Beyond Value Functions: Single-Loop Bilevel Optimization under Flatness Conditions

Liuyuan Jiang[*,◇], **Quan Xiao**[†,◇], **Lisha Chen**[*], **Tianyi Chen**[†,◇]
[◇]Rensselaer Polytechnic Institute, Troy, NY
[*]University of Rochester, Rochester, NY
[†]Cornell Tech, Cornell University, New York, NY
ljiang24@ur.rochester.edu, qx232@cornell.edu
lisha.chen@rochester.edu, tianyi.chen@cornell.edu [*]

Abstract

Bilevel optimization, a hierarchical optimization paradigm, has gained significant attention in a wide range of practical applications, notably in the fine-tuning of generative models. However, due to the nested problem structure, most existing algorithms require either the Hessian vector calculation or the nested loop updates, which are computationally inefficient in large language model (LLM) fine-tuning. In this paper, building upon the fully first-order penalty-based approach, we propose an efficient value function-free (PBGD-Free) algorithm that eliminates the loop of solving the lower-level problem and admits fully single-loop updates. Inspired by the landscape analysis of representation learning-based LLM fine-tuning problem, we propose a relaxed flatness condition for the upper-level function and prove the convergence of the proposed value-function-free algorithm. We test the performance of the proposed algorithm in various applications and demonstrate its superior computational efficiency over the state-of-the-art bilevel methods.

1 Introduction

Bi-level optimization (BLO) has gained significant attention for its powerful modeling capabilities in hierarchical learning across a wide range of real-world applications, such as distributed learning [69, 28], meta learning [27, 12, 27], model pruning [111, 96], reinforcement learning [108, 84, 79, 85], continual learning [8, 33], fine-tuning large language models (LLMs) [76, 68, 64, 56, 107] and diffusion models [93, 17, 61]. In this paper, we consider the BLO problem with $f : \mathbb{R}^{d_x} \times \mathbb{R}^{d_y} \to \mathbb{R}$ and $g : \mathbb{R}^{d_x} \times \mathbb{R}^{d_y} \to \mathbb{R}$ being the upper-level (UL) and lower-level (LL) objectives that are continuously differentiable but not necessarily convex. Since the LL problem may contain multiple solutions in $S_g^*(x)$, we consider the *optimistic* BLO formulation which selects the one y that minimizes the UL objective, given by

$$\min_{x,y} \ f(x,y) \ \text{ s.t. } \ y \in S_g^*(x) := \arg\min_y g(x,y). \tag{1}$$

In large-scale machine learning problems, efficiency is given a higher priority [110] and it is critical to use gradient-based approaches to solve the above problem. One can perform a direct gradient descent (GD) on the hyper-objective $\phi(x) := \min_{y \in S_g^*(x)} f(x,y)$. A popular GD-based method is the implicit gradient descent (IGD) method with second-order Hessian evaluation [29, 34, 39, 15, 42, 47, 82]. However, evaluating Hessian or its inverse in IGD is costly. To reduce the computational burden, especially in large-scale problems, first-order gradient-based methods, including the penalty-based BLO methods [105, 50, 77, 44, 45, 40, 57], have been developed. For example, penalizing the LL

[*]The work was supported by the National Science Foundation Projects 2401297, 2532349 and 2532653, and by the Cisco Research Award.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

objective optimality gap into the UL via a large penalty constant γ has been proposed [105, 78, 45, 58], yielding the following objective

$$\min_x F_\gamma(x) := \min_y \tilde{F}_\gamma(x,y) := f(x,y) + \gamma\big(g(x,y) - \min_z g(x,z)\big). \tag{2}$$

Under a proper curvature assumption for the LL problem, the penalty reformulation proves to be differentiable and smooth [78, 45, 40, 11], which enables the design of penalty-based gradient descent algorithms (PBGD) [78, 44, 45, 10]. Furthermore, the function value gap $|F_\gamma(x) - \phi(x)| = \mathcal{O}(\gamma^{-1})$ ensures the solution to the reformulated problem is an approximate solution to the original problem. The reformulation in (2) provides two choices of algorithm update: jointly updating x and y to minimize $\tilde{F}_\gamma(x,y)$ [105, 78], or alternatively optimizing y then updating x to minimize the hyper objective $F_\gamma(x)$ [45, 11]. Each has its pros and cons. For example, joint update eliminates the inner loop of y so that it has low per-iteration cost, but high smoothness constant of $\tilde{F}_\gamma(x,y)$, which increases with γ, making the convergence rate suboptimal [78, 105]. In contrast, the smoothness constant of $F_\gamma(x)$ remains $\mathcal{O}(1)$ since the value function gap $\gamma(g(x,y) - \min_z g(x,z))$ remains in $\mathcal{O}(1)$ when y minimizes $\tilde{F}(x,y)$, but estimating $\nabla F_\gamma(x)$ often requires running inner loops to obtain $y_g^*(x) \in S_g^*(x)$ and $y_\gamma^*(x) \in S_\gamma^*(x) := \operatorname{argmin}_y \tilde{F}_\gamma(x,y)$ [11]. Then a natural question is:

(Q1) *Can we develop an efficient algorithm that combines the best of both worlds?*

The idea is to update x by $\nabla_x f(x, y_\gamma^*(x))$ and skip the inner loop estimation for $y_g^*(x)$, which we term as PBGD Free of value function evaluation (PBGD-Free). To be more specific, we illustrate the updates for standard PBGD, its variants, and PBGD-Free in Figure 1.

Positive empirical observations on Q1: *PBGD-Free largely reduces computation and memory cost while preserving the accuracy in LLM parameter efficient fine-tuning (PEFT) [72, 2].* See Figure 2 and experiments in Section 4. We prioritize supervised fine-tuning (SFT) loss at the LL to ensure a capable base LLM model, while we keep direct preference optimization (DPO) loss [70] in the UL to keep alignment with human preferences:

$$\min_{x,y} \; f_{\text{DPO}}(x,y;\mathcal{D}_{\text{DPO}}) \tag{3}$$
$$\text{s.t. } y \in \arg\min_y g_{\text{SFT}}(x,y;\mathcal{D}_{\text{SFT}}),$$

where x is a pretrained LLM model, and y is an easy-to-fine-tune head. This design is aligned with both theoretical and practical needs in LLM deployment, as detailed in Section 4.

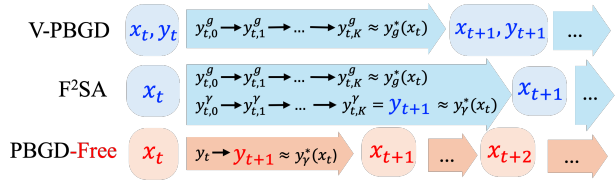

Figure 1: Update schemes for V-PBGD, F^{2}SA and PBGD-Free. V-PBGD [78] (**top**) and F^{2}SA [45] (**middle**) refine the LL variable over multiple steps before updating x_t via $\nabla_x \tilde{F}_\gamma(x_t, y_t)$ for V-PBGD or $\nabla_x \tilde{F}_\gamma(x_t, y_t)$ for F^{2}SA while PBGD-Free (**bottom**) applies a 1-step inner update to find a more efficient yet potentially less accurate $\nabla_x f(x_t, y_{t+1}) \approx \nabla F_\gamma(x_t)$.

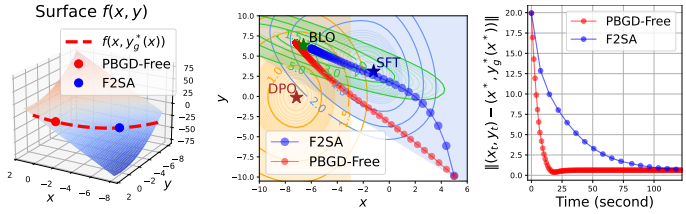

Figure 2: **An Illustration to show PBGD-Free does not work in Example 1, but works well in PEFT.** The **left** plot shows the $f(x,y)$ and $f(x, y_g^*(x))$ in Example 1, with red and blue dots as the converged points using PBGD-Free and F^{2}SA method. The **middle** plot shows the trajectory of updates in PEFT. The orange, cyan, and green contours are the landscapes of $f_{\text{DPO}}(x,y)$, $g_{\text{SFT}}(x,y)$, and $\tilde{F}_\gamma(x,y)$, respectively. The **right** plot presents the convergence vs. time in PEFT, showing faster convergence of PBGD-Free. (See Appendix C.1 for details.)

Negative theoretical observations on Q1: *There are some counterexamples where PBGD-Free does not converge.* **c1)** When the UL objective solely depends on the LL variable $f(x,y) = f(y)$, as in data hyper-cleaning [75, 34, 76] and meta-learning [27, 12, 27], the LL penalty gradient term contains all the gradient information about the UL variable x, so it cannot be omitted; and, **c2)** When $f(x,y)$ jointly depends on both variables, omission of the penalty gradient term can lead to a different update direction; see more details in Example 1 and Proposition 2.

Property	PBGD-Free	V-PBGD	BOME	F^{2}SA-MA	F^{2}SA	BVFSM
$f(x,\cdot)$	Flat	Lip	Lip & B	Lip	Lip	Diff
$g(x,\cdot)$	PL	PL	PL & B	PL	PL	Diff
$f(x,y)+\gamma g(x,y)$	PL	/	/	PL	PL	/
Single-loop	✓	✘	✘	✓	✘	✘
Memory cost	d_x+d_y	d_x+2d_y	$3d_x+4d_y$	d_x+5d_y	d_x+2d_y	d_x+2d_y
Complexity	$\mathcal{O}(\epsilon^{-1})$	$\tilde{\mathcal{O}}(\epsilon^{-1.5})$	$\tilde{\mathcal{O}}(\epsilon^{-4})$	$\mathcal{O}(\epsilon^{-1.5})$	$\tilde{\mathcal{O}}(\epsilon^{-1})$	Asym

Table 1: Comparison of the proposed method (PBGD-Free) with the existing first-order approaches for BLO with nonconvex LL problem (PBGD [78], BOME [105], F^{2}SA-MA [45] (with momentum assistance), F^{2}SA [11] and BVFSM [51]) in deterministic setting. The notation $\tilde{\mathcal{O}}$ hides dependency on $\log(\epsilon^{-1})$ terms. 'Flat', 'Lip', 'B', 'Diff', and 'Asym' stand for 'flatness condition' in Def. 1, 'Lipschitz continuous', 'bounded', 'differentiable', and 'asymptotic convergence'.

Example 1. *For the BLO problem in* (1) *with* $f(x,y)=x^2+10y$ *and* $g(x,y)=(y-x+1)^2$*, the gradients* $\langle \nabla F_\gamma(x), \nabla_x f(x, y_\gamma^*(x))\rangle < 0$ *exhibit* ***opposite directions*** *for* $x\in(-5,0)$*. As a result,* $\nabla F_\gamma(x)=2x+10$ *converges to* $x=-5$ *while* $\nabla_x f(x,y_\gamma^*(x))=2x$ *converges to* $x=0$*.*

These findings leave it unclear when PBGD-Free can be applied without sacrificing accuracy. In this paper, we focus on the case with UL joint dependency, where the objective $f(x,y)$ depends intrinsically on both x and y (i.e., cannot be simplified to $f(y)$). We explore the following question:

(Q2) *Can we identify sufficient conditions under which the PBGD-Free algorithm is guaranteed to converge to the stationary solution of the original problem?*

We give an affirmative answer to the above question. Specifically, **our key contributions** are summarized as follows, and the broader impact is discussed in Appendix D.

- **C1)** We propose PBGD-Free, a computationally efficient fully-single-loop, value-function-free, first-order algorithm. See a detailed comparison with other algorithms in Table 1. Specifically, compared to V-PBGD, it reduces the memory cost from $O(d_x+2d_y)$ to $O(d_x+d_y)$, and the per-iteration computational complexity cost from $O(K)$ to $O(1)$, where K is the number of inner iterations. Furthermore, we show that empirically, it works in large-scale problems such as PEFT (3). But theoretically, under a *Lipschitz condition* on the UL objective, PBGD-Free only converges to an $\Theta(1)$-neighborhood of a stationary point.
- **C2)** We then introduce a Hölder-alike condition to describe the flatness of $f(x,\cdot)$ (see Definition 1), which relaxes the standard $l_{f,0}$-Lipschitz continuity assumption when $l_{f,0}$ is small. This condition allows us to establish an improved complexity of the PBGD-Free algorithm in $\mathcal{O}(\epsilon^{-1})$ (Theorem 3) to a necessary stationary condition of the original problem.
- **C3)** We validate our methods through applications to LLM with PEFT and bilevel low-rank adaptation. Across all experiments, PBGD-Free demonstrates much better efficiency and comparable or better accuracy than the state-of-the-art baselines. See Section 4.

1.1 Prior art

Second-order BLO methods. The convergence for IGD-based BLO approaches was firstly established for the unconstrained strongly-convex LL problem [29], with later literature focused on improving finite time convergence rate [29, 34, 38, 14, 15, 42, 47, 82, 16, 97, 39]. Another branch of methods is based on iterative differentiation (ITD) methods [60, 26, 63, 75, 7], but they generally lack finite-time guarantee under stochastic setting [32, 37]. However, convergence analysis for both ITD and IGD methods mentioned above is limited to the setting where the LL problem is strongly convex over y. This assumption does not align with large-scale machine learning applications, where the LL objective represents the loss of a neural network and is inherently nonconvex [86, 41]. Recent studies have generalized the IGD and ITD methods to address BLO with convex [81, 52] or even nonconvex LL problem [92, 3, 49, 50, 66, 98]. Nevertheless, both ITD and IGD require the computation of second-order information, making them inefficient for large-scale machine learning problems.

First-order BLO methods. Fully first-order bilevel methods based on equilibrium backpropagation [74, 74, 113] and penalty reformulation [78, 44, 45, 105, 11, 103] have become increasingly popular due to their computational efficiency and the ability to handle nonconvex LL problems. Later, penalty

approaches have been generalized to address BLO with constrained LL problem [100, 40] and distributed learning settings [99, 87]. However, the iteration complexity of fully first-order approaches remains suboptimal, exhibiting a logarithmic dependency due to the inner-loop overhead. To reduce the cost in inner loops for both $y_g^*(x)$ and $y_\gamma^*(x)$, PBGD [78] eliminated the inner loop for $y_\gamma^*(x)$ by jointly optimizing x and y_γ in (2), F^{2}SA [45] managed to be fully single-loop using momentum and warm-start techniques. However, both methods incur a suboptimal iteration complexity of $\mathcal{O}(\epsilon^{-1.5})$. [11] further improved iteration complexity of double-loop version of F^{2}SA by exploiting the fact that $F_\gamma(x)$ is $\mathcal{O}(1)$-Lipschitz smooth, but its inner loop leads to a high per-iteration computational cost and suboptimal convergence rate as $\mathcal{O}(\epsilon^{-1}\log(\epsilon^{-1}))$.

Landscape-aware optimization. Landscape-aware optimization leverages structural properties of objective functions into algorithm design to accelerate the convergence or improve the generalization. Newton-type methods, which use second-order curvature information to rescale gradients, have been utilized in BLO [23, 71, 21] for efficient Hessian-vector calculation in IGD-based BLO methods. Sharpness-aware minimization [25], which seeks solutions robust to local perturbations and promotes convergence to flat minima, has also been incorporated into BLO [1] for improved generalization. Other landscape conditions in single-level optimization, such as relaxed smoothness [109, 46] and Hessian spectrum [112, 30], are key to explaining the theoretical benefits of empirically effective algorithms like gradient clipping and Adam [43]. However, most existing works focus on second-order BLO algorithms, and none have explored BLO tailored landscape conditions.

2 Value Function Free Algorithm for BLO Problems

In this section, we introduce the value function free algorithm for bilevel problems and show that it does not always converge under the traditional Lipschitz condition.

2.1 Preliminary: the Lipschitzness condition and the penalty-based reformulation

We begin by introducing the standard Lipschitz condition on the UL objective $f(x,y)$, which is common in BLO analysis.

Assumption 1. *Assume that for all x, the UL objective $f(x,\cdot)$ is $l_{f,0}$-Lipschitz in y at $y_g^*(x)$ with some $l_{f,0}>0$, i.e.,*

$$|f(x,y_g^*(x)) - f(x,y)| \le l_{f,0}\|y - y_g^*(x)\|. \tag{4}$$

For differentiable f, Assumption 1 implies $\|\nabla_y f(x,y_g^*(x))\| \le l_{f,0}$. This assumption is crucial for the key results in BLO literature; e.g., [78, 45, 11, 15, 39, 34, 38, 105]. Together with the following standard assumption, it enables a good approximation of $F_\gamma(x)$ to the original problem.

Assumption 2. *Suppose that i) f and g are respectively $l_{f,1}$ and $l_{g,1}$-smooth; ii) $\nabla^2 f$ and $\nabla^2 g$ are respectively $l_{f,2}$ and $l_{g,2}$-Lipschitz continuous; and, iii) there exists a finite $\gamma^*>0$ such that $cf(x,y)+g(x,y)$ is μ-Polyak-Łojasiewicz (PL) in y for all $c\in[0,1/\gamma^*]$ for some $\mu>0$.*

We provide the definition of smoothness and PL in Appendix A. Here, the smoothness condition is standard [29, 34, 44, 38, 15, 40]. The Hessian Lipschitzness helps establish the smoothness of $F_\gamma(x)$ with constant nonincreasing with γ and is also conventional [45, 11, 19]. PL condition is weaker than the strong convexity assumption [15, 34, 29, 38, 19] and is conventional in the first-order BLO literature [45, 78, 11]. Under these, the penalty objective is differentiable [78, 45, 11, 63] and

$$\nabla F_\gamma(x) = \nabla_x f(x,y_\gamma^*(x)) + \gamma\left(\nabla_x g(x,y_\gamma^*(x)) - \nabla_x g(x,y_g^*(x))\right) \tag{5}$$

with $\forall y_g^*(x)\in S_g^*(x)$ and $\forall y_\gamma^*(x)\in S_\gamma^*(x) := \operatorname{argmin}_y F_\gamma(x,y)$. Moreover, the following proposition shows that $F_\gamma(x)$ is a good approximation of the original objective $\phi(x)$.

