Penalty-Based First-Order Methods for Bilevel Optimization with Minimax and Constrained Lower-Level Problems

Yiyang Shen[*], **Yutian He**[*], **Weiran Wang, Qihang Lin**
University of Iowa
Iowa City, IA
{yiyang-shen,yutian-he,weiran-wang,qihang-lin}@uiowa.edu

Abstract

We study a class of bilevel optimization problems in which both the upper- and lower-level problems have minimax structures. This setting captures a broad range of emerging applications. Despite the extensive literature on bilevel optimization and minimax optimization separately, existing methods mainly focus on bilevel optimization with lower-level minimization problems, often under strong convexity assumptions, and are not directly applicable to the minimax lower-level setting considered here. To address this gap, we develop penalty-based first-order methods for bilevel minimax optimization without requiring strong convexity of the lower-level problem. In the deterministic setting, we establish that the proposed method finds a lifted ϵ-KKT point with $\tilde{O}(\epsilon^{-4})$ oracle complexity. We further show that bilevel problems with convex constrained lower-level minimization can be reformulated as special cases of our framework via Lagrangian duality, leading to an $\tilde{O}(\epsilon^{-4})$ complexity bound that improves upon the existing $\tilde{O}(\epsilon^{-7})$ result when the lower-level Slater's condition holds. Finally, we extend our approach to the stochastic setting, where only stochastic gradient oracles are available, and prove that the proposed stochastic method finds a nearly lifted ϵ-KKT point with $\tilde{O}(\epsilon^{-9})$ oracle complexity.

1 Introduction

In this paper, we consider the following bilevel optimization problem with minimax upper- and lower-level problems:

$$\min_{\mathbf{x}_1,\mathbf{y}_1,\mathbf{y}_2} \left\{ \max_{\mathbf{x}_2} f(\mathbf{x}_1,\mathbf{x}_2,\mathbf{y}_1,\mathbf{y}_2) \Big| (\mathbf{y}_1,\mathbf{y}_2) \in \arg\min_{\mathbf{z}_1} \max_{\mathbf{z}_2} \tilde{f}(\mathbf{x}_1,\mathbf{z}_1,\mathbf{z}_2) \right\}, \tag{1}$$

where $(\mathbf{y}_1,\mathbf{y}_2) \in \arg\min_{\mathbf{z}_1} \max_{\mathbf{z}_2} \tilde{f}(\mathbf{x}_1,\mathbf{z}_1,\mathbf{z}_2)$ means $(\mathbf{y}_1,\mathbf{y}_2)$ is a saddle-point of $\tilde{f}(\mathbf{x}_1,\cdot,\cdot)$, i.e.,

$$\mathbf{y}_1 \in \arg\min_{\mathbf{z}_1} \tilde{f}(\mathbf{x}_1,\mathbf{z}_1,\mathbf{y}_2) \quad \text{and} \quad \mathbf{y}_2 \in \arg\max_{\mathbf{z}_2} \tilde{f}(\mathbf{x}_1,\mathbf{y}_1,\mathbf{z}_2).$$

We assume the saddle-point of $\tilde{f}(\mathbf{x}_1,\cdot,\cdot)$ always exists for any $\mathbf{x}_1$ and assume

$$f(\mathbf{x}_1,\mathbf{x}_2,\mathbf{y}_1,\mathbf{y}_2) = f_1(\mathbf{x}_1,\mathbf{x}_2,\mathbf{y}_1,\mathbf{y}_2) + f_2(\mathbf{x}_1) - f_3(\mathbf{x}_2), \tag{2}$$

$$\tilde{f}(\mathbf{x}_1,\mathbf{y}_1,\mathbf{y}_2) = \tilde{f}_1(\mathbf{x}_1,\mathbf{y}_1,\mathbf{y}_2) + \tilde{f}_2(\mathbf{y}_1) - \tilde{f}_3(\mathbf{y}_2). \tag{3}$$

[*]Equal contribution. Author ordering determined by a coin flip.
Code available at: https://github.com/weiranwang19/minimax

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

where f_2, f_3, $\tilde{f}_2$, and $\tilde{f}_3$ are proper closed convex functions, f_1 is smooth and concave on $\mathbf{x}_2$, and $\tilde{f}_1$ is smooth, convex in $\mathbf{y}_1$ and concave in $\mathbf{y}_2$.

Bilevel optimization has found numerous applications in machine learning, including but not limited to hyperparameter tuning [4, 16], meta-learning [16, 5, 60], reinforcement learning [23, 72, 41], and neural architecture search [45]. On the other hand, minimax optimization problems also arise in a wide range of applications, such as robust learning [13], distributionally robust optimization (DRO, [3]), generative adversarial networks [19], and adversarial training [55]. Incorporating such minimax problems as the lower-level problem in bilevel optimization leads to a broad class of novel and practically relevant applications of (1). We will present the examples of (1) from hyperparameter tuning for distributionally robust deep learning in Section 7 with numerical results. Additional applications of (1) are given in Section A.

Despite the extensive literature on both bilevel optimization and minimax optimization, existing bilevel formulations typically assume that the lower-level problem is a single-objective minimization problem (often strongly convex), with or without constraints. The setting where the lower-level problem itself is a minimax problem has only been studied in a very few papers [79, 78, 73]. Moreover, general first-order oracle complexity guarantees for finding approximate KKT points of bilevel optimization with minimax lower-level problems remain largely unexplored. To fill this gap, this work proposes both deterministic and stochastic first-order algorithms for (1) based on a value-function penalty reformulation. Such reformulation has been mainly studied for bilevel optimization with a lower-level minimization problem [63, 73, 74, 30, 40, 39, 8, 54, 52] and for bilevel optimization with a lower-level minimax problem [73, 79]. We establish the (stochastic) gradient oracle complexity of the proposed methods for finding an ϵ-KKT point of (1) (see Definition 3.2) in the deterministic case and for finding a nearly ϵ-KKT point of (1) (see Definition 6.2) in the stochastic case.

Furthermore, we consider a bilevel optimization problem in which the lower-level problem is a convex minimization problem with convex nonlinear inequality constraints. In particular, we study the following problem

$$\min_{\mathbf{x}_1,\mathbf{y}_1} \left\{ f(\mathbf{x}_1,\mathbf{y}_1) \middle| \mathbf{y}_1 \in \arg\min_{\mathbf{z}_1} \left\{ \bar{f}(\mathbf{x}_1,\mathbf{z}_1) \middle| \bar{\mathbf{g}}(\mathbf{x}_1,\mathbf{z}_1) \leq 0 \right\} \right\}, \tag{4}$$

where $\bar{\mathbf{g}}(\mathbf{x}_1,\mathbf{y}_1) = (\bar{g}_1(\mathbf{x}_1,\mathbf{y}_1),\ldots,\bar{g}_l(\mathbf{x}_1,\mathbf{y}_1)) \in \mathbb{R}^l$ and $\bar{g}_i(\mathbf{x}_1,\mathbf{y}_1)$ is smooth and convex in $\mathbf{y}_1$ and satisfies Slater's condition for $i = 1,\ldots,l$. We also assume

$$f(\mathbf{x}_1,\mathbf{y}_1) = f_1(\mathbf{x}_1,\mathbf{y}_1) + f_2(\mathbf{x}_1), \qquad \bar{f}(\mathbf{x}_1,\mathbf{y}_1) = \bar{f}_1(\mathbf{x}_1,\mathbf{y}_1) + \bar{f}_2(\mathbf{y}_1), \tag{5}$$

where f_2 and $\bar{f}_2$ are proper closed convex functions, f_1 is smooth, and $\bar{f}_1$ is smooth and convex in $\mathbf{y}_1$. Assuming Slater's condition holds, strong duality holds for the lower-level problem and any of its optimal Lagrangian multiplier must fall in a box $[0,B]^l$ for some constant B, we can reformulate (4) as a special case of (1) as follows

$$\min_{\mathbf{x}_1,\mathbf{y}_1,\mathbf{y}_2} \left\{ f(\mathbf{x}_1,\mathbf{y}_1) \middle| (\mathbf{y}_1,\mathbf{y}_2) \in \arg\min_{\mathbf{z}_1} \max_{\mathbf{z}_2} \tilde{f}(\mathbf{x}_1,\mathbf{z}_1,\mathbf{z}_2) \right\}, \tag{6}$$

where there is no variable $\mathbf{x}_2$ in the upper-level problem and

$$\tilde{f}(\mathbf{x}_1,\mathbf{y}_1,\mathbf{y}_2) = \bar{f}(\mathbf{x}_1,\mathbf{y}_1) + \mathbf{y}_2^\top \bar{\mathbf{g}}(\mathbf{x}_1,\mathbf{y}_1) - \mathbf{1}_{[0,B]^l}(\mathbf{y}_2) \tag{7}$$

with $\mathbf{1}_{[0,B]^l}(\cdot)$ being the indicator function of the set $[0,B]^l$. It follows that $\tilde{f}$ in (7) satisfies (2) with $\tilde{f}_1(\mathbf{x}_1,\mathbf{y}_1,\mathbf{y}_2) = \mathbf{y}_2^\top \bar{\mathbf{g}}(\mathbf{x}_1,\mathbf{y}_1) + \bar{f}_1(\mathbf{x}_1,\mathbf{y}_1)$, $\tilde{f}_2(\mathbf{y}_1) = \bar{f}_2(\mathbf{y}_1)$, and $\tilde{f}_3(\mathbf{y}_2) = \mathbf{1}_{[0,B]^l}(\mathbf{y}_2)$. We are then able to apply the proposed first-order method to (4) and show that our method finds an ϵ-KKT point of (4) (see Definition 5.2) with a lower oracle complexity than Lu and Mei [54, 52] although their methods can be applied to a more general setting, including the case where the strong duality does not hold in the lower level problem.

Summary of contributions.

1. We propose a deterministic first-order method for (1) based on a penalty reformulation, and show that it finds an ϵ-KKT point of (1) with $\tilde{O}(\epsilon^{-4})$ oracle complexity. We further extend the proposed method to the stochastic setting and show that our method finds a nearly ϵ-KKT point of (1) with $\tilde{O}(\epsilon^{-9})$ stochastic oracle complexity. To the best of our knowledge, these are the first complexity results for deterministic and stochastic first-order methods for (1).

2. We apply the proposed method to (4) after reformulating it as (6), which is an instance of (1). This reformulation holds due to strong duality and a uniform bound on the dual variables guaranteed by the assumed Slater's condition. The resulting method finds an ϵ-KKT point of (4) with complexity $\tilde{O}(\epsilon^{-4})$, improving upon the best existing complexity bound of $\tilde{O}(\epsilon^{-7})$ for a merely convex case lower-level problem [54] and $\tilde{O}(\epsilon^{-6})$ for a strongly convex lower-level problem [52]. That said, we also note that the methods in [52, 54] utilize a primal penalty reformulation of the lower-level problem so they do not require strong duality in the lower level and can be applied in a more general setting than ours, e.g., when Slater's condition does not hold.

3. We demonstrate the proposed algorithms on both synthetic and real-world datasets. On (4), our method outperforms existing penalty-based methods in terms of convergence speed. We further consider a bilevel formulation for hyperparameter tuning (28) of a real-world distributionally robust optimization problem [62], which gives rise to an instance of (1) that substantially improves over an existing DRO method in terms of test worst-group accuracy.

2 Related works

Most first-order methods for bilevel optimization focus on the smooth setting in which the lower-level problem is a strongly convex constrained or unconstrained minimization problem. Under this assumption (and constraint qualification in the constrained case), the lower-level solution is a single-valued and differentiable mapping of the upper-level solution, and the gradient of the outer objective can be computed through the chain rule, the implicit function theorem and Danskin's theorem. This leads to the class of hypergradient or implicit-gradient methods [59, 18, 60, 20, 29, 71, 21, 9, 66, 28, 11, 10, 42, 34, 68, 48, 70, 24, 23, 39, 40, 8, 30]. Their assumptions exclude applications in which the lower-level problem is merely convex and has multiple solutions.

Several recent works have replaced the lower-level strong convexity assumption to other assumptions, such as the Polyak-Łojasiewicz condition, or consider an optimization framework without using a hypergradient, for example, the penalty-based, value-function-based, and optimality-condition-based a [46, 49, 47, 2, 50, 7, 65, 44, 40, 69, 26, 27, 25, 31]. Shen and Chen [63], Shen et al. [64] consider a generic penalty method when the lower-level optimal set allows a squared-distance bound function. These methods can be applied to bilevel optimization with merely convex or non-convex lower-level objectives, but not directly to (1) whose lower-level problem has a minimax structure. A more detailed comparison between our work and some penalty-based methods is given in Table 3 in Appendix K.

More recently, Lu and Mei [52] proposed a first-order penalty framework that solves bilevel optimization through a minimax reformulation using the algorithmic framework from Lu and Mei [53]. Their method finds an ϵ-KKT point of (4) with complexity $\tilde{O}(\epsilon^{-4})$ and $\tilde{O}(\epsilon^{-7})$ when the lower-level problem is unconstrained and constrained, respectively. By a similar penalty approach, Lu and Mei [54] developed a sequential minimax optimization method that reduces the complexity to $\tilde{O}(\epsilon^{-6})$ when the lower-level problem is constrained but has a strongly convex objective function. Our work is motivated by Lu and Mei [52, 53, 54]. In fact, our Algorithm 1 in the deterministic case is exactly Algorithm 2 from [53]. However, we study a more general bilevel minimax structure in (1), which covers lower-level constrained optimization as a special case if Slater's condition holds. In that case, we improve the complexity in [52, 54] from $\tilde{O}(\epsilon^{-7})$ and $\tilde{O}(\epsilon^{-6})$ to $\tilde{O}(\epsilon^{-4})$. That said, because the methods in [52, 54] apply a penalty formulation to handle the lower-level constraints instead of using duality gap, they can be applied to a more general setting than ours, e.g., when the lower-level problem does not satisfy Slater's condition or does not even have strong duality.

Ouattara and Aswani [58], Jiang et al. [30], Nie et al. [57] study bilevel problems in which the lower-level problem involves nonlinear constraints, and reformulate the bilevel problem as a single-level problem via Lagrangian duality in a way similar to our approach. Ouattara and Aswani [58] do not provide a complexity analysis, however. Jiang et al. [30] and Nie et al. [57] consider a penalty formulation of the single-level problem, but require strong convexity of the lower-level problem to ensure uniqueness of the lower-level solution; this assumption is not needed in our work. Yao et al. [74] propose a related reformulation based on a proximal Lagrangian value function and, similarly to our approach, impose a box constraint on the dual variable. However, their algorithm requires projection onto the constraint set, which can be computationally inefficient.

Yao et al. [73] study bilevel optimization with convex, nonlinearly constrained lower-level minimization. They introduce a doubly regularized Lagrangian gap function that yields an equivalent single-level formulation with a smooth constraint and develop a penalty-based first-order method. In the same paper, they further extend their gap-function reformulation to bilevel optimization with a convex-concave minimax lower-level problem and propose a corresponding first-order algorithm. However, the convergence analysis of this extension is left for future work.

Zheng et al. [79] study bilevel reinforcement learning with a regularized zero-sum Markov game at the lower level and establish convergence guarantees for a penalty-based first-order policy-gradient method. Although their analysis does not require strong convexity-concavity of the full lower-level objective, it exploits state-wise strongly convex-concave regularization, uniqueness of the lower-level equilibrium policy pair, and a Polyak-Lojasiewicz condition on the penalized objective. In contrast, our framework allows general convex-concave lower-level problems with possibly nonunique saddle points, without requiring these additional structural properties.

Finally, Khanduri et al. [35], Kornowski et al. [37], Tsaknakis et al. [66] study bilevel problems in which the lower-level constraints are linear, whereas we allow general nonlinear convex constraints. Li et al. [43], Allende and Still [1] consider reformulations of (4) based on Wolfe duality and the KKT conditions of the lower-level problem, respectively, which can accommodate general lower-level constraints. Ye et al. [75] and Gao et al. [17] adopt value-function-based reformulations and solve (4) using techniques from difference-of-convex optimization. However, these works do not provide complexity guarantees for finding an ϵ-KKT point. Recent works by Khalafi and Boob [33], Dvurechensky et al. [14] study stochastic bilevel variational inequality problems. Although minimax problems can be formulated as variational inequality problems, their models use the same decision variables at the upper and lower levels, which is substantially different from our setting.

3 Preliminaries and assumptions

A function or mapping ϕ on $\mathbb{R}^n$ is L_ϕ-*Lipschitz* if $\|\phi(\mathbf{x}) - \phi(\mathbf{x}')\| \le L_\phi \|\mathbf{x} - \mathbf{x}'\|$ for any $\mathbf{x}$ and $\mathbf{x}'$ in $\mathbb{R}^n$, and $L_{\nabla\phi}$-*smooth* if it is differentiable and $\|\nabla\phi(\mathbf{x}) - \nabla\phi(\mathbf{x}')\| \le L_{\nabla\phi}\|\mathbf{x} - \mathbf{x}'\|$ for any $\mathbf{x}$ and $\mathbf{x}'$ in $\mathbb{R}^n$. For a proper closed convex function ϕ on $\mathbb{R}^n$, its *proximal mapping* $\mathbf{prox}_\phi(\cdot)$, is defined as

$$\mathbf{prox}_\phi(\mathbf{x}) := \arg\min_{\mathbf{x}' \in \mathbb{R}^n} \left\{ \frac{1}{2}\|\mathbf{x}' - \mathbf{x}\|^2 + \phi(\mathbf{x}') \right\}.$$

For a proper closed function ϕ on $\mathbb{R}^n$, its *subdifferential* is defined as

$$\partial\phi(\mathbf{x}) := \{v \in \mathbb{R}^n : \phi(\mathbf{x}') \ge \phi(\mathbf{x}) + \langle v, \mathbf{x}' - \mathbf{x}\rangle + o(\|\mathbf{x}' - \mathbf{x}\|), \quad \text{as } \mathbf{x}' \to \mathbf{x}\}.$$

Any vector $v \in \partial\phi(\mathbf{x})$ is called a *subgradient* of ϕ at $\mathbf{x}$. If $\mathbf{z}$ is a sub-vector of $\mathbf{x}$, we use $\partial_{\mathbf{z}}\phi(\mathbf{x})$ to denote the (partial) subdifferential of ϕ with respect to $\mathbf{z}$.

We define the primal and dual objective functions of the lower-level problem in (1) as

$$p(\mathbf{x}_1, \mathbf{y}_1) := \max_{\mathbf{z}_2} \tilde{f}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{z}_2) \quad \text{and} \quad d(\mathbf{x}_1, \mathbf{y}_2) := \min_{\mathbf{z}_1} \tilde{f}(\mathbf{x}_1, \mathbf{z}_1, \mathbf{y}_2), \tag{8}$$

respectively, which can be non-smooth. By weak duality, it holds that $p(\mathbf{x}_1, \mathbf{y}_1) \ge d(\mathbf{x}_1, \mathbf{y}_2)$ for any $\mathbf{x}_1$, $\mathbf{y}_1$ and $\mathbf{y}_2$, and $(\mathbf{y}_1, \mathbf{y}_2)$ is the saddle-point of $\tilde{f}(\mathbf{x}_1, \cdot, \cdot)$ if and only if $p(\mathbf{x}_1, \mathbf{y}_1) = d(\mathbf{x}_1, \mathbf{y}_2)$. Therefore, bilevel optimization (1) is essentially a non-convex concave minimax problem with a non-convex non-smooth inequality constraint, i.e.,

$$\min_{\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2} \left\{ \max_{\mathbf{x}_2} f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2) \Big| p(\mathbf{x}_1, \mathbf{y}_1) - d(\mathbf{x}_1, \mathbf{y}_2) \le 0 \right\}. \tag{9}$$

We make the following assumptions throughout the paper.

Assumption 3.1. Problem (1) has the structures in (2) and (3) and the following statements hold.

1. $\mathcal{X}_1 := \mathrm{dom} f_2$, $\mathcal{X}_2 := \mathrm{dom} f_3$, $\mathcal{Y}_1 := \mathrm{dom} \tilde{f}_2$, and $\mathcal{Y}_2 := \mathrm{dom} \tilde{f}_3$ are compact.
2. $f_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2)$ is concave in $\mathbf{x}_2$ and $L_{\nabla f_1}$-smooth on $\mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{Y}_1 \times \mathcal{Y}_2$. $\tilde{f}_1(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2)$ is convex in $\mathbf{y}_1$, concave in $\mathbf{y}_2$, and $L_{\nabla \tilde{f}_1}$-smooth on $\mathcal{X}_1 \times \mathcal{Y}_1 \times \mathcal{Y}_2$.

3. f_2, f_3, $\tilde{f}_2$, and $\tilde{f}_3$ are proper closed convex, and their proximal operators can be computed exactly.

We define the following quantities, which are all finite according to Assumption 3.1.

$$D_1 := \max\{\|(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2) - (\mathbf{x}_1', \mathbf{y}_1', \mathbf{y}_2')\| \mid (\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2), (\mathbf{x}_1', \mathbf{y}_1', \mathbf{y}_2') \in \mathcal{X}_1 \times \mathcal{Y}_1 \times \mathcal{Y}_2\}, \quad (10)$$

$$D_2 := \max\{\|(\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2) - (\mathbf{x}_2', \mathbf{z}_1', \mathbf{z}_2')\| \mid (\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2), (\mathbf{x}_2', \mathbf{z}_1', \mathbf{z}_2') \in \mathcal{X}_2 \times \mathcal{Y}_1 \times \mathcal{Y}_2\}, \quad (11)$$

$$f_{\text{low}} := \min\{f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2) \mid (\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2) \in \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{Y}_1 \times \mathcal{Y}_2\}. \quad (12)$$

We call $(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2)$ a *Karush–Kuhn–Tucker* (KKT) point of (1) if there exists $\rho \geq 0$ such that

$$\begin{aligned} &\mathbf{0} \in \partial_{(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2)} \left[\max_{\mathbf{x}_2} f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2)\right] + \rho \partial_{(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2)} \left[p(\mathbf{x}_1, \mathbf{y}_1) - d(\mathbf{x}_1, \mathbf{y}_2)\right], \\ &p(\mathbf{x}_1, \mathbf{y}_1) - d(\mathbf{x}_1, \mathbf{y}_2) \leq 0. \end{aligned}$$

Given that the maximization problems in $\max_{\mathbf{x}_2} f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2)$ and in the definition of $p(\mathbf{x}_1, \mathbf{y}_1) - d(\mathbf{x}_1, \mathbf{y}_2)$ are concave, by Danskin's theorem [6], $(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2)$ as a KKT point of (1) if there exist $\rho \geq 0$ and $(\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2)$ such that

$$\begin{aligned} &\mathbf{0} \in \partial_{(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2)} \left[f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2) + \rho\big(\tilde{f}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{z}_2) - \tilde{f}(\mathbf{x}_1, \mathbf{z}_1, \mathbf{y}_2)\big)\right], \\ &\mathbf{0} \in \partial_{\mathbf{x}_2} \left[-f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2)\right], \quad \mathbf{0} \in \rho \partial_{\mathbf{z}_1} \tilde{f}(\mathbf{x}_1, \mathbf{z}_1, \mathbf{y}_2), \quad \mathbf{0} \in \rho \partial_{\mathbf{z}_2} \left[-\tilde{f}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{z}_2)\right], \\ &p(\mathbf{x}_1, \mathbf{y}_1) - d(\mathbf{x}_1, \mathbf{y}_2) \leq 0. \end{aligned}$$

In general, a KKT point can only be approached as a limiting solution of an algorithm. Hence, we consider a lifted ϵ-KKT point of (1) which satisfies all conditions of a KKT point up to an error of ϵ.

Definition 3.2. $(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2)$ is a lifted ϵ-KKT point (1) if there exist $\rho \geq 0$ and $(\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2)$ such that

$$\begin{aligned} &\text{dist}\left(\mathbf{0}, \partial_{(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2)} \left[f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2) + \rho\big(\tilde{f}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{z}_2) - \tilde{f}(\mathbf{x}_1, \mathbf{z}_1, \mathbf{y}_2)\big)\right]\right) \leq \epsilon, \\ &\text{dist}\left(\mathbf{0}, \partial_{\mathbf{x}_2} \left[-f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2)\right]\right) \leq \epsilon, \quad \text{dist}\left(\mathbf{0}, \rho \partial_{\mathbf{z}_1} \tilde{f}(\mathbf{x}_1, \mathbf{z}_1, \mathbf{y}_2)\right) \leq \epsilon, \\ &\text{dist}\left(\mathbf{0}, \rho \partial_{\mathbf{z}_2} \left[-\tilde{f}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{z}_2)\right]\right) \leq \epsilon, \quad p(\mathbf{x}_1, \mathbf{y}_1) - d(\mathbf{x}_1, \mathbf{y}_2) \leq \epsilon. \end{aligned}$$

We propose a first-order method for finding a lifted ϵ-KKT point of (1). The efficiency of our method is characterized by its *oracle complexity*, i.e., the total number of gradient evaluations of f and $\tilde{f}$.

4 A deterministic first-order method for (1) based on penalty formulation

Following the existing literature, e.g., [63, 39, 8, 54, 52], we reformulate (1) and (9) into an unconstrained penalty problem using a penalty parameter $\rho \geq 0$:

$$\begin{aligned} f^* &:= \min_{\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2} \left\{\max_{\mathbf{x}_2} f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2) + \rho\left[p(\mathbf{x}_1, \mathbf{y}_1) - d(\mathbf{x}_1, \mathbf{y}_2)\right]\right\} && (13) \\ &= \min_{\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2} \max_{\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2} P_\rho(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2, \mathbf{z}_1, \mathbf{z}_2) && (14) \end{aligned}$$

where

$$\begin{aligned} P_\rho(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2, \mathbf{z}_1, \mathbf{z}_2) &:= f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2) + \rho\big(\tilde{f}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{z}_2) - \tilde{f}(\mathbf{x}_1, \mathbf{z}_1, \mathbf{y}_2)\big) && (15) \\ &= P_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2, \mathbf{z}_1, \mathbf{z}_2) + P_2(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2) - P_3(\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2) && (16) \\ P_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2, \mathbf{z}_1, \mathbf{z}_2) &:= f_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2) + \rho\big(\tilde{f}_1(\mathbf{x}_1, \mathbf{y}_1, \mathbf{z}_2) - \tilde{f}_1(\mathbf{x}_1, \mathbf{z}_1, \mathbf{y}_2)\big), && (17) \\ P_2(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2) &:= f_2(\mathbf{x}_1) + \rho\tilde{f}_2(\mathbf{y}_1) + \rho\tilde{f}_3(\mathbf{y}_2) && (18) \\ P_3(\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2) &:= f_3(\mathbf{x}_2) + \rho\tilde{f}_2(\mathbf{z}_1) + \rho\tilde{f}_3(\mathbf{z}_2). && (19) \end{aligned}$$

The strategy we adopt is to first find an ϵ-primal-dual stationary point of (14) defined below using a primal-dual method and show that such a point is also a lifted $O(\epsilon)$-KKT point of (1).

Definition 4.1. $(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2, \mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2)$ is an ϵ-primal-dual stationary point of (13) if

$$\text{dist}\left(\mathbf{0}, \partial_{(\mathbf{x}_1,\mathbf{y}_1,\mathbf{y}_2)}\left[f(\mathbf{x}_1,\mathbf{x}_2,\mathbf{y}_1,\mathbf{y}_2) + \rho\big(\tilde{f}(\mathbf{x}_1,\mathbf{y}_1,\mathbf{z}_2) - \tilde{f}(\mathbf{x}_1,\mathbf{z}_1,\mathbf{y}_2)\big)\right]\right) \le \epsilon,$$
$$\text{dist}\left(\mathbf{0}, \partial_{(\mathbf{x}_2,\mathbf{z}_1,\mathbf{z}_2)}\left[-f(\mathbf{x}_1,\mathbf{x}_2,\mathbf{y}_1,\mathbf{y}_2) + \rho\big(-\tilde{f}(\mathbf{x}_1,\mathbf{y}_1,\mathbf{z}_2) + \tilde{f}(\mathbf{x}_1,\mathbf{z}_1,\mathbf{y}_2)\big)\right]\right) \le \epsilon.$$

The lemma below characterizes the relationship between Definition 3.2 and Definition 4.1. The proof can be found in Appendix B.

Lemma 4.2. *Suppose* $(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2, \mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2)$ *is an* ϵ*-primal-dual stationary point of* (13) *and*

$$\rho^{-1}\left(\max_{\mathbf{x}_2',\mathbf{z}_1',\mathbf{z}_2'} P_\rho(\mathbf{x}_1,\mathbf{x}_2',\mathbf{y}_1,\mathbf{y}_2,\mathbf{z}_1',\mathbf{z}_2') - f_{low}\right) \le O(\epsilon), \tag{20}$$

where P_ρ *and* f_{low} *are defined in* (15) *and* (12)*. Then* $(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2)$ *is a lifted* $O(\epsilon)$*-KKT point of* (1)*.*

Based on Lemma 4.2, our approach will find a lifted $O(\epsilon)$-KKT point of (1) by computing an ϵ-primal-dual stationary point $(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2, \mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2)$ of (14) and, simultaneously, ensure it satisfies (20) by showing that $\max_{\mathbf{x}_2',\mathbf{z}_1',\mathbf{z}_2'} P_\rho(\mathbf{x}_1,\mathbf{x}_2',\mathbf{y}_1,\mathbf{y}_2,\mathbf{z}_1',\mathbf{z}_2') = O(1)$ and setting $\rho = O(\epsilon^{-1})$.

