as suggested by the theoretical results in Table 1.

Linear inverse problems. In the next experiment, we concentrate on a problem that fulfills the Hölderian Error Bound condition for some $r > 1$. We aim to evaluate the performance of our method in this specific context and verify the validity of our theoretical results for this scenario. Specifically, we focus on the so-called linear inverse problems, commonly used to evaluate convex



(a) Infeas. ($n = 3$) (b) Subopt. ($n = 3$) (c) Infeas. ($n = 100$) (d) Subopt. ($n = 100$)

Figure 2: Comparison of a-IRG, Bi-SG, SEA, R-APM, PB-APG, Bisec-BiO, and AGM-BiO for solving the linear inverse problem.

bilevel optimization algorithms, which originate from [35]. The goal of linear inverse problems is to obtain a solution $\mathbf{x} \in \mathbb{R}^n$ to the system of linear equation $\mathbf{A}\mathbf{x} = \mathbf{b}$. Note that if $\mathbf{A}$ is rank-deficient, there can be multiple solutions, or there might be no exact solution due to noise. To address this issue, we chase a solution that has the smallest weighted norm with respect to some positive definite matrix $\mathbf{Q}$, i.e., $\|\mathbf{x}\|_{\mathbf{Q}} := \sqrt{\mathbf{x}^\top \mathbf{Q}\mathbf{x}}$. This problem can be also cast as the following simple bilevel problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \triangleq \frac{1}{2}\|\mathbf{x}\|_{\mathbf{Q}}^2 \quad \text{s.t.} \quad \mathbf{x} \in \operatorname*{argmin}_{\mathbf{z}\in\mathcal{Z}} g(\mathbf{z}) \triangleq \frac{1}{2}\|\mathbf{A}\mathbf{z} - \mathbf{b}\|_2^2$$

For this class of problem, if $\mathbf{Q}$, $\mathbf{A}$, and $\mathbf{b}$ are generated randomly or by the "regularization tools" like [14, 35], we are not able to obtain the exact optimal value f^*. To the best of our knowledge, no existing solver could obtain the exact optimal value f^* for this bilevel problem. Specifically, the existing solvers either fail to solve this bilevel problem or return an inaccurate solution by solving a relaxed version of the problem. Hence, in [14, 35] they only reported the upper-level function value. However, in this paper, we intend to obtain the complexity bounds for finding (ϵ_f, ϵ_g)-optimal and (ϵ_f, ϵ_g)-absolute optimal solutions. Without knowing f^*, we can not characterize the behavior of $|f(\mathbf{x}_k) - f^*|$. Therefore, we choose an example where we can obtain the exact solution. Specifically, we set $\mathbf{Q} = \mathbf{I}_n$, $\mathbf{A} = \mathbf{1}_n^\top$, $\mathbf{b} = 1$, and the constraint set $\mathcal{Z} = \mathbb{R}_+^n$. In this case, the optimal solution $\mathbf{x}^* = \frac{1}{n}\mathbf{1}_n$ and optimal value $f^* = \frac{1}{2n}$. This specific example essentially involves seeking the minimum norm for an under-determined system. Note that the lower-level objective in this problem satisfies the Hölderian Error Bound condition with order $r = 2$ [36]. Hence, we do not need the constraint set $\mathcal{Z}$ to be compact as shown in Theorem 4.4. Due to the unbounded nature of the constraint set, Frank-Wolfe-type methods are not viable options. Consequently, we have opted not to incorporate CG-BiO in this experiment.

We explored examples with two distinct dimensions: $n = 3$ and $n = 100$, evaluating a total of 2000 gradients. In Figures 2(a) and 2(c), AGM-BiO shows superior performance in terms of infeasibility. In Figures 2(b) and 2(d), we compare methods in terms of absolute error of suboptimality. The gap between R-APM and AGM-BiO is smaller for $n = 3$, but for $n = 100$, AGM-BiO significantly outperforms all other methods, including R-APM. Since the regularization and penalty parameters in R-APM and PB-APG are fixed, they might get stuck at a certain accuracy level, as seen in Figures 2(a) and 2(c). In contrast, AGM-BiO uses a dynamic framework for minimizing the upper and lower-level functions, consistently reducing both suboptimality and infeasibility. Although Bisec-BiO theoretically has the best complexity results due to the ease of projecting onto the sublevel set of f, its performance in the last iteration is inconsistent, as shown in Figure 2.

6 Conclusion

In this paper, we introduced an accelerated gradient-based algorithm for solving a specific class of bilevel optimization problems with convex objective functions in both the upper and lower levels. Our proposed algorithm achieves a computational complexity of $\mathcal{O}(\max\{\epsilon_f^{-0.5}, \epsilon_g^{-1}\})$. When an additional weak sharpness condition is applied to the lower-level function g, the iteration complexity improves to $\tilde{\mathcal{O}}(\max\{\epsilon_f^{-0.5}, \epsilon_g^{-0.5}\})$, matching the well-known fastest convergence rate for single-level convex optimization problems. We further extended this result to an iteration complexity of $\tilde{\mathcal{O}}(\max\{\epsilon_f^{-\frac{2r-1}{2r}}, \epsilon_g^{-\frac{2r-1}{2r}}\})$ when the lower-level loss satisfies the Hölderian error bound assumption.

Acknowledgements

The research of J. Cao, R. Jiang, and A. Mokhtari is supported in part by NSF Grant 2127697 and the NSF AI Institute for Foundations of Machine Learning (IFML) at UT Austin. The research of E. Yazdandoost Hamedani is supported by NSF Grant 2127696.

References

[1] Zalán Borsos, Mojmir Mutny, and Andreas Krause. Coresets via bilevel optimization for continual learning and streaming. *Advances in Neural Information Processing Systems*, 33:14879–14890, 2020.

[2] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*, pages 1568–1577. PMLR, 2018.

[3] Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. Truncated back-propagation for bilevel optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1723–1732. PMLR, 2019.

[4] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. *Advances in neural information processing systems*, 32, 2019.

[5] Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018.

[6] Ruichen Jiang, Nazanin Abolfazli, Aryan Mokhtari, and Erfan Yazdandoost Hamedani. A conditional gradient-based method for simple bilevel optimization with convex lower-level problem. In *International Conference on Artificial Intelligence and Statistics*, pages 10305–10323. PMLR, 2023.

[7] Jincheng Cao, Ruichen Jiang, Nazanin Abolfazli, Erfan Yazdandoost Hamedani, and Aryan Mokhtari. Projection-free methods for stochastic simple bilevel optimization with convex lower-level problem. *Advances in Neural Information Processing Systems*, 36, 2024.

[8] Sepideh Samadi, Daniel Burbano, and Farzad Yousefian. Achieving optimal complexity guarantees for a class of bilevel convex optimization problems. *arXiv preprint arXiv:2310.12247*, 2023.

[9] Stephen Dempe, Nguyen Dinh, and Joydeep Dutta. Optimality conditions for a simple convex bilevel programming problem. *Variational Analysis and Generalized Differentiation in Optimization and Control: In Honor of Boris S. Mordukhovich*, pages 149–161, 2010.

[10] Joydeep Dutta and Tanushree Pandit. Algorithms for simple bilevel programming. *Bilevel Optimization: Advances and Next Challenges*, pages 253–291, 2020.

[11] Yekini Shehu, Phan Tu Vuong, and Alain Zemkoho. An inertial extrapolation method for convex simple bilevel optimization. *Optimization Methods and Software*, 36(1):1–19, 2021.

[12] Harshal D Kaushik and Farzad Yousefian. A method with convergence rates for optimization problems with variational inequality constraints. *SIAM Journal on Optimization*, 31(3):2171–2198, 2021.

[13] Roey Merchav and Shoham Sabach. Convex bi-level optimization problems with non-smooth outer objective function. *arXiv preprint arXiv:2307.08245*, 2023.

[14] Lingqing Shen, Nam Ho-Nguyen, and Fatma Kılınç-Karzan. An online convex optimization-based framework for convex bilevel optimization. *Mathematical Programming*, 198(2):1519–1582, 2023.

[15] Jiulin Wang, Xu Shi, and Rujun Jiang. Near-optimal convex simple bilevel optimization with a bisection method. *arXiv preprint arXiv:2402.05415*, 2024.

[16] Pengyu Chen, Xu Shi, Rujun Jiang, and Jiulin Wang. Penalty-based methods for simple bilevel optimization under h\"{o} lderian error bounds. *arXiv preprint arXiv:2402.02155*, 2024.

[17] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[18] Alexandre d'Aspremont, Damien Scieur, Adrien Taylor, et al. Acceleration methods. *Foundations and Trends® in Optimization*, 5(1-2):1–245, 2021.

[19] Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM J. Optim.*, 2008.

[20] Heinz H Bauschke, Regina S Burachik, Patrick L Combettes, Veit Elser, D Russell Luke, and Henry Wolkowicz. *Fixed-point algorithms for inverse problems in science and engineering*, volume 49. Springer Science & Business Media, 2011.

[21] Yao-Liang Yu. On decomposing the proximal map. *Advances in neural information processing systems*, 26, 2013.

