A Conditional Gradient-based Method for Simple Bilevel Optimization with Convex Lower-level Problem

Ruichen Jiang UT Austin
Nazanin Abolfazli University of Arizona
Aryan Mokhtari UT Austin
Erfan Yazdandoost Hamedani University of Arizona

Abstract

In this paper, we study a class of bilevel optimization problems, also known as simple bilevel optimization, where we minimize a smooth objective function over the optimal solution set of another convex constrained optimization problem. Several iterative methods have been developed for tackling this class of problems. Alas, their convergence guarantees are either asymptotic for the upper-level objective, or the convergence rates are slow and sub-optimal. To address this issue, in this paper, we introduce a novel bilevel optimization method that locally approximates the solution set of the lower-level problem via a cutting plane and then runs a conditional gradient update to decrease the upper-level objective. When the upper-level objective is convex, we show that our method requires $\mathcal{O}(\max\{1/\epsilon_f, 1/\epsilon_g\})$ iterations to find a solution that is ϵ_f-optimal for the upper-level objective and ϵ_g-optimal for the lower-level objective. Moreover, when the upper-level objective is non-convex, our method requires $\mathcal{O}(\max\{1/\epsilon_f^2, 1/(\epsilon_f\epsilon_g)\})$ iterations to find an (ϵ_f, ϵ_g)-optimal solution. We also prove stronger convergence guarantees under the Hölderian error bound assumption on the lower-level problem. To the best of our knowledge, our method achieves the best-known iteration complexity for the considered class of bilevel problems.

1 INTRODUCTION

Bilevel optimization is a form of optimization where one problem is embedded within another. It captures a hierarchical structure, where an *upper-level* function is minimized over the solution set of a *lower-level* problem. This class of problems has attracted great attention due to their applications in hyper-parameter optimization (Franceschi et al., 2018; Shaban et al., 2019), meta-learning (Rajeswaran et al., 2019; Bertinetto et al., 2019), and reinforcement learning (Hong et al., 2020). In this paper, we focus on a specific form of bilevel optimization formally defined as

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

where $\mathcal{Z}$ is a compact convex set and $f, g : \mathbb{R}^d \to \mathbb{R}$ are continuously differentiable functions on an open set containing $\mathcal{Z}$. We assume that g is convex but not necessarily strongly convex, and hence the lower-level problem in (1) could have multiple optimal solutions. We remark that problem (1) is often referred to as the "simple bilevel problem" in the literature (Dempe et al., 2010; Dutta and Pandit, 2020; Shehu et al., 2021) to differentiate it from the more general settings where the lower-level problem is parameterized by some upper-level variables. This class of bilevel problems appears in several settings as discussed in Section 2. The key challenge to solve problem (1) stems from the fact that its feasible set—the solution set of the lower-level problem—does not admit a simple characterization and is not explicitly given. This rules out the possibility of directly applying projection-based methods as well as the conditional gradient (CG) type methods, since projection onto or minimizing a linear objective over the feasible set is intractable.

A possible scheme in this case is reformulating problem (1) as a constrained optimization problem with functional constraints and applying primal-dual methods. Specifically, problem (1) can be written as

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{x} \in \mathcal{Z},\, g(\mathbf{x}) \le g^*, \tag{2}$$

where g^* is the optimal value of the lower-level problem. However, a critical issue is that problem (2) does not satisfy strict feasibility and hence the Slater's condition fails, which is required for most primal-dual methods. Even relaxing the constraint ($g(\mathbf{x}) \le g^* + \epsilon$) to ensure strict feasibility would inevitably lead to numerical issues. In fact, as ϵ approaches zero and the problem becomes nearly degenerate, the dual optimal variable may tend to infinity, which slows down the convergence and leads to numerical instability

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

Table 1: Summary of bilevel optimization algorithms. The abbreviations "SC", "C", and "non-C" stand for "strongly convex", "convex", and "non-convex", respectively.

References	Upper level	Lower level		Convergence		Oracle needed
	Objective f	Objective g	Feasible set $\mathcal{Z}$	Upper level	Lower level	
MNG (Beck and Sabach, 2014)	SC, differentiable	C, smooth	Closed	Asymptotic	$\mathcal{O}(1/\epsilon^2)$	projection
BiG-SAM (Sabach and Shtern, 2017)	SC, smooth	C, composite	Closed	Asymptotic	$\mathcal{O}(1/\epsilon)$	projection
Tseng's method (Malitsky, 2017)	C, composite	C, composite	Closed	Asymptotic	$o(1/\epsilon)$	projection
a-IRG (Kaushik and Yousefian, 2021)	C, Lipschitz	C, Lipschitz	Closed	$\mathcal{O}(\max\{1/\epsilon_f^4, 1/\epsilon_g^4\})$		projection
This paper, Theorem 1	C, smooth	C, smooth	Compact	$\mathcal{O}(\max\{1/\epsilon_f, 1/\epsilon_g\})$		linear solver
This paper, Theorem 2	Non-C, smooth	C, smooth	Compact	$\mathcal{O}(\max\{1/\epsilon_f^2, 1/(\epsilon_f\epsilon_g)\})$		linear solver

(Bonnans and Shapiro, 2013) (A detailed discussion about this instability issue is provided in Appendix D). Therefore, problem (1) cannot be simply treated as a classic constrained optimization problem and calls for new theories and algorithms tailored to its hierarchical structure.

Another approach to solving problem (1) is the Tikhonov-type regularization (Tikhonov and Arsenin, 1977), where the objective functions of both levels are combined using a regularization parameter $\sigma > 0$ to form a single-level problem. It is known that as $\sigma \to 0$, any cluster point of the solutions of the regularized single-level problem is a solution to the bilevel problem in (1). In fact, under certain assumptions as shown in (Friedlander and Tseng, 2008; Dempe et al., 2021), the solution set of problem (1) exactly matches with the regularized problem for sufficiently small σ. Alas, checking such conditions and finding the threshold are often difficult in practice. To avoid this issue, Cabot (2005) and Solodov (2007) proposed adjusting the regularization parameter σ dynamically and proved an asymptotic convergence guarantee. Along another line of research, several works (Yamada, 2001; Xu, 2004) have studied the more general problem of solving a variational inequality over the fixed-point set of a nonexpansive mapping, to which the problem in (1) is a special case. In particular, the hybrid steepest descent method by Yamada (2001) and the sequential averaging method (SAM) by Xu (2004) converge asymptotically to the optimal solution when the parameters are properly chosen. However, these results fail to provide any non-asymptotic guarantee for either the upper- or lower-level objectives.

More recently, there has been a surge of interest in establishing non-asymptotic convergence rates for problem (1). One of the first methods of this kind is the minimal norm gradient (MNG) method proposed by Beck and Sabach (2014). When the upper-level function f is strongly-convex and the lower-level function g is convex and smooth, they showed that MNG converges asymptotically to the optimal solution and achieves a complexity bound of $\mathcal{O}(1/\epsilon^2)$ in terms of the lower-level objective value. Subsequently, built on the SAM framework, Bilevel Gradient SAM (BiG-SAM) was proposed by Sabach and Shtern (2017) and it was shown to achieve a complexity of $\mathcal{O}(1/\epsilon)$ for the lower-level problem; see also Shehu et al. (2021) for a related method. Malitsky (2017) studied a version of Tseng's accelerated gradient method that obtains a convergence rate of $o(1/k)$ for the lower-level problem. However, these prior works only establish convergence rates for the lower-level problem, while the rate for the upper-level objective is missing. The only exception is the work by Kaushik and Yousefian (2021): when f and g are convex and Lipschitz continuous, they showed that an iterative regularization-based method achieves a convergence rate of $\mathcal{O}(1/k^{0.5-b})$ for the upper-level objective and a rate of $\mathcal{O}(1/k^b)$ for the lower-level, where $b \in (0, 0.5)$ is a user-defined parameter. As stated in Table 1, if one sets $b = 0.25$ to balance the two rates, then finding a solution that is ϵ_f-optimal for the upper-level problem and ϵ_g-optimal for the lower-level problem would require a complexity of $\mathcal{O}(\max\{1/\epsilon_f^4, 1/\epsilon_g^4\})$.

Contributions. In this paper, we present a novel conditional gradient-based bilevel optimization (CG-BiO) method with tight non-asymptotic guarantees for both upper- and lower-level problems. At each iteration, our proposed CG-BiO method uses a cutting plane to locally approximate the solution set of the lower-level problem, and then combines it with a CG-type update on the upper-level objective. Our theoretical guarantees for CG-BiO are the following:

- When the upper-level function f is convex, we show that CG-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 $\mathcal{O}(\max\{1/\epsilon_f, 1/\epsilon_g\})$ iterations, where f^* is the optimal value of problem (1) and g^* is the optimal value of the lower-level problem. This guarantee matches the best-known results in terms of the lower-level objective and is optimal for bilevel projection-free methods.
- When f is non-convex, CG-BiO finds $\hat{\mathbf{x}}$ that satisfies $\mathcal{G}(\hat{\mathbf{x}}) \leq \epsilon_f$ and $g(\hat{\mathbf{x}}) - g^* \leq \epsilon_g$ within $\mathcal{O}(\max\{1/\epsilon_f^2, 1/(\epsilon_f\epsilon_g)\})$ iterations, where $\mathcal{G}(\hat{\mathbf{x}})$ is the Frank-Wolfe (FW) gap function (cf. (7)).
- With an additional r-th-order ($r \geq 1$) Hölderian error bound assumption on the lower-level problem, CG-

BiO finds $\hat{\mathbf{x}}$ with $|f(\hat{\mathbf{x}}) - f^*| \leq \epsilon_f$ within $\mathcal{O}(1/\epsilon_f^r)$ iterations in the convex case, and $\hat{\mathbf{x}}$ with $|\mathcal{G}(\hat{\mathbf{x}})| \leq \epsilon_f$ within $\mathcal{O}(1/\epsilon_f^{r+1})$ iterations in the non-convex case.

It is worth noting that the state-of-the-art methods for solving simple bilevel problems (stated in Table 1) require projection onto the set $\mathcal{Z}$ at each iteration. In contrast, as our proposed method is a CG-based method, it requires access to a linear solver instead of projection at each iteration, which is suitable for the settings where projection is computationally costly; e.g., when $\mathcal{Z}$ is a polyhedron.

Additional Related Work. In the general form of bilevel problems, the upper-level function f may also depend on an additional variable $\mathbf{w} \in \mathbb{R}^m$ that in turn influences the lower-level problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d, \mathbf{w}\in\mathbb{R}^m} f(\mathbf{x}, \mathbf{w}) \quad \text{s.t.} \quad \mathbf{x} \in \operatorname*{argmin}_{\mathbf{z}\in\mathcal{Z}} g(\mathbf{z}, \mathbf{w}). \tag{3}$$

Problem (3) has been studied in depth, and we refer the readers to the extensive survey by Dempe (2020). Note that for any fixed $\mathbf{w}$, the above problem boils down to a simple bilevel problem in (1). In recent years, gradient-based methods for problem (3) have become increasingly popular including implicit differentiation (Domke, 2012; Pedregosa, 2016; Gould et al., 2016; Ji et al., 2021) and iterative differentiation (Maclaurin et al., 2015; Franceschi et al., 2018). However, most of the existing methods work under the assumption that the lower-level problem is strongly convex in $\mathbf{z}$ for any $\mathbf{w}$ and thus has a unique minimum. Note that such an assumption would render the simple bilevel problem in (1) trivial, as it amounts to solving the lower-level problem only. More relevant to our work, some concurrent papers consider the case where the lower-level problem can have multiple minima (Liu et al., 2020; Li et al., 2020; Liu et al., 2021a,b; Sow et al., 2022; Gao et al., 2022). They either reformulate problem (3) as a constrained optimization problem in the same spirit as (2), or build upon existing methods (in particular, BiG-SAM by Sabach and Shtern (2017)) for solving the simple bilevel problem. Moreover, as they consider a more general problem than ours, their theoretical results are necessarily weaker, providing only asymptotic convergence guarantees or slower rates. In this paper, we explore a fundamentally different approach for solving the bilevel problem in (1) directly and provide tight non-asymptotic convergence guarantees for our method.

2 PRELIMINARIES

In this section, we first discuss a few motivating examples for problem (1), which can be generalized to two broader classes of problems: lexicographic optimization (Gong et al., 2021) and lifelong learning (Chaudhry et al., 2019). Additional discussions and examples are provided in Appendix E. Then, we state the required assumptions and notions of optimality that we use for our theoretical results.

2.1 Motivating Examples

Several machine learning applications consist of a main objective g, such as the training loss, and a secondary objective f, such as a regularization term or an auxiliary loss. In this case, a natural approach is to fully optimize the main objective and use the secondary objective as a criterion to select one of the optimal solutions. This approach is also known as lexicographic optimization (Gong et al., 2021) and can be formulated as the simple bilevel problem in (1). In Examples 1 and 2, we provide instances of such problems.

In the paradigm of Lifelong Learning, the learner faces a stream of possibly related tasks and the central theme is to accumulate the knowledge learned from the past and continually improves it given new tasks. We can cast this problem as a simple bilevel problem, where the lower-level loss corresponds to samples from seen tasks, while the upper-level loss captures the error on a new task. The goal is to improve the model using the new task while ensuring that it still performs well over the previous tasks. We illustrate an instance of this class of problems in Example 3.

