Journal of Machine Learning Research 23 (2022) 1-56 Submitted 8/21; Published 12/22

Lower Bounds and Accelerated Algorithms for Bilevel Optimization

Kaiyi Ji KAIYIJI@BUFFALO.EDU
Department of Computer Science and Engineering
University at Buffalo
Buffalo, NY 14260-2500, USA

Yingbin Liang LIANG.889@OSU.EDU
Department of Electrical and Computer Engineering
The Ohio State University
Columbus, OH 98195-4322, USA

Editor: Sathiya Keerthi

Abstract

Bilevel optimization has recently attracted growing interests due to its wide applications in modern machine learning problems. Although recent studies have characterized the convergence rate for several such popular algorithms, it is still unclear how much further these convergence rates can be improved. In this paper, we address this fundamental question from two perspectives. First, we provide the first-known lower complexity bounds of $\widetilde{\Omega}\left(\sqrt{\frac{L_y\widetilde{L}_{xy}^2}{\mu_x\mu_y^2}}\right)$ and $\widetilde{\Omega}\big(\frac{1}{\sqrt{\epsilon}}\min\{\kappa_y, \frac{1}{\sqrt{\epsilon^3}}\}\big)$ respectively for strongly-convex-strongly-convex and convex-strongly-convex bilevel optimizations. Second, we propose an accelerated bilevel optimizer named AccBiO, for which we provide the first-known complexity bounds without the gradient boundedness assumption (which was made in existing analyses) under the two aforementioned geometries. We also provide significantly tighter upper bounds than the existing complexity when the bounded gradient assumption does hold. We show that AccBiO achieves the optimal results (i.e., the upper and lower bounds match up to logarithmic factors) when the inner-level problem takes a quadratic form with a constant-level condition number. Interestingly, our lower bounds under both geometries are larger than the corresponding optimal complexities of minimax optimization, establishing that bilevel optimization is provably more challenging than minimax optimization. Our theoretical results are validated by numerical experiments.

Keywords: Bilevel optimization, lower bounds, accelerated algorithms, computational complexity, convergence rate, optimality.

1. Introduction

Bilevel optimization was first introduced by Bracken and McGill (1973), and since then has been studied for decades (Hansen et al., 1992; Shi et al., 2005; Moore, 2010). Recently, bilevel optimization has attracted growing interests due to its important role in various machine learning applications including meta-learning (Franceschi et al., 2018; Rajeswaran et al., 2019), hyperparameter optimization (Franceschi et al., 2018; Feurer and Hutter, 2019), imitation learning (Arora et al., 2020), and network architecture search (Liu et al., 2019; He et al., 2020). A general formulation of unconstrained bilevel optimization can be written as follows.

$$\min_{x\in\mathbb{R}^p} \Phi(x) := f(x, y^*(x)), \quad \text{s.t. } y^*(x) = \arg\min_{y\in\mathbb{R}^q} g(x, y), \tag{1}$$

where f and g are continuously differentiable functions. The problem eq. (1) contains two optimization procedures: at the inner level we search $y^*(x)$ as the minimizer of $g(x, y)$ with respect to (w.r.t.) y given x, and at the outer level we minimize the objective function $\Phi(x)$ w.r.t. x, which includes the compositional dependence on x via $y^*(x)$.

Most theoretical studies of bilevel optimization algorithms have focused on *the asymptotic* analysis without the convergence rate characterization. For example, Franceschi et al. (2018); Shaban et al. (2019) established the asymptotic convergence for gradient-based approaches when there is one single solution for the inner-level problem, and Liu et al. (2020); Li et al. (2020) extended the analysis to the setting where the inner-level problem allows multiple solutions. The *finite-time* analysis that characterizes the convergence rate of bilevel optimization algorithms is rather limited except a few studies recently. Grazzi et al. (2020) provided the iteration complexity of two dominant types of strategies, i.e., approximate implicit differentiation (AID) and iterative differentiation (ITD), for approximating the hypergradient $\nabla\Phi(x)$, but did not characterize the finite-time convergence for the entire execution of algorithms. Ghadimi and Wang (2018) proposed an AID-based bilevel approximation (BA) algorithm as well as an accelerated variant ABA, and analyzed their finite-time complexities under different loss geometries. In particular, the complexity upper bounds of BA and ABA are given by $\widetilde{\mathcal{O}}\Big(\frac{U^2\widetilde{L}_{xy}^4\rho_{yy}^2}{\mu_x^2\mu_y^6}\Big)$ and $\widetilde{\mathcal{O}}\Big(\frac{U\widetilde{L}_{xy}^2\rho_{yy}}{\mu_x\mu_y^3}\Big)$ for the strongly-convex-strongly-convex setting where $\Phi(\cdot)$ is μ_x-strongly-convex and $g(x,\cdot)$ is μ_y-strongly-convex, $\mathcal{O}\big(\frac{\kappa_y^{11.25}}{\epsilon^{1.25}}\big)$ and $\mathcal{O}\big(\frac{\kappa_y^{6.75}}{\epsilon^{0.75}}\big)$ for the convex-strongly-convex setting, and $\mathcal{O}\big(\frac{\kappa_y^{6.25}}{\epsilon^{1.25}}\big)$ for the nonconvex-strongly-convex setting, where $U, \widetilde{L}_{xy}$ and ρ_{yy} are Lipschitz parameters of objective functions (see Section 2.1). Ji et al. (2021) further improved the bound for the nonconvex-strongly-convex setting to $\mathcal{O}\big(\frac{\kappa_y^4}{\epsilon}\big)$.

In this paper, we address several open and important questions about bilevel optimization. We first observe that the existing complexity results on **bilevel** optimization are much worse than those on **minimax** optimization, which is a special case of bilevel optimization with $f(x, y) = g(x, y)$. For example, for the convex-strongly-convex case, it was shown in Lin et al. (2020) that the optimal complexity for **minimax** optimization is given by $\widetilde{\mathcal{O}}\big(\frac{\kappa_y^{0.5}}{\epsilon^{0.5}}\big)$, which is much smaller than the best known $\widetilde{\mathcal{O}}\big(\frac{\kappa_y^{6.75}}{\epsilon^{0.75}}\big)$ for **bilevel** optimization. Similar observations hold for the strongly-convex-strongly-convex setting. Therefore, one fundamental question arises.

1. *What is the performance limit of bilevel optimization in terms of computational complexity? Whether bilevel optimization is provably more challenging (i.e., requires more computations) than minimax optimization?*

Furthermore, existing analyses reply on a strong assumption on the boundedness of the outer-level gradient $\nabla_y f(x,\cdot)$[1] to guarantee that the smoothness parameter of $\Phi(\cdot)$ and the

1. Grazzi et al. (2020) assume that the inner-problem solution $y^*(x)$ is uniformly bounded for all x so that $\nabla_y f(x, y^*(x))$ is bounded.

hyperparameter estimation error are bounded as the algorithm runs. Then the following question needs to be addressed.

2. *Can we design a new bilevel optimization algorithm, which provably converges without the gradient boundedness? If so, whether such an algorithm achieves the optimal computational complexity?*

In addition, even when the boundedness assumption holds, existing complexity bounds show pessimistic complexity dependences on the condition numbers, e.g., $\mathcal{O}(\kappa_y^{6.75})$ for the convex-strongly-convex case. Then, the following question arises.

3. *Under the bounded gradient assumption, can we provide new upper bounds with tighter complexity dependences on the condition numbers for strongly-convex-strongly-convex and convex-strongly-convex bilevel optimizations?*

In this paper, we provide affirmative answers to the above questions.

1.1 Summary of Contributions

Our main contributions lie in developing several new results for bilevel optimization, including the first-known lower bounds on the computational complexity, a new convergence analysis without the gradient boundedness assumption, and significantly tighter upper bounds for bilevel optimization under different geometries. Our upper bounds meet the lower bounds in various cases, suggesting the tightness of the lower bounds and the optimality of the proposed algorithms. We summarize our results as follows.

- We provide the first-known lower bound of $\widetilde{\Omega}\Big(\sqrt{\frac{L_y\widetilde{L}_{xy}^2}{\mu_x\mu_y^2}}\Big)$ for solving the strongly-convex-strongly-convex bilevel optimization. We then propose a new accelerated bilevel optimizer named AccBiO. In contrast to existing bilevel optimizers, we show that AccBiO converges to the ϵ-accurate solution without the requirement on the boundedness of the gradient $\nabla_y f(x,\cdot)$ for any x. In particular, Table 1 shows that AccBiO achieves an upper complexity bound of $\widetilde{\mathcal{O}}\Big(\sqrt{\frac{L_y\widetilde{L}_{xy}^2\widetilde{L}_y}{\mu_x\mu_y^3}}+\sqrt{\frac{\widetilde{L}_{xy}^2\rho_{xy}\widetilde{L}_y}{\mu_y^4}}Gr^*+\sqrt{\frac{\widetilde{L}_{xy}^3\rho_{yy}L_y\widetilde{L}_y}{\mu_x\mu_y^5}}Fr^*\Big)$. When the inner-level function $g(x,y)$ takes the quadratic form as $g(x,y)=y^THy+x^TJy+b^Ty+h(x)$, we further improve the upper bounds to $\widetilde{\mathcal{O}}\Big(\sqrt{\frac{L_y\widetilde{L}_{xy}^2\widetilde{L}_y}{\mu_x\mu_y^3}}\Big)$. For such a quadratic subclass of bilevel problems with $\widetilde{L}_y=\Theta(\mu_y)$, our upper bound matches the lower bound up to logarithmic factors, suggesting that AccBiO is near-optimal. Technically, our analysis of the lower bound involves careful construction of quadratic f and g functions with a properly structured bilinear term, as well as novel characterization of subspaces of iterates for updating x and y. For upper bounds, our analysis controls the finiteness of all iterates $x_k, k=0,....$ as the algorithm runs via an induction proof to ensure that the hypergradient estimation error will not explode after the acceleration steps.
- We next provide lower and upper bounds for solving convex-strongly-convex bilevel optimization. As shown in Table 1, AccBiO achieves an upper bound of $\widetilde{\mathcal{O}}\Big(\sqrt{\frac{L_y\widetilde{L}_{xy}^2\widetilde{L}_y}{\epsilon\mu_y^3}}+\sqrt{\frac{\widetilde{L}_{xy}^2\rho_{xy}\widetilde{L}_y}{\epsilon\mu_y^4}}\widetilde{G}r^*+\sqrt{\frac{\widetilde{L}_{xy}^3\rho_{yy}L_y\widetilde{L}_y}{\epsilon\mu_y^5}}\widetilde{F}_r^*\Big)$, which is further improved to $\widetilde{\mathcal{O}}\Big(\sqrt{\frac{L_y\widetilde{L}_{xy}^2\widetilde{L}_y}{\epsilon\mu_y^3}}\Big)$ for the quadratic

Table 1: Comparison of computational complexities for finding an ϵ-approximate point **without the gradient boundedness assumption**. All listed results are from this paper, as all existing results were developed under the gradient boundedness condition, which we compare in Table 2. The complexity is measured by $\tau(n_J + n_H) + n_G$ (Theorem 3), where n_G, n_J, n_H are the numbers of gradients, Jacobian- and Hessian-vector products, and τ is a universal constant. In the 'references' column, quadratic $g(x,y)$ means that g takes a quadratic form as $g(x,y) = y^T Hy + x^T Jy + b^T y + h(x)$ for the constant matrices H, J and a constant vector b. 'LowerB' represents 'lower bound'. In the 'computational complexity' column, L_y, $\widetilde{L}_y$, $\widetilde{L}_{xy}$, ρ_{xy} and ρ_{yy} are the Lipschitz parameters of objective functions as shown in Section 2.1. $Gr^* = \sqrt{\frac{\|\nabla_y f(x^*, y^*(x^*))\|}{\mu_x}}$, and $Fr^* = \Big(\frac{2}{\mu_x}(\Phi(0) - \Phi(x^*)) + \|x^*\|^2 + \frac{\epsilon}{\mu_x}\Big)^{\frac{1}{4}}$, $\widetilde{G}r^*$ and $\widetilde{F}r^*$ take the same form as Gr^* and Fr^* but with x^*, f, Φ, μ_x replaced by $\widetilde{x}^*$, $\widetilde{f}$, $\widetilde{\Phi}$ and $\frac{\epsilon}{R}$ (see Section 4.3 for details). The lower bounds hold for both the general and quadratic $g(x,y)$ cases. By '$L_y = \Theta(\mu_y)$', we mean that there exists universal constants $C_1, C_2 > 0$ that is independent of μ_y s.t. $C_1\mu_y < L_y < C_2\mu_y$.

Types	**References**	**Computational Complexity**
Strongly-Convex-Strongly-Convex	**AccBiO** (Theorem 9)	$\widetilde{\mathcal{O}}\Big(\sqrt{\frac{L_y\widetilde{L}_{xy}^2\widetilde{L}_y}{\mu_x\mu_y^3}} + \sqrt{\frac{\widetilde{L}_{xy}^2\rho_{xy}\widetilde{L}_y}{\mu_y^4}}Gr^* + \sqrt{\frac{\widetilde{L}_{xy}^3\rho_{yy}L_y\widetilde{L}_y}{\mu_x\mu_y^5}}Fr^*\Big)$
	AccBiO (quadratic g, Coro 10)	$\widetilde{\mathcal{O}}\Big(\sqrt{\frac{L_y\widetilde{L}_{xy}^2\widetilde{L}_y}{\mu_x\mu_y^3}}\Big)$
	LowerB (Theorem 4)	$\widetilde{\Omega}\Bigg(\sqrt{\frac{L_y\widetilde{L}_{xy}^2}{\mu_x\mu_y^2}}\Bigg)$
Convex-Strongly-Convex	**AccBiO** (Theorem 11)	$\widetilde{\mathcal{O}}\Big(\sqrt{\frac{L_y\widetilde{L}_{xy}^2\widetilde{L}_y}{\epsilon\mu_y^3}} + \sqrt{\frac{\widetilde{L}_{xy}^2\rho_{xy}\widetilde{L}_y}{\epsilon\mu_y^4}}\widetilde{G}r^* + \sqrt{\frac{\widetilde{L}_{xy}^3\rho_{yy}L_y\widetilde{L}_y}{\epsilon\mu_y^5}}\widetilde{F}_r^*\Big)$
	AccBiO (quadratic g, Coro 12)	$\widetilde{\mathcal{O}}\Big(\sqrt{\frac{L_y\widetilde{L}_{xy}^2\widetilde{L}_y}{\epsilon\mu_y^3}}\Big)$
	LowerB (Coro 7, $\widetilde{L}_y = \Theta(\mu_y)$)	$\widetilde{\Omega}\Big(\sqrt{\frac{\widetilde{L}_{xy}^2 L_y}{\mu_y^2\epsilon}}\Big)$
	LowerB (Coro 8, $\widetilde{L}_y = \Theta(1)$)	$\widetilde{\Omega}(\frac{1}{\sqrt{\epsilon}}\min\{\kappa_y, \frac{1}{\sqrt{\epsilon^3}}\})$

$g(x,y)$. For such a quadratic case with $\widetilde{L}_y = \Theta(\mu_y)$, our upper bound matches the lower bound up to logarithmic factors, suggesting the optimality of AccBiO. Technically, the analysis of the lower bound is different from that for the strongly-convex $\Phi(\cdot)$, and exploits the structures of different powers of an unnormalized graph Laplacian matrix Z.

- Furthermore, when the gradient $\nabla_y f(x,\cdot)$ is bounded, as assumed by existing studies, we provide new upper bounds with significantly tighter dependence on the condition numbers. For example, as shown in Table 2, our upper bounds outperform the best known result by an order of $\frac{\kappa_y^{4.75}}{\epsilon^{0.25}}$ for the convex-strongly-convex case.

- To compare between bilevel optimization and minimax optimization, for the strongly-convex-strongly-convex case, our lower bound is larger than the optimal complexity for the same type of minimax optimization by a factor of $\sqrt{\kappa_y}$. Similar observation

Table 2: Comparison of computational complexities for finding an ϵ-approximate point **with the gradient boundedness assumption**. U is the gradient bound.

Types	**References**	**Computational Complexity**
Strongly-Convex-Strongly-Conconvex	BA (Ghadimi and Wang, 2018)	$\widetilde{\mathcal{O}}\Big(\max\Big\{\frac{U^2\widetilde{L}_{xy}^4\rho_{yy}^2}{\mu_x^2\mu_y^6}, \frac{\widetilde{L}_y^2}{\mu_y^2}\Big\}\Big)$
	ABA (Ghadimi and Wang, 2018)	$\widetilde{\mathcal{O}}\Big(\max\Big\{\frac{U\widetilde{L}_{xy}^2\rho_{yy}}{\mu_x\mu_y^3}, \frac{\widetilde{L}_y^2}{\mu_y^2}\Big\}\Big)$
	AccBiO-BG (this paper, Theorem 13)	$\widetilde{\mathcal{O}}\Big(\sqrt{\frac{U\widetilde{L}_{xy}^2\rho_{yy}\widetilde{L}_y}{\mu_x\mu_y^4}}\Big)$
Convex-Strongly-Convex	BA (Ghadimi and Wang, 2018)	$\widetilde{\mathcal{O}}\Big(\frac{\kappa_y^{11.25}}{\epsilon^{1.25}}\Big)$
	ABA (Ghadimi and Wang, 2018)	$\widetilde{\mathcal{O}}\Big(\frac{\kappa_y^{6.75}}{\epsilon^{0.75}}\Big)$
	AccBiO-BG (this paper, Theorem 14)	$\widetilde{\mathcal{O}}\Big(\sqrt{\frac{\widetilde{L}_{xy}^2\rho_{yy}\widetilde{L}_y}{\epsilon\mu_y^4}}\Big)$

holds for the convex-strongly-convex case. This establishes that bilevel optimization is fundamentally more challenging than minimax optimization.

1.2 Related Works

The studies of bilevel optimization problems and algorithms can be dated back to Bracken and McGill (1973), and since then, different types of approaches have been proposed. Earlier approaches in Aiyoshi and Shimizu (1984); Edmunds and Bard (1991); Al-Khayyal et al. (1992); Hansen et al. (1992); Shi et al. (2005); Lv et al. (2007); Moore (2010) reduced the bilevel problem to a single-level optimization problem using the Karush-Kuhn-Tucker (KKT) conditions or penalty function methods. In comparison, gradient-based approaches are more attractive due to their efficiency and effectiveness. Such a type of approaches estimate the hypergradient $\nabla\Phi(x)$ for iterative updates, and are generally divided into AID- and ITD-based categories. ITD-based approaches (Maclaurin et al., 2015; Franceschi et al., 2017; Finn et al., 2017; Grazzi et al., 2020) estimate the hypergradient $\nabla\Phi(x)$ in either a reverse (automatic differentiation) or forward manner. AID-based approaches (Domke, 2012; Pedregosa, 2016; Grazzi et al., 2020; Ji et al., 2021) estimate the hypergradient via implicit differentiation.

Theoretically, bilevel optimization has been studied via both the asymptotic and finite-time (non-asymptotic) analysis. Franceschi et al. (2018) characterized the asymptotic convergence of a backpropagation-based approach as one of ITD-based algorithms by assuming the inner-level problem is strongly convex. Shaban et al. (2019) provided a similar analysis for a truncated backpropagation scheme. Liu et al. (2020); Li et al. (2020); Sow et al. (2022) analyzed the asymptotic performance of bilevel approaches when the inner-level problem is convex. The finite-time complexity analysis for bilevel optimization has also been explored. In particular, Ghadimi and Wang (2018) provided a finite-time convergence analysis for an AID-based algorithm under two different loss geometries, where $\Phi(\cdot)$ is strongly convex, convex or nonconvex, and $g(x, \cdot)$ is strongly convex. Ji et al. (2021, 2022) provided an improved finite-time analysis for AID- and ITD-based algorithms under the nonconvex-strongly-convex geometry. In this paper, we provide the first-known lower bounds on complexity as well as tighter upper bounds under these two geometries.

When the objective functions can be expressed in an expected or finite-time form, Ghadimi and Wang (2018); Ji et al. (2021); Hong et al. (2020) developed stochastic bilevel algorithms and provided the finite-time analysis. In particular, Ji et al. (2021) proposed a SGD type of bilevel optimization algorithm named stocBiO with a sample efficient hypergradient estimator. Since then, there have been a few subsequent studies on accelerating SGD-type bilevel optimization via momentum-based variance reduction Chen et al. (2021); Guo et al. (2021); Khanduri et al. (2021); Yang et al. (2021); Huang and Huang (2021). For example, Guo et al. 2021 proposed a single-loop algorithm SEMA based on the momentum-based technique introduced by Cutkosky and Orabona 2019. Chen et al. 2021 proposed a single-loop method named STABLE by using a similar momentum scheme for the Hessian updates. Yang et al. 2021 improved the sample complexity of stocBiO via both single-loop and double-loop variance reduction. While the stochastic setting is not within the scope of this paper, the accelerating algorithms and lower bounds developed here can be extended to the stochastic setting.

Bilevel optimization has been applied to meta-learning and led to various algorithms such as model-agnostic meta-learning (MAML) (Finn et al., 2017), implicit MAML (iMAML) (Rajeswaran et al., 2019), and almost no inner loop (ANIL) (Raghu et al., 2019). Theoretically, Rajeswaran et al. (2019) analyzed the complexity of iMAML via implicit differentiation under the strongly-convex setting. Ji et al. (2020b); Fallah et al. (2020) characterized the convergence of MAML under the nonconvex function geometry. Ji et al. (2020a) analyzed the convergence and complexity of ANIL with either strongly-convex or nonconvex inner-level geometries.

Bilevel optimization has been applied to study various machine learning problems. For example, bilevel optimization has exhibited great effectiveness in hyperparameter optimization, and received tremendous attention recently in automatic machine learning (autoML) (Okuno et al., 2018; Yu and Zhu, 2020; Ji and Liang, 2018). A variety of bilevel optimization algorithms have been proposed for this area, which include but not limited to AID-based (Pedregosa, 2016; Franceschi et al., 2018), ITD-based (Franceschi et al., 2018; Shaban et al., 2019; Grazzi et al., 2020), self-tuning network based (Mackay et al., 2018; Bae and Grosse, 2020), penalty-based (Mehra and Hamm, 2019; Sinha et al., 2020; Liu et al., 2021), and proximal approximation based (Jenni and Favaro, 2018) approaches. Bilevel optimization has also been exploited to improve the search efficiency for neural architecture search (NAS) (Liu et al., 2019; Xie et al., 2018; He et al., 2020). For example, Liu et al. (2019) proposed a continuous relaxation of the discrete architecture representation, and tremendously accelerated the architecture search via a gradient-based bilevel optimization method named DARTS. Xie et al. (2018) further proposed a new stochastic reformulation of NAS coupled with a sampling process to address the bias issue of DARTS. He et al. (2020) reformulated the bilevel objective function of NAS into a mixed-level optimization procedure, and proposed an efficient MiLeNAS method with a lower validation error. We anticipate that the proposed acceleration schemes will be useful for the aforementioned applications.

2. Preliminaries on Bilevel Optimization

2.1 Bilevel Problem Class

In this section, we introduce the problem class we are interested in. First, we suppose functions $f(x,y)$ and $g(x,y)$ satisfy the following smoothness property.

Assumption 1 *The outer-level function f satisfies, for $\forall x_1, x_2, x \in \mathbb{R}^p$ and $y_1, y_2, y \in \mathbb{R}^q$, there exist constants $L_x, L_{xy}, L_y \geq 0$ such that*

$$\begin{aligned}\|\nabla_x f(x_1,y) - \nabla_x f(x_2,y)\| \leq & L_x\|x_1 - x_2\|, \ \|\nabla_x f(x,y_1) - \nabla_x f(x,y_2)\| \leq L_{xy}\|y_1 - y_2\| \\ \|\nabla_y f(x_1,y) - \nabla_y f(x_2,y)\| \leq & L_{xy}\|x_1 - x_2\|, \ \|\nabla_y f(x,y_1) - \nabla_y f(x,y_2)\| \leq L_y\|y_1 - y_2\|. \end{aligned} \tag{2}$$

The inner-level function g satisfies that, there exist $\widetilde{L}_{xy}, \widetilde{L}_y \geq 0$ such that

$$\|\nabla_y g(x_1,y) - \nabla_y g(x_2,y)\| \leq \widetilde{L}_{xy}\|x_1 - x_2\|, \ \|\nabla_y g(x,y_1) - \nabla_y g(x,y_2)\| \leq \widetilde{L}_y\|y_1 - y_2\|. \tag{3}$$

The hypergradient $\nabla\Phi(x)$ plays an important role for designing bilevel optimization algorithms. The computation of $\nabla\Phi(x)$ involves Jacobians $\nabla_x\nabla_y g(x,y)$ and Hessians $\nabla_y^2 g(x,y)$. In this paper, we are interested in the following inner-level problem with general Lipschitz continuous Jacobians and Hessians, as adopted by Ghadimi and Wang (2018); Ji et al. (2021); Hong et al. (2020). For notational convenience, let $z := (x,y)$ denote both variables.

Assumption 2 *There exist constants $\rho_{xy}, \rho_{yy} \geq 0$ such that for any $z_1 \in \mathbb{R}^p \times \mathbb{R}^q, z_2 \in \mathbb{R}^p \times \mathbb{R}^q$,*

$$\|\nabla_x\nabla_y g(z_1) - \nabla_x\nabla_y g(z_2)\| \leq \rho_{xy}\|z_1 - z_2\|, \ \ \|\nabla_y^2 g(z_1) - \nabla_y^2 g(z_2)\| \leq \rho_{yy}\|z_1 - z_2\|. \tag{4}$$

In this paper, we study the following two classes of bilevel optimization problems.

Definition 1 (Bilevel Problem Classes) *Suppose f and g satisfy Assumptions 1, 2. We define the following two classes of bilevel problems under different geometries.*

- **Strongly-convex-strongly-convex class** $\mathcal{F}_{scsc}$: $\Phi(\cdot)$ *is μ_x-strongly-convex and $g(x,\cdot)$ is μ_y-strongly-convex.*
- **Convex-strongly-convex class** $\mathcal{F}_{csc}$: $\Phi(\cdot)$ *is convex and $g(x,\cdot)$ is μ_y-strongly-convex. In addition, assume that there exists a constant $B > 0$ such that $\|x^*\| = B$, where $x^* \in \arg\min_{x\in\mathbb{R}^p}\Phi(x)$.*

A simple but important subclass of the bilevel problem class in Theorem 1 includes the following quadratic inner-level functions $g(x,y)$.

$$\text{(Quadratic } g \text{ subclass:)} \quad g(x,y) = \frac{1}{2}y^T H y + x^T J y + b^T y + h(x), \tag{5}$$

where the Hessian H and the Jacobian J satisfy $H \preceq \widetilde{L}_y I$ and $J \preceq \widetilde{L}_{xy} I$ for $\forall x \in \mathbb{R}^p$ and $\forall y \in \mathbb{R}^q$. Note that the above quadratic subclass also covers a large collection of applications such as few-shot meta-learning with shared embedding model (Bertinetto et al., 2018) and biased regularization in hyperparameter optimization (Grazzi et al., 2020). Further note that it is possible to extend the **constant** coupling matrix H of the quadratic subclass to the more general matrix $H(x)$ that is Lipschitz continuous, i.e., $\|H(x_1)-H(x_2)\| \leq L_H\|x_1-x_2\|$ for a constant L_H.

Another convex-strongly-convex example of practical interest is the fair resource allocation problem over communication networks (Srikant and Ying, 2013). Consider a network with n users/devices. Each user i is associated with a convex cost function $C(x_i)$ (or $U(x_i)$ for utility function U), where x_i is the resource (e.g., the package transmission rate). In the network implementation, $C(x_i)$ is fabricated to be a simple function, e.g., α-fairness. Let $x = [x_1, x_2, ..., x_n]^T$. Then, the lower-level strongly-convex problem is given by

$$x^*(\lambda) = \arg\max_x g(x,\lambda) = \sum_{i=1}^{n} C(x_i) + \frac{\lambda}{2}\|x\|^2, \tag{6}$$

where $\lambda\|x\|^2$ is used for capacity and traffic control. However, the underlying true network cost function can be more complicated, e.g., the aggregation loss $\widetilde{C}(x) = \sum_{i=1}^n \tau_i C(x_i)$. Then, the upper-level problem is to find the best regularization constant λ that minimizes the true cost

$$\max_\lambda \widetilde{C}(x^*(\lambda)) = \sum_{i=1}^{n} \tau_i C(x_i^*(\lambda)).$$

We next justify that the total function $\widetilde{C}(x^*(\lambda))$ is convex w.r.t. λ. Using the implicit differentiation and the optimality condition of x_i^* , it can be shown that

$$\begin{aligned} \nabla C(x_i^*) + \lambda x_i^* &= 0 \\ (\nabla^2 C(x_i^*) + \lambda)\frac{\partial x_i^*}{\partial \lambda} + x_i^* &= 0, \end{aligned}$$

which, in conjunction with the second-order derivative of the total upper-level function $\frac{\partial^2 \widetilde{C}(x^*(\lambda))}{\partial^2 \lambda} = \sum_{i=1}^n \big(\tau_i \nabla^2 C(x_i^*)(\frac{\partial x_i^*}{\partial \lambda})^2 + \tau_i \nabla C(x_i^*)\frac{\partial x_i^*}{\partial \lambda}\big)$, yields

$$\frac{\partial^2 \widetilde{C}(x^*(\lambda))}{\partial^2 \lambda} = \sum_{i=1}^{n} \Big(\tau_i \nabla^2 C(x_i^*)\Big(\frac{\partial x_i^*}{\partial \lambda}\Big)^2 + \frac{\tau_i (\nabla C(x_i^*))^2}{\lambda(\nabla^2 C(x_i^*) + \lambda)}\Big) \geq 0.$$

This implies that the total objective function $\widetilde{C}(x^*(\lambda))$ is **convex** w.r.t. λ. Besides, there are some quasiconvex-strongly-convex and convex-strongly-convex examples in batch selection problems for model fairness (Roh et al., 2021).