Proposition 1 (Approximation error [78, 45]**).** *Under Assumptions 1–2, for any x, we have*

$$\begin{aligned}\|F_\gamma(x)-\phi(x)\| &\le \mathcal{O}(l_{f,0}^2\mu^{-1}\gamma^{-1}), \quad \textit{and} \\ \|\nabla\phi(x)-\nabla F_\gamma(x)\| &= \mathcal{O}(\|y_g^*(x)-y_\gamma^*(x)\|) \le \mathcal{O}(l_{f,0}\mu^{-1}\gamma^{-1}).\end{aligned} \tag{6}$$

for some $y_g^*(x)$, $y_\gamma^*(x)$ defined in (5)*. Moreover, the bound for $\|y_g^*(x)-y_\gamma^*(x)\|$ is tight.*

Therefore, the PBGD type of algorithms [11, 44, 45, 105] proceed by approximating $y_t^\gamma \approx y_\gamma^*(x_t)$ and $y_t^g \approx y_g^*(x_t)$ via gradient descent and updating x via

$$x_{t+1} = x_t - \eta g_t \quad \text{where} \quad g_t = \nabla_x f(x,y_t^\gamma) + \gamma\left(\nabla_x g(x,y_t^\gamma) - \nabla_x g(x,y_t^g)\right). \tag{7}$$

Algorithm 1 PBGD Free from value function (PBGD-Free) algorithm

Inputs: initial point x_0, y_0; step sizes η, η^γ; counters T, K ▷ $K = 1$ is a common choice
for $t = 0, 1, \ldots, T-1$ **do**
  for $k = 0, 1, \ldots, K-1$ **do**
    $y^\gamma_{t,k+1} = y^\gamma_{t,k} - \eta^\gamma\big(\gamma^{-1}\nabla_y f(x_t, y^\gamma_{t,k}) + \nabla_y g(x_t, y^\gamma_{t,k})\big)$ ▷ set $y^\gamma_{t,0} = y^\gamma_{t-1}$
  end for
  $x_{t+1} = x_t - \eta g_t$, where $g_t = \nabla_x f(x_t, y^\gamma_t)$ ▷ set $y^\gamma_t = y^\gamma_{t,K}$
end for
Outputs: (x_T, y^γ_T)

2.2 Negative theoretical results of the PBGD-Free under Lipschitz condition

Although PBGD-type algorithms can achieve the state-of-the-art complexity $\mathcal{O}(\epsilon^{-1}\log(\epsilon^{-1}))$ in [11], their reliance on two inner loops can become computationally expensive for large-scale problems. While the overhead is manageable in small-scale settings, it may pose practical challenges as the model size grows. Nevertheless, empirical evidence in Figure 2 and real-world applications in Section 4 illustrate that it sometimes gives satisfactory results even if it directly updates x_{t+1} and $y^\gamma_{t,k}$ by

$$x_{t+1} = x_t - \eta\nabla_x f(x_t, y^\gamma_t) \quad \text{and} \quad y^\gamma_{t,k+1} = y^\gamma_{t,k} - \eta^\gamma(\gamma^{-1}\nabla_y f(x_t, y^\gamma_{t,k}) + \nabla_y g(x_t, y^\gamma_{t,k})) \tag{8}$$

which we name as PBGD-Free algorithm and is summarized in Algorithm 1.

Although PBGD-Free is computationally efficient by eliminating the inner loop estimates of $y^*_g(x)$, the removal of the value function part $b(x_t) := \gamma(\nabla_x g(x, y^*_\gamma(x)) - \nabla_x g(x, y^*_g(x)))$ in PBGD-Free introduces a non-negligible bias shown in Example 1. To see this, by Taylor's expansion, the omitted value function part $b(x_t)$ is in the order of

$$\|b(x_t)\| = \gamma\|\nabla_{xy} g(x, y^*_g(x))\big(y^*_\gamma(x) - y^*_g(x)\big)\| + \mathcal{O}(\gamma\|y^*_g(x) - y^*_\gamma(x)\|^2). \tag{9}$$

Here, the second term $\mathcal{O}(\gamma\|y^*_g(x) - y^*_\gamma(x)\|^2 = \mathcal{O}(l^2_{f,0}\gamma^{-1})$ can be small enough with enlarging γ following Proposition 1. For general settings where $\nabla_{xy} g(x, y^*_g(x)) \neq 0$, due to the first term and according to Proposition 1, the bias in (9) is tight as $\Omega(1)$. Therefore, in the general case where $\nabla_{xy} g(x, y^*_g(x)) \neq 0$, the PBGD-Free algorithm only drives the iterates to a neighborhood of the stationary point, which we will formally quantify as follows.

Proposition 2 (Lower bound on asymptotic error)**.** *Under Assumptions 1 and 2, there exists a BLO problem where the iterates generated by PBGD-Free (Algorithm 1) converge to a neighborhood of a stationary point with a non-vanishing residual even when choosing step sizes appropriately, i.e.,*

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla F_\gamma(x_t)\|^2 = \lim_{T\to\infty}\Theta\left(\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla_y f(x_t, y^*_g(x_t))\|^2\right) = \Theta\left(l^2_{f,0}\right). \tag{10}$$

The proof of Proposition 2 is available at Appendix A.1. Proposition 2 illustrates that PBGD-Free converges to the ϵ stationary point only when the bound for $\|\nabla_y f(x, y^*_g(x_t))\|$ (a.k.a $\ell_{f,0}$) is small. However, this is difficult to guarantee even in scenarios where PBGD-Free is effective, such as in representation learning based PEFT (3). This motivates us to explore a weaker condition than the small Lipschitz assumption on $f(x, \cdot)$, one that is more likely to hold in practice.

3 Theoretical Analysis under the (δ, α)-Flatness Condition

In this section, we will introduce a new relaxed condition to replace the widely used Lipschitz condition of the UL objective, discuss its use cases, and establish the convergence rate of the PBGD-Free algorithm under this condition.

3.1 A relaxed condition: UL (δ, α)-flatness and its validation on PEFT problem (3)

We first introduce a relaxed condition that is less restrictive, and therefore more general, than the conventional uniform Lipschitz assumption on $f(x, \cdot)$.

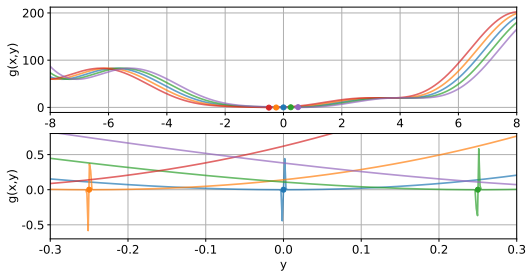

Figure 3: Visualization of $f(x, y)$ in Example 2 with details deferred to Appendix B.2. Colored curves represent $f(x, \cdot)$ for different x; dots show $(y_g^*(x), f(x, y_g^*(x)))$. The **upper** plot shows $f(x, \cdot)$ on a larger scale, and the **lower** one illustrates the fluctuation around $y_g^*(x)$.

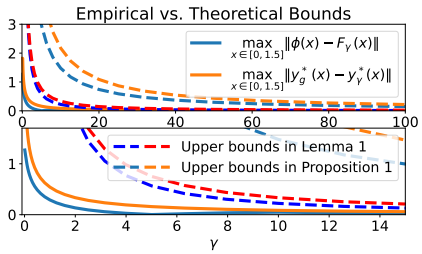

Figure 4: Empirical bounds for $\|\phi(x) - F_\gamma(x)\|$ and $\|y_\gamma^*(x) - y_g^*(x)\|$ versus theoretical upper bounds in Proposition 1 and Lemma 1 for the illustration of representation learning PEFT (3). The lower plot shows a smaller scale.

Definition 1 ((δ, α)-flatness)**.** *Function $f(x, \cdot) : \mathcal{Y} \to \mathbb{R}$ is called (δ, α)-flat with modulus $c \geq 0$ at $y_g^*(x) \in S_g^*(x)$ with $\delta \geq 0, \alpha \geq 1$ if $|f(x, y_g^*(x)) - f(x, y)| \leq c\|y_g^*(x) - y\|^\alpha + \delta$ holds for all y.*

When $\delta = 0$ and $y_g^*(x)$ is replaced by an arbitrary y', Definition 1 reduces to the standard Hölder condition. Under Assumption 1, the function $f(x, \cdot)$ satisfies $(0, 1)$-flatness with modulus $c = l_{f,0}$. However, setting $\delta = 0$ naturally imposes the constraint $\alpha \leq 1$ whenever $\nabla_y f(x, y_g^*(x)) \neq 0$. Unless otherwise specified, we assume $\delta > 0$ and $\alpha > 1$ when referring to flatness in the following.

We then discuss the relations of Lipschitz condition in Assumption 1 and the new flatness condition in Definition 1 through several observations.

Observation 1. *Under the $l_{f,1}$-smoothness condition of $f(x, \cdot)$, if $\|\nabla_y f(x, y_g^*(x))\| = \delta^{\frac{1}{\alpha}}$, then $f(x, \cdot)$ is (δ, α)-flat with some modulus $0 \leq c \leq \mathcal{O}(l_{f,1})$.*

The proof of Observation 1 is provided in Appendix B.1. It demonstrates that assuming small $\|\nabla_y f(x, y_g^*(x))\|$ is stronger than assuming flatness since $\ell_{f,0} = \delta^{\frac{1}{\alpha}} > \delta$ when $\alpha > 1$. Below, we will show that the flatness condition automatically holds near the LL optimal solution $y_g^*(x)$.

Observation 2. *Under the smoothness condition of f, the (δ, α)-flatness condition holds automatically for all $y \in \{y : |f(x, y) - f(x, y_g^*(x))| \leq \delta\}$.*

Since f is continuous and smooth, this observation implies that the flatness condition permits abrupt, unstable changes in the $\mathcal{O}(\delta)$-neighborhood of $y_g^*(x)$. This demonstrates that the flatness condition is relatively mild and further confirms that it is strictly weaker than the small Lipschitz condition, which explicitly requires $\|\nabla_y f(x, y_g^*(x))\|$ to be small. Figure 3 visualize an example that is $(3e^{-3}, 1.1)$-flat with modulus $c = 5$ at $y_g^*(x)$, but it exhibits a sharp change leading to a large Lipschitz continuity constant $\nabla_y f(x, y_g^*(x)) = 1000$. The details of Figure 3 are deferred to Appendix B.2.

Owing to the Hölder-alike condition, the following observation shows that outside of the $\mathcal{O}(\delta)$ neighborhood, the curvature of the flatness condition is also milder than the Lipschitz condition.

Observation 3. *Under (δ, α)-flatness, the growth rate of $f(x, \cdot)$ outside the $\mathcal{O}(\delta)$ neighborhood is*

$$\frac{|f(x, y) - f(x, y_g^*(x))|}{\|y - y_g^*(x)\|} \leq \begin{cases} \mathcal{O}(1), & \text{if } \mathcal{O}(\delta) \leq \|y - y_g^*(x)\| \leq \mathcal{O}(1), \\ \mathcal{O}\left(\|y - y_g^*(x)\|^{\alpha-1}\right), & \text{if } \|y - y_g^*(x)\| > \mathcal{O}(1). \end{cases} \tag{11}$$

This is obtained by dividing both sides of the flatness inequality by $\|y_g^*(x) - y\|$. For small $\|y_g^*(x) - y\|$, the second term dominates and leads to a $\mathcal{O}(1)$ bound, which is the same as the Lipschitz condition. However, for large $\|y_g^*(x) - y\|$, since $\alpha > 1$, the bound $\mathcal{O}(\|y_g^*(x) - y\|^{\alpha-1})$ can be larger than $\mathcal{O}(1)$. This observation further demonstrates that the flatness condition relaxes the Lipschitzness of $f(x, \cdot)$ in Assumption 1. Specifically, while Lipschitz continuity would require a uniform bound on the gradient, flatness allows for a higher growth rate of $\mathcal{O}(\|y - y_g^*(x)\|^{\alpha-1})$. For UL objective $f(x, \cdot)$ with fixed x, given a *pre-determined α and modulus c*, the δ constant for flatness condition in Definition 1 can be calculated via

$$\delta(x) := \max\{0, |f(x, y_g^*(x)) - f(x, y_\gamma^*(x))| - c\|y_g^*(x) - y_\gamma^*(x)\|^\alpha\}. \tag{12}$$

When $\|y_\gamma^*(x) - y_g^*(x)\| > \mathcal{O}(1)$, the last term in (12) dominates and $\delta(x)$ can effectively be 0. Therefore, together with Observation 2, the flatness condition with small δ not only encompasses a broader function class than small Lipschitz continuous functions, but is easier to hold in practice. For example, modern loss functions used in deep learning, such as cross-entropy, squared error, or exponential losses, are nonlinear and locally curved. Around $y_g^*(x)$, we can write $f(x, y') \approx f(x, y_g^*(x)) + c\|y' - y_g^*(x)\|^\alpha$ for some $\alpha > 1$ and constant $c > 0$. In such cases, the additive term in (12) vanishes and $\delta(x)$ is effectively zero. This implies that the flatness condition can hold even when no Lipschitz bound on $f(x, \cdot)$ is available, particularly for locally curved objectives. We next illustrate this behavior concretely through a parameter-efficient fine-tuning (PEFT) problem in representation learning.

3.2 The flatness of the representation learning PEFT problem.

In our PEFT framework in (3), the model, which can be any structure (e.g. CNN), is parameterized with (x, y) by $\pi_{x,y}(r|z) := \text{softmax}(\text{model}_{x,y}(z))_r$. It gives the model's predicted probability for response r given input question z. The DPO loss [70] over preference data $\mathcal{D}_{\text{DPO}}$, compares outputs $\pi_{x,y}$ against a reference π_{ref} via

$$f_{\text{DPO}}(x, y) := -\frac{1}{|\mathcal{D}_{\text{DPO}}|} \sum_{(z, r_w, r_\ell) \in \mathcal{D}_{\text{DPO}}} \log\left(\sigma\left(q_\beta(x, y; z, r_w, r_\ell)\right)\right), \tag{13}$$

where $q_\beta(x, y; z, r_w, r_\ell) := \beta \log \frac{\pi_{x,y}(r_w|z)}{\pi_{\text{ref}}(r_w|z)} - \beta \log \frac{\pi_{x,y}(r_\ell|z)}{\pi_{\text{ref}}(r_\ell|z)}$, r_w and r_ℓ are the preferred and rejected responses to input z. The SFT loss operates on supervised dataset $\mathcal{D}_{\text{SFT}}$ through

$$g_{\text{SFT}}(x, y) := -\frac{1}{|\mathcal{D}_{\text{SFT}}|} \sum_{(z, r_{\text{SFT}}) \in \mathcal{D}_{\text{SFT}}} \log\left(\pi_{x,y}(r_{\text{SFT}}|z)\right). \tag{14}$$

Both objectives are differentiable with the following gradients

$$\nabla f_{\text{DPO}} = -(1 - \sigma(q_\beta))\nabla q_\beta, \quad \nabla g_{\text{SFT}} = -\nabla \pi / \pi. \tag{15}$$

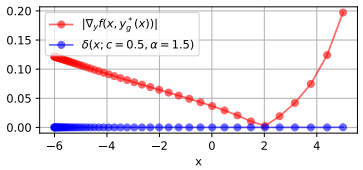

Figure 5: Comparisons of $\delta(x)$ and $\nabla_y f(x, y_g^*(x))$ during PBGD-Free updates. The Lipschitz constant $\ell_{f,0} = \max_x \|\nabla_y f(x, y_g^*(x))\|$ is large but $\delta(x)$ is small.

While the Lipschitz constant for this problem is large, it satisfies the flatness condition with small δ. To illustrate, we revisit the example in Figure 2; see the detailed setting in Appendix C.1. As in Figure 5, the flatness constant $\delta(x) \leq 0.0003$ in the blue line is small throughout optimization for $c = 0.5$ and $\alpha = 1.5$, despite a large Lipschitz constant in red. This confirms that the loss landscape analysis under the Lipschitz condition is not tight, as $l_{f,0}$ remains non-negligible even in local neighborhoods, whereas the flatness condition allows for a tighter analysis. The small $\delta(x)$ values along the PBGD-Free trajectory validate that *the PEFT problem* (3) *satisfies the flatness condition*, which inspired us to establish the enhanced analysis of the PBGD-Free algorithm under the flatness condition. In real-world PEFT problems, e.g. ones in Section 4, $\delta(x)$ in (12) is typically small because the distance between $y_g^*(x)$ and $y_\gamma^*(x)$ are non-negligible, whereas their impact on f_{DPO} is marginal.

3.3 Convergence analysis for the PBGD-Free algorithm

As shown in (9), the first term is the major bottleneck of the divergence issue of the PBGD-Free algorithm under the Lipschitz condition in Proposition 2. The key to establishing the convergence guarantee for the PBGD-Free algorithm is the tighter bound of $\|y_g^*(x) - y_\gamma^*(x)\|$ and $\|\phi(x) - F_\gamma(x)\|$ under the flatness condition, compared to the results in Lemma 1. We highlight the results as follows.

Lemma 1 (Tighter analysis on function value gap). *Suppose Assumption 2.(iii) holds. For fixed x, suppose $f(x, \cdot)$ is (δ, α)-flat at any $y_g^*(x) \in S_g^*(x)$ with $\alpha \in (1, 1.5]$. Then, there exists $\gamma^* > 0$ such that for $\gamma \geq \gamma^*$, we have*

$$\|\phi(x) - F_\gamma(x)\| = \mathcal{O}(\gamma^{-\frac{\alpha}{2-\alpha}} + \delta), \text{ and } d_{S_\gamma^*(x)}(y_g), d_{S_g^*(x)}(y_\gamma) = \mathcal{O}\big(\gamma^{-\frac{1}{2-\alpha}} + \delta^{\frac{1}{2}}\gamma^{-\frac{1}{2}}\big), \tag{16}$$

for any $y_g \in S_g^*(x)$ and $y_\gamma \in S_\gamma^*(x)$.

The proof of Lemma 1 is available at Appendix B.3. When δ is smaller than target accuracy ϵ, achieving $\|\phi(x) - F_\gamma(x)\|, \|y_g^*(x) - y_\gamma^*(x)\|^2 = \mathcal{O}(\epsilon)$ only requires $\gamma = \mathcal{O}(\epsilon^{-\frac{2-\alpha}{2}})$, which is

strictly smaller than the choice of $\gamma = \mathcal{O}(\epsilon^{-\frac{1}{2}})$ in previous literature [78, 10, 45, 44]. This also aligns with common practice, where the penalty constant γ does not need to be excessively large. For instance, the UL objective in Example 2 is $(10^{-3}, 1.1)$-flat and therefore choosing $\gamma = 15$ gives desired accuracy, supporting the rule of thumb: $\gamma \approx 15$ is a reasonable choice. In Figure 4, we also show that our bound under the flatness condition in Lemma 1 is tighter than the one under the Lipschitz condition in Proposition 1 for the representation learning PEFT (3).

Since Lemma 1 provides a per-iterate bound with fixed x, the next step is to analyze the Lipschitz continuity of the flatness constant $\delta(x)$ with respect to x, enabling a uniform bound across iterations.

Lemma 2 (Lipschitz continuity of flatness constant $\delta(x)$). *Suppose Assumption 2 holds. Then fixing some $c \geq 0$ and $\alpha \in (1, 2)$, there exists some trajectory of $y_g^*(x)$, $y_\gamma^*(x)$ such that the flatness constant of $f(x, \cdot)$, $\delta(x)$ defined in* (12)*, is $\mathcal{O}(c\gamma^{-(\alpha-1)})$-Lipschitz-continuous in x.*

The proof of the Lemma 2 is available in Appendix B.4. However, Lemma 1 and Lemma 2 only enable the convergence of PBGD-Free to the stationary point of the penalized objective $F_\gamma(x)$. We next establish the approximate equivalence of the stationary points to the original BLO problem (1).