Example 1 (Over-parameterized regression). In a constrained regression problem, we aim to find a parameter vector $\boldsymbol{\beta} \in \mathbb{R}^d$ that minimizes the loss $\ell_{\rm tr}(\boldsymbol{\beta})$ with respect to the training dataset $\mathcal{D}_{\rm tr}$. We also constrain $\boldsymbol{\beta}$ to be in some set $\mathcal{Z} \subseteq \mathbb{R}^d$ representing some prior knowledge. For instance, we have $\mathcal{Z} = \{\boldsymbol{\beta} \mid \|\boldsymbol{\beta}\|_1 \leq \lambda\}$ for some $\lambda > 0$ in a sparse regression problem. Without an explicit regularization, an over-parameterized regression problem over the training dataset possesses multiple global minima. In fact, while any optimization algorithm can achieve one of these many global minima, not all optimal regression coefficients perform equally. Hence, one can consider a secondary objective, such as the loss over a validation set $\mathcal{D}_{\rm val}$, to select one from the minimizers of the training loss. This leads to the following bilevel problem:

$$\begin{aligned} &\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \ f(\boldsymbol{\beta}) \triangleq \ell_{\rm val}(\boldsymbol{\beta}) \\ &\text{s.t.} \quad \boldsymbol{\beta} \in \operatorname*{argmin}_{\mathbf{z}\in\mathcal{Z}} g(\mathbf{z}) \triangleq \ell_{\rm tr}(\mathbf{z}). \end{aligned} \tag{4}$$

We note that problem (4) can also appear as a subproblem in hyperparameter selection problems (Gao et al., 2022). In this case, both the upper-level and lower-level objectives are smooth and convex if the loss ℓ is smooth and convex.

Example 2 (Fair classification). In a binary classification problem, we aim to find a mapping from the feature vectors $\mathbf{x}_i$ to the target labels y_i. Due to the bias in the dataset, standard training procedures could lead to a model that discriminates against certain social groups. To alleviate this issue, we can use a fairness metric as a secondary objective to promote fairness in the decision of the model. One common criterion is the $p\%$-rule: given a sensitive attribute v such as race or sex, we require that for any a and b,

$$\min\left(\frac{\mathbb{P}(\hat{y}=1\,|\,\mathbf{x}, v=a)}{\mathbb{P}(\hat{y}=1\,|\,\mathbf{x}, v=b)}, \frac{\mathbb{P}(\hat{y}=1\,|\,\mathbf{x}, v=b)}{\mathbb{P}(\hat{y}=1\,|\,\mathbf{x}, v=a)}\right) \geq \frac{p}{100},$$

where $\hat{y}$ is the prediction of the model. However, this objective is hard to optimize and hence we use the covariance as a surrogate loss as suggested in (Zafar et al., 2017; Gong et al., 2021). Let $h(\mathbf{x};\boldsymbol{\beta})$ be the output of the model parameterized by $\boldsymbol{\beta}$, and consider the following problem:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \ (\mathrm{cov}(h(\mathbf{x};\boldsymbol{\beta}),v))^2 \quad \text{s.t.} \quad \boldsymbol{\beta}\in \operatorname*{argmin}_{\mathbf{z}\in\mathcal{Z}} \ \ell_{\mathrm{tr}}(\mathbf{z}).$$

Specifically, we aim to minimize the correlation between our prediction model and the sensitive feature v without sacrificing its performance over the training set.

Example 3 (Dictionary learning). The goal of dictionary learning is to learn a concise representation of the input data from a massive dataset. Let $\mathbf{A}=\{\mathbf{a}_1,\ldots,\mathbf{a}_n\}$ denote a dataset of n points with $\mathbf{a}_i\in\mathbb{R}^m$ for any $i\in\mathcal{N}\triangleq\{1,\ldots,n\}$. We aim to find a dictionary $\mathbf{D}=[\mathbf{d}_1,\ldots,\mathbf{d}_p]\in\mathbb{R}^{m\times p}$ such that each data point $\mathbf{a}_i$ can be well approximated by a linear combination of a few basis vectors in $\mathbf{D}$. A common approach is to formulate this as the following non-convex optimization problem (Kreutz-Delgado et al., 2003; Yaghoobi et al., 2009; Rakotomamonjy, 2013; Bao et al., 2016):

$$\begin{aligned}&\min_{\mathbf{D}\in\mathbb{R}^{m\times p}} \min_{\mathbf{X}\in\mathbb{R}^{p\times n}} \ \frac{1}{2n}\sum_{i\in\mathcal{N}}\|\mathbf{a}_i-\mathbf{D}\mathbf{x}_i\|_2^2\\ &\text{s.t.} \quad \|\mathbf{d}_j\|_2\le 1, j=1,\ldots,p;\ \|\mathbf{x}_i\|_1\le\delta, i\in\mathcal{N}.\end{aligned} \tag{5}$$

Note that we normalize the basis vectors to have bounded ℓ_2-norm and impose ℓ_1-norm constraints to encourage sparsity in $\{\mathbf{x}_i\}_{i=1}^n$. Further, we refer to $\mathbf{X}=[\mathbf{x}_1,\ldots,\mathbf{x}_n]\in\mathbb{R}^{p\times n}$ as the coefficient matrix.

In real applications, the data points typically arrive sequentially and the underlying representation may be gradually evolving. Thus, it is desirable to update our dictionary in a continuous manner. Suppose that we already have learned a dictionary $\hat{\mathbf{D}}\in\mathbb{R}^{m\times p}$ and the corresponding coefficient matrix $\hat{\mathbf{X}}\in\mathbb{R}^{p\times n}$ for the dataset $\mathbf{A}$. When a new dataset $\mathbf{A}'=\{\mathbf{a}'_1,\ldots,\mathbf{a}'_{n'}\}$ arrives, we hope to expand our dictionary by learning new basis vectors from $\mathbf{A}'$ while retaining the learned information in $\hat{\mathbf{D}}$. To achieve so, we aim to find the dictionary $\tilde{\mathbf{D}}\in\mathbb{R}^{m\times q}$ $(q>p)$ and the coefficient matrix $\tilde{\mathbf{X}}\in\mathbb{R}^{q\times n'}$ for the new dataset $\mathbf{A}'$, and at the same time enforce $\tilde{\mathbf{D}}$ to perform well on the old dataset $\mathbf{A}$ together with the learned coefficient matrix $\hat{\mathbf{X}}$. This leads to the following bilevel problem:

$$\begin{aligned}&\min_{\tilde{\mathbf{D}}\in\mathbb{R}^{m\times q}} \min_{\tilde{\mathbf{X}}\in\mathbb{R}^{q\times n'}} f(\tilde{\mathbf{D}},\tilde{\mathbf{X}})\\ &\text{s.t.} \quad \|\tilde{\mathbf{x}}_k\|_1\le\delta, k=1,\ldots,n';\ \tilde{\mathbf{D}}\in\operatorname*{argmin}_{\|\tilde{\mathbf{d}}_j\|_2\le 1} g(\tilde{\mathbf{D}}),\end{aligned} \tag{6}$$

where the objective $f(\tilde{\mathbf{D}},\tilde{\mathbf{X}})\triangleq\frac{1}{2n'}\sum_{k=1}^{n'}\|\mathbf{a}'_k-\tilde{\mathbf{D}}\tilde{\mathbf{x}}_k\|_2^2$ is the average reconstruction error on the new dataset $\mathbf{A}'$, the lower-level objective $g(\tilde{\mathbf{D}})\triangleq\frac{1}{2n}\sum_{i=1}^{n}\|\mathbf{a}_i-\tilde{\mathbf{D}}\hat{\mathbf{x}}_i\|_2^2$ is the error on the old dataset $\mathbf{A}$, and with a slight abuse of notation we let $\hat{\mathbf{x}}_i$ denote the extended vector in $\mathbb{R}^q$ by appending zeros at the end. Note that in problem (6), the upper-level objective is non-convex while the lower-level objective is convex with multiple minima.

2.2 Assumptions and Definitions

We focus on the case where the lower-level function g is smooth and convex, while the upper-level function f is smooth but not necessarily convex. Formally, we make the following assumptions.

Assumption 1. *Let $\|\cdot\|$ be an arbitrary norm on $\mathbb{R}^d$ and $\|\cdot\|_*$ be its dual norm. We assume*

(i) $\mathcal{Z}\subset\mathbb{R}^d$ is convex and compact with diameter D, i.e., $\|\mathbf{x}-\mathbf{y}\|\le D$ 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 Lipschitz with constant L_g, i.e., $\|\nabla g(\mathbf{x})-\nabla g(\mathbf{y})\|_*\le L_g\|\mathbf{x}-\mathbf{y}\|$ for all $\mathbf{x},\mathbf{y}\in\mathcal{Z}$.

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

Remark 2.1. Instead of the Lipschitz gradient assumptions above, we may assume that f and g have bounded *curvature constants*. Such an assumption is common in the analysis of the CG method and has the advantage of being affine-invariant, e.g., see (Jaggi, 2013; Lacoste-Julien, 2016).

Throughout the paper, we use $g^*\triangleq\min_{\mathbf{z}\in\mathcal{Z}} g(\mathbf{z})$ and $\mathcal{X}_g^*\triangleq\operatorname{argmin}_{\mathbf{z}\in\mathcal{Z}} g(\mathbf{z})$ to denote the optimal value and the optimal solution set of the lower-level problem, respectively. Note that by Assumption 1, the set $\mathcal{X}_g^*$ is nonempty, compact and convex, but in general not a singleton as g could have multiple minima on $\mathcal{Z}$. Moreover, we use f^* to denote the optimal value and $\mathbf{x}^*$ to denote an optimal solution of problem (1), which are guaranteed to exist as f is continuous and $\mathcal{X}_g^*$ is compact.

For generality, we allow different target accuracies ϵ_f and ϵ_g for the upper-level and lower-level problems, respectively, and define an (ϵ_f,ϵ_g)-optimal solution as follows.

Definition 1 ((ϵ_f,ϵ_g)-optimal solution)**.** *When f is convex, a point $\hat{\mathbf{x}}\in\mathcal{Z}$ is (ϵ_f,ϵ_g)-optimal for problem* (1) *if*

$$f(\hat{\mathbf{x}})-f^*\le\epsilon_f \quad \textit{and} \quad g(\hat{\mathbf{x}})-g^*\le\epsilon_g.$$

When f is non-convex, $\hat{\mathbf{x}}\in\mathcal{Z}$ is (ϵ_f,ϵ_g)-optimal if

$$\mathcal{G}(\hat{\mathbf{x}})\le\epsilon_f \quad \textit{and} \quad g(\hat{\mathbf{x}})-g^*\le\epsilon_g,$$

where $\mathcal{G}(\hat{\mathbf{x}})$ is the FW gap (Jaggi, 2013; Lacoste-Julien, 2016) defined by

$$\mathcal{G}(\hat{\mathbf{x}})\triangleq\max_{\mathbf{s}\in\mathcal{X}_g^*}\{\langle\nabla f(\hat{\mathbf{x}}),\hat{\mathbf{x}}-\mathbf{s}\rangle\}. \tag{7}$$

3 PROPOSED ALGORITHM

Before stating our proposed method, we start by the standard CG method (Frank and Wolfe, 1956; Levitin and Polyak,

Algorithm 1 CG-based bilevel optimization (CG-BiO)

Input: Target accuracy $\epsilon_f, \epsilon_g > 0$, stepsizes $\{\gamma_k\}_k$
Initialization: Set $\mathbf{x}_0 \in \mathcal{Z}$ s.t. $g(\mathbf{x}_0) - g^* \leq \epsilon_g/2$
for $k = 0, \ldots, K-1$ **do**
  Compute $\mathbf{s}_k \leftarrow \operatorname{argmin}_{\mathbf{s}\in\mathcal{X}_k} \langle \nabla f(\mathbf{x}_k), \mathbf{s}\rangle$
  $\mathcal{X}_k \triangleq \{\mathbf{s} \in \mathcal{Z} : \langle \nabla g(\mathbf{x}_k), \mathbf{s} - \mathbf{x}_k\rangle \leq g(\mathbf{x}_0) - g(\mathbf{x}_k)\}$
  if $\langle \nabla f(\mathbf{x}_k), \mathbf{x}_k - \mathbf{s}_k\rangle \leq \epsilon_f$ and $\langle \nabla g(\mathbf{x}_k), \mathbf{x}_k - \mathbf{s}_k\rangle \leq \epsilon_g/2$ **then**
    Return $\mathbf{x}_k$ and STOP
  else
    $\mathbf{x}_{k+1} \leftarrow (1-\gamma_k)\mathbf{x}_k + \gamma_k \mathbf{s}_k$
  end if
end for

1966) for solving problem (1). Recall that $\mathcal{X}_g^*$ denotes the solution set of the lower-level problem. If we assume $\mathbf{x}_0 \in \mathcal{X}_g^*$, then the CG update at iteration k is given by

$$\mathbf{x}_{k+1} = (1-\gamma_k)\mathbf{x}_k + \gamma_k \mathbf{s}_k,$$

where

$$\mathbf{s}_k = \operatorname*{argmin}_{\mathbf{s}\in\mathcal{X}_g^*} \langle \nabla f(\mathbf{x}_k), \mathbf{s}\rangle, \tag{8}$$

and $\gamma_k \in [0,1]$ is the stepsize. However, the main challenge here is that the solution set $\mathcal{X}_g^*$ for the lower-level problem is not explicitly given, and hence the linear minimization required in (8) is computationally intractable. Moreover, the standard CG method needs to be initialized with a feasible point. In this case, $\mathbf{x}_0$ has to be an optimal solution of the lower-level problem, which is hard to guarantee in general—in finite number of iterations one may not be able to find an *exact* optimal solution for the lower-level problem. Similar issues also hold if we try to use projection-based methods such as projected gradient descent to solve problem (2).

