
Penalty-based Methods for Simple Bilevel Optimization under Hölderian Error Bounds

Pengyu Chen*
School of Data Science
Fudan University
pychen22@m.fudan.edu.cn

Xu Shi*
School of Data Science
Fudan University
xshi22@m.fudan.edu.cn

Rujun Jiang†
School of Data Science
Fudan University
rjjiang@fudan.edu.cn

Jiulin Wang
School of Data Science
Fudan University
wangjiulin@fudan.edu.cn

Abstract

This paper investigates simple bilevel optimization problems where we minimize an upper-level objective over the optimal solution set of a convex lower-level objective. Existing methods for such problems either only guarantee asymptotic convergence, have slow sublinear rates, or require strong assumptions. To address these challenges, we propose a penalization framework that delineates the relationship between approximate solutions of the original problem and its reformulated counterparts. This framework accommodates varying assumptions regarding smoothness and convexity, enabling the application of specific methods with different complexity results. Specifically, when both upper- and lower-level objectives are composite convex functions, under an α -Hölderian error bound condition and certain mild assumptions, our algorithm attains an $(\epsilon, \epsilon^\beta)$ -optimal solution of the original problem for any $\beta > 0$ within $\mathcal{O}\left(\sqrt{1/\epsilon^{\max\{\alpha, \beta\}}}\right)$ iterations. The result can be improved further if the smooth part of the upper-level objective is strongly convex. We also establish complexity results when the upper- and lower-level objectives are general nonsmooth functions. Numerical experiments demonstrate the effectiveness of our algorithms.

1 Introduction

Bilevel optimization involves embedding one optimization problem within another, creating a hierarchical structure where the upper-level problem’s feasible set is influenced by the lower-level problem. This framework frequently occurs in various real-world scenarios, such as meta-learning [Bertinetto et al., 2018, Rajeswaran et al., 2019], hyper-parameter optimization [Chen et al., 2024, Franceschi et al., 2018, Shaban et al., 2019], reinforcement learning [Mingyi et al., 2020] and adversarial learning [Bishop et al., 2020, Wang et al., 2021, 2022]. In this paper, we concentrate on a subset of bilevel optimization known as simple bilevel optimization (SBO), which has garnered significant interest in the machine learning community due to its relevance in dictionary learning [Beck and Sabach, 2014, Jiang et al., 2023], lexicographic optimization [Kissel et al., 2020, Gong et al., 2021], lifelong learning [Malitsky, 2017, Jiang et al., 2023]; see more details in Appendix A.

*Equal contribution

†Corresponding author

SBO aims to find an optimal solution that minimizes the upper-level objective over the solution set of the lower-level problem. In other words, we are interested in solving the following problem:

$$\min_{\mathbf{x} \in \mathbb{R}^n} F(\mathbf{x}) \quad \text{s.t. } \mathbf{x} \in \arg \min_{\mathbf{z} \in \mathbb{R}^n} G(\mathbf{z}). \quad (\text{P})$$

Here $F, G : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$ are proper, convex, and lower semi-continuous functions. We also assume that the optimal solution set of the lower-level problem, denoted as X_{opt} , is nonempty. Moreover, since G is convex and lower semi-continuous, it holds that X_{opt} is closed and convex [Bertsekas et al., 2003, Proposition 1.2.2 and Page 49].

In this paper, we first reformulate problem (P) into the constrained form:

$$\min_{\mathbf{x} \in \mathbb{R}^n} F(\mathbf{x}) \quad \text{s.t. } G(\mathbf{x}) - G^* \leq 0, \quad (\text{P}_{\text{Val}})$$

where G^* represents the optimal value of the unconstrained lower-level problem.

Based on this reformulation, we consider the following penalization of (P_{Val}),

$$\min_{\mathbf{x} \in \mathbb{R}^n} \Phi_\gamma(\mathbf{x}) = F(\mathbf{x}) + \gamma p(\mathbf{x}), \quad (\text{P}_\gamma)$$

where $p(\mathbf{x}) := G(\mathbf{x}) - G^*$ is the so-called residual function and $\gamma > 0$ is the penalized parameter. Obviously, we have $p(\mathbf{x}) \geq 0$, and $p(\mathbf{x}) = 0$ if and only if $\mathbf{x} \in X_{\text{opt}}$.

Denote F^* and G^* as the optimal values of problem (P) and the lower-level problem $\min_{\mathbf{x} \in \mathbb{R}^n} G(\mathbf{x})$, respectively. We aim to find an (ϵ_F, ϵ_G) -optimal solution $\tilde{\mathbf{x}}^*$ of problem (P), which satisfies

$$F(\tilde{\mathbf{x}}^*) - F^* \leq \epsilon_F, \quad G(\tilde{\mathbf{x}}^*) - G^* \leq \epsilon_G. \quad (1)$$

Moreover, a point $\tilde{\mathbf{x}}_\gamma^*$ is said to be an ϵ -optimal solution of problem (P_γ) if

$$\Phi_\gamma(\tilde{\mathbf{x}}_\gamma^*) - \Phi_\gamma^* \leq \epsilon,$$

where Φ_γ^* is the optimal value of problem (P_γ).

1.1 Related work

Various approaches have been developed to solve problem (P) [Cabot, 2005, Solodov, 2007, Sabach and Shtern, 2017, Dutta and Pandit, 2020, Gong et al., 2021]. Among those, one category that is the most related to penalization formulation (P_γ) is the regularization method, which integrates the upper- and lower-level objectives through Tikhonov regularization [Tikhonov and Arsenin, 1977]

$$\min_{\mathbf{x} \in \mathbb{R}^n} \eta(\mathbf{x}) := \sigma F(\mathbf{x}) + G(\mathbf{x}), \quad (\text{P}_{\text{Reg}})$$

where σ is the so-called regularization parameter. When F is strongly convex and its domain is compact, Amini and Yousefian [2019] extended the IR-PG method from Solodov [2007], which achieved a asymptotic convergence rate for the upper-level problem and a convergence rate of $\mathcal{O}(1/K^{0.5-b})$ for the lower-level problem, where $b \in (0, 0.5)$. Malitsky [2017] studied a version of Tseng's accelerated gradient method and showed a convergence rate of $\mathcal{O}(1/K)$ for the lower-level problem, while the convergence rate for the upper-level objective is not explicitly provided. Kaushik and Yousefian [2021] proposed an iteratively regularized gradient (a-IRG) method which obtains a complexity of $\mathcal{O}(1/K^{0.5-b})$ and $\mathcal{O}(1/K^b)$ for the upper- and lower-level objective, respectively, where $b \in (0, 0.5)$. Inspired by this research, and under a quasi-Lipschitz assumption for F , Merchav and Sabach [2023] introduced a bi-subgradient (Bi-SG) method. This method demonstrates convergence rates of $\mathcal{O}(1/K^b)$ and $\mathcal{O}(1/K^{1-b})$ for the lower- and upper-level objectives, respectively, where $b \in (0.5, 1)$. In their framework, the convergence rate of the upper-level objective can be improved to be linear when F is strongly convex. Recently, under the weak-sharp minima assumption of the lower-level problem, Samadi et al. [2023] proposed a regularized accelerated proximal method (R-APM), showing a convergence rate of $\mathcal{O}(1/K^2)$ for both upper- and lower-level objectives. When the domain is compact and F, G are both smooth, Giang-Tran et al. [2023] proposed an iteratively regularized conditional gradient (IR-CG) method, which ensures convergence rates of $\mathcal{O}(1/K^p)$ and $\mathcal{O}(1/K^{1-p})$ for upper- and lower-level objectives, respectively, where $p \in (0, 1)$.

Despite the abundance of existing methodologies yielding non-asymptotic convergence outcomes, their efficacy is frequently contingent upon additional assumptions. Denote L_{f_1} and L_{g_1} as the Lipschitz constants for the gradients of the smooth components in the upper- and lower-level objectives,

respectively. Specifically, when F is strongly convex and G is smooth, Beck and Sabach [2014] presented the Minimal Norm Gradient (MNG) method and provided the asymptotic convergence to the optimal solution set and a convergence rate of $\mathcal{O}(L_{g_1}^2/\epsilon^2)$ for the lower-level problem. When F is assumed to be smooth, Jiang et al. [2023] introduced a conditional gradient-based bilevel optimization (CG-BiO) method, which invokes at most $\mathcal{O}(\max\{L_{f_1}/\epsilon_F, L_{g_1}/\epsilon_G\})$ of linear optimization oracles to achieve an (ϵ_F, ϵ_G) -optimal solution. Shen et al. [2023] combined an online framework with the mirror descent algorithm and established a convergence rate of $\mathcal{O}(1/\epsilon^3)$ for both upper- and lower-level objectives, assuming a compact domain and boundedness of the functions and gradients at both levels. Furthermore, they showed that the convergence rate can be improved to $\mathcal{O}(1/\epsilon^2)$ under additional structural assumptions. For a concise overview of overall methodologies, including their assumptions and convergence outcomes, refer to Table 1 in Appendix B.

For general bilevel optimization problems, there have been recent results on convergent guarantees [Shen and Chen, 2023, Sow et al., 2022, Chen et al., 2023, Huang, 2023]. Among those, the one that is the most related to ours is [Shen and Chen, 2023]. It investigates the case when the upper-level objective is nonconvex and gives convergence results under additional assumptions [Shen and Chen, 2023, Theorem 3 and 4]. However, as the general bilevel optimization problem is nonconvex, the algorithms in the literature often converge to weak stationary points, while our method for SBO converges to global optimal solution.

1.2 Our approach

Our approach is straightforward. Firstly, we introduce a penalization framework delineating the connection between approximate solutions of problems (P) and (P_γ) . This framework enables the attainment of an (ϵ_F, ϵ_G) -optimal solution by solving problem (P_γ) approximately. Subsequently, our focus shifts solely to resolving the unconstrained problem (P_γ) . Depending on varying assumptions regarding smoothness and convexity, we can employ different methods such as the accelerated proximal gradient (APG) methods [Beck and Teboulle, 2009, Nesterov, 2013, Lin and Xiao, 2014] to solve problem (P_γ) . We summarize our main contributions as follows.

- We propose a framework that explicitly examines the relationship between an ϵ -optimal solution of penalty formulation (P_γ) and an (ϵ_F, ϵ_G) -optimal solution of problem (P). We also provide a lower bound for the metric $F(\mathbf{x}) - F^*$.
- When F and G are both composite convex functions, we provide a penalty-based APG algorithm that attains an $(\epsilon, \epsilon^\beta)$ -optimal solution of problem (P) within $\mathcal{O}(\sqrt{1/\epsilon^{\max\{\alpha, \beta\}}})$ iterations. If the upper-level objective is strongly convex, the complexity can be improved to $\mathcal{O}(\sqrt{1/\epsilon^{\max\{\alpha-1, \beta-1\}}} \log \frac{1}{\epsilon})$. We also apply our method for the scenario where both the upper- and lower-level objectives are generalized nonsmooth convex functions.
- We present adaptive versions of PB-APG and PB-APG-sc with warm-start, which dynamically adjust the penalty parameters, and solve the associated penalized problem with adaptive accuracy. The adaptive ones have similar complexity results as their primal counterparts but can achieve superior performance in some experiments.

Utilizing the penalization method to address the original SBO problem is a novel approach. While Tikhonov regularization may seem similar to our framework, its principles differ. Implementing Tikhonov regularization necessitates the "slow condition" ($\lim_{k \rightarrow \infty} \sigma_k = 0, \sum_{k=0}^{\infty} \sigma_k = +\infty$), which requires iterative solutions for each iteration. In contrast, our method simply involves solving a single optimization problem (P_γ) for a given γ . Furthermore, we establish a relationship between the approximate solutions of the original bilevel problem and those of the reformulated single-level problem (P_γ) for a specific γ . This is the first theoretical result connecting the original bilevel problem to the penalization problem, accompanied by an optimal non-asymptotic complexity result.

2 The penalization framework

We begin by outlining specific assumptions for F and G , as detailed below.

Assumption 2.1. The set $S := \bigcup_{\mathbf{x} \in X_{\text{opt}}} \partial F(\mathbf{x})$ is bounded with a diameter $l_F := \max_{\xi \in S} \|\xi\|$.

Note that the type of subdifferential ∂F used here is the most general form for a convex function, as detailed in [Bertsekas et al., 2003, Section 4.2]. When the upper-level objective F is non-convex,

we replace the assumption with the condition that the upper-level objective is Lipschitz continuous (cf. Theorems 2.7 and 2.8).

Assumption 2.2 (Hölderian error bound). The function $p(\mathbf{x}) := G(\mathbf{x}) - G^*$ satisfies the Hölderian error bound with exponent $\alpha \geq 1$ and $\rho > 0$. Namely,

$$\text{dist}(\mathbf{x}, X_{\text{opt}})^\alpha \leq \rho p(\mathbf{x}), \forall \mathbf{x} \in \text{dom}(G),$$

where $\text{dist}(\mathbf{x}, X_{\text{opt}}) := \inf_{\mathbf{y} \in X_{\text{opt}}} \|\mathbf{x} - \mathbf{y}\|$.

We remark that Hölderian error bounds are satisfied by many practical problems and widely used in optimization literature [Pang, 1997, Bolte et al., 2017, Zhou and So, 2017, Roulet and d'Aspremont, 2020, Jiang and Li, 2022]. There are two notable special cases: (i) when $\alpha = 1$, we often refer to X_{opt} as a set of weak sharp minima of G [Burke and Ferris, 1993, Studniarski and Ward, 1999, Burke and Deng, 2005, Samadi et al., 2023]; (ii) when $\alpha = 2$, Assumption 2.2 is known as the quadratic growth condition [Drusvyatskiy and Lewis, 2018a]. Additional examples of functions exhibiting Hölderian error bound, along with their corresponding parameters, are presented in Appendix C.

We are now ready to establish the connection between approximate solutions of problems (P) and (P_γ) . The subsequent two lemmas build upon the work of Shen and Chen [2023] for (general) bilevel optimization. Compared with their work, we generalize the exponent α from 2 to $\alpha \geq 1$, providing a more general result. Furthermore, we also derive a lower bound for the penalized parameter for all $\alpha \geq 1$ and present a theoretical framework for these scenarios.

Lemma 2.3. *Suppose that Assumptions 2.1 and 2.2 hold with $\alpha > 1$. Then, for any $\epsilon > 0$, an optimal solution of problem (P) is an ϵ -optimal solution of problem (P_γ) when $\gamma \geq \rho l_F^\alpha (\alpha - 1)^{\alpha-1} \alpha^{-\alpha} \epsilon^{1-\alpha}$.*

Lemma 2.3 establishes the relationship between an optimal solution of problem (P) and an ϵ -optimal solution of problem (P_γ) when $\alpha > 1$. It also provides a lower bound for γ , which plays a pivotal role in the complexity results. The proofs of this paper are deferred to Appendix E.

The lemma presented below yields a more favorable outcome when $\alpha = 1$, which is referred to as exact penalization. Notably, this specific result is not discussed in Shen and Chen [2023].

Lemma 2.4. *Suppose that Assumptions 2.1 and 2.2 hold with $\alpha = 1$. Then an optimal solution of problem (P) is also an optimal solution of problem (P_γ) if $\gamma \geq \rho l_F$, and vice versa if $\gamma > \rho l_F$. In this case, we say that there is an exact penalization between problems (P) and (P_γ) .*

For simplicity, we define

$$\gamma^* = \begin{cases} \rho l_F^\alpha (\alpha - 1)^{\alpha-1} \alpha^{-\alpha} \epsilon^{1-\alpha} & \text{if } \alpha > 1 \\ \rho l_F & \text{if } \alpha = 1 \end{cases}. \quad (2)$$

Based on Lemmas 2.3 and 2.4, we give an overall relationship of approximate solutions between problems (P_γ) and (P).

Theorem 2.5. *Suppose that Assumptions 2.1 and 2.2 hold. For any given $\epsilon > 0$ and $\beta > 0$, let*

$$\gamma = \gamma^* + \begin{cases} 2l_F^\beta \epsilon^{1-\beta} & \text{if } \alpha > 1, \\ l_F^\beta \epsilon^{1-\beta} & \text{if } \alpha = 1, \end{cases}$$

with γ^ defined in (2). If $\tilde{\mathbf{x}}_\gamma^*$ is an ϵ -optimal solution of problem (P_γ) , then $\tilde{\mathbf{x}}_\gamma^*$ is an $(\epsilon, l_F^{-\beta} \epsilon^\beta)$ -optimal solution of problem (P).*

Particularly, we are also able to establish a lower bound for $F(\tilde{\mathbf{x}}_\gamma^*) - F^*$ under the same conditions outlined in Theorem 2.5.

Theorem 2.6. *Suppose that the conditions in Theorem 2.5 hold. Then, $\tilde{\mathbf{x}}_\gamma^*$ satisfies the following suboptimality lower bound,*

$$F(\tilde{\mathbf{x}}_\gamma^*) - F^* \geq -l_F (\rho l_F^{-\beta} \epsilon^\beta)^{\frac{1}{\alpha}}.$$

By setting $\beta = \alpha$, we obtain $F(\tilde{\mathbf{x}}_\gamma^*) - F^* \geq -\rho^{\frac{1}{\alpha}} \epsilon$, which along with Theorem 2.5 gives

$$|F(\tilde{\mathbf{x}}_\gamma^*) - F^*| \leq \max\{\epsilon, \rho^{\frac{1}{\alpha}} \epsilon\}.$$

We emphasize that the lower bound established in Theorem 2.6 is an intrinsic property of problem (P) under Assumptions 2.1 and 2.2. This property is independent of the algorithms we present.

2.1 The upper-level function is non-convex

Note that the upper-level objective F is required to be convex in the above context (cf. Theorem 2.5). This raises a question: while Theorem 2.5 establishes the relationship between approximate solutions of problems (P) and (P_γ) , the distinction between the global or local optimal solutions of problem (P) and (P_γ) remains unclear when F is non-convex.

We first establish the relationship between global optimal solutions of problems (P) and (P_γ) when F is non-convex, which is similar to Theorem 2.5.

