SHARPER ANALYSIS OF SINGLE-LOOP METHODS FOR BILEVEL OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Bilevel optimization underpins many machine learning applications, including hyperparameter optimization, meta-learning, neural architecture search, and reinforcement learning. While hypergradient-based methods have advanced significantly, a gap persists between theoretical guarantees—typically derived for multi-loop algorithms—and practical single-loop implementations required for efficiency. This work narrows that gap by establishing sharper convergence results for single-loop approximate implicit differentiation (AID) and iterative differentiation (ITD) methods. For AID, we improve the convergence rate from $\mathcal{O}(\kappa^6/K)$ to $\mathcal{O}(\kappa^5/K)$, where κ is the condition number of the inner-level problem. For ITD, we prove that the asymptotic error is $\mathcal{O}(\kappa^2)$, exactly matching the known lower bound and improving upon the previous $\mathcal{O}(\kappa^3)$ guarantee. We further validate the refined analyses by the experiments on synthetic bilevel optimization tasks.

1 INTRODUCTION

Bilevel optimization has attracted extensive attention in various applications of machine learning, including hyperparameter optimization (Maclaurin et al., 2015; Franceschi et al., 2017; Shaban et al., 2019; Shen et al., 2024), meta-learning (Chen et al., 2017; Finn et al., 2017; Franceschi et al., 2018), neural architecture search (Liu et al., 2018; He et al., 2020), and reinforcement learning (Zhang et al., 2020; Wang et al., 2020; Shen et al., 2025). Bilevel optimization corresponds to solving one optimization problem subject to constraints defined by another optimization problem. In this paper, we focus on the following bilevel optimization problem:

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

where the outer- and inner-level functions f and g are both jointly continuously differentiable on $\mathbb{R}^m \times \mathbb{R}^n$. We focus on the setting where g is strongly convex with respect to (w.r.t.) the inner-level variable y, which can guarantee the uniqueness of the inner solution (Chen et al., 2024).

Hypergradient-based algorithms have recently gained significant attention for their balance of simplicity and efficiency. Two prominent approaches are approximate implicit differentiation (AID) (Domke, 2012; Pedregosa, 2016; Ghadimi & Wang, 2018; Grazzi et al., 2020; Ji et al., 2021) and iterative differentiation (ITD) (Franceschi et al., 2017; Shaban et al., 2019; Grazzi et al., 2020; Ji et al., 2021; Liu et al., 2021). The key distinction lies in how they estimate the hypergradient $\nabla\Phi(x)$: AID leverages the implicit function theorem, while ITD applies automatic differentiation (see Section 2). Despite this difference, both methods require solving the inner problem to obtain the optimal solution y^*. In practice, however, closed-form solutions are rarely available, and one typically resorts to gradient descent to compute an approximate solution $\hat{y}$.

Most theoretical studies of bilevel optimization analyze algorithms that employ **multi-loop** updates (multi-step gradient descent) for the inner problem and linear-system (Ghadimi & Wang, 2018; Ji et al., 2021; Dong et al., 2025; Fang et al., 2025). In contrast, practical algorithms overwhelmingly adopt **single-loop** updates, where only one inner update is performed per outer iteration. The main appeal of single-loop methods is computational efficiency: they significantly reduce training cost while maintaining competitive performance. This design has become standard across a wide range of applications. For instance, in neural architecture search, DARTS (Liu et al., 2018) updates the

Algorithms	Convergence rate	**MV**(ϵ)	**Gc**(ϵ)
AID (Ji et al., 2022)	$\mathcal{O}(\kappa^6/K)$	$\mathcal{O}(\kappa^6\epsilon^{-1})$	$\mathcal{O}(\kappa^6\epsilon^{-1})$
AID (this paper)	$\mathcal{O}(\kappa^5/K)$	$\mathcal{O}(\kappa^5\epsilon^{-1})$	$\mathcal{O}(\kappa^5\epsilon^{-1})$
ITD (Ji et al., 2022)	$\mathcal{O}(\kappa^3/K+\kappa^3)$	N/A	N/A
ITD (this paper)	$\mathcal{O}(\kappa^3/K+\kappa^2)$	N/A	N/A
Lower bound of ITD	$\Omega(\kappa^2)$	N/A	N/A

Table 1: Comparison of computational complexities of both single-loop AID-based and ITD-based algorithms for finding an ϵ-stationary point. For the last three columns, 'N/A' means that the complexities to achieve an ϵ-accuracy are not measurable due to the nonvanishing convergence error. MV(ϵ): the total number of Jacobian- and Hessian-vector product computations. Gc(ϵ): the total number of gradient computations.

network parameters (y) via single-loop while optimizing architecture coefficients (x). In few-shot meta-learning, MAML (Finn et al., 2017) applies single-loop adaptation to task-specific parameters. In data reweighting for imbalanced or noisy samples, methods such as Ren et al. (2018); Shu et al. (2019) also rely on single-loop updates. These examples underscore a critical gap: while existing theory primarily addresses multi-loop schemes, the algorithms most relevant in practice depend on single-loop updates, making it essential to establish their convergence guarantees.

Recently, Liu et al. (2024) propose MEHA, a Moreau-envelope-based single-loop method with convergence rate $O(1/K^{1/2-p}+1/K^p)$, where K is the number of outer iterations and $p\in(0,1/2)$. Kwon et al. (2023) design F^{3}SA by incorporating momentum, achieving a rate of $O(K^{-2/3})$. However, these single-loop methods remain slower than AID and ITD, both of which can reach $O(K^{-1})$ as shown in Table 1. Motivated by this gap, we focus on the AID and ITD methods and seek sharper analyses for their single-loop variants.

Along similar lines, Ji et al. (2022) analyze different loop structures in bilevel optimization and establish corresponding theoretical results. For AID, Ji et al. (2022) establish a convergence of $\mathcal{O}(\kappa^6/K)$ in the single-loop setting, where $\kappa=\frac{L}{\mu}$ denotes the condition number (L and μ are the gradient Lipschitz and strong convexity constants defined respectively in Assumptions 1 and 3). This is still inferior to the $\mathcal{O}(\kappa^4/K)$ rate achieved by the multi-loop AID. Therefore, our work first aims to narrow the gap of the convergence between the single-loop and multi-loop AID-based methods:

- Our first contribution is that, via a refined analysis and a novel analytical methodology, we show that the single-loop AID algorithm can achieve a convergence rate of $\mathcal{O}(\kappa^5/K)$, thereby providing a more practical and theoretically grounded alternative for large-scale bilevel optimization tasks where previous guarantees of $\mathcal{O}(\kappa^6/K)$ limited reliability.

For ITD, Ji et al. (2022) show that single-loop suffers from an inherent error of order $\mathcal{O}(\kappa^3)$, leaving a gap of $\alpha\mu$ (with α the inner-level step size) from the fundamental lower bound. They identify closing this gap as an open problem.

- Our second contribution is that the single-loop ITD method can attain a convergence error of order $\mathcal{O}(\kappa^2)$, exactly matching the lower bound of Ji et al. (2022), thereby establishing its theoretical optimality and potentially supporting it as an efficient alternative to more costly multi-loop methods.

Moreover, our key technical contribution is a novel analytical framework that departs from the standard proof template. Prior analyses bound the squared error norm directly, which inflates the dependence on κ. We instead decouple the analysis by first bounding the error norm and only then squaring it. This delicate treatment avoids the overestimation and yields sharper bounds, providing a more accurate characterization of both AID and ITD.

2 Algorithms

In this section, we introduce two popular bilevel optimization algorithms to solve problem (1). It is worth noting that we provide the single-loop algorithms, as this aligns with practical choices in related applications.

Algorithm 1 Single-Loop AID-based bilevel optimization algorithm

1: **Input:** Learning rates $\alpha, \beta, \eta > 0$, initializations x_0, y_0, v_0.
2: **for** $k = 0, 1, 2, ..., K$ **do**
3: Set $y_k^0 = \hat{y}_{k-1}$ if $k > 0$ and y_0 otherwise ***(warm start initialization)***
4: Update $\hat{y}_k = y_k^0 - \alpha \nabla_y g(x_k, y_k^0)$
5: Set $v_k^0 = \hat{v}_{k-1}$ if $k > 0$ and v_0 otherwise ***(warm start initialization)***
6: Update $\hat{v}_k = (I - \eta \nabla_y^2 g(x_k, \hat{y}_k)) v_k^0 + \eta \nabla_y f(x_k, \hat{y}_k)$
7: Compute $\widehat{\nabla} \Phi(x_k) = \nabla_x f(x_k, \hat{y}_k) - \nabla_{xy}^2 g(x_k, \hat{y}_k) \hat{v}_k$
8: Update $x_{k+1} = x_k - \beta \widehat{\nabla} \Phi(x_k)$
9: **end for**

Algorithm 2 Single-Loop ITD-based bilevel optimization algorithm

1: **Input:** Learning rate $\alpha, \beta > 0$, initializations x_0 and y_0.
2: **for** $k = 0, 1, 2, ..., K$ **do**
3: Set $y_k^0 = \hat{y}_{k-1}$ if $k > 0$ and y_0 otherwise ***(warm start initialization)***
4: Update $\hat{y}_k(x_k) = y_k^0 - \alpha \nabla_y g(x_k, y_k^0)$
5: Compute $\widehat{\nabla} \Phi(x_k) = \nabla_x f(x_k, \hat{y}_k) - \alpha \nabla_{xy}^2 g(x_k, y_k^0) \nabla_y f(x_k, \hat{y}_k)$
6: Update $x_{k+1} = x_k - \beta \widehat{\nabla} \Phi(x_k)$
7: **end for**

2.1 AID-based Bilevel Optimization Algorithm

We provide the single-loop AID-based bilevel optimization algorithm (for simplicity, hereafter referred to as AID) in Algorithm 1. In each outer-level iteration k, AID first performs one step of gradient descent on the inner-level function $g(x, y)$ to find a point $\hat{y}_k$ that approximates y_k^*, where y_k^* denotes $\arg\min_y g(x_k, y)$. Moreover, to accelerate the practical training process, AID usually adopts a warm-start strategy. In other words, the initial value y_k^0 of the inner-level problem at iteration k is set to the updated value $\hat{y}_{k-1}$ from iteration $k-1$.

In the outer-level, AID first obtain $\hat{v}_k$ via solving a linear system $\nabla_y^2 g(x_k, \hat{y}_k) v = \nabla_y f(x_k, \hat{y}_k)$ by one step of gradient descent starting form v_k^0, and then AID can estimate the gradient $\nabla \Phi(x_k) = \nabla_x f(x_k, y_k^*) - \nabla_{xy}^2 g(x_k, y_k^*) \hat{v}_k$ of the outer-level function w.r.t. x (called hypergradient) by the form of $\widehat{\nabla} \Phi(x_k) = \nabla_x f(x_k, \hat{y}_k) - \nabla_{xy}^2 g(x_k, \hat{y}_k) \hat{v}_k$.