Our key idea is to perform a CG update over a local approximation set $\mathcal{X}_k$ at the k-th iteration instead of the hard-to-characterize set $\mathcal{X}_g^*$. To this end, we borrow the idea of *cutting plane* from the optimization literature (Boyd and Vandenberghe, 2018) and let $\mathcal{X}_k$ be the intersection of $\mathcal{Z}$ and the halfspace $\mathcal{H}_k$:

$$\begin{aligned} &\mathcal{X}_k \triangleq \mathcal{Z} \cap \mathcal{H}_k, \\ &\mathcal{H}_k \triangleq \{\mathbf{s} \in \mathbb{R}^d \mid \langle \nabla g(\mathbf{x}_k), \mathbf{s} - \mathbf{x}_k\rangle \leq g(\mathbf{x}_0) - g(\mathbf{x}_k)\}. \end{aligned} \tag{9}$$

Indeed $\mathcal{X}_k$ is potentially more tractable than $\mathcal{X}_g^*$, as the difficult nonlinear inequality $g(\mathbf{x}) \leq g^*$ in (2) is replaced by a single linear inequality. Also, by using the convexity of g, we can show that the hyperplane $\mathcal{H}_k$ eliminates those points known to have a larger value than $g(\mathbf{x}_0)$. Thus, if we initialize our algorithm such that $\mathbf{x}_0$ is near-optimal for the lower-level problem, the linear inequality in (9) ensures improvement in terms of the lower-level function. Further, this also implies that $\mathcal{X}_k$ contains the solution set $\mathcal{X}_g^*$, so we are guaranteed to make progress on the upper-level objective f. We formalize this claim in the following lemma.

Lemma 1. *Recall $\mathcal{X}_g^*$ as the solution set for the lower-level problem in* (1) *and recall the definition of the set $\mathcal{X}_k$ in* (9)*. Then, for any $k \geq 0$, we have $\mathcal{X}_g^* \subseteq \mathcal{X}_k$.*

Now we are ready to introduce our proposed CG-BiO method. We first initialize $\mathbf{x}_0 \in \mathcal{Z}$ as a near-optimal solution for the lower-level problem, i.e., $g(\mathbf{x}_0) - g^* \leq \epsilon_g/2$ for some prescribed accuracy ϵ_g. This can be done by running the standard CG method on the lower-level problem, which requires at most $\mathcal{O}(1/\epsilon_g)$ iterations. Once the initialization step is done, we simply run CG with respect to the approximation sets $\mathcal{X}_k$. Specificallly, at iteration k, we solve the following subproblem over the set $\mathcal{X}_k$ defined in (9):

$$\mathbf{s}_k = \operatorname*{argmin}_{\mathbf{s}\in\mathcal{X}_k} \langle \nabla f(\mathbf{x}_k), \mathbf{s}\rangle, \tag{10}$$

and update the iterate by $\mathbf{x}_{k+1} = (1-\gamma_k)\mathbf{x}_k + \gamma_k \mathbf{s}_k$ with stepsize $\gamma_k \in [0,1]$. We assume access to a linear optimization oracle that returns the solution of the subproblem in (10), which is standard for projection-free methods (Jaggi, 2013; Lacoste-Julien, 2016; Mokhtari et al., 2018). In particular, if $\mathcal{Z}$ can be described by a system of linear inequalities, then problem (10) corresponds to a linear program and can be solved efficiently by a standard solver as we will show in our experiments. We repeat the process above until we reach an accuracy of ϵ_f for the upper-level objective and an accuracy of ϵ_g for the lower-level objective. The steps of our proposed CG-BiO method are summarized in Algorithm 1.

4 CONVERGENCE ANALYSIS

In this section, we analyze the iteration complexity of our CG-BiO method. We first consider the case where the upper-level function f is convex. In this case, we choose the stepsize as $\gamma_k = 2/(k+2)$, which is a typical choice in the standard CG method (Jaggi, 2013).

Theorem 1 (Convex upper-level)**.** *Suppose that Assumption 1 holds and f is convex. Let $\{\mathbf{x}_k\}_{k=0}^{K-1}$ be the sequence generated by Algorithm 1 with stepsize $\gamma_k = 2/(k+2)$ for $k \geq 0$. Then we have*

$$f(\mathbf{x}_K) - f^* \leq \frac{2L_f D^2}{K+1}, \;\; g(\mathbf{x}_K) - g^* \leq \frac{2L_g D^2}{K+1} + \frac{\epsilon_g}{2}.$$

Theorem 1 shows that the gap of the upper-level objective can be upper bounded by $\mathcal{O}(1/K)$, similar to the convergence bound of standard CG. At the same time, the gap of the lower-level objective can also be controlled by a term of order $\mathcal{O}(1/K)$ in addition to the initial error $\epsilon_g/2$. As a corollary, Algorithm 1 will return an (ϵ_f, ϵ_g)-optimal solution when the number of iterations K exceeds

$$\max\left\{\frac{2L_f D^2}{\epsilon_f}, \frac{4L_g D^2}{\epsilon_g}\right\} = \mathcal{O}\left(\max\left\{\frac{1}{\epsilon_f}, \frac{1}{\epsilon_g}\right\}\right).$$

Our complexity bound improves over the result by Kaushik and Yousefian (2021), who considered a different setup

where both the upper-level and lower-level functions are Lipschitz but not necessarily smooth. Also, comparing with existing works in the same setup, our convergence rate for the lower-level objective matches those by Sabach and Shtern (2017); Malitsky (2017), while we also provide a non-asymptotic convergence bound for the upper-level objective. To the best of our knowledge, our result provides the best-known bound for the considered setting. We also remark that our rate is tight at least within the family of projection-free methods, since it is known that their worst-case complexity is $\Theta(1/\epsilon_f)$ even for a single-level problem (Jaggi, 2013; Lan, 2013).

Remark 4.1. The initialization step requires $\mathcal{O}(1/\epsilon_g)$ iterations, and hence, this additional term does not change the overall complexity. The same applies for the non-convex case below.

Now we turn to the case where f is non-convex. In this case, we choose the stepsize as a constant depending on the target accuracies as well as the problem parameters.

Theorem 2 (Non-convex upper-level)**.** *Suppose that Assumption 1 holds. Let $\{\mathbf{x}_k\}_{k=0}^{K-1}$ be the sequence generated by Algorithm 1 with stepsize $\gamma_k = \min\left\{\frac{\epsilon_f}{L_f D^2}, \frac{\epsilon_g}{L_g D^2}\right\}$ for all $k \geq 0$. Define $\underline{f} = \min_{\mathbf{x}\in Z} f(\mathbf{x})$. Then for*

$$K \geq \max\Big\{\frac{2L_f D^2(f(\mathbf{x}_0)-\underline{f})}{\epsilon_f^2}, \frac{2L_g D^2(f(\mathbf{x}_0)-\underline{f})}{\epsilon_f \epsilon_g}\Big\},$$

there exists $k^* \in \{0, 1, \ldots, K-1\}$ such that $\mathcal{G}(\mathbf{x}_{k^*}) \leq \epsilon_f$ and $g(\mathbf{x}_{k^*}) - g^* \leq \epsilon_g$.

As a corollary of Theorem 2, the number of iterations required to find an (ϵ_f, ϵ_g)-optimal solution can be upper bounded by $\mathcal{O}(\max\{1/\epsilon_f^2, 1/(\epsilon_f\epsilon_g)\})$. We note that the dependence on the upper-level accuracy ϵ_f also matches that in the standard CG method for a single-level problem (Lacoste-Julien, 2016; Mokhtari et al., 2018).

We end this section with the following remark. Since the algorithm's output $\hat{\mathbf{x}}$ may lie outside of the feasible set $\mathcal{X}_g^*$, both $f(\hat{\mathbf{x}}) - f^*$ in Theorem 1 and $\mathcal{G}(\hat{\mathbf{x}})$ in Theorem 2 are not necessarily positive. While this might seem unconventional, we note that Kaushik and Yousefian (2021) also used $f(\hat{\mathbf{x}}) - f^*$ as the performance metric. In fact, this is also common in the literature on constrained optimization, where the generated iterate could be infeasible and thus $f(\hat{\mathbf{x}}) - f^*$ could be negative (see, e.g., Beck (2017)). On the other hand, we note that it is in general impossible to prove convergence in terms of $|f(\hat{\mathbf{x}}) - f^*|$ due to a negative result by Chen et al. (2023). Specifically, for any first-order method and a given number of iterations K, they showed that there exists an instance of problem (1) where $|f(\mathbf{x}_k) - f^*| \geq 1$ for all $0 \leq k \leq K-1$. Therefore, to provide convergence bounds on $|f(\hat{\mathbf{x}}) - f^*|$ or $|\mathcal{G}(\hat{\mathbf{x}})|$, it is necessary to impose additional regularity conditions on problem (1), which we discuss in the next section.

4.1 Convergence under Hölderian Error Bound

In this section, we complement the convergence results in Theorems 1 and 2 by giving a lower bound on $f(\hat{\mathbf{x}}) - f^*$ and $\mathcal{G}(\hat{\mathbf{x}})$. Let $\hat{\mathbf{x}}$ be an (ϵ_f, ϵ_g)-optimal solution as defined in Definition 1. Intuitively, since $\hat{\mathbf{x}}$ is ϵ_g-optimal for the lower-level problem, it should be close to the optimal solution set $\mathcal{X}_g^*$ under some regularity condition on g. As such, we can lower bound $f(\hat{\mathbf{x}}) - f^*$ by using the smoothness of f. Formally, we assume that the lower-level objective satisfies the Hölderian error bound, which quantifies the growth rate of the objective value $g(\mathbf{x})$ as the point $\mathbf{x}$ deviates from the optimal solution set $\mathcal{X}_g^*$.

Assumption 2. *The function g satisfies the Hölderian error bound for some $\alpha > 0$ and $r \geq 1$, i.e,*

$$\frac{\alpha}{r}\,\mathrm{dist}(\mathbf{x}, \mathcal{X}_g^*)^r \leq g(\mathbf{x}) - g^*, \qquad \forall \mathbf{x} \in \mathcal{Z}, \tag{11}$$

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

We note that the error bound condition in (11) is well-studied in the optimization literature (see (Pang, 1997; Bolte et al., 2017; Roulet and d'Aspremont, 2020) and the references therein) and is known to hold generally when the function g is analytic and the set $\mathcal{Z}$ is bounded (Łojasiewicz, 1959; Luo and Pang, 1994). Two important special cases are: 1) g satisfies (11) with $r = 1$, i.e., $\mathcal{X}_g^*$ is a set of weak sharp minima of g (Burke and Ferris, 1993; Burke and Deng, 2005); 2) g satisfies (11) with $r = 2$ known as quadratic growth condition (Drusvyatskiy and Lewis, 2018).

Under Assumption 2, we can establish the following lower bounds on $f(\hat{\mathbf{x}}) - f^*$ and $\mathcal{G}(\hat{\mathbf{x}})$. Notably, the following result is an intrinsic property of problem (1) and independent of the algorithm we use.

Proposition 1. *Assume that g satisfies Assumption 2, and define $M = \max_{\mathbf{x}\in\mathcal{X}_g^*} \|\nabla f(\mathbf{x})\|_*$. Then for any $\hat{\mathbf{x}}$ that satisfies $g(\hat{\mathbf{x}}) - g^* \leq \epsilon_g$, it holds that:*

(i) If f is convex, then $f(\hat{\mathbf{x}}) - f^* \geq -M\left(\frac{r\epsilon_g}{\alpha}\right)^{\frac{1}{r}}$.

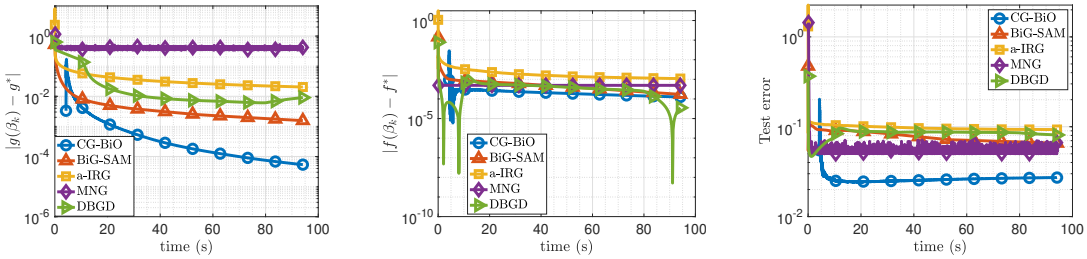
(ii) If f is non-convex and has L_f-Lipschitz gradient, then $\mathcal{G}(\hat{\mathbf{x}}) \geq -M\left(\frac{r\epsilon_g}{\alpha}\right)^{\frac{1}{r}} - L_f\left(\frac{r\epsilon_g}{\alpha}\right)^{\frac{2}{r}}$.

By combining Theorems 1 and 2 with Proposition 1, we obtain the following stronger convergence guarantees for the output of our proposed method.

Corollary 1. *Suppose that Assumption 1 holds and g satisfies the Hölderian error bound in Assumption 2 with $\alpha > 0$ and $r \geq 1$. Let $M = \max_{\mathbf{x}\in\mathcal{X}_g^*} \|\nabla f(\mathbf{x})\|_*$.*

(i) If f in problem (1) is convex and we set $\epsilon_g = \frac{\alpha}{r}\left(\frac{\epsilon_f}{M}\right)^r$, then after $K = \mathcal{O}(1/\epsilon_f^r)$ iterations we have $|f(\mathbf{x}_K) - f^*| \leq \epsilon_f$ and $g(\mathbf{x}_K) - g^* \leq \epsilon_g$.

(ii) If f in problem (1) is non-convex and we set $\epsilon_g = \min\{\frac{\alpha}{r}\left(\frac{\epsilon_f}{2M}\right)^r, \frac{\alpha}{r}\left(\frac{\epsilon_f}{2L_f}\right)^{r/2}\}$, then after $K = \mathcal{O}(1/\epsilon_f^{r+1})$ iterations there exists $k^* \in \{0, \ldots, K-1\}$ such that $|\mathcal{G}(\mathbf{x}_{k^*})| \leq \epsilon_f$ and $g(\mathbf{x}_{k^*}) - g^* \leq \epsilon_g$.



