TO SOLVE BILEVEL OPTIMIZATION WITH NONCONVEX LOWER LEVELS, WE NEED SECOND-ORDER STATIONARITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Although bilevel optimization (BLO) has emerged as a powerful framework for addressing many complex and nested machine learning problems in recent years, most existing studies are confined to the lower-level strongly convex (LLSC) or lower-level generally convex (LLGC) settings (i.e., the lower-level objective function is assumed to be, at least, convex). While the LLSC/LLGC assumptions render more tractable algorithmic design and theoretical analysis, they are too rigid to encompass many machine learning problems in practice. The limitations of LLSC/LLGC assumptions in BLO motivate us to investigate solving the BLO problem in the general lower-level nonconvex (LLNC) settings, which remains in its infancy. In the literature on LLNC-BLO, most of the existing works either require additional structures in the lower-level objective function for tractable theoretical analysis, or adopt the first-order stationarity reformulation as a lower-level surrogate problem, which is inherited from the LLSC/LLGC settings but could lose their effectiveness in the LLNC setting. To bridge this gap, in this work, we propose to reformulate the nonconvex lower-level problem using a *second-order stationarity-based surrogate,* the solution of which guarantees a local optimal solution at the lower level. Based on this reformulation, we propose the PROBE (Perturbed gradient algorithm for bilevel problem) and show that it overcomes the limitations of prior works by probing and escaping lower-level saddle points. We theoretically prove that PROBE achieves a finite-time convergence rate of $\mathcal{O}(T^{-\frac{2}{5}})$, where T denotes iterations. To our knowledge, this work is the first to establish the finite-time convergence rate guarantee for achieving lower-level second-order stationary solutions in general LLNC-BLO. Our experiments on both a large language model-based data curation task and a meta-learning task also show that PROBE outperforms state-of-the-art methods.

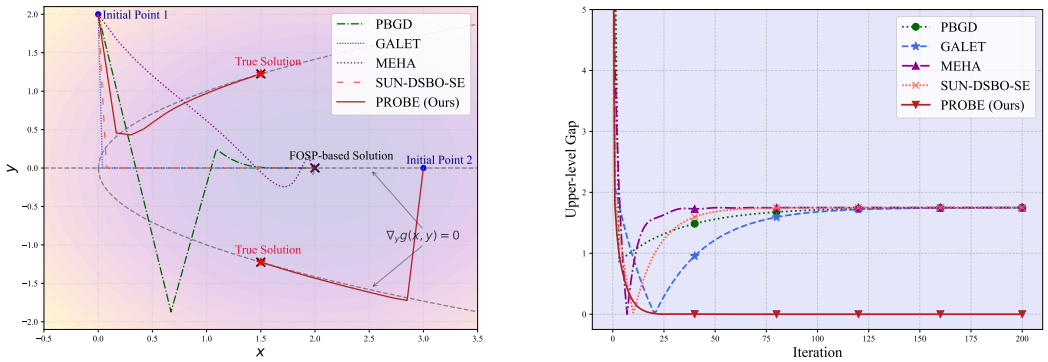
1 INTRODUCTION

In recent years, bilevel optimization (BLO) (Bracken & McGill, 1973; Liu et al., 2021; Zhang et al., 2024) has received significant attention in the machine learning community due to the rise of a wide range of complex and nested machine learning problems, such as reinforcement learning (Hong et al., 2023; Kudo et al., 2026), meta-learning (Franceschi et al., 2018; Ji et al., 2021), adversarial training (Zhang et al., 2022b), large language model (LLM) fine-tuning (Shen et al., 2024a), to name just a few. In general, a BLO problem can be formulated as follows:

$$\text{BLO}: \min_{x\in\mathbb{R}^p, y\in\mathbb{R}^q} f(x,y) \qquad \text{subject to } \; y\in\mathcal{S}(x) := \arg\min_y g(x,y), \tag{1}$$

where $f(x,y)$ and $g(x,y)$ are the upper- and lower-level objective functions, respectively. Clearly, the challenge of solving a BLO problem stems from the nested structure, i.e., part of the decision variables y in the upper-level objective $f(x,y)$ is obtained from the set of optimizers $S(x)$ given an upper-level variable $x\in\mathbb{R}^p$. To address this challenge and for tractable algorithmic design and analysis, most existing works in the BLO literature have relied on the *restrictive* lower-level strong convexity (LLSC) assumption, i.e., $g(x,\cdot)$ is *strongly convex* with respect to (w.r.t.) y for any given $x\in\mathbb{R}^p$ (Ghadimi & Wang, 2018; Ji et al., 2021; Yang et al., 2021; Dagréou et al., 2022),

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Converging trajectories. *(b) Convergence gaps in (a).*

Figure 1: Example: $x, y \in \mathbb{R}$, $f(x,y) = (x-2)^2 + y^2$, $g(x,y) = \frac{1}{4}y^4 - \frac{1}{2}xy^2$. *Fig. 1a shows the contour plot of* $f(x,y)$ *along with converging trajectories with initial points* $(0,2)$ *and* $(3,0)$*, and Fig. 1b plots* $|f(x_t,y_t) - f^*|$ *vs. iteration* t*, illustrating how far the convergent sequences are from solving the LLNC-BLO problem. Our* PROBE *(Alg. 1) is the* ***only*** *method that* ***escapes the saddle point*** *and* ***finds true solutions****, compared to existing baselines methods:* PBGD *(Shen & Chen, 2023),* GALET *(Xiao et al., 2023),* MEHA *(Liu et al., 2024), and* SUN-DSBO-SE *(Ma et al., 2026).*

thereby ensuring a singleton solution set for the lower-level problem and well-defined upper-level hypergradient through the implicit function theorem. To address the limitation of the stringent LLSC assumption, a recent line of works in the BLO literature relaxes LLSC to lower-level general convexity (LLGC), i.e., $g(x,\cdot)$ is *convex* w.r.t. y (Cao et al., 2023; Jiang et al., 2023; Liu et al., 2023). Unfortunately, LLGC remains highly restrictive, since most modern learning models are based on deep neural networks, which are highly *nonconvex* and violate the LLGC assumption.

To date, research on bilevel optimization (BLO) problems with nonconvex lower levels (LLNC) remains in its infancy, and existing results are limited. To the best of our knowledge, early attempts to address LLNC-BLO (Kwon et al., 2023; Shen & Chen, 2023; Xiao et al., 2023; Chen et al., 2024; Liu et al., 2024; Jiang et al., 2025a;b; Ma et al., 2026) have achieved only partial success and exhibit various limitations. One key reason for this stagnation is that, under LLNC, finding an optimal solution $y^*(x)$ to the lower-level problem is already intractable in general (typically NP-hard). This difficulty is further compounded by the potential non-uniqueness of $y^*(x)$ and the challenge of establishing well-defined upper-level hypergradients given a $y^*(x)$. Consequently, the aforementioned works either impose specific structural assumptions or consider relaxations of the LLNC-BLO problem. For example, some existing works (Kwon et al., 2023; Jiang et al., 2025a;b) assume bounded domains for the variables (either x, y, or both), which limits their practical applicability. Other works relax the LLNC-BLO formulation by replacing the lower-level optimality condition $y \in \arg\min_y g(x,y)$ with the first-order stationarity condition, i.e., seeking y-solutions that satisfy $\nabla_y g(x,y) = 0$. Although some studies further assume that the lower-level objective $g(x,y)$ satisfies the Polyak–Łojasiewicz (PL) condition (Shen & Chen, 2023; Xiao et al., 2023; Liu et al., 2024; Ma et al., 2026), thereby ensuring that any first-order stationary point (FOSP) is globally optimal. However, the PL condition is restrictive, since nonconvex problems may generally exhibit flat regions, saddle points, or suboptimal local minima, under which the PL condition fails to hold.

In this work, we argue that algorithmic designs using the FOSP relaxation as a surrogate for the lower-level problem could be easily trapped in undesirable saddle points of the lower-level objective (i.e., points that are FOSP but not local minima), thereby yielding poor solutions to the overall LLNC-BLO problem. For example, consider the simple LLNC-BLO problem in Fig. 1 where $x, y \in \mathbb{R}$, $f(x,y) = (x-2)^2 + y^2$, $g(x,y) = \frac{1}{4}y^4 - \frac{1}{2}xy^2$. It can be readily verified that the PL condition fails to hold even for this simple example. Consequently, sequences generated by existing algorithms based on the FOSP surrogate converge only to *saddle points* of $g(x,\cdot)$ (cf. Fig. 1a), leading to solutions that are significantly suboptimal compared to the true optimum (cf. Fig. 1b).

To address the limitations of FOSP-based lower-level surrogates, in this paper, we propose using **second-order stationarity** as a surrogate for LLNC-BLO problems. Mathematically, second-order stationarity requires not only $\nabla_y g(x,y) = 0$, but also that the Hessian with respect to y is positive semidefinite (PSD), i.e., $\lambda_{\min}(\nabla^2_{yy} g(x,y)) \geq 0$. Our rationale for adopting this second-order-

Table 1: Comparison of PROBE *(Alg. 1) with state-of-the-art LLNC-BLO methods.* †*:* BLO Setting *states the assumptions on the lower level of the LLNC-BLO.* ‡*:* Surrogate Problem *refers to the relaxed surrogate of the lower level of the LLNC-BLO problem. Moreau envelope-based surrogates reduce to first-order stationarity under smoothness conditions (Ma et al., 2026).*

Algorithm	BLO Setting†	Surrogate Problem‡	Escape Saddle Points
PBGD	LLNC	First-order stationarity	✗
GALET	LLNC	First-order stationarity	✗
MEHA	LLNC	Moreau envelope	✗
SUN-DSBO-SE	LLNC	Moreau envelope	✗
PROBE **(Ours)**	**LLNC**	**Second-order stationarity**	✓

stationarity-based lower-level surrogate (SOS-LLS) is that a PSD Hessian enforces *local convexity* in the lower-level problem, thereby ruling out undesirable saddle points and guaranteeing at least local optimality with respect to y. As a concrete example, Fig. 1 shows that our PROBE method, based on SOS-LLS, successfully recovers true optimal solutions to the LLNC-BLO problem.

Despite the significant advantage of avoiding undesirable saddle points, solving LLNC-BLO problems through the SOS-LLS reformulation remains largely underexplored, and both algorithm design and analysis are highly nontrivial. This is primarily due to two fundamental challenges: **(i)** the SOS-LLS reformulation leads to a constrained problem that is inherently non-smooth, posing difficulties for algorithmic design; and **(ii)** under the LLNC setting, the upper-level hypergradient is ill-defined, leaving no natural metric to quantify the upper-level stationarity gap. In this paper, we address these challenges through the following key contributions:

- **Theoretical Foundations:** For the first time in the literature, we propose to incorporate second-order stationarity as the lower-level surrogate for LLNC-BLO problems, which allows to avoid undesirable saddle point solutions at the lower level without assuming the overly restrictive PL condition (cf. Table 1). Building on the SOS-LLS reformulation, we introduce a new augmented convergence metric that leverages second-order stationarity to induce local curvature. By progressively shrinking the augmentation parameter, the proposed metric recovers solutions to the original LLNC-BLO problem, thereby providing a principled foundation for both algorithm design and analysis.
- **Algorithmic Design and Analysis:** Based on the proposed SOS-LLS reformulation, we propose PROBE (Perturbed gradient algorithm for bilevel problem), a new algorithm that efficiently probes and guarantees escaping saddle points in the lower level of LLNC-BLO problems. Theoretically, we show that PROBE achieves a finite-time convergence rate of $\mathcal{O}(T^{-\frac{2}{5}})$ at the upper level while guaranteeing local optimality at the lower level for solving LLNC-BLO problems, where T denotes iteration counts.
- **Empirical Validation:** We validate the effectiveness of our PROBE method through a dataset curation task for LLM fine-tuning and a meta-learning task, both of which can be formulated as a BLO problem. Our experimental results demonstrate that PROBE outperforms existing baselines. We further conduct ablation studies to further reveal the significance of the key components in our proposed PROBE method.

2 Related Work

In this section, we provide an overview on two related lines of research: (i) bilevel optimization (BLO), with a focus on methods that address BLO problems in the LLNC settings; and (ii) mechanisms for escaping saddle points in single-level nonconvex optimization theory for machine learning.