[22] Nelly Pustelnik and Laurent Condat. Proximity operator of a sum of functions; application to depth map estimation. *IEEE Signal Processing Letters*, 24(12):1827–1831, 2017.

[23] Heinz H Bauschke, Minh N Bui, and Xianfu Wang. Projecting onto the intersection of a cone and a sphere. *SIAM Journal on Optimization*, 28(3):2158–2188, 2018.

[24] Samir Adly, Loïc Bourdin, and Fabien Caubet. On a decomposition formula for the proximal operator of the sum of two convex functions. *Journal of Convex Analysis*, 26(2):699–718, 2019.

[25] Lesi Chen, Jing Xu, and Jingzhao Zhang. On bilevel optimization without lower-level strong convexity. *arXiv preprint arXiv:2301.00712*, 2023.

[26] Jong-Shi Pang. Error bounds in mathematical programming. *Mathematical Programming*, 79(1-3):299–332, 1997.

[27] Jérôme Bolte, Trong Phong Nguyen, Juan Peypouquet, and Bruce W Suter. From error bounds to the complexity of first-order descent methods for convex functions. *Mathematical Programming*, 165:471–507, 2017.

[28] Vincent Roulet and Alexandre d'Aspremont. Sharpness, restart and acceleration. *Advances in Neural Information Processing Systems*, 30, 2017.

[29] Zhi-Quan Luo and Jong-Shi Pang. Error bounds for analytic systems and their applications. *Mathematical Programming*, 67(1-3):1–28, 1994.

[30] James V Burke and Michael C Ferris. Weak sharp minima in mathematical programming. *SIAM Journal on Control and Optimization*, 31(5):1340–1359, 1993.

[31] James V Burke and Sien Deng. Weak sharp minima revisited, part ii: application to linear regularity and error bounds. *Mathematical programming*, 104:235–261, 2005.

[32] Dmitriy Drusvyatskiy and Adrian S Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods. *Mathematics of Operations Research*, 43(3):919–948, 2018.

[33] Chengyue Gong and Xingchao Liu. Bi-objective trade-off with dynamic barrier gradient descent. *NeurIPS 2021*, 2021.

[34] Benedek Rozemberczki, Paul Scherer, Yixuan He, George Panagopoulos, Alexander Riedel, Maria Astefanoaei, Oliver Kiss, Ferenc Beres, Guzmán López, Nicolas Collignon, et al. Pytorch geometric temporal: Spatiotemporal signal processing with neural machine learning models. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 4564–4573, 2021.

[35] Shoham Sabach and Shimrit Shtern. A first order method for solving convex bilevel optimization problems. *SIAM Journal on Optimization*, 27(2):640–660, 2017.

[36] Ion Necoara, Yu Nesterov, and Francois Glineur. Linear convergence of first order methods for non-strongly convex optimization. *Mathematical Programming*, 175:69–107, 2019.

[37] Andreĭ Nikolaevich Tikhonov and VIAK Arsenin. Solutions of ill-posed problems. *(No Title)*, 1977.

[38] Mikhail Solodov. An explicit descent method for bilevel convex optimization. *Journal of Convex Analysis*, 14(2):227, 2007.

[39] Mikhail V Solodov. A bundle method for a class of bilevel nonsmooth convex minimization problems. *SIAM Journal on Optimization*, 18(1):242–259, 2007.

[40] Elias S Helou and Lucas EA Simões. ϵ-subgradient algorithms for bilevel convex optimization. *Inverse Problems*, 33(5):055020, 2017.

[41] Amir Beck and Shoham Sabach. A first order method for finding minimal norm-like solutions of convex optimization problems. *Mathematical Programming*, 147(1-2):25–46, 2014.

[42] Yura Malitsky. Chambolle-pock and tseng's methods: relationship and extension to the bilevel optimization. *arXiv preprint arXiv:1706.02602*, page 3, 2017.

[43] Boris T Polyak. Introduction to optimization. 1987.

Appendix / supplemental material

A Proof of the Main Results

A.1 Proof of Theorem 4.1

To prove Theorem 4.1, we start with the following general lemma that holds for any choice of the step sizes $\{a_k\}$.

Lemma A.1. *Let $\{\mathbf{x}_k\}$ be the sequence of iterates generated by Algorithm 1 with stepsize $a_k > 0$ for $k \geq 0$. Then we have*

$$\begin{aligned} A_{k+1}(f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*)) + \frac{1}{2}\|\mathbf{z}_{k+1} - \mathbf{x}^*\|^2 - \Big(A_k(f(\mathbf{x}_k) - f(\mathbf{x}^*)) + \frac{1}{2}\|\mathbf{z}_k - \mathbf{x}^*\|^2\Big) \\ \leq \left(\frac{L_f a_k^2}{2A_{k+1}} - \frac{1}{2}\right)\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2, \end{aligned} \tag{10}$$

$$A_{k+1}(g(\mathbf{x}_{k+1}) - g(\mathbf{x}^*)) - A_k(g(\mathbf{x}_k) - g(\mathbf{x}^*)) \leq a_k(g_k - g(\mathbf{x}^*)) + \frac{L_g a_k^2}{2A_{k+1}}\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2. \tag{11}$$

Proof of Lemma A.1. Let $\mathbf{x}^*$ be any optimal solution of (1). We first consider the upper-level objective f. Since f is convex, we have

$$f(\mathbf{y}_k) - f(\mathbf{x}^*) \leq \langle \nabla f(\mathbf{y}_k), \mathbf{y}_k - \mathbf{x}^* \rangle, \quad f(\mathbf{y}_k) - f(\mathbf{x}_k) \leq \langle \nabla f(\mathbf{y}_k), \mathbf{y}_k - \mathbf{x}_k \rangle. \tag{12}$$

Now given the update rule $A_{k+1} = A_k + a_k$, we can write

$$A_{k+1}(f(\mathbf{y}_k) - f(\mathbf{x}^*)) - A_k(f(\mathbf{x}_k) - f(\mathbf{x}^*)) = a_k(f(\mathbf{y}_k) - f(\mathbf{x}^*)) + A_k(f(\mathbf{y}_k) - f(\mathbf{x}_k)) \tag{13}$$

Combining (12) and (13), we have

$$\begin{aligned} & A_{k+1}(f(\mathbf{y}_k) - f(\mathbf{x}^*)) - A_k(f(\mathbf{x}_k) - f(\mathbf{x}^*)) \\ & \leq a_k(\langle \nabla f(\mathbf{y}_k), \mathbf{y}_k - \mathbf{x}^* \rangle) + A_k(\langle \nabla f(\mathbf{y}_k), \mathbf{y}_k - \mathbf{x}_k \rangle) \\ & = \langle \nabla f(\mathbf{y}_k), a_k \mathbf{y}_k + A_k(\mathbf{y}_k - \mathbf{x}_k) - a_k \mathbf{x}^* \rangle \\ & = a_k \langle \nabla f(\mathbf{y}_k), \mathbf{z}_k - \mathbf{x}^* \rangle, \end{aligned} \tag{14}$$

where the last equality follows from the definition of $\mathbf{y}_k$. Furthermore, since f is L_f-smooth, we have

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{y}_k) + \langle \nabla f(\mathbf{y}_k), \mathbf{x}_{k+1} - \mathbf{y}_k \rangle + \frac{L_f}{2}\|\mathbf{x}_{k+1} - \mathbf{y}_k\|^2. \tag{15}$$

If we multiply both sides of (15) by A_{k+1} and combine the resulting inequality with (14), we obtain

$$\begin{aligned} & A_{k+1}(f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*)) - A_k(f(\mathbf{x}_k) - f(\mathbf{x}^*)) \\ & \leq a_k \langle \nabla f(\mathbf{y}_k), \mathbf{z}_k - \mathbf{x}^* \rangle + A_{k+1} \langle \nabla f(\mathbf{y}_k), \mathbf{x}_{k+1} - \mathbf{y}_k \rangle + \frac{L_f A_{k+1}}{2}\|\mathbf{x}_{k+1} - \mathbf{y}_k\|^2 \\ & = a_k \langle \nabla f(\mathbf{y}_k), \mathbf{z}_k - \mathbf{x}^* \rangle + a_k \langle \nabla f(\mathbf{y}_k), \mathbf{z}_{k+1} - \mathbf{z}_k \rangle + \frac{L_f a_k^2}{2A_{k+1}}\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2 \\ & = a_k \langle \nabla f(\mathbf{y}_k), \mathbf{z}_{k+1} - \mathbf{x}^* \rangle + \frac{L_f a_k^2}{2A_{k+1}}\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2, \end{aligned} \tag{16}$$

where we used the fact that $a_k(\mathbf{z}_{k+1} - \mathbf{z}_k) = A_{k+1}(\mathbf{x}_{k+1} - \mathbf{y}_k)$ in the first equality. Moreover, since $\mathbf{x}^* \in \mathcal{X}_k$, we obtain from the update rule in (3) that

$$\begin{aligned} & \langle \mathbf{z}_{k+1} - \mathbf{z}_k + a_k \nabla f(\mathbf{y}_k), \mathbf{x}^* - \mathbf{z}_{k+1} \rangle \geq 0 \\ \Leftrightarrow \quad & a_k \langle \nabla f(\mathbf{y}_k), \mathbf{z}_{k+1} - \mathbf{x}^* \rangle \leq \langle \mathbf{z}_{k+1} - \mathbf{z}_k, \mathbf{x}^* - \mathbf{z}_{k+1} \rangle \\ \Leftrightarrow \quad & a_k \langle \nabla f(\mathbf{y}_k), \mathbf{z}_{k+1} - \mathbf{x}^* \rangle \leq \frac{1}{2}\|\mathbf{z}_k - \mathbf{x}^*\|^2 - \frac{1}{2}\|\mathbf{z}_{k+1} - \mathbf{x}^*\|^2 - \frac{1}{2}\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2. \end{aligned} \tag{17}$$

Combining (16) and (17) leads to

$$\begin{aligned} A_{k+1}(f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*)) + \frac{1}{2}\|\mathbf{z}_{k+1} - \mathbf{x}^*\|^2 - \left(A_k(f(\mathbf{x}_k) - f(\mathbf{x}^*)) + \frac{1}{2}\|\mathbf{z}_k - \mathbf{x}^*\|^2\right) \\ \leq \frac{1}{2}\left(\frac{L_f a_k^2}{A_{k+1}} - 1\right)\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2, \end{aligned}$$

which proves the claim in (10).