2.2 Algorithm Class for Bilevel Optimization

Compared to **minimization** and **minimax** problems, the most different and challenging component of **bilevel optimization** lies in the computation of the *hypergradient* $\nabla\Phi(\cdot)$. In specific, when functions f and g are continuously twice differentiable, it has been shown in Foo et al. (2008) that $\nabla\Phi(\cdot)$ takes the form of

$$\nabla\Phi(x) =\nabla_x f(x, y^*(x)) - \nabla_x\nabla_y g(x, y^*(x))[\nabla_y^2 g(x, y^*(x))]^{-1}\nabla_y f(x, y^*(x)). \tag{7}$$

In practice, exactly calculating the Hessian inverse $(\nabla_y^2 g(\cdot))^{-1}$ in eq. (7) is computationally infeasible, and hence two types of hypergradient estimation approaches named AID and ITD have been proposed, where only efficient **Hessian- and Jacobian-vector products** need to be computed. We present AID-based bilevel optimization algorithms as follows, and the introduction of ITD-based methods can be found in Appendix A.

Example 1 (AID-based Bilevel Algorithms) *(Domke, 2012; Pedregosa, 2016; Grazzi et al., 2020; Ji et al., 2021) Such a class of algorithms use AID-based approaches for hypergradient computation, and take the following updates.*

For each outer iteration $m = 0, ..., Q-1$,

- *Update variable y using gradient decent (GD) or accelerated gradient descent (AGD)*

$$\begin{aligned} &\text{(GD:)} \quad y_m^t = y_m^{t-1} - \eta \nabla_y g(x_m, y_m^{t-1}), t = 1, ..., N \\ &\text{(AGD:)} \quad y_m^t = z_m^{t-1} - \eta \nabla_y g(x_m, z_m^{t-1}), \\ &\qquad\qquad z_m^t = \Big(1 + \frac{\sqrt{\kappa_y}-1}{\sqrt{\kappa_y}+1}\Big) y_m^t - \frac{\sqrt{\kappa_y}-1}{\sqrt{\kappa_y}+1} y_m^{t-1}, t = 1, ..., N \end{aligned} \tag{8}$$

where $\kappa_y = \widetilde{L}_y/\mu_y$ denotes the condition number of the inner-level function $g(x,\cdot)$.

- *Update x via $x_{m+1} = x_m - \beta G_m$, where G_m is constructed via AID and takes the form of*

$$G_m = \nabla_x f(x_m, y_m^N) - \nabla_x \nabla_y g(x_m, y_m^N) v_m^S, \tag{9}$$

where vector v_m^S is obtained by running S steps of GD (with initialization $v_m^0 = 0$) or accelerated gradient methods (e.g., heavy-ball method with $v_m^0 = v_m^1 = 0$) to solve a quadratic programming

$$\min_v Q(v) := \frac{1}{2} v^T \nabla_y^2 g(x_m, y_m^N) v - v^T \nabla_y f(x_m, y_m^N). \tag{10}$$

We next verify that Example 1 belongs to the algorithm class defined in Theorem 2. For the case when S-steps GD with initialization $\mathbf{0}$ is applied to solve the quadratic program in eq. (10), simple telescoping yields

$$v_m^S = \alpha \sum_{t=0}^{S-1} (I - \alpha \nabla_y^2 g(x_m, y_m^N))^t \nabla_y f(x_m, y_m^N),$$

which, incorporated into eq. (9), implies that G_m falls into the span subspaces in eq. (13), and hence all updates fall into the subspaces $\mathcal{H}_x^k, \mathcal{H}_y^k, k = 0, ..., K$ defined in Theorem 2. For the case when heavy-ball method, i.e., $v_m^{t+1} = v_m^t - \eta_t \nabla Q(v_m^t) + \theta_t (v_m^t - v_m^{t-1})$, with initialization $v_m^0 = v_m^1 = \mathbf{0}$ is applied to eq. (10), expressing the updates via a dynamic system perspective yields

$$\begin{bmatrix} v_m^S \\ v_m^{S-1} \end{bmatrix} = \sum_{s=2}^{S} \prod_{t=s}^{S-1} \begin{bmatrix} (1+\theta_t) I - \eta_t \nabla_y^2 g(x_m, y_m^N) & -\theta_t I \\ I & \mathbf{0} \end{bmatrix} \begin{bmatrix} \eta_t \nabla_y f(x_m, y_m^N) \\ \mathbf{0} \end{bmatrix}. \tag{11}$$

Combining v_m^S in eq. (11) with eq. (9), we can see that the resulting G_m falls into the span subspaces in eq. (13), and hence this case still belongs to the algorithm class in Theorem 2. Note that the algorithm class considered in Theorem 2 also includes single-loop bilevel optimization algorithms, e.g., by setting $N = 1$ in Example 1.

We next introduce a general hypergradient-based algorithm class, which includes popular AID-based (given above) and ITD-based (in Appendix A) bilevel optimization algorithms.

Definition 2 (Hypergradient-Based Algorithm Class) *Suppose there are totally K iterations and x is updated for Q times at iterations indexed by $s_i, i = 1, ..., Q-1$ with $s_0 < ... < s_{Q-1} \leq K$. Note that Q is an arbitrary positive integer in $0, ..., K$ and $s_i, i = 1, ..., Q-1$ are Q arbitrary distinct integers in $0, ..., K$. The iterates $\{(x_k, y_k)\}_{k=0,...,K}$ are generated according to $(x_k, y_k) \in \mathcal{H}_x^k, \mathcal{H}_y^k$, where the linear subspaces $\mathcal{H}_x^k, \mathcal{H}_y^k, k = 0, ..., K$ with $\mathcal{H}_x^0 = \mathcal{H}_y^0 = \{\mathbf{0}\}$ are given as follows.*

$$H_y^{k+1} = \operatorname{Span}\left\{y_i, \nabla_y g(\widetilde{x}_i, \widetilde{y}_i), \forall \widetilde{x}_i \in \mathcal{H}_x^i, \forall y_i, \widetilde{y}_i \in \mathcal{H}_y^i, 1 \leq i \leq k\right\}. \tag{12}$$

For x, we have, for all $m = 0, ..., Q-1$,

$$\begin{aligned}
\mathcal{H}_x^{s_m} = \operatorname{Span}\Big\{& x_i, \nabla_x f(\widetilde{x}_i, \widetilde{y}_i), \nabla_x \nabla_y g(x_i^t, y_i^t) \prod_{j=1}^{t} (I - \alpha \nabla_y^2 g(x_{i,j}^t, y_{i,j}^t)) \nabla_y f(\hat{x}_i, \hat{y}_i), \\
& t = 0, ..., T, \forall x_i, \hat{x}_i, x_i^t, x_{i,j}^t \in \mathcal{H}_x^i, \forall \hat{y}_i, y_i^t, y_{i,j}^t \in \mathcal{H}_y^i, 1 \leq i \leq s_m - 1, \forall \alpha \in \mathbb{R}, T \in \mathbb{N}\Big\} \\
\mathcal{H}_x^n = \mathcal{H}_x^{s_m}&, \forall s_m \leq n \leq s_{m+1} - 1 \text{ with } s_Q = K + 1.
\end{aligned} \tag{13}$$

We note that this algorithm class in Theorem 2 can be enlarged by replacing $\nabla_y f(x, y)$ in eq. (13) with $\operatorname{Span}(y_1, \nabla_y g(x_2, y_2), \nabla_y f(x_3, y_3))$, where y_1, x_2, y_2, x_3, y_3 can be any points in previous iterations.

Further note that in the algorithm class in Theorem 2, x can be updated at any iteration due to the arbitrary choices of $Q, s_i, i = 1, ..., Q-1$, and the hypergradient estimate can be constructed using any combination of points in the historical search space (which holds similarly for y). Moreover, this algorithm class allows to update x and y at the same time or alternatively, and hence include both single- and double-loop bilevel optimization algorithms. Note that the above hypergradient-based algorithm class include popular examples such as HOAG (Pedregosa, 2016), AID-FP (Grazzi et al., 2020), reverse (Franceschi et al., 2017), K-RMD (Shaban et al., 2019), AID-BiO and ITD-BiO (Ji et al., 2021).

2.3 Complexity Measures

We introduce the criterion for measuring the computational complexity of bilevel optimization algorithms. Note that the updates of x and y of bilevel algorithms involve computing gradients, Jacobian- and Hessian-vector products. In practice, it has been shown in Griewank (1993); Rajeswaran et al. (2019) that the time and memory cost for computing a Hessian-vector product $\nabla^2 f(\cdot)v$ (similarly for a Jacobian-vector product) via automatic differentiation (e.g., the widely-used reverse mode in PyTorch or TensorFlow) is no more than a (universal) constant order (e.g., usually 2-5 times) over the cost for computing gradient $\nabla f(\cdot)$. For this reason, we take the following complexity measures.

Definition 3 (Complexity Measure) *The total complexity $\mathcal{C}_{\text{fun}}(\mathcal{A},\epsilon)$ of a bilevel optimization algorithm $\mathcal{A}$ to find a point $\bar{x}$ such that the suboptimality gap $f(\bar{x})-\min_x f(x)\le\epsilon$ is given by $\mathcal{C}_{\text{fun}}(\mathcal{A},\epsilon)=\tau(n_J+n_H)+n_G$, where n_J,n_H and n_G are the total numbers of Jacobian- and Hessian-vector product, and gradient evaluations, and $\tau>0$ is a universal constant. Similarly, we define $\mathcal{C}_{\text{grad}}(\mathcal{A},\epsilon)=\tau(n_J+n_H)+n_G$ as the complexity to find a point $\bar{x}$ such that the gradient norm $\|\nabla f(\bar{x})\|\le\epsilon$.*

3. Lower Bounds for Bilevel Optimization

3.1 Strongly-Convex-Strongly-Convex Bilevel Optimization

We first study the case when $\Phi(\cdot)$ is μ_x-strongly-convex and the inner-level function $g(x,\cdot)$ is μ_y-strongly-convex. We present our lower bound result for this case as below.

Theorem 4 *Let $M=K+QT+Q+2$ with K,T,Q given by Theorem 2. There exists a problem instance in $\mathcal{F}_{scsc}$ defined in Theorem 1 with dimensions $p=q=d$ as in eq. (40) such that for this problem, any output x^K belonging to the subspace $\mathcal{H}_x^K$, i.e., generated by any algorithm in the hypergradient-based algorithm class defined in Theorem 2, satisfies*

$$\Phi(x^K)-\Phi(x^*)=\Omega\Big(\frac{\Phi(x_0)-\Phi(x^*)}{\kappa_x}r^{2M}\Big), \tag{14}$$

where $x^*=\arg\min_{x\in\mathbb{R}^d}\Phi(x)$ and the parameter r satisfies $1-\Big(\frac{1}{2}+\sqrt{\xi+\frac{1}{4}}\Big)^{-1}<r<1$ with ξ given by $\xi\ge\frac{\widetilde{L}_y}{4\mu_y}+\frac{L_x}{8\mu_x}+\frac{L_y\widetilde{L}_{xy}^2}{8\mu_x\mu_y^2}-\frac{3}{8}$. To achieve $\Phi(x^K)-\Phi(x^*)\le\epsilon$, the total complexity $\mathcal{C}_{\text{fun}}(\mathcal{A},\epsilon)$ satisfies

$$\mathcal{C}_{\text{fun}}(\mathcal{A},\epsilon)=\Omega\Bigg(\sqrt{\frac{L_y\widetilde{L}_{xy}^2}{\mu_x\mu_y^2}}\log\frac{\Phi(x_0)-\Phi(x^*)}{\kappa_x\epsilon}\Bigg),$$

where $\kappa_x=\frac{L_\Phi}{\mu_x}$ is the condition number for the total objective function.

Note that the inner-level function $g(x,y)$ in our constructed worst-case instance takes the same quadratic form as in eq. (5) so that the lower bound in Theorem 4 also applies to the quadratic g subclass. We provide a proof sketch of Theorem 4 as follows, and present the complete proof in Appendix B.

Proof Sketch of Theorem 4

The proof of Theorem 4 is divided into four main steps: 1) constructing a worst-case instance $(f,g)\in\mathcal{F}_{scsc}$; 2) characterizing the optimal point $x^*=\arg\min_{x\in\mathbb{R}^d}\Phi(x)$; 3) characterizing the subspaces $\mathcal{H}_x^K,\mathcal{H}_y^K$; and 4) lower-bounding the convergence rate and complexity.

Step 1 (construct a worse-case instance): We first construct the following instance functions f and g.

$$f(x,y)=\frac{1}{2}x^T(\alpha Z^2+\mu_x I)x-\frac{\alpha\beta}{\widetilde{L}_{xy}}x^TZ^3y+\frac{\bar{L}_{xy}}{2}x^TZy+\frac{L_y}{2}\|y\|^2+\frac{\bar{L}_{xy}}{\widetilde{L}_{xy}}b^Ty,$$

$$g(x,y) = \frac{1}{2}y^T(\beta Z^2 + \mu_y I)y - \frac{\widetilde{L}_{xy}}{2}x^T Z y + b^T y, \tag{15}$$

where $\alpha = \frac{L_x - \mu_x}{4}$, $\beta = \frac{\widetilde{L}_y - \mu_y}{4}$, and the coupling matrices Z, Z^2, Z^4 take the forms of

$$Z = \begin{bmatrix} & & & 1 \\ & & 1 & -1 \\ & \iddots & \iddots & \\ 1 & -1 & & \end{bmatrix}, Z^2 = \begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}, Z^4 = \begin{bmatrix} 2 & -3 & 1 & & & \\ -3 & 6 & -4 & 1 & & \\ 1 & -4 & 6 & -4 & 1 & \\ & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & 1 & -4 & 6 & -4 \\ & & & 1 & -4 & 5 \end{bmatrix}. \tag{16}$$

The above matrices play an important role in developing lower bounds due to their following zero-chain properties (Nesterov, 2003; Zhang et al., 2019). Let $\mathbb{R}^{k,d} = \{x \in \mathbb{R}^d | x_i = 0 \text{ for } k+1 \leq i \leq d\}$, where x_i denotes the i^{th} coordinate of the vector x.

Lemma 5 (Zero-Chain Property) *For any given vector $v \in \mathbb{R}^{k,d}$, we have $Z^2 v \in \mathbb{R}^{k+1,d}$.*

Theorem 5 indicates that if a vector v has nonzero entries only at the first k coordinates, then multiplying it with a matrix Z^2 has at most one more nonzero entry at position $k+1$. We demonstrate the validity of the constructed instance by showing that f and g in eq. (15) satisfy Assumptions 1 and 2, and $\Phi(x)$ is μ_x-strongly-convex. The proof of these statements can be found in Step 1 in Appendix B.

Step 2 (characterize the minimizer x^*): We show that the unique minimizer x^* satisfies the following equation

$$Z^4 x^* + \lambda Z^2 x^* + \tau x^* = \gamma Z b, \tag{17}$$

where $\lambda = \Theta(1)$ and $\gamma = \Theta(1)$, $\tau = \Theta(\mu_x \mu_y^2)$ (see eq. (33) for its complete form). We choose b in eq. (17) such that $(Zb)_t = 0$ for all $t \geq 3$, which is feasible because we show that Z is invertible. Using the structure of Z in eq. (16), we show that there exists a vector $\hat{x}$ with its i^{th} coordinate $\hat{x}_i = r^i$ such that

$$\|x^* - \hat{x}\| = \mathcal{O}(r^d), \tag{18}$$

where $0 < r < 1$ satisfies $\frac{1-r}{=}\Theta(\frac{L_y \widetilde{L}_{xy}^2}{\mu_x \mu_y^2})$. Then, based on the above eq. (18), we are able to characterize x^*, e.g., its norm $\|x^*\|$, using its approximate (exponentially close) $\hat{x}$.

Step 3 (characterize the iterate subspaces): By exploiting the forms of the subspaces $\{\mathcal{H}_x^k, \mathcal{H}_y^k\}_{k=1}^K$ defined in Theorem 2, we use the induction to show that

$$H_x^K \subseteq \text{Span}\{Z^{2(K+QT+Q)}(Zb), ..., Z^2(Zb), (Zb)\}.$$

Then, noting that $(Zb)_t = 0$ for all $t \geq 3$ and using the zero-chain property of Z^2, we have the t^{th} coordinate of the output x^K to be zero, i.e., $(x^K)_t = 0$, for all $t \geq M+1$.

Step 4 (combine Steps $1, 2, 3$ and characterize the complexity): By choosing $d > \max\left\{2M, M+1+\log_r\left(\frac{\tau}{4(7+\lambda)}\right)\right\}$, and based on **Steps** 2 and 3, we have $\|x^K - x^*\| \geq$

$\frac{\|x^*-x_0\|}{3\sqrt{2}}r^M$ which, in conjunction with the form of $\Phi(x)$, yields the result in eq. (14). The complexity result then follows because $\frac{1}{1-r} = \Theta(\frac{L_y\widetilde{L}_{xy}^2}{\mu_x\mu_y^2})$ and from the definition of the complexity measure in Theorem 3.

Remark. We note that the introduction of the term $\frac{\alpha\beta}{\widetilde{L}_{xy}}x^TZ^3y$ in f is necessary to obtain the lower bound $\widetilde{\Omega}(\frac{L_y\widetilde{L}_{xy}^2}{\mu_x\mu_y^2})$. Without such a term, there will be an additional high-order term $\Omega(A^6x)$ at the left hand side of eq. (17). Then, following the same steps as in Step 2, we would obtain a result similar to eq. (18), but with a parameter r satisfying $0 < \frac{1}{1-r} = \Theta\big(\frac{L_y\widetilde{L}_{xy}}{\mu_x\mu_y}\big)$. Then, following the same steps as in Steps 3 and 4, the final overall complexity $\mathcal{C}_{\text{fun}}(\mathcal{A},\epsilon) = \Omega\big(\frac{L_y\widetilde{L}_{xy}}{\mu_x\mu_y}\big)$, which is not as tight as $\Omega\big(\frac{L_y\widetilde{L}_{xy}^2}{\mu_x\mu_y^2}\big)$ obtained under the selection in eq. (15).

3.2 Convex-Strongly-Convex Bilevel Optimization

We next characterize the lower complexity bound for the convex-strongly-convex setting, where $\Phi(\cdot)$ is **convex** and the inner-level function $g(x,\cdot)$ is μ_y-strongly-convex. We state our main result for this case in the following theorem.

Theorem 6 *Let $M = K + QT - Q + 3$ with K, T, Q given by Theorem 2, and let x^K be an output belonging to the subspace $\mathcal{H}_x^K$, i.e., generated by any algorithm in the hypergradient-based algorithm class defined in Theorem 2. There exists an instance in $\mathcal{F}_{csc}$ defined in Theorem 1 with dimensions $p = q = d$ such that in order to achieve $\|\nabla\Phi(x^K)\| \leq \epsilon$, it requires $M \geq \lfloor r^*\rfloor - 3$, where r^* is the solution of the equation*

$$r^4 + r\Big(\frac{2\beta^4}{\mu_y^4} + \frac{4\beta^3}{\mu_y^3} + \frac{4\beta^2}{\mu_y^2}\Big) = \frac{B^2(\widetilde{L}_{xy}^2L_y + L_x\mu_y^2)^2}{128\mu_y^4\epsilon^2}, \tag{19}$$

where $\beta = \frac{\widetilde{L}_y-\mu_y}{4}$ and B is given in Theorem 1. The complexity satisfies $\mathcal{C}_{\text{grad}}(\mathcal{A},\epsilon) = \Omega(r^*)$.

Note that Theorem 6 uses the gradient norm $\|\nabla\Phi(x)\| \leq \epsilon$ rather than the suboptimality gap $\Phi(x^K) - \Phi(x^*)$ as the convergence criteria. This is because for the convex-strongly-convex case, lower-bounding the suboptimality gap requires the Hessian matrix A in the worst-case construction of the total objective function $\Phi(x)$ to have a nice structure, e.g., the solution of $A'x = e_1$ (e_1 has a single non-zero value 1 at the first coordinate) is explicit, where A' is derived by removing last k columns and rows of A. However, in bilevel optimization, A often contains different powers of the zero-chain matrix Z, and does not have such a structure. We will leave the lower bound under the suboptimality criteria for the future study. Note that r^* in Theorem 6 has a complicated form. The following two corollaries simplify the complexity results by considering specific parameter regimes.

Corollary 7 *Under the same setting of Theorem 6, consider the case when $\beta = \Theta(\mu_y)$. Then, we have $\mathcal{C}_{\text{grad}}(\mathcal{A},\epsilon) = \Omega\Big(\frac{B^{\frac{1}{2}}(\widetilde{L}_{xy}^2L_y + L_x\mu_y^2)^{\frac{1}{2}}}{\mu_y\epsilon^{\frac{1}{2}}}\Big)$.*

Corollary 8 *Under the same setting of Theorem 6, consider the case when $\beta = \Theta(1)$, i.e., at a constant level. Then, we have $\mathcal{C}_{\text{grad}}(\mathcal{A},\epsilon) = \widetilde{\Omega}(\frac{1}{\sqrt{\epsilon}}\min\{\kappa_y, \frac{1}{\sqrt{\epsilon^3}}\})$.*

The proof sketch of Theorem 6 is provided as follows. The complete proof is provided in Appendix C.

Proof Sketch of Theorem 6

Step 1(construct the worst-case instance): We construct the instance functions f and g as follows.

$$\begin{aligned} f(x,y) &= \frac{L_x}{8}x^T Z^2 x + \frac{L_y}{2}\|y\|^2, \\ g(x,y) &= \frac{1}{2}y^T(\beta Z^2 + \mu_y I)y - \frac{\widetilde{L}_{xy}}{2}x^T Z y + b^T y, \end{aligned} \tag{20}$$

where $\beta = \frac{\widetilde{L}_y - \mu_y}{4}$. Here, the coupling matrix Z is different from that eq. (16) for the strongly-convex-strongly-convex case, which takes the form of

$$Z := \begin{bmatrix} & & & 1 & -1 \\ & & 1 & -1 & \\ & \iddots & \iddots & & \\ -1 & & & & \end{bmatrix}, \quad Z^2 := \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{bmatrix}. \tag{21}$$

It can be verified that Z is invertible and Z^2 in eq. (21) also satisfies the zero-chain property, i.e., Theorem 5. We can further verify that $\Phi(x)$ is convex and functions f, g satisfy Assumptions 1 and 2.

Step 2 (characterize the minimizer x^*): Recall that $x^* \in \arg\min_{x\in\mathbb{R}^d} \Phi(x)$. We then show that x^* satisfies the equation

$$\Big(\frac{L_x\beta^2}{4}Z^6 + \frac{L_x\beta^2\beta\mu_y}{2}Z^4 + \Big(\frac{L_y\widetilde{L}_{xy}^2}{4} + \frac{L_x\mu_y^2}{4}\Big)Z^2\Big)x^* = \frac{L_y\widetilde{L}_{xy}}{2}Zb.$$

Let $\widetilde{b} = \frac{L_y\widetilde{L}_{xy}}{2}Zb$ and choose b such that $\widetilde{b}_t = 0$ for all $t \geq 4$. Then, by choosing $\widetilde{b}_1, \widetilde{b}_2, \widetilde{b}_3$ properly, we derive that $x^* = \frac{B}{\sqrt{d}}\mathbf{1}$, where $\mathbf{1}$ is the all-one vector, and hence $\|x^*\| = B$.

Step 3 (characterize the gradient norm): In this step, we show that for any x whose last three coordinates are zeros, the gradient norm of $\nabla\Phi(x)$ is lower-bounded. Namely, we prove that

$$\min_{x\in\mathbb{R}^d:\, x_{d-2}=x_{d-1}=x_d=0} \|\nabla\Phi(x)\|^2 \geq \frac{B^2\Big(\frac{\widetilde{L}_{xy}^2 L_y}{4} + \frac{L_x\mu_y^2}{4}\Big)^2}{8\mu_y^4 d^4 + 16d\beta^4 + 32d\beta^3\mu_y + 32d\beta^2\mu_y^2}. \tag{22}$$

Step 4 (characterize the iterate subspaces): By exploiting the forms of the subspaces $\{\mathcal{H}_x^k, \mathcal{H}_y^k\}_{k=1}^K$ defined in Theorem 2 and by induction, we show that

$$H_x^K \subseteq \mathrm{Span}\{Z^{2(K+QT-Q)}(Zb), ..., Z^2(Zb), (Zb)\}.$$

Since $(Zb)_t = 0$ for all $t \geq 4$ and using the zero-chain property of Z^2, we have the t^{th} coordinate of the output x^K is zero, i.e., $(x^K)_t = 0$, for all $t \geq M+1$, where $M = K + QT - Q + 3$.

Step 5 (combine Steps $1,2,3,4$ and characterize the complexity): Choose d such that the right hand side of eq. (22) equals ϵ by solving eq. (19). Then, using the results in Steps 3 and 4, it follows that for any $M \leq d-3$, $\|\nabla\Phi(x^K)\| \geq \epsilon$. Thus, to achieve $\|\nabla\Phi(x)\| \leq \epsilon$, it requires $M > d-3$ and the complexity result follows as $\mathcal{C}_{\text{grad}}(\mathcal{A},\epsilon) = \Omega(M)$.

4. Accelerated Gradient Method and Upper Bounds for Bilevel Optimization

In this section, we propose a new bilevel optimization algorithm, and characterize its computational complexity, which serves as new upper bounds for bilevel optimization.

4.1 Accelerated Bilevel Optimization Algorithm: AccBiO

As shown in Algorithm 1, we propose a new accelerated algorithm named AccBiO for bilevel optimization. At the beginning of each outer iteration, we run N steps of accelerated gradient descent (AGD) to get y_k^N as an approximate of $y_k^* = \arg\min_y g(x_k, y)$. Then, based on the inner-level output y_k^N, we construct a hypergradient estimate via $G_k := \nabla_x f(x_k, y_k^N) - \nabla_x\nabla_y g(x_k, y_k^N) v_k^M$, where v_k^M is the output of an M-step heavy ball method with stepsizes η and θ for solving a quadratic problem as shown in line 7. Finally, as shown in lines 8-9, we update the variables z_k and x_k using Nesterov's momentum acceleration scheme (Nesterov et al., 2018) over the estimated hypergradient G_k. Next, we analyze the convergence and complexity performance of AccBiO for the two bilevel optimization classes $\mathcal{F}_{scsc}$ and $\mathcal{F}_{csc}$ described in Theorem 1.

4.2 Strongly-Convex-Strongly-Convex Bilevel Optimization

In this setting, $\Phi(x)$ is μ_x-strongly-convex and $g(x,\cdot)$ is μ_y-strongly-convex. The following theorem provides a performance guarantee for AccBiO. Recall $x^* = \arg\min_x \Phi(x)$.