Lemma 3 (Approximate equivalence of stationary points). *Suppose Assumption 2 holds. Let x^* be an ϵ stationary point of $F_\gamma(x^*)$ and suppose $f(x^*, \cdot)$ is (δ, α)-flat at any $y_g^*(x^*) \in S_g^*(x^*)$ with $\alpha \in (1, 1.5]$ and $\delta \leq \mathcal{O}(\epsilon^{\frac{\alpha}{2}})$. Then there exists $\gamma^* = \mathcal{O}(\epsilon^{-\frac{2-\alpha}{2}})$ and $y_\gamma \in S_\gamma^*(x^*)$ such that for $\gamma \geq \gamma^*$, (x^*, y_γ) is the $\mathcal{O}(\epsilon)$ stationary point of the original BLO problem* (1)*.*

The proof of the Lemma 3 is available in Appendix B.5, which generalizes the definition of stationary condition for (1) [91, 105] and its relations to that of the penalty problem [78, 45, 105] under flatness condition instead of Lipschitz continuity in Assumption 1. In this way, building upon Lemma 1 and Lemma 2, the convergence result for PBGD-Free in Algorithm 1 is stated as follows.

Theorem 3 (Convergence of PBGD-Free). *Suppose Assumption 2 holds, and for all x_t on the trajectory, $f(x_t, \cdot)$ is $(\delta(x_t), \alpha)$-flat at all $y_g(x_t) \in S_g^*(x_t)$ with the same $\alpha \in (1, 1.5]$ and modulus $c = \mathcal{O}(1)$. For iterations generated by Algorithm 1 with $K = 1$ and step size $\eta \leq l_{F,1}^{-1}$, where $l_{F,1}$ is the smoothness constant of $F_\gamma(x)$, and suppose for target accuracy ϵ, there exists δ such that $\frac{1}{T}\sum_{t=0}^{T-1} \delta(x_t) \leq \delta$, then by choosing $\gamma = \mathcal{O}(\delta^{-\frac{2-\alpha}{2}})$,*

$$\frac{1}{T}\sum_{t=0}^{T-1} \|\nabla F_\gamma(x_t)\|^2 \leq \mathcal{O}(T^{-1} + \delta^{\frac{2(\alpha-1)}{\alpha}}). \tag{17}$$

The proof of Theorem 3 is provided in Appendix B.6. Here, the smoothness constant $l_{F,1}$ is not scalable with γ [11], therefore leading to a constant step size choice. Theorem 3 establishes the convergence rate of the fully-single-loop version of PBGD-Free in Algorithm 1. The result shows that the algorithm converges to the neighborhood of a stationary point for $F_\gamma(x)$, where the stationary gap is controlled by the flatness parameter (δ, α). Specifically, for a (δ, α)-flat function with $\alpha \in (1, 1.5)$, the convergence error scales as $\mathcal{O}(\delta^{\frac{2(\alpha-1)}{\alpha}})$, ensuring that the suboptimality gap remains small. For instance, for the PEFT problem in (3), $\delta(x)$ is often negligible, as per the discussion in Section 3.2. Moreover, the method follows a single-loop update scheme, which is computationally more efficient than other fully first-order methods [44, 45, 105, 78, 10], as elaborated in the Appendix B.7. A comparison of the proposed algorithm with state-of-the-art fully first-order BLO methods is provided in Table 1.

4 Numerical Experiments

In this section, we empirically validate our theoretical results through experiments on real-world tasks. In the main paper, we will focus on the LLM PEFT problem (3). Additional experiments, including fair representation learning problem on the NLSY-7k dataset [73, 80], and BiDORA fine-tuning [68], are provided in Appendix C.

4.1 Representation learning based LLM PEFT and its flatness

SFT enhances pre-trained LLMs for downstream task adaptation, whereas DPO aligns them with human preferences. A straightforward way to achieve both goals is to sequentially optimize the

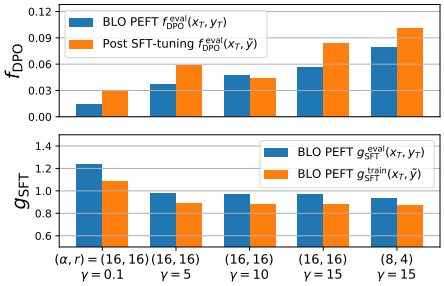

Figure 6: Ablation study on penalty γ and LoRA configuration [35] for PYTHIA-1b [6].

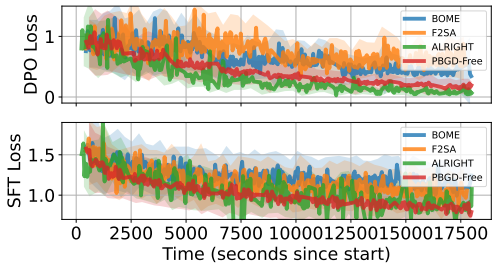

Figure 7: Train losses vs. time for different algorithms in solving (3) (or bi-objective learning for ALRIGHT) on LLAMA-3-3B [31].

Methods	$f_{\text{DPO}}^{\text{eval}}(x_T, y_T)$	$g_{\text{SFT}}^{\text{eval}}(x_T, y_T)$	$f_{\text{DPO}}^{\text{eval}}(x_T, \tilde{y})$	$g_{\text{SFT}}^{\text{train}}(x_T, \tilde{y})$
V-PBGD [45]	0.818	1.0309	0.8423	0.9533
BOME [105]	0.8332	1.1552	0.8402	0.9842
ALRIGHT [24]	0.8055	0.8656	0.8201	0.7855
PBGD-Free	**0.7837**	**0.8516**	**0.8088**	**0.6688**

Table 2: Comparison of different algorithms for PEFT LLAMA-3-3B [31]. Results show the DPO Loss ↓, SFT Loss ↓ for both the outcome (x_T, y_T) trained on solving (3) for different methods or and the outcome $(x_T, \tilde{y})$ from post-SFT-tuning on another dataset with fixed-backbone, using the same dataset fixed time of training for each.

two objectives. However, this often leads to catastrophic forgetting [24], where applying DPO after SFT overwrites task-specific knowledge. To address this, we adopt the bilevel framework (3), which prioritizes SFT at the LL to create a more reliable base model and applies DPO at the UL to guide the human preference alignment. This hierarchical formulation effectively optimizes DPO conditioned on a near-optimal SFT, thereby preserving downstream task performance. Moreover, this design is natural, as user preferences are generally consistent across tasks due to underlying psychological regularities. Additionally, the proposed BLO framework aligns with the post-SFT paradigm, where DPO is fine-tuned from a pre-trained model and SFT is applied to downstream tasks. In practical settings, it is common to fine-tune only a lightweight head while keeping the backbone fixed or lightly updated [67, 106, 72].

In this paper, we adopt a decomposition of LLM into a backbone model x (e.g., attention weights) and an output head y to formulate a BLO PEFT framework (3). Our method conforms to the PEFT practice by allowing the head to specialize in SFT tasks while training the backbone through DPO to capture generalizable preference representations. In our experiments, we adopt the low rank adaptation (LoRA) [35] to the backbone x for PEFT on LLAMA-3-3B [31] and PYTHIA-1B [6], using the `Dahoas/rm-hh-rlhf` dataset for DPO and the `OpenOrca` dataset [55] for SFT. Our code is adapted from the bilevel LLM post-training library `https://github.com/Post-LLM/BIPOST` and experiment details are referred to Appendix C.3. As preliminarily demonstrated in Figure 5, this *BLO PEFT problem* in (3) features flatness (small δ), which is further corroborated by the observation in experiment that the LL solution $y_g^*(x)$ and $y_\gamma^*(x)$ have ℓ_2- distance greater than 1, suggesting a negligible flatness constant $\delta(x)$ by (12).

4.2 Ablation study and main experimental results for the PEFT problem (3)

In this experiment, we consider evaluating methods on both **S1)** *BLO PEFT learning phase via* (3) to obtain a preference backbone x, and **S2)** *post-SFT tuning on a new dataset* with the obtained preference backbone model x, to verify the representation quality and transferability of the backbone.

We first conduct an ablation study on the PYTHIA-1b to test the impact of the penalty constant γ and LoRA configuration on the PBGD-Free method. We report the DPO and SFT loss under different settings for both (x_T, y_T) learned from **S1)** and $(x_T, \tilde{y})$ from **S2)** in Figure 6.

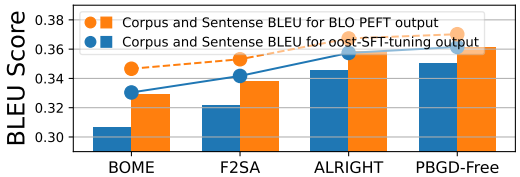

Figure 8: BLEU-4 Corpus and BLEU-4 Sentence Score (↑) for different algorithms for PEFT on LLAMA-3-3B [31].

Trade-off between DPO and SFT under different γ. According to Figure 6, increasing γ degrades DPO performance while improving SFT for **S1)**, indicating that a larger γ provides better LL optimality. Notably, the SFT improvement beyond $\gamma = 10$ is marginal for **S1)**, while the DPO performance significantly deteriorates, suggesting that $\gamma \approx 10$ offers the best balance as our theory predicts.

Faster convergence over BLO baselines and stable training over bi-objective. Since second-order BLO algorithms are inefficient in large-scale LLM training, we consider first-order methods F^{2}SA [45] and BOME [105] as BLO baselines. As shown in Figure 7, PBGD-Free converges faster than the BLO baselines. We additionally compare with ALRIGHT [24], an effective bi-objective algorithm, to validate the representation capability of *BLO PEFT* (3) formulation. ALRIGHT [24] exhibits less stability during training (Figure 7), likely due to alternating between DPO and SFT objectives.

Transferability of preference backbone and strong SFT performance. Compared with **S1)**, PBGD-Free in Figure 6 shows enhanced SFT with comparable DPO performance on **S2)**, suggesting it learns a transferable preference backbone x through BLO (3). Table 2 further quantifies these findings for other baselines, demonstrating that PBGD-Free achieves superior DPO and SFT performance. Notably, the backbone model x obtained by PBGD-Free attains the lowest SFT and DPO loss on **S2)**, verifying the transferability of PBGD-Free. To further evaluate the quality of generated output, Figure 8 corroborates the SFT performance using the evaluation metrics BLEU score [65], where our method outperforms all baselines, further justifying its superiority in learning a good representation. More experimental results, including semantic analysis (Table 6) are provided in Appendix C.3.

5 Concluding remarks

In this paper, we propose PBGD-Free, a penalty-based method for efficiently solving the nonconvex BLO problem without solving the value-function subproblem of $y_g^*(x)$. We first show that, under a general Lipschitz condition, the convergence of PBGD-Free has a constant lower bound by the Lipschitz constant, which does not vanish unless the Lipschitz constant is sufficiently small. Motivated by empirical findings in representation learning, we then introduce a Hölder-like condition and prove that, when its constant is sufficiently small, the fully single-loop PBGD-Free algorithm achieves an iteration complexity of $\mathcal{O}(\epsilon^{-1})$. We further demonstrate that this Hölder-like condition with a small constant is strictly weaker than the small Lipschitz condition, and we verify this condition in representation-learning-based LLM PEFT, fair representation learning, and BiDORA fine-tuning. Numerical experiments in the above problems demonstrate that the PBGD-Free algorithm is computationally efficient and can outperform the existing baselines across all three applications.

References

[1] Momin Abbas, Quan Xiao, Lisha Chen, Pin-Yu Chen, and Tianyi Chen. Sharp-maml: Sharpness-aware model-agnostic meta learning. In *International conference on machine learning*, pages 10–32, 2022.

[2] Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. Muppet: Massive multi-task representations with pre-finetuning. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, 2021.

[3] Michael Arbel and Julien Mairal. Non-convex bilevel games with critical point selection maps. In *Proc. Advances in Neural Information Processing Systems*, New Orleans, LA, 2022.

[4] Sanjeev Arora, Simon Du, Sham Kakade, Yuping Luo, and Nikunj Saunshi. Provable representation learning for imitation learning via bi-level optimization. In *International Conference on Machine Learning*, pages 367–376. PMLR, 2020.

[5] Guillaume O Berger, P-A Absil, Raphaël M Jungers, and Yurii Nesterov. On the quality of first-order approximation of functions with hölder continuous gradient. *Journal of Optimization Theory and Applications*, 185:17–33, 2020.

[6] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, 2023.

[7] Jérôme Bolte, Edouard Pauwels, and Samuel Vaiter. Automatic differentiation of nonsmooth iterative algorithms. In *Advances in Neural Information Processing Systems*, 2022.

[8] Zalán Borsos, Mojmir Mutny, and Andreas Krause. Coresets via bilevel optimization for continual learning and streaming. In *Advances in Neural Information Processing Systems*, virtual, 2020.

[9] Sébastien Bubeck. *Convex Optimization: Algorithms and Complexity*. Foundations and Trends® in Machine Learning, Boston, MA, USA, 2015.

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

[11] Lesi Chen, Jing Xu, and Jingzhao Zhang. On finding small hyper-gradients in bilevel optimization: Hardness results and improved analysis. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 947–980. PMLR, 2024.

[12] Lisha Chen, Sharu Theresa Jose, Ivana Nikoloska, Sangwoo Park, Tianyi Chen, and Osvaldo Simeone. Learning with limited samples – meta-learning and applications to communication systems. *Foundations and Trends in Signal Processing*, 1 2023.

[13] Lisha Chen, Quan Xiao, Ellen Hidemi Fukuda, Xinyi Chen, Kun Yuan, and Tianyi Chen. Efficient first-order optimization on the pareto set for multi-objective learning under preference guidance. *arXiv preprint arXiv:2504.02854*, 2025.

[14] Tianyi Chen, Yuejiao Sun, Quan Xiao, and Wotao Yin. A single-timescale method for stochastic bilevel optimization. In *Proc. International Conference on Artificial Intelligence and Statistics*, 2022.

[15] Tianyi Chen, Yuejiao Sun, and Wotao Yin. Closing the gap: Tighter analysis of alternating stochastic gradient methods for bilevel problems. In *Advances in Neural Information Processing Systems*, Virtual, 2021.

[16] Ziyi Chen, Bhavya Kailkhura, and Yi Zhou. A fast and convergent proximal algorithm for regularized nonconvex and nonsmooth bi-level optimization. *arXiv preprint arXiv:2203.16615*, 2022.

[17] Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations*, 2024.

[18] Frank H Clarke. *Optimization and nonsmooth analysis*. SIAM, Philadelphia, PA, 1990.

[19] Mathieu Dagréou, Pierre Ablin, Samuel Vaiter, and Thomas Moreau. A framework for bilevel optimization that enables stochastic and global variance reduction algorithms. In *Advances in Neural Information Processing Systems*, 2022.

[20] Bill Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Third international workshop on paraphrasing (IWP2005)*, 2005.

[21] Youran Dong, Junfeng Yang, Wei Yao, and Jin Zhang. Efficient curvature-aware hypergradient approximation for bilevel optimization. *arXiv preprint arXiv:2505.02101*, 2025.

[22] Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. Llama-omni: Seamless speech interaction with large language models. *arXiv preprint arXiv:2409.06666*, 2024.

[23] Sheng Fang, Yong-Jin Liu, Wei Yao, Chengming Yu, and Jin Zhang. qnbo: quasi-newton meets bilevel optimization. In *Proc. International Conference on Learning Representations*, 2025.

[24] Heshan Fernando, Han Shen, Parikshit Ram, Yi Zhou, Horst Samulowitz, Nathalie Baracaldo, and Tianyi Chen. Mitigating forgetting in llm supervised fine-tuning and preference learning. *arXiv preprint arXiv:2410.15483*, 2024.

[25] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *Proc. International Conference on Learning Representations*, virtual, 2021.

[26] Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. Forward and reverse gradient-based hyperparameter optimization. In *International Conference on Machine Learning*, pages 1165–1173. PMLR, 2017.

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

[28] Hongchang Gao, Bin Gu, and My T Thai. On the convergence of distributed stochastic bilevel optimization algorithms over a network. In *International conference on artificial intelligence and statistics*, pages 9238–9281. PMLR, 2023.

[29] Saeed Ghadimi and Mengdi Wang. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*, 2018.

[30] Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In *Proc. International Conference on Machine Learning*, pages 2232–2241, 2019.

[31] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[32] Riccardo Grazzi, Luca Franceschi, Massimiliano Pontil, and Saverio Salzo. On the iteration complexity of hypergradient computation. In *International Conference on Machine Learning*, pages 3748–3758, virtual, 2020.

[33] Jie Hao, Kaiyi Ji, and Mingrui Liu. Bilevel coreset selection in continual learning: A new formulation and algorithm. In *Advances in Neural Information Processing Systems*, New Orleans, LA, 2023.

[34] Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A two-timescale stochastic algorithm framework for bilevel optimization: Complexity analysis and application to actor-critic. *SIAM Journal on Optimization*, 33(1):147–180, 2023.

[35] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[36] Yifan Hu, Jie Wang, Yao Xie, Andreas Krause, and Daniel Kuhn. Contextual stochastic bilevel optimization. In *Proc. Advances in Neural Information Processing Systems*, 2023.

[37] Kaiyi Ji, Mingrui Liu, Yingbin Liang, and Lei Ying. Will bilevel optimizers benefit from loops. *arXiv preprint arXiv:2205.14224*, 2022.

[38] Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. In *International conference on machine learning*, pages 4882–4892. PMLR, 2021.

[39] Kaiyi Ji, Junjie Yang, and Yingbin Liang. Provably faster algorithms for bilevel optimization and applications to meta-learning. In *Advances in Neural Information Processing Systems*, 2021.

[40] Liuyuan Jiang, Quan Xiao, Victor M Tenorio, Fernando Real-Rojas, Antonio Marques, and Tianyi Chen. A primal-dual-assisted penalty approach to bilevel optimization with coupled constraints. In *Advances in Neural Information Processing Systems*, 2024.

[41] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I 16*, pages 795–811, 2016.

[42] Prashant Khanduri, Siliang Zeng, Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A near-optimal algorithm for stochastic bilevel optimization via double-momentum. In *Advances in Neural Information Processing Systems*, Virtual, 2021.

[43] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint:1412.6980*, December 2014.

[44] Jeongyeol Kwon, Dohyun Kwon, Stephen Wright, and Robert D Nowak. A fully first-order method for stochastic bilevel optimization. In *International Conference on Machine Learning*, pages 18083–18113, 2023.