Under Assumption 3.1, (14) is a non-convex-concave (NCC) minimax problem. We directly apply the *inexact proximal point method* by Lu and Mei [53, Algorithm 2] to find its ϵ-primal-dual stationary point. In this method, the solutions are updated by solving a sequence of minimax subproblems:

$$\Big((\mathbf{x}_1^{(k+1)}, \mathbf{y}_1^{(k+1)}, \mathbf{y}_2^{(k+1)}), (\mathbf{x}_2^{(k+1)}, \mathbf{z}_1^{(k+1)}, \mathbf{z}_2^{(k+1)})\Big) \approx \arg\min_{\mathbf{x}_1,\mathbf{y}_1,\mathbf{y}_2}\max_{\mathbf{x}_2,\mathbf{z}_1,\mathbf{z}_2}\Big\{P_\rho(\mathbf{x}_1,\mathbf{x}_2,\mathbf{y}_1,\mathbf{y}_2,\mathbf{z}_1,\mathbf{z}_2)$$
$$+\frac{\rho_1}{2}\left\|(\mathbf{x}_1,\mathbf{y}_1,\mathbf{y}_2) - (\mathbf{x}_1^{(k)},\mathbf{y}_1^{(k)},\mathbf{y}_2^{(k)})\right\|^2 - \frac{\rho_2}{2}\left\|(\mathbf{x}_2,\mathbf{z}_1,\mathbf{z}_2) - (\mathbf{x}_2^{(0)},\mathbf{z}_1^{(0)},\mathbf{z}_2^{(0)})\right\|^2\Big\}, \tag{21}$$

where $\rho_1 \ge 0$ is a proximal parameter and $\rho_2 \ge 0$ is a smoothing parameter. If ρ_1 and ρ_2 are set appropriately, subproblem (21) is a strongly-convex-strongly-concave (SCSC) minimax problem, which can be solved by the optimal method from [38, Algorithm 4] and [53, Algorithm 1]. We present this optimal method in Algorithm 2 in Section C and denote it by `OptFOM`. Using Algorithm 2 as a subroutine for (21) leads to [53, Algorithm 2], which is presented in the deterministic case in Algorithm 1. Its oracle complexity is presented in Theorem 4.3 with the proof given in Appendix D.

Theorem 4.3. *Suppose Assumption 3.1 holds and* $(\mathbf{x}_1^{(0)}, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)})$ *in Algorithm 1 satisfies*

$$p(\mathbf{x}_1^{(0)}, \mathbf{y}_1^{(0)}) - d(\mathbf{x}_1^{(0)}, \mathbf{y}_2^{(0)}) \le \epsilon. \tag{22}$$

Algorithm 1 outputs a lifted $O(\epsilon)$*-KKT point* $(\mathbf{x}_1^\epsilon, \mathbf{y}_1^\epsilon, \mathbf{y}_2^\epsilon)$ *of* (1) *with oracle complexity* $\tilde{O}(\epsilon^{-4})$*.*

Remark 4.4. To satisfy (22), one can simply choose any $\mathbf{x}_1^{(0)} \in \mathcal{X}_1$ and then apply any optimal first-order method, e.g., the one by [56], to the convex-concave minimax problem $\min_{\mathbf{y}_1}\max_{\mathbf{y}_2}\tilde{f}(\mathbf{x}_1^{(0)}, \mathbf{y}_1, \mathbf{y}_2)$ to find an ϵ-saddle point $(\mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)})$ that satisfies (22) in oracle complexity $O(1/\epsilon)$. This initialization procedure does not increase the complexity of Algorithm 1.

Under additional strong convexity and strong concavity assumptions on the lower-level problem, the oracle complexity established in Theorem 4.3 can be improved by appropriately adjusting the parameters in Algorithm 1. Specifically, under Assumptions 3.1 and J.1, the oracle complexity improves to $\tilde{O}(\epsilon^{-3.5})$. Under Assumptions 3.1 and J.3, it further improves to $\tilde{O}(\epsilon^{-3})$. These improved complexity bounds are established in Theorems J.2 and J.4, respectively, in the appendix.

5 Application on bilevel optimization with a constrained lower-level problem

In this section, we discuss the application of Algorithm 1 to (4). We first present assumptions on (4).

Assumption 5.1. Problem (4) has the structures in (5) and the following statements hold.

1. $\mathcal{X}_1 := \text{dom} f_2$ and $\mathcal{Y}_1 := \text{dom}\bar{f}_2$ are compact.
2. $f_1(\mathbf{x}_1, \mathbf{y}_1)$ is $L_{\nabla f_1}$-smooth on $\mathcal{X}_1 \times \mathcal{Y}_1$. $\bar{f}_1(\mathbf{x}_1, \mathbf{y}_1)$ is convex in $\mathbf{y}_1$ and $L_{\nabla \bar{f}_1}$-smooth on $\mathcal{X}_1 \times \mathcal{Y}_1$. $\bar{f}(\mathbf{x}_1, \mathbf{y}_1)$ is $L_{\bar{f}}$-Lipschitz on $\mathcal{X}_1 \times \mathcal{Y}_1$. $\bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{y}_1)$ is convex in $\mathbf{y}_1$ in each component, $L_{\nabla \bar{\mathbf{g}}}$-smooth and and $L_{\bar{\mathbf{g}}}$-Lipschitz on on $\mathcal{X}_1 \times \mathcal{Y}_1$.

Algorithm 1 A (deterministic or stochastic) inexact proximal point method for (1)

Input: $K > 0$, $\epsilon > 0$, $(\mathbf{x}_1^{(0)}, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)}) \in \mathcal{X}_1 \times \mathcal{Y}_1 \times \mathcal{Y}_2$, $(\mathbf{x}_2^{(0)}, \mathbf{z}_1^{(0)}, \mathbf{z}_2^{(0)}) \in \mathcal{X}_2 \times \mathcal{Y}_1 \times \mathcal{Y}_2$

Set $\rho = 1/\epsilon$, $\quad \rho_1 = 2L_{\nabla f} + 4\rho L_{\nabla \tilde{f}}$, $\quad \rho_2 = \epsilon/(2D_2)$ in (21).

Set $\hat{\epsilon} = \epsilon^{1.5}$, $\quad L_{\nabla P_1} = L_{\nabla f} + 2\rho L_{\nabla \tilde{f}}$, $\quad \delta_P^2 = 3\delta_f^2 + 6\rho^2 \delta_{\tilde{f}}^2$

for $k = 0, 1, 2, \ldots$ **do**

Set

$$\bar{h}(x, y) \leftarrow P_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2, \mathbf{z}_1, \mathbf{z}_2) + \frac{\rho_1}{2} \left\| (\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2) - (\mathbf{x}_1^{(k)}, \mathbf{y}_1^{(k)}, \mathbf{y}_2^{(k)}) \right\|^2 - \frac{\rho_2}{2} \left\| (\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2) - (\mathbf{x}_2^{(0)}, \mathbf{z}_1^{(0)}, \mathbf{z}_2^{(0)}) \right\|^2,$$

$$p(x) \leftarrow P_2(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2), \quad q(y) \leftarrow P_3(\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2),$$

$$L_{\nabla \bar{h}} \leftarrow L_{\nabla P_1} + \max\{\rho_1, \rho_2\}, \quad \sigma_x \leftarrow L_{\nabla P_1}, \quad \sigma_y \leftarrow \epsilon/(2D_2), \quad \delta^2 \leftarrow \delta_P^2$$

if (1) is deterministic (i.e., $\delta_f = \delta_{\tilde{f}} = 0$) **then**

Apply `OptFOM` in Algorithm 2 as follows.

$$\left((\mathbf{x}_1^{(k+1)}, \mathbf{y}_1^{(k+1)}, \mathbf{y}_2^{(k+1)}), (\mathbf{x}_2^{(k+1)}, \mathbf{z}_1^{(k+1)}, \mathbf{z}_2^{(k+1)}) \right)$$
$$\leftarrow \texttt{OptFOM}\left(\epsilon_k := \frac{\hat{\epsilon}}{k+1}, (\mathbf{x}_1^{(k)}, \mathbf{y}_1^{(k)}, \mathbf{y}_2^{(k)}), (\mathbf{x}_2^{(k)}, \mathbf{z}_1^{(k)}, \mathbf{z}_2^{(k)}), \bar{h}, p, q, L_{\nabla \bar{h}}, \sigma_x, \sigma_y \right)$$

if $\left\| (\mathbf{x}_1^{(k+1)}, \mathbf{y}_1^{(k+1)}, \mathbf{y}_2^{(k+1)}) - (\mathbf{x}_1^{(k)}, \mathbf{y}_1^{(k)}, \mathbf{y}_2^{(k)}) \right\| \le \epsilon/(4L_{\nabla P_1})$ **then**

Return $((\mathbf{x}_1^{\epsilon}, \mathbf{y}_1^{\epsilon}, \mathbf{y}_2^{\epsilon}), (\mathbf{x}_2^{\epsilon}, \mathbf{z}_1^{\epsilon}, \mathbf{z}_2^{\epsilon})) = \left((\mathbf{x}_1^{(k+1)}, \mathbf{y}_1^{(k+1)}, \mathbf{y}_2^{(k+1)}), (\mathbf{x}_2^{(k+1)}, \mathbf{z}_1^{(k+1)}, \mathbf{z}_2^{(k+1)}) \right)$

end if

else

Apply `SAPD` in Algorithm 4 as follows.

$$\left((\mathbf{x}_1^{(k+1)}, \mathbf{y}_1^{(k+1)}, \mathbf{y}_2^{(k+1)}), (\mathbf{x}_2^{(k+1)}, \mathbf{z}_1^{(k+1)}, \mathbf{z}_2^{(k+1)}) \right)$$
$$\leftarrow \texttt{SAPD}\left(\hat{\epsilon}, (\mathbf{x}_1^{(k)}, \mathbf{y}_1^{(k)}, \mathbf{y}_2^{(k)}), (\mathbf{x}_2^{(k)}, \mathbf{z}_1^{(k)}, \mathbf{z}_2^{(k)}), \bar{h}, p, q, L_{\nabla \bar{h}}, \sigma_x, \sigma_y, \delta^2 \right)$$

if $k = K - 1$ **then**

Return $((\mathbf{x}_1^{\epsilon}, \mathbf{y}_1^{\epsilon}, \mathbf{y}_2^{\epsilon}), (\mathbf{x}_2^{\epsilon}, \mathbf{z}_1^{\epsilon}, \mathbf{z}_2^{\epsilon})) = \left((\mathbf{x}_1^{(k')}, \mathbf{y}_1^{(k')}, \mathbf{y}_2^{(k')}), (\mathbf{x}_2^{(k')}, \mathbf{z}_1^{(k')}, \mathbf{z}_2^{(k')}) \right)$, where k' is uniformly randomly sampled from $\{1, 2, \ldots, K\}$.

end if

end if

end for

3. f_2 and $\bar{f}_2$ are proper closed convex and their proximal operators can be computed exactly.
4. (Slater's condition) There exists $G > 0$ such that, for any $\mathbf{x}_1 \in \mathcal{X}_1$, there exists $\hat{\mathbf{y}}_1 \in \mathcal{Y}_1$ such that $\bar{g}_i(\mathbf{x}_1, \hat{\mathbf{y}}_1) \le -G$ for $i = 1, \ldots, l$.

We define the following quantities, which are all finite according to Assumption 5.1.

$$D_{\mathcal{Y}_1} := \max\{\|\mathbf{y}_1 - \mathbf{y}_1'\| \mid \mathbf{y}_1, \mathbf{y}_1' \in \mathcal{Y}_1\}, \quad B := \frac{2L_{\bar{f}} D_{\mathcal{Y}_1}}{G}. \tag{23}$$

The value of B above is used in (6). Our goal is to find a lifted ϵ-KKT point of (4) defined by [54, 52].

Definition 5.2. $(\mathbf{x}_1, \mathbf{y}_1)$ is a lifted ϵ-KKT point of (4) if there exists $(\mathbf{z}_1, \rho, \boldsymbol{\lambda}, \bar{\boldsymbol{\lambda}}) \in \mathcal{Y}_1 \times \mathbb{R}_+ \times \mathbb{R}_+^l \times \mathbb{R}_+^l$ such that

$$\operatorname{dist}\left(\mathbf{0}, \partial_{(\mathbf{x}_1, \mathbf{y}_1)} \left[f(\mathbf{x}_1, \mathbf{y}_1) + \rho\big(\bar{f}(\mathbf{x}_1, \mathbf{y}_1) - \bar{f}(\mathbf{x}_1, \mathbf{z}_1) - \bar{\boldsymbol{\lambda}}^T \bar{g}(\mathbf{x}_1, \mathbf{z}_1)\big) + \boldsymbol{\lambda}^T \bar{g}(\mathbf{x}_1, \mathbf{y}_1) \right] \right) \le \epsilon,$$

$$\operatorname{dist}\left(\mathbf{0}, \rho \partial_{\mathbf{z}_1} \left[\bar{f}(\mathbf{x}_1, \mathbf{z}_1) + \bar{\boldsymbol{\lambda}}^T \bar{g}(\mathbf{x}_1, \mathbf{z}_1) \right] \right) \le \epsilon, \quad \|[\bar{g}(\mathbf{x}_1, \mathbf{z}_1)]_+\| \le \epsilon, \quad \left| \bar{\boldsymbol{\lambda}}^T \bar{g}(\mathbf{x}_1, \mathbf{z}_1) \right| \le \epsilon,$$

$$\bar{f}(\mathbf{x}_1, \mathbf{y}_1) - \bar{f}^*(\mathbf{x}_1) \le \epsilon, \quad \|[\bar{g}(\mathbf{x}_1, \mathbf{y}_1)]_+\| \le \epsilon, \quad \left| \boldsymbol{\lambda}^T \bar{g}(\mathbf{x}_1, \mathbf{y}_1) \right| \le \epsilon,$$

where $\bar{f}^*(\mathbf{x}_1) := \min_{\mathbf{y}_1} \left\{ \bar{f}(\mathbf{x}_1, \mathbf{y}_1) | \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{y}_1) \le 0 \right\}$ for any $\mathbf{x}_1 \in \mathcal{X}_1$.

It is well known that, under Slater's condition for the lower-level problem, (4) can be reformulated as (6), which is a special case of (1). This reformulation holds because of the lemma below whose proof is standard and given in Appendix E.1.

Lemma 5.3. *Suppose Assumption 5.1 holds. For any fixed $\mathbf{x}_1 \in \mathcal{X}_1$, a $\mathbf{y}_1 \in \mathcal{Y}_1$ is an optimal solution to the lower-level problem of* (4) *if and only if there exists $\mathbf{y}_2 \in [0, B]^l$ such that $(\mathbf{y}_1, \mathbf{y}_2)$ is a saddle-point of $\min_{\mathbf{z}_1} \max_{\mathbf{z}_2} \tilde{f}(\mathbf{x}_1, \mathbf{z}_1, \mathbf{z}_2)$, where $\tilde{f}(\mathbf{x}_1, \mathbf{z}_1, \mathbf{z}_2)$ is defined in* (7)*.*

According to Lemma 5.3, we can solve (4) by solving (6) using Algorithm 1. Below we show that a lifted ϵ-KKT point of (6) is also a lifted $O(\epsilon)$-KKT point of (4). The proof is given in Appendix E.2.

Lemma 5.4. *Suppose $\epsilon \leq \min\{\frac{\rho L_{\bar{f}}}{4}, \frac{\rho G}{4}\}$. A lifted ϵ-KKT point for* (6) *under Definition 3.2 is a lifted $O(\epsilon)$-KKT point of* (4) *under Definition 5.2.*

With this lemma, we can characterize the complexity for Algorithm 1 to find a lifted $O(\epsilon)$-KKT point of (4) in the following theorem whose proof is given in Appendix F.

Theorem 5.5. *Suppose $\epsilon \leq \min\{\frac{\rho L_{\bar{f}}}{4}, \frac{\rho G}{4}\}$ and Algorithm 1 is applied to* (6)*. Suppose Assumption 5.1 holds and $(\mathbf{x}_1^{(0)}, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)})$ in Algorithm 1 satisfies*

$$\max_{\mathbf{z}_2 \in [0,B]} \left\{ \bar{f}(\mathbf{x}_1^{(0)}, \mathbf{y}_1^{(0)}) + \mathbf{z}_2^\top \bar{\mathbf{g}}(\mathbf{x}_1^{(0)}, \mathbf{y}_1^{(0)}) \right\} - \min_{\mathbf{z}_1} \left\{ \bar{f}(\mathbf{x}_1^{(0)}, \mathbf{z}_1) + (\mathbf{y}_2^{(0)})^\top \bar{\mathbf{g}}(\mathbf{x}_1^{(0)}, \mathbf{z}_1) \right\} \leq \epsilon \tag{24}$$

Algorithm 1 outputs a lifted $O(\epsilon)$-KKT point $(\mathbf{x}_1^\epsilon, \mathbf{y}_1^\epsilon)$ of (4) *in oracle complexity $\tilde{O}(\epsilon^{-4})$.*

Remark 5.6. The methods of Lu and Mei [52, 54] achieve an oracle complexity of $\tilde{O}(\epsilon^{-7})$ for finding a lifted $O(\epsilon)$-KKT point of (4). This bound can be improved to $\tilde{O}(\epsilon^{-6})$ under strong convexity of the lower-level problem [54]. The relatively high oracle complexity of these methods stems in part from the use of two penalty parameters, $\rho = \Theta(\epsilon^{-1})$ and $\mu = \Theta(\epsilon^{-2})$, associated with the outer- and lower-level problems, respectively. The resulting minimax penalty reformulation has a smoothness constant of order $\Theta(\rho\mu) = \Theta(\epsilon^{-3})$. In contrast, our reformulation (6) does not require the additional penalty parameter μ, yielding a smoothness constant of order $\Theta(\rho) = \Theta(\epsilon^{-1})$. Nevertheless, our approach relies on the equivalence established in Lemma 5.3. If this equivalence fails, for instance, due to the lack of strong duality or uniformly bounded optimal dual solutions over $\mathbf{x}_1 \in \mathcal{X}$, our method is not applicable. By contrast, the methods of Lu and Mei [52, 54] do not rely on the reformulation (6) and can therefore be stilll applied.

6 Stochastic bilevel optimization with a minimax lower level problem

In this section, we still consider problem (1) when gradients f_1 and $\tilde{f}_1$ can only be accessed through stochastic oracles. In particular, in addition to Assumption 3.1, we make the following assumptions.

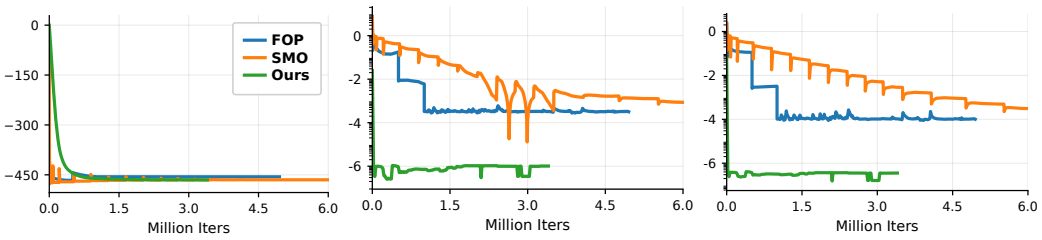
Assumption 6.1. There exist a random variable ω, two constants δ_f and $\delta_{\tilde{f}}$, and stochastic gradient estimators of $\nabla f_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2)$ and $\nabla \tilde{f}_1(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2)$, which are mappings of $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2)$ and ω denoted by $\hat{\nabla} f_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2; \omega)$ and $\hat{\nabla} \tilde{f}_1(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2; \omega)$, respectively, and satisfy

$$\begin{aligned}
&\mathbb{E}[\hat{\nabla} f_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2; \omega)] = \nabla f_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2), \quad \mathbb{E}[\hat{\nabla} \tilde{f}_1(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2; \omega)] = \nabla \tilde{f}_1(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2), \\
&\mathbb{E}[\|\hat{\nabla} f_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2; \omega) - \nabla f_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2)\|^2] \leq \delta_f^2, \\
&\mathbb{E}[\|\hat{\nabla} \tilde{f}_1(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2; \omega) - \nabla \tilde{f}_1(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2)\|^2] \leq \delta_{\tilde{f}}^2.
\end{aligned}$$

A key design that allows Algorithm 1 to find a lifted ϵ-KKT point of (1) in the deterministic case is the proximal steps (see lines 26 and 27) in Algorithm 2, which require deterministic gradients and thus are not implementable in the stochastic setting. Therefore, we consider a slightly different notion of a lifted ϵ-KKT point in the stochastic case.

Definition 6.2. A solution $(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2)$ is a nearly lifted ϵ-KKT point of (1) if there exists $(\hat{\mathbf{x}}_1, \hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2)$ such that $\|(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2) - (\hat{\mathbf{x}}_1, \hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2)\| \leq \epsilon$ and $(\hat{\mathbf{x}}_1, \hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2)$ is a lifted ϵ-KKT point (1).

Under Assumptions 3.1 and 6.1, (14) is a stochastic NCC minimax problem, which can be solved by a stochastic inexact proximal point method such as the SAPD+ method by [76]. SAPD+ updates solutions also in the way of (21) except that the subproblem (21) is also solved by the stochastic accelerated primal-dual (SAPD) method [77]. The SAPD+ method applied to (14) is described in



(a) f(x_1, y_1) (b) $\log_{10}(p(\mathbf{x}_1, \mathbf{y}_1) - d(\mathbf{x}_1, \mathbf{y}_2))$ (c) $\log_{10}(\|[\tilde{\mathbf{g}}(\mathbf{x}_1, \mathbf{z}_1)]_+\|)$

Figure 1: Performance for an instance with $(n, m, l) = (300, 300, 15)$ and tuned Lipschitz constants.

Algorithm 1 when the problem is stochastic (i.e., the condition of the if-statement in Line 6 is false) and SAPD is presented in Algorithm 4 in Section G for a generic SCSC minimax problem. The complexity of Algorithm 1 in the stochastic case is given below with proof provided in Appendix H.

Theorem 6.3. *Suppose Assumptions 3.1 and 6.1 hold,* $(\mathbf{x}_1^{(0)}, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)})$ *in Algorithm 1 satisfies*

$$\mathbb{E}[p(\mathbf{x}_1^{(0)}, \mathbf{y}_1^{(0)}) - d(\mathbf{x}_1^{(0)}, \mathbf{y}_2^{(0)})] \leq \epsilon, \tag{25}$$

and K *in Algorithm 1 satisfies* $K = \left\lceil \left(\mathbb{E}\left[\max_{\mathbf{x}_2} f(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)}) \right] + 1 - f^* + \epsilon D_2/4 \right) \epsilon^{-3} \right\rceil$. *Algorithm 1 terminates and outputs a nearly lifted* $O(\epsilon)$*-KKT point of* (1) *in expectation in stochastic oracle complexity* $\tilde{O}(\epsilon^{-9})$.

Under additional strong convexity and strong concavity assumptions on the lower-level problem, the oracle complexity established in Theorem 6.3 can be improved by appropriately adjusting the parameters in Algorithm 1. Specifically, under Assumptions 3.1 and J.1, the oracle complexity improves to $\tilde{O}(\epsilon^{-8})$. Under Assumptions 3.1 and J.3, it further improves to $\tilde{O}(\epsilon^{-7})$. These improved complexity bounds are established in Theorems J.5 and J.6, respectively, in the appendix.

7 Numerical experiments

In this section, we evaluate the numerical performance of the proposed methods. Additional experimental details are provided in Section L.

7.1 Constrained linear optimization

We consider the following instance of (4) from [52, 54]:

$$\min_{\mathbf{x},\mathbf{y}} \; c^T\mathbf{x} + d^T\mathbf{y} + \mathbf{1}_{[-1,1]^n}(\mathbf{x}) \quad \textbf{s.t.} \quad \mathbf{y} \in \arg\min_{\mathbf{z}} \left\{ \tilde{d}^T\mathbf{z} + \mathbf{1}_{[-1,1]^m}\mathbf{z} \mid \widetilde{A}\mathbf{x} + \widetilde{B}\mathbf{z} - \tilde{b} \leq 0 \right\}, \tag{26}$$

where $c \in \mathbb{R}^n$, $d, \tilde{d} \in \mathbb{R}^m$, $\tilde{b} \in \mathbb{R}^l$, $\widetilde{A} \in \mathbb{R}^{l\times n}$, and $\widetilde{B} \in \mathbb{R}^{l\times m}$. Five instances are generated per tuple (n, m, l) as described in Appendix L.1.1, and we set the initial solution as $(\mathbf{x}^{(0)}, \mathbf{y}^{(0)}) = (0, 0)$.

We compare our method against FOP [52] and SMO [54]. Overall, our method achieves competitive results against the two baselines when all methods use respective theoretical Lipschitz constants to determine step sizes, and our method performs significantly better after we tune the Lipschitz constants for each method. In Figure 1, we plot for one instance the values of $f(\mathbf{x}_1, \mathbf{y}_1)$, $p(\mathbf{x}_1, \mathbf{y}_1) - d(\mathbf{x}_1, \mathbf{y}_2)$, and $\|[\tilde{\mathbf{g}}(\mathbf{x}_1, \mathbf{z}_1)]_+\|$ over the number of gradient steps. Our method reduces the last two quantities faster than the others. In Table 1, we report $f(\mathbf{x}_1, \mathbf{y}_1)$ at termination.

7.2 Hyperparameter tuning for group distributionally robust optimization (Group DRO)

Empirical risk minimization (ERM) assigns uniform weights to training samples, resulting in potentially poor worst-group performance on an unbalanced dataset [62]. Instead of creating balanced data, which are expensive [61], we propose a deep bilevel DRO framework by decomposing the model into a feature extractor $\mathbf{x}_1$ and a linear classifier $\mathbf{y}_1$, where a representation of $\mathbf{a}$, denoted by $\texttt{Enc}(\mathbf{a}; \mathbf{x}_1)$, is produced by $\mathbf{x}_1$, on which $\mathbf{y}_1$ predicts label b. This separation isolates non-convex representation learning from convex classification [32, 36]. We denote the loss on data $(\mathbf{a}, b)$ by $\ell^{\mathbf{x}_1}(\mathbf{y}_1, (\mathbf{a}, b))$.

(n, m, ℓ)	FOP	FOP-t	SMO	SMO-t	Ours	Ours-t
$(100, 100, 5)$	-141.67	-141.69	-145.78	-145.63	-143.14	**-145.85**
$(200, 200, 10)$	-278.67	-282.31	-305.95	-305.97	-300.67	**-310.84**
$(300, 300, 15)$	-439.92	-448.36	-455.91	-461.45	-438.58	**-462.92**

Table 1: Upper objective $f(\mathbf{x}_1, \mathbf{y}_1)$ at termination for (26). "-t" means using tuned a Lipschitz constant to construct step sizes, and it means using the theoretical Lipschitz constant without "-t".

Lower Level: Group DRO Let $D_{\text{train}} = \cup_{g=1}^{G} D_{\text{train}}^{(g)}$. We define

$$\min_{\mathbf{y}_1} \max_{\mathbf{y}_2 \in \mathcal{Y}} \ \mathcal{L}_{\text{train}}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2; \eta) := \mathbf{y}_2^T \mathcal{L}^{\mathbf{x}_1}(\mathbf{y}_1) \ - \ \frac{\eta}{2} \|\mathbf{y}_2 - \tfrac{1}{G}\mathbf{1}\|^2 \ + \ \lambda \|\mathbf{y}_1\|^2, \tag{27}$$

where $\mathcal{L}^{\mathbf{x}_1}(\mathbf{y}_1) := (\mathcal{L}_1^{\mathbf{x}_1}, \ldots, \mathcal{L}_G^{\mathbf{x}_1})$, $\mathcal{L}_g^{\mathbf{x}_1}(\mathbf{y}_1) := \frac{1}{|D_{\text{train}}^{(g)}|} \sum_{(\mathbf{a},b) \in D_{\text{train}}^{(g)}} \ell^{\mathbf{x}_1}(\mathbf{y}_1, (\mathbf{a}, b))$, and $\mathcal{Y} = \{\mathbf{y} \geq 0 : \mathbf{1}^T \mathbf{y} = 1\}$. The learnable weight $\mathbf{y}_2$ emphasizes high-loss groups with η controlling deviation from uniform weights. The lower problem is convex–concave in $(\mathbf{y}_1, \mathbf{y}_2)$ due to the linear classifier.

Upper Level: Hyperparameter Tuning We tune η and $\mathbf{x}_1$ using validation data D_{val}, partitioned analogously. The bilevel problem is

$$\min_{\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2, \eta \geq 0} \max_{\mathbf{x}_2 \in \mathcal{Y}} \ \mathbf{x}_2^T \mathcal{L}_{\text{val}}^{\mathbf{x}_1}(\mathbf{y}_1) + \lambda \|\mathbf{x}_1\|^2 \quad \textbf{s.t.} \quad (\mathbf{y}_1, \mathbf{y}_2) \in \arg\min_{\mathbf{z}_1} \max_{\mathbf{z}_2 \in \mathcal{Y}} \mathcal{L}_{\text{train}}(\mathbf{x}_1, \mathbf{z}_1, \mathbf{z}_2; \eta), \tag{28}$$

where $\mathcal{L}_{\text{val}}^{\mathbf{x}_1}(\mathbf{y}_1) := (\mathcal{L}_{\text{val},1}^{\mathbf{x}_1}, \ldots, \mathcal{L}_{\text{val},G}^{\mathbf{x}_1})$. Here, $\max_{\mathbf{x}_2 \in \mathcal{Y}}$ computes the worst-group validation loss.

Results We use the stochastic version of our method with $\rho = 100$ and minibatch size of 256 for CelebA and 128 for Waterbirds. We evaluate our method on CelebFaces Attributes (CelebA) [51] and Waterbirds [62] datasets, where binary labels are confounded by two spurious attributes, yielding 4 groups, where the minority group contributes to 3% of the entire training set. Our method yields average performance close to DRO while outperforming it on worst group accuracy, with and without ℓ_2 penalty (See Table 2). See Appendix L.2 for additional experimental details.

Dataset	ℓ_2 **Penalty**	**Worst Group**		**Average**	
	λ	Group DRO	Ours	Group DRO	Ours
CelebA	1e-4	42.04 ± 2.10	$\mathbf{86.06 \pm 2.48}$	94.84 ± 0.06	91.11 ± 1.07
	0.1	85.37 ± 1.40	$\mathbf{87.52 \pm 0.83}$	93.70 ± 0.26	90.26 ± 1.13
Waterbirds	1e-4	76.53 ± 1.72	$\mathbf{83.61 \pm 5.58}$	97.25 ± 0.07	90.05 ± 2.00
	1.0	84.22 ± 1.32	$\mathbf{87.13 \pm 1.57}$	96.14 ± 0.10	90.89 ± 2.77

Table 2: Worst-group and average accuracy for DRO and our method on the test set.