1) Bilevel Optimization (BLO): The study of BLO traces its roots back to as early as Bracken & McGill (1973); Ye & Zhu (1995). In recent years, BLO has received significant attention within the machine learning community, primarily driven by the growing necessity of solving nested-structured optimization problems in modern machine learning applications. As mentioned earlier, most existing works in the BLO literature are based on the restrictive LLSC condition (Franceschi et al., 2018; Ghadimi & Wang, 2018; Shaban et al., 2019; Arbel & Mairal, 2021; Ji et al., 2021; Yang et al., 2021; Dagréou et al., 2022), which significantly limits the practical applicability of the BLO framework. Due to the limitations of LLSC, researchers have recently started to relax the LLSC assumption

to the LLGC assumption (Cao et al., 2023; Chen et al., 2023; Jiang et al., 2023; Liu et al., 2023; Shen et al., 2024b). However, the convexity requirement in LLGC still excludes a wide range of practical scenarios. The widening gap between the demand for solving general BLO problems in the LLNC setting and the limited theoretical understanding of this BLO setting motivates the studies of LLNC-BLO. However, this area remains in its infancy, and existing works either impose strong structural assumptions or consider FOSP-based relaxations of the LLNC-BLO problem (Kwon et al., 2023; Shen & Chen, 2023; Xiao et al., 2023; Chen et al., 2024; Liu et al., 2024; Jiang et al., 2025a;b; Ma et al., 2026). As shown in the illustrative example in Figure 1 and Table 1, these methods fail to distinguish local minima from saddle points at the lower level, leading to solutions that are significantly suboptimal compared to the true optimum of the original LLNC-BLO problems.

2) Escaping Saddle Points in Single-Level Nonconvex Optimization: It is well-known that in the absence of convexity, traditional single-level optimization becomes significantly more challenging: finding a global optimum is NP-hard in general (Murty et al., 1987). In fact, even finding a local minimum solution in nonconvex optimization is highly nontrivial. One of the primary challenges stems from the ubiquity of saddle points, particularly in the landscape of high-dimensional objective functions (Dauphin et al., 2014), which is the typical regime where modern machine learning tasks reside. Although second-order methods naturally tackle this issue by actively probing and escaping saddle points, their dimension-dependent convergence rates preclude them from practical large-scale applications. In the machine learning literature, the question of how to escape saddle points in nonconvex optimization was first systematically studied in Ge et al. (2015). However, the convergence rate established in Ge et al. (2015) scales polynomially with the dimension. Inspired by this breakthrough, a significant amount of research effort has been dedicated to designing algorithms that reduce the dimension dependence to scale at a polylog fashion (Agarwal et al., 2017; Jin et al., 2017; Allen-Zhu & Li, 2018; Fang et al., 2018; Jin et al., 2018; Liu et al., 2018; Jin et al., 2021; Zhang & Li, 2021). Another closely related line of research focuses on zeroth-order methods for escaping saddle points (Vlatakis-Gkaragkounis et al., 2019; Zhang et al., 2022a; Zhang & Gu, 2022; Ren et al., 2023). To our knowledge, the only works that incorporated saddle point escaping techniques in the BLO framework are Huang et al. (2025); Chen et al. (2025). However, these studies are limited to the LLSC-based setting and only considered escaping saddle points at the upper level, which are *fundamentally different from* and *much simpler than* our setting, where we investigate saddle point escaping at the lower level for the LLNC setting.

3 The Second-Order Stationarity-Based Lower-Level Surrogate

In this section, we present the second-order stationarity-based lower-level surrogate reformulation for the LLNC-BLO problem. We begin by introducing several standard smoothness conditions that are needed to formally define second-order stationary points:

Assumption 1 (Smoothness of the Upper- and Lower-Level Objective Functions)**.** The upper- and lower-level objective functions f and g are twice differentiable, and there exist positive constants ν, ℓ, ρ, such that for any $z = (x, y)$ and $z' = (x', y')$, it holds that:

$$|f(z) - f(z')| \le \nu\|z - z'\|, \quad \|\nabla f(z) - \nabla f(z')\| \le \ell\|z - z'\|,$$
$$\|\nabla g(z) - \nabla g(z')\| \le \ell\|z - z'\|, \quad \|\nabla^2 g(z) - \nabla^2 g(z')\| \le \rho\|z - z'\|,$$

where $\|\cdot\|$ denotes Euclidean and spectral norm for vectors and matrices hereafter, respectively.

We note that the smoothness assumptions above are not only widely adopted in the BLO literature (Ghadimi & Wang, 2018; Ji et al., 2021; Yang et al., 2021; Xiao et al., 2023), but they have also been numerically observed and verified in large-scale modern machine learning models, such as LLMs (Castin et al., 2023; Malladi et al., 2023; Li et al., 2024).

Equipped with these smoothness properties, we now formally define stationary points, local minima, and saddle points for the lower-level of the LLNC-BLO problem.

Definition 1 (First- and Second-Order Stationary Point)**.** Suppose Assumption 1 holds, and let $\lambda_{\min}(\cdot)$ denote the minimal eigenvalue of a matrix. For any given $x \in \mathbb{R}^p$ and any positive ϵ,

1. y is a first-order stationary point (FOSP) if $\nabla_y g(x, y) = 0$. It is an ϵ-FOSP if $\|\nabla_y g(x, y)\| \le \epsilon$.
2. y is a second-order stationary point (SOSP) if $\nabla_y g(x, y) = 0$ and $\lambda_{\min}(\nabla^2_{yy} g(x, y)) \ge 0$. It is an ϵ-SOSP if $\|\nabla_y g(x, y)\| \le \epsilon$ and $\lambda_{\min}(\nabla^2_{yy} g(x, y)) \ge -\sqrt{\rho\epsilon}$.

We note that the definition of ϵ-second-order stationary point here has also been adopted in the literature (Nesterov & Polyak, 2006; Jin et al., 2017; 2018; Zhang & Li, 2021).

Definition 2 (Local Minimum and Saddle Point)**.** Suppose that g is differentiable. For any given $x \in \mathbb{R}^p$, y is a local minimum if $\nabla_y g(x, y) = 0$ and there exists $\delta > 0$ such that $g(x, y) \leq g(x, y')$ for all $y' \in \mathbb{B}_y(\delta)$, where $\mathbb{B}_y(\delta)$ denotes an open ball of radius δ centered at y. In contrast, y is a saddle point if $\nabla_y g(x, y) = 0$ and it is not a local minimum.

When all saddle points are strict, i.e., $\lambda_{\min}(\nabla^2_{yy} g(x, y)) < 0$ holds for all saddle points, attaining a second-order stationary point is equivalent to reaching a local minimum. In fact, this "strict" property has been extensively verified and documented for saddle points across various nonconvex landscapes (e.g., tensor decomposition or low-rank matrix recovery, see, e.g., (Ge et al., 2015; Bhojanapalli et al., 2016; Ge et al., 2016; Jin et al., 2017; Sun et al., 2018), and see Appendix D for more discussions.) [1].

Given the NP-hardness of finding global optima for general nonconvex objective functions, it is standard practice to target local optimal solutions instead (i.e., points that satisfy the necessary conditions for global optimality). However, as previously noted, simply relaxing the lower-level optimality constraint to merely finding an FOSP (Shen & Chen, 2023; Xiao et al., 2023; Liu et al., 2024; Ma et al., 2026) is problematic, as it introduces the risk of converging to undesirable saddle points. Fortunately, under the strict saddle assumption, identifying an SOSP yields local minima. Thus, we propose to reformulate the LLNC-BLO problem with the following SOSP-based surrogate:

$$\min_{x \in \mathbb{R}^p, y \in \mathbb{R}^q} f(x, y), \qquad \text{subject to } \nabla_y g(x, y) = 0, \;\; \lambda_{\min}(\nabla^2_{yy} g(x, y)) \geq 0. \tag{2}$$

4 The Proposed PROBE Algorithm

Based on the SOSP-based lower-level surrogate for LLNC-BLO problems, in this section, we will investigate how to design an efficient algorithm for solving in Eq. 2. We start with defining an appropriate metric to measure the convergence error in designing algorithms for solving Eq. 2.

Suppose that $(\tilde{x}, \tilde{y})$ is an ϵ'-SOSP for $g(\tilde{x}, \cdot)$ for some positive ϵ'. According to Definition 1, we have that $\nabla^2_{yy} g(\tilde{x}, \tilde{y}) + (\sqrt{\rho\epsilon'} + \sigma) I_q$ is positive definite for any arbitrarily small positive value $\sigma > 0$ (which can be adaptively selected according to ϵ'). It then follows that the augmented objective function $g(\tilde{x}, y) + \frac{\mu}{2}\|y - \tilde{y}\|^2$ is locally $\frac{\sigma}{2}$-strongly convex within a sufficiently small neighborhood of $\tilde{y}$, where the parameter $\mu \triangleq \mu(\epsilon', \sigma) := \sqrt{\rho\epsilon'} + \sigma$. By leveraging this local quadratic curvature, we can define the following *μ-augmented implicit function* with a small positive radius $r' = \frac{\sigma}{\rho}$:

$$\Phi_\mu(\tilde{x}) := f(\tilde{x}, y^*_\mu(\tilde{x})), \;\; \text{where } y^*_\mu(\tilde{x}) := \underset{y \in \mathbb{B}_{\tilde{y}}(r')}{\arg\min} \{g(\tilde{x}, y) + \frac{\mu}{2}\|y - \tilde{y}\|^2\},$$

where the existence and uniqueness of $y^*_\mu(\tilde{x})$ can be guaranteed by the local strong convexity. Indeed, the introduction of μ-regularization restores a locally strongly convex curvature, thereby enabling the use of existing techniques from the LLSC-BLO setting, such as hypergradient-based methods (Ghadimi & Wang, 2018; Grazzi et al., 2020; Ji et al., 2021; Huang et al., 2025). Consequently, following from the Implicit Function Theorem (IFT), the gradient of $\Phi_\mu(\tilde{x})$, referred to as the μ-augmented hypergradient in this paper, can be computed as:

$$\begin{aligned}\nabla\Phi_\mu(\tilde{x}) &= \nabla_x f(\tilde{x}, y^*_\mu(\tilde{x})) + \frac{dy^*_\mu(\tilde{x})}{dx} \nabla_y f(\tilde{x}, y^*_\mu(\tilde{x})) \\ &= \nabla_x f(\tilde{x}, y^*_\mu(\tilde{x})) - \nabla^2_{xy} g(\tilde{x}, y^*_\mu(\tilde{x})) \left[\nabla^2_{yy} g(\tilde{x}, y^*_\mu(\tilde{x})) + \mu I_q\right]^{-1} \nabla_y f(\tilde{x}, y^*_\mu(\tilde{x})).\end{aligned}$$

The μ-augmented hypergradient naturally gives rise to the following convergence metric:

[1] While finding an SOSP does not theoretically preclude convergence to non-strict saddle points that could occur in highly overparameterized deep learning landscapes, it remains strictly far more advantageous than relying on FOSP. SOSP guarantees the evasion of all strict saddle points, including local maxima, by explicitly exploiting directions of negative curvature. Furthermore, because escaping non-strict saddle points generally requires computationally intractable higher-order derivatives, targeting an SOSP represents the strongest practically verifiable optimality condition. Empirically, the non-strict saddles satisfying SOSP conditions in deep networks often lie on flat manifolds that yield highly desirable generalization properties.

Algorithm 1: The overall PROBE algorithmic framework.