Theorem 2.7. *Suppose that Assumption 2.2 holds, G is convex, and F is l -Lipschitz continuous on $\text{dom}(F)$. For any given $\epsilon > 0$ and $\beta > 0$, let*

$$\gamma = \gamma^* + \begin{cases} 2l^\beta \epsilon^{1-\beta} & \text{if } \alpha > 1, \\ l^\beta \epsilon^{1-\beta} & \text{if } \alpha = 1, \end{cases} \quad (3)$$

where γ^* is given by (2). If $\tilde{\mathbf{x}}_\gamma^*$ is an ϵ -global optimal solution of problem (P_γ) , then $\tilde{\mathbf{x}}_\gamma^*$ is an $(\epsilon, l^{-\beta} \epsilon^\beta)$ -global optimal solution of problem (P).

Theorem 2.7 provides the relationship between the global optimal solutions of problems (P_γ) and (P). However, the relationship between local optimal solutions of these problems is more intricate than those of the global ones [Shen and Chen, 2023]. Given $r > 0$ and $\mathbf{z} \in \mathbb{R}^n$, define $\mathcal{B}(\mathbf{z}, r) := \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{z}\| \leq r\}$. We present the following theorem, which demonstrates that local optimal solutions of problem (P_γ) can serve as approximate local optimal solutions of problem (P).

Theorem 2.8. *Suppose that Assumption 2.2 holds and G is convex. Let \mathbf{x}_γ^* be a local optimal solution of problem (P_γ) on $\mathcal{B}(\mathbf{x}_\gamma^*, r)$. Assume F is l -Lipschitz continuous on $\mathcal{B}(\mathbf{x}_\gamma^*, r)$. Then \mathbf{x}_γ^* is an approximate local optimal solution of problem (P) that satisfies $F(\mathbf{x}_\gamma^*) - F_{\mathcal{B}}^* \leq 0$ and $G(\mathbf{x}_\gamma^*) - G^* \leq \epsilon$ when $\alpha > 1$ and $\gamma \geq (\frac{\rho l^\alpha}{\epsilon^{\alpha-1}})^{\frac{1}{\alpha}}$, where $F_{\mathcal{B}}^*$ is the optimal value of problem (P) on $\mathcal{B}(\mathbf{x}_\gamma^*, r) \cap X_{\text{opt}}$. Furthermore, \mathbf{x}_γ^* is a local optimal solution of problem (P) when $\alpha = 1$ and $\gamma > \rho l$.*

Indeed, the relationship between approximate local optimal solutions of problems (P_γ) and (P) is more intricate than the connection among global solutions presented in Theorem 2.5. These interactions will be the focus of our future work. The proofs of Theorems 2.7 and 2.8 are presented in Appendixes E.5 and E.6.

3 Main algorithms

In this section, we concentrate on addressing problem (P), making various assumptions, and offering distinct convergence outcomes.

3.1 Both objectives are convex composite functions

In this scenario, we address problem (P) where F and G are both composite functions, i.e., $F = f_1 + f_2$ and $G = g_1 + g_2$.

Assumption 3.1. F and G satisfy the following assumptions.

- (1) The gradient of $f_1(\mathbf{x})$, denoted as ∇f_1 , is L_{f_1} -Lipschitz continuous on $\text{dom}(F)$;
- (2) The gradient of $g_1(\mathbf{x})$, denoted as ∇g_1 , is L_{g_1} -Lipschitz continuous on $\text{dom}(G)$;
- (3) f_2 and g_2 are proper, convex, lower semicontinuous, and possibly non-smooth.

We remark that Assumption 3.1(1)(3) is more general than many existing papers in the literature. Specifically, while previous works such as Beck and Sabach [2014], Amini and Yousefian [2019], Jiang et al. [2023], Giang-Tran et al. [2023] require the upper-level objective to be smooth or strongly convex, we simply assume that F is a composite function composed of a smooth convex function and a possibly non-smooth convex function. For the lower-level objective, previous works such as Beck and Sabach [2014], Amini and Yousefian [2019], Jiang et al. [2023], Giang-Tran et al. [2023] impose smoothness assumptions and, in some cases, convexity and compactness constraints on the domain; while our approach does not require these additional constraints, allowing for more flexibility and generality as presented in Assumption 3.1(2)(3).

We are now prepared to introduce two algorithms: the penalty-based accelerated proximal gradient (PB-APG) algorithm and its adaptive counterpart, the aPB-APG to solve problem (P_γ) and, subsequently, to obtain an (ϵ_F, ϵ_G) -optimal solution of problem (P).

To simplify notations, we omit the constant term $-\gamma G^*$, and rewrite problem (P_γ) as follows,

$$\min_{\mathbf{x} \in \mathbb{R}^n} \Phi_\gamma(\mathbf{x}) := \phi_\gamma(\mathbf{x}) + \psi_\gamma(\mathbf{x}), \quad (P_\Phi)$$

where $\phi_\gamma(\mathbf{x}) = f_1(\mathbf{x}) + \gamma g_1(\mathbf{x})$ and $\psi_\gamma(\mathbf{x}) = f_2(\mathbf{x}) + \gamma g_2(\mathbf{x})$ represent the smooth and nonsmooth parts, respectively. Then, it follows that the gradient of $\phi_\gamma(\mathbf{x})$ is L_γ -Lipschitz continuous with $L_\gamma = L_{f_1} + \gamma L_{g_1}$.

To implement the APG methods, we need another assumption concerning $\psi_\gamma(\mathbf{x})$.

Assumption 3.2. For any $\gamma > 0$, the function $\psi_\gamma(\mathbf{x})$ is prox-friendly, i.e., the proximal mapping

$$\text{prox}_{t\psi_\gamma}(\mathbf{y}) := \arg \min_{\mathbf{x} \in \mathbb{R}^n} \left\{ \psi_\gamma(\mathbf{x}) + \frac{1}{2t} \|\mathbf{x} - \mathbf{y}\|^2 \right\},$$

is easy to compute for any $t > 0$.

The function $\psi_\gamma(\mathbf{x})$ represents the sum of two non-smooth functions, and proximal mapping for such function sums is widely studied and used in the literature [Yu, 2013, Pustelnik and Condat, 2017, Adly et al., 2019, Boob et al., 2023, Latafat et al., 2023]. This assumption is also a more general requirement compared to many existing algorithms [Sabach and Shtern, 2017, Giang-Tran et al., 2023]. It is important to note that our assumption is more general than existing literature. In the simple bilevel literature, when employing proximal mappings, researchers often consider the scenario where only one level contains a nonsmooth term (see, e.g., [Jiang et al., 2023, Doron and Shtern, 2023, Samadi et al., 2023, Merchav and Sabach, 2023]). In this case, the proximal mapping of the sum $f_2 + \gamma g_2$ is then reduced to the proximal mapping of either f_2 or g_2 , which is a more easily satisfied condition.

3.1.1 Accelerated proximal gradient-based algorithm

We apply the APG algorithm [Beck and Teboulle, 2009, Lin and Xiao, 2014, Nesterov, 2013] to solve problem (P_Φ) , as outlined in Algorithm 1. Moreover, if the Lipschitz constant L_γ is unknown or computationally infeasible, line search [Beck and Teboulle, 2009] can be adopted and will yield almost the same complexity bound. For brevity, we denote Algorithm 1 as $\hat{\mathbf{x}} = \text{PB-APG}(\phi_\gamma, \psi_\gamma, L_{f_1}, L_{g_1}, \mathbf{x}_0, \epsilon)$, where $\hat{\mathbf{x}}$ represents an ϵ -optimal solution of (P_Φ) .

Algorithm 1 Penalty-based APG (PB-APG)

- 1: **Input:** $\gamma, L_\gamma = L_{f_1} + \gamma L_{g_1}, \mathbf{x}_{-1} = \mathbf{x}_0 \in \mathbb{R}^n, R > 0, t_{-1} = t_0 = 1, k = 0, \epsilon > 0$ and $\{t_k\}$.
 - 2: **for** $k \geq 0$ **do**
 - 3: $\mathbf{y}_k = \mathbf{x}_k + t_k (t_{k-1}^{-1} - 1) (\mathbf{x}_k - \mathbf{x}_{k-1})$
 - 4: $\mathbf{x}_{k+1} = \text{prox}_{L_\gamma^{-1} \psi_\gamma}(\mathbf{y}_k - L_\gamma^{-1} \nabla \phi_\gamma(\mathbf{y}_k))$
 - 5: **end for**
-

In Algorithm 1, we stop the loop of Line. 3 - 4 if the number of iterations satisfies that:

$$\frac{2(L_f + \gamma L_g)R^2}{(k+1)^2} \leq \epsilon,$$

where R is a constant that satisfies $\|\mathbf{x}_0 - \mathbf{x}^*\| \leq R$.

Combining Theorem 2.5 and [Tseng, 2008, Corollary 2], we establish the following complexity result for problem (P).

Theorem 3.3. *Suppose that Assumptions 2.1, 2.2, 3.1 and 3.2 hold and the sequence $\{t_k\}$ in Algorithm 1 satisfies $\frac{1-t_{k+1}}{t_{k+1}^2} \leq \frac{1}{t_k^2}$. Let γ be given as in Theorem 2.5. Algorithm 1 generates an $(\epsilon, l_F^{-\beta} \epsilon^\beta)$ -optimal solution of problem (P) after at most K iterations, where*

$$K = \mathcal{O} \left(\sqrt{\frac{L_{f_1}}{\epsilon}} + \sqrt{\frac{l_F^{\max\{\alpha, \beta\}} L_{g_1}}{\epsilon^{\max\{\alpha, \beta\}}}} \right).$$

Note that Theorem 3.3 encompasses all possible relationships between the magnitudes of ϵ_F and ϵ_G in (1), as $\alpha \geq 1$ and $\beta > 0$ are arbitrary. Specially, if $\alpha = 1$ and $\beta \leq \alpha$, the number of iterations is $K = \mathcal{O}\left(\sqrt{(L_{f_1} + l_F L_{g_1})/\epsilon}\right)$. This result matches the lower bound complexity for unconstrained smooth or convex composite optimization [Nemirovsky and Yudin, 1983, Woodworth and Srebro, 2016]. Additionally, if $g_1 \equiv 0$, the number of iterations for obtaining an $(\epsilon, \epsilon^\beta)$ -optimal solution of problems (P) is independent of γ , which can be improved to $K = \mathcal{O}(\sqrt{L_{f_1}/\epsilon})$.

Remark 3.4. It is noteworthy that Theorem 1 in a previous paper Samadi et al. [2023] provides the first method that needs $\mathcal{O}(\sqrt{(L_{g_1} + l_F L_{g_1})/\epsilon})$ iterations to achieve an (ϵ, ϵ) solution if $\alpha = 1$ and F is smooth. Nevertheless, our methodology diverges in various respects. First, our approach is rooted in the penalization formulation of problem (P_{Val}), while the approach proposed by Samadi et al. [2023] is based on the Tikhonov regularization [Tikhonov and Arsenin, 1977]. Second, we provide a theoretical framework that clearly delineates the relationship between approximate solutions of problems (P) and (P_γ) for all cases of $\alpha \geq 1$ and F is non-convex, as indicated in Lemmas 2.3, 2.4 and Theorems 2.5, 2.7, 2.8. Therefore, we can first shift our focus from (P) to (P_γ) based on the penalization framework and then use various methods to solve (P_γ), not limited to using the APG methods. Besides, the association between approximate solutions of problem (P) and (P_γ) differs significantly based on whether $\alpha > 1$ or $\alpha = 1$. For the case of $\alpha > 1$, the lower bound comprehensively integrates the accuracy parameter ϵ , which results in a more sophisticated analysis of the convergence result, while Samadi et al. [2023] did not consider the situation when $\alpha > 1$. Third, our method applies to the case that F is composite, while Samadi et al. [2023] requires F to be smooth. Finally, we also propose an adaptive version of our algorithm (see Algorithm 2) that does not require an estimate of γ .

3.1.2 Adaptive version with warm-start mechanism

In practice, the penalty parameter γ might be difficult to determine. This motivates us to propose Algorithm 2, which adaptively updates γ and invokes PB-APG with dynamic γ and solution accuracies.

Algorithm 2 Adaptive PB-APG method (aPB-APG)

- 1: **Input:** $\mathbf{x}_0 \in \mathbb{R}^n$, $\gamma_0 = \gamma_1 > 0$, $L_{f_1}, L_{g_1}, \nu > 1, \eta > 1, \epsilon_0 > 0$.
 - 2: **for** $k \geq 0$ **do**
 - 3: $\phi_k(\mathbf{x}) = f_1(\mathbf{x}) + \gamma_k g_1(\mathbf{x})$
 - 4: $\psi_k(\mathbf{x}) = f_2(\mathbf{x}) + \gamma_k g_2(\mathbf{x})$
 - 5: Invoke $\mathbf{x}_k = \text{PB-APG}(\phi_k, \psi_k, L_{f_1}, L_{g_1}, \mathbf{x}_{k-1}, \epsilon_k)$
 - 6: $\epsilon_{k+1} = \epsilon_k/\eta$
 - 7: $\gamma_{k+1} = \nu\gamma_k$
 - 8: **end for**
-

In Algorithm 2, we adaptively update the penalty parameter γ_k , and invoke the PB-APG to generate an approximate solution for (P_γ) with accuracy $\epsilon = \epsilon_k$. Meanwhile, a warm-start mechanism is employed, meaning that the initial point for each subproblem is the output of the preceding subproblem. The convergence result of Algorithm 2 is as follows.

Theorem 3.5. *Suppose that Assumptions 2.1, 2.2, 3.1, and 3.2 hold. Also assume that for every outcome of inner loop in Algorithm 2, $\|\mathbf{x}_k - \mathbf{x}_k^*\| \leq R$. Let $\epsilon_0 > 0$ be given.*

- When $\alpha > 1$, set $\nu > \eta^{\alpha-1}$, and define $N := \lceil \log_{\eta^{1-\alpha\nu}}(\rho L_F^\alpha (\alpha-1)^{\alpha-1} \alpha^{-\alpha} \epsilon_0^{1-\alpha} / \gamma_0) \rceil_+$ and $\gamma_k^* := \rho L_F^\alpha (\alpha-1)^{\alpha-1} \alpha^{-\alpha} \epsilon_0^{1-\alpha} \eta^{k(\alpha-1)}$.
- When $\alpha = 1$, set $\nu > 1$, and define $N := \lceil \log_\nu(\rho l_F / \gamma_0) \rceil_+$ and $\gamma_k^* := \rho L_F$.

Then, for any $k \geq N$, Algorithm 2 generates an $(\frac{\epsilon_0}{\eta^k}, \frac{2\epsilon_0}{\eta^k(\gamma_0\nu^k - \gamma_k^*)})$ -optimal solution of problem (P) after at most K iterations, where K satisfies

$$K = \mathcal{O}\left(\sqrt{\frac{L_{f_1}\eta^k}{\epsilon_0}} + \sqrt{\frac{L_{g_1}\gamma_0(\eta\nu)^k}{\epsilon_0}}\right).$$

Theorem 3.5 shows that for any given initial accuracy $\epsilon_0 > 0$, Algorithm 2 can produce an approximate solution of problem (P) with the desired accuracy.

Remark 3.6. From Theorem 3.5, one can obtain an $(\epsilon, \frac{\epsilon}{\gamma_0\nu^k - \gamma_k^*})$ -optimal solution of problem (P) within $\mathcal{O}(\sqrt{L_{f_1}/\epsilon} + \sqrt{L_{g_1}/\epsilon^\alpha})$ iterations when $\epsilon/\eta \leq \epsilon_0/\eta^k \leq \epsilon$, which is similar to the complexity results in Theorem 3.3.

3.1.3 The upper-level objective is strongly convex

We investigate the convergence outcomes when the smooth part of the upper-level objective exhibits strong convexity.

Assumption 3.7. $f_1(\mathbf{x})$ is μ -strongly convex on $\text{dom}(F)$ with $\mu > 0$.

Assumption 3.7 is another widely adopted setting in the existing SBO literature [Beck and Sabach, 2014, Sabach and Shtern, 2017, Amini and Yousefian, 2019, Merchav and Sabach, 2023]. Here, we propose a variant of PB-APG that can provide better complexity results than existing methods. Our main integration is an APG-based algorithm, which has been studied in the existing literature [Nesterov, 2013, Lin and Xiao, 2014, Xu, 2022]. In this paper, we adopt the algorithm proposed in Lin and Xiao [2014] and modify it with a constant step-size for simplicity as in Algorithm 3. Similar to Algorithm 1, we denote Algorithm 3 by $\hat{\mathbf{x}} = \text{PB-APG-sc}(\phi_\gamma, \psi_\gamma, \mu, L_{f_1}, L_{g_1}, \mathbf{y}_0, \epsilon)$.

Algorithm 3 PB-APG method for Strong Convexity Case (PB-APG-sc)

- 1: **Input:** $\mu, \gamma, L_\gamma = L_{f_1} + \gamma L_{g_1}, \mathbf{x}_{-1}, \mathbf{y}_0 \in \mathbb{R}^n$.
 - 2: $\tilde{\mathbf{y}} = \mathbf{y}_0 - L_\gamma^{-1} \nabla \phi_\gamma(\mathbf{x}_{-1})$
 - 3: $\tilde{\mathbf{x}} = \text{prox}_{L_\gamma^{-1} \psi_\gamma}(\tilde{\mathbf{y}} - L_\gamma^{-1} \nabla \phi_\gamma(\tilde{\mathbf{y}}))$
 - 4: **Initialization:** Let $\mathbf{x}_{-1} = \mathbf{x}_0 = \tilde{\mathbf{x}}, k = 0$
 - 5: **for** $k \geq 0$ **do**
 - 6: $\mathbf{y}_k = \mathbf{x}_k + \frac{\sqrt{L_\gamma - \sqrt{\mu}}}{\sqrt{L_\gamma + \sqrt{\mu}}}(\mathbf{x}_k - \mathbf{x}_{k-1})$
 - 7: $\mathbf{x}_{k+1} = \text{prox}_{L_\gamma^{-1} \psi_\gamma}(\mathbf{y}_k - L_\gamma^{-1} \nabla \phi_\gamma(\mathbf{y}_k))$
 - 8: **end for**
-

The convergence analysis of Algorithm 3 is in the existing literature [Nesterov, 2013, Lin and Xiao, 2014]. Combining [Lin and Xiao, 2014, Theorem 1] and Theorem 2.5, we have the following complexity result.