Next, we proceed to prove the claim in (11). To do so, we first leverage the convexity of g which leads to

$$g(\mathbf{y}_k) - g(\mathbf{x}_k) \leq \langle \nabla g(\mathbf{y}_k), \mathbf{y}_k - \mathbf{x}_k \rangle . \tag{18}$$

Also, since g is L_g-smooth, we have

$$g(\mathbf{x}_{k+1}) \leq g(\mathbf{y}_k) + \langle \nabla g(\mathbf{y}_k), \mathbf{x}_{k+1} - \mathbf{y}_k \rangle + \frac{L_g}{2} \|\mathbf{x}_{k+1} - \mathbf{y}_k\|^2. \tag{19}$$

By multiplying both sides of (18) and (19) by A_k and A_{k+1}, respectively, and adding the resulted inequalities we obtain

$$\begin{aligned}
&A_{k+1}(g(\mathbf{x}_{k+1}) - g(\mathbf{y}_k)) + A_k(g(\mathbf{y}_k) - g(\mathbf{x}_k)) \\
&\leq A_{k+1} \langle \nabla g(\mathbf{y}_k), \mathbf{x}_{k+1} - \mathbf{y}_k \rangle + A_k \langle \nabla g(\mathbf{y}_k), \mathbf{y}_k - \mathbf{x}_k \rangle + \frac{L_g A_{k+1}}{2} \|\mathbf{x}_{k+1} - \mathbf{y}_k\|^2 \\
&= a_k \langle \nabla g(\mathbf{y}_k), \mathbf{z}_{k+1} - \mathbf{z}_k \rangle + A_k \langle \nabla g(\mathbf{y}_k), \mathbf{y}_k - \mathbf{x}_k \rangle + \frac{L_g a_k^2}{2A_{k+1}} \|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2 \\
&= a_k \langle \nabla g(\mathbf{y}_k), \mathbf{z}_{k+1} - \mathbf{y}_k \rangle + \frac{L_g a_k^2}{2A_{k+1}} \|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2,
\end{aligned}$$

where the first equality holds since $a_k(\mathbf{z}_{k+1} - \mathbf{z}_k) = A_{k+1}(\mathbf{x}_{k+1} - \mathbf{y}_k)$, and the second equality holds since $a_k(\mathbf{z}_k - \mathbf{y}_k) = A_k(\mathbf{y}_k - \mathbf{x}_k)$. Lastly, by the definition of the constructed cutting plane, we know that $g(\mathbf{y}_k) + \langle \nabla g(\mathbf{y}_k), \mathbf{z} - \mathbf{y}_k \rangle \leq g_k$ for any $\mathbf{z} \in \mathcal{X}_k$. Hence, $\langle \nabla g(\mathbf{y}_k), \mathbf{z}_{k+1} - \mathbf{y}_k \rangle$ is upper bounded by $g_k - g(\mathbf{y}_k)$. Applying this substitution into to the above expression would lead to the claim in (11). □

Now we are ready to prove Theorem 4.1.

Proof of Theorem 4.1. To begin with, note that by our choice of a_k, we have

$$a_k = \frac{k+1}{4L_f} \quad \text{and} \quad A_{k+1} = \frac{(k+1)(k+2)}{8L_f}. \tag{20}$$

Thus, it can be verified that $L_f a_k^2 \leq \frac{1}{2} A_{k+1}$. Then it follows from Lemma A.1 that

$$\begin{aligned}
A_{k+1}(f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*)) + \frac{1}{2}\|\mathbf{z}_{k+1} - \mathbf{x}^*\|^2 - \Big(A_k(f(\mathbf{x}_k) - f(\mathbf{x}^*)) + \frac{1}{2}\|\mathbf{z}_k - \mathbf{x}^*\|^2\Big) \\
\leq \left(\frac{L_f a_k^2}{2A_{k+1}} - \frac{1}{2}\right) \|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2,
\end{aligned} \tag{21}$$

$$A_{k+1}(g(\mathbf{x}_{k+1}) - g(\mathbf{x}^*)) - A_k(g(\mathbf{x}_k) - g(\mathbf{x}^*)) \leq a_k(g_k - g(\mathbf{x}^*)) + \frac{L_g a_k^2}{2A_{k+1}} \|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2. \tag{22}$$

We first prove the convergence guarantee for the upper-level objective. By using induction on (21), we obtain that for any $k \geq 0$

$$A_k(f(\mathbf{x}_k) - f(\mathbf{x}^*)) + \frac{1}{2}\|\mathbf{z}_k - \mathbf{x}^*\|^2 \leq A_0(f(\mathbf{x}_0) - f(\mathbf{x}^*)) + \frac{1}{2}\|\mathbf{z}_0 - \mathbf{x}^*\|^2 = \frac{1}{2}\|\mathbf{z}_0 - \mathbf{x}^*\|^2, \tag{23}$$

which implies

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{\|\mathbf{z}_0 - \mathbf{x}^*\|^2}{2A_k} = \frac{4L_f\|\mathbf{z}_0 - \mathbf{x}^*\|^2}{k(k+1)}.$$

We proceed to establish an upper bound on $g(\mathbf{x}_k) - g(\mathbf{x}^*)$. By summing the inequality in (22) from 0 to $k-1$ we obtain

$$\begin{aligned}
A_k(g(\mathbf{x}_k) - g(\mathbf{x}^*)) &\leq \sum_{i=0}^{k-1} a_i(g_i - g^*) + \frac{L_g}{4L_f} \sum_{i=0}^{k-1} \|\mathbf{z}_{i+1} - \mathbf{z}_i\|^2 \\
&\leq \sum_{i=0}^{k-1} \frac{i+1}{4L_f} \frac{2L_g\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{(i+1)^2} + \frac{L_g}{4L_f} \sum_{i=0}^{k-1} D^2 \\
&\leq \frac{L_g}{2L_f}\|\mathbf{x}_0 - \mathbf{x}^*\|^2(\ln k + 1) + \frac{L_g}{4L_f} D^2 k.
\end{aligned} \tag{24}$$

Note that the second inequality holds due to the condition in (5). Thus, we obtain

$$g(\mathbf{x}_k) - g(\mathbf{x}^*) \leq \frac{4L_g\|\mathbf{x}_0 - \mathbf{x}^*\|^2(\ln k + 1)}{k(k+1)} + \frac{2L_g D^2}{k+1}.$$

The above upper bound on $g(\mathbf{x}_k) - g(\mathbf{x}^*)$ without any additional condition, but next we show that if $f(\mathbf{x}_k) \geq f(\mathbf{x}^*)$ the above upper bound can be further improved as we can upper bound $\sum_{i=0}^{k-1} \|\mathbf{z}_{i+1} - \mathbf{z}_i\|^2$ by a constant independent of k instead of kD^2. To prove this claim, by summing the inequality in (10) from 0 to $k-1$ we obtain

$$\frac{1}{4}\sum_{i=0}^{k-1} \|\mathbf{z}_{i+1} - \mathbf{z}_i\|^2 \leq \frac{1}{2}\|\mathbf{z}_0 - \mathbf{x}^*\|^2 - \left(A_k(f(\mathbf{x}_k) - f(\mathbf{x}^*)) + \frac{1}{2}\|\mathbf{z}_k - \mathbf{x}^*\|^2\right). \tag{25}$$

Hence, if $f(\mathbf{x}_k) \geq f(\mathbf{x}^*)$, then it holds

$$\sum_{i=0}^{k-1} \|\mathbf{z}_{i+1} - \mathbf{z}_i\|^2 \leq 2\|\mathbf{z}_0 - \mathbf{x}^*\|^2. \tag{26}$$

Thus if replace $\sum_{i=0}^{k-1} \|\mathbf{z}_{i+1} - \mathbf{z}_i\|^2$ in (24) by $2\|\mathbf{z}_0 - \mathbf{x}^*\|^2$, we would obtain the following improve bound:

$$g(\mathbf{x}_k) - g(\mathbf{x}^*) \leq \frac{4L_g\|\mathbf{x}_0 - \mathbf{x}^*\|^2(\ln k + 1)}{k(k+1)} + \frac{4L_g\|\mathbf{z}_0 - \mathbf{x}^*\|^2}{k(k+1)}.$$

□

Recall Remark 4.1. The main difficulty in obtaining an accelerated rate of $\mathcal{O}(1/K^2)$ for g is that it is unclear how to control $\sum_{k=0}^{K-1} \|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2$. This, in turn, is because we don't know how to prove a **lower bound** on f. Instead, we used the compactness of $\mathcal{Z}$ to achieve the $\mathcal{O}(1/K)$ for g.