[45] Jeongyeol Kwon, Dohyun Kwon, Steve Wright, and Robert Nowak. On penalty methods for nonconvex bilevel optimization and first-order stochastic approximation. In *International Conference on Learning Representations*, 2024.

[46] Haochuan Li, Alexander Rakhlin, and Ali Jadbabaie. Convergence of adam under relaxed assumptions. In *Proc. Advances in Neural Information Processing Systems*, New Orleans, LA, 2023.

[47] Junyi Li, Bin Gu, and Heng Huang. A fully single loop algorithm for bilevel optimization without hessian inverse. In *Association for the Advancement of Artificial Intelligence*, pages 7426–7434, virtual, 2022.

[48] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

[49] Risheng Liu, Jiaxin Gao, Jin Zhang, Deyu Meng, and Zhouchen Lin. Investigating bilevel optimization for learning and vision from a unified perspective: A survey and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):10045–10067, 2021.

[50] Risheng Liu, Xuan Liu, Xiaoming Yuan, Shangzhi Zeng, and Jin Zhang. A value-function-based interior-point method for non-convex bi-level optimization. In *International conference on machine learning*, pages 6882–6892, 2021.

[51] Risheng Liu, Xuan Liu, Shangzhi Zeng, Jin Zhang, and Yixuan Zhang. Value-function-based sequential minimization for bi-level optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[52] Risheng Liu, Yaohua Liu, Wei Yao, Shangzhi Zeng, and Jin Zhang. Averaged method of multipliers for bi-level optimization without lower-level strong convexity. In *Proc. International Conference on Machine Learning*, Honolulu, HI, 2023.

[53] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353*, 2024.

[54] Xu-Hui Liu, Yali Du, Jun Wang, and Yang Yu. On the optimization landscape of low rank adaptation methods for large language models. In *Proc. International Conference on Learning Representations*, 2025.

[55] Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. The flan collection: Designing data and methods for effective instruction tuning. In *Proc. International Conference on Machine Learning*, 2023.

[56] Han Lu, Yichen Xie, Xiaokang Yang, and Junchi Yan. Boundary matters: A bi-level active finetuning method. In *Advances in Neural Information Processing Systems*, 2024.

[57] Songtao Lu. Tsp: A two-sided smoothed primal-dual method for nonconvex bilevel optimization. In *Forty-second International Conference on Machine Learning*, 2025.

[58] Zhaosong Lu and Sanyou Mei. First-order penalty methods for bilevel optimization. *SIAM Journal on Optimization*, 34(2):1937–1969, 2024.

[59] Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150, 2011.

[60] Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *International Conference on Machine Learning*, pages 2113–2122, Lille, France, 2015.

[61] Pierre Marion, Anna Korba, Peter Bartlett, Mathieu Blondel, Valentin De Bortoli, Arnaud Doucet, Felipe Llinares-López, Courtney Paquette, and Quentin Berthet. Implicit diffusion: Efficient optimization through stochastic sampling. In *International Conference on Artificial Intelligence and Statistics*, 2025.

[62] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, Boston, MA, 2013.

[63] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[64] Rui Pan, Jipeng Zhang, Xingyuan Pan, Renjie Pi, Xiaoyu Wang, and Tong Zhang. Scalebio: Scalable bilevel optimization for llm data reweighting. *arXiv preprint arXiv:2406.19976*, 2024.

[65] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[66] Ieva Petrulionytė, Julien Mairal, and Michael Arbel. Functional bilevel optimization for machine learning. In *Proc. Advances in Neural Information Processing Systems*, Vancouver, Canada, 2024.

[67] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*, 2020.

[68] Peijia Qin, Ruiyi Zhang, and Pengtao Xie. Bidora: Bi-level optimization-based weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2410.09758*, 2024.

[69] Zhen Qin, Zhuqing Liu, Songtao Lu, Yingbin Liang, and Jia Liu. Duet: Decentralized bilevel optimization without lower-level strong convexity. In *The Thirteenth International Conference on Learning Representations*, 2025.

[70] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

[71] Zaccharie Ramzi, Florian Mannel, Shaojie Bai, Jean-Luc Starck, Philippe Ciuciu, and Thomas Moreau. Shine: Sharing the inverse estimate from the forward pass for bi-level optimization and implicit models. In *Proc. International Conference on Learning Representations*, 2022.

[72] Yi Ren, Shangmin Guo, Wonho Bae, and Danica J Sutherland. How to prepare your task head for finetuning. In *International Conference on Learning Representations*, 2023.

[73] Donna S Rothstein, Deborah Carr, and Elizabeth Cooksey. Cohort profile: The national longitudinal survey of youth 1979 (nlsy79). *International journal of epidemiology*, 48(1):22–22e, 2019.

[74] Benjamin Scellier and Yoshua Bengio. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in computational neuroscience*, 11:24, 2017.

[75] 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.

[76] Han Shen, Pin-Yu Chen, Payel Das, and Tianyi Chen. Seal: Safety-enhanced aligned llm fine-tuning via bilevel data selection. In *International Conference on Learning Representations*, 2025.

[77] Han Shen and Tianyi Chen. A single-timescale analysis for stochastic approximation with multiple coupled sequences. *Advances in Neural Information Processing Systems*, 35:17415–17429, 2022.

[78] Han Shen, Quan Xiao, and Tianyi Chen. On penalty-based bilevel gradient descent method. In *International Conference on Machine Learning*, Honolulu, HI, 2023.

[79] Han Shen, Zhuoran Yang, and Tianyi Chen. Principled penalty-based methods for bilevel reinforcement learning and RLHF. In *International Conference on Machine Learning*, Vienna, Austria, 2024.

[80] Changjian Shui, Qi Chen, Jiaqi Li, Boyu Wang, and Christian Gagné. Fair representation learning through implicit path alignment. In *International Conference on Machine Learning*, pages 20156–20175. PMLR, 2022.

[81] Daouda Sow, Kaiyi Ji, Ziwei Guan, and Yingbin Liang. A constrained optimization approach to bilevel optimization with multiple inner minima. *arXiv preprint arXiv:2203.01123*, 2022.

[82] Daouda Sow, Kaiyi Ji, and Yingbin Liang. On the convergence theory for hessian-free bilevel algorithms. In *Advances in Neural Information Processing Systems*, volume 35, pages 4136–4149, 2022.

[83] James H Stock and Francesco Trebbi. Retrospectives: Who invented instrumental variable regression? *Journal of Economic Perspectives*, 17(3):177–194, 2003.

[84] Mao Tan, Zhuocen Dai, Yongxin Su, Caixue Chen, Ling Wang, and Jie Chen. Bi-level optimization of charging scheduling of a battery swap station based on deep reinforcement learning. *Engineering Applications of Artificial Intelligence*, 118:105557, 2023.

[85] Vinzenz Thoma, Barna Pásztor, Andreas Krause, Giorgia Ramponi, and Yifan Hu. Contextual bilevel reinforcement learning for incentive alignment. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[86] Paul Vicol, Jonathan Lorraine, Fabian Pedregosa, David Duvenaud, and Roger Grosse. On implicit bias in overparameterized bilevel optimization. In *Proc. International Conference on Machine Learning*, 2022.

[87] Xiaoyu Wang, Xuxing Chen, Shiqian Ma, and Tong Zhang. Fully first-order methods for decentralized bilevel optimization. *arXiv preprint arXiv:2410.19319*, 2024.

[88] Rachel Ward and Tamara Kolda. Convergence of alternating gradient descent for matrix factorization. In *Proc. Advances in Neural Information Processing Systems*, New Orleans, LA, 2023.

[89] Per-Åke Wedin. Perturbation theory for pseudo-inverses. *BIT Numerical Mathematics*, 13:217–232, 1973.

[90] Quan Xiao and Tianyi Chen. Unlocking global optimality in bilevel optimization: A pilot study. In *Proc. International Conference on Learning Representations*, 2025.

[91] Quan Xiao, Songtao Lu, and Tianyi Chen. A generalized alternating method for bilevel optimization under the polyak-łojasiewicz condition. In *Advances in Neural Information Processing Systems*, New Orleans, LA, 2023.

[92] Quan Xiao, Han Shen, Wotao Yin, and Tianyi Chen. Alternating implicit projected sgd and its efficient variants for equality-constrained bilevel optimization. In *International Conference on Artificial Intelligence and Statistics*, 2023.

[93] Quan Xiao, Hui Yuan, AFM Saif, Gaowen Liu, Ramana Kompella, Mengdi Wang, and Tianyi Chen. A first-order generative bilevel optimization framework for diffusion models. *arXiv preprint arXiv:2502.08808*, 2025.

[94] Di Xie, Jiang Xiong, and Shiliang Pu. All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6176–6185, 2017.

[95] Liyuan Xu, Heishiro Kanagawa, and Arthur Gretton. Deep proxy causal learning and its application to confounded bandit policy evaluation. *Advances in Neural Information Processing Systems*, 34:26264–26275, 2021.

[96] Changdi Yang, Pu Zhao, Yanyu Li, Wei Niu, Jiexiong Guan, Hao Tang, Minghai Qin, Bin Ren, Xue Lin, and Yanzhi Wang. Pruning parameterization with bi-level optimization for efficient semantic segmentation on the edge. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[97] Junjie Yang, Kaiyi Ji, and Yingbin Liang. Provably faster algorithms for bilevel optimization. *arXiv preprint arXiv:2106.04692*, June 2021.

[98] Yifan Yang, Hao Ban, Minhui Huang, Shiqian Ma, and Kaiyi Ji. Tuning-free bilevel optimization: New algorithms and convergence analysis. In *The Thirteenth International Conference on Learning Representations*, 2025.

[99] Yifan Yang, Peiyao Xiao, Shiqian Ma, and Kaiyi Ji. First-order federated bilevel learning. In *Proc. of the AAAI Conference on Artificial Intelligence*, volume 39, pages 22029–22037, 2025.

[100] Wei Yao, Chengming Yu, Shangzhi Zeng, and Jin Zhang. Constrained bi-level optimization: Proximal lagrangian value function approach and hessian-free algorithm. *arXiv preprint arXiv:2401.16164*, 2024.

[101] Can Yaras, Peng Wang, Laura Balzano, and Qing Qu. Compressible dynamics in deep overparameterized low-rank learning & adaptation. In *Proc. International Conference on Machine Learning*, Vienna, Austria, 2024.

[102] Jane J. Ye. Constraint qualifications and necessary optimality conditions for optimization problems with variational inequality constraints. *SIAM Journal on Optimization*, 10(4):943–962, 2000.

[103] Jane J Ye, Xiaoming Yuan, Shangzhi Zeng, and Jin Zhang. Difference of convex algorithms for bilevel programs with applications in hyperparameter selection. *Mathematical Programming*, 198(2):1583–1616, 2023.

[104] Jane J Ye, Daoli Zhu, and Qiji Jim Zhu. Exact penalization and necessary optimality conditions for generalized bilevel programming problems. *SIAM Journal on optimization*, 7(2):481–507, 1997.

[105] Mao Ye, Bo Liu, Stephen Wright, Peter Stone, and Qiang Liu. Bome! bilevel optimization made easy: A simple first-order approach. In *Proc. Advances in Neural Information Processing Systems*, New Orleans, LA, 2022.

[106] Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*, 2021.

[107] Emanuele Zangrando, Sara Venturini, Francesco Rinaldi, and Francesco Tudisco. debora: Efficient bilevel optimization-based low-rank adaptation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[108] Haifeng Zhang, Weizhe Chen, Zeren Huang, Minne Li, Yaodong Yang, Weinan Zhang, and Jun Wang. Bi-level actor-critic for multi-agent coordination. In *Proc. of the AAAI Conference on Artificial Intelligence*, 2020.

[109] Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *Proc. International Conference on Learning Representations*, virtual, 2020.

[110] Weipu Zhang, Gang Wang, Jian Sun, Yetian Yuan, and Gao Huang. Storm: Efficient stochastic transformer based world models for reinforcement learning. *Advances in Neural Information Processing Systems*, 36:27147–27166, 2023.

[111] Yihua Zhang, Yuguang Yao, Parikshit Ram, Pu Zhao, Tianlong Chen, Mingyi Hong, Yanzhi Wang, and Sijia Liu. Advancing model pruning via bi-level optimization. In *Advances in Neural Information Processing Systems*, 2022.

[112] Yushun Zhang, Congliang Chen, Tian Ding, Ziniu Li, Ruoyu Sun, and Zhiquan Luo. Why transformers need adam: A hessian perspective. In *Proc. Advances in Neural Information Processing Systems*, Vancouver, Canada, 2024.

[113] Nicolas Zucchet and João Sacramento. Beyond backpropagation: bilevel optimization through implicit differentiation and equilibrium propagation. *Neural Computation*, 34(12):2309–2346, 2022.

Supplementary Material for "Beyond Value Functions: Single-Loop Bilevel Optimization under Flatness Conditions"

Table of Contents

A Preliminaries

Notations. We define $v(x) = \min_z g(x,z)$ and $v_\gamma(x) = \min_y \gamma^{-1} f(x,y) + g(x,y)$. We denote $S_g^*(x) = \arg\min_z g(x,z)$, $S_\gamma^*(x) = \arg\min_y \gamma^{-1} f(x,y) + g(x,y)$, $d_S(y) := \min_{z\in S} \|y - z\|$.

Definition 2 (Lipschitz Continuity and Smoothness)**.** *We say a function $f(x,y)$ is $l_{f,0}$-Lipschitz if*

$$\|f(x_1,y_1) - f(x_2,y_2)\| \le l_{f,1}\|[x_1;y_1] - [x_2;y_2]\|, \quad \forall (x_1,y_1),(x_2,y_2) \tag{18}$$

If f is differentiable, we say f is $l_{f,1}$-smooth on if ∇f is $l_{f,1}$-Lipschitz, i.e. $\forall (x_1,y_1),(x_2,y_2)$:

$$\|[\nabla_x f(x_1,y_1) - \nabla_x f(x_2,y_2); \nabla_y f(x_1,y_1) - \nabla_y f(x_2,y_2)]\| \le l_{f,1}\|[x_1;y_1] - [x_2;y_2]\|. \tag{19}$$

Definition 3 (PL condition)**.** *We say $g(x,y)$ satisfies μ-Polyak-Łojasiewicz (PL) condition in y if*

$$\|\nabla_y g(x,y)\| \ge 2\mu(g(x,y) - v(x)). \tag{20}$$

Lemma 4 ([41, Theorem 2])**.** *If $g(x,y)$ is $\ell_{g,1}$-Lipschitz smooth and PL in y with μ_g, then it satisfies the error bound (EB) condition with μ_g, i.e.*

$$\|\nabla_y g(x,y)\| \ge \mu_g d_{S_g^*(x)}(y). \tag{21}$$

Moreover, it also satisfies the quadratic growth (QG) condition with μ_g, i.e.

$$g(x,y) - v(x) \ge \frac{\mu_g}{2} d_{S_g^*(x)}(y)^2. \tag{22}$$

Conversely, if $g(x,y)$ is $\ell_{g,1}$-Lipschitz smooth and satisfies EB with μ_g, then it is PL in y with $\mu_g/\ell_{g,1}$.

Proposition 4 (Complete version to Proposition 1 [78, 45])**.** *Under Assumption 1–2, for any x, there is*

$$\|F_\gamma(x) - \phi(x)\| \le \mathcal{O}(\|\nabla_y f(x, y_g^*(x))\|^2 \mu^{-1}\gamma^{-1}) = \mathcal{O}(l_{f,0}^2 \mu^{-1}\gamma^{-1}). \tag{23}$$

Additionally, for any $y_g^*(x) \in S_g^*(x)$, $y_\gamma^*(x) \in S_\gamma^*(x)$,

$$d_{S_\gamma^*(x)}(y_g^*(x)), d_{S_g^*(x)}(y_\gamma^*(x)) \le \Omega(\|\nabla_y f(x, y_g^*(x))\|\mu^{-1}\gamma^{-1}) = \Omega(l_{f,0}\mu^{-1}\gamma^{-1}). \tag{24}$$

Moreover, for $y_g^*(x) = \arg\min_{z\in S_g^*(x)} f(x,z)$, there is

$$\|\nabla\phi(x) - \nabla F_\gamma(x)\| = \mathcal{O}\left(d_{S_\gamma^*(x)}(y_g^*(x))\right) = \mathcal{O}(l_{f,0}\mu^{-1}\gamma^{-1}).$$

A.1 Proof of Proposition 2

First of all, Algorithm 1 can be viewed as a biased PBGD algorithm with bias being

$$\begin{aligned}\|b_t\| =&\|\nabla F_\gamma(x_t) - \nabla_x f(x_t, y_t^\gamma)\| \\ \overset{(a)}{=}&\|\nabla_x f(x_t, y_\gamma^*(x)) + \gamma(\nabla_x g(x_t, y_\gamma^*(x_t)) - \nabla_x g(x_t, y_g^*(x_t))) - \nabla_x f(x_t, y_t^\gamma)\| \\ \overset{(b)}{\leq}&\|\nabla_x f(x_t, y_\gamma^*(x)) - \nabla_x f(x_t, y_t^\gamma)\| + \gamma\|\nabla_x g(x_t, y_\gamma^*(x_t)) - \nabla_x g(x_t, y_g^*(x_t))\| \\ \overset{(c)}{\leq}&l_{f,1}\|y_\gamma^*(x_t) - y_t^\gamma\| + \gamma l_{g,1}\|y_\gamma^*(x_t) - y_g^*(x_t)\| \\ \overset{(d)}{\leq}&l_{f,1}\sqrt{\gamma^{-1}f(x_t, y_t^\gamma) - v_\gamma(x_t)} + \mathcal{O}(\|\nabla_y f(x, y_g^*(x_t))\|) \\ \overset{(e)}{\leq}&l_{f,1}\sqrt{(1-\eta^\gamma\mu)^K(\gamma^{-1}f(x_t, y_{t-1}^\gamma) - v_\gamma(x_t))} + \mathcal{O}(\|\nabla_y f(x, y_g^*(x_t))\|)\end{aligned}$$

where (a) is by plugging in $\nabla F_\gamma(x_t)$ in (5), this holds for arbitrary $y_g^*(x)$, $y_\gamma(x)$ as solutions to problems in (5); (b) follows triangle-inequality; (c) uses the smoothness of f and g; the first term in (d) is obtained by the QG property as ensured by PL condition and smoothness as per Lemma 4, via choosing $y_\gamma^*(x_t) = \arg\min_{y\in S_\gamma^*(x_t)} \|y - y_t^\gamma\|$, and the second term follows Proposition 4 by choosing $y_g^*(x) = \arg\min_{z\in S_g^*(x_t)} \|y_\gamma^*(x_t) - z\|$; and (e) follows the linear convergence result of PL function [41] as y_t^γ is the results from K-step inner update starting at y_{t-1}^γ. In this way, when taking K sufficiently large, there is $\|b_t\| \leq \mathcal{O}(\|\nabla_y f(x, y_g^*(x_t))\| = \mathcal{O}(l_{f,0})$.