8 Limitations

This work has the following limitations: (1) Assumption 5.1 requires compactness of the feasible domains. (2) The lifted ϵ-KKT point in Definition 3.2 is defined in a lifted space with additional variables $\mathbf{z}_1$ and $\mathbf{z}_3$. It would be desirable to study a stronger stationarity notion that only involves the original variables. (3) In the stochastic case, the $\tilde{O}(\epsilon^{-9})$ complexity is high. We believe using the variance-reduction technique can reduce this complexity, which we leave it as future work.

9 AI use statement

All mathematical results and proofs in this paper were developed by the authors except Lemma 5.3, which was initially generated by Gemini and polished and verified by the authors. GPT-6 was used to assist with language editing and to improve the clarity and presentation of the manuscript. It was also used to assist in writing portions of the code for the numerical experiments. All AI-assisted code was carefully reviewed and verified by the authors. The authors take full responsibility for the correctness of all mathematical results and proofs, experimental implementations, and the final manuscript.

References

[1] Gemayqzel Bouza Allende and Georg Still. Solving bilevel programs with the kkt-approach. *Mathematical programming*, 138(1):309–332, 2013.

[2] Michael Arbel and Julien Mairal. Non-convex bilevel games with critical point selection maps. *Advances in Neural Information Processing Systems*, 35:8013–8026, 2022.

[3] Aharon Ben-Tal, Dick Den Hertog, Anja De Waegenaere, Bertrand Melenberg, and Gijs Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013.

[4] Kristin P. Bennett, Gautam Kunapuli, Jing Hu, and Jong-Shi Pang. Bilevel optimization and machine learning. In *IEEE World Congress on Computational Intelligence*, pages 25–47. Springer, 2008.

[5] Luca Bertinetto, Joao F. Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2018.

[6] Dimitri P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48 (3):334–334, 1997.

[7] 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.

[8] 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. URL `http://jmlr.org/papers/v26/23-1104.html`.

[9] Tianyi Chen, Yuejiao Sun, and Wotao Yin. Closing the gap: Tighter analysis of alternating stochastic gradient methods for bilevel problems. *Advances in Neural Information Processing Systems*, 34:25294–25307, 2021.

[10] Tianyi Chen, Yuejiao Sun, Quan Xiao, and Wotao Yin. A single-timescale method for stochastic bilevel optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 2466–2488, 2022.

[11] 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.

[12] Liam Collins, Aryan Mokhtari, and Sanjay Shakkottai. Task-robust model-agnostic meta-learning. *Advances in Neural Information Processing Systems*, 33:18860–18871, 2020.

[13] John C Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49(3):1378–1406, 2021.

[14] Pavel Dvurechensky, Andrea Ebner, Johannes Carl Schnebel, Shimrit Shtern, and Mathias Staudigl. Stochastic variance reduced extragradient methods for solving hierarchical variational inequalities. *arXiv preprint arXiv:2602.13510*, 2026.

[15] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[16] 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, 2018.

[17] Lucy L Gao, Jane Ye, Haian Yin, Shangzhi Zeng, and Jin Zhang. Value function based difference-of-convex algorithm for bilevel hyperparameter selection problems. In *International conference on machine learning*, pages 7164–7182. PMLR, 2022.

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

[19] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[20] 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, 2020.

[21] Zhishuai Guo, Quanqi Hu, Lijun Zhang, and Tianbao Yang. Randomized stochastic variance-reduced methods for multi-task stochastic bilevel optimization. *arXiv preprint arXiv:2105.02266*, 2021.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

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

[24] Xiaoyin Hu, Nachuan Xiao, Xin Liu, and Kim-Chuan Toh. An improved unconstrained approach for bilevel optimization. *SIAM Journal on Optimization*, 33(4):2801–2829, 2023.

[25] Feihu Huang. Adaptive mirror descent bilevel optimization. *arXiv preprint arXiv:2311.04520*, 2023.

[26] Feihu Huang. On momentum-based gradient methods for bilevel optimization with nonconvex lower-level. *arXiv preprint arXiv:2303.03944*, 2023.

[27] Feihu Huang. Optimal hessian/jacobian-free nonconvex-pl bilevel optimization. In *Proceedings of the 41st International Conference on Machine Learning*, pages 19598–19621, 2024.

[28] Feihu Huang, Junyi Li, Shangqian Gao, and Heng Huang. Enhanced bilevel optimization via bregman distance. *Advances in Neural Information Processing Systems*, 35:28928–28939, 2022.

[29] Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. In *ICML*, pages 4882–4892, 2021.

[30] Liuyuan Jiang, Quan Xiao, Victor M. Tenorio, Fernando Real-Rojas, Antonio G. Marques, and Tianyi Chen. A primal-dual-assisted penalty approach to bilevel optimization with coupled constraints. *Advances in Neural Information Processing Systems*, 37:95026–95066, 2024.

[31] Liuyuan Jiang, Quan Xiao, Lisha Chen, and Tianyi Chen. Beyond value functions: Single-loop bilevel optimization under flatness conditions. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

[32] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations (ICLR)*, 2020.

[33] Mohammad Khalafi and Digvijay Boob. Regularized operator extrapolation method for stochastic bilevel variational inequality problems. *arXiv preprint arXiv:2505.09778*, 2025.

[34] 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 *NeurIPS*, 2021.

[35] Prashant Khanduri, Ioannis Tsaknakis, Yihua Zhang, Jia Liu, Sijia Liu, Jiawei Zhang, and Mingyi Hong. Linearly constrained bilevel optimization: A smoothed implicit gradient approach. In *International Conference on Machine Learning*, pages 16291–16325. PMLR, 2023.

[36] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. In *International Conference on Learning Representations (ICLR)*, 2023.

[37] Guy Kornowski, Swati Padmanabhan, Kai Wang, Jimmy Zhang, and Suvrit Sra. First-order methods for linearly constrained bilevel optimization. *Advances in neural information processing systems*, 37.

[38] Dmitry Kovalev and Alexander Gasnikov. The first optimal algorithm for smooth minimax optimization. In *NeurIPS*, 2022.

[39] 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. PMLR, 2023.

[40] Jeongyeol Kwon, Dohyun Kwon, Stephen Wright, and Robert Nowak. On penalty methods for nonconvex bilevel optimization and first-order stochastic approximation. In *International Conference on Learning Representations*, volume 2024, pages 47473–47518, 2024.

[41] Chenliang Li, Siliang Zeng, Zeyi Liao, Jiaxiang Li, Dongyeop Kang, Alfredo Garcia, and Mingyi Hong. Learning reward and policy jointly from demonstration and preference improves alignment. *arXiv preprint arXiv:2406.06874*, 2024.

[42] Junyi Li, Bin Gu, and Heng Huang. A fully single loop algorithm for bilevel optimization without hessian inverse. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7426–7434, 2022.

[43] Yuwei Li, Gui-Hua Lin, Jin Zhang, and Xide Zhu. A novel approach for bilevel programs based on wolfe duality. *arXiv preprint arXiv:2302.06838*, 2023.

[44] Bo Liu, Mao Ye, Stephen Wright, Peter Stone, and Qiang Liu. Bome! bilevel optimization made easy. In *NeurIPS*, 2022.

[45] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *ICLR*, 2018.

[46] 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. PMLR, 2021.

[47] Risheng Liu, Yaohua Liu, Shangzhi Zeng, and Jin Zhang. Towards gradient-based bilevel optimization with non-convex followers and beyond. *Advances in Neural Information Processing Systems*, 34:8662–8675, 2021.

[48] Risheng Liu, Pan Mu, Xiaoming Yuan, Shangzhi Zeng, and Jin Zhang. A general descent aggregation framework for gradient-based bi-level optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):38–57, 2022.

[49] 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*, 45(12):15930–15948, 2023.

[50] 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*, pages 21839–21866. PMLR, 2023.

[51] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738, 2015.

[52] Zhaosong Lu and Sanyou Mei. First-order penalty methods for bilevel optimization. *SIAM Journal on Optimization*, 34(2):1937–1969, 2024. doi: 10.1137/23M1566753. URL `https://doi.org/10.1137/23M1566753`.

[53] Zhaosong Lu and Sanyou Mei. A first-order augmented lagrangian method for constrained minimax optimization. *Mathematical Programming*, 213(1):1063–1104, September 2025. doi: 10.1007/s10107-024-02163-3. URL https://doi.org/10.1007/s10107-024-02163-3.

[54] Zhaosong Lu and Sanyou Mei. Solving bilevel optimization via sequential minimax optimization. *Mathematics of Operations Research*, 2026. doi: 10.1287/moor.2024.0521. URL https://doi.org/10.1287/moor.2024.0521. Published online January 6, 2026.

[55] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[56] Arkadi Nemirovski. Prox-method with rate of convergence o(1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

[57] Hantao Nie, Jiaxiang Li, and Zaiwen Wen. An augmented lagrangian value function method for lower-level constrained stochastic bilevel optimization. *arXiv preprint arXiv:2509.24249*, 2025.

[58] Aurélien Ouattara and Anil Aswani. Duality approach to bilevel programs with a convex lower level. In *2018 Annual American Control Conference (ACC)*, pages 1388–1395. IEEE, 2018.

[59] Fabian Pedregosa. Hyperparameter optimization with approximate gradient. In *ICML*, 2016.

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

[61] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning (ICML)*, pages 4334–4343. PMLR, 2018.

[62] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations (ICLR)*, 2020.

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

[64] Han Shen, Quan Xiao, and Tianyi Chen. On penalty-based bilevel gradient descent method. *Mathematical Programming*, pages 1–51, 2025.

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

[66] Ioannis Tsaknakis, Prashant Khanduri, and Mingyi Hong. An implicit gradient-type method for linearly constrained bilevel problems. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5438–5442. IEEE, 2022.

[67] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[68] Quan Xiao, Han Shen, Wotao Yin, and Tianyi Chen. Alternating implicit projected sgd and its efficient variants for equality-constrained bilevel optimization. *arXiv preprint arXiv:2211.07096*, 2022.

[69] Quan Xiao, Songtao Lu, and Tianyi Chen. An alternating optimization method for bilevel problems under the polyak-łojasiewicz condition. *Advances in Neural Information Processing Systems*, 36:63847–63873, 2023.

[70] Siyuan Xu and Minghui Zhu. Efficient gradient approximation method for constrained bilevel optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12509–12517, 2023.

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

[72] Yan Yang, Bin Gao, and Ya-xiang Yuan. Bilevel reinforcement learning via the development of hyper-gradient without lower-level convexity. *arXiv preprint arXiv:2405.19697*, 2024.

[73] Wei Yao, Haian Yin, Shangzhi Zeng, and Jin Zhang. Overcoming lower-level constraints in bilevel optimization: A novel approach with regularized gap functions. In *The Thirteenth International Conference on Learning Representations*, .

[74] Wei Yao, Chengming Yu, Shangzhi Zeng, and Jin Zhang. Constrained bi-level optimization: Proximal lagrangian value function approach and hessian-free algorithm. In *The Twelfth International Conference on Learning Representations*, .

[75] 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.

[76] Xuan Zhang, Necdet Serhat Aybat, and Mert Gurbuzbalaban. Sapd+: An accelerated stochastic method for nonconvex-concave minimax problems. *Advances in Neural Information Processing Systems*, 35:21668–21681, 2022.

[77] Xuan Zhang, Necdet Serhat Aybat, and Mert Gürbüzbalaban. Robust accelerated primal-dual methods for computing saddle points. *SIAM Journal on Optimization*, 34(1):1097–1130, 2024.

[78] Xuelin Zhang and Peipei Yuan. On the stability and generalization of first-order bilevel minimax optimization. *arXiv preprint arXiv:2604.20115*, 2026.

[79] Zihao Zheng, Irwin King, and Songtao Lu. Bilevel optimization over saddle points of zero-sum markov games. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 163179–163238. PMLR, 2026.

[80] Bolei Zhou, Àgata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP:1–1, 07 2017. doi: 10.1109/TPAMI.2017.2723009.

A Additional applications of (1)

Model (1) can be potentially applied to not only many machine learning problems but also bilevel optimization with a convex lower-level problem with nonlinear convex inequality constraints. We list some of these applications here.

Distributionally robust learning with tuning hyperparameters Let $\mathbf{a}$ be a feature vector and b be a target variable. Suppose a model $s(\boldsymbol{\theta}, \mathbf{a})$ parameterized by $\boldsymbol{\theta}$ is used to predict b based on $\mathbf{a}$. Let $\ell(s(\boldsymbol{\theta}, \mathbf{a}), b)$ be the loss of the prediction. Given a training dataset $D_{\text{train}} = \{(\mathbf{a}_t^i, b_t^i)\}_{i=1}^{n_{\text{train}}}$ and a validation dataset $D_{\text{val}} = \{(\mathbf{a}_v^i, b_v^i)\}_{i=1}^{n_{\text{val}}}$, let $\mathbf{y}_2 \in \mathbb{R}^{n_{\text{train}}}$ and $\mathbf{x}_2 \in \mathbb{R}^{n_{\text{val}}}$ be the weights on the training and validation instances, respectively. The weighted average losses on the training and validation sets are defined as

$$\mathcal{L}_{\text{DRO}}(D_{\text{train}}; \boldsymbol{\theta}, \mathbf{y}_2) := \sum_{i=1}^{n_{\text{train}}} y_2^i \cdot \ell(s(\boldsymbol{\theta}, \mathbf{a}_t^i), b_t^i) \tag{29}$$

$$\mathcal{L}_{\text{DRO}}(D_{\text{val}}; \boldsymbol{\theta}, \mathbf{x}_2) := \sum_{i=1}^{n_{\text{val}}} x_2^i \cdot \ell(s(\boldsymbol{\theta}, \mathbf{a}_v^i), b_v^i). \tag{30}$$

The distributionally robust learning problem with tuning hyperparameters can be formulated as

$$\begin{aligned} &\min_{\eta \geq 0, \boldsymbol{\theta}} \max_{\mathbf{x}_2 \in \Delta_{n_{\text{val}}, p}} \mathcal{L}_{\text{DRO}}(D_{\text{val}}; \boldsymbol{\theta}, \mathbf{x}_2) \\ &\text{s.t.} \quad \boldsymbol{\theta} \in \arg\min_{\boldsymbol{\theta}'} \max_{\mathbf{y}_2 \in \Delta_{n_{\text{train}}, 1}} \left\{ \mathcal{L}_{\text{DRO}}(D_{\text{train}}; \boldsymbol{\theta}', \mathbf{y}_2) - \frac{\eta}{2} \|\mathbf{y}_2 - \frac{1}{n_{\text{train}}} \mathbf{1}\|^2 \right\} \end{aligned} \tag{31}$$

where $\Delta_{n,p} = \{\mathbf{y} \in \mathbb{R}_+^n | \mathbf{1}^\top \mathbf{y} = 1, y_i \in [0, p], i = 1, \ldots, n\}$ is the truncated probability simplex. The objective function above is chosen to minimize the average loss on the $1/(pn) \times 100\%$ worst cases. However, if one directly minimizes the average loss on the $1/(pn) \times 100\%$ worst training instances, the model is not guaranteed to minimize the $1/(pn) \times 100\%$ worst instances in the populations due to the generalization error, especially, when n_{train} is small. Therefore, a regularization term $\frac{\eta}{2}\|\mathbf{y}_2 - 1/n_{\text{train}}\mathbf{1}\|^2$ with a changeable hyperparameter $\eta \geq 0$ is introduced to the lower level problem. The goal is to minimize the average loss on the $1/(pn) \times 100\%$ worst cases on the validation set by the trained model through optimizing the parameter η. (31) is thus an instance of (1) with $\mathbf{x}_1 = \eta$ and $\mathbf{y}_1 = \boldsymbol{\theta}$.

Distributionally robust meta-learning The robust learning framework in (31) can be extended to meta learning [16, 15, 60, 12]. Suppose there are $|\mathcal{T}|$ tasks, indexed by $\{1, \ldots, \mathcal{T}\}$, and the training and testing datasets for task τ are denoted by $D_{\text{train}}^\tau = \{(\mathbf{a}_t^{\tau,i}, b_t^{\tau,i})\}_{i=1}^{n_{\text{train}}}$ and $D_{\text{val}}^\tau = \{(\mathbf{a}_v^{\tau,i}, b_v^{\tau,i})\}_{i=1}^{n_{\text{val}}}$, respectively, for $\tau \in \mathcal{T}$. Let $\boldsymbol{\theta}$ be the meta-parameter and $\boldsymbol{\theta}^\tau$ be the task-specifc parameter for task τ. The distributionally meta-learning can be formulated as follows

$$\begin{aligned} &\min_{\boldsymbol{\theta}, \eta \geq 0, \lambda \in [\underline{\lambda}, \bar{\lambda}], \boldsymbol{\theta}^\tau, \tau \in \mathcal{T}} \max_{\mathbf{x}_2^\tau \in \Delta_{n_{\text{val}}, p}, \tau \in \mathcal{T}} \sum_{\tau \in \mathcal{T}} \mathcal{L}_{\text{DRO}}(D_{\text{val}}^\tau; \boldsymbol{\theta}^\tau, \mathbf{x}_2^\tau) \\ &\text{s.t.} \quad \boldsymbol{\theta}^\tau \in \arg\min_{\boldsymbol{\theta}'} \max_{\mathbf{y}_2^\tau \in \Delta_{n_{\text{train}}, 1}} \left\{ \begin{array}{l} \mathcal{L}_{\text{DRO}}(D_{\text{train}}^\tau; \boldsymbol{\theta}', \mathbf{y}_2^\tau) \\ -\frac{\eta}{2}\|\mathbf{y}_2^\tau - \frac{1}{n_{\text{train}}}\mathbf{1}\|^2 + \frac{\lambda}{2}\|\boldsymbol{\theta}' - \boldsymbol{\theta}\|^2 \end{array} \right\} \text{ for } \tau \in \mathcal{T}. \end{aligned} \tag{32}$$

In (32), we perform distributionally robust learning with tuning hyperparameters for each task $\tau \in \mathcal{T}$ just like (31). The main difference is that a meta $\boldsymbol{\theta}$ is introduced and an additional regularization term $\frac{\lambda}{2}\|\boldsymbol{\theta}' - \boldsymbol{\theta}\|^2$ is included in each lower-level problem to ensure $\boldsymbol{\theta}^\tau$ is not too far away from $\boldsymbol{\theta}$. It is still an instance of (1) with $\mathbf{x}_1 = (\boldsymbol{\theta}, \eta, \lambda)$, $\mathbf{y}_1 = (\boldsymbol{\theta}^\tau)_{\tau \in \mathcal{T}}$, $\mathbf{x}_2 = (\mathbf{x}_2^\tau)_{\tau \in \mathcal{T}}$ and $\mathbf{y}_2 = (\mathbf{y}_2^\tau)_{\tau \in \mathcal{T}}$.

B Proof of Lemma 4.2

Proof of Lemma 4.2. Suppose $(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2, \mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2)$ is an ϵ-primal-dual stationary point of (14). The first four conditions in Definition 3.2 are trivially implied by the two conditions in Definition 4.1. By (12), (15) and the fact that $p(\mathbf{x}_1, \mathbf{y}_1) - d(\mathbf{x}_1, \mathbf{y}_2) \geq 0$, we have

$$\begin{aligned} f_{\text{low}} + \rho\left[p(\mathbf{x}_1, \mathbf{y}_1) - d(\mathbf{x}_1, \mathbf{y}_2)\right] &\leq \max_{\mathbf{x}_2'} f(\mathbf{x}_1, \mathbf{x}_2', \mathbf{y}_1, \mathbf{y}_2) + \rho\left[p(\mathbf{x}_1, \mathbf{y}_1) - d(\mathbf{x}_1, \mathbf{y}_2)\right] \\ &= \max_{\mathbf{x}_2', \mathbf{z}_1', \mathbf{z}_2'} P_\rho(\mathbf{x}_1, \mathbf{x}_2', \mathbf{y}_1, \mathbf{y}_2, \mathbf{z}_1', \mathbf{z}_2'). \end{aligned}$$

This implies that

$$p(\mathbf{x}_1, \mathbf{y}_1) - d(\mathbf{x}_1, \mathbf{y}_2) \leq \rho^{-1} \left(\max_{\mathbf{x}_2', \mathbf{z}_1', \mathbf{z}_2'} P_\rho(\mathbf{x}_1, \mathbf{x}_2', \mathbf{y}_1, \mathbf{y}_2, \mathbf{z}_1', \mathbf{z}_2') - f_{\text{low}} \right),$$

which further implies $p(\mathbf{x}_1, \mathbf{y}_1) - d(\mathbf{x}_1, \mathbf{y}_2) \leq O(\epsilon)$ because of (20). Hence, $(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2)$ is a lifted $O(\epsilon)$-KKT point of (1). □

C A deterministic first-order method for nonconvex-concave minimax problem

Our goal is to solve (1) and (9) by solving its penalty formulation (14). Under Assumption 3.1, (14) is an instance of the following nonconvex-concave minimax problem

$$H^* := \min_x \max_y \{H(x, y) := h(x, y) + p(x) - q(y)\} \tag{33}$$

that satisfies the following assumptions.

Assumption C.1.

1. $\mathrm{dom} p$ and $\mathrm{dom} q$ are compact.
2. $h(x, y)$ is concave in y and $L_{\nabla h}$-smooth on $\mathrm{dom} p \times \mathrm{dom} q$.
3. p and q are proper closed convex, and their proximal operators can be computed exactly.

We introduce the following notations

$$D_p := \max\{\|u - v\| | u, v \in \mathrm{dom}\, p\}, \quad D_q := \max\{\|u - v\| | u, v \in \mathrm{dom}\, q\}, \tag{34}$$

$$H_{\text{low}} := \min\{H(x, y) | (x, y) \in \mathrm{dom} p \times \mathrm{dom} q\}. \tag{35}$$

We say $(x^\epsilon, y^\epsilon) \in \mathrm{dom} p \times \mathrm{dom} q$ is an *ϵ-primal-dual stationary point* of (33) if

$$\mathrm{dist}(0, \partial_x H(x^\epsilon, y^\epsilon)) \leq \epsilon \ \text{ and } \ \mathrm{dist}(0, \partial_y(-H(x^\epsilon, y^\epsilon))) \leq \epsilon. \tag{36}$$

To find an ϵ-primal-dual stationary point of (33), we directly use the state-of-the-art method, the inexact proximal-point method by Lu and Mei [53, Algorithm 2], which generates the next iterate $\left(x^{k+1}, y^{k+1}\right)$ from the current iterate $\left(x^k, y^k\right)$ by approximately solving a sequence of strongly-convex-strongly-concave (SCSC) minimax subproblems as follows

$$\left(x^{k+1}, y^{k+1}\right) \approx \arg\min_x \max_y \left\{H_k(x, y) := h_k(x, y) + p(x) - q(y)\right\}, \tag{37}$$

where

$$h_k(x, y) = h(x, y) - \epsilon \|y - y^0\|^2/(4D_q) + L_{\nabla h} \|x - x^k\|^2. \tag{38}$$

The subproblem (37) is an instance of the following SCSC minimax problem

$$\min_x \max_y \left\{\bar{h}(x, y) + p(x) - q(y)\right\}, \tag{39}$$

which satisfies the following assumptions.

Assumption C.2.

1. $\mathrm{dom} p$ and $\mathrm{dom} q$ are compact.
2. $\bar{h}(x, y)$ is σ_x-strongly convex in x and σ_y-strongly concave in y and $L_{\nabla \bar{h}}$-smooth on $\mathrm{dom} p \times \mathrm{dom} q$.
3. p and q are proper closed convex and their proximal operators can be computed exactly.

Subproblem (37) is an instance of (39) with

$$\bar{h}(x, y) \leftarrow h(x, y) - \epsilon \|y - y^0\|^2/(4D_q) + L_{\nabla h} \|x - x^k\|^2. \tag{40}$$

Algorithm 2 An optimal first-order method for (39): $\texttt{OptFOM}(\bar{\epsilon}, x^0, y^0, \bar{h}, p, q, L_{\nabla\bar{h}}, \sigma_x, \sigma_y)$

Input: targeted stationarity level $\bar{\epsilon} > 0$, initial solution $(x^0, y^0) \in \operatorname{dom} p \times \operatorname{dom} q$, $(\bar{h}, p, q)$ from (39), and $(L_{\nabla\bar{h}}, \sigma_x, \sigma_y)$ from Assumption C.2.

$z_f^0 = z^0 = -\sigma_x x^0$, $y_f^0 = y^0$, $\bar{\alpha} = \min\left\{1, \sqrt{8\sigma_y/\sigma_x}\right\}$, $\eta_z = \sigma_x/2$, $\eta_y = \min\{1/(2\sigma_y), 4/(\bar{\alpha}\sigma_x)\}$, $\zeta = \left(2\sqrt{5}(1 + 8L_{\nabla\bar{h}}/\sigma_x)\right)^{-1}$, $\gamma_x = \gamma_y = 8\sigma_x^{-1}$, and $\hat{\zeta} = \min\{\sigma_x, \sigma_y\}/L_{\nabla\bar{h}}^2$.

for $k = 0, 1, 2, \ldots$ **do**

  $(z_g^k, y_g^k) = \bar{\alpha}(z^k, y^k) + (1 - \bar{\alpha})(z_f^k, y_f^k)$.

  $(x^{k,-1}, y^{k,-1}) = (-\sigma_x^{-1} z_g^k, y_g^k)$.

  $x^{k,0} = \mathbf{prox}_{\zeta\gamma_x p}(x^{k,-1} - \zeta\gamma_x a_x^k(x^{k,-1}, y^{k,-1}))$.

  $y^{k,0} = \mathbf{prox}_{\zeta\gamma_y q}(y^{k,-1} - \zeta\gamma_y a_y^k(x^{k,-1}, y^{k,-1}))$.

  $b_x^{k,0} = \frac{1}{\zeta\gamma_x}(x^{k,-1} - \zeta\gamma_x a_x^k(x^{k,-1}, y^{k,-1}) - x^{k,0})$.

  $b_y^{k,0} = \frac{1}{\zeta\gamma_y}(y^{k,-1} - \zeta\gamma_y a_y^k(x^{k,-1}, y^{k,-1}) - y^{k,0})$.

  $t = 0$.

  while $\gamma_x\|a_x^k(x^{k,t}, y^{k,t}) + b_x^{k,t}\|^2 + \gamma_y\|a_y^k(x^{k,t}, y^{k,t}) + b_y^{k,t}\|^2 > \gamma_x^{-1}\|x^{k,t} - x^{k,-1}\|^2 + \gamma_y^{-1}\|y^{k,t} - y^{k,-1}\|^2$ **do**

    $\beta_t = 2/(t + 3)$

    $x^{k,t+1/2} = x^{k,t} + \beta_t(x^{k,0} - x^{k,t}) - \zeta\gamma_x(a_x^k(x^{k,t}, y^{k,t}) + b_x^{k,t})$.

    $y^{k,t+1/2} = y^{k,t} + \beta_t(y^{k,0} - y^{k,t}) - \zeta\gamma_y(a_y^k(x^{k,t}, y^{k,t}) + b_y^{k,t})$.

    $x^{k,t+1} = \mathbf{prox}_{\zeta\gamma_x p}(x^{k,t} + \beta_t(x^{k,0} - x^{k,t}) - \zeta\gamma_x a_x^k(x^{k,t+1/2}, y^{k,t+1/2}))$.

    $y^{k,t+1} = \mathbf{prox}_{\zeta\gamma_y q}(y^{k,t} + \beta_t(y^{k,0} - y^{k,t}) - \zeta\gamma_y a_y^k(x^{k,t+1/2}, y^{k,t+1/2}))$.

    $b_x^{k,t+1} = \frac{1}{\zeta\gamma_x}(x^{k,t} + \beta_t(x^{k,0} - x^{k,t}) - \zeta\gamma_x a_x^k(x^{k,t+1/2}, y^{k,t+1/2}) - x^{k,t+1})$.

    $b_y^{k,t+1} = \frac{1}{\zeta\gamma_y}(y^{k,t} + \beta_t(y^{k,0} - y^{k,t}) - \zeta\gamma_y a_y^k(x^{k,t+1/2}, y^{k,t+1/2}) - y^{k,t+1})$.

    $t \leftarrow t + 1$.

  end while

  $(x_f^{k+1}, y_f^{k+1}) = (x^{k,t}, y^{k,t})$.

  $(z_f^{k+1}, w_f^{k+1}) = (\nabla_x\hat{h}(x_f^{k+1}, y_f^{k+1}) + b_x^{k,t}, -\nabla_y\hat{h}(x_f^{k+1}, y_f^{k+1}) + b_y^{k,t})$.

  $z^{k+1} = z^k + \eta_z\sigma_x^{-1}(z_f^{k+1} - z^k) - \eta_z(x_f^{k+1} + \sigma_x^{-1} z_f^{k+1})$.

  $y^{k+1} = y^k + \eta_y\sigma_y(y_f^{k+1} - y^k) - \eta_y(w_f^{k+1} + \sigma_y y_f^{k+1})$.

  $x^{k+1} = -\sigma_x^{-1} z^{k+1}$.

  $\hat{x}^{k+1} = \mathbf{prox}_{\hat{\zeta}p}(x^{k+1} - \hat{\zeta}\nabla_x\bar{h}(x^{k+1}, y^{k+1}))$.

  $\hat{y}^{k+1} = \mathbf{prox}_{\hat{\zeta}q}(y^{k+1} + \hat{\zeta}\nabla_y\bar{h}(x^{k+1}, y^{k+1}))$.

  Terminate the algorithm and output $(\hat{x}^{k+1}, \hat{y}^{k+1})$ if

$$\|\hat{\zeta}^{-1}(x^{k+1} - \hat{x}^{k+1}, \hat{y}^{k+1} - y^{k+1}) - (\nabla\bar{h}(x^{k+1}, y^{k+1}) - \nabla\bar{h}(\hat{x}^{k+1}, \hat{y}^{k+1}))\| \le \bar{\epsilon}.$$

end for

Additionally, when (33) satisfies Assumption C.1, (37) is an instance of (39) satisfying Assumption (C.2) with

$$\begin{aligned} L_{\nabla\bar{h}} &\leftarrow L_{\nabla h} + \max\{2L_{\nabla h}, \epsilon/(2D_q)\}, \\ \sigma_x &\leftarrow L_{\nabla h}, \\ \sigma_y &\leftarrow \epsilon/(2D_q). \end{aligned} \tag{41}$$

where D_q is defined in (34).