(a) Lower-level gap (b) Upper-level gap (c) Test error objective

Figure 1: The performance of CG-BiO compared with BiG-SAM, a-IRG and MNG on problem (4).

Corollary 1 shows that under the r-th Hölderian error bound assumption, we can find an iterate to be ϵ_f-close to optimality within $\mathcal{O}(1/\epsilon_f^r)$ iterations in the convex case, and to be ϵ_f-close to stationarity within $\mathcal{O}(1/\epsilon_f^{r+1})$ iterations in the non-convex case.

5 NUMERICAL EXPERIMENTS

In this section, we test our method on three different bilevel optimization problems, described in Section 2, with real and synthetic datasets and compare our method with other existing methods in the literature (Beck and Sabach, 2014; Sabach and Shtern, 2017; Kaushik and Yousefian, 2021; Gong et al., 2021).

5.1 Over-parameterized Regression

For Example 1, we consider a sparse linear regression problem on the Wikipedia Math Essential dataset (Rozemberczki et al., 2021), which consists of a data matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ with $n = 1068$ instances and $d = 730$ attributes and an outcome vector $\mathbf{b} \in \mathbb{R}^n$. We assign 60% of the dataset as the training set $(\mathbf{A}_{\rm tr}, \mathbf{b}_{\rm tr})$, 20% as the validation set $(\mathbf{A}_{\rm val}, \mathbf{b}_{\rm val})$ and the rest as the test set $(\mathbf{A}_{\rm test}, \mathbf{b}_{\rm test})$. Then the lower-level objective in (4) is the training error $g(\mathbf{z}) = \frac{1}{2}\|\mathbf{A}_{\rm tr}\mathbf{z} - \mathbf{b}_{\rm tr}\|_2^2$, the upper-level objective is the validation error $f(\boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{A}_{\rm val}\boldsymbol{\beta} - \mathbf{b}_{\rm val}\|_2^2$, and the constraint set is $\mathcal{Z} = \{\boldsymbol{\beta} \mid \|\boldsymbol{\beta}\|_1 \le \lambda\}$ for some $\lambda > 0$ to induce sparsity in $\boldsymbol{\beta}$. We also use the test error $\frac{1}{2}\|\mathbf{A}_{\rm test}\boldsymbol{\beta} - \mathbf{b}_{\rm test}\|_2^2$ as our performance metric. Note that the regression problem is over-parameterized since the number of features d is larger than the number of data instances in the training set.

We compare the performance of our method CG-BiO against the MNG method by Beck and Sabach (2014), the Bilevel Gradient SAM (BiG-SAM) by Sabach and Shtern (2017), the averaging iteratively regularized gradient (a-IRG) by Kaushik and Yousefian (2021), and the Dynamic Barrier Gradient Descent (DBGD) by Gong et al. (2021). It is worth noting that while we can implement these methods numerically, some of them are not directly applicable to our problem setting and thus lack any convergence guarantee; see Appendix F for more discussions. For benchmarking purposes, we use CVX (Grant and Boyd, 2014, 2008) to solve the lower-level problem and the constrained reformulation in (2) to obtain the optimal values g^* and f^*, respectively.

In Fig. 1(a), we observe that CG-BiO converges at a faster rate than the other baseline methods in terms of the lower-level objective, which confirms our theoretical result (cf. Table 1). Fig. 1(b) and (c) show that CG-BiO achieves a smaller upper-level objective gap as well as a smaller test error compared to other methods within the same run time. Interestingly, after the initial stage, the upper-level objective $f(\boldsymbol{\beta}_k)$ of CG-BiO *increases*, while the optimality gap $|f(\boldsymbol{\beta}_k) - f^*|$ *decreases*. This suggests that CG-BiO may "overshoot" at the beginning due to its relatively large stepsize. Nevertheless, as the number of iterations increases and the level of infeasibility decreases, the upper-level objective of CG-BiO approaches the optimal value of the bilevel problem, which is in line with Proposition 1.

5.2 Fair Classification

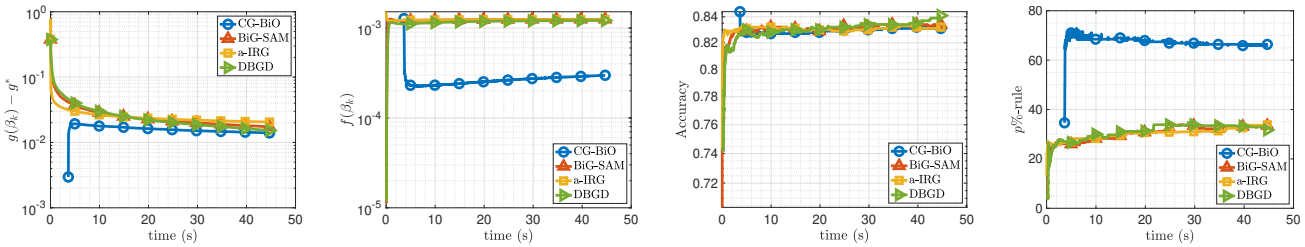
We use the Adult income dataset (Dua and Graff, 2019) containing 48,842 subjects each with 14 attributes, where the task is to predict whether the annual income of a given subject exceeds \$50K. For efficiency, we randomly sample 2,000 data points as the training set and 1,000 as the test set. We choose "sex" as the sensitive attribute v. Further, we adopt the logistic regression classifier as our model, where the posterior probability is given by $\mathbb{P}(\hat{y}_i = 1 \,|\, \mathbf{x}_i; \boldsymbol{\beta}) = 1/(1 + e^{-\mathbf{x}_i^\top \boldsymbol{\beta}})$ for a given feature vector $\mathbf{x}_i$ and parameter $\boldsymbol{\beta}$. Concretely, the lower-level problem is a sparse logistic regression problem for some $\lambda > 0$:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \; g(\boldsymbol{\beta}) = -\frac{1}{n}\sum_{i=1}^{n} \log \mathbb{P}(\hat{y}_i = y_i \,|\, \mathbf{x}_i; \boldsymbol{\beta}) \;\; \text{s.t. } \|\boldsymbol{\beta}\|_1 \le \lambda, \tag{12}$$

while the upper-level objective is the squared covariance:

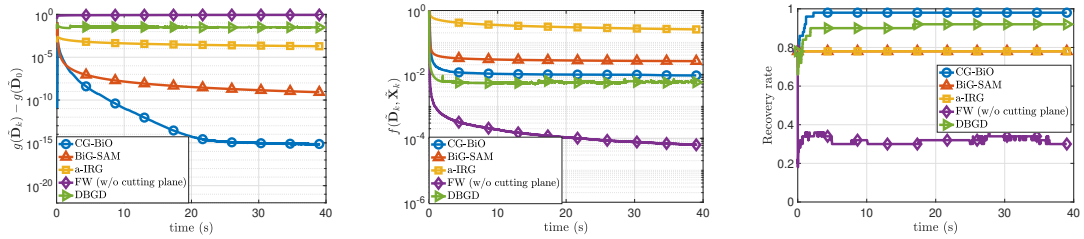
$$f(\boldsymbol{\beta}) = \Big(\frac{1}{n}\sum_{i=1}^{n}(v_i - \bar{v})\mathbb{P}(\hat{y}_i = 1 \,|\, \mathbf{x}_i; \boldsymbol{\beta})\Big)^2, \tag{13}$$

where $\bar{v} = \frac{1}{n}\sum_i v_i$. Note that the lower-level objective in (12) is convex while the upper-level in (13) is non-convex. We numerically verified that the lower-level problem can possess multiple optimal solutions.



(a) Lower-level gap (b) Upper-level objective (c) Test set accuracy (d) $p\%$-rule

Figure 2: The performance of CG-BiO compared with BiG-SAM, a-IRG on the bilevel problem defined in (12) and (13).



(a) Lower-level objective (b) Upper-level objective (c) Recovery rate

Figure 3: The performance of CG-BiO compared with BiG-SAM, a-IRG and the baseline CG method on problem (6).

In Fig. 2, we compare CG-BiO against BiG-SAM, a-IRG and DBGD. For this example, we did not implement the MNG method as it is computationally intractable; see Appendix F. For a fair comparison, we manually tune the hyperparameters such that all achieve a similar performance on the lower-level problem as shown in Fig. 2(a), and judge their efficiency by the error of the upper-level objective. Fig. 2(b) shows that CG-BiO is able to achieve a smaller upper-level objective (smaller covariance). As a result, we observe in Fig. 2(c) and (d) that it reaches a much better $p\%$-rule with a similar level of accuracy on the test set.

5.3 Dictionary Learning

We evaluate our CG-BiO method for problem (6) on a synthetic dataset, similar to the setup in Rakotomamonjy (2013). We first generate the true dictionary $\tilde{\mathbf{D}}^* \in \mathbb{R}^{25\times 50}$ consisting of 50 basis vectors in $\mathbb{R}^{25}$, each of which has its entries drawn from a standard Gaussian distribution and is normalized to have unit ℓ_2-norm. We further construct the two dictionaries $\mathbf{D}^*$ and $\mathbf{D}'^*$ consisting of 40 and 20 basis vectors in $\tilde{\mathbf{D}}^*$, respectively (and hence they share 10 bases in common). The two datasets $\mathbf{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_{250}\}$ and $\mathbf{A}' = \{\mathbf{a}'_1, \ldots, \mathbf{a}'_{200}\}$ are generated according to the rules:

$$\mathbf{a}_i = \mathbf{D}^*\mathbf{x}_i + \mathbf{n}_i, \qquad i = 1, \ldots, 250;$$
$$\mathbf{a}'_k = \mathbf{D}'^*\mathbf{x}'_k + \mathbf{n}'_k, \quad k = 1, \ldots, 200,$$

where $\{\mathbf{x}_i\}_{i=1}^{250}, \{\mathbf{x}'_k\}_{k=1}^{200}$ are sparse coefficient vectors and $\{\mathbf{n}_i\}_{i=1}^{250}, \{\mathbf{n}'_k\}_{k=1}^{200}$ are random Gaussian noise vectors. Since neither $\mathbf{A}$ nor $\mathbf{A}'$ contains the full information of the true dictionary $\tilde{\mathbf{D}}^*$, it is crucial for our learning algorithm to update our dictionary given the new dataset $\mathbf{A}'$ while retaining our knowledge from the old dataset $\mathbf{A}$.

In our experiment, we first solve the standard dictionary learning problem in (5) using the dataset $\mathbf{A}$ to obtain the initial dictionary $\hat{\mathbf{D}}$ and the coefficient vectors $\{\hat{\mathbf{x}}_i\}_{i=1}^{250}$. Then we use the reconstruction error on $\mathbf{A}$ with respect to $\{\hat{\mathbf{x}}_i\}_{i=1}^{250}$ to define the lower-level objective in problem (6), and use the error on the new dataset $\mathbf{A}'$ to define the upper-level objective. In this case, $\hat{\mathbf{D}}$ serves as a near-optimal solution for the lower-level problem. We compare CG-BiO with BiG-SAM and a-IRG. Similar to the previous experiment, we exclude MNG due to its computational intractability. Moreover, to demonstrate the necessity of the cutting plane in (9), we also run a baseline method that performs the CG update over the set $\mathcal{Z}$ instead of $\mathcal{X}_k$ (cf. the update in (10)). This method ignores the lower-level objective and may be regarded as applying the standard CG method solely on the upper-level objective. In all algorithms, we initialize $\tilde{\mathbf{D}}$ with the dictionary $\hat{\mathbf{D}}$ learned from $\mathbf{A}$ and initialize $\tilde{\mathbf{X}}$ randomly.

We report our results in Fig. 3. In addition to the the upper- and lower-level objective values, we use the recovery rate of the true basis vectors as our performance metric. Specifically, a basis vector $\tilde{\mathbf{d}}^*_i$ in $\tilde{\mathbf{D}}^*$ is regarded as successfully recovered if there exists $\tilde{\mathbf{d}}_j$ in $\tilde{\mathbf{D}}$ such that $|\langle \tilde{\mathbf{d}}^*_i, \tilde{\mathbf{d}}_j \rangle| > 0.9$. In Fig. 3(a) and (b), we observe that CG-BiO converges faster than BiG-SAM and a-IRG, and it also achieves smaller errors in terms of both the upper- and lower-level objectives. DBGD achieves a similar upper-level objective value as CG-BiO, but performs poorly in terms of the lower-level

objective. On the other hand, the baseline CG method only focuses on the upper-level objective and as a result incurs a much larger error on the lower-level objective. In terms of recovery rate, Fig. 3(c) shows that CG-BiO recovers almost all basis vectors in $\tilde{\mathbf{D}}^*$ at the end of its execution and performs slightly better than DBGD. In contrast, both BiG-SAM and a-IRG only learn from the dataset $\mathbf{A}$ due to their slow convergence, while the baseline CG method "forgets" the basis vectors previously learned and only recovers those underlying the new dataset $\mathbf{A}'$.

6 CONCLUSION

In this paper, we proposed a CG-based method to solve the class of simple bilevel optimization problems. We closed an important gap in the existing literature by providing a tight non-asymptotic complexity bound for the upper-level objective. Specifically, we proved that our CG-BiO method finds an (ϵ_f, ϵ_g)-optimal solution after at most $\mathcal{O}(\max\{1/\epsilon_f, 1/\epsilon_g\})$ iterations when the upper-level objective f is convex, and after at most $\mathcal{O}(\max\{1/\epsilon_f^2, 1/(\epsilon_f\epsilon_g)\})$ iterations when f is non-convex. We further strengthened our results when the lower-level problem satisfies the Hölderian error bound assumption. The numerical results also showed the superior performance of our method compared to existing algorithms.