Theorem 3.8. *Suppose that Assumptions 2.1, 2.2, 3.1, 3.2, and 3.7 hold. Algorithm 3 can produce an $(\epsilon, l_F^{-\beta} \epsilon^\beta)$ -optimal solution of problem (P) after at most K iterations, where K satisfies*

$$K = \mathcal{O} \left(\sqrt{\frac{L_{f_1}}{\mu}} \log \frac{1}{\epsilon} + \sqrt{\frac{l_F^{\max\{\alpha, \beta\}} L_{g_1}}{\epsilon^{\max\{\alpha-1, \beta-1\}}}} \log \frac{1}{\epsilon} \right).$$

Theorem 3.8 improves the complexity results of Theorem 3.3 significantly. Specifically, when $0 < \beta \leq \alpha = 1$, the convergence rate can be improved to be linear, i.e., $K = \mathcal{O}(\sqrt{L_{f_1}/\mu} \log \frac{1}{\epsilon})$.

Additionally, we present an adaptive variant of PB-APG-sc, termed aPB-APG-sc, which adaptively executes $\mathbf{x}_k = \text{PB-APG-sc}(\phi_k, \psi_k, \mu, L_{f_1}, L_{g_1}, \mathbf{x}_{k-1}, \epsilon_k)$ and enjoys the similar complexity results of Algorithm 3, as delineated in Algorithm 4 within Appendix D.1.

3.2 Both objectives are non-smooth

In this section, we focus on the scenario where both the upper- and lower-level objectives are non-smooth, namely, $f_1 = g_1 \equiv 0$. Additionally, we assume that there is a point $x \in C$ in the lower level problem, where C is either \mathbb{R}^n (the unconstrained case) or a nonempty closed and convex set satisfying $C \subseteq \text{int}(\text{dom}(F) \cap \text{dom}(G))$.

It is worth noting that in the case where both F and G are non-smooth, the convergence result may not be as favorable as those in the previous scenarios. This is primarily due to the limited availability of information and unfavorable properties concerning F and G . In this case, we employ a subgradient method to solve problem (P_γ) , which has been extensively studied in the existing literature [Shor, 2012, Bubeck et al., 2015, Beck, 2017, Nesterov, 2018]. Specifically, we update

$$\mathbf{x}_{k+1} = \text{Proj}_C(\mathbf{x}_k - \eta_k \xi_k), \quad (4)$$

where $\xi_k \in \partial \Phi_\gamma(\mathbf{x}_k)$ is a subgradient of $\Phi_\gamma(\mathbf{x}_k)$, and $\text{Proj}_C(\mathbf{x})$ is the projection of \mathbf{x} onto C .

Let \mathbf{x}_γ^* be an optimal solution of problem (P_γ) and suppose that there exists a constant R such that $\|\mathbf{x}_0 - \mathbf{x}_\gamma^*\| \leq R$. Motivated by Theorem 8.28 in Beck [2017], we establish the subsequent complexity result for problem (P).

Theorem 3.9. *Suppose that Assumption 3.1(3) holds, f_2 and g_2 are l_{f_2} - and l_{g_2} -Lipschitz continuous, respectively. Set step-size $\eta_k = \frac{R}{l_\gamma \sqrt{k+1}}$ in (4). Then, the subgradient method produces an $(\epsilon, l_{f_2}^{-\beta} \epsilon^\beta)$ -optimal*

solution of problem (P) after at most K iterations, where K satisfies

$$K = \mathcal{O} \left(\frac{l_{f_2}^2}{\epsilon^2} + \frac{l_{f_2}^{\max\{2\alpha, 2\beta\}} l_{g_2}^2}{\epsilon^{\max\{2\alpha, 2\beta\}}} \right).$$

For non-smooth SBO problems, our method has lower complexity compared to existing approaches. Specifically, under a bounded domain assumption, Helou and Simões [2017] simply proposed an ϵ -subgradient method with an asymptotic rate towards the optimal solution set. The a-IRG method in Kaushik and Yousefian [2021] achieved convergence rates of $\mathcal{O}(1/\epsilon^{0.5-b})$ and $\mathcal{O}(1/\epsilon^{\frac{1}{b}})$ for the upper- and lower-level objectives, respectively, where $b \in (0, 0.5)$. Setting $b = 0.25$ yields the convergence rates of $\mathcal{O}(1/\epsilon^4)$ for both upper- and lower-level objectives, which indicates that our complexity is more efficient than theirs when $\alpha < 2$ and $\beta \leq \alpha$. Furthermore, the online framework proposed in Shen et al. [2023] performed a complexity of $\mathcal{O}(1/\epsilon^3)$ for both upper- and lower-level objectives. Similarly, our approach prevails over theirs when $\alpha < 1.5$ and $\beta \leq \alpha$.

Strongly convex upper-level objective. Based on Theorem 8.31 in Beck [2017], we next explore the improved complexity result for problem (P) when f_2 is additionally strongly convex.

Theorem 3.10. *Suppose that Assumption 3.1(3) holds, $C \subseteq \text{int}(\text{dom}(F) \cap \text{dom}(G))$, f_2 is l_{f_2} -Lipschitz continuous and μ_{f_2} -strongly convex³, and g_2 is l_{g_2} -Lipschitz continuous. Choose step-size $\eta_k = \frac{2}{\mu_{f_2}(k+1)}$ in (4). Then, the subgradient method produces an $(\epsilon, l_{f_2}^{-\beta} \epsilon^\beta)$ -optimal solution of problem (P) after at most K iterations, where K satisfies*

$$K = \mathcal{O} \left(\frac{l_{f_2}^2}{\mu_{f_2} \epsilon} + \frac{l_{f_2}^{\max\{2\alpha, 2\beta\}} l_{g_2}^2}{\mu_{f_2} \epsilon^{\max\{2\alpha-1, 2\beta-1\}}} \right).$$

To our knowledge, within the context of Theorem 3.10, current findings fail to exploit strong convexity to enhance results. However, our approach capitalizes on distinct structural characteristics that yield superior complexity outcomes relative to Theorem 3.9 in cases where $\alpha < 2$ and $\beta \leq \alpha$.

4 Numerical experiments

We apply our Algorithms 1, 2, 3 and 4 to two simple bilevel optimization problems from the motivating examples in Appendix A. The performances of our methods are compared with several existing methods: MNG [Beck and Sabach, 2014], BiG-SAM [Sabach and Shtern, 2017], DBGD [Gong et al., 2021], a-IRG [Kaushik and Yousefian, 2021], CG-BiO [Jiang et al., 2023], Bi-SG [Merchav and Sabach, 2023] and R-APM [Samadi et al., 2023]. For practical efficiency, we use the Greedy FISTA algorithm proposed in Liang et al. [2022] as the APG method in our approach. Detailed settings and additional experimental results are presented in Appendix F.

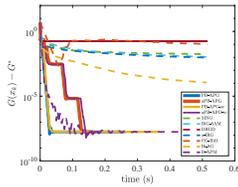


Figure 1: Performances of methods in LRP.

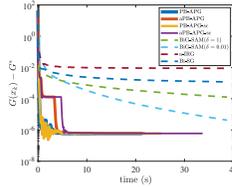
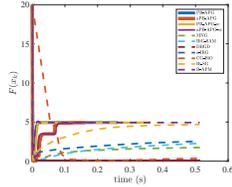
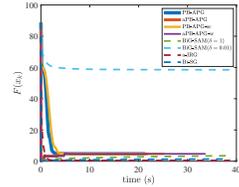


Figure 2: Performances of methods in LSRP.



4.1 Logistic regression problem (LRP)

The LRP reads

$$\min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{x}\|^2 \quad \text{s.t.} \quad \mathbf{x} \in \arg \min_{\mathbf{z} \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^m \log(1 + \exp(-\mathbf{a}_i^T \mathbf{z} b_i)) + I_C(\mathbf{z}), \quad (5)$$

where $I_C(\mathbf{x})$ is the indicator function of the set $C = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_1 \leq \theta\}$ with $\theta = 10$. Our goal is to find a solution to the lower-level problem with the smallest Euclidean norm. The upper-level objective only

³In this case, we must have C bounded, as f_2 is both strongly convex and Lipschitz continuous.

consists of the smooth part, which is 1-strongly convex and 1-smooth; meanwhile, the lower-level objective is a composite function, where the smooth part is $\frac{1}{4m} \lambda_{\max}(A^T A)$ -smooth, and the nonsmooth part is prox-friendly [Duchi et al., 2008].

In this experiment, we compare our methods with MNG, BiG-SAM, DBGD, a-IRG, CG-BiO, and Bi-SG. We plot the values of residuals of the lower-level objective $G(\mathbf{x}_k) - G^*$ and the upper-level objective over time in Figure 1.

As shown in Figure 1, the PB-APG, aPB-APG, PB-APG-sc, and aPB-APG-sc algorithms exhibit significantly faster convergence performance than the other methods for both lower- and upper-level objectives, although R-APM attains similar outcomes, our PB-APG and PB-APG-sc ensure a more rapid decline than it, as shown in the first subfigure of Figure 1. This is because our methods achieve lower optimal gaps and desired function values of the lower- and upper-level objectives with less execution time. This observation confirms the improved complexity results stated in the theorems above. Although the high exactness of our methods for the lower-level problem leads to larger upper-level objectives, Table 3 in Appendix F.1 shows that our methods are much closer to the optimal value. This is reasonable because the other methods exhibit lower accuracy at the lower-level problem, resulting in larger feasible sets compared to the lower-level optimal solution set X_{opt} . In addition, Figure 1 demonstrates that aPB-APG and aPB-APG-sc outperform PB-APG and PB-APG-sc in terms of convergence rate. This improvement can be attributed to the adaptiveness incorporated in Algorithms 2 and 4.

4.2 Least squares regression problem (LSRP)

The LSRP has the following form:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \frac{\tau}{2} \|\mathbf{x}\|^2 + \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \mathbf{x} \in \arg \min_{\mathbf{z} \in \mathbb{R}^n} \frac{1}{2m} \|\mathbf{A}\mathbf{z} - \mathbf{b}\|^2, \quad (6)$$

where $\tau = 0.02$ regulates the trade-off between ℓ_1 and ℓ_2 norms. We aim to find a sparse solution for the lower-level problem. The upper-level objective is formulated as a composite function, which consists of a τ -strongly convex and τ -smooth component, along with a proximal-friendly non-smooth component [Beck, 2017]. The lower-level objective is a smooth function with a smoothness parameter of $\frac{1}{m} \lambda_{\max}(A^T A)$.

In this experiment, we compare the performances of our methods with a-IRG, BiG-SAM, and Bi-SG. We plot the values of residuals of lower-level objective $G(\mathbf{x}_k) - G^*$ and the upper-level objective over time in Figure 2.

Figure 2 shows that the proposed PB-APG, aPB-APG, PB-APG-sc, and aPB-APG-sc converge faster than the compared methods for both the lower- and upper-level objectives, as well. For the upper-level objective, our methods achieve larger function values than other methods, except BiG-SAM ($\delta = 0.01$). This is because our methods attain higher accuracy for the lower-level objective than other methods. We have similar observations in Section 4.1. Furthermore, Figure 1 also demonstrates that the adaptive mechanism produces staircase-shaped curves for aPB-APG and aPB-APG-sc, which might prevent undesirable fluctuations in PB-APG and PB-APG-sc.

5 Conclusion

This paper proposes a penalization framework that effectively addresses the challenges inherent in simple bilevel optimization problems. By delineating the relationship between approximate solutions of the original problem and its penalized reformulation, we enable the application of specific methods under varying assumptions for the original problem. Under the Hölderian error bound condition, our methods achieve superior complexity results compared to the existing methods. The performance is further improved when the smooth component of the upper-level objective is strongly convex. Additionally, we extend our framework to scenarios involving general nonsmooth objectives. Numerical experiments also validate the effectiveness of our algorithms.

Acknowledgements

This work is partly supported by the National Key R&D Program of China under grant 2023YFA1009300, National Natural Science Foundation of China under grants 12171100 and the Major Program of NFSC (72394360,72394364).

References

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

- Mostafa Amini and Farzad Yousefian. An iterative regularized incremental projected subgradient method for a class of bilevel optimization problems. In *2019 American Control Conference (ACC)*, pages 4069–4074. IEEE, 2019.
- Amir Beck. *First-order methods in optimization*. SIAM, 2017.
- Amir Beck and Shoham Sabach. A first order method for finding minimal norm-like solutions of convex optimization problems. *Mathematical Programming*, 147(1-2):25–46, 2014.
- Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.
- Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018.
- Dimitri Bertsekas, Angelia Nedic, and Asuman Ozdaglar. *Convex analysis and optimization*, volume 1. Athena Scientific, 2003.
- Nicholas Bishop, Long Tran-Thanh, and Enrico Gerding. Optimal learning from verified training data. *Advances in Neural Information Processing Systems*, 33:9520–9529, 2020.
- Jérôme Bolte, Trong Phong Nguyen, Juan Peypouquet, and Bruce W Suter. From error bounds to the complexity of first-order descent methods for convex functions. *Mathematical Programming*, 165:471–507, 2017.
- Digvijay Boob, Qi Deng, and Guanghui Lan. Stochastic first-order methods for convex and nonconvex functional constrained optimization. *Mathematical Programming*, 197(1):215–279, 2023.
- Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.
- James V. Burke and Sien Deng. Weak sharp minima revisited, part ii: application to linear regularity and error bounds. *Mathematical Programming*, 104(2-3):235–261, 2005.
- James V Burke and Michael C Ferris. Weak sharp minima in mathematical programming. *SIAM Journal on Control and Optimization*, 31(5):1340–1359, 1993.
- Alexandre Cabot. Proximal point algorithm controlled by a slowly vanishing term: applications to hierarchical minimization. *SIAM Journal on Optimization*, 15(2):555–572, 2005.
- H. Chen, H. Xu, R. Jiang, et al. Lower-level duality based reformulation and majorization minimization algorithm for hyperparameter optimization. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 784–792. PMLR, 2024.
- L. Chen, J. Xu, and J. Zhang. Bilevel optimization without lower-level strong convexity from the hyper-objective perspective. *arXiv preprint arXiv:2301.00712*, 2023.
- Damek Davis and Dmitriy Drusvyatskiy. Stochastic model-based minimization of weakly convex functions. *SIAM Journal on Optimization*, 29(1):207–239, 2019.
- Stephan Dempe, Nguyen Dinh, Joydeep Dutta, and Tanushree Pandit. Simple bilevel programming and extensions. *Mathematical Programming*, 188:227–253, 2021.
- Lior Doron and Shimrit Shtern. Methodology and first-order algorithms for solving nonsmooth and non-strongly convex bilevel optimization problems. *Mathematical Programming*, 201:521–558, 2023.
- Dmitriy Drusvyatskiy and Adrian S. Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods. *Mathematics of operations research*, 43(3):919–948, 2018a.
- Dmitriy Drusvyatskiy and Adrian S Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods. *Mathematics of Operations Research*, 43(3):919–948, 2018b.
- John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l_1 -ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279, 2008.
- Joydeep Dutta and Tanushree Pandit. Algorithms for simple bilevel programming. *Bilevel Optimization: Advances and Next Challenges*, pages 253–291, 2020.

- Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International conference on machine learning*, pages 1568–1577. PMLR, 2018.
- Michael P Friedlander and Paul Tseng. Exact regularization of convex programs. *SIAM Journal on Optimization*, 18(4):1326–1350, 2008.
- Khanh-Hung Giang-Tran, Nam Ho-Nguyen, and Dabeen Lee. Projection-free methods for solving convex bilevel optimization problems. *arXiv preprint arXiv:2311.09738*, 2023.
- Chengyue Gong, Xingchao Liu, and Qiang Liu. Bi-objective trade-off with dynamic barrier gradient descent. In *International Conference on Neural Information Processing Systems*, pages 29630–29642, 2021.
- Elias S Helou and Lucas EA Simões. ϵ -subgradient algorithms for bilevel convex optimization. *Inverse Problems*, 33(5):055020, 2017.
- F. Huang. On momentum-based gradient methods for bilevel optimization with nonconvex lower-level. *arXiv preprint arXiv:2303.03944*, 2023.
- Ruichen Jiang, Nazanin Abolfazli, Aryan Mokhtari, and Erfan Yazdandoost Hamedani. A conditional gradient-based method for simple bilevel optimization with convex lower-level problem. In *International Conference on Artificial Intelligence and Statistics*, pages 10305–10323. PMLR, 2023.
- Rujun Jiang and Xudong Li. Hölderian error bounds and kurdyka-łojasiewicz inequality for the trust region subproblem. *Mathematics of Operations Research*, 47(4):3025–3050, 2022.
- Harshal D. Kaushik and Farzad Yousefian. A method with convergence rates for optimization problems with variational inequality constraints. *SIAM Journal on Optimization*, 31(3):2171–2198, 2021.
- Matthias Kissel, Martin Gottwald, and Klaus Diepold. Neural network training with safe regularization in the null space of batch activations. In *Artificial Neural Networks and Machine Learning—ICANN 2020: 29th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 15–18, 2020, Proceedings, Part II 29*, pages 217–228. Springer, 2020.
- Puya Latafat, Andreas Themelis, Silvia Villa, and Panagiotis Patrinos. Adabim: An adaptive proximal gradient method for structured convex bilevel optimization. *arXiv preprint arXiv:2305.03559*, 2023.
- Jingwei Liang, Tao Luo, and Carola-Bibiane Schonlieb. Improving fast iterative shrinkage-thresholding algorithm: Faster, smarter, and greedier. *SIAM Journal on Scientific Computing*, 44(3):A1069–A1091, 2022.
- Qihang Lin and Lin Xiao. An adaptive accelerated proximal gradient method and its homotopy continuation for sparse optimization. In *International Conference on Machine Learning*, pages 73–81. PMLR, 2014.
- Zhi-Quan Luo, Jong-Shi Pang, Daniel Ralph, and Shi-Quan Wu. Exact penalization and stationarity conditions of mathematical programs with equilibrium constraints. *Mathematical Programming*, 75(1):19–76, 1996.
- Yura Malitsky. Chambolle-Pock and Tsengs methods: relationship and extension to the bilevel optimization. *arXiv preprint arXiv:1706.02602*, 2017.
- Roey Merchav and Shoham Sabach. Convex bi-level optimization problems with nonsmooth outer objective function. *SIAM Journal on Optimization*, 33(4):3114–3142, 2023.
- Hong Mingyi, Wai Hoi-To, Wang Zhaoran, and Zhuoran Yang. A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic. *arXiv preprint arXiv:2007.05170*, 2020.
- Arkadij Semenovič Nemirovsky and David Borisovich Yudin. *Problem complexity and method efficiency in optimization*. Wiley, 1983.
- Yurii Nesterov. Gradient methods for minimizing composite functions. *Mathematical programming*, 140(1): 125–161, 2013.
- Yurii Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018.
- Jong Shi Pang. Error bounds in mathematical programming. *Mathematical Programming*, 79(1-3):299–332, 1997.
- Nelly Pustelnik and Laurent Condat. Proximity operator of a sum of functions; application to depth map estimation. *IEEE Signal Processing Letters*, 24(12):1827–1831, 2017.

- Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. *Advances in neural information processing systems*, 32, 2019.
- Vincent Roulet and Alexandre d’Aspremont. Sharpness, restart and acceleration. *SIAM Journal on Optimization*, 30(1):262–289, 2020.
- Shoham Sabach and Shimrit Shtern. A first order method for solving convex bilevel optimization problems. *SIAM Journal on Optimization*, 27(2):640–660, 2017.
- Sepideh Samadi, Daniel Burbano, and Farzad Yousefian. Achieving optimal complexity guarantees for a class of bilevel convex optimization problems. *arXiv preprint arXiv:2310.12247*, 2023.
- Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. Truncated back-propagation for bilevel optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1723–1732. PMLR, 2019.
- Han Shen and Tianyi Chen. On penalty-based bilevel gradient descent method. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 30992–31015. PMLR, 23–29 Jul 2023.
- Lingqing Shen, Nam Ho-Nguyen, and Fatma Kılınc-Karzan. An online convex optimization-based framework for convex bilevel optimization. *Mathematical Programming*, 198(2):1519–1582, 2023.
- Naum Zuselevich Shor. *Minimization methods for non-differentiable functions*, volume 3. Springer Science & Business Media, 2012.
- Mikhail Solodov. An explicit descent method for bilevel convex optimization. *Journal of Convex Analysis*, 14(2):227–237, 2007.
- D. Sow, K. Ji, Z. Guan, and et al. A primal-dual approach to bilevel optimization with multiple inner minima. *arXiv preprint arXiv:2203.01123*, 2022.
- Marcin Studniarski and Doug E Ward. Weak sharp minima: characterizations and sufficient conditions. *SIAM Journal on Control and Optimization*, 38(1):219–236, 1999.
- Andre Nikolaevich Tikhonov and V. I. A. K. Arsenin. *Solutions of ill-posed problems*. Wiley, 1977.
- Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. *unpublished manuscript*, 2008.
- Jiali Wang, He Chen, Rujun Jiang, Xudong Li, and Zihao Li. Fast algorithms for stackelberg prediction game with least squares loss. In *International Conference on Machine Learning*, pages 10708–10716. PMLR, 2021.
- Jiali Wang, Wen Huang, Rujun Jiang, Xudong Li, and Alex L Wang. Solving stackelberg prediction game with least squares loss via spherically constrained least squares reformulation. In *International Conference on Machine Learning*, pages 22665–22679. PMLR, 2022.
- Blake E Woodworth and Nati Srebro. Tight complexity bounds for optimizing composite objectives. *Advances in neural information processing systems*, 29, 2016.
- Yangyang Xu. First-order methods for problems with $o(1)$ functional constraints can have almost the same convergence rate as for unconstrained problems. *SIAM Journal on Optimization*, 32(3):1759–1790, 2022.
- Yao-Liang Yu. On decomposing the proximal map. *Advances in neural information processing systems*, 26, 2013.
- Jinshan Zeng, Tim Tsz-Kit Lau, Shaobo Lin, and Yuan Yao. Global convergence of block coordinate descent in deep learning. In *International conference on machine learning*, pages 7313–7323. PMLR, 2019.
- Zirui Zhou and Anthony Man-Cho So. A unified approach to error bounds for structured convex optimization problems. *Mathematical Programming*, 165:689–728, 2017.

A Motivating examples

Many machine learning applications involve a primary objective G , which usually represents the training loss, and a secondary objective F , which can be a regularization term or an auxiliary loss. A common approach for such problems is to optimize G fully and then use F to select the optimal solutions from the ones obtained for G . This is called lexicographic optimization [Kissel et al., 2020, Gong et al., 2021]. Two classes of lexicographic optimization problems are the regularized problem, also known as the ill-posed optimization problem [Amini and Yousefian, 2019, Jiang et al., 2023], and the over-parameterized regression [Jiang et al., 2023], where the upper-level objectives are the regularization terms or loss functions, and the lower-level objectives are the loss functions and the constraint terms. We present some examples of these classes of problems as follows.

Example A.1 (Linear Inverse Problems). Linear inverse problems aim to reconstruct a vector $\mathbf{x} \in \mathbb{R}^n$ from measurements $b \in \mathbb{R}^m$ that satisfy $b = A\mathbf{x} + \rho\varepsilon$, where $A : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a linear mapping, $\varepsilon \in \mathbb{R}^m$ is unknown noise, and $\rho > 0$ is its magnitude. Various optimization techniques can address these problems. We focus on the bilevel formulation, widely adopted in the literature [Beck and Sabach, 2014, Sabach and Shtern, 2017, Dempe et al., 2021, Latafat et al., 2023, Merchav and Sabach, 2023].

The lower-level objective in the bilevel formulation is given by

$$G(\mathbf{x}) = \frac{1}{2m} \|A\mathbf{x} - b\|^2 + I_C(\mathbf{x}), \quad (7)$$

where $I_C(\mathbf{x})$ is the indicator function of a set C that satisfies $I_C(\mathbf{x}) = 0$ if $\mathbf{x} \in C$, and $I_C(\mathbf{x}) = +\infty$ if $\mathbf{x} \notin C$. The set C is a closed, convex set that can be chosen as $C = \mathbb{R}^n$, $C = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} \geq 0\}$, or $C = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_1 \leq \theta\}$ for some $\theta > 0$.

This problem may have multiple minimizer solutions. Hence, a reasonable option is to consider the minimal norm solution problem, i.e., find the optimal solution with the smallest Euclidean norm [Beck and Sabach, 2014, Sabach and Shtern, 2017, Latafat et al., 2023]:

$$F(\mathbf{x}) = \frac{1}{2} \|\mathbf{x}\|^2.$$

We need to solve the simple bilevel optimization problem:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{x}\|^2 \quad \text{s.t.} \quad \mathbf{x} \in \arg \min_{\mathbf{z} \in \mathbb{R}^n} \frac{1}{2m} \|A\mathbf{z} - b\|^2 + I_C(\mathbf{z}).$$

Example A.2 (Sparse Solution of Linear Inverse Problems). Consider the same setting as in Example A.1, but with the additional goal of finding a sparse solution among all the minimizers of the linear inverse problem (7). This can simplify the model and improve computational efficiency. To achieve sparsity, we can use any function that encourages it. One such function is the well-known elastic net regularization [Friedlander and Tseng, 2008, Amini and Yousefian, 2019, Merchav and Sabach, 2023], which is defined as

$$F(\mathbf{x}) = \|\mathbf{x}\|_1 + \frac{\tau}{2} \|\mathbf{x}\|^2,$$

where $\tau > 0$ regulates the trade-off between ℓ_1 and ℓ_2 norms.

This example corresponds to our second experiment in Section 4.2.

Example A.3 (Logistic Regression Problem). The logistic regression problem aims to map the feature vectors \mathbf{a}_i to the target labels b_i . A standard machine learning technique for this problem is to minimize the logistic loss function over the given dataset [Amini and Yousefian, 2019, Gong et al., 2021, Jiang et al., 2023, Latafat et al., 2023, Merchav and Sabach, 2023]. We assume that the dataset consists of a feature matrix $A \in \mathbb{R}^{m \times n}$ and a label vector $b \in \mathbb{R}^m$, with $b_i \in \{-1, 1\}$ for each i . The logistic loss function is defined as

$$g_1(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^m \log(1 + \exp(-\mathbf{a}_i^T \mathbf{x} b_i)). \quad (8)$$

Over-fitting is a common issue when the number of features is large compared to the number of instances m . A possible approach is to regularize the logistic objective function with a specific function or a constraint [Jiang et al., 2023, Merchav and Sabach, 2023]. For instance, we can use $g_2(\mathbf{x}) = I_C(\mathbf{x})$, where $I_C(\mathbf{x})$ is the indicator of the set $C = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_1 \leq \theta\}$, as in Example A.1.

This problem may also have multiple optimal solutions. Hence, a natural extension is to consider the minimal norm solution problem [Gong et al., 2021, Jiang et al., 2023, Latafat et al., 2023], as in Example A.1. This requires solving the following problem:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{x}\|^2 \quad \text{s.t.} \quad \mathbf{x} \in \arg \min_{\mathbf{z} \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^m \log(1 + \exp(-\mathbf{a}_i^T \mathbf{z} b_i)) + I_C(\mathbf{z}).$$

When choosing $C = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_1 \leq \theta\}$ for some $\theta > 0$, it corresponds to our first experiment in Section 4.1.

Example A.4 (Over-parameterized Regression Problem). The linear regression problem aims to find a parameter vector $\mathbf{x} \in \mathbb{R}^n$ that minimizes the training loss $\ell_{\text{tr}}(\mathbf{x})$ over the training dataset \mathcal{D}_{tr} . Without explicit regularization, the over-parameterized regression problem has multiple minima. However, these minima may have different generalization performance. Therefore, we introduce a secondary objective, such as the validation loss over a validation set \mathcal{D}_{val} , to select one of the global minima of the training loss. This results in the following bilevel problem:

$$\min_{\mathbf{x} \in \mathbb{R}^n} F(\mathbf{x}) := \ell_{\text{val}}(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{x} \in \arg \min_{\mathbf{z} \in \mathbb{R}^n} G(\mathbf{z}) := \ell_{\text{tr}}(\mathbf{z}). \quad (9)$$

For instance, we can consider the sparse linear regression problem, where the lower-level objective consists of the training error and a regularization term, namely, $G(\mathbf{x}) = \frac{1}{2} \|A_{\text{tr}}\mathbf{x} - b_{\text{tr}}\|^2 + I_C(\mathbf{x})$. Here, $I_C(\mathbf{x})$ denotes the indicator of a convex set, as in Example A.2. The upper-level objective is the validation error, i.e., $F(\mathbf{x}) = \frac{1}{2} \|A_{\text{val}}\mathbf{x} - b_{\text{val}}\|^2$. The linear regression problem is over-parameterized when the number of features n is larger than the number of data instances in the training set.

B Comparison between simple bilevel optimization methods

Table 1: Summary of simple bilevel optimization algorithms. The abbreviations ‘‘SC,’’ ‘‘C,’’ ‘‘diff,’’ ‘‘comp,’’ ‘‘WS’’ and ‘‘C3’’ represent ‘‘strongly convex,’’ ‘‘convex,’’ ‘‘differentiable,’’ ‘‘composite,’’ ‘‘weak sharpness’’ and ‘‘Convex objective with Convex Compact constraints,’’ respectively. The abbreviation α -HEB refers to Holderian error bound with exponent parameter α . We only include the gradients Lipschitz constant in the complexity result when its relation to the complexity is clear; otherwise, we omit it. Notation l_F is the upper bound of subdifferentials of F , L_{f_1} and L_{g_1} are the Lipschitz constants of ∇f_1 and ∇g_1 , respectively.

Methods	Upper-level	Lower-level	(ϵ_F, ϵ_G) -optimal	Convergence	
	Objective F	Objective G		Solution	Upper-level
MNG [Beck and Sabach, 2014]	SC, diff	C, smooth	(f, ϵ_G)	Asymptotic	$\mathcal{O}(L_{g_1}^2/\epsilon_G^2)$
BiG-SAM [Sabach and Shtern, 2017]	SC, smooth	C, comp	(f, ϵ_G)	Asymptotic	$\mathcal{O}(L_{g_1}/\epsilon_G)$
IR-IG [Amini and Yousefian, 2019]	SC	C3, Finite sum	(f, ϵ_G)	Asymptotic	$\mathcal{O}\left(\frac{1}{\epsilon_G^{2.5-\epsilon}}\right), \epsilon \in (0, 0.5)$
IR-CG [Giang-Tran et al., 2023]	C, smooth	C3, smooth	(ϵ_F, ϵ_G)		$\mathcal{O}\left(\max\{1/\epsilon_F^{\frac{1}{1-p}}, 1/\epsilon_G^2\}\right), p \in (0, 1)$
Tseng’s method [Malitsky, 2017]	C, comp	C, comp	(f, ϵ_G)	Asymptotic	$\mathcal{O}(1/\epsilon_G)$
ITALEX [Doron and Shtern, 2023]	C, comp	C, comp	(ϵ, ϵ^2)		$\mathcal{O}(1/\epsilon^2)$
a-IRG [Kaushik and Yousefian, 2021]	C, Lip	C, Lip	(ϵ_F, ϵ_G)		$\mathcal{O}\left(\max\{1/\epsilon_F^{\frac{1}{1-b}}, 1/\epsilon_G^2\}\right), b \in (0, 0.5)$
CG-BiO [Jiang et al., 2023]	C, smooth	C3, smooth	(ϵ_F, ϵ_G)		$\mathcal{O}(\max\{L_{f_1}/\epsilon_F, L_{g_1}/\epsilon_G\})$
Bi-SG [Merchav and Sabach, 2023]	C, quasi-Lip/comp	C, comp	(ϵ_F, ϵ_G)		$\mathcal{O}\left(\max\{1/\epsilon_F^{\frac{1}{1-a}}, 1/\epsilon_G^2\}\right), a \in (0.5, 1)$
	μ -SC, comp	C, comp	(ϵ_F, ϵ_G)		$\mathcal{O}\left(\max\left\{\frac{\log 1/\epsilon_G}{\mu}, 1/\epsilon_G^2\right\}\right), a \in (0.5, 1)$
R-APM [Samadi et al., 2023]	C, smooth	C, comp, WS	(ϵ, ϵ)		$\mathcal{O}(\sqrt{1/\epsilon})$
Online Framework [Shen et al., 2023]	C, Lip	C3, Lip	(ϵ_F, ϵ_G)		$\mathcal{O}(\max\{1/\epsilon_F^2, 1/\epsilon_G^3\})$
	C, comp	C, comp, α -HEB	$(\epsilon, l_F^{-\beta} \epsilon^\beta)$		$\mathcal{O}\left(\sqrt{\frac{L_{f_1}}{\epsilon}} + \sqrt{\frac{L_{g_1}}{\epsilon^{\max\{\alpha, \beta\}}}}\right), \alpha \geq 1, \beta > 0$
Our method	μ -SC, comp	C, comp, α -HEB	$(\epsilon, l_F^{-\beta} \epsilon^\beta)$		$\mathcal{O}\left(\sqrt{\frac{L_{f_1}}{\mu}} \log \frac{1}{\epsilon} + \sqrt{\frac{L_{g_1}}{\epsilon^{\max\{\alpha-1, \beta-1\}}}} \log \frac{1}{\epsilon}\right), \alpha \geq 1, \beta > 0$
	nonsmooth, Lip	nonsmooth, Lip, α -HEB	$(\epsilon, l_F^{-\beta} \epsilon^\beta)$		$\mathcal{O}\left(\frac{l_F^2}{\epsilon^2} + \frac{l_{g_1}^{\max\{2\alpha, 2\beta\}}}{\epsilon^{\max\{2\alpha, 2\beta\}}}\right), \alpha \geq 1, \beta > 0$

C Examples of functions satisfying the Holderian error bound

We present several examples of functions that satisfy the Holderian error bound Assumption 2.2 and their corresponding exponent parameter α in Table 2. We also provide some clarifications for Table 2 below. The abbreviations ‘‘ $Q \in \mathbb{S}^n$ ’’ and ‘‘ $Q \succ 0$ ’’ stand for ‘‘ Q is a symmetric matrix of order n and a positive definite matrix, respectively. We refer the reader to Pang [1997], Bolte et al. [2017], Zhou and So [2017], Jiang and Li [2022], Doron and Shtern [2023] and the references therein for more examples of functions that satisfy Holderian error bound Assumption 2.2. Furthermore, it is noteworthy that numerous applications in neural networks, such as deep neural networks (DNNs), also comply with this assumption, as discussed in Bolte et al. [2017], Zeng et al. [2019].

⁴According to Table 2 of Doron and Shtern [2023], the parameter α can take values of either 1 or 2. Particularly, when $\alpha = 1$, we have $\rho = 1$; when $\alpha = 2$, we have $\rho = 2/\tau$.

Table 2: Summary of some functions satisfying Hölderian error bound with corresponding exponents.