A.2 Proof of Lemma 4.3

Proof of Lemma 4.3. Note that by multiplying both sides of (10) by $\lambda > 0$ we have

$$\begin{aligned} A_{k+1}(\lambda(f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*))) + \frac{\lambda}{2}\|\mathbf{z}_{k+1} - \mathbf{x}^*\|^2 - \left(A_k(\lambda(f(\mathbf{x}_k) - f(\mathbf{x}^*))) + \frac{\lambda}{2}\|\mathbf{z}_k - \mathbf{x}^*\|^2\right) \\ \leq \frac{\lambda}{2}\left(\frac{L_f a_k^2}{A_{k+1}} - 1\right)\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2 \end{aligned} \tag{27}$$

Further note that in this case we have $a_k = \frac{\gamma(k+1)}{4L_f}$ and $A_{k+1} = \gamma\frac{(k+1)(k+2)}{8L_f}$. Hence, a_k^2/A_{k+1} is bounded above by $\frac{\gamma}{2L_f}$. Therefore, we can replace a_k^2/A_{k+1} in the above expression by $\frac{\gamma}{2L_f}$ to obtain

$$\begin{aligned} A_{k+1}(\lambda(f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*))) + \frac{\lambda}{2}\|\mathbf{z}_{k+1} - \mathbf{x}^*\|^2 - \left(A_k(\lambda(f(\mathbf{x}_k) - f(\mathbf{x}^*))) + \frac{\lambda}{2}\|\mathbf{z}_k - \mathbf{x}^*\|^2\right) \\ \leq \frac{\lambda}{2}\left(\frac{\gamma}{2} - 1\right)\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2 \end{aligned} \tag{28}$$

Similarly, we can replace a_k^2/A_{k+1} in (11) by $\frac{\gamma}{2L_f}$ to obtain

$$A_{k+1}(g(\mathbf{x}_{k+1}) - g(\mathbf{x}^*)) - A_k(g(\mathbf{x}_k) - g(\mathbf{x}^*)) \leq a_k(g_k - g(\mathbf{x}^*)) + \frac{\gamma L_g}{4L_f}\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2 \tag{29}$$

Note if we sum up the two inequalities above, we obtain

$$\begin{aligned} &A_{k+1}(\lambda(f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*)) + g(\mathbf{x}_{k+1}) - g(\mathbf{x}^*)) + \frac{\lambda}{2}\|\mathbf{z}_{k+1} - \mathbf{x}^*\|^2 \\ &\quad - \left(A_k(\lambda(f(\mathbf{x}_k) - f(\mathbf{x}^*)) + g(\mathbf{x}_k) - g(\mathbf{x}^*)) + \frac{\lambda}{2}\|\mathbf{z}_k - \mathbf{x}^*\|^2\right) \\ \leq (\lambda(\frac{\gamma}{4} - \frac{1}{2}) + \frac{\gamma}{4}\frac{L_g}{L_f})\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2 + a_k(g_k - g(\mathbf{x}^*)) &\leq a_k(g_k - g(\mathbf{x}^*)) \end{aligned} \tag{30}$$

Note that the last inequality holds since the first term is negative due to the choice of λ. By summing the inequalities from 0 to $k-1$ we obtain

$$A_k(\lambda(f(\mathbf{x}_k)-f(\mathbf{x}^*))+g(\mathbf{x}_k)-g(\mathbf{x}^*))+\frac{1}{2}\lambda\|\mathbf{z}_k-\mathbf{x}^*\|^2 \leq \sum_{i=0}^{k-1} a_i(g_i-g(\mathbf{x}^*))+\frac{1}{2}\lambda\|\mathbf{z}_0-\mathbf{x}^*\|^2 \quad (31)$$

Now using the condition on g_i in (5) and the definition of a_i we have

$$\sum_{i=0}^{k-1} a_i(g_i - g(\mathbf{x}^*)) \leq \sum_{i=0}^{k-1} \frac{\gamma(i+1)}{4L_f}\frac{2L_g\|\mathbf{x}_0-\mathbf{x}^*\|^2}{(i+1)^2} \leq \frac{\gamma L_g}{2L_f}\|\mathbf{x}_0-\mathbf{x}^*\|^2(\ln k+1) \quad (32)$$

By applying this upper bound into (31) we obtain

$$A_k(\lambda(f(\mathbf{x}_k)-f(\mathbf{x}^*))+g(\mathbf{x}_k)-g(\mathbf{x}^*))+\frac{1}{2}\lambda\|\mathbf{z}_k-\mathbf{x}^*\|^2 \leq \frac{\gamma L_g}{2L_f}\|\mathbf{x}_0-\mathbf{x}^*\|^2(\ln k+1)+\frac{1}{2}\lambda\|\mathbf{z}_0-\mathbf{x}^*\|^2 \quad (33)$$

If we drop the $\frac{1}{2}\lambda\|\mathbf{z}_k-\mathbf{x}^*\|^2$ in the left-hand side and divide both sides of the resulted inequality by A_k which is equal to $A_k = \gamma\frac{k(k+1)}{8L_f}$ we obtain

$$\lambda(f(\mathbf{x}_k)-f(\mathbf{x}^*))+g(\mathbf{x}_k)-g(\mathbf{x}^*) \leq \frac{4L_g\|\mathbf{x}_0-\mathbf{x}^*\|^2(\ln k+1)}{k(k+1)}+\frac{4\lambda L_f\|\mathbf{z}_0-\mathbf{x}^*\|^2}{\gamma k(k+1)}, \quad (34)$$

and the proof is complete. □

A.3 Proof of Theorem 4.4

Proof of Theorem 4.4. Recall the result of Lemma 4.3 that if $a_k = \gamma(k+1)/(4L_f)$, where $0<\gamma\leq 1$ and $\lambda \geq \frac{L_g}{(2/\gamma-1)L_f}$ then after K iterations we have

$$\lambda(f(\mathbf{x}_K)-f(\mathbf{x}^*))+g(\mathbf{x}_K)-g(\mathbf{x}^*) \leq \frac{4L_g\|\mathbf{x}_0-\mathbf{x}^*\|^2(\ln K+1)}{K(K+1)}+\frac{4\lambda L_f\|\mathbf{z}_0-\mathbf{x}^*\|^2}{\gamma K(K+1)}. \quad (35)$$

Now if we replace γ by $1/(\frac{2L_g}{L_f}K^{\frac{2r-2}{2r-1}}+2)$ as suggested in the statement of the theorem, we would obtain

$$\begin{aligned}&\lambda(f(\mathbf{x}_K)-f(\mathbf{x}^*))+g(\mathbf{x}_K)-g(\mathbf{x}^*)\\ &\leq \frac{4L_g\|\mathbf{x}_0-\mathbf{x}^*\|^2(\ln K+1)}{K(K+1)}+\frac{8(\frac{L_g}{L_f}K^{\frac{2r-2}{2r-1}}+1)\lambda L_f\|\mathbf{z}_0-\mathbf{x}^*\|^2}{K(K+1)}.\end{aligned} \quad (36)$$

Now we proceed to prove the first claim which is an upper bound on $f(\mathbf{x}_K)-f(\mathbf{x}^*)$. Note that given the fact that $g(\mathbf{x}_K)-g(\mathbf{x}^*)>0$ and $\lambda>0$ we can show that

$$f(\mathbf{x}_K)-f(\mathbf{x}^*) \leq \frac{4L_g\|\mathbf{x}_0-\mathbf{x}^*\|^2(\ln K+1)}{\lambda K(K+1)}+\frac{8((\frac{L_g}{L_f}K^{\frac{2r-2}{2r-1}}+1))L_f\|\mathbf{z}_0-\mathbf{x}^*\|^2}{K(K+1)}. \quad (37)$$

If we select λ which is a free parameter as $\lambda = K^{-\frac{2r-2}{2r-1}} \geq \frac{L_g}{(2/\gamma-1)L_f}$ then we obtain

$$f(\mathbf{x}_K)-f(\mathbf{x}^*) \leq \frac{4L_g\|\mathbf{x}_0-\mathbf{x}^*\|^2(\ln K+1)}{K^{\frac{1}{2r-1}}(T+1)}+\frac{8L_g\|\mathbf{z}_0-\mathbf{x}^*\|^2}{K^{\frac{1}{2r-1}}(T+1)}+\frac{8L_f\|\mathbf{z}_0-\mathbf{x}^*\|^2}{K(K+1)}. \quad (38)$$