2.2 ITD-based Bilevel Optimization Algorithm

We present the single-loop ITD-based bilevel optimization algorithm (for simplicity, hereafter referred to as ITD) in Algorithm 2. Similar to AID, ITD also performs one step of gradient descent and employs a warm-start strategy on the inner-level function $g(x, y)$ to obtain $\hat{y}_k$. Unlike AID, however, ITD does not rely on the implicit gradient formula when estimating the hypergradient, but instead estimates the hypergradient directly via automatic differentiation. Since the update of $\hat{y}_k$ depends on x_k, ITD needs to store the iterative trajectory for backpropagation. In this work, because we consider the more practical single-step gradient descent, the hypergradient estimate takes the following form: $\widehat{\nabla} \Phi(x_k) = \nabla_x f(x_k, \hat{y}_k) - \alpha \nabla_{xy}^2 g(x_k, y_k^0) \nabla_y f(x_k, \hat{y}_k)$.

3 Definitions and Assumptions

In bilevel optimization, the objective is to minimize the hyper-objective function $\nabla \Phi(x)$, which is typically nonconvex. Because finding a global minimum for such functions can be computationally prohibitive (Nemirovski & IUdin, 1983), this work aims to find an approximate stationary point following the literature (Carmon et al., 2017; Ji et al., 2021).

Definition 1. *We call $\bar{x}$ is an ϵ-stationary point of problem (1) if $\|\nabla \Phi(\bar{x})\|^2 \leq \epsilon$.*

In this work, we focus on the problem (1) under the following standard assumptions, as also widely adopted by Ghadimi & Wang (2018); Ji et al. (2021). Let $z = (x, y)$ denote all parameters.

Assumption 1. *The inner-level function $g(x, y)$ is μ-strong-convex w.r.t. y.*

Assumption 2. *The function $f(z)$ is M-Lipschitz, i.e., for any z, z',*

$$|f(z) - f(z')| \leq M \|z - z'\|.$$

Assumption 3. *Gradients $\nabla f(z)$ and $\nabla g(z)$ are L-Lipschitz, i.e., for any z, z',*

$$\|\nabla f(z) - \nabla f(z')\| \leq L \|z - z'\|, \quad \|\nabla g(z) - \nabla g(z')\| \leq L \|z - z'\|.$$

Assumption 4. *Suppose the derivatives $\nabla^2_{xy} g(z)$ and $\nabla^2_y g(z)$ are ρ-Lipschitz, i.e., for any z, z',*

$$\left\|\nabla^2_{xy} g(z) - \nabla^2_{xy} g(z')\right\| \leq \rho \|z - z'\|, \quad \left\|\nabla^2_y g(z) - \nabla^2_y g(z')\right\| \leq \rho \|z - z'\|.$$

4 Main Results

In this section, we will provide the convergence analysis and characterize the overall computational complexity for both single-loop AID- and ITD-based algorithms.

4.1 Challenges in the Analysis and Our Approach

The conventional analytical path (Ji et al., 2021; 2022), which we term *Direct Squared Norm Analysis* (DSNA), relies on bounding the squared norm of the error vector at each iteration. Let's consider a simplified one-step error recurrence of the form $e_{k+1} = Ae_k + \delta_k$, where A represents the contraction operator and δ_k is the accumulated error term (e.g., from the inexact inner-loop solution). The standard approach proceeds by analyzing its squared norm: $\|e_{k+1}\|^2 = \|Ae_k + \delta_k\|^2 = \|Ae_k\|^2 + 2\langle Ae_k, \delta_k\rangle + \|\delta_k\|^2$. The primary challenge arises from the cross-term, $2\langle Ae_k, \delta_k\rangle$. To make this term tractable, existing analyses invariably resort to "pessimistic" inequalities, such as the Cauchy-Schwarz or Young's inequality (e.g., $2\langle a, b\rangle \leq \|a\|^2 + \|b\|^2$). For example, Ji et al. (2022) adopted this approach when analyzing the error upper bounds of the inner variable and the solution of the linear system. While this decouples the terms, it does so at a great cost. This step fundamentally ignores any potential underlying structure or cancellation effects between e_k and δ_k. The repeated application of such loose bounds over many iterations causes the dependencies on the problem's condition number, κ, to compound, ultimately leading to the inflated convergence rate. Our key insight is that this pessimistic rate is not an inherent property of the algorithm itself, but rather an analysis artifact stemming from the premature squaring of the norm. This step discards crucial information too early in the derivation.

We introduce a more delicate analytical strategy, *Decoupled Norm Analysis* (DNA), that sidesteps this bottleneck. Instead of immediately squaring the error recurrence, we first analyze the error norm in its linear form by applying the triangle inequality: $\|e_{k+1}\| = \|Ae_k + \delta_k\| \leq \|Ae_k\| + \|\delta_k\|$. By keeping the analysis in the linear domain of norms for as long as possible, we can establish a tighter recursive relationship (Lemmas 1 and 2 for AID, Lemma 5 for ITD). This approach allows for a more refined handling of the error terms, preserving more of the underlying geometric structure. The squaring operation is deferred to the very end of the analysis, after the full recurrence has been unrolled (Lemma 4 for AID, Lemma 7 for ITD). This seemingly simple change of order—analyzing the norm before squaring it—prevents the compounding of pessimistic estimates associated with the cross-term. It is this principled deviation from the standard analytical template that allows us to break the rate barrier and establish the significantly improved convergence rate, providing a more faithful theoretical picture of the algorithm's efficiency.

4.2 Convergence Analysis of AID

Proof Sketch: The proof for AID consists of three main steps: 1) Decomposing the hypergradient estimation error into the approximation error of the inner-level solution and the error from solving the linear system. (Lemma 3). 2) Bounding these two types of errors based on the errors in previous iterations (Lemmas 1 and 2). 3) Combining the results from the preceding steps to provide a convergence guarantee for the AID algorithm (Theorem 1).

Before presenting the convergence analysis on AID, we first give the following useful lemmas. Now we study the convergence of $\|\hat{v}_k - v_k^*\|$ and $\|\hat{y}_k - y_k^*\|$ for $k = 1, 2, \ldots, K$, where v_k^* is the exact solution of the linear system $\nabla_y^2 g(x_k, \hat{y}_k)v = \nabla_y f(x_k, \hat{y}_k)$. Note that the descent of the overall outer-level objectives also depends on the error of y_k. We next analyze these errors.

Lemma 1. *Consider single-loop AID-based algorithm in Algorithm 1. Suppose Assumptions 1-4 hold. Let $\alpha \leq \frac{1}{L}$, then we have*

$$\left\|y_k^0 - \hat{y}_k\right\| \leq \alpha L \left(\left\|\hat{y}_{k-1} - y_{k-1}^*\right\| + \|x_{k-1} - x_k\|\right), \tag{2}$$

$$\|\hat{y}_k - y_k^*\| \leq (1 - \mu\alpha)\left\|\hat{y}_{k-1} - y_{k-1}^*\right\| + \frac{L}{\mu}\|x_{k-1} - x_k\|. \tag{3}$$

Remark 1. *Lemma 1 demonstrates that: 1) for $k = 1, \ldots, K$, the error between the initial point and the iterated solution of the inner-level problem in single-loop AID can be bounded by the error from the previous iteration; 2) the error between the approximate solution and the exact solution of the inner-level problem in single-loop AID can also be bounded by the error from the previous iteration, which serves as a crucial foundation for the analysis of the algorithm's convergence.*

Then, we decompose $\|\hat{v}_k - v_k^*\|$ and then estimate the upper bound.

Lemma 2. *Consider single-loop AID-based algorithm in Algorithm 1. Suppose Assumptions 1-4 hold. Let $C_0 = \frac{\rho M}{\mu^2} + \frac{L}{\mu}$. Then, we have*

$$\|\hat{v}_k - v_k^*\| \leq \|\hat{v}_k - \tilde{v}_k^*\| + C_0\|\hat{y}_k - y_k^*\|, \tag{4}$$

$$\|\hat{v}_k - \tilde{v}_k^*\| \leq (1 - \mu\eta)\left\|\hat{v}_{k-1} - \tilde{v}_{k-1}^*\right\| + C_0\left(\left\|y_k^0 - \hat{y}_k\right\| + \|x_{k-1} - x_k\|\right), \tag{5}$$

where $\tilde{v}_k^* = (\nabla_y^2 g(x_k, \hat{y}_k))^{-1}\nabla_y f(x_k, \hat{y}_k)$.

Remark 2. *The purpose of Lemma 2 is to conduct a more detailed decomposition of the error between $\hat{v}_k$ and v_k^*, because this error originates from two aspects: 1) The use of $\hat{y}_k$ to approximate y_k^* in the inner-level problem. 2) The use of $\hat{v}_k$, obtained from solving the linear system $\nabla_y^2 g(x_k, \hat{y}_k)v = \nabla_y f(x_k, \hat{y}_k)$, to approximate v_k^*. Therefore, Lemma 2 decouples these two factors and controls them separately. Specifically, the first and second terms in Eq. (4) are only related to the precision of the linear equation solution and the inner-level problem solution, respectively. 3) Eq. (5) further expands the first term on the right-hand side of Eq. (4).*

In Lemmas 1 and 2, we have already provided the relevant error terms of y_k and v_k. Therefore, we will utilize the above results to analyze the error between the estimated hypergradient $\widehat{\nabla}\Phi(x_k)$ and the true hypergradient $\nabla\Phi(x_k)$.

Lemma 3. *Consider single-loop AID-based algorithm in Algorithm 1. Suppose Assumptions 1-4 hold. Define C_0 as in Lemma 2. Then we have*

$$\left\|\widehat{\nabla}\Phi(x_k) - \nabla\Phi(x_k)\right\| \leq \left(L + \frac{\rho M}{\mu} + C_0 L\right)\|\hat{y}_k - y_k^*\| + L\|\hat{v}_k - \tilde{v}_k^*\|. \tag{6}$$

Unlike the previous DSNA, our proposed DNA avoids the inflation of the condition number κ caused by repeated squaring. Combine Eq. (6) with the former lemmas, we can get the following lemma.