$G(\mathbf{x})$	Remarks	Name	α
$\max_{i \in [m]} \{\langle \mathbf{a}_i, \mathbf{x} \rangle - b_i\}$	$\mathbf{a}_i \in \mathbb{R}^n, i \in [m], b \in \mathbb{R}^m$	piece-wise maximum	1
$\ \mathbf{x} - \mathbf{x}_0\ _Q = \sqrt{(\mathbf{x} - \mathbf{x}_0)^T Q (\mathbf{x} - \mathbf{x}_0)}$	$Q \in \mathbb{S}^n, Q \succ 0, \mathbf{x}_0 \in \mathbb{R}^n$	Q -norm	1
$\ \mathbf{x} - \mathbf{x}_0\ _p$	$\mathbf{x}_0 \in \mathbb{R}^n, p \geq 1$	ℓ_p -norm	1
$\ x\ _1 + \frac{\tau}{2} \ x\ ^2$	$\tau > 0$	Elastic net	1 or 2^4
$\ A\mathbf{x} - b\ ^2$	$A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$	Least squares	2
$\frac{1}{m} \sum_{i=1}^m \log(1 + \exp(-\mathbf{a}_i^T \mathbf{x} b_i))$	$\mathbf{a}_i \in \mathbb{R}^n, i \in [m], b \in \mathbb{R}^m, A \in \mathbb{R}^{m \times n}$	Logistic loss	2
$\eta(\mathbf{x}) + \frac{\sigma}{2} \ \mathbf{x}\ ^2$	η convex, $\sigma > 0$	Strongly-convex	2

D Supplementary results

D.1 Adaptive version of PB-APG method with strong convexity assumption

Algorithm 4 Adaptive PB-APG-sc method (aPB-APG-sc)

- 1: **Input:** $\mathbf{x}_{-1} = \mathbf{x}_0 \in \mathbb{R}^n, \gamma_0 = \gamma_1 > 0, L_{f_1}, L_{g_1}, \nu > 1, \eta > 1, \epsilon_0 > 0.$
- 2: **for** $k \geq 0$ **do**
- 3: $\phi_k(\mathbf{x}) = f_1(\mathbf{x}) + \gamma_k g_1(\mathbf{x})$
- 4: $\psi_k(\mathbf{x}) = f_2(\mathbf{x}) + \gamma_k g_2(\mathbf{x})$
- 5: Invoke $\mathbf{x}_k = \text{PB-APG-sc}(\phi_k, \psi_k, \mu, L_{f_1}, L_{g_1}, \mathbf{x}_{k-1}, \epsilon_k)$
- 6: $\epsilon_{k+1} = \frac{1}{\eta} \epsilon_k$
- 7: $\gamma_{k+1} = \nu \gamma_k$
- 8: **end for**

Similar to Algorithm 2, we have the following convergence results of Algorithm 4.

Theorem D.1. *Suppose that Assumptions 2.1, 2.2, 3.1, 3.2, and 3.7 hold. Let $\epsilon_0 > 0$ be given.*

- *When $\alpha > 1$, set $\nu > \eta^{\alpha-1}$, $N = \lceil \log_{\eta^{1-\alpha\nu}}(\rho L_F^\alpha (\alpha-1)^{\alpha-1} \alpha^{-\alpha} \epsilon_0^{1-\alpha} / \gamma_0) \rceil_+$ and $\gamma_k^* = \rho L_F^\alpha (\alpha-1)^{\alpha-1} \alpha^{-\alpha} \epsilon_0^{1-\alpha} \eta^{k(\alpha-1)}$;*
- *When $\alpha = 1$, set $\nu > 1$, $N = \lceil \log_\nu(\rho L_F / \gamma_0) \rceil_+$ and $\gamma_k^* = \rho L_F.$*

Then, for any $k \geq N$, Algorithm 2 generates an $(\frac{\epsilon_0}{\eta^k}, \frac{2\epsilon_0}{\eta^k(\gamma_0 \nu^k - \gamma_k^)})$ -optimal solution of problem (P) after at most K iterations, where K satisfies*

$$K = \mathcal{O} \left(\sqrt{\frac{L_{f_1}}{\mu}} \log \frac{\eta^k}{\epsilon_0} + \sqrt{\frac{\nu^k L_F^{\max\{\alpha, \beta\}} L_{g_1}}{\epsilon^{\max\{\alpha-1, \beta-1\}}}} \log \frac{\eta^k}{\epsilon_0} \right).$$

The proof is similar to the proof of Theorem 3.5 in Appendix E.8. So we omit it here.

E Proofs of main results

In this section, we propose the proofs of our main convergence results in this paper.

E.1 Proof of Lemma 2.3

Proof. Since X_{opt} is closed and convex [Beck and Sabach, 2014], the projection of any $\mathbf{x} \in \mathbb{R}^n$ onto X_{opt} , denoted as $\bar{\mathbf{x}}$, exists and is unique. Furthermore, it holds that $\text{dist}(\mathbf{x}, X_{\text{opt}}) = \|\mathbf{x} - \bar{\mathbf{x}}\|.$

Then, by Assumption 2.1, we have

$$F(\mathbf{x}) - F(\bar{\mathbf{x}}) \geq -\xi^T (\mathbf{x} - \bar{\mathbf{x}}) \geq -\|\xi\| \|\mathbf{x} - \bar{\mathbf{x}}\| \geq -l_F \|\mathbf{x} - \bar{\mathbf{x}}\|, \forall \xi \in \partial F(\bar{\mathbf{x}}). \quad (10)$$

Choosing $\gamma^* = \rho l_F^\alpha (\alpha - 1)^{\alpha-1} \alpha^{-\alpha} \epsilon^{1-\alpha}$, it follows that

$$\begin{aligned}
F(\mathbf{x}) - F(\bar{\mathbf{x}}) + \gamma^* p(\mathbf{x}) &\stackrel{(10)}{\geq} -l_F \|\mathbf{x} - \bar{\mathbf{x}}\| + \gamma^* p(\mathbf{x}) \\
&\stackrel{(a)}{\geq} -l_F \|\mathbf{x} - \bar{\mathbf{x}}\| + \frac{\gamma^*}{\rho} \|\mathbf{x} - \bar{\mathbf{x}}\|^\alpha \\
&\geq \min_{\mathbf{z} \geq 0} -l_F \mathbf{z} + \frac{\gamma^*}{\rho} \mathbf{z}^\alpha \\
&\stackrel{(b)}{=} -\epsilon,
\end{aligned} \tag{11}$$

where (a) follows from the Hölderian error bound assumption of $p(\mathbf{x})$, and (b) is from the fact that $\mathbf{y} = -l_F \mathbf{z} + \frac{\gamma^*}{\rho} \mathbf{z}^\alpha$ attains its minimum at $\mathbf{z}^* = \left(\frac{\rho l_F}{\alpha \gamma^*}\right)^{\frac{1}{\alpha-1}}$.

Since $\bar{\mathbf{x}} \in X_{\text{opt}}$ is feasible for problem (P), we have $F(\bar{\mathbf{x}}) \geq F^*$. This along with (11) indicates

$$F(\mathbf{x}) + \gamma p(\mathbf{x}) - F^* \geq F(\mathbf{x}) + \gamma^* p(\mathbf{x}) - F(\bar{\mathbf{x}}) \geq -\epsilon, \quad \forall \mathbf{x} \in \mathbb{R}^d \text{ and } \gamma \geq \gamma^*. \tag{12}$$

Let \mathbf{x}^* be an optimal solution of (P) so that $F(\mathbf{x}^*) = F^*$. In addition, since $\mathbf{x}^* \in X_{\text{opt}}$, we have $p(\mathbf{x}^*) = 0$. Combine these results with (12), we have

$$F(\mathbf{x}^*) + \gamma p(\mathbf{x}^*) = F^* \stackrel{(12)}{\leq} F(\mathbf{x}) + \gamma p(\mathbf{x}) + \epsilon, \quad \forall \mathbf{x} \in \mathbb{R}^d \text{ and } \gamma \geq \gamma^*. \tag{13}$$

This demonstrates that an optimal solution of (P) is an ϵ -optimal solution for (P_γ) . \square

E.2 Proof of Lemma 2.4

Proof. The proof is motivated by Theorem 1 in Luo et al. [1996]. Denote $\mathbf{x}^*, \mathbf{x}_\gamma^*$ as optimal solutions of problem (P) and (P_γ) , respectively.

For any $\mathbf{x} \in \mathbb{R}^n$, let $\bar{\mathbf{x}}$ be the projection of \mathbf{x} onto X_{opt} . Then $\bar{\mathbf{x}}$ is a feasible solution of (P) and $F(\bar{\mathbf{x}}) \geq F(\mathbf{x}^*)$ holds. Then we have

$$\begin{aligned}
F(\mathbf{x}) + \gamma p(\mathbf{x}) &= F(\bar{\mathbf{x}}) + F(\mathbf{x}) - F(\bar{\mathbf{x}}) + \gamma p(\mathbf{x}) \\
&\geq F(\mathbf{x}^*) + F(\mathbf{x}) - F(\bar{\mathbf{x}}) + \gamma p(\mathbf{x}) \\
&\stackrel{(a)}{\geq} F(\mathbf{x}^*) - l_F \|\mathbf{x} - \bar{\mathbf{x}}\| + \frac{\gamma}{\rho} \|\mathbf{x} - \bar{\mathbf{x}}\| \\
&= F(\mathbf{x}^*) + \left(\frac{\gamma}{\rho} - l_F\right) \|\mathbf{x} - \bar{\mathbf{x}}\| \\
&\stackrel{(b)}{\geq} F(\mathbf{x}^*) = F(\mathbf{x}^*) + \gamma p(\mathbf{x}^*),
\end{aligned} \tag{14}$$

where (a) follows from (10) and the Hölderian error bound assumption of $p(\mathbf{x})$, and (b) follows from $\gamma \geq \rho l_F$. Therefore, we conclude that \mathbf{x}^* is an optimal solution of (P_γ) .

For the converse part, let $\bar{\mathbf{x}}_\gamma^*$ be the projection of \mathbf{x}_γ^* onto X_{opt} . Then $\bar{\mathbf{x}}_\gamma^*$ is a feasible solution of (P). Therefore, it holds that $F(\bar{\mathbf{x}}_\gamma^*) \geq F(\mathbf{x}^*)$. Similarly, we have

$$\begin{aligned}
F(\mathbf{x}^*) &= F(\mathbf{x}^*) + \gamma p(\mathbf{x}^*) \\
&\geq F(\mathbf{x}_\gamma^*) + \gamma p(\mathbf{x}_\gamma^*) \\
&= F(\mathbf{x}_\gamma^*) - F(\mathbf{x}^*) + F(\mathbf{x}^*) + \gamma p(\mathbf{x}_\gamma^*) \\
&\geq F(\mathbf{x}^*) + F(\mathbf{x}_\gamma^*) - F(\bar{\mathbf{x}}_\gamma^*) + \gamma p(\mathbf{x}_\gamma^*) \\
&\stackrel{(c)}{\geq} F(\mathbf{x}^*) - l_F \|\mathbf{x}_\gamma^* - \bar{\mathbf{x}}_\gamma^*\| + \frac{\gamma}{\rho} \|\mathbf{x}_\gamma^* - \bar{\mathbf{x}}_\gamma^*\| \\
&\geq F(\mathbf{x}^*) + \left(\frac{\gamma}{\rho} - l_F\right) \|\mathbf{x}_\gamma^* - \bar{\mathbf{x}}_\gamma^*\| \\
&\geq F(\mathbf{x}^*),
\end{aligned} \tag{15}$$

where the inequality (c) follows from (10) and the Hölderian error bound assumption of $p(\mathbf{x})$.

Therefore, all inequalities in (15) become equalities. We deduce that $\|\mathbf{x}_\gamma^* - \bar{\mathbf{x}}_\gamma^*\| = 0$ if $\gamma > \rho l_F$, implying that \mathbf{x}_γ^* is in X_{opt} , i.e., $p(\mathbf{x}_\gamma^*) = 0$. Furthermore, as the first inequality of (15) becomes an equality, we obtain

$$F(\mathbf{x}^*) = F(\mathbf{x}_\gamma^*) + \gamma p(\mathbf{x}_\gamma^*) = F(\mathbf{x}_\gamma^*).$$

Therefore, \mathbf{x}_γ^* is also an optimal solution of (P). \square

E.3 Proof of Theorem 2.5

Proof. Denote \mathbf{x}^* , \mathbf{x}_γ^* as optimal solutions of problem (P) and (P_γ) , respectively.

- **Case of $\alpha > 1$.** Since $\tilde{\mathbf{x}}_\gamma^*$ is an ϵ -optimal solution of (P_γ) , we have

$$F(\tilde{\mathbf{x}}_\gamma^*) + \gamma p(\tilde{\mathbf{x}}_\gamma^*) \leq F(\mathbf{x}) + \gamma p(\mathbf{x}) + \epsilon, \quad \forall \mathbf{x} \in \mathbb{R}^n. \quad (16)$$

Note that the arguments in the proof of Lemma 2.3 still hold. Substituting $\mathbf{x} = \mathbf{x}^*$ into (16) and utilizing $p(\mathbf{x}^*) = 0$, we have

$$F(\tilde{\mathbf{x}}_\gamma^*) + \gamma p(\tilde{\mathbf{x}}_\gamma^*) \leq F(\mathbf{x}^*) + \epsilon = F(\mathbf{x}^*) + \gamma^* p(\mathbf{x}^*) + \epsilon \leq F(\tilde{\mathbf{x}}_\gamma^*) + \gamma^* p(\tilde{\mathbf{x}}_\gamma^*) + 2\epsilon,$$

where the last inequality follows from setting $\mathbf{x} = \tilde{\mathbf{x}}_\gamma^*$ in (13). Then, it holds that

$$p(\tilde{\mathbf{x}}_\gamma^*) \leq \frac{2\epsilon}{\gamma - \gamma^*} = \frac{2\epsilon}{2l_F^\beta \epsilon^{1-\beta}} = l_F^{-\beta} \epsilon^\beta. \quad (17)$$

By setting $\mathbf{x} = \mathbf{x}^*$ in (16), we have

$$F(\tilde{\mathbf{x}}_\gamma^*) - F(\mathbf{x}^*) \leq \gamma(p(\mathbf{x}^*) - p(\tilde{\mathbf{x}}_\gamma^*)) + \epsilon.$$

Using the fact that $p(\mathbf{x}^*) = 0 \leq p(\tilde{\mathbf{x}}_\gamma^*)$, we have

$$F(\tilde{\mathbf{x}}_\gamma^*) - F(\mathbf{x}^*) \leq \epsilon. \quad (18)$$

Combing (18) with (17), we conclude that $\tilde{\mathbf{x}}_\gamma^*$ is an $(\epsilon, l_F^{-\beta} \epsilon^\beta)$ -optimal solution of (P).

- **Case of $\alpha = 1$.** Since $\tilde{\mathbf{x}}_\gamma^*$ is an ϵ -optimal solution of (P_γ) , we have

$$F(\tilde{\mathbf{x}}_\gamma^*) + \gamma p(\tilde{\mathbf{x}}_\gamma^*) \leq F(\mathbf{x}_\gamma^*) + \gamma p(\mathbf{x}_\gamma^*) + \epsilon. \quad (19)$$

On the one hand, as $\gamma = \gamma^* + l_F^\beta \epsilon^{1-\beta} > \gamma^*$, by Lemma 2.4, \mathbf{x}_γ^* is an optimal solution of (P). On the other hand, since $\gamma \geq \gamma^*$, according to Lemma 2.4, \mathbf{x}^* is also an optimal solution of (P_γ) . Therefore, $p(\mathbf{x}^*) = 0$ and $p(\mathbf{x}_\gamma^*) = 0$, it holds that

$$\begin{aligned} F(\mathbf{x}^*) &\leq F(\tilde{\mathbf{x}}_\gamma^*) + \gamma p(\tilde{\mathbf{x}}_\gamma^*) \\ &\stackrel{(19)}{\leq} F(\mathbf{x}_\gamma^*) + \gamma p(\mathbf{x}_\gamma^*) + \epsilon \\ &= F(\mathbf{x}_\gamma^*) + \gamma^* p(\mathbf{x}_\gamma^*) + \epsilon \\ &= F(\mathbf{x}^*) + \gamma^* p(\mathbf{x}^*) + \epsilon \\ &\leq F(\tilde{\mathbf{x}}_\gamma^*) + \gamma^* p(\tilde{\mathbf{x}}_\gamma^*) + \epsilon, \end{aligned} \quad (20)$$

where the first inequality follows from the fact that \mathbf{x}^* is an optimal solution of (P_γ) , and the last inequality follows from the optimality of \mathbf{x}^* to (P_γ) when $\gamma \geq \gamma^*$.

The second inequality of (20) and $p(\tilde{\mathbf{x}}_\gamma^*) \geq 0$ imply that

$$F(\tilde{\mathbf{x}}_\gamma^*) \leq F(\mathbf{x}_\gamma^*) + \gamma p(\mathbf{x}_\gamma^*) + \epsilon = F(\mathbf{x}^*) + \gamma p(\mathbf{x}^*) + \epsilon \leq F(\mathbf{x}^*) + \epsilon.$$

That is, it holds that

$$F(\tilde{\mathbf{x}}_\gamma^*) \leq F(\mathbf{x}^*) + \epsilon. \quad (21)$$

In addition, from (20), we have $F(\tilde{\mathbf{x}}_\gamma^*) + \gamma p(\tilde{\mathbf{x}}_\gamma^*) \leq F(\tilde{\mathbf{x}}_\gamma^*) + \gamma^* p(\tilde{\mathbf{x}}_\gamma^*) + \epsilon$, which implies that

$$p(\tilde{\mathbf{x}}_\gamma^*) \leq \frac{\epsilon}{\gamma - \gamma^*} = \frac{\epsilon}{l_F^\beta \epsilon^{1-\beta}} = l_F^{-\beta} \epsilon^\beta. \quad (22)$$

This result along with (21) demonstrate that $\tilde{\mathbf{x}}_\gamma^*$ is an $(\epsilon, l_F^{-\beta} \epsilon^\beta)$ -optimal solution of (P). □

E.4 Proof of Theorem 2.6

Proof. Let $\hat{\mathbf{x}}_\gamma^*$ be the projection of $\tilde{\mathbf{x}}_\gamma^*$ on X_{opt} , we have $\|\tilde{\mathbf{x}}_\gamma^* - \hat{\mathbf{x}}_\gamma^*\| = \text{dist}(\tilde{\mathbf{x}}_\gamma^*, X_{\text{opt}})$.

By Assumption 2.2, the following inequality holds,

$$\|\tilde{\mathbf{x}}_\gamma^* - \hat{\mathbf{x}}_\gamma^*\|^\alpha \leq \rho p(\tilde{\mathbf{x}}_\gamma^*) \stackrel{(a)}{\leq} \rho l_F^{-\beta} \epsilon^\beta \implies \|\tilde{\mathbf{x}}_\gamma^* - \hat{\mathbf{x}}_\gamma^*\| \leq \left(\rho l_F^{-\beta} \epsilon^\beta \right)^{\frac{1}{\alpha}}, \quad (23)$$

where (a) follows from (17) when $\alpha > 1$ or from (22) when $\alpha = 1$.