Given the fact that $\mathbf{x}_0=\mathbf{z}_0$ we can simplify the upper bound to

$$f(\mathbf{x}_K)-f(\mathbf{x}^*) \leq \frac{12L_g\|\mathbf{x}_0-\mathbf{x}^*\|^2(\ln K+1)}{K^{\frac{2r}{2r-1}}}+\frac{8L_f\|\mathbf{z}_0-\mathbf{x}^*\|^2}{K^2}. \quad (39)$$

Next, we proceed to establish an upper bound on $g(\mathbf{x}_K)-g(\mathbf{x}^*)$. We will use the following inequality that holds due to the HEB condition and formally stated in Proposition 4.2:

$$f(\mathbf{x}_K)-f^* \geq -M\left(\frac{r(g(\mathbf{x}_K)-g(\mathbf{x}^*)))}{\alpha}\right)^{\frac{1}{r}} \quad (40)$$

Now if we replace this lower bound into (36) we would obtain

$$\begin{aligned} &- \lambda M(\frac{r}{\alpha})^{\frac{1}{r}}(g(\mathbf{x}_K) - g(\mathbf{x}^*))^{\frac{1}{r}} + g(\mathbf{x}_K) - g(\mathbf{x}^*) \\ &\leq \frac{4L_g\|\mathbf{x}_0 - \mathbf{x}^*\|^2(\ln K + 1)}{K(K+1)} + \frac{8((\frac{L_g}{L_f}K^{\frac{2r-2}{2r-1}} + 1))\lambda L_f\|\mathbf{z}_0 - \mathbf{x}^*\|^2}{K(K+1)}. \end{aligned} \tag{41}$$

Next we consider two different cases: In the first case we assume $\lambda M(\frac{r}{\alpha})^{\frac{1}{r}}(g(\mathbf{x}_K) - g(\mathbf{x}^*))^{\frac{1}{r}} \leq \frac{1}{2}(g(\mathbf{x}_K) - g(\mathbf{x}^*))$ holds and in the second case we assume the opposite of this inequality holds.

If we are in the first case and $\lambda M(\frac{r}{\alpha})^{\frac{1}{r}}(g(\mathbf{x}_K) - g(\mathbf{x}^*))^{\frac{1}{r}} \leq \frac{1}{2}(g(\mathbf{x}_K) - g(\mathbf{x}^*))$, then the inequality in (41) leads to

$$\frac{1}{2}(g(\mathbf{x}_K) - g(\mathbf{x}^*)) \leq \frac{4L_g\|\mathbf{x}_0 - \mathbf{x}^*\|^2(\ln K + 1)}{K(K+1)} + \frac{8((\frac{L_g}{L_f}K^{\frac{2r-2}{2r-1}} + 1))\lambda L_f\|\mathbf{z}_0 - \mathbf{x}^*\|^2}{K(K+1)}. \tag{42}$$

Since $\lambda = K^{-\frac{2r-2}{2r-1}}$, it further leads to the following upper bound

$$g(\mathbf{x}_K) - g(\mathbf{x}^*) \leq \frac{8L_g\|\mathbf{x}_0 - \mathbf{x}^*\|^2(\ln K + 1)}{K(K+1)} + \frac{16L_g\|\mathbf{z}_0 - \mathbf{x}^*\|^2}{K(K+1)} + \frac{16L_f\|\mathbf{z}_0 - \mathbf{x}^*\|^2}{K^{\frac{1}{2r-1}}(K+1)}. \tag{43}$$

Now given the fact that $\mathbf{x}_0 = \mathbf{z}_0$, we obtain

$$g(\mathbf{x}_K) - g(\mathbf{x}^*) \leq \frac{24L_g\|\mathbf{x}_0 - \mathbf{x}^*\|^2(\ln K + 1)}{K^2} + \frac{16L_f\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{K^{\frac{2r}{2r-1}}}. \tag{44}$$

If we are in the second case and $\lambda M(\frac{r}{\alpha})^{\frac{1}{r}}(g(\mathbf{x}_K) - g(\mathbf{x}^*))^{\frac{1}{r}} > \frac{1}{2}(g(\mathbf{x}_K) - g(\mathbf{x}^*))$ then this inequality is equivalent to

$$(g(\mathbf{x}_K) - g(\mathbf{x}^*))^{1-1/r} \leq 2\lambda M(\frac{r}{\alpha})^{\frac{1}{r}}, \tag{45}$$

leading to

$$g(\mathbf{x}_K) - g(\mathbf{x}^*) \leq \left(2K^{-\frac{2r-2}{2r-1}}M(\frac{r}{\alpha})^{\frac{1}{r}}\right)^{\frac{r}{r-1}} = \frac{(2M)^{\frac{r}{r-1}}(\frac{r}{\alpha})^{\frac{1}{r-1}}}{K^{\frac{2r}{2r-1}}} \tag{46}$$

By combining the bounds in (44) and (46) we realize that

$$g(\mathbf{x}_K) - g(\mathbf{x}^*) \leq \max\{\frac{24L_g\|\mathbf{x}_0 - \mathbf{x}^*\|^2(\ln K + 1)}{K^2} + \frac{16L_f\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{K^{\frac{2r}{2r-1}}}, \frac{(2M)^{\frac{r}{r-1}}(\frac{r}{\alpha})^{\frac{1}{r-1}}}{K^{\frac{2r}{2r-1}}}\} \tag{47}$$

Finally by using the above bound in (47) and the result of Proposition 4.2 we can prove the the second claim and establish a lower bound on $f(\mathbf{x}_K) - f^*$ which is

$$\begin{aligned} &f(\mathbf{x}_K) - f^* \geq \\ &- M\left(\frac{r}{\alpha}\right)^{\frac{1}{r}}\left(\max\{\frac{24L_g\|\mathbf{x}_0 - \mathbf{x}^*\|^2(\ln K + 1)}{K^2} + \frac{16L_f\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{K^{\frac{2r}{2r-1}}}, \frac{(2M)^{\frac{r}{r-1}}(\frac{r}{\alpha})^{\frac{1}{r-1}}}{K^{\frac{2r}{2r-1}}}\}\right)^{1/r} \end{aligned} \tag{48}$$

leading to

$$\begin{aligned} &f(\mathbf{x}_K) - f^* \geq -M\left(\frac{r}{\alpha}\right)^{\frac{1}{r}} \\ &\left(\max\{\frac{(24L_g\|\mathbf{x}_0 - \mathbf{x}^*\|^2(\ln K + 1))^{1/r}}{K^{2/r}} + \frac{(16L_f\|\mathbf{x}_0 - \mathbf{x}^*\|^2)^{1/r}}{K^{\frac{2}{2r-1}}}, \frac{((2M)^{\frac{r}{r-1}}(\frac{r}{\alpha})^{\frac{1}{r-1}})^{1/r}}{K^{\frac{2}{2r-1}}}\}\right) \end{aligned} \tag{49}$$

□

A.4 Proof of Theorem 4.5

Proof of Theorem 4.5. To upper bound $f(\mathbf{x}_k)-f(\mathbf{x}^*)$, we follow a similar analysis as in Theorem 4.1. Specifically, first note that by our choice of a_k, we have $a_k = \gamma\frac{k+1}{4L_f}$ and $A_{k+1} = \gamma\frac{(k+1)(k+2)}{8L_f}$, where $\gamma \in (0,1)$. Hence, we can obtain that $L_f a_k^2 \leq \frac{\gamma}{2}A_{k+1}$. By using Lemma A.1 and the fact that $\gamma \in (0,1)$, we have

$$A_{k+1}(f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*)) + \frac{1}{2}\|\mathbf{z}_{k+1} - \mathbf{x}^*\|^2 - \Big(A_k(f(\mathbf{x}_k) - f(\mathbf{x}^*)) + \frac{1}{2}\|\mathbf{z}_k - \mathbf{x}^*\|^2\Big) \leq \left(\frac{\gamma}{4} - \frac{1}{2}\right)\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2 \leq 0.$$

By using induction, we obtain that for any $k \geq 0$

$$A_k(f(\mathbf{x}_k) - f(\mathbf{x}^*)) + \frac{1}{2}\|\mathbf{z}_k - \mathbf{x}^*\|^2 \leq A_0(f(\mathbf{x}_0) - f(\mathbf{x}^*)) + \frac{1}{2}\|\mathbf{z}_0 - \mathbf{x}^*\|^2 = \frac{1}{2}\|\mathbf{z}_0 - \mathbf{x}^*\|^2, \quad (50)$$

Since $A_k = \gamma\frac{k(k+1)}{8L_f}$ and $\mathbf{z}_0 = \mathbf{x}_0$, this further implies that

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{\|\mathbf{z}_0 - \mathbf{x}^*\|^2}{2A_k} = \frac{4L_f\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{\gamma k(k+1)}.$$

Next, we will prove the upper bound on $g(\mathbf{x}_k) - g(\mathbf{x}^*)$. By Lemma 4.3, we have for any $k \geq 0$

$$\lambda(f(\mathbf{x}_k) - f(\mathbf{x}^*)) + g(\mathbf{x}_k) - g(\mathbf{x}^*) \leq \frac{4L_g\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{k(k+1)}(\ln k + 1) + \frac{4\lambda L_f\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{\gamma k(k+1)} \quad (51)$$