Theorem 9 *Suppose that (f,g) belong to the strongly-convex-strongly-convex class $\mathcal{F}_{scsc}$ in Theorem 1. Choose the inner iteration number $N = \widetilde{\Theta}(\kappa_y)$. Choose stepsizes $\lambda = \frac{4}{(\sqrt{\widetilde{L}_y}+\sqrt{\mu_y})^2}$ and $\theta = \max\big\{\big(1-\sqrt{\lambda\mu_y}\big)^2, \big(1-\sqrt{\lambda\widetilde{L}_y}\big)^2\big\}$ and $M = \widetilde{\Theta}(\kappa_y)$ for the heavy-ball method. Let $\kappa_y = \frac{\widetilde{L}_y}{\mu_y}$ be the condition number for the inner-level function $g(x,\cdot)$ and $L_\Phi = \Theta\Big(L_x + \frac{L_y\widetilde{L}_{xy}^2}{\mu_y^2} + \frac{\widetilde{L}_{xy}^2\rho_{xy}}{\mu_y^3}\|\nabla_y f(x^*, y^*(x^*))\| + \frac{\widetilde{L}_{xy}^3\rho_{yy}L_y}{\mu_y^4}\sqrt{\frac{2}{\mu_x}(\Phi(0)-\Phi(x^*)) + \|x^*\|^2 + \frac{\epsilon}{\mu_x}}\Big)$ be the smoothness parameter of the objective $\Phi(\cdot)$. Then, we have*

$$\Phi(z_K) - \Phi(x^*) \leq \Big(1 - \frac{1}{\sqrt{\kappa_x}}\Big)^K (\Phi(0) - \Phi(x^*) + \frac{\mu_x}{2}\|x^*\|^2) + \frac{\epsilon}{2},$$

Algorithm 1 Accelerated Bilevel Optimization (AccBiO) Algorithm

Input: Initialization $z_0 = x_0 = y_0 = 0$, parameters λ and θ
for $k = 0, 1, ..., K$ **do**
  Set $y_k^0 = 0$ as initialization
  for $t = 1, ..., N$ **do**

$$y_k^t = s_k^{t-1} - \frac{1}{\widetilde{L}_y}\nabla_y g(x_k, s_k^{t-1}),\ s_k^t = \frac{2\sqrt{\kappa_y}}{\sqrt{\kappa_y}+1}y_k^t - \frac{\sqrt{\kappa_y}-1}{\sqrt{\kappa_y}+1}y_k^{t-1}.$$

  end for
  Hypergradient computation:
    1) Get v_k^M after running M steps of heavy-ball method $v_k^{t+1} = v_k^t - \lambda\nabla Q(v_k^t) + \theta(v_k^t - v_k^{t-1})$ with initialization $v_k^0 = v_k^1 = 0$ over

$$\min_v Q(v) := \frac{1}{2}v^T\nabla_y^2 g(x_k, y_k^N)v - v^T\nabla_y f(x_k, y_k^N)$$

    2) Compute $\nabla_x\nabla_y g(x_k, y_k^N)v_k^M$ via automatic differentiation;
    3) compute $G_k := \nabla_x f(x_k, y_k^N) - \nabla_x\nabla_y g(x_k, y_k^N)v_k^M$.
  Update $z_{k+1} = x_k - \frac{1}{L_\Phi}G_k$
  Update $x_{k+1} = \Big(1 + \frac{\sqrt{\kappa_x}-1}{\sqrt{\kappa_x}+1}\Big)z_{k+1} - \frac{\sqrt{\kappa_x}-1}{\sqrt{\kappa_x}+1}z_k$
end for

where $\kappa_x = \frac{L_\Phi}{\mu_x}$ is the condition number for $\Phi(\cdot)$. To achieve $\Phi(z_K) - \Phi(x^*) < \epsilon$, the complexity satisfies

$$\mathcal{C}_{\text{fun}}(\mathcal{A},\epsilon) = \mathcal{O}\Big(\sqrt{\frac{L_x\widetilde{L}_y}{\mu_x\mu_y}} + \sqrt{\frac{L_y\widetilde{L}_{xy}^2\widetilde{L}_y}{\mu_x\mu_y^3}} + \sqrt{\frac{\widetilde{L}_{xy}^2\rho_{xy}\widetilde{L}_y}{\mu_y^4}}\sqrt{\frac{\|\nabla_y f(x^*, y^*(x^*))\|}{\mu_x}} + \sqrt{\frac{\widetilde{L}_{xy}^3\rho_{yy}L_y\widetilde{L}_y}{\mu_x\mu_y^5}}\Big(\frac{2}{\mu_x}(\Phi(0) - \Phi(x^*)) + \|x^*\|^2 + \frac{\epsilon}{\mu_x}\Big)^{\frac{1}{4}}\Big), \tag{23}$$

To the best of our knowledge, our result in Theorem 9 is the first-known upper bound on the computational complexity for strongly-convex bilevel optimization under only mild assumptions on the Lipschitz continuity of the first- and second-order derivatives of the outer- and inner-level functions f, g. As a comparison, existing results in Ghadimi and Wang (2018); Ji et al. (2021) for bilevel optimization further make a strong assumption that the gradient norm $\|\nabla_y f(x,y)\|$ is bounded for all $(x,y) \in \mathbb{R}^p \times \mathbb{R}^q$ to upper-bound the smoothness parameter L_{Φ_k} of $\Phi(x_k)$ and the hypergradient estimation error $\|G_k - \nabla\Phi(x_k)\|$ at the k^{th} iteration. This is because L_{Φ_k} and $\|G_k - \nabla\Phi(x_k)\|$ turn out to be increasing with the gradient norm $\|\nabla_y f(x_k, y^*(x_k))\|$, for which it is challenging to prove the boundedness given the theoretical frameworks in Ghadimi and Wang (2018); Ji et al. (2021) where no results on bounded iterates are established. Our analysis does not require such a restrictive assumption because we show by induction that the optimality gap $\|x_k - x^*\|$ is well bounded as the algorithm runs. As a result, we can guarantee the boundedness of the smoothness parameter L_{Φ_k} and the error $\|G_k - \nabla\Phi(x_k)\|$ during the entire optimization process. In Section 5, we further develop tighter upper bounds than existing results under this additional bounded gradient assumption.

Based on Theorem 9, we next study the quadratic g subclass, where the inner-level function $g(x,y)$ takes a quadratic form as in eq. (5). The following corollary provides upper bounds on the convergence rate and complexity of AccBiO under this case.

Corollary 10 (Quadratic g subclass) *Under the same setting of Theorem 9, consider the quadratic inner-level function $g(x,y)$ in eq. (5), where $\nabla_x\nabla_y g(\cdot,\cdot)$ and $\nabla_y^2 g(\cdot,\cdot)$ are constant. To achieve $\Phi(z_K)-\Phi(x^*)<\epsilon$, the complexity satisfies $\mathcal{C}_{\rm fun}(\mathcal{A},\epsilon)=\widetilde{\mathcal{O}}\Big(\sqrt{\frac{L_y\widetilde{L}_{xy}^2\widetilde{L}_y}{\mu_x\mu_y^3}}\Big)$.*

Theorem 10 shows that for the quadratic g subclass, the complexity upper bound in Theorem 9 specializes to $\widetilde{\mathcal{O}}\Big(\sqrt{\frac{L_y\widetilde{L}_{xy}^2\widetilde{L}_y}{\mu_x\mu_y^3}}\Big)$. This improvement over the complexity for the general case in eq. (23) comes from tighter upper bounds on the smoothness parameter L_Φ of the objective function $\Phi(x)$ and a smaller hypergradient estimation error $\|G_k-\nabla\Phi(x_k)\|$. In addition, it can be seen that when the inner-level problem is easy to solve, i.e., $\widetilde{L}_y=C\mu_y$ for any constant $C\geq 1$ that is independent of μ_y, the complexity becomes $\widetilde{\mathcal{O}}\big(\sqrt{\frac{L_y\widetilde{L}_{xy}^2}{\mu_x\mu_y^2}}\big)$, which matches the lower bound established by Theorem 4 up to logarithmic factors.

4.3 Convex-Strongly-Convex Bilevel Optimization

We next provide an upper bound for convex-strongly-convex bilevel optimization, where the function $\Phi(x)$ is convex. Recall from Theorem 1 that $\|x^*\|=B$ for some constant $B>0$, where x^* is one minimizer of $\Phi(\cdot)$. For this case, we construct a strongly-convex-strongly-convex function $\widetilde{\Phi}(\cdot)=\widetilde{f}(x,y^*(x))$ by adding a small quadratic regularization to the outer-level function $f(x,y)$, i.e.,

$$\widetilde{f}(x,y)=f(x,y)+\frac{\epsilon}{2R}\|x\|^2. \tag{24}$$

Then, we can apply the results in Theorem 9 to $\widetilde{\Phi}(x)$, and obtain the following theorem.

Theorem 11 *Suppose that (f,g) belong to the convex-strongly-convex class $\mathcal{F}_{csc}$ in Theorem 1. Let $L_{\widetilde{\Phi}}$ be the smoothness parameter of function $\widetilde{\Phi}(\cdot)$, which takes the same form as L_Φ in Theorem 9 except that L_x, f, x^* and Φ become $L_x+\frac{\epsilon}{R}, \widetilde{f}, \widetilde{x}^*$ and $\widetilde{\Phi}$, respectively. Choose $M=\widetilde{\Theta}(\kappa_y)$ for the heavy-ball method. We consider two widely-used convergence criterions as follows.*

- **(Suboptimality gap)** *Choose $R=B^2$ in eq. (24), and choose the same parameters as in Theorem 9 with ϵ and μ_x being replaced by $\epsilon/2$ and $\frac{\epsilon}{R}$, respectively. To achieve $\Phi(z_K)-\Phi(x^*)\leq\epsilon$, the required complexity is at most*

$$\begin{aligned}\mathcal{C}_{\rm fun}(\mathcal{A},\epsilon)\leq&\widetilde{\mathcal{O}}\Big(\sqrt{\frac{B^2L_y\widetilde{L}_{xy}^2\widetilde{L}_y}{\epsilon\mu_y^3}}+\sqrt{\frac{B^2\widetilde{L}_{xy}^2\rho_{xy}\widetilde{L}_y}{\epsilon\mu_y^4}}\sqrt{\|\nabla_y\widetilde{f}(\widetilde{x}^*,y^*(\widetilde{x}^*))\|}\\&+\sqrt{\frac{B^2\widetilde{L}_{xy}^3\rho_{yy}L_y\widetilde{L}_y}{\epsilon\mu_y^5}}\Big(\frac{2B^2}{\epsilon}(\widetilde{\Phi}(0)-\widetilde{\Phi}(\widetilde{x}^*))+\|\widetilde{x}^*\|^2+B^2\Big)^{\frac{1}{4}}\Big).\end{aligned}$$

- **(Gradient norm)** *Choose $R = B$ in eq. (24), and choose the same parameters as in Theorem 9 with ϵ and μ_x being replaced by $\epsilon^2/(4L_{\widetilde{\Phi}} + \frac{8\epsilon}{R})$ and $\frac{\epsilon}{R}$, respectively. To achieve $\|\nabla\Phi(z_k)\| \leq 5\epsilon$, the required complexity is at most*

$$\mathcal{C}_{\text{grad}}(\mathcal{A},\epsilon) \leq \widetilde{\mathcal{O}}\Big(\sqrt{\frac{BL_y\widetilde{L}_{xy}^2\widetilde{L}_y}{\epsilon\mu_y^3}} + \sqrt{\frac{B\widetilde{L}_{xy}^2\rho_{xy}\widetilde{L}_y}{\epsilon\mu_y^4}}\sqrt{\|\nabla_y\widetilde{f}(\widetilde{x}^*, y^*(\widetilde{x}^*))\|} + \sqrt{\frac{B\widetilde{L}_{xy}^3\rho_{yy}L_y\widetilde{L}_y}{\epsilon\mu_y^5}}\Big(\frac{2B}{\epsilon}(\widetilde{\Phi}(0) - \widetilde{\Phi}(\widetilde{x}^*)) + \|\widetilde{x}^*\|^2 + B\Big)^{\frac{1}{4}}\Big).$$

As far as we know, Theorem 11 is the first convergence result for convex-strongly-convex bilevel optimization without the bounded gradient assumption. Then, similarly to Theorem 10, we also study the quadratic $g(x,y)$ case where the inner-level function$g(x,y)$ takes the quadratic form as given in eq. (5).

Corollary 12 (Quadratic g subclass) *Under the same setting of Theorem 11, consider the quadratic $g(x,y)$ where $\nabla_x\nabla_y g(\cdot,\cdot)$ and $\nabla_y^2 g(\cdot,\cdot)$ are constant. Then, we have*

- **(Suboptimality gap)** *To achieve $\Phi(z_K) - \Phi(x^*) \leq \epsilon$, $\mathcal{C}_{\text{fun}}(\mathcal{A},\epsilon) = \widetilde{\mathcal{O}}\Big(B\sqrt{\frac{L_y\widetilde{L}_{xy}^2\widetilde{L}_y}{\epsilon\mu_y^3}}\Big)$.*
- **(Gradient norm)** *To achieve $\|\nabla\Phi(z_k)\| \leq \epsilon$, we have $\mathcal{C}_{\text{grad}}(\mathcal{A},\epsilon) = \widetilde{\mathcal{O}}\Big(\sqrt{\frac{BL_y\widetilde{L}_{xy}^2\widetilde{L}_y}{\epsilon\mu_y^3}}\Big)$.*

It can be seen from Theorem 12 that for the quadratic g subclass, AccBiO achieves a computational complexity of $\widetilde{\mathcal{O}}\Big(\sqrt{\frac{BL_y\widetilde{L}_{xy}^2\widetilde{L}_y}{\epsilon\mu_y^3}}\Big)$ in term of the gradient norm. For the case where $\widetilde{L}_y = \Theta(\mu_y)$, the complexity becomes $\widetilde{\mathcal{O}}\Big(\sqrt{\frac{BL_y\widetilde{L}_{xy}^2}{\epsilon\mu_y^2}}\Big)$, which matches the lower bound in Theorem 7 up to logarithmic factors.

4.4 Optimality of Bilevel Optimization and Discussion

We compare the lower and upper bounds and make the following remarks on the optimality of bilevel optimization and its comparison to minimax optimization.

Optimality of results for quadratic g subclass. We compare the developed lower and upper bounds and make a few remarks on the optimality of the proposed AccBiO algorithms. Let us first focus on the quadratic g subclass where $g(x,y)$ takes the quadratic form as in eq. (5). For the strongly-convex-strongly-convex setting, comparison of Theorem 4 and Theorem 10 implies that AccBiO achieves the optimal complexity for $\widetilde{L}_y = \Theta(\mu_y)$, i.e., the inner-level problem is easy to solve. For the general case, there is still a gap of $\sqrt{\kappa_y}$ between lower and upper bounds. For the convex-strongly-convex setting, comparison of Theorem 6 and Theorem 12 shows that AccBiO is optimal for $\widetilde{L}_y = \Theta(\mu_y)$, and there is a gap for the general case. Such a gap is mainly due to the large smoothness parameter L_Φ of $\Phi(\cdot)$. We note that a similar issue also occurs for minimax optimization, which has been addressed by Lin et al. (2020) using an accelerated proximal point method for the inner-level problem and exploiting Sion's minimax theorem $\min_x \max_y f(x,y) = \max_y \min_x f(x,y)$. However, such an approach is not applicable for bilevel optimization due to the asymmetry of x and y, e.g., $\min_x f(x, y^*(x)) \neq \min_y g(x^*(y), y)$. This gap between lower and upper bounds deserves future efforts.

Optimality of results for general g. We now discuss the optimality of our results for a more general g whose second-order derivatives are Lipschitz continuous. For the strongly-convex-strongly-convex setting, it can be seen from the comparison of Theorem 4 and Theorem 9 that there is a gap between the lower and upper bounds. This gap is because the lower bounds construct the bilinearly coupled worst-case $g(x,y)$ whose Hessians and Jacobians are constant, rather than generally ρ_{yy}- and ρ_{xy}-Lipschitz continuous as considered in the upper bounds. Hence, tighter lower bounds need to be provided for this setting, which requires more sophisticated worst-case instances with Lipschitz continuous Hessians $\nabla_y^2 g(x,y)$ and Jacobians $\nabla_x \nabla_y g(x,y)$. For example, it is possible to construct $g(x,y)$ as $g(x,y) = \sigma(y) y^T Z y - x^T Z y + b^T y$, where $\sigma(\cdot) : \mathbb{R}^d \to \mathbb{R}$ satisfies a certain Lipschitz property. For example, if σ is Lipchitz continuous, simple calculation shows that L_Φ scales at an order of κ_y^3. However, it still requires significant efforts to determine the form of σ such that the optimal point of $\Phi(\cdot)$ and the subspaces $\mathcal{H}_x, \mathcal{H}_y$ are easy to characterize and satisfy the properties outlined in the proof of Theorem 9.

Comparison to minimax optimization. We compare the optimality between minimax optimization and bilevel optimization. For the strongly-convex-strongly-convex minimax optimization, Zhang et al. (2019) developed a lower bound of $\widetilde{\Omega}(\sqrt{\kappa_x \kappa_y})$ for minimax optimization, which is achieved by the accelerated proximal point method proposed by Lin et al. (2020) up to logarithmic factors. For the same type of bilevel optimization, we provide a lower bound of $\widetilde{\Omega}(\sqrt{\kappa_x}\kappa_y)$ in Theorem 4, which is larger than that of minimax optimization by a factor of $\sqrt{\kappa_y}$. A similar comparison is also observed for the convex-strongly-convex bilevel optimization. This establishes that bilevel optimization is fundamentally more challenging than minimax optimization. This is because bilevel optimization needs to handle the different structures of the outer- and inner-level functions f and g (e.g., second-order derivatives in the hypergradient), whereas for minimax optimization, the fact of $f = g$ simplifies the problem (e.g., no second-order derivatives) and allows more efficient algorithm designs.

5. Upper Bounds with Gradient Boundedness Assumption

Our study in Section 4 does not make the bounded gradient assumption, which has been commonly taken in the existing studies (Ghadimi and Wang, 2018; Ji et al., 2021; Hong et al., 2020; Ji et al., 2020a). In this section, we establish tighter upper bounds than those in existing works (Ghadimi and Wang, 2018; Ji et al., 2021) under such an additional assumption.

Assumption 3 (Bounded gradient) *There exists a constant U such that for any $(x', y') \in \mathbb{R}^p \times \mathbb{R}^q$, $\|\nabla_y f(x', y')\| \leq U$.*

Algorithm 2 Accelerated Bilevel Optimization Method under Bounded Gradient Assumption (AccBiO-BG)

Input: Initialization $z_0 = x_0 = y_0 = 0$, parameters $\eta_k, \tau_k.\alpha_k, \beta_k, \lambda$ and θ
for $k = 0, ..., K$ **do**
  Set $\widetilde{x}_k = \eta_k x_k + (1-\eta_k) z_k$
  Set $y_k^0 = y_{k-1}^N$ if $k > 0$ and y_0 otherwise (warm start)
  for $t = 1, ..., N$ **do**

$$\text{(AGD:)} \quad y_k^t = s_k^{t-1} - \frac{1}{\widetilde{L}_y} \nabla_y g(\widetilde{x}_k, s_k^{t-1}), \quad s_k^t = \frac{2\sqrt{\kappa_y}}{\sqrt{\kappa_y}+1} y_k^t - \frac{\sqrt{\kappa_y}-1}{\sqrt{\kappa_y}+1} y_k^{t-1}.$$

  end for
  Hypergradient computation:
    1) Get v_k^M after running M steps of heavy-ball method $v_k^{t+1} = v_k^t - \lambda \nabla Q(v_k^t) + \theta(v_k^t - v_k^{t-1})$ with initialization $v_k^0 = v_k^1 = 0$ over

$$\text{(Quadratic programming:)} \quad \min_v Q(v) := \frac{1}{2} v^T \nabla_y^2 g(\widetilde{x}_k, y_k^N) v - v^T \nabla_y f(\widetilde{x}_k, y_k^N);$$

    2) Compute Jacobian-vector product $\nabla_x \nabla_y g(\widetilde{x}_k, y_k^N) v_k^M$ via automatic differentiation;
    3) compute **hypergradient estimate** $G_k := \nabla_x f(\widetilde{x}_k, y_k^N) - \nabla_x \nabla_y g(\widetilde{x}_k, y_k^N) v_k^M$.
  Update $x_{k+1} = \tau_k \widetilde{x}_k + (1-\tau_k) x_k - \beta_k G_k$
  Update $z_{k+1} = \widetilde{x}_k - \alpha_k G_k$
end for

5.1 Accelerated Bilevel Optimization Algorithm: AccBiO-BG

We propose an accelerated algorithm named AccBiO-BG in Algorithm 2 for bilevel optimization under the additional bounded gradient assumption. Similarly to AccBiO, AccBiO-BG first runs N steps of accelerated gradient descent (AGD) at each outer iteration. Note that AccBiO-BG here adopts a warm start strategy with $y_k^0 = y_{k-1}^N$ so that our analysis does not require the boundedness of $y^*(x_k), k = 0..., K$ and reduces the total computational complexity. Then, AccBiO-BG constructs the hypergradient estimate $G_k := \nabla_x f(\widetilde{x}_k, y_k^N) - \nabla_x \nabla_y g(\widetilde{x}_k, y_k^N) v_k^M$ following the same steps as in AccBiO. Finally, we update variables x_k, z_k via two accelerated gradient steps, where we incorporate a variant (Ghadimi and Lan, 2016) of Nesterov's momentum. We use this variant instead of vanilla Nesterov's momentum (Nesterov et al., 2018) in Algorithm 1, because the resulting analysis is easier to handle the warm start strategy, which backpropagates the tracking error $\|y_k^N - y^*(x_k)\|$ to previous loops.

5.2 Strongly-Convex-Strongly-Convex Bilevel Optimization

The following theorem provides a theoretical performance guarantee for AccBiO-BG.

Theorem 13 *Suppose that (f, g) belong to the strongly-convex-strongly-convex class $\mathcal{F}_{scsc}$ in Theorem 1 and further suppose Assumption 3 is satisfied. Choose $\alpha_k = \alpha \leq \frac{1}{2L_\Phi}$, $\eta_k = \frac{\sqrt{\alpha\mu_x}}{\sqrt{\alpha\mu_x}+2}$, $\tau_k = \frac{\sqrt{\alpha\mu_x}}{2}$, $\beta_k = \sqrt{\frac{\alpha}{\mu_x}}$ and $N = \widetilde{\Theta}(\kappa_y)$, where L_Φ is the smoothness parameter of $\Phi(x)$. Choose stepsizes $\lambda = \frac{4}{(\sqrt{\widetilde{L}_y}+\sqrt{\mu_y})^2}$, $\theta = \max\big\{\big(1-\sqrt{\lambda\mu_y}\big)^2, \big(1-\sqrt{\lambda\widetilde{L}_y}\big)^2\big\}$ and $M = \widetilde{\Theta}(\kappa_y)$ for the heavy-ball method. Then, to achieve $\Phi(z_K) - \Phi(x^*) \leq \epsilon$, the required complexity*

$\mathcal{C}_{\text{fun}}(\mathcal{A},\epsilon)$ *is at most*

$$\mathcal{C}_{\text{fun}}(\mathcal{A},\epsilon)=\widetilde{\mathcal{O}}\Big(\sqrt{\frac{U\widetilde{L}_{xy}^2\rho_{yy}\widetilde{L}_y}{\mu_x\mu_y^4}}\Big),$$

The proof of Theorem 13 is provided in Appendix J. Theorem 13 shows that the upper bound achieved by our proposed AccBiO-BG algorithm is $\widetilde{\mathcal{O}}(\sqrt{\frac{U\widetilde{L}_{xy}^2\rho_{yy}\widetilde{L}_y}{\mu_x\mu_y^4}})$. This bound improves the best known $\widetilde{\mathcal{O}}\big(\max\big\{\frac{U\widetilde{L}_{xy}^2\rho_{yy}}{\mu_x\mu_y^3},\frac{\widetilde{L}_y^2}{\mu_y^2}\big\}\big)$ (see eq. (2.10) and (2.60) therein) achieved by the accelerated bilevel approximation algorithm (ABA) in Ghadimi and Wang (2018) by a factor of $\mathcal{O}(\sqrt{\frac{U\widetilde{L}_{xy}^2\rho_{yy}}{\widetilde{L}_y\mu_x\mu_y^2}})$.

5.3 Convex-Strongly-Convex Bilevel Optimization

Similarly to Theorem 11, we consider a strongly-convex-strongly-convex function $\widetilde{\Phi}(\cdot)=\widetilde{f}(x,y^*(x))$ with $\widetilde{f}(x,y)=f(x,y)+\frac{\epsilon}{2B^2}\|x\|^2$, where $B=\|x^*\|$ as defined in Theorem 1. Then, we have the following theorem.

Theorem 14 *Suppose that (f,g) belong to the convex-strongly-convex class $\mathcal{F}_{csc}$ in Theorem 1 and further suppose Assumption 3 is satisfied. Let $L_{\widetilde{\Phi}}$ be the smoothness parameter of $\widetilde{\Phi}(\cdot)$, which takes the same form as L_Φ in Theorem 13 but with L_x being replaced by $L_x+\frac{\epsilon}{B^2}$. Choose the same parameter as in Theorem 13 with $\alpha=\frac{1}{2L_{\widetilde{\Phi}}}$, $N=\widetilde{\Theta}(\kappa_y)$, $M=\widetilde{\Theta}(\kappa_y)$ and $\mu_x=\frac{\epsilon}{B^2}$. Then, to achieve $\Phi(z_K)-\Phi(x^*)\leq\epsilon$, the required complexity $\mathcal{C}_{\text{fun}}(\mathcal{A},\epsilon)$ is at most*

$$\mathcal{C}_{\text{fun}}(\mathcal{A},\epsilon)=\widetilde{\mathcal{O}}\Big(B\sqrt{\frac{\widetilde{L}_{xy}^2\rho_{yy}\widetilde{L}_y}{\epsilon\mu_y^4}}\Big).$$

As shown in Theorem 14, our proposed AccBiO-BG algorithm achieves a complexity of $\widetilde{\mathcal{O}}(\sqrt{\frac{\widetilde{L}_{xy}^2\rho_{yy}\widetilde{L}_y}{\epsilon\mu_y^4}})$, which significantly improves the best known result achieved by the ABA algorithm in Ghadimi and Wang (2018) (see Table 2).

Extension to nonconvex-strongly-convex case. By applying the acceleration scheme in Algorithm 2 with $\tau_k=0,\alpha_k=\Theta(\frac{1}{L_\Phi}),\beta_k=\frac{k}{4L_\Phi}$ and $\eta_k=\frac{2}{k+1}$ (following the ABA method in Ghadimi and Wang (2018)), we can also improve the complexity by a factor of $\mathcal{O}(\sqrt{\kappa_y})$. In this case, we need the bounded gradient assumption because we cannot guarantee the boundedness of the iterates under the nonconvex geometry unlike the convex case.

6. Numerical Experiments

n this section, we conduct experiments to validate our theoretical results. We consider the following bilevel optimization problem, where the upper and lower functions are given by

$$f(x,y)=\frac{1}{2}x^TU^2x+\frac{1}{2}\|y\|^2$$

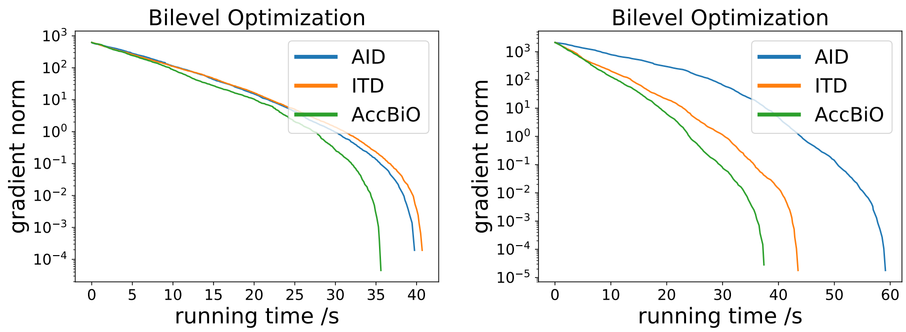

Figure 1: Gradient norm $\|\nabla\Phi(\hat{x})\|$ v.s. running time (in seconds). Left plot: dimension $d = 30$; right plot: dimension $d = 50$.

$$g(x, y) = \frac{1}{2}y^T(H^2 + I)y - \frac{1}{2}x^T V y + b^T y, \tag{25}$$

where U, H and V are random matrices with each entry sampled from $[0, 1)$ uniformly at random. We compare the proposed AccBiO method with two widely-used bilevel benchmark algorithms AID and ITD in Grazzi et al. (2020); Ji et al. (2021).

Hyperparameter setting. For AID and ITD, we choose their best inner- and outer-loop learning rates from $\{10^i, i = -6, -5, -4, -3, -2, -1, 0, 1, 2, ..., 6\}$. For AccBiO, we choose the coefficients α and β in the inner-loop acceleration updates $y_k^t = s_k^{t-1} - \alpha\nabla_y g(x_k, s_k^{t-1}), s_k^t = (1+\beta)y_k^t - \beta y_k^{t-1}$ from $\{10^i, i = -5, ..., 4, 5\}$. Similarly, we choose the coefficients in the outer-loop acceleration updates (i.e., line 8-9 in Algorithm 2) from $\{10^i, i = -7, -6, ..., 6, 7\}$. For ITD, we choose the number N of inner-loop iterations from $\{5, 10, 15, 20\}$. For AID and our AccBiO method, we choose N from $\{5, 10, 15, 20\}$ and the number M of the iterations for solving the linear system from $\{5, 10, 15\}$.

In Figure 1, we plot the gradient norm $\|\nabla\Phi(\hat{x})\|$ v.s. running time (in seconds) over the problem dimension $d = 30$ and $d = 50$, where $\hat{x}$ is the output of the algorithm. It can be seen that our proposed AccBiO has the fastest convergence rate, and AID achieves a convergence performance similarly to ITD. This validates our theoretical result that AccBiO achieves a lower computational complexity than non-acceleration methods of ITD and AID.