By Assumption 2.1, we have

$$F(\tilde{\mathbf{x}}_\gamma^*) - F^* \geq F(\tilde{\mathbf{x}}_\gamma^*) - F(\hat{\mathbf{x}}_\gamma^*) \stackrel{(10)}{\geq} -l_F \|\tilde{\mathbf{x}}_\gamma^* - \hat{\mathbf{x}}_\gamma^*\| \geq -l_F \left(\rho l_F^{-\beta} \epsilon^\beta \right)^{\frac{1}{\alpha}},$$

where the first inequality follows from $F(\hat{\mathbf{x}}_\gamma^*) \geq F^*$ and $\hat{\mathbf{x}}_\gamma^* \in X_{\text{opt}}$. □

E.5 Proof of Theorem 2.7

Proof. For any $\mathbf{x} \in \text{dom}(F)$, let $\bar{\mathbf{x}}$ be the projection of \mathbf{x} onto X_{opt} , where the existence and uniqueness of $\bar{\mathbf{x}}$ follows from that X_{opt} is closed and convex. Since F is l -Lipschitz continuous, similar to (10), we have

$$F(\mathbf{x}) - F(\bar{\mathbf{x}}) \geq -l\|\mathbf{x} - \bar{\mathbf{x}}\|, \quad \forall \xi \in \partial F(\bar{\mathbf{x}}). \quad (24)$$

Therefore, all the requirements of (10) in equations (11), (14) and (15) can be replaced by (24). This implies that Lemmas 2.3 and 2.4 also hold for the global solutions of problems (P) and (P $_{\gamma}$) when F is non-convex. Then, the final result follows a similar pattern to Theorem 2.5. Here we omit it. \square

E.6 Proof of Theorem 2.8

Proof. Let $\bar{\mathbf{x}}_{\gamma}^*$ be the projection of \mathbf{x}_{γ}^* onto X_{opt} and $\hat{\mathbf{x}}_{\gamma}^* = c\mathbf{x}_{\gamma}^* + (1-c)\bar{\mathbf{x}}_{\gamma}^*$ with $c = \min\{1, 1 - \frac{r}{\|\mathbf{x}_{\gamma}^* - \bar{\mathbf{x}}_{\gamma}^*\|}\}$, which implies that $\hat{\mathbf{x}}_{\gamma}^* \in \mathcal{B}(\mathbf{x}_{\gamma}^*, r)$. Then, we have

$$F(\mathbf{x}_{\gamma}^*) + \gamma p(\mathbf{x}_{\gamma}^*) \leq F(\hat{\mathbf{x}}_{\gamma}^*) + \gamma p(\hat{\mathbf{x}}_{\gamma}^*) \stackrel{(i)}{\leq} F(\hat{\mathbf{x}}_{\gamma}^*) + \gamma(cp(\mathbf{x}_{\gamma}^*) + (1-c)p(\bar{\mathbf{x}}_{\gamma}^*)) = F(\hat{\mathbf{x}}_{\gamma}^*) + \gamma cp(\mathbf{x}_{\gamma}^*), \quad (25)$$

where inequality (i) follows from the convexity of $p(\mathbf{x})$.

Inequality (25) demonstrates that

$$\gamma(1-c)p(\mathbf{x}_{\gamma}^*) \leq F(\hat{\mathbf{x}}_{\gamma}^*) - F(\mathbf{x}_{\gamma}^*) \leq l\|\hat{\mathbf{x}}_{\gamma}^* - \mathbf{x}_{\gamma}^*\| = l(1-c)\|\mathbf{x}_{\gamma}^* - \bar{\mathbf{x}}_{\gamma}^*\| \leq l(1-c)(\rho p(\mathbf{x}_{\gamma}^*))^{\frac{1}{\alpha}},$$

where the second inequality follows from the l -Lipschitz continuity of F on $\mathcal{B}(\mathbf{x}_{\gamma}^*, r)$. Therefore, it holds that

$$\gamma p(\mathbf{x}_{\gamma}^*) \leq l(\rho p(\mathbf{x}_{\gamma}^*))^{\frac{1}{\alpha}}. \quad (26)$$

- **Case of $\alpha > 1$.** By (26), we have $p(\mathbf{x}_{\gamma}^*) \leq (\frac{\rho l^{\alpha}}{\gamma})^{\frac{1}{\alpha-1}}$, which demonstrates that $p(\mathbf{x}_{\gamma}^*) \leq \epsilon$ if $\gamma \geq (\frac{\rho l^{\alpha}}{\epsilon^{\alpha-1}})^{\frac{1}{\alpha}}$.

Then, for any $\mathbf{x}_{\gamma} \in \mathcal{B}(\mathbf{x}_{\gamma}^*, r)$ that also satisfies $p(\mathbf{x}_{\gamma}) \leq p(\mathbf{x}_{\gamma}^*) \leq \epsilon$, we have

$$F(\mathbf{x}_{\gamma}^*) + \gamma p(\mathbf{x}_{\gamma}^*) \leq F(\mathbf{x}_{\gamma}) + \gamma p(\mathbf{x}_{\gamma}), \quad (27)$$

which implies that $F(\mathbf{x}_{\gamma}^*) - F(\mathbf{x}_{\gamma}) \leq \gamma(p(\mathbf{x}_{\gamma}) - p(\mathbf{x}_{\gamma}^*)) \leq 0$. The desired result follows.

- **Case of $\alpha = 1$.** By (26), we have $p(\mathbf{x}_{\gamma}^*) = 0$ if $\gamma > \rho l$. Therefore, for any $\mathbf{x}_{\gamma} \in \mathcal{B}(\mathbf{x}_{\gamma}^*, r) \cap X_{\text{opt}}$, by the definition of \mathbf{x}_{γ}^* , it holds that

$$F(\mathbf{x}_{\gamma}^*) + \gamma p(\mathbf{x}_{\gamma}^*) \leq F(\mathbf{x}_{\gamma}) + \gamma p(\mathbf{x}_{\gamma}),$$

which demonstrates that $F(\mathbf{x}_{\gamma}^*) \leq F(\mathbf{x}_{\gamma})$. The desired result follows. \square

E.7 Proof of Theorem 3.3

Proof. From [Beck, 2017, Theorem 10.34], the objective value after K iterations can be bounded by

$$\Phi_{\gamma}(\mathbf{x}_K) - \Phi_{\gamma}^* \leq \frac{2L_{\gamma}\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{(K+1)^2},$$

where $L_{\gamma} = L_{f_1} + \gamma L_{g_1}$.

Combining this with our stopping criterion, we find that after K iterations,

$$\Phi_{\gamma}(\mathbf{x}_K) - \Phi_{\gamma}^* \leq \epsilon.$$

This indicates that we obtain an ϵ -optimal solution to problem (P $_{\gamma}$). The value of K satisfies:

$$K = \sqrt{\frac{2(L_{f_1} + \gamma L_{g_1})}{\epsilon}} R - 1.$$

Specifically, we analyze the value of K in various scenarios in the form of $\mathcal{O}(\cdot)$.

- **Case of $\alpha > 1$.** In this case, $\gamma = \gamma^* + 2l_F^{\beta}\epsilon^{1-\beta}$ comprises two components: γ^* and $2l_F^{\beta}\epsilon^{1-\beta}$. Therefore, it is natural to discuss which of these two components plays the dominant role in the complexity results. First, we write K in the form:

$$K = \sqrt{\frac{2(L_{f_1} + (\rho l_F^{\alpha}(\alpha-1)^{\alpha-1}\alpha^{-\alpha}\epsilon^{1-\alpha} + 2l_F^{\beta}\epsilon^{1-\beta})L_{g_1})}{\epsilon}} R - 1.$$

If $\beta < \alpha$, the dominating term in γ is $\gamma^* = \rho l_F^\alpha (\alpha - 1)^{\alpha-1} \alpha^{-\alpha} \epsilon^{1-\alpha}$. Then, the number of iterations is

$$K = \mathcal{O} \left(\sqrt{\frac{L_{f_1} + l_F^\alpha \epsilon^{1-\alpha} L_{g_1}}{\epsilon}} \right) = \mathcal{O} \left(\sqrt{\frac{L_{f_1}}{\epsilon}} + \sqrt{\frac{l_F^\alpha L_{g_1}}{\epsilon^\alpha}} \right).$$

If $\beta = \alpha$, we have $\gamma = (\rho(\alpha - 1)^{\alpha-1} \alpha^{-\alpha} + 2) l_F^\alpha \epsilon^{1-\alpha}$. Then, the number of iterations is

$$K = \mathcal{O} \left(\sqrt{\frac{L_{f_1} + l_F^\alpha \epsilon^{1-\alpha} L_{g_1}}{\epsilon}} \right) = \mathcal{O} \left(\sqrt{\frac{L_{f_1}}{\epsilon}} + \sqrt{\frac{l_F^\alpha L_{g_1}}{\epsilon^\alpha}} \right).$$

If $\beta > \alpha$, the dominating term in γ is $2l_F^\beta \epsilon^{1-\beta}$. Then, the number of iterations is

$$K = \mathcal{O} \left(\sqrt{\frac{L_{f_1} + 2l_F^\beta \epsilon^{1-\beta} L_{g_1}}{\epsilon}} \right) = \mathcal{O} \left(\sqrt{\frac{L_{f_1}}{\epsilon}} + \sqrt{\frac{l_F^\beta L_{g_1}}{\epsilon^\beta}} \right).$$

- **Case of $\alpha = 1$.** In this case, $\gamma = \gamma^* + l_F^\beta \epsilon^{1-\beta}$, where $\gamma^* = \rho l_F$. Similarly, we explore which of these two elements plays a more significant role.

If $\beta < 1$, the dominating term in γ is γ^* . Then, the number of iterations is

$$K = \mathcal{O} \left(\sqrt{\frac{L_{f_1} + \rho l_F L_{g_1}}{\epsilon}} \right) = \mathcal{O} \left(\sqrt{\frac{L_{f_1}}{\epsilon}} + \sqrt{\frac{l_F L_{g_1}}{\epsilon}} \right).$$

If $\beta = 1$, we have $\gamma = (\rho + 1) l_F \epsilon^{1-\alpha}$. Then the number of iterations is

$$K = \mathcal{O} \left(\sqrt{\frac{L_{f_1} + (\rho + 1) l_F L_{g_1}}{\epsilon}} \right) = \mathcal{O} \left(\sqrt{\frac{L_{f_1}}{\epsilon}} + \sqrt{\frac{l_F L_{g_1}}{\epsilon}} \right).$$

If $\beta > 1$, the dominating term in γ is $l_F^\beta \epsilon^{1-\beta}$. Then, the number of iterations is

$$K = \mathcal{O} \left(\sqrt{\frac{L_{f_1} + l_F^\beta \epsilon^{1-\beta} L_{g_1}}{\epsilon}} \right) = \mathcal{O} \left(\sqrt{\frac{L_{f_1}}{\epsilon}} + \sqrt{\frac{l_F^\beta L_{g_1}}{\epsilon^\beta}} \right).$$

Combining the above results, we conclude that

$$K = \mathcal{O} \left(\sqrt{\frac{L_{f_1}}{\epsilon}} + \sqrt{\frac{l_F^{\max\{\alpha, \beta\}} L_{g_1}}{\epsilon^{\max\{\alpha, \beta\}}}} \right).$$

□

E.8 Proof of Theorem 3.5

Proof. In this proof, we denote Φ_k^* as the optimal value of problem (P_γ) when $\gamma = \gamma_k$, and \mathbf{x}_k as the output of PB-APG (Algorithm 1) in the k -th iteration.

- **Case of $\alpha > 1$.** Suppose that N is the smallest nonnegative integer such that $\gamma_N \geq \gamma_N^* := \rho l_F^\alpha (\alpha - 1)^{\alpha-1} \alpha^{-\alpha} \epsilon_N^{1-\alpha}$. In this case, we have

$$\gamma_N = \gamma_0 \nu^N \geq \rho l_F^\alpha (\alpha - 1)^{\alpha-1} \alpha^{-\alpha} \epsilon_N^{1-\alpha} = \rho l_F^\alpha (\alpha - 1)^{\alpha-1} \alpha^{-\alpha} \epsilon_0^{1-\alpha} (1/\eta)^{(1-\alpha)N}, \quad (28)$$

which is equivalent to

$$\gamma_0 (\nu \eta^{1-\alpha})^N \geq \rho l_F^\alpha (\alpha - 1)^{\alpha-1} \alpha^{-\alpha} \epsilon_0^{1-\alpha}. \quad (29)$$

From (29), after at most $N := \lceil \log_{\nu \eta^{1-\alpha}} \left(\frac{\rho l_F^\alpha (\alpha - 1)^{\alpha-1} \alpha^{-\alpha} \epsilon_0^{1-\alpha}}{\gamma_0} \right) \rceil_+$ iterations, (28) holds.

Since $x_N = \text{PB-APG}(\phi_N, \psi_N, L_{f_1}, L_{g_1}, \mathbf{x}_{N-1}, \epsilon_N)$, we have

$$\Phi_N(\mathbf{x}_N) - \Phi_N^* \leq \epsilon_N, \quad \gamma_N \geq \gamma_N^*,$$

which shows that \mathbf{x}_N is an ϵ_N -optimal solution of (P_γ) with $\gamma = \gamma_N$. From the proof in Theorem 2.5 (see inequalities (17) and (18) in Appendix E.3), \mathbf{x}_N is also an $(\frac{\epsilon_0}{\eta^N}, \frac{2\epsilon_0}{\eta^N(\gamma_0\nu^N - \gamma_N^*)})$ -optimal solution of problem (P).

Furthermore, note that for any iteration $k \geq N$, inequality (29) always holds, which means that the following statement holds for any $k \geq N$:

$$\Phi_k(\mathbf{x}_k) - \Phi_k^* \leq \epsilon_k, \quad \gamma_k \geq \gamma_k^*. \quad (30)$$

Let I_k be the number of iterations of PB-APG required to satisfy (30) at the k -th iteration of aPB-APG. Then, for any $k \geq N$, the total number of iterations is

$$K = I_0 + I_1 + \dots + I_k.$$

From [Beck, 2017, Theorem 10.34], the number of iterations in i -th inner loop satisfies:

$$I_i = \sqrt{\frac{2(L_{f_1} + \gamma_i L_{g_1})}{\epsilon_i}} \|\mathbf{x}_{i-1} - \mathbf{x}_i^*\| - 1,$$

where \mathbf{x}_i^* is the optimal solution in i -th inner loop. Then we have that

$$\begin{aligned} K &= \sum_{i=0}^k \sqrt{\frac{2(L_{f_1} + \gamma_i L_{g_1})}{\epsilon_i}} \|\mathbf{x}_{i-1} - \mathbf{x}_i^*\| - k \\ &\leq \sum_{i=0}^k \sqrt{\frac{2(L_{f_1} + \gamma_k L_{g_1})}{\epsilon_i}} R - k \\ &= \frac{\eta^{\frac{k}{2}} - 1}{\eta^{\frac{1}{2}} - 1} \sqrt{\frac{2(L_{f_1} + \gamma_0 \nu^k L_{g_1})}{\epsilon_0}} - k. \end{aligned}$$

For simplicity, we can also use $\mathcal{O}(\cdot)$ to show the value of K .

$$\begin{aligned} K &= \mathcal{O}\left(\sqrt{\frac{L_{f_1} + \gamma_0 L_{g_1}}{\epsilon_0}}\right) + \dots + \mathcal{O}\left(\sqrt{\frac{L_{f_1} + \gamma_k L_{g_1}}{\epsilon_k}}\right) \\ &\leq \mathcal{O}\left(\sqrt{\frac{L_{f_1} + \gamma_k L_{g_1}}{\epsilon_0}}\right) + \dots + \mathcal{O}\left(\sqrt{\frac{L_{f_1} + \gamma_k L_{g_1}}{\epsilon_k}}\right) \\ &= \mathcal{O}\left(\sqrt{\frac{L_{f_1} + \gamma_k L_{g_1}}{\epsilon_k}} \left(1 + \sqrt{1/\eta} + \sqrt{1/\eta^2} + \dots + \sqrt{1/\eta^k}\right)\right) \\ &= \mathcal{O}\left(\sqrt{\frac{L_{f_1} + \gamma_k L_{g_1}}{\epsilon_k}}\right) \\ &= \mathcal{O}\left(\sqrt{\frac{L_{f_1} \eta^k}{\epsilon_0}} + \sqrt{\frac{L_{g_1} \gamma_0 (\eta \nu)^k}{\epsilon_0}}\right). \end{aligned}$$

- **Case of $\alpha = 1$.** Suppose that after N updates, we have $\gamma_N \geq \rho l_F$, i.e.,

$$\gamma_0 \nu^N \geq \rho l_F. \quad (31)$$

This demonstrates that after for all $k \geq N := \log_\nu\left(\frac{\rho l_F}{\gamma_0}\right)$, (31) always holds.

Similar to the case of $\alpha > 1$, the total iteration number is:

$$\begin{aligned} K &= \mathcal{O}\left(\sqrt{\frac{L_{f_1} + \gamma_0 L_{g_1}}{\epsilon_0}}\right) + \dots + \mathcal{O}\left(\sqrt{\frac{L_{f_1} + \gamma_k L_{g_1}}{\epsilon_k}}\right) \\ &= \mathcal{O}\left(\sqrt{\frac{L_{f_1} + \gamma_k L_{g_1}}{\epsilon_k}}\right) \\ &= \mathcal{O}\left(\sqrt{\frac{L_{f_1} \eta^k}{\epsilon_0}} + \sqrt{\frac{L_{g_1} \gamma_0 (\eta \nu)^k}{\epsilon_0}}\right). \end{aligned}$$

□

E.9 Proof of Theorem 3.8

Before proving Theorem 3.8, we need the following lemma that is modified from Theorem 1 in Lin and Xiao [2014], we state it in the subsequent lemma for completeness.