Lemma 4. *Consider single-loop AID-based algorithm in Algorithm 1. Suppose Assumptions 1-4 hold. Define C_0 as in Lemma 2. Let $\alpha = \eta = \frac{1}{L}$, $C_1 = \frac{4C_0 L}{\mu}$, $C_2 = \frac{\alpha L^2 C_0}{\mu} + \frac{\rho M}{\mu^2} + \frac{L}{\mu} + \frac{LC_1}{\mu}$ and $C_3 = L + \frac{\rho M}{\mu} + C_0 L$. Choose the outer stepsize β such that $\beta = \min\{\frac{C_1\mu\alpha}{4C_2C_3}, \frac{\eta\mu}{2LC_2}\}$. Then, we have*

$$\begin{aligned}\left\|\widehat{\nabla}\Phi(x_k) - \nabla\Phi(x_k)\right\|^2 \leq & L^2\left(1 - \frac{\mu}{4L}\right)^k \cdot \left(\left\|v_0^Q - \tilde{v}_0^*\right\| + C_1\left\|y_0^N - y_0^*\right\|\right)^2 \\ & + \frac{3\beta^2 C_2^2 L^3}{\mu}\sum_{t=0}^{k}\left(1 - \frac{\mu}{4L}\right)^{k-1-t}\|\nabla\Phi(x_t)\|^2.\end{aligned} \tag{7}$$

Remark 3. *Lemma 4 is a key result that supports the convergence analysis of single-loop AID-based algorithm. Compared to the work of (Ji et al., 2022), we relax the limit of the step-size for solving the linear system. Specifically, Ji et al. (2022) in their Corollary 2 required that $\eta = \mathcal{O}(\kappa^{-2})$, whereas we, through a more fine-grained analysis, set eta to $1/L$. This indirectly allows for a more aggressive choice of the outer-level step size β, thereby achieving a faster convergence rate.*

Based on the above conclusions, the following theorem provides a convergence analysis for single-loop AID-based algorithm.

Theorem 1. *Consider single-loop AID-based algorithm in Algorithm 1. Suppose Assumptions 1-4 hold. Choose parameters* $\alpha = \eta = \frac{1}{L}$. *Let* $L_\Phi = L + \frac{2L^2+\rho M^2}{\mu^2} + \frac{2\rho LM+L^3}{\mu^2} + \frac{\rho L^2 M}{\mu^3}$ *be the smoothness parameter of* $\Phi(\cdot)$. *Choose the outer stepsize* β *such that* $\beta = \min\{\frac{C_1\mu\alpha}{4C_2C_3}, \frac{\eta\mu}{2LC_2}\}$. *Then,* $\frac{1}{K}\sum_{k=0}^{K-1}\|\nabla\Phi(x_k)\|^2 = \mathcal{O}(\frac{\kappa^5}{K})$, *and the complexity is* $Gc(\epsilon) = \widetilde{\mathcal{O}}(\kappa^5\epsilon^{-1})$, $Mv(\epsilon) = \widetilde{\mathcal{O}}(\kappa^5\epsilon^{-1})$.

Remark 4. *Compared with the work of Ji et al. (2022), our core improvement lies in controlling the errors of both the inner solution* y *and the linear system solution* v, *where we relax the requirement on the outer objective learning rate* β *from* $\mathcal{O}(\kappa^{-6})$ *to* $\mathcal{O}(\kappa^{-5})$. *Consequently, we improve the convergence rate of single-loop AID-based algorithm from* $\mathcal{O}(\kappa^6/K)$ *to* $\mathcal{O}(\kappa^5/K)$. *This indicates that the convergence gap between such algorithms and the AID algorithms with multi-step gradient descent is not as large as the* $\mathcal{O}(\kappa^2)$ *gap shown by Ji et al. (2022), but rather a smaller* $\mathcal{O}(\kappa^1)$. *This also partially supports the practice that most bilevel optimization algorithms perform only one or a few inner updates.*

Theorem 2. [Simplified version of the upper bound in Ji et al. (2022)]. *Consider single-loop AID-based algorithm in Algorithm 1. Under the same setting of Theorem 1, we have* $\frac{1}{K}\sum_{k=0}^{K-1}\|\nabla\Phi(x_k)\|^2 = \mathcal{O}\big(\frac{\kappa^6}{K}\big)$.

4.3 Convergence Analysis of ITD

Proof Sketch: Unlike AID, the hypergradient estimation error of the single-loop ITD-based algorithm is introduced only by solving the inner problem. Therefore, our proof consists of three main steps: 1) Establishing the connection between the hypergradient estimation error and the approximation error of the inner-level solution (Lemma 6). 2) Bounding the approximation error of the solution to the inner-level problem (Lemma 5). 3) Combining the results from the previous steps to provide a convergence analysis for the ITD algorithm (Lemma 7 and Theorem 3).

To this end, we first present several useful lemmas, which will subsequently be used to prove Theorem 3.

Lemma 5. *Consider the single-loop ITD-based algorithm in Algorithm 2. Suppose Assumptions 1-4 hold. Let* $\alpha \le \frac{1}{L}$, $C_4 = L + \alpha L^2 + \alpha\rho M$, $C_5 = M(1-\alpha\mu)\frac{L}{\mu} + \alpha^2\rho M^2$, $C_6 = 1 - \mu\alpha + \frac{L\beta C_4}{\mu}$ *and* $C_7 = \frac{L\beta C_5}{\mu}$. *Then, we have*

$$\|\hat{y}_k - y^*(x_k)\| \le C_6\|\hat{y}_{k-1} - y^*(x_{k-1})\| + \frac{L\beta}{\mu}\|\nabla\Phi(x_{k-1})\| + C_7, \tag{8}$$

$$\|\hat{y}_k - y^*(x_k)\| \le \Big(1 - \frac{\mu}{2L}\Big)^k\|\hat{y}_0 - y^*(x_0)\| + \frac{L\beta^2}{\mu}\sum_{j=0}^{k-1}\Big(1 - \frac{\mu}{2L}\Big)^{k-1-j}\left(\|\nabla\Phi(x_j)\| + C_5\right). \tag{9}$$

Using the error bound for $\|\hat{y}_k - y_k^*\|$, we will analyze the error between the estimated hypergradient $\widehat{\nabla}\Phi(x_k)$ and the true hypergradient $\nabla\Phi(x_k)$ of the ITD algorithm in the following lemma.

Lemma 6. *Consider the single-loop ITD-based algorithm in Algorithm 2. Suppose Assumptions 1-4 hold. Define* C_4 *and* C_5 *in Lemma 5. Let* $\alpha \le \frac{1}{L}$, *we have*

$$\Big\|\widehat{\nabla}\Phi(x_k) - \nabla\Phi(x_k)\Big\| \le C_4\|\hat{y}_k - y_k^*\| + C_5. \tag{10}$$

Remark 5. *Lemma 6 shows that the error between the true hypergradient and the estimated hypergradient is controlled by the accuracy of the inner-level problem solution and an inherent error, part of which arises from* $\big\|y_k^0 - \hat{y}_k\big\|$. *This indicates that this non-vanishing convergence error is related to the refinement of the inner-level problem solution, and that the single-loop method is insufficient to bridge this gap.*

Lemma 7. *Consider the single-loop ITD-based algorithm in Algorithm 2. Suppose Assumptions 1-4 hold. Define C_4 and C_5 in Lemma 5. Let $\alpha \leq \frac{1}{L}$ and $\beta \leq \frac{\mu^3}{2L(2L^2+\rho M)}$. Then we have*

$$\left\|\widehat{\nabla}\Phi(x_k) - \nabla\Phi(x_k)\right\|^2 \leq C_4^2\left(1-\frac{\mu}{4L}\right)^k \|\hat{y}_0 - y^*(x_0)\|^2 + \frac{3L\beta^2 C_4^2}{\mu}\sum_{j=0}^{k-1}\left(1-\frac{\mu}{4L}\right)^{k-1-j}\left(\|\nabla\Phi(x_j)\| + C_5\right)^2 + 3C_5^2.$$

Based on the above results, the following theorem provides a convergence analysis for single-loop ITD-based algorithm.

Theorem 3. *Consider the single-loop ITD-based algorithm in Algorithm 2. Suppose Assumptions 1-4 hold. Choose parameters $\alpha = \eta = \frac{1}{L}$. Let $L_\Phi = L + \frac{2L^2+\rho M^2}{\mu^2} + \frac{2\rho LM+L^3}{\mu^2} + \frac{\rho L^2 M}{\mu^3}$ be the smoothness parameter of $\Phi(\cdot)$. Choose the outer stepsize β such that $\beta \leq \frac{\mu^3}{2L(2L^2+\rho M)}$. Then, $\frac{1}{K}\sum_{k=0}^{K-1}\|\nabla\Phi(x_k)\|^2 = \mathcal{O}\left(\frac{\kappa^3}{K} + \kappa^2\right)$.*

Remark 6. *Theorem 3 demonstrates that for the single-loop ITD-based algorithm, the convergence bound contains a non-vanishing error of order $\mathcal{O}(\kappa^2)$. Under the standard Assumptions 1-4, such an error is unavoidable. Moreover, this error upper bound of order $\mathcal{O}(\kappa^2)$ matches the error lower bound (Theorem 4), which indicates that we have achieved a tighter error upper bound through more refined analysis. This resolves the issue in Ji et al. (2022) where there exists a gap of $\alpha\mu$ between the upper and lower bounds.*

Theorem 4. [Simplified version of the lower bound in Ji et al. (2022)]. *Consider the single-loop ITD-based algorithm in Algorithm 2. Suppose Assumptions 1-4 hold. Let $\alpha \leq \frac{1}{L}$, $\beta \leq \frac{1}{L_\Phi}$ and $L_\Phi = L + \frac{2L^2+\rho M^2}{\mu^2} + \frac{2\rho LM+L^3}{\mu^2} + \frac{\rho L^2 M}{\mu^3}$. Then, we have $\|\nabla\Phi(x_K)\|^2 \geq \Theta(\kappa^2)$.*

5 Experiments

Experimental setup. We consider the following bilevel optimization problem:

$$f(x,y) = \frac{1}{2}x^T Z_x x + \frac{1}{10}\mathbf{1}^T y, \qquad g(x,y) = \frac{1}{2}y^T Z_y y - Lx^T y + \mathbf{1}^T y,$$

where $x, y \in \mathbb{R}^2$ and $Z_x = Z_y = \begin{bmatrix} L & 0 \\ 0 & \mu \end{bmatrix}$. Thus the optimal solution of the inner-level subproblem and the exact hypergradient have the following form:

$$y^*(x) = Z_y^{-1}(Lx - \mathbf{1}), \qquad \nabla\Phi(x) = Z_x x + LZ_y^{-1}\mathbf{1}. \tag{11}$$

Based on the updates of single-loop ITD-based method, we have $\hat{y}_k = y_k^0 - \alpha(Z_y y_k^0 - Lx_k + \mathbf{1})$. Let the hyperparameters set as $\mu = 0.1$, $M = 0.1$, $\rho = 0.1$, $K = 10000$ and $\alpha = 1/L$.

Results of AID-based Algorithm. Figure 1 presents the error curves of the single-loop AID-based Algorithm. In Figure 1 (Left), we compare the error upper bound derived by Thoerem 1 with that given by Ji et al. (2022) under different condition numbers κ. It can be observed that, under varying condition numbers, our upper bound curve consistently lies closer above the $\|\nabla\Phi(x_k)\|^2$ curve. This is achieved by refining the analysis and reducing the theoretical order of the upper bound from $\mathcal{O}(\kappa^6)$ to $\mathcal{O}(\kappa^5)$. In Figure 1 (Right), under the condition number $\kappa = 2$, we compare the variation of the error upper bound with respect to the number of outer iterations K. It can be seen that the $\|\nabla\Phi(x_k)\|^2$ curve keeps decreasing as the number of iterations increases, which indicates that the single-loop AID-based algorithm converges as K grows, thereby confirming the correctness of Theorem 1. Moreover, we observe that our upper bound curve consistently outperforms that of Ji et al. (2022), which demonstrates that, theoretically, we provide a tighter error upper bound for this algorithm, thus verifying the correctness and effectiveness of our theoretical results.