Moreover, according to [11], $F_\gamma(x)$ is $\mathcal{O}(1)$-smooth. Therefore,

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla F_\gamma(x_t)\|^2 \leq& \mathcal{O}(T^{-1}) + \sum_{t=0}^{T-1}\|b_t\|^2 \\ \leq& \mathcal{O}(T^{-1}) + \mathcal{O}\Big(\frac{1}{T}\sum_{t=0}^{T}\|\nabla_y f(x, y_g^*(x_t))\|^2\Big) = \mathcal{O}(T^{-1}) + \mathcal{O}(l_{f,0}^2). \quad (25)\end{aligned}$$

In this way, for sufficiently large T, $\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla F_\gamma(x)\|^2 \leq \mathcal{O}(l_{f,0}^2)$.

Next, we will prove the lower bound of Algorithm 1 by constructing a counterexample, and show that the upper bound is tight. Consider $f(x, y) = x^2 + l_{f,0}y$ and $g(x, y) = (y - x + 1)^2$. In this problem, $\nabla F_\gamma(x) = 2x + l_{f,0}$ while $\nabla_x f(x, y_\gamma^*(x)) = 2x$. Using fixed stepsize η, $x_{t+1} = x_t - \eta x_t = (1-\eta)x_t$, implying $\|\nabla_x f(x_{t+1}, y_\gamma^*(x_{t+1}))\| = \|2x_{t+1}\| = 2(1-\eta)\|x_t\| = 2(1-\eta)^{t+1}\|x_0\|$. Therefore, for arbitrary small $\epsilon > 0$, there exists some $T_0 = \mathcal{O}(\ln(\epsilon^{-1}))$ such that Algorithm 1 converges to $\|x_t\| < \epsilon$ for all $t \geq T_0$, whereas $\nabla F_\gamma(x_t) = l_{f,0}$. In this way, we have

$$\frac{1}{T - T_0}\sum_{t=T_0}^{T}\|\nabla F_\gamma(x_t)\|^2 = \mathcal{O}(\epsilon) + l_{f,0}^2. \quad (26)$$

This indicates $\Omega(l_{f,0}^2)$ is a tight bound.

B Improved Analysis under Flatness

B.1 Proof of Observation 1

For $\|y - y_g^*(x)\| \geq 1$, by the $l_{f,0} = \delta^{\frac{1}{\alpha}}$-Lipschitzness of $f(x, y)$ in y, there is

$$\|f(x, y) - f(x, y_g^*(x))\| \leq l_{f,0}\|y - y_g^*(x)\| \leq \delta^{\frac{1}{\alpha}}\|y - y_g^*(x)\|^\alpha. \quad (27)$$

For small $\|y - y_g^*(x)\| < 1$, Taylor's expansion gives

$$f(x, y) - f(x, y_g^*(x)) = \langle\nabla_y f(x, y_g^*(x)), y - y_g^*(x)\rangle + R(x, y). \quad (28)$$

Here, $R(x,y)$ is a remainder. By Hölder-Continuous Gradient Condition [5], [62, Section 2], which is implied by the smoothness, there exists some $0 \leq c \leq l_{f,1}/2$, $1 < \alpha < 2$ such that $\|R(x,y)\| \leq c\|y - y_g^*(x)\|^\alpha$. By Cauchy-Schwartz's inequality, there is

$$\begin{aligned}\|\langle \nabla_y f(x, y_g^*(x)), y - y_g^*(x)\rangle\| \leq & \|\nabla_y f(x, y_g^*(x))\| \|y - y_g^*(x)\| \\ \leq & \delta^{\frac{1}{\alpha}} \|y - y_g^*(x)\| \\ \leq & \delta + \|y - y_g^*(x)\|^\alpha \end{aligned} \tag{29}$$

where the last inequality holds as for $a, b \in (0,1)$ and $\alpha \in (1,2)$, there is $ab \leq \max\{a,b\}^2 \leq \max\{a,b\}^\alpha \leq a^\alpha + b^\alpha$ and here $a = \delta^{1/\alpha}$ and $b = \|y - y_g^*(x)\|$. The observation therefore, holds.

B.2 Detailed example for Observation 2

The following example visualizes Observation 2.

Example 2. *We consider the LL objective $g(x,y) = (y-x)^2$ and the UL objective*

$$f(x,y) = (\sin(y-x)+2)\,|y-x|^2 + 10\exp\left(-\frac{(y-x)^2}{2(0.005)^2}\right)\sin\left(100(y-x)\right).$$

The LL problem $g(x,y)$ is strongly convex in y, with $y_g^*(x) = x$. Therefore, $\nabla_y f(x, y_g^*(x)) = 1000$ is extremely large, which leads to a loose upper bound for $\|y_g^*(x) - y_\gamma^*(x)\|$ or $\|\phi(x) - F_\gamma(x)\|$ following Lemma 1. However, this problem is $(3e^{-3}, 1.1)$-flat with $c = 5$ at $y_g^*(x) = x$ for $x \in [-10, 10]$. As shown in Figure 3, $f(x,\cdot)$ exhibits fluctuations around $y_g^*(x)$ while remaining relatively stable elsewhere. This shows that flatness is weaker than requiring small $\|\nabla_y f(x,y)\|$.

B.3 Proof of Lemma 1

Proof. For any $y_g^*(x) \in S_g^*(x)$, $y_\gamma^*(x) \in S_\gamma^*(x)$, there is

$$\begin{aligned} & \gamma^{-1} f(x, y_\gamma^*(x)) + g(x, y_\gamma^*(x)) \leq \gamma^{-1} f(x, y_g^*(x)) + g(x, y_g^*(x)) \\ \Rightarrow \quad & \gamma^{-1} f(x, y_\gamma^*(x)) + g(x, y_\gamma^*(x)) - v(x) \leq \gamma^{-1} f(x, y_g^*(x)) + g(x, y_g^*(x)) - v(x) \\ \Rightarrow \quad & \gamma^{-1} f(x, y_\gamma^*(x)) + g(x, y_\gamma^*(x)) - v(x) \leq \gamma^{-1} f(x, y_g^*(x)) \\ & \Rightarrow \quad f(x, y_\gamma^*(x)) \leq f(x, y_g^*(x)). \end{aligned} \tag{30}$$

In this way, according to the definition of $\phi(x)$ and $F_\gamma(x)$, we have

$$\begin{aligned} \|\phi(x) - F_\gamma(x)\| = & \min_{z \in S_g^*(x)} f(x,z) - \left(f(x, y_\gamma^*(x)) + \gamma(g(x, y_\gamma^*(x)) - v(x))\right) \\ \overset{(a)}{\leq} & f(x, y_g^*(x)) - \left(f(x, y_\gamma^*(x)) + \gamma(g(x, y_\gamma^*(x)) - v(x))\right) \\ \overset{(b)}{\leq} & f(x, y_g^*(x)) - \left(f(x, y_\gamma^*(x)) + \gamma\frac{\mu_g}{2}\|y_g^*(x) - y_\gamma^*(x)\|^2\right) \\ \overset{(c)}{=} & \|f(x, y_g^*(x)) - f(x, y_\gamma^*(x))\| - \gamma\frac{\mu_g}{2}\|y_g^*(x) - y_\gamma^*(x)\|^2 \\ \overset{(d)}{\leq} & c\|y_g^*(x) - y_\gamma^*(x)\|^\alpha + \delta - \gamma\frac{\mu}{2}\|y_g^*(x) - y_\gamma^*(x)\|^2 \\ \overset{(e)}{\leq} & \max_{z: z \geq 0} cz^\alpha - \gamma\frac{\mu}{2}z^2 + \delta \\ \overset{(f)}{=} & c^{\frac{2}{2-\alpha}}(2\alpha)^{\frac{\alpha}{2-\alpha}}(1-\alpha/2)(\mu\gamma)^{-\frac{\alpha}{2-\alpha}} + \delta = \mathcal{O}(\gamma^{-\frac{\alpha}{2-\alpha}} + \delta) \end{aligned} \tag{31}$$

Here, (a) holds for arbitrary $y_g^*(x) \in S_g^*(x), y_\gamma^*(x) \in S_\gamma^*(x)$ by (30); (b) is from the μ_g quadratic growth condition of $g(x,\cdot)$ which is implied by μ_g-PL according to Lemma 4, via choosing $y_g^*(x) = \arg\min_{z \in S_g^*(x)} \|z - y_\gamma^*(x)\|$; (c) again uses (30); (d) follows the flatness of $f(x,y)$ at $y_g^*(x)$, (e) is by formulating the problem as a maximization problem over $z = \|y_g^*(x) - y_\gamma^*(x)\|$; and (f) is the solution to this polynomial problem. Therefore, the first part is proved.

For the second part, as $\frac{1}{\gamma}f(x,\cdot)+g(x,\cdot)$ being μ-PL for $\gamma>\gamma^*$, it is also μ-QG by Lemma 4. In this way, fixed any $\gamma>\gamma^*$, for any $y_\gamma^*(x)\in S_\gamma^*(x)$ and any $y_g^*(x)\in S_g^*(x)$, there is

$$\gamma\left(\left(\frac{1}{\gamma}f(x,y_g^*(x))+g(x,y_g^*(x))\right)-\left(\frac{1}{\gamma}f(x,y_\gamma^*(x))+g(x,y_\gamma^*(x))\right)\right)\geq\gamma\frac{\mu}{2}d^2_{S_\gamma^*(x)}(y_g^*(x)). \tag{32}$$

Moreover, following steps (a)-(d) as in (31), there is

$$\begin{aligned}\text{left of (32)}=&\big(f(x,y_g^*(x))+\gamma g(x,y_g^*(x))-\gamma v(x)\big)-\big(f(x,y_\gamma^*(x))+\gamma g(x,y_\gamma^*(x))-\gamma v(x)\big)\\ =&f(x,y_g^*(x))-f(x,y_\gamma^*(x))-\gamma\big(g(x,y_\gamma^*(x))-v(x)\big)\\ \leq&c\|y_g^*(x)-y_\gamma^*(x)\|^\alpha+\delta-\gamma\frac{\mu}{2}d^2_{S_\gamma^*(x)}(y_\gamma^*(x))\end{aligned} \tag{33}$$

Combining (32) and the above, there is

$$c\|y_g^*(x)-y_\gamma^*(x)\|^\alpha+\delta-\gamma\frac{\mu}{2}d^2_{S_g^*(x)}(y_\gamma^*(x))\geq\gamma\frac{\mu}{2}d^2_{S_\gamma^*(x)}(y_g^*(x)) \tag{34}$$

for any $y_g^*(x)\in S_g^*(x)$ and $y_\gamma^*(x)\in S_\gamma^*(x)$. In this way, for any $y_g^*(x)\in S_g^*(x)$, choose $y_\gamma^*(x)=\arg\min_{y\in S_\gamma^*(x)}\|y-y_g^*(x)\|$, there is

$$cd^\alpha_{S_\gamma^*(x)}(y_g^*(x))+\delta\geq\gamma\frac{\mu}{2}d^2_{S_\gamma^*(x)}(y_g^*(x)) \tag{35}$$

Similarly, for any $y_\gamma^*(x)\in S_\gamma^*(x)$, choose $y_g^*(x)=\arg\min_{z\in S_g^*(x)}\|z-y_g^*(x)\|$, there is

$$cd^\alpha_{S_g^*(x)}(y_\gamma^*(x))+\delta\geq\gamma\frac{\mu}{2}d^2_{S_g^*(x)}(y_\gamma^*(x)). \tag{36}$$

For simplicity, denote $x=d_{S_g^*(x)}(y_\gamma^*(x))$ (or $x=d_{S_\gamma^*(x)}(y_g^*(x))$), there is

$$x^{2-\alpha}\leq2c\mu^{-1}\gamma^{-1}+2\delta\mu^{-1}\gamma^{-1}x^{-\alpha}. \tag{37}$$

As $\alpha\in(1,1.5]$, for $x\geq\sqrt{\frac{\delta}{\gamma}}$,

$$x^{2-\alpha}\leq2c\mu^{-1}\gamma^{-1}+2\delta\mu^{-1}\gamma^{-1}\left(\frac{\delta}{\gamma}\right)^{-\frac{\alpha}{2}}. \tag{38}$$

Since $|a+b|^p\leq2^{p-1}(|a|^p+|b|^p)$ for all $p\geq1$ (as $|\cdot|^p$ is convex), there is

$$\begin{aligned}x=&(x^{2-\alpha})^{\frac{1}{2-\alpha}}\leq\left(2c\mu^{-1}\gamma^{-1}+2\delta\mu^{-1}\gamma^{-1}\left(\frac{\delta}{\gamma}\right)^{-\frac{\alpha}{2}}\right)^{\frac{1}{2-\alpha}}\\ \leq&2^{\frac{1}{2-\alpha}-1}\left((2c\mu^{-1})^{\frac{1}{2-\alpha}}\gamma^{-\frac{1}{2-\alpha}}+(2\mu^{-1})^{\frac{1}{2-\alpha}}\delta^{\frac{1}{2}}\gamma^{-\frac{1}{2}}\right)=Oc(\gamma^{-\frac{1}{2-\alpha}}+\delta^{\frac{1}{2}}\gamma^{-\frac{1}{2}})\end{aligned} \tag{39}$$

In this way, we can conclude the following to include the scenario that $x\leq\sqrt{\frac{\delta}{\gamma}}$.

$$x=\mathcal{O}(\gamma^{-\frac{1}{2-\alpha}}+\delta^{\frac{1}{2}}\gamma^{-\frac{1}{2}}). \tag{40}$$

□

B.4 Proof of Lemma 2

Define

$$\begin{aligned}\delta'(x):=&|f(x,y_g^*(x))-f(x,y_\gamma^*(x))|-c\|y_g^*(x)-y_\gamma^*(x)\|^\alpha\\ =&f(x,y_g^*(x))-f(x,y_\gamma^*(x))-c\|y_g^*(x)-y_\gamma^*(x)\|^\alpha.\end{aligned} \tag{41}$$

where the equality is from $f(x,y_g^*(x))\geq f(x,y_\gamma^*(x))$ as per (30). We firstly show that $f(x,y_g^*(x))-f(x,y_\gamma^*(x))$ and $c\|y_g^*(x)-y_\gamma^*(x)\|^\alpha$ are both Lipschitz continuous.

For $f(x, y_g^*(x)) - f(x, y_\gamma^*(x))$, according to [11, Lemma F.3], there is

$$\begin{aligned}
&\Big\| \frac{\partial}{\partial x} \Big[f(x, y_g^*(x)) - f(x, y_\gamma^*(x)) \Big] \Big\| \\
=&\big\| \nabla_x f(x, y_g^*(x)) - \nabla_x f(x, y_\gamma^*(x)) + \nabla_y f(x, y_g^*(x)) \nabla_{yy} g(x, y_g^*(x))^\dagger \nabla_{yx} g(x, y_g^*(x)) \\
&- \nabla_y f(x, y_\gamma^*(x)) \left[\gamma^{-1} \nabla_{yy} f(x, y_\gamma^*(x)) + \nabla_{yy} g(x, y_\gamma^*(x)) \right]^\dagger \\
&\quad \times \left[\gamma^{-1} \nabla_{yx} f(x, y_\gamma^*(x)) + \nabla_{yx} g(x, y_\gamma^*(x)) \right] \big\| \\
\leq&\big\| E_1 \| + \| \nabla_y f(x, y_g^*(x)) \nabla_{yy} g(x, y_g^*(x))^\dagger \nabla_{yx} g(x, y_g^*(x)) \\
&- [\nabla_y f(x, y_g^*(x)) + E_2][\nabla_{yy} g(x, y_g^*(x)) + E_3]^\dagger [\nabla_{yx} g(x, y_g^*(x)) + E_4] \| \qquad (42)
\end{aligned}$$

where the inequality is by the triangle inequality and by denoting

$$\begin{cases}
E_1 = & \nabla_x f(x, y_\gamma^*(x)) - \nabla_x f(x, y_g^*(x)) \\
E_2 = & \nabla_y f(x, y_\gamma^*(x)) - \nabla_y f(x, y_g^*(x)) \\
E_3 = & \gamma^{-1} \nabla_{yy} f(x, y_\gamma^*(x)) + \nabla_{yy} g(x, y_\gamma^*(x)) - \nabla_{yy} g(x, y_g^*(x)) \\
E_4 = & \gamma^{-1} \nabla_{yx} f(x, y_\gamma^*(x)) + \nabla_{yx} g(x, y_\gamma^*(x)) - \nabla_{yx} g(x, y_g^*(x))
\end{cases} \qquad (43)$$

By the smoothness of f, the Lipschitzness of $\nabla^2 g$ and by Proposition 1, we know that

$$\|E_1\|, \|E_2\|, \|E_3\|, \|E_4\| = \mathcal{O}(\gamma^{-1}). \qquad (44)$$

Additionally, according to [89], we know

$$\|[\nabla_{yy} g(x, y_g^*(x)) + E_3]^\dagger - \nabla_{yy} g(x, y_g^*(x))^\dagger\| \leq \frac{1+\sqrt{5}}{\mu} \|E_3\| = \mathcal{O}(\gamma^{-1}). \qquad (45)$$

In this way, by the smoothness of f and g, we know $\|\gamma^{-1} \nabla^2 f + \nabla^2 g\| \leq \gamma^{-1} l_{f,1} + l_{g,1}$ and therefore,

$$\Big\| \frac{\partial}{\partial x} \Big[f(x, y_g^*(x)) - f(x, y_\gamma^*(x)) \Big] \Big\| \leq (42) \leq \mathcal{O}(\gamma^{-1}). \qquad (46)$$

This shows that $f(x, y_g^*(x)) - f(x, y_\gamma^*(x))$ is Lipschitz-continuous.

Fix any x, denote arbitrary $y_g^*(x) \in S_g^*(x)$, $y_\gamma^*(x) \in S_\gamma^*(x)$, then for any $x' \in \mathcal{X}$, there exists some $y_g^*(x') \in S_g^*(x')$, $y_\gamma^*(x') \in S_\gamma^*(x')$ such that

$$\begin{aligned}
&\big| c\|y_g^*(x) - y_\gamma^*(x)\|^\alpha - c\|y_g^*(x') - y_\gamma^*(x')\|^\alpha \big| \\
\overset{(a)}{\leq}& c \max_{z \in [\|y_g^*(x) - y_\gamma^*(x)\|, \|y_g^*(x') - y_\gamma^*(x')\|]} z^{\alpha-1} \big| \|y_g^*(x) - y_\gamma^*(x)\| - \|y_g^*(x') - y_\gamma^*(x')\| \big| \\
\overset{(b)}{\leq}& \mathcal{O}(c\gamma^{-(\alpha-1)}) \big\| \big(y_g^*(x) - y_\gamma^*(x)\big) - \big(y_g^*(x') - y_\gamma^*(x')\big) \big\| \\
\overset{(c)}{\leq}& \mathcal{O}(c\gamma^{-(\alpha-1)}) \big(\|y_g^*(x) - y_g^*(x')\| + \|y_\gamma^*(x) - y_\gamma^*(x')\| \big) \\
\overset{(d)}{=}& \mathcal{O}(c\gamma^{-(\alpha-1)}) \|x - x'\| \qquad (47)
\end{aligned}$$

where (a) follows the mean value theorem, as $|\cdot|^\alpha$ is continuous; (b) is from $\|y_g^*(x) - y_\gamma^*(x)\| = \mathcal{O}(\gamma^{-1})$, and the 1-Lipschitzness of the norm function; (c) uses triangle-inequality; and (d) is achieved by knowing that $y_g^*(x)$ and $y_\gamma^*(x)$ are, respectively, L_g, L_γ-Lipschitz for some constant L_g, L_γ [78].