Lemma E.1. *Suppose that Assumptions 2.1, 3.1, 3.2, and 3.7 hold. Let \mathbf{x}_γ^* be an optimal solution of problem (P_γ) and suppose that there exists a constant R such that $\max\{\|\mathbf{y}_0 - \mathbf{x}_\gamma^*\|, \|\tilde{\mathbf{x}} - \mathbf{x}_\gamma^*\|\} \leq R$. Then, the sequence $\{\mathbf{x}_k\}$ generated by Algorithm 3 satisfy*

$$\Phi_\gamma(\mathbf{x}_k) - \Phi_\gamma(\mathbf{x}_\gamma^*) \leq \left(\frac{L_\gamma + \mu}{2} R^2 \right) \left(1 - \sqrt{\frac{\mu}{L_\gamma}} \right)^k. \quad (32)$$

Proof. Denote $L_\gamma = L_{f_1} + \gamma L_{g_1}$. By Theorem 3.1 in Beck and Teboulle [2009], we have

$$\Phi_\gamma(\tilde{\mathbf{x}}) - \Phi_\gamma(\mathbf{x}_\gamma^*) \leq \frac{L_\gamma}{2} \|\mathbf{y}_0 - \mathbf{x}_\gamma^*\|^2. \quad (33)$$

Utilize Theorem 1 in Lin and Xiao [2014], we have

$$\begin{aligned} \Phi_\gamma(\mathbf{x}_k) - \Phi_\gamma(\mathbf{x}_\gamma^*) &\leq \left(\Phi_\gamma(\tilde{\mathbf{x}}) - \Phi_\gamma(\mathbf{x}_\gamma^*) + \frac{\mu}{2} \|\tilde{\mathbf{x}} - \mathbf{x}_\gamma^*\|^2 \right) \left(1 - \sqrt{\frac{\mu}{L_\gamma}} \right)^k \\ &\stackrel{(33)}{\leq} \left(\frac{L_\gamma}{2} \|\mathbf{y}_0 - \mathbf{x}_\gamma^*\|^2 + \frac{\mu}{2} \|\tilde{\mathbf{x}} - \mathbf{x}_\gamma^*\|^2 \right) \left(1 - \sqrt{\frac{\mu}{L_\gamma}} \right)^k \\ &\leq \left(\frac{L_\gamma + \mu}{2} R^2 \right) \left(1 - \sqrt{\frac{\mu}{L_\gamma}} \right)^k. \end{aligned} \quad (34)$$

□

By Lemma E.1, we are now prepared to prove Theorem 3.8.

Proof. By Lemma E.1, the number of iterations required to achieve an ϵ -optimal solution for problem (P_γ) is

$$K = \mathcal{O} \left(\sqrt{\frac{L_\gamma}{\mu}} \log \left(\frac{L_\gamma + \mu}{2\epsilon} R^2 \right) \right) = \mathcal{O} \left(\sqrt{\frac{L_\gamma}{\mu}} \log \frac{1}{\epsilon} \right).$$

- **Case of $\alpha > 1$.** In this case, $\gamma = \gamma^* + 2l_F^\beta \epsilon^{1-\beta}$, where $\gamma^* = \rho l_F^\alpha (\alpha - 1)^{\alpha-1} \alpha^{-\alpha} \epsilon^{1-\alpha}$.

If $\beta < \alpha$, the dominating term in γ is γ^* . Then, the number of iterations is

$$K = \mathcal{O} \left(\sqrt{\frac{L_{f_1} + l_F^\alpha \epsilon^{1-\alpha} L_{g_1}}{\mu}} \log \frac{1}{\epsilon} \right) = \mathcal{O} \left(\sqrt{\frac{L_{f_1}}{\mu}} \log \frac{1}{\epsilon} + \sqrt{\frac{l_F^\alpha L_{g_1}}{\epsilon^{\alpha-1}}} \log \frac{1}{\epsilon} \right).$$

If $\beta = \alpha$, we have $\gamma = (\rho(\alpha - 1)^{\alpha-1} \alpha^{-\alpha} + 2) l_F^\alpha \epsilon^{1-\alpha}$. Then, the number of iterations is

$$K = \mathcal{O} \left(\sqrt{\frac{L_{f_1} + l_F^\alpha \epsilon^{1-\alpha} L_{g_1}}{\mu}} \log \frac{1}{\epsilon} \right) = \mathcal{O} \left(\sqrt{\frac{L_{f_1}}{\mu}} \log \frac{1}{\epsilon} + \sqrt{\frac{l_F^\alpha L_{g_1}}{\epsilon^{\alpha-1}}} \log \frac{1}{\epsilon} \right).$$

If $\beta > \alpha$, the dominating term in γ is $2l_F^\beta \epsilon^{1-\beta}$. Then, the number of iterations is

$$K = \mathcal{O} \left(\sqrt{\frac{L_{f_1} + 2l_F^\beta \epsilon^{1-\beta} L_{g_1}}{\mu}} \log \frac{1}{\epsilon} \right) = \mathcal{O} \left(\sqrt{\frac{L_{f_1}}{\mu}} \log \frac{1}{\epsilon} + \sqrt{\frac{l_F^\beta L_{g_1}}{\epsilon^{\beta-1}}} \log \frac{1}{\epsilon} \right).$$

- **Case of $\alpha = 1$.** When $\alpha = 1$, γ can be written as $\gamma = \gamma^* + l_F^\beta \epsilon^{1-\beta}$, where $\gamma^* = \rho l_F$.

If $\beta < 1$, the dominating term in γ is γ^* . Then, the number of iterations is

$$K = \mathcal{O} \left(\sqrt{\frac{L_{f_1} + \rho l_F L_{g_1}}{\mu}} \log \frac{1}{\epsilon} \right) = \mathcal{O} \left(\sqrt{\frac{L_{f_1}}{\mu}} \log \frac{1}{\epsilon} + \sqrt{\frac{l_F L_{g_1}}{\epsilon^{\alpha-1}}} \log \frac{1}{\epsilon} \right).$$

If $\beta = 1$, we have $\gamma = (\rho + 1) l_F \epsilon^{1-\alpha}$. Then, the number of iterations is

$$K = \mathcal{O} \left(\sqrt{\frac{L_{f_1} + \rho l_F L_{g_1}}{\mu}} \log \frac{1}{\epsilon} \right) = \mathcal{O} \left(\sqrt{\frac{L_{f_1}}{\mu}} \log \frac{1}{\epsilon} + \sqrt{\frac{l_F L_{g_1}}{\epsilon^{\alpha-1}}} \log \frac{1}{\epsilon} \right).$$

If $\beta > 1$, the dominating term in γ is $l_F^\beta \epsilon^{1-\beta}$. Then, we have

$$K = \mathcal{O} \left(\sqrt{\frac{L_{f_1} + l_F^\beta \epsilon^{1-\beta} L_{g_1}}{\mu}} \log \frac{1}{\epsilon} \right) = \mathcal{O} \left(\sqrt{\frac{L_{f_1}}{\mu}} \log \frac{1}{\epsilon} + \sqrt{\frac{l_F^\beta L_{g_1}}{\epsilon^{\beta-1}}} \log \frac{1}{\epsilon} \right).$$

Combining the above results, we conclude that

$$K = \mathcal{O} \left(\sqrt{\frac{L_{f_1}}{\mu}} \log \frac{1}{\epsilon} + \sqrt{\frac{l_F^{\max\{\alpha, \beta\}} L_{g_1}}{\epsilon^{\max\{\alpha-1, \beta-1\}}}} \log \frac{1}{\epsilon} \right).$$

□

E.10 Proof of Theorem 3.9

Proof. Denote $l_\gamma = l_{f_2} + \gamma l_{g_2}$. Define $\Phi_{\gamma, best}^K = \min_{i=0, \dots, K} \Phi_\gamma(\mathbf{x}_i)$ and $\hat{\Phi}_{\gamma, best}^{K, j} = \min_{i=j, \dots, K} \Phi_\gamma(\mathbf{x}_i)$ for all $0 \leq j \leq K$. We claim that the sequence generated by the subgradient method satisfies

$$\Phi_{\gamma, best}^K - \Phi_\gamma^* \leq \frac{l_\gamma R^2 + 2 \log 2}{4 \sqrt{K+2}}. \quad (35)$$

Specifically, from Lemma 8.24 in Beck [2017], for all $0 \leq j \leq K$, we have

$$\hat{\Phi}_{\gamma, best}^{K, j} - \Phi_\gamma^* \leq \frac{1}{2} \frac{R^2 + \sum_{k=j}^K \eta_k^2 \|\xi_k\|^2}{\sum_{k=j}^K \eta_k}. \quad (36)$$

Define $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ as rounding up and rounding down, respectively. Let $j = \lfloor \frac{K}{2} \rfloor$ in (36), by the definition of step-size $\eta_k = \frac{R}{l_\gamma \sqrt{k+1}}$, we have

$$\hat{\Phi}_{\gamma, best}^{K, j} - \Phi_\gamma^* \leq \frac{l_\gamma R^2 + \sum_{k=\lfloor \frac{K}{2} \rfloor}^K \frac{1}{k+1}}{2 \sum_{k=\lfloor \frac{K}{2} \rfloor}^K \frac{1}{\sqrt{k+1}}} \leq \frac{l_\gamma R^2 + 2 \log 2}{4 \sqrt{K+2}}, \quad (37)$$

where the second inequality follows from that $\sum_{k=\lfloor \frac{K}{2} \rfloor}^K \frac{1}{k+1} \leq \int_{\lfloor \frac{K}{2} \rfloor - 1}^K \frac{1}{s+1} ds \leq 2 \log 2$ and $\sum_{k=\lfloor \frac{K}{2} \rfloor}^K \frac{1}{\sqrt{k+1}} \geq \int_{\lfloor \frac{K}{2} \rfloor}^{K+1} \frac{1}{\sqrt{s+1}} ds \geq \frac{1}{2} \sqrt{K+2}$.

From the fact that $\Phi_{\gamma, best}^K \leq \hat{\Phi}_{\gamma, best}^{K, j}$, The desired result of (35) follows.

Then, inequality (35) demonstrates that the number of iterations to obtain an ϵ -optimal solution for problem (P_γ) is

$$K = \mathcal{O} \left(\frac{l_{f_2} + \gamma l_{g_2}}{\epsilon} \right)^2.$$

- **Case of $\alpha > 1$.** we have $\gamma = \gamma^* + 2l_{f_2}^\beta \epsilon^{1-\beta}$ and $\gamma^* = \rho l_{f_2}^\alpha (\alpha - 1)^{\alpha-1} \alpha^{-\alpha} \epsilon^{1-\alpha}$.

If $\beta < \alpha$, the dominating term in γ is γ^* . Then, the number of iterations is

$$K = \mathcal{O} \left(\frac{l_{f_2} + l_{f_2}^\alpha \epsilon^{1-\alpha} l_{g_2}}{\epsilon} \right)^2 = \mathcal{O} \left(\frac{l_{f_2}^2}{\epsilon^2} + \frac{l_{f_2}^{2\alpha} l_{g_2}^2}{\epsilon^{2\alpha}} \right).$$

If $\beta = \alpha$, we have $\gamma = (\rho(\alpha - 1)^{\alpha-1} \alpha^{-\alpha} + 2) l_F^\alpha \epsilon^{1-\alpha}$. Then, the number of iterations is

$$K = \mathcal{O} \left(\frac{l_{f_2} + l_{f_2}^\alpha \epsilon^{1-\alpha} l_{g_2}}{\epsilon} \right)^2 = \mathcal{O} \left(\frac{l_{f_2}^2}{\epsilon^2} + \frac{l_{f_2}^{2\alpha} l_{g_2}^2}{\epsilon^{2\alpha}} \right).$$

If $\beta > \alpha$, the dominating term in γ is $2l_F^\beta \epsilon^{1-\beta}$. Then, the number of iterations is

$$K = \mathcal{O} \left(\frac{l_{f_2} + 2l_{f_2}^\beta \epsilon^{1-\beta} l_{g_2}}{\epsilon} \right)^2 = \mathcal{O} \left(\frac{l_{f_2}^2}{\epsilon^2} + \frac{l_{f_2}^{2\beta} l_{g_2}^2}{\epsilon^{2\beta}} \right).$$

- **Case of $\alpha = 1$.** we have $\gamma = \gamma^* + l_{f_2}^\beta \epsilon^{1-\beta}$ and $\gamma^* = \rho l_{f_2}$.

If $\beta < 1$, the dominating term in γ is γ^* . Then, the number of iterations is

$$K = \mathcal{O}\left(\frac{(l_{f_2} + \rho l_{f_2} l_{g_2})^2}{\epsilon}\right) = \mathcal{O}\left(\frac{l_{f_2}^2}{\epsilon^2} + \frac{l_{f_2}^2 l_{g_2}^2}{\epsilon^2}\right).$$

If $\beta = 1$, we have $\gamma = (\rho + 1) l_{f_2} \epsilon^{1-\alpha}$. Then, the number of iterations is

$$K = \mathcal{O}\left(\frac{(l_{f_2} + \rho l_{f_2} l_{g_2})^2}{\epsilon}\right) = \mathcal{O}\left(\frac{l_{f_2}^2}{\epsilon^2} + \frac{l_{f_2}^2 l_{g_2}^2}{\epsilon^2}\right).$$

If $\beta > 1$, the dominating term in γ is $l_{f_2}^\beta \epsilon^{1-\beta}$. Then, the number of iterations is

$$K = \mathcal{O}\left(\frac{(l_{f_2} + l_{f_2}^\beta l_{g_2} \epsilon^{1-\beta})^2}{\epsilon}\right) = \mathcal{O}\left(\frac{l_{f_2}^2}{\epsilon^2} + \frac{l_{f_2}^{2\beta} l_{g_2}^2}{\epsilon^{2\beta}}\right).$$

Combining the above results, we conclude that

$$K = \mathcal{O}\left(\frac{l_{f_2}^2}{\epsilon^2} + \frac{l_{f_2}^{\max\{2\alpha, 2\beta\}} l_{g_2}^2}{\epsilon^{\max\{2\alpha, 2\beta\}}}\right).$$

□

E.11 Proof of Theorem 3.10

Proof. Denote $l_\gamma = l_{f_2} + \gamma l_{g_2}$, define $\Phi_{\gamma, best}^K = \min_{i=0, \dots, K} \Phi_\gamma(\mathbf{x}_i)$. From Theorem 8.31 in Beck [2017], the sequence generated by the subgradient method satisfies

$$\Phi_{\gamma, best}^K - \Phi_\gamma^* \leq \frac{2l_\gamma^2}{\mu_{f_2}(K+1)}.$$

This demonstrates that the number of iterations to obtain an ϵ -optimal solution for problem (P_γ) is

$$K = \mathcal{O}\left(\frac{(l_{f_2} + \gamma l_{g_2})^2}{\mu_{f_2} \epsilon}\right).$$

- **Case of $\alpha > 1$.** we have $\gamma = \gamma^* + 2l_{f_2}^\beta \epsilon^{1-\beta}$ and $\gamma^* = \rho l_{f_2}^\alpha (\alpha - 1)^{\alpha-1} \alpha^{-\alpha} \epsilon^{1-\alpha}$.

If $\beta < \alpha$, the dominating term in γ is γ^* . Then, the number of iterations is

$$K = \mathcal{O}\left(\frac{(l_{f_2} + l_{f_2}^\alpha \epsilon^{1-\alpha} l_{g_2})^2}{\mu_{f_2} \epsilon}\right) = \mathcal{O}\left(\frac{l_{f_2}^2}{\mu_{f_2} \epsilon} + \frac{l_{f_2}^{2\alpha} l_{g_2}^2}{\mu_{f_2} \epsilon^{2\alpha-1}}\right).$$

If $\beta = \alpha$, we have $\gamma = (\rho(\alpha - 1)^{\alpha-1} \alpha^{-\alpha} + 2) l_{f_2}^\alpha \epsilon^{1-\alpha}$. Then, the number of iterations is

$$K = \mathcal{O}\left(\frac{(l_{f_2} + l_{f_2}^\alpha \epsilon^{1-\alpha} l_{g_2})^2}{\mu_{f_2} \epsilon}\right) = \mathcal{O}\left(\frac{l_{f_2}^2}{\mu_{f_2} \epsilon} + \frac{l_{f_2}^{2\alpha} l_{g_2}^2}{\mu_{f_2} \epsilon^{2\alpha-1}}\right).$$

If $\beta > \alpha$, the dominating term in γ is $2l_{f_2}^\beta \epsilon^{1-\beta}$. Then, the number of iterations is

$$K = \mathcal{O}\left(\frac{(l_{f_2} + 2l_{f_2}^\beta \epsilon^{1-\beta} l_{g_2})^2}{\mu_{f_2} \epsilon}\right) = \mathcal{O}\left(\frac{l_{f_2}^2}{\mu_{f_2} \epsilon} + \frac{l_{f_2}^{2\beta} l_{g_2}^2}{\mu_{f_2} \epsilon^{2\beta-1}}\right).$$

- **Case of $\alpha > 1$.** we have $\gamma = \gamma^* + l_{f_2}^\beta \epsilon^{1-\beta}$ and $\gamma^* = \rho l_{f_2}$.

If $\beta < 1$, the dominating term in γ is γ^* . Then, the number of iterations is

$$K = \mathcal{O}\left(\frac{(l_{f_2} + \rho l_{f_2} l_{g_2})^2}{\mu_{f_2} \epsilon}\right) = \mathcal{O}\left(\frac{l_{f_2}^2}{\mu_{f_2} \epsilon} + \frac{l_{f_2}^2 l_{g_2}^2}{\mu_{f_2} \epsilon}\right).$$

If $\beta = 1$, we have $\gamma = (\rho + 1)l_F\epsilon^{1-\alpha}$. Then, the number of iterations is

$$K = \mathcal{O}\left(\frac{(l_{f_2} + \rho l_{f_2} l_{g_2})^2}{\mu_{f_2} \epsilon}\right) = \mathcal{O}\left(\frac{l_{f_2}^2}{\mu_{f_2} \epsilon} + \frac{l_{f_2}^2 l_{g_2}^2}{\mu_{f_2} \epsilon}\right).$$

If $\beta > 1$, the dominating term in γ is $l_{f_2}^\beta \epsilon^{1-\beta}$. Then, the number of iterations is

$$K = \mathcal{O}\left(\frac{(l_{f_2} + l_{f_2}^\beta l_{g_2} \epsilon^{1-\beta})^2}{\mu_{f_2} \epsilon}\right) = \mathcal{O}\left(\frac{l_{f_2}^2}{\mu_{f_2} \epsilon} + \frac{l_{f_2}^{2\beta} l_{g_2}^2}{\mu_{f_2} \epsilon^{2\beta-1}}\right).$$

Combining the above results, we conclude that

$$K = \mathcal{O}\left(\frac{l_{f_2}^2}{\mu_{f_2} \epsilon} + \frac{l_{f_2}^{\max\{2\alpha, 2\beta\}} l_{g_2}^2}{\mu_{f_2} \epsilon^{\max\{2\alpha-1, 2\beta-1\}}}\right).$$

□

F Implementation details

In this section, we provide supplementary experiment settings and results. Specifically, in Appendix F.1, we present the detailed experimental settings, and in Appendix F.2, we provide the detailed experimental results. Additionally, in Appendix F.3 and F.4, we conduct experiments with different values of penalty parameter γ and solution accuracy ϵ , respectively.