The optimal algorithms for (39) under Assumption (C.2) include Kovalev and Gasnikov [38, Algorithm 4] and Lu and Mei [53, Algorithm 1], which are also used in [52, 54] as a subroutine for solving bilevel optimization. We present Lu and Mei [53, Algorithm 1] in Algorithm 2, where the notations

Algorithm 3 An inexact proximal point method [53, Algorithm 2] for (33)

Input: Outer targeted stationarity level $\epsilon > 0$, inner targeted stationarity level $\hat{\epsilon} > 0$, initial solution $(x^0, y^0) \in \operatorname{dom} p \times \operatorname{dom} q$, (h, p, q) from (33), $L_{\nabla h}$ from Assumption C.1, and D_q in (34).
for $k = 0, 1, 2, \ldots$ **do**
  Set $\bar{h}$ as in (40) and set $(L_{\nabla \bar{h}}, \sigma_x, \sigma_y)$ as in (41).
  $$(x^{k+1}, y^{k+1}) \leftarrow \texttt{OptFOM}\left(\frac{\hat{\epsilon}}{k+1}, x^k, y^k, \bar{h}, p, q, L_{\nabla \bar{h}}, \sigma_x, \sigma_y\right)$$
  if $\|x^{k+1} - x^k\| \le \epsilon/(4L_{\nabla h})$ **then**
    Return $(x^\epsilon, y^\epsilon) = (x^{k+1}, y^{k+1})$.
  end if
end for

$\hat{h}$, a_x^k and a_y^k are defined as follows:

$$\hat{h}(x, y) = \bar{h}(x, y) - \sigma_x \|x\|^2/2 + \sigma_y \|y\|^2/2, \tag{42}$$

$$a_x^k(x, y) = \nabla_x \hat{h}(x, y) + \sigma_x (x - \sigma_x^{-1} z_g^k)/2, \tag{43}$$

$$a_y^k(x, y) = -\nabla_y \hat{h}(x, y) + \sigma_y y + \sigma_x (y - y_g^k)/8, \tag{44}$$

and y_g^k and z_g^k are generated at iteration k of Algorithm 2. According to Lu and Mei [53, Theorem 1], under Assumption C.2, Algorithm 2 finds an $\bar{\epsilon}$-primal-dual stationary point of (39) with $O(\frac{L_{\nabla \bar{h}}}{\sqrt{\sigma_x \sigma_y}} \ln(1/\bar{\epsilon}))$ oracle complexity. Here, $\bar{\epsilon}$ is the first input of Algorithm 2.

Using Algorithm 2 to approximately solve subprpoblem (37) in each main iteration leads to the inexact proximal-point method in [53, Algorithm 2] for (33), which is presented in Algorithm 3. The oracle complexity of Algorithm 3 is established in Lu and Mei [53, Theorem 2] and is re-stated in Theorem C.3.

Theorem C.3. *Suppose Assumption C.1 holds. Let H, H^*, D_p, D_q and H_{low} be defined in* (33)*,* (34) *and* (35)*. Let*

$$\begin{aligned}
\alpha &= \min\{1, \sqrt{4\epsilon/(D_q L_{\nabla h})}\},\\
\delta &= (2 + \alpha^{-1}) L_{\nabla h} D_p^2 + \max\{\epsilon/D_q, \alpha L_{\nabla h}/4\} D_q^2,\\
K &= \left\lceil 16(\max_y H(x^0, y) - H^* + \epsilon D_q/4) L_{\nabla h} \epsilon^{-2} + 32\hat{\epsilon}^2 (1 + 4 D_q^2 L_{\nabla h}^2 \epsilon^{-2}) \epsilon^{-2} - 1) \right\rceil_+\\
C &= \frac{4 \max\{\frac{1}{2L_{\nabla h}}, \min\{\frac{D_q}{\epsilon}, \frac{4}{\alpha L_{\nabla h}}\}\} \Big(\delta + 2\alpha^{-1}(H^* - H_{low} + \epsilon D_q/4 + L_{\nabla h} D_p^2)\Big)}{[(3L_{\nabla h} + \epsilon/(2D_q))^2 / \min\{L_{\nabla h}, \epsilon/(2D_q)\} + 3L_{\nabla h} + \epsilon/(2D_q)]^{-2} \hat{\epsilon}^2}\\
N &= (\left\lceil 96\sqrt{2}(1 + (24 L_{\nabla h} + 4\epsilon/D_q) L_{\nabla h}^{-1}) \right\rceil + 2) \max\{2, \sqrt{D_q L_{\nabla h} \epsilon^{-1}}\}\\
&\qquad \times \Big((K+1) \times (\log(C))_+ + K + 1 + 2K \log(K+1)\Big).
\end{aligned}$$

Then Algorithm 3 outputs an ϵ-primal-dual stationary point (x^ϵ, y^ϵ) of (33) *(e.g.,* (36) *holds) in oracle complexity no more than N. Moreover, (x^ϵ, y^ϵ) satisfies*

$$\max_y H(x^\epsilon, y) \le \max_y H(x^0, y) + \epsilon D_q/4 + 2\hat{\epsilon}^2 (L_{\nabla h}^{-1} + 4 D_q^2 L_{\nabla h} \epsilon^{-2}). \tag{45}$$

Remark C.4. According to this theorem, Algorithm 3 finds an ϵ-primal-dual stationary point of (33) in $K = O(L_{\nabla h}\epsilon^{-2} + \hat{\epsilon}^2 L_{\nabla h}^2 \epsilon^{-4})$ iterations with a total oracle complexity of $N = \tilde{O}(\sqrt{L_{\nabla h}} \epsilon^{-0.5} K)$. If $\hat{\epsilon}$ is further set to be $O(\epsilon)$ and $L_{\nabla h} = O(1)$, the complexity becomes $N = \tilde{O}(\epsilon^{-2.5})$.

D Proof of Theorem 4.3

Proof of Theorem 4.3. We first verify that the deterministic case of Algorithm 1 is exactly Algorithm 3 applied to (14).

By (16), problem (14) is in the form of (33) with

$$\begin{aligned} x &\leftarrow (\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2), \\ y &\leftarrow (\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2), \\ h(x,y) &\leftarrow P_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2, \mathbf{z}_1, \mathbf{z}_2), \\ p(x) &\leftarrow P_2(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2), \\ q(y) &\leftarrow P_3(\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2). \end{aligned} \tag{46}$$

where P_1, P_2 and P_3 are defined in (17), (18) and (19). Moreover, according to Assumption 3.1, the smoothness parameters of (33) and the diameters of the domains, when specified to (14), are

$$\begin{aligned} L_{\nabla h} &\leftarrow L_{\nabla P_1} = L_{\nabla f} + 2\rho L_{\nabla \tilde{f}}, \\ D_p &\leftarrow D_1, \\ D_q &\leftarrow D_2, \end{aligned} \tag{47}$$

where D_1 and D_2 are defined in (10) and (11). Given the mappings in (46) and (47), the procedure in (37), when applied to (14), becomes exactly (21) if we set ρ_1 and ρ_2 as in Algorithm 1, i.e.,

$$\rho_1 = 2L_{\nabla P_1} = 2L_{\nabla f} + 4\rho L_{\nabla \tilde{f}}, \quad \rho_2 = \epsilon/(2D_2). \tag{48}$$

Similarly, subproblem (21) is an instance of (39) with

$$\begin{aligned} x &\leftarrow (\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2), \\ y &\leftarrow (\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2), \\ \bar{h}(x,y) &\leftarrow P_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2, \mathbf{z}_1, \mathbf{z}_2) \\ &\quad + \frac{\rho_1}{2}\left\|(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2) - (\mathbf{x}_1^{(k)}, \mathbf{y}_1^{(k)}, \mathbf{y}_2^{(k)})\right\|^2 - \frac{\rho_2}{2}\left\|(\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2) - (\mathbf{x}_2^{(0)}, \mathbf{z}_1^{(0)}, \mathbf{z}_2^{(0)})\right\|^2 \\ p(x) &\leftarrow P_2(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2) \\ q(y) &\leftarrow P_3(\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2), \end{aligned} \tag{49}$$

Additionally, when (1) satisfies Assumption 3.1 and ρ_1 and ρ_2 are set as in Algorithm 1 (i.e., as in (48)), (21) is an instance of (39) satisfying Assumption (C.2) with

$$\begin{aligned} L_{\nabla \bar{h}} &\leftarrow L_{\nabla f} + 2\rho L_{\nabla \tilde{f}} + \max\{\rho_1, \rho_2\}, \\ \sigma_x &\leftarrow L_{\nabla f} + 2\rho L_{\nabla \tilde{f}}, \\ \sigma_y &\leftarrow \epsilon/(2D_2), \end{aligned} \tag{50}$$

where D_2 is defined in (11). Since problem (14) is also an instance of (33) under the setting of (46) and (47), the deterministic caes of Algorithm 1 is exactly Algorithm 3 under the setting of (46) and (47). Therefore, we can directly apply Theorem C.3 and Lemma 4.2 to prove Theorem 4.3. While doing so, we just need to set $\hat{\epsilon} = \epsilon^{1.5}$ in Theorem C.3 according to Line 3 of Algorithm 1.

Let f^*, D_1, D_2 and f_{low} be defined in (13), (10), (11) and (12). Let

$$\begin{aligned} L_{\nabla P_1} &= L_{\nabla f_1} + 2\rho L_{\nabla \tilde{f}_1}, \\ \alpha_P &= \min\{1, \sqrt{4\epsilon/(D_2 L_{\nabla P_1})}\}, \\ \delta_P &= (2 + \alpha_P^{-1}) L_{\nabla P_1} D_1^2 + \max\{\epsilon/D_2, \alpha_P L_{\nabla P_1}/4\} D_2^2, \\ K_P &= \left\lceil \begin{array}{l} 16(\max_{\mathbf{x}_2} f(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)}) + 1 - f_{\text{low}} + \epsilon D_2/4) L_{\nabla P_1} \epsilon^{-2} \\ +32\epsilon^3(1 + 4(D_2^2 L_{\nabla P_1}^2 \epsilon^{-2})\epsilon^{-2} - 1) \end{array} \right\rceil_+ \\ C_P &= \frac{4\max\{\frac{1}{2L_{\nabla P_1}}, \min\{\frac{D_2}{\epsilon}, \frac{4}{\alpha_P L_{\nabla P_1}}\}\}\left(\delta_P + 2\alpha_P^{-1}(f^* - f_{\text{low}} + \epsilon D_2/4 + L_{\nabla P_1} D_1^2)\right)}{[(3L_{\nabla P_1} + \epsilon/(2D_2))^2/\min\{L_{\nabla P_1}, \epsilon/(2D_2)\} + 3L_{\nabla P_1} + \epsilon/(2D_2)]^{-2}\epsilon^3} \\ N_P &= (\lceil 96\sqrt{2}(1 + (24L_{\nabla P_1} + 4\epsilon/D_2)L_{\nabla P_1}^{-1})\rceil + 2)\max\{2, \sqrt{D_2 L_{\nabla P_1}\epsilon^{-1}}\} \\ &\quad \times \Big((K_P + 1) \times (\log(C_P))_+ + K_P + 1 + 2K_P \log(K_P + 1)\Big). \end{aligned}$$

Note that $\max_y H(x^0, y)$ in Theorem C.3 becomes

$$\begin{aligned}
&\max_{\mathbf{x}_2,\mathbf{z}_1,\mathbf{z}_2} P_\rho(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)}, \mathbf{z}_1, \mathbf{z}_2) \\
=&\max_{\mathbf{x}_2} f(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)}) + \rho\left(p(\mathbf{x}_1^{(0)}, \mathbf{y}_1^{(0)}) - d(\mathbf{x}_1^{(0)}, \mathbf{y}_2^{(0)})\right) \\
\le&\max_{\mathbf{x}_2} f(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)}) + 1 \qquad (51)
\end{aligned}$$

because of the initial condition (22) and the fact that $\rho = \frac{1}{\epsilon}$ by Algorithm 1. Recall that $\hat{\epsilon}$ is set to be $\epsilon^{1.5}$ in Algorithm 1. Applying Theorem C.3 under this setting with $\hat{\epsilon} = \epsilon^{1.5}$, we can show that Algorithm 1 outputs an ϵ-primal-dual stationary point $((\mathbf{x}_1^\epsilon, \mathbf{y}_1^\epsilon, \mathbf{y}_2^\epsilon), (\mathbf{x}_2^\epsilon, \mathbf{z}_1^\epsilon, \mathbf{z}_2^\epsilon))$ of (14) in oracle complexity no more than N_P above. Since $L_{\nabla P_1} = O(\rho) = O(\epsilon^{-1})$, we have $K_P = O(\epsilon^{-3})$ and $N_P = \tilde{O}(\epsilon^{-4})$.

Moreover, (45) in Theorem C.3 also implies that $((\mathbf{x}_1^\epsilon, \mathbf{y}_1^\epsilon, \mathbf{y}_2^\epsilon), (\mathbf{x}_2^\epsilon, \mathbf{z}_1^\epsilon, \mathbf{z}_2^\epsilon))$ satisfies

$$\begin{aligned}
\max_{\mathbf{x}_2,\mathbf{z}_1,\mathbf{z}_2} P_\rho(\mathbf{x}_1^\epsilon, \mathbf{x}_2, \mathbf{y}_1^\epsilon, \mathbf{y}_2^\epsilon, \mathbf{z}_1, \mathbf{z}_2) \le &\max_{\mathbf{x}_2,\mathbf{z}_1,\mathbf{z}_2} P_\rho(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)}, \mathbf{z}_1, \mathbf{z}_2) \\
&+ \epsilon D_2/4 + 2\epsilon^3(L_{\nabla P_1}^{-1} + 4D_2^2 L_{\nabla P_1}\epsilon^{-2}).
\end{aligned}$$

Combining the inequality above with (51) gives

$$\begin{aligned}
&\rho^{-1}\left(\max_{\mathbf{x}_2,\mathbf{z}_1,\mathbf{z}_2} P_\rho(\mathbf{x}_1^\epsilon, \mathbf{x}_2, \mathbf{y}_1^\epsilon, \mathbf{y}_2^\epsilon, \mathbf{z}_1, \mathbf{z}_2) - f_{\text{low}}\right) \\
\le&\rho^{-1}\left(\max_{\mathbf{x}_2,\mathbf{z}_1,\mathbf{z}_2} P_\rho(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)}, \mathbf{z}_1, \mathbf{z}_2) - f_{\text{low}} + \epsilon D_2/4 + 2\epsilon^3(L_{\nabla P_1}^{-1} + 4D_2^2 L_{\nabla P_1}\epsilon^{-2})\right) \\
\le&\rho^{-1}\left(\max_{\mathbf{x}_2} f(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)}) + 1 - f_{\text{low}} + \epsilon D_2/4 + 2\epsilon^3(L_{\nabla P_1}^{-1} + 4D_2^2 L_{\nabla P_1}\epsilon^{-2})\right) \\
\le&O(\epsilon),
\end{aligned}$$

where the last inequality is because $L_{\nabla P_1} = O(\rho) = O(\epsilon^{-1})$. By Lemma 4.2, $(\mathbf{x}_1^\epsilon, \mathbf{y}_1^\epsilon, \mathbf{y}_2^\epsilon)$ is a lifted $O(\epsilon)$-KKT point of (1). □

Dependence of total oracle complexity on problem parameters. We further examine the dependence of the deterministic first-order oracle complexity on the problem parameters. Let

$$\Delta_0 := \max_{\mathbf{x}_2} f(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)}) - f_{\text{low}}$$

denote the initial objective gap relative to the lower bound f_{low}. Recall that $\rho = \epsilon^{-1}$ and $L_{\nabla P_1} = L_{\nabla f_1} + 2\epsilon^{-1}L_{\nabla \tilde{f}_1}$. By the definitions of K_P and N_P, we have

$$K_P = O\left(\frac{(\Delta_0 + \epsilon D_2)(L_{\nabla f_1} + \epsilon^{-1}L_{\nabla \tilde{f}_1})}{\epsilon^2}\right), \quad N_P = \tilde{O}\left(\sqrt{\frac{D_2(L_{\nabla f_1} + \epsilon^{-1}L_{\nabla \tilde{f}_1})}{\epsilon}}K_P\right).$$

These bounds explicitly characterize the dependence on the smoothness constants $L_{\nabla f_1}$ and $L_{\nabla \tilde{f}_1}$, as well as the diameter D_2.

Note that the above complexity bounds guarantee a lifted $O(\epsilon)$-KKT point of (1) satisfying all conditions in Definition 3.2, except that the last condition is guaranteed only up to a constant factor, namely,

$$p(\mathbf{x}_1, \mathbf{y}_1) - d(\mathbf{x}_1, \mathbf{y}_2) \le O(\epsilon).$$

To obtain the explicit bound $p(\mathbf{x}_1, \mathbf{y}_1) - d(\mathbf{x}_1, \mathbf{y}_2) \le \epsilon$, it suffices to choose $\rho = c_\rho/\epsilon$ with a sufficiently large constant $c_\rho > 0$. In particular, the last inequality in the proof of Theorem 4.3 shows that $c_\rho = \Theta(1 + \Delta_0)$ is sufficient. The corresponding oracle complexity bounds can then be obtained by replacing $\rho = 1/\epsilon$ to $\rho = c_\rho/\epsilon$ and retaining the dependence on c_ρ in $L_{\nabla P_1}$ and all other quantities involving ρ.

E Proof of technical lemmas in Section 5

E.1 Proof of Lemma 5.3

Proof. Let's consider the convex program

$$\min_{\mathbf{z}_1} \bar{f}(\mathbf{x}_1, \mathbf{z}_1) \qquad \text{s.t.} \quad \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{z}_1) \leq 0. \tag{52}$$

By Assumption 5.1.2, $\bar{f}(\mathbf{x}_1, \mathbf{z}_1)$ is convex in $\mathbf{z}_1$, $\bar{g}_i(\mathbf{x}_1, \mathbf{z}_1)$ is convex in $\mathbf{z}_1$. By Assumption 5.1.4, Slater holds with margin $G > 0$. Then the strong duality holds and there exists a Lagrange multiplier $\boldsymbol{\lambda}^* \in \mathbb{R}^l_+$ such that $(\mathbf{z}_1^*, \boldsymbol{\lambda}^*)$ is a saddle point of the Lagrangian $\bar{f}(\mathbf{x}_1, \mathbf{z}_1) + \boldsymbol{\lambda}^T \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{z}_1)$. That is

$$\bar{f}(\mathbf{x}_1, \mathbf{z}_1^*) + \boldsymbol{\lambda}^T \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{z}_1^*) \leq \bar{f}(\mathbf{x}_1, \mathbf{z}_1^*) + \boldsymbol{\lambda}^{*T} \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{z}_1^*) \leq \bar{f}(\mathbf{x}_1, \mathbf{z}_1) + \boldsymbol{\lambda}^{*T} \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{z}_1) \tag{53}$$

for any $\mathbf{x}_1 \in \mathcal{X}_1$, $\mathbf{z}_1 \in \mathcal{Y}_1$ and $\boldsymbol{\lambda} \in \mathbb{R}^l_+$. Due to complementary slackness, we have

$$\bar{f}(\mathbf{x}_1, \mathbf{z}_1^*) = \bar{f}(\mathbf{x}_1, \mathbf{z}_1^*) + \boldsymbol{\lambda}^{*T} \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{z}_1^*)$$

By Assumption 5.1.4, given any $\mathbf{x}_1 \in \mathcal{X}_1$, there exists $\hat{\mathbf{z}}_1 \in \mathcal{Y}_1$ satisfying $\bar{g}_i(\mathbf{x}_1, \hat{\mathbf{z}}_1) \leq -G$. Setting $\mathbf{z}_1 = \hat{\mathbf{z}}_1$ in the second inequality of (53), we have

$$\bar{f}(\mathbf{x}_1, \mathbf{z}_1^*) \leq \bar{f}(\mathbf{x}_1, \hat{\mathbf{z}}_1) + \boldsymbol{\lambda}^{*T} \bar{\mathbf{g}}(\mathbf{x}_1, \hat{\mathbf{z}}_1) = \bar{f}(\mathbf{x}_1, \hat{\mathbf{z}}_1) + \sum_{i=1}^{l} \lambda_i^* \bar{g}_i(\mathbf{x}_1, \hat{\mathbf{z}}_1) \leq \bar{f}(\mathbf{x}_1, \hat{\mathbf{z}}_1) - G \sum_{i=1}^{l} \lambda_i^*$$

$$\Rightarrow \sum_{i=1}^{l} \lambda_i^* \leq \frac{\bar{f}(\mathbf{x}_1, \hat{\mathbf{z}}_1) - \bar{f}(\mathbf{x}_1, \mathbf{z}_1^*)}{G}$$

Now, using Assumption 5.1.2 and (23), we have $\bar{f}(\mathbf{x}_1, \hat{\mathbf{z}}_1) - \bar{f}(\mathbf{x}_1, \mathbf{z}_1^*) \leq L_{\bar{f}} D_{\mathcal{Y}_1}$. Therefore,

$$\sum_{i=1}^{l} \lambda_i^* \leq \frac{L_{\bar{f}} D_{\mathcal{Y}_1}}{G} < B.$$

Since $\boldsymbol{\lambda}^* \geq 0$, this implies the componentwise $\lambda_i^* < B,\ \forall i$. Hence, $\boldsymbol{\lambda}^* \in [0, B)^l$.

We now prove that a solution to (52) corresponds to a saddle point of $\tilde{f}(\mathbf{x}_1, \mathbf{z}_1, \mathbf{z}_2)$ defined in (7).

"⇒": Let $\mathbf{y}_1$ be an optimal solution to the lower-level problem (52). As proven above, there exists an optimal dual multiplier $\boldsymbol{\lambda}^* \geq 0$ such that $(\mathbf{y}_1, \boldsymbol{\lambda}^*)$ forms a global saddle point for the standard Lagrangian (53) and the optimal multiplier satisfies the bound $\boldsymbol{\lambda}^* \in [0, B)^l$.

Let $\mathbf{y}_2 = \boldsymbol{\lambda}^*$. We must prove that $(\mathbf{y}_1, \mathbf{y}_2)$ is a saddle point of the specific minimax objective $\tilde{f}(\mathbf{x}_1, \mathbf{z}_1, \mathbf{z}_2) = \overline{f}(\mathbf{x}_1, \mathbf{z}_1) + \mathbf{z}_2^T \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{z}_1) - \mathbf{1}_{[0,B]^l}(\mathbf{z}_2)$.

1. **Maximization over the dual variable ($\mathbf{z}_2$):** From the first inequality of (53), we have $\overline{f}(\mathbf{x}_1, \mathbf{y}_1) + \mathbf{z}_2^T \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{y}_1) \leq \overline{f}(\mathbf{x}_1, \mathbf{y}_1) + (\boldsymbol{\lambda}^*)^T \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{y}_1)$ for all $\mathbf{z}_2 \geq 0$. This inequality trivially holds for any $\mathbf{z}_2$ restricted to the subset $[0, B]^l$. For any $\mathbf{z}_2 \notin [0, B]^l$, the indicator function $-\mathbf{1}_{[0,B]^l}(\mathbf{z}_2)$ evaluates to $-\infty$, making the inequality $\tilde{f}(\mathbf{x}_1, \mathbf{y}_1,\ \mathbf{z}_2) \leq \tilde{f}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2)$ strictly and trivially true.
2. **Minimization over the primal variable ($\mathbf{z}_1$):** From the second inequality of (53), we have $\overline{f}(\mathbf{x}_1, \mathbf{y}_1) + (\boldsymbol{\lambda}^*)^T \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{y}_1) \leq \overline{f}(\mathbf{x}_1, \mathbf{z}_1) + (\boldsymbol{\lambda}^*)^T \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{z}_1)$ for all $\mathbf{z}_1 \in \mathcal{Y}_1$. Since $\mathbf{y}_2 = \boldsymbol{\lambda}^* \in [0, B]^l$, the indicator penalty is zero ($-\mathbf{1}_{[0,B]^l}(\mathbf{y}_2) = 0$). Adding this zero penalty to both sides yields exactly $\tilde{f}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2) \leq \tilde{f}(\mathbf{x}_1, \mathbf{z}_1, \mathbf{y}_2)$ for all $\mathbf{z}_1 \in \mathcal{Y}_1$.

Since both inequalities hold for all $\mathbf{z}_1 \in \mathcal{Y}_1$ and all $\mathbf{z}_2$, the pair $(\mathbf{y}_1, \mathbf{y}_2)$ is exactly a saddle point of $\min_{\mathbf{z}_1} \max_{\mathbf{z}_2} \tilde{f}(\mathbf{x}_1, \mathbf{z}_1, \mathbf{z}_2)$.

"⇐": Conversely, suppose there exists $\mathbf{y}_2 \in [0, B]^l$ such that $(\mathbf{y}_1, \mathbf{y}_2)$ is a saddle point of $\min_{\mathbf{z}_1} \max_{\mathbf{z}_2} \tilde{f}(\mathbf{x}_1, \mathbf{z}_1, \mathbf{z}_2)$, and equivalently a saddle point of $\min_{\mathbf{z}_1} \max_{\mathbf{z}_2 \in [0,B]^l} \{\overline{f}(\mathbf{x}_1, \mathbf{z}_1) + \mathbf{z}_2^T \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{z}_1)\}$. The saddle point definition imposes two inequalities:

1. **Maximization over the dual variable ($\mathbf{z}_2$):** We have $\overline{f}(\mathbf{x}_1, \mathbf{y}_1) + \mathbf{z}_2^T \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{y}_1) \leq \overline{f}(\mathbf{x}_1, \mathbf{y}_1) + \mathbf{y}_2^T \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{y}_1)$ for all $\mathbf{z}_2 \in [0, B]^l$. For this to hold, $\mathbf{y}_2$ must satisfy a component-wise maximization rule: if $\bar{\mathbf{g}}_i(\mathbf{x}_1, \mathbf{y}_1) > 0$, it forces $y_{2,i} = B$; if $\bar{\mathbf{g}}_i(\mathbf{x}_1, \mathbf{y}_1) < 0$, it forces $y_{2,i} = 0$.

2. **Minimization over the primal variable ($\mathbf{z}_1$):** We have $\overline{f}(\mathbf{x}_1, \mathbf{y}_1) + \mathbf{y}_2^T \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{y}_1) \leq \overline{f}(\mathbf{x}_1, \mathbf{z}_1) + \mathbf{y}_2^T \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{z}_1)$ for all $\mathbf{z}_1 \in \mathcal{Y}_1$.

To prove primal feasibility, suppose for contradiction that $\bar{\mathbf{g}}_j(\mathbf{x}_1, \mathbf{y}_1) > 0$ for some index j. By the maximization rule, this forces $y_{2,j} = B$, which implies $\mathbf{y}_2^T \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{y}_1) \geq 0$. Evaluating the minimization inequality at the Slater point $\hat{\mathbf{z}}_1$ (where $\bar{\mathbf{g}}_i(\mathbf{x}_1, \hat{\mathbf{z}}_1) \leq -G$ for all i) yields:

$$\overline{f}(\mathbf{x}_1, \mathbf{y}_1) + \mathbf{y}_2^T \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{y}_1) \leq \overline{f}(\mathbf{x}_1, \hat{\mathbf{z}}_1) + \mathbf{y}_2^T \bar{\mathbf{g}}(\mathbf{x}_1, \hat{\mathbf{z}}_1)$$

Rearranging and utilizing the Lipschitz bound $\overline{f}(\mathbf{x}_1, \hat{\mathbf{z}}_1) - \overline{f}(\mathbf{x}_1, \mathbf{y}_1) \leq L_{\overline{f}} D_{\mathcal{Y}_1}$, we obtain:

$$-\mathbf{y}_2^T \bar{\mathbf{g}}(\mathbf{x}_1, \hat{\mathbf{z}}_1) \leq \overline{f}(\mathbf{x}_1, \hat{\mathbf{z}}_1) - \overline{f}(\mathbf{x}_1, \mathbf{y}_1) - \mathbf{y}_2^T \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{y}_1) \leq L_{\overline{f}} D_{\mathcal{Y}_1}$$

Since $y_{2,j} = B$ and $\mathbf{y}_2 \geq 0$, we can strictly bound the left side: $-\mathbf{y}_2^T \bar{\mathbf{g}}(\mathbf{x}_1, \hat{\mathbf{z}}_1) \geq G \sum_{i=1}^{l} y_{2,i} \geq G \cdot B$. This gives $G \cdot B \leq L_{\overline{f}} D_{\mathcal{Y}_1}$, or $B \leq \frac{L_{\overline{f}} D_{\mathcal{Y}_1}}{G}$, which strictly contradicts our chosen bound $B = \frac{2 L_{\overline{f}} D_{\mathcal{Y}_1}}{G}$.

Therefore, no constraint can be violated, guaranteeing primal feasibility ($\bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{y}_1) \leq 0$). Because all constraints are feasible, the maximization rule forces $y_{2,i} = 0$ whenever $\bar{\mathbf{g}}_i(\mathbf{x}_1, \mathbf{y}_1) < 0$, which establishes exact complementary slackness ($\mathbf{y}_2^T \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{y}_1) = 0$).