Input: Iteration T, K_t, N_t, initial points x_0, y_0, v_0, step-sizes α_t, β_t, aggregation parameters μ_t, and PGD parameters $r_t, \varrho_t, \Delta_{G,t}, \Delta_{K,t}$, where $t \in \{0, \ldots, T-1\}$

for $t = 0, \ldots, T-1$ **do**

  Update $y_{t+1} \leftarrow \mathrm{PGD}\big(K_t, x_t, y_t, \beta_t, r_t, \varrho_t, \Delta_{G,t}, \Delta_{K,t}\big)$

  Solve $\big(\nabla^2_{yy} g(x_t, y_{t+1}) + \mu_t I_q\big) v = \nabla_y f(x_t, y_{t+1})$ from v_t by N_t-step Conjugate Gradient with a warm start to get v_{t+1}

  Update $x_{t+1} \leftarrow x_t - \alpha_t \widehat{\nabla} \Phi_{\mu_t}(x_t)$, where $\widehat{\nabla} \Phi_{\mu_t}(x_t) = \nabla_x f(x_t, y_{t+1}) - \nabla^2_{xy} g(x_t, y_{t+1}) v_{t+1}$

return (x_T, y_T)

Definition 3 (μ-Augmented Hypergradient-Based Convergence Metric)**.** Suppose y is an ϵ'-SOSP for a given x (as defined in Def. 1) and all saddle points are strict. For any positive ϵ, $\mu = \sqrt{\rho\epsilon'} + \sigma$, (x, y) is said to be an (ϵ, μ)-stationary solution to Eq. 2 if it satisfies: $\Pi(x, y) := \|\nabla \Phi_\mu(x)\|^2 + \epsilon' \le \epsilon$.

With the formal definition of this convergence metric and the SOSP-based lower-level surrogate for the LLNC-BLO problem, the following key question naturally arises:

Question: Is the μ-augmented hypergradient a good estimation of the "steepest descent update direction" of the original LLNC-BLO problem in terms of x? More importantly, how to control the μ-parameter in the SOSP-based surrogate in order to ensure that we still converge to a good solution to the original LLNC-BLO problem with a lower-level local optimality guarantee?

Although this question is challenging to answer in the highly complex LLNC setting, one may obtain some high-level qualitative understanding by measuring the "bias" of the μ-augmented hypergradient when a hypergradient is well-defined (note that in LLNC-BLO, a hypergradient could be ill-posed if the Hessian of the lower-level problem is rank-deficient). More specifically, note that if the sequence generated by an algorithm is able to escape strict saddle points, it typically reaches a local minimum, where the lower-level objective function usually exhibits sufficient local curvature in a bounded local neighborhood, so that its landscape can be bounded from below by a quadratic function. Consider a scenario in which $\nabla^2_{yy} g(\tilde{x}, y^*_\mu(\tilde{x})) \succeq \mu_0 I_q$ holds for some $\mu_0 > 0$, so that a hypergradient is well-defined. Due to the fact that $A^{-1} - B^{-1} = A^{-1}(B - A)B^{-1}$, we can bound the discrepancy between the μ-augmented hypergradient and the true hypergradient as follows:

$$
\begin{aligned}
\|\nabla \Phi_\mu(\tilde{x}) - \nabla \Phi_0(\tilde{x})\| &= \left\| \nabla^2_{xy} g \left(\left[\nabla^2_{yy} g\right]^{-1} (\mu I_q) \left[\nabla^2_{yy} g + \mu I_q\right]^{-1} \right) \nabla_y f \right\| \\
&\le \|\nabla^2_{xy} g\| \cdot \left\| \left[\nabla^2_{yy} g\right]^{-1} \right\| \cdot \mu \cdot \left\| \left[\nabla^2_{yy} g + \mu I_q\right]^{-1} \right\| \cdot \|\nabla_y f\| \\
&\overset{(a)}{\le} \ell \cdot \frac{1}{\mu_0} \cdot \mu \cdot \frac{1}{\mu_0 + \mu} \cdot \nu \le \frac{\ell \nu}{\mu_0^2} (\sqrt{\rho\epsilon'} + \sigma) = \mathcal{O}(\sqrt{\epsilon'} + \sigma),
\end{aligned}
$$

where (a) follows from Assumption 1. This shows that the μ-augmented hypergradient incurs an approximation bias that scales at most linearly with both the square root of the SOSP violation $\sqrt{\epsilon'}$ and the augmentation parameter σ. To mitigate the impact of such "bias", we can adaptively shrink both ϵ_t and σ_t in the design of our algorithm to progressively refine the lower-level solutions. Specifically, by using a dynamic sequence $\{\mu_t = \sqrt{\rho\epsilon_t} + \sigma_t\}_t$ that approaches 0^+ as $t \to \infty$, the "bias" can be gradually eliminated to induce convergence to a solution with vanishing bias to the original LLNC-BLO problem. Based on this intuition, we propose PROBE (Perturbed gradient algorithm for bilevel problem), which is summarized in Algorithms 1 and 2. In what follows, we will first describe the basic idea and the key steps of PROBE. The theoretical convergence analysis of PROBE will be provided in Section 5.

1) Basic Idea: PROBE employs a double-loop framework, where the inner loop is dedicated to finding an ϵ_t-SOSP of the lower-level objective $g(x_t, \cdot)$ for a given upper-level variable x_t in step t, while the outer loop utilizes this approximation to construct a μ_t-augmented hypergradient, which guides the update of the upper-level variable x_t towards the minimum of $\Phi_{\mu_t}(\cdot)$ with a shrinking μ_t.

2) SOSP Approximation via PGD (Inner-Loop): The inner-loop (Algorithm 2) employs Perturbed Gradient Descent (PGD, Jin et al. (2017)) to identify SOSP approximations. In order to escape saddle

Algorithm 2: The perturbed gradient descent subroutine PGD($K, x, y^0, \beta, r, \varrho, \Delta_G, \Delta_K$).

$\tau \leftarrow -\Delta_K - 1$
for $k = 0, \ldots, K-1$ **do**
  if $\|\nabla_y g(x, y^k)\| \le \varrho$ ***and*** $k - \tau > \Delta_K$ **then**
    $\tilde{y} \leftarrow y^k, \quad \tau \leftarrow k$ `// Store anchor point & Update timestamp`
    $\xi \sim \text{Unif}(\mathbb{B}_{\mathbf{0}}(r)), \quad y^k \leftarrow y^k + \xi$ `// Inject uniform noise`
  if $k - \tau = \Delta_K$ ***and*** $g(x, y^k) - g(x, \tilde{y}) > -\Delta_G$ **then**
    return $\tilde{y}$ `// Return confirmed SOSP`
  $y^{k+1} \leftarrow y^k - \beta \nabla_y g(x, y^k)$

points, PGD monitors the gradient: if $\|\nabla_y g(x, y^k)\| \le \varrho$, it injects a uniform noise $\xi \sim \text{Unif}(\mathbb{B}_{\mathbf{0}}(r))$ for a more aggressive exploration, where r denotes the radius of a ball and will be selected later. After several steps, the algorithm confirms that the iterate has reached an approximate SOSP if the function value has an insufficient reduction (i.e., $g(x, y^k) - g(x, \tilde{y}) > -\Delta_G$). Finally, the inner-loop returns the identified ϵ_t-SOSP to the outer-loop at each outer iteration t by adaptively and properly selecting parameters $(K_t, \varrho_t, \Delta_{G,t}, \Delta_{K,t})$.

3) μ_t-Hypergradient Estimation and Update (Outer-Loop): Building upon the lower-level solution y_{t+1} identified by the inner-loop, the outer-loop (PROBE , Algorithm 1) proceeds to update the upper-level variable. In the μ_t-augmented hypergradient, the primary computational bottleneck lies in estimating the Hessian inverse $[\nabla^2_{yy} g(x_t, y_{t+1}) + \mu_t I_q]^{-1}$. We address this by employing the Conjugate Gradient (CG) method (Grazzi et al., 2020; Ji et al., 2021), an inversion-free approach that only queries Hessian-vector products. This allows PROBE to efficiently compute the adjoint variable v_{t+1} within N_t iterations at every time slot t. Furthermore, we equip the N_t-step CG with a warm start mechanism (conducting CG in the t-round beginning from the last output v_t), which significantly accelerates convergence to reach a faster rate. Finally, using the estimated μ_t-augmented hypergradient $\widehat{\nabla}\Phi_{\mu_t}(x_t)$, we perform a gradient descent step to update upper-level variable to x_{t+1}. As will be mentioned in the next section, to ensure the convergence to a good solution of the original LLNC-BLO problem, we dynamically increase and decrease the parameters K_t and σ_t, respectively, to shrink $\mu_t = \sqrt{\rho \epsilon_t} + \sigma_t$ toward 0 at an appropriate rate.

5 Finite-Time Convergence Analysis of PROBE

In this section, we will conduct theoretical convergence analysis of our proposed PROBE algorithm. Toward this end, we first restate the iteration complexity performance guarantee of the PGD method (Jin et al., 2017), which is adapted in the context of LLNC-BLO in Algorithm 2.

Lemma 1 (Iteration Complexity of PGD). *Under Assumption 1, with hyperparameter choices in Appendix B, for any $\delta' \in (0,1), \epsilon_t \le \ell^2/\rho, \Delta_{g,t} \ge g(x_t, y_t) - \inf_y g(x_t, y)$, with probability at least $1 - \delta'$, Algorithm 2 outputs an ϵ_t-SOSP with iterations no more than $K_t = \mathcal{O}\big(\frac{\ell \Delta_{g,t}}{\epsilon_t^2} \log^4\big(\frac{q \ell \Delta_{g,t}}{\epsilon_t^2 \delta'}\big)\big)$.*

We note that, besides PGD, several algorithms have been proposed to escape saddle points while maintaining nearly dimension-free convergence rates (Jin et al., 2018; 2021; Zhang & Li, 2021). While these approaches offer improvements in iteration complexity to some degree (from ϵ_t^{-2} to $\epsilon_t^{-1.75}$ at most), they typically introduce additional algorithmic modules, such as Negative Curvature Finding (NCF) and Nesterov-style acceleration, which significantly complicate the implementation of the inner-loop algorithm. Moreover, rather than offering a "last-iteration" convergence guarantee, these accelerated methods only guarantee an approximate SOSP being visited at some point within the trajectory. This renders them ill-suited for algorithm designs for LLNC-BLO, which requires a terminal SOSP estimate. We are now ready to state the main convergence result for PROBE .

Theorem 1. *By selecting $K_t = \widetilde{\Theta}\big((t+1)^{\frac{8}{5}}\big)$, $\sigma_t = \Theta\big((t+1)^{-p}\big)$ and $\alpha_t = \Theta\big((t+1)^{-q}\big)$ with $p \in (0, \frac{1}{3}), 3p \le q < 1$, for any $\delta \in (0,1)$, with probability at least $1 - \delta$, the sequence $\{x_t, y_t\}_t$ generated by Algorithm 1 satisfies the following convergence guarantee: $\min_{\frac{T}{2} \le t \le T} \Pi(x_t, y_t) = \mathcal{O}(\frac{1}{T^{1-q}} + \frac{1}{T^{2p}} + \frac{1}{T^{\frac{4}{5}}})$. By selecting $(p, q) = (\frac{1}{5}, \frac{3}{5})$, we have $\min_{\frac{T}{2} \le t \le T} \Pi(x_t, y_t) = \mathcal{O}\big(T^{-\frac{2}{5}}\big)$.*

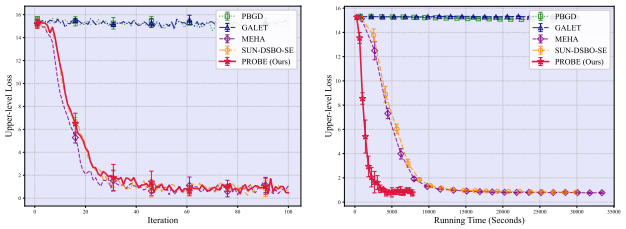
Proof Sketch. Our convergence analysis is organized in three key steps (proof details of Theorem 1 are relegated to Appendix B due to limited space):

- **Step 1) Bounding the Variable Drifts:** We first control the tracking errors of the lower-level variable y and the adjoint multiplier v. A major challenge here is decomposing and bounding several intertwined gaps: the inner-iteration solver error (i.e., taking y for example hereafter, $\|y_{t+1} - y^*_{\mu_t}(x_t)\|$), and the parameter-induced shift across varying μ (i.e., $\|y^*_{\mu_{t+1}}(x_t) - y^*_{\mu_t}(x_t)\|$).
- **Step 2) Controlling the Augmented Implicit Function Shift:** Next, we analyze the dynamics of $\Phi_{\mu_{t+1}}(x_{t+1}) - \Phi_{\mu_t}(x_t)$, which is challenging due to simultaneous updates in both the variable x and the parameter μ. We decompose this shift into two components: (i) the contribution from the dynamics of x, which can be controlled using the smoothness of the augmented implicit function and, consequently, the descent lemma; and (ii) the contribution induced by the drift in μ, which can be converted to the drift in y that has already been controlled in the previous step.
- **Step 3) Lyapunov Analysis and Telescoping:** Finally, building upon the aforementioned results, we construct a dynamic Lyapunov function to decouple the descent of Φ_{μ_t} from the tracking errors. By using a telescoping sum, we show that all error terms are bounded by the sum of a constant and a convergent series, thereby establishing the finite-time convergence rate. □