Results of ITD-based Algorithm. Figure 2 illustrates the performance of the ITD-based algorithm. From Figure 2 (Left), we first observe that in Ji et al. (2022), the gap between the reported upper and

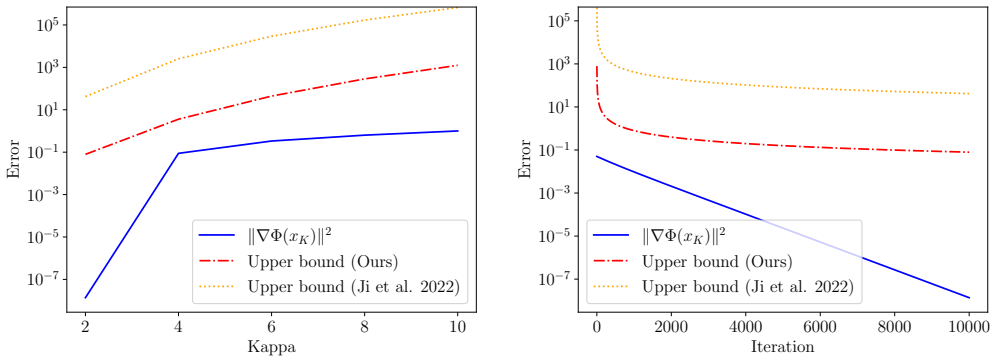

Figure 1: Comparison of error curves of the single-loop AID-based Algorithm. **Left:** Curves of various error terms (the squared norm of the true hypergradient $\|\nabla\Phi(x_k)\|^2$, the upper bound provided in Theorem 1 by us, and the upper bound provided in Theorem 2 by Ji et al. (2022)) with respect to different condition numbers κ. **Right:** Curves of various error terms with respect to the number of iterations K when the condition number $\kappa = 2$.

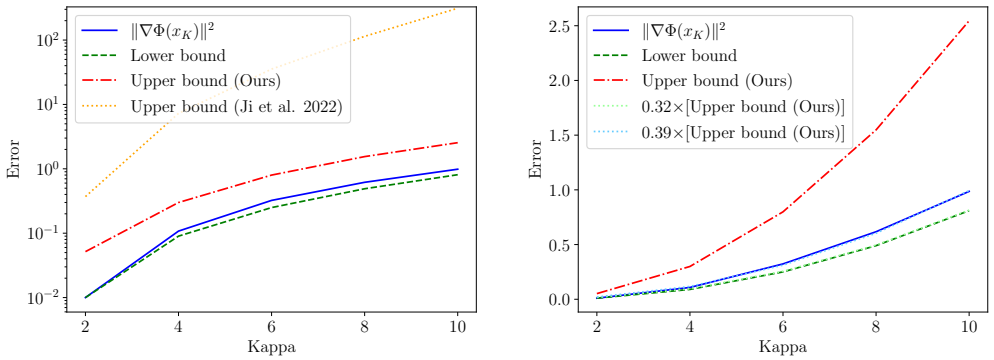

Figure 2: Comparison of error curves of the single-loop ITD-based Algorithm. **Left:** Curves of various error terms (the squared norm of the true hypergradient $\|\nabla\Phi(x_k)\|^2$, the upper bound provided in Theorem 3 by us, and the upper bound provided by Ji et al. (2022), the lower bound provided in Theorem 4) with respect to different condition numbers κ. **Right:** Curves of the scaled upper bound ($\times 0.32$ and $\times 0.39$) with respect to different condition numbers κ.

lower bounds remains large, confirming their conclusion that both bounds still differ by an error of order $\alpha\mu$. In contrast, our theoretical upper bound is substantially tighter: it lies much closer to the empirical $\|\nabla\Phi(x_K)\|^2$ curve while remaining strictly above it. This demonstrates that our bound provides a sharper characterization of the true convergence behavior.

To further verify the validity of our theoretical results, in addition to the curve of the true hypergradient norm, the upper bound curve (according to Theorem 3), and the lower bound curve, we also scale the upper bound curve in Figure 2 (Right). Specifically, we multiply it by 0.32 and 0.39, respectively. The results show that, after scaling the upper bound curve with different factors, its error values almost coincide with the true hypergradient norm curve and the lower bound curve, respectively. This indicates that the difference between the upper bound and the lower bound arises from constant factors introduced by scaling, rather than from differences in order. Thus, this supports the conclusion of Theorem 3, namely that we have reduced the inherent error to $\mathcal{O}(\kappa^2)$.

6 Related Work

Hypergradient-based bilevel optimization. A variety of hypergradient-based bilevel algorithms have been proposed, differing mainly in how they estimate hypergradients. Methods based on approximate implicit differentiation (AID) (Domke, 2012; Pedregosa, 2016; Ghadimi & Wang, 2018;

Grazzi et al., 2020; Ji et al., 2021) estimate the product of the inverse hessian and a vector by solving linear systems with efficient iterative solvers. In contrast, iterative differentiation (ITD) methods (Maclaurin et al., 2015; Franceschi et al., 2017; Shaban et al., 2019; Liu et al., 2021) compute hypergradients by backpropagating through the inner optimization trajectory. The convergence properties of AID- and ITD-based algorithms have been the subject of extensive study. For example, Ghadimi & Wang (2018) and Ji et al. (2021) analyzed the convergence rates and complexities of both approaches, while Ji et al. (2022) provided a unified framework covering different inner-loop choices and established lower bounds on the inherent error of ITD. Despite this progress, a notable gap remains between the convergence rate of the single-loop and multi-loop algorithms. Motivated by this gap, our work develops sharper convergence guarantees for single-loop methods, which are widely used in practice. Compared with Ji et al. (2022), our analysis for AID achieves an improved convergence order, while for ITD we refine the upper bound on the inherent error to match its known lower bound.

Gradient-based bilevel optimization. In recent years, some first-order gradient-based bilevel optimization methods have also attracted attention. Chen et al. (2025) proposed an algorithm that achieves near-optimal complexity under the nonconvex–strongly convex setting; however, they still require a relatively large number of inner iterations, $O(\kappa \log(\lambda\kappa))$, where $\lambda = O(\kappa^3)$ denotes the penalty strength, which is also large. This, to some extent, affects practical applicability. In addition, Liu et al. (2024) proposed MEHA based on Moreau-envelope, where they considered the single-loop setting and provided a convergence rate of $O(1/K^{1/2-p} + 1/K^p)$, with $p \in (0, 1/2)$. Kwon et al. (2023), by introducing momentum, designed F^{3}SA, which is also a single-loop method and achieves a convergence rate of $O(K^{-2/3})$. However, compared with hypergradient-based methods, its convergence rate is relatively slower. Therefore, this paper focuses on providing a sharper analysis for hypergradient-based methods. From a technical perspective, DNA has the potential to be applied to such gradient-based methods (Chen et al., 2022; Hong et al., 2023; Liu et al., 2024; Fang et al., 2025), which we leave for future work.

The single-loop bilevel optimization algorithms. The single-loop methods have shown potential in many applications. In few-shot meta-learning, MAML (Finn et al., 2017), as a classic method, performs single-step gradient descent on the support set for multiple tasks in the inner-level, retaining the iteration path, while the outer-level updates the network's initial values using the query set. In hyperparameter optimization, sample reweighting is a widely used application of bilevel optimization algorithms (Ren et al., 2018; Shu et al., 2019; Wang et al., 2024), as bilevel optimization can efficiently assign different weights to each sample. Such methods typically use the training set in the inner-level to perform single-step gradient descent to optimize model parameters, and the validation set in the outer loop to optimize sample weights or weighted networks. In neural architecture search, DARTS (Liu et al., 2018) method uses a one-step update in the inner-level to update the model, and the outer-level optimizes the architecture using validation data. It is worth noting that most of these algorithms achieve efficiency by single-loop, which is also crucial for the large-scale practice of bilevel optimization techniques (Choe et al., 2023; Shen et al., 2024). Therefore, in this work, we focus on the single-loop bilevel optimization algorithms, consistent with practical applications, and are committed to establishing sharper convergence guarantees for these algorithms.

7 Conclusion

In this work, we advance the theoretical understanding of single-loop bilevel optimization algorithms, a setting of growing practical relevance. For the AID method, our refined analysis improves the convergence rate to $\mathcal{O}(\kappa^5/K)$, narrowing the gap with multi-loop approaches. For the ITD method, we establish that its convergence error is exactly $\mathcal{O}(\kappa^2)$, thereby closing the open question raised in prior work regarding its tightness. Our experimental results can corroborate the theory, demonstrating that single-loop methods can achieve both efficiency and favorable convergence behavior. These findings not only bridge an important gap between theory and practice, but also potentially suggest that the single-loop bilevel optimization methods can be strong candidates for large-scale machine learning tasks. Beyond the specific result for the algorithm, we believe our proposed analytical paradigm of the decoupling norm analysis opens new path for studying other bilevel optimization algorithms, potentially tightening bounds for methods where previous analyses have been overly pessimistic. Future work includes extending our refined analyses to nonconvex inner problems and hessian-free methods.

Reproducibility Statement. All results are theoretical, and complete proofs are provided in the appendix with clear assumptions and detailed derivations. This ensures that all claims can be independently verified without reliance on external data.

REFERENCES

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points i. *arXiv preprint arXiv:1710.11606*, 2017.

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

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

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

Yutian Chen, Matthew W Hoffman, Sergio Gómez Colmenarejo, Misha Denil, Timothy P Lillicrap, Matt Botvinick, and Nando Freitas. Learning to learn without gradient descent by gradient descent. In *International Conference on Machine Learning*, pp. 748–756. PMLR, 2017.

Sang Choe, Sanket Vaibhav Mehta, Hwijeen Ahn, Willie Neiswanger, Pengtao Xie, Emma Strubell, and Eric Xing. Making scalable meta learning practical. *Advances in neural information processing systems*, 36:26271–26290, 2023.

Justin Domke. Generic methods for optimization-based modeling. In *Artificial Intelligence and Statistics*, pp. 318–326. PMLR, 2012.

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

Sheng Fang, Yong-Jin Liu, Wei Yao, Chengming Yu, and Jin Zhang. qnbo: quasi-newton meets bilevel optimization. *arXiv preprint arXiv:2502.01076*, 2025.

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

Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. Forward and reverse gradient-based hyperparameter optimization. In *International conference on machine learning*, pp. 1165–1173. PMLR, 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*, pp. 1568–1577. PMLR, 2018.

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

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

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*, pp. 11993–12002, 2020.

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

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

Kaiyi Ji, Mingrui Liu, Yingbin Liang, and Lei Ying. Will bilevel optimizers benefit from loops. *Advances in Neural Information Processing Systems*, 35:3011–3023, 2022.