7. Conclusion and Discussion

In this paper, we provide the first-known lower bounds and new upper bounds with relaxed assumptions and tighter characterizations for bilevel optimization under various function geometries. We here discuss the extensions and applications of our results as follows.

Other loss geometries. In this paper, we study two typical loss geometries, i.e., the strongly-convex-strongly-convex and convex-strongly-convex geometries. It will be interesting to investigate other types of loss landscapes. For example, when the total objective function $\Phi(x)$ involves neural networks and is generally nonconvex, new efforts are needed to address the boundedness of iterates x_k as the algorithm runs, e.g., by adding a projection onto a bounded domain or a regularizer to force such a boundedness. Moreover, existing convergence rate analysis relies on the strong convexity of the inner problem to better capture the inner-level convergence behavior. It is interesting to extend to more general geometries that allows more than one unique solution, e.g., convexity or star-convexity, which, however, requires us to revise the hypergradient form in eq. (7) or explore the convergence under other criterions such as stationarity based on the Moreau envelope (Davis and Drusvyatskiy, 2019) due to the nonsmoothness of the inner-level solution $y^*(x)$ and the objective function $\Phi(x)$.

Applications of results. We note that some of our analysis can be applied to other problem domains such as minimax optimization. For example, our lower-bounding technique for Theorem 6 can be extended to **convex-concave** or **convex-strongly-concave minimax** optimization, where the objective function $f(x, y)$ satisfies the general smoothness property as in eq. (2) with the general smoothness parameters $L_x, L_{xy}, L_y \geq 0$. The resulting lower bound will be different from that in Ouyang and Xu (2019), which considered a special case with $L_y = 0$ and the convergence is measured in terms of the suboptimality gap $\mathcal{O}(\Phi(x) - \Phi(x^*))$ rather than the gradient norm $\|\nabla\Phi(x)\|$ considered in this paper. Thus, such an extension will serve as a new contribution to lower complexity bounds for minimax optimization.

Appendix

Table of Contents

Appendix A. ITD-Based Bilevel Algorithms

In this section, we present existing ITD-based bilevel optimization algorithms, and show that they belong to the hypergradient-based algorithm class we consider in Theorem 2.

Example 2 (ITD-based Bilevel Algorithms) *(Maclaurin et al., 2015; Franceschi et al., 2017; Ji et al., 2021; Grazzi et al., 2020) Such type of algorithms use ITD-based approaches for hypergradient computation, and take the following updates.*

For each outer iteration $m = 0, ..., Q-1$,

- *Update variable y for N times via iterative algorithms (e.g., gradient descent, accelerated gradient methods).*

$$\text{(Gradient descent:)} \quad y_m^t = y_m^{t-1} - \eta \nabla_y g(x_m, y_m^{t-1}), t = 1, ..., N. \tag{26}$$

- *Compute the hypergradient estimate* $G_m = \frac{\partial f(x_m, y_m^N(x_m))}{\partial x_m}$ *via backpropagation. Under the gradient updates in eq. (26),* G_m *takes the form of*

$$G_m = \nabla_x f(x_m, y_m^N) - \eta \sum_{t=0}^{N-1} \nabla_x \nabla_y g(x_m, y_m^t) \prod_{j=t+1}^{N-1} (I - \eta \nabla_y^2 g(x_m, y_m^j)) \nabla_y f(x_m, y_m^N). \quad (27)$$

A similar form holds for case when updating y *with accelerated gradient methods.*

- *Update* x *based on* G_m *via gradient-based iterative methods.*

It can be seen from eq. (27) that only Hessian-vector products $\nabla_y^2 g(x_m, y_m^j)v_j, j = 1, ..., N$ and Jacobian-vector products $\nabla_x \nabla_y g(x_m, y_m^j)v_j, j = 1, ..., N$ are computed, where each v_j is obtained recursively via

$$v_{j-1} = \underbrace{(I - \alpha \nabla_y^2 g(x_m, y_m^j))v_j}_{\text{Hessian-vector product}} \text{ with } v_N = \nabla_y f(x_m, y_m^N).$$

Appendix B. Proof of Theorem 4

In this section, we provide a complete proof of Theorem 4 under the strongly-convex-strongly-convex geometry. Note that our construction sets the dimensions of variables x and y to be the same, i.e., $p = q = d$. The main proofs are divided into four steps: 1) constructing the worst-case instance that belongs to the problem class $\mathcal{F}_{scsc}$ defined in Theorem 1; 2) characterizing the optimal point $x^* = \arg\min_{x \in \mathbb{R}^d} \Phi(x)$; 3) characterizing the subspaces $\mathcal{H}_x^k, \mathcal{H}_y^k$; and 4) developing lower bounds on the convergence and complexity.

Step 1: Constructing the worst-case instance that satisfies Theorem 1.

In this step, we show that the constructed f, g in eq. (15) satisfy Assumptions 1 and 2, and $\Phi(x)$ is μ_x-strongly-convex. It can be seen from eq. (15) that f, g satisfy eq. (2) (3) and (4) in Assumptions 1 and 2 with arbitrary constants $L_x, L_y, \widetilde{L}_y, \widetilde{L}_{xy}$ and $\rho_{xy} = \rho_{yy} = 0$ but requires $L_{xy} \geq \frac{(L_x - \mu_x)(\widetilde{L}_y - \mu_y)}{2\widetilde{L}_{xy}}$ (which is still at a constant level) due to the introduction of the term $\frac{\alpha\beta}{\widetilde{L}_{xy}} x^T Z^3 y$ in f. We note that such a term introduces necessary connection between f and g, and yields a tighter lower bound, as pointed out in the remark at the end of Section 3.1.

We next show that the overall objective function $\Phi(x) = f(x, y^*(x))$ is μ_x-strongly-convex. According to eq. (15), we have $g(x, \cdot)$ to be μ_y-strongly-convex with a single minimizer $y^*(x) = (\beta Z^2 + \mu_y I)^{-1}\big(\frac{\widetilde{L}_{xy}}{2} Zx - b\big)$, and hence we obtain from eq. (1) that $\Phi(x)$ is given by

$$\begin{aligned}
\Phi(x) =& \frac{1}{2} x^T (\alpha Z^2 + \mu_x I)x - \frac{\alpha\beta}{\widetilde{L}_{xy}} x^T Z^3 (\beta Z^2 + \mu_y I)^{-1}\Big(\frac{\widetilde{L}_{xy}}{2} Zx - b\Big) \\
&+ \frac{\bar{L}_{xy}}{2} x^T Z (\beta Z^2 + \mu_y I)^{-1}\Big(\frac{\widetilde{L}_{xy}}{2} Zx - b\Big) + \frac{\bar{L}_{xy}}{\widetilde{L}_{xy}} b^T (\beta Z^2 + \mu_y I)^{-1}\Big(\frac{\widetilde{L}_{xy}}{2} Zx - b\Big) \\
&+ \frac{L_y}{2}\Big(\frac{\widetilde{L}_{xy}}{2} Zx - b\Big)^T (\beta Z^2 + \mu_y I)^{-1} (\beta Z^2 + \mu_y I)^{-1}\Big(\frac{\widetilde{L}_{xy}}{2} Zx - b\Big). \quad (28)
\end{aligned}$$

Note that Z is symmetric and invertible with Z^{-1} given by

$$Z^{-1} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \vdots & \iddots & 1 & \\ 1 & \iddots & & \\ 1 & & & \end{bmatrix}, \tag{29}$$

and hence the eigenvalue decomposition of Z can be written as $Z = U\,\mathrm{Diag}\{\lambda_1, ..., \lambda_d\}U^T$, where $\lambda_i \neq 0, i = 1, ..., d$ and U is an orthonormal matrix. Then, for any integers $i, j > 0$, simple calculation yields

$$Z^i(\beta Z^2 + \mu_y I)^{-j} = U\mathrm{Diag}\left\{\frac{\lambda_1^i}{(\beta\lambda_1^2 + \mu_y)^j}, ..., \frac{\lambda_d^i}{(\beta\lambda_d^2 + \mu_y)^j}\right\}U^T = (\beta Z^2 + \mu_y I)^{-j} Z^i. \tag{30}$$

Using the relationship in eq. (30), we have

$$\frac{1}{2}x^T\alpha Z^2 x = \frac{\alpha\beta}{2}x^T Z^4(\beta Z^2 + \mu_y I)^{-1}x + \frac{\alpha\mu_y}{2}x^T Z^2(\beta Z^2 + \mu_y I)^{-1}x,$$

which, in conjunction with eq. (28) and eq. (30), yields

$$\begin{aligned}\Phi(x) = \frac{1}{2}\mu_x\|x\|^2 + \frac{2\alpha\mu_y + \bar{L}_{xy}\widetilde{L}_{xy}}{4}x^T Z^2(\beta Z^2 + \mu_y I)^{-1}x - \frac{\bar{L}_{xy}}{\widetilde{L}_{xy}}b^T(\beta Z^2 + \mu_y I)^{-1}b \\ + \frac{L_y}{2}\Big(\frac{\widetilde{L}_{xy}}{2}Zx - b\Big)^T(\beta Z^2 + \mu_y I)^{-2}\Big(\frac{\widetilde{L}_{xy}}{2}Zx - b\Big) + \frac{2\alpha\beta}{\widetilde{L}_{xy}^2}b^T Z^2(\beta Z^2 + \mu_y I)^{-1}b,\end{aligned} \tag{31}$$

which is μ_x-strongly-convex.

Step 2: Characterizing $x^* = \arg\min_{x\in\mathbb{R}^d}\Phi(\cdot)$.

Based on the form of $\Phi(\cdot)$, we have

$$\begin{aligned}\nabla\Phi(x) =& (\beta Z^2 + \mu_y I)^2\mu_x x + \Big(\alpha\mu_y + \frac{\bar{L}_{xy}\widetilde{L}_{xy}}{2}\Big)(\beta Z^2 + \mu_y I)Z^2 x + \frac{L_y\widetilde{L}_{xy}}{2}\Big(\frac{\widetilde{L}_{xy}}{2}Z^2x - Zb\Big) \\ =& \Big(\beta^2\mu_x + \alpha\beta\mu_y + \frac{\beta\bar{L}_{xy}\widetilde{L}_{xy}}{2}\Big)Z^4x + (2\beta\mu_x\mu_y + \alpha\mu_y^2 + \frac{\mu_y\bar{L}_{xy}\widetilde{L}_{xy}}{2} + \frac{L_y\widetilde{L}_{xy}^2}{4})Z^2x \\ &+ \mu_x\mu_y^2 x - \frac{L_y\widetilde{L}_{xy}}{2}Zb.\end{aligned} \tag{32}$$

By setting $\nabla\Phi(x^*) = 0$, we have

$$\begin{aligned}Z^4x^* + \underbrace{\frac{2\beta\mu_x\mu_y + \alpha\mu_y^2 + \frac{\mu_y\bar{L}_{xy}\widetilde{L}_{xy}}{2} + \frac{L_y\widetilde{L}_{xy}^2}{4}}{\beta^2\mu_x + \alpha\beta\mu_y + \frac{\beta\bar{L}_{xy}\widetilde{L}_{xy}}{2}}}_{\lambda} Z^2x^* \\ + \underbrace{\frac{\mu_x\mu_y^2}{\beta^2\mu_x + \alpha\beta\mu_y + \frac{\beta\bar{L}_{xy}\widetilde{L}_{xy}}{2}}}_{\tau} x^* = \underbrace{\frac{L_y\widetilde{L}_{xy}Zb}{2(\beta^2\mu_x + \alpha\beta\mu_y + \frac{\beta\bar{L}_{xy}\widetilde{L}_{xy}}{2})}}_{\widetilde{b}},\end{aligned} \tag{33}$$

where we define $\lambda, \tau, \widetilde{b}$ for notational convenience. The following lemma establishes useful properties of x^* under a specific selection of $\widetilde{b}$.

Lemma 15 *Let b be chosen such that $\widetilde{b}$ as defined in eq. (33) satisfies $\widetilde{b}_1 = (2+\lambda+\tau)r - (3+\lambda)r^2 + r^3, \widetilde{b}_2 = r - 1$ and $\widetilde{b}_t = 0, t = 3, ..., d$, where $0 < r < 1$ is a solution of equation*

$$1 - (4+\lambda)r + (6+2\lambda+\tau)r^2 - (4+\lambda)r^3 + r^4 = 0. \tag{34}$$

Let $\hat{x}$ be a vector with each coordinate $\hat{x}_i = r^i$. Then, we have

$$\|\hat{x} - x^*\| \leq \frac{(7+\lambda)}{\tau} r^d. \tag{35}$$

Proof Note that the choice of b is achievable because Z is invertible with Z^{-1} given by

$$Z^{-1} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \vdots & \iddots & 1 & \\ 1 & \iddots & & \\ 1 & & & \end{bmatrix}.$$

Then, define a vector $\hat{b}$ with $\hat{b}_t = \widetilde{b}_t$ for $t = 1, ..., d-2$ and

$$\begin{aligned} \hat{b}_{d-1} =& r^{d-3} - (4+\lambda)r^{d-2} + (6+2\lambda+\tau)r^{d-1} - (4+\lambda)r^d \overset{(34)}{=} -r^{d+1} \\ \hat{b}_d =& r^{d-2} - (4+\lambda)r^{d-1} + (5+2\lambda+\tau)r^d \overset{(34)}{=} -r^d + (4+\lambda)r^{d+1} - r^{d+2}. \end{aligned} \tag{36}$$

Then, it can be verified that $\hat{x}$ satisfies the following equations

$$\begin{aligned} (2+\lambda+\tau)\hat{x}_1 - \hat{(}3+\lambda)x_2 + \hat{x}_3 &= \hat{b}_1 \\ -(3+\lambda)\hat{x}_1 + (6+2\lambda+\tau)\hat{x}_2 - (4+\lambda)\hat{x}_3 + \hat{x}_4 &= \hat{b}_2 \\ \hat{x}_t - (4+\lambda)\hat{x}_{t+1} + (6+2\lambda+\tau)\hat{x}_{t+2} - (4+\lambda)\hat{x}_{t+3} + \hat{x}_{t+4} &= \hat{b}_{t+2}, \text{ for } 1 \leq t \leq d-4 \\ \hat{x}_{d-3} - (4+\lambda)\hat{x}_{d-2} + (6+2\lambda+\tau)\hat{x}_{d-1} - (4+\lambda)\hat{x}_d &= \hat{b}_{d-1} \\ \hat{x}_{d-2} - (4+\lambda)\hat{x}_{d-1} + (5+2\lambda+\tau)\hat{x}_d &= \hat{b}_d, \end{aligned}$$

which, in conjunction with the forms of Z^2 and Z^4 in eq. (16), yields

$$Z^4\hat{x} + \lambda Z^2\hat{x} + \tau\hat{x} = \hat{b}.$$

Noting that $Z^4x^* + \lambda Z^2x^* + \tau x^* = \widetilde{b}$, we have

$$\tau\|x^* - \hat{x}\| \leq \|(Z^4 + \lambda Z^2 + \tau I)(x^* - \hat{x})\| = \|\widetilde{b} - \hat{b}\| \overset{(i)}{\leq} (7+\lambda)r^d$$

where (i) follows from the definition of $\hat{b}$ in eq. (36). ■

Step 3: Characterizing subspaces $\mathcal{H}_x^K$ and $\mathcal{H}_y^K$.

In this step, we characterize the forms of the subspaces $\mathcal{H}_x^K$ and $\mathcal{H}_y^K$ for bilevel optimization algorithms considered in Theorem 2. Based on the constructions of f, g in eq. (15), we have

$$\nabla_x f(x,y) = (\alpha Z^2 + \mu_x I)x - \frac{\alpha\beta}{\widetilde{L}_{xy}} Z^3 y + \frac{\bar{L}_{xy}}{2} Zy$$
$$\nabla_y f(x,y) = -\frac{\alpha\beta}{\widetilde{L}_{xy}} Z^3 x + \frac{\bar{L}_{xy}}{2} Zx + L_y y + \frac{\bar{L}_{xy}}{\widetilde{L}_{xy}} b - \frac{2\alpha\beta}{\widetilde{L}_{xy}^2} Z^2 b$$
$$\nabla_x\nabla_y g(x,y) = -\frac{\widetilde{L}_{xy}}{2} Z,\ \nabla_y^2 g(x,y) = \beta Z^2 + \mu_y I,\ \nabla_y g(x,y) = (\beta Z^2 + \mu_y I)y - \frac{\widetilde{L}_{xy}}{2} Zx + b,$$

which, in conjunction with eq. (12) and eq. (13), yields

$$\begin{aligned} &\mathcal{H}_y^0 = \mathrm{Span}\{0\}, ..., \mathcal{H}_y^{s_0} = \mathrm{Span}\{Z^{2(s_0-1)}b, ..., Z^2 b, b\} \\ &\mathcal{H}_x^0 = ...\mathcal{H}_x^{s_0-1} = \mathrm{Span}\{0\}, \mathcal{H}_x^{s_0} \subseteq \mathrm{Span}\{Z^{2(T+s_0)}(Zb), ..., Z^2(Zb), (Zb)\}. \end{aligned} \tag{37}$$

Repeating the same steps as in eq. (37), it can be verified that

$$H_x^{s_{Q-1}} \subseteq \mathrm{Span}\{Z^{2(s_{Q-1}+QT+Q)}(Zb), ..., Z^{2j}(Zb), ..., Z^2(Zb), (Zb)\}. \tag{38}$$

Recall eq. (13) that $\mathcal{H}_x^K = \mathcal{H}_x^{s_{Q-1}}$ and $s_{Q-1} \le K$. Then, we obtain from eq. (38) that H_x^K satisfies

$$H_x^K \subseteq \mathrm{Span}\{Z^{2(K+QT+Q)}(Zb), ..., Z^2(Zb), (Zb)\}. \tag{39}$$

Step 4: Characterizing convergence and complexity.

Based on the results in Steps 1 and 2, we are now ready to provide a lower bound on the convergence rate and complexity of bilevel optimization algorithms. Let $M = K+QT+Q+2$ and $x_0 = \mathbf{0}$, and let the dimension d satisfy

$$d > \max\Big\{2M, M + 1 + \log_r\Big(\frac{\tau}{4(7+\lambda)}\Big)\Big\}. \tag{40}$$

Recall Theorem 15 that Zb has zeros at all coordinates with $t = 3, ..., d$. Then, based on the form of subspaces $\mathcal{H}_x^K$ in eq. (39) and using the zero-chain property in Theorem 5, we have x^K has zeros at the coordinates with $t = M+1, ..., d$, and hence

$$\|x^K - \hat{x}\| \ge \sqrt{\sum_{i=M+1}^{d} \|\hat{x}_i\|} = r^M\sqrt{r^2 + ... + r^{2(d-M)}} \overset{(i)}{\ge} \frac{r^M}{\sqrt{2}}\|\hat{x} - x_0\|, \tag{41}$$

where (i) follows from eq. (40). Then, based on Theorem 15 and eq. (40), we have

$$\|\hat{x} - x^*\| \le \frac{7+\lambda}{\tau} < \frac{r^M}{2\sqrt{2}} r \overset{(i)}{\le} \frac{r^M}{2\sqrt{2}}\|\hat{x} - x_0\|, \tag{42}$$

where (i) follows from the fact that $\|\hat{x} - x_0\| = \|\hat{x}\| \ge r$. Combining eq. (41) and eq. (42) further yields

$$\|x^K - x^*\| \ge \|x^K - \hat{x}\| - \|\hat{x} - x^*\| \ge \frac{r^M}{\sqrt{2}}\|\hat{x} - x_0\| - \frac{r^M}{2\sqrt{2}}\|\hat{x} - x_0\| = \frac{r^M}{2\sqrt{2}}\|\hat{x} - x_0\|. \tag{43}$$

In addition, note that

$$\|x^* - \hat{x}\| \leq \frac{7+\lambda}{\tau} r^d \overset{(40)}{\leq} \frac{1}{4} r \leq \frac{1}{4}\|\hat{x}\| \leq \frac{1}{4}\|\hat{x} - x^*\| + \frac{1}{4}\|x^*\|,$$

which, in conjunction with $\|x_0 - \hat{x}\| \geq \|x^* - x_0\| - \|x^* - \hat{x}\|$, yields

$$\|x_0 - \hat{x}\| \geq \frac{2}{3}\|x^* - x_0\|. \tag{44}$$

Combining eq. (43) and eq. (44) yields

$$\|x^K - x^*\| \geq \frac{\|x^* - x_0\|}{3\sqrt{2}} r^M. \tag{45}$$

Then, since the objective function $\Phi(x)$ is μ_x-strongly-convex, we have $\Phi(x^K) - \Phi(x^*) \geq \frac{\mu_x}{2}\|x^K - x^*\|^2$ and $\|x_0 - x^*\|^2 \geq \frac{2}{L_\Phi}(\Phi(x_0) - \Phi(x^*))$, and hence eq. (45) yields

$$\Phi(x^K) - \Phi(x^*) = \Omega\Big(\frac{\Phi(x_0) - \Phi(x^*)}{36\kappa_x} r^{2M}\Big). \tag{46}$$

Recall that r is the solution of the equation $1 - (4+\lambda)r + (6+2\lambda+\tau)r^2 - (4+\lambda)r^3 + r^4 = 0$. Based on Lemma 4.2 in Zhang et al. (2019), we have

$$1 - \frac{1}{\frac{1}{2} + \sqrt{\frac{\lambda}{2\tau} + \frac{1}{4}}} < r < 1, \tag{47}$$

which, in conjunction with the definitions of λ and τ in eq. (33) and the fact $\bar{L}_{xy} \geq 0$, yields the first result eq. (14) in Theorem 4. Then, in order to achieve an ϵ-accurate solution, i.e., $\Phi(x^K) - \Phi(x^*) \leq \epsilon$, it requires

$$\begin{aligned} M = K + QT + Q + 2 &= \Omega\Big(\frac{\log \frac{\Phi(x_0) - \Phi(x^*)}{\kappa_x \epsilon}}{2\log\frac{1}{r}}\Big) \\ &\overset{(i)}{=} \Omega\Big(\sqrt{\frac{\lambda}{2\tau}} \log \frac{\Phi(x_0) - \Phi(x^*)}{\epsilon \kappa_x}\Big) = \Omega\Big(\sqrt{\frac{L_y \widetilde{L}_{xy}^2}{\mu_x \mu_y^2}} \log \frac{\Phi(x_0) - \Phi(x^*)}{\kappa_x \epsilon}\Big), \end{aligned} \tag{48}$$

where (i) follows from eq. (47). Recall that the complexity measure is given by $\mathcal{C}_{\text{fun}}(\mathcal{A}, \epsilon) = \Omega(n_J + n_H + n_G)$, where the numbers n_J, n_H of Jacobian- and Hessian-vector products are given by $n_J = Q$ and $n_H = QT$ and the number n_G of gradient evaluations is given by $n_G = K$. Then, the total complexity $\mathcal{C}_{\text{fun}}(\mathcal{A}, \epsilon) = \Omega(Q + QT + K)$, which combined with eq. (48) implies

$$\mathcal{C}_{\text{fun}}(\mathcal{A}, \epsilon) = \Omega\Big(\sqrt{\frac{L_y \widetilde{L}_{xy}^2}{\mu_x \mu_y^2}} \log \frac{\Phi(x_0) - \Phi(x^*)}{\kappa_x \epsilon}\Big).$$

Then, the proof is complete.

Appendix C. Proof of Theorem 6

In this section, we provide the proof for Theorem 6 under the convex-strongly-convex geometry. The proof is divided into the following steps: 1) constructing the worst-case instance that belongs to the convex-strongly-convex problem class $\mathcal{F}_{csc}$ defined in Theorem 1; 2) characterizing $x^* \in \arg\min_{x\in\mathbb{R}^d} \Phi(x)$; 3) developing the lower bound on the gradient norm $\|\nabla\Phi(x)\|$ when the last several coordinates of x are zeros; 4) characterizing the subspaces $\mathcal{H}_x^k$ and $\mathcal{H}_x^k$; and 5) characterizing the convergence and complexity.

Step 1: Constructing the worst-case instance that satisfies Theorem 1.

It can be verified that the constructed f, g in eq. (20) satisfy eq. (2) (3) and (4) in Assumptions 1 and 2. Then, similarly to the proof of Theorem 4, we have $y^*(x) = (\beta Z^2 + \mu_y I)^{-1}(\frac{\widetilde{L}_{xy}}{2}Zx - b)$ and hence $\Phi(x) = f(x, y^*(x))$ takes the form of

$$\Phi(x) = \frac{L_x}{8}x^T Z^2 x + \frac{L_y}{2}\Big(\frac{\widetilde{L}_{xy}}{2}Zx - b\Big)^T(\beta Z^2 + \mu_y I)^{-2}\Big(\frac{\widetilde{L}_{xy}}{2}Zx - b\Big),$$

which can be verified to be convex.

Step 2: Characterizing x^*.

Note that the gradient $\nabla\Phi(x)$ is given by

$$\nabla\Phi(x) = \frac{L_x}{4}Z^2 x + \frac{L_y\widetilde{L}_{xy}}{2}Z(\beta Z^2 + \mu_y I)^{-2}\Big(\frac{\widetilde{L}_{xy}}{2}Zx - b\Big). \tag{49}$$

Then, setting $\nabla\Phi(x^*) = 0$ and using eq. (30), we have

$$\Big(\frac{L_x\beta^2}{4}Z^6 + \frac{L_x\beta^2\beta\mu_y}{2}Z^4 + \Big(\frac{L_y\widetilde{L}_{xy}^2}{4} + \frac{L_x\mu_y^2}{4}\Big)Z^2\Big)x^* = \frac{L_y\widetilde{L}_{xy}}{2}Zb. \tag{50}$$

Let $\widetilde{b} = \frac{L_y\widetilde{L}_{xy}}{2}Zb$, and we choose b such that $\widetilde{b}_t = 0$ for $t = 4, ..., d$ and

$$\begin{aligned}
\widetilde{b}_1 =& \frac{B}{\sqrt{d}}\Big(\frac{5}{4}L_x\beta^2 + L_x\beta\mu_y + \frac{\widetilde{L}_{xy}^2 L_y}{4} + \frac{L_x}{4}\mu_y^2\Big), \\
\widetilde{b}_2 =& \frac{B}{\sqrt{d}}(-L_x\beta^2 - \frac{L_x\beta}{2}\mu_y),\ \widetilde{b}_3 = \frac{B}{\sqrt{d}}\frac{L_x\beta^2}{4},
\end{aligned} \tag{51}$$

where the selection of b is achievable because Z is invertible with Z^{-1} given by

$$Z^{-1} = \begin{bmatrix} & & & -1 \\ & & -1 & -1 \\ & \iddots & \iddots & \vdots \\ -1 & -1 & -1 & -1 \end{bmatrix}.$$

Based on the form of Z^2 in eq. (21) and the forms of Z^4, Z^6 given by

$$Z^4 = \begin{bmatrix} 5 & -4 & 1 & & & \\ -4 & 6 & -4 & 1 & & \\ 1 & -4 & 6 & -4 & 1 & \\ & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & 1 & -4 & 6 & -3 \\ & & & 1 & -3 & 2 \end{bmatrix}, \; Z^6 = \begin{bmatrix} 14 & -14 & 6 & -1 & & & & \\ -14 & 20 & -15 & 6 & -1 & & & \\ 6 & -15 & 20 & -15 & 6 & -1 & & \\ -1 & 6 & -15 & 20 & -15 & 6 & -1 & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & -1 & 6 & -15 & 20 & -15 & 5 \\ & & & -1 & 6 & -15 & 19 & -9 \\ & & & & -1 & 5 & -9 & 5 \end{bmatrix}, \quad (52)$$

it can be checked from eq. (50) that $x^* = \frac{B}{\sqrt{d}}\mathbf{1}$, where $\mathbf{1}$ denotes the all-one vector and thus $\|x^*\| = B$.

Step 3: Characterizing lower bound on $\|\nabla\Phi(x)\|$.