Three remarks on Theorem 1 are in order: **First**, to our knowledge, this work is the *first* that incorporates second-order stationarity as a surrogate for LLNC-BLO and propose an algorithm with a finite-time convergence rate of $\mathcal{O}(T^{-\frac{2}{5}})$ with lower-level SOSP guarantee. While existing LLNC-BLO methods (Shen & Chen, 2023; Xiao et al., 2023; Liu et al., 2024; Ma et al., 2026) also provided finite-time convergence analysis, a direct comparison with them is difficult due to the fact that most of these works assume the PL condition, which is not needed in this work. **Second**, there is a trade-off in choosing $\sigma_t = \sigma_0(t+1)^{-p}$: a larger p-value yields faster convergence rate $\mathcal{O}(T^{-2p})$ in Theorem 1. However, the condition $q \geq 3p$ implies a small learning rate $\alpha_t = \alpha_0(t+1)^{-q}$. Also, the smoothness coefficient of Φ_{μ_t} scales as $L_t = \Theta(\sigma_t^{-3})$, implying that $\alpha_t = \Theta(L_t^{-1}) = \Theta((t+1)^{-3p})$ when $q = 3p$ is required for a fast convergence, which in turn necessitates a relatively small p. To balance these two directions, we choose $(p, q) = (\frac{1}{5}, \frac{3}{5})$ to achieve the most efficient convergence rate $\mathcal{O}(T^{-\frac{2}{5}})$. **Third**, instead of using a fixed number of inner-loop iterations K, PROBE gradually increases K_t over time, leading to a *two-timescale* algorithmic design. Specifically, we introduce a μ-augmentation to allow a well-defined hypergradient in LLNC-BLO. This term introduces an error that can be controlled by gradually letting $\{\mu_t\} \to 0$. Hence, K_t needs to be increasing rather than fixed to decrease the inner-loop error ϵ_t and thus μ_t, leading to this two-timescale design.

6 Experimental Results

In this section, we will validate the efficacy of our PROBE method by using a data curation task for LLM fine-tuning. Specifically, we first curate a high quality dataset from multiple datasets of unknown qualities, which is then used to fine-tune an LLM to mitigate toxicity, reduce verbosity, and improve coherence. This data curation task can be formulated as an LLNC-BLO problem (Shen et al., 2024a; Pan et al., 2025), where the lower level aims to determine the mixing proportion of each dataset, and the upper level further fine-tunes the model as a validation process:



(a) Loss vs Iteration. *(b) Loss vs Running time.*

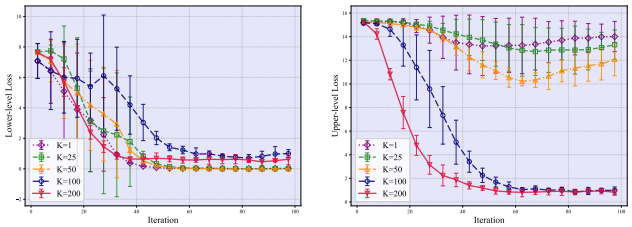
Figure 2: Baseline comparison.

$$\min_{x,y} \sum_{i=1}^{n_0} \mathcal{L}\big(r_{y(x)}(p_i), r_i\big), \quad \text{s.t. } y(x) \in \arg\min_{y} \sum_{j=1}^{\mathcal{D}} \text{SoftMax}_x(j) \sum_{i=1}^{n_j} \mathcal{L}\big(r_y(p_i), r_i\big),$$

where x is the weight vector for training datasets, y represents the LLM parameters, (p, r) denotes a prompt-response pair, $\mathcal{L}$ is the loss, $\text{SoftMax}_x(j) = \exp(x_j)/\sum_{j'} \exp(x_{j'})$, index 0 stands for validation set, and $1, \ldots, \mathcal{D}$ are training indices. Clearly, the lower-level problem is nonconvex. We conduct experiments on the HelpSteer (Wang et al., 2024) dataset, which incorporates data of varying quality. We employ Llama-3.2-3B-Instruct (Meta, 2024) as the base model. More implementation details are relegated to Appendix A due to space limitation.

1) Baseline Comparison: We compare our PROBE method against state-of-the-art LLNC-BLO baselines: PBGD (Shen & Chen, 2023), GALET (Xiao et al., 2023), MEHA (Liu et al., 2024), and SUN-DSBO-SE (Ma et al., 2026). Figure 2 illustrates the upper-level loss along with the standard error bars over 5 trials, which shows that PBGD and GALET fail to converge on this data curation task. As detailed in Appendix A, PBGD performs well at the lower level. However, its oversimplified FOSP-based surrogate could not avoid being trapped at undesirable saddle points at the lower level, leading to weak performance at the upper level. In contrast, MEHA, SUN-DSBO-SE, and PROBE converge to much better solutions at upper level. Although the theoretical convergence rates of these methods are not directly comparable due to different metrics, PROBE outperforms all baselines in terms of wall-clock running time, demonstrating its efficiency.

2) Hyper-parameter Experimentation and Ablation Study: In this experiment, we study how the selections of hyperparameters K and μ impact the performance of PROBE, as shown in Fig. 3. For simplicity, we fix K and μ for each experiment. As shown in Figures 3a and 3b, increasing the inner-loop PGD iterations, K, yields a significant improvement in the upper-level loss. This aligns with our theoretical insights: a small K restricts PGD's ability to probe and escape saddle points. Consequently, the lower-level problem stagnates at a suboptimal position, which in turn leads to unsatisfactory upper-level performance. In contrast, when K is sufficiently large, PROBE reliably finds a lower-level SOSP, yielding substantial overall performance improvements. Figures 3c and 3d show the influence of the augmentation parameter μ, both of which show that the larger μ facilitates a slightly lower loss value. Also, Figure 3d shows that the trajectory fluctuates dramatically when μ is insufficient to restore the Hessian positive definiteness. This highlights the important tradeoff and provides a guidance in selecting μ.

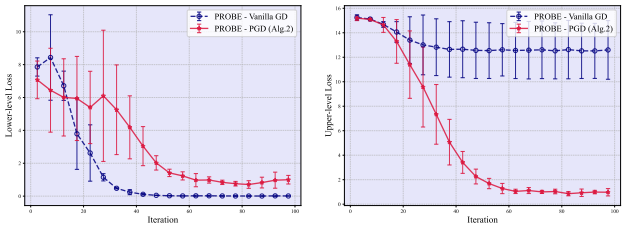


(a) Lower-level loss with K. *(b) Upper-level loss with K.*

(c) Lower-level loss with μ. *(d) Upper-level loss with μ.*

Figure 3: Hyperparameter investigation.

We also conduct the following ablation study by replacing the PGD method in Algorithm 2 with the vanilla gradient descent method, keeping all other components of PROBE unchanged. As shown in Figure 4, without saddle points escaping, PROBE is easily trapped in solutions seemingly better at the lower level but are suboptimal at the upper level, highlighting the importance of our SOSP-based lower-level surrogate and the efficacy of our proposed PROBE . Due to space limitation, we provide further numerical details and a meta-learning experiment in Appendix A.



(a) Lower-level loss. *(b) Upper-level loss.*

Figure 4: Ablation Study.

7 Conclusion

In this paper, we investigated bilevel optimization with non-convex lower levels. We showed that existing first-order stationarity-based surrogates often fail to escape lower-level saddle points. To address this challenge, we proposed a second-order stationarity-based surrogate, based on which we developed a new algorithm called PROBE and established its finite-time convergence rate. We conducted experiments on dataset curation tasks for LLM finetuning to validate our PROBE algorithm. We note that the focus of this work is on deterministic LLNC-BLO problems. Future directions include extending our framework into stochastic LLNC-BLO problems.

AI USE STATEMENT

We used generative AI tools solely to polish the language. All AI-assisted revisions were reviewed by the authors, who take full responsibility for the final content of this work.

ETHICS STATEMENT

We have not identified any specific ethical concerns that require additional discussion in this work.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide the code for our experiments in the supplementary materials and describe the implementation details in Section A.

REFERENCES

Naman Agarwal, Zeyuan Allen-Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma. Finding approximate local minima faster than gradient descent. In *Proceedings of the 49th annual ACM SIGACT symposium on theory of computing*, pp. 1195–1199, 2017.

Zeyuan Allen-Zhu and Yuanzhi Li. Neon2: Finding local minima via first-order oracles. *Advances in Neural Information Processing Systems*, 31, 2018.

Michael Arbel and Julien Mairal. Amortized implicit differentiation for stochastic bilevel optimization. *arXiv preprint arXiv:2111.14580*, 2021.

Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Global optimality of local search for low rank matrix recovery. *Advances in Neural Information Processing Systems*, 29, 2016.

Jerome Bracken and James T McGill. Mathematical programs with optimization problems in the constraints. *Operations research*, 21(1):37–44, 1973.

Jincheng Cao, Ruichen Jiang, Nazanin Abolfazli, Erfan Yazdandoost Hamedani, and Aryan Mokhtari. Projection-free methods for stochastic simple bilevel optimization with convex lower-level problem. *Advances in Neural Information Processing Systems*, 36:6105–6131, 2023.

Valérie Castin, Pierre Ablin, and Gabriel Peyré. How smooth is attention? *arXiv preprint arXiv:2312.14820*, 2023.

Lesi Chen, Jing Xu, and Jingzhao Zhang. Bilevel optimization without lower-level strong convexity from the hyper-objective perspective. 2023.

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*, pp. 947–980. PMLR, 2024.

Lesi Chen, Yaohua Ma, and Jingzhao Zhang. Near-optimal nonconvex-strongly-convex bilevel optimization with fully first-order oracles. *Journal of Machine Learning Research*, 26(109):1–56, 2025.

Mathieu Dagréou, Pierre Ablin, Samuel Vaiter, and Thomas Moreau. A framework for bilevel optimization that enables stochastic and global variance reduction algorithms. *Advances in Neural Information Processing Systems*, 35:26698–26710, 2022.

Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *Advances in neural information processing systems*, 27, 2014.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in neural information processing systems*, 31, 2018.

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*, pp. 1568–1577. PMLR, 2018.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on learning theory*, pp. 797–842. PMLR, 2015.

Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. *Advances in neural information processing systems*, 29, 2016.

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

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

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.

Yifan Hu, Jie Wang, Yao Xie, Andreas Krause, and Daniel Kuhn. Contextual stochastic bilevel optimization. *Advances in Neural Information Processing Systems*, 36:78412–78434, 2023.

Minhui Huang, Xuxing Chen, Kaiyi Ji, Shiqian Ma, and Lifeng Lai. Efficiently escaping saddle points in bilevel optimization. *Journal of machine learning research*, 26(1):1–61, 2025.

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

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

Xiaotian Jiang, Jiaxiang Li, Jiawen Bi, Mingyi Hong, and Shuzhong Zhang. A correspondence-driven approach for bilevel decision-making with nonconvex lower-level problems. *arXiv preprint arXiv:2509.01148*, 2025a.

Xiaotian Jiang, Ioannis Tsaknakis, Prashant Khanduri, and Mingyi Hong. A discretization approach for bilevel optimization with low-dimensional and non-convex lower-level. *arXiv preprint arXiv:2505.10830*, 2025b.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *International conference on machine learning*, pp. 1724–1732. PMLR, 2017.

Chi Jin, Praneeth Netrapalli, and Michael I Jordan. Accelerated gradient descent escapes saddle points faster than gradient descent. In *Conference on learning theory*, pp. 1042–1085. PMLR, 2018.