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

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

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

Risheng Liu, Zhu Liu, Wei Yao, Shangzhi Zeng, and Jin Zhang. Moreau envelope for nonconvex bilevel optimization: A single-loop and hessian-free solution strategy. In *International Conference on Machine Learning*, pp. 31566–31596. PMLR, 2024.

Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *International conference on machine learning*, pp. 2113–2122. PMLR, 2015.

A.S. Nemirovski and D.B. IUdin. *Problem Complexity and Method Efficiency in Optimization*. A Wiley-Interscience publication. Wiley, 1983. ISBN 9780471103455. URL https://books.google.co.jp/books?id=6ULvAAAAMAAJ.

Fabian Pedregosa. Hyperparameter optimization with approximate gradient. In *International conference on machine learning*, pp. 737–746. PMLR, 2016.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pp. 4334–4343. PMLR, 2018.

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

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

Han Shen, Zhuoran Yang, and Tianyi Chen. Principled penalty-based methods for bilevel reinforcement learning and rlhf. *Journal of Machine Learning Research*, 26(114):1–49, 2025.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32, 2019.

Lingxiao Wang, Qi Cai, Zhuoran Yang, and Zhaoran Wang. On the global optimality of model-agnostic meta-learning. In *International conference on machine learning*, pp. 9837–9846. PMLR, 2020.

Quanziang Wang, Renzhen Wang, Yuexiang Li, Dong Wei, Hong Wang, Kai Ma, Yefeng Zheng, and Deyu Meng. Relational experience replay: Continual learning by adaptively tuning task-wise relationship. *IEEE Transactions on Multimedia*, 26:9683–9698, 2024.

Haifeng Zhang, Weizhe Chen, Zeren Huang, Minne Li, Yaodong Yang, Weinan Zhang, and Jun Wang. Bi-level actor-critic for multi-agent coordination. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7325–7332, 2020.

A PROOF OF THE SINGLE-LOOP AID-BASED ALGORITHM

A.1 PROOF OF LEMMA 1

Proof. By the update rule of y_k, we have for each $k = 1, \ldots, K$,

$$\begin{aligned}\left\|y_k^0 - \hat{y}_k\right\| =& \alpha \left\|\nabla_y g(x_k, y_k^0)\right\| = \alpha \left\|\nabla_y g(x_k, \hat{y}_{k-1})\right\| \\ =& \alpha \left\|\nabla_y g(x_k, \hat{y}_{k-1}) - \nabla_y g(x_k, y_{k-1}^*) + \nabla_y g(x_k, y_{k-1}^*) - \nabla_y g(x_{k-1}, y_{k-1}^*)\right\| \\ \leq& \alpha L \left(\left\|\hat{y}_{k-1} - y_{k-1}^*\right\| + \left\|x_{k-1} - x_k\right\|\right).\end{aligned}$$

The second conclusion holds that

$$\begin{aligned}\left\|\hat{y}_k - y_k^*\right\| \leq& (1 - \mu\alpha) \left\|y_k^0 - y_k^*\right\| \leq (1 - \mu\alpha) \left\|\hat{y}_{k-1} - y_{k-1}^*\right\| + \left\|y_{k-1}^* - y_k^*\right\| \\ \overset{\text{(i)}}{\leq}& (1 - \mu\alpha) \left\|\hat{y}_{k-1} - y_{k-1}^*\right\| + \frac{L}{\mu} \left\|x_{k-1} - x_k\right\|,\end{aligned}$$

where (i) follows from Lemma 2.2 in Ghadimi & Wang (2018). □

A.2 PROOF OF LEMMA 2

In the following two proofs, we will respectively present the two conclusions (Eq. (4) and Eq. (5)) in Lemma 2.

Proof. According to the triangle inequality, we have $\|\hat{v}_k - v_k^*\| \leq \|\hat{v}_k - \tilde{v}_k^*\| + \|\tilde{v}_k^* - v_k^*\|$ for $k = 1, 2, \ldots, K$. Then we focus on using $\|\hat{y}_k - y_k^*\|$ to bound $\|\tilde{v}_k^* - v_k^*\|$:

$$\begin{aligned}\left\|\tilde{v}_k^* - v_k^*\right\| =& \left\|[\nabla_y^2 g(x_k, \hat{y}_k)]^{-1} \nabla_y f(x_k, \hat{y}_k) - [\nabla_y^2 g(x_k, y_k^*)]^{-1} \nabla_y f(x_k, y_k^*)\right\| \\ \leq& \left\|[\nabla_y^2 g(x_k, \hat{y}_k)]^{-1} \nabla_y f(x_k, \hat{y}_k) - [\nabla_y^2 g(x_k, y_k^*)]^{-1} \nabla_y f(x_k, \hat{y}_k)\right\| \\ &+ \left\|[\nabla_y^2 g(x_k, y_k^*)]^{-1} \nabla_y f(x_k, \hat{y}_k) - [\nabla_y^2 g(x_k, y_k^*)]^{-1} \nabla_y f(x_k, y_k^*)\right\| \\ \leq& \left\|[\nabla_y^2 g(x_k, \hat{y}_k)]^{-1} - [\nabla_y^2 g(x_k, y_k^*)]^{-1}\right\| \cdot \left\|\nabla_y f(x_k, \hat{y}_k)\right\| \\ &+ \left\|[\nabla_y^2 g(x_k, y_k^*)]^{-1}\right\| \cdot \left\|\nabla_y f(x_k, \hat{y}_k) - \nabla_y f(x_k, y_k^*)\right\| \\ \leq& \frac{\rho M \left\|\hat{y}_k - y_k^*\right\|}{\mu^2} + \frac{L}{\mu} \left\|\hat{y}_k - y_k^*\right\| = \left(\frac{\rho M}{\mu^2} + \frac{L}{\mu}\right) \left\|\hat{y}_k - y_k^*\right\|.\end{aligned}$$

Then, we can get the conclusion of Eq. (4). □

Proof. By the updated rule, we can obtain that

$$\left\|\hat{v}_k - \tilde{v}_k^*\right\| \leq (1 - \mu\eta) \left\|v_k^0 - \tilde{v}_k^*\right\| \leq (1 - \mu\eta) \left\|\hat{v}_{k-1} - \tilde{v}_{k-1}^*\right\| + \left\|\tilde{v}_{k-1}^* - \tilde{v}_k^*\right\|.$$

For the second term $\left\|\tilde{v}_{k-1}^* - \tilde{v}_k^*\right\|$, we have

$$\begin{aligned}\left\|\tilde{v}_{k-1}^* - \tilde{v}_k^*\right\| =& \left\|[\nabla_y^2 g(x_{k-1}, \hat{y}_{k-1})]^{-1} \nabla_y f(x_{k-1}, \hat{y}_{k-1}) - [\nabla_y^2 g(x_k, \hat{y}_k)]^{-1} \nabla_y f(x_k, \hat{y}_k)\right\| \\ \leq& \left\|[\nabla_y^2 g(x_{k-1}, \hat{y}_{k-1})]^{-1} \nabla_y f(x_{k-1}, \hat{y}_{k-1}) - [\nabla_y^2 g(x_k, \hat{y}_k)]^{-1} \nabla_y f(x_{k-1}, \hat{y}_{k-1})\right\| \\ &+ \left\|[\nabla_y^2 g(x_k, \hat{y}_k)]^{-1} \nabla_y f(x_{k-1}, \hat{y}_{k-1}) - [\nabla_y^2 g(x_k, \hat{y}_k)]^{-1} \nabla_y f(x_k, \hat{y}_k)\right\| \\ \leq& \left\|[\nabla_y^2 g(x_{k-1}, \hat{y}_{k-1})]^{-1} - [\nabla_y^2 g(x_k, \hat{y}_k)]^{-1}\right\| \cdot \left\|\nabla_y f(x_{k-1}, \hat{y}_{k-1})\right\| \\ &+ \left\|[\nabla_y^2 g(x_k, \hat{y}_k)]^{-1}\right\| \left\|\nabla_y f(x_{k-1}, \hat{y}_{k-1}) - \nabla_y f(x_k, \hat{y}_k)\right\|.\end{aligned}$$

Furthermore,

$$\begin{aligned}&\left\|\nabla_y f(x_{k-1}, \hat{y}_{k-1}) - \nabla_y f(x_k, \hat{y}_k)\right\| \\ \leq& \left\|\nabla_y f(x_{k-1}, \hat{y}_{k-1}) - \nabla_y f(x_k, y_k^0)\right\| + \left\|\nabla_y f(x_k, y_k^0) - \nabla_y f(x_k, \hat{y}_k)\right\| \\ \leq& L \left\|x_{k-1} - x_k\right\| + L \left\|y_k^0 - \hat{y}_k\right\|.\end{aligned}$$

Then, we have

$$\begin{aligned}
&\left\|[\nabla_y^2 g(x_{k-1},\hat{y}_{k-1})]^{-1}-[\nabla_y^2 g(x_k,\hat{y}_k)]^{-1}\right\|\cdot\|\nabla_y f(x_{k-1},\hat{y}_{k-1})\| \\
\leq&\left\|[\nabla_y^2 g(x_{k-1},\hat{y}_{k-1})]^{-1}\right\|\left\|\nabla_y^2 g(x_{k-1},\hat{y}_{k-1})-\nabla_y^2 g(x_k,\hat{y}_k)\right\|\left\|[\nabla_y^2 g(x_k,\hat{y}_k)]^{-1}\right\| \\
&\cdot\|\nabla_y f(x_{k-1},\hat{y}_{k-1})\| \\
\leq&\frac{\rho\left(\|\hat{y}_{k-1}-\hat{y}_k\|+\|x_{k-1}-x_k\|\right)}{\mu^2}\|\nabla_y f(x_{k-1},\hat{y}_{k-1})\| \\
\leq&\frac{\rho M}{\mu^2}\left(\|\hat{y}_{k-1}-\hat{y}_k\|+\|x_{k-1}-x_k\|\right).
\end{aligned}$$

Thus, we can obtain that

$$\begin{aligned}
\left\|\tilde{v}_{k-1}^*-\tilde{v}_k^*\right\|\leq&\frac{\rho M\left(\|\hat{y}_{k-1}-\hat{y}_k\|+\|x_{k-1}-x_k\|\right)}{\mu^2}+\frac{L\|x_{k-1}-x_k\|+L\left\|y_k^0-\hat{y}_k\right\|}{\mu} \\
=&\left(\frac{\rho M}{\mu^2}+\frac{L}{\mu}\right)\left\|y_k^0-\hat{y}_k\right\|+\left(\frac{\rho M}{\mu^2}+\frac{L}{\mu}\right)\|x_{k-1}-x_k\|.
\end{aligned}$$