Substituting this complementary slackness equality into the minimization condition simplifies it to $\overline{f}(\mathbf{x}_1, \mathbf{y}_1) \leq \overline{f}(\mathbf{x}_1, \mathbf{z}_1)$ for all feasible $\mathbf{z}_1$ (where $\mathbf{y}_2^T \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{z}_1) \leq 0$). Hence, $\mathbf{y}_1$ is an optimal solution to the lower-level problem. □

E.2 Proof of Lemma 5.4

Proof of Lemma 5.4. Note that (6) is an instance of (1) without $\mathbf{x}_2$. Suppose $(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2) \in \mathcal{X}_1 \times \mathcal{Y}_1 \times [0, B]^l$ is a lifted ϵ-KKT solution of problem (6) under Definition 3.2. There must exist $\rho \geq 0$ and $(\mathbf{z}_1, \mathbf{z}_2) \in \mathcal{Y}_1 \times [0, B]^l$ such that the following inequalities hold.

$$\operatorname{dist}\left(\mathbf{0}, \partial_{(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2)} \left[f(\mathbf{x}_1, \mathbf{y}_1) + \rho \begin{pmatrix} \bar{f}(\mathbf{x}_1, \mathbf{y}_1) + \mathbf{z}_2^\top \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{y}_1) - \mathbf{1}_{[0,B]^l}(\mathbf{z}_2) \\ -\bar{f}(\mathbf{x}_1, \mathbf{z}_1) - \mathbf{y}_2^\top \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{z}_1) + \mathbf{1}_{[0,B]^l}(\mathbf{y}_2) \end{pmatrix} \right] \right) \leq \epsilon, \quad (54)$$

$$\operatorname{dist}\left(\mathbf{0}, \rho \partial_{\mathbf{z}_1} \left[\bar{f}(\mathbf{x}_1, \mathbf{z}_1) + \mathbf{y}_2^\top \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{z}_1) \right] \right) \leq \epsilon, \quad (55)$$

$$\operatorname{dist}\left(\mathbf{0}, \rho \partial_{\mathbf{z}_2} \left[-\mathbf{z}_2^\top \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{y}_1) + \mathbf{1}_{[0,B]^l}(\mathbf{z}_2) \right] \right) \leq \epsilon, \quad (56)$$

$$\max_{\mathbf{z}_2 \in [0,B]^l} \left\{ \bar{f}(\mathbf{x}_1, \mathbf{y}_1) + \mathbf{z}_2^T \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{y}_1) \right\} - \min_{\mathbf{z}_1} \left\{ \bar{f}(\mathbf{x}_1, \mathbf{z}_1) + \mathbf{y}_2^\top \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{z}_1) \right\} \leq \epsilon. \quad (57)$$

Let

$$\bar{\boldsymbol{\lambda}} = \mathbf{y}_2 \text{ and } \boldsymbol{\lambda} = \rho \mathbf{z}_2.$$

Then (55) can be written as

$$\operatorname{dist}\left(\mathbf{0}, \rho \partial_{\mathbf{z}_1} \left[\bar{f}(\mathbf{x}_1, \mathbf{z}_1) + \bar{\boldsymbol{\lambda}}^T \bar{g}(\mathbf{x}_1, \mathbf{z}_1) \right] \right) \leq \epsilon. \quad (58)$$

It is easy to see that (54) implies

$$\operatorname{dist}\left(\mathbf{0}, \partial_{(\mathbf{x}_1, \mathbf{y}_1)} \left[f(\mathbf{x}_1, \mathbf{y}_1) + \rho\big(\bar{f}(\mathbf{x}_1, \mathbf{y}_1) - \bar{f}(\mathbf{x}_1, \mathbf{z}_1) - \bar{\boldsymbol{\lambda}}^T \bar{g}(\mathbf{x}_1, \mathbf{z}_1)\big) + \boldsymbol{\lambda}^T \bar{g}(\mathbf{x}_1, \mathbf{y}_1) \right] \right) \leq \epsilon \quad (59)$$

and

$$\operatorname{dist}\left(\mathbf{0}, \rho \partial_{\mathbf{y}_2} \left[-\mathbf{y}_2^\top \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{z}_1) + \mathbf{1}_{[0,B]^l}(\mathbf{y}_2) \right] \right) \leq \epsilon. \quad (60)$$

Equation (60) is equivalent to

$$\operatorname{dist}\left(\rho\bar{\mathbf{g}}(\mathbf{x}_1,\mathbf{z}_1), \partial_{\mathbf{y}_2}\mathbf{1}_{[0,B]^l}(\mathbf{y}_2)\right) \le \epsilon. \tag{61}$$

Let $\mathbf{y}_2 = (y_{2,1},\ldots,y_{2,l})^\top$ and we can split the indexes $\{1,\ldots,l\}$ into three subsets:

$$I_1 := \{i | y_{2,i} = 0\}, \quad I_2 := \{i | y_{2,i} \in (0,B)\}, \quad I_3 := \{i | y_{2,i} = B\}.$$

Let $\mathbf{y}_{2,I_1}$, $\mathbf{y}_{2,I_2}$ and $\mathbf{y}_{2,I_3}$ be the sub-vectors of $\mathbf{y}_2$ where the indexes of the components are in I_1, I_2 and I_3, respectively. We define $\mathbf{g}_{I_1}(\mathbf{x}_1,\mathbf{z}_1)$, $\mathbf{g}_{I_2}(\mathbf{x}_1,\mathbf{z}_1)$ and $\mathbf{g}_{I_3}(\mathbf{x}_1,\mathbf{z}_1)$ in a similar way. Because

$$\partial\mathbf{1}_{[0,B]}(y) = \begin{cases} (-\infty,0] & \text{if } y = 0, \\ \{0\} & \text{if } y \in (0,B), \\ [0,+\infty) & \text{if } y = B, \end{cases}$$

(61) implies

$$\rho\sqrt{\|[\mathbf{g}_{I_1}(\mathbf{x}_1,\mathbf{z}_1)]_+\|^2 + \|\mathbf{g}_{I_2}(\mathbf{x}_1,\mathbf{z}_1)\|^2 + \|[\mathbf{g}_{I_3}(\mathbf{x}_1,\mathbf{z}_1)]_-\|^2} \le \epsilon, \tag{62}$$

where $[\cdot]_+ := \max\{\cdot,0\}$ and $[\cdot]_- := \min\{\cdot,0\}$. Therefore,

$$\begin{aligned} -\rho\mathbf{y}_2^\top\bar{\mathbf{g}}(\mathbf{x}_1,\mathbf{z}_1) =& -\rho\mathbf{y}_{2,I_1}^\top\bar{\mathbf{g}}_{2,I_1}(\mathbf{x}_1,\mathbf{z}_1) - \rho\mathbf{y}_{2,I_2}^\top\bar{\mathbf{g}}_{2,I_2}(\mathbf{x}_1,\mathbf{z}_1) - \rho\mathbf{y}_{2,I_3}^\top\bar{\mathbf{g}}_{2,I_3}(\mathbf{x}_1,\mathbf{z}_1) \\ \le& -\rho\mathbf{y}_{2,I_1}^\top[\bar{\mathbf{g}}_{2,I_1}(\mathbf{x}_1,\mathbf{z}_1)]_+ - \rho\mathbf{y}_{2,I_2}^\top\bar{\mathbf{g}}_{2,I_2}(\mathbf{x}_1,\mathbf{z}_1) - \rho\mathbf{y}_{2,I_3}^\top[\bar{\mathbf{g}}_{2,I_3}(\mathbf{x}_1,\mathbf{z}_1)]_- \\ \le& \rho\|\mathbf{y}_2\|\sqrt{\|[\mathbf{g}_{I_1}(\mathbf{x}_1,\mathbf{z}_1)]_+\|^2 + \|\mathbf{g}_{I_2}(\mathbf{x}_1,\mathbf{z}_1)\|^2 + \|[\mathbf{g}_{I_3}(\mathbf{x}_1,\mathbf{z}_1)]_-\|^2} \\ \le& \epsilon\|\mathbf{y}_2\| \le \epsilon\|\mathbf{y}_2\|_1. \end{aligned} \tag{63}$$

We next prove that $I_3 = \emptyset$. Let $\hat{\mathbf{y}}_1$ be the solution that satisfies Assumption 5.2.4, so $\bar{g}_i(\mathbf{x}_1,\hat{\mathbf{y}}_1) \le -G$ for $i = 1,\ldots,l$. By (55), we have

$$\rho(\partial_{\mathbf{z}_1}\bar{f}(\mathbf{x}_1,\mathbf{z}_1) + \partial_{\mathbf{z}_1}\bar{\mathbf{g}}(\mathbf{x}_1,\mathbf{z}_1)\mathbf{y}_2)^\top(\mathbf{z}_1 - \hat{\mathbf{y}}_1) \le D_{\mathcal{Y}_1}\epsilon,$$

or equivalently,

$$\rho\partial_{\mathbf{z}_1}\bar{f}(\mathbf{x}_1,\mathbf{z}_1)^\top(\mathbf{z}_1 - \hat{\mathbf{y}}_1) \le \rho\left\langle\nabla_{\mathbf{z}_1}\bar{\mathbf{g}}(\mathbf{x}_1,\mathbf{z}_1)\mathbf{y}_2, \hat{\mathbf{y}}_1 - \mathbf{z}_1\right\rangle + D_{\mathcal{Y}_1}\epsilon,$$

which implies

$$\begin{aligned} -\rho L_{\bar{f}} D_{\mathcal{Y}_1} &\le \rho\left(\bar{f}(\mathbf{x}_1,\mathbf{z}_1) - \bar{f}(\mathbf{x}_1,\hat{\mathbf{y}}_1)\right) \le \rho\left\langle\partial_{\mathbf{z}_1}\bar{f}(\mathbf{x}_1,\mathbf{z}_1), \mathbf{z}_1 - \hat{\mathbf{y}}_1\right\rangle \\ &\le \rho\left\langle\nabla_{\mathbf{z}_1}\bar{\mathbf{g}}(\mathbf{x}_1,\mathbf{z}_1)\mathbf{y}_2, \hat{\mathbf{y}}_1 - \mathbf{z}_1\right\rangle + D_{\mathcal{Y}_1}\epsilon \\ &\le \rho\mathbf{y}_2^\top\left(\bar{\mathbf{g}}(\mathbf{x}_1,\hat{\mathbf{y}}_1) - \bar{\mathbf{g}}(\mathbf{x}_1,\mathbf{z}_1)\right) + D_{\mathcal{Y}_1}\epsilon \\ &\le -\rho G\|\mathbf{y}_2\|_1 + \epsilon\|\mathbf{y}_2\|_1 + D_{\mathcal{Y}_1}\epsilon, \end{aligned}$$

where the first inequality is because of the $L_{\bar{f}}$-Lipschitz continuity of $\bar{f}$, the second is by the convexity of $\bar{f}$ in its second argument, the fourth is by the convexity of $\bar{g}$ in its second argument, and the last inequality is because of (63) and the fact that $\bar{g}_i(\mathbf{x}_1,\hat{\mathbf{y}}_1) \le -G$ for $i = 1,\ldots,l$.

Suppose $I_3 \neq \emptyset$ so there is one coordinate of $\mathbf{y}_2$ equals B. We have $\|\mathbf{y}_2\|_1 \ge B$. Since $\epsilon \le \rho G$, the inequality above implies

$$B(\rho G - \epsilon) \le D_{\mathcal{Y}_1}\epsilon + \rho L_{\bar{f}} D_{\mathcal{Y}_1}.$$

Since $\epsilon \le \min\{\frac{\rho L_{\bar{f}}}{4}, \frac{\rho G}{4}\}$, by (23), it holds that

$$\frac{2L_{\bar{f}}D_{\mathcal{Y}_1}}{G} = B \le \frac{D_{\mathcal{Y}_1}\epsilon + \rho L_{\bar{f}}D_{\mathcal{Y}_1}}{(\rho G - \epsilon)} \le \frac{5L_{\bar{f}}D_{\mathcal{Y}_1}}{3G}.$$

This contradiction means $I_3 = \emptyset$.

Therefore, by (62), we have

$$\begin{aligned} |\rho\mathbf{y}_2^\top\bar{\mathbf{g}}(\mathbf{x}_1,\mathbf{z}_1)| =& |\rho\mathbf{y}_{2,I_1}^\top\bar{\mathbf{g}}_{2,I_1}(\mathbf{x}_1,\mathbf{z}_1) + \rho\mathbf{y}_{2,I_2}^\top\bar{\mathbf{g}}_{2,I_2}(\mathbf{x}_1,\mathbf{z}_1) + \rho\mathbf{y}_{2,I_3}^\top\bar{\mathbf{g}}_{2,I_3}(\mathbf{x}_1,\mathbf{z}_1)| \\ =& |\rho\mathbf{y}_{2,I_2}^\top\bar{\mathbf{g}}_{2,I_2}(\mathbf{x}_1,\mathbf{z}_1)| \\ \le& \rho\|\mathbf{y}_2\|\|\mathbf{g}_{I_2}(\mathbf{x}_1,\mathbf{z}_1)\| \\ \le& \epsilon\sqrt{l}B, \end{aligned}$$

which implies

$$|\mathbf{y}_2^\top \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{z}_1)| \leq \frac{\epsilon\sqrt{l}B}{\rho}. \tag{64}$$

Moreover, (62) and the fact that $I_3 = \emptyset$ imply that

$$\|[\bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{z}_1)]_+\| \leq \sqrt{\|[\mathbf{g}_{I_1}(\mathbf{x}_1, \mathbf{z}_1)]_+\|^2 + \|\mathbf{g}_{I_2}(\mathbf{x}_1, \mathbf{z}_1)\|^2} \leq \frac{\epsilon}{\rho}. \tag{65}$$

By the same argument as the one above, we can derive from (56) that

$$|\mathbf{z}_2^\top \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{y}_1)| \leq \frac{\epsilon\sqrt{l}B}{\rho}. \tag{66}$$

and

$$\|[\bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{y}_1)]_+\| \leq \frac{\epsilon}{\rho}. \tag{67}$$

Finally, let $\mathbf{z}_1^* \in \arg\min_{\mathbf{z}_1'} \left\{\bar{f}(\mathbf{x}_1, \mathbf{z}_1') | \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{z}_1') \leq 0\right\}$. We then have

$$\begin{aligned}
&\bar{f}(\mathbf{x}_1, \mathbf{y}_1) - \bar{f}^*(\mathbf{x}_1) \\
\leq& \left\{\bar{f}(\mathbf{x}_1, \mathbf{y}_1) + (\mathbf{0})^\top \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{y}_1)\right\} - \left\{\bar{f}(\mathbf{x}_1, \mathbf{z}_1^*) + \mathbf{y}_2^\top \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{z}_1^*)\right\} \\
\leq& \max_{\mathbf{z}_2' \in [0,B]} \left\{\bar{f}(\mathbf{x}_1, \mathbf{y}_1) + (\mathbf{z}_2')^\top \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{y}_1)\right\} - \min_{\mathbf{z}_1'} \left\{\bar{f}(\mathbf{x}_1, \mathbf{z}_1') + \mathbf{y}_2^\top \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{z}_1')\right\} \\
\leq& \epsilon
\end{aligned} \tag{68}$$

where the last inequality is due to (57).

The conclusion is thus proved as we have proved that (59), (58), (64), (65), (66), (67), and (68) hold. □

F Proof of Theorem 5.5

Proof. Since Theorem 5.5 is an application of Theorem 4.3 to a particular form of lower level objective, it remains to derive the Lipschitz constant $L_{\nabla P_1}$ for the penalty function. Based on the reformulation, we decompose P_1 as follows:

$$P_1 = f_1(\mathbf{x}_1, \mathbf{y}_1) + \rho \underbrace{\left(\overline{f}_1(\mathbf{x}_1, \mathbf{y}_1) + \mathbf{z}_2^\top \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{y}_1)\right)}_{=:H_1(\mathbf{x}_1, \mathbf{y}_1, \mathbf{z}_2)} - \rho \underbrace{\left(\overline{f}_1(\mathbf{x}_1, \mathbf{z}_1) + \mathbf{y}_2^\top \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{z}_1)\right)}_{=:H_2(\mathbf{x}_1, \mathbf{z}_1, \mathbf{y}_2)}$$

Let $\mathbf{u} = (\mathbf{x}_1, \mathbf{y}_1)$. The gradient of $H_1(\mathbf{u}, \mathbf{z}_2)$ is $\nabla H_1 = [\nabla \overline{f}_1(\mathbf{u}) + J_{\bar{\mathbf{g}}}(\mathbf{u})^\top \mathbf{z}_2,\ \bar{\mathbf{g}}(\mathbf{u})]^\top$. For any two points $(\mathbf{u}, \mathbf{z}_2)$ and $(\mathbf{u}', \mathbf{z}_2')$, we bound the difference in the $\mathbf{u}$-gradients:

$$\begin{aligned}
&\|\nabla_{\mathbf{u}} H_1(\mathbf{u}, \mathbf{z}_2) - \nabla_{\mathbf{u}} H_1(\mathbf{u}', \mathbf{z}_2')\| \\
\leq& \|\nabla \overline{f}_1(\mathbf{u}) - \nabla \overline{f}_1(\mathbf{u}')\| + \|J_{\bar{\mathbf{g}}}(\mathbf{u})^\top \mathbf{z}_2 - J_{\bar{\mathbf{g}}}(\mathbf{u}')^\top \mathbf{z}_2'\| \\
\leq& L_{\nabla \overline{f}_1} \|\mathbf{u} - \mathbf{u}'\| + \|(J_{\bar{\mathbf{g}}}(\mathbf{u}) - J_{\bar{\mathbf{g}}}(\mathbf{u}'))^\top \mathbf{z}_2\| + \|J_{\bar{\mathbf{g}}}(\mathbf{u}')^\top (\mathbf{z}_2 - \mathbf{z}_2')\| \\
\leq& (L_{\nabla \overline{f}_1} + B\sqrt{l} L_{\nabla \bar{\mathbf{g}}}) \|\mathbf{u} - \mathbf{u}'\| + L_{\bar{\mathbf{g}}} \|\mathbf{z}_2 - \mathbf{z}_2'\|
\end{aligned}$$

where we utilized the constraint bounds $\|\mathbf{z}_2\| \leq B\sqrt{l}$ (since $\mathbf{z}_2 \in [0, B]^l$), the $L_{\nabla \bar{\mathbf{g}}}$-smoothness and the $L_{\bar{\mathbf{g}}}$-Lipschitz continuity ($\|J_{\bar{\mathbf{g}}}(\mathbf{u})\| \leq L_{\bar{\mathbf{g}}}$) of the constraint mapping $\bar{\mathbf{g}}$ with respect to operator norm.

For the dual variable gradient, we have:

$$\|\nabla_{\mathbf{z}_2} H_1(\mathbf{u}, \mathbf{z}_2) - \nabla_{\mathbf{z}_2} H_1(\mathbf{u}', \mathbf{z}_2')\| = \|\bar{\mathbf{g}}(\mathbf{u}) - \bar{\mathbf{g}}(\mathbf{u}')\| \leq L_{\bar{\mathbf{g}}} \|\mathbf{u} - \mathbf{u}'\|$$

Since

$$\begin{aligned}
&\|\nabla H_1(\mathbf{u}, \mathbf{z}_2) - \nabla H_1(\mathbf{u}', \mathbf{z}_2')\| \\
=& \sqrt{\|\nabla_{\mathbf{u}} H_1(\mathbf{u}, \mathbf{z}_2) - \nabla_{\mathbf{u}} H_1(\mathbf{u}', \mathbf{z}_2')\|^2 + \|\nabla_{\mathbf{z}_2} H_1(\mathbf{u}, \mathbf{z}_2) - \nabla_{\mathbf{z}_2} H_1(\mathbf{u}', \mathbf{z}_2')\|^2} \\
\leq& \|\nabla_{\mathbf{u}} H_1(\mathbf{u}, \mathbf{z}_2) - \nabla_{\mathbf{u}} H_1(\mathbf{u}', \mathbf{z}_2')\| + \|\nabla_{\mathbf{z}_2} H_1(\mathbf{u}, \mathbf{z}_2) - \nabla_{\mathbf{z}_2} H_1(\mathbf{u}', \mathbf{z}_2')\|,
\end{aligned}$$

summing the above bounds provides the total Lipschitz constant for H_1:

$$L_{\nabla H_1} \le L_{\nabla \bar{f}_1} + B\sqrt{l} L_{\nabla \bar{\mathbf{g}}} + 2L_{\bar{\mathbf{g}}}.$$

By structural symmetry, because the dual variable $\mathbf{y}_2$ is constrained to the exact same box $[0, B]^l$, the gradient of $H_2(\mathbf{x}_1, \mathbf{z}_1, \mathbf{y}_2)$ shares is $L_{\nabla H_2}$-Lipschitz continuous with $L_{\nabla H_2} = L_{\nabla H_1}$.

Combining $L_{\nabla H_1}$ and $L_{\nabla H_2}$ with the $L_{\nabla f_1}$-smoothness of the upper-level objective f_1 yields the final overall bound:

$$L_{\nabla P_1} \le L_{\nabla f_1} + 2\rho(L_{\nabla \bar{f}_1} + B\sqrt{l} L_{\nabla \bar{\mathbf{g}}} + 2L_{\bar{\mathbf{g}}}).$$

We define the following quantities, which are all finite according to Assumption 5.1.

$$D_1 := \max\{\|(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2) - (\mathbf{x}_1', \mathbf{y}_1', \mathbf{y}_2')\| \mid (\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2), (\mathbf{x}_1', \mathbf{y}_1', \mathbf{y}_2') \in \mathcal{X}_1 \times \mathcal{Y}_1 \times [0, B]^l\}, \tag{69}$$

$$D_2 := \max\{\|(\mathbf{z}_1, \mathbf{z}_2) - (\mathbf{z}_1', \mathbf{z}_2')\| \mid (\mathbf{z}_1, \mathbf{z}_2), (\mathbf{z}_1', \mathbf{z}_2') \in \mathcal{Y}_1 \times [0, B]^l\}, \tag{70}$$

$$f_{\text{low}} := \min\{f(\mathbf{x}_1, \mathbf{y}_1) \mid (\mathbf{x}_1, \mathbf{y}_1) \in \mathcal{X}_1 \times \mathcal{Y}_1\},. \tag{71}$$

Then we can apply Theorem 4.3 on (6) and (7). Let D_1, D_2 and f_{low} be defined in (69), (70) and (71). Let

$$\begin{aligned}
f^* &= \min_{\mathbf{x}_1, \mathbf{y}_1} \left\{ f(\mathbf{x}_1, \mathbf{y}_1) + \rho \left[\bar{f}(\mathbf{x}_1, \mathbf{y}_1) + \max_{\mathbf{z}_2 \in [0,B]} \mathbf{z}_2^\top \bar{\mathbf{g}}(\mathbf{x}_1, \mathbf{y}_1) - \bar{f}^*(\mathbf{x}_1) \right] \right\}, \\
L_{\nabla P_1} &= L_{\nabla f_1} + 2\rho(L_{\nabla \bar{f}_1} + B\sqrt{l} L_{\nabla \bar{g}} + 2L_{\bar{g}}), \\
\alpha_c &= \min\{1, \sqrt{4\epsilon/(D_2 L_{\nabla P_1})}\}, \\
\delta_c &= (2 + \alpha_c^{-1}) L_{\nabla P_1} D_1^2 + \max\{\epsilon/D_2, \alpha_c L_{\nabla P_1}/4\} D_2^2, \\
K_c &= \left\lceil \begin{array}{l} 16(f(\mathbf{x}_1^{(0)}, \mathbf{y}_1^{(0)}) + 1 - f_{\text{low}} + \epsilon D_2/4) L_{\nabla P_1} \epsilon^{-2} \\ +32\epsilon^3(1 + 4(D_2^2 L_{\nabla P_1}^2 \epsilon^{-2})\epsilon^{-2} - 1) \end{array} \right\rceil_+ \\
C_c &= \frac{4 \max\{\frac{1}{2L_{\nabla P_1}}, \min\{\frac{D_2}{\epsilon}, \frac{4}{\alpha_c L_{\nabla P_1}}\}\} \Big(\delta_c + 2\alpha_c^{-1}(f^* - f_{\text{low}} + \epsilon D_2/4 + L_{\nabla P_1} D_1^2) \Big)}{[(3L_{\nabla P_1} + \epsilon/(2D_2))^2 / \min\{L_{\nabla P_1}, \epsilon/(2D_2)\} + 3L_{\nabla P_1} + \epsilon/(2D_2)]^{-2} \epsilon^3} \\
N_c &= (\lceil 96\sqrt{2}(1 + (24L_{\nabla P_1} + 4\epsilon/D_2) L_{\nabla P_1}^{-1}) \rceil + 2)\{2, \sqrt{D_2 L_{\nabla P_1} \epsilon^{-1}}\} \\
&\qquad \times \Big((K_c + 1) \times (\log(C_c))_+ + K_c + 1 + 2K_c \log(K_c + 1) \Big).
\end{aligned}$$

Then by Theorem 4.3 and Lemma 5.4, Algorithm 1 outputs a lifted $O(\epsilon)$-KKT point $(\mathbf{x}_1^\epsilon, \mathbf{y}_1^\epsilon)$ of (4) in oracle complexity no more than N_c. Here, $L_{\nabla P_1} = O(\rho) = O(\epsilon^{-1})$, $K_c = O(\epsilon^{-3})$, and $N_c = \tilde{O}(\epsilon^{-4})$. □

Dependence of total oracle complexity on problem parameters. We further examine the dependence of the deterministic first-order oracle complexity on the problem parameters in the constrained bilevel setting. Let

$$\Delta_0 := f(\mathbf{x}_1^{(0)}, \mathbf{y}_1^{(0)}) - f_{\text{low}}$$

denote the initial objective gap relative to f_{low}, and define

$$\widetilde{L} := L_{\nabla \bar{f}_1} + B\sqrt{l}\, L_{\nabla \bar{g}} + 2L_{\bar{g}},$$

which can be viewed as an overall smoothness parameter of the lower-level constrained problem. With the choice $\rho = \epsilon^{-1}$, we have $L_{\nabla P_1} = L_{\nabla f_1} + 2\epsilon^{-1}\widetilde{L}$. By the definition of K_c and N_c above, we obtain

$$K_c = O\left(\frac{(\Delta_0 + \epsilon D_2)(L_{\nabla f_1} + \epsilon^{-1}\widetilde{L})}{\epsilon^2} \right), \quad N_c = \tilde{O}\left(\sqrt{\frac{D_2(L_{\nabla f_1} + \epsilon^{-1}\widetilde{L})}{\epsilon}} K_c \right).$$

These bounds explicitly characterize the dependence on the upper-level smoothness constant $L_{\nabla f_1}$.

We next examine the dependence on the number of lower-level constraints and the margin of Slater's condition, i.e., G. Since the dual feasible set is $[0, B]^l$, we have

$$D_2 \le D_{\mathcal{Y}_1} + B\sqrt{l}.$$

For simplicity, we next only keep the terms with leading-order dependence on ϵ, B and l, and the preceding bound implies

$$K_c = O\left(\frac{B\sqrt{l}}{\epsilon^3}\right), \quad N_c = \tilde{O}\left(\frac{B^2 l}{\epsilon^4}\right).$$

Furthermore, by (23), the bound B depends on the Slater margin G in the order of $O(G^{-1})$. Consequently, when the remaining problem parameters are fixed, the complexity's dependence on the Slater margin is $O(G^{-2})$ in the leading term.

G A stochastic first-order method for nonconvex-concave minimax problem

In this section, our goal is to solve (1) and (9) by solving its penalty formulation (14) under the stochastic setting. We still view (14) as an instance of (33) and, in addition to Assumption C.1, we assume the existence of the stochastic first-order oracle of h.

Assumption G.1. There exist a random variable ω and a stochastic gradient estimator of $\nabla h(x,y)$, which is a mapping of (x,y) and ω denoted by $\hat{\nabla} h(x,y;\omega)$, and satisfies

1. $\mathbb{E}[\hat{\nabla} h(x,y;\omega)] = \nabla h(x,y)$.
2. $\mathbb{E}[\|\hat{\nabla} h(x,y;\omega) - \nabla h(x,y)\|^2] \le \delta^2$.

Our goal here is to find a *nearly ϵ-primal-dual stationary point* of (33), which is a solution (x^ϵ, y^ϵ) such that there exists $(\hat{x}^\epsilon, \hat{y}^\epsilon)$ such that $\|(x^\epsilon, y^\epsilon) - (\hat{x}^\epsilon, \hat{y}^\epsilon)\| \le \epsilon$ and $(\hat{x}^\epsilon, \hat{y}^\epsilon)$ is a ϵ-primal-dual stationary point of (33) (defined in (36)). To find a nearly ϵ-primal-dual stationary point of (33), we adopt the inexact proximal-point method again, which generates the next iterate by approximately solving the subproblem (37) using a stochastic first-order method. Similarly to the deterministic case, this subproblem is an instance of (39) satisfying the following assumption in addition to Assumption C.2.

Assumption G.2. There exist a random variable ω and a stochastic gradient estimator of $\nabla \bar{h}(x,y)$, which is a mapping of (x,y) and ω denoted by $\hat{\nabla} \bar{h}(x,y;\omega)$, and satisfies

1. $\mathbb{E}[\hat{\nabla} \bar{h}(x,y;\omega)] = \nabla \bar{h}(x,y)$.
2. $\mathbb{E}[\|\hat{\nabla} \bar{h}(x,y;\omega) - \nabla \bar{h}(x,y)\|^2] \le \delta^2$.

Note that, in the case of (37) where $\bar{h}$ (40) holds, we are under the setting where

$$\hat{\nabla} \bar{h}(x,y;\omega) \leftarrow \hat{\nabla} h(x,y;\omega) + (2L_{\nabla h}(x - x^k), -\epsilon/(2D_q)(y - y^0)) \tag{72}$$

and $\hat{\nabla} \bar{h}(x,y;\omega)$ and $\hat{\nabla} h(x,y;\omega)$ in (72) have the same vriance bound δ^2.