Next, we characterize a lower bound on $\|\nabla\Phi(x)\|$ when the last three coordinates of x are zeros, i.e., $x_{d-2} = x_{d-1} = x_d = 0$. Let $\Omega = [I_{d-3}, \mathbf{0}]^T$ and define $\widetilde{x} \in \mathbb{R}^{d-3}$ such that $\widetilde{x}_i = x_i$ for $i = 1, ..., d-3$. Then for any matrix H, $H\Omega$ is equivalent to removing the last three columns of H. Then, based on the form of $\nabla\Phi(x)$ in eq. (49), we have

$$\min_{x\in\mathbb{R}^d: x_{d-2}=x_{d-1}=x_d=0} \|\nabla\Phi(x)\|^2 = \min_{\widetilde{x}\in\mathbb{R}^{d-3}} \|H\Omega\widetilde{x} - (\beta Z^2 + \mu_y I)^{-2}\widetilde{b}\|^2 \quad (53)$$

where the matrix H is given by

$$H = (\beta Z^2 + \mu_y I)^{-2} \underbrace{\Big(\frac{L_x\beta^2}{4} Z^6 + \frac{L_x\beta^2\beta\mu_y}{2} Z^4 + \Big(\frac{L_y\widetilde{L}_{xy}^2}{4} + \frac{L_x\mu_y^2}{4}\Big) Z^2\Big)}_{\widetilde{H}}. \quad (54)$$

Then using an approach similar to (7) in Carmon et al. (2019), we have

$$\min_{\widetilde{x}\in\mathbb{R}^{d-3}} \|H\Omega\widetilde{x} - (\beta Z^2 + \mu_y I)^{-2}\widetilde{b}\|^2 = \big(\widetilde{b}^T(\beta Z^2 + \mu_y I)^{-2} z\big)^2, \quad (55)$$

where z is the normalized (i.e., $\|z\| = 1$) solution of equation $(H\Omega)^T z = 0$. Next we characterize the solution z. Since $H = (\beta Z^2 + \mu_y I)^{-2}\widetilde{H}$, we have

$$(H\Omega)^T z = (\widetilde{H}\Omega)^T (\beta Z^2 + \mu_y I)^{-2} z = 0. \quad (56)$$

Based on the definition of $\widetilde{H}$ in eq. (54) and the forms of Z^2, Z^4, Z^6 in eq. (21) and eq. (52), we have that the solution z takes the form of $z = \lambda(\beta Z^2 + \mu_y I)^2 h$, where λ is a factor such that $\|z\| = 1$ and h is a vector satisfying $h_t = t$ for $t = 1, ..., d$. Based on the definition of Z^2 in eq. (21), we have

$$\begin{aligned} 1 = \|z\| =& \lambda\sqrt{\sum_{i=1}^{d-2}(i\mu_y^2)^2 + ((d-1)\mu_y^2 - \beta^2)^2 + (d\mu_y^2 + \beta^2 + 2\beta\mu_y)^2} \\ \leq& \lambda\sqrt{\sum_{i=1}^{d-2}(i\mu_y^2)^2 + 2(d-1)^2\mu_y^4 + 2\beta^4 + 2d^2\mu_y^4 + 2(\beta^2 + 2\beta\mu_y)^2} \end{aligned}$$

$$<\lambda\sqrt{\frac{2}{3}\mu_y^4(d+1)^3+4\beta^4+8\beta^3\mu_y+8\beta^2\mu_y^2},$$

which further implies that

$$\lambda>\frac{1}{\sqrt{\frac{2}{3}\mu_y^4(d+1)^3+4\beta^4+8\beta^3\mu_y+8\beta^2\mu_y^2}}. \tag{57}$$

Then, combining eq. (53), eq. (55) and eq. (57) yields

$$\begin{aligned}
\min_{x:x_{d-2}=x_{d-1}=x_d=0}\|\nabla\Phi(x)\|^2 =&\Big(\widetilde{b}^T(\beta Z^2+\mu_y I)^{-2}z\Big)^2=(\lambda\widetilde{b}^T h)^2=\lambda^2(\widetilde{b}_1+2\widetilde{b}_2+3\widetilde{b}_3)^2\\
\overset{(i)}{=}&\lambda^2\frac{B^2}{4d}\Big(\frac{\widetilde{L}_{xy}^2L_y}{4}+\frac{L_x\mu_y^2}{4}\Big)^2\\
\geq&\frac{B^2\Big(\frac{\widetilde{L}_{xy}^2L_y}{4}+\frac{L_x\mu_y^2}{4}\Big)^2}{\frac{8}{3}\mu_y^4d(d+1)^3+16d\beta^4+32d\beta^3\mu_y+32d\beta^2\mu_y^2}\\
\overset{(ii)}{\geq}&\frac{B^2\Big(\frac{\widetilde{L}_{xy}^2L_y}{4}+\frac{L_x\mu_y^2}{4}\Big)^2}{8\mu_y^4d^4+16d\beta^4+32d\beta^3\mu_y+32d\beta^2\mu_y^2}
\end{aligned} \tag{58}$$

where (i) follows from the definition of $\widetilde{b}$ in eq. (51), and (ii) follows because $d\geq 3$.

Step 4: Characterizing subspaces $\mathcal{H}_x^k$ and $\mathcal{H}_x^k$.

Based on the constructions of f,g in eq. (20), we have

$$\begin{aligned}
&\nabla_x f(x,y)=\frac{L_x}{4}Z^2x,\ \nabla_y f(x,y)=L_y y,\ \nabla_x\nabla_y g(x,y)=-\frac{\widetilde{L}_{xy}}{2}Z\\
&\nabla_y^2 g(x,y)=\beta Z^2+\mu_y I,\ \nabla_y g(x,y)=(\beta Z^2+\mu_y I)y-\frac{\widetilde{L}_{xy}}{2}Zx+b,
\end{aligned}$$

which, in conjunction with eq. (12) and eq. (13), yields

$$\begin{aligned}
&\mathcal{H}_y^0=\mathrm{Span}\{0\},....,\mathcal{H}_y^{s_0}=\mathrm{Span}\{Z^{2(s_0-1)}b,...,Z^2b,b\}\\
&\mathcal{H}_x^0=....\mathcal{H}_x^{s_0-1}=\mathrm{Span}\{0\},\mathcal{H}_x^{s_0}=\mathrm{Span}\{Z^{2(T+s_0-2)}(Zb),....,Z^2(Zb),(Zb)\}.
\end{aligned}$$

Repeating the above procedure and noting that $s_{Q-1}\leq K$ yield

$$\begin{aligned}
\mathcal{H}_x^K=\mathcal{H}_x^{s_{Q-1}}&=\mathrm{Span}\{Z^{2(s_{Q-1}+QT-Q-1)}(Zb),....,Z^2(Zb),(Zb)\}\\
&\subseteq\mathrm{Span}\{Z^{2(K+QT-Q)}(Zb),....,Z^2(Zb),(Zb)\}.
\end{aligned} \tag{59}$$

Step 5: Characterizing convergence and complexity.

Let $M=K+QT-Q+3$ and consider the following equation

$$r^4+r\Big(\frac{2\beta^4}{\mu_y^4}+\frac{4\beta^3}{\mu_y^3}+\frac{4\beta^2}{\mu_y^2}\Big)=\frac{B^2\Big(\widetilde{L}_{xy}^2L_y+L_x\mu_y^2\Big)^2}{128\mu_y^4\epsilon^2}, \tag{60}$$

which has a solution denoted as r^*. We choose $d = \lfloor r^* \rfloor$. Then, based on eq. (58), we have

$$\min_{x:x_{d-2}=x_{d-1}=x_d=0} \|\nabla\Phi(x)\|^2 \geq \frac{B^2\Big(\frac{\widetilde{L}_{xy}^2 L_y}{4} + \frac{L_x\mu_y^2}{4}\Big)^2}{8\mu_y^4(r^*)^4 + 16r^*\beta^4 + 32r^*\beta^3\mu_y + 32r^*\beta^2\mu_y^2} = \epsilon^2. \tag{61}$$

Then, to achieve $\|\nabla\Phi(x^K)\| < \epsilon$, it requires that $M > d-3$. Otherwise (i.e., if $M \leq d-3$), based on eq. (59) and the fact that Zb has nonzeros only at the first three coordinates, we have x^K has zeros at the last three coordinates, and hence eq. (61) yields $\|\nabla\Phi(x^K)\| \geq \epsilon$, which leads to a contradiction. Therefore, we have $M > \lfloor r^* \rfloor - 3$.

To characterize the total complexity, using the metric in Theorem 3, we have

$$\mathcal{C}_{\text{grad}}(\mathcal{A}, \epsilon) = \Omega(Q + QT + K) = \Omega(M) = \Omega(r^*).$$

Then, the proof is complete.

Appendix D. Proof of Theorem 7

In this case, the condition number κ_y satisfies $\kappa_y = \frac{\widetilde{L}_y}{\mu_y} = \Theta(1)$. Then, it can be verified that r^* satisfies $(r^*)^3 = \Omega(\frac{2\beta^4}{\mu_y^4} + \frac{4\beta^3}{\mu_y^3} + \frac{4\beta^2}{\mu_y^2})$, and hence it follows from eq. (19) that

$$\mathcal{C}_{\text{grad}}(\mathcal{A}, \epsilon) \geq r^* = \Omega\Big(\frac{B^{\frac{1}{2}}(\widetilde{L}_{xy}^2 L_y + L_x\mu_y^2)^{\frac{1}{2}}}{\mu_y\epsilon^{\frac{1}{2}}}\Big).$$

Appendix E. Proof of Theorem 8

In this case, $\widetilde{L}_y = \Theta(1)$. Then, we consider two cases $\frac{1}{\kappa_y} = \Omega(\epsilon^{\frac{3}{2}})$ and $\frac{1}{\kappa_y} = \mathcal{O}(\epsilon^{\frac{3}{2}})$ separately.

Case 1: $\frac{1}{\kappa_y} = \Omega(\epsilon^{\frac{3}{2}})$. For this case, we have $\big(\frac{2\beta^4}{\mu_y^4} + \frac{4\beta^3}{\mu_y^3} + \frac{4\beta^2}{\mu_y^2}\big) = \mathcal{O}\big(\frac{\kappa_y^3}{\epsilon^{3/2}}\big)$. Then, it follows from eq. (19) that $\mathcal{C}_{\text{grad}}(\mathcal{A}, \epsilon) \geq r^* = \Omega\big(\frac{\kappa_y}{\epsilon^{1/2}}\big)$.

Case 2: $\frac{1}{\kappa_y} = \mathcal{O}(\epsilon^{\frac{3}{2}})$. For this case, first suppose $(r^*)^3 = \mathcal{O}\big(\frac{2\beta^4}{\mu_y^4} + \frac{4\beta^3}{\mu_y^3} + \frac{4\beta^2}{\mu_y^2}\big)$, and then it follows from eq. (19) that $r^* = \Omega(\frac{1}{\epsilon^2})$. On the other hand, if $(r^*)^3 = \Omega\big(\frac{2\beta^4}{\mu_y^4} + \frac{4\beta^3}{\mu_y^3} + \frac{4\beta^2}{\mu_y^2}\big)$, then we obtain from eq. (19) that $r^* = \Omega(\frac{\kappa_y}{\epsilon^{1/2}}) = \Omega(\frac{1}{\epsilon^2})$, which yields $\mathcal{C}_{\text{grad}}(\mathcal{A}, \epsilon) \geq r^* = \Omega(\frac{1}{\epsilon^2})$. Then, combining these two cases finishes the proof.

Appendix F. Proof of Theorem 9

To simplify the notations, we define the following quantities.

$$\begin{aligned}
\mathcal{M}_k =& \|y^*(x^*)\| + \frac{\widetilde{L}_{xy}}{\mu_y}\|x_k - x^*\|, \;\; \mathcal{N}_k = \|\nabla_y f(x^*, y^*(x^*))\| + \Big(L_{xy} + \frac{L_y\widetilde{L}_{xy}}{\mu_y}\Big)\|x_k - x^*\| \\
\mathcal{M}_* =& \|y^*(x^*)\| + \frac{3\widetilde{L}_{xy}}{\mu_y}\sqrt{\frac{2}{\mu_x}(\Phi(0) - \Phi(x^*)) + \|x^*\|^2 + \frac{\epsilon}{\mu_x}} \\
\mathcal{N}_* =& \|\nabla_y f(x^*, y^*(x^*))\| + 3\Big(L_{xy} + \frac{L_y\widetilde{L}_{xy}}{\mu_y}\Big)\sqrt{\frac{2}{\mu_x}(\Phi(0) - \Phi(x^*)) + \|x^*\|^2 + \frac{\epsilon}{\mu_x}},
\end{aligned} \tag{62}$$

where $\mathcal{M}_k$ and $\mathcal{N}_k$ change with the optimality gap $\|x_k - x^*\|$ at the k^{th} iteration, and $\mathcal{M}_*$ and $\mathcal{N}_*$ are two positive constants depending on the information of the objective function at the optimal point x^*. We first establish the following lemma to upper-bound the hypergradient estimation error $\|\nabla\Phi(x_k) - G_k\|$.

Lemma 16 *Let G_k be the hypergradient estimator used in Algorithm 1 at iteration k. Then, we have*

$$\begin{aligned}\|G_k - \nabla\Phi(x_k)\| \leq & \sqrt{\frac{\widetilde{L}_y + \mu_y}{\mu_y}}\Big(L_y + \frac{2\widetilde{L}_{xy}L_y}{\mu_y} + \Big(\frac{\rho_{xy}}{\mu_y} + \frac{\widetilde{L}_{xy}\rho_{yy}}{\mu_y^2}\Big)\mathcal{N}_k\Big)\mathcal{M}_k \exp\Big(-\frac{N}{2\sqrt{\kappa_y}}\Big) \\ & + \frac{\widetilde{L}_{xy}}{\mu_y}\Big(\frac{\sqrt{\kappa_y}-1}{\sqrt{\kappa_y}+1}\Big)^M \mathcal{N}_k, \end{aligned} \tag{63}$$

where the quantities $\mathcal{M}_k$ and $\mathcal{N}_k$ are defined in eq. (62).

Theorem 16 shows that the estimation error $\|\nabla\Phi(x_k) - G_k\|$ is bounded given that the optimality gap $\|x_k - x^*\|$ is bounded. We will show in the proof of Theorem 9 that $\|x_k - x^*\|$ is bounded as the algorithm runs due to the strongly-convex geometry of the objective function $\Phi(x)$. In addition, it can be seen that this error decays exponentially with respect to the number N of inner-level steps and the number M of steps of the heavy-ball method for solving the linear system in Algorithm 1. Then, to prove the convergence of Algorithm 1, we set $N = M = c\sqrt{\kappa_y}\log(\kappa_y)$ in the proof of Theorem 9, where c is a constant independent of κ_y.

Proof Recall line 7 of Algorithm 1 that

$$G_k := \nabla_x f(x_k, y_k^N) - \nabla_x\nabla_y g(x_k, y_k^N) v_k^M, \tag{64}$$

where v_k^M is the M^{th} step output of the heavy-ball method for solving

$$\min_v Q(v) := \frac{1}{2} v^T \nabla_y^2 g(x_k, y_k^N) v - v^T \nabla_y f(x_k, y_k^N).$$

Recall the smoothness parameter $\widetilde{L}_y$ of $g(x,\cdot)$ defined in Assumption 1. Then, based on the convergence result of the heavy-ball method in Badithela and Seiler (2019) with stepsizes $\lambda = \frac{4}{(\sqrt{\widetilde{L}_y} + \sqrt{\mu_y})^2}$ and $\theta = \max\big\{\big(1 - \sqrt{\lambda\mu_y}\big)^2, \big(1 - \sqrt{\lambda\widetilde{L}_y}\big)^2\big\}$ and noting that $v_k^0 = v_k^1 = 0$, we have

$$\begin{aligned}\|v_k^M & - \nabla_y^2 g(x_k, y_k^N)^{-1}\nabla_y f(x_k, y_k^N)\| \\ & \leq \Big(\frac{\sqrt{\kappa_y}-1}{\sqrt{\kappa_y}+1}\Big)^M \Big\|\big(\nabla_y^2 g(x_k, y_k^N)\big)^{-1}\nabla_y f(x_k, y_k^N)\Big\| \\ & \leq \frac{L_y}{\mu_y}\Big(\frac{\sqrt{\kappa_y}-1}{\sqrt{\kappa_y}+1}\Big)^M \|y^*(x_k) - y_k^N\| + \frac{\|\nabla_y f(x_k, y^*(x_k))\|}{\mu_y}\Big(\frac{\sqrt{\kappa_y}-1}{\sqrt{\kappa_y}+1}\Big)^M \\ & \overset{(i)}{\leq} \frac{L_y}{\mu_y}\|y^*(x_k) - y_k^N\| + \frac{\|\nabla_y f(x_k, y^*(x_k))\|}{\mu_y}\Big(\frac{\sqrt{\kappa_y}-1}{\sqrt{\kappa_y}+1}\Big)^M \end{aligned} \tag{65}$$

where $y^*(x_k) = \arg\min_{y\in\mathbb{R}^q} g(x_k, y)$ and (i) follows because $\frac{\sqrt{\kappa_y}-1}{\sqrt{\kappa_y}+1} \leq 1$. Then, based on the forms of G_k and $\nabla\Phi(x)$ in eq. (64) and eq. (7), and using Assumptions 1 and 2, we have

$$
\begin{aligned}
&\|G_k - \nabla\Phi(x_k)\| \\
&\overset{(i)}{\leq}\|\nabla_x f(x_k, y_k^N) - \nabla_x f(x_k, y^*(x_k))\| + \widetilde{L}_{xy}\|v_k^M - \nabla_y^2 g(x_k, y^*(x_k))^{-1}\nabla_y f(x_k, y^*(x_k))\| \\
&\quad + \frac{\|\nabla_y f(x_k, y^*(x_k))\|}{\mu_y}\|\nabla_x\nabla_y g(x_k, y_k^N) - \nabla_x\nabla_y g(x_k, y^*(x_k))\| \\
&\leq L_y\|y^*(x_k) - y_k^N\| + \widetilde{L}_{xy}\|v_k^M - \nabla_y^2 g(x_k, y_k^N)^{-1}\nabla_y f(x_k, y_k^N)\| \\
&\quad + \widetilde{L}_{xy}\big\|\nabla_y^2 g(x_k, y_k^N)^{-1}\nabla_y f(x_k, y_k^N) - \nabla_y^2 g(x_k, y^*(x_k))^{-1}\nabla_y f(x_k, y^*(x_k))\big\| \\
&\quad + \frac{\rho_{xy}}{\mu_y}\|y_k^N - y^*(x_k)\|\|\nabla_y f(x_k, y^*(x_k))\| \\
&\leq\Big(L_y + \frac{\widetilde{L}_{xy}L_y}{\mu_y} + \frac{\rho_{xy}}{\mu_y}\|\nabla_y f(x_k, y^*(x_k))\|\Big)\|y_k^N - y^*(x_k)\| \\
&\quad + \frac{\widetilde{L}_{xy}\rho_{yy}\|y_k^N - y^*(x_k)\|}{\mu_y^2}\|\nabla_y f(x_k, y^*(x_k))\| + \widetilde{L}_{xy}\|v_k^M - \nabla_y^2 g(x_k, y_k^N)^{-1}\nabla_y f(x_k, y_k^N)\| \\
&\overset{(ii)}{\leq}\Big(L_y + \frac{2\widetilde{L}_{xy}L_y}{\mu_y} + \Big(\frac{\rho_{xy}}{\mu_y} + \frac{\widetilde{L}_{xy}\rho_{yy}}{\mu_y^2}\Big)\|\nabla_y f(x_k, y^*(x_k))\|\Big)\|y_k^N - y^*(x_k)\| \\
&\quad + \frac{\widetilde{L}_{xy}}{\mu_y}\left(\frac{\sqrt{\kappa_y}-1}{\sqrt{\kappa_y}+1}\right)^M\|\nabla_y f(x_k, y^*(x_k))\|,
\end{aligned}
\tag{66}
$$

where (i) follows from Assumption 1 that $\|\nabla_x\nabla_y g(\cdot,\cdot)\| \leq \widetilde{L}_{xy}$ and $\|(\nabla_y^2 g(\cdot,\cdot))^{-1}\| \leq \frac{1}{\mu_y}$ and (ii) follows from eq. (65). Note that y_k^N is obtained as the N-step output of AGD for minimizing the inner-level loss function $g(x_k,\cdot)$ and recall $y^*(x_k) = \arg\min_{y\in\mathbb{R}^q} g(x_k, y)$. Then, based on the analysis in Nesterov (2003) for AGD, we have

$$
\begin{aligned}
\|y_k^N - y^*(x_k)\| &\leq\sqrt{\frac{\widetilde{L}_y + \mu_y}{\mu_y}}\|y_k^0 - y^*(x_k)\|\exp\Big(-\frac{N}{2\sqrt{\kappa_y}}\Big) \\
&\leq\sqrt{\frac{\widetilde{L}_y + \mu_y}{\mu_y}}\Big(\|y^*(x^*)\| + \frac{\widetilde{L}_{xy}}{\mu_y}\|x_k - x^*\|\Big)\exp\Big(-\frac{N}{2\sqrt{\kappa_y}}\Big),
\end{aligned}
\tag{67}
$$

where $x^* = \arg\min_{x\in\mathbb{R}^p}\Phi(x)$. Moreover, based on Lemma 2.2 in Ghadimi and Wang (2018), we have $\|y^*(x_1) - y^*(x_2)\| \leq \frac{\widetilde{L}_{xy}}{\mu_y}\|x_1 - x_2\|$ for any $x_1, x_2 \in \mathbb{R}^p$, and hence

$$
\|\nabla_y f(x_k, y^*(x_k))\| \leq \|\nabla_y f(x^*, y^*(x^*))\| + \Big(L_{xy} + \frac{L_y\widetilde{L}_{xy}}{\mu_y}\Big)\|x_k - x^*\|. \tag{68}
$$

Substituting eq. (67) and eq. (68) into eq. (66), and using the definition of $\mathcal{M}_k$ and $\mathcal{N}_k$ in eq. (62), we have

$$
\|G_k - \nabla\Phi(x_k)\| \leq\sqrt{\frac{\widetilde{L}_y + \mu_y}{\mu_y}}\Big(L_y + \frac{2\widetilde{L}_{xy}L_y}{\mu_y} + \Big(\frac{\rho_{xy}}{\mu_y} + \frac{\widetilde{L}_{xy}\rho_{yy}}{\mu_y^2}\Big)\mathcal{N}_k\Big)\mathcal{M}_k\exp\Big(-\frac{N}{2\sqrt{\kappa_y}}\Big)
$$

$$+\frac{\widetilde{L}_{xy}}{\mu_y}\Big(\frac{\sqrt{\kappa_y}-1}{\sqrt{\kappa_y}+1}\Big)^M\mathcal{N}_k,$$

which completes the proof. ■

We then establish the following lemma to characterize the smoothness parameter of the objective function $\Phi(x)$ around the iterate x_k. Recall eq. (7) that $\nabla\Phi(x)$ is given by

$$\nabla\Phi(x)=\nabla_x f(x,y^*(x))-\nabla_x\nabla_y g(x,y^*(x))[\nabla_y^2 g(x,y^*(x))]^{-1}\nabla_y f(x,y^*(x)), \tag{69}$$

where $y^*(x)=\arg\min_y g(x,\cdot)$ denotes the minimizer of the inner-level function $g(x,\cdot)$.

Lemma 17 *Consider the hypergradient $\nabla\Phi(x)$ given by eq. (69). For any $x\in\mathbb{R}^p$, we have*

$$\begin{aligned}&\|\nabla\Phi(x)-\nabla\Phi(x_k)\|\\ &\quad\leq\Big(\underbrace{L_x+\frac{2L_{xy}\widetilde{L}_{xy}}{\mu_y}+\frac{L_y\widetilde{L}_{xy}^2}{\mu_y^2}+\Big(\frac{\widetilde{L}_{xy}\rho_{yy}}{\mu_y^2}+\frac{\rho_{xy}}{\mu_y}\Big)\Big(1+\frac{\widetilde{L}_{xy}}{\mu_y}\Big)\mathcal{N}_k}_{L_{\Phi_k}}\Big)\|x-x_k\|,\end{aligned} \tag{70}$$

where $\mathcal{N}_k$ is defined in eq. (62). Furthermore, eq. (70) implies that, for any $x\in\mathbb{R}^p$,

$$\Phi(x)\leq\Phi(x_k)+\langle\nabla\Phi(x_k),x-x_k\rangle+\frac{L_{\Phi_k}}{2}\|x-x_k\|^2. \tag{71}$$

Theorem 17 shows that $\nabla\Phi(x)$ is Lipschitz continuous around the iterate x_k, i.e., $\Phi(x)$ is smooth, where the smoothness parameter L_{Φ_k} contains a term proportional to $\|x_k-x^*\|$. We will show in the proof of Theorem 9 that the optimality distance $\|x_k-x^*\|$ is bounded as the algorithm runs, and hence the smoothness parameter L_{Φ_k} is bounded by $\mathcal{O}(\frac{1}{\mu_y^3})$ during the entire process.

Proof Based on the form of $\nabla\Phi(x)$ in eq. (69), we have

$$\begin{aligned}&\|\nabla\Phi(x)-\nabla\Phi(x_k)\|\\ &\leq\|\nabla_x f(x,y^*(x))-\nabla_x f(x_k,y^*(x_k))\|+\frac{\widetilde{L}_{xy}}{\mu_y}\|\nabla_y f(x,y^*(x))-\nabla_y f(x_k,y^*(x_k))\|\\ &\quad+\underbrace{\|\nabla_x\nabla_y g(x,y^*(x))\nabla_y^2 g(x,y^*(x))^{-1}-\nabla_x\nabla_y g(x_k,y^*(x_k))\nabla_y^2 g(x_k,y^*(x_k))^{-1}\|}_{P}\|\nabla_y f(x_k,y^*(x_k))\|,\end{aligned}$$

which, in conjunction with the inequality

$$\begin{aligned}P\leq&\frac{\widetilde{L}_{xy}\rho_{yy}}{\mu_y^2}(\|x-x_k\|+\|y^*(x)-y^*(x_k)\|)+\frac{\rho_{xy}}{\mu_y}(\|x-x_k\|+\|y^*(x)-y^*(x_k)\|)\\ \overset{(i)}{\leq}&\Big(\frac{\widetilde{L}_{xy}\rho_{yy}}{\mu_y^2}+\frac{\rho_{xy}}{\mu_y}\Big)\Big(1+\frac{\widetilde{L}_{xy}}{\mu_y}\Big)\|x-x_k\|,\end{aligned}$$

and using Assumption 1, yields

$$\|\nabla\Phi(x)-\nabla\Phi(x_k)\|\leq\Big(L_x+\frac{2L_{xy}\widetilde{L}_{xy}}{\mu_y}+\frac{L_y\widetilde{L}_{xy}^2}{\mu_y^2}\Big)\|x-x_k\|$$

$$+\Big(\frac{\widetilde{L}_{xy}\rho_{yy}}{\mu_y^2}+\frac{\rho_{xy}}{\mu_y}\Big)\Big(1+\frac{\widetilde{L}_{xy}}{\mu_y}\Big)\|\nabla_y f(x_k,y^*(x_k))\|\|x-x_k\|, \tag{72}$$

where (i) follows from the $\frac{\widetilde{L}_{xy}}{\mu_y}$-smoothness of $y^*(\cdot)$. Substituting eq. (68) into eq. (72) and using the definition of $\mathcal{N}_k$ in eq. (62), we have

$$\begin{aligned}&\|\nabla\Phi(x)-\nabla\Phi(x_k)\|\\ &\quad\leq\underbrace{\Big(L_x+\frac{2L_{xy}\widetilde{L}_{xy}}{\mu_y}+\frac{L_y\widetilde{L}_{xy}^2}{\mu_y^2}+\Big(\frac{\widetilde{L}_{xy}\rho_{yy}}{\mu_y^2}+\frac{\rho_{xy}}{\mu_y}\Big)\Big(1+\frac{\widetilde{L}_{xy}}{\mu_y}\Big)\mathcal{N}_k\Big)}_{L_{\Phi_k}}\|x-x_k\|.\end{aligned} \tag{73}$$

Based on eq. (73), we further obtain

$$\begin{aligned}|\Phi(x)-\Phi(x_k)-&\langle\nabla\Phi(x_k),x-x_k\rangle|\\ &=\Big|\int_0^1\langle\nabla\Phi(x_k+t(x-x_k)),x-x_k\rangle dt-\langle\nabla\Phi(x_k),x-x_k\rangle\Big|\\ &\leq\Big|\int_0^1\langle\nabla\Phi(x_k+t(x-x_k))-\nabla\Phi(x_k),x-x_k\rangle dt\Big|\\ &\leq\Big|\int_0^1\|\nabla\Phi(x_k+t(x-x_k))-\nabla\Phi(x_k)\|\|x-x_k\|dt\Big|\\ &\leq\Big|\int_0^1 L_{\Phi_k}\|(x-x_k)\|^2tdt\Big|=\frac{L_{\Phi_k}}{2}\|x-x_k\|^2.\end{aligned}$$

Then, the proof is now complete. ■

Based on Theorem 16 and Theorem 17, we are ready to prove Theorem 9.