Then, we can get the conclusion of Eq. (5). □

A.3 Proof of Lemma 3

Proof. According to the definition of the hypergradient, we have

$$\begin{aligned}
\left\|\widehat{\nabla}\Phi(x_k)-\nabla\Phi(x_k)\right\|=&\left\|\nabla_x f(x_k,\hat{y}_k)-\nabla_{xy}^2 g(x_k,\hat{y}_k)\hat{v}_k-\nabla_x f(x_k,y_k^*)+\nabla_{xy}^2 g(x_k,y_k^*)v_k^*\right\| \\
\leq&\|\nabla_x f(x_k,y_k^*)-\nabla_x f(x_k,\hat{y}_k)\|+\left\|\nabla_{xy}^2 g(x_k,\hat{y}_k)(v_k^*-\hat{v}_k)\right\| \\
&+\left\|\left(\nabla_{xy}^2 g(x_k,y_k^*)-\nabla_{xy}^2 g(x_k,\hat{y}_k)\right)v_k^*\right\| \\
\leq&\left(L+\frac{\rho M}{\mu}\right)\|\hat{y}_k-y_k^*\|+L\|\hat{v}_k-v_k^*\| \\
\overset{Eq.\ (4)}{\leq}&\left(L+\frac{\rho M}{\mu}+C_0 L\right)\|\hat{y}_k-y_k^*\|+L\|\hat{v}_k-\tilde{v}_k^*\|.
\end{aligned}$$

Then, the proof is compeleted. □

A.4 Proof of Lemma 4

Proof. Firstly, we have

$$\begin{aligned}
\|\hat{v}_k-\tilde{v}_k^*\|\leq&(1-\mu\eta)\left\|\hat{v}_{k-1}-\tilde{v}_{k-1}^*\right\|+C_0\left\|y_k^0-\hat{y}_k\right\|+\left(\frac{\rho M}{\mu^2}+\frac{L}{\mu}\right)\|x_{k-1}-x_k\| \\
\overset{Eq.\ (2)}{\leq}&(1-\mu\eta)\left\|\hat{v}_{k-1}-\tilde{v}_{k-1}^*\right\|+C_0\alpha L\left\|\hat{y}_{k-1}-y_{k-1}^*\right\| \\
&+\left(\frac{\alpha L^2 C_0}{\mu}+\frac{\rho M}{\mu^2}+\frac{L}{\mu}\right)\|x_{k-1}-x_k\|.
\end{aligned}$$

Then we have

$$\begin{aligned}
&\|\hat{v}_k-\tilde{v}_k^*\|+C_1\|\hat{y}_k-y_k^*\| \\
\leq&(1-\mu\eta)\left\|\hat{v}_{k-1}-\tilde{v}_{k-1}^*\right\|+C_0\alpha L\left\|\hat{y}_{k-1}-y_{k-1}^*\right\|+\left(\frac{\alpha L^2 C_0}{\mu}+\frac{\rho M}{\mu^2}+\frac{L}{\mu}\right)\|x_{k-1}-x_k\| \\
&+(1-\mu\alpha)C_1\left\|\hat{y}_{k-1}-y_{k-1}^*\right\|+\frac{LC_1}{\mu}\|x_{k-1}-x_k\| \\
=&(1-\mu\eta)\left\|\hat{v}_{k-1}-\tilde{v}_{k-1}^*\right\|+\left(1-\mu\alpha+\frac{C_0\alpha L}{C_1}\right)\cdot C_1\left\|\hat{y}_{k-1}-y_{k-1}^*\right\| \\
&+\left(\frac{\alpha L^2 C_0}{\mu}+\frac{\rho M}{\mu^2}+\frac{L}{\mu}+\frac{LC_1}{\mu}\right)\|x_{k-1}-x_k\|.
\end{aligned}$$

By the update rule of $\{x_k\}$, we can obtain that

$$\begin{aligned}\|x_{k-1}-x_k\| =&\beta\left\|\widehat{\nabla}\Phi(x_{k-1})\right\| \leq \beta\|\nabla\Phi(x_{k-1})\| + \beta\left\|\widehat{\nabla}\Phi(x_{k-1})-\nabla\Phi(x_{k-1})\right\| \\ \overset{Eq.\ (6)}{\leq}& \beta\|\nabla\Phi(x_{k-1})\| + \beta\left(L+\frac{\rho M}{\mu}+C_0L\right)\left\|\hat{y}_{k-1}-y^*_{k-1}\right\| + \beta L\left\|\hat{v}_{k-1}-\tilde{v}^*_{k-1}\right\|.\end{aligned}$$

Thus, we have

$$\begin{aligned}&\|\hat{v}_k-\tilde{v}^*_k\| + C_1\|\hat{y}_k-y^*_k\| \\ \leq&\left(1-\mu\eta+\beta L\left(\frac{\alpha L^2C_0}{\mu}+\frac{\rho M}{\mu^2}+\frac{L}{\mu}+\frac{LC_1}{\mu}\right)\right)\left\|\hat{v}_{k-1}-\tilde{v}^*_{k-1}\right\| \\ &+\left(1-\mu\alpha+\frac{C_0\alpha L}{C_1}+\frac{\beta}{C_1}\left(L+\frac{\rho M}{\mu}+C_0L\right)\left(\frac{\alpha L^2C_0}{\mu}+\frac{\rho M}{\mu^2}+\frac{L}{\mu}+\frac{LC_1}{\mu}\right)\right) \\ &\cdot C_1\left\|\hat{y}_{k-1}-y^*_{k-1}\right\| + \beta\left(\frac{\alpha L^2C_0}{\mu}+\frac{\rho M}{\mu^2}+\frac{L}{\mu}+\frac{LC_1}{\mu}\right)\|\nabla\Phi(x_{k-1})\|.\end{aligned}$$

We denote that $C_2=\frac{\alpha L^2C_0}{\mu}+\frac{\rho M}{\mu^2}+\frac{L}{\mu}+\frac{LC_1}{\mu}$ and $C_3=L+\frac{\rho M}{\mu}+C_0L$. Then the above equation can rewrite as follows

$$\begin{aligned}&\|\hat{v}_k-\tilde{v}^*_k\| + C_1\|\hat{y}_k-y^*_k\| \\ \leq&(1-\mu\eta+\beta LC_2)\left\|\hat{v}_{k-1}-\tilde{v}^*_{k-1}\right\| + \left(1-\mu\alpha+\frac{C_0\alpha L}{C_1}+\frac{\beta C_2C_3}{C_1}\right)\cdot C_1\left\|\hat{y}_{k-1}-y^*_{k-1}\right\| \\ &+\beta C_2\|\nabla\Phi(x_{k-1})\|.\end{aligned}$$

We only need to $1-\mu\eta+\beta LC_2\leq 1-\frac{\mu\eta}{2}$, $\frac{C_0\alpha L}{C_1}=\frac{\mu\alpha}{4}$ and $\frac{\beta C_2C_3}{C_1}\leq\frac{\mu\alpha}{4}$. Then we can get

$$\begin{aligned}&\beta\leq\frac{\eta\mu}{2LC_2},\quad \beta\leq\frac{C_1\mu\alpha}{4C_2C_3},\quad C_1=\frac{4C_0L}{\mu}, \\ &C_2=\left(\frac{\alpha L^2C_0}{\mu}+\frac{\rho M}{\mu^2}+\frac{L}{\mu}+\frac{LC_1}{\mu}\right)\overset{\alpha=\frac{1}{L}}{=}\frac{4L^3}{\mu^3}+\frac{4L^2\rho M}{\mu^4}+\frac{L^2}{\mu^2}+\frac{L\rho M}{\mu^3}+\frac{L}{\mu}+\frac{\rho M}{\mu^2}, \\ &C_3=L+\frac{\rho M}{\mu}+C_0L=L+\frac{\rho M}{\mu}+\frac{\rho ML}{\mu^2}+\frac{L^2}{\mu}.\end{aligned}$$

Then, we have

$$\begin{aligned}\beta\leq\frac{C_1\mu\alpha}{4C_2C_3}=&\frac{\mu^4(\rho M+L\mu)}{(4L^3\mu+4L^2\rho M+2L^2\mu^2+2L\mu\rho M+L\mu^3)(L\mu^2+\rho M\mu+\rho ML+L^2\mu)} \\ =&\mathcal{O}(\kappa^{-4}),\end{aligned}$$

$$\beta\leq\frac{\eta\mu}{2LC_2}\overset{\eta=\frac{1}{L}}{=}\frac{\mu^5}{2L^2(4L^3\mu+4L^2\rho M+L^2\mu^2+L\mu\rho M+L\mu^3)}=\mathcal{O}(\kappa^{-5}).$$

Then, we have $\beta\leq\min\{\mathcal{O}(\kappa^{-4}),\mathcal{O}(\kappa^{-5})\}=\mathcal{O}(\kappa^{-5})$. Thus, we have

$$\begin{aligned}&\|\hat{v}_k-\tilde{v}^*_k\| + C_1\|\hat{y}_k-y^*_k\| \\ \leq&\max\{1-\frac{\mu\eta}{2},1-\frac{\alpha\mu}{2}\}\cdot\left(\left\|\hat{v}_{k-1}-\tilde{v}^*_{k-1}\right\| + C_1\left\|\hat{y}_{k-1}-y^*_{k-1}\right\|\right)+\beta C_2\|\nabla\Phi(x_{k-1})\| \\ =&\left(1-\frac{\mu}{2L}\right)\cdot\left(\left\|\hat{v}_{k-1}-\tilde{v}^*_{k-1}\right\| + C_1\left\|\hat{y}_{k-1}-y^*_{k-1}\right\|\right)+\beta C_2\|\nabla\Phi(x_{k-1})\|\end{aligned}$$

Accordingly, we have

$$\begin{aligned}\left(\|\hat{v}_k-\tilde{v}^*_k\| + C_1\|\hat{y}_k-y^*_k\|\right)^2\leq&\left(1-\frac{\mu}{4L}\right)\cdot\left(\left\|\hat{v}_{k-1}-\tilde{v}^*_{k-1}\right\| + C_1\left\|\hat{y}_{k-1}-y^*_{k-1}\right\|\right)^2 \\ &+\frac{3\beta^2C_2^2L}{\mu}\|\nabla\Phi(x_{k-1})\|^2.\end{aligned}$$

Moreover, we have

$$\|\hat{v}_k - \tilde{v}_k^*\| + C_1 \|\hat{y}_k - y_k^*\| \leq \left(1 - \frac{\mu}{2L}\right)^k \cdot \left(\left\|v_0^Q - \tilde{v}_0^*\right\| + C_1 \left\|y_0^N - y_0^*\right\|\right) + \beta C_2 \sum_{t=0}^{k} \left(1 - \frac{\mu}{2L}\right)^{k-1-t} \|\nabla\Phi(x_t)\|.$$

Thus, we can obtain that

$$(\|\hat{v}_k - \tilde{v}_k^*\| + C_1 \|\hat{y}_k - y_k^*\|)^2 \leq \left(1 - \frac{\mu}{4L}\right)^k \cdot \left(\left\|v_0^Q - \tilde{v}_0^*\right\| + C_1 \left\|y_0^N - y_0^*\right\|\right)^2 + \frac{3\beta^2 C_2^2 L}{\mu} \sum_{t=0}^{k} \left(1 - \frac{\mu}{4L}\right)^{k-1-t} \|\nabla\Phi(x_t)\|^2. \tag{12}$$