Chi Jin, Praneeth Netrapalli, Rong Ge, Sham M Kakade, and Michael I Jordan. On nonconvex optimization for machine learning: Gradients, stochasticity, and saddle points. *Journal of the ACM (JACM)*, 68(2):1–29, 2021.

Mikoto Kudo, Takumi Tanabe, Akifumi Wachi, and Youhei Akimoto. Sample-efficient hypergradient estimation for decentralized bi-level reinforcement learning. *arXiv preprint arXiv:2603.14867*, 2026.

Jeongyeol Kwon, Dohyun Kwon, Stephen Wright, and Robert Nowak. On penalty methods for nonconvex bilevel optimization and first-order stochastic approximation. *arXiv preprint arXiv:2309.01753*, 2023.

Jiaxiang Li, Siliang Zeng, Hoi-To Wai, Chenliang Li, Alfredo Garcia, and Mingyi Hong. Getting more juice out of the sft data: Reward learning from human demonstration improves sft for llm alignment. *Advances in Neural Information Processing Systems*, 37:124292–124318, 2024.

Mingrui Liu, Zhe Li, Xiaoyu Wang, Jinfeng Yi, and Tianbao Yang. Adaptive negative curvature descent with applications in non-convex optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

Risheng Liu, Jiaxin Gao, Jin Zhang, Deyu Meng, and Zhouchen Lin. Investigating bi-level 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.

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 *International Conference on Machine Learning*, pp. 21839–21866. PMLR, 2023.

Risheng Liu, Zhu Liu, Wei Yao, Shangzhi Zeng, and Jin Zhang. Moreau envelope for nonconvex bi-level optimization: A single-loop and hessian-free solution strategy. *arXiv preprint arXiv:2405.09927*, 2024.

Yaoshuai Ma, Xiao Wang, Wei Yao, and Jin Zhang. Sun-dsbo: A structured unified framework for nonconvex decentralized stochastic bilevel optimization. *arXiv preprint arXiv:2601.22682*, 2026.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. *Advances in Neural Information Processing Systems*, 36:53038–53075, 2023.

Meta. meta-llama/llama-3.2-3b-instruct. 2024. URL https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct.

Katta G Murty, Santosh N Kabadi, et al. Some np-complete problems in quadratic and nonlinear programming. *Mathematical programming*, 39(2):117–129, 1987.

Yurii Nesterov and Boris T Polyak. Cubic regularization of newton method and its global performance. *Mathematical programming*, 108(1):177–205, 2006.

Rui Pan, Dylan Zhang, Hanning Zhang, Xingyuan Pan, Minrui Xu, Jipeng Zhang, Renjie Pi, Xiaoyu Wang, and Tong Zhang. Scalebio: Scalable bilevel optimization for llm data reweighting. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31959–31982, 2025.

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.

Zhaolin Ren, Yujie Tang, and Na Li. Escaping saddle points in zeroth-order optimization: the power of two-point estimators. In *International Conference on Machine Learning*, pp. 28914–28975. PMLR, 2023.

Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. Truncated back-propagation for bilevel optimization. In *The 22nd international conference on artificial intelligence and statistics*, pp. 1723–1732. PMLR, 2019.

Han Shen and Tianyi Chen. On penalty-based bilevel gradient descent method. In *International conference on machine learning*, pp. 30992–31015. PMLR, 2023.

Han Shen, Pin-Yu Chen, Payel Das, and Tianyi Chen. Seal: Safety-enhanced aligned llm fine-tuning via bilevel data selection. *arXiv preprint arXiv:2410.07471*, 2024a.

Han Shen, Santiago Paternain, Gaowen Liu, Ramana Kompella, and Tianyi Chen. A method for bilevel optimization with convex lower-level problem. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 9426–9430. IEEE, 2024b.

Ju Sun, Qing Qu, and John Wright. A geometric analysis of phase retrieval. *Foundations of Computational Mathematics*, 18(5):1131–1198, 2018.

Emmanouil-Vasileios Vlatakis-Gkaragkounis, Lampros Flokas, and Georgios Piliouras. Efficiently avoiding saddle points with zero order methods: No gradients required. *Advances in neural information processing systems*, 32, 2019.

Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makesh Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Scowcroft, Neel Kant, Aidan Swope, et al. Helpsteer: Multi-attribute helpfulness dataset for steerlm. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3371–3384, 2024.

Quan Xiao, Songtao Lu, and Tianyi Chen. A generalized alternating method for bilevel learning under the polyak-{\L} ojasiewicz condition. *arXiv preprint arXiv:2306.02422*, 2023.

Junjie Yang, Kaiyi Ji, and Yingbin Liang. Provably faster algorithms for bilevel optimization. *Advances in Neural Information Processing Systems*, 34:13670–13682, 2021.

Jane J Ye and DL Zhu. Optimality conditions for bilevel programming problems. *Optimization*, 33 (1):9–27, 1995.

Chenyi Zhang and Tongyang Li. Escape saddle points by a simple gradient-descent based algorithm. *Advances in Neural Information Processing Systems*, 34:8545–8556, 2021.

Hualin Zhang and Bin Gu. Faster gradient-free methods for escaping saddle points. In *The Eleventh International Conference on Learning Representations*, 2022.

Hualin Zhang, Huan Xiong, and Bin Gu. Zeroth-order negative curvature finding: Escaping saddle points without gradients. *Advances in Neural Information Processing Systems*, 35:38332–38344, 2022a.

Yihua Zhang, Guanhua Zhang, Prashant Khanduri, Mingyi Hong, Shiyu Chang, and Sijia Liu. Revisiting and advancing fast adversarial training through the lens of bi-level optimization. In *International Conference on Machine Learning*, pp. 26693–26712. PMLR, 2022b.

Yihua Zhang, Prashant Khanduri, Ioannis Tsaknakis, Yuguang Yao, Mingyi Hong, and Sijia Liu. An introduction to bilevel optimization: Foundations and applications in signal processing and machine learning. *IEEE Signal Processing Magazine*, 41(1):38–59, 2024.

A Additional Numerical Results

A.1 Meta-Learning Experiments

Meta-learning seeks to improve model generalization and is naturally formulated as a BLO problem (Franceschi et al., 2018; Ji et al., 2021; Hu et al., 2023; Qin et al., 2025). Specifically, the lower level implements task-specific adaptation on the training data, while the upper level enhances the representation utility to ensure generalization ability of the model.

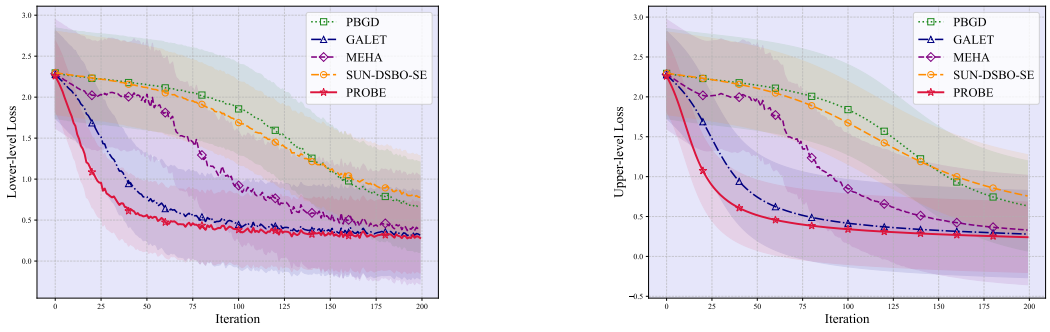
We train a multi-layer perceptron (MLP, corresponding to x) with depth 2, width 512 and ReLU activation, along with an additional 256-dimensional linear layer (corresponding to y) on MNIST dataset. In particular, we construct 5 MNIST heterogeneous training datasets, each containing 80% of a selected pair of digits (e.g., $(0, 1)$) and 20% of other digit pairs. The validation set are constructed from the MNIST test split, using the same ratio of five digit-pair tasks. The formal formulation is given as follows:

$$\min_{x,y} \sum_{s\in\mathcal{D}_0} \mathcal{L}\big((x\circ y)(s), l(s)\big)$$

$$\text{subject to } y \in \mathcal{S}(x) := \arg\min_{y} \sum_{j=1}^{5} \sum_{s\in\mathcal{D}_j} \mathcal{L}\big((x\circ y)(s), l(s)\big),$$

where s denotes the digit sample, $l(s)$ denotes its label, $x \circ y(s)$ denotes the model output, $\mathcal{L}$ denotes the cross-entropy loss function, and $\mathcal{D}_j, j \in \{0, \ldots, 5\}$ are the datasets. Since the loss function is convex, and y corresponds to a linear layer, the lower-level problem is convex (but not strongly convex) w.r.t. y. We consider this meta-learning task to demonstrate the capability of PROBE in the LLGC setting.

As shown in Figure 5, all baselines and our PROBE converge at both upper and lower levels. All baseline methods succeed in this meta-learning task because the FOSP-based relaxed surrogate aligns exactly with the original BLO problem in this LLGC environment, preventing them from getting trapped at saddle points (In fact, no saddle points exist due to the convexity). PROBE, on the other hand, still achieves the fastest convergence, demonstrating both its validity in the LLGC setting and its consistent efficiency.

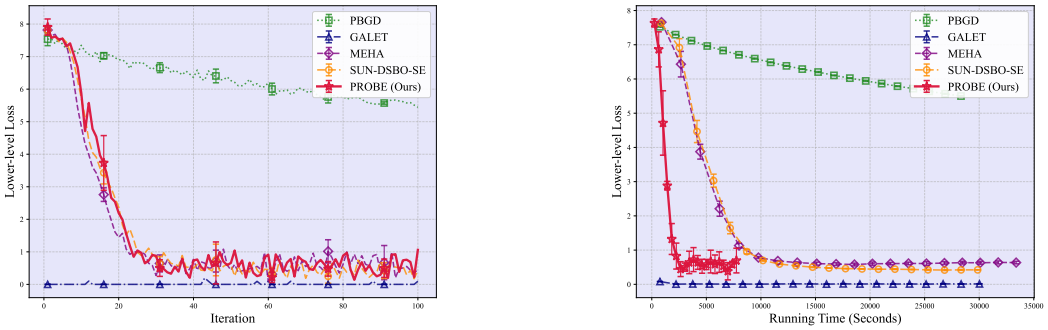


(a) Lower-level loss. *(b) Upper-level loss.*

Figure 5: Baseline comparison in meta-learning task.

A.2 Implementation Details and Additional Results on Data Curation Task

In the data curation task, we fine-tune an LLM, Llama-3.2-3B-Instruct (Meta, 2024) model, on the HelpSteer (Wang et al., 2024) dataset, which naturally contains prompts with responses of varying quality. During training, corresponding to the lower-level problem in our formulation, the LLM learns to distinguish between two datasets and is expected to assign higher weights to the one with higher response quality. During validation, corresponding to the upper-level problem, the performance of the fine-tuned LLM is evaluated on high-quality prompt-response data. Specifically, the datasets are constructed as follows: for each response data in HelpSteer, we compute its average score,



(a) Loss vs Iteration. *(b) Loss vs Running time.*

Figure 6: Baseline (lower-level) comparison.

and include the corresponding prompt-response pair in the high-quality set if $s \geq 2.5$, or in the low-quality set if $s \leq 2$ (the only validation set merely include high-quality validation data). Llama-3.2-3B-Instruct is trained using LoRA technique with rank $= 8$, and all data curation experiments are conducted on a cluster of 2 NVIDIA H200 GPUs (approximately 140GB each) using PyTorch's DistributedDataParallel.