To solve the generic problem (39) in the stochastic setting, one of the optimal algorithms is the SAPD method [77, 76] presented in Algorithm 4. According to Zhang et al. [77, Theorem 2.11] and inequalities (2.28) and (2.32) in its proof, to achieve the optimal complexity, the number of iterations T and the control parameters (τ, σ, θ) are set as follows.

$$T = \left\lceil 1 + \ln\left(\frac{6\sigma_x D_p^2 + 6\sigma_y D_q^2}{\hat{\epsilon}^2}\right) \max\left\{(1-\overline{\theta}_1)^{-1}, (1-\overline{\theta}_2)^{-1}, (1-\overline{\overline{\theta}}_1)^{-1}, (1-\overline{\overline{\theta}}_2)^{-1}\right\}\right\rceil. \tag{73}$$

$$\tau = \frac{1-\theta}{\sigma_x \theta}, \quad \sigma = \frac{1-\theta}{\sigma_y \theta}, \quad \theta = \max\{\overline{\theta}_1,\ \overline{\theta}_2,\ \overline{\overline{\theta}}_1,\ \overline{\overline{\theta}}_2\}, \tag{74}$$

where

$$\overline{\theta}_1 := 1 - \frac{\beta(L_{\nabla \bar{h}} + \sigma_x)\sigma_y}{4L_{\nabla \bar{h}}^2}\left(\sqrt{1 + \frac{8\sigma_x L_{\nabla \bar{h}}^2}{\beta\sigma_y(L_{\nabla \bar{h}} + \sigma_x)^2}} - 1\right), \quad \overline{\theta}_2 := 1 - \frac{(1-\beta)^2}{32}\frac{\sigma_y^2}{L_{\nabla \bar{h}}^2}\left(\sqrt{1 + \frac{64 L_{\nabla \bar{h}}^2}{(1-\beta)^2\sigma_y^2}} - 1\right). \tag{75}$$

Algorithm 4 A stochastic accelerated primal-pual (SAPD) algorithm for (39):$\texttt{SAPD}(\bar{\epsilon}, x^0, y^0, \bar{h}, p, q, L_{\nabla \bar{h}}, \sigma_x, \sigma_y, \delta^2)$

Input: targeted stationarity level $\bar{\epsilon} > 0$, initial solution $(x^0, y^0) \in \operatorname{dom} p \times \operatorname{dom} q$, $(\bar{h}, p, q)$ from (39), $(L_{\nabla \bar{h}}, \sigma_x, \sigma_y)$ from Assumption C.2, and δ^2 from Assumption G.1.
Set T, τ, σ, and θ as in (73), (74), (75) and (76).
$\tilde{q}_0 \leftarrow 0$
for $k = 0, 1, 2, ..., T-1$ **do**
  $s^k \leftarrow \hat{\nabla}_y \bar{h}(x^k, y^k; \omega_y^k) + \theta \tilde{q}_k$
  $y^{k+1} \leftarrow \mathbf{prox}_{\sigma q}(y^k + \sigma s_k)$
  $x^{k+1} \leftarrow \mathbf{prox}_{\tau p}(x^k - \tau \hat{\nabla}_x \bar{h}(x^k, y^{k+1}; \omega_x^k))$
  $q_{k+1} \leftarrow \hat{\nabla}_y \bar{h}(x^{k+1}, y^{k+1}; \omega_y^{k+1}) - \hat{\nabla}_y \bar{h}(x^k, y^k; \omega_y^k)$
end for
Output:(x^T, y^T)

Algorithm 5 A stochastic inexact proximal point method for (33)

Input: number of iterations $K > 0$, inner targeted stationarity level $\hat{\epsilon} > 0$, initial solution $(x^0, y^0) \in \operatorname{dom} p \times \operatorname{dom} q$, (h, p, q) from (33), $L_{\nabla h}$ from Assumption C.1, D_q in (34), and δ^2 from Assumption G.1.
for $k = 0, 1, 2, \ldots$ **do**
  Set $\bar{h}$ as in (40) and set $(L_{\nabla \bar{h}}, \sigma_x, \sigma_y)$ as in (41).
$$(x^{k+1}, y^{k+1}) \leftarrow \texttt{SAPD}\left(\hat{\epsilon}, x^k, y^k, \bar{h}, p, q, L_{\nabla \bar{h}}, \sigma_x, \sigma_y, \delta^2\right)$$
  if $k = K$ **then**
    Return $(x^\epsilon, y^\epsilon) = (x^{k'+1}, y^{k'+1})$, where k' is uniformaly randomly sampled from $\{0, 1, \ldots, K-1\}$.
  end if
end for

$$\bar{\bar{\theta}}_1 := \max\left\{0, 1 - \frac{1}{12\Xi^x(\beta)} \frac{\sigma_x}{\delta^2} \hat{\epsilon}^2\right\}, \quad \bar{\bar{\theta}}_2 := \max\left\{0, 1 - \frac{1}{12\Xi^y(\beta)} \frac{\sigma_y}{\delta^2} \hat{\epsilon}^2\right\}, \tag{76}$$

$$\Xi^x(\beta) := 1 + \Psi(\beta), \quad \Xi^y(\beta) := \frac{27 + 3\beta}{2} + \frac{\sigma_y}{\sigma_x} \Psi(\beta),$$
$$\Psi(\beta) := \min\left\{\sqrt{\frac{\beta}{2} \frac{\sigma_x}{\sigma_y}}, \frac{1-\beta}{4}\right\}, \quad \beta := \min\left\{\frac{1}{2}, \frac{\sigma_y}{\sigma_x}, \frac{\sigma_x}{\sigma_y}\right\}.$$

Using Algorithm 4 to approximately solve subprpoblem (37) in each main iteration leads to a variant of the SAPD+ algorithm [76], which is presented in Algorithm 5 for solving (33). The stochastic case of Algorithm 1 is the specification of Algorithm 5 when applied to (14).

The complexity of Algorithm 4 is established in Zhang et al. [77, 76] and Algorithm 5 is established in Zhang et al. [76]. However, the result in Zhang et al. [76] cannot be directly applied to our problem because their convergence theory is for finding a nearly ϵ-stationary point of the outer function, or more specifically, a point x such that $\|\nabla \phi_{2L_{\nabla h}}(x)\| \leq \epsilon$, where

$$\phi_{2L_{\nabla h}}(x) := \min_{x'} \max_{y} \left[H(x', y) + L_{\nabla h} \|x' - x\|^2\right] \tag{77}$$

and H is from (33). However, our goal is instead to find a nearly ϵ-primal-dual stationary point of (33). Given this difference, we prove the following theorem that characterizes the complexity of Algorithm 5 for finding a nearly ϵ-primal-dual stationary point. This theorem is not only the key result for establishing the complexity of finding a nearly lifted ϵ-KKT point of (1) but also an interesting result itself.

Theorem G.3. *Suppose Assumptions C.1 and G.1 hold. Let H, H^*, D_p, and D_q be defined in* (33), (34) *and* (35). *Let*

$$K = \left\lceil \left(\mathbb{E}\left[\max_y H(x^0, y)\right] - H^* + \epsilon D_q/4 \right) \hat{\epsilon}^{-2} \right\rceil,$$

$$C = \left\lceil 1 + \ln\left(\frac{6L_{\nabla h}D_p^2 + 3\epsilon D_q}{\hat{\epsilon}^2}\right) \left[\left(L_{\nabla h} + \max\left\{2L_{\nabla h}, \frac{\epsilon}{2D_q}\right\}\right) \left(\frac{2\sqrt{2D_q}}{\sqrt{L_{\nabla h}\epsilon}} + \frac{3}{L_{\nabla h}} + \frac{24D_q}{\epsilon}\right) + \frac{\delta^2}{\hat{\epsilon}^2}\left(\frac{21}{L_{\nabla h}} + \frac{360D_q}{\epsilon}\right)\right]\right\rceil,$$

$$N = KC.$$

Then Algorithm 5 terminates and outputs a solution (x^ϵ, y^ϵ) such that there exists a stochastic solution $(\hat{x}^\epsilon, \hat{y}^\epsilon)$ such that

$$\mathbb{E}\|(x^\epsilon, y^\epsilon) - (\hat{x}^\epsilon, \hat{y}^\epsilon)\| \le \max\left\{\frac{\hat{\epsilon}}{\sqrt{L_{\nabla h}}}, \frac{\hat{\epsilon}}{\sqrt{\epsilon/(2D_q)}}\right\} \tag{78}$$

and $(\hat{x}^\epsilon, \hat{y}^\epsilon)$ is a $\max\left\{\sqrt{2L_{\nabla h}}\hat{\epsilon}, \frac{\epsilon}{2}\right\}$-primal-dual stationary point of (33) *in expectation. As a consequence, Algorithm 5 finds a nearly $\max\left\{\sqrt{2L_{\nabla h}}\hat{\epsilon}, \frac{\epsilon}{2}, \frac{\hat{\epsilon}}{\sqrt{L_{\nabla h}}}, \frac{\hat{\epsilon}}{\sqrt{\epsilon/(2D_q)}}\right\}$-primal-dual stationary point (x^ϵ, y^ϵ) of* (33) *in expectation in complexity no more than N. Moreover, it holds that*

$$\mathbb{E}\left[\max_y H(\hat{x}^\epsilon, y)\right] \le 2\mathbb{E}\left[\max_y H(x^0, y)\right] - H^* + \epsilon D_q/2, \tag{79}$$

Proof. We first focus on the the kth main iteration. Let $(\hat{x}^k, \hat{y}^k)$ be the unique saddle point of (37). Recall that (x^{k+1}, y^{k+1}) is returned by Algorithm 4 initialized at (x^k, y^k) and after running for T iterations. The parameters τ, σ and θ in Algorithm 4 are set as in (74), (75) and (76) according to Zhang et al. [77, Theorem 2.11]. Therefore, by Zhang et al. [77, Theorem 2] and inequalities (2.27) and (2.28) in its proof in their paper, Algorithm 4 ensures

$$\mathbb{E}[\sigma_x\|x^{k+1} - \hat{x}^k\|^2 + \sigma_y\|y^{k+1} - \hat{y}^k\|^2] \le \hat{\epsilon}^2 \tag{80}$$

if

$$T \ge 1 + \ln\left(\frac{6\sigma_x\|x^k - \hat{x}^k\|^2 + 6\sigma_y\|y^k - \hat{y}^k\|^2}{\hat{\epsilon}^2}\right) \max\left\{(1-\overline{\theta}_1)^{-1}, (1-\overline{\theta}_2)^{-1}, (1-\overline{\overline{\theta}}_1)^{-1}, (1-\overline{\overline{\theta}}_2)^{-1}\right\}. \tag{81}$$

By (34), we have $\|x^k - \hat{x}^k\|^2 \le D_p^2$ and $\|y^k - \hat{y}^k\|^2 \le D_q^2$. Therefore, the choice of T in Algorithm 4 does satisfy (81), so (80) holds.

Note that

$$\begin{aligned}
(1-\overline{\theta}_1)^{-1} =& \frac{1}{2}\left(\frac{L_{\nabla\bar{h}}}{\sigma_x} + 1\right) + \sqrt{\frac{1}{4}\left(\frac{L_{\nabla\bar{h}}}{\sigma_x} + 1\right)^2 + \frac{2L_{\nabla\bar{h}}^2}{\beta\sigma_x\sigma_y}},\\
(1-\overline{\theta}_2)^{-1} =& \frac{1}{2} + \sqrt{\frac{1}{4} + \frac{16L_{\nabla\bar{h}}^2}{(1-\beta)^2\sigma_y^2}},\\
(1-\overline{\overline{\theta}}_1)^{-1} =& 12\Xi^x(\beta)\frac{\delta^2}{\sigma_x\hat{\epsilon}^2}, \qquad (1-\overline{\overline{\theta}}_2)^{-1} = 12\Xi^y(\beta)\frac{\delta^2}{\sigma_y\hat{\epsilon}^2}.
\end{aligned}$$

Using the facts that $\sqrt{a^2+b^2} \le a+b$, that $1/\min\{a,b\} \le \frac{1}{a} + \frac{1}{b}$, that $\max\{\sigma_x, \sigma_y\} \le L_{\nabla\bar{h}}$, and the definitions of β, $\Xi^x(\beta)$ and $\Xi^y(\beta)$, we have $\Xi^x(\beta) \le 1 + \sqrt{1/2}$, $\Xi^y(\beta) \le (27+3/2)/2 + 1/\sqrt{2}$,

$$\begin{aligned}
(1-\overline{\theta}_1)^{-1} \le& \frac{2L_{\nabla\bar{h}}}{\sqrt{\sigma_x\sigma_y}} + \frac{3L_{\nabla\bar{h}}}{\sigma_x} + \frac{3L_{\nabla\bar{h}}}{\sigma_y}, \qquad (1-\overline{\theta}_2)^{-1} \le \frac{9L_{\nabla\bar{h}}}{\sigma_y},\\
(1-\overline{\overline{\theta}}_1)^{-1} =& \frac{21\delta^2}{\sigma_x\hat{\epsilon}^2}, \qquad (1-\overline{\overline{\theta}}_2)^{-1} = \frac{180\delta^2}{\sigma_y\hat{\epsilon}^2}.
\end{aligned}$$

These results imply that

$$T \leq \left\lceil 1 + \ln\left(\frac{6\sigma_x D_p^2 + 6\sigma_y D_q^2}{\hat{\epsilon}^2}\right)\left(\frac{2L_{\nabla\bar{h}}}{\sqrt{\sigma_x\sigma_y}} + \frac{3L_{\nabla\bar{h}}}{\sigma_x} + \frac{12L_{\nabla\bar{h}}}{\sigma_y} + \frac{21\delta^2}{\sigma_x\hat{\epsilon}^2} + \frac{180\delta^2}{\sigma_y\hat{\epsilon}^2}\right)\right\rceil.$$

Since we are analyzing Algorithm 4 when it is applied (37) under the setting of (40) and (41), replacing $L_{\nabla\bar{h}}$, σ_x, and σ_y in the right-hand side above gives

$$\begin{aligned} T \leq \left\lceil 1 + \ln\left(\frac{6L_{\nabla h}D_p^2 + 3\epsilon D_q}{\hat{\epsilon}^2}\right)\right. & \left[\left(L_{\nabla h} + \max\left\{2L_{\nabla h}, \frac{\epsilon}{2D_q}\right\}\right)\left(\frac{2\sqrt{2D_q}}{\sqrt{L_{\nabla h}\epsilon}} + \frac{3}{L_{\nabla h}} + \frac{24D_q}{\epsilon}\right)\right. \\ & \left.\left. + \frac{\delta^2}{\hat{\epsilon}^2}\left(\frac{21}{L_{\nabla h}} + \frac{360D_q}{\epsilon}\right)\right]\right\rceil = C. \end{aligned} \tag{82}$$

This result shows that the complexity of each call of Algorithm 4 is bounded by C in each main iteration of Algorithm 5.

Nest, we will analyze the number of main iterations needed in Algorithm 5. Let $\phi^{\epsilon}_{2L_{\nabla h}}(x)$ be the Moreau envelope of $\max_y \left[H(x,y) - \epsilon\|y - y^0\|^2/(4D_q)\right]$, i.e.,

$$\phi^{\epsilon}_{2L_{\nabla h}}(x) := \min_{x'}\max_{y}\left[H(x',y) - \epsilon\|y - y^0\|^2/(4D_q) + L_{\nabla h}\|x' - x\|^2\right]. \tag{83}$$

We then have

$$\begin{aligned} & \phi^{\epsilon}_{2L_{\nabla h}}(x^{k+1}) \\ =& \min_{x}\max_{y}\left[H(x,y) - \epsilon\|y - y^0\|^2/(4D_q) + L_{\nabla h}\|x - x^{k+1}\|^2\right] \\ \leq& \max_{y}\left[H(\hat{x}^k,y) - \epsilon\|y - y^0\|^2/(4D_q)\right] + L_{\nabla h}\|\hat{x}^k - x^{k+1}\|^2 \\ =& \max_{y}\left[H(\hat{x}^k,y) - \epsilon\|y - y^0\|^2/(4D_q)\right] + L_{\nabla h}\|\hat{x}^k - x^k\|^2 - L_{\nabla h}\|\hat{x}^k - x^k\|^2 + L_{\nabla h}\|\hat{x}^k - x^{k+1}\|^2 \\ =& \phi^{\epsilon}_{2L_{\nabla h}}(x^k) - L_{\nabla h}\|\hat{x}^k - x^k\|^2 + L_{\nabla h}\|\hat{x}^k - x^{k+1}\|^2. \end{aligned}$$

Taking the expectation and summing the inequality above for $k = 0, 1, \ldots, K-1$ gives

$$\begin{aligned} \sum_{k=0}^{K-1} L_{\nabla h}\mathbb{E}\|\hat{x}^k - x^k\|^2 \leq& \mathbb{E}\left[\phi^{\epsilon}_{2L_{\nabla h}}(x^0) - \phi^{\epsilon}_{2L_{\nabla h}}(x^K)\right] + \sum_{k=0}^{K-1} L_{\nabla h}\mathbb{E}\|\hat{x}^k - x^{k+1}\|^2 \\ \leq& \mathbb{E}\left[\phi^{\epsilon}_{2L_{\nabla h}}(x^0) - \phi^{\epsilon}_{2L_{\nabla h}}(x^K)\right] + K\hat{\epsilon}^2 \end{aligned} \tag{84}$$

where the last inequality is because of (80) and the fact that $\sigma_x = L_{\nabla h}$. Moreover, by definitions, it is easy to show that

$$\phi^{\epsilon}_{2L_{\nabla h}}(x^0) \leq \max_y H(x^0, y), \quad \phi^{\epsilon}_{2L_{\nabla h}}(x^K) \geq H^* - \epsilon D_q/4.$$

Recall that k' is uniformly randomly sampled from $\{0, 1, \ldots, K-1\}$. The two inequalities above and (84) together imply

$$L^2_{\nabla h}\mathbb{E}\|\hat{x}^{k'} - x^{k'}\|^2 \leq \frac{L_{\nabla h}(\mathbb{E}[\max_y H(x^0,y)] - H^* + \epsilon D_q/4)}{K} + L_{\nabla h}\hat{\epsilon}^2 \leq 2L_{\nabla h}\hat{\epsilon}^2, \tag{85}$$

where the second inequality is from the definition of K.

Recall that Algorithm 5 returns $(x^{\epsilon}, y^{\epsilon}) = (x^{k'+1}, y^{k'+1})$, where k' is uniformaly randomly sampled from $\{0, 1, \ldots, K-1\}$. We will show that $(\hat{x}^{\epsilon}, \hat{y}^{\epsilon}) = (\hat{x}^{k'}, \hat{y}^{k'})$ will be the stochastic solution in the statement of the theorem.

By the optimal condition of $(\hat{x}^k, \hat{y}^k)$, it holds for any $k \geq 0$ that

$$\begin{aligned} \mathbf{0} \in& \partial p(\hat{x}^k) + \nabla_x h(\hat{x}^k, \hat{y}^k) + L_{\nabla h}(\hat{x}^k - x^k), \\ \mathbf{0} \in& \partial q(\hat{x}^k) - \nabla_y h(\hat{x}^k, \hat{y}^k) + \epsilon(\hat{y}^k - y^0)/(2D_q), \end{aligned}$$

By (85) and Jensen's inequality, the two inclusions above imply that

$$\mathbb{E}\text{dist}\left(\mathbf{0}, \partial p(\hat{x}^{k'}) + \nabla_x h(\hat{x}^{k'}, \hat{y}^{k'})\right) \leq L_{\nabla h}\mathbb{E}\|\hat{x}^{k'} - x^{k'}\| \leq \sqrt{2L_{\nabla h}}\hat{\epsilon}, \tag{86}$$

$$\mathbb{E}\text{dist}\left(\mathbf{0}, \partial q(\hat{x}^{k'}) - \nabla_y h(\hat{x}^{k'}, \hat{y}^{k'})\right) \leq \epsilon\mathbb{E}\|\hat{y}^{k'} - y^0\|/(2D_q) \leq \frac{\epsilon}{2}, \tag{87}$$

which means $(\hat{x}^\epsilon, \hat{y}^\epsilon) = (\hat{x}^{k'}, \hat{y}^{k'})$ is an $\max\left\{\sqrt{2L_{\nabla h}}\hat{\epsilon}, \frac{\epsilon}{2}\right\}$-primal-dual solution of (33) in expectation.

Additionally, by (80), Jensen's inequality and the facts that $\sigma_x = L_{\nabla h}$ and $\sigma_y = \epsilon/(2D_q)$, it also holds that

$$\begin{aligned}\mathbb{E}\|(x^\epsilon, y^\epsilon) - (\hat{x}^\epsilon, \hat{y}^\epsilon)\| =& \mathbb{E}\|(x^{k'+1}, y^{k'+1}) - (\hat{x}^{k'}, \hat{y}^{k'})\| \leq \sqrt{\mathbb{E}\|(x^{k'+1}, y^{k'+1}) - (\hat{x}^{k'}, \hat{y}^{k'})\|^2} \\ \leq& \sqrt{\frac{\hat{\epsilon}^2}{\min\{\sigma_x, \sigma_y\}}} \leq \max\left\{\frac{\hat{\epsilon}}{\sqrt{L_{\nabla h}}}, \frac{\hat{\epsilon}}{\sqrt{\epsilon/(2D_q)}}\right\}.\end{aligned} \tag{88}$$

Comparing the orders of the right-hand sides in (86), (87), and (88), we conclude by definition that $(x^\epsilon, y^\epsilon) = \left(x^{k'+1}, y^{k'+1}\right)$ is a nearly $\max\left\{\sqrt{2L_{\nabla h}}\hat{\epsilon}, \frac{\epsilon}{2}, \frac{\hat{\epsilon}}{\sqrt{L_{\nabla h}}}, \frac{\hat{\epsilon}}{\sqrt{\epsilon/(2D_q)}}\right\}$-primal-dual stationary point of (33) in expectation. It is found in complexity no more than $TK \leq CK = N$.

Next we prove the second conclusion. By the same argument we used to show (84), it is easy to see that (84) holds even if we replace K by $k = 0, 1, \ldots, K-1$. In fact, we have, for $k = 0, 1, \ldots, K-1$, that

$$0 \leq \sum_{l=0}^{k-1} L_{\nabla h}\mathbb{E}\|\hat{x}^l - x^l\|^2 \leq \mathbb{E}\left[\phi^\epsilon_{2L_{\nabla h}}(x^0) - \phi^\epsilon_{2L_{\nabla h}}(x^k)\right] + k\hat{\epsilon}^2.$$

which implies

$$\begin{aligned}\mathbb{E}\phi^\epsilon_{2L_{\nabla h}}(x^k) \leq& \phi^\epsilon_{2L_{\nabla h}}(x^0) + k\hat{\epsilon}^2 \leq \mathbb{E}[\max_y H(x^0, y)] + k\hat{\epsilon}^2 \\ \leq& \mathbb{E}[\max_y H(x^0, y)] + K\hat{\epsilon}^2 \leq 2\mathbb{E}[\max_y H(x^0, y)] - H^* + \epsilon D_q/4,\end{aligned} \tag{89}$$

for $k = 0, \ldots, K-1$, where the last inequality is from the definition of K. On the other hand, we have

$$\phi^\epsilon_{2L_{\nabla h}}(x^k) = \max_y \left[H(\hat{x}^k, y) - \epsilon\|y - \hat{y}^0\|^2/(4D_q) + L_{\nabla h}\|\hat{x}^k - x^k\|^2\right] \geq \max_y H(\hat{x}^k, y) - \epsilon D_q/4.$$

Applying the inequality above to (89) leads to

$$\mathbb{E}\left[\max_y H(\hat{x}^k, y)\right] \leq 2\mathbb{E}[\max_y H(x^0, y)] - H^* + \epsilon D_q/2,$$

for $k = 0, 1, \ldots, K-1$. Letting $k = k'$ above and taking expectation over k' leads to (79). □

Remark G.4. According to this theorem, when $L_{\nabla h} = O(1)$, Algorithm 5 finds a nearly $O\left(\max\left\{\epsilon, \frac{\hat{\epsilon}}{\sqrt{\epsilon}}\right\}\right)$-primal-dual stationary point of (33) in $K = O(\hat{\epsilon}^{-2})$ iterations with a total stochastic oracle complexity of $N = \tilde{O}((\epsilon^{-1} + \delta^2\epsilon^{-1}\hat{\epsilon}^{-2})K)$. If $\hat{\epsilon}$ is further set to be $\Theta\left(\epsilon^{1.5}\right)$ and $L_{\nabla h} = O(1)$, the complexity becomes $N = \tilde{O}((\epsilon^{-3} + \delta^2\epsilon^{-7}))$.

H Proof of Theorem 6.3

Proof of Theorem 6.3. Just like the deterministic case, subproblem (21) is solved by Algorithm 4 as an instance of (39) under the setting of (49) and (50). In this setting, the stochastic gradient of $\bar{h}$ is set as

$$\begin{aligned}\hat{\nabla}\bar{h}(x, y; \omega) \leftarrow& \hat{\nabla} f_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2; \omega) + \rho\big(\hat{\nabla}\tilde{f}_1(\mathbf{x}_1, \mathbf{y}_1, \mathbf{z}_2; \omega) - \hat{\nabla}\tilde{f}_1(\mathbf{x}_1, \mathbf{z}_1, \mathbf{y}_2; \omega)\big) \\ &+ \begin{pmatrix} 2L_{\nabla P_1}\left[(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2) - (\mathbf{x}_1^{(k)}, \mathbf{y}_1^{(k)}, \mathbf{y}_2^{(k)})\right], \\ -\frac{\epsilon}{D_2}\left[(\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2) - (\mathbf{x}_2^{(k)}, \mathbf{z}_1^{(k)}, \mathbf{z}_2^{(k)})\right] \end{pmatrix},\end{aligned} \tag{90}$$

where $L_{\nabla P_1} = L_{\nabla f} + 2\rho L_{\nabla \tilde{f}}$ and D_2 is defined in (11). Under Assumption 6.1, we have

$$\begin{aligned}
&\mathbb{E}\left\| \begin{array}{l} \hat{\nabla} f_1(\mathbf{x}_1,\mathbf{x}_2,\mathbf{y}_1,\mathbf{y}_2;\omega) + \rho\big(\hat{\nabla}\tilde{f}_1(\mathbf{x}_1,\mathbf{y}_1,\mathbf{z}_2;\omega) - \hat{\nabla}\tilde{f}_1(\mathbf{x}_1,\mathbf{z}_1,\mathbf{y}_2;\omega)\big) \\ -\nabla f_1(\mathbf{x}_1,\mathbf{x}_2,\mathbf{y}_1,\mathbf{y}_2) + \rho\big(\nabla\tilde{f}_1(\mathbf{x}_1,\mathbf{y}_1,\mathbf{z}_2) - \nabla\tilde{f}_q(\mathbf{x}_1,\mathbf{z}_1,\mathbf{y}_2)\big) \end{array} \right\|^2 \\
\leq\ & 3\mathbb{E}\left\|\hat{\nabla} f_1(\mathbf{x}_1,\mathbf{x}_2,\mathbf{y}_1,\mathbf{y}_2;\omega) - \nabla f_1(\mathbf{x}_1,\mathbf{x}_2,\mathbf{y}_1,\mathbf{y}_2)\right\|^2 \\
&+ 3\rho^2\mathbb{E}\left\|\hat{\nabla}\tilde{f}_1(\mathbf{x}_1,\mathbf{y}_1,\mathbf{z}_2;\omega) - \nabla\tilde{f}_1(\mathbf{x}_1,\mathbf{y}_1,\mathbf{z}_2)\right\|^2 \\
&+ 3\rho^2\mathbb{E}\left\|\hat{\nabla}\tilde{f}_1(\mathbf{x}_1,\mathbf{z}_1,\mathbf{y}_2;\omega) - \nabla\tilde{f}_q(\mathbf{x}_1,\mathbf{z}_1,\mathbf{y}_2)\right\|^2 \\
\leq\ & 3\delta_f^2 + 6\rho^2\delta_{\tilde{f}}^2 = \delta_P^2,
\end{aligned}$$

where δ_P^2 is the same as in Algorithm 1. This means $\hat{\nabla}\bar{h}(x,y;\omega)$ specified in (90) satisfies Assumption G.2 with

$$\delta^2 = \delta_P^2 = 3\delta_f^2 + 6\rho^2\delta_{\tilde{f}}^2 = O(\epsilon^{-2}). \tag{91}$$

Since problem (14) is also an instance of (33) under the setting of (46) and (47), Algorithm 1 is exactly Algorithm 5 under the setting of (46) and (47). Therefore, we can directly apply Theorem G.3 and Lemma 4.2 to prove Theorem 6.3 as we show below.

Let f^*, D_1, D_2 and f_{low} be defined in (13), (10), (11) and (12). Let

$$\begin{aligned}
&L_{\nabla P_1} = L_{\nabla f_1} + 2\rho L_{\nabla \tilde{f}_1}, \\
&\delta_P^2 = 3\delta_f^2 + 6\rho^2\delta_{\tilde{f}}^2, \\
&C_P = \left\lceil 1 + \ln\left(\frac{6L_{\nabla P_1}D_p^2 + 3\epsilon D_2}{\hat{\epsilon}^2}\right)\right. \\
&\left.\times\left[\left(L_{\nabla P_1} + \max\left\{2L_{\nabla P_1}, \frac{\epsilon}{2D_2}\right\}\right)\left(\frac{2\sqrt{2D_2}}{\sqrt{L_{\nabla P_1}\epsilon}} + \frac{3}{L_{\nabla P_1}} + \frac{24D_2}{\epsilon}\right) + \frac{\delta_P^2}{\hat{\epsilon}^2}\left(\frac{21}{L_{\nabla P_1}} + \frac{360D_2}{\epsilon}\right)\right]\right\rceil \\
&K_P = \left\lceil \left(\mathbb{E}\left[\max_{\mathbf{x}_2} f(\mathbf{x}_1^{(0)},\mathbf{x}_2,\mathbf{y}_1^{(0)},\mathbf{y}_2^{(0)})\right] + 1 - f^* + \epsilon D_2/4\right)\hat{\epsilon}^{-2}\right\rceil \\
&N_P = C_P K_P.
\end{aligned}$$

Note that

$$\mathbb{E}\left[\max_{\mathbf{x}_2,\mathbf{z}_1,\mathbf{z}_2} P_\rho(\mathbf{x}_1^{(0)},\mathbf{x}_2,\mathbf{y}_1^{(0)},\mathbf{y}_2^{(0)},\mathbf{z}_1,\mathbf{z}_2)\right] \leq \mathbb{E}\left[\max_{\mathbf{x}_2} f(\mathbf{x}_1^{(0)},\mathbf{x}_2,\mathbf{y}_1^{(0)},\mathbf{y}_2^{(0)})\right] + 1 \tag{92}$$

because of the initial condition (25) and the fact that $\rho = \frac{1}{\epsilon}$.