Moreover, since g satisfies the weak sharpness condition, we can use Proposition 4.2 with $r = 1$ to write

$$f(\mathbf{x}_k) - f^* \geq -\frac{M}{\alpha}(g(\mathbf{x}_k) - g^*). \quad (52)$$

Combining (51) and (52) leads to

$$-\lambda\frac{M}{\alpha}(g(\mathbf{x}_k) - g(\mathbf{x}^*)) + g(\mathbf{x}_k) - g(\mathbf{x}^*) \leq \frac{4L_g\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{k(k+1)}(\ln k + 1) + \frac{4\lambda L_f\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{\gamma k(k+1)} \quad (53)$$

Note that we can choose λ to be any number satisfying $\lambda \geq \frac{L_g}{(2/\gamma-1)L_f}$ (cf. Lemma 4.3). Specifically, since $\gamma \leq \frac{2\alpha L_f}{2ML_g+\alpha L_f}$, we can set $\lambda = \alpha/(2M)$ and accordingly (53) can be simplified to

$$g(\mathbf{x}_k) - g(\mathbf{x}^*) \leq \frac{8L_g\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{k(k+1)}(\ln k + 1) + \frac{4\alpha L_f\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{\gamma M k(k+1)}.$$

Finally, we use (52) again together with the above upper bound on $g(\mathbf{x}_k) - g(\mathbf{x}^*)$ to obtain

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \geq -\frac{M}{\alpha}(g(\mathbf{x}_k) - g(\mathbf{x}^*)) \geq -\left(\frac{8ML_g\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{\alpha k(k+1)}(\ln k + 1) + \frac{4L_f\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{\gamma k(k+1)}\right).$$

□

B Extension to the Non-smooth/Composite Setting

In this section, we would like to mention the possible extension to the non-smooth/composite setting. In the general non-smooth settings, we believe it is not possible to extend our results and achieve the purpose of the acceleration. This is because, even in the single-level setting, the best achievable rate in the general non-smooth setting is $\mathcal{O}(1/\sqrt{K})$ achieved by sub-gradient method. That said, it should be possible to extend our accelerated bilevel framework to a special non-smooth setting where the upper- and lower-level objective functions have a composite structure, i.e., they can be written as the sum of a convex smooth function and a convex non-smooth function that is easy to compute its proximal operator.

Note that the properties of smoothness of f and g have only been used in the proof of Lemma A.1 in Section A. None of the other results will break if (10) and (11) in Lemma A.1 still hold in the composite setting. Now, we present and prove the counterpart of Lemma A.1 in the composite setting.

Lemma B.1. *Suppose f_1, f_2, g_1, g_2 are convex and f_1, g_1 are L_f-smooth and L_g-smooth, respectively. Let $\{\mathbf{x}_k\}$ be the sequence of iterates generated by Algorithm 2 with stepsize $a_k > 0$ for $k \geq 0$. Moreover, suppose Assumption 3.1 holds. Then we have*

$$\begin{aligned} A_{k+1}(f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*)) + \frac{1}{2}\|\mathbf{z}_{k+1} - \mathbf{x}^*\|^2 - \Big(A_k(f(\mathbf{x}_k) - f(\mathbf{x}^*)) + \frac{1}{2}\|\mathbf{z}_k - \mathbf{x}^*\|^2\Big) \\ \leq \left(\frac{L_f a_k^2}{2A_{k+1}} - \frac{1}{2}\right)\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2, \end{aligned} \tag{54}$$

$$A_{k+1}(g(\mathbf{x}_{k+1}) - g(\mathbf{x}^*)) - A_k(g(\mathbf{x}_k) - g(\mathbf{x}^*)) \leq a_k(g_k - g(\mathbf{x}^*)) + \frac{L_g a_k^2}{2A_{k+1}}\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2. \tag{55}$$

Proof of Lemma B.1. Let $\mathbf{x}^*$ be any optimal solution of (6).

We first consider the upper-level objective f. Since f_1 is convex, we have

$$f_1(\mathbf{y}_k) - f_1(\mathbf{x}^*) \leq \langle \nabla f_1(\mathbf{y}_k), \mathbf{y}_k - \mathbf{x}^* \rangle, \quad f_1(\mathbf{y}_k) - f_1(\mathbf{x}_k) \leq \langle \nabla f_1(\mathbf{y}_k), \mathbf{y}_k - \mathbf{x}_k \rangle. \tag{56}$$

Now given the update rule $A_{k+1} = A_k + a_k$, we can write

$$A_{k+1}(f_1(\mathbf{y}_k) - f_1(\mathbf{x}^*)) - A_k(f_1(\mathbf{x}_k) - f_1(\mathbf{x}^*)) = a_k(f_1(\mathbf{y}_k) - f_1(\mathbf{x}^*)) + A_k(f_1(\mathbf{y}_k) - f_1(\mathbf{x}_k)) \tag{57}$$

Combining (56) and (57), we have

$$\begin{aligned} &A_{k+1}(f_1(\mathbf{y}_k) - f_1(\mathbf{x}^*)) - A_k(f_1(\mathbf{x}_k) - f_1(\mathbf{x}^*)) \\ &\leq a_k(\langle \nabla f_1(\mathbf{y}_k), \mathbf{y}_k - \mathbf{x}^* \rangle) + A_k(\langle \nabla f_1(\mathbf{y}_k), \mathbf{y}_k - \mathbf{x}_k \rangle) \\ &= \langle \nabla f_1(\mathbf{y}_k), a_k\mathbf{y}_k + A_k(\mathbf{y}_k - \mathbf{x}_k) - a_k\mathbf{x}^* \rangle \\ &= a_k \langle \nabla f_1(\mathbf{y}_k), \mathbf{z}_k - \mathbf{x}^* \rangle, \end{aligned} \tag{58}$$

where the last equality follows from the definition of $\mathbf{y}_k$. Furthermore, since f_1 is L_f-smooth, we have

$$f_1(\mathbf{x}_{k+1}) \leq f_1(\mathbf{y}_k) + \langle \nabla f_1(\mathbf{y}_k), \mathbf{x}_{k+1} - \mathbf{y}_k \rangle + \frac{L_f}{2}\|\mathbf{x}_{k+1} - \mathbf{y}_k\|^2. \tag{59}$$

If we multiply both sides of (59) by A_{k+1} and combine the resulting inequality with (58), we obtain

$$\begin{aligned} &A_{k+1}(f_1(\mathbf{x}_{k+1}) - f_1(\mathbf{x}^*)) - A_k(f_1(\mathbf{x}_k) - f_1(\mathbf{x}^*)) \\ &\leq a_k \langle \nabla f_1(\mathbf{y}_k), \mathbf{z}_k - \mathbf{x}^* \rangle + A_{k+1} \langle \nabla f_1(\mathbf{y}_k), \mathbf{x}_{k+1} - \mathbf{y}_k \rangle + \frac{L_f A_{k+1}}{2}\|\mathbf{x}_{k+1} - \mathbf{y}_k\|^2 \\ &= a_k \langle \nabla f_1(\mathbf{y}_k), \mathbf{z}_k - \mathbf{x}^* \rangle + a_k \langle \nabla f_1(\mathbf{y}_k), \mathbf{z}_{k+1} - \mathbf{z}_k \rangle + \frac{L_f a_k^2}{2A_{k+1}}\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2 \\ &= a_k \langle \nabla f_1(\mathbf{y}_k), \mathbf{z}_{k+1} - \mathbf{x}^* \rangle + \frac{L_f a_k^2}{2A_{k+1}}\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2, \end{aligned} \tag{60}$$

where we used the fact that $a_k(\mathbf{z}_{k+1} - \mathbf{z}_k) = A_{k+1}(\mathbf{x}_{k+1} - \mathbf{y}_k)$ in the first equality. Moreover, from the step 6 in Algorithm 2, we have $\mathbf{z}_k - a_k\nabla f_1(\mathbf{y}_k) - \mathbf{z}_{k+1} \in a_k\partial(f_2(\mathbf{z}_{k+1}) + \delta_{\mathcal{X}_k}(\mathbf{z}_{k+1}))$. Using this, from the definition of subgradients for $f_2 + \delta_{\mathcal{X}_k}$, we have

$$\begin{aligned} &\langle \mathbf{x}^* - \mathbf{z}_{k+1}, \mathbf{z}_k - a_k\nabla f_1(\mathbf{y}_k) - \mathbf{z}_{k+1} \rangle \leq a_k(f_2(\mathbf{x}^*) + \delta_{\mathcal{X}_k}(\mathbf{x}^*) - f_2(\mathbf{z}_{k+1}) - \delta_{\mathcal{X}_k}(\mathbf{z}_{k+1})) \\ \Leftrightarrow \quad &\langle \mathbf{z}_{k+1} - \mathbf{z}_k + a_k\nabla f(\mathbf{y}_k), \mathbf{x}^* - \mathbf{z}_{k+1} \rangle \geq a_k f_2(\mathbf{z}_{k+1}) - a_k f_2(\mathbf{x}^*) \\ \Leftrightarrow \quad &a_k \langle \nabla f(\mathbf{y}_k), \mathbf{z}_{k+1} - \mathbf{x}^* \rangle \leq \langle \mathbf{z}_{k+1} - \mathbf{z}_k, \mathbf{x}^* - \mathbf{z}_{k+1} \rangle - a_k f_2(\mathbf{z}_{k+1}) + a_k f_2(\mathbf{x}^*) \\ \Leftrightarrow \quad &a_k \langle \nabla f(\mathbf{y}_k), \mathbf{z}_{k+1} - \mathbf{x}^* \rangle \leq \frac{1}{2}\|\mathbf{z}_k - \mathbf{x}^*\|^2 - \frac{1}{2}\|\mathbf{z}_{k+1} - \mathbf{x}^*\|^2 - \frac{1}{2}\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2 \\ &\qquad\qquad - a_k f_2(\mathbf{z}_{k+1}) + a_k f_2(\mathbf{x}^*). \end{aligned} \tag{61}$$