Proof [**Proof of Theorem 9**] Algorithm 1 conducts the following updates

$$\begin{aligned}z_{k+1}=&x_k-\frac{1}{L_\Phi}G_k,\\ x_{k+1}=&\Big(1+\frac{\sqrt{\kappa_x}-1}{\sqrt{\kappa_x}+1}\Big)z_{k+1}-\frac{\sqrt{\kappa_x}-1}{\sqrt{\kappa_x}+1}z_k,\end{aligned} \tag{74}$$

where the smoothness parameter L_Φ takes the form of

$$\begin{aligned}L_\Phi=&L_x+\frac{2L_{xy}\widetilde{L}_{xy}}{\mu_y}+\frac{L_y\widetilde{L}_{xy}^2}{\mu_y^2}+\Big(\frac{\widetilde{L}_{xy}\rho_{yy}}{\mu_y^2}+\frac{\rho_{xy}}{\mu_y}\Big)\Big(1+\frac{\widetilde{L}_{xy}}{\mu_y}\Big)\|\nabla_y f(x^*,y^*(x^*))\|\\ &+3\Big(\frac{\widetilde{L}_{xy}\rho_{yy}}{\mu_y^2}+\frac{\rho_{xy}}{\mu_y}\Big)\Big(1+\frac{\widetilde{L}_{xy}}{\mu_y}\Big)\Big(L_{xy}+\frac{L_y\widetilde{L}_{xy}}{\mu_y}\Big)\sqrt{\frac{2}{\mu_x}(\Phi(0)-\Phi(x^*))+\|x^*\|^2+\frac{\epsilon}{\mu_x}}\\ =&\Theta\Big(L_x+\frac{L_y\widetilde{L}_{xy}^2}{\mu_y^2}+\frac{\widetilde{L}_{xy}^2\rho_{xy}}{\mu_y^3}\|\nabla_y f(x^*,y^*(x^*))\|\\ &\qquad+\frac{\widetilde{L}_{xy}^3\rho_{yy}L_y}{\mu_y^4}\sqrt{\frac{2}{\mu_x}(\Phi(0)-\Phi(x^*))+\|x^*\|^2+\frac{\epsilon}{\mu_x}}\Big),\end{aligned} \tag{75}$$

and $\kappa_x = \frac{L_\Phi}{\mu_x}$ is the condition number of the objective function $\Phi(x)$.

The remaining proof adapts the results in Section 2.2.5 of Nesterov et al. (2018), but with two key differences: we need to (a) prove the boundedness of the iterates as the algorithm runs, and (b) carefully handle the hypergradient estimation error in the convergence analysis for accelerated gradient methods. In specific, we first construct the estimate sequences as follows.

$$\begin{aligned} S_0(x) =&\Phi(x_0) + \frac{\mu_x}{2}\|x-x_0\|^2 \\ S_{k+1}(x) =&\Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)S_k(x) + \frac{1}{\sqrt{\kappa_x}}\Big(\Phi(x_k) + \langle G_k, x-x_k\rangle + \frac{\mu_x}{2}\|x-x_k\|^2 + \frac{\epsilon}{4}\Big). \end{aligned} \tag{76}$$

Note that $\nabla^2 S_0(x) = \mu_x I$ and $\nabla^2 S_{k+1}(x) = \big(1-\frac{1}{\sqrt{\kappa_x}}\big)\nabla^2 S_k(x) + \frac{\mu_x}{\sqrt{\kappa_x}} I$. Then, by induction, it can be verified that $\nabla^2 S_k(x) = \mu_x I$ for all $k = 0, ..., K$. This implies that $S_k(x)$ can be written as $S_k(x) = S_k^* + \frac{\mu_x}{2}\|x - v_k\|^2$, where $v_k = \arg\min_{x\in\mathbb{R}^p} S_k(x)$. Next, we show by induction that

1. $\|z_k - x^*\| \le \sqrt{\frac{2}{\mu_x}(\Phi(0)-\Phi(x^*)) + \|x^*\|^2 + \frac{\epsilon}{\mu_x}}$ for all $k = 0, ..., K$. (77)
2. $S_k^* \ge \Phi(z_k)$ for all $k = 0, ..., K$. (78)

Combining the first item in eq. (77) with the updates in eq. (74) also implies the boundedness of the sequence $x_k, k = 0, ..., K$ by noting that

$$\begin{aligned} \|x_k - x^*\| \le& \Big(1+\frac{\sqrt{\kappa_x}-1}{\sqrt{\kappa_x}+1}\Big)\|z_k - x^*\| + \frac{\sqrt{\kappa_x}-1}{\sqrt{\kappa_x}+1}\|z_{k-1}-x^*\| \\ \le& 3\sqrt{\frac{2}{\mu_x}(\Phi(0)-\Phi(x^*)) + \|x^*\|^2 + \frac{\epsilon}{\mu_x}}. \end{aligned} \tag{79}$$

Next, we prove the above two items given in eq. (77) and eq. (78) by induction. First, it can be verified that they hold for $k = 0$ by noting that $\|z_0 - x^*\| = \|x^*\|$ and $S_0^* = \Phi(x_0)$. Then, we suppose that they hold for all $k = 0, ..., k'$ and prove the $k'+1$ case.

Based on Theorem 17, we have, for all $k = 0, ..., k'$,

$$\begin{aligned} \Phi(z_{k+1}) \le& \Phi(x_k) + \langle \nabla\Phi(x_k), z_{k+1}-x_k\rangle + \frac{L_{\Phi_k}}{2}\|z_{k+1}-x_k\|^2 \\ \overset{(i)}{=}& \Phi(x_k) - \frac{1}{L_\Phi}\langle \nabla\Phi(x_k), G_k\rangle + \frac{L_{\Phi_k}}{2L_\Phi^2}\|G_k\|^2, \end{aligned} \tag{80}$$

where (i) follows from the updates in eq. (74). Note that for $k = 0, ..., k'$, it is seen from eq. (79) that the optimality gap $\|x_k - x^*\| \le 3\sqrt{\frac{2}{\mu_x}(\Phi(0)-\Phi(x^*)) + \|x^*\|^2 + \frac{\epsilon}{\mu_x}}$, which, combined with the definition of L_{Φ_k} in eq. (70), yields $L_{\Phi_k} \le L_\Phi$ for all $k = 0, ..., k'$, where L_Φ is given by eq. (75). Then, we obtain from eq. (80) that for all $k = 0, ..., k'$,

$$\begin{aligned} \Phi(z_{k+1}) \le& \Phi(x_k) - \frac{1}{L_\Phi}\langle \nabla\Phi(x_k), G_k\rangle + \frac{1}{2L_\Phi}\|G_k\|^2 \\ =& \Phi(x_k) - \frac{1}{L_\Phi}\|\nabla\Phi(x_k)\|^2 - \frac{1}{L_\Phi}\langle \nabla\Phi(x_k), G_k - \nabla\Phi(x_k)\rangle + \frac{1}{2L_\Phi}\|G_k\|^2 \end{aligned}$$

$$
\begin{aligned}
&=\Phi(x_k) - \frac{1}{L_\Phi}\|\nabla\Phi(x_k)\|^2 + \frac{1}{2L_\Phi}\|\nabla\Phi(x_k)\|^2 + \frac{1}{2L_\Phi}\|G_k - \nabla\Phi(x_k)\|^2 \\
&=\Phi(x_k) - \frac{1}{2L_\Phi}\|\nabla\Phi(x_k)\|^2 + \frac{1}{2L_\Phi}\|G_k - \nabla\Phi(x_k)\|^2, \qquad (81)
\end{aligned}
$$

which, in conjunction with the strong convexity of $\Phi(\cdot)$, yields

$$
\begin{aligned}
\Phi(z_{k+1}) \leq &\Big(1 - \frac{1}{\sqrt{\kappa_x}}\Big)\Phi(z_k) + \Big(1 - \frac{1}{\sqrt{\kappa_x}}\Big)\langle \nabla\Phi(x_k), x_k - z_k\rangle + \frac{1}{\sqrt{\kappa_x}}\Phi(x_k) \\
&- \frac{1}{2L_\Phi}\|\nabla\Phi(x_k)\|^2 + \frac{1}{2L_\Phi}\|G_k - \nabla\Phi(x_k)\|^2 \\
\overset{(i)}{\leq} &\Big(1 - \frac{1}{\sqrt{\kappa_x}}\Big)S_k^* + \Big(1 - \frac{1}{\sqrt{\kappa_x}}\Big)\langle \nabla\Phi(x_k), x_k - z_k\rangle + \frac{1}{\sqrt{\kappa_x}}\Phi(x_k) \\
&- \frac{1}{2L_\Phi}\|\nabla\Phi(x_k)\|^2 + \frac{1}{2L_\Phi}\|G_k - \nabla\Phi(x_k)\|^2, \qquad (82)
\end{aligned}
$$

where (i) follows because $S_k^* \geq \Phi(z_k)$ for $k = 0, ..., k'$. Next, based on the definition of $S_k(x)$ in eq. (76) and taking derivative w.r.t. x on both sides of eq. (76), we have

$$
\begin{aligned}
\nabla S_{k+1}(x) &\overset{(i)}{=} \Big(1 - \frac{1}{\sqrt{\kappa_x}}\Big)\nabla S_k(x) + \frac{1}{\sqrt{\kappa_x}}G_k + \frac{\mu_x}{\sqrt{\kappa_x}}(x - x_k) \\
&= \mu_x\Big(1 - \frac{1}{\sqrt{\kappa_x}}\Big)(x - v_k) + \frac{1}{\sqrt{\kappa_x}}G_k + \frac{\mu_x}{\sqrt{\kappa_x}}(x - x_k), \qquad (83)
\end{aligned}
$$

where (i) follows because $S_k(x) = S_k^* + \frac{\mu_x}{2}\|x - v_k\|^2$. Noting that $\nabla S_{k+1}(v_{k+1}) = 0$, we obtain from eq. (83) that

$$
\mu_x\Big(1 - \frac{1}{\sqrt{\kappa_x}}\Big)(v_{k+1} - v_k) + \frac{1}{\sqrt{\kappa_x}}G_k + \frac{\mu_x}{\sqrt{\kappa_x}}(v_{k+1} - x_k) = 0,
$$

which yields

$$
v_{k+1} = \Big(1 - \frac{1}{\sqrt{\kappa_x}}\Big)v_k + \frac{1}{\sqrt{\kappa_x}}x_k - \frac{1}{\mu_x\sqrt{\kappa_x}}G_k. \qquad (84)
$$

Based on eq. (76) and using $S_k(x) = S_k^* + \frac{\mu_x}{2}\|x - v_k\|^2$, we have

$$
S_{k+1}^* + \frac{\mu_x}{2}\|x_k - v_{k+1}\|^2 = \Big(1 - \frac{1}{\sqrt{\kappa_x}}\Big)\Big(S_k^* + \frac{\mu_x}{2}\|x_k - v_k\|^2\Big) + \frac{1}{\sqrt{\kappa_x}}\Phi(x_k) + \frac{\epsilon}{4\sqrt{\kappa_x}},
$$

which, in conjunction with eq. (84), yields

$$
\begin{aligned}
S_{k+1}^* =&\Big(1 - \frac{1}{\sqrt{\kappa_x}}\Big)S_k^* + \Big(1 - \frac{1}{\sqrt{\kappa_x}}\Big)\frac{\mu_x}{2}\|x_k - v_k\|^2 + \frac{1}{\sqrt{\kappa_x}}\Phi(x_k) + \frac{\epsilon}{4\sqrt{\kappa_x}} \\
&- \Big(1 - \frac{1}{\sqrt{\kappa_x}}\Big)^2\frac{\mu_x}{2}\|x_k - v_k\|^2 - \frac{1}{2\mu_x\kappa_x}\|G_k\|^2 + \Big(1 - \frac{1}{\sqrt{\kappa_x}}\Big)\frac{1}{\sqrt{\kappa_x}}\langle v_k - x_k, G_k\rangle \\
=&\Big(1 - \frac{1}{\sqrt{\kappa_x}}\Big)S_k^* + \Big(1 - \frac{1}{\sqrt{\kappa_x}}\Big)\frac{1}{\sqrt{\kappa_x}}\frac{\mu_x}{2}\|x_k - v_k\|^2 + \frac{1}{\sqrt{\kappa_x}}\Phi(x_k) + \frac{\epsilon}{4\sqrt{\kappa_x}}
\end{aligned}
$$

$$
-\frac{1}{2\mu_x\kappa_x}\|G_k\|^2+\Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)\frac{1}{\sqrt{\kappa_x}}\langle v_k-x_k,G_k\rangle. \tag{85}
$$

Based on the definition of κ_x, we simplify eq. (85) to

$$
\begin{aligned}
S_{k+1}^*\geq&\Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)S_k^*+\frac{1}{\sqrt{\kappa_x}}\Phi(x_k)+\frac{\epsilon}{4\sqrt{\kappa_x}}-\frac{1}{2L_\Phi}\|G_k\|^2\\
&+\Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)\frac{1}{\sqrt{\kappa_x}}\langle v_k-x_k,G_k\rangle.
\end{aligned} \tag{86}
$$

Next, we prove $v_k-x_k=\sqrt{\kappa_x}(x_k-z_k)$ by induction. First note that this equality holds for $k=0$ based on the fact that $v_0-x_0=\sqrt{\kappa_x}(x_0-z_0)=0$. Then, suppose that it holds for iteration k, and for iteration $k+1$, we obtain from eq. (84) that

$$
\begin{aligned}
v_{k+1}-x_{k+1}=&\Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)v_k+\frac{1}{\sqrt{\kappa_x}}x_k-x_{k+1}-\frac{1}{\mu_x\sqrt{\kappa_x}}G_k\\
\overset{(i)}{=}&\Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)\Big(1+\sqrt{\kappa_x}\Big)x_k-\Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)\sqrt{\kappa_x}z_k+\frac{1}{\sqrt{\kappa_x}}x_k-x_{k+1}-\frac{1}{\mu_x\sqrt{\kappa_x}}G_k\\
=&\sqrt{\kappa_x}\Big(x_k-\frac{1}{L_\Phi}G_k\Big)-(\sqrt{\kappa_x}-1)z_k-x_{k+1}\\
\overset{(ii)}{=}&\sqrt{\kappa_x}(x_{k+1}-z_{k+1}),
\end{aligned} \tag{87}
$$

where (i) follows because $v_k-x_k=\sqrt{\kappa_x}(x_k-z_k)$ and (ii) follows from the updating step in eq. (74). Then, by induction, we have that $v_k-x_k=\sqrt{\kappa_x}(x_k-z_k)$ holds for all iterations. Combining this equality with eq. (86), we have

$$
\begin{aligned}
S_{k+1}^*\geq&\Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)S_k^*+\frac{1}{\sqrt{\kappa_x}}\Phi(x_k)+\frac{\epsilon}{4\sqrt{\kappa_x}}-\frac{1}{2L_\Phi}\|G_k\|^2+\Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)\langle x_k-z_k,G_k\rangle\\
=&\Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)S_k^*+\frac{1}{\sqrt{\kappa_x}}\Phi(x_k)+\frac{\epsilon}{4\sqrt{\kappa_x}}-\frac{1}{2L_\Phi}\|\nabla\Phi(x_k)\|^2+\Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)\langle x_k-z_k,\nabla\Phi(x_k)\rangle\\
&+\Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)\langle x_k-z_k,G_k-\nabla\Phi(x_k)\rangle-\frac{1}{2L_\Phi}\|G_k-\nabla\Phi(x_k)\|^2-\frac{1}{L_\Phi}\langle G_k-\nabla\Phi(x_k),\nabla\Phi(x_k)\rangle\\
\overset{(i)}{\geq}&\Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)S_k^*+\frac{1}{\sqrt{\kappa_x}}\Phi(x_k)-\frac{1}{2L_\Phi}\|\nabla\Phi(x_k)\|^2+\Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)\langle x_k-z_k,\nabla\Phi(x_k)\rangle+\frac{\epsilon}{4\sqrt{\kappa_x}}\\
&-\Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)\|x_k-z_k\|\|G_k-\nabla\Phi(x_k)\|-\frac{1}{2L_\Phi}\|G_k-\nabla\Phi(x_k)\|^2\\
&-\|G_k-\nabla\Phi(x_k)\|\|x_k-x^*\|
\end{aligned} \tag{88}
$$

where (i) follows from Theorem 17 with $L_{\Phi_k}\leq L_\Phi$ for $k=0,...,k'$. Based on $\|z_k-x^*\|\leq\sqrt{\frac{2}{\mu_x}(\Phi(0)-\Phi(x^*))+\|x^*\|^2+\frac{\epsilon}{\mu_x}}$ and $\|x_k-x^*\|<3\sqrt{\frac{2}{\mu_x}(\Phi(0)-\Phi(x^*))+\|x^*\|^2+\frac{\epsilon}{\mu_x}}$ for $k=0,...,k'$, and using $\|x_k-z_k\|\leq\|z_k-x^*\|+\|x_k-x^*\|$, we obtain from eq. (88) that

$$
\begin{aligned}
S_{k+1}^*\geq&\Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)S_k^*+\frac{1}{\sqrt{\kappa_x}}\Phi(x_k)-\frac{1}{2L_\Phi}\|\nabla\Phi(x_k)\|^2+\Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)\langle x_k-z_k,\nabla\Phi(x_k)\rangle\\
&+\frac{\epsilon}{4\sqrt{\kappa_x}}-\Big(7-\frac{4}{\sqrt{\kappa_x}}\Big)\sqrt{\frac{2}{\mu_x}(\Phi(0)-\Phi(x^*))+\|x^*\|^2+\frac{\epsilon}{\mu_x}}\|G_k-\nabla\Phi(x_k)\|
\end{aligned}
$$

$$- \frac{1}{2L_\Phi}\|G_k - \nabla\Phi(x_k)\|^2. \tag{89}$$

Next, we upper-bound the hypergradient estimation error $\|G_k-\nabla\Phi(x_k)\|$ in eq. (89). Based on Theorem 16, we have

$$\begin{aligned}\|G_k - \nabla\Phi(x_k)\| \leq& \sqrt{\frac{\widetilde{L}_y+\mu_y}{\mu_y}}\Big(L_y + \frac{2\widetilde{L}_{xy}L_y}{\mu_y} + \Big(\frac{\rho_{xy}}{\mu_y}+\frac{\widetilde{L}_{xy}\rho_{yy}}{\mu_y^2}\Big)\mathcal{N}_k\Big)\mathcal{M}_k\exp\Big(-\frac{N}{2\sqrt{\kappa_y}}\Big)\\ &+\frac{\widetilde{L}_{xy}}{\mu_y}\Big(\frac{\sqrt{\kappa_y}-1}{\sqrt{\kappa_y}+1}\Big)^M\mathcal{N}_k,\end{aligned}$$

which, combined with $\|x_k - x^*\| \leq 3\sqrt{\frac{2}{\mu_x}(\Phi(0)-\Phi(x^*))+\|x^*\|^2+\frac{\epsilon}{\mu_x}}$ for $k=0,...,k'$ and the definitions of $\mathcal{M}_k,\mathcal{N}_k$ in eq. (62), yields

$$\begin{aligned}\|G_k - \nabla\Phi(x_k)\| \leq& \sqrt{\frac{\widetilde{L}_y+\mu_y}{\mu_y}}\Big(L_y + \frac{2\widetilde{L}_{xy}L_y}{\mu_y} + \Big(\frac{\rho_{xy}}{\mu_y}+\frac{\widetilde{L}_{xy}\rho_{yy}}{\mu_y^2}\Big)\mathcal{N}_*\Big)\mathcal{M}_*\exp\Big(-\frac{N}{2\sqrt{\kappa_y}}\Big)\\ &+\frac{\widetilde{L}_{xy}}{\mu_y}\Big(\frac{\sqrt{\kappa_y}-1}{\sqrt{\kappa_y}+1}\Big)^M\mathcal{N}_*,\end{aligned}$$

where the constants $\mathcal{M}_*$ and $\mathcal{N}_*$ are defined in eq. (62). We choose sufficiently large

$$\begin{aligned}N &= \Theta\Big(\sqrt{\kappa_y}\log\big(\kappa_x,\kappa_y,\|x^*\|,\|y^*(x^*)\|,\Phi(0)-\Phi(x^*),\epsilon\big)\Big),\\ M &= \Theta\Big(\sqrt{\kappa_y}\log\big(\kappa_x,\kappa_y,\|x^*\|,\|y^*(x^*)\|,\Phi(0)-\Phi(x^*),\epsilon\big)\Big),\end{aligned} \tag{90}$$

such that

$$\begin{aligned}&\|G_k-\nabla\Phi(x_k)\| \leq \frac{\sqrt{\epsilon L_\Phi}}{2\sqrt{2}\kappa_x^{1/4}}\\ &\big(7-\frac{4}{\sqrt{\kappa_x}}\big)\sqrt{\frac{2}{\mu_x}(\Phi(0)-\Phi(x^*))+\|x^*\|^2+\frac{\epsilon}{\mu_x}}\|G_k-\nabla\Phi(x_k)\| \leq \frac{\epsilon}{8\sqrt{\kappa_x}}.\end{aligned}$$

Substituting these two inequalities into eq. (89) yields, for any $k=0,...,k'$,

$$\begin{aligned}S_{k+1}^* \geq& \Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)S_k^* + \frac{1}{\sqrt{\kappa_x}}\Phi(x_k) - \frac{1}{2L_\Phi}\|\nabla\Phi(x_k)\|^2\\ &+\Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)\langle x_k - z_k, \nabla\Phi(x_k)\rangle + \frac{\epsilon}{16\sqrt{\kappa_x}}\\ \overset{(i)}{\geq}& \Phi(z_{k+1}),\end{aligned} \tag{91}$$

where (i) follows from $\|G_k-\nabla\Phi(x_k)\| \leq \frac{\sqrt{\epsilon L_\Phi}}{2\sqrt{2}\kappa_x^{1/4}}$ in eq. (82), which, by induction, finishes the proof of the second item eq. (78). To prove the first item eq. (77), letting $x = x^*$ in eq. (76) yields, for $x=0,...,k'$,

$$S_{k+1}(x^*) = \Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)S_k(x^*) + \frac{1}{\sqrt{\kappa_x}}\Big(\Phi(x_k) + \langle\nabla\Phi(x_k), x^*-x_k\rangle + \frac{\mu_x}{2}\|x^*-x_k\|^2 + \frac{\epsilon}{4}\Big)$$

$$
\begin{aligned}
&+\frac{1}{\sqrt{\kappa_x}}\langle G_k-\nabla\Phi(x_k),x^*-x_k\rangle\\
\leq&\Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)S_k(x^*)+\frac{1}{\sqrt{\kappa_x}}\Phi(x^*)+\frac{\epsilon}{4\sqrt{\kappa_x}}+\frac{1}{\sqrt{\kappa_x}}\|x_k-x^*\|\|G_k-\nabla\Phi(x_k)\|\\
\overset{(i)}{\leq}&\Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)S_k(x^*)+\frac{1}{\sqrt{\kappa_x}}\Phi(x^*)+\frac{\epsilon}{2\sqrt{\kappa_x}},
\end{aligned}\tag{92}
$$

where (i) follows because $\|x_k-x^*\|\|G_k-\nabla\Phi(x_k)\|\leq\frac{\epsilon}{8\sqrt{\kappa_x}}/(7-\frac{4}{\sqrt{\kappa_x}})<\frac{\epsilon}{24\sqrt{\kappa_x}}<\frac{\epsilon}{4}$. Subtracting both sides of eq. (92) by $\Phi(x^*)$ yields, for all $k=0,...,k'$,

$$
S_{k+1}(x^*)-\Phi(x^*)\leq\Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)(S_k(x^*)-\Phi(x^*))+\frac{\epsilon}{2\sqrt{\kappa_x}}.\tag{93}
$$

Telescoping eq. (93) over k from 0 to k' and using $S_0(x^*)=\Phi(0)+\frac{\mu_x}{2}\|x^*\|^2$, we have

$$
\begin{aligned}
S_{k'+1}(x^*)-\Phi(x^*)\leq&\Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)^{k'+1}(\Phi(0)-\Phi(x^*)+\frac{\mu_x}{2}\|x^*\|^2)+\frac{\epsilon}{2}\\
\leq&\,\Phi(0)-\Phi(x^*)+\frac{\mu_x}{2}\|x^*\|^2+\frac{\epsilon}{2},
\end{aligned}
$$

which, in conjunction with $S_{k'+1}(x^*)\geq S^*_{k'+1}\geq\Phi(z_{k'+1})$ and $\Phi(z_{k'+1})-\Phi(x^*)\geq\frac{\mu_x}{2}\|z_{k'+1}-x^*\|^2$, yields

$$
\|z^{k'+1}-x^*\|\leq\sqrt{\frac{2}{\mu_x}(\Phi(0)-\Phi(x^*))+\|x^*\|^2+\frac{\epsilon}{\mu_x}}.
$$

Then, by induction, we finish the proof of the first item eq. (77). Therefore, based on eq. (77) and eq. (78) and using an approach similar to eq. (93), we have

$$
\Phi(z_K)-\Phi(x^*)\leq S_K(x^*)-\Phi(x^*)\leq\Big(1-\frac{1}{\sqrt{\kappa_x}}\Big)^K(\Phi(0)-\Phi(x^*)+\frac{\mu_x}{2}\|x^*\|^2)+\frac{\epsilon}{2}.\tag{94}
$$

In order to achieve $\Phi(z_K)-\Phi(x^*)\leq S_K(x^*)-\Phi(x^*)\leq\epsilon$, it requires at most

$$
\begin{aligned}
K=&\,\mathcal{O}\Big(\sqrt{\frac{L_\Phi}{\mu_x}}\log\Big(\frac{\Phi(0)-\Phi(x^*)+\frac{\mu_x}{2}\|x^*\|^2}{\epsilon}\Big)\Big)\\
=&\,\mathcal{O}\Big(\sqrt{\frac{L_x}{\mu_x}}+\sqrt{\frac{L_y\widetilde{L}_{xy}^2}{\mu_x\mu_y^2}}+\sqrt{\frac{\widetilde{L}_{xy}^2\rho_{xy}}{\mu_y^3}}\sqrt{\frac{\|\nabla_y f(x^*,y^*(x^*))\|}{\mu_x}}\\
&+\sqrt{\frac{\widetilde{L}_{xy}^3\rho_{yy}L_y}{\mu_x\mu_y^4}}\Big(\frac{2}{\mu_x}(\Phi(0)-\Phi(x^*))+\|x^*\|^2+\frac{\epsilon}{\mu_x}\Big)^{\frac{1}{4}}\Big),
\end{aligned}\tag{95}
$$

Following from the choice of $M=N=\widetilde{\Theta}(\sqrt{\kappa_y})$, the complexity of Algorithm 1 is given by

$$
\mathcal{C}_{\text{fun}}(\mathcal{A},\epsilon)=\mathcal{O}(n_J+n_H+n_G)=\mathcal{O}(K+KM+KN)
$$

$$
\begin{aligned}
=\mathcal{O}\Big(&\sqrt{\frac{L_x\widetilde{L}_y}{\mu_x\mu_y}}+\sqrt{\frac{L_y\widetilde{L}_{xy}^2\widetilde{L}_y}{\mu_x\mu_y^3}}+\sqrt{\frac{\widetilde{L}_{xy}^2\rho_{xy}\widetilde{L}_y}{\mu_y^4}}\sqrt{\frac{\|\nabla_y f(x^*,y^*(x^*))\|}{\mu_x}}\\
&+\sqrt{\frac{\widetilde{L}_{xy}^3\rho_{yy}L_y\widetilde{L}_y}{\mu_x\mu_y^5}}\Big(\frac{2}{\mu_x}(\Phi(0)-\Phi(x^*))+\|x^*\|^2+\frac{\epsilon}{\mu_x}\Big)^{\frac{1}{4}}\Big),
\end{aligned}
$$

which finishes the proof. ∎

Appendix G. Proof of Theorem 10

The proof follows a procedure similar to that for Theorem 9 except that the smoothness parameter of $\Phi(\cdot)$ at iterate x_k and the hypergradient estimation error $\|G_k-\nabla\Phi(x_k)\|$ are different. In specific, for the quadratic inner problem, we have that $\nabla_y^2 g(x,y)\equiv H, \nabla_x\nabla_y g(x,y)\equiv J, \forall x\in\mathbb{R}^p, y\in\mathbb{R}^q$. Then, based on the form of $\nabla\Phi(x)$ in eq. (69), we have

$$
\begin{aligned}
&\|\nabla\Phi(x_1)-\nabla\Phi(x_2)\|\\
&\quad\le\|\nabla_x f(x_1,y^*(x_1))-\nabla_x f(x_2,y^*(x_2))\|\\
&\quad\quad+\|JH^{-1}\nabla_y f(x_1,y^*(x_1))-JH^{-1}\nabla_y f(x_2,y^*(x_2))\|\\
&\quad\le L_x\|x_1-x_2\|+L_{xy}\|y^*(x_1)-y^*(x_2)\|+\frac{\widetilde{L}_{xy}}{\mu_y}(L_{xy}\|x_1-x_2\|+L_y\|y^*(x_1)-y^*(x_2)\|)
\end{aligned}
$$

which, in conjunction with $\|y^*(x_1)-y^*(x_2)\|\le\frac{\widetilde{L}_{xy}}{\mu_y}\|x_1-x_2\|$, yields

$$
\|\nabla\Phi(x_1)-\nabla\Phi(x_2)\|\le\underbrace{\Big(L_x+\frac{2\widetilde{L}_{xy}L_{xy}}{\mu_y}+\frac{L_y\widetilde{L}_{xy}^2}{\mu_y^2}\Big)}_{L_\Phi}\|x_1-x_2\|. \tag{96}
$$

Note that eq. (96) shows that the objective function $\Phi(\cdot)$ is globally smooth, i.e., the smoothness parameter is bounded at all $x\in\mathbb{R}^p$. This is different from the proof in Theorem 9, where the smoothness parameter is unbounded over $x\in\mathbb{R}^p$, but can be bounded at all iterates $x_k, k=0,...,K$ along the optimization path of the algorithm. Therefore, the proof for such a quadratic special case is simpler.