The hyperparameter selection is detailed as follows. The batch size is set to 32. For a fair comparison, we fix the total number of iterations to $T = 20,000$, which implies: (i) for single-loop methods, the algorithm runs for $20,000$ rounds; and (ii) for double-loop methods, the product of inner-loop steps and outer-loop steps equals $20,000$. Additionally, we simplify our method by using fixed values for K, N, and μ to ensure fairness in the comparison. Accordingly, we uniformly select 100 iterations to report. The learning rates for all algorithms are set to 10^{-5}. For PROBE, we set $T = 100$, $K = 200$, $\mu = 0.1$. We set $\Delta_K = 10$, and other PGD parameters are set to 10^{-2} as our default setup. For these baselines, the detailed setups are specified as follows. For MEHA, we examined additional learning rate $\eta \in \{0.1, 2.0\}$ and penalty coefficient $c \in \{0.1, 1, 2, 5, 10\}$. For SUN-DSBO-SE, we examined proximal parameter $\gamma \in \{0.01, 0.05, 0.1\}$ and penalty parameter $\mu_0 \in \{0.01, 0.1, 1, 10\}$. For PBGD, we examined the penalty coefficient in $\{0.1, 1, 10\}$. For GALET, we also tested larger inner-loop step counts $(10, 20)$, but these configurations resulted in numerical errors; we therefore used the best stable configuration with both inner-loop counts set to 5. The reported baseline performances correspond to the best-performing configurations among those tested rather than unfavorable or arbitrarily selected settings. Each experiment is repeated 5 times, and the corresponding standard error bars are shown in all figures. We also apply an exponential moving average (EMA) to enhance visibility.

The lower-level performances of all methods are shown in Figure 6. As noted earlier, GALET finds some seemingly favorable solutions for the lower-level problem (Its loss values also decrease from around 8, and the seemingly immediate convergence observed in the figure is due to the use of EMA). Nevertheless, when considered alongside Figure 2, it is clear that these solutions are actually suboptimal in general. In stark contrast, MEHA, SUN-DSBO-SE, and our PROBE not only converge during training, but also performs consistently well in the validation process. Among them, our PROBE is the most efficient method, achieving convergence in the shortest running time.

B Proof of Main Result

Theorem 2 (Formally restated Theorem 1)**.** *By selecting parameters as follows:*

$$\sigma_t = \Theta\left((t+1)^{-p}\right), \quad \epsilon_t = \frac{\sigma_t^4}{\rho}, \quad \mu_t = \sqrt{\rho\epsilon_t} + \sigma_t, \quad \alpha_t = \Theta\left((t+1)^{-q}\right), \quad \beta_t = \Theta\left(\frac{1}{\ell}\right),$$

$$N_t = \widetilde{\Theta}((t+1)^{\frac{p}{2}}), \quad K_t = \widetilde{\Theta}((t+1)^{8p}),$$

$$\delta' \in (0, \frac{1}{T}), \quad \Delta_{g,t} \geq g(x_t, y_t) - \inf_y g(x_t, y), \quad \chi_t = 3\max\left\{\log\left(\frac{q\ell\Delta_{g,t}}{\epsilon_t^2\delta'}\right), 4\right\},$$

$$r_t = \Theta\left(\frac{\epsilon_t}{\ell\chi_t^2}\right), \quad \varrho_t = \Theta\left(\frac{\epsilon_t}{\chi_t^2}\right), \quad \Delta_{G,t} = \Theta\left(\frac{\epsilon_t^{1.5}}{\chi_t^3\rho^{0.5}}\right), \quad \Delta_{K,t} = \Theta\left(\frac{\ell\chi_t}{\rho^{0.5}\epsilon_t^{0.5}}\right),$$

for any $\delta = T\delta' \in (0,1)$, $p \in (0, \frac{1}{3})$, $3p \le q < 1$, the sequence $\{x_t, y_t\}_t$ generated by Algorithm 1 satisfies the following convergence guarantee w.p. $1-\delta$:

$$\min_{\frac{T}{2} \le t \le T} \Pi(x_t, y_t) = \mathcal{O}\left(\frac{1}{T^{1-q}} + \frac{1}{T^{2p}} + \frac{1}{T^{\frac{4}{5}}}\right).$$

When selecting $p = \frac{1}{5}$ and $q = \frac{3}{5}$, we have $\min_{\frac{T}{2} \le t \le T} \Pi(x_t, y_t) = \mathcal{O}\left(T^{-\frac{2}{5}}\right)$.

Proof. We prove this result with several sub-steps.

Part $\mathcal{A}$. We first control $\|v^*_{\mu_{t+1}}(x_t) - v^*_{\mu_t}(x_t)\|$. To begin with, we define:

$$v^*_\mu(x_t) := -\left[\nabla^2_{yy} g(x_t, y^*_\mu(x_t)) + \mu I_q\right]^{-1} \nabla_y f(x_t, y^*_\mu(x_t)),$$

For any fixed $t \ge 0$, let $H_\mu = \nabla^2_{yy} g(x_t, y^*_\mu(x_t)) + \mu I_q$ and $w_\mu = \nabla_y f(x_t, y^*_\mu(x_t))$. Then, we have:

$$\begin{aligned}
\|v^*_{\mu_{t+1}}(x_t) - v^*_{\mu_t}(x_t)\| &= \|H^{-1}_{\mu_{t+1}} w_{\mu_{t+1}} - H^{-1}_{\mu_t} w_{\mu_t}\| \\
&\le \|H^{-1}_{\mu_{t+1}}(w_{\mu_{t+1}} - w_{\mu_t})\| + \|(H^{-1}_{\mu_{t+1}} - H^{-1}_{\mu_t}) w_{\mu_t}\| \\
&\overset{\flat}{\le} \frac{\ell}{\sigma_{t+1}} \|y^*_{\mu_{t+1}}(x_t) - y^*_{\mu_t}(x_t)\| + \|H^{-1}_{\mu_{t+1}}\| \cdot \|H_{\mu_t} - H_{\mu_{t+1}}\| \cdot \|H^{-1}_{\mu_t}\| \cdot \nu,
\end{aligned} \tag{3}$$

where $\flat$ is due to Assumption 1 and $A^{-1} - B^{-1} = A^{-1}(B-A)B^{-1}$. For the second term, we have:

$$\|H_{\mu_t} - H_{\mu_{t+1}}\| \le \rho \|y^*_{\mu_t}(x_t) - y^*_{\mu_{t+1}}(x_t)\| + |\mu_t - \mu_{t+1}|.$$

Thus, to tackle Equation (3), we need to control the gap $\|y^*_{\mu_t}(x_t) - y^*_{\mu_{t+1}}(x_t)\|$. To this end, we first define the following supporting function for any fixed t:

$$h_\mu(y) = g(x_t, y) + \frac{\mu}{2}\|y - y_{t+1}\|^2.$$

Since our Alg. 2 ensures that w.p. $1-\delta'$, y_{t+1} is an ϵ_t-SOSP, we know that $\|\nabla_y g(x_t, y_{t+1})\| \le \epsilon_t$ and $\lambda_{\min}(\nabla^2_{yy} g(x_t, y_{t+1})) \ge -\sqrt{\rho\epsilon_t}$. According to our parameter selection, we can get $\mu_t = \Theta(\sigma_t)$ and $\sqrt{\rho\epsilon_t} = \sigma_t^2$. Accordingly, we have:

$$\begin{aligned}
\lambda_{\min}(\nabla^2 h_{\mu_t}(y_{t+1})) &\ge \mu_t - \sqrt{\rho\epsilon_t} =: \iota_t, \\
\lambda_{\min}(\nabla^2 h_{\mu_{t+1}}(y_{t+1})) &\ge \mu_{t+1} - \sqrt{\rho\epsilon_t} := \tilde{\iota}_{t+1}.
\end{aligned}$$

Note that $\sigma_t = \Theta((t+1)^{-p})$ with $p \in (0, \frac{1}{3})$, we can easily get $\iota_t = \Theta(\sigma_t)$ and $\tilde{\iota}_{t+1} = \Theta(\sigma_t)$. Therefore, within a sufficiently small ball $\mathcal{B}_{y_{t+1}}(\frac{\iota_t}{2\rho})$, we have $\lambda_{\min}(\nabla^2 h_{\mu_t}(y)) \ge \frac{\iota_t}{2}$. Thus, noting that $\nabla h_{\mu_t}(y^*_{\mu_t}(x_t)) = 0$, we can obtain:

$$\begin{aligned}
&\|y^*_{\mu_t}(x_t) - y_{t+1}\| \le \frac{2}{\iota_t}\|\nabla h_{\mu_t}(y_{t+1})\| = \frac{2}{\iota_t}\|\nabla_y g(x_t, y_{t+1})\| \le \frac{2\epsilon_t}{\iota_t} = \frac{2\sigma_t^4/\rho}{\iota_t}, \\
\Longrightarrow\ &\|y^*_{\mu_t}(x_t) - y_{t+1}\| = \mathcal{O}(\sigma_t^3).
\end{aligned}$$

Similarly, within a small ball centered at y_{t+1}, $h_{\mu_{t+1}}$ remains $\frac{\tilde{\iota}_{t+1}}{2}$-strong convex, which implies:

$$\begin{aligned}
\frac{\tilde{\iota}_{t+1}}{2}\|y^*_{\mu_{t+1}}(x_t) - y^*_{\mu_t}(x_t)\|^2 &\le \langle \nabla h_{\mu_{t+1}}(y^*_{\mu_t}(x_t)) - \nabla h_{\mu_{t+1}}(y^*_{\mu_{t+1}}(x_t)), y^*_{\mu_t}(x_t) - y^*_{\mu_{t+1}}(x_t)\rangle \\
&\overset{\dagger}{=} \langle \nabla_y g(x_t, y^*_{\mu_t}(x_t)) + \mu_{t+1}(y^*_{\mu_t}(x_t) - y_{t+1}), y^*_{\mu_t}(x_t) - y^*_{\mu_{t+1}}(x_t)\rangle \\
&\overset{\ddagger}{=} \langle (\mu_{t+1} - \mu_t)(y^*_{\mu_t}(x_t) - y_{t+1}), y^*_{\mu_t}(x_t) - y^*_{\mu_{t+1}}(x_t)\rangle \\
&\overset{\flat}{\le} |\mu_{t+1} - \mu_t| \cdot \|y^*_{\mu_t}(x_t) - y_{t+1}\| \cdot \|y^*_{\mu_{t+1}}(x_t) - y^*_{\mu_t}(x_t)\|,
\end{aligned}$$

where † is because $y^*_{\mu_{t+1}}(x_t)$ minimizes $h_{\mu_{t+1}}$, ‡ is due to $\nabla_y g(x_t, y^*_{\mu_t}(x_t)) + \mu_t(y^*_{\mu_t}(x_t) - y_{t+1}) = 0$, and ♭ is induced by the Cauchy-Schwarz inequality. Thus, we have:

$$\|y^*_{\mu_{t+1}}(x_t) - y^*_{\mu_t}(x_t)\| \leq \frac{2|\mu_{t+1} - \mu_t|}{\tilde{\iota}_{t+1}} \|y^*_{\mu_t}(x_t) - y_{t+1}\|.$$

Substituting the bound $\|y^*_{\mu_t}(x_t) - y_{t+1}\| = \mathcal{O}(\sigma_t^3)$ and applying the rigorous lower bound $\tilde{\iota}_{t+1} = \Theta(\sigma_t)$, we can get:

$$\|y^*_{\mu_{t+1}}(x_t) - y^*_{\mu_t}(x_t)\| = \mathcal{O}(\sigma_t^2)|\mu_t - \mu_{t+1}|.$$

We then substitute the above results into Equation (3) to get:

$$\|v^*_{\mu_{t+1}}(x_t) - v^*_{\mu_t}(x_t)\| \leq \frac{\ell}{\sigma_{t+1}} \|y^*_{\mu_{t+1}}(x_t) - y^*_{\mu_t}(x_t)\| + \|H^{-1}_{\mu_{t+1}}\| \cdot \|H_t - H_{t+1}\| \cdot \|H^{-1}_{\mu_t}\| \cdot \nu$$

$$\implies \|v^*_{\mu_{t+1}}(x_t) - v^*_{\mu_t}(x_t)\| = \left(\frac{\ell}{\sigma_{t+1}} + \frac{\nu\rho}{\sigma_t\sigma_{t+1}}\right)\mathcal{O}(\sigma_t^2)|\mu_t - \mu_{t+1}| + \frac{\nu}{\sigma_t\sigma_{t+1}}|\mu_t - \mu_{t+1}|.$$

This indicates:

$$\begin{aligned}\|v^*_{\mu_{t+1}}(x_t) - v^*_{\mu_t}(x_t)\| &= \mathcal{O}\left(\frac{1}{\sigma_t\sigma_{t+1}}|\mu_t - \mu_{t+1}|\right) = \mathcal{O}\left(\frac{(t+1)^{-p-1}}{(t+1)^{-p}(t+2)^{-p}}\right) \\ &= \mathcal{O}\left(\frac{(t+1)^{-p-1}}{(t+1)^{-2p}2^{-p}}\right) = \mathcal{O}\big((t+1)^{p-1}\big).\end{aligned}$$