The first step holds since $\mathbf{x}^*, \mathbf{z}_{k+1} \in \mathcal{X}_k$, i.e. $\delta_{\mathcal{X}_k}(\mathbf{x}^*) = \delta_{\mathcal{X}_k}(\mathbf{z}_{k+1}) = 0$. Combining (60) and (61) leads to

$$\begin{aligned} &A_{k+1}(f_1(\mathbf{x}_{k+1}) - f_1(\mathbf{x}^*)) + \frac{1}{2}\|\mathbf{z}_{k+1} - \mathbf{x}^*\|^2 - \left(A_k(f_1(\mathbf{x}_k) - f_1(\mathbf{x}^*)) + \frac{1}{2}\|\mathbf{z}_k - \mathbf{x}^*\|^2\right) \\ &\leq \frac{1}{2}\left(\frac{L_f a_k^2}{A_{k+1}} - 1\right)\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2 - a_k f_2(\mathbf{z}_{k+1}) + a_k f_2(\mathbf{x}^*), \end{aligned} \tag{62}$$

Then we add $(A_{k+1}f_2(\mathbf{x}_{k+1}) - a_k f_2(\mathbf{x}^*) - A_k f_2(\mathbf{x}_k))$ on both sides to obtain,

$$\begin{aligned} &A_{k+1}(f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*)) + \frac{1}{2}\|\mathbf{z}_{k+1} - \mathbf{x}^*\|^2 - \left(A_k(f(\mathbf{x}_k) - f(\mathbf{x}^*)) + \frac{1}{2}\|\mathbf{z}_k - \mathbf{x}^*\|^2\right) \\ &\leq \frac{1}{2}\left(\frac{L_f a_k^2}{A_{k+1}} - 1\right)\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2 - a_k f_2(\mathbf{z}_{k+1}) + A_{k+1}f_2(\mathbf{x}_{k+1}) - A_k f_2(\mathbf{x}_k), \end{aligned} \tag{63}$$

Finally, by the convexity of f_2, $A_{k+1} = A_k + a_k$, and $\mathbf{x}_{k+1} = \frac{A_k}{A_k+a_k}\mathbf{x}_k + \frac{a_k}{A_k+a_k}\mathbf{z}_{k+1}$, i.e. $-a_k f_2(\mathbf{z}_{k+1}) + A_{k+1}f_2(\mathbf{x}_{k+1}) - A_k f_2(\mathbf{x}_k) \leq 0$, the first inequality of this Lemma can be obtained.

Next, we proceed to prove the claim for the lower-level objective g. To do so, we first leverage the convexity of the smooth part g_1 which leads to

$$g_1(\mathbf{y}_k) - g_1(\mathbf{x}_k) \leq \langle \nabla g_1(\mathbf{y}_k), \mathbf{y}_k - \mathbf{x}_k \rangle . \tag{64}$$

Also, since g_1 is L_g-smooth, we have

$$g_1(\mathbf{x}_{k+1}) \leq g_1(\mathbf{y}_k) + \langle \nabla g_1(\mathbf{y}_k), \mathbf{x}_{k+1} - \mathbf{y}_k \rangle + \frac{L_g}{2}\|\mathbf{x}_{k+1} - \mathbf{y}_k\|^2. \tag{65}$$

By multiplying both sides of (64) and (65) by A_k and A_{k+1}, respectively, and adding the resulted inequalities we obtain

$$\begin{aligned} &A_{k+1}(g_1(\mathbf{x}_{k+1}) - g_1(\mathbf{y}_k)) + A_k(g_1(\mathbf{y}_k) - g_1(\mathbf{x}_k)) \\ &\leq A_{k+1}\langle \nabla g_1(\mathbf{y}_k), \mathbf{x}_{k+1} - \mathbf{y}_k \rangle + A_k \langle \nabla g_1(\mathbf{y}_k), \mathbf{y}_k - \mathbf{x}_k \rangle + \frac{L_g A_{k+1}}{2}\|\mathbf{x}_{k+1} - \mathbf{y}_k\|^2 \\ &= a_k \langle \nabla g_1(\mathbf{y}_k), \mathbf{z}_{k+1} - \mathbf{z}_k \rangle + A_k \langle \nabla g_1(\mathbf{y}_k), \mathbf{y}_k - \mathbf{x}_k \rangle + \frac{L_g a_k^2}{2A_{k+1}}\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2 \\ &= a_k \langle \nabla g_1(\mathbf{y}_k), \mathbf{z}_{k+1} - \mathbf{y}_k \rangle + \frac{L_g a_k^2}{2A_{k+1}}\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2, \end{aligned}$$

where the first equality holds since $a_k(\mathbf{z}_{k+1} - \mathbf{z}_k) = A_{k+1}(\mathbf{x}_{k+1} - \mathbf{y}_k)$, and the second equality holds since $a_k(\mathbf{z}_k - \mathbf{y}_k) = A_k(\mathbf{y}_k - \mathbf{x}_k)$. Lastly, by the definition of the constructed approximated set $\mathcal{X}_k$, we know that $g_1(\mathbf{y}_k) + \langle \nabla g_1(\mathbf{y}_k), \mathbf{z} - \mathbf{y}_k \rangle + g_2(\mathbf{z}) \leq g_k$ for any $\mathbf{z} \in \mathcal{X}_k$. Hence, $\langle \nabla g_1(\mathbf{y}_k), \mathbf{z}_{k+1} - \mathbf{y}_k \rangle$ is upper bounded by $g_k - g_1(\mathbf{y}_k) - g_2(\mathbf{z}_{k+1})$. Applying this substitution into to the above expression to obtain,

$$\begin{aligned} &A_{k+1}(g_1(\mathbf{x}_{k+1}) - g_1(\mathbf{y}_k)) + A_k(g_1(\mathbf{y}_k) - g_1(\mathbf{x}_k)) \\ &\leq a_k g_k - a_k g_1(\mathbf{y}_k) - a_k g_2(\mathbf{z}_{k+1}) + \frac{L_g a_k^2}{2A_{k+1}}\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2 \end{aligned} \tag{66}$$

By adding $a_k g_1(\mathbf{y}_k) - a_k g_1(\mathbf{x}^*)$ on both sides, we have,

$$\begin{aligned} &A_{k+1}(g_1(\mathbf{x}_{k+1}) - g_1(\mathbf{x}^*)) + A_k(g_1(\mathbf{x}^*) - g_1(\mathbf{x}_k)) \\ &\leq a_k g_k - a_k g_1(\mathbf{x}^*) - a_k g_2(\mathbf{z}_{k+1}) + \frac{L_g a_k^2}{2A_{k+1}}\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2 \end{aligned} \tag{67}$$

Lastly, we add $(A_{k+1}g_2(\mathbf{x}_{k+1}) - a_k g_2(\mathbf{x}^*) - A_k g_2(\mathbf{x}_k))$ on both sides to obtain,

$$\begin{aligned} &A_{k+1}(g(\mathbf{x}_{k+1}) - g(\mathbf{x}^*)) + A_k(g(\mathbf{x}^*) - g(\mathbf{x}_k)) \\ &\leq a_k g_k - a_k g(\mathbf{x}^*) + A_{k+1}g_2(\mathbf{x}_{k+1}) - A_k g_2(\mathbf{x}_k) - a_k g_2(\mathbf{z}_{k+1}) + \frac{L_g a_k^2}{2A_{k+1}}\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2 \end{aligned} \tag{68}$$

By the convexity of g_2, $A_{k+1} = A_k + a_k$, and $\mathbf{x}_{k+1} = \frac{A_k}{A_k+a_k}\mathbf{x}_k + \frac{a_k}{A_k+a_k}\mathbf{z}_{k+1}$ (outlined in Algorithm 2), i.e. $A_{k+1}g_2(\mathbf{x}_{k+1}) - A_k g_2(\mathbf{x}_k) - a_k g_2(\mathbf{z}_{k+1}) \leq 0$, the second inequality of this Lemma can be achieved. □

Hence, with the additional Assumption 3.1, by replicating the analysis outlined in Section A, we can derive identical complexity results for Algorithm 2 in either the compact domain setting or with the Hölderian error bounds on g.