We next upper-bound the hypergradient estimation error $\|G_k-\nabla\Phi(x_k)\|$. Using an approach similar to eq. (66), we have

$$
\begin{aligned}
&\|G_k-\nabla\Phi(x_k)\|\\
&\quad\le L_y\|y^*(x_k)-y_k^N\|+\widetilde{L}_{xy}\|v_k^M-H^{-1}\nabla_y f(x_k,y_k^N)\|\\
&\quad\quad+\widetilde{L}_{xy}\big\|H^{-1}\nabla_y f(x_k,y_k^N)-H^{-1}\nabla_y f(x_k,y^*(x_k))\big\|\\
&\quad\le\Big(L_y+\frac{\widetilde{L}_{xy}L_y}{\mu_y}\Big)\|y_k^N-y^*(x_k)\|+\widetilde{L}_{xy}\|v_k^M-H^{-1}\nabla_y f(x_k,y_k^N)\|
\end{aligned}
$$

$$
\begin{aligned}
&\leq\Big(L_y+\frac{\widetilde{L}_{xy}L_y}{\mu_y}\Big)\|y_k^N-y^*(x_k)\|+\frac{\widetilde{L}_{xy}}{\mu_y}\left(\frac{\sqrt{\kappa_y}-1}{\sqrt{\kappa_y}+1}\right)^M\|\nabla_y f(x_k,y^*(x_k))\|\\
&\leq\sqrt{\frac{\widetilde{L}_y+\mu_y}{\mu_y}}\Big(L_y+\frac{\widetilde{L}_{xy}L_y}{\mu_y}\Big)\mathcal{M}_*\exp\Big(-\frac{N}{2\sqrt{\kappa_y}}\Big)+\frac{\widetilde{L}_{xy}}{\mu_y}\Big(\frac{\sqrt{\kappa_y}-1}{\sqrt{\kappa_y}+1}\Big)^M\mathcal{N}_*, \qquad (97)
\end{aligned}
$$

where $\mathcal{M}_*$ and $\mathcal{N}_*$ are given by eq. (62). Based on eq. (96), eq. (97), we choose $N=\widetilde{\Theta}(\sqrt{\kappa_y}), M=\widetilde{\Theta}(\sqrt{\kappa_y})$. Then, using an approach similar to eq. (94) with $\rho_{xy}=\rho_{yy}=0$, we have

$$
\Phi(z_K)-\Phi(x^*)\leq\Big(1-\sqrt{\frac{\mu_x}{L_\Phi}}\Big)^K(\Phi(0)-\Phi(x^*)+\frac{\mu_x}{2}\|x^*\|^2)+\frac{\epsilon}{2}, \qquad (98)
$$

where L_Φ is given in eq. (96). Then, in order to achieve $\Phi(z_K)-\Phi(x^*)\leq\epsilon$, it requires at most

$$
\mathcal{C}_{\text{fun}}(\mathcal{A},\epsilon)=\mathcal{O}(n_J+n_H+n_G)=\mathcal{O}(K+KM+KN)=\widetilde{\mathcal{O}}\Big(\sqrt{\frac{L_y\widetilde{L}_{xy}^2\widetilde{L}_y}{\mu_x\mu_y^3}}\Big),
$$

which finishes the proof.

Appendix H. Proof of Theorem 11

Recall that $\widetilde{\Phi}(\cdot)=\widetilde{f}(x,y^*(x))$ with $\widetilde{f}(x,y)=f(x,y)+\frac{\epsilon}{2R}\|x\|^2$. Then, we have $\widetilde{\Phi}(x)=\Phi(x)+\frac{\epsilon}{2R}\|x\|^2$ is strongly-convex with parameter $\mu_x=\frac{\epsilon}{R}$. Note that the smoothness parameters of $\widetilde{f}(x,y)$ are the same as those of $f(x,y)$ except that L_x in eq. (2) becomes $L_x+\frac{\epsilon}{R}$ for $\widetilde{f}(x,y)$. Let $x^*\in\arg\min_{x\in\mathbb{R}^p}\Phi(x)$ be one minimizer of the original objective function $\Phi(\cdot)$ and let $\widetilde{x}^*=\arg\min_{x\in\mathbb{R}^p}\widetilde{\Phi}(x)$ be the minimizer of the regularized objective function $\widetilde{\Phi}(\cdot)$. We next characterize some useful inequalities between x^* and $\widetilde{x}^*$. Based on the definition of x^* and $\widetilde{x}^*$, we have $\nabla\widetilde{\Phi}(\widetilde{x}^*)=0$ and $\nabla\widetilde{\Phi}(x^*)=\nabla\Phi(x^*)+\frac{\epsilon}{R}x^*=\frac{\epsilon}{R}x^*$, which, combined with the strong convexity of $\widetilde{\Phi}(\cdot)$, implies that $\frac{\epsilon}{R}\|x^*-\widetilde{x}^*\|\leq\|\nabla\widetilde{\Phi}(\widetilde{x}^*)-\nabla\widetilde{\Phi}(x^*)\|=\frac{\epsilon}{R}\|x^*\|$ and hence $\|\widetilde{x}^*\|\leq 2\|x^*\|$. Similarly, the following (in)equalities hold:

$$
\begin{aligned}
\|y^*(\widetilde{x}^*)\|&\leq\|y^*(x^*)\|+\frac{3\widetilde{L}_{xy}}{\mu_y}\|x^*\|,\\
\|\nabla_y\widetilde{f}(\widetilde{x}^*,y^*(\widetilde{x}^*))\|&\leq\|\nabla_y f(\widetilde{x}^*,y^*(\widetilde{x}^*))\|+\frac{\epsilon}{R}\|\widetilde{x}^*\|\\
&\leq\|\nabla_y f(x^*,y^*(x^*))\|+\Big(3L_{xy}+\frac{3L_y\widetilde{L}_{xy}}{\mu_y}+\frac{2\epsilon}{R}\Big)\|x^*\|\\
\widetilde{\Phi}(0)-\widetilde{\Phi}(\widetilde{x}^*)&=\Phi(0)-\Phi(\widetilde{x}^*)-\frac{\epsilon}{2R}\|\widetilde{x}^*\|^2\overset{(i)}{\leq}\Phi(0)-\Phi(x^*), \qquad (99)
\end{aligned}
$$

where (i) follows from the definition of $x^*\in\arg\min_x\Phi(x)$.

Let $L_{\widetilde{\Phi}}$ be one smoothness parameter of the function $\widetilde{\Phi}(\cdot)$, which takes the same form as L_Φ in eq. (75) except that L_x,f,x^* , μ_x and Φ become $L_x+\frac{\epsilon}{R},\widetilde{f},\widetilde{x}^*$, $\frac{\epsilon}{R}$ and $\widetilde{\Phi}$ in eq. (75), respectively. Similarly to eq. (90), we choose

$$
N=\Theta(\sqrt{\kappa_y}\log(\text{poly}(\epsilon,R,\kappa_y,\|\widetilde{x}^*\|,\|y^*(\widetilde{x}^*)\|,\|\nabla_y\widetilde{f}(\widetilde{x}^*,y^*(\widetilde{x}^*))\|,\widetilde{\Phi}(0)-\widetilde{\Phi}(\widetilde{x}^*)))),
$$

$$M = \Theta(\sqrt{\kappa_y}\log(\text{poly}(\epsilon, R, \kappa_y, \|\widetilde{x}^*\|, \|y^*(\widetilde{x}^*)\|, \|\nabla_y \widetilde{f}(\widetilde{x}^*, y^*(\widetilde{x}^*))\|, \widetilde{\Phi}(0) - \widetilde{\Phi}(\widetilde{x}^*)))). \quad (100)$$

We first prove the case when the convergence is measured in term of the suboptimality gap. Note that in this case we choose $R = B^2$. Using an approach similar to eq. (94) in the proof of Theorem 9 with ϵ and μ_x being replaced by $\epsilon/2$ and $\frac{\epsilon}{B^2}$, respectively, we have

$$\widetilde{\Phi}(z_K) - \widetilde{\Phi}(\widetilde{x}^*) \leq \Big(1 - \sqrt{\frac{\epsilon}{B^2 L_{\widetilde{\Phi}}}}\Big)^K (\widetilde{\Phi}(0) - \widetilde{\Phi}(\widetilde{x}^*) + \frac{\epsilon}{2B^2}\|\widetilde{x}^*\|^2) + \frac{\epsilon}{4},$$

which, in conjunction with $\widetilde{\Phi}(z_K) \geq \Phi(z_K)$ and $\widetilde{\Phi}(\widetilde{x}^*) \leq \widetilde{\Phi}(x^*) = \Phi(x^*) + \frac{\epsilon}{2B^2}\|x^*\|^2$, yields

$$\Phi(z_K) - \Phi(x^*) \leq \Big(1 - \sqrt{\frac{\epsilon}{B^2 L_{\widetilde{\Phi}}}}\Big)^K (\widetilde{\Phi}(0) - \widetilde{\Phi}(\widetilde{x}^*) + \frac{\epsilon}{2B^2}\|\widetilde{x}^*\|^2) + \frac{\epsilon}{4} + \frac{\epsilon}{2B^2}\|x^*\|^2. \quad (101)$$

Recall $\|x^*\| = B$. Similarly to eq. (95), we choose

$$\begin{aligned} K =& \Theta\Big(\sqrt{\frac{B^2 L_{\widetilde{\Phi}}}{\epsilon}}\log\Big(\frac{\widetilde{\Phi}(0) - \widetilde{\Phi}(\widetilde{x}^*) + \frac{\epsilon}{2B^2}\|\widetilde{x}^*\|^2}{\epsilon}\Big)\Big) \\ =& \widetilde{\Theta}\Big(\sqrt{\frac{B^2 L_y \widetilde{L}_{xy}^2}{\epsilon\mu_y^2}} + \sqrt{\frac{B^2 \widetilde{L}_{xy}^2 \rho_{xy}}{\epsilon\mu_y^3}}\sqrt{\|\nabla_y \widetilde{f}(\widetilde{x}^*, y^*(\widetilde{x}^*))\|} \\ & + \sqrt{\frac{B^2 \widetilde{L}_{xy}^3 \rho_{yy} L_y}{\epsilon\mu_y^4}}\Big(\frac{2B^2}{\epsilon}(\widetilde{\Phi}(0) - \widetilde{\Phi}(\widetilde{x}^*)) + \|\widetilde{x}^*\|^2 + B^2\Big)^{\frac{1}{4}}\Big), \end{aligned} \quad (102)$$

Then, we obtain from eq. (101) that $\Phi(z_K) - \Phi(x^*) \leq \epsilon$, and the complexity $\mathcal{C}_{\text{fun}}(\mathcal{A}, \epsilon)$ after substituting eq. (99) into eq. (100) and eq. (102) is given by

$$\begin{aligned} &\mathcal{C}_{\text{fun}}(\mathcal{A}, \epsilon) = \mathcal{O}(n_J + n_H + n_G) = \mathcal{O}(K + KM + KN) \\ &= \widetilde{\mathcal{O}}\Big(\sqrt{\frac{B^2 L_y \widetilde{L}_{xy}^2 \widetilde{L}_y}{\epsilon\mu_y^3}} + \sqrt{\frac{B^2 \widetilde{L}_{xy}^2 \rho_{xy} \widetilde{L}_y}{\epsilon\mu_y^4}}\sqrt{\|\nabla_y \widetilde{f}(\widetilde{x}^*, y^*(\widetilde{x}^*))\|} \\ &\quad + \sqrt{\frac{B^2 \widetilde{L}_{xy}^3 \rho_{yy} L_y \widetilde{L}_y}{\epsilon\mu_y^5}}\Big(\frac{2B^2}{\epsilon}(\widetilde{\Phi}(0) - \widetilde{\Phi}(\widetilde{x}^*)) + \|\widetilde{x}^*\|^2 + B^2\Big)^{\frac{1}{4}}\Big) \end{aligned} \quad (103)$$

Next, we characterize the convergence rate and complexity under the gradient norm metric. Note that in this case we choose $R = B$. Using eq. (9.14) in Boyd et al. (2004), we have $\|\nabla\widetilde{\Phi}(z_k)\|^2 \leq 2L_{\widetilde{\Phi}}(\widetilde{\Phi}(z_K) - \widetilde{\Phi}(\widetilde{x}^*))$, which, combined with $\|\nabla\widetilde{\Phi}(z_k)\|^2 \geq \frac{1}{2}\|\nabla\Phi(z_k)\|^2 - \frac{\epsilon^2}{B^2}\|z_k\|^2 \geq \frac{1}{2}\|\nabla\Phi(z_k)\|^2 - \frac{\epsilon^2}{B^2}(2\|z_k - \widetilde{x}^*\|^2 + 2\|\widetilde{x}^*\|^2)$ yields

$$\begin{aligned} \|\nabla\Phi(z_k)\|^2 \leq& 4L_{\widetilde{\Phi}}(\widetilde{\Phi}(z_K) - \widetilde{\Phi}(\widetilde{x}^*)) + \frac{4\epsilon^2}{B^2}\|z_k - \widetilde{x}^*\|^2 + \frac{4\epsilon^2}{B^2}\|\widetilde{x}^*\|^2 \\ \overset{(i)}{\leq}& 4L_{\widetilde{\Phi}}(\widetilde{\Phi}(z_K) - \widetilde{\Phi}(\widetilde{x}^*)) + \frac{8\epsilon}{B}(\widetilde{\Phi}(z_K) - \widetilde{\Phi}(\widetilde{x}^*)) + \frac{16\epsilon^2}{B^2}\|x^*\|^2 \\ =& \Big(4L_{\widetilde{\Phi}} + \frac{8\epsilon}{B}\Big)(\widetilde{\Phi}(z_K) - \widetilde{\Phi}(\widetilde{x}^*)) + \frac{16\epsilon^2}{B^2}\|x^*\|^2, \end{aligned} \quad (104)$$

where (i) follows from the strong convexity of $\widetilde{\Phi}(\cdot)$ and $\|\widetilde{x}^*\| \leq 2\|x^*\|$, and $L_{\widetilde{\Phi}}$ takes the same form as L_Φ in eq. (75) except that L_x, f, x^* and Φ become $L_x + \frac{\epsilon}{B}, \widetilde{f}, \widetilde{x}^*$ and $\widetilde{\Phi}$ in eq. (75), respectively. Then, using an approach similar to eq. (94) in the proof of Theorem 9 with ϵ and μ_x being replaced by $\epsilon^2/(4L_{\widetilde{\Phi}}+\frac{8\epsilon}{B})$ and $\frac{\epsilon}{B}$, respectively, we have

$$\widetilde{\Phi}(z_K)-\widetilde{\Phi}(\widetilde{x}^*)\leq\Big(1-\sqrt{\frac{\epsilon}{BL_{\widetilde{\Phi}}}}\Big)^K(\widetilde{\Phi}(0)-\widetilde{\Phi}(\widetilde{x}^*)+\frac{\epsilon}{2B}\|\widetilde{x}^*\|^2)+\frac{\epsilon^2}{2(4L_{\widetilde{\Phi}}+\frac{8\epsilon}{B})},$$

which, in conjunction with eq. (90) and eq. (104), yields

$$\|\nabla\Phi(z_k)\|^2\leq\Big(1-\sqrt{\frac{\epsilon}{BL_{\widetilde{\Phi}}}}\Big)^K\Big(\widetilde{\Phi}(0)-\widetilde{\Phi}(\widetilde{x}^*)+\frac{\epsilon}{2B}\|\widetilde{x}^*\|^2\Big)\Big(4L_{\widetilde{\Phi}}+\frac{8\epsilon}{B}\Big)+\frac{\epsilon^2}{2}+\frac{16\epsilon^2}{B^2}\|x^*\|^2.$$

Note that $\|x^*\|=B$. Then, to achieve $\|\nabla\Phi(z_k)\|\leq 5\epsilon$, it suffices to choose M,N similarly to eq. (100), and choose

$$K=\Theta\Big(\sqrt{\frac{BL_{\widetilde{\Phi}}}{\epsilon}}\log\Big(\frac{(\widetilde{\Phi}(0)-\widetilde{\Phi}(\widetilde{x}^*)+\frac{\epsilon}{2B}\|\widetilde{x}^*\|^2)(4L_{\widetilde{\Phi}}+\frac{8\epsilon}{B})}{\epsilon}\Big)\Big).$$

This in conjunction with eq. (99) yields

$$\begin{aligned}\mathcal{C}_{\text{grad}}(\mathcal{A},\epsilon)=&\mathcal{O}(n_J+n_H+n_G)=\mathcal{O}(K+KM+KN)\\=&\widetilde{\mathcal{O}}\Big(\sqrt{\frac{BL_y\widetilde{L}_{xy}^2\widetilde{L}_y}{\epsilon\mu_y^3}}+\sqrt{\frac{B\widetilde{L}_{xy}^2\rho_{xy}\widetilde{L}_y}{\epsilon\mu_y^4}}\sqrt{\|\nabla_y\widetilde{f}(\widetilde{x}^*,y^*(\widetilde{x}^*))\|}\\&+\sqrt{\frac{B\widetilde{L}_{xy}^3\rho_{yy}L_y\widetilde{L}_y}{\epsilon\mu_y^5}}\Big(\frac{2B}{\epsilon}(\widetilde{\Phi}(0)-\widetilde{\Phi}(\widetilde{x}^*))+\|\widetilde{x}^*\|^2+B\Big)^{\frac{1}{4}}\Big)\end{aligned}$$

which finishes the proof.

Appendix I. Proof of Theorem 12

Note that for the quadratic inner problem, the Jacobians $\nabla_x\nabla_y g(x,y)$ and Hessians $\nabla_y^2 g(x,y)$ are **constant** matrices, which imply that the parameters $\rho_{xx}=\rho_{xy}=0$ in Assumption 2. Then, letting $\rho_{xx}=\rho_{xy}=0$ in the results of Theorem 11 finishes the proof.

Appendix J. Proof of Theorem 13

Based on the update in line 9 of Algorithm 2, we have, for any $x\in\mathbb{R}^p$

$$\langle\beta_kG_k,x_{k+1}-x\rangle=\tau_k\underbrace{\langle x-x_{k+1},x_{k+1}-\widetilde{x}_k\rangle}_{P}+(1-\tau_k)\underbrace{\langle x-x_{k+1},x_{k+1}-x_k\rangle}_{Q}. \tag{105}$$

Note that P in the above eq. (105) satisfies

$$\begin{aligned}P=&\langle\widetilde{x}_k-x_{k+1},x-\widetilde{x}_k\rangle+\|x-\widetilde{x}_k\|^2-\|x-x_{k+1}\|^2\\=&-P+\|x-\widetilde{x}_k\|^2-\|\widetilde{x}_k-x_{k+1}\|^2-\|x-x_{k+1}\|^2,\end{aligned}$$

which yields $P = \frac{1}{2}(\|x-\widetilde{x}_k\|^2 - \|\widetilde{x}_k - x_{k+1}\|^2 - \|x - x_{k+1}\|^2)$. Taking an approach similar to the derivation of P, we can obtain $Q = \frac{1}{2}(\|x - x_k\|^2 - \|x - x_{k+1}\|^2 - \|x_k - x_{k+1}\|^2)$. Then, substituting the forms of P, Q to eq. (105) and using the choices of τ_k and β_k, we have

$$
\begin{aligned}
\Big\langle G_k, \frac{\sqrt{\alpha\mu_x}}{2}(x_{k+1} - x)\Big\rangle = & \frac{\sqrt{\alpha\mu_x}\mu_x}{8}(\|x-\widetilde{x}_k\|^2 - \|\widetilde{x}_k - x_{k+1}\|^2 - \|x - x_{k+1}\|^2) \\
& + \frac{2\mu_x - \sqrt{\alpha\mu_x}\mu_x}{8}(\|x - x_k\|^2 - \|x - x_{k+1}\|^2 - \|x_k - x_{k+1}\|^2). \quad (106)
\end{aligned}
$$

Based on the update $z_{k+1} = \widetilde{x}_k - \alpha_k G_k$ and the choice of $\alpha_k = \alpha$, we have, for any $x' \in \mathbb{R}^p$,

$$
\begin{aligned}
\langle z_{k+1} - x', G_k\rangle = & \frac{1}{\alpha}\langle x' - z_{k+1}, z_{k+1} - \widetilde{x}_k\rangle \\
= & \frac{1}{2\alpha}(\|x' - \widetilde{x}_k\| - \|x' - z_{k+1}\|^2 - \|z_{k+1} - \widetilde{x}_k\|^2). \quad (107)
\end{aligned}
$$

Let $x' = (1 - \frac{\sqrt{\alpha\mu_x}}{2})z_k + \frac{\sqrt{\alpha\mu_x}}{2}$ and recall $\widetilde{x}_k = \eta_k x_k + (1-\eta_k)z_k$. Then, we have

$$
\begin{aligned}
\|x' - \widetilde{x}_k\|^2 = & \Big\|\frac{\sqrt{\alpha\mu_x}}{2}(x_{k+1} - z_k) + \frac{\sqrt{\alpha\mu_x}}{\sqrt{\alpha\mu_x}+2}(z_k - x_k)\Big\|^2 \\
= & \Big\|\frac{\sqrt{\alpha\mu_x}}{2}(x_{k+1} - x_k) + \frac{\alpha\mu_x}{2(\sqrt{\alpha\mu_x}+2)}(z_k - x_k)\Big\|^2 \\
\overset{(i)}{=} & \frac{\alpha\mu_x}{4}\Big\|(1 - \frac{\sqrt{\alpha\mu_x}}{2})(x_{k+1} - x_k) + \frac{\sqrt{\alpha\mu_x}}{2}(x_{k+1} - \widetilde{x}_k)\Big\|^2 \\
\leq & \frac{\alpha\mu_x}{4}\big(1 - \frac{\sqrt{\alpha\mu_x}}{2}\big)\|x_{k+1} - x_k\|^2 + \frac{\alpha\mu_x\sqrt{\alpha\mu_x}}{8}\|x_{k+1} - \widetilde{x}_k\|^2, \quad (108)
\end{aligned}
$$

where (i) follows because $\widetilde{x}_k - x_k = \frac{2}{2+\sqrt{\alpha\mu_x}}(z_k - x_k)$. Then, substituting eq. (108) into eq. (107), adding eq. (106) and eq. (107), and cancelling out several negative terms, we have

$$
\begin{aligned}
& \Big\langle G_k, \frac{\sqrt{\alpha\mu_x}}{2}(z_{k+1} - x) + (1 - \frac{\sqrt{\alpha\mu_x}}{2})(z_{k+1} - z_k)\Big\rangle \\
\leq & \frac{\sqrt{\alpha\mu_x}\mu_x}{8}\|x - \widetilde{x}_k\|^2 - \frac{1}{2\alpha}\|z_{k+1} - \widetilde{x}_k\|^2 - \frac{\mu_x\sqrt{\alpha\mu_x}}{16}\|x_{k+1} - \widetilde{x}_k\|^2 \\
& - \frac{\mu_x}{4}\|x - x_{k+1}\|^2 - \frac{2\mu_x - \sqrt{\alpha\mu_x}\mu_x}{16}\|x_k - x_{k+1}\|^2. \quad (109)
\end{aligned}
$$

Next, we characterize the smoothness property of $\Phi(x)$. Using the form of $\nabla\Phi(x)$ in eq. (7), and based on Assumptions 1, 2 and Assumption 3 that $\|\nabla_y f(\cdot,\cdot)\| \leq U$, we have, for any $x_1, x_2 \in \mathbb{R}^p$,

$$
\begin{aligned}
& \|\nabla\Phi(x_1) - \nabla\Phi(x_2)\| \\
& \quad \leq \|\nabla_x f(x_1, y^*(x_1)) - \nabla_x f(x_2, y^*(x_2))\| \\
& \quad\quad + \|\nabla_x\nabla_y g(x_1, y^*(x_1))\nabla_y^2 g(x_1, y^*(x_1))^{-1}\nabla_y f(x_1, y^*(x_1)) \\
& \quad\quad\quad - \nabla_x\nabla_y g(x_2, y^*(x_2))\nabla_y^2 g(x_2, y^*(x_2))^{-1}\nabla_y f(x_2, y^*(x_2))\| \\
& \quad \leq L_x\|x_1 - x_2\| + L_{xy}\|y^*(x_1) - y^*(x_2)\| + \frac{\widetilde{L}_{xy}}{\mu_y}(L_{xy}\|x_1 - x_2\| + L_y\|y^*(x_1) - y^*(x_2)\|)
\end{aligned}
$$

$$+\Big(\frac{U\rho_{xy}}{\mu_y}+\frac{\widetilde{L}_{xy}U\rho_{yy}}{\mu_y^2}\Big)(\|x_1-x_2\|+\|y^*(x_1)-y^*(x_2)\|),$$

which, combined with Lemma 2.2 in Ghadimi and Wang (2018) that $\|y^*(x_1)-y^*(x_2)\|\leq \frac{\widetilde{L}_{xy}}{\mu_y}\|x_1-x_2\|$, yields

$$\begin{aligned}&\|\nabla\Phi(x_1)-\nabla\Phi(x_2)\|\\ &\quad\leq \Big(\underbrace{L_x+\frac{2L_{xy}\widetilde{L}_{xy}}{\mu_y}+\Big(\frac{U\rho_{xy}}{\mu_y}+\frac{U\widetilde{L}_{xy}\rho_{yy}}{\mu_y^2}\Big)\Big(1+\frac{\widetilde{L}_{xy}}{\mu_y}\Big)+\frac{\widetilde{L}_{xy}^2L_y}{\mu_y^2}}_{L_\Phi}\Big)\|x_1-x_2\|.\end{aligned}\tag{110}$$

Then, based on the above L_Φ-smoothness of $\Phi(\cdot)$, we have

$$\begin{aligned}\Phi(z_{k+1})\leq&\Phi(\widetilde{x}_k)+\langle\nabla\Phi(\widetilde{x}_k),z_{k+1}-\widetilde{x}_k\rangle+\frac{L_\Phi}{2}\|z_{k+1}-\widetilde{x}_k\|^2\\ =&\big(1-\frac{\sqrt{\alpha\mu_x}}{2}\big)(\Phi(\widetilde{x}_k)+\langle\nabla\Phi(\widetilde{x}_k),z_{k+1}-\widetilde{x}_k\rangle)\\ &+\frac{\sqrt{\alpha\mu_x}}{2}(\Phi(\widetilde{x}_k)+\langle\nabla\Phi(\widetilde{x}_k),z_{k+1}-\widetilde{x}_k\rangle)+\frac{L_\Phi}{2}\|z_{k+1}-\widetilde{x}_k\|^2.\end{aligned}\tag{111}$$

Adding eq. (109) and eq. (111) yields

$$\begin{aligned}\Phi(z_{k+1})\leq&\big(1-\frac{\sqrt{\alpha\mu_x}}{2}\big)(\Phi(\widetilde{x}_k)+\langle\nabla\Phi(\widetilde{x}_k),z_k-\widetilde{x}_k\rangle)+\frac{\sqrt{\alpha\mu_x}}{2}(\Phi(\widetilde{x}_k)+\langle\nabla\Phi(\widetilde{x}_k),x-\widetilde{x}_k\rangle)\\ &+\Big\langle\nabla\Phi(\widetilde{x}_k)-G_k,\frac{\sqrt{\alpha\mu_x}}{2}(z_{k+1}-x)+(1-\frac{\sqrt{\alpha\mu_x}}{2})(z_{k+1}-z_k)\Big\rangle\\ &+\frac{\sqrt{\alpha\mu_x}\mu_x}{8}\|x-\widetilde{x}_k\|^2-\frac{1}{2\alpha}(1-\alpha L_\Phi)\|z_{k+1}-\widetilde{x}_k\|^2-\frac{\mu_x\sqrt{\alpha\mu_x}}{16}\|x_{k+1}-\widetilde{x}_k\|^2\\ &-\frac{\mu_x}{4}\|x-x_{k+1}\|^2-\frac{2\mu_x-\sqrt{\alpha\mu_x}\mu_x}{16}\|x_k-x_{k+1}\|^2,\end{aligned}$$

which, in conjunction with the strong-convexity of $\Phi(\cdot)$, $\sqrt{\alpha\mu_x}\leq 1$ and $\alpha\leq\frac{1}{2L_\Phi}$, yields

$$\begin{aligned}\Phi(z_{k+1})\leq&\big(1-\frac{\sqrt{\alpha\mu_x}}{2}\big)\big(\Phi(z_k)-\frac{\mu_x}{2}\|z_k-\widetilde{x}_k\|^2\big)+\frac{\sqrt{\alpha\mu_x}}{2}\big(\Phi(x)-\frac{\mu_x}{2}\|x-\widetilde{x}_k\|^2\big)\\ &+\Big\langle\nabla\Phi(\widetilde{x}_k)-G_k,\frac{\sqrt{\alpha\mu_x}}{2}(z_{k+1}-x)+(1-\frac{\sqrt{\alpha\mu_x}}{2})(z_{k+1}-z_k)\Big\rangle\\ &+\frac{\sqrt{\alpha\mu_x}\mu_x}{8}\|x-\widetilde{x}_k\|^2-\frac{1}{4\alpha}\|z_{k+1}-\widetilde{x}_k\|^2-\frac{\mu_x\sqrt{\alpha\mu_x}}{16}\|x_{k+1}-\widetilde{x}_k\|^2.\end{aligned}\tag{112}$$