Part $\mathcal{B}$. We then consider the error introduced by the Conjugate Gradient step. Recall that $v^*_{\mu_t}(x_t) = [\nabla^2_{yy} g(x_t, y^*_{\mu_t}(x_t)) + \mu_t I_q]^{-1} \nabla_y f(x_t, y^*_{\mu_t}(x_t))$, and we now define $E_t := \|v_t - v^*_{\mu_{t-1}}(x_{t-1})\|^2$. Note that Conjugate Gradient actually solves a linear system $H_t v = b_t$, where $H_t = \nabla^2_{yy} g(x_t, y_{t+1}) + \mu_t I_q$ and $b_t = \nabla_y f(x_t, y_{t+1})$ at each step t. We correspondingly denote the exact solution to this linear system as $v_t^\circ = H_t^{-1} b_t$. According to Grazzi et al. (2020), by selecting $N_t = \widetilde{\Theta}(\sqrt{(t+1)^p})$, we can obtain:

$$\|v_{t+1} - v_t^\circ\| \leq 2(t+1)^{\frac{p}{2}} \left(\frac{(t+1)^{\frac{p}{2}} - 1}{(t+1)^{\frac{p}{2}} + 1}\right)^{N_t} \|v_t - v_t^\circ\|, \quad \implies \quad \|v_{t+1} - v_t^\circ\| \leq \frac{1}{\sqrt{2}} \|v_t - v_t^\circ\|.$$

Therefore, we have:

$$\sqrt{E_{t+1}} = \|v_{t+1} - v^*_{\mu_t}(x_t)\| \leq \|v_{t+1} - v_t^\circ\| + \|v_t^\circ - v^*_{\mu_t}(x_t)\|.$$

Thus, we need to bound $\|v_t - v_t^\circ\|$, which can be controlled by:

$$\|v_t - v_t^\circ\| \leq \sqrt{E_t} + \underbrace{\|v^*_{\mu_{t-1}}(x_{t-1}) - v^*_{\mu_{t-1}}(x_t)\|}_{①} + \underbrace{\|v^*_{\mu_{t-1}}(x_t) - v^*_{\mu_t}(x_t)\|}_{②} + \underbrace{\|v^*_{\mu_t}(x_t) - v_t^\circ\|}_{③}.$$

We now control each of these terms. For ①, we first note that:

$$\begin{aligned}&\|v^*_\mu(x) - v^*_\mu(x')\| \\ &\leq \left\| \left[\nabla^2_{yy} g(x, y^*_\mu(x)) + \mu I_q\right]^{-1} \left(\nabla_y f(x, y^*_\mu(x)) - \nabla_y f(x', y^*_\mu(x'))\right) \right\| \\ &\quad + \left\| \left(\left[\nabla^2_{yy} g(x, y^*_\mu(x)) + \mu I_q\right]^{-1} - \left[\nabla^2_{yy} g(x', y^*_\mu(x')) + \mu I_q\right]^{-1} \right) \nabla_y f(x', y^*_\mu(x')) \right\| \\ &\leq \left(\frac{\ell}{\sigma} + \frac{\nu\rho}{\sigma^2}\right)\left(1 + \frac{\ell}{\sigma}\right)\|x - x'\| = \Theta(\sigma^{-3})\|x - x'\|,\end{aligned}$$

meaning that its Lipschitz continuity coefficient is in the order of $\Theta(\sigma^{-3})$. Therefore, we have:

$$① \leq \Theta(\sigma_t^{-3})\alpha_{t-1}\|\widehat{\nabla}\Phi_{\mu_{t-1}}(x_{t-1})\| \quad \implies \quad ① = \mathcal{O}\big((t+1)^{3p-q}\big)\|\widehat{\nabla}\Phi_{\mu_{t-1}}(x_{t-1})\|.$$

For ②, we can follow Part $\mathcal{A}$ to get ② $= \mathcal{O}((t+1)^{p-1})$.

For ③, we have:

$$\begin{aligned}
③ &\le \left\| \left[\nabla^2_{yy} g(x_t, y^*_{\mu_t}(x_t)) + \mu_t I_q\right]^{-1} \left(\nabla_y f(x_t, y^*_{\mu_t}(x_t)) - \nabla_y f(x_t, y_{t+1})\right) \right\| \\
&\quad + \left\| \left(\left[\nabla^2_{yy} g(x_t, y^*_{\mu_t}(x_t)) + \mu_t I_q\right]^{-1} - H_t^{-1} \right) \nabla_y f(x_t, y_{t+1}) \right\|, \\
\implies ③ &= \frac{\ell}{\sigma_t} \mathcal{O}(\sigma_t^3) + \frac{\nu\rho}{\sigma_t^2} \mathcal{O}(\sigma_t^3) = \mathcal{O}\big((t+1)^{-p}\big).
\end{aligned}$$

Substituting them back, we can get:

$$\begin{aligned}
\|v_t - v_t^\circ\| &= \sqrt{E_t} + \mathcal{O}\big((t+1)^{3p-q}\big) \|\widehat{\nabla} \Phi_{\mu_{t-1}}(x_{t-1})\| + \mathcal{O}\big((t+1)^{-p}\big), \\
\implies \sqrt{E_{t+1}} &= \frac{1}{\sqrt{2}} \sqrt{E_t} + \mathcal{O}\big((t+1)^{3p-q}\big) \|\widehat{\nabla} \Phi_{\mu_{t-1}}(x_{t-1})\| + \mathcal{O}\big((t+1)^{-p}\big).
\end{aligned}$$

According to the Young's inequality with constant $1/2$, we have:

$$\begin{aligned}
E_{t+1} &= \left(1 + \frac{1}{2}\right) \left(\frac{1}{\sqrt{2}} \sqrt{E_t}\right)^2 + (1+2) \left(\mathcal{O}\big((t+1)^{3p-q}\big) \|\widehat{\nabla} \Phi_{\mu_{t-1}}(x_{t-1})\| + \mathcal{O}\big((t+1)^{-p}\big)\right)^2 \\
&= \frac{3}{4} E_t + \mathcal{O}\big((t+1)^{6p-2q}\big) \|\widehat{\nabla} \Phi_{\mu_{t-1}}(x_{t-1})\|^2 + \mathcal{O}\big((t+1)^{-2p}\big).
\end{aligned}$$

Part $\mathcal{C}$. In this part, we control the shift $\Phi_{\mu_{t+1}}(x_{t+1}) - \Phi_{\mu_t}(x_t)$. We first compute the smoothness coefficient of Φ_μ as follows. According to Assumption 1, for any x, x' we have:

$$\begin{aligned}
&\|\nabla \Phi_\mu(x) - \nabla \Phi_\mu(x')\| \\
\le &\|\nabla_x f(x, y^*_\mu(x)) - \nabla_x f(x', y^*_\mu(x'))\| \\
&+ \left\| \left(\nabla^2_{xy} g(x, y^*_\mu(x)) - \nabla^2_{xy} g(x', y^*_\mu(x'))\right) v^*_\mu(x) \right\| + \left\| \nabla^2_{xy} g(x', y^*_\mu(x')) \left(v^*_\mu(x) - v^*_\mu(x')\right) \right\| \\
\le &\left(\ell + \|v^*_\mu(x)\| \cdot \rho\right) \left(\|x - x'\| + \|y^*_\mu(x) - y^*_\mu(x')\|\right) + \ell \|v^*_\mu(x) - v^*_\mu(x')\| \\
\le &\left(\ell + \frac{\nu}{\sigma} \cdot \rho\right) \left(\|x - x'\| + \|y^*_\mu(x) - y^*_\mu(x')\|\right) \\
&+ \ell \left\| \left[\nabla^2_{yy} g(x, y^*_\mu(x)) + \mu I_q\right]^{-1} \left(\nabla_y f(x, y^*_\mu(x)) - \nabla_y f(x', y^*_\mu(x'))\right) \right\| \\
&+ \ell \left\| \left(\left[\nabla^2_{yy} g(x, y^*_\mu(x)) + \mu I_q\right]^{-1} - \left[\nabla^2_{yy} g(x', y^*_\mu(x')) + \mu I_q\right]^{-1} \right) \nabla_y f(x', y^*_\mu(x')) \right\| \\
\le &\left(\ell + \frac{\nu\rho}{\sigma} + \frac{\ell^2}{\sigma} + \frac{\nu\ell\rho}{\sigma^2}\right) \left(\|x - x'\| + \frac{\ell}{\sigma} \|x - x'\|\right) \\
\le &\left(\ell + \frac{\nu\rho}{\sigma} + \frac{\ell^2}{\sigma} + \frac{\nu\ell\rho}{\sigma^2}\right) \left(1 + \frac{\ell}{\sigma}\right) \|x - x'\| \triangleq L_\mu \|x - x'\|.
\end{aligned}$$

We denote $L_t := L_{\mu_t}$. Given the L_{t+1}-smoothness of $\Phi_{\mu_{t+1}}$ with $L_{t+1} = \Theta((t+1)^{3p})$, and applying the update rule $x_{t+1} = x_t - \alpha_t \widehat{\nabla} \Phi_{\mu_t}(x_t)$ with $q \ge 3p$, the descent lemma yields:

$$\begin{aligned}
&\Phi_{\mu_{t+1}}(x_{t+1}) - \Phi_{\mu_{t+1}}(x_t) \\
\le &\langle \nabla \Phi_{\mu_{t+1}}(x_t), x_{t+1} - x_t \rangle + \frac{L_{t+1}}{2} \|x_{t+1} - x_t\|^2 \\
= &- \alpha_t \langle \nabla \Phi_{\mu_{t+1}}(x_t), \widehat{\nabla} \Phi_{\mu_t}(x_t) \rangle + \frac{L_{t+1} \alpha_t^2}{2} \|\widehat{\nabla} \Phi_{\mu_t}(x_t)\|^2 \\
\le &- \frac{\alpha_t}{2} \|\nabla \Phi_{\mu_t}(x_t)\|^2 - \frac{\alpha_t}{4} \|\widehat{\nabla} \Phi_{\mu_t}(x_t)\|^2 + \frac{\alpha_t}{2} \|\nabla \Phi_{\mu_{t+1}}(x_t) - \widehat{\nabla} \Phi_{\mu_t}(x_t)\|^2,
\end{aligned}$$

where in the last inequality, we use the fact that $\|a - b\|^2 = \|a\|^2 + \|b\|^2 - 2\langle a, b \rangle$. For the last term, we can control it by:

$$\begin{aligned}
\|\nabla \Phi_{\mu_{t+1}}(x_t) - \widehat{\nabla} \Phi_{\mu_t}(x_t)\| \le &\underbrace{\|\nabla_x f(x_t, y^*_{\mu_{t+1}}(x_t)) - \nabla_x f(x_t, y_{t+1})\|}_{\text{Term A}} \\
&+ \underbrace{\|\nabla^2_{xy} g(x_t, y^*_{\mu_{t+1}}(x_t)) v^*_{\mu_{t+1}}(x_t) - \nabla^2_{xy} g(x_t, y_{t+1}) v_{t+1}\|}_{\text{Term B}}.
\end{aligned}$$

Then, we have:

$$\text{Term A} \le \ell\|y^*_{\mu_{t+1}}(x_t) - y_{t+1}\| = \mathcal{O}\big(\|y^*_{\mu_{t+1}}(x_t) - y^*_{\mu_t}(x_t)\| + \|y^*_{\mu_t}(x_t) - y_{t+1}\|\big),$$
$$\implies \text{Term A} = \mathcal{O}(t^{-3p-1}) + \mathcal{O}(t^{-3p}) = \mathcal{O}(t^{-3p}).$$

Besides, we have:

$$\text{Term B} \le \rho\|y^*_{\mu_{t+1}}(x_t) - y_{t+1}\| \cdot \|v^*_{\mu_{t+1}}(x_t)\| + \ell(\|v^*_{\mu_{t+1}}(x_t) - v^*_{\mu_t}(x_t)\| + \sqrt{E_{t+1}}),$$
$$\implies \text{Term B} = \mathcal{O}(t^{-3p} \cdot t^{p}) + \mathcal{O}(t^{p-1}) + \mathcal{O}(\sqrt{E_{t+1}}) = \mathcal{O}(t^{-2p} + t^{p-1} + \sqrt{E_{t+1}}).$$

Thus, we combine them to get:

$$\|\nabla\Phi_{\mu_{t+1}}(x_t) - \widehat{\nabla}\Phi_{\mu_t}(x_t)\|^2 = \mathcal{O}(t^{-4p} + t^{2p-2} + E_{t+1}).$$

Also, we consider the the gap $|\Phi_{\mu_{t+1}}(x_t) - \Phi_{\mu_t}(x_t)|$. According to Assumption 1, we have:

$$|\Phi_{\mu_{t+1}}(x_t) - \Phi_{\mu_t}(x_t)| = |f(x_t, y^*_{\mu_{t+1}}(x_t)) - f(x_t, y^*_{\mu_t}(x_t))| \le \nu\|y^*_{\mu_{t+1}}(x_t) - y^*_{\mu_t}(x_t)\|,$$
$$\implies |\Phi_{\mu_{t+1}}(x_t) - \Phi_{\mu_t}(x_t)| \overset{\flat}{=} \mathcal{O}(\sigma_t^2)|\mu_t - \mu_{t+1}| = \mathcal{O}\big((t+1)^{-3p-1}\big),$$

where $\flat$ can be implied by Part $\mathcal{A}$.