Recall that $\hat{\epsilon}$ is set to be $\epsilon^{1.5}$ in Algorithm 1. According to the discussion above, Algorithm 1 is exactly Algorithm 5 under the setting of (46) and (47). In this setting, $\max_y H(x^0,y)$ in Theorem G.3 corresponds to

$$\max_{\mathbf{x}_2,\mathbf{z}_1,\mathbf{z}_2} P_\rho(\mathbf{x}_1^{(0)},\mathbf{x}_2,\mathbf{y}_1^{(0)},\mathbf{y}_2^{(0)},\mathbf{z}_1,\mathbf{z}_2).$$

Applying Theorem G.3 under this setting with $\hat{\epsilon} = \epsilon^{1.5}$ and with $\max_y H(x^0,y)$ in K in Theorem G.3 replaced by its upper bound

$$\mathbb{E}\left[\max_{\mathbf{x}_2} f(\mathbf{x}_1^{(0)},\mathbf{x}_2,\mathbf{y}_1^{(0)},\mathbf{y}_2^{(0)})\right] + 1, \tag{93}$$

we can show that, in complexity no more than N_P, Algorithm 1 outputs $((\mathbf{x}_1^\epsilon,\mathbf{y}_1^\epsilon,\mathbf{y}_2^\epsilon),(\mathbf{x}_2^\epsilon,\mathbf{z}_1^\epsilon,\mathbf{z}_2^\epsilon))$ such that there exists a stochastic solution $((\hat{\mathbf{x}}_1^\epsilon,\hat{\mathbf{y}}_1^\epsilon,\hat{\mathbf{y}}_2^\epsilon),(\hat{\mathbf{x}}_2^\epsilon,\hat{\mathbf{z}}_1^\epsilon,\hat{\mathbf{z}}_2^\epsilon))$ such that

$$\begin{aligned}
&\mathbb{E}\left\|(\mathbf{x}_1^\epsilon,\mathbf{y}_1^\epsilon,\mathbf{y}_2^\epsilon) - (\hat{\mathbf{x}}_1^\epsilon,\hat{\mathbf{y}}_1^\epsilon,\hat{\mathbf{y}}_2^\epsilon)\right\| \\
\leq\ &\mathbb{E}\left\|((\mathbf{x}_1^\epsilon,\mathbf{y}_1^\epsilon,\mathbf{y}_2^\epsilon),(\mathbf{x}_2^\epsilon,\mathbf{z}_1^\epsilon,\mathbf{z}_2^\epsilon)) - ((\hat{\mathbf{x}}_1^\epsilon,\hat{\mathbf{y}}_1^\epsilon,\hat{\mathbf{y}}_2^\epsilon),(\hat{\mathbf{x}}_2^\epsilon,\hat{\mathbf{z}}_1^\epsilon,\hat{\mathbf{z}}_2^\epsilon))\right\| \\
\leq\ &\max\left\{\frac{\hat{\epsilon}}{\sqrt{L_{\nabla P_1}}}, \frac{\hat{\epsilon}}{\sqrt{\epsilon/(2D_2)}}\right\}
\end{aligned}$$

and $((\hat{\mathbf{x}}_1^\epsilon, \hat{\mathbf{y}}_1^\epsilon, \hat{\mathbf{y}}_2^\epsilon), (\hat{\mathbf{x}}_2^\epsilon, \hat{\mathbf{z}}_1^\epsilon, \hat{\mathbf{z}}_2^\epsilon))$ is an $\max\left\{\sqrt{2L_{\nabla P_1}}\hat{\epsilon}, \frac{\epsilon}{2}\right\}$-primal-dual stationary point of (14) in expectation.

Since $\hat{\epsilon} = \epsilon^{1.5}$ and $L_{\nabla P_1} = O(\rho) = O(\epsilon^{-1})$, we have

$$\max\left\{\sqrt{2L_{\nabla P_1}}\hat{\epsilon}, \frac{\epsilon}{2}, \frac{\hat{\epsilon}}{\sqrt{L_{\nabla P_1}}}, \frac{\hat{\epsilon}}{\sqrt{\epsilon/(2D_2)}}\right\} = O(\epsilon)$$

meaning that

$$\mathbb{E}\left\|(\mathbf{x}_1^\epsilon, \mathbf{y}_1^\epsilon, \mathbf{y}_2^\epsilon) - (\hat{\mathbf{x}}_1^\epsilon, \hat{\mathbf{y}}_1^\epsilon, \hat{\mathbf{y}}_2^\epsilon)\right\| = O(\epsilon) \tag{94}$$

and $((\hat{\mathbf{x}}_1^\epsilon, \hat{\mathbf{y}}_1^\epsilon, \hat{\mathbf{y}}_2^\epsilon), (\hat{\mathbf{x}}_2^\epsilon, \hat{\mathbf{z}}_1^\epsilon, \hat{\mathbf{z}}_2^\epsilon))$ is an $O(\epsilon)$-primal-dual stationary point of (14) in expectation.

Additionally, (79) in Theorem G.3 also implies that $((\hat{\mathbf{x}}_1^\epsilon, \hat{\mathbf{y}}_1^\epsilon, \hat{\mathbf{y}}_2^\epsilon), (\hat{\mathbf{x}}_2^\epsilon, \hat{\mathbf{z}}_1^\epsilon, \hat{\mathbf{z}}_2^\epsilon))$ satisfies

$$\begin{aligned}
&\mathbb{E}\left[\max_{\mathbf{x}_2,\mathbf{z}_1,\mathbf{z}_2} P_\rho(\hat{\mathbf{x}}_1^\epsilon, \mathbf{x}_2, \hat{\mathbf{y}}_1^\epsilon, \hat{\mathbf{y}}_2^\epsilon, \mathbf{z}_1, \mathbf{z}_2)\right] \\
\le& 2\mathbb{E}\left[\max_{\mathbf{x}_2,\mathbf{z}_1,\mathbf{z}_2} P_\rho(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)}, \mathbf{z}_1, \mathbf{z}_2)\right] - f^* + \epsilon D_2/2.
\end{aligned}$$

On the other hand, by condition (25), it holds that

$$\begin{aligned}
&\mathbb{E}\left[\max_{\mathbf{x}_2,\mathbf{z}_1,\mathbf{z}_2} P_\rho(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)}, \mathbf{z}_1, \mathbf{z}_2)\right] \\
=& \mathbb{E}\left[\max_{\mathbf{x}_2} f(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)}) + \rho\big(p(\mathbf{x}_1^{(0)}, \mathbf{y}_1^{(0)}) - d(\mathbf{x}_1^{(0)}, \mathbf{y}_2^{(0)})\big)\right] \\
\le& \mathbb{E}\left[\max_{\mathbf{x}_2} f(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)})\right] + 1.
\end{aligned}$$

Combining the two inequalities above gives

$$\begin{aligned}
&\rho^{-1}\left(\mathbb{E}\left[\max_{\mathbf{x}_2,\mathbf{z}_1,\mathbf{z}_2} P_\rho(\hat{\mathbf{x}}_1^\epsilon, \mathbf{x}_2, \hat{\mathbf{y}}_1^\epsilon, \hat{\mathbf{y}}_2^\epsilon, \mathbf{z}_1, \mathbf{z}_2)\right] - f_{\text{low}}\right) \\
\le& \rho^{-1}\left(2\mathbb{E}\left[\max_{\mathbf{x}_2,\mathbf{z}_1,\mathbf{z}_2} P_\rho(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)}, \mathbf{z}_1, \mathbf{z}_2)\right] - f^* - f_{\text{low}} + \epsilon D_2/2\right) \\
\le& \rho^{-1}\left(2\mathbb{E}\left[\max_{\mathbf{x}_2} f(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)})\right] + 2 - f^* - f_{\text{low}} + \epsilon D_2/2\right) \\
\le& O(\epsilon)
\end{aligned} \tag{95}$$

where the last inequality is because $L_{\nabla P_1} = O(\rho) = O(\epsilon^{-1})$. By Lemma 4.2, $(\hat{\mathbf{x}}_1^\epsilon, \hat{\mathbf{y}}_1^\epsilon, \hat{\mathbf{y}}_2^\epsilon)$ is a lifted $O(\epsilon)$-KKT point of (1) in expectation. Because of (94), $(\mathbf{x}_1^\epsilon, \mathbf{y}_1^\epsilon, \mathbf{y}_2^\epsilon)$ is a nearly lifted $O(\epsilon)$-KKT point of (1) in expectation.

Moreover, since $L_{\nabla P_1} = O(\rho) = O(\epsilon^{-1})$, $\delta_P^2 = O(\rho^2) = O(\epsilon^{-2})$, we have $C_P = \tilde{O}(\epsilon^{-6})$, $K_P = O(\epsilon^{-3})$ and $N_P = \tilde{O}(\epsilon^{-9})$. Therefore, Algorithm 1 has a complexity of $\tilde{O}(\epsilon^{-9})$ for finding a nearly lifted $O(\epsilon)$-KKT point of (1) in expectation. □

I Deterministic and stochastic first-order methods for nonconvex-strongly-concave minimax problem

In this section, we first consider the oracle complexity for finding an ϵ-primal-dual stationary point of (33) (defined in (36)) when h is nonconvex in x but strongly concave in y. More specifically, we make the following assumption in addition to Assumption C.1.

Assumption I.1. $h(x, y)$ is μ_y-strongly concave in y.

To find an ϵ-primal-dual stationary point of (33), we still apply the procedure (37), but the difference is that subproblem (37) becomes an instance of (39) under the setting of (40) and

$$\begin{aligned}
L_{\nabla\bar{h}} &\leftarrow L_{\nabla h} + \max\{2L_{\nabla h}, \epsilon/(2D_q)\}, \\
\sigma_x &\leftarrow L_{\nabla h}, \\
\sigma_y &\leftarrow \epsilon/(2D_q) + \mu_y.
\end{aligned} \tag{96}$$

where D_q is defined in (34) and μ_y is from Assumption I.1.

We can still apply the optimal Algorithm 2 to (39) under the setting of (40) and (96), which still leads to Algorithm 3 except that we need to set $(L_{\nabla \bar{h}}, \sigma_x, \sigma_y)$ as in (96) in Line 3 of Algorithm 3. The oracle complexity of this modified Algorithm 3 can be proved after making a small change in the proof of [53, Theorem 2] to adapt it for the case when (33) is nonconvex-strongly-concave. In particular, the only change we need is to replace the strong concavity module σ_y of subproblem (38) in their proof by $\epsilon/(2D_q)$ to $\epsilon/(2D_q) + \mu_h$. This gives us the following oracle complexity of the modified Algorithm 3, which covers Theorem C.3 as a special case when $\mu_y = 0$. Since the proof is almost identical to that of Theorem C.3, we omit it here.

Theorem I.2. *Suppose Assumptions C.1 and I.1 hold. Also, suppose $(L_{\nabla \bar{h}}, \sigma_x, \sigma_y)$ is set as in* (96) *in Line 3 of Algorithm 3. Let H, H^*, D_p, D_q and H_{low} be defined in* (33)*,* (34) *and* (35)*. Let σ_y be defined in* (96)*. Let*

$$
\begin{aligned}
\alpha =& \ \min\{1, \sqrt{8\sigma_y/L_{\nabla h}}\},\\
\delta =& \ (2+\alpha^{-1})L_{\nabla h}D_p^2 + \max\{2\sigma_y, \alpha L_{\nabla h}/4\}D_q^2,\\
K =& \ \left\lceil 16(\max_y H(x^0,y) - H^* + \epsilon D_q/4)L_{\nabla h}\epsilon^{-2} + 32\hat{\epsilon}^2(1+L_{\nabla h}^2/\sigma_y^2)\epsilon^{-2} - 1)\right\rceil_+\\
C =& \ \frac{4\max\{\frac{1}{2L_{\nabla h}}, \min\{\frac{1}{2\sigma_y}, \frac{4}{\alpha L_{\nabla h}}\}\}\Big(\delta + 2\alpha^{-1}(H^* - H_{low} + \epsilon D_q/4 + L_{\nabla h}D_p^2)\Big)}{[(3L_{\nabla h} + \epsilon/(2D_q))^2/\min\{L_{\nabla h}, \sigma_y\} + 3L_{\nabla h} + \epsilon/(2D_q)]^{-2}\hat{\epsilon}^2}\\
N =& \ (\left\lceil 96\sqrt{2}(1 + (24L_{\nabla h} + 4\epsilon/D_q)L_{\nabla h}^{-1})\right\rceil + 2)\max\{2, \sqrt{L_{\nabla h}/(2\sigma_y)}\}\\
&\times \Big((K+1)\times(\log(C))_+ + K + 1 + 2K\log(K+1)\Big).
\end{aligned}
$$

Then Algorithm 3 outputs an ϵ-primal-dual stationary point (x^ϵ, y^ϵ) of (33) *(e.g.,* (36) *holds) in oracle complexity no more than N. Moreover, (x^ϵ, y^ϵ) satisfies*

$$
\max_y H(x^\epsilon, y) \le \max_y H(x^0, y) + \epsilon D_q/4 + 2\hat{\epsilon}^2(L_{\nabla h}^{-1} + L_{\nabla h}/\sigma_y^2). \tag{97}
$$

Remark I.3. Note that $\sigma_y = \Omega(\mu_y)$. According to this theorem, when h is nonconvex-strongly-concave, the modified Algorithm 3 finds an ϵ-primal-dual stationary point of (33) in $K = O(L_{\nabla h}\epsilon^{-2} + \hat{\epsilon}^2(L_{\nabla h}^2/\mu_y^2)\epsilon^{-2})$ iterations with a total oracle complexity of $N = \tilde{O}(\sqrt{L_{\nabla h}/\mu_y}K)$. If $\hat{\epsilon}$ is further set to be $O(1)$ and $L_{\nabla h} = O(1)$ and $\mu_y = O(1)$, the complexity becomes $N = \tilde{O}(\epsilon^{-2})$.

Next, we consider the stochastic setting where Assumptions C.1, G.1 and I.1 hold, and study the stochastic oracle complexity for finding a nearly ϵ-primal-dual stationary point of (33) defined at the beginning of Section G. Just like the deterministic case above, we apply the procedure (37), where subproblem (37) is an instance of (39) under the setting of (40) and (96).

Due to the stochastic setting, we still apply Algorithm 4 to (39) under the setting of (40) and (96), which still leads to Algorithm 5 except that we need to set $(L_{\nabla \bar{h}}, \sigma_x, \sigma_y)$ as in (96) in Line 3 of Algorithm 5. The oracle complexity of this modified Algorithm 5 can be proved after making a small change in the proof of Theorem G.3 to adapt it for the case when (33) is nonconvex-strongly-concave. In particular, the only change we need is to replace the strong concavity module σ_y of subproblem (38) in that proof by $\epsilon/(2D_q)$ to $\epsilon/(2D_q) + \mu_h$. This gives us the following oracle complexity of the modified Algorithm 5, which covers Theorem G.3 as a special case when $\mu_y = 0$. Since the proof is almost identical to that of Theorem G.3, we omit it here.

Theorem I.4. *Suppose Assumptions C.1, G.1 and I.1 hold. Also, suppose $(L_{\nabla \bar{h}}, \sigma_x, \sigma_y)$ is set as in* (96) *in Line 3 of Algorithm 5. Let H, H^*, D_p, and D_q be defined in* (33)*,* (34) *and* (35)*. Let σ_y be*

defined in (96). *Let*

$$K = \left\lceil \left(\mathbb{E}\left[\max_y H(x^0, y) \right] - H^* + \epsilon D_q/4 \right) \hat{\epsilon}^{-2} \right\rceil,$$

$$C = \left\lceil 1 + \ln\left(\frac{6L_{\nabla h} D_p^2 + 6\sigma_y D_q^2}{\hat{\epsilon}^2} \right) \left[\left(L_{\nabla h} + \max\left\{ 2L_{\nabla h}, \frac{\epsilon}{2D_q} \right\} \right) \left(\frac{2}{\sqrt{L_{\nabla h}\sigma_y}} + \frac{3}{L_{\nabla h}} + \frac{12}{\sigma_y} \right) + \frac{\delta^2}{\hat{\epsilon}^2} \left(\frac{21}{L_{\nabla h}} + \frac{180}{\sigma_y} \right) \right] \right\rceil,$$

$N = KC$.

Then Algorithm 5 terminates and outputs a solution (x^ϵ, y^ϵ) *such that there exists a stochastic solution* $(\hat{x}^\epsilon, \hat{y}^\epsilon)$ *such that*

$$\mathbb{E}\|(x^\epsilon, y^\epsilon) - (\hat{x}^\epsilon, \hat{y}^\epsilon)\| \le \max\left\{ \frac{\hat{\epsilon}}{\sqrt{L_{\nabla h}}}, \frac{\hat{\epsilon}}{\sqrt{\sigma_y}} \right\} \tag{98}$$

and $(\hat{x}^\epsilon, \hat{y}^\epsilon)$ *is a* $\max\left\{\sqrt{2L_{\nabla h}}\hat{\epsilon}, \frac{\epsilon}{2}\right\}$*-primal-dual stationary point of* (33) *in expectation. As a consequence, Algorithm 5 finds a nearly* $\max\left\{\sqrt{2L_{\nabla h}}\hat{\epsilon}, \frac{\epsilon}{2}, \frac{\hat{\epsilon}}{\sqrt{L_{\nabla h}}}, \frac{\hat{\epsilon}}{\sqrt{\sigma_y}}\right\}$*-primal-dual stationary point* (x^ϵ, y^ϵ) *of* (33) *in expectation in complexity no more than* N*. Moreover,* (79) *holds.*

Remark I.5. Note that $\sigma_y = \Omega(\mu_y) = \Omega(1)$. According to this theorem, when $L_{\nabla h} = O(1)$, the modified Algorithm 5 finds a nearly $O\left(\max\left\{\epsilon, \hat{\epsilon}\right\}\right)$-primal-dual stationary point of (33) in $K = O(\hat{\epsilon}^{-2})$ iterations with a total stochastic oracle complexity of $N = \tilde{O}((1 + \delta^2\epsilon^{-1}\hat{\epsilon}^{-2})K)$. If $\hat{\epsilon}$ is further set to be $\Theta\left(\epsilon\right)$ and $L_{\nabla h} = O(1)$, the complexity becomes $N = \tilde{O}((\epsilon^{-2} + \delta^2\epsilon^{-5}))$.

J Complexity under strong convexity and strong concavity

In this section, we present additional complexity results for Algorithm 1 in both the deterministic and stochastic cases when f and $\tilde{f}$ further satisfy strong convexity and strong concavity in addition to Assumption 3.1.

J.1 Reduced complexity in deterministic case

We first state the following assumption.

Assumption J.1. $f_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2)$ is $\mu_{\mathbf{x}_2}$-strongly concave in $\mathbf{x}_2$. $\tilde{f}_1(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2)$ is $\mu_{\mathbf{y}_1}$-strongly convex in $\mathbf{y}_1$ and $\mu_{\mathbf{y}_2}$-strongly concave in $\mathbf{y}_2$.

Theorem J.2. *Suppose Assumptions 3.1 and J.1 hold and* $(\mathbf{x}_1^{(0)}, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)})$ *in Algorithm 1 satisfies* (22). *Also, suppose* σ_y *in Line 5 of Algorithm 1 is set to* $\epsilon/(2D_2) + \min\{\mu_{\mathbf{x}_2}, \rho\mu_{\mathbf{y}_1}, \rho\mu_{\mathbf{y}_2}\}$ *and* $\hat{\epsilon}$ *in Line 3 is set to* $\epsilon^{0.5}$*. Algorithm 1 outputs a lifted* $O(\epsilon)$*-KKT point* $(\mathbf{x}_1^\epsilon, \mathbf{y}_1^\epsilon, \mathbf{y}_2^\epsilon)$ *of* (1) *with oracle complexity* $\tilde{O}(\epsilon^{-3.5})$.

Proof of Theorem J.2. Due to Assumption J.1, $P_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2, \mathbf{z}_1, \mathbf{z}_2)$ is $\min\{\mu_{\mathbf{x}_2}, \rho\mu_{\mathbf{y}_1}, \rho\mu_{\mathbf{y}_2}\}$-strongly concave in $(\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2)$. By the same argument as in the proof of Theorem 4.3, we can still show that the deterministic case of Algorithm 1 with σ_y in Line 5 set to $\epsilon/(2D_2) + \min\{\mu_{\mathbf{x}_2}, \rho\mu_{\mathbf{y}_1}, \rho\mu_{\mathbf{y}_2}\}$ is exactly Algorithm 3 with $(L_{\nabla\bar{h}}, \sigma_x, \sigma_y)$ in Line 3 set as in (96). Similarly, when (1) satisfies Assumptions 3.1 and J.1 and ρ_1 and ρ_2 are set as in Algorithm 1 (i.e., as in (48)), (21) is an instance of (39) satisfying Assumption (C.2) with

$$\begin{aligned} L_{\nabla\bar{h}} &\leftarrow L_{\nabla f} + 2\rho L_{\nabla\tilde{f}} + \max\{\rho_1, \rho_2\}, \\ \sigma_x &\leftarrow L_{\nabla f} + 2\rho L_{\nabla\tilde{f}}, \\ \sigma_y &\leftarrow \epsilon/(2D_2) + \min\{\mu_{\mathbf{x}_2}, \rho\mu_{\mathbf{y}_1}, \rho\mu_{\mathbf{y}_2}\}. \end{aligned} \tag{99}$$

Therefore, we can directly apply Theorem I.2 and Lemma 4.2 to prove Theorem J.2. While doing so, we just need to set $\hat{\epsilon} = \epsilon^{0.5}$ in Theorem I.2 according to the new Line 3 of Algorithm 1.

Let σ_y, f^*, D_1, D_2 and f_{low} be defined in (99), (13), (10), (11) and (12). Let

$$
\begin{aligned}
L_{\nabla P_1} &= L_{\nabla f_1} + 2\rho L_{\nabla \tilde{f}_1},\\
\alpha_P &= \min\{1, \sqrt{8\sigma_y / L_{\nabla P_1}}\},\\
\delta_P &= (2+\alpha_P^{-1}) L_{\nabla P_1} D_1^2 + \max\{2\sigma_y, \alpha_P L_{\nabla P_1}/4\} D_2^2,\\
K_P &= \left\lceil \begin{array}{l} 16(\max_{\mathbf{x}_2} f(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)}) + 1 - f_{\text{low}}) L_{\nabla P_1} \epsilon^{-2} \\ +32\epsilon(1+\sigma_y^{-2} L_{\nabla P_1}^2)\epsilon^{-2} - 1 \end{array} \right\rceil_+\\
C_P &= \Big(\frac{4\max\{\frac{1}{2L_{\nabla P_1}}, \min\{\frac{1}{2\sigma_y}, \frac{4}{\alpha_P L_{\nabla P_1}}\}\}}{[(3L_{\nabla P_1} + \epsilon/(2D_2))^2 / \min\{L_{\nabla P_1}, \sigma_y\} + 3L_{\nabla P_1} + \epsilon/(2D_2)]^{-2}\epsilon} \Big)\\
&\qquad \times \Big(\delta_P + 2\alpha_P^{-1}(f^* - f_{\text{low}} + L_{\nabla P_1} D_1^2) \Big),\\
N_P &= (\left\lceil 96\sqrt{2}(1 + (24 L_{\nabla P_1} + 4\epsilon/D_2) L_{\nabla P_1}^{-1}) \right\rceil + 2) \max\{2, \sqrt{L_{\nabla P_1}/(2\sigma_y)}\}\\
&\qquad \times \Big((K_P+1) \times (\log(C_P))_+ + K_P + 1 + 2K_P \log(K_P+1) \Big).
\end{aligned}
$$

Recall that $\hat{\epsilon}$ is set to be $\epsilon^{0.5}$ in the modified Algorithm 1. Applying Theorem I.2 under this setting with $\hat{\epsilon} = \epsilon^{0.5}$, we can show that Algorithm 1 outputs an ϵ-primal-dual stationary point $((\mathbf{x}_1^\epsilon, \mathbf{y}_1^\epsilon, \mathbf{y}_2^\epsilon), (\mathbf{x}_2^\epsilon, \mathbf{z}_1^\epsilon, \mathbf{z}_2^\epsilon))$ of (14) in oracle complexity no more than N_P above. Since $\sigma_y = \Omega(\mu_{\mathbf{x}_2}) = \Omega(1)$ and $L_{\nabla P_1} = O(\rho) = O(\epsilon^{-1})$, we have $K_P = O(\epsilon^{-3})$ and $N_P = \tilde{O}(\epsilon^{-3.5})$.

Moreover, (97) in Theorem I.2 also implies that $((\mathbf{x}_1^\epsilon, \mathbf{y}_1^\epsilon, \mathbf{y}_2^\epsilon), (\mathbf{x}_2^\epsilon, \mathbf{z}_1^\epsilon, \mathbf{z}_2^\epsilon))$ satisfies

$$
\begin{aligned}
\max_{\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2} P_\rho(\mathbf{x}_1^\epsilon, \mathbf{x}_2, \mathbf{y}_1^\epsilon, \mathbf{y}_2^\epsilon, \mathbf{z}_1, \mathbf{z}_2) \le & \max_{\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2} P_\rho(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)}, \mathbf{z}_1, \mathbf{z}_2)\\
& + \epsilon D_2/4 + 2\epsilon(L_{\nabla P_1}^{-1} + L_{\nabla P_1}/\sigma_y^2).
\end{aligned}
$$

Combining the inequality above with (51) gives

$$
\begin{aligned}
&\rho^{-1}\left(\max_{\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2} P_\rho(\mathbf{x}_1^\epsilon, \mathbf{x}_2, \mathbf{y}_1^\epsilon, \mathbf{y}_2^\epsilon, \mathbf{z}_1, \mathbf{z}_2) - f_{\text{low}} \right)\\
\le &\rho^{-1}\left(\max_{\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2} P_\rho(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)}, \mathbf{z}_1, \mathbf{z}_2) - f_{\text{low}} + \epsilon D_2/4 + 2\epsilon(L_{\nabla P_1}^{-1} + L_{\nabla P_1}/\sigma_y^2 \right)\\
\le &\rho^{-1}\left(\max_{\mathbf{x}_2} f(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)}) + 1 - f_{\text{low}} + \epsilon D_2/4 + 2\epsilon(L_{\nabla P_1}^{-1} + L_{\nabla P_1}/\sigma_y^2 \right)\\
\le &O(\epsilon),
\end{aligned}
$$

where the last inequality is because $\sigma_y = \Omega(\mu_{\mathbf{x}_2}) = \Omega(1)$ and $L_{\nabla P_1} = O(\rho) = O(\epsilon^{-1})$. By Lemma 4.2, $(\mathbf{x}_1^\epsilon, \mathbf{y}_1^\epsilon, \mathbf{y}_2^\epsilon)$ is a lifted $O(\epsilon)$-KKT point of (1). □

We further consider the assumption below.

Assumption J.3. The upper-level maximization variable $\mathbf{x}_2$ is absent; that is, $f_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2) = f_1(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2)$ and $f_3 \equiv 0$. Furthermore, $\tilde{f}_1(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2)$ is $\mu_{\mathbf{y}_1}$-strongly convex in $\mathbf{y}_1$ and $\mu_{\mathbf{y}_2}$-strongly concave in $\mathbf{y}_2$.

Theorem J.4. *Suppose Assumptions 3.1 and J.3 hold and $(\mathbf{x}_1^{(0)}, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)})$ in Algorithm 1 satisfies (22). Also, suppose σ_y in Line 5 of Algorithm 1 is set to $\epsilon/(2D_2) + \min\{\rho\mu_{\mathbf{y}_1}, \rho\mu_{\mathbf{y}_2}\}$ and $\hat{\epsilon}$ in Line 3 is set to 1. Algorithm 1 outputs a lifted $O(\epsilon)$-KKT point $(\mathbf{x}_1^\epsilon, \mathbf{y}_1^\epsilon, \mathbf{y}_2^\epsilon)$ of (1) with oracle complexity $\tilde{O}(\epsilon^{-3})$.*

Proof of Theorem J.4. Under Assumption J.3, a lifted ϵ-KKT point is still defined as in Definition 3.2 except that the conditions related to $\mathbf{x}_2$ are removed. Due to Assumption J.3, $P_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2, \mathbf{z}_1, \mathbf{z}_2)$ is $\min\{\rho\mu_{\mathbf{y}_1}, \rho\mu_{\mathbf{y}_2}\}$-strongly concave in $(\mathbf{z}_1, \mathbf{z}_2)$. Then, the proof steps are almost the same as the proof of Theorem J.2. The only difference is that (21) is an instance of

(39) satisfying Assumption (C.2) with

$$\begin{aligned} L_{\nabla\bar{h}} &\leftarrow L_{\nabla f} + 2\rho L_{\nabla\tilde{f}} + \max\{\rho_1, \rho_2\}, \\ \sigma_x &\leftarrow L_{\nabla f} + 2\rho L_{\nabla\tilde{f}}, \\ \sigma_y &\leftarrow \epsilon/(2D_2) + \min\{\rho\mu_{\mathbf{y}_1}, \rho\mu_{\mathbf{y}_2}\}. \end{aligned} \tag{100}$$

Let

$$D_2 := \max\{\|(\mathbf{z}_1, \mathbf{z}_2) - (\mathbf{z}_1', \mathbf{z}_2')\| \mid (\mathbf{z}_1, \mathbf{z}_2), (, \mathbf{z}_1', \mathbf{z}_2') \in \mathcal{Y}_1 \times \mathcal{Y}_2\}. \tag{101}$$

Let σ_y, f^*, D_1, and f_{low} be defined in (100), (13), (10), and (12). Let

$$\begin{aligned} L_{\nabla P_1} &= L_{\nabla f_1} + 2\rho L_{\nabla\tilde{f}_1}, \\ \alpha_P &= \min\{1, \sqrt{8\sigma_y/L_{\nabla P_1}}\}, \\ \delta_P &= (2 + \alpha_P^{-1}) L_{\nabla P_1} D_1^2 + \max\{2\sigma_y, \alpha_P L_{\nabla P_1}/4\} D_2^2, \\ K_P &= \left\lceil \begin{array}{l} 16(\max_{\mathbf{x}_2} f(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)}) + 1 - f_{\text{low}}) L_{\nabla P_1} \epsilon^{-2} \\ +32(1 + \sigma_y^{-2} L_{\nabla P_1}^2)\epsilon^{-2} - 1 \end{array} \right\rceil_+ \\ C_P &= \Big(\frac{4\max\{\frac{1}{2L_{\nabla P_1}}, \min\{\frac{1}{2\sigma_y}, \frac{4}{\alpha_P L_{\nabla P_1}}\}\}}{[(3L_{\nabla P_1} + \epsilon/(2D_2))^2/\min\{L_{\nabla P_1}, \sigma_y\} + 3L_{\nabla P_1} + \epsilon/(2D_2)]^{-2}} \Big) \\ &\qquad \times \Big(\delta_P + 2\alpha_P^{-1}(f^* - f_{\text{low}} + L_{\nabla P_1} D_1^2) \Big), \\ N_P &= (\left\lceil 96\sqrt{2}(1 + (24L_{\nabla P_1} + 4\epsilon/D_2)L_{\nabla P_1}^{-1}) \right\rceil + 2)\max\{2, \sqrt{L_{\nabla P_1}/(2\sigma_y)}\} \\ &\qquad \times \Big((K_P + 1) \times (\log(C_P))_+ + K_P + 1 + 2K_P \log(K_P + 1) \Big). \end{aligned}$$

Recall that $\hat{\epsilon}$ is set to be 1 in the modified Algorithm 1. Applying Theorem I.2 under this setting with $\hat{\epsilon} = 1$, we can show that Algorithm 1 outputs an ϵ-primal-dual stationary point $((\mathbf{x}_1^\epsilon, \mathbf{y}_1^\epsilon, \mathbf{y}_2^\epsilon), (\mathbf{x}_2^\epsilon, \mathbf{z}_1^\epsilon, \mathbf{z}_2^\epsilon))$ of (14) in oracle complexity no more than N_P above. Since $\sigma_y = \Omega(\rho) = \Omega(\epsilon^{-1})$ and $L_{\nabla P_1} = O(\rho) = O(\epsilon^{-1})$, we have $K_P = O(\epsilon^{-3})$ and $N_P = \tilde{O}(\epsilon^{-3})$.