Therefore, we have

$$\begin{aligned}
\left\|\widehat{\nabla}\Phi(x_k) - \nabla\Phi(x_k)\right\|^2 \leq & L^2 \left(\|\hat{v}_k - \tilde{v}_k^*\| + \frac{C_3}{L} \|\hat{y}_k - y_k^*\|\right)^2 \\
\leq & L^2 \left(\|\hat{v}_k - \tilde{v}_k^*\| + C_1 \|\hat{y}_k - y_k^*\|\right)^2 \\
\leq & L^2 \left(1 - \frac{\mu}{4L}\right)^k \cdot \left(\left\|v_0^Q - \tilde{v}_0^*\right\| + C_1 \left\|y_0^N - y_0^*\right\|\right)^2 \\
& + \frac{3\beta^2 C_2^2 L^3}{\mu} \sum_{t=0}^{k} \left(1 - \frac{\mu}{4L}\right)^{k-1-t} \|\nabla\Phi(x_t)\|^2,
\end{aligned}$$

where the second inequality is because of $C_3 \leq LC_1$ and the specific derivation process is as follows

$$\frac{C_3}{LC_1} = \frac{\mu(L+\mu)}{L^2} = \frac{1}{\kappa} + \frac{1}{\kappa^2} < 1,$$

where $C_3 = \frac{(L\mu+\rho M)\cdot(L+\mu)}{\mu^2}$ and $LC_1 = \frac{L^2(L\mu+\rho M)}{\mu^3}$. □

A.5 Proof of Theorem 1

Proof. First, based on Lemma 2 in Ji et al. (2021), we have $\nabla\Phi(\cdot)$ is L_Φ-Lipschitz, where $L_\Phi = L + \frac{2L^2+\rho M^2}{\mu} + \frac{2\rho LM+L^3}{\mu^2} + \frac{\rho L^2 M}{\mu^3} = \Theta(\kappa^3)$. Then, we have

$$\begin{aligned}
\Phi(x_{k+1}) \leq & \Phi(x_k) + \langle \nabla\Phi(x_k), x_{k+1} - x_k\rangle + \frac{L_\Phi}{2} \|x_{k+1} - x_k\|^2 \\
\leq & \Phi(x_k) - \left(\frac{\beta}{2} - \beta^2 L_\Phi\right) \|\nabla\Phi(x_k)\|^2 + \left(\frac{\beta}{2} + \beta^2 L_\Phi\right) \left\|\nabla\Phi(x_k) - \widehat{\nabla}\Phi(x_k)\right\|^2 \\
\overset{Eq.\ (7)}{\leq} & \Phi(x_k) - \left(\frac{\beta}{2} - \beta^2 L_\Phi\right) \|\nabla\Phi(x_k)\|^2 + \left(\frac{\beta}{2} + \beta^2 L_\Phi\right) L^2 \left(1 - \frac{\mu}{4L}\right)^k \cdot \\
& \left(\left\|v_0^Q - \tilde{v}_0^*\right\| + C_1 \left\|y_0^N - y_0^*\right\|\right)^2 + \left(\frac{\beta}{2} + \beta^2 L_\Phi\right) \frac{3\beta^2 C_2^2 L^3}{\mu} \sum_{t=0}^{k} \left(1 - \frac{\mu}{4L}\right)^{k-1-t} \|\nabla\Phi(x_t)\|^2.
\end{aligned}$$

Telescoping above equation over k from 0 to $K-1$, we can obtain that

$$\begin{aligned}
\Phi(x_{K-1}) \leq & \Phi(x_0) - \left(\frac{\beta}{2} - \beta^2 L_\Phi\right) \sum_{k=0}^{K-1} \|\nabla\Phi(x_k)\|^2 + \left(\frac{\beta}{2} + \beta^2 L_\Phi\right) \frac{4L^3}{\mu} \\
& \cdot \left(\left\|v_0^Q - \tilde{v}_0^*\right\| + C_1 \left\|y_0^N - y_0^*\right\|\right)^2 + \left(\frac{\beta}{2} + \beta^2 L_\Phi\right) \frac{12\beta^2 C_2^2 L^4}{\mu^2} \sum_{k=0}^{K-1} \|\nabla\Phi(x_k)\|^2 \\
= & \Phi(x_0) - \beta \left(\frac{1}{2} - \beta L_\Phi - \left(\frac{1}{2} + \beta L_\Phi\right) \frac{12\beta^2 C_2^2 L^4}{\mu^2}\right) \sum_{k=0}^{K-1} \|\nabla\Phi(x_k)\|^2 \\
& + \left(\frac{\beta}{2} + \beta^2 L_\Phi\right) \frac{4L^3}{\mu} \cdot \left(\left\|v_0^Q - \tilde{v}_0^*\right\| + C_1 \left\|y_0^N - y_0^*\right\|\right)^2.
\end{aligned}$$

Because $\beta = \min\{\frac{1}{8L_\Phi} = \mathcal{O}(\kappa^{-3}), \mathcal{O}(\kappa^{-5})\} = \mathcal{O}(\kappa^{-5})$, we can obtain that

$$\frac{1}{K}\sum_{k=0}^{K-1}\|\nabla\Phi(x_k)\|^2 \leq \frac{\Phi(x_0)-\Phi(x^*)}{\beta AK} + \frac{4L^3(1+2\beta L_\Phi)}{2\mu AK}\cdot\left(\left\|v_0^Q-\tilde{v}_0^*\right\| + C_1\left\|y_0^N-y_0^*\right\|\right)^2,$$

where $A = \frac{1}{2} - \beta L_\Phi - \left(\frac{1}{2}+\beta L_\Phi\right)\frac{12\beta^2C_2^2L^4}{\mu^2}$, $L_\Phi = L + \frac{2L^2+\rho M^2}{\mu} + \frac{2\rho LM+L^3}{\mu^2} + \frac{\rho L^2 M}{\mu^3} = \mathcal{O}(\kappa^3)$. We rewrite y_0^N as $y_0^{N_0}$ and Let $N_0 \geq \frac{\ln(\mu)}{\ln(\mu/(\mu-L))}$, Thus, we have

$$\left\|v_0^Q-\tilde{v}_0^*\right\| + C_1\left\|y_0^N-y_0^*\right\| \leq \frac{M}{\mu} + \frac{2}{\mu}(L\|y_0^*\|+M) + 4L\left(\frac{\rho M}{\mu^2}+\frac{L}{\mu}\right)\|y_0^*\| = \mathcal{O}(\kappa^2),$$

because $\left\|y_0^{N_0}-y_0^*\right\| \leq (1-\alpha\mu)^{N_0}\left\|y_0^0-y_0^*\right\| \leq \mu\|y_0^*\|$. For the first term, we have

$$\frac{\Phi(x_0)-\Phi(x^*)}{\beta A} = \frac{2\mu^2(\Phi(x_0)-\Phi(x^*))}{\beta\mu^2 - 2\beta^2L_\Phi - 12\beta^3C_2^2L^4 - 24\beta^4L_\Phi C_2^2L^4} = \mathcal{O}(\kappa^5).$$

For the second term, we have

$$\begin{aligned}&\frac{4L^3(1+2\beta L_\Phi)}{2\mu AK}\cdot\left(\left\|v_0^Q-\tilde{v}_0^*\right\| + C_1\left\|y_0^N-y_0^*\right\|\right)^2\\ =&\frac{4L^3\mu+8\beta L_\Phi L^3\mu}{(1-2\beta L_\Phi)\mu^2 - 12\beta^2C_2^2L^4(1+2\beta L_\Phi)}\cdot\left(\left\|v_0^Q-\tilde{v}_0^*\right\| + C_1\left\|y_0^N-y_0^*\right\|\right)^2 = \mathcal{O}(\kappa^5).\end{aligned}$$

Then, we have

$$\frac{1}{K}\sum_{k=0}^{K-1}\|\nabla\Phi(x_k)\|^2 = \mathcal{O}\left(\frac{\kappa^5}{K}+\frac{\kappa^5}{K}\right) = \mathcal{O}\left(\frac{\kappa^5}{K}\right).$$

Then, to achieve an ϵ-accurate stationary point, we have $K = \mathcal{O}(\kappa^5\epsilon^{-1})$, and hence we have the following complexity results. 1) Gradient complexity: $\mathrm{Gc}(\epsilon) = 3K = \widetilde{\mathcal{O}}(\kappa^5\epsilon^{-1})$. 2) Matrix-vector product complexities: $\mathrm{Mv}(\epsilon) = K + KQ = \widetilde{\mathcal{O}}(\kappa^5\epsilon^{-1})$. □

B Proofs of the single-loop ITD-based algorithm

B.1 Additional useful Lemma

Lemma 8. *Consider the single-loop ITD-based algorithm in Algorithm 2. Suppose Assumptions 1-4 hold. Let $\alpha \leq \frac{1}{L}$, we have*

$$\left\|\nabla_x y_k^N(x_k) - \nabla_x y_k^*(x_k)\right\| \leq (1-\alpha\mu)\left\|\nabla_x y^*(x_k)\right\| + \alpha\rho\left\|y_k^0 - y^*(x_k)\right\|,$$

where $y_k^N(x_k) = y_k^0 - \alpha\nabla_y g(x_k, y_k^0)$ and $y_k^* = \arg\min_y g(x_k, y)$ for $k = 1, \ldots, K$.