Acknowledgements

The research of R. Jiang and A. Mokhtari is supported in part by NSF Grants 2127697, 2019844, and 2112471, ARO Grant W911NF2110226, the Machine Learning Lab (MLL) at UT Austin, and the Wireless Networking and Communications Group (WNCG) Industrial Affiliates Program. The research of N. Abolfazli and E. Yazdandoost Hamedani is supported by NSF Grant 2127696.

References

C. Bao, H. Ji, Y. Quan, and Z. Shen. Dictionary learning for sparse coding: Algorithms and convergence analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1356–1369, 2016.

A. Beck. *First-order methods in optimization*. SIAM, 2017.

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

L. Bertinetto, J. F. Henriques, P. Torr, and A. Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2019.

J. Bolte, T. P. Nguyen, J. Peypouquet, and B. W. Suter. From error bounds to the complexity of first-order descent methods for convex functions. *Mathematical Programming*, 165(2):471–507, 2017.

J. F. Bonnans and A. Shapiro. *Perturbation analysis of optimization problems*. Springer Science & Business Media, 2013.

S. Boyd and L. Vandenberghe. Localization and cutting-plane methods. `https://web.stanford.edu/class/ee364b/lectures/localization_methods_notes.pdf`, 2018.

J. V. Burke and S. Deng. Weak sharp minima revisited, part ii: application to linear regularity and error bounds. *Mathematical Programming*, 104(2):235–261, 2005.

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

A. Cabot. Proximal point algorithm controlled by a slowly vanishing term: Applications to hierarchical minimization. *SIAM Journal on Optimization*, 15(2):555–572, 2005.

A. Chambolle and T. Pock. On the ergodic convergence rates of a first-order primal–dual algorithm. *Mathematical Programming*, 159(1):253–287, 2016.

A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny. Efficient lifelong learning with a-GEM. In *International Conference on Learning Representations*, 2019.

L. Chen, J. Xu, and J. Zhang. On bilevel optimization without lower-level strong convexity. *arXiv preprint arXiv:2301.00712*, 2023.

L. Collins, A. Mokhtari, S. Oh, and S. Shakkottai. MAML and ANIL provably learn representations. In *Proceedings of the 39th International Conference on Machine Learning*, pages 4238–4310. PMLR, 2022.

S. Dempe. Bilevel optimization: Theory, algorithms, applications and a bibliography. In *Bilevel Optimization: Advances and Next Challenges*, pages 581–672. Springer International Publishing, Cham, 2020.

S. Dempe, N. Dinh, and J. Dutta. Optimality conditions for a simple convex bilevel programming problem. In *Variational Analysis and Generalized Differentiation in Optimization and Control: In Honor of Boris S. Mordukhovich*, pages 149–161. Springer New York, New York, NY, 2010.

S. Dempe, N. Dinh, J. Dutta, and T. Pandit. Simple bilevel programming and extensions. *Mathematical Programming*, 188(1):227–253, 2021.

J. Domke. Generic methods for optimization-based modeling. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, pages 318–326, 2012.

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

S. S. Du, W. Hu, S. M. Kakade, J. D. Lee, and Q. Lei. Few-shot learning via learning the representation, provably. In *International Conference on Learning Representations*, 2021.

D. Dua and C. Graff. UCI machine learning repository, 2019. URL `http://archive.ics.uci.edu/ml`.

J. Dutta and T. Pandit. Algorithms for simple bilevel programming. In *Bilevel Optimization: Advances and Next Challenges*, pages 253–291. Springer International Publishing, Cham, 2020.

L. Franceschi, P. Frasconi, S. Salzo, R. Grazzi, and M. Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1568–1577, 2018.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2): 95–110, 1956.

M. P. Friedlander and P. Tseng. Exact regularization of convex programs. *SIAM Journal on Optimization*, 18(4): 1326–1350, 2008.

L. L. Gao, J. Ye, H. Yin, S. Zeng, and J. Zhang. Value function based difference-of-convex algorithm for bilevel hyperparameter selection problems. In *Proceedings of the 39th International Conference on Machine Learning*, pages 7164–7182, 2022.

C. Gong, X. Liu, and Q. Liu. Bi-objective trade-off with dynamic barrier gradient descent. In *Advances in Neural Information Processing Systems*, 2021.

S. Gould, B. Fernando, A. Cherian, P. Anderson, R. S. Cruz, and E. Guo. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *arXiv preprint arXiv:1607.05447*, 2016.

M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. `http://stanford.edu/~boyd/graph_dcp.html`.

M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, Mar. 2014.

E. Y. Hamedani and N. S. Aybat. A primal-dual algorithm with line search for general convex-concave saddle point problems. *SIAM Journal on Optimization*, 31(2):1299–1329, 2021.

N. He, A. Juditsky, and A. Nemirovski. Mirror prox algorithm for multi-term composite minimization and semi-separable problems. *Computational Optimization and Applications*, 61(2):275–319, 2015.

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

M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning*, pages 427–435, 2013.

K. Ji, J. Yang, and Y. Liang. Bilevel optimization: Convergence analysis and enhanced design. In *International Conference on Machine Learning*, pages 4882–4892, 2021.

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

K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski. Dictionary Learning Algorithms for Sparse Representation. *Neural Computation*, 15(2): 349–396, 02 2003.

S. Lacoste-Julien. Convergence rate of Frank-Wolfe for non-convex objectives. *arXiv preprint arXiv:1607.00345*, 2016.

G. Lan. The complexity of large-scale convex programming under a linear optimization oracle. *arXiv preprint arXiv:1309.5550*, 2013.

E. S. Levitin and B. T. Polyak. Constrained minimization methods. *USSR Computational mathematics and mathematical physics*, 6(5):1–50, 1966.

J. Li, B. Gu, and H. Huang. Improved bilevel model: Fast and optimal algorithm with theoretical guarantee. *arXiv preprint arXiv:2009.00690*, 2020.

R. Liu, P. Mu, X. Yuan, S. Zeng, and J. Zhang. A generic first-order algorithmic framework for bi-level programming beyond lower-level singleton. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

R. Liu, X. Liu, X. Yuan, S. Zeng, and J. Zhang. A value-function-based interior-point method for non-convex bi-level optimization. In *Proceedings of the 38th International Conference on Machine Learning*, 2021a.

R. Liu, Y. Liu, S. Zeng, and J. Zhang. Towards gradient-based bilevel optimization with non-convex followers and beyond. In *Advances in Neural Information Processing Systems*, 2021b.

S. Łojasiewicz. Sur la problème de la division. *Studia Mathematica*, 18:87–136, 1959.

Z.-Q. Luo and J.-S. Pang. Error bounds for analytic systems and their applications. *Mathematical Programming*, 67 (1):1–28, 1994.

D. Maclaurin, D. Duvenaud, and R. Adams. Gradient-based hyperparameter optimization through reversible learning. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2113–2122, 2015.

Y. Malitsky. Chambolle-pock and tseng's methods: relationship and extension to the bilevel optimization. *arXiv preprint arXiv:1706.02602*, 2017.

A. Mokhtari, A. Ozdaglar, and A. Jadbabaie. Escaping saddle points in constrained optimization. *Advances in Neural Information Processing Systems (NIPS)*, 2018.

J.-S. Pang. Error bounds in mathematical programming. *Mathematical Programming*, 79(1):299–332, 1997.

F. Pedregosa. Hyperparameter optimization with approximate gradient. In *Proceedings of the 33nd International Conference on Machine Learning (ICML)*, 2016.

F. Pedregosa, G. Negiar, A. Askari, and M. Jaggi. Linearly convergent frank-wolfe with backtracking line-search. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pages 1–10. PMLR, 2020.

A. Rajeswaran, C. Finn, S. M. Kakade, and S. Levine. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems*, pages 113–124, 2019.

A. Rakotomamonjy. Direct optimization of the dictionary learning problem. *IEEE Transactions on Signal Processing*, 61(22):5495–5506, 2013.

V. Roulet and A. d'Aspremont. Sharpness, restart, and acceleration. *SIAM Journal on Optimization*, 30(1):262–289, 2020.

B. Rozemberczki, P. Scherer, Y. He, G. Panagopoulos, A. Riedel, M. Astefanoaei, O. Kiss, F. Beres, , G. Lopez, N. Collignon, and R. Sarkar. PyTorch Geometric Temporal: Spatiotemporal Signal Processing with Neural Machine Learning Models. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, page 4564–4573, 2021.

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

A. Shaban, C.-A. Cheng, N. Hatch, and B. Boots. Truncated back-propagation for bilevel optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1723–1732, 2019.

Y. Shehu, P. T. Vuong, and A. Zemkoho. An inertial extrapolation method for convex simple bilevel optimization. *Optimization Methods and Software*, 36(1):1–19, 2021.

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

D. Sow, K. Ji, Z. Guan, and Y. Liang. A constrained optimization approach to bilevel optimization with multiple inner minima. *arXiv preprint arXiv:2203.01123*, 2022.

A. Tikhonov and V. Arsenin. *Solutions of Ill-Posed Problems*. Wiley, New York, 1977.

N. Tripuraneni, C. Jin, and M. Jordan. Provable meta-learning of linear representations. In *Proceedings of the 38th International Conference on Machine Learning*, pages 10434–10443, 2021.

H.-K. Xu. Viscosity approximation methods for nonexpansive mappings. *Journal of Mathematical Analysis and Applications*, 298(1):279–291, 2004.

Y. Xu. Iteration complexity of inexact augmented lagrangian methods for constrained convex programming. *Mathematical Programming*, 185(1):199–244, 2021.

M. Yaghoobi, T. Blumensath, and M. E. Davies. Dictionary learning for sparse approximations with the majorization method. *IEEE Transactions on Signal Processing*, 57(6): 2178–2191, 2009.

I. Yamada. The hybrid steepest-descent method for variational inequalityproblems over the intersection of the fixed-point sets of nonexpansive mappings. In *Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications*, pages 473–504. North-Holland, 2001.

M. B. Zafar, I. Valera, M. G. Rogriguez, and K. P. Gummadi. Fairness Constraints: Mechanisms for Fair Classification. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 962–970. PMLR, 2017.

A SUPPORTING LEMMAS

A.1 Proof of Lemma 1

Let $\mathbf{x}_g^*$ be any point in $\mathcal{X}_g^*$, i.e., any optimal solution of the lower-level problem. By definition, we have $g(\mathbf{x}_g^*) = g^*$. Since g is convex and $g^* \leq g(\mathbf{x}_0)$, we have

$$g(\mathbf{x}_0) - g(\mathbf{x}_k) \geq g^* - g(\mathbf{x}_k) = g(\mathbf{x}_g^*) - g(\mathbf{x}_k) \geq \left\langle \nabla g(\mathbf{x}_k), \mathbf{x}_g^* - \mathbf{x}_k \right\rangle,$$

which implies $\mathbf{x}_g^* \in \mathcal{X}_k$. Hence, we conclude that $\mathcal{X}_g^* \subseteq \mathcal{X}_k$.

A.2 Improvement in One Step

The following lemma characterizes the improvement of both the upper-level and lower-level objective values after one step of Algorithm 1.

Lemma 2. *Let $\{\mathbf{x}_k\}_{k=0}^K$ be the sequence generated by Algorithm 1. Suppose Assumption 1 holds, then for any $k \geq 0$ we have*

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) - \gamma_k \mathcal{G}(\mathbf{x}_k) + \frac{1}{2}\gamma_k^2 L_f D^2, \tag{14}$$

$$g(\mathbf{x}_{k+1}) \leq (1-\gamma_k) g(\mathbf{x}_k) + \gamma_k g(\mathbf{x}_0) + \frac{1}{2}\gamma_k^2 L_g D^2, \tag{15}$$

Proof. Since the gradient of f is L_f-Lipschitz and $\mathcal{Z}$ is bounded with diameter D, we have

$$\begin{aligned} f(\mathbf{x}_{k+1}) &\leq f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{x}_{k+1} - \mathbf{x}_k \rangle + \frac{1}{2} L_f \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \\ &= f(\mathbf{x}_k) + \gamma_k \langle \nabla f(\mathbf{x}_k), \mathbf{s}_k - \mathbf{x}_k \rangle + \frac{1}{2} L_f \gamma_k^2 \|\mathbf{s}_k - \mathbf{x}_k\|^2 \\ &\leq f(\mathbf{x}_k) + \gamma_k \langle \nabla f(\mathbf{x}_k), \mathbf{s}_k - \mathbf{x}_k \rangle + \frac{1}{2} L_f \gamma_k^2 D^2. \end{aligned} \tag{16}$$

Now using the definition of $\mathbf{s}_k$ in (10), the definition of $\mathcal{G}(\mathbf{x})$ in (7) and Lemma 1, we obtain

$$\langle \nabla f(\mathbf{x}_k), \mathbf{s}_k - \mathbf{x}_k \rangle = \min_{\mathbf{s} \in \mathcal{X}_k} \langle \nabla f(\mathbf{x}_k), \mathbf{s} - \mathbf{x}_k \rangle \leq \min_{\mathbf{s} \in \mathcal{X}_g^*} \langle \nabla f(\mathbf{x}_k), \mathbf{s} - \mathbf{x}_k \rangle = -\mathcal{G}(\mathbf{x}_k). \tag{17}$$

Then (14) follows from (16) and (17).

