Accelerating Inexact HyperGradient Descent for Bilevel Optimization

Haikuo Yang YANGHK22@M.FUDAN.EDU.CN
Luo Luo LUOLUO@FUDAN.EDU.CN
Fudan University
Chris Junchi Li JUNCHILI@BERKELEY.EDU
Michael I. Jordan JORDAN@CS.BERKELEY.EDU
University of California, Berkeley
Maryam Fazel MFAZEL@UW.EDU
University of Washington

Abstract

We present a method for solving general nonconvex-strongly-convex bilevel optimization problems. Our method—the *Restarted Accelerated HyperGradient Descent* (RAHGD) method—finds an ϵ-first-order stationary point of the objective with $\tilde{\mathcal{O}}(\kappa^{3.25}\epsilon^{-1.75})$ oracle complexity, where κ is the condition number of the lower-level objective and ϵ is the desired accuracy. We also propose a perturbed variant of RAHGD for finding an $\big(\epsilon, \mathcal{O}(\kappa^{2.5}\sqrt{\epsilon}\,)\big)$-second-order stationary point within the same order of oracle complexity. Our results achieve the best-known theoretical guarantees for finding stationary points in bilevel optimization and also improve upon the existing upper complexity bound for finding second-order stationary points in nonconvex-strongly-concave minimax optimization problems, setting a new state-of-the-art benchmark. Empirical studies are conducted to validate the theoretical results in this paper.

1. Introduction

Bilevel optimization is emerging as a key unifying problem formulation in machine learning, encompassing a variety of applications including meta-learning, model-free reinforcement learning and hyperparameter optimization [16, 46]. Our work focuses on a version of the general problem that is particularly relevant to machine learning—the *nonconvex-strongly-convex bilevel optimization problem*:

$$\min_{x\in\mathbb{R}^{d_x}} \; \Phi(x) \triangleq f(x, y^*(x)), \tag{1a}$$

$$\text{s.t} \quad y^*(x) = \arg\min_{y\in\mathbb{R}^{d_y}} \; g(x, y), \tag{1b}$$

where the upper-level function $f(x, y)$ is smooth and possibly nonconvex, and the lower-level function $g(x, y)$ is smooth and strongly convex with respect to y for any given x.Bilevel optimization is more expressive but harder to solve than classical single-level optimization since the objective $\Phi(x)$ in (1a) involves the argument input $y^*(x)$ which is the solution of the lower-level problem (1b). In contradistinction to classical optimization, bilevel optimization problem (1) involves solving an

* Full version available at https://arxiv.org/abs/2307.00126.

optimization problem where the minimization variable is taken as the minimizer of a lower-level optimization problem.

Most existing work on nonconvex-strongly-convex bilevel optimization [17, 25, 26] focuses on finding approximate *first-order stationary points* (FOSP) of the objective. Recently, Huang et al. [23] extended the scope of work in this area, proposing the (perturbed) *approximate implicit differentiation* (AID) algorithm which can find an $\big(\epsilon, \mathcal{O}(\kappa^{2.5}\sqrt{\epsilon}\,)\big)$- *second-order stationary points* (SOSP) within a $\tilde{\mathcal{O}}(\kappa^4\epsilon^{-2})$ oracle complexity, where $\kappa \geq 1$ is the condition number of any $g(x,\cdot)$ and $\epsilon > 0$ is the desired accuracy. Given this result, a key further challenge is to study whether the ϵ^{-2}-dependency in the upper complexity bound can be improved under additional Lipschitz assumptions on high-order derivatives [23].

Given this context, a natural question to ask is: *Can we design an algorithm that improves upon known algorithmic complexities for finding approximate first-order and second-order stationary points in nonconvex-strongly-convex bilevel optimization?* We provide an affirmative answer to this question, by designing a particular form of acceleration of hypergradient descent and thereby improving the oracle complexity.

Organization. The rest of this work is organized as follows. Section 2 delineates the assumptions and specific algorithmic subroutines. Section 3 formally presents the `RAHGD` algorithm along with its complexity bound for finding approximation first-order stationary points. Section 4 proposes the `PRAHGD`, the perturbed version of `RAHGD`, along with its complexity bound for finding approximate second-order stationary points. Further presentation of contributions, related work, technical analysis and additional experiments are deferred to the supplementary materials.

Notation. We let $\|\cdot\|_2$ be the spectral norm of matrices and the Euclidean norm of vectors. Given a real symmetric matrix A, we let $\lambda_{\max}(A)$ ($\lambda_{\min}(A)$) denote its largest (smallest) eigenvalue. We use the notation $\mathbb{B}(r)$ to present the closed Euclidean ball with radius r centered at the origin. We denote $Gc(f,\epsilon)$, $JV(f,\epsilon)$ and $HV(f,\epsilon)$ as the oracle complexities of gradients, Jacobian-vector products and Hessian-vector products, respectively. Finally, we adopt the notation $\mathcal{O}(\cdot)$ to hide only absolute constants which do not depend on any problem parameters, and also $\tilde{\mathcal{O}}(\cdot)$ for constants that include a polylogarithmic factor.

2. Preliminaries

In this section, we first proceed to establish convergence of the algorithmic subroutines related to our algorithm—*accelerated gradient descent* and the *conjugate gradient method*. Then, we present the notations and assumptions necessary for our problem setting. We proceed to establish convergence of these two algorithmic subroutines in the following paragraphs.

Subroutine 1: Accelerated Gradient Descent. Our first component is *Nesterov's accelerated gradient descent* (AGD), which is an acceleration of the first-order method in smooth convex optimization. We describe the details of AGD for minimizing a given smooth and strongly convex function in Algorithm 1, which exhibits the following *optimal* convergence rate [39]:

Lemma 1 ([39]) *Running Algorithm 1 on an ℓ_h-smooth and μ_h-strongly convex objective function $h(\cdot)$ with $\alpha = 1/\ell_h$ and $\beta = (\sqrt{\kappa_h} - 1)/(\sqrt{\kappa_h}+1)$ produces an output z_T satisfying*

$$\|z_T - z^*\|_2^2 \leq (1+\kappa_h)\left(1 - 1/\sqrt{\kappa_h}\right)^T \|z_0 - z^*\|_2^2,$$

where $z^* = \arg\min_z h(z)$ and $\kappa_h = \ell_h/\mu_h$ denotes the condition number of the objective h.

Subroutine 2: Conjugate Gradient Method. The *(linear) conjugate gradient* (CG) method was proposed by Hestenes and Stiefel in the 1950s as an iterative method for solving linear systems with positive definite coefficient matrices. It serves as an alternative to Gaussian elimination that is well-suited for solving large problems. CG can be formulated as the minimization of the quadratic objective function

$$\frac{1}{2}q^\top Aq - q^\top b, \tag{2}$$

where $A \in \mathbb{R}^{d\times d}$ is a positive definite matrix and $b \in \mathbb{R}^d$ is a fixed vector. We summarize the setup of CG for minimizing function (2) in Algorithm 2, and record the following convergence property [41]:

Lemma 2 ([41]) *Running Algorithm 2 for minimizing quadratic function (2) produces q_T satisfying*

$$\|q_T - q^*\|_2 \le 2\sqrt{\kappa_A}\left(\frac{\sqrt{\kappa_A}-1}{\sqrt{\kappa_A}+1}\right)^T \|q_0 - q^*\|_2,$$

where $q^* = A^{-1}b$ denotes the unique minimizer of Eq. (2), and $\kappa_A = \lambda_{\max}(A)/\lambda_{\min}(A)$ denotes the condition number of (positive definite) matrix A.

In the rest of this section we impose the following assumptions on the upper-level function f and the lower-level function g. We then turn to the details of our theoretical analysis:

Assumption 3 *The upper-level function $f(x, y)$ and lower-level function $g(x, y)$ satisfy the following conditions:*

(i) Function $g(x, y)$ is three times differentiable and μ-strongly convex with respect to y for any fixed x;

(ii) Function $f(x, y)$ is twice differentiable and M-Lipschitz continuous with respect to y;

(iii) Gradient $\nabla f(x, y)$ and $\nabla g(x, y)$ are ℓ-Lipschitz continuous with respect to x and y;

(iv) Jacobian $\nabla^2_{xy} f(x, y)$, $\nabla^2_{xy} g(x, y)$ and Hessians $\nabla^2_{xx} f(x, y)$, $\nabla^2_{yy} f(x, y)$, $\nabla^2_{yy} g(x, y)$ are ρ-Lipschitz continuous with respect to x and y;

(v) Third-order derivatives $\nabla^3_{xyx} g(x, y)$, $\nabla^3_{yxy} g(x, y)$ and $\nabla^3_{yyy} g(x, y)$ are ν-Lipschitz continuous with respect to x and y.

These assumptions are standard for the bilevel optimization problem we are studying. We also introduce an appropriate notion of condition number for the lower-level function $g(x, y)$.

Definition 4 *Under Assumption 3, we refer to $\kappa \triangleq \ell/\mu$ the* condition number *of the lower-level objective $g(x, y)$.*

Leveraging such a notion, we can show that the solution to the lower-level optimization problem $y^*(x) = \arg\min_{y\in\mathbb{R}^{d_y}} g(x, y)$ is κ-Lipschitz continuous in x under Assumption 3, as indicated in the following lemma:

Lemma 5 *Suppose Assumption 3 holds, then $y^*(x)$ is κ-Lipschitz continuous, that is, we have $\|y^*(x) - y^*(x')\|_2 \le \kappa \|x - x'\|_2$ for any $x, x' \in \mathbb{R}^{d_x}$.*

We also can show that $\Phi(x)$ admits Lipschitz continuous gradients and Lipschitz continuous Hessians, as shown in the following lemmas:

Lemma 6 *Suppose Assumption 3 holds, then $\Phi(x)$ is $\tilde{L}$-gradient Lipschitz continuous, that is, we have $\|\nabla\Phi(x) - \nabla\Phi(x')\| \le \tilde{L}\|x - x'\|$ for any $x, x' \in \mathbb{R}^{d_x}$, where $\tilde{L} = \mathcal{O}(\kappa^3)$.*

Lemma 7 *Suppose Assumption 3 holds, then $\Phi(x)$ is $\tilde{\rho}$-Hessian Lipschitz continuous, that is, $\|\nabla^2\Phi(x) - \nabla^2\Phi(x')\| \le \tilde{\rho}\|x - x'\|$ for any $x, x' \in \mathbb{R}^{d_x}$, where $\tilde{\rho} = \mathcal{O}(\kappa^5)$.*

The detailed form of $\tilde{L}$ and $\tilde{\rho}$ can be found in Appendix C. Finally, with the definition of $\tilde{\rho}$ in hand, we give the formal definition of an ϵ-first-order stationary point as well as an (ϵ, τ)-second-order stationary point, as follows:

Definition 8 (Approximate First-Order Stationary Point) *Under Assumption 3, we call x an* ϵ-first-order stationary point *of $\Phi(x)$ if $\|\nabla\Phi(x)\|_2 \le \epsilon$.*

Definition 9 (Approximate Second-Order Stationary Point) *Under Assumption 3, we call x an* (ϵ, τ)-second-order stationary point *of $\Phi(x)$ if $\|\nabla\Phi(x)\|_2 \le \epsilon$ and $\lambda_{\min}(\nabla^2\Phi(x)) \ge -\tau$.*

We remark that these concepts are commonly used in the nonconvex optimization literature [40]. The approximate second-order stationary point is sometimes referred to as an "approximate local minimizer." With all these preliminaries at hand, we are ready to proceed with the (perturbed) restarted accelerated hypergradient descent method.

3. Restarted Accelerated HyperGradient Descent Algorithm

In this section, we present our *restarted accelerated hypergradient descent* (`RAHGD`) algorithm and provide corresponding query complexity upper bound results. We present the details of `RAHGD` in Algorithm 3, which has a nested loop structure. The outer loop, indexed by k, uses the accelerated gradient descent method to find the solver of (1a). The AGD step in Line 5 is used to find the inexact solver of (1b). The CG step is added to compute the Hessian-vector product, as shown in (5). We note that the iteration numbers of the AGD and CG steps play an important role in the convergence analysis of Algorithm 3; moreover, at the end of this section we will show that the total iteration number of AGD and CG can be bounded sharply. Finally, note that there is a restarting step in Line 12 where the option `Perturbation` is taken as $= 0$.

We let subscript t index the times of restarting. We note that the subscript t of epoch number is added in Algorithm 3 purely for the sake of an easier convergence analysis. The incurred storage of iterations across all epochs can be avoided when implementing Algorithm 3 in practice.

In accelerated nonconvex optimization, a straightforward application of AGD cannot ensure consistent decrements of the objective function. Inspired by the work of Li and Lin [33], we add a restarting step in Line 12—we define $\mathcal{K}$ to be the iteration number when the "if condition" triggers, and hence the iterates from $k = 0$ to $k = \mathcal{K}$ constructs one single epoch, where $\mathcal{K} = \min_k \left\{k \ge 1 : \, k\sum_{t=0}^{k-1} \|x_{t+1} - x_t\|_2^2 > B^2\right\}$. Then we can have the objective function consistently decrease with respect to each epoch when we run Algorithm 3. We provide the convergence results for `RAHGD` in the rest of this section.

Denote $v_k^* = \left(\nabla_{yy}^2 g(w_k, y_k)\right)^{-1}\nabla_y f(w_k, y_k)$. Due to the bilevel optimization problem we are considering the following conditions on the inexact gradient. Recall that the overall objective function $\Phi(x)$ is $\tilde{L}$-gradient Lipschitz continuous, and both the upper-level function $f(x,y)$ and the lower-level function $g(x,y)$ are ℓ-gradient Lipschitz continuous:

Condition 10 *Let $w_{-1} = x_{-1}$. Then for some $\sigma > 0$, we assume that the estimators $y_k \in \mathbb{R}^{d_y}$ and $v_k \in \mathbb{R}^{d_y}$ satisfy the conditions*

$$\|y_k - y^*(w_k)\|_2 \le \frac{\sigma}{2\tilde{L}}, \qquad \text{for each } k = -1, 0, 1, 2, \dots \tag{3}$$

and

$$\|v_k - v_k^*\| \le \frac{\sigma}{2\ell}, \qquad \text{for each } k = 0, 1, 2, \dots \tag{4}$$

Remark 11 *We will show at the end of this section that Condition 10 is guaranteed to hold after running AGD and CG for a sufficient number of iterations.*

Under Condition 10, the bias of $\hat{\nabla}\Phi(x_k)$ defined in equation (7) can be bounded as shown in the following lemma:

Lemma 12 (Inexact gradients) *Suppose Assumption 3 and Condition 10 hold, then we have $\|\nabla\Phi(w_k) - \hat{\nabla}\Phi(w_k)\|_2 \le \sigma$.*

In the following theorem we show that the iteration complexity in the outer loop is bounded.

Theorem 13 (`RAHGD` finding FOSP) *Suppose that Assumptions 3 and Condition 10 hold. Let*

$$\eta = \frac{1}{4\tilde{L}}, \quad B = \sqrt{\frac{\epsilon}{\tilde{\rho}}}, \quad \theta = 4(\tilde{\rho}\epsilon\eta^2)^{1/4}, \quad K = \frac{1}{\theta}, \quad \alpha = \frac{1}{\ell}, \quad \beta = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}, \quad \sigma = \epsilon^2.$$

Denote $\Delta = \Phi(x_{\text{int}}) - \min_x \Phi(x)$. Then `RAHGD` in Algorithm 3 terminates within $\mathcal{O}(\Delta\tilde{L}^{0.5}\tilde{\rho}^{0.25}\epsilon^{-1.75})$ iterations, outputting $\hat{w}$ satisfying $\|\nabla\Phi(\hat{w})\|_2 \le 83\epsilon$.

Theorem 13 says that Algorithm 3 can find an ϵ-first-order stationary point with $\mathcal{O}(\kappa^{2.75}\epsilon^{-1.75})$ iterations in the outer loop. The following result indicates that Condition 10 holds if we run AGD and CG for a sufficient number of iterations. In addition, the total number of iterations in one epoch is at most $\mathcal{O}(\kappa^{0.5}\mathcal{K}\log(1/\epsilon))$:

Proposition 14 *Suppose Assumption 3 holds. In the t-th epoch, we set the inner loop iteration number $T_{t,k}$ and the CG iteration number $T'_{t,k}$. We run Algorithm 3 with the parameter chosen in Theorem 13. Then all $y_{t,k}$ and $v_{t,k}$ satisfy Condition 10. For each t, we also have the following bounds for the inner loops $\sum_{k=-1}^{\mathcal{K}-1} T_{t,k} \le \mathcal{O}(\kappa^{0.5}\mathcal{K}\log(1/\epsilon))$ and $\sum_{k=0}^{\mathcal{K}-1} T'_{t,k} \le \mathcal{O}(\kappa^{0.5}\mathcal{K}\log(1/\epsilon))$.*

The detailed forms of $T_{t,k}$ and $T'_{t,k}$ can be found in Appendix D. Combined with Theorem 13, we finally obtain the total number of oracle calls as follows:

Corollary 15 (Oracle complexity of `RAHGD`) *Under Assumption 3, we run `RAHGD` in Algorithm 3 with the parameters set as in Theorem 13 and Proposition 14. The output $\hat{w}$ is then an ϵ-first-order stationary point of $\Phi(x)$. Additionally, the oracle complexities satisfy $Gc(f,\epsilon) = \tilde{\mathcal{O}}(\kappa^{2.75}\epsilon^{-1.75})$, $Gc(g,\epsilon) = \tilde{\mathcal{O}}(\kappa^{3.25}\epsilon^{-1.75})$, $JV(g,\epsilon) = \tilde{\mathcal{O}}(\kappa^{2.75}\epsilon^{-1.75})$ and $HV(g,\epsilon) = \tilde{\mathcal{O}}(\kappa^{3.25}\epsilon^{-1.75})$.*

The algorithm can be adapted to solving the single-level nonconvex minimization problem where κ reduces to 1, and the given complexity matches the state-of-the-art [1, 6, 7, 28, 33]. The best known lower bound in this setting is $O(\epsilon^{-1.714})$ [8]. Closing this $O(\epsilon^{-0.036})$-gap remains open even in nonconvex minimization settings.