Proof. According to the definition, we have $\nabla_x y_k^N(x_k) = -\alpha\nabla_{xy}^2 g(x_k, y_k^0)$ and $\nabla_x y^*(x_k) = -[\nabla_{yy}^2 g(x_k, y_k^*)]^{-1}\nabla_{xy}^2 g(x_k, y_k^*)$. Thus, we have

$$\begin{aligned}\left\|\nabla_x y_k^N(x_k) - \nabla_x y_k^*(x_k)\right\| =& \left\|-\alpha\nabla_{xy}^2 g(x_k, y_k^0) + [\nabla_{yy}^2 g(x_k, y_k^*)]^{-1}\nabla_{xy}^2 g(x_k, y_k^*)\right\|\\ \leq& \left\|\left(I - \alpha\nabla_{yy}^2 g(x_k, y_k^*)\right)[\nabla_{yy}^2 g(x_k, y_k^*)]^{-1}\nabla_{xy}^2 g(x_k, y_k^*)\right\|\\ &+ \left\|\alpha\left(\nabla_{xy}^2 g(x_k, y_k^*) - \nabla_{xy}^2 g(x_k, y_k^0)\right)\right\|\\ \leq& (1-\alpha\mu)\left\|\nabla_x y^*(x_k)\right\| + \alpha\rho\left\|y_k^0 - y^*(x_k)\right\|.\end{aligned}$$

Then, the proof is completed. □

B.2 Proof of Lemma 5

Proof. Accordingly, we have

$$\begin{aligned}\|\hat{y}_k - y^*(x_k)\| \leq & (1-\mu\alpha)\|\hat{y}_{k-1} - y^*(x_{k-1})\| + \frac{L}{\mu}\|x_{k-1} - x_k\| \\ \leq & (1-\mu\alpha)\|\hat{y}_{k-1} - y^*(x_{k-1})\| + \frac{L\beta}{\mu}\left\|\widehat{\nabla}\Phi(x_{k-1})\right\| \\ \leq & (1-\mu\alpha)\|\hat{y}_{k-1} - y^*(x_{k-1})\| + \frac{L\beta}{\mu}\left(\|\nabla\Phi(x_{k-1})\| + \left\|\widehat{\nabla}\Phi(x_{k-1}) - \nabla\Phi(x_{k-1})\right\|\right) \\ \leq & (1-\mu\alpha)\|\hat{y}_{k-1} - y^*(x_{k-1})\| + \frac{L\beta}{\mu}\left(\|\nabla\Phi(x_{k-1})\| + C_4\|\hat{y}_{k-1} - y^*_{k-1}\| + C_5\right) \\ \leq & \left(1 - \mu\alpha + \frac{L\beta C_4}{\mu}\right)\|\hat{y}_{k-1} - y^*(x_{k-1})\| + \frac{L\beta}{\mu}\|\nabla\Phi(x_{k-1})\| + \frac{L\beta C_5}{\mu}.\end{aligned}$$

We rewrite the above equation as $\|\hat{y}_k - y^*(x_k)\| \leq C_6\|\hat{y}_{k-1} - y^*(x_{k-1})\| + \frac{L\beta}{\mu}\|\nabla\Phi(x_{k-1})\| + C_7$, where $C_6 = 1 - \mu\alpha + \frac{L\beta C_4}{\mu}$ and $C_7 = \frac{L\beta C_5}{\mu}$. Then, the proof of Eq. (8) is completed.

Since $\beta \leq \frac{\mu^3}{2L(2L^2+\rho M)}$, we have $C_6 \leq 1 - \frac{\mu}{2L}$. Accordingly, we have

$$\|\hat{y}_k - y^*(x_k)\| \leq \left(1 - \frac{\mu}{2L}\right)^k\|\hat{y}_0 - y^*(x_0)\| + \frac{L\beta^2}{\mu}\sum_{j=0}^{k-1}\left(1 - \frac{\mu}{2L}\right)^{k-1-j}\left(\|\nabla\Phi(x_j)\| + C_5\right).$$

Then, the proof of Eq. (9) is completed. Similar with AID in Eq. (12), we can obtain

$$\|\hat{y}_k - y^*(x_k)\|^2 \leq \left(1 - \frac{\mu}{4L}\right)^k\|\hat{y}_0 - y^*(x_0)\|^2 + \frac{3L\beta^2}{\mu}\sum_{j=0}^{k-1}\left(1 - \frac{\mu}{4L}\right)^{k-1-j}\left(\|\nabla\Phi(x_j)\| + C_5\right)^2. \tag{13}$$

□

B.3 Proof of Lemma 6

Proof. First, according to the definition of $\widehat{\nabla}\Phi(x_k)$ and $\nabla\Phi(x_k)$, we have

$$\begin{aligned}&\left\|\widehat{\nabla}\Phi(x_k) - \nabla\Phi(x_k)\right\| \\ \leq & \|\nabla_1 f(x_k, \hat{y}_k) + \nabla_x\hat{y}_k(x_k)\nabla_2 f(x_k, \hat{y}_k) - \nabla_1 f(x_k, y_k^*) - \nabla_x y_k^*(x_k)\nabla_2 f(x_k, y_k^*)\| \\ \leq & L\|\hat{y}_k - y_k^*\| + \|\nabla_x\hat{y}_k(x_k)\nabla_2 f(x_k, \hat{y}_k) - \nabla_x\hat{y}_k(x_k)\nabla_2 f(x_k, y_k^*)\| \\ & + \|\nabla_x\hat{y}_k(x_k)\nabla_2 f(x_k, y_k^*) - \nabla_x y_k^*(x_k)\nabla_2 f(x_k, y_k^*)\| \\ \leq & L\|\hat{y}_k - y_k^*\| + \alpha L^2\|\hat{y}_k - y_k^*\| + M\left((1-\alpha\mu)\frac{L}{\mu} + \alpha\rho\left\|y_k^0 - y_k^*\right\|\right).\end{aligned}$$

For the relationship of $\left\|y_k^0 - y^*(x_k)\right\|$ and $\|\hat{y}_k - y^*(x_k)\|$, we have

$$\left\|y_k^0 - y^*(x_k)\right\| \leq \leq \alpha\left\|\nabla_y g(x_k, y_k^0)\right\| + \|\hat{y}_k - y^*(x_k)\| \leq \alpha M + \|\hat{y}_k - y^*(x_k)\|.$$

Then, we have

$$\left\|\widehat{\nabla}\Phi(x_k) - \nabla\Phi(x_k)\right\| \leq C_4\|\hat{y}_k - y_k^*\| + C_5, \tag{14}$$

where $C_4 = L + \alpha L^2 + \alpha\rho M$ and $C_5 = M(1-\alpha\mu)\frac{L}{\mu} + \alpha^2\rho M^2$. □

B.4 Proof of Lemma 7

Proof. According to Lemma 6, we have

$$\begin{aligned}&\left\|\widehat{\nabla}\Phi(x_k) - \nabla\Phi(x_k)\right\|^2 \leq \left(C_4\|\hat{y}_k - y_k^*\| + C_5\right)^2 \leq C_4^2\|\hat{y}_k - y_k^*\|^2 + 3C_5^2 \\ \leq & C_4^2\left(1 - \frac{\mu}{4L}\right)^k\|\hat{y}_0 - y_0^*\|^2 + \frac{3L\beta^2C_4^2}{\mu}\sum_{j=0}^{k-1}\left(1 - \frac{\mu}{4L}\right)^{k-1-j}\left(\|\nabla\Phi(x_j)\| + C_5\right)^2 + 3C_5^2,\end{aligned}$$

where the last inequality holds since Eq. (13). □

B.5 Proof of Theorem 3

Proof. First, based on Lemma 2 in Ji et al. (2021), we have $\nabla\Phi(\cdot)$ is L_Φ-Lipschitz, where $L_\Phi = L + \frac{2L^2+\rho M^2}{\mu} + \frac{2\rho LM+L^3}{\mu^2} + \frac{\rho L^2 M}{\mu^3} = \Theta(\kappa^3)$. Then, we have

$$\begin{aligned}
\Phi(x_{k+1}) \leq & \Phi(x_k) - \left(\frac{\beta}{2} - \beta^2 L_\Phi\right) \|\nabla\Phi(x_k)\|^2 + \left(\frac{\beta}{2} + \beta^2 L_\Phi\right) \left\|\widehat{\nabla}\Phi(x_k) - \nabla\Phi(x_k)\right\|^2 \\
\leq & \Phi(x_k) - \left(\frac{\beta}{2} - \beta^2 L_\Phi\right) \|\nabla\Phi(x_k)\|^2 + \left(\frac{\beta}{2} + \beta^2 L_\Phi\right) C_4^2 \left(1 - \frac{\mu}{4L}\right)^k \|\hat{y}_0 - y_0^*\|^2 \\
& + \left(\frac{\beta}{2} + \beta^2 L_\Phi\right) \frac{3L\beta^2 C_4^2}{\mu} \sum_{j=0}^{k-1} \left(1 - \frac{\mu}{4L}\right)^{k-1-j} (\|\nabla\Phi(x_j)\| + C_5)^2 + \left(\frac{\beta}{2} + \beta^2 L_\Phi\right) 3C_5^2.
\end{aligned}$$

Telescoping the above equation over k from 0 to $K-1$ yields

$$\begin{aligned}
\Phi(x_{K-1}) \leq & \Phi(x_0) - \left(\frac{\beta}{2} - \beta^2 L_\Phi\right) \sum_{k=0}^{K-1} \|\nabla\Phi(x_k)\|^2 + \left(\frac{\beta}{2} + \beta^2 L_\Phi\right) C_4^2 \frac{4L}{\mu} \|\hat{y}_0 - y_0^*\|^2 \\
& + \left(\frac{\beta}{2} + \beta^2 L_\Phi\right) \frac{3L\beta^2 C_4^2}{\mu} \frac{4L}{\mu} \sum_{k=0}^{K-1} (\|\nabla\Phi(x_k)\| + C_5)^2 + \left(\frac{\beta}{2} + \beta^2 L_\Phi\right) 3C_5^2 K \\
\leq & \Phi(x_0) - A \sum_{k=0}^{K-1} \|\nabla\Phi(x_k)\|^2 + B_1 + B_2 + \left(\frac{\beta}{2} + \beta^2 L_\Phi\right) 3C_5^2 K,
\end{aligned}$$

where

$$A = \left(\frac{\beta}{2} - \beta^2 L_\Phi\right) - \left(\frac{\beta}{2} + \beta^2 L_\Phi\right) \frac{12L^2\beta^2 C_4^2}{\mu^2}, \quad B_1 = \left(\frac{\beta}{2} + \beta^2 L_\Phi\right) C_4^2 \frac{4L}{\mu} \|\hat{y}_0 - y_0^*\|^2,$$
$$B_2 = \left(\frac{\beta}{2} + \beta^2 L_\Phi\right) \frac{36L^2\beta^2 C_4^2 C_5^2}{\mu^2}.$$

Thus we have

$$\frac{1}{K} \sum_{k=0}^{K-1} \|\nabla\Phi(x_j)\|^2 \leq \frac{\Phi(x_0) - \Phi(x^*)}{AK} + \frac{B_1 + B_2}{AK} + \left(\frac{\beta}{2} + \beta^2 L_\Phi\right) \frac{3C_5^2}{A},$$

where $\beta = \mathcal{O}(\kappa^{-3})$, $L_\Phi = \mathcal{O}(\kappa^3)$, $C_4 = \mathcal{O}(1)$, $C_5 = \mathcal{O}(\kappa^1)$. Thus we have $\frac{1}{A} = \mathcal{O}(\kappa^3)$, $\frac{B_1}{A} = \mathcal{O}(\kappa^1)$, $\frac{B_2}{A} = \mathcal{O}(\kappa^{-2})$. Therefore, we have

$$\frac{1}{K} \sum_{k=0}^{K-1} \|\nabla\Phi(x_k)\|^2 = \mathcal{O}\left(\frac{\kappa^3}{K} + \kappa^2\right).$$

Therefore the proof is completed. □

C The Use of Large Language Models

In preparing this paper, we made limited use of ChatGPT (an OpenAI large language model) solely for language polishing and minor improvements in clarity and readability of a few sections. The LLM did not contribute to research ideation, technical content, experimental design, analysis, or writing of substantive material. All research ideas, methods, results, and conclusions are entirely those of the authors.