Similarly, since the gradient of g is L_g-Lipschitz, we have

$$g(\mathbf{x}_{k+1}) \leq g(\mathbf{x}_k) + \gamma_k \langle \nabla g(\mathbf{x}_k), \mathbf{s}_k - \mathbf{x}_k \rangle + \frac{1}{2} L_g \gamma_k^2 D^2. \tag{18}$$

Moreover, since $\mathbf{s}_k \in \mathcal{X}_k$, from the definition of $\mathcal{X}_k$ in (10) we get $\langle \nabla g(\mathbf{x}_k), \mathbf{s}_k - \mathbf{x}_k \rangle \leq g(\mathbf{x}_0) - g(\mathbf{x}_k)$. Combining this with (18) leads to (15). □

B PROOF OF THE MAIN THEOREMS

B.1 Proof of Theorem 1

We first prove the convergence rate of the upper-level objective f, which largely mirrors the standard analysis of the CG method (Jaggi, 2013). Since $\mathbf{x}^* \in \mathcal{X}_g^*$ and f is convex, from the definition of $\mathcal{G}(\mathbf{x}_k)$ in (7) we have

$$\mathcal{G}(\mathbf{x}_k) = \max_{\mathbf{s} \in \mathcal{X}_g^*} \{\langle \nabla f(\mathbf{x}_k), \mathbf{x}_k - \mathbf{s} \rangle\} \geq \langle \nabla f(\mathbf{x}_k), \mathbf{x}_k - \mathbf{x}^* \rangle \geq f(\mathbf{x}_k) - f^*. \tag{19}$$

Subtracting f^* from both sides of (14) in Lemma 2 and using (19), we obtain that

$$f(\mathbf{x}_{k+1}) - f^* \leq (1-\gamma_k)(f(\mathbf{x}_k) - f^*) + \frac{1}{2}\gamma_k^2 L_f D^2. \tag{20}$$

Now define $A_k = k(k+1)$. By substituting $\gamma_k = 2/(k+2)$ and multiplying both sides of (20) by A_{k+1}, we get

$$A_{k+1}(f(\mathbf{x}_{k+1}) - f^*) \leq A_k(f(\mathbf{x}_k) - f^*) + \frac{2(k+1)}{k+2} L_f D^2 \leq A_k(f(\mathbf{x}_k) - f^*) + 2L_f D^2.$$

Hence, if follows from induction that

$$A_K(f(\mathbf{x}_K) - f^*) \leq A_0(f(\mathbf{x}_0) - f^*) + 2KL_f D^2 \quad \Rightarrow \quad f(\mathbf{x}_K) - f^* \leq \frac{2KL_f D^2}{A_k} = \frac{2L_f D^2}{K+1}.$$

This completes the first part of the proof.

The proof for the lower-level problem follows from similar arguments. By subtracting $g(\mathbf{x}_0)$ from both sides of (15) in Lemma 2, we have

$$g(\mathbf{x}_{k+1}) - g(\mathbf{x}_0) \leq (1-\gamma_k)(g(\mathbf{x}_k) - g(\mathbf{x}_0)) + \frac{1}{2}\gamma_k^2 L_g D^2. \tag{21}$$

By substituting $\gamma_k = 2/(k+2)$ and multiplying both sides of (21) by A_{k+1}, we obtain

$$A_{k+1}(g(\mathbf{x}_{k+1}) - g(\mathbf{x}_0)) \leq A_k(g(\mathbf{x}_k) - g(\mathbf{x}_0)) + 2L_g D^2.$$

Hence, if follows from induction that

$$A_K(g(\mathbf{x}_K) - g(\mathbf{x}_0)) \leq 2KL_g D^2 \quad \Rightarrow \quad g(\mathbf{x}_K) - g(\mathbf{x}_0) \leq \frac{2KL_g D^2}{A_k} = \frac{2L_g D^2}{K+1}.$$

Since $g(\mathbf{x}_0) - g^* \leq \epsilon_g/2$, we obtain

$$g(\mathbf{x}_K) - g^* \leq \frac{2L_g D^2}{K+1} + \frac{1}{2}\epsilon_g,$$

which completes the proof.

B.2 Proof of Theorem 2

Since we use a fixed stepsize in Theorem 2, in the following we will write $\gamma_k = \gamma$.

We first consider the upper-level objective f. The analysis here is similar to the one by Mokhtari et al. (2018). By using (14) in Lemma 2, we have

$$\mathcal{G}(\mathbf{x}_k) \leq \frac{f(\mathbf{x}_k) - f(\mathbf{x}_{k+1})}{\gamma} + \frac{1}{2}\gamma L_f D^2.$$

Summing both sides of the above inequality from $k=0$ to $K-1$, we get

$$\sum_{k=0}^{K-1} \mathcal{G}(\mathbf{x}_k) \leq \frac{f(\mathbf{x}_0) - f(\mathbf{x}_K)}{\gamma} + \frac{1}{2}K\gamma L_f D^2 \leq \frac{f(\mathbf{x}_0) - \underline{f}}{\gamma} + \frac{1}{2}K\gamma L_f D^2,$$

where we used the fact that $f(\mathbf{x}_K) \geq \underline{f} = \min_{\mathbf{x}\in Z} f(\mathbf{x})$. This further implies that

$$\min_{0\leq k\leq K-1} \mathcal{G}(\mathbf{x}_k) \leq \frac{1}{K}\sum_{k=0}^{K-1} \mathcal{G}(\mathbf{x}_k) \leq \frac{f(\mathbf{x}_0) - \underline{f}}{\gamma K} + \frac{1}{2}\gamma L_f D^2. \tag{22}$$

To upper bound the right-hand side of (22), note that our choices of the stepsize γ and the number of iterations K satisfy

$$\gamma \leq \frac{\epsilon_f}{L_f D^2} \quad \text{and} \quad K \geq \frac{2(f(\mathbf{x}_0) - \underline{f})}{\epsilon_f \gamma}.$$

Thus, we have

$$\min_{0\leq k\leq K-1} \mathcal{G}(\mathbf{x}_k) \leq \frac{f(\mathbf{x}_0) - \underline{f}}{\gamma K} + \frac{1}{2}\gamma L_f D^2 \leq \frac{\epsilon_f}{2} + \frac{\epsilon_f}{2} = \epsilon_f.$$

This guarantees that $\mathcal{G}(\mathbf{x}_{k^*}) \leq \epsilon_f$ by choosing $k^* = \operatorname{argmin}_{0\leq k\leq K-1} \mathcal{G}(\mathbf{x}_k)$.

Now we move to the analysis of the lower-level objective g. For any $k \geq 0$, by applying induction on (15) in Lemma 2, it follows that

$$g(\mathbf{x}_k) - g(\mathbf{x}_0) \leq \frac{1}{2} L_g D^2 \sum_{j=0}^{k-1} \gamma^2 (1-\gamma)^j \leq \frac{1}{2} L_g D^2 \gamma,$$

where we used $\sum_{j=0}^{k-1}(1-\gamma)^j \leq 1/\gamma$ in the last inequality. Furthermore, since $g(\mathbf{x}_0) - g^* \leq \epsilon_g/2$ and $\gamma \leq \frac{\epsilon_g}{L_g D^2}$, this implies that $g(\mathbf{x}_k) - g^* \leq \frac{1}{2}\epsilon_g + \frac{1}{2}\epsilon_g = \epsilon_g$ for any $0 \leq k \leq K-1$. In particular, we can take $k = k^*$ and conclude that $g(\mathbf{x}_{k^*}) - g^* \leq \epsilon_g$. This completes the proof.

C PROOFS UNDER HÖLDERIAN ERROR BOUND ASSUMPTION

C.1 Proof of Proposition 1

Since $\mathcal{X}_g^*$ is closed and compact, we can let $\hat{\mathbf{x}}^* = \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}_g^*} \|\mathbf{x} - \hat{\mathbf{x}}\|$ such that $\|\hat{\mathbf{x}}^* - \hat{\mathbf{x}}\| = \operatorname{dist}(\hat{\mathbf{x}}, \mathcal{X}_g^*)$. By Assumption 2, we obtain

$$\frac{\alpha}{r}\|\hat{\mathbf{x}}^* - \hat{\mathbf{x}}\|^r \leq g(\hat{\mathbf{x}}) - g^* \leq \epsilon_g \quad \Leftrightarrow \quad \|\hat{\mathbf{x}}^* - \hat{\mathbf{x}}\| \leq \left(\frac{r\epsilon_g}{\alpha}\right)^{\frac{1}{r}}.$$

When f is convex, we have

$$f(\hat{\mathbf{x}}) - f^* \geq f(\hat{\mathbf{x}}) - f(\hat{\mathbf{x}}^*) \geq \langle \nabla f(\hat{\mathbf{x}}^*), \hat{\mathbf{x}} - \hat{\mathbf{x}}^* \rangle \geq -\|\nabla f(\hat{\mathbf{x}}^*)\|_* \|\hat{\mathbf{x}} - \hat{\mathbf{x}}^*\| \geq -M\left(\frac{r\epsilon_g}{\alpha}\right)^{\frac{1}{r}},$$

where we used the convexity of f in the second inequality. When f is non-convex, we have

$$\begin{aligned}
\mathcal{G}(\hat{\mathbf{x}}) = \max_{\mathbf{s} \in \mathcal{X}_g^*} \{\langle \nabla f(\hat{\mathbf{x}}), \hat{\mathbf{x}} - \mathbf{s} \rangle\} &\geq \langle \nabla f(\hat{\mathbf{x}}), \hat{\mathbf{x}} - \hat{\mathbf{x}}^* \rangle \\
&= \langle \nabla f(\hat{\mathbf{x}}) - \nabla f(\hat{\mathbf{x}}^*), \hat{\mathbf{x}} - \hat{\mathbf{x}}^* \rangle + \langle \nabla f(\hat{\mathbf{x}}^*), \hat{\mathbf{x}} - \hat{\mathbf{x}}^* \rangle \\
&\geq -\|\nabla f(\hat{\mathbf{x}}) - \nabla f(\hat{\mathbf{x}}^*)\|_* \|\hat{\mathbf{x}} - \hat{\mathbf{x}}^*\| - \|\nabla f(\hat{\mathbf{x}}^*)\| \|\hat{\mathbf{x}} - \hat{\mathbf{x}}^*\| \\
&\geq -L_f \|\hat{\mathbf{x}} - \hat{\mathbf{x}}^*\|^2 - M\|\hat{\mathbf{x}} - \hat{\mathbf{x}}^*\| \qquad (23) \\
&\geq -M\left(\frac{r\epsilon_g}{\alpha}\right)^{\frac{1}{r}} - L_f\left(\frac{r\epsilon_g}{\alpha}\right)^{\frac{2}{r}},
\end{aligned}$$

where we used the fact that ∇f is L_f-Lipschitz in (23). This completes the proof.

C.2 Proof of Corollary 1

In the first case where f is convex, we set $\epsilon_g = \frac{\alpha}{r}\left(\frac{\epsilon_f}{M}\right)^r$. By Theorem 1, we have $f(\mathbf{x}_K) - f^* \leq \epsilon_f$ and $g(\mathbf{x}_K) - g^* \leq \epsilon_g$ when

$$K \geq \max\left\{\frac{2L_f D^2}{\epsilon_f} - 1, \frac{4L_g D^2}{\epsilon_g} - 1\right\} = \max\left\{\frac{2L_f D^2}{\epsilon_f} - 1, \frac{4rM^r L_g D^2}{\alpha \epsilon_f^r} - 1\right\} = \mathcal{O}\left(\frac{1}{\epsilon_f^r}\right).$$

Moreover, Proposition 1 implies that $f(\mathbf{x}_K) - f^* \geq -M\left(\frac{r\epsilon_g}{\alpha}\right)^{\frac{1}{r}} \geq -\epsilon_f$. Putting all pieces together, we conclude that $|f(\mathbf{x}_K) - f^*| \leq \epsilon_f$ and $g(\mathbf{x}_K) - g^* \leq \epsilon_g$ after $K = \mathcal{O}(1/\epsilon_f^r)$ iterations.

In the second case where f is non-convex, we set $\epsilon_g = \min\{\frac{\alpha}{r}\left(\frac{\epsilon_f}{2M}\right)^r, \frac{\alpha}{r}\left(\frac{\epsilon_f}{2L_f}\right)^{r/2}\}$. By Theorem 2, we can find $k^* \in \{0, 1, \ldots, K-1\}$ such that $\mathcal{G}(\mathbf{x}_{k^*}) \leq \epsilon_f$ and $g(\mathbf{x}_{k^*}) - g^* \leq \epsilon_g$ when

$$\begin{aligned}
K &\geq (f(\mathbf{x}_0) - \underline{f}) \cdot \max\left\{\frac{2L_f D^2}{\epsilon_f^2}, \frac{2L_g D^2}{\epsilon_f \epsilon_g}\right\} \\
&= (f(\mathbf{x}_0) - \underline{f}) \cdot \max\left\{\frac{2L_f D^2}{\epsilon_f^2}, \frac{2r(2M)^r L_g D^2}{\alpha \epsilon_f^{r+1}}, \frac{2r(2L_f)^{\frac{r}{2}} L_g D^2}{\alpha \epsilon_f^{\frac{r}{2}+1}}\right\} = \mathcal{O}\left(\frac{1}{\epsilon_f^{r+1}}\right).
\end{aligned}$$

Moreover, Proposition 1 implies that $\mathcal{G}(\mathbf{x}_{k^*}) \geq -M\left(\frac{r\epsilon_g}{\alpha}\right)^{\frac{1}{r}} - L_f\left(\frac{r\epsilon_g}{\alpha}\right)^{\frac{2}{r}} \geq -\frac{\epsilon_f}{2} - \frac{\epsilon_f}{2} = -\epsilon_f$. Thus, we conclude $|\mathcal{G}(\mathbf{x}_{k^*})| \leq \epsilon_f$ and $g(\mathbf{x}_{k^*}) - g^* \leq \epsilon_g$ after $K = \mathcal{O}(1/\epsilon_f^{r+1})$ iterations.