4. Perturbed Restarted Accelerated HyperGradient Descent Algorithm

In this section, we introduce perturbation to our `RAHGD` algorithm. In many nonconvex problems encountered in practice in machine learning, most first-order stationary points presented are saddle points [12, 27, 32]. Recall that the notion of second-order stationary points consists of not only zero gradient value, but positive semidefinite Hessian matrix as well. Earlier work of Jin et al. [28], Li and Lin [33] shows that one can obtain an approximate second-order stationary point by intermittently *perturbing* the algorithm using random noise. We present the details of our *perturbed restarted accelerated hypergradient descent* (`PRAHGD`) in Algorithm 3. Compared with `RAHGD`, a noise-perturbation step is added in Algorithm 3 [Line 12, option `Perturbation` = 1].

We proceed with the complexity analysis for `PRAHGD`, where we show that `PRAHGD` in Algorithm 3 outputs an $(\epsilon, \sqrt{\tilde{\rho}\epsilon})$-second-order stationary point within $\tilde{\mathcal{O}}(\kappa^{3.25}\epsilon^{-1.75})$ oracle queries:

Theorem 16 (PRAHGD finding SOSP) *Suppose that Assumption 3 and Condition 10 hold. Let*

$$\chi = \mathcal{O}\left(\log\frac{d_x}{\zeta\epsilon}\right), \quad \eta = \frac{1}{4\tilde{L}}, \quad K = \frac{2\chi}{\theta}, \quad B = \frac{1}{288\chi^2}\sqrt{\frac{\epsilon}{\tilde{\rho}}}, \quad \theta = \frac{1}{2}(\tilde{\rho}\epsilon\eta^2)^{1/4},$$
$$\sigma = \min\left\{\frac{\tilde{\rho}B\zeta r\theta}{2\sqrt{d_x}}, \epsilon^2\right\}, \ \alpha = \frac{1}{\ell}, \ \beta = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}, \ r = \min\left\{\frac{\tilde{L}B^2}{4C}, \frac{B+B^2}{\sqrt{2}}, \frac{\theta B}{20K}, \sqrt{\frac{\theta B^2}{2K}}\right\}$$

for some positive constant C. Denote $\Delta = \Phi(x_{\text{int}}) - \min_{x\in\mathbb{R}^{d_x}}\Phi(x)$. Then `PRAHGD` in Algorithm 3 terminates in at most $\mathcal{O}\big(\Delta\tilde{L}^{0.5}\tilde{\rho}^{0.25}\chi^6 \cdot \epsilon^{-1.75}\big)$ iterations and the output satisfies $\|\nabla\Phi(\hat{w})\|_2 \le \epsilon$ and $\lambda_{\min}(\nabla^2\Phi(\hat{w})) \ge -1.011\sqrt{\tilde{\rho}\epsilon}$ with probability at least $1-\zeta$.

Theorem 16 says that `PRAHGD` in Algorithm 3 can find an $(\epsilon, \sqrt{\tilde{\rho}\epsilon})$-second-order stationary point within $\tilde{\mathcal{O}}(\kappa^{2.75}\epsilon^{-1.75})$ iterations in the outer loop. The following proposition shows that Condition 10 holds in this setting. In addition, the total number of iterations in one epoch is at most $\mathcal{O}(\kappa^{0.5}\mathcal{K}\log(1/\epsilon))$:

Proposition 17 *Suppose Assumption 3 holds. In the t-th epoch, we set the inner loop iteration number $T_{t,k}$ and the CG iteration number $T'_{t,k}$. We run Algorithm 3 with the parameters chosen in Theorem 16. Then all $y_{t,k}$ and $v_{t,k}$ satisfy the Condition 10. For each t, we also have the inner loops $\sum_{k=-1}^{\mathcal{K}-1} T_{t,k} \le \mathcal{O}\left(\kappa^{0.5}\mathcal{K}\log(1/\epsilon)\right)$ and $\sum_{k=0}^{\mathcal{K}-1} T'_{t,k} \le \mathcal{O}\left(\kappa^{0.5}\mathcal{K}\log(1/\epsilon)\right)$ holds.*

The detailed form of $T_{t,k}$ and $T'_{t,k}$ can be found in Appendix E. Combining this result with Theorem 16, we finally obtain the total number of gradient oracle calls as follows:

Corollary 18 (Oracle complexity of PRAHGD) *Under Assumption 3, we run `PRAHGD` in Algorithm 3 with all parameters set as in Theorem 16. The output $\hat{w}$ is then an $\left(\epsilon, \sqrt{\tilde{\rho}\epsilon}\right)$-second-order stationary point of $\Phi(x)$. Additionally, the oracle complexities satisfy that $Gc(f,\epsilon) = \tilde{\mathcal{O}}(\kappa^{2.75}\epsilon^{-1.75})$, $Gc(g,\epsilon) = \tilde{\mathcal{O}}(\kappa^{3.25}\epsilon^{-1.75})$, $JV(g,\epsilon) = \tilde{\mathcal{O}}(\kappa^{2.75}\epsilon^{-1.75})$ and $HV(g,\epsilon) = \tilde{\mathcal{O}}(\kappa^{3.25}\epsilon^{-1.75})$.*

Acknowledgments and Disclosure of Funding

This work is supported in part by the Mathematical Data Science program of the Office of Naval Research under grant number N00014-18-1-2764 and also the Vannevar Bush Faculty Fellowship program under grant number N00014-21-1-2941 and National Science Foundation (NSF) grant IIS-1901252 to MIJ, and in part by awards NSF TRIPODS II-DMS 2023166, NSF CCF 2007036, NSF CCF 2212261, NSF AF 2312775 to MF. This work is also supported by the National Natural Science Foundation of China (No. 62206058) and the Shanghai Sailing Program (22YF1402900) to LL.

References

[1] Naman Agarwal, Zeyuan Allen-Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma. Finding approximate local minima faster than gradient descent. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1195–1199, 2017.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223. PMLR, 2017.

[3] Luca Bertinetto, Joao F. Henriques, Philip H.S. Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018.

[4] Rajendra Bhatia. *Matrix Analysis*, volume 169. Springer, 1997.

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

[6] Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Convex until proven guilty: Dimension-free acceleration of gradient descent on non-convex functions. In *International Conference on Machine Learning*, pages 654–663. PMLR, 2017.

[7] Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Accelerated methods for nonconvex optimization. *SIAM Journal on Optimization*, 28(2):1751–1772, 2018.

[8] Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points II: first-order methods. *Mathematical Programming*, 185(1-2):315–355, 2021.

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

[10] Ziyi Chen, Qunwei Li, and Yi Zhou. Escaping saddle points in nonconvex minimax optimization via cubic-regularized gradient descent-ascent. *arXiv preprint arXiv:2110.07098*, 2021.

[11] Frank E. Curtis, Daniel P. Robinson, and Mohammadreza Samadi. A trust region algorithm with a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ for nonconvex optimization. *Mathematical Programming*, 162:1–32, 2017.

[12] Yann N. Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *Advances in Neural Information Processing Systems*, 27, 2014.

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

[14] Tanner Fiez and Lillian Ratliff. Gradient descent-ascent provably converges to strict local minmax equilibria with a finite timescale separation. In *International Conference on Learning Representations*, 2021.

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

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

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

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[19] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

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

[21] 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.

[22] Minhui Huang, Shiqian Ma, and Lifeng Lai. A Riemannian block coordinate descent method for computing the projection robust wasserstein distance. In *International Conference on Machine Learning*, pages 4446–4455. PMLR, 2021.

[23] Minhui Huang, Kaiyi Ji, Shiqian Ma, and Lifeng Lai. Efficiently escaping saddle points in bilevel optimization. *arXiv preprint arXiv:2202.03684*, 2022.

[24] Kaiyi Ji, Jason D. Lee, Yingbin Liang, and H. Vincent Poor. Convergence of meta-learning with task-specific adaptation over partial parameters. *Advances in Neural Information Processing Systems*, 33:11490–11500, 2020.

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

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

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

[28] Chi Jin, Praneeth Netrapalli, and Michael I. Jordan. Accelerated gradient descent escapes saddle points faster than gradient descent. In *Conference On Learning Theory*, pages 1042–1085. PMLR, 2018.

[29] Chi Jin, Praneeth Netrapalli, and Michael I. Jordan. What is local optimality in nonconvex-nonconcave minimax optimization? In *International Conference on Machine Learning*, pages 4880–4889. PMLR, 2020.

[30] Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in Neural Information Processing Systems*, 12, 1999.

[31] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[32] Jason D. Lee, Ioannis Panageas, Georgios Piliouras, Max Simchowitz, Michael I. Jordan, and Benjamin Recht. First-order methods almost always avoid strict saddle points. *Mathematical Programming*, 176:311–337, 2019.

[33] Huan Li and Zhouchen Lin. Restarted nonconvex accelerated gradient descent: No more polylogarithmic factor in the $O(\epsilon^{-7/4})$ complexity. In *International Conference on Machine Learning*, pages 12901–12916. PMLR, 2022.

[34] Tianyi Lin, Chenyou Fan, Nhat Ho, Marco Cuturi, and Michael I. Jordan. Projection robust Wasserstein distance and Riemannian optimization. *Advances in Neural Information Processing Systems*, 33:9383–9397, 2020.

[35] Tianyi Lin, Chi Jin, and Michael I. Jordan. On gradient descent ascent for nonconvex-concave minimax problems. In *International Conference on Machine Learning*, pages 6083–6093. PMLR, 2020.

[36] Songtao Lu, Ioannis Tsaknakis, Mingyi Hong, and Yongxin Chen. Hybrid block successive approximation for one-sided non-convex min-max problems: algorithms and applications. *IEEE Transactions on Signal Processing*, 68:3676–3691, 2020.

[37] Luo Luo, Haishan Ye, Zhichao Huang, and Tong Zhang. Stochastic recursive gradient descent ascent for stochastic nonconvex-strongly-concave minimax problems. *Advances in Neural Information Processing Systems*, 33:20566–20577, 2020.

[38] Luo Luo, Yujun Li, and Cheng Chen. Finding second-order stationary points in nonconvex-strongly-concave minimax optimization. In *Advances in Neural Information Processing Systems*, 2022.

[39] Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87. Springer Science & Business Media, 2013.

[40] Yurii Nesterov and Boris T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

[41] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

[42] Maher Nouiehed, Maziar Sanjabi, Tianjian Huang, Jason D. Lee, and Meisam Razaviyayn. Solving a class of non-convex min-max games using iterative first order methods. *Advances in Neural Information Processing Systems*, 32, 2019.

[43] Walter Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, New York, 1976.

[44] Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. Truncated back-propagation for bilevel optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1723–1732. PMLR, 2019.

[45] Aman Sinha, Hongseok Namkoong, Riccardo Volpi, and John Duchi. Certifying some distributional robustness with principled adversarial training. In *International Conference on Learning Representations*, 2018.

[46] Bradly Stadie, Lunjun Zhang, and Jimmy Ba. Learning intrinsic rewards as a bi-level optimization problem. In *Conference on Uncertainty in Artificial Intelligence*, pages 111–120. PMLR, 2020.

[47] Nilesh Tripuraneni, Mitchell Stern, Chi Jin, Jeffrey Regier, and Michael I. Jordan. Stochastic cubic regularization for fast nonconvex optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

[48] Jiawei Zhang, Peijun Xiao, Ruoyu Sun, and Zhi-Quan Luo. A single-loop smoothed gradient descent-ascent algorithm for nonconvex-concave min-max problems. *Advances in Neural Information Processing Systems*, 33:7377–7389, 2020.

Appendix A. Overview and Contributions

We provide the presentation of contributions, related work, technical analysis and additional experiments in this supplementary materials.

A.1. Contributions

Our contributions are four-fold:

(i) We propose a method that we refer to as *Restarted Accelerated Hypergradient Descent* (RAHGD) that applies Nesterov's *accelerated gradient descent* (AGD) to approximate the solution $y^*(x)$ of the inner problem (1b) and combines it with the *conjugate gradient* (CG) method to construct an inexact hypergradient of the objective. The algorithm makes use of proper restarting and acceleration to optimize the objective $\Phi(\cdot)$ based on the obtained inexact hypergradient. We show that RAHGD can find an ϵ-FOSP of the objective within $\mathcal{O}(\kappa^{3.25}\epsilon^{-1.75})$ first-order oracle queries [§3].

(ii) For the task of finding approximate second-order stationary points, we add a perturbation step to RAHGD and introduce the *Perturbed Restarted Accelerated HyperGradient Descent* (PRAHGD) algorithm. We show that PRAHGD can efficiently escape saddle points and find an $\big(\epsilon, \mathcal{O}(\kappa^{2.5}\sqrt{\epsilon}\,)\big)$-second-order stationary point of the objective Φ within $\tilde{\mathcal{O}}\big(\kappa^{3.25}\epsilon^{-1.75}\big)$ oracle queries. This improves over the best known complexity in bilevel optimization due to Huang et al. [23] by a factor of $\tilde{\mathcal{O}}(\kappa^{0.75}\epsilon^{-0.25})$ [§4].

(iii) We apply the theoretical framework of PRAHGD to the problem of minimax optimization. Specially, we propose a PRAHGD variant crafted for nonconvex-strongly-concave minimax optimization. We refer to the resulting algorithm as *Perturbed Restarted Accelerate Gradient Descent Ascent* (PRAGDA). We show that PRAGDA provably finds an $\mathcal{O}\big(\epsilon, \mathcal{O}(\kappa^{1.5}\sqrt{\epsilon}\,)\big)$-SOSP with a first-order oracle query complexity of $\tilde{\mathcal{O}}(\kappa^{1.75}\epsilon^{-1.75})$. This improves upon the best known first-order (including gradient/Hessian-vector/Jacobian-vector-product) oracle query complexity bound of $\tilde{\mathcal{O}}(\kappa^{1.5}\epsilon^{-2} + \kappa^{2}\epsilon^{-1.5})$ due to Luo et al. [38] [§B].

(iv) We conduct a variety of empirical studies of bilevel optimization. Specifically, we evaluate the effectiveness of our proposed algorithms (RAHGD / PRAHGD / PRAGDA) by applying them to three different tasks: data hypercleaning for the MNIST dataset, hyperparameter optimization for logistic regression and a synthetic minimax problem. Our studies demonstrate that our algorithms outperform several established baseline algorithms, such as BA, AID-BiO, ITD-BiO, PAID-BiO and iMCN, with inevitably faster empirical convergence. The results provide empirical evidence in support of the effectiveness of our proposed algorithmic framework for bilevel and minimax optimization [§G].

A.2. Overview of Our Algorithm Design and Main Techniques

We overview the algorithm design in this subsection. Inspired by the success of the accelerated gradient descent method for nonconvex optimization [see, e.g., 28, 33], we propose a novel method called the *restarted accelerated hypergradient descent* (RAHGD) algorithm. The gradient of $\Phi(x)$, which we called a *hypergradient*, can be computed via following equation [17, 25]:

$$\nabla\Phi(x) = \nabla_x f(x, y^*(x)) - \nabla_{xy}^2 g(x, y^*(x))\big(\nabla_{yy}^2 g(x, y^*(x))\big)^{-1}\nabla_y f(x, y^*(x)). \tag{5}$$

Unfortunately, directly applying first-order algorithms by iterating with the exact hypergradient $\nabla\Phi(x)$ is costly or intractable for large-scale problems, given the need to obtain $y^*(x)$ and particularly given the need to invert the matrix $\nabla^2_{yy}g(x, y^*(x))$.

For given $x = x_k \in \mathbb{R}^{d_x}$, we aim to construct an estimate of $\nabla\Phi(x_k)$ with reasonable computational cost and sufficient accuracy. The strong convexity of $g(x_k, \cdot)$ motivates us to apply AGD for finding $y_k \approx y^*(x_k)$. To avoid direct computation of the term $\left(\nabla^2_{yy}g(x_k, y_k)\right)^{-1}\nabla_y f(x_k, y_k)$, we observe that it is the solution of the following quadratic problem:

$$\min_{v\in\mathbb{R}^{d_y}} \ \frac{1}{2}v^\top\nabla^2_{yy}g(x_k, y_k)v - v^\top\nabla_y f(x_k, y_k). \tag{6}$$

Accordingly, we can estimate $v_k \approx \left(\nabla^2_{yy}g(x_k, y_k)\right)^{-1}\nabla_y f(x_k, y_k)$ and solve (6) using a conjugate gradient subroutine. Based on y_k and v_k, we obtain an expression for an *inexact hypergradient*:

$$\hat{\nabla}\Phi(x_k) = \nabla_x f(x_k, y_k) - \nabla^2_{xy}g(x_k, y_k)v_k, \tag{7}$$

which can serve as a surrogate of the true hypergradient $\nabla\Phi(x_k)$ in first-order algorithms.

We formally present `RAHGD` in Algorithm 3. The main issue to address for `RAHGD` is the computational cost for achieving sufficient accuracy of $\hat{\nabla}\Phi(x_k)$. Interestingly, our theoretical analysis shows that all of the cost arises from the computations of y_k and v_k, and it can thus be bounded sharply. As a result, our algorithm can find approximate first-order stationary points with less oracle complexity than existing methods [17, 25]. We also introduce the *perturbed* `RAHGD` (`PRAHGD`) in Algorithm 3 for escaping saddle points. Adapting the analysis of `RAHGD`, we show that `PRAHGD` can find approximate second-order stationary points more efficiently than existing methods [23].

A.3. Related Work

The subject of bilevel optimization problem has a long history with early work tracing back to the 1970s [5]. Recent algorithmic advances in this field have driven successful applications in areas such as meta-learning [3, 16, 24], reinforcement learning [21, 30, 46] and hyperparameter optimization [13, 20, 44].

There have also been theoretical advances in bilevel optimization in recent years. Ghadimi and Wang [17] presented a convergence rate for the AID approach when $f(x, y)$ is convex, analyzing the complexity of an accelerated algorithm that uses gradient descent to approximate $y^*(x_k)$ in the inner loop and uses AGD in the outer loop. Further improvements in dependence on the condition number and analysis of the convergence were achieved via the *iterative differentiation* (ITD) approach by Ji et al. [25, 26], who analyzed the complexity of AID and ITD and also provided a complexicity analysis for a randomized version. Hong et al. [21] proposed the TTSA algorithm—a provable single-loop algorithm that updates two variables in an alternating manner—and presented applications to the problem of reinforcement learning under randomized scenarios. For stochastic bilevel problems, various methods have been proposed, such as BSA by Ghadimi and Wang [17], TTSA by Hong et al. [21], stocBiO by Ji et al. [25], and ALSET by Chen et al. [9]. More recent research on this front has focused on variance reduction and momentum techniques, resulting in cutting-edge stochastic first-order oracle complexities.