We can introduce positive constants C_1, C_1' to construct the following result:

$$\begin{aligned}
&\Phi_{\mu_{t+1}}(x_{t+1}) - \Phi_{\mu_t}(x_t)\\
=&[\Phi_{\mu_{t+1}}(x_{t+1}) - \Phi_{\mu_{t+1}}(x_t)] + [\Phi_{\mu_{t+1}}(x_t) - \Phi_{\mu_t}(x_t)]\\
\le& -\frac{\alpha_t}{2}\|\nabla\Phi_{\mu_t}(x_t)\|^2 - \frac{\alpha_t}{4}\|\widehat{\nabla}\Phi_{\mu_t}(x_t)\|^2 + C_1\alpha_t E_{t+1} + C_1'\big(\alpha_t t^{-4p} + \alpha_t t^{2p-2} + t^{-3p-1}\big).
\end{aligned}$$

Part $\mathcal{D}$. Recall Part $\mathcal{B}$, with positive constants C_2, C_2', we have:

$$E_{t+1} \le \frac{3}{4}E_t + C_2 t^{6p-2q}\|\widehat{\nabla}\Phi_{\mu_{t-1}}(x_{t-1})\|^2 + C_2' t^{-2p}.$$

We construct the following Lyapunov function:

$$V_t = \Phi_{\mu_t}(x_t) + M\alpha_t E_t,$$

where $M = 3C_1$. Then, we have:

$$\begin{aligned}
V_{t+1} - V_t &= \Phi_{\mu_{t+1}}(x_{t+1}) - \Phi_{\mu_t}(x_t) + M(\alpha_{t+1}E_{t+1} - \alpha_t E_t)\\
&\le -\frac{\alpha_t}{2}\|\nabla\Phi_{\mu_t}(x_t)\|^2 - \frac{\alpha_t}{4}\|\widehat{\nabla}\Phi_{\mu_t}(x_t)\|^2 + C_1'\big(\alpha_t t^{-4p} + \alpha_t t^{2p-2} + t^{-3p-1}\big)\\
&\quad + (C_1\alpha_t + M\alpha_{t+1})E_{t+1} - M\alpha_t E_t\\
&\le -\frac{\alpha_t}{2}\|\nabla\Phi_{\mu_t}(x_t)\|^2 - \frac{\alpha_t}{4}\|\widehat{\nabla}\Phi_{\mu_t}(x_t)\|^2 + C_1'\big(\alpha_t t^{-4p} + \alpha_t t^{2p-2} + t^{-3p-1}\big)\\
&\quad + (C_1\alpha_t + M\alpha_{t+1})\left(\frac{3}{4}E_t + C_2 t^{6p-2q}\|\widehat{\nabla}\Phi_{\mu_{t-1}}(x_{t-1})\|^2 + C_2' t^{-2p}\right) - M\alpha_t E_t\\
&\overset{\flat}{\le} -\frac{\alpha_t}{2}\|\nabla\Phi_{\mu_t}(x_t)\|^2 - \frac{\alpha_t}{4}\|\widehat{\nabla}\Phi_{\mu_t}(x_t)\|^2 + \underbrace{4C_1C_2\alpha_t t^{6p-2q}}_{\eta_t}\|\widehat{\nabla}\Phi_{\mu_{t-1}}(x_{t-1})\|^2\\
&\quad + \underbrace{C_1''\big(t^{-4p-q} + t^{2p-2-q} + t^{-2p-q} + t^{-3p-1}\big)}_{R_t},
\end{aligned}$$

where $\flat$ is due to $M = 3C_1$ and $\alpha_t \ge \alpha_{t+1}$, and C_1'' is another constant. When $q \ge 3p$, we can select proper α_t to ensure $\eta_{t+1} \le \frac{1}{8}\alpha_t$, and denote $T_0 = \lceil \frac{T}{2} \rceil$. Then, we can get:

$$\begin{aligned}
\sum_{t=T_0}^{T}\frac{\alpha_t}{2}\|\nabla\Phi_{\mu_t}(x_t)\|^2 &\le \sum_{t=T_0}^{T}\frac{\alpha_t}{2}\|\nabla\Phi_{\mu_t}(x_t)\|^2 - \sum_{t=T_0}^{T}\frac{\alpha_t}{8}\|\widehat{\nabla}\Phi_{\mu_t}(x_t)\|^2 + \eta_1\|\widehat{\nabla}\Phi_{\mu_{T_0}}(x_0)\|^2\\
&\le V_{T_0} - V_{T+1} + \eta_1\|\widehat{\nabla}\Phi_{\mu_{T_0}}(x_0)\|^2 + \sum_{t=T_0}^{T}R_t.
\end{aligned}$$

Noting that V_{T+1} is bounded below by $\inf \Phi$, the constant terms can be bounded by $\mathcal{O}(1)$. For the residual sum $\sum_{t=T_0}^{T} R_t$, the dominant term is $\mathcal{O}(t^{-q-2p})$. Integrating this term yields $\sum_{t=T_0}^{T} R_t = \mathcal{O}(T^{1-q-2p})$. Dividing both sides by $\sum_{t=T_0}^{T} \alpha_t = \Theta(T^{1-q})$, we finally get:

$$\min_{1\le t\le T} \|\nabla \Phi_{\mu_t}(x_t)\|^2 = \frac{\mathcal{O}(1) + \mathcal{O}(T^{1-q-2p})}{\Theta(T^{1-q})} = \mathcal{O}\left(\frac{1}{T^{1-q}} + \frac{1}{T^{2p}}\right).$$

When selecting $p = 1/5$ and $q = 3/5$, and noting $\min_{T_0\le t\le T} \epsilon_t = \mathcal{O}(T^{-\frac{4}{5}})$ the convergence rate is $\min_{T_0\le t\le T} \Pi(x_t, y_t) = \mathcal{O}(T^{-\frac{2}{5}})$. □

Given Theorem 1, we can immediately obtain the following result:

Corollary 1. *Whenever the lower-level local minimum satisfies the local curvature condition $\nabla^2_{yy} g(x, y) \succeq \mu_0 I_q$ for some $\mu_0 > 0$, the bias bound yields $\|\nabla \Phi_\mu(x) - \nabla \Phi_0(x)\| = \mathcal{O}(\mu)$. Combining this bound with the parameter selection in Theorem 1 gives:*

$$\min_{\lceil T/2\rceil \le t\le T} \|\nabla \Phi_0(x_t)\|^2 = \mathcal{O}(T^{-2/5}).$$

C Details of Illustrative Example

To intuitively show the rationale of our SOSP-based relaxed surrogate, we consider the following concrete example with $x, y \in \mathbb{R}$:

$$\min_{x\in\mathbb{R}^p, y\in\mathbb{R}^q} f(x, y) = (x-2)^2 + y^2$$
$$\text{subject to } \; y \in \mathcal{S}(x) := \arg\min_y g(x, y) = \frac{1}{4}y^4 - \frac{1}{2}xy^2.$$

As mentioned earlier, to tackle this nonconvex lower-level BLO problem, existing works instead solve a surrogate lower-level problem by finding a first-order stationary point, i.e., a point where $\nabla_y g(x, y) = 0$, or by solving its equivalent formulations under certain other assumptions (Xiao et al., 2023; Shen & Chen, 2023; Liu et al., 2024; Ma et al., 2026).

Specifically, the required derivatives are:

$$\nabla_y g(x, y) = y^3 - xy = y(y^2 - x), \quad \nabla^2_{yy} g(x, y) = 3y^2 - x.$$

Setting the gradient to 0, we obtain the following FOSP set:

$$\{(x, 0), (x', \sqrt{x'}), (x', -\sqrt{x'}) : \forall x \in \mathbb{R}, x' \ge 0\}.$$

When substituting $\nabla_y g(x, y) = 0$ into the upper-level problem, we obtain the following solutions: $(x^*, y^*) \in \{(2, 0), (1.5, \sqrt{1.5}), (1.5, -\sqrt{1.5})\}$, where $(2, 0)$ is a saddle point of $g(x, \cdot)$, and the other two serve as the true solutions. As shown in Figure 1, these aforementioned methods can be entirely drawn toward the saddle point, failing to solve the original BLO problem. In contrast, our PROBE escapes the saddle point and finds the optimal solution.

D Scope and Implications of SOSP Guarantees

While we established a finite-time convergence rate of $\mathcal{O}(T^{-\frac{2}{5}})$ for PROBE in Section 5, it remains important to clarify the class of solutions to which the method converges. In this section, we discuss our SOSP-based solutions in detail.

The theoretical guarantee is to achieve some SOSP solution. Our theoretical results guarantee that PROBE reaches a *lower-level SOSP* and thereby avoids strict saddle points. However, they do *not* guarantee the selection of the "best" solution among multiple lower-level local minima. Although the objective values of any given collection of local minima can be compared directly, guaranteeing the selection of the best solution among all local minima is equivalent to finding a global minimum of the lower-level problem, which is computationally intractable for general nonconvex problems (NP-hard). Therefore, without additional structural assumptions, the effects of different lower-level local solutions on the upper-level problem cannot be characterized in general.

Trajectory	Candidates	Strict	Non-strict	Undetermined
1	8	8	0	0
2	20	19	0	1
3	7	7	0	0
4	11	10	0	1
5	18	17	0	1
Total	**64**	**61**	**0**	**3**

Table 2: Numerical classification of the perturbation-triggered lower-level stationary-point candidates encountered along five PROBE *trajectories.* ***Undetermined*** *refers to candidates that could not be classified conclusively because of numerical instability.*

An SOSP provides a strictly stronger stationarity guarantee than an FOSP. The lower-level guarantee that PROBE reaches an SOSP does *not* rely on the strict-saddle property. Rather, this property is used only to conclude that an SOSP corresponds to a lower-level local minimum. Even when strictness does not hold, an SOSP still excludes any stationary point that admits a direction of negative curvature and therefore provides a *strictly stronger* stationarity guarantee than an FOSP. This advantage is also well documented in the literature (Dauphin et al., 2014; Allen-Zhu & Li, 2018). This distinction is particularly important in BLO: as shown in Figure 1, FOSP-based methods can become trapped at lower-level saddle points and consequently converge to incorrect upper-level solutions, whereas PROBE escapes these saddle points and finds the correct solution.

Empirical evidence shows the prevalence of strict saddles in practice. We conducted an experiment using the same LLM-based setting as in Section A. Specifically, we identified all the saddle points encountered along each trajectory and computed the minimum eigenvalue of the lower-level Hessian at each of them. As shown in Table 2, across five repeated trials, all 61 encountered saddle points exhibited a negative minimum eigenvalue, providing empirical evidence that the saddles encountered in these runs are strict. This implies that most saddle points are strict in practice, demonstrating the applicability of our theories.