Note that we have the equality that

$$\begin{aligned}\frac{\sqrt{\alpha\mu_x}}{2}(z_{k+1}-x)+&(1-\frac{\sqrt{\alpha\mu_x}}{2})(z_{k+1}-z_k)\\ &=(z_{k+1}-\widetilde{x}_k)+\frac{\sqrt{\alpha\mu_x}}{2}(\widetilde{x}_k-x)+(1-\frac{\sqrt{\alpha\mu_x}}{2})(\widetilde{x}_k-z_k).\end{aligned}\tag{113}$$

Then, using eq. (113) and the Cauchy-Schwarz inequality, we have

$$\begin{aligned}
&\Big\langle \nabla\Phi(\widetilde{x}_k)-G_k, \frac{\sqrt{\alpha\mu_x}}{2}(z_{k+1}-x)+(1-\frac{\sqrt{\alpha\mu_x}}{2})(z_{k+1}-z_k)\Big\rangle \\
&\quad\leq\Big(2\alpha+\frac{1}{2\mu_x}+\frac{\sqrt{\alpha\mu_x}}{4\mu_x}\Big)\|\nabla\Phi(\widetilde{x}_k)-G_k\|^2+\frac{1}{8\alpha}\|z_{k+1}-\widetilde{x}_k\|^2+\frac{\sqrt{\alpha\mu_x}\mu_x}{8}\|\widetilde{x}_k-x\| \\
&\qquad+(1-\frac{\sqrt{\alpha\mu_x}}{2})\frac{\mu_x}{2}\|z_k-\widetilde{x}_k\|^2. \qquad (114)
\end{aligned}$$

Substituting eq. (114) into eq. (112) and cancelling out negative terms, we have

$$\begin{aligned}
\Phi(z_{k+1})\leq&\big(1-\frac{\sqrt{\alpha\mu_x}}{2}\big)\Phi(z_k)+\frac{\sqrt{\alpha\mu_x}}{2}\Phi(x)-\frac{1}{8\alpha}\|z_{k+1}-\widetilde{x}_k\|^2-\frac{\mu_x\sqrt{\alpha\mu_x}}{16}\|x_{k+1}-\widetilde{x}_k\|^2 \\
&+\Big(2\alpha+\frac{1}{2\mu_x}+\frac{\sqrt{\alpha\mu_x}}{4\mu_x}\Big)\|\nabla\Phi(\widetilde{x}_k)-G_k\|^2. \qquad (115)
\end{aligned}$$

We next upper-bound the hypergradient estimation error $\|\nabla\Phi(\widetilde{x}_k)-G_k\|^2$. Recall that

$$G_k:=\nabla_x f(\widetilde{x}_k,y_k^N)-\nabla_x\nabla_y g(\widetilde{x}_k,y_k^N)v_k^M, \qquad (116)$$

where v_k^M is the M^{th} step output of the heavy-ball method for solving

$$\min_v Q(v):=\frac{1}{2}v^T\nabla_y^2 g(\widetilde{x}_k,y_k^N)v-v^T\nabla_y f(\widetilde{x}_k,y_k^N)$$

Then, based on the convergence result of the heavy-ball method in Badithela and Seiler (2019) with the stepsizes $\lambda=\frac{4}{(\sqrt{\widetilde{L}_y}+\sqrt{\mu_y})^2}$ and $\theta=\max\big\{\big(1-\sqrt{\lambda\mu_y}\big)^2,\big(1-\sqrt{\lambda\widetilde{L}_y}\big)^2\big\}$, we have

$$\begin{aligned}
\|v_k^M-\nabla_y^2 g(\widetilde{x}_k,y_k^N)^{-1}\nabla_y f(\widetilde{x}_k,y_k^N)\|\leq&\Big(\frac{\sqrt{\kappa_y}-1}{\sqrt{\kappa_y}+1}\Big)^M\Big\|\nabla_y^2 g(\widetilde{x}_k,y_k^N)^{-1}\nabla_y f(\widetilde{x}_k,y_k^N)\Big\| \\
\overset{(i)}{\leq}&\frac{U}{\mu_y}\Big(\frac{\sqrt{\kappa_y}-1}{\sqrt{\kappa_y}+1}\Big)^M, \qquad (117)
\end{aligned}$$

where (i) follows from Assumption 3 that $\|\nabla_y f(\cdot,\cdot)\|\leq U$. Let $y_k^*=\arg\min_y g(\widetilde{x}_k,y)$. Then, based on the form of $\nabla\Phi(x)$ in eq. (7), we have

$$\begin{aligned}
\|G_k-&\nabla\Phi(\widetilde{x}_k)\| \\
\leq&\|\nabla_x f(\widetilde{x}_k,y_k^N)-\nabla_x f(\widetilde{x}_k,y_k^*)\|+\widetilde{L}_{xy}\|v_k^M-\nabla_y^2 g(\widetilde{x}_k,y_k^*)^{-1}\nabla_y f(\widetilde{x}_k,y_k^*)\| \\
&+\frac{\|\nabla_y f(\widetilde{x}_k,y_k^*)\|}{\mu_y}\|\nabla_x\nabla_y g(\widetilde{x}_k,y_k^N)-\nabla_x\nabla_y g(\widetilde{x}_k,y_k^*)\| \\
\leq&L_y\|y_k^*-y_k^N\|+\widetilde{L}_{xy}\|v_k^M-\nabla_y^2 g(\widetilde{x}_k,y_k^N)^{-1}\nabla_y f(\widetilde{x}_k,y_k^N)\| \\
&+\widetilde{L}_{xy}\big\|\nabla_y^2 g(\widetilde{x}_k,y_k^N)^{-1}\nabla_y f(\widetilde{x}_k,y_k^N)-\nabla_y^2 g(\widetilde{x}_k,y_k^*)^{-1}\nabla_y f(\widetilde{x}_k,y_k^*)\big\|+\frac{U\rho_{xy}}{\mu_y}\|y_k^N-y_k^*\| \\
\overset{(i)}{\leq}&\Big(L_y+\frac{\widetilde{L}_{xy}L_y}{\mu_y}+\Big(\frac{\rho_{xy}}{\mu_y}+\frac{\widetilde{L}_{xy}\rho_{yy}}{\mu_y^2}\Big)U\Big)\|y_k^N-y_k^*\|+\frac{U\widetilde{L}_{xy}}{\mu_y}\Big(\frac{\sqrt{\kappa_y}-1}{\sqrt{\kappa_y}+1}\Big)^M, \qquad (118)
\end{aligned}$$

where (i) follows from eq. (117). Note that y_k^N is obtained as the N^{th} step output of AGD. Then, based on the analysis in Nesterov (2003) for AGD, we have

$$\begin{aligned}\|y_k^N - y_k^*\|^2 \leq & \frac{\widetilde{L}_y + \mu_y}{\mu_y}\|y_k^0 - y_k^*\|^2 \exp\Big(-\frac{N}{\sqrt{\kappa_y}}\Big) = \frac{\widetilde{L}_y + \mu_y}{\mu_y}\|y_{k-1}^N - y_k^*\|^2 \exp\Big(-\frac{N}{\sqrt{\kappa_y}}\Big) \\ \leq & \frac{2(\widetilde{L}_y + \mu_y)}{\mu_y}\exp\Big(-\frac{N}{\sqrt{\kappa_y}}\Big)(\|y_{k-1}^N - y_{k-1}^*\|^2 + \|y_{k-1}^* - y_k^*\|^2) \\ \leq & \underbrace{\frac{2(\widetilde{L}_y + \mu_y)}{\mu_y}\exp\Big(-\frac{N}{\sqrt{\kappa_y}}\Big)}_{\tau_N}(\|y_{k-1}^N - y_{k-1}^*\|^2 + \kappa_y\|\widetilde{x}_k - \widetilde{x}_{k-1}\|^2), \end{aligned} \tag{119}$$

which, in conjunction with $\widetilde{x}_k - \widetilde{x}_{k-1} = \eta_k(x_k - \widetilde{x}_{k-1}) + (1-\eta_k)(z_k - \widetilde{x}_{k-1})$, yields

$$\begin{aligned}\|y_k^N - y_k^*\|^2 \leq & \tau_N\|y_{k-1}^N - y_{k-1}^*\|^2 + \kappa_y\eta_k\tau_N\|x_k - \widetilde{x}_{k-1}\|^2 \\ & + \kappa_y(1-\eta_k)\tau_N\|z_k - \widetilde{x}_{k-1}\|^2. \end{aligned} \tag{120}$$

Telescoping eq. (120) over k yields

$$\|y_k^N - y_k^*\|^2 \leq \tau_N^k\|y_0^N - y_0^*\|^2 + \sum_{i=0}^{k-1}\tau_N^{k-i}\kappa_y\eta_k\|x_{i+1} - \widetilde{x}_i\|^2 + \sum_{i=0}^{k-1}\tau_N^{k-i}\kappa_y(1-\eta_k)\|z_{i+1} - \widetilde{x}_i\|^2,$$

which, in conjunction with eq. (115) and eq. (118) and letting $x = x^*$, yields

$$\begin{aligned}\Phi(z_{k+1}) - \Phi(x^*) \leq & \big(1 - \frac{\sqrt{\alpha\mu_x}}{2}\big)(\Phi(z_k) - \Phi(x^*) - \frac{1}{8\alpha}\|z_{k+1} - \widetilde{x}_k\|^2 - \frac{\mu_x\sqrt{\alpha\mu_x}}{16}\|x_{k+1} - \widetilde{x}_k\|^2 \\ & + \lambda\sum_{i=0}^{k-1}\tau_N^{k-i}\kappa_y\eta_k\|x_{i+1} - \widetilde{x}_i\|^2 + \lambda\sum_{i=0}^{k-1}\tau_N^{k-i}\kappa_y(1-\eta_k)\|z_{i+1} - \widetilde{x}_i\|^2 \\ & + \Delta + \lambda\tau_N^k\|y_0^* - y_0^N\|^2, \end{aligned} \tag{121}$$

where Δ and λ are given by

$$\begin{aligned}\Delta = & \Big(4\alpha + \frac{1}{\mu_x} + \frac{\sqrt{\alpha\mu_x}}{2\mu_x}\Big)\frac{U^2\widetilde{L}_{xy}^2}{\mu_y^2}\Big(\frac{\sqrt{\kappa_y}-1}{\sqrt{\kappa_y}+1}\Big)^{2M} \\ \lambda = & \Big(4\alpha + \frac{1}{\mu_x} + \frac{\sqrt{\alpha\mu_x}}{2\mu_x}\Big)\Big(L_y + \frac{\widetilde{L}_{xy}L_y}{\mu_y} + \Big(\frac{\rho_{xy}}{\mu_y} + \frac{\widetilde{L}_{xy}\rho_{yy}}{\mu_y^2}\Big)U\Big)^2. \end{aligned} \tag{122}$$

Telescoping eq. (121) over k from 0 to $K-1$ and noting that $0 < \eta_k \leq 1$, we have

$$\begin{aligned}\Phi(z_K) - \Phi(x^*) \leq & \big(1 - \frac{\sqrt{\alpha\mu_x}}{2}\big)^K(\Phi(z_0) - \Phi(x^*)) - \frac{1}{8\alpha}\sum_{k=0}^{K-1}\big(1 - \frac{\sqrt{\alpha\mu_x}}{2}\big)^{K-1-k}\|z_{k+1} - \widetilde{x}_k\|^2 \\ & - \frac{\mu_x\sqrt{\alpha\mu_x}}{16}\sum_{k=0}^{K-1}\big(1 - \frac{\sqrt{\alpha\mu_x}}{2}\big)^{K-1-k}\|x_{k+1} - \widetilde{x}_k\|^2 + \frac{2\Delta}{\sqrt{\alpha\mu_x}} \\ & + \sum_{k=0}^{K-1}\big(1 - \frac{\sqrt{\alpha\mu_x}}{2}\big)^{K-1-k}\lambda\tau_N^k\|y_0^* - y_0^N\|^2 \end{aligned}$$

$$
\begin{aligned}
&+ \lambda \sum_{k=0}^{K-1} \big(1-\frac{\sqrt{\alpha\mu_x}}{2}\big)^{K-1-k} \sum_{i=0}^{k-1} \tau_N^{k-i}\kappa_y \|x_{i+1}-\widetilde{x}_i\|^2 \\
&+ \lambda \sum_{k=0}^{K-1} \big(1-\frac{\sqrt{\alpha\mu_x}}{2}\big)^{K-1-k} \sum_{i=0}^{k-1} \tau_N^{k-i}\kappa_y \|z_{i+1}-\widetilde{x}_i\|^2,
\end{aligned}
$$

which, in conjunction with the fact that $k \leq K-1$, yields

$$
\begin{aligned}
\Phi(z_K)-\Phi(x^*) \leq{}& \big(1-\frac{\sqrt{\alpha\mu_x}}{2}\big)^K(\Phi(z_0)-\Phi(x^*)) - \frac{1}{8\alpha}\sum_{k=0}^{K-1}\big(1-\frac{\sqrt{\alpha\mu_x}}{2}\big)^{K-1-k}\|z_{k+1}-\widetilde{x}_k\|^2 \\
&- \frac{\mu_x\sqrt{\alpha\mu_x}}{16}\sum_{k=0}^{K-1}\big(1-\frac{\sqrt{\alpha\mu_x}}{2}\big)^{K-1-k}\|x_{k+1}-\widetilde{x}_k\|^2 + \frac{2\Delta}{\sqrt{\alpha\mu_x}} \\
&+ \sum_{k=0}^{K-1}\big(1-\frac{\sqrt{\alpha\mu_x}}{2}\big)^{K-1-k}\lambda\tau_N^k\|y_0^*-y_0^N\|^2 \\
&+ \frac{2\tau_N\lambda\kappa_y}{\sqrt{\alpha\mu_x}}\sum_{i=0}^{K-2}\tau_N^{K-2-i}\|x_{i+1}-\widetilde{x}_i\|^2 + \frac{2\tau_N\lambda\kappa_y}{\sqrt{\alpha\mu_x}}\sum_{i=0}^{K-2}\tau_N^{K-2-i}\|z_{i+1}-\widetilde{x}_i\|^2. \qquad (123)
\end{aligned}
$$

Recall the definition of τ_N in eq. (119). Then, choose N such that

$$
\tau_N = \frac{2(\widetilde{L}_y+\mu_y)}{\mu_y}\exp\Big(-\frac{N}{\sqrt{\kappa_y}}\Big) \leq \min\Big\{\frac{\sqrt{\mu_x}}{16\lambda\kappa_y\sqrt{\alpha}}, \frac{\alpha\mu_x^2}{32\lambda\kappa_y}, \big(1-\frac{\sqrt{\alpha\mu_x}}{2}\big)^2\Big\}, \qquad (124)
$$

which, in conjunction with eq. (123), yields

$$
\Phi(z_K)-\Phi(x^*) \leq \big(1-\frac{\sqrt{\alpha\mu_x}}{2}\big)^K\Big(\Phi(z_0)-\Phi(x^*)+\frac{2\lambda\|y_0^*-y_0^N\|^2}{\sqrt{\alpha\mu_x}}\Big) + \frac{2\Delta}{\sqrt{\alpha\mu_x}}.
$$

Then, based on the definitions of λ and Δ in eq. (122) and L_Φ in eq. (110), to achieve $\Phi(z^K)-\Phi(x^*) \leq \epsilon$, we have

$$
K = \widetilde{\mathcal{O}}\Big(\sqrt{\frac{U\widetilde{L}_{xy}^2\rho_{yy}}{\mu_x\mu_y^3}}\Big), \quad M = \widetilde{\mathcal{O}}\Big(\sqrt{\frac{\widetilde{L}_y}{\mu_y}}\Big). \qquad (125)
$$

In addition, it follows from eq. (124) that

$$
N = \widetilde{\mathcal{O}}\Big(\sqrt{\frac{\widetilde{L}_y}{\mu_y}}\Big). \qquad (126)
$$

Based on eq. (125) and eq. (126), the total complexity is given by

$$
\begin{aligned}
\mathcal{C}_{\text{fun}}(\mathcal{A},\epsilon) &= \mathcal{O}(n_J+n_H+n_G) = \mathcal{O}(K+KM+KN) \\
&= \widetilde{\mathcal{O}}\Big(\sqrt{\frac{U\widetilde{L}_{xy}^2\rho_{yy}\widetilde{L}_y}{\mu_x\mu_y^4}}\Big),
\end{aligned}
$$

which finishes the proof.

Appendix K. Proof of Theorem 14

Let $\widetilde{x}^*$ be the minimizer of $\widetilde{\Phi}(\cdot)$. Then, applying the results in Theorem 13 to $\widetilde{\Phi}(x)$ with the strongly-convex parameter $\mu_x = \frac{\epsilon}{B^2}$ and choosing $N = \Theta\Big(\sqrt{\frac{\widetilde{L}_y}{\mu_y}}\log(\text{poly}(B,\epsilon,\mu_y,U))\Big)$, we have

$$\widetilde{\Phi}(z_K) - \widetilde{\Phi}(\widetilde{x}^*) \leq \big(1 - \frac{\sqrt{\epsilon}}{2\sqrt{2L_{\widetilde{\Phi}}}B}\big)^K \Big(\widetilde{\Phi}(z_0) - \widetilde{\Phi}(\widetilde{x}^*) + \frac{2\sqrt{2L_{\widetilde{\Phi}}}B\widetilde{\lambda}\|y_0^* - y_0^N\|^2}{\sqrt{\alpha\epsilon}}\Big) + \frac{2\widetilde{\Delta}\sqrt{2L_{\widetilde{\Phi}}}B}{\sqrt{\epsilon}},$$

where $\widetilde{\Delta}$ and $\widetilde{\lambda}$ take the same forms as Δ and λ in eq. (122) with μ_x being replaced by $\frac{\epsilon}{B^2}$. By choosing $M = \Theta\Big(\sqrt{\frac{\widetilde{L}_y}{\mu_y}}\log\frac{\text{poly}(B,\epsilon,\mu_y,U)}{\epsilon}\Big)$ in $\widetilde{\Delta}$, we have $\frac{2\widetilde{\Delta}\sqrt{2L_{\widetilde{\Phi}}}B}{\sqrt{\epsilon}} \leq \frac{\epsilon}{4}$, and hence

$$\widetilde{\Phi}(z_K) - \widetilde{\Phi}(\widetilde{x}^*) \leq \big(1 - \frac{\sqrt{\epsilon}}{2\sqrt{2L_{\widetilde{\Phi}}}B}\big)^K \Big(\widetilde{\Phi}(z_0) - \widetilde{\Phi}(\widetilde{x}^*) + \frac{2\sqrt{2L_{\widetilde{\Phi}}}B\widetilde{\lambda}\|y_0^* - y_0^N\|^2}{\sqrt{\alpha\epsilon}}\Big) + \frac{\epsilon}{4},$$

which, in conjunction with $\widetilde{\Phi}(z_K) \geq \Phi(z_K)$, $\widetilde{\Phi}(\widetilde{x}^*) \leq \widetilde{\Phi}(x^*) = \Phi(x^*) + \frac{\epsilon}{2B^2}\|x^*\|^2$ and $z_0 = 0$, yields

$$\begin{aligned}\Phi(z_K) - \Phi(x^*) \leq& \big(1 - \frac{\sqrt{\epsilon}}{2\sqrt{2L_{\widetilde{\Phi}}}B}\big)^K \Big(\Phi(0) - \widetilde{\Phi}(\widetilde{x}^*) + \frac{2\sqrt{2L_{\widetilde{\Phi}}}B\widetilde{\lambda}\|y_0^* - y_0^N\|^2}{\sqrt{\alpha\epsilon}}\Big) \\ &+ \frac{\epsilon}{4} + \frac{\epsilon}{2B^2}\|x^*\|^2. \end{aligned} \tag{127}$$

Based on eq. (99), we have $\Phi(0) - \widetilde{\Phi}(\widetilde{x}^*) \leq \Phi(0) - \Phi(x^*)$, which, combined with $\|x^*\| = B$ and $K = \widetilde{\Theta}\Big(B\sqrt{\frac{\widetilde{L}_{xy}^2\rho_{yy}}{\epsilon\mu_y^3}}\Big)$, yields $\Phi(z_K) - \Phi(x^*) \leq \epsilon$. Then, the total complexity satisfies

$$\mathcal{C}_{\text{fun}}(\mathcal{A},\epsilon) = \mathcal{O}(n_J + n_H + n_G) = \mathcal{O}(K + KM + KN) = \widetilde{\mathcal{O}}\Big(B\sqrt{\frac{\widetilde{L}_{xy}^2\rho_{yy}\widetilde{L}_y}{\epsilon\mu_y^4}}\Big), \tag{128}$$

which finishes the proof.

References

Eitaro Aiyoshi and Kiyotaka Shimizu. A solution method for the static constrained stackelberg problem via penalty method. *IEEE Transactions on Automatic Control*, 29(12): 1111–1114, 1984.

Faiz A Al-Khayyal, Reiner Horst, and Panos M Pardalos. Global optimization of concave functions subject to quadratic constraints: an application in nonlinear bilevel programming. *Annals of Operations Research*, 34(1):125–147, 1992.

Sanjeev Arora, Simon S Du, Sham Kakade, Yuping Luo, and Nikunj Saunshi. Provable representation learning for imitation learning via bi-level optimization. In *Proc. International Conference on Machine Learning (ICML)*, 2020.

Apurva Badithela and Peter Seiler. Analysis of the heavy-ball algorithm using integral quadratic constraints. In *2019 American Control Conference (ACC)*, pages 4081–4085. IEEE, 2019.

Juhan Bae and Roger Grosse. Delta-STN: Efficient bilevel optimization for neural networks using structured response Jacobians. *arXiv preprint arXiv:2010.13514*, 2020.

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

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

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

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points i. *Mathematical Programming*, pages 1–50, 2019.

Tianyi Chen, Yuejiao Sun, and Wotao Yin. A single-timescale stochastic bilevel optimization method. *arXiv preprint arXiv:2102.04671*, 2021.

Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Damek Davis and Dmitriy Drusvyatskiy. Stochastic model-based minimization of weakly convex functions. *SIAM Journal on Optimization*, 29(1):207–239, 2019.

Justin Domke. Generic methods for optimization-based modeling. In *Artificial Intelligence and Statistics (AISTATS)*, pages 318–326, 2012.

Thomas Arthur Edmunds and Jonathan F Bard. Algorithms for nonlinear bilevel mathematical programs. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(1):83–89, 1991.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1082–1092. PMLR, 2020.

Matthias Feurer and Frank Hutter. Hyperparameter optimization. In *Automated Machine Learning*, pages 3–33. Springer, Cham, 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proc. International Conference on Machine Learning (ICML)*, pages 1126–1135, 2017.

Chuan-sheng Foo, Chuong B Do, and Andrew Y Ng. Efficient multiple hyperparameter learning for log-linear models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 377–384, 2008.

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

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 (ICML)*, pages 1568–1577, 2018.

Saeed Ghadimi and Guanghui Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming*, 156(1-2):59–99, 2016.

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

Riccardo Grazzi, Luca Franceschi, Massimiliano Pontil, and Saverio Salzo. On the iteration complexity of hypergradient computation. In *Proc. International Conference on Machine Learning (ICML)*, 2020.

Andreas Griewank. Some bounds on the complexity of gradients, jacobians, and hessians. In *Complexity in Numerical Optimization*, pages 128–162. World Scientific, 1993.

Zhishuai Guo, Yi Xu, Wotao Yin, Rong Jin, and Tianbao Yang. On stochastic moving-average estimators for non-convex optimization. *arXiv preprint arXiv:2104.14840*, 2021.

Pierre Hansen, Brigitte Jaumard, and Gilles Savard. New branch-and-bound rules for linear bilevel programming. *SIAM Journal on Scientific and Statistical Computing*, 13(5):1194–1217, 1992.

Chaoyang He, Haishan Ye, Li Shen, and Tong Zhang. Milenas: Efficient neural architecture search via mixed-level reformulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11993–12002, 2020.

Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic. *arXiv preprint arXiv:2007.05170*, 2020.

Feihu Huang and Heng Huang. Biadam: Fast adaptive bilevel optimization methods. *arXiv preprint arXiv:2106.11396*, 2021.

Simon Jenni and Paolo Favaro. Deep bilevel learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 618–633, 2018.

Kaiyi Ji and Yingbin Liang. Minimax estimation of neural net distance. *Advances in Neural Information Processing Systems*, 31, 2018.

Kaiyi Ji, Jason D Lee, Yingbin Liang, and H Vincent Poor. Convergence of meta-learning with task-specific adaptation over partial parameter. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020a.

Kaiyi Ji, Junjie Yang, and Yingbin Liang. Multi-step model-agnostic meta-learning: Convergence and improved algorithms. *arXiv preprint arXiv:2002.07836*, 2020b.

Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. In *International Conference on Machine Learning (ICML)*, pages 4882–4892. PMLR, 2021.

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

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. *arXiv preprint arXiv:2102.07367*, 2021.

Junyi Li, Bin Gu, and Heng Huang. Improved bilevel model: Fast and optimal algorithm with theoretical guarantee. *arXiv preprint arXiv:2009.00690*, 2020.

Tianyi Lin, Chi Jin, and Michael I Jordan. Near-optimal algorithms for minimax optimization. In *Conference on Learning Theory (COLT)*, pages 2738–2779. PMLR, 2020.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *International Conference on Learning Representations (ICLR)*, 2019.

Risheng Liu, Pan Mu, Xiaoming Yuan, Shangzhi Zeng, and Jin Zhang. A generic first-order algorithmic framework for bi-level programming beyond lower-level singleton. In *International Conference on Machine Learning (ICML)*, 2020.

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 *Proc. International Conference on Machine Learning (ICML)*, 2021.

Yibing Lv, Tiesong Hu, Guangmin Wang, and Zhongping Wan. A penalty function method based on Kuhn–Tucker condition for solving linear bilevel programming. *Applied Mathematics and Computation*, 188(1):808–813, 2007.

Matthew Mackay, Paul Vicol, Jonathan Lorraine, David Duvenaud, and Roger Grosse. Self-tuning networks: Bilevel optimization of hyperparameters using structured best-response functions. In *International Conference on Learning Representations (ICLR)*, 2018.

Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *International Conference on Machine Learning (ICML)*, pages 2113–2122, 2015.

Akshay Mehra and Jihun Hamm. Penalty method for inversion-free deep bilevel optimization. *arXiv preprint arXiv:1911.03432*, 2019.

Gregory M Moore. *Bilevel programming algorithms for machine learning model selection*. Rensselaer Polytechnic Institute, 2010.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87. Springer Science & Business Media, 2003.

Yurii Nesterov et al. *Lectures on Convex Optimization*, volume 137. Springer, 2018.

Takayuki Okuno, Akiko Takeda, and Akihiro Kawana. Hyperparameter learning via bilevel nonsmooth optimization. *arXiv preprint arXiv:1806.01520*, 2018.

Yuyuan Ouyang and Yangyang Xu. Lower complexity bounds of first-order methods for convex-concave bilinear saddle-point problems. *Mathematical Programming*, pages 1–35, 2019.

Fabian Pedregosa. Hyperparameter optimization with approximate gradient. In *International Conference on Machine Learning (ICML)*, pages 737–746, 2016.

Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of MAML. *International Conference on Learning Representations (ICLR)*, 2019.

Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 113–124, 2019.

Yuji Roh, Kangwook Lee, Steven Whang, and Changho Suh. Sample selection for fair and robust training. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2021.

Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. Truncated back-propagation for bilevel optimization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1723–1732, 2019.

Chenggen Shi, Jie Lu, and Guangquan Zhang. An extended Kuhn–Tucker approach for linear bilevel programming. *Applied Mathematics and Computation*, 162(1):51–63, 2005.

Ankur Sinha, Tanmay Khandait, and Raja Mohanty. A gradient-based bilevel optimization approach for tuning hyperparameters in machine learning. *arXiv preprint arXiv:2007.11022*, 2020.

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

Rayadurgam Srikant and Lei Ying. *Communication networks: an optimization, control, and stochastic networks perspective.* Cambridge University Press, 2013.

Sirui Xie, Hehui Zheng, Chunxiao Liu, and Liang Lin. Snas: stochastic neural architecture search. In *International Conference on Learning Representations (ICLR)*, 2018.

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

Tong Yu and Hong Zhu. Hyper-parameter optimization: A review of algorithms and applications. *arXiv preprint arXiv:2003.05689*, 2020.

Junyu Zhang, Mingyi Hong, and Shuzhong Zhang. On lower iteration complexity bounds for the saddle point problems. *arXiv preprint arXiv:1912.07481*, 2019.