While much of the literature on bilevel optimization has focused on finding first-order stationary points, the problem of finding second-order stationary points has been largely unadressed. Huang et al.

[23] recently proposed a perturbed algorithm for finding approximate second-order stationary points. The algorithm adopts gradient descent (GD) to approximately solve the lower-level minimization problem and conjugate gradient (CG) to solve for Hessian-vector product with GD used in the outer loop. For the problem of classical optimization, second-order methods such as those proposed in Curtis et al. [11], Nesterov and Polyak [40] have been used to obtain ϵ-accurate SOSPs in single-level optimization with a complexity of $\mathcal{O}(\epsilon^{-1.5})$; however, they require expensive operations such as inverting Hessian matrices. A significant body of recent literature has been focusing on first-order methods for obtaining an approximate $\big(\epsilon, \mathcal{O}(\kappa^{2.5}\sqrt{\epsilon}\,)\big)$-SOSP, with the best-known query complexity of $\tilde{\mathcal{O}}(\epsilon^{-1.75})$ of gradient and Hessian-vector products [1, 6, 7, 27, 28, 33].

An important special case of the bilevel optimization problem (1)—the problem of minimax optimization, where $g = -f$ in Eq. (1b)—has been extensively studied in the literature. Minimax optimization has been the focus of attention in the machine learning community recently due to its applications to training GANs [2, 18], to adversarial learning [19, 45] and to optimal transport [22, 34]. On the theoretically font, Jin et al. [29], Nouiehed et al. [42] studied the complexity of Multistep Gradient Descent Ascent (GDmax), and Lin et al. [35], Lu et al. [36] provided the first convergence analysis for the single-loop *gradient descent ascent* (GDA) algorithm. More recently, Luo et al. [37] applied the stochastic variance reduction technique to the nonconvex-strongly-concave case, achieving the first optimal complexity upper bound when κ is treated as an $O(1)$-constant. Zhang et al. [48] proposed a stabilized smoothed GDA algorithm that achieves a better complexity for the nonconvex-concave problem. Fiez and Ratliff [14] provided asymptotic results showing that GDA converges to a local minimax point almost surely. Nevertheless, to the best of our knowledge, all the previous works targeted finding approximate stationary points of $\Phi(x)$, and the theory for finding the local minimax points is absent in the literature. It was not until very recently that Chen et al. [10], Luo et al. [38] independently proposed (inexact) cubic-regularized Newton methods for solving this problem; these are second-order algorithms that provably converge to a local minimax point. These algorithms are limited, however, to minimax optimization and they cannot be used to solve the more general bilevel optimization problems.

Appendix B. Improved Convergence for Accelerating Minimax Optimization

This section applies the ideas of `PRAHGD` to find approximate second-order stationary points in minimax optimization problem of the form

$$\min_{x\in\mathbb{R}^{d_x}} \left\{ \bar{\Phi}(x) \triangleq \max_{y\in\mathbb{R}^{d_y}} \bar{f}(x,y) \right\}, \tag{8}$$

where $\bar{f}(x,y)$ is strongly concave in y but possibly nonconvex in x. Problems of form (8) can be regarded as a special case of a bilevel optimization problem by taking $f(x,y) = \bar{f}(x,y)$ and $g(x,y) = -\bar{f}(x,y)$. Danskin's theorem yields $\nabla\bar{\Phi}(x) = \nabla_x \bar{f}(x, y^*(x))$, in this case, which is in fact consistent with hypergradient of form (5) with the optimality condition for the lower-level problem invoked, that is, $\nabla_y f(x, y^*(x)) = 0$. This implies that when applying `PRAHGD` to the minimax optimization problem (8), *no* CG subroutine is called and *no* Jacobian-vector or Hessian-vector product operation is invoked.

We first show in Lemma 19 that the minimax problem enjoys tighter Lipschitz continuouity parameters than the general bilevel problem:

Lemma 19 *Suppose that $\bar{f}(x,y)$ is ℓ-smooth, ρ-Hessian Lipschitz continuous with respect to x and y and μ-strongly concave in y but possibly nonconvex in x. Then the objective $\bar{\Phi}(x)$ is $(\kappa+1)\ell$-smooth and admits $(4\sqrt{2}\kappa^3\rho)$-Lipschitz continuous Hessians.*

We formally introduce the *perturbed restarted accelerated gradient descent ascent* (PRAGDA) as in Algorithm 4. Utilizing the PRAHGD complexity result as in Theorems 16 and 17 together with Lemma 19, we can take $\tilde{L} = (\kappa+1)\ell$ and $\tilde{\rho} = 4\sqrt{2}\kappa^3\rho$ to conclude an improved oracle complexity upper bounds for finding second-order stationary points for this particular problem, as in the following result:

Theorem 20 (PRAGDA finding SOSP) *Under the settings of Lemma 19, Algorithm 4 outputs an $\big(\epsilon, \mathcal{O}(\kappa^{1.5}\sqrt{\epsilon}\,)\big)$-second-order stationary point of $\bar{\Phi}(x)$ in equation (8) within $\tilde{\mathcal{O}}(\kappa^{1.75}\epsilon^{-1.75})$ gradient oracle calls.*

Prior to this work, the state-of-the-art algorithm was attained by the *inexact minimax cubic Newton* (iMCN) method [38], which under comparable settings outputs an $\big(\epsilon, \mathcal{O}(\kappa^{1.5}\sqrt{\epsilon}\,)\big)$-approximate SOSP within oracle quries of $\tilde{\mathcal{O}}(\kappa^2\epsilon^{-1.5})$ gradients, $\tilde{\mathcal{O}}(\kappa^{1.5}\epsilon^{-2})$ Hessian-vector products and $\tilde{\mathcal{O}}(\kappa\epsilon^{-2})$ Jacobian-vector products. We compare the query complexity upper bound of PRAGDA with iMCN in detail. As can be observed, the total oracle complexity of PRAGDA is no worse than that of iMCN since $\tilde{\mathcal{O}}(\kappa^{1.75}\epsilon^{-1.75}) \le \tilde{\mathcal{O}}(\kappa^2\epsilon^{-1.5} + \kappa^{1.5}\epsilon^{-2})$, a simple application of AM-GM inequality. Moreover, PRAGDA only requires gradient oracle calls while iMCN additionally requires Hessian-vector and Jacobian-vector oracle calls. To summarize, PRAGDA enjoys an oracle complexity that is no inferior than that of iMCN, whereas in both of the regimes $\kappa \gg \epsilon^{-1}$ and $\kappa \ll \epsilon^{-1}$ PRAGDA's complexity is strictly superior.

Appendix C. Basic Lemmas

In this section, we provide some basic lemmas.

Lemma 21 *Suppose Assumption 3 holds, then $y^*(x)$ is κ-Lipschitz continuous, that is,*

$$\left\|y^*(x) - y^*(x')\right\|_2 \le \kappa \left\|x - x'\right\|_2$$

for any $x, x' \in \mathbb{R}^{d_x}$.

Proof Recall that

$$y^*(x) = \arg\min_{y \in \mathbb{R}^{d_y}} g(x,y).$$

The optimality condition leads to $\nabla_y g(x, y^*(x)) = 0$ for each $x \in \mathbb{R}^{d_x}$. By taking a further derivative with respect to x on both sides and applying the chain rule [43], we obtain

$$\nabla^2_{yx} g(x, y^*(x)) + \nabla^2_{yy} g(x, y^*(x)) \frac{\partial y^*(x)}{\partial x} = 0.$$

The smoothness and strong convexity of g in y immediately indicate

$$\frac{\partial y^*(x)}{\partial x} = -(\nabla^2_{yy} g(x, y^*(x)))^{-1} \nabla^2_{yx} g(x, y^*(x)) .$$

Thus we have

$$\left\|\frac{\partial y^*(x)}{\partial x}\right\|_2 = \left\|(\nabla^2_{yy} g(x, y^*(x)))^{-1}\nabla^2_{yx} g(x, y^*(x))\right\|_2 \leq \frac{\ell}{\mu} = \kappa \ ,$$

where the inequality is based on the fact that $g(x, y)$ is ℓ-smooth with respect to x and y and μ-strongly convex with respect to y for any x.

Therefore, we proved that $y^*(x)$ is κ-Lipschitz continuous. ■

We also can show that $\Phi(x)$ admits Lipschitz continuous gradients and Lipschitz continuous Hessians, as shown in the following lemmas:

Lemma 22 *Suppose Assumption 3 holds, then $\Phi(x)$ is $\tilde{L}$-gradient Lipschitz continuous, that is,*

$$\|\nabla\Phi(x) - \nabla\Phi(x')\| \leq \tilde{L}\|x - x'\|$$

for any $x, x' \in \mathbb{R}^{d_x}$, where

$$\tilde{L} = \ell + \frac{2\ell^2 + \rho M}{\mu} + \frac{\ell^3 + 2\rho\ell M}{\mu^2} + \frac{\rho\ell^2 M}{\mu^3} \ .$$

Proof Recall that

$$\nabla\Phi(x) = \nabla_x f(x, y^*(x)) - \nabla^2_{xy} g(x, y^*(x))\big(\nabla^2_{yy} g(x, y^*(x))\big)^{-1}\nabla_y f(x, y^*(x)) \ .$$

We denote $\mathcal{H}_1(x) = \nabla_x f(x, y^*(x))$, $\mathcal{H}_2(x) = \nabla^2_{xy} g(x, y^*(x))$, $\mathcal{H}_3(x) = \big(\nabla^2_{yy} g(x, y^*(x))\big)^{-1}$ and $\mathcal{H}_4(x) = \nabla_y f(x, y^*(x))$, then

$$\nabla\Phi(x) = \mathcal{H}_1(x) - \mathcal{H}_2(x)\mathcal{H}_3(x)\mathcal{H}_4(x) \ .$$

We first consider $\mathcal{H}_1(x),\ \mathcal{H}_2(x)$ and $\mathcal{H}_4(x)$. For any $x,\ x' \in \mathbb{R}^{d_x}$, we have

$$\begin{aligned}\|\mathcal{H}_1(x) - \mathcal{H}_1(x')\| &\leq \ell(\|x - x'\| + \|y^*(x) - y^*(x')\|)\\ &\leq \ell(1+\kappa)\|x - x'\| \ ,\end{aligned}$$

where we use triangle inequality in the first inequality and Lemma 22 in the second one.

We also have

$$\begin{aligned}\|\mathcal{H}_2(x) - \mathcal{H}_2(x')\| &\leq \rho(\|x - x'\| + \|y^*(x) - y^*(x')\|)\\ &\leq \rho(1+\kappa)\|x - x'\|\end{aligned}$$

and

$$\begin{aligned}\|\mathcal{H}_4(x) - \mathcal{H}_4(x')\| &\leq \ell(\|x - x'\| + \|y^*(x) - y^*(x')\|)\\ &\leq \ell(1+\kappa)\|x - x'\|.\end{aligned}$$

We then consider $\mathcal{H}_3(x)$. For any $x,\ x' \in \mathbb{R}^{d_x}$, we have

$$\|\mathcal{H}_3(x) - \mathcal{H}_3(x')\|$$

$$
\begin{aligned}
&= \left\| \left(\nabla_{yy}^2 g(x, y^*(x))\right)^{-1} - \left(\nabla_{yy}^2 g(x', y^*(x'))\right)^{-1} \right\| \\
&\le \left\| \left(\nabla_{yy}^2 g(x, y^*(x))\right)^{-1} \right\| \left\| \nabla_{yy}^2 g(x', y^*(x')) - \nabla_{yy}^2 g(x, y^*(x)) \right\| \left\| \left(\nabla_{yy}^2 g(x', y^*(x'))\right)^{-1} \right\| \\
&\le \frac{1}{\mu^2} \rho(\|x - x'\| + \|y^*(x) - y^*(x')\|) \\
&\le \frac{\rho(1+\kappa)}{\mu^2} \|x - x'\| \, .
\end{aligned}
$$

We also have

$$
\|\mathcal{H}_2(x)\| \le \ell, \quad \|\mathcal{H}_3(x)\| \le \frac{1}{\mu} \quad \text{and} \quad \|\mathcal{H}_4(x)\| \le M.
$$

for any $x \in \mathbb{R}^{d_x}$. Then for any $x, \ x' \in \mathbb{R}^{d_x}$ we have

$$
\begin{aligned}
&\|\nabla \Phi(x) - \nabla \Phi(x')\| \\
&\le \|\mathcal{H}_1(x) - \mathcal{H}_1(x')\| + \|\mathcal{H}_2(x)\mathcal{H}_3(x)\mathcal{H}_4(x) - \mathcal{H}_2(x')\mathcal{H}_3(x')\mathcal{H}_4(x')\| \\
&\le \ell(1+\kappa)\|x - x'\| + \|\mathcal{H}_2(x)\mathcal{H}_3(x)\mathcal{H}_4(x) - \mathcal{H}_2(x)\mathcal{H}_3(x)\mathcal{H}_4(x')\| \\
&\quad + \|\mathcal{H}_2(x)\mathcal{H}_3(x)\mathcal{H}_4(x') - \mathcal{H}_2(x)\mathcal{H}_3(x')\mathcal{H}_4(x')\| \\
&\quad + \|\mathcal{H}_2(x)\mathcal{H}_3(x')\mathcal{H}_4(x') - \mathcal{H}_2(x')\mathcal{H}_3(x')\mathcal{H}_4(x')\| \\
&\le \ell(1+\kappa)\|x - x'\| + \|\mathcal{H}_2(x)\|\|\mathcal{H}_3(x)\|\|\mathcal{H}_4(x) - \mathcal{H}_4(x')\| \\
&\quad + \|\mathcal{H}_2(x)\|\|\mathcal{H}_4(x')\|\|\mathcal{H}_3(x) - \mathcal{H}_3(x')\| \\
&\quad + \|\mathcal{H}_3(x')\|\|\mathcal{H}_4(x')\|\|\mathcal{H}_2(x) - \mathcal{H}_2(x')\| \\
&\le \ell(1+\kappa)\|x - x'\| + \frac{\ell^2}{\mu}(1+\kappa)\|x - x'\| + \frac{\ell\rho M}{\mu^2}(1+\kappa)\|x - x'\| + \frac{M\rho}{\mu}(1+\kappa)\|x - x'\| \\
&= \left(\ell + \frac{2\ell^2 + \rho M}{\mu} + \frac{\ell^3 + 2\rho\ell M}{\mu^2} + \frac{\rho\ell^2 M}{\mu^3}\right)\|x - x'\| \, .
\end{aligned}
$$

■

Lemma 23 ([23, Lemma 3.4]). *Suppose Assumption 3 holds, then $\Phi(x)$ is $\tilde{\rho}$-Hessian Lipschitz continuous, that is, $\|\nabla^2\Phi(x) - \nabla^2\Phi(x')\| \le \tilde{\rho}\|x - x'\|$ for any $x, x' \in \mathbb{R}^{d_x}$, where*

$$
\begin{aligned}
\tilde{\rho} = \Bigg[&\left(\rho + \frac{2\ell\rho + M\nu}{\mu} + \frac{2M\ell\nu + \rho\ell^2}{\mu^2} + \frac{M\ell^2\nu}{\mu^3}\right)\left(1 + \frac{\ell}{\mu}\right) \\
&+ \left(\frac{2\ell\rho}{\mu} + \frac{4M\rho^2 + 2\ell^2\rho}{\mu^2} + \frac{2M\ell\rho^2}{\mu^3}\right)\left(1 + \frac{\ell}{\mu}\right)^2 + \left(\frac{M\rho^2}{\mu^2} + \frac{\rho\ell}{\mu}\right)\left(1 + \frac{\ell}{\mu}\right)^3 \Bigg] \, .
\end{aligned}
$$

Lemma 24 (Inexact gradients) *Suppose Assumption 3 and Condition 10 hold, then we have*

$$
\|\nabla\Phi(w_k) - \hat{\nabla}\Phi(w_k)\|_2 \le \sigma \, .
$$

Proof Recall that

$$
\nabla\Phi(x) = \nabla_x f(x, y^*(x)) - \nabla_{xy}^2 g(x, y^*(x))\left(\nabla_{yy}^2 g(x, y^*(x))\right)^{-1}\nabla_y f(x, y^*(x))
$$

and

$$\hat{\nabla}\Phi(x_k) = \nabla_x f(x_k, y_k) - \nabla^2_{xy} g(x_k, y_k) v_k \; .$$

We define

$$\bar{\nabla}\Phi(x_k) = \nabla_x f(x_k, y_k) - \nabla^2_{xy} g(x_k, y_k) \big(\nabla^2_{yy} g(x_k, y_k)\big)^{-1} \nabla_y f(x_k, y_k) \; ,$$

then we have

$$\begin{aligned} ||\nabla\Phi(w_k) - \hat{\nabla}\Phi(w_k)||_2 &= ||\nabla\Phi(w_k) - \bar{\nabla}\Phi(w_k) + \bar{\nabla}\Phi(w_k) - \hat{\nabla}\Phi(w_k)||_2 \\ &\le ||\nabla\Phi(w_k) - \bar{\nabla}\Phi(w_k)||_2 + ||\bar{\nabla}\Phi(w_k) - \hat{\nabla}\Phi(w_k)||_2 \\ &\le \tilde{L}||y_k - y^*(w_k)||_2 + \ell \left|\left| v_k - \big(\nabla^2_{yy} g(w_k, y_k)\big)^{-1} \nabla_y f(w_k, y_k) \right|\right|_2 \\ &\le \sigma \; , \end{aligned}$$

where we use triangle inequality in the first inequality, Lemma 6 and Assumption 3(c) in the second inequality and Condition 10 in the last one. ■

Lemma 25 ([38, Lemmas 1 and 3]). *Assume that $\bar{f}(x, y)$ is ℓ-smooth, ρ-Hessian Lipschitz continuous with respect to x and y and μ-strongly concave in y but possibly nonconvex in x, then the objective $\bar{\Phi}(x)$ is $(\kappa + 1)\ell$-smooth and $(4\sqrt{2}\kappa^3\rho)$-Hessian Lipschitz continuous.*

Appendix D. Proofs for Section 3

In this section, we provide the proofs for theorems in Section 3. We separate our proof into three parts. We first prove that $\Phi(x)$ decrease at least $\mathcal{O}(\epsilon^{3/2})$ in one epoch and thus the total number of epochs is bounded. Then we show that our `RAHGD` in Algorithm 3 can output an ϵ-FOSP. Finally, we provide the oracle calls complexity analysis.