C Additional related work

Previous work has explored "asymptotic" results for simple bilevel problems, dating back to Tikhonov-type regularization introduced in [37]. In this approach, the objectives of both levels are combined into a single-level problem using a regularization parameter $\sigma > 0$ and as $\sigma \to 0$ the solutions of the regularized single-level problem approaches a solution to the bilevel problem in (1). Further, the authors in [38] proposed the explicit descent method that solves problem (1) when upper and lower-level functions are smooth and convex. This result was further extended to a non-smooth setting in [39]. The results in both [38] and [39] only indicated that both upper and lower-level objective functions converge asymptotically. Moreover, the authors in [40] proposed the ϵ-subgradient method to solve simple bilevel problems and showed its asymptotic convergence. Specifically, they assumed the upper-level objective function to be convex and utilized two different algorithms, namely, the Fast Iterative Bilevel Algorithm (FIBA) and Incremental Iterative Bilevel Algorithm (IIBA), that consider smooth and non-smooth lower-level objective functions, respectively.

Some studies have only established non-asymptotic convergence rates for the lower-level problem. One of the pioneering methods in this category is the minimal norm gradient (MNG) method, introduced in [41]. This method assumes that the upper-level objective function is smooth and strongly convex, while the lower-level objective function is smooth and convex. The authors showed that the lower-level objective function reaches an iteration complexity of $\mathcal{O}(1/\epsilon^2)$. Subsequently, the Bilevel Gradient SAM (BiS-SAM) method was introduced in [35], and it was proven to achieve a complexity of $\mathcal{O}(1/\epsilon)$ for the lower-level problem. A similar rate of convergence was also attained in [42].

D Connection with the Polyak Step Size

In this section, we would like to highlight the connection between our algorithm's projection step (outlined in Step 6 of Algorithm 1) and the Polyak step size. To make this connection, we first without loss of generality, replace g_k with g^*. It is a reasonable argument, as g_k values are close to g^*, a point highlighted in (5). In addition, we further assume that the set $\mathcal{Z} = \mathbb{R}^n$ to simplify the expressions. Given these substitutions, the projection step in our AGM-BiO method is equivalent to solving the following problem:

$$\begin{aligned} \min \quad & \|\mathbf{x} - \mathbf{x}_k\|^2 \\ \text{s.t.} \quad & g(\mathbf{x}_k) + \langle \nabla g(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k \rangle \leq g^* \end{aligned}$$

In other words,$\mathbf{x}_{k+1}$ is the unique solution of the above quadratic program with a linear constraint. By writing the optimality conditions for the above problem and considering λ as the Lagrange multipliers associated with the linear constraint, we obtain that

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k - \lambda \nabla g(\mathbf{x}_k) \\ \lambda(g(\mathbf{x}_k) + \langle \nabla g(\mathbf{x}_k), \mathbf{x}_{k+1} - \mathbf{x}_k \rangle - g^*) = 0 \\ \lambda \geq 0 \end{cases}$$

Given the fact that $\mathbf{x}_{k+1} \neq \mathbf{x}_k$, we can conclude that $\lambda \neq 0$, and hence we have

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k - \lambda \nabla g(\mathbf{x}_k) \\ g(\mathbf{x}_k) + \langle \nabla g(\mathbf{x}_k), \mathbf{x}_{k+1} - \mathbf{x}_k \rangle - g^* = 0 \\ \lambda > 0 \end{cases}$$

By replacing $\mathbf{x}_{k+1}$ in the second expression with its expression in the first equation we obtain that

$$\lambda = \frac{g(\mathbf{x}_k) - g^*}{\|\nabla g(\mathbf{x}_k)\|^2}.$$

which is exactly the Polyak step size in the literature [43]. To solve a bilevel optimization problem, we intend to do gradient descent for both upper- and lower-level functions. Tuning the ratio of upper- and lower-level step size is generally hard. However, by connecting the projection step with the Polyak step size, we observe that the stepsize for the lower-level objective is auto-selected as the Polyak stepsize in our method. In other words, it is one of the advantages of our algorithm that we do not need to choose the lower-level stepsize or ratio of the upper- and lower-level stepsize theoretically or empirically.

E Experiment Details

In this section, we include more details of the numerical experiments in Section 5. All simulations are implemented using MATLAB R2022a on a PC running macOS Sonoma with an Apple M1 Pro chip and 16GB Memory.

E.1 Over-parametrized Regression

Dataset generation. The original Wikipedia Math Essential dataset [34] composes of a data matrix of size 1068×731. We randomly select one of the columns as the outcome vector $\mathbf{b} \in \mathbb{R}^{1068}$ and the rest to be a new matrix $\mathbf{A} \in \mathbb{R}^{1068\times 730}$. We set the constraint parameter $\lambda = 1$ in this experiment, i.e., the constraint set is given by $\mathcal{Z} = \{\boldsymbol{\beta} \mid \|\boldsymbol{\beta}\|_2 \leq 1\}$.

Implementation details. To be fair, all the algorithms start from the origin as the initial point. For our AGM-BiO method, we set the target tolerances for the absolute suboptimality and infeasibility to $\epsilon_f = 10^{-4}$ and $\epsilon_g = 10^{-4}$, respectively. We choose the stepsizes as $a_k = 10^{-2}(k+1)/(4L_f)$. In each iteration, we need to do a projection onto an intersection of a L_2-ball and a halfspace, which has a closed-form solution. For a-IRG, we set $\eta_0 = 10^{-3}$ and $\gamma_0 = 10^{-3}$. For CG-BiO, we obtain an initial point with FW gap of the lower-level problem less than $\epsilon_g/2 = 5 \times 10^{-5}$ and choose stepsize $\gamma_k = 10^{-2}/(k+2)$. For Bi-SG, we set $\eta_k = 10^{-2}/(k+1)^{0.75}$ and $t_k = 1/L_g = 1/\lambda_{max}(\mathbf{A}_{tr}^\top \mathbf{A}_{tr}) = 1.5 \times 10^{-4}$. For SEA, we set both the lower- and upper-level stepsizes to be 10^{-4}. For R-APM, since the lower-level problem does not satisfy the weak sharpness condition, we set $\eta = 1/(K+1) = 1.25 \times 10^{-5}$ and $\gamma = 10^{-4} \leq 1/(L_g + \eta L_f)$. For PB-APG, we set the penalty parameter $\gamma = 10^4$. Note that the lower-level problem in this experiment does not satisfy Höderian error bound assumption, so there is no theoretical guarantee for PB-APG.

E.2 Linear inverse problems

Dataset generation. We set $\mathbf{Q} = \mathbf{I}_n$, $\mathbf{A} = \mathbf{1}_n^\top$, and $\mathbf{b} = 1$. The constraint set is selected as $\mathcal{Z} = \mathbb{R}_+^n$. We choose a low dimensional $(n = 3)$ and a high dimensional $(n = 100)$ example and run $K = 10^3$ number of iterations to compare the numerical performance of these algorithms, respectively.

Implementation details. To be fair, all the algorithms start from the same initial point randomly chosen from $\mathbb{R}_+^n$. For our AGM-BiO method, we set the stepsizes as $a_k = \gamma(k+1)/(4L_f)$, where $\gamma = 1/(\frac{2L_g}{L_f}K^{2/3} + 2)$ as suggested in Theorem 4.4. In each iteration, we need to project onto an intersection of a halfspace and $\mathbb{R}_+^n$. Since halfspaces and $\mathbb{R}_+^n$ are both convex and closed set, the projection subproblem can be solved by Dykstra's projection algorithm in [20]. For a-IRG, we set $\eta_0 = 10^{-2}$ and $\gamma_0 = 10^{-2}$. For Bi-SG, we set $\eta_k = 1/(k+1)^{0.75}$ and $t_k = 1/L_g$. For SEA, we set the lower-level stepsize to be 10^{-2} and the upper-level stepsize to be 10^{-2}. For R-APM, since the lower-level problem does not satisfy the weak sharpness condition, we set $\eta = 1/(K+1)$ and $\gamma = 1/(L_g + \eta L_f)$. For PB-APG, we set the penalty parameter $\gamma = 10^4$. For Bisec-BiO, we choose the target tolerances to $\epsilon_f = \epsilon_g = 10^{-4}$. For comparison purposes, we limit the maximum number of gradient evaluations for each APG call to 10^2. In this experiment, $L_f = 1$ and $L_g = n$, where n is the number of dimensions.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We clearly stated our contributions in the introduction aligned with the main claims in the abstract.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The main limitation is that our algorithm requires the compact domain as we stated in Section 3 and 4. We also explained why such an assumption is necessary in Remark 4.1.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Our paper provides the full set of assumptions in Section 2.1 and a complete proof in Section A.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The experimental results are stated in Section 5. The implementation details are included in Section E. The code and data are attached in the supplementary material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code and data are attached in the supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We explained how we performed the experiments in Section 5. Moreover, the implementation details are included in Section E.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Our algorithm is designed for deterministic simple bilevel optimization, which does not include any randomness.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The compute resources we used are stated in Section E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper conforms, in every respect, with the NeurIPS Code of Ethics

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed in this paper.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The data used in the paper are properly credited and cited in Section 5.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.