In this way, we can conclude that $\delta'(x)$ is $\mathcal{O}(c\gamma^{-(\alpha-1)})$ Lipschitz continuous. As $\delta(x)$ is a ReLu function works on $\delta'(x)$, it is also $\mathcal{O}(c\gamma^{-(\alpha-1)})$ Lipschitz continuous.

B.5 Stationary relations under flatness condition

Similar to [91], we first derive the stationary conditions for the original BLO problem (1), under the flatness condition in Definition 1 instead of the Lipschitz continuity. Then we prove the stationary equivalence of the penalty problem with the original BLO problem (1).

B.5.1 Stationary conditions for original BLO problem (1)

First, the original BLO problem can be equivalently written as its gradient based constrained form under LL PL condition as follows [105, 91].

$$\min_{x,y} \ f(x,y), \quad \text{s.t.} \quad \nabla_y g(x,y) = 0. \tag{48}$$

We aim to show that the Karush–Kuhn–Tucker conditions (KKT) conditions of (48) are necessary for the global optimality of the original BLO problem (1), thereby serving as its stationary conditions. Prior works [104, 102] have discussed that under the calmness condition, the KKT condition is a necessary optimality condition. Similar to [104, 102, 91, 13], we will prove that the calmness condition holds for (48) for our problem, even under our relaxed assumptions, so that KKT conditions, which our proposed algorithm converges to, are necessary for the optimality in the original BLO problem (1). Notably, the key difference is that the prior works require a global [91] or local [13] Lipschitz condition of the upper-level objective, while we prove this under a more relaxed flatness condition of the upper-level objective. This makes the result applicable to a much wider set of problems. We first review the definition of the calmness condition below.

Definition 4 (Calmness, [18, Definition 6.4.1]). *Let (x^*, y^*) be the global minimizer of the constrained problem*

$$\min_{x,y} \ f(x,y) \quad \text{s.t.} \ \ f_c(x,y) = 0. \tag{49}$$

where $f_c : \mathbb{R}^{d_x+d_y} \to \mathbb{R}^d$ and $d \geq 1$. If there exist positive ϵ and M such that for any $q \in \mathbb{R}^d$ with $\|q\| \leq \epsilon$ and any $\|(x', y') - (x^*, y^*)\| \leq \epsilon$ which satisfies $f_c(x', y') + q = 0$, one has

$$f(x', y') - f(x^*, y^*) + M\|q\| \geq 0 \tag{50}$$

then the problem (49) *is said to be calm with M and ϵ.*

We will prove a general version for establishing that the KKT conditions of problem (48) serve as the stationary conditions of the BLO problem (1), which only requires the UL objective to be continuously differentiable.

Lemma 5. *Suppose that $g(x, \cdot)$ satisfies the PL condition and is smooth, and $f(x, \cdot)$ is continuously differentiable. For the global minimizer (x^*, y^*) of BLO problem in* (1) *(a.k.a* (48)*), then* (48) *is calm at its global minimizer (x^*, y^*), and therefore, the KKT conditions of* (48) *are the necessary conditions of the global optimality in* (1)*.*

Proof. First, for any x, since $f(x, \cdot)$ is continuously differentiable, then $f(x, \cdot)$ is locally Lipschitz continuous around any y', i.e. there exists a neighborhood of $\mathbb{B}_\epsilon(y') := \{y : \|y - y'\| \leq \epsilon\}$ and constant $L := \max_{y \in \mathbb{B}_\epsilon(y')} \|\nabla_y f(x,y)\| < \infty$ such that $f(x, \cdot)$ is Lipschitz continuous with constant L over $y \in \mathbb{B}_\epsilon(y')$.

Then consider $\forall q$ with $\|q\| \leq \epsilon$, and $\forall x' \in \mathbb{B}_\epsilon(x^*)$ and y', s.t. $\nabla_y g(x', y') + q = 0$, then letting $y_q \in \mathrm{Proj}_{S_g^*(x')}(y')$ and according to Lemma 4, one has

$$\epsilon \geq \|q\| = \|\nabla_y g(x', y')\| \geq \mu_g \|y' - y_q\|$$

Since (x^*, y^*) solves (48) and (x', y_q) is also feasible to (48), one has $f(x^*, y^*) \leq f(x', y_q)$. Thus,

$$f(x', y') - f(x^*, y^*) \geq f(x', y') - f(x', y_q) \overset{(a)}{\geq} -L\|y' - y_q\| \geq -L\|q\| \tag{51}$$

where (a) is due to the local Lipschitz continuity of $f(x, \cdot)$ with $L := \max_{y \in \mathbb{B}_\epsilon(y')} \|\nabla_y f(x,y)\|$.

(51) justifies the calmness definition in (50) with $M := \frac{L}{\mu_g}$ and ϵ. □

Therefore, under Assumption 2, the smoothness of $f(x, \cdot)$ implies that $f(x, \cdot)$ is continuously differentiable so that Lemma 5 holds. Note that Lemma 5 also generalizes the results in [91, 13] by relaxing the global/local Lipschitz continuity assumption on $f(x, \cdot)$ via continuously differentiable (ensuring local Lipschitz continuity).

We then aim to prove in Lemma 3 that the stationary point of the penalty reformulation (2) is approximately the stationary point of the original BLO problem in (1) (i.e., the KKT point of (48)).

B.5.2 Proof of Lemma 3

Proof. Let x^* be the stationary point of the penalty problem (2), then we have $\|\nabla F_\gamma(x^*)\| \leq \epsilon$. Then according to (5), for $\forall y_\gamma \in S_\gamma^*(x^*)$ and $\forall y_g \in S_g^*(x^*)$, we have

$$\|\nabla_x f(x^*, y_\gamma) + \gamma(\nabla_x g(x^*, y_\gamma) - \nabla_x g(x^*, y_g))\|^2 \leq \epsilon \tag{52a}$$
$$\|\nabla_y f(x^*, y_\gamma) + \gamma(\nabla_y g(x^*, y_\gamma) - \nabla_y g(x^*, y_g))\|^2 \leq \epsilon. \tag{52b}$$

We aim to prove that (x^*, y_γ) is approximately the KKT point of (48), i.e. $\exists$ finite $w \in \mathbb{R}^{d_y}$, s.t.

$$\|\nabla_x f(x^*, y_\gamma) + \nabla_{xy} g(x^*, y_\gamma) w\|^2 \leq \mathcal{O}(\epsilon) \tag{53a}$$
$$\|\nabla_y f(x^*, y_\gamma) + \nabla_{yy} g(x^*, y_\gamma) w\|^2 \leq \mathcal{O}(\epsilon) \tag{53b}$$
$$\|\nabla_y g(x^*, y_\gamma)\|^2 \leq \mathcal{O}(\epsilon). \tag{53c}$$

The approximate LL optimality in (53c) is earned by Lemma 1, which gives

$$d_{S_g^*(x)}(y_\gamma) = \mathcal{O}(\gamma^{-\frac{1}{2-\alpha}} + \delta^{\frac{1}{2}}\gamma^{-\frac{1}{2}}) = \mathcal{O}(\epsilon^{0.5} + \delta^{0.5}\epsilon^{\frac{2-\alpha}{4}})$$

when $\gamma = \mathcal{O}(\epsilon^{-\frac{2-\alpha}{2}})$. Therefore, when $\delta \leq \epsilon^{\frac{\alpha}{2}}$, it holds that

$$\|\nabla_y g(x^*, y_\gamma)\|^2 \leq \mathcal{O}(d^2_{S_g^*(x)}(y_\gamma)) = \mathcal{O}(\epsilon + \delta\epsilon^{\frac{2-\alpha}{2}}) \leq \mathcal{O}(\epsilon) \tag{54}$$

where the first inequality is earned by the smoothness condition. Moreover, by Taylor expansion of (52) and letting $y_g = \operatorname{argmin}_{y \in S_g^*(x)} \|y_\gamma - y_g\|$, it holds that

$$\|\nabla_x f(x^*, y_\gamma) + \gamma\nabla_{xy} g(x^*, y_\gamma)(y_\gamma - y_g)\|^2 \leq \epsilon + \mathcal{O}(\|y_\gamma - y_g\|^2) \leq \mathcal{O}(\epsilon),$$
$$\|\nabla_y f(x^*, y_\gamma) + \gamma\nabla_{yy} g(x^*, y_\gamma)(y_\gamma - y_g)\|^2 \leq \epsilon + \mathcal{O}(\|y_\gamma - y_g\|^2) \leq \mathcal{O}(\epsilon).$$

where the last two inequalities are due to (54). Together with (54) and defining $w = \gamma(y_\gamma - y_g)$ with finite norm $\|w\| = \mathcal{O}(\epsilon^{-\frac{2-\alpha}{2}}\epsilon) = \mathcal{O}(\epsilon^{\frac{\alpha}{2}}) \leq 1$, the point (x^*, y_γ, w) satisfies the approximate KKT conditions in (53). □

B.6 Proof of Theorem 3

In the following, we start with a more general setting where x is bounded in a domain $\mathcal{X}$ and the update of x is conducted via projected gradient descent.

Denote the gradient approximate $g_t = \nabla_x f(x, y_{t+1}^\gamma)$. According to smoothness, we have

$$\begin{aligned}
F_\gamma(x_{t+1}) - F_\gamma(x_t) \leq& \langle \nabla F_\gamma(x_t) - g_t + g_t, x_{t+1} - x_t\rangle + \frac{l_{F,1}}{2}\|x_{t+1} - x_t\|^2 \\
\leq& -\frac{1}{\eta}\|x_{t+1} - x_t\|^2 + \frac{1}{2\eta}\|x_{t+1} - x_t\|^2 + \|\nabla F_\gamma(x_{t+1}) - g_t\|\|x_{t+1} - x_t\| \\
\leq& -\frac{1}{2\eta}\|x_{t+1} - x_t\|^2 + \frac{1}{4\eta}\|x_{t+1} - x_t\|^2 + \eta\|\nabla F_\gamma(x_{t+1}) - g_t\|^2 \\
=& -\frac{1}{4\eta}\|x_{t+1} - x_t\|^2 + \eta\|\nabla F_\gamma(x_{t+1}) - g_t\|^2
\end{aligned} \tag{55}$$

where the second inequality uses $\eta \leq l_{F,1}^{-1}$, $\langle g_t, x_{t+1} - x_t\rangle \leq -\frac{1}{\eta}\|x_{t+1} - x_t\|^2$ by [9, Lemma 3.1] and Cauchy-Schwartz inequality; the third applies Young's inequality

For simplicity, denote $h(x, y) = \gamma^{-1} f(x, y) + g(x, y)$, $v^h(x) = \min_{y \in \mathcal{Y}} h(x, y)$, $y_t^{\gamma,*} = \arg\min_{y \in \mathcal{Y}} h(x_t, y)$ and $y_t^{g,*} \in \arg\min_{y \in \mathcal{Y}} g(x_t, y)$, and the update bias $b(x_t) = \nabla F_\gamma(x_t) - g_t$. In this way,

$$\begin{aligned}
\|b(x_t)\|^2 =& \|\nabla_x f(x_t, y_t^{\gamma,*}) + \gamma(\nabla_x g(x_t, y_t^{\gamma,*}) - \nabla_x g(x_t, y_t^{g,*})) - \nabla_x f(x_t, y_{t+1}^\gamma)\|^2 \\
\overset{(a)}{\leq}& 2\|\nabla_x f(x_t, y_t^{\gamma,*}) - \nabla_x f(x_t, y_{t+1}^\gamma)\| + 2\gamma^2\|\nabla_x g(x_t, y_t^{\gamma,*}) - \nabla_x g(x_t, y_t^{g,*})\|^2 \\
\overset{(b)}{\leq}& 2l_{f,1}^2\|y_{t+1} - y_t^\gamma\|^2 + \mathcal{O}\big(\gamma^{-\frac{2(\alpha-1)}{2-\alpha}} + \delta\gamma\big)
\end{aligned}$$

$$
\begin{aligned}
&\overset{(c)}{\leq} \frac{4}{\mu_\gamma^*} l_{f,1}^2 (h(x_t, y_{t+1}^\gamma) - v^h(x_t)) + \mathcal{O}\big(\gamma^{-\frac{2(\alpha-1)}{2-\alpha}} + \delta\gamma\big) \\
&\overset{(d)}{\leq} \frac{4}{\mu_\gamma^*} l_{f,1}^2 (1-\eta^\gamma\mu)(h(x_t, y_t^\gamma) - v^h(x_t)) + \mathcal{O}\big(\gamma^{-\frac{2(\alpha-1)}{2-\alpha}} + \delta\gamma\big)
\end{aligned} \tag{56}
$$

where (a) applies the Young's inequality; (b) follows the smoothness of f and Lemma 1; (c) employs the property of strong convexity; and (d) is by the descent theory for applying projected gradient descent on problems satisfying PL condition, see e.g. [41, Theorem 5].

Plugging (56) in (55), there is

$$
\begin{aligned}
F_\gamma(x_{t+1}) - F_\gamma(x_t) \leq & -\frac{\eta}{4}\|\nabla F_\gamma(x_t)\|^2 + \eta \frac{4}{\mu_\gamma^*} l_{f,1}^2 (1-\eta^\gamma\mu)(h(x_t, y_t^\gamma) - v^h(x_t)) \\
& + \eta\mathcal{O}\big(\gamma^{-\frac{2(\alpha-1)}{2-\alpha}} + \delta\gamma\big)
\end{aligned} \tag{57}
$$

Moreover, as $h(x,y)$ is $l_{h,1} = \gamma^{-1} l_{f,1} + l_{g,1}$-smooth and $v^h(x)$ is $l_{v^h,1} = l_{h,1}(1 + L_y^\gamma)$-smooth, there is

$$
\begin{aligned}
& h(x_{t+1}, y_{t+1}^\gamma) - v^h(x_{t+1}) \\
\overset{(a)}{\leq} & h(x_t, y_{t+1}^\gamma) - v^h(x_t) + \langle \nabla_x h(x_t, y_{t+1}^\gamma) - \nabla v^h(x_t), x_{t+1} - x_t \rangle + \frac{\eta^2 (l_{h,1} + l_{v^h,1})}{2} \|\frac{x_{t+1} - x_t}{\eta}\|^2 \\
\overset{(b)}{\leq} & h(x_t, y_{t+1}^\gamma) - v^h(x_t) + \eta l_{h,1} \|y_{t+1}^\gamma - y_t^{\gamma,*}\| \|\frac{x_{t+1} - x_t}{\eta}\| + \frac{\eta^2 (l_{h,1} + l_{v^h,1})}{2} \|\frac{x_{t+1} - x_t}{\eta}\|^2 \\
\overset{(c)}{\leq} & h(x_t, y_{t+1}^\gamma) - v^h(x_t) + \eta l_{h,1} \frac{z}{2} \|y_{t+1}^\gamma - y_t^{\gamma,*}\|^2 + \frac{\eta l_{h,1}}{2z} \|\frac{x_{t+1} - x_t}{\eta}\|^2 + \frac{\eta^2 (l_{h,1} + l_{v^h,1})}{2} \|\frac{x_{t+1} - x_t}{\eta}\|^2 \\
\overset{(d)}{\leq} & (1 + \frac{\eta l_{h,1} z}{2})(h(x_t, y_{t+1}^\gamma) - v^h(x_t)) + (\frac{\eta l_{h,1}}{2z} + \frac{\eta^2 (l_{h,1} + l_{v^h,1})}{2}) \|\frac{x_{t+1} - x_t}{\eta}\|^2 \\
\overset{(e)}{\leq} & (1 + \frac{\eta l_{h,1} z}{2})(1-\eta^\gamma\mu)(h(x_t, y_t^\gamma) - v^h(x_t)) + (\frac{\eta l_{h,1}}{2z} + \frac{\eta^2 (l_{h,1} + l_{v^h,1})}{2}) \|\frac{x_{t+1} - x_t}{\eta}\|^2, \quad \forall z > 0.
\end{aligned} \tag{58}
$$

Here, (a) follows the smoothness of $h(x,y) + v^h(x)$ in x; (b) applies Cauchy-Schwartz inequality and the smoothness of h in y; (c) uses Young's inequality for any $z > 0$; (d) is from the PL condition of $h(x,y)$ in y; (e) is similarly by the descent theory for applying projected gradient descent on $h(x,\cdot)$ satisfying PL condition [41, Theorem 5].