D.1. Proof of Theorem 13

In this section, we mainly consider the progress in one epoch. We omit the subscript t for notation simplicity. For each epoch except the last one, we have $1 \le \mathcal{K} \le K$,

$$\mathcal{K} \sum_{i=0}^{\mathcal{K}-1} \|x_{i-1} - x_i\|_2^2 > B^2, \tag{9}$$

$$\|x_k - x_0\|_2^2 \le k \sum_{i=0}^{k-1} \|x_{i+1} - x_i\|_2^2 \le B^2, \qquad \forall k < \mathcal{K}, \tag{10}$$

$$\|w_k - x_0\|_2 \le \|x_k - x_0\|_2 + \|x_k - x_{k-1}\|_2 \le 2B, \qquad \forall k < \mathcal{K}, \tag{11}$$

$$\|w_k - w_{k-1}\|_2 \le 2B, \qquad \forall k < \mathcal{K}, \tag{12}$$

where equation (12) can be proved by induction as follows. For $k = 0$, we have

$$\|w_0 - w_{-1}\|_2 = 0 \le 2B.$$

For $k = 1$, we have

$$\|w_1 - w_0\|_2 = \|(x_1 - x_0) + (1-\theta)(x_1 - x_0)\|_2 \le 2B.$$

For $k \ge 2$, we have

$$\begin{aligned}\|w_k - w_{k-1}\|_2 &\le (2-\theta)\|x_k - x_{k-1}\|_2 + (1-\theta)\|x_{k-1} - x_{k-2}\|_2 \\ &\le 2\sqrt{2\|x_k - x_{k-1}\|_2^2 + 2\|x_{k-1} - x_{k-2}\|_2^2} \le 2B.\end{aligned}$$

In the last epoch, the "if condition" does not trigger and the while loop breaks until $k = K$. Hence, we have

$$\|x_k - x_0\|_2^2 \le k \sum_{i=0}^{k-1} \|x_{i+1} - x_i\|_2^2 \le B^2, \qquad \forall k \le K, \tag{13}$$

$$\|w_k - x_0\|_2 \le 2B, \qquad \forall k \le K. \tag{14}$$

We first consider the case when $\|\nabla\Phi(w_{\mathcal{K}-1})\|_2$ is large in the following lemma.

Lemma 26 *Suppose that Assumption 3 and Condition 10 hold. Let $\eta \le \frac{1}{4\tilde{L}}$ and $0 \le \theta \le 1$. When the "if condition" triggers and $\|\nabla\Phi(w_{\mathcal{K}-1})\|_2 > \frac{B}{\eta}$, we have*

$$\Phi(x_{\mathcal{K}}) - \Phi(x_0) \le -\frac{B^2}{4\eta} + \sigma B + \frac{5\eta\sigma^2\mathcal{K}}{8}.$$

Proof Since $\Phi(x)$ has $\tilde{L}$-Lipschitz continuous gradient, we have

$$\begin{aligned}\Phi(x_{k+1}) &\le \Phi(w_k) + \langle \nabla\Phi(w_k), x_{k+1} - w_k\rangle + \frac{\tilde{L}}{2}\|x_{k+1} - w_k\|_2^2 \\ &\le \Phi(w_k) - \eta\langle \nabla\Phi(w_k), \hat{\nabla}\Phi(w_k)\rangle + \frac{\eta}{8}\|\hat{\nabla}\Phi(w_k)\|_2^2,\end{aligned}$$

where we use $\eta \le \frac{\tilde{L}}{4}$. We also have

$$\Phi(x_k) \ge \Phi(w_k) + \langle \nabla\Phi(w_k), x_k - w_k\rangle - \frac{\tilde{L}}{2}\|x_k - w_k\|_2^2.$$

Combining above inequalities leads to

$$\begin{aligned}&\Phi(x_{k+1}) - \Phi(x_k) \\ &\le -\langle \nabla\Phi(w_k), x_k - w_k\rangle + \frac{\tilde{L}}{2}\|x_k - w_k\|_2^2 - \eta\langle \nabla\Phi(w_k), \hat{\nabla}\Phi(w_k)\rangle + \frac{\eta}{8}\|\hat{\nabla}\Phi(w_k)\|_2^2 \\ &= \frac{1}{\eta}\langle x_{k+1} - w_k, x_k - w_k\rangle + \langle \hat{\nabla}\Phi(w_k) - \nabla\Phi(w_k), x_k - w_k\rangle + \frac{\tilde{L}}{2}\|x_k - w_k\|_2^2 \\ &\qquad - \eta\langle \nabla\Phi(w_k), \hat{\nabla}\Phi(w_k)\rangle + \frac{\eta}{8}\|\hat{\nabla}\Phi(w_k)\|_2^2 \\ &= \frac{1}{2\eta}\left(\|x_{k+1} - w_k\|_2^2 + \|x_k - w_k\|_2^2 - \|x_{k+1} - x_k\|_2^2\right) + \langle \hat{\nabla}\Phi(w_k) - \nabla\Phi(w_k), x_k - w_k\rangle\end{aligned}$$

$$
\begin{aligned}
&\quad + \frac{\tilde{L}}{2}\|x_k - w_k\|_2^2 - \eta\langle\nabla\Phi(w_k), \hat{\nabla}\Phi(w_k)\rangle + \frac{\eta}{8}\|\hat{\nabla}\Phi(w_k)\|_2^2 \\
&\overset{(a)}{\leq} \frac{5}{8\eta}\|x_k - w_k\|_2^2 - \frac{1}{2\eta}\|x_{k+1} - x_k\|_2^2 + \langle\hat{\nabla}\Phi(w_k) - \nabla\Phi(w_k), x_k - w_k\rangle + \frac{5\eta}{8}\|\hat{\nabla}\Phi(w_k)\|_2^2 \\
&\quad - \eta\langle\nabla\Phi(w_k), \hat{\nabla}\Phi(w_k)\rangle \\
&\overset{(b)}{\leq} \frac{5}{8\eta}\|x_k - x_{k-1}\|_2^2 - \frac{1}{2\eta}\|x_{k+1} - x_k\|_2^2 + \|\hat{\nabla}\Phi(w_k) - \nabla\Phi(w_k)\|_2 \cdot \|x_k - x_{k-1}\|_2 \\
&\quad + \frac{5\eta}{8}\|\hat{\nabla}\Phi(w_k)\|_2^2 - \eta\langle\nabla\Phi(w_k), \hat{\nabla}\Phi(w_k)\rangle \\
&= \frac{5}{8\eta}\|x_k - x_{k-1}\|_2^2 - \frac{1}{2\eta}\|x_{k+1} - x_k\|_2^2 + \|\hat{\nabla}\Phi(w_k) - \nabla\Phi(w_k)\|_2 \cdot \|x_k - x_{k-1}\|_2 \\
&\quad + \frac{5\eta}{8}\|\hat{\nabla}\Phi(w_k)\|_2^2 - \frac{\eta}{2}\left(\|\nabla\Phi(w_k)\|_2^2 + \left\|\hat{\nabla}\Phi(w_k)\right\|_2^2 - \left\|\nabla\Phi(w_k) - \hat{\nabla}\Phi(w_k)\right\|_2^2\right) \\
&\overset{(c)}{\leq} \frac{5}{8\eta}\|x_k - x_{k-1}\|_2^2 - \frac{1}{2\eta}\|x_{k+1} - x_k\|_2^2 + \|\hat{\nabla}\Phi(w_k) - \nabla\Phi(w_k)\|_2 \cdot \|x_k - x_{k-1}\|_2 \\
&\quad - \frac{3\eta}{8}\|\nabla\Phi(w_k)\|_2^2 + \frac{5\eta}{8}\left\|\nabla\Phi(w_k) - \hat{\nabla}\Phi(w_k)\right\|_2^2 \\
&\overset{(d)}{\leq} \frac{5}{8\eta}\|x_k - x_{k-1}\|_2^2 - \frac{1}{2\eta}\|x_{k+1} - x_k\|_2^2 - \frac{3\eta}{8}\|\nabla\Phi(w_k)\|_2^2 + \sigma\|x_k - x_{k-1}\|_2 + \frac{5\eta}{8}\sigma^2\,,
\end{aligned}
$$

where we use $\tilde{L} \leq \frac{1}{4\eta}$ in $\overset{(a)}{\leq}$, $\|x_k - w_k\|_2 = (1-\theta)\|x_k - x_{k-1}\|_2 \leq \|x_k - x_{k-1}\|_2$ in $\overset{(b)}{\leq}$, triangle inequality in $\overset{(c)}{\leq}$ and Lemma 12 in $\overset{(d)}{\leq}$.

Summing over above inequality with $k = 0, 1, \cdots, \mathcal{K}-1$ and using $x_0 = x_{-1}$, we have

$$
\begin{aligned}
&\Phi(x_{\mathcal{K}}) - \Phi(x_0) && (15)\\
&\leq \frac{1}{8\eta}\sum_{k=0}^{\mathcal{K}-2}\|x_{k+1} - x_k\|_2^2 - \frac{3\eta}{8}\sum_{k=0}^{\mathcal{K}-1}\|\nabla\Phi(w_k)\|_2^2 + \sigma\sum_{k=0}^{\mathcal{K}-1}\|x_k - x_{k-1}\|_2 + \frac{5\eta\sigma^2\mathcal{K}}{8} && (16)\\
&\overset{(e)}{\leq} \frac{1}{8\eta}\sum_{k=0}^{\mathcal{K}-2}\|x_{k+1} - x_k\|_2^2 - \frac{3\eta}{8}\sum_{k=0}^{\mathcal{K}-1}\|\nabla\Phi(w_k)\|_2^2 + \sigma\sqrt{\mathcal{K}-1}\sqrt{\sum_{k=0}^{\mathcal{K}-2}\|x_{k+1} - x_k\|_2^2} + \frac{5\eta\sigma^2\mathcal{K}}{8} && (17)\\
&\overset{(f)}{\leq} \frac{B^2}{8\eta} - \frac{3\eta}{8}\|\nabla\Phi(w_{\mathcal{K}-1})\|_2^2 + \sigma B + \frac{5\eta\sigma^2\mathcal{K}}{8} && (18)\\
&\overset{(g)}{\leq} -\frac{B^2}{4\eta} + \sigma B + \frac{5\eta\sigma^2\mathcal{K}}{8}\,, && (19)
\end{aligned}
$$

where we use Cauchy-Schwarz inequality in $\overset{(e)}{\leq}$, the "if condition" in $\overset{(f)}{\leq}$ and $\|\nabla\Phi(w_{\mathcal{K}-1})\|_2 > \frac{B}{\eta}$ in $\overset{(g)}{\leq}$. ■

Now we consider the case when $\|\nabla\Phi(w_{\mathcal{K}-1})\|_2$ is small.

If $\|\nabla\Phi(w_{\mathcal{K}-1})\|_2 \leq \frac{B}{\eta}$, then equation (11) and Lemma 12 lead to

$$\begin{aligned}\|x_{\mathcal{K}} - x_0\|_2 &\leq \|w_{\mathcal{K}-1} - x_0\|_2 + \eta\,\|\nabla\Phi(w_{\mathcal{K}-1})\|_2 + \eta\left\|\hat{\nabla}\Phi(w_{\mathcal{K}-1}) - \nabla\Phi(w_{\mathcal{K}-1})\right\|_2 \\ &\leq 3B + \eta\sigma.\end{aligned}$$

For each epoch, we denote $\mathbf{H} = \nabla^2\Phi(x_0)$ and $\mathbf{H} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ to be its eigenvalue decomposition with orthogonal matrix $\mathbf{\Lambda} \in \mathbb{R}^{d_x \times d_x}$ and diagonal matrix $\mathbf{U} \in \mathbb{R}^{d_x \times d_x}$. Let λ_j be the j-th eigenvalue of $\mathbf{H}$. We denote $\tilde{x} = \mathbf{U}^T x$, $\tilde{w} = \mathbf{U}^T w$ and $\tilde{\nabla}\Phi(w) = \mathbf{U}^T\nabla\Phi(w)$. Let $\tilde{x}^j$ and $\tilde{\nabla}^j\Phi(w)$ be the j-th coordinate of $\tilde{x}$ and $\tilde{\nabla}\Phi(w)$, respectively. Since Φ has $\tilde{\rho}$-Lipschitz continuous Hessian, we have

$$\begin{aligned}\Phi(x_{\mathcal{K}}) - \Phi(x_0) &\leq \langle\nabla\Phi(x_0), x_{\mathcal{K}} - x_0\rangle + \frac{1}{2}(x_{\mathcal{K}} - x_0)^\top\mathbf{H}(x_{\mathcal{K}} - x_0) + \frac{\tilde{\rho}}{6}\|x_{\mathcal{K}} - x_0\|_2^3 \\ &= \langle\tilde{\nabla}\Phi(x_0), \tilde{x}_{\mathcal{K}} - \tilde{x}_0\rangle + \frac{1}{2}(\tilde{x}_{\mathcal{K}} - \tilde{x}_0)^\top\mathbf{\Lambda}(\tilde{x}_{\mathcal{K}} - \tilde{x}_0) + \frac{\tilde{\rho}}{6}\|x_{\mathcal{K}} - x_0\|_2^3 \\ &\leq \phi(\tilde{x}_{\mathcal{K}}) - \phi(\tilde{x}_0) + \frac{\tilde{\rho}}{6}(3B + \eta\sigma)^3,\end{aligned} \tag{20}$$

where we denote

$$\phi(x) = \langle\tilde{\nabla}\Phi(x_0), x - \tilde{x}_0\rangle + \frac{1}{2}(x - \tilde{x}_0)^\top\mathbf{\Lambda}(x - \tilde{x}_0)$$

and

$$\phi_j(x) = \langle\tilde{\nabla}^j\Phi(x_0), x - \tilde{x}_0^j\rangle + \frac{1}{2}\lambda_j(x - \tilde{x}_0^j)^2.$$

Let

$$\tilde{\delta}_k^j = (\mathbf{U}^\top\hat{\nabla}\Phi(w_k))^j - \nabla\phi_j(\tilde{w}_k^j) \quad \text{and} \quad \tilde{\delta}_k = \mathbf{U}^\top\hat{\nabla}\Phi(w_k) - \nabla\phi(\tilde{w}_k)\,,$$

then the iteration of the algorithm means

$$\tilde{w}_k^j = \tilde{x}_k^j + (1-\theta)(\tilde{x}_k^j - \tilde{x}_{k-1}^j), \tag{21}$$

and

$$\tilde{x}_{k+1}^j = \tilde{w}_k^j - \eta(\mathbf{U}^\top\hat{\nabla}\Phi(w_k))^j = \tilde{w}_k^j - \eta\nabla\phi_j(\tilde{w}_k^j) - \eta\tilde{\delta}_k^j. \tag{22}$$

For any $k < \mathcal{K}$, we can bound $\|\tilde{\delta}_k\|_2$ as follows

$$\begin{aligned}\|\tilde{\delta}_k\| &= \left\|\mathbf{U}^\top\hat{\nabla}\Phi(w_k) - \tilde{\nabla}\Phi(w_k) + \tilde{\nabla}\Phi(w_k) - \nabla\phi(\tilde{w}_k)\right\|_2 \\ &\leq \left\|\tilde{\nabla}\Phi(w_k) - \tilde{\nabla}\Phi(x_0) - \mathbf{\Lambda}(\tilde{w}_k - \tilde{x}_0)\right\|_2 + \left\|\mathbf{U}^\top\hat{\nabla}\Phi(w_k) - \tilde{\nabla}\Phi(w_k)\right\|_2 \\ &= \|\nabla\Phi(w_k) - \nabla\Phi(x_0) - \mathbf{H}(w_k - x_0)\|_2 + \left\|\hat{\nabla}\Phi(w_k) - \nabla\Phi(w_k)\right\|_2 \\ &\leq \left\|\int_0^1 \langle\nabla^2\Phi(x_0 + t(w_k - x_0)) - \mathbf{H}, w_k - x_0\rangle\,\mathrm{d}t\right\|_2 + \sigma \\ &\leq \frac{\tilde{\rho}}{2}\|w_k - x_0\|_2^2 + \sigma \\ &\leq 2\tilde{\rho}B^2 + \sigma,\end{aligned}$$

where the first inequality uses triangle inequality; the second one is based on Lemma 12; the third one is based on the Lipschitz continuity of Hessian and the last one uses equation (11).

Notice that quadratic function $\phi(x)$ equals to the sum of d_x scalar functions $\phi_j(x^j)$. Then we decompose $\phi(x)$ into $\sum_{j\in S_1}\phi_j(x^j)$ and $\sum_{j\in S_2}\phi_j(x^j)$, where

$$S_1 = \left\{j : \lambda_j \geq -\frac{\theta}{\eta}\right\} \qquad \text{and} \qquad S_2 = \left\{j : \lambda_j < -\frac{\theta}{\eta}\right\} .$$

We first consider the term $\sum_{j\in S_1}\phi_j(x^j)$ in the following lemma.