D PRIMAL-DUAL METHOD FOR THE BILEVEL PROBLEM

In this section, we discuss the convergence rate of primal-dual type methods for solving the bilevel problem in (1). We consider the setting as in Theorem 1, in which both f and g are convex and smooth. To simplify the discussion, we further assume $\mathcal{Z} = \{\mathbf{z} \in \mathcal{X} \mid \mathbf{A}\mathbf{z} \leq \mathbf{b}\}$ where $\mathbf{A} \in \mathbb{R}^{m\times d}$, $\mathbf{b} \in \mathbb{R}^m$, and $\mathcal{X}$ is a convex and easy-to-project compact set.

To obtain the reformulation in (2), one first needs to estimate the optimal value g^* of the lower-level problem. Since it is a convex program with linear constraints, we can implement a first-order primal-dual method (see, e.g., (Chambolle and Pock, 2016)) to find g_0 such that $|g_0 - g^*| \leq \epsilon_g/4$ within at most $\mathcal{O}(\frac{L_g+\|\mathbf{A}\|}{\epsilon_g})$ iterations[1]. Next, problem (1) can be cast as the following convex optimization problem with linear and nonlinear convex constraints:

$$\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{A}\mathbf{x} \leq \mathbf{b},\, g(\mathbf{x}) \leq g_0 + \frac{\epsilon_g}{2}, \tag{24}$$

where we add the term $\frac{\epsilon_g}{2}$ to ensure that the Slater's condition holds. Now we can apply any classic or accelerated first-order primal-dual methods (He et al., 2015; Xu, 2021; Hamedani and Aybat, 2021) to find a solution of problem (24) that is both ϵ_f-suboptimal and $\frac{\epsilon_g}{4}$-infeasible. For example, the optimal convergence rates obtained by Xu (2021) and Hamedani and Aybat (2021) imply that after K iterations, the average iterate $\bar{\mathbf{x}}_K$ satisfies

$$\max\left\{|f(\bar{\mathbf{x}}_K) - f(\mathbf{x}^*_\epsilon)|, |g(\bar{\mathbf{x}}_K) - g(\mathbf{x}^*_\epsilon)|\right\} \leq \Delta/K,$$

where $\mathbf{x}^*_\epsilon$ denotes an optimal solution of problem (24), $\Delta \triangleq \mathcal{O}((L_f + L_g + C_g)D^2 + C_g\,|\lambda_1^*|^2 + \|\mathbf{A}\|\,\|\boldsymbol{\lambda}_2^*\|^2)$, C_g is the Lipschtiz constant of g, and $\lambda_1^* \in \mathbb{R}$ and $\boldsymbol{\lambda}_2^* \in \mathbb{R}^m$ denote an arbitrary dual optimal solution corresponding to the nonlinear and linear constraints in problem (24), respectively. Using the fact that $f(\mathbf{x}^*_\epsilon) \leq f(\mathbf{x}^*)$ and $g(\mathbf{x}^*_\epsilon) \leq g_0 + \frac{\epsilon_g}{2} \leq g^* + \frac{3}{4}\epsilon_g$, we conclude

$$f(\bar{\mathbf{x}}_K) - f(\mathbf{x}^*) \leq \Delta/K \quad \text{and} \quad |g(\bar{\mathbf{x}}_K) - g(\mathbf{x}^*)| \leq \Delta/K + \frac{3}{4}\epsilon_g.$$

Therefore, to achieve an (ϵ_f, ϵ_g)-optimal solution of problem (1), a primal-dual method overall requires $\mathcal{O}\left(\frac{L_g+\|\mathbf{A}\|}{\epsilon_g} + \frac{\Delta}{\min\{\epsilon_f,\epsilon_g\}}\right)$ primal-dual gradient calls, whereas our proposed method overall requires $\mathcal{O}\left(\frac{L_g}{\epsilon_g} + \frac{(L_f+L_g)D^2}{\min\{\epsilon_f,\epsilon_g\}}\right)$ linear minimization oracle calls. In particular, we observe that the convergence guarantee of primal-dual methods heavily relies on the norm of the dual optimal variable $|\lambda_1^*|$, which may tend to infinity as ϵ approaches zero and the problem in (24) becomes nearly degenerate.

D.1 Numerical Example

Here we consider a simple two-dimensional example to illustrate the numerical instability of primal-dual methods applied to the relaxed problem (24). To this end, consider the following problem

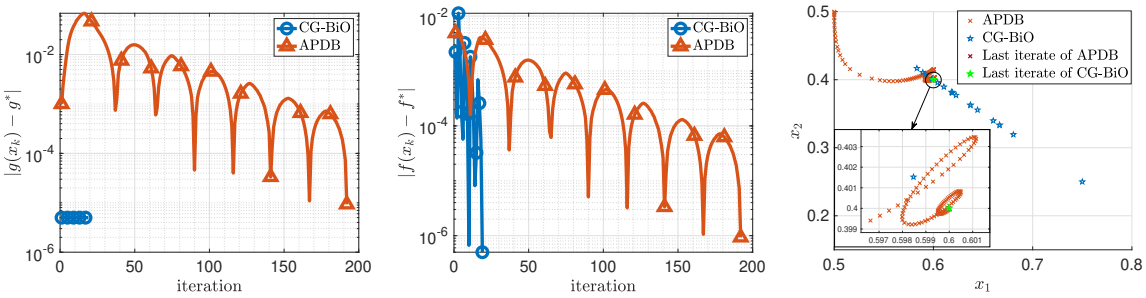
$$\min_{\mathbf{x}\in\mathbb{R}^2} 0.5x_1^2 - 0.5x_1 + 0.1x_2 \quad \text{s.t.} \quad \mathbf{x} \in \operatorname*{argmin}_{\mathbf{z}\in\mathcal{Z}}\{-z_1 - z_2\}, \tag{25}$$

where $\mathcal{Z} = \{\mathbf{z} \in \mathbb{R}^2_+ \mid z_1 + z_2 \leq 1, 4z_1 + 6z_2 \leq 5\}$. The lower-level problem has multiple solutions which can be described by $\mathcal{X}_g^* = \{\mathbf{x} \in \mathbb{R}^2 \mid x_1 + x_2 = 1, x_1 \in [0.5, 1], x_2 \in [0, 0.5]\}$ and the optimal solution of (25) is $(x_1^*, x_2^*) = (0.6, 0.4)$. We implemented accelerated primal-dual method with backtracking (APDB) proposed by Hamedani and Aybat (2021), one of the state-of-the-art primal-dual methods, and compared it with our proposed method CG-BiO. Figure 4 illustrates the iteration trajectories of both methods. We selected the relaxing parameter in (24) as $\epsilon = 10^{-5}$ for APDB. We also used the same accuracy for ϵ_g and ϵ_f when implementing CG-BiO. The primal-dual method finds an ϵ-solution (dark red cross) within 193 iterations while CG-BiO finds an ϵ-solution (green star) within 20 iterations. Furthermore, we observe a more stable numerical behavior for CG-BiO in comparison with APDB, which corroborates our theoretical analysis above.

E ADDITIONAL MOTIVATING EXAMPLES

In this section, we provide some additional remarks and two more examples for the bilevel problem in (1).

[1]Note that this complexity can be improved to the optimal rate of $\mathcal{O}(\sqrt{\frac{L_g}{\epsilon_g}} + \frac{\|\mathbf{A}\|}{\epsilon_g})$ using an accelerated method.



(a) Lower-level gap (b) Upper-level gap (c) Iteration trajectory

Figure 4: The performance of CG-BiO compared with APDB on problem (25).

E.1 Lexicographic Optimization

In Section 2.1, we have seen two instances of lexicographic optimization, where we use the secondary loss to improve generalization (Example 1) or promote fairness (Example 2). In the following, we describe another standard example where we use regularization to tackle ill-posed problems.

Example 4 (Ill-posed Optimization). Without an explicit regularization, the empirical risk minimization problem $\min_{\mathbf{z}\in\mathcal{Z}} \ell_{\rm tr}(\mathbf{z})$ can be ill-posed, i.e., it has multiple optimal solutions or is sensitive to small perturbation in the input data. To tackle this issue, we can consider a secondary objective function $\mathcal{R}(\cdot)$ as another criterion to select one of the optimal solutions with some desired property. For example, we can find the minimal ℓ_2-norm solution by choosing $\mathcal{R}(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$. Such a problem can be formulated as the following bilevel problem:

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

E.2 Lifelong Learning

A popular framework known as A-GEM (Chaudhry et al., 2019) formulates the lifelong learning problem as follows:

$$\min_{\boldsymbol{\beta}} \frac{1}{n'}\sum_{i=1}^{n'} \ell(\langle \mathbf{x}_i', \boldsymbol{\beta}\rangle, y_i') \quad \text{s.t.} \quad \sum_{(\mathbf{x}_i,y_i)\in\mathcal{M}} \ell(\langle \mathbf{x}_i, \boldsymbol{\beta}\rangle, y_i) \le \sum_{(\mathbf{x}_i,y_i)\in\mathcal{M}} \ell(\langle \mathbf{x}_i, \boldsymbol{\beta}^{(t-1)}\rangle, y_i). \tag{26}$$

Here, the objective function is the training loss on the current task $\mathcal{D}_t = \{(\mathbf{x}_i', y_i')\}_{i=1}^{n'}$, and the inequality constraint ensures that the model with parameter $\boldsymbol{\beta}$ performs no worse than the previous one on the episodic memory $\mathcal{M}$ (i.e., some stored data samples from the previous tasks).

In this paper, we consider a variant of problem (26), where we further tighten the constraint and require that the model also minimizes the error on the episodic memory. This leads to the following bilevel problem:

$$\min_{\boldsymbol{\beta}} \frac{1}{n'}\sum_{i=1}^{n'} \ell(\langle \mathbf{x}_i', \boldsymbol{\beta}\rangle, y_i') \quad \text{s.t.} \quad \boldsymbol{\beta} \in \operatorname*{argmin}_{\mathbf{z}} \sum_{(\mathbf{x}_i,y_i)\in\mathcal{M}} \ell(\langle \mathbf{x}_i, \mathbf{z}\rangle, y_i). \tag{27}$$

Example 3 can be viewed as an instance of problem (27) where the learning problem at hand is dictionary learning. Below we present another related example from representation learning.

Example 5 (Representation Learning). In meta-learning problems, we aim to pre-train a model that can be easily fine-tuned to new tasks. This can be often achieved by learning a compact representation that is shared among multiple tasks (Tripuraneni et al., 2021; Du et al., 2021; Collins et al., 2022). In particular, consider a multi-task linear representation learning problem with T tasks at the training time. We assume that the data points for the i-th task are generated according to $y_i^j = \mathbf{w}_i^{*\top}\mathbf{B}^{*\top}\mathbf{x}_i^j + n_i^j$ for $j = 1,\ldots,m_i$, where n_i^j is some random noise and $\mathbf{B}^* \in \mathbb{R}^{k\times d}$ is a common representation that maps the input in $\mathbb{R}^d$ to a lower dimensional feature vector in $\mathbb{R}^k$. When we have access to a diverse set of tasks such that their heads $\{\mathbf{w}_i^*\}_{i=1}^T$ span $\mathbb{R}^k$, it is shown that one can find the ground truth representation $\mathbf{B}^*$ by solving the following problem:

$$\min_{\mathbf{B}} \min_{\mathbf{w}_1,\ldots,\mathbf{w}_T} \sum_{i=1}^{T}\sum_{j=1}^{m_i} \left(y_i^j - \mathbf{w}_i^\top \mathbf{B}^\top \mathbf{x}_i^j\right)^2 \qquad \text{s.t.} \quad \|\mathbf{B}\|_F \le \Delta,\ \|\mathbf{w}_i\|_1 \le \delta,\ i = 1,\ldots,T,$$

where we impose the norm constraints on $\mathbf{B}$ and $\{\mathbf{w}_i\}_{i=1}^T$ for some parameters $\Delta, \delta > 0$ to resolve the scale invariance of the problem.

However, if the tasks at the training time are not diverse enough, then we can only learn a partial represention, i.e., a subset of the feature maps in $\mathbf{B}^*$. One way to further improve the learned representation is to leverage the new tasks we observe during the test time. Concretely, let $\hat{\mathbf{w}}_1^*, \ldots, \hat{\mathbf{w}}_T^*$ and $\hat{\mathbf{B}}_{\text{tr}}^*$ denote the output of the training procedure. When we are given a new task at the test time, we can improve the representation $\hat{\mathbf{B}}_{\text{tr}}^*$ by solving the following bilevel problem:

$$\min_{\mathbf{B}\in\mathbb{R}^{k\times d}} \min_{\mathbf{w}_{T+1}\in\mathbb{R}^k} f(\mathbf{B}, \mathbf{w}_{T+1}) \qquad \text{s.t.} \quad \mathbf{B} \in \operatorname*{argmin}_{\|\mathbf{B}'\|_F\le\Delta} g(\mathbf{B}),\ \|\mathbf{w}_{T+1}\|_1 \le \delta, \tag{28}$$

where $f(\mathbf{B}, \mathbf{w}_{T+1}) \triangleq \sum_{j=1}^{m_{T+1}} \left(y_{T+1}^j - \mathbf{w}_{T+1}^\top \mathbf{B}^\top \mathbf{x}_{T+1}^j\right)^2$ is the loss over the test set and $g(\mathbf{B}) \triangleq \sum_{i=1}^T \sum_{j=1}^{m_i} \left(y_i^j - \hat{\mathbf{w}}_i^{*\top} \mathbf{B}'^\top \mathbf{x}_i^j\right)^2$ is the loss over the training set. The rationale is that the solution of problem (28) can fit to both the old training tasks and the new test task, and hence is a better approximation of $\mathbf{B}^*$ compared to $\hat{\mathbf{B}}_{\text{tr}}^*$. This way, we maintain the feature maps learned at the training time and at the same time learn new feature maps from the test task. Note that in problem (28) the upper-level function is nonconvex, while the lower-level problem is convex with multiple solutions.