In this way, adding $c(h(x_{t+1}, y_{t+1}^\gamma) - v^h(x_{t+1}))$ to both side of (56), there is

$$
\begin{aligned}
& F_\gamma(x_{t+1}) + c(h(x_{t+1}, y_{t+1}^\gamma) - v^h(x_{t+1})) \\
\leq & F_\gamma(x_t) + \left(-\frac{\eta}{4} + c\Big(\frac{\eta l_{h,1}}{2z} + \frac{\eta^2 (l_{h,1} + l_{v^h,1})}{2}\Big)\right) \|\frac{x_{t+1} - x_t}{\eta}\|^2 \\
& + c\Big(\big(1 + \eta\big(\frac{l_{h,1} z}{2} + l_{f,1}^2 \frac{4}{\mu_\gamma^* c}\big)\big)(1-\eta^\gamma\mu)(h(x_t, y_t^\gamma) - v^h(x_t))\Big) + \eta\mathcal{O}\big(\gamma^{-\frac{2(\alpha-1)}{2-\alpha}} + \delta\gamma\big).
\end{aligned}
$$

In this way, choose the following hyper-parameter,

$$
\begin{cases}
c = \mu^{-\frac{1}{2}} \\
z = 8 c l_{h,1} \\
\eta^\gamma \leq l_{h,1}^{-1} \\
\eta \leq \min\left\{ \frac{1}{8c(l_{h,1} + l_{v^h,1})}, \frac{\eta^\gamma\mu/(1-\eta^\gamma\mu)}{\frac{l_{h,1} z}{2} + \frac{4 l_{f,1}^2}{\mu c}} \right\}
\end{cases} \tag{59}
$$

i.e. $c = \mathcal{O}(1)$, $\eta = \mathcal{O}(1)$, there is

$$
F_\gamma(x_{t+1}) - F_\gamma(x_t) + c(h(x_{t+1}, y_{t+1}^\gamma) - v^h(x_{t+1}))
$$

Algorithm 2 Fully-single-loop F^{2}SA [45] without momentum

inputs: initial points x_0; step size η, η^g, η^γ; counters T.
for $t = 0, 1, \ldots, T-1$ **do**
  update $y_{t+1}^g = y_t^g - \eta^g \nabla_y g(x_t, y_t)$
  update $y_{t+1}^\gamma = y_t^\gamma - \eta^\gamma \big(\nabla_y \gamma^{-1} f(x_t, y_t^\gamma) + \nabla_y g(x_t, y_t^\gamma)\big)$
  update $x_{t+1} = x_t - \eta g_t$ where $g_t = \nabla_x f(x, y_t^\gamma) + \gamma(\nabla_x g(x, y_t^\gamma) - \nabla_x g(x, y_t^g))$.
end for
outputs: (x_T, y_t^γ)

$$\leq -\frac{\eta}{8}\|\nabla F_\gamma(x_t)\|^2 + c(h(x_t, y_t^\gamma) - v^h(x_t)) + \eta\mathcal{O}\big(\gamma^{-\frac{2(\alpha-1)}{2-\alpha}} + \delta\gamma\big).$$

Denote $D_1 = F_\gamma(x_0) - F_\gamma(x_T)$, $D_2 = (h(x_0, y_0^\gamma) - v^h(x_0)) - (h(x_T, y_T^\gamma) - v^h(x_T))$. Rearranging and telescoping gives

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla F_\gamma(x_t)\|^2 \leq& \frac{8\,(D_1 + cD_2)}{\eta T} + \mathcal{O}\big(\gamma^{-\frac{2(\alpha-1)}{2-\alpha}} + \delta\gamma\big)\\ =&\mathcal{O}(T^{-1} + \delta^{\frac{2(\alpha-1)}{\alpha}})\end{aligned} \tag{60}$$

where the last equality is achieved as $c = \mathcal{O}(1)$ and $\eta = \mathcal{O}(1)$ and by setting $\gamma = \mathcal{O}(\delta^{-\frac{2-\alpha}{\alpha}})$. This confirms that the trajectory x_t stabilizes on average. Moreover, the hyper-parameter choices in (59) reformulate (58), which can be plugged in (56) to obtain

$$\frac{1}{T}\sum_{t=0}^{T-1}\|b_t\|^2 \leq \frac{4}{\mu_\gamma^*} l_{f,1}^2 (1-\eta^\gamma\mu)\frac{1}{T}\sum_{t=0}^{T-1}(h(x_t, y_t^\gamma) - v^h(x_t)) + \mathcal{O}\big(\delta^{\frac{2(\alpha-1)}{\alpha}}\big) \leq \mathcal{O}\big(T^{-1} + \delta^{\frac{2(\alpha-1)}{\alpha}}\big). \tag{61}$$

where the last inequality follows by applying Abel's summation formula on series $\sum_{k=1}^K a_k b_k$ where $a_k = \mathcal{O}((1-\eta^\gamma\mu/2)^k)$ and $K^{-1}\sum_{k=0}^K b_k = \mathcal{O}(T^{-1} + \delta^{\frac{2(\alpha-1)}{\alpha}})$.

In this way, there is

$$\begin{aligned}\|\nabla F_\gamma(x_t)\|^2 =& \Big\|\frac{x_t - (x_t - \eta\nabla F_\gamma(x_t))}{\eta}\Big\|^2\\ =&\Big\|\frac{x_t - (x_t - \eta g_t - \eta b_t)}{\eta}\Big\|^2\\ \leq& 2\Big\|\frac{x_t - x_{t+1}}{\eta}\Big\|^2 + 2\|b_t\|^2\end{aligned} \tag{62}$$

where inequality is by Young's inequality. Therefore,

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla F_\gamma(x_t)\|^2 \leq& 2\frac{1}{T}\sum_{t=0}^{T-1}\Big\|\frac{x_t - x_{t+1}}{\eta}\Big\|^2 + 2\frac{1}{T}\sum_{t=0}^{T-1}\|b_t\|^2\\ \leq&\mathcal{O}\big(T^{-1} + \delta^{\frac{2(\alpha-1)}{\alpha}}\big)\end{aligned} \tag{63}$$

where the last inequality is obtained by plugging in (60) and (61). $\alpha \leq 1.5$ and rearranging.

B.7 Additional discussion on fully-single-loop version of F^{2}SA [45]

Since momentum updates in F^{2}SA [45] introduce additional memory cost, in this section, we look into the fully-single-loop version of F^{2}SA [45] without momentum, i.e. at each iteration t, it updates:

$$y_{t+1}^g = y_t^g - \eta^g \nabla_y g(x_t, y_t), \quad \text{and} \tag{64}$$

$$y_{t+1}^\gamma = y_t^\gamma - \eta^\gamma\big(\nabla_y\gamma^{-1} f(x_t, y_t^\gamma) + \nabla_y g(x_t, y_t^\gamma)\big) \tag{65}$$

where $\eta^g \leq l_{g,1}^{-1}$ and $\eta^\gamma \leq (l_{g,1} + \gamma^{-1} l_{f,1})^{-1}$ are the step sizes. We summarize the algorithm in Algorithm 2 present the convergence results in the following theorem.

Proposition 5. *Suppose all assumptions in Proposition 2 hold. For iterations using the fully-single-loop version of Algorithm 2 with $\eta = \mathcal{O}(\gamma^{-1})$ gives*

$$\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla F_\gamma(x_t)\|^2 = \mathcal{O}(\gamma^2 T^{-1}). \tag{66}$$

Proof. Denote the gradient approximate $g_t = \nabla_x f(x, y_{t+1}^\gamma) + \gamma\nabla_x g(x, y_{t+1}^\gamma) - \gamma\nabla_x g(x, y_{t+1}^g)$. According to smoothness, we have

$$\begin{aligned}
F_\gamma(x_{t+1}) - F_\gamma(x_t) \leq& \langle \nabla F_\gamma(x_t) - g_t + g_t, x_{t+1} - x_t\rangle + \frac{l_{F,1}}{2}\|x_{t+1} - x_t\|^2 \\
\leq& -\frac{1}{\eta}\|x_{t+1} - x_t\|^2 + \frac{1}{2\eta}\|x_{t+1} - x_t\|^2 + \|\nabla F_\gamma(x_{t+1}) - g_t\|\|x_{t+1} - x_t\| \\
\leq& -\frac{1}{2\eta}\|x_{t+1} - x_t\|^2 + \frac{1}{4\eta}\|x_{t+1} - x_t\|^2 + \eta\|\nabla F_\gamma(x_{t+1}) - g_t\|^2 \\
=& -\frac{\eta}{4}\|\nabla F_\gamma(x_t)\|^2 + \eta\|\nabla F_\gamma(x_{t+1}) - g_t\|^2
\end{aligned} \tag{67}$$

where the second inequality uses $\eta \leq l_{F,1}^{-1}$, $\langle g_t, x_{t+1} - x_t\rangle \leq -\frac{1}{\eta}\|x_{t+1} - x_t\|^2$ by [9, Lemma 3.1] and Cauchy-Schwartz inequality; the third applies Young's inequality

Moreover, denote $h(x, y) = \gamma^{-1}f(x, y) + g(x, y)$, $v^h(x) = \min_{y\in\mathcal{Y}} h(x, y)$, $y_t^{\gamma,*} = \arg\min_{y\in\mathcal{Y}} h(x_t, y)$ and $y_t^{g,*} \in \arg\min_{y\in\mathcal{Y}} g(x_t, y)$, by triangle inequality and Young's inequality, there is

$$\begin{aligned}
&\|\nabla F_\gamma(x_t) - g_t\|^2 \\
\leq& 2\gamma^2\|\nabla_x h(x_t, y_{t+1}^\gamma) - \nabla v^h(x_t)\|^2 + 2\gamma^2\|\nabla_x g(x_t, y_{t+1}^g) - \nabla v(x_t)\|^2 \\
=& 2\gamma^2 l_{h,1}^2\|y_{t+1}^\gamma - y_t^{\gamma,*}\|^2 + 2\gamma^2 l_{g,1}^2\|y_{t+1}^g - y_t^{g,*}\|^2 \\
\leq& 2\gamma^2 l_{h,1}^2\frac{2}{\gamma\mu_h}\gamma(h(x_t, y_{t+1}) - v^h(x_t)) + 2\gamma^2 l_{g,1}^2\frac{2}{\gamma\mu}\gamma(g(x_t, y_{t+1}) - v(x_t)) \\
\leq& 2\gamma^2 l_{h,1}^2\frac{2}{\gamma\mu_h}(1 - \eta^y\mu_h)\gamma(h(x_t, y_t) - v^h(x_t)) + 2\gamma^2 l_{g,1}^2\frac{2}{\gamma\mu}(1 - \eta^y\mu)\gamma(g(x_t, z_t) - v(x_t)) \\
\leq& 2\gamma^2 l_{h,1}^2\frac{2}{\gamma\mu_h}(1 - \eta^y\mu_h)\gamma(h(x_t, y_t) - v^h(x_t)) + 2\gamma^2 l_{g,1}^2\frac{2}{\gamma\mu}(1 - \eta^y\mu)\gamma(g(x_t, z_t) - v(x_t)) \\
\leq& 2\gamma^2 l_{h,1}^2\frac{1}{\gamma^2\mu_h^2}(1 - \eta^y\mu_h)\|\gamma\nabla_y h(x_t, y_t)\|^2 + 2\gamma^2 l_{g,1}^2\frac{1}{\gamma^2\mu^2}(1 - \eta^y\mu)\|\gamma\nabla_y g(x_t, z_t)\|^2
\end{aligned} \tag{68}$$

The second to last inequality follows PL condition and the last inequality is by the descent theory for applying projected gradient descent on problems satisfying PL condition, see e.g. [41, Theorem 5].

Plugging (68) in (67), there is

$$\begin{aligned}
F_\gamma(x_{t+1}) - F_\gamma(x_t) \leq& -\frac{\eta}{4}\|\nabla F_\gamma(x_t)\|^2 + \eta\frac{4}{\mu_\gamma^*}\gamma^2(\gamma^{-1}l_{f,1} + l_{g,1})^2(1 - \eta^\gamma\mu)(h(x_t, y_t^\gamma) - v^h(x_t)) \\
&+ \eta\frac{4}{\mu}\gamma^2(l_{g,1})^2(1 - \eta^g\mu)(h(x_t, y_t^g) - v^h(x_t)).
\end{aligned} \tag{69}$$

Moreover, as $h(x, y)$ is $l_{h,1} = \gamma^{-1}l_{f,1} + l_{g,1}$-smooth and $v^h(x)$ is $l_{v^h,1} = l_{h,1}(1 + L_y^\gamma)$-smooth, there is

$$\begin{aligned}
&h(x_{t+1}, y_{t+1}^\gamma) - v^h(x_{t+1}) \\
\overset{(a)}{\leq}& h(x_t, y_{t+1}^\gamma) - v^h(x_t) + \langle\nabla_x h(x_t, y_{t+1}^\gamma) - \nabla v^h(x_t), x_{t+1} - x_t\rangle + \frac{\eta^2(l_{h,1} + l_{v^h,1})}{2}\|\nabla F_\gamma(x_t)\|^2 \\
\overset{(b)}{\leq}& h(x_t, y_{t+1}^\gamma) - v^h(x_t) + \eta l_{h,1}\|y_{t+1}^\gamma - y_t^{\gamma,*}\|\|\nabla F_\gamma(x_t)\| + \frac{\eta^2(l_{h,1} + l_{v^h,1})}{2}\|\nabla F_\gamma(x_t)\|^2 \\
\overset{(c)}{\leq}& h(x_t, y_{t+1}^\gamma) - v^h(x_t) + \eta l_{h,1}\frac{z}{2}\|y_{t+1}^\gamma - y_t^{\gamma,*}\|^2 + \frac{\eta l_{h,1}}{2z}\|\nabla F_\gamma(x_t)\|^2 + \frac{\eta^2(l_{h,1} + l_{v^h,1})}{2}\|\nabla F_\gamma(x_t)\|^2
\end{aligned}$$

$$
\begin{aligned}
&\overset{(d)}{\leq}(1+\frac{\eta l_{h,1}z}{2})(h(x_t,y_{t+1}^{\gamma})-v^h(x_t))+(\frac{\eta l_{h,1}}{2z}+\frac{\eta^2(l_{h,1}+l_{v^h,1})}{2})\|\nabla F_\gamma(x_t)\|^2\\
&\overset{(e)}{\leq}(1+\frac{\eta l_{h,1}z}{2})(1-\eta^\gamma\mu)(h(x_t,y_t^{\gamma})-v^h(x_t))+(\frac{\eta l_{h,1}}{2z}+\frac{\eta^2(l_{h,1}+l_{v^h,1})}{2})\|\nabla F_\gamma(x_t)\|^2,\quad \forall z>0.
\end{aligned}
\tag{70}
$$

Here, (a) follows the smoothness of $h(x,y)+v^h(x)$ in x; (b) applies Cauchy-Schwartz inequality and the smoothness of h in y; (c) uses Young's inequality for any $z>0$; (d) is from the PL condition of $h(x,y)$ in y; (e) is similarly by the descent theory for applying projected gradient descent on $h(x,\cdot)$ satisfying PL condition [41, Theorem 5].

Following similar analysis, as $g(x,y)$ is $l_{g,1}$-smooth and $v(x)$ is $l_{v,1}=l_{g,1}(1+L_y^g)$-smooth, there is

$$
\begin{aligned}
&g(x_{t+1},y_{t+1}^g)-v(x_{t+1})\\
\leq&(1+\frac{\eta l_{g,1}z'}{2})(1-\eta^g\mu_{g^*})(g(x_t,y_t^g)-v(x_t))+(\frac{\eta l_{g,1}}{2z'}+\frac{\eta^2(l_{g,1}+l_{v,1})}{2})\|\nabla F_\gamma(x_t)\|^2.
\end{aligned}
\tag{71}
$$

In this way, adding $c(h(x_{t+1},y_{t+1}^{\gamma})-v^h(x_{t+1}))$ and $c'(g(x_{t+1},y_{t+1}^g)-v(x_{t+1}))$ to both side of (69), there is

$$
\begin{aligned}
&F_\gamma(x_{t+1})-F_\gamma(x_t)+c(h(x_{t+1},y_{t+1}^{\gamma})-v^h(x_{t+1}))+c'(g(x_{t+1},y_{t+1}^g)-v(x_{t+1}))\\
\leq&\Big(-\frac{\eta}{4}+c\Big(\frac{\eta l_{h,1}}{2z}+\frac{\eta^2(l_{h,1}+l_{v^h,1})}{2}\Big)+c'\Big(\frac{\eta l_{g,1}}{2z'}+\frac{\eta^2(l_{g,1}+l_{v,1})}{2}\Big)\Big)\|\nabla F_\gamma(x_t)\|^2\\
&+c\Big(\big(1+\eta\big(\frac{l_{h,1}z}{2}+\gamma^2l_{h,1}^2\frac{4}{\mu_\gamma^* c}\big)\big)(1-\eta^\gamma\mu)(h(x_t,y_t^{\gamma})-v^h(x_t))\Big)\\
&+c'\Big(\big(1+\eta\big(\frac{l_{g,1}z'}{2}+\gamma^2l_{g,1}^2\frac{4}{\mu c'}\big)\big)(1-\eta^g\mu_g)(g(x_t,y_t^g-v(x_t))\Big).
\end{aligned}
$$

In this way, choose the following hyper-parameter,

$$
\begin{cases}
c=\gamma\mu^{-\frac{1}{2}}\\
c'=\gamma\mu^{-\frac{1}{2}}\\
z=16cl_{h,1}\\
z'=16c'l_{g,1}\\
\eta^g\leq l_{g,1}^{-1}\\
\eta^\gamma\leq l_{h,1}^{-1}\\
\eta\leq\min\left\{\frac{1}{16c(l_{h,1}+l_{v^h,1})},\frac{1}{16c'(l_{g,1}+l_{v,1})},\frac{\eta^\gamma\mu/(1-\eta^\gamma\mu)}{\frac{l_{h,1}z}{2}+\frac{4\gamma^2l_{h,1}^2}{\mu c}},\frac{\eta^g\mu_g/(1-\eta^g\mu_g)}{\frac{l_{g,1}z'}{2}+\frac{4\gamma^2l_{g,1}^2}{\mu_g c}}\right\}
\end{cases}
\tag{72}
$$

i.e. $c,c'=\mathcal{O}(\gamma)$, $\eta=\mathcal{O}(\gamma^{-1})$, there is

$$
\begin{aligned}
&F_\gamma(x_{t+1})-F_\gamma(x_t)+c(h(x_{t+1},y_{t+1}^{\gamma})-v^h(x_{t+1}))+c'(g(x_{t+1},y_{t+1}^g)-v(x_{t+1}))\\
\leq&-\frac{\eta}{8}\|\nabla F_\gamma(x_t)\|^2+c(h(x_t,y_t^{\gamma})-v^h(x_t))+c'l_{g,1}^2(g(x_t,y_t^g-v(x_t)).
\end{aligned}
$$

Denote $D_1=F_\gamma(x_0)-F_\gamma(x_T)$, $D_2=(h(x_0,y_0^{\gamma})-v^h(x_0))-(h(x_T,y_T^{\gamma})-v^h(x_T))$, and $D_3=(g(x_0,y_0^g)-v(x_0))-(g(x_T,y_T^g)-v(x_T))$. Rearranging and telescoping gives

$$
\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla F_\gamma(x_t)\|^2\leq\frac{8\left(D_1+cD_2+c'D_3\right)}{\eta T}=\mathcal{O}(\gamma^2T^{-1})
\tag{73}
$$

where the last equality is because $c,c'=\mathcal{O}(\gamma)$, and $\eta=\mathcal{O}(\gamma^{-1})$. □

The convergence of the fully-single-loop F^{2}SA without momentum is hindered by a larger γ, which regulates the UL violation rate. While the general case requires $\gamma=\mathcal{O}(\epsilon^{-0.5})$ as per Lemma 1. This shows that the fully-single-loop version of F^{2}SA, though computationally efficient, suffers from higher international cost.