Lemma 27 *Suppose that Assumption 3 and Condition 10 hold. Let $\eta \leq \frac{1}{4\tilde{L}}$ and $0 \leq \theta \leq 1$. When the "if condition" triggers and $\|\nabla\Phi(w_{\mathcal{K}-1})\|_2 \leq \frac{B}{\eta}$, then we have*

$$\sum_{j\in S_1}\phi_j(\tilde{x}^j_{\mathcal{K}}) - \sum_{j\in S_1}\phi_j(\tilde{x}^j_0) \leq -\sum_{j\in S_1}\frac{3\theta}{8\eta}\sum_{k=0}^{\mathcal{K}-1}|\tilde{x}^j_{k+1}-\tilde{x}^j_k|^2 + \frac{2\eta\mathcal{K}}{\theta}(2\tilde{\rho}B^2+\sigma)^2 . \tag{23}$$

Proof Since $\phi_j(x)$ is quadratic, we have

$$\begin{aligned}
&\phi_j(\tilde{x}^j_{k+1}) \\
&= \phi_j(\tilde{x}^j_k) + \langle\nabla\phi_j(\tilde{x}^j_k), \tilde{x}^j_{k+1}-\tilde{x}^j_k\rangle + \frac{\lambda_j}{2}|\tilde{x}^j_{k+1}-\tilde{x}^j_k|^2 \\
&\overset{(a)}{=} \phi_j(\tilde{x}^j_k) - \frac{1}{\eta}\langle\tilde{x}^j_{k+1}-\tilde{w}^j_k+\eta\tilde{\delta}^j_k, \tilde{x}^j_{k+1}-\tilde{x}^j_k\rangle + \langle\nabla\phi_j(\tilde{x}^j_k)-\nabla\phi_j(\tilde{w}^j_k), \tilde{x}^j_{k+1}-\tilde{x}^j_k\rangle \\
&\qquad + \frac{\lambda_j}{2}|\tilde{x}^j_{k+1}-\tilde{x}^j_k|^2 \\
&= \phi_j(\tilde{x}^j_k) - \frac{1}{\eta}\langle\tilde{x}^j_{k+1}-\tilde{w}^j_k, \tilde{x}^j_{k+1}-\tilde{x}^j_k\rangle - \langle\tilde{\delta}^j_k, \tilde{x}^j_{k+1}-\tilde{x}^j_k\rangle + \lambda_j\langle\tilde{x}^j_k-\tilde{w}^j_k, \tilde{x}^j_{k+1}-\tilde{x}^j_k\rangle \\
&\qquad + \frac{\lambda_j}{2}|\tilde{x}^j_{k+1}-\tilde{x}^j_k|^2 \\
&= \phi_j(\tilde{x}^j_k) + \frac{1}{2\eta}\left(|\tilde{x}^j_k-\tilde{w}^j_k|^2 - |\tilde{x}^j_{k+1}-\tilde{w}^j_k|^2 - |\tilde{x}^j_{k+1}-\tilde{x}^j_k|^2\right) - \langle\tilde{\delta}^j_k, \tilde{x}^j_{k+1}-\tilde{x}^j_k\rangle \\
&\qquad + \frac{\lambda_j}{2}\left(|\tilde{x}^j_{k+1}-\tilde{w}^j_k|^2 - |\tilde{x}^j_k-\tilde{w}^j_k|^2\right) \\
&\leq \phi_j(\tilde{x}^j_k) + \frac{1}{2\eta}\left(|\tilde{x}^j_k-\tilde{w}^j_k|^2 - |\tilde{x}^j_{k+1}-\tilde{w}^j_k|^2 - |\tilde{x}^j_{k+1}-\tilde{x}^j_k|^2\right) + \frac{1}{2\alpha}|\tilde{\delta}^j_k|^2 + \frac{\alpha}{2}|\tilde{x}^j_{k+1}-\tilde{x}^j_k|^2 \\
&\qquad + \frac{\lambda_j}{2}\left(|\tilde{x}^j_{k+1}-\tilde{w}^j_k|^2 - |\tilde{x}^j_k-\tilde{w}^j_k|^2\right),
\end{aligned}$$

where we use equation (21) in $\overset{(a)}{=}$.

Using the fact $\tilde{L} \geq \lambda_j \geq -\frac{\theta}{\eta}$ for $j \in S_1$ and

$$\left(-\frac{1}{2\eta}+\frac{\lambda_j}{2}\right)|\tilde{x}^j_{k+1}-\tilde{w}^j_k|^2 \leq \left(-2\tilde{L}+\frac{\tilde{L}}{2}\right)|\tilde{x}^j_{k+1}-\tilde{w}^j_k|^2 \leq 0,$$

we have

$$
\begin{aligned}
&\phi_j(\tilde{x}_{k+1}^j) \\
&\leq \phi_j(\tilde{x}_k^j) + \frac{1}{2\eta}\left(|\tilde{x}_k^j - \tilde{w}_k^j|^2 - |\tilde{x}_{k+1}^j - \tilde{x}_k^j|^2\right) + \frac{1}{2\alpha}|\tilde{\delta}_k^j|^2 + \frac{\alpha}{2}|\tilde{x}_{k+1}^j - \tilde{x}_k^j|^2 + \frac{\theta}{2\eta}|\tilde{x}_k^j - \tilde{w}_k^j|^2 \\
&\overset{(b)}{=} \phi_j(\tilde{x}_k^j) + \frac{(1-\theta)^2(1+\theta)}{2\eta}|\tilde{x}_k^j - \tilde{x}_{k-1}^j|^2 - \left(\frac{1}{2\eta} - \frac{\alpha}{2}\right)|\tilde{x}_{k+1}^j - \tilde{x}_k^j|^2 + \frac{1}{2\alpha}|\tilde{\delta}_k^j|^2 \\
&= \phi_j(\tilde{x}_k^j) + \frac{(1-\theta)^2(1+\theta)}{2\eta}\left(|\tilde{x}_k^j - \tilde{x}_{k-1}^j|^2 - |\tilde{x}_{k+1}^j - \tilde{x}_k^j|^2\right) \\
&\qquad - \left(\frac{1}{2\eta} - \frac{\alpha}{2} - \frac{(1-\theta)^2(1+\theta)}{2\eta}\right)|\tilde{x}_{k+1}^j - \tilde{x}_k^j|^2 + \frac{1}{2\alpha}|\tilde{\delta}_k^j|^2 \\
&\overset{(c)}{\leq} \phi_j(\tilde{x}_k^j) + \frac{(1-\theta)^2(1+\theta)}{2\eta}\left(|\tilde{x}_k^j - \tilde{x}_{k-1}^j|^2 - |\tilde{x}_{k+1}^j - \tilde{x}_k^j|^2\right) - \frac{3\theta}{8\eta}|\tilde{x}_{k+1}^j - \tilde{x}_k^j|^2 + \frac{2\eta}{\theta}|\tilde{\delta}_k^j|^2
\end{aligned}
$$

for each $j \in S_1$, where we use equation (22) in $\overset{(b)}{=}$ and let $\alpha = \frac{\theta}{4\eta}$ in $\overset{(c)}{\leq}$ which leads to

$$
\frac{1}{2\eta} - \frac{\alpha}{2} - \frac{(1-\theta)^2(1+\theta)}{2\eta} = \frac{1}{2\eta} - \frac{\theta}{8\eta} - \frac{(1-\theta)^2(1+\theta)}{2\eta} = \frac{3\theta}{8\eta} + \frac{\theta^2 - \theta^3}{2\eta} \geq \frac{3\theta}{8\eta}.
$$

Summing over above result with $k = 0, 1, \cdots, \mathcal{K}-1$ for $j \in S_1$ and using $x_0 = x_{-1}$, we have

$$
\begin{aligned}
&\sum_{j\in S_1} \phi_j(\tilde{x}_{\mathcal{K}}^j) \\
&\leq \sum_{j\in S_1} \phi_j(\tilde{x}_0^j) - \sum_{j\in S_1} \frac{3\theta}{8\eta}\sum_{k=0}^{\mathcal{K}-1}|\tilde{x}_{k+1}^j - \tilde{x}_k^j|^2 + \frac{2\eta}{\theta}\sum_{k=0}^{\mathcal{K}-1}\left\|\tilde{\delta}_k\right\|_2^2 - \frac{(1-\theta)^2(1+\theta)}{2\eta}|\tilde{x}_{\mathcal{K}}^j - \tilde{x}_{\mathcal{K}-1}^j|^2 \\
&\leq \sum_{j\in S_1} \phi_j(\tilde{x}_0^j) - \sum_{j\in S_1} \frac{3\theta}{8\eta}\sum_{k=0}^{\mathcal{K}-1}|\tilde{x}_{k+1}^j - \tilde{x}_k^j| + \frac{2\eta\mathcal{K}}{\theta}(2\tilde{\rho}B^2 + \sigma)^2.
\end{aligned}
$$

This completes the proof. ■

Next, we consider the term $\sum_{j\in S_2} \phi_j(x^j)$.

Lemma 28 *Suppose that Assumption 3 and Condition 10 hold. Let $\eta \leq \frac{1}{4\tilde{L}}$ and $0 \leq \theta \leq 1$. When the "if condition" triggers and $\|\nabla\Phi(w_{\mathcal{K}-1})\|_2 \leq \frac{B}{\eta}$, then we have*

$$
\sum_{j\in S_2} \phi_j(\tilde{x}_{\mathcal{K}}^j) - \sum_{j\in S_2} \phi_j(\tilde{x}_0^j) \leq -\sum_{j\in S_2} \frac{\theta}{2\eta}\sum_{k=0}^{\mathcal{K}-1}|\tilde{x}_{k+1}^j - \tilde{x}_k^j|^2 + \frac{\eta\mathcal{K}}{2\theta}(2\tilde{\rho}B^2 + \sigma)^2 + \frac{\eta\mathcal{K}}{2\theta}\sigma^2 . \quad (24)
$$

Proof We denote $\nu_j = \tilde{x}_0^j - \frac{1}{\lambda_j}\tilde{\nabla}^j\Phi(x_0)$, then $\phi_j(x)$ can be rewritten as

$$
\phi_j(x) = \frac{\lambda_j}{2}\left(x - \tilde{x}_0^j + \frac{1}{\lambda_j}\tilde{\nabla}^j\Phi(x_0)\right)^2 - \frac{1}{2\lambda_j}|\tilde{\nabla}^j\Phi(x_0)|^2 = \frac{\lambda_j}{2}(x - \nu_j)^2 - \frac{1}{2\lambda_j}|\tilde{\nabla}^j\Phi(x_0)|^2 .
$$

For each $j \in S_2 = \{j : \lambda_j < -\frac{\theta}{\eta}\}$, we have

$$\begin{aligned}\phi_j(\tilde{x}_{k+1}^j) - \phi_j(\tilde{x}_k^j) &= \frac{\lambda_j}{2}|\tilde{x}_{k+1}^j - \nu_j|^2 - \frac{\lambda_j}{2}|\tilde{x}_k^j - \nu_j|^2 \\ &= \frac{\lambda_j}{2}|\tilde{x}_{k+1}^j - \tilde{x}_k^j|^2 + \lambda_j\langle \tilde{x}_{k+1}^j - \tilde{x}_k^j, \tilde{x}_k^j - \nu_j\rangle \\ &\le -\frac{\theta}{2\eta}|\tilde{x}_{k+1}^j - \tilde{x}_k^j|^2 + \lambda_j\langle \tilde{x}_{k+1}^j - \tilde{x}_k^j, \tilde{x}_k^j - \nu_j\rangle\ .\end{aligned} \tag{25}$$

So we only need to bound the second part. From equation (21) and equation (22), we have

$$\begin{aligned}\tilde{x}_{k+1}^j - \tilde{x}_k^j &= \tilde{w}_k^j - \tilde{x}_k^j - \eta\nabla\phi_j(\tilde{w}_k^j) - \eta\tilde{\delta}_k^j \\ &= (1-\theta)(\tilde{x}_k^j - \tilde{x}_{k-1}^j) - \eta\lambda_j(\tilde{w}_k^j - \nu_j) - \eta\tilde{\delta}_k^j \\ &= (1-\theta)(\tilde{x}_k^j - \tilde{x}_{k-1}^j) - \eta\lambda_j(\tilde{x}_k^j - \nu_j + (1-\theta)(\tilde{x}_k^j - \tilde{x}_{k-1}^j)) - \eta\tilde{\delta}_k^j\ .\end{aligned}$$

So for each $j \in S_2$, we have

$$\begin{aligned}&\langle \tilde{x}_{k+1}^j - \tilde{x}_k^j, \tilde{x}_k^j - \nu_j\rangle \\ &= (1-\theta)\langle \tilde{x}_k^j - \tilde{x}_{k-1}^j, \tilde{x}_k^j - \nu_j\rangle - \eta\lambda_j|\tilde{x}_k^j - \nu_j|^2 - \eta\lambda_j(1-\theta)\langle \tilde{x}_k^j - \tilde{x}_{k-1}^j, \tilde{x}_k^j - \nu_j\rangle - \eta\langle \tilde{\delta}_k^j, \tilde{x}_k^j - \nu_j\rangle \\ &\ge (1-\theta)\langle \tilde{x}_k^j - \tilde{x}_{k-1}^j, \tilde{x}_k^j - \nu_j\rangle - \eta\lambda_j|\tilde{x}_k^j - \nu_j|^2 \\ &\quad + \frac{\eta\lambda_j(1-\theta)}{2}(|\tilde{x}_k^j - \tilde{x}_{k-1}^j|^2 + |\tilde{x}_k^j - \nu_j|^2) + \frac{\eta}{2\lambda_j(1+\theta)}|\tilde{\delta}_k^j|^2 + \frac{\eta\lambda_j(1+\theta)}{2}|\tilde{x}_k^j - \nu_j|^2 \\ &= (1-\theta)\langle \tilde{x}_k^j - \tilde{x}_{k-1}^j, \tilde{x}_k^j - \nu_j\rangle + \frac{\eta\lambda_j(1-\theta)}{2}|\tilde{x}_k^j - \tilde{x}_{k-1}^j|^2 + \frac{\eta}{2\lambda_j(1+\theta)}|\tilde{\delta}_k^j|^2 \\ &= (1-\theta)\langle \tilde{x}_k^j - \tilde{x}_{k-1}^j, \tilde{x}_{k-1}^j - \nu_j\rangle + (1-\theta)|\tilde{x}_k^j - \tilde{x}_{k-1}^j|^2 + \frac{\eta\lambda_j(1-\theta)}{2}|\tilde{x}_k^j - \tilde{x}_{k-1}^j|^2 + \frac{\eta}{2\lambda_j(1+\theta)}|\tilde{\delta}_k^j|^2 \\ &\ge (1-\theta)\langle \tilde{x}_k^j - \tilde{x}_{k-1}^j, \tilde{x}_{k-1}^j - \nu_j\rangle + \frac{\eta}{2\lambda_j}|\tilde{\delta}_k^j|^2\ ,\end{aligned}$$

where we use the fact that $\lambda_j < 0$ when $j \in S_2$ in the first inequality and the fact

$$\left(1 + \frac{\eta\lambda_j}{2}\right)(1-\theta) \ge \left(1 - \frac{\eta\tilde{L}}{2}\right)(1-\theta) \ge 0$$

indicates the second inequality. Then we have

$$\begin{aligned}&\langle \tilde{x}_{k+1}^j - \tilde{x}_k^j, \tilde{x}_k^j - \nu_j\rangle \\ &\ge (1-\theta)^k\langle \tilde{x}_1^j - \tilde{x}_0^j, \tilde{x}_0^j - \nu_j\rangle + \frac{\eta}{2\lambda_j}\sum_{i=1}^k(1-\theta)^{k-i}|\tilde{\delta}_i^j|^2 \\ &\overset{(a)}{=} -\frac{\eta}{2\lambda_j}(1-\theta)^k\langle \nabla^j\Phi(x_0), \hat{\nabla}^j\Phi(x_0)\rangle + \frac{\eta}{2\lambda_j}\sum_{i=1}^k(1-\theta)^{k-i}|\tilde{\delta}_i^j|^2 \\ &= -\frac{\eta}{2\lambda_j}(1-\theta)^k\left(|\nabla^j\Phi(x_0)|^2 + |\hat{\nabla}^j\Phi(x_0)|^2 - |\nabla^j\Phi(x_0) - \hat{\nabla}^j\Phi(x_0)|^2\right) + \frac{\eta}{2\lambda_j}\sum_{i=1}^k(1-\theta)^{k-i}|\tilde{\delta}_i^j|^2 \\ &\overset{(b)}{\ge} \frac{\eta}{2\lambda_j}(1-\theta)^k\left(|\nabla^j\Phi(x_0) - \hat{\nabla}^j\Phi(x_0)|^2\right) + \frac{\eta}{2\lambda_j}\sum_{i=1}^k(1-\theta)^{k-i}|\tilde{\delta}_i^j|^2,\end{aligned}$$

where we use

$$\tilde{x}_1^j - \tilde{x}_0^j = \tilde{x}_1^j - \tilde{w}_0^j = -\eta\left(\mathbf{U}^\top \hat{\nabla}\Phi(x_0)\right)^j \quad \text{and} \quad \tilde{x}_0^j - \nu_j = -\frac{1}{\lambda_j}\tilde{\nabla}^j\Phi(x_0)$$

in $\overset{(a)}{=}$ and $\lambda_j < 0$ in $\overset{(b)}{\geq}$. Plugging above inequality into equation (25) and using $\lambda_j < 0$, we have

$$\phi_j(\tilde{x}_{k+1}^j) - \phi_j(\tilde{x}_k^j) \leq -\frac{\theta}{2\eta}|\tilde{x}_{k+1}^j - \tilde{x}_k^j|^2 + \frac{\eta}{2}(1-\theta)^k\left(|\nabla^j\Phi(x_0) - \hat{\nabla}^j\Phi(x_0)|^2\right) + \frac{\eta}{2}\sum_{i=1}^{k}(1-\theta)^{k-i}|\tilde{\delta}_i^j|^2.$$

Summing over above result with $k = 0, 1, \ldots, \mathcal{K}-1$ for $j \in S_2$, we have

$$\begin{aligned}
&\sum_{j\in S_2}\phi_j(\tilde{x}_{\mathcal{K}}^j) - \sum_{j\in S_2}\phi_j(\tilde{x}_0^j) \\
&\leq -\sum_{j\in S_2}\frac{\theta}{2\eta}\sum_{k=0}^{\mathcal{K}-1}|\tilde{x}_{k+1}^j - \tilde{x}_k^j|^2 + \frac{\eta}{2}\left\|\nabla\Phi(x_0) - \hat{\nabla}\Phi(x_0)\right\|_2^2\sum_{k=0}^{\mathcal{K}-1}(1-\theta)^k + \frac{\eta}{2}\sum_{k=0}^{\mathcal{K}-1}\sum_{i=1}^{k}(1-\theta)^{k-i}\|\tilde{\delta}_i\|_2^2 \\
&\leq -\sum_{j\in S_2}\frac{\theta}{2\eta}\sum_{k=0}^{\mathcal{K}-1}|\tilde{x}_{k+1}^j - \tilde{x}_k^j|^2 + \frac{\eta}{2}\sigma^2\sum_{k=0}^{\mathcal{K}-1}(1-\theta)^k + \frac{\eta}{2}\sum_{k=0}^{\mathcal{K}-1}\sum_{i=1}^{k}(1-\theta)^{k-i}\|\tilde{\delta}_i\|_2^2 \\
&\leq -\sum_{j\in S_2}\frac{\theta}{2\eta}\sum_{k=0}^{\mathcal{K}-1}|\tilde{x}_{k+1}^j - \tilde{x}_k^j|^2 + \frac{\eta\mathcal{K}}{2\theta}\sigma^2 + \frac{\eta\mathcal{K}}{2\theta}(2\tilde{\rho}B + \sigma)^2,
\end{aligned}$$

which completes the proof. ∎

Putting Lemma 27 and 28 together, we can show the decrease of $\Phi(x)$ in each epoch.