F EXPERIMENT DETAILS

In this section, we include more details of the numerical experiments in Section 5.

For completeness, we briefly review the update rules of MNG (Beck and Sabach, 2014), BiG-SAM (Sabach and Shtern, 2017), a-IRG (Kaushik and Yousefian, 2021) and DBGD (Gong et al., 2021) in the setup of problem (1). In the following, we use $\Pi_{\mathcal{Z}}(\cdot)$ to denote the Euclidean projection onto the set $\mathcal{Z}$.

- Each step of MNG requires solving the following subproblem:

$$\mathbf{x}_{k+1} = \operatorname*{argmin}_{\mathbf{x}\in Q_k\cap W_k} f(\mathbf{x}), \tag{29}$$

  where

$$\begin{aligned} Q_k &\triangleq \left\{\mathbf{z}\in\mathbb{R}^d \mid \langle G_M(\mathbf{x}_k), \mathbf{x}_k - \mathbf{z}\rangle \ge \frac{3}{4M}\|G_M(\mathbf{x}_k)\|^2\right\}, \\ W_k &\triangleq \left\{\mathbf{z}\in\mathbb{R}^d \mid \langle \nabla f(\mathbf{x}_k), \mathbf{z} - \mathbf{x}_k\rangle \ge 0\right\}, \\ G_M(\mathbf{x}) &\triangleq M\left[\mathbf{x} - \Pi_{\mathcal{Z}}\left(\mathbf{x} - \frac{1}{M}\nabla g(\mathbf{x})\right)\right], \end{aligned}$$

  and $M \ge L_g$ is a hyperparameter. As we can see, the implementation of MNG is only feasible when the subproblem in (29) is easy to solve. In particular, it is typically computationally intractable when the upper-level objective f is non-convex.

- BiG-SAM is given by

$$\begin{aligned} \mathbf{y}_{k+1} &= \Pi_{\mathcal{Z}}(\mathbf{x}_k - \eta_g \nabla g(\mathbf{x}_k)), \\ \mathbf{z}_{k+1} &= \mathbf{x}_k - \eta_f \nabla f(\mathbf{x}_k), \\ \mathbf{x}_{k+1} &= \alpha_{k+1}\mathbf{z}_{k+1} + (1-\alpha_{k+1})\mathbf{y}_{k+1}, \end{aligned}$$

  where $\eta_f \le \frac{2}{L_f}$ and $\eta_g \le \frac{1}{L_g}$ are stepsizes and $\alpha_k = \min\{\frac{\gamma}{k}, 1\}$ for some $\gamma > 0$. We note that the analysis by Sabach and Shtern (2017) requires the upper-level objective to be strongly convex, and therefore is not directly applicable in our setting. Nevertheless, we also implement their method and manually set the hyperparameters.

- The a-IRG algorithm is given by

$$\mathbf{x}_{k+1} = \Pi_{\mathcal{Z}}\left(\mathbf{x}_k - \gamma_k(\nabla g(\mathbf{x}_k) + \eta_k \nabla f(\mathbf{x}_k))\right),$$

  where γ_k is the stepsize and η_k is the regularization parameter. In our experiment, we choose $\gamma_k = \gamma_0/\sqrt{k+1}$ and $\eta_k = \eta_0/(k+1)^{1/4}$ for some constants γ_0 and η_0.

- The DBGD algorithm is given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \gamma_k(\nabla f(\mathbf{x}_k) + \lambda_k \nabla g(\mathbf{x}_k)), \tag{30}$$

where γ_k is the stepsize and we set λ_k as

$$\lambda_k = \max\left\{\frac{\phi(\mathbf{x}_k) - \langle \nabla f(\mathbf{x}_k), \nabla g(\mathbf{x}_k)\rangle}{\|\nabla g(\mathbf{x}_k)\|^2}, 0\right\} \quad \text{and} \quad \phi(\mathbf{x}) = \min\{\alpha(g(\mathbf{x}) - \hat{g}), \beta\|\nabla g(\mathbf{x})\|^2\}.$$

Here, α and β are hyperparameters and $\hat{g}$ is a lower bound on g^*. In our experiment, we choose $\alpha = \beta = 1$ and $\hat{g} = 0$. We also note that Gong et al. (2021) only considered unconstrained simple bilevel problems where $\mathcal{Z} = \mathbb{R}^d$. To enforce the constraint, we replace (30) with the update rule $\mathbf{x}_{k+1} = \Pi_{\mathcal{Z}}(\mathbf{x}_k - \gamma_k(\nabla f(\mathbf{x}_k) + \lambda_k \nabla g(\mathbf{x}_k)))$.

F.1 Over-parameterized Regression

Dataset Generation. The original Wikipedia Math Essential dataset (Rozemberczki et al., 2021) consists of an 1068×731 matrix. We randomly select one of the columns as the outcome vector $\mathbf{b} \in \mathbb{R}^{1068}$ and the rest as the data matrix $\mathbf{A} \in \mathbb{R}^{1068\times730}$. We let $\lambda = 1$ in the experiment, i.e., the constraint set is given by $\mathcal{Z} = \{\boldsymbol{\beta} \mid \|\boldsymbol{\beta}\|_1 \le 1\}$.

Initialization. We run the standard CG method with the stepsizes chosen as $2/(k+2)$ on the lower-level problem in (4). We terminate the procedure once the FW gap is no more than $\epsilon_g/2 = 5\times10^{-5}$ or we have reached the maximum number of iterations $N_{\max} = 10^4$.

Implementation Details. For our CG-BiO method, we set the target accuracies for the upper-level and lower-level problems to $\epsilon_f = 10^{-4}$ and $\epsilon_g = 10^{-4}$, respectively. We choose the stepsizes as $\gamma_k = 2/(k+12)$ to avoid instability due to large initial stepsizes. In each iteration, we need to solve a subproblem in the form of

$$\min_{\mathbf{s}} \ \langle \nabla f(\boldsymbol{\beta}_k), \mathbf{s}\rangle \qquad \text{s.t.} \quad \|\mathbf{s}\|_1 \le \lambda, \ \langle \nabla g(\boldsymbol{\beta}_k), \mathbf{s} - \boldsymbol{\beta}_k\rangle \le g(\boldsymbol{\beta}_0) - g(\boldsymbol{\beta}_k). \tag{31}$$

We can reformulate the above problem as a linear program by introducing $\mathbf{s}^+, \mathbf{s}^- \ge 0$ such that $\mathbf{s} = \mathbf{s}^+ - \mathbf{s}^-$. Specifically, problem (31) becomes

$$\begin{aligned}&\min_{\mathbf{s}^+,\mathbf{s}^-} \ \langle \nabla f(\boldsymbol{\beta}_k), \mathbf{s}^+ - \mathbf{s}^-\rangle \\ &\text{s.t.} \quad \mathbf{s}^+, \mathbf{s}^- \ge 0, \ \langle \mathbf{s}^+, \mathbf{1}\rangle + \langle \mathbf{s}^-, \mathbf{1}\rangle \le \lambda, \ \langle \nabla g(\boldsymbol{\beta}_k), \mathbf{s}^+ - \mathbf{s}^- - \boldsymbol{\beta}_k\rangle \le g(\boldsymbol{\beta}_0) - g(\boldsymbol{\beta}_k),\end{aligned}$$

where $\mathbf{1} \in \mathbb{R}^d$ is the all-one vector.

For MNG, we set $M = \lambda_{\max}(\mathbf{A}_{\text{tr}}^\top \mathbf{A}_{\text{tr}})$. For BiG-SAM, we set $\eta_f = 2/\lambda_{\max}(\mathbf{A}_{\text{val}}^\top \mathbf{A}_{\text{val}})$, $\eta_g = 1/\lambda_{\max}(\mathbf{A}_{\text{tr}}^\top \mathbf{A}_{\text{tr}})$ and $\gamma = 10$. For a-IRG, we set $\gamma_0 = 0.01$ and $\eta_0 = 1$. For DBGD, we set $\gamma_k = 10^{-4}$.

F.2 Fair Classification

Dataset Generation. We preprocess the original Adult income dataset (Dua and Graff, 2019) with the same procedure as in (Zafar et al., 2017), leading to a dataset with 50 attributes for prediction. Moreover, we standardize all the attributes such that they lie between 0 and 1. In our experiment, we set $\lambda = 100$.

Initialization. We run the standard CG method with backtracking line search (Pedregosa et al., 2020) on the sparse logistic regression problem in (12). We terminate the procedure once the FW gap is no more than $\epsilon_g/2 = 5\times10^{-5}$ or we have reached the maximum number of iterations $N_{\max} = 10^4$.

Implementation Details. For our CG-BiO method, we set $\epsilon_f = 10^{-4}$ and $\epsilon_g = 10^{-4}$, respectively. We choose the stepsize as $\gamma_k = 0.005/\sqrt{k+1}$ instead of a constant stepsize as suggested by Theorem 2. Empirically, we observe that this leads to faster convergence. The subproblem we need to solve is in the same form as problem (31), which is also solved by a LP solver.

For BiG-SAM, we set $\eta_f = \eta_g = 0.1$ and $\gamma = 1$. For a-IRG, we set $\gamma_0 = 5$ and $\eta_0 = 0.1$. For DBGD, we set $\gamma_k = 0.08$.

F.3 Dictionary Learning

Dataset Generation. Each of the sparse coefficient vectors $\{\mathbf{x}_i\}_{i=1}^{250}$ and $\{\mathbf{x}'_k\}_{k=1}^{200}$ has 5 nonzero entries, whose locations are randomly chosen. Also, the absolute values of those nonzero weights are drawn uniformly from the interval $[0.2, 1]$.

The entries of the random noise vectors $\{\mathbf{n}_i\}_{i=1}^{250}$ and $\{\mathbf{n}'_k\}_{k=1}^{200}$ follow i.i.d. Gaussian distribution with mean 0 and standard deviation 0.01.

Initialization. The initialization consists of two phases. In the first phase, we run the standard CG algorithm on both the variables $\mathbf{D} \in \mathbb{R}^{25\times 40}$ and $\mathbf{X} \in \mathbb{R}^{40\times 250}$ for 10^4 iterations with the stepsize chosen as $1/\sqrt{k+1}$ ($k \geq 0$ is the iteration counter). Then in the second phase, we keep the variable $\mathbf{X}$ fixed and update $\mathbf{D}$ using the standard CG algorithm with exact line search. We terminate the procedure and output $\hat{\mathbf{D}}$ and $\hat{\mathbf{X}}$ until the FW gap with respect to $\mathbf{D}$ is no more than $\epsilon_g = 10^{-6}$.

Implementation Details. We choose $\delta = 3$ in both problems (5) and (6). All three algorithms start from the same initial point. We initialize $\tilde{\mathbf{D}} \in \mathbb{R}^{25\times 50}$ as the concatenation of $\hat{\mathbf{D}} \in \mathbb{R}^{25\times 40}$ and 10 columns of all zeros. Moreover, we initialize the variable $\tilde{\mathbf{X}}$ randomly by drawing its entries from a standard Gaussian distribution and normalizing each column to have a ℓ_1-norm of δ. For our CG-BiO method, we choose the stepsize as $\gamma_k = 0.3/\sqrt{k+1}$ instead of a constant stepsize as suggested by Theorem 2. Empirically, we observe that this leads to faster convergence. The same stepsize rule is also used in the baseline CG method. In each iteration, we need to solve a subproblem in the form of

$$\min_{\tilde{\mathbf{D}}} \ \langle \nabla f_{\tilde{\mathbf{D}}}(\tilde{\mathbf{D}}_k, \tilde{\mathbf{X}}_k), \tilde{\mathbf{D}} \rangle \qquad \text{s.t.} \quad \|\tilde{\mathbf{d}}_i\|_2 \leq 1, \ \langle \nabla g(\tilde{\mathbf{D}}_k), \tilde{\mathbf{D}} - \tilde{\mathbf{D}}_k \rangle \leq g(\tilde{\mathbf{D}}_0) - g(\tilde{\mathbf{D}}_k). \tag{32}$$

By using the KKT condition, it can be shown that the above problem is equivalent to finding a zero of the following one-dimensional nonlinear equation involving $\lambda \geq 0$:

$$\tilde{\mathbf{D}} = \Pi_{\mathcal{Z}}(\nabla f_{\tilde{\mathbf{D}}}(\tilde{\mathbf{D}}_k, \tilde{\mathbf{X}}_k) + \lambda \nabla g(\tilde{\mathbf{D}}_k)), \quad \langle \nabla g(\tilde{\mathbf{D}}_k), \tilde{\mathbf{D}} - \tilde{\mathbf{D}}_k \rangle = g(\tilde{\mathbf{D}}_0) - g(\tilde{\mathbf{D}}_k),$$

where the projection on $\mathcal{Z} = \{\tilde{\mathbf{D}} \in \mathbb{R}^{25\times 40} : \|\tilde{\mathbf{d}}_i\|_2 \leq 1, \ i = 1, \ldots, 40\}$ amounts to a column-wise projection on the Euclidean ball. In practice, we find that it can be solved efficiently by MATLAB's root-finding solver.

For BiG-SAM, we set $\eta_f = \eta_g = 0.1$ and $\gamma = 10$. For a-IRG, we set $\gamma_0 = 0.01$ and $\eta_0 = 1$. For DBGD, we set $\gamma_k = 0.1$.