Moreover, (97) in Theorem I.2 also implies that $((\mathbf{x}_1^\epsilon, \mathbf{y}_1^\epsilon, \mathbf{y}_2^\epsilon), (\mathbf{x}_2^\epsilon, \mathbf{z}_1^\epsilon, \mathbf{z}_2^\epsilon))$ satisfies

$$\begin{aligned} \max_{\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2} P_\rho(\mathbf{x}_1^\epsilon, \mathbf{x}_2, \mathbf{y}_1^\epsilon, \mathbf{y}_2^\epsilon, \mathbf{z}_1, \mathbf{z}_2) \leq & \max_{\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2} P_\rho(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)}, \mathbf{z}_1, \mathbf{z}_2) \\ & + \epsilon D_2/4 + 2(L_{\nabla P_1}^{-1} + L_{\nabla P_1}/\sigma_y^2). \end{aligned}$$

Combining the inequality above with (51) gives

$$\begin{aligned} & \rho^{-1} \left(\max_{\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2} P_\rho(\mathbf{x}_1^\epsilon, \mathbf{x}_2, \mathbf{y}_1^\epsilon, \mathbf{y}_2^\epsilon, \mathbf{z}_1, \mathbf{z}_2) - f_{\text{low}} \right) \\ \leq & \rho^{-1} \left(\max_{\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2} P_\rho(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)}, \mathbf{z}_1, \mathbf{z}_2) - f_{\text{low}} + \epsilon D_2/4 + 2\epsilon(L_{\nabla P_1}^{-1} + L_{\nabla P_1}/\sigma_y^2 \right) \\ \leq & \rho^{-1} \left(\max_{\mathbf{x}_2} f(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)}) + 1 - f_{\text{low}} + \epsilon D_2/4 + 2(L_{\nabla P_1}^{-1} + L_{\nabla P_1}/\sigma_y^2 \right) \\ \leq & O(\epsilon), \end{aligned}$$

where the last inequality is because $\sigma_y = \Omega(\mu_{\mathbf{x}_2}) = \Omega(\rho) = \Omega(\epsilon^{-1})$ and $L_{\nabla P_1} = O(\rho) = O(\epsilon^{-1})$. By Lemma 4.2, $(\mathbf{x}_1^\epsilon, \mathbf{y}_1^\epsilon, \mathbf{y}_2^\epsilon)$ is a lifted $O(\epsilon)$-KKT point of (1). □

J.2 Reduced complexity in stochastic case

Theorem J.5. *Suppose Assumptions 3.1, 6.1 and J.1 hold, $(\mathbf{x}_1^{(0)}, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)})$ in Algorithm 1 satisfies (25) and K in Algorithm 1 satisfies $K = \left\lceil \left(\mathbb{E}\left[\max_{\mathbf{x}_2} f(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)}) \right] + 1 - f^* + \epsilon D_2/4 \right) \epsilon^{-3} \right\rceil$. Also, suppose σ_y in Line 5 of Algorithm 1 is set to $\epsilon/(2D_2) + \min\{\mu_{\mathbf{x}_2}, \rho\mu_{\mathbf{y}_1}, \rho\mu_{\mathbf{y}_2}\}$. Algorithm 1 terminates and outputs a nearly lifted $O(\epsilon)$-KKT point of (1) in expectation in stochastic oracle complexity $\tilde{O}(\epsilon^{-8})$.*

Proof of Theorem J.5. Due to Assumption J.1, $P_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2, \mathbf{z}_1, \mathbf{z}_2)$ is $\min\{\mu_{\mathbf{x}_2}, \rho\mu_{\mathbf{y}_1}, \rho\mu_{\mathbf{y}_2}\}$-strongly concave in $(\mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2)$. By the same argument as in the proof of Theorem 6.3, we can still show that the stochastic case of Algorithm 1 with σ_y in Line 5 set to $\epsilon/(2D_2)+\min\{\mu_{\mathbf{x}_2}, \rho\mu_{\mathbf{y}_1}, \rho\mu_{\mathbf{y}_2}\}$ is exactly Algorithm 5 with $(L_{\nabla\bar{h}}, \sigma_x, \sigma_y)$ in Line 3 set as in (96). Similarly, when (1) satisfies Assumptions 3.1 and J.1 and ρ_1 and ρ_2 are set as in Algorithm 1 (i.e., as in (48)), (21) is an instance of (39) satisfying Assumption (C.2) with (99) and satisfying Assumption (G.1) with (91). Therefore, we can directly apply Theorem I.4 and Lemma 4.2 to prove Theorem J.5. While doing so, we just need to set $\hat{\epsilon} = \epsilon^{1.5}$ in Theorem I.4 according to the Line 3 of Algorithm 1.

Let σ_y, f^*, D_1, D_2 and f_{low} be defined in (99), (13), (10), (11) and (12). Let

$L_{\nabla P_1} = L_{\nabla f_1} + 2\rho L_{\nabla \tilde{f}_1}$,

$\delta_P^2 = 3\delta_f^2 + 6\rho^2\delta_{\tilde{f}}^2$,

$$C_P = \left\lceil 1 + \ln\left(\frac{6L_{\nabla P_1}D_1^2 + 6\sigma_y D_2^2}{\hat{\epsilon}^2}\right)\left[\left(L_{\nabla P_1} + \max\left\{2L_{\nabla P_1}, \frac{\epsilon}{2D_2}\right\}\right)\left(\frac{2}{\sqrt{L_{\nabla P_1}\sigma_y}} + \frac{3}{L_{\nabla P_1}} + \frac{12}{\sigma_y}\right) + \frac{\delta_P^2}{\hat{\epsilon}^2}\left(\frac{21}{L_{\nabla P_1}} + \frac{180}{\sigma_y}\right)\right]\right\rceil$$

$$K_P = \left\lceil \left(\mathbb{E}\left[\max_{\mathbf{x}_2} f(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)})\right] + 1 - f^* + \epsilon D_2/4\right)\hat{\epsilon}^{-2}\right\rceil$$

$N_P = C_P K_P$.

Note that (92) still holds because of (25) and the fact that $\rho = \frac{1}{\epsilon}$. Applying Theorem I.4 under this setting with $\max_y H(x^0, y)$ in K in Theorem I.4 replaced by its upper bound (93), we can show that, in complexity no more than N_P, Algorithm 1 Algorithm 1 outputs $((\mathbf{x}_1^\epsilon, \mathbf{y}_1^\epsilon, \mathbf{y}_2^\epsilon), (\mathbf{x}_2^\epsilon, \mathbf{z}_1^\epsilon, \mathbf{z}_2^\epsilon))$ such that there exists a stochastic solution $((\hat{\mathbf{x}}_1^\epsilon, \hat{\mathbf{y}}_1^\epsilon, \hat{\mathbf{y}}_2^\epsilon), (\hat{\mathbf{x}}_2^\epsilon, \hat{\mathbf{z}}_1^\epsilon, \hat{\mathbf{z}}_2^\epsilon))$ such that

$$\mathbb{E}\left\|(\mathbf{x}_1^\epsilon, \mathbf{y}_1^\epsilon, \mathbf{y}_2^\epsilon) - (\hat{\mathbf{x}}_1^\epsilon, \hat{\mathbf{y}}_1^\epsilon, \hat{\mathbf{y}}_2^\epsilon)\right\| \le \max\left\{\frac{\hat{\epsilon}}{\sqrt{L_{\nabla P_1}}}, \frac{\hat{\epsilon}}{\sqrt{\sigma_y}}\right\}$$

and $((\hat{\mathbf{x}}_1^\epsilon, \hat{\mathbf{y}}_1^\epsilon, \hat{\mathbf{y}}_2^\epsilon), (\hat{\mathbf{x}}_2^\epsilon, \hat{\mathbf{z}}_1^\epsilon, \hat{\mathbf{z}}_2^\epsilon))$ is a $\max\left\{\sqrt{2L_{\nabla P_1}}\hat{\epsilon}, \frac{\epsilon}{2}\right\}$-primal-dual stationary point of (14) in expectation.

Since $\hat{\epsilon} = \epsilon^{1.5}$, $\sigma_y = \Omega(\mu_{\mathbf{x}_2}) = \Omega(1)$ and $L_{\nabla P_1} = O(\rho) = O(\epsilon^{-1})$, we have

$$\max\left\{\sqrt{2L_{\nabla P_1}}\hat{\epsilon}, \frac{\epsilon}{2}, \frac{\hat{\epsilon}}{\sqrt{L_{\nabla P_1}}}, \frac{\hat{\epsilon}}{\sqrt{\sigma_y}}\right\} = O(\epsilon)$$

meaning that (94) holds and $((\hat{\mathbf{x}}_1^\epsilon, \hat{\mathbf{y}}_1^\epsilon, \hat{\mathbf{y}}_2^\epsilon), (\hat{\mathbf{x}}_2^\epsilon, \hat{\mathbf{z}}_1^\epsilon, \hat{\mathbf{z}}_2^\epsilon))$ is an $O(\epsilon)$-primal-dual stationary point of (14) in expectation.

Additionally, (79) is also ensured by Theorem I.4, which implies (95) by the same argument used in the end of the proof of Theorem 6.3.

Moreover, since $\sigma_y = \Omega(\mu_{\mathbf{x}_2}) = \Omega(1)$, $L_{\nabla P_1} = O(\rho) = O(\epsilon^{-1})$, $\delta_P^2 = O(\rho^2) = O(\epsilon^{-2})$, we have $C_P = \tilde{O}(\epsilon^{-5})$, $K_P = O(\epsilon^{-3})$ and $N_P = \tilde{O}(\epsilon^{-8})$. Therefore, Algorithm 1 has a complexity of $\tilde{O}(\epsilon^{-8})$ for finding a nearly lifted $O(\epsilon)$-KKT point of (1) in expectation. □

Theorem J.6. *Suppose Assumptions 3.1, 6.1 and J.3 hold,* $(\mathbf{x}_1^{(0)}, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)})$ *in Algorithm 1 satisfies* (25) *and* K *in Algorithm 1 satisfies* $K = \left\lceil \left(\mathbb{E}\left[\max_{\mathbf{x}_2} f(\mathbf{x}_1^{(0)}, \mathbf{x}_2, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)})\right] + 1 - f^* + \epsilon D_2/4\right)\epsilon^{-3}\right\rceil$. *Also, suppose* σ_y *in Line 5 of Algorithm 1 is set to* $\epsilon/(2D_2) + \min\{\rho\mu_{\mathbf{y}_1}, \rho\mu_{\mathbf{y}_2}\}$. *Algorithm 1 terminates and outputs a nearly lifted* $O(\epsilon)$*-KKT point of* (1) *in expectation in stochastic oracle complexity* $\tilde{O}(\epsilon^{-7})$.

Proof of Theorem J.6. Under Assumption J.3, a lifted ϵ-KKT point is still defined as in Definition 3.2 except that the conditions related to $\mathbf{x}_2$ are removed. Due to Assumption J.3, $P_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2, \mathbf{z}_1, \mathbf{z}_2)$ is $\min\{\rho\mu_{\mathbf{y}_1}, \rho\mu_{\mathbf{y}_2}\}$-strongly concave in $(\mathbf{z}_1, \mathbf{z}_2)$. Then, the proof steps are almost the same as the proof of Theorem J.5. The only difference is that (21) is an instance of (39) satisfying Assumption (C.2) with (100) and satisfying Assumption (G.1) with (91).

Work	Lower level	Key assumptions	Stationarity guarantee
Jiang et al. [31]	Unconstrained min	PL condition on perturbed lower level; (δ, α)-flat upper objective	ϵ-KKT of gradient-constrained reformulation
Kwon et al. [40]	Constrained min	Proximal error bound; solution-path regularity for hyperobjective stationarity	ϵ-KKT of value-function reformulation; hyperobjective stationarity under additional regularity
Kwon et al. [39]	Unconstrained min	Strongly convex lower level	ϵ-stationarity of original hyperobjective
Our work	**Min–max**	**Convex–concave lower level**	**Lifted ϵ-KKT of value-function reformulation**

Table 3: Comparison with related penalty-based bilevel optimization methods.

In this case, $\sigma_y = \Omega(\rho) = \Omega(\epsilon^{-1})$. Moreover, since $L_{\nabla P_1} = O(\rho) = O(\epsilon^{-1})$, $\delta_P^2 = O(\rho^2) = O(\epsilon^{-2})$, we have $C_P = \tilde{O}(\epsilon^{-4})$, $K_P = O(\epsilon^{-3})$ and $N_P = \tilde{O}(\epsilon^{-7})$. Therefore, Algorithm 1 has a complexity of $\tilde{O}(\epsilon^{-7})$ for finding a nearly lifted $O(\epsilon)$-KKT point of (1) in expectation. □

K Detailed comparison with related penalty methods

Table 3 compares our method with three closely related penalty-based approaches in terms of their lower-level problem structures, key assumptions, and stationarity guarantees.

Oracle models and penalty parameters. Jiang et al. [31] use deterministic gradients with a penalty parameter of order $\Omega(\epsilon^{-(2-\alpha)})$. Kwon et al. [39] use stochastic gradients and a polynomially increasing penalty parameter that eventually reaches $\Omega(\epsilon^{-1})$. Kwon et al. [40] consider both deterministic and stochastic oracles, with an improved stochastic complexity bound under mean-squared smoothness. Both their method and ours use a penalty parameter of order $\Omega(\epsilon^{-1})$, while our approach supports both deterministic and stochastic gradients.

In summary, unlike the above methods, which primarily address lower-level minimization problems, our framework directly handles convex-concave minimax lower-level problems without requiring strong convexity or a Polyak–Lojasiewicz condition, while providing deterministic and stochastic first-order complexity guarantees for approximate KKT points.

L Additional details on numerical experiments

Synthetic experiments are run on Intel Xeon Gold 6438Y+, and deep DRO experiments are run on one H100 GPU.

L.1 Constrained linear optimization

L.1.1 Instance generation

We first randomly generate c and d with all the entries independently chosen from $\mathcal{N}(0, 1)$. We then randomly generate $\widetilde{A}$ and $\widetilde{B}$ with all the entries independently chosen from $\mathcal{N}(0, 0.01)$. In addition, we randomly generate $\hat{y} \in [-1, 1]^m$ with all the entries independently chosen from $\mathcal{N}(0, 0.1)$ and then projected to $[-1, 1]^m$ and choose $\tilde{d}$ and $\tilde{b}$ based on the KKT conditions that make $\hat{y}$ the optimal solution of the lower level optimization of Problem (26) with $x = 0$. More specifically, $\tilde{d}$ and $\tilde{b}$ are chosen as follows. First notice that $\hat{y}$ already satisfies the box constraint. We can construct the Lagrangian of the lower-level problem

$$\mathcal{L}(y, \lambda) = \tilde{d}^\top y + \lambda^\top (\tilde{B}y - \tilde{b}),$$

Then we generate λ randomly with $\lambda \sim |\mathcal{N}(0, I)|$. We then set $\tilde{b}$ and $\tilde{d}$ such that

$$\nabla_y \mathcal{L}(\hat{y}, \lambda) = \tilde{d} + \tilde{B}^\top \lambda = 0, \quad \tilde{B}\hat{y} - \tilde{b} = 0 \leq 0.$$

This way, the KKT conditions for the lower level problem hold and thus $\hat{y}$ is optimal.

L.1.2 Hyperparameters and implementations

For our method, we used a Lagrange bound of $B = 200$ and set input $\epsilon = 0.01$ in Algorithm 1. Hyperparameters of FOP and SMO, e.g., ϵ_k, μ_k and ρ_k, are set based on the description of Lu and Mei [54, Section 3]. It is worth mentioning that for the FOP baseline, they did not implement the pseudocode of Lu and Mei [52, Algorithm 2] but instead used a practical variant with a schedule for $(\epsilon_k, \mu_k, \rho_k)$ described in their experiments section. We similarly implemented the variant for the FOP baseline. For our method and each instance, we tune the Lipschitz constant $L_{\nabla P_1}$ used to construct the step sizes by performing a grid search on $\{1, 5, 6, 7, 8, 9, 10, 100\}$ based on the output performance. We find that 6 is the best value for all instances. For the two baselines, since the Lipschitz constants of their subproblems change with the outer iteration due to the dynamic penalty parameters, we set their Lipschitz constant to be $c \cdot L_k$, where L_k is the theoretical Lipschitz constant for the subproblem in their kth outer iteration, and c is chosen from $\{0.01, 0.05, 0.1, 0.2\}$ by a grid search based on the output performance.

By Step 2 of Algorithm 1 in [54], SMO needs to solve the following the subproblem up to an optimality gap ϵ_k in outer iteration k:

$$\min_{\mathbf{y}_1} \bar{f}(\mathbf{x}_1^{(k)}, \mathbf{y}_1) + \bar{f}_2(\mathbf{y}_1) + \frac{1}{2\rho_k \mu_k} \left\| [\mathbf{z}_2^{(k)} + \mu_k \bar{\mathbf{g}}(\mathbf{x}_1^{(k)}, \mathbf{y}_1)]_+ \right\|^2.$$

The goal is to obtain a good initial solution $\mathbf{y}_1^{(k)}$ for the minimax problem in iteration k. Following [54], we solve this problem using the optimal first-order method given in [54][Algorithm A.2], which guarantees an ϵ_k optimality gap at termination. For the same purpose, FOP needs to solve the following subproblem up to an optimality gap ϵ_k in outer iteration k:

$$\min_{\mathbf{y}_1} \bar{f}(\mathbf{x}_1^{(k)}, \mathbf{y}_1) + \bar{f}_2(\mathbf{y}_1) + \mu_k \left\| [\bar{\mathbf{g}}(\mathbf{x}_1^{(k)}, \mathbf{y}_1)]_+ \right\|^2.$$

We also solve it by [54][Algorithm A.2]. For the similar reason, we also need to obtain a good initial primal-dual solution $(\mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)})$ but we need it only at the beginning of Algorithm 1 instead of each outer iteration. We use the same $\mathbf{y}_1^{(0)}$ as the one solved from the FOP's subproblem above with $k = 0$ and set $\mathbf{y}_2^{(0)} = \mu_0 \nabla \bar{\mathbf{g}}(\mathbf{x}_1^{(0)}, \mathbf{y}_1^{(0)})[\bar{\mathbf{g}}(\mathbf{x}_1^{(0)}, \mathbf{y}_1^{(0)})]_+$, where μ_0 is set as in FOP.

All three methods use Algorithm 2 as a subroutine. We set the maximum number of outer iterations of Algorithm 2 to 200 and set the maximum number of inner iterations (Line 11) to 10000 for each method. Each of the three methods involves a precision-related parameter ϵ_k. We terminate each method once its $\epsilon_k \leq 0.01$, $\|[\bar{\mathbf{g}}(\mathbf{x}_1^{(k)}, \mathbf{y}_1^{(k)})]_+\| \leq 0.01$ and $\bar{f}(\mathbf{x}_1^{(k)}, \mathbf{y}_1^{(k)}) - \bar{f}^*(\mathbf{x}_1^{(k)}) \leq 0.01$.

L.1.3 Additional results

In Table 4, we compare the bound for the dual variables, i.e., B, used in our experiments with its theoretical values in (23) based on the lower-level Slater margin of the generated instances. To do so, we compute $G = -\min_{\|z\|_\infty \leq 1} \max_{i=1}^{l} \{\tilde{B}_i z - \tilde{b}_i\}$ as an estimate of the Slater margin and present its value together with the theoretical multiplier bounds calculated as (23) and the multiplier bound we used in experiments. The values in Table 4 are the averaged values over the five random instances.

(n, m, l)	**Slater Margin**	**Theoretical** B	B **Used**
$(100, 100, 5)$	0.34	32.86	200
$(200, 200, 10)$	0.49	52.60	200
$(300, 300, 15)$	0.58	60.49	200
$(400, 400, 20)$	0.70	93.31	200
$(500, 500, 25)$	0.78	126.65	200

Table 4: Slater margin and theoretical B for different instances.

In Table 5, we calculate the theoretical Lipschitz constant $L_{\nabla P_1}$ derived in in Appendix F, i.e.,

$$L_{\nabla P_1} = L_{\nabla f_1} + 2\rho(L_{\nabla \bar{f}_1} + B\sqrt{l} L_{\nabla \bar{\mathbf{g}}} + 2L_{\bar{\mathbf{g}}})$$

for each instance. Each value in Table 5 is the average over the five random instances. We observe that the gap between the theoretical values and the one we used (i.e., 6) increases with the problem size.

(n, m, l)	**Theoretical** $L_{\nabla P_1}$
$(100, 100, 5)$	31.54
$(200, 200, 10)$	45.12
$(300, 300, 15)$	56.08
$(400, 400, 20)$	64.42
$(500, 500, 25)$	72.74

Table 5: Theoretical Lipschitz constant $L_{\nabla P_1}$ for different instances.

L.2 Group DRO with hyperparameter tuning

L.2.1 Datasets

We use the same split for both datasets as provided by Sagawa et al. [62]. The primary task of the target label prediction is severely confounded by a spurious attribute (see Table 6). In the CelebA dataset, standard ERM models tend to exploit the spurious correlation, effectively using gender as a proxy for hair color to minimize average loss. This shortcut learning leads to disastrous performance on minority group, i.e., blond male [62]. To show the severity of this issue along the same strand, CUB images [67] are overlaid on top of Places images [80] with imbalanced proportions to create the Waterbirds dataset as done by Sagawa et al. [62], where the labels are the type of the bird (waterbirds vs. landbirds), confounded by the background (water vs. land). Because the data is defined by the intersection of the target label and the spurious attribute, it naturally partitions into $G = 4$ predefined groups. With deep learning paradigms, we use the proposed stochastic bilevel optimization from Section 6.

CelebA Groups	$\lvert D_{\textbf{train}}\rvert$	$\lvert D_{\textbf{val}}\rvert$
Dark Hair, Female	71,629	8,535
Dark Hair, Male	66,874	8,276
Blond Hair, Female	22,880	2,874
Blond Hair, Male	1,387	182

Waterbirds Groups	$\lvert D_{\textbf{train}}\rvert$	$\lvert D_{\textbf{val}}\rvert$
Landbirds, land background	3,498	467
Waterbirds, water background	1,057	466
Landbirds, water background	184	133
Waterbirds, land background	56	133

Table 6: Statistics for Training and Validation Sets. Group descriptions are {label, confounder}.

L.2.2 Comparison to Group DRO

Group DRO [62] developed a method to solve a minimax problem in the form of

$$\min_{\mathbf{x}_1, \mathbf{y}_1} \max_{\mathbf{y}_2 \in \mathcal{Y}} \mathcal{L}_{\text{train}}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{y}_2; C) := \mathbf{y}_2^\top \mathcal{L}^{\mathbf{x}_1}(\mathbf{y}_1) + \lambda(\|\mathbf{x}_1\|^2 + \|\mathbf{y}_1\|^2) + C \sum_{g=1}^{G} \frac{(\mathbf{y}_2)_g}{\sqrt{n_g}}. \tag{102}$$

There are several differences between our formulation in (27) and Group DRO (102): the latter (1) does not have the η-weighted regularization term for $\mathbf{y}_2$, (2) jointly trains the entire model (both the feature extractor $\mathbf{x}_1$ and the linear classifier $\mathbf{y}_1$), (3) has the loss adjustment term $C/\sqrt{n_g}$ for each group g, and (4) uses only the training set in the optimization model. For our method, we perform tuning of the feature extractor $\mathbf{x}_1$ through the loss on the validation set at the upper level, which is not exposed to the lower-level training problem just like the hold-out validation technique. Per SAPD algorithm 4, we sample one batch per gradient with replacement, so in each outer most iteration of our method, we sample three batches. We used balanced sampling for the gradient of the upper-level validation loss and uniform sampling for the gradient of the lower-level training loss.

L.2.3 Training settings and sensitivity analysis

We use ResNet-50 [22] as a feature extractor for representation learning on both datasets. To initialize the subproblem, the feature extractor ($\mathbf{x}_1$) uses the train-from-scratch option and classifier ($\mathbf{y}_1$) is initialized by sampling each coordinate from $\mathcal{N}(0, 0.01)$. In Table 7, we report the worst-group validation accuracy obtained by our methods when using different values of some tuning parameters in our algorithm. We find $T = 1$, i.e., running one iteration in Algorithm 4 to solve the proximal-point subproblem, leads to relatively good performance. We initialize η with 1.0 and note that gradient multiplication, while not used here, can be applied to accelerate learning. For all experiments, we set $D_y = 100$ and $\hat{\epsilon} = 10.0$. For CelebA, we set the batch size for constructing stochastic gradients to be 256. For Waterbirds, we set the batch size to be 128. We evaluate and report the test performance using the solution from the iteration of Algorithm 1 that achieves the best validation performance.

For other unmentioned hyperparameters, we use the default value provided in the code by Sagawa et al. [62].

Hyperparameter	**Value**	**Accuracy**	
		CelebA	Waterbirds
ρ in Algorithm 1	10	82.29 ± 0.90	83.33 ± 3.61
	100	86.46 ± 2.39	81.25 ± 5.41
	500	84.90 ± 3.25	39.59 ± 10.97
$L_{\nabla P_1}$	500	84.90 ± 3.25	80.21 ± 10.05
	1000	86.46 ± 2.38	81.25 ± 5.41
	2000	87.50 ± 4.13	83.34 ± 1.81
Total iteration T in Algorithm 4	**1**	86.46 ± 2.39	81.25 ± 5.41
	3	34.90 ± 5.49	37.50 ± 14.32
	5	31.77 ± 10.41	28.13 ± 10.83

Table 7: The worst-group validation accuracy using different tuning parameters. The best choice is bold.

L.2.4 Additional results

In Table 8, we show the training accuracies by DRO and our method with one random seed. Our method performs worse on the training set, and we believe it is because the feature extractor and the hyperparameter are updated mainly using the information from the validation set. In Table 9, we compare the runtimes of both methods. We want to emphasize that, although the runtime of Group DRO looks much longer in Table 9, we cannot conclude that Group DRO is more time-consuming than our method. This is because the two methods run for different numbers of iterations during training. We terminate the training algorithm of Group DRO after 40 epochs, according to the original training setup implemented in their code. On the contrary, we terminate our algorithm after 100 outer iterations, which use fewer than 400 epochs.

Dataset	ℓ_2 **Penalty**	**Average**		**Worst Group**	
		Group DRO	Ours	Group DRO	Ours
CelebA	1e-4	100.0	94.4	100.0	91.8
	0.1	95.0	95.4	93.4	93.0
Waterbirds	1e-4	100.0	96.1	100.0	92.6
	1.0	99.1	95.2	97.5	90.3

Table 8: Average and worst-group accuracy for DRO and our method on the training set on one seed.

Dataset	ℓ_2 Penalty	Group DRO	Ours
CelebA	1e-4	2h 37m 39s	**55m 19s**
	0.1	7h 38m 31s	**56m 26s**
Waterbirds	1e-4	1h 49m 01s	**28m 27s**
	1.0	1h 45m 21s	**28m 22s**

Table 9: Wall-clock comparison between Group DRO and our method on CelebA and Waterbirds.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We give the summary of contribution in the introduction. The main theoretical contribution we claim is given in Theorem 4.3. The good computational results of our methods claimed in the abstract is justified by the numerical results in Section 7.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: We discuss the limitations of the work in Section 8.

   Guidelines:

   - The answer [N/A] 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: Each theoretical result has been rigorously proved with complete details. The proofs are provided in Sections B, D, F, G, H, and E. The assumptions we need are clearly presented in Assumptions 3.1, 5.1, and 6.1.

   Guidelines:

   - The answer [N/A] 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: We fully disclose all the details of our main experimental results in Sections 7, and L, which are sufficient to reproduce the results. Code will be released upon publication.

   Guidelines:

   - The answer [N/A] 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: In the submitted supplemental materials, we have included all the codes we developed for the numerical experiments in this paper, which are sufficient to reproduce our results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The details are included in the Implementation Details of the Appendix L.

   Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [No]

Justification: When comparing with the baseline methods, we directly use the performance metrics reported in the corresponding papers. Since these papers do not report error bars or standard errors, we omit them as well for consistency.

Guidelines:

- The answer [N/A] 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: Provided in Appendix Implementation Details in Appendix L.

   Guidelines:

   - The answer [N/A] 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 research conducted for this paper complies with the NeurIPS Code of Ethics in all aspects. No human objects are involved, and we do not foresee any potential harms.

   Guidelines:

   - The answer [N/A] 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: [N/A]

    Justification: This is a theoretical paper. There's no any negative societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: All used assets are cited.

Guidelines:

- The answer [N/A] 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: [N/A]

Justification: This paper does not release new assets.

Guidelines:

- The answer [N/A] 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: [N/A]

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

Guidelines:

- The answer [N/A] 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: [N/A]

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

Guidelines:

- The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.