Lemma 29 *Suppose that Assumption 3 and Condition 10 hold. Let* $\eta \leq \frac{1}{4\tilde{L}}$ *and* $0 \leq \theta \leq 1$. *When the "if condition" triggers and* $\|\nabla\Phi(w_{\mathcal{K}-1})\|_2 \leq \frac{B}{\eta}$, *then we have*

$$\Phi(x_{\mathcal{K}}) - \Phi(x_0) \leq -\frac{\epsilon^{3/2}}{\sqrt{\tilde{\rho}}}\,.$$

Proof Summing over equation (23) and equation (24), we have

$$\begin{aligned}
\phi(\tilde{x}_{\mathcal{K}}) - \phi(\tilde{x}_0) &= \sum_{j\in S_1\cup S_2}\phi_j(\tilde{x}_{\mathcal{K}}^j) - \phi_j(\tilde{x}_0^j) \\
&\leq \frac{3\theta}{8\eta}\sum_{k=0}^{\mathcal{K}-1}\|\tilde{x}_{k+1} - \tilde{x}_k\|_2^2 + \frac{5\eta\mathcal{K}}{2\theta}(2\tilde{\rho}B^2 + \sigma)^2 + \frac{\eta\mathcal{K}}{2\theta}\sigma^2 \qquad (26) \\
&\leq -\frac{3\theta B^2}{8\eta\mathcal{K}} + \frac{5\eta\mathcal{K}}{2\theta}(2\tilde{\rho}B^2 + \sigma)^2 + \frac{\eta\mathcal{K}}{2\theta}\sigma^2,
\end{aligned}$$

where we use equation (9) in the last inequality. Plugging into equation (20) and using $\mathcal{K} \leq K$, we have

$$\begin{aligned}\Phi(x_{\mathcal{K}}) - \Phi(x_0) &\leq -\frac{3\theta B^2}{8\eta\mathcal{K}} + \frac{5\eta\mathcal{K}}{2\theta}(2\tilde{\rho}B^2+\sigma)^2 + \frac{\tilde{\rho}}{6}(3B+\eta\sigma)^3 + \frac{\eta\mathcal{K}}{2\theta}\sigma^2 \\ &\leq -\frac{3\theta B^2}{8\eta K} + \frac{5\eta K}{2\theta}(2\tilde{\rho}B^2+\sigma)^2 + \frac{\tilde{\rho}}{6}(3B+\eta\sigma)^3 + \frac{\eta\mathcal{K}}{2\theta}\sigma^2 \qquad (27) \\ &\leq -\frac{\epsilon^{3/2}}{\sqrt{\tilde{\rho}}}\,.\end{aligned}$$

This completes the proof. ■

Then we can proof that the number of epochs is bounded as shown in the following lemma.

Lemma 30 *Consider the setting of Theorem 13, and we run* `RAHGD` *in Algorithm 3, then the algorithm will terminate in at most $\Delta\sqrt{\tilde{\rho}}\epsilon^{-3/2}$ epochs.*

Proof From Lemma 26 and 29, we have

$$\Phi(x_{\mathcal{K}}) - \Phi(x_0) \leq -\min\left\{\frac{\epsilon^{3/2}}{\sqrt{\tilde{\rho}}}, \frac{\epsilon\tilde{L}}{\tilde{\rho}}\right\}. \qquad (28)$$

Notice that in Algorithm 3, we set x_0 to be the last iterate $x_{\mathcal{K}}$ in the previous epoch. Summing over all epochs, say N total epochs, we have

$$\min_{x\in\mathbb{R}^{d_x}} \Phi(x) - \Phi(x_{\text{int}}) \leq -N\min\left\{\frac{\epsilon^{3/2}}{\sqrt{\tilde{\rho}}}, \frac{\epsilon\tilde{L}}{\tilde{\rho}}\right\}. \qquad (29)$$

So the algorithm will terminate in at most $\Delta\sqrt{\tilde{\rho}}\epsilon^{-3/2}$ epochs. ■

Now we are prepared to prove Theorem 13.

Proof Lemma 30 says that `RAHGD` will terminate in at most $\Delta\sqrt{\tilde{\rho}}\epsilon^{-3/2}$ epochs. Since each epoch needs at most $K = \frac{1}{2}\big(\tilde{L}^2/(\tilde{\rho}\epsilon)\big)^{1/4}$ iterations, the total iterations must be less than $\Delta\tilde{L}^{1/2}\tilde{\rho}^{1/4}\epsilon^{-7/4}$. Recall that we have $\tilde{L} = \mathcal{O}(\kappa^3)$ and $\tilde{\rho} = \mathcal{O}(\kappa^5)$, thus the total iterations is at most $\mathcal{O}(\kappa^{11/4}\epsilon^{-7/4})$.

Now we consider the last epoch. Denote $\tilde{w} = \mathbf{U}^\top\hat{w} = \frac{1}{K_0+1}\sum_{k=0}^{K_0}\mathbf{U}^\top w_k = \frac{1}{K_0+1}\sum_{k=0}^{K_0}\tilde{w}_k$. Since ϕ is quadratic, we have

$$\begin{aligned}\|\phi(\tilde{w})\|_2 &= \left\|\frac{1}{K_0+1}\sum_{k=0}^{K_0}\nabla\phi(\tilde{w}_k)\right\|_2 \\ &\overset{(a)}{=} \frac{1}{\eta(K_0+1)}\left\|\sum_{k=0}^{K_0}\left(\tilde{x}_{k+1} - \tilde{w}_k + \eta\tilde{\delta}_k\right)\right\|_2 \\ &= \frac{1}{\eta(K_0+1)}\left\|\sum_{k=0}^{K_0}\left(\tilde{x}_{k+1} - \tilde{x}_k - (1-\theta)(\tilde{x}_k - \tilde{x}_{k-1}) + \eta\tilde{\delta}_k\right)\right\|_2 \\ &\overset{(b)}{=} \frac{1}{\eta(K_0+1)}\left\|\tilde{x}_{K_0+1} - \tilde{x}_0 - (1-\theta)(\tilde{x}_{K_0} - \tilde{x}_0 + \eta\sum_{k=0}^{K_0}\tilde{\delta}_k\right\|_2\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{\eta(K_0+1)} \left\| \tilde{x}_{K_0+1} - \tilde{x}_{K_0} + \theta(\tilde{x}_{K_0} - \tilde{x}_0) + \eta \sum_{k=0}^{K_0} \tilde{\delta}_k \right\|_2 \\
&\leq \frac{1}{\eta(K_0+1)} \left(\|\tilde{x}_{K_0+1} - \tilde{x}_{K_0}\|_2 + \theta \|\tilde{x}_{K_0} - \tilde{x}_0\|_2 + \eta \sum_{k=0}^{K_0} \left\|\tilde{\delta}_k\right\|_2 \right) \\
&\overset{\text{(c)}}{\leq} \frac{2}{\eta K} \|\tilde{x}_{K_0+1} - \tilde{x}_{K_0}\|_2 + \frac{2\theta B}{\eta K} + 2\tilde{\rho} B^2 + \sigma,
\end{aligned}
$$

where we use equation (22) in $\overset{\text{(a)}}{=}$, $x_{-1} = x_0$ in $\overset{\text{(b)}}{=}$; $K_0 + 1 \geq \frac{K}{2}$, equation (13) and equation (14) in $\overset{\text{(c)}}{\leq}$.

From $K_0 = \arg\min_{\lfloor \frac{K}{2} \rfloor \leq k \leq K-1} \ \|x_{k+1} - x_k\|_2$, we have

$$
\begin{aligned}
\|x_{K_0+1} - x_{K_0}\|_2^2 &\leq \frac{1}{K - \lfloor K/2 \rfloor} \sum_{k=\lfloor K/2 \rfloor}^{K-1} \|x_{k+1} - x_k\|_2^2 \\
&\leq \frac{1}{K - \lfloor K/2 \rfloor} \sum_{k=0}^{K-1} \|x_{k+1} - x_k\|_2^2 \\
&\overset{\text{(d)}}{\leq} \frac{1}{K - \lfloor K/2 \rfloor} \frac{B^2}{K} \\
&\leq \frac{2B^2}{K^2},
\end{aligned}
$$

where we use equation (13) in $\overset{\text{(d)}}{\leq}$. On the other hand, we also have

$$
\begin{aligned}
\|\nabla \Phi(\hat{w})\|_2 &= \left\| \tilde{\nabla} \Phi(\hat{w}) \right\|_2 \\
&\leq \|\nabla \phi(\tilde{w})\|_2 + \left\| \tilde{\nabla} \Phi(\hat{w}) - \nabla \phi(\tilde{w}) \right\|_2 \\
&= \|\nabla \phi(\tilde{w})\|_2 \left\| \tilde{\nabla} \Phi(\hat{w}) - \tilde{\nabla} \Phi(x_0) - \mathbf{\Lambda}(\tilde{w} - \tilde{x}_0) \right\|_2 \\
&= \|\nabla \phi(\tilde{w})\|_2 + \|\nabla \Phi(\hat{w}) - \nabla \Phi(x_0) - \mathbf{H}(\hat{w} - x_0)\|_2 \\
&\leq \|\nabla \phi(\tilde{w})\|_2 + \frac{\tilde{\rho}}{2} \|\hat{w} - x_0\|_2^2 \\
&\overset{\text{(e)}}{\leq} \|\nabla \phi(\tilde{w})\|_2 + 2\tilde{\rho} B^2 \ ,
\end{aligned}
$$

where we use $\|\hat{w} - x_0\|_2 \leq \frac{1}{K_0+1} \sum_{k=0}^{K_0} \|w_k - x_0\|_2 \leq 2B$ from equation (14) in $\overset{\text{(e)}}{\leq}$. So we have

$$
\|\nabla \Phi(\hat{w})\|_2 \leq \frac{2\sqrt{2}B}{\eta K^2} + \frac{2\theta B}{\eta K} + 4\tilde{\rho} B^2 + \sigma \leq 83\epsilon.
$$

This completes our proof of Theorem 13. ■

D.2. Proof of Proposition 14

Proof We first consider the iterations of CG in Algorithm 3 in one epoch. We set $T'_{t,k}$ as

$$T'_{t,k} = \begin{cases} \left\lceil \frac{\sqrt{\kappa}+1}{2} \log\left(\frac{4\ell\sqrt{\kappa}}{\sigma}\left(\|v_{0,-1}\|_2 + \frac{M}{\mu}\right)\right)\right\rceil, & k=0, \\ \left\lceil \frac{\sqrt{\kappa}+1}{2} \log\left(\frac{4\ell\sqrt{\kappa}}{\sigma}\left(\frac{\sigma}{2\ell} + \frac{2M}{\mu}\right)\right)\right\rceil, & k\geq 1. \end{cases} \tag{30}$$

We denote

$$v^*(x,y) = \left(\nabla^2_{yy} g(x,y)\right)^{-1} \nabla_y f(x,y),$$

then

$$\|v^*(x,y)\|_2 \leq \frac{M}{\mu}, \quad \forall x \in \mathbb{R}^{d_x}, y \in \mathbb{R}^{d_y}.$$

We use induction to show that

$$\left\|v_{t,k} - v^*_{t,k}\right\|_2 \leq \frac{\sigma}{2\ell}$$

holds for any $k \geq 0$. For $k=0$, Lemma 2 straightforwardly implies that

$$\left\|v_{t,0} - v^*_{t,0}\right\|_2 \leq \frac{\left\|v_{0,-1} - v^*_{t,0}\right\|_2}{\|v_{0,-1}\|_2 + M/\mu} \cdot \frac{\sigma}{2\ell} \leq \frac{\sigma}{2\ell}.$$

Suppose it holds that $\left\|v_{t,k} - v^*_{t,k}\right\|_2 \leq \frac{\sigma}{2\ell}$ for any $k = k'-1$, then we have

$$\begin{aligned} \left\|v_{t,k'} - v^*_{t,k'}\right\|_2 &\leq 2\sqrt{\kappa}\left(1 - \frac{2}{1+\sqrt{\kappa}}\right)^{T'_{t,k'}} \left\|v_{t,k'-1} - v^*_{t,k'}\right\|_2 \\ &\leq 2\sqrt{\kappa}\left(1 - \frac{2}{1+\sqrt{\kappa}}\right)^{T'_{t,k'}} \left(\left\|v_{t,k'-1} - v^*_{t,k'-1}\right\|_2 + \left\|v^*_{t,k'-1} - v^*_{t,k'}\right\|_2\right) \\ &\leq 2\sqrt{\kappa}\left(1 - \frac{2}{1+\sqrt{\kappa}}\right)^{T'_{t,k'}} \left(\frac{\sigma}{2\ell} + \frac{2M}{\mu}\right) \leq \frac{\sigma}{2\ell}, \end{aligned}$$

where the first inequality is based on Lemma 2, the second one uses triangle inequality, the third one uses the definition of T'_k. Therefore, equation (4) in Condition 10 can hold.

The total iteration number of CG in Algorithm 3 in one epoch satisfies

$$\begin{aligned} &\sum_{k=0}^{\mathcal{K}-1} T'_k \\ &\leq \mathcal{K} + \frac{\sqrt{\kappa}+1}{2}\left(\frac{2T'_0}{\sqrt{\kappa}+1} + \sum_{k=1}^{\mathcal{K}-1} \log\left(\frac{4\ell\sqrt{\kappa}}{\sigma}\left(\frac{\sigma}{2\ell} + \frac{2M}{\mu}\right)\right)\right) \\ &= \mathcal{K} + \frac{\sqrt{\kappa}+1}{2}\left(\frac{2T'_0}{\sqrt{\kappa}+1} + (\mathcal{K}-1)\log\left(\frac{4\ell\sqrt{\kappa}}{\sigma}\left(\frac{\sigma}{2\ell} + \frac{2M}{\mu}\right)\right)\right) \\ &= \mathcal{K} + \frac{\sqrt{\kappa}+1}{2}\mathcal{K}\left(\frac{1}{\mathcal{K}}\log\left(\frac{4\ell\sqrt{\kappa}}{\sigma}\left(\|v_{0,-1}\|_2 + \frac{M}{\mu}\right)\right) + \left(1 - \frac{1}{\mathcal{K}}\right)\log\left(\frac{4\ell\sqrt{\kappa}}{\sigma}\left(\frac{\sigma}{2\ell} + \frac{2M}{\mu}\right)\right)\right). \end{aligned}$$

Now we consider the iterations of AGD in Algorithm 3.We first show the following lemma.

Lemma 31 *Consider the setting of Theorem 13, and we run Algorithm 3, then we have*

$$\|y^*(w_{t,-1})\|_2 \leq \hat{C}$$

for any $t > 0$*, where* $\hat{C} = \|y^*(x_{0,0})\|_2 + (2B + \eta\sigma + \eta C)\kappa\Delta\sqrt{\tilde{\rho}}\epsilon^{-3/2}$.

Then consider the iteratons of AGD in Algorithm 3. We choose $T_{t,k}$ as

$$T_{t,k} = \begin{cases} \left\lceil 2\sqrt{\kappa}\log\left(\frac{2\tilde{L}\sqrt{\kappa+1}}{\sigma}\hat{C}\right)\right\rceil, & k = -1. \\ \left\lceil 2\sqrt{\kappa}\log\left(\frac{2\tilde{L}\sqrt{\kappa+1}}{\sigma}\left(\frac{\sigma}{2\tilde{L}} + 2\kappa B\right)\right)\right\rceil, & k \geq 0. \end{cases} \tag{31}$$

We will use induction to show that Lemma 31 as well as equation (3) in Condition 10 will hold. For $t = 0$, Lemma 31 hold trivially. Then we use induction with respect to k to prove that

$$\|y_{t,k} - y^*(w_{t,k})\|_2 \leq \frac{\sigma}{2\tilde{L}}$$

holds for any $k \geq -1$. For $k = -1$, Lemma 1 directly implies

$$\|y_{t,-1} - y^*(w_{t,-1})\|_2 \leq \frac{\left\|y^*(w_{t,-1})\right\|_2}{\hat{C}} \cdot \frac{\sigma}{2\tilde{L}} \leq \frac{\sigma}{2\tilde{L}},$$

where the second inequality is based on Lemma 31. Suppose it holds that

$$\|y_{t,k-1} - y^*(w_{t,k-1})\|_2 \leq \frac{\sigma}{2\tilde{L}}$$

for any $k \leq k' - 1$, then we have

$$\begin{aligned}
&\left\|y_{t,k'} - y^*(w_{t,k'})\right\|_2 \\
&\leq \sqrt{1+\kappa}\left(1 - \frac{1}{\sqrt{\kappa}}\right)^{T_{t,k'}/2} \left\|y_{t,k'-1} - y^*(w_{t,k'})\right\|_2 \\
&\leq \sqrt{1+\kappa}\left(1 - \frac{1}{\sqrt{\kappa}}\right)^{T_{t,k'}/2} \left(\left\|y_{t,k'-1} - y^*(w_{t,k'-1})\right\|_2 + \left\|y^*(w_{t,k'-1}) - y^*(w_{t,k'})\right\|_2\right) \\
&\leq \sqrt{1+\kappa}\left(1 - \frac{1}{\sqrt{\kappa}}\right)^{T_{t,k'}/2} \left(\frac{\sigma}{2\tilde{L}} + \kappa\left\|w_{t,k'-1} - w_{t,k'}\right\|_2\right) \\
&\leq \sqrt{1+\kappa}\left(1 - \frac{1}{\sqrt{\kappa}}\right)^{T_{t,k'}/2} \left(\frac{\sigma}{2\tilde{L}} + 2\kappa B\right) \\
&\leq \frac{\sigma}{2\tilde{L}}\,,
\end{aligned}$$

where the first inequality is based on Lemma 1, the second one uses triangle inequality, the third one is based on induction hypothesis and Lemma 5, the fourth one uses equation (12), and the last step use the definition of $T_{t,k}$. Therefore, equation (3) in Condition 10 can hold.