C Additional Experimental Details

C.1 Additional details for toy example in Figure 2

In this section, we provide details for the toy example of PEFT BLO problem in Figure 2. We consider a binary classification setting where the model parameters $\theta = (x, y)$ consist of the UL variable x and LL variable y, with $\theta \in \mathbb{R}^2$. The model implements a 1D convolutional network with softmax activation:

```
class SoftmaxNN(nn.Module):
    def __init__(self):
        super().__init__()
        self.hidden = nn.Conv1d(in_channels=1, out_channels=1,
                      kernel_size=2, stride=2, bias=False)
        self.activation = nn.Softmax(dim=1)
        self._init_weight()
```

We specify the SFT datasets $\mathcal{D}_{\text{SFT}} = \{(X_1, y)\}$, and the DPO dataset $\mathcal{D}_{\text{DPO}} = \{(X_2, y_w, y_\ell)\}$ in Table 3. The BLO problem is specified in (3), where f_{DPO} consists of a DPO loss with $\beta = 1$ [70] plus an ℓ_2 regularization term (weight 0.01) and g_{SFT} consists of a negative log-likelihood loss and the same regulariztion. The reference model is obtained via learning on $g_{\text{SFT}}(x, y)$ (parameterized with $(x = -5.34, y = -9.94)$).

We apply our PBGD-Free algorithm in Algorithm 1 in comparison with F^{2}SA [45] with $\gamma = 15$, $K = 10$ inner loop to solve (5), and $T = 5000$ outer loop for both algorithms.

Table 3: Dataset specification for toy illustration

Input	Output	Feature
X_1	$y = 0$	$[1.0, 1.0, 0.5, 0.5]^\top$
X_1	$y' = 1$	$[1.0, 0.5, 0, 0.5]^\top$
X_2	$y_w = 1$	$[1.0, 0.5, 0.5, 0.5]^\top$
X_2	$y_\ell = 0$	$[0.5, 1.0, 1.0, 1.0]^\top$

C.2 Representation learning problem on NLSY dataset [73]

BLO has proven effective in representation learning for obtaining a joint backbone model x that captures unified task features and generalizes well to downstream tasks by only tuning the head y [4, 95, 83, 36, 80]. We test our algorithm on a representation learning problem on the National Longitudinal Survey of Youth (NLSY) dataset [73], following the experimental setup in [80]. This problem aims to learn representations to predict normalized income via $\min_{x,y} f_{\text{MSE}}(x, y; D_1)$ s.t. $y \in \arg\min_y f_{\text{MSE}}(x, y; D_2)$, where D_1, D_2 are datasets partitioned by gender. The representation model, parameterized x, consists of two fully connected layers (hidden size 200, ReLU activation), and the predictor, parameterized y, is a linear classification head.

We compare our fully-single-loop PBGD-Free (with inner iteration $K = 1$) against F^{2}SA [45] (with $K = 2$) and the ITD algorithm from [80], following the experimental setup in [80]. As shown in Table 4, the performance gap is particularly notable in efficiency, where PBGD-Free is over twice as fast as F^{2}SA [45] and more than 30 times faster than the ITD-based approach [80], primarily because it omits the value-function part and the inner loop of $y_g^*(x)$. Moreover, PBGD-Free achieves lower MSE than PBGD [44]. This improvement stems from PBGD-Free's ability to avoid the bias γ-propagation inherent in PBGD's design. When both algorithms are single-loop (or nearly single-loop for small $K = 2$), PBGD's reliance on a fixed penalty parameter γ amplifies initial inner update biases throughout training, slowing convergence, detailed in Appendix B.7 while PBGD-Free eliminates these γ-dependent value function terms.

C.3 LLM PEFT problem (3)

General Setup. We evaluate our PEFT framework (3) using the `Dahoas/rm-hh-rlhf` dataset for DPO loss and the `OpenOrca` dataset for SFT loss. For training, we test one PYTHIA-1b [6]

Methods	MSE	Time (s)
F^{2}SA [45]	1.9331 ± 0.0794	12.33 ± 0.34
Implicit [80]	2.1530 ± 0.0455	169.69 ± 0.36
PBGD-Free	$\mathbf{1.8916 \pm 0.1245}$	$\mathbf{5.15 \pm 0.06}$

Table 4: Performance results for different training methods on representation learning problem on NLSY-7k Dataset [73]. The mean $\pm$ standard deviation is reported for both the mean MSE and the mean time over 5 random experiments on the test dataset.

Method	Avg Memory Used (MB)	Peak Memory Used (MB)
BOME	18834.53	21535.96
F^{2}SA	16213.78	17622.43
ALRIGHT	16031.86	**16107.45**
PBGD-Free	**16016.94**	16180.89

Table 5: Empirical GPU memory usage for each method over 3 epoches of training.

model with 1800 samples for each dataset (batch size 16) and the LLAMA-3-3B [22] model with 4800 samples (batch size 32). Both models are adapted with LoRA (ALPHA 16, RANK 16) and we treat LoRA PEFT weights on the attention layers as x, the last layer linear head as y. The learning rate is set to 1×10^{-5}, using Adam [43] as the optimizer. All experiments were conducted on a cluster of NVIDIA A6000 GPUs, each with 40 GB of memory. Training was performed using PyTorch with the DeepSpeed library `https://github.com/deepspeedai/DeepSpeed` to optimize memory usage and distributed training efficiency. We consider a time-limited experiment under a consistent computational budget, reflecting real-world constraints where training time is often a critical factor.

Algorithm hyperparameter. We use a penalty constant of $\gamma = 10$ for our proposed PBGD-Free algorithm (Algorithm 1) with a single inner loop ($K = 1$). For the baseline F^{2}SA algorithm [11, 45], we set $\gamma = 10$ with $K = 3$ inner updates for training LLAMA-3-3B [31], and $K = 5$ for PYTHIA-1b [6]. For the BOME algorithm, we similarly use $K = 3$ and $K = 5$ inner loops, adopting its hyperparameter $\eta = 0.5$ for calculating the penalty constant, as suggested in [105]. For the ALRIGHT algorithm [24], we use its default setting of $\lambda = 0.5$ as suggested in literature [24]. Since the ALRIGHT algorithm in [24] is a bi-objective learning algorithm that does not have the representation learning capability, we examine it on an alternative formulation $\min_{x,y}[f_{\text{DPO}(x,y)}, g_{\text{SFT}}(y)]$.

Faster training than BLO baselines and more stable over bi-objective. As presented in Figure 9, when training PYTHIA-1b [6] on the PEFT problem (3), our PBGD-Free algorithm demonstrates the fastest convergence compared to the baseline BLO methods F^{2}SA [11, 45] and BOME [105], both of which fail to converge within the given time budget. Additionally, PBGD-Free shows greater stability compared to its bi-objective counterpart ALRIGHT [24]. Table 5 reports the average and peak GPU memory consumption over 3 epochs for all compared methods, illustrating that PBGD-Free maintains a memory footprint comparable to the baselines.

Better transferability to new task. Figure 10 further illustrates the performance of the outputs from BLO PEFT learning (3) in the subsequent post-SFT-tuning phase **S2)**. The BLO baselines (F^{2}SA [11, 45] and BOME [105]), which did not achieve convergence in the initial PEFT phase due to their higher time complexity, tend to sacrifice DPO performance when improving SFT performance during post-SFT tuning **S2)**. In contrast, PBGD-Free algorithm and its bi-objective counterpart ALRIGHT [24] demonstrate the ability to preserve strong preference alignment (DPO) while conducting SFT training in **S2)**. This shows that the preference backbone x learned by both of them can be adapted to new task by fine-tuning only the linear head to achieve strong SFT performance. Notably, the BLO PEFT outputs trained by PBGD-Free achieves better SFT performance with substantially lower SFT loss, highlighting the advantage of the prioritization of SFT in our BLO formulation (3). This structure allows for a more powerful SFT tuning head, whereas bi-objective training methods tend to oscillate between potentially conflicting objectives, thereby limiting their post-SFT performance.

Example of SFT Evaluation Performance		
Human: Generate an approximately fifteen-word sentence that describes all this data: Midsummer House eatType restaurant; Midsummer House food Chinese; Midsummer House priceRange moderate; Midsummer House customer rating 3 out of 5; Midsummer House near All Bar One		
PYTHIA-1b [6]: Midsummer House staff a restaurant priced restaurant restaurant. a good-5 star rating. and in All Bar One.	**BLO-PEFT (F^2SA [45])**: Midsummer House is a restaurant priced restaurant restaurant with a 3-5 customer rating. located near All Bar One.	**BLO-PEFT (PBGD-Free)**: Midsummer House is a moderately priced Chinese restaurant with a 3/5 customer rating. located near All Bar One.
Human: You will be given a definition of a task first, then some input of the task. This task is about using the specified sentence and converting the sentence to Resource Description Framework (RDF) triplets of the form (subject, predicate, object). The RDF triplets generated must be such that the triplets accurately capture the structure and semantics of the input sentence. The input is a sentence and the output is a list of triplets of the form [subject, predicate, object] that capture the relationships present in the sentence. When a sentence has more than 1 RDF triplet possible, the output must contain all of them. AFC Ajax (amateurs)'s ground is Sportpark De Toekomst where Ajax Youth Academy also play.		
LLAMA-3-3B [31]: [["AjaxFC Ajax (amateurs)", "playsGround", "Sportpark De Toekomst"], ["Ajax Youth Academy", "has at", "Sportpark De Toekomst"]]	**BLO-PEFT (F^2SA [45])**: [["AjaxFC Ajax (amateurs)", "plays ground", "Sportpark De Toekomst"], ["Ajax Youth Academy", "has at", "Sportpark De Toekomst"]]	**BLO-PEFT (PBGD-Free)**: [["AFC Ajax (amateurs)", "has ground", "Sportpark De Toekomst"], ["Ajax Youth Academy", "plays at", "Sportpark De Toekomst"]]

Table 6: Examples of SFT evaluation performance for PYTHIA-1b [6], LLAMA-3-3B [31] and their corresponding BLO-PEFT (3) results via our PBGD-Free Algorithm 1 and basline F^2SA [45]. Text marked in red indicates incorrect outputs, orange indicates partially correct outputs that follow some of the instructions, and green indicates fully correct outputs that match the expected instructions.

Better SFT performance while maintaining preference learning. As illustrated in Table 6, the outputs generated by PBGD-Free demonstrate more precise and semantically accurate extraction, highlighting its superior SFT performance. In Figure 12, we present the loss metrics performance throughout post-SFT-tuning phase for LLAMA-3-3B [31] in addition to the results presented in Section 4. We observe that our backbone x trained on BLO PEFT (3) via PBGD-Free retains its lowest DPO rates throughout post-SFT-tuning. Figure 11 shows the quantitative results of the preference alignment using average reward gap and win rate. Together with Figure 8, they indicate that the backbone preference model x by the PBGD-Free maintains the first-tier DPO performance for preference alignment while enhancing the SFT performance by only fine-tuning the linear head. The slight DPO drop in Figure 11 of PBGD-Free compared with ALRIGHT is because it prioritizes better SFT performance, which restricts the feasible search space of representation model optimizing at the UL. However, since the representation evaluation criterion prioritizes strong SFT performance achieved by fine-tuning only the linear head, and treats preference alignment as a secondary goal, PBGD-Free remains the top-performing method. Moreover, according to Figure 7, PBGD-Free is more stable during the training compared with ALRIGHT. Additionally, Table 6 provides the SFT output comparison given by PBGD-Free and F^2SA on PYTHIA-1b [6], LLAMA-3-3B [31], from which we can see that both methods improve the response quality over the pre-trained model through BLO PEFT (3), while PBGD-Free generates better responses and follows the human instructions well.

Higher-rank LoRA enables finding better preference backbone via PBGD-Free. The last 2 columns in Figure 6 show that a higher-rank LoRA better preserves DPO with comparable SFT performance. It is likely because a higher-rank LoRA provides more over-parameterization, which ensures a more benign optimization landscape for the representation parameter x [101, 88, 54] and thus enables globally finding a better representation model x [90].

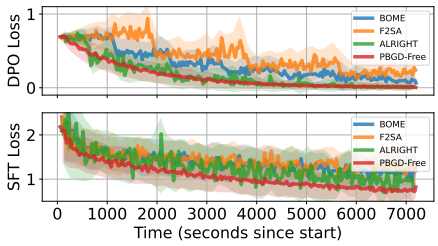

Figure 9: Train losses vs. time with STRIDE = 50 for different algorithms in solving (3) (or biobjective learning for ALRIGHT [24]) on PYTHIA-1b [6].

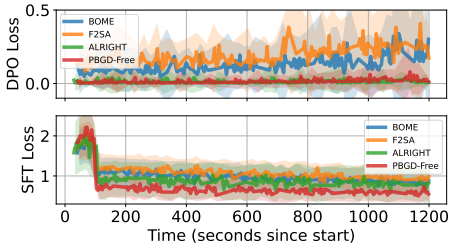

Figure 10: Train losses vs. time with STRIDE = 50 for different algorithms in Post SFT-tuning phase on PYTHIA-1b [6].

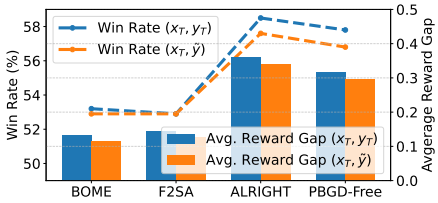

Figure 11: Average Reward Gap (↑) and Win Rate (↑) for different algorithms for PEFT LLAMMA-3-3B on [31] with the output (x_T, y_T) via each method in **S1)** and the outcome $(x_T, \tilde{y})$ from post-SFT-tuning on another dataset with fixed-backbone in **S2)**.

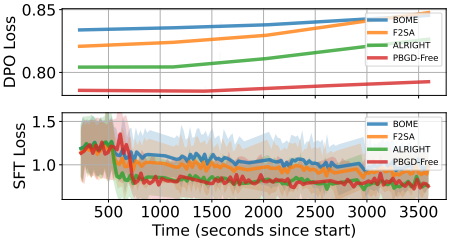

Figure 12: Evaluation of DPO losses and train SFT loss vs. time with STRIDE = 50 for different algorithms in Post SFT-tuning phase on for PEFT LLAMMA-3-3B [31].

C.4 BiDoRa fine-tuning problem

One of the recent applications of bilevel optimization lies in the field of Large Language Finetuning. [68] proposed BiDoRa, which considers fine-tuning using DoRa [53] by training on a BLO problem

$$\min_{m} l_{\text{tr}}^{l}(m, v^*(m)) \quad \text{s.t} \quad v^*(m) = l_{\text{tr}}^{s}(m, v^*(m)) + \rho R(v) \tag{74}$$

where m is the magnitude and v is the direction matrix for the low-rank incremental direction, $l_{\text{tr}}^{l}(m, v)$ and $l_{\text{tr}}^{l}(m, v)$ are respectively loss functions for fine-tuning training dataset splitting into large and small on proportion 0.66 to 0.67, and $R(v)$ is the Gram regularization loss [94] with constant ρ taken as $1e^{-3}$.

We conduct experiments on Microsoft Research Paraphrase Corpus (MRPC) dataset [20], and Internet Movie Database (IMDb) in Hugging Face by fine-tuning Bert model [59]. We apply fully-single-loop versions of PBGD and PBGD-Free in Algorithm 1 to solve the problem in Section 74 and compare it with training using DARTS [48], the algorithm used in the original BiDoRa algorithm [68], and the naive results trained on $\min_{m,v} l_{tr}(m, v)$ where l_{tr} is the combined loss for training dataset including the ones used for both l_{tr}^{l} and l_{tr}^{s} for DoRa [53]. The experiment is conducted on a single NVIDIA RTX A5000 GPU (24GB) using CUDA 12.2 and NVIDIA driver version 535.113.01.

As illustrated in Table 7, training BiDoRa using PBGD-Free in Algorithm 1 achieves the best performance in terms of test accuracy. It is more efficient than training using PBGD as it cuts the inner loop. Notably, it performs even better than the fully-single-loop of F^{2}SA [11]. This is consistent with the convergence results in Proposition 5 and Theorem 3.

Methods	**MRPC**	**IMDb**
BiDoRa-PBGD-Free	$\mathbf{0.839 \pm 0.006}$	$\mathbf{0.873 \pm 0.007}$
BiDoRa-F^{2}SA	0.820 ± 0.014	0.866 ± 0.016
BiDoRa-DARTS	/	/
DoRa	0.832 ± 0.010	0.872 ± 0.010

Table 7: Test accuracy (%) on training the finetuning parameters using BiDoRa-PBGD in comparison with DARTS [48], the algorithm used in [68], and with directly training DoRa [53]. It represents the accuracy mean $\pm$ standard deviation on 20 random training experiments. The "/" represents didn't converge in 10 times the time used in training DoRa.

D Broader Impact

This paper mainly focuses on developing an efficient BLO algorithm with a theoretical guarantee. By applying the proposed method to representation learning based PEFT and BiDORA, our work contributes to the broader development of parameter efficient LLM fine-tuning. Potential societal impacts include applications in creative content generation, data augmentation, and machine learning-based simulations. While we acknowledge the possibility of unintended uses, we do not identify any specific societal risks that need to be highlighted in this context.

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: The abstract and introduction clearly state the main contributions, including the development of PBGD-Free, a fully first-order, single-loop bilevel optimization algorithm that eliminates value function evaluations. The claims are supported by theoretical analysis and experiments, demonstrating improved convergence and scalability for LLM fine-tuning, aligning well with the paper's core contributions.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses limitations related to the reliance on flatness conditions, which are critical for the efficiency of the proposed PBGD-Free algorithm. It also acknowledges the assumption of joint (x, y) dependency in the upper-level objective (i.e., $f(x, y) \neq f(y)$).

   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: The paper provides a complete set of assumptions and detailed proofs for each theoretical result, including clear numbering and cross-referencing in both the main text and supplementary material. The assumptions are explicitly stated alongside theorems, ensuring that the analysis is rigorous and well-supported. Additionally, the main paper includes observations that guide intuition, with full formal derivations available in the appendix.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The paper provides sufficient details for reproducing the main experimental results, including hyperparameter settings, model architectures, and dataset descriptions. It clearly outlines the training procedures, evaluation metrics, and computational resources used, ensuring that the main claims and conclusions can be independently verified. Additionally, the supplementary material includes further implementation details, enhancing reproducibility.

   Guidelines:

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

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The code and data required to reproduce the main experimental results of this paper are included in the supplementary material.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The paper provides a comprehensive description of the training and test settings for all experiments. We provide key hyper-parameters as described in the main paper or in the Appendix, and we provide the detailed implementation in supplementary material.

   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: [Yes]

Justification: See Section 4 and Appendix C

Guidelines: This paper provides experiment results with multiple rounds of experiments, justifying the statistical significance of the experiments.

- 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: See Section C. For fair representation learning, we report the total runtime in Table 4. We report the computation resources for representation learning based PEFT in Appendix C.3 and C.4 for BiDoRA fine-tuning.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The authors obey the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: See Appendix D.

    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: We do not release new models and dataset, but utilize the existing ones. We focus on new problem formulation via BLO and proposing efficient algorithms using existing LLM models and dataset.

    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: We cite all the code based and datasets we are using in Section 4.

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: We release the code through anonymized zip file and provide all the details of training parameters in in Section 4 and Section C.

    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.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method we developed is original and completely without LLM.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.