F.1 Experiment setting

All simulations are implemented using MATLAB R2023a on a PC running Windows 11 with an AMD (R) Ryzen (TM) R7-7840H CPU (3.80GHz) and 16GB RAM.

F.1.1 Experiment setting of Section 4.1

We conduct the first experiment using the `a1a.t` data from LIBSVM datasets⁵. This data consists of 30,956 instances, each with $n = 123$ features. For this experiment, a sample of 1,000 instances is taken from the data, denoted as A . The corresponding labels for these instances are denoted as b , where each label b_i is either -1 or 1 , corresponding to the i -th instance \mathbf{a}_i .

The Greedy FISTA algorithm [Liang et al., 2022] is used as a benchmark to compute G^* . To compute the proximal mapping of $f_2(\mathbf{x}) + \gamma g_2(\mathbf{x})$ in problem (P_γ) , i.e., projection onto a 1-norm ball, we utilize the method proposed in Duchi et al. [2008], which performs exact projection in $\mathcal{O}(n)$ expected time, where n is the dimension of \mathbf{x} .

For the PB-APG and PB-APG-sc algorithms, we set the value of $\gamma = 10^5$, and we terminate the algorithms when $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| \leq 10^{-10}$. For the aPB-APG and aPB-APG-sc algorithms, we set $\gamma_0 = \frac{1}{2^5}$, $\nu = 20$, $\eta = 10$, and $\epsilon_0 = 10^{-6}$. The iterations of these two algorithms continue until ϵ_k reaches 10^{-10} (meanwhile, $\gamma = 10^5$).

We compare our methods with MNG, BiG-SAM, DBGD, a-IRG, CG-BiO, Bi-SG, and R-APM in this experiment. Specifically, for R-APM [Samadi et al., 2023], the regularization parameter η is set to $\eta = 1/\gamma$, reflecting the equivalence of the penalty formulation (P_γ) to (P_{Reg}) , with $\sigma = 1/\gamma$, as previously discussed.

We note that the termination criterion $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| \leq 10^{-10}$ used in our experiments is different from the one proposed in our algorithms since the parameters required for the latter are not easily measurable. Nevertheless, this termination criterion is also widely used in the literature, as it corresponds to a gradient mapping [Beck, 2017, Nesterov, 2018, Davis and Drusvyatskiy, 2019]. Furthermore, Theorem 3.5 of Drusvyatskiy and Lewis [2018b] implies that $\|\mathbf{x}_{k+1} - \mathbf{x}_k\|$ also measures the distance to the optimal solution set.

F.1.2 Experiment setting of Section 4.2

In the second experiment, we address the problem of least squares regression using the `YearPredictionMSD` data from the UCI Machine Learning Repository⁶. This data consists of 515,345 songs with release years ranging from 1992 to 2011. Each song has 90 features, and the corresponding release year is used as the label.

⁵<https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary/a1a.t>

⁶<https://archive.ics.uci.edu/ml/datasets/YearPredictionMSD>

For this experiment, a sample of $m = 1,000$ songs is taken from the data, and the feature matrix and release years vector are denoted as A and b , respectively.

Following Section 5.2 in Merchav and Sabach [2023], we apply the min-max scaling technique to normalize the feature matrix A . Additionally, we add an intercept term and 90 collinear features to A such that the resulting matrix $A^T A$ becomes positive semi-definite, which implies that the feasible set X_{opt} is not a singleton.

We compare our methods with a-IRG, BiG-SAM, and Bi-SG in this experiment. Specifically, for BiG-SAM [Sabach and Shtern, 2017], we consider the accuracy parameter δ for the Moreau envelope with two values, namely $\delta = 1$ and $\delta = 0.01$.

To benchmark the performance, we utilize the MATLAB function `lsqminnorm` to compute G^* . Moreover, we follow the parameter settings outlined in Section 4.1.

F.2 Detailed results of experiments

To approximate the optimal value F^* , we use the MATLAB function `fmincon` to solve a relaxed version of the function-value-based reformulations in equation (P_{Val}). In this relaxed version, we replace the constraint in (P_{Val}) with $G(\mathbf{x}) - G^* \leq \varepsilon$, where $\varepsilon = 10^{-10}$. This allows us to obtain an approximation of the optimal value while allowing for a small deviation from the true optimal value G^* .

We gather the total number of iterations for our methods, as well as the lower- and upper-level objective values and the optimal gaps for all the methods, in Table 3. Subsequently, we compare the optimal gaps of all methods, which are defined as $G(\mathbf{x}) - G^*$ and $F(\mathbf{x}) - F^*$ for the lower- and upper-level optimal gaps, respectively.

Table 3: Methods comparison: lower- and upper-level objectives and optimal gaps

Logistic Regression Problem (5)					
Method	Total iterations	Lower-level value	Lower-level gap	Upper-level value	Upper-level gap
PB-APG	1470	3.2794e-01	1.7630e-08	4.9382e+00	-3.3998e-03
aPB-APG	1010	3.2794e-01	1.7630e-08	4.9382e+00	-3.3998e-03
PB-APG-sc	2278	3.2794e-01	1.7630e-08	4.9382e+00	-3.3998e-03
aPB-APG-sc	1046	3.2794e-01	1.7630e-08	4.9382e+00	-3.3998e-03
MNG	/	3.4540e-01	1.7459e-02	1.7469e+00	-3.1947e+00
BiG-SAM	/	3.3878e-01	1.0840e-02	2.2873e+00	-2.6543e+00
DBGD	/	5.2681e-01	1.9887e-01	8.8408e-02	-4.8532e+00
a-IRG	/	3.3765e-01	9.7121e-03	2.5401e+00	-2.4016e+00
CG-BIO	/	4.3040e-01	1.0246e-01	3.7684e-01	-4.5648e+00
Bi-SG	/	3.2806e-01	1.1530e-04	4.6873e+00	-2.5432e-01
R-APM	/	3.2794e-01	1.7645e-08	4.9382e+00	-3.4013e-03
Least Squares Regression Problem (6)					
Method	Total iterations	Lower-level value	Lower-level gap	Upper-level value	Upper-level gap
PB-APG	39314	7.3922e-03	6.0034e-07	4.7236e+00	-1.1888e-01
aPB-APG	40784	7.3922e-03	6.0030e-07	4.7236e+00	-1.1887e-01
PB-APG-sc	46446	7.3922e-03	6.0034e-07	4.7236e+00	-1.1888e-01
aPB-APG-sc	61777	7.3922e-03	6.0035e-07	4.7236e+00	-1.1888e-01
BiG-SAM ($\delta = 1$)	/	7.5189e-03	1.2733e-04	3.5081e+00	-1.3344e+00
BiG-SAM ($\delta = 0.01$)	/	7.3958e-03	4.2281e-06	5.8510e+01	5.3668e+01
a-IRG	/	1.6224e-02	8.8328e-03	4.7745e-01	-4.3651e+00
Bi-SG	/	8.5782e-03	1.1866e-03	1.3832e+00	-3.4593e+00

Table 3 reveals that for the logistic regression problem (5), our PB-APG, aPB-APG, PB-APG-sc, and aPB-APG-sc exhibit almost identical function values for both objectives, surpassing other methods in terms of optimal gaps for the lower- and upper-level objectives (measured by the numerical value of the upper-level objective). In the case of the least squares regression problem (6), aPB-APG achieves the smallest optimal gaps for both objectives, followed by PB-APG and PB-APG-sc. These results demonstrate that our methods, despite yielding larger upper-level function values, generate solutions that are significantly closer to the optimal solution, as depicted in Figure 1. Additionally, for the problem in (5), both aPB-APG and aPB-APG-sc require fewer iterations than PB-APG and PB-APG-sc, respectively. This can be attributed to the warm-start mechanism employed in aPB-APG and aPB-APG-sc. Moreover, for the problem in (6), both aPB-APG and aPB-APG-sc require more iterations than PB-APG and PB-APG-sc, respectively. However, they exhibit staircase-shaped curves, which avoid the unwanted oscillations in PB-APG and PB-APG-sc, we have a similar observation in Figure 2.

F.3 Supplementary experiments for different penalty parameters

In this section, we investigate the impact of different values of penalty parameter γ on the experimental results of problems (5) and (6). We set γ to be either 2×10^4 or 5×10^5 for PB-APG and PB-APG-sc, and choose the

Table 5: Lower- and upper-level objectives and optimal gaps with different penalty parameters for problem (6).

$\gamma = 2 \times 10^4$					
Method	Total iterations	Lower-level value	Lower-level gap	Upper-level value	Upper-level gap
PB-APG	17153	7.4052e-03	1.3619e-05	4.2843e+00	-5.5818e-01
aPB-APG	20877	7.4052e-03	1.3619e-05	4.2843e+00	-5.5818e-01
PB-APG-sc	27501	7.4052e-03	1.3619e-05	4.2843e+00	-5.5818e-01
aPB-APG-sc	40077	7.4052e-03	1.3619e-05	4.2843e+00	-5.5818e-01
$\gamma = 5 \times 10^5$					
Method	Total iterations	Lower-level value	Lower-level gap	Upper-level value	Upper-level gap
PB-APG	85511	7.3916e-03	2.4094e-08	4.8198e+00	-2.2752e-02
aPB-APG	85502	7.3916e-03	2.4093e-08	4.8198e+00	-2.2752e-02
PB-APG-sc	173731	7.3916e-03	2.4071e-08	4.8198e+00	-2.2740e-02
aPB-APG-sc	166324	7.3916e-03	2.4091e-08	4.8198e+00	-2.2751e-02

F.4 Supplementary experiments for different solution accuracies

In this section, we investigate the impact of different solution accuracies on the experimental results of problems (5) and (6). We set ϵ to be either 10^{-4} or 10^{-7} and terminate the algorithms for PB-APG and PB-APG-sc when $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| \leq \epsilon$. For aPB-APG and aPB-APG-sc, we choose the corresponding ϵ_0 values as 1 or 10^{-3} . The remaining settings are the same as in Section 4.

We also plot the values of the residuals of the lower-level objective $G(\mathbf{x}_k) - G^*$ and the upper-level objective over time in Figures 5 and 6. Additionally, we also collect the total number of iterations, the lower- and upper-level objective values, and the optimal gaps of our methods in Table 6 for problems (5) and (6) with different solution accuracies.

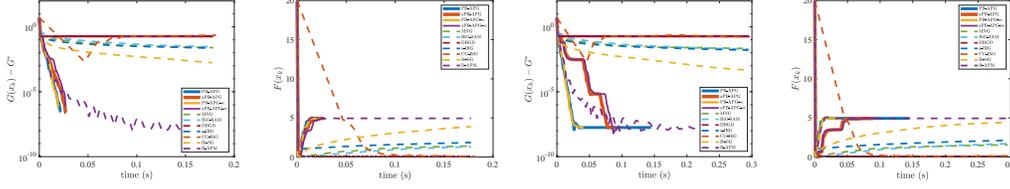


Figure 5: LRP (5) with $\epsilon = 10^{-4}$ (left two subfigures) and $\epsilon = 10^{-7}$ (right two subfigures).

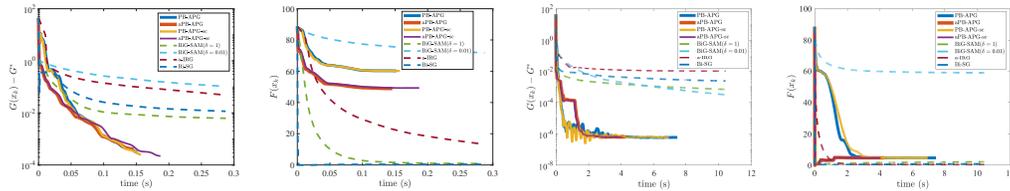


Figure 6: LSRP (6) with $\epsilon = 10^{-4}$ (left two subfigures) and $\epsilon = 10^{-7}$ (right two subfigures).

From Figures 5 and 6, it is evident that in most cases, our methods outperform the other methods in terms of both the lower- and upper-level objectives. However, there is an exception in the case of the upper-level objective for problem (6) when $\epsilon = 10^{-4}$. As illustrated in the second subfigure in Figure 6, our methods exhibit larger function values for the upper-level objective compared to the other methods (except BiG-SAM ($\delta = 0.01$)), despite still achieving smaller optimal gaps for the lower-level objective. This discrepancy actually indicates that our methods have not yet achieved the desired accuracy when $\epsilon = 10^{-4}$, and it is important to note that $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| \leq \epsilon$ is not the termination criterion in our proposed algorithms, as explained in Appendix F.1. Therefore, the larger optimality gaps for the upper-level objective in this case may be attributed to the termination criterion.

Table 6: Lower- and upper-level objectives and optimal gaps with different solution accuracies for problem (5).

$\epsilon = 10^{-4}$					
Method	Total iterations	Lower-level value	Lower-level gap	Upper-level value	Upper-level gap
PB-APG	124	3.2794e-01	2.8671e-07	4.9483e+00	6.7024e-03
aPB-APG	148	3.2794e-01	2.3660e-07	4.9419e+00	2.9831e-04
PB-APG-sc	100	3.2794e-01	5.4674e-07	4.9287e+00	-1.2956e-02
aPB-APG-sc	149	3.2794e-01	7.9015e-07	4.9302e+00	-1.1404e-02
$\epsilon = 10^{-7}$					
Method	Total iterations	Lower-level value	Lower-level gap	Upper-level value	Upper-level gap
PB-APG	841	3.2794e-01	1.7631e-08	4.9382e+00	-3.3999e-03
aPB-APG	551	3.2794e-01	1.7707e-08	4.9382e+00	-3.4075e-03
PB-APG-sc	225	3.2794e-01	1.7493e-08	4.9383e+00	-3.3691e-03
aPB-APG-sc	614	3.2794e-01	1.7507e-08	4.9382e+00	-3.3874e-03

Table 7: Lower- and upper-level objectives and optimal gaps with different solution accuracies for problem (6).

$\epsilon = 10^{-4}$					
Method	Total iterations	Lower-level value	Lower-level gap	Upper-level value	Upper-level gap
PB-APG	426	7.6950e-03	3.0342e-04	6.0249e+01	5.5407e+01
aPB-APG	432	7.8018e-03	4.1016e-04	4.8967e+01	4.4125e+01
PB-APG-sc	437	7.6456e-03	2.5400e-04	6.0196e+01	5.5354e+01
aPB-APG-sc	517	7.6143e-03	2.2274e-04	4.9292e+01	4.4449e+01
$\epsilon = 10^{-7}$					
Method	Total iterations	Lower-level value	Lower-level gap	Upper-level value	Upper-level gap
PB-APG	13707	7.3922e-03	5.9756e-07	4.7279e+00	-1.1460e-01
aPB-APG	7803	7.3923e-03	6.5025e-07	4.7300e+00	-1.1248e-01
PB-APG-sc	12724	7.3922e-03	5.7840e-07	4.7354e+00	-1.0714e-01
aPB-APG-sc	7429	7.3922e-03	6.3816e-07	4.7326e+00	-1.0992e-01

Tables 3, 6, and 7 demonstrate that the number of iterations for our methods also increases with the solution accuracy, while the optimal gaps of the lower- and upper-level objectives decrease correspondingly. This finding confirms that the number of iterations and the optimal gaps are influenced by the solution accuracy, as illustrated in the expressions for the number of iterations provided by Theorem 3.3 and other related theorems.

NeurIPS Paper Checklist

1. Claims

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

Answer: [Yes]

Justification: Please refer to Abstract and Section 1.

Guidelines:

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

2. Limitations

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

Answer: [Yes]

Justification: Please refer to the assumptions adopted in this paper (e.g. Assumptions 2.1, 2.2, and 3.2), our study is based on these assumptions.

Guidelines:

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

3. Theory Assumptions and Proofs

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

Answer: [Yes]

Justification: Please refer to the theorems and the proofs of them in this paper, please refer to Appendix E. For example, Theorem 3.3 and its proof in Appendix E.7.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.

- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

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

Answer: [Yes]

Justification: Please refer to Section 4 and Appendix F.

Guidelines:

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

5. Open access to data and code

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

Answer: [Yes]

Justification: Please refer to Section 4 and Appendix F and the supplemental material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so No is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

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

Answer: [Yes]

Justification: Please refer to Section 4 and Appendix F.

Guidelines:

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

7. Experiment Statistical Significance

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

Answer: [No]

Justification: The error bars are not applicable in this paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

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

Answer: [Yes]

Justification: Please refer to Section 4 and Appendix F.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

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

Answer: [Yes]

Justification: Please refer to Section 4 and Appendix F.

Guidelines:

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

10. Broader Impacts

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

Answer: [NA]

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

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

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

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

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

12. Licenses for existing assets

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

Answer: [Yes]

Justification: All the creators or original owners of assets are properly credited, and the license and terms of use are explicitly mentioned and properly respected, please refer to Section 4 and Appendix F.

Guidelines:

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

13. New Assets

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

Answer: [Yes]

Justification: Please refer to the supplemental materials and the 'README.m' file.

Guidelines:

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

14. Crowdsourcing and Research with Human Subjects

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

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

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

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

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

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

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