Suppose Lemma 31 and equation (3) in Condition 10 hold for any $t \leq t' - 1$, then we have shown that when we choose $T'_{t,k}$ as defined in equation (30), then equation (4) in Condition 10 can hold. Thus, from Lemma 12 we obtain that:

$$\|\nabla\Phi(w_{t,k}) - \hat{\nabla}\Phi(w_{t,k})\|_2 \leq \sigma\,. \tag{32}$$

We claim that for any t, we can find some constant C to satisfy:

$$\|\nabla\Phi(w_{t,\mathcal{K}-1})\|_2 \le C\ . \tag{33}$$

Otherwise, equation (18) in Lemma 26 shows that $\Phi(w_{t,\mathcal{K}})$ can go to $-\infty$ and contradict with the assumption $\min_{x\in\mathbf{R}^{d_x}}\Phi(x) > -\infty$.

For any epoch $t \le t'-1$, we have

$$\begin{aligned}
&\|x_{t,\mathcal{K}} - x_{t,0}\|_2 \\
&= \|x_{t,\mathcal{K}} - x_{t,\mathcal{K}-1} + x_{t,\mathcal{K}-1} - x_{t,0}\|_2 \\
&= \left\|(1-\theta)(x_{t,\mathcal{K}-1} - x_{t,\mathcal{K}-2}) - \eta\hat{\nabla}\Phi(w_{t,\mathcal{K}-1}) + x_{t,\mathcal{K}-1} - x_{t,0}\right\|_2 \\
&\le \|x_{t,\mathcal{K}-1} - x_{t,\mathcal{K}-2}\|_2 + \|x_{t,\mathcal{K}-1} - x_{t,0}\|_2 + \eta\left\|\hat{\nabla}\Phi(w_{t,\mathcal{K}-1})\right\|_2 \\
&\le 2B + \eta\left\|\hat{\nabla}\Phi(w_{t,\mathcal{K}-1}) - \nabla\Phi(w_{t,\mathcal{K}-1}) + \nabla\Phi(w_{t,\mathcal{K}-1})\right\|_2 \\
&\le 2B + \eta\left\|\hat{\nabla}\Phi(w_{t,\mathcal{K}-1}) - \nabla\Phi(w_{t,\mathcal{K}-1})\right\|_2 + \eta\|\nabla\Phi(w_{t,\mathcal{K}-1})\|_2 \\
&\le 2B + \eta\sigma + \eta\|\nabla\Phi(w_{t,\mathcal{K}-1})\|_2 \\
&\le 2B + \eta(\sigma + C)
\end{aligned} \tag{34}$$

for some constant C. Here we use triangle inequality in the first inequality; equation (10) in the second one; triangle inequality again in the third one; equation (32) in the fourth one and equation (33) in the last one.

Then for t'-th epoch, we have

$$\begin{aligned}
\left\|y^*(w_{t',-1}) - y^*(x_{0,0})\right\|_2 &\le \kappa\left\|w_{t',-1} - x_{0,0}\right\|_2 \\
&= \kappa\left\|x_{t',0} - x_{0,0}\right\|_2 \\
&= \kappa\left\|x_{t'-1,\mathcal{K}} - x_{0,0}\right\|_2 \\
&\le \kappa(\left\|x_{t'-1,0} - x_{0,0}\right\|_2 + \left\|x_{t'-1,\mathcal{K}} - x_{t'-1,0}\right\|_2) \\
&\le \kappa(\left\|x_{t'-1,0} - x_{1,0}\right\|_2 + (2B + \eta\sigma + \eta C)) \\
&\le (2B + \eta\sigma + \eta C)\kappa t\ ,
\end{aligned}$$

where the first inequality is based on the Lipschitz continuous of $y^*(x)$ shown in Lemma 5; the second one uses triangle inequality; the third one is based on equation (34), and the last one uses induction. Then we have

$$\begin{aligned}
\left\|y^*(w_{t',-1})\right\|_2 &\le \|y^*(x_{0,0})\|_2 + B\kappa t' \\
&\le \|y^*(x_{0,0})\|_2 + \frac{(2B + \eta\sigma + \eta C)\kappa\Delta\sqrt{\rho}}{\epsilon^{3/2}},
\end{aligned}$$

where we use Lemma 30 in the last inequality.

Similarly with the case $t = 0$, we use induction with respect to k again, we have that equation (3) in Condition 10 hold.

This also finishes the proof for Lemma 31.

The total gradient calls from AGD in Algorithm 3 in one epoch satisfies

$$\begin{aligned}\sum_{k=-1}^{\mathcal{K}-1} T_{t,k} &\le 2\sqrt{\kappa}\left(\frac{T_{-1}}{2\sqrt{\kappa}} + \sum_{k=0}^{\mathcal{K}-1}\log\left(\sqrt{\kappa+1} + \frac{4\tilde{L}\kappa\sqrt{\kappa+1}B}{\sigma}\right)\right) + \mathcal{K} + 1 \\ &= 2\sqrt{\kappa}\left(\frac{T_{-1}}{2\sqrt{\kappa}} + \mathcal{K}\log\left(\sqrt{\kappa+1} + \frac{4\tilde{L}\kappa\sqrt{\kappa+1}B}{\sigma}\right)\right) + \mathcal{K} + 1 \\ &= 2\sqrt{\kappa}\mathcal{K}\left(\frac{1}{\mathcal{K}}\log\left(\frac{2\tilde{L}\sqrt{\kappa+1}}{\sigma}\hat{C}\right) + \log\left(\sqrt{\kappa+1} + \frac{4\tilde{L}\kappa\sqrt{\kappa+1}B}{\sigma}\right)\right) + \mathcal{K} + 1.\end{aligned}$$

This completes our proof of Proposition 14. ■

D.3. Proof of Corollary 15

Proof Theorem 13 says that RAHGD can output an ϵ-FOSP within at most $\mathcal{O}\left(\Delta\tilde{L}^{1/2}\tilde{\rho}^{1/4}\epsilon^{-7/4}\right)$ iterations in the outer loop. Then we have

$$Gc(f,\epsilon) = \mathcal{O}\left(\frac{\Delta\tilde{L}^{1/2}\tilde{\rho}^{1/4}}{\epsilon^{7/4}}\right) \qquad \text{and} \qquad JV(g,\epsilon) = \mathcal{O}\left(\frac{\Delta\tilde{L}^{1/2}\tilde{\rho}^{1/4}}{\epsilon^{7/4}}\right).$$

Recall that $\tilde{L} = \mathcal{O}(\kappa^3)$ and $\tilde{\rho} = \mathcal{O}(\kappa^5)$, we have

$$Gc(f,\epsilon) = \mathcal{O}\left(\kappa^{11/4}\epsilon^{-7/4}\right) \qquad \text{and} \qquad JV(g,\epsilon) = \mathcal{O}\left(\kappa^{11/4}\epsilon^{-7/4}\right).$$

Gradients of $g(x,\cdot)$ and Hessian-vector products are occurred in AGD and CG respectively. Proposition 14 shows that we only require $\mathcal{O}\left(\sqrt{\kappa}\mathcal{K}\log(\frac{1}{\epsilon})\right)$ iterates of AGD and CG in one epoch to have Condition 10 hold. From Lemma 30 we know that RAHGD will terminate in at most $\Delta\sqrt{\tilde{\rho}}\epsilon^{-3/2}$ epochs. Recall that $\mathcal{K} \le K = \frac{1}{2}\left(\tilde{L}^2/(\tilde{\rho}\epsilon)\right)^{1/4}$, we have

$$Gc(g,\epsilon) = \mathcal{O}\left(\frac{\Delta\tilde{L}^{1/2}\tilde{\rho}^{1/4}\kappa^{1/2}\log(1/\epsilon)}{\epsilon^{7/4}}\right) \quad \text{and} \quad HV(g,\epsilon) = \mathcal{O}\left(\frac{\Delta\tilde{L}^{1/2}\tilde{\rho}^{1/4}\kappa^{1/2}\log(1/\epsilon)}{\epsilon^{7/4}}\right).$$

Hiding polylogarithmic factor and pluging $\tilde{L} = \mathcal{O}(\kappa^3)$ and $\tilde{\rho} = \mathcal{O}(\kappa^5)$ into it, we have

$$Gc(g,\epsilon) = \tilde{\mathcal{O}}\left(\kappa^{13/4}\epsilon^{-7/4}\right) \qquad \text{and} \qquad HV(g,\epsilon) = \tilde{\mathcal{O}}\left(\kappa^{13/4}\epsilon^{-7/4}\right).$$

■

Appendix E. Proofs for Section 4

In this section, we provide the proofs for theorems in Section 4. We first show that the number of epochs can be bounded. Then we prove that PRAHGD can output an $(\epsilon, \sqrt{\tilde{\rho}\epsilon})$-SOSP. Finally, we provide the oracle complexity analysis.

E.1. Proof of Theorem 16

Lemma 32 *Consider the setting of Theorem 16, and we run Algorithm 3, then the algorithm will terminate in at most* $\mathcal{O}\left(\Delta\sqrt{\tilde{\rho}}\chi^5\epsilon^{-3/2}\right)$ *epochs.*

Proof

From the Lipschitz continuity of gradient, we have

$$\begin{aligned}\Phi(x_{t+1,0}) - \Phi(x_{t,\mathcal{K}}) &\leq \langle \nabla\Phi(x_{t,\mathcal{K}}), x_{t+1,0} - x_{t,\mathcal{K}}\rangle + \frac{\tilde{L}}{2}\|x_{t+1,0} - x_{t,\mathcal{K}}\|_2^2 \\ &= \langle \nabla\Phi(x_{t,\mathcal{K}}), \xi_t\rangle + \frac{\tilde{L}}{2}\|\xi_t\|_2^2 \\ &\leq \|\nabla\Phi(x_{t,\mathcal{K}})\|_2 r + \frac{\tilde{L}r^2}{2}.\end{aligned}$$

If $\|\nabla\Phi(w_{\mathcal{K}-1})\|_2 > \frac{B}{\eta}$, then Lemma 26 means when the "if condition" triggers, we have

$$\Phi(x_{\mathcal{K}}) - \Phi(x_0) \leq -\frac{B^2}{4\eta} + \sigma B + \frac{5\eta\sigma^2 K}{8}. \tag{35}$$

We say that $\|\nabla\Phi(x_{t,\mathcal{K}})\|_2$ is bounded. Otherwise, one gradient descent step $z = x_{t,\mathcal{K}} - \eta\nabla\Phi(x_{t,\mathcal{K}})$ leads to

$$\begin{aligned}\Phi(z) &\leq \Phi(x_{t,\mathcal{K}}) + \langle \nabla\Phi(x_{t,\mathcal{K}}), -\eta\nabla\Phi(x_{t,\mathcal{K}})\rangle + \frac{\tilde{L}\eta^2}{2}\|\nabla\Phi(x_{t,\mathcal{K}})\|_2^2 \\ &= \Phi(x_{t,\mathcal{K}}) - \frac{7\eta}{8}\|\nabla\Phi(x_{t,\mathcal{K}})\|_2^2,\end{aligned}$$

which means $\Phi(z) \sim -\infty$ and contradicts with the assumption $\min_{x\in\mathbb{R}^{d_x}}\Phi(x) > -\infty$. Let $\|\nabla\Phi(x_{t,\mathcal{K}})\|_2 \leq C$, then we have

$$\Phi(x_{t+1,0}) - \Phi(x_{t,\mathcal{K}}) \leq Cr + \frac{\tilde{L}r^2}{2} \leq \frac{B^2}{8\eta}, \tag{36}$$

where we use the definition of r in the second inequality. Summing over equation (35) and equation (36), we obtain

$$\Phi(x_{t+1,0}) - \Phi(x_{t,0}) \leq -\frac{B^2}{8\eta} + \sigma B + \frac{5\eta\sigma^2 K}{8} \leq -\frac{B^2}{8\eta} = -\frac{\epsilon\tilde{L}}{165888\tilde{\rho}\chi^4}$$

for all epochs. On the other hand, if $\|\nabla\Phi(w_{\mathcal{K}-1})\|_2 \leq \frac{B}{\eta}$, Lemma 29 means

$$\Phi(x_{\mathcal{K}}) - \Phi(x_0) \leq -\frac{3\theta B^2}{8\eta K} + \frac{5\eta K}{2\theta}(2\tilde{\rho}B^2 + \sigma)^2 + \frac{\tilde{\rho}}{6}(3B + \eta\sigma)^3 + \frac{\eta K}{2\theta}\sigma^2.$$

We also have

$$\begin{aligned}\|\nabla\Phi(x_{\mathcal{K}})\|_2 &\leq \|\nabla\Phi(w_{\mathcal{K}-1})\|_2 + \|\nabla\Phi(x_{\mathcal{K}}) - \nabla\Phi(w_{\mathcal{K}-1})\|_2 \\ &\leq \|\nabla\Phi(w_{\mathcal{K}-1})\|_2 + \tilde{L}\|x_{\mathcal{K}} - w_{\mathcal{K}-1}\|_2\end{aligned}$$

$$\leq \|\nabla\Phi(w_{\mathcal{K}-1})\|_2 + \tilde{L}\eta\left(\|\nabla\Phi(w_{\mathcal{K}-1})\|_2 + \left\|\hat{\nabla}\Phi(w_{\mathcal{K}-1}) - \nabla\Phi(w_{\mathcal{K}-1})\right\|_2\right)$$
$$\leq \frac{B}{\eta} + \tilde{L}B + \frac{\sigma}{4} = \frac{5B}{4\eta} + \frac{\sigma}{4}\,.$$

So we obatin

$$\Phi(x_{t+1,0}) - \Phi(x_{t,\mathcal{K}}) \leq \frac{5Br}{4\eta} + \frac{\sigma r}{4} + \frac{\tilde{L}r^2}{2} \leq \frac{\theta B^2}{8\eta K} + \frac{\sigma B^2}{4},$$

and

$$\Phi(x_{t+1,0}) - \Phi(x_{t,0}) \leq -\frac{\theta B^2}{4\eta K} + \frac{5\eta K}{2\theta}(2\tilde{\rho}B^2 + \sigma)^2 + \frac{\tilde{\rho}}{6}(3B + \eta\sigma)^3 + \frac{\eta K}{2\theta}\sigma^2 + \frac{\sigma B^2}{4}$$
$$\leq -\frac{\epsilon^{1.5}}{663552\sqrt{\tilde{\rho}}\chi^5}\,.$$

Hence, the algorithm will terminate in at most $\mathcal{O}\left(\Delta\sqrt{\tilde{\rho}}\chi^5\epsilon^{-3/2}\right)$ epochs. ■

Before proving that PRAHGD can output an $(\epsilon, \sqrt{\tilde{\rho}\epsilon}\,)$-SOSP, we first show the following lemma.

Lemma 33 *Following the setting of Theorem 16, we additionally suppose that* $\lambda_{\min}(\mathbf{H}) < -\sqrt{\epsilon\tilde{\rho}}$, *where* $\mathbf{H} = \nabla^2\Phi(x)$ *for given* $x \in \mathbb{R}^{d_x}$. *We suppose points* $x_0', x_0'' \in \mathbb{R}^{d_x}$ *satisfy* $\|x_0' - x\|_2 \leq r$, $\|x_0'' - x\|_2 \leq r$ *and* $x_0' - x_0'' = r_0 e_1$, *where* e_1 *is the minimum eigen-direction of* $\mathbf{H}$ *and* $r_0 = \frac{\zeta r}{\sqrt{d_x}}$. *Running* PRAHGD *in Algorithm 3 with initialization* $x_{0,0} = x_0'$ *and* $x_{0,0} = x_0''$, *respectively, then at least one of these two initial points leads to its iterations trigger the "if condition".*

Proof Recall that the update rule of PRAHGD can be written as:

$$x_{k+1} = (2-\theta)x_k - (1-\theta)x_{k-1} - \eta\hat{\nabla}\Phi((2-\theta)x_k - (1-\theta)x_{k-1}).$$

We denote $z_k = x_k' - x_k''$, then

$$z_{k+1} = (2-\theta)z_k - (1-\theta)z_{k-1} - \eta(\hat{\nabla}\Phi(w_k') - \hat{\nabla}\Phi(w_k'')))$$
$$= (2-\theta)(\mathbf{I} - \eta\mathbf{H} - \eta\mathbf{\Omega}_k)z_k - (1-\theta)(\mathbf{I} - \eta\mathbf{H} - \eta\mathbf{\Omega}_k)z_{k-1} - \eta(\varsigma_k - \varsigma_k'),$$

where

$$\mathbf{\Omega}_k = \int_0^1 (\nabla^2\Phi(tw_k + (1-t)w_k') - K)\,\mathrm{d}t,\;\; \varsigma_k' = \nabla\Phi(w_k') - \hat{\nabla}\Phi(w_k') \;\text{ and }\; \varsigma_k'' = \nabla\Phi(w_k'') - \hat{\nabla}\Phi(w_k'').$$

In the last step, we use

$$\nabla\Phi(w_k') - \nabla\Phi(w_k'') = (\mathbf{H} + \mathbf{\Omega}_k)(w_k' - w_k'') = (\mathbf{H} + \mathbf{\Omega}_k)((2-\theta)z_k - (1-\theta)z_{t-1})\,.$$

We thus get the update of z_k in matrix form as follows

$$\begin{pmatrix} z_{k+1} \\ z_k \end{pmatrix} = \begin{pmatrix} (2-\theta)(\mathbf{I} - \eta\mathbf{H}) & -(1-\theta)(\mathbf{I} - \eta\mathbf{H}) \\ \mathbf{I} & 0 \end{pmatrix}\begin{pmatrix} w_k \\ w_{k-1} \end{pmatrix}$$
$$+ \eta\begin{pmatrix} (2-\theta)\mathbf{\Omega}_k z_k - (1-\theta)\mathbf{\Omega}_k z_{k-1} + \varsigma_k' - \varsigma_k'' \\ 0 \end{pmatrix}$$

$$= \mathbf{A}\begin{pmatrix} z_k \\ z_{k-1} \end{pmatrix} - \eta \begin{pmatrix} \omega_k \\ 0 \end{pmatrix} = \mathbf{A}^{k+1}\begin{pmatrix} z_0 \\ z_{-1} \end{pmatrix} - \eta \sum_{i=0}^{k} \mathbf{A}^{k-i}\begin{pmatrix} \omega_i \\ 0 \end{pmatrix} ,$$

where $\omega_k = (2-\theta)\mathbf{\Omega}_k z_k - (1-\theta)\mathbf{\Omega}_k z_{k-1} + \varsigma_k' - \varsigma_k''$. Then we have

$$z_k = \begin{pmatrix} \mathbf{I} & 0 \end{pmatrix} \mathbf{A}^k \begin{pmatrix} z_0 \\ z_0 \end{pmatrix} - \eta \begin{pmatrix} \mathbf{I} & 0 \end{pmatrix} \sum_{i=0}^{k-1} \mathbf{A}^{k-i-1}\begin{pmatrix} \omega_i \\ 0 \end{pmatrix} .$$

Assume that none of the iteration on $(x_0', x_1', \ldots, x_K')$ and $(x_0'', x_1'', \ldots, x_K'')$ trigger the "if condition", then we have

$$\begin{aligned} \left\|x_k' - x_0'\right\|_2 \le B,\ \left\|w_k' - x_0'\right\|_2 \le 2B,\ \forall k \le K, \\ \left\|x_k'' - x_0''\right\|_2 \le B,\ \left\|w_k'' - x_0''\right\|_2 \le 2B,\ \forall k \le K. \end{aligned} \tag{37}$$

Then we achieve

$$\begin{aligned} \|\mathbf{\Omega}_k\|_2 &\le \tilde{\rho} \max\left(\left\|w_k' - x\right\|_2, \left\|w_k'' - x\right\|_2\right) \\ &\le \tilde{\rho} \max\left(\left\|w_k' - x_0'\right\|_2, \left\|w_k'' - x_0''\right\|_2\right) + \tilde{\rho} r \le 3\tilde{\rho} B \end{aligned}$$

and

$$\begin{aligned} \|\omega_k\|_2 &\le 6\tilde{\rho} B \left(\|z_k\|_2 + \|z_{k-1}\|_2\right) + \left\|\varsigma_k' - \varsigma_k''\right\|_2 \\ &\le 6\tilde{\rho} B \left(\|z_k\|_2 + \|z_{k-1}\|_2\right) + 2\sigma, \end{aligned}$$

where we use Lemma 12 in the last step. We can show the following inequality for all $k \le K$ by induction:

$$\left\| \eta \begin{pmatrix} \mathbf{I} & 0 \end{pmatrix} \sum_{i=0}^{k-1} \mathbf{A}^{k-1-i} \begin{pmatrix} \omega_i \\ 0 \end{pmatrix} \right\|_2 \le \frac{1}{2} \left\| \begin{pmatrix} \mathbf{I} & 0 \end{pmatrix} \mathbf{A}^k \begin{pmatrix} z_0 \\ z_0 \end{pmatrix} \right\|_2 .$$

It is easy to check the base case holds for $k = 0$. Assume the inequality holds for all steps equal to or less than k. Then we have

$$\|z_k\|_2 \le \frac{3}{2} \left\| \begin{pmatrix} \mathbf{I} & 0 \end{pmatrix} \mathbf{A}^k \begin{pmatrix} z_0 \\ z_0 \end{pmatrix} \right\|_2 \qquad \text{and} \qquad \|\omega_k\|_2 \le 18\tilde{\rho} B \left\| \begin{pmatrix} \mathbf{I} & 0 \end{pmatrix} \mathbf{A}^k \begin{pmatrix} z_0 \\ z_0 \end{pmatrix} \right\|_2 + 2\sigma,$$

where we use the monotonicity of $\left\| \begin{pmatrix} \mathbf{I} & 0 \end{pmatrix} \mathbf{A}^k \begin{pmatrix} z_0 \\ z_0 \end{pmatrix} \right\|_2$ in k (Lemma 38 in [28]) in the last inequality.

We define

$$\begin{pmatrix} a_k & -b_k \end{pmatrix} = \begin{pmatrix} 1 & 0 \end{pmatrix} \mathbf{A}_{\min}^k \quad \text{and} \quad \mathbf{A}_{\min} = \begin{pmatrix} (2-\theta)(1-\eta\lambda_{\min}) & -(1-\theta)(1-\eta\lambda_{\min}) \\ 1 & 0 \end{pmatrix},$$

then

$$\begin{aligned} &\left\| \eta \begin{pmatrix} \mathbf{I} & 0 \end{pmatrix} \sum_{i=0}^{k} \mathbf{A}^{k-i} \begin{pmatrix} \omega_i \\ 0 \end{pmatrix} \right\|_2 \\ \le\ & \eta \sum_{i=0}^{k} \left\| \begin{pmatrix} \mathbf{I} & 0 \end{pmatrix} \sum_{i=0}^{k} \mathbf{A}^{k-i} \begin{pmatrix} \mathbf{I} \\ 0 \end{pmatrix} \right\|_2 \|\omega_i\|_2 \end{aligned}$$

$$
\begin{aligned}
&\leq \eta \sum_{i=0}^{k} \left\| \begin{pmatrix} \mathbf{I} & 0 \end{pmatrix} \sum_{i=0}^{k} \mathbf{A}^{k-i} \begin{pmatrix} \mathbf{I} \\ 0 \end{pmatrix} \right\|_2 \left(18\tilde{\rho} B \left\| \begin{pmatrix} \mathbf{I} & 0 \end{pmatrix} \mathbf{A}^{i} \begin{pmatrix} z_0 \\ z_0 \end{pmatrix} \right\|_2 + 2\sigma \right) \\
&\overset{\text{(a)}}{=} \eta \sum_{i=0}^{k} |a_{k-i}| \left(18\tilde{\rho} B r_0 |a_i - b_i| + 2\sigma \right) \\
&\overset{\text{(b)}}{\leq} \eta \sum_{i=0}^{k} |a_{k-i}| \left(20\tilde{\rho} B r_0 |a_i - b_i| \right) \\
&\overset{\text{(c)}}{\leq} 20\tilde{\rho} B \eta \sum_{i=0}^{k} \left(\frac{2}{\theta} + k + 1 \right) |a_{k+1} - b_{k+1}| r_0 \\
&\leq 20\tilde{\rho} B \eta K \left(\frac{2}{\theta} + K \right) \left\| \begin{pmatrix} \mathbf{I} & 0 \end{pmatrix} \mathbf{A}^{k+1} \begin{pmatrix} z_0 \\ z_0 \end{pmatrix} \right\|_2 ,
\end{aligned}
$$

where the step $\overset{\text{(a)}}{=}$ uses the fact that $z_0 = r_0 e_1$ is along the minimum eigenvector direction of $\mathbf{H}$; the step $\overset{b}{\leq}$ is based on the fact that $\sigma \leq \tilde{\rho} B r_0 |a_i - b_i|$; the step $\overset{c}{\leq}$ uses Lemma 36 in [28]. From Lemma 38 in [28], we have

$$
|a_i - b_i| \geq \frac{\theta}{2} \left(1 + \frac{\theta}{2} \right)^i \geq \frac{\theta}{2},
$$

and thus $\tilde{\rho} B r_0 |a_i - b_i| \geq \dfrac{\tilde{\rho} B \zeta r \theta}{2\sqrt{d_x}} \geq \sigma$. From the parameter settings, we have

$$
20\tilde{\rho} B \eta K \left(\frac{2}{\theta} + K \right) \leq \frac{1}{2}.
$$

Therefore, we complete the induction, which yields

$$
\begin{aligned}
\|z_K\|_2 &\geq \left\| \begin{pmatrix} \mathbf{I} & 0 \end{pmatrix} \mathbf{A}^{K} \begin{pmatrix} z_0 \\ z_0 \end{pmatrix} \right\|_2 - \left\| \eta \begin{pmatrix} \mathbf{I} & 0 \end{pmatrix} \sum_{i=0}^{K-1} \mathbf{A}^{K-i-1} \begin{pmatrix} \omega_i \\ 0 \end{pmatrix} \right\|_2 \\
&\geq \frac{1}{2} \left\| \begin{pmatrix} \mathbf{I} & 0 \end{pmatrix} \mathbf{A}^{K} \begin{pmatrix} z_0 \\ z_0 \end{pmatrix} \right\|_2 = \frac{r_0}{2} |a_K - b_K| \\
&\geq \frac{\theta r_0}{4} \left(1 + \frac{\theta}{2} \right)^K \geq 5B,
\end{aligned}
$$

where we use Lemma 38 in [28] and $\eta \lambda_{\min} \leq -\theta^2$ in the third inequality and the last step comes from $K = \frac{2}{\theta} \log(\frac{20B}{\theta r_0})$. However, from equation (37) we can obtain:

$$
\|z_K\|_2 \leq \left\| x'_K - x'_0 \right\|_2 + \left\| x''_K - x''_0 \right\|_2 + \left\| x'_0 - x''_0 \right\|_2 \leq 2B + 2r \leq 4B,
$$

which leads to contradiction. Thus we conclude that at least one of the iteration triggers the "if condition" and we finish the proof. ■

Now we prove Theorem 16.

Proof From Lemma 32, we know that Algorithm 3 will terminate in at most $\mathcal{O}\left(\Delta\sqrt{\tilde{\rho}}\chi^5\epsilon^{-3/2}\right)$ epochs. Since each epoch needs at most $K = \mathcal{O}\left(\chi\left(\tilde{L}^2/(\epsilon\tilde{\rho})\right)^{1/4}\right)$ gradient evaluations, the total number of gradient evaluations must be less than $\mathcal{O}\left(\Delta\tilde{L}^{1/2}\tilde{\rho}^{1/4}\chi^6\epsilon^{-1.75}\right)$.
Now we consider the last epoch. Similar to the proof of Theorem 13, we also have

$$\|\nabla\Phi(\hat{w})\|_2 \le \frac{2\sqrt{2}B}{\eta K^2} + \frac{2\theta B}{\eta K} + 4\tilde{\rho}B^2 + \sigma \le \frac{\epsilon}{\chi^3} + \epsilon^2 \le \epsilon\,.$$

If $\lambda_{\min}(\nabla^2\Phi(x_{t,\mathcal{K}})) \ge -\sqrt{\epsilon\tilde{\rho}}$, then from the perturbation theory of eigenvalues of Bhatia [4], we have

$$\begin{aligned}
|\lambda_j(\nabla^2\Phi(\hat{w}_{t+1})) - \lambda_j(\nabla^2\Phi(x_{t,\mathcal{K}}))| &\le \left\|\nabla^2\Phi(\hat{w}_{t+1}) - \nabla^2\Phi(x_{t,\mathcal{K}})\right\|_2 \\
&\le \tilde{\rho}\,\|\hat{w}_{t+1} - x_{t,\mathcal{K}}\|_2 \\
&\le \tilde{\rho}\,\|\hat{w}_{t+1} - x_{t+1,0}\|_2 + \tilde{\rho}r \\
&\le 3\tilde{\rho}B
\end{aligned}$$

for any j, where we use $\|\hat{w}_{t+1} - x_{t+1,0}\|_2 \le \frac{1}{K_0+1}\sum_{k=0}^{K_0}\|w_{t+1,k} - x_{t+1,0}\|_2 \le 2B$ in the last inequality. Then we have

$$\begin{aligned}
\lambda_j(\nabla^2\Phi(\hat{w}_{t+1})) &\ge \lambda_j(\nabla^2\Phi(x_{t,\mathcal{K}})) - |\lambda_j(\nabla^2\Phi(\hat{w}_{t+1})) - \lambda_j(\nabla^2\Phi(x_{t,\mathcal{K}}))| \\
&\ge -\sqrt{\epsilon\tilde{\rho}} - 3\tilde{\rho}B \ge -1.011\sqrt{\epsilon\tilde{\rho}}.
\end{aligned}$$

Now we consider $\lambda_{\min}(\nabla^2\Phi(x_{t,\mathcal{K}})) < -\sqrt{\epsilon\tilde{\rho}}$. Define the stuck region in $\mathbb{B}(r)$ centered at $x_{t,\mathcal{K}}$ to be the set of points starting from which the "if condition" does not trigger in K iterations, that is, the algorithm terminates and outputs a saddle point. From Lemma 33, we know that the length along the minimum eigen-direction of $\nabla^2\Phi(x_{t,\mathcal{K}})$ is less than r_0. Therefore, the probability of the starting point $x_{t+1,0} = x_{t,\mathcal{K}} + \xi_t$ located in the stuck region is less than

$$\frac{r_0 V_{d-1}(r)}{V_d(r)} \le \frac{r_0\sqrt{d}}{r} \le \zeta,$$

where we let $r_0 = \frac{\zeta r}{\sqrt{d}}$. Thus, the output $\hat{w}$ satisfies $\lambda_{\min}(\nabla^2\Phi(\hat{w}) \ge -1.011\sqrt{\epsilon\tilde{\rho}}$ with probability at least $1-\zeta$. This completes our whole proof of Theorem 16. ■

E.2. Proof of Proposition 17

The proof of Proposition 17 is similar to that of Proposition 14. We provide the poof for Proposition 17 as follows.
Proof We first consider the iterations of CG in Algorithm 3 in one epoch. We choose $T'_{t,k}$ as

$$T'_{t,k} = \begin{cases} \left\lceil \frac{\sqrt{\kappa}+1}{2}\log\left(\frac{4\ell\sqrt{\kappa}}{\sigma}\left(\|v_{0,-1}\|_2 + \frac{M}{\mu}\right)\right)\right\rceil, & k=0, \\ \left\lceil \frac{\sqrt{\kappa}+1}{2}\log\left(\frac{4\ell\sqrt{\kappa}}{\sigma}\left(\frac{\sigma}{2\ell} + \frac{2M}{\mu}\right)\right)\right\rceil, & k \ge 1. \end{cases} \tag{38}$$

Following the proof of that in Section D.2 in exact fashions we arrive at that equation (4) in Condition 10 can hold.

The total iterates of CG when running `PRAHGD` in Algorithm 3 in one epoch satisfies

$$\begin{aligned}
&\sum_{k=0}^{\mathcal{K}-1} T'_k \\
&\leq \mathcal{K} + \frac{\sqrt{\kappa}+1}{2}\left(\frac{2T'_0}{\sqrt{\kappa}+1} + \sum_{k=1}^{\mathcal{K}-1}\log\left(\frac{4\ell\sqrt{\kappa}}{\sigma}\left(\frac{\sigma}{2\ell}+\frac{2M}{\mu}\right)\right)\right) \\
&= \mathcal{K} + \frac{\sqrt{\kappa}+1}{2}\left(\frac{2T'_0}{\sqrt{\kappa}+1} + (\mathcal{K}-1)\log\left(\frac{4\ell\sqrt{\kappa}}{\sigma}\left(\frac{\sigma}{2\ell}+\frac{2M}{\mu}\right)\right)\right) \\
&= \mathcal{K} + \frac{\sqrt{\kappa}+1}{2}\mathcal{K}\left(\frac{1}{\mathcal{K}}\log\left(\frac{4\ell\sqrt{\kappa}}{\sigma}\left(\|v_{0,-1}\|_2+\frac{M}{\mu}\right)\right) + \left(1-\frac{1}{\mathcal{K}}\right)\log\left(\frac{4\ell\sqrt{\kappa}}{\sigma}\left(\frac{\sigma}{2\ell}+\frac{2M}{\mu}\right)\right)\right) .
\end{aligned}$$

Now we consider the iterations of AGD `PRAHGD` in Algorithm 3 in one epoch.

We first show the following lemma.

Lemma 34 *Consider the setting of Theorem 16, and we run* `PRAHGD` *in Algorithm 3, then we have*

$$\|y^*(w_{t,-1})\|_2 \leq \tilde{C}$$

for any $t > 0$*, where* $\tilde{C} = \|y^*(x_{0,0})\|_2 + (2B + B^2 + \eta\sigma + \eta C)\kappa\Delta\sqrt{\tilde{\rho}}\epsilon^{-3/2}$.

Then we choose $T_{t,k}$ as

$$T_{t,k} = \begin{cases} \left\lceil 2\sqrt{\kappa}\log\left(\frac{2\tilde{L}\sqrt{\kappa+1}}{\sigma}\tilde{C}\right)\right\rceil, & k=-1 \\ \left\lceil 2\sqrt{\kappa}\log\left(\frac{2\tilde{L}\sqrt{\kappa+1}}{\sigma}\left(\frac{\sigma}{2\tilde{L}}+2\kappa B\right)\right)\right\rceil, & k \geq 0 \end{cases} \tag{39}$$

We will use induction to show that Lemma 34 as well as equation (3) in Condition 10 will hold.

For $t = 0$, Lemma 34 hold trivially. Then we use induction with respect to k to prove that

$$\|y_{t,k} - y^*(w_{t,k})\|_2 \leq \frac{\sigma}{2\tilde{L}}$$

holds for any $k \geq -1$. For $k = -1$, Lemma 1 directly implies

$$\|y_{t,-1} - y^*(w_{t,-1})\|_2 \leq \frac{\left\|y^*(w_{t,-1})\right\|_2}{\tilde{C}} \cdot \frac{\sigma}{2\tilde{L}} \leq \frac{\sigma}{2\tilde{L}},$$

where the second inequality is based on Lemma 34. Suppose it holds that $\|y_{t,k-1} - y^*(w_{t,k-1})\|_2 \leq \frac{\sigma}{2\tilde{L}}$ for any $k \leq k'-1$, then we have

$$\begin{aligned}
&\left\|y_{t,k'} - y^*(w_{t,k'})\right\|_2 \\
&\leq \sqrt{1+\kappa}\left(1-\frac{1}{\sqrt{\kappa}}\right)^{T_{t,k'}/2}\left\|y_{t,k'-1} - y^*(w_{t,k'})\right\|_2 \\
&\leq \sqrt{1+\kappa}\left(1-\frac{1}{\sqrt{\kappa}}\right)^{T_{t,k'}/2}\left(\left\|y_{t,k'-1} - y^*(w_{t,k'-1})\right\|_2 + \left\|y^*(w_{t,k'-1}) - y^*(w_{t,k'})\right\|_2\right)
\end{aligned}$$

$$
\begin{aligned}
&\le \sqrt{1+\kappa}\left(1-\frac{1}{\sqrt{\kappa}}\right)^{T_{t,k'}/2}\left(\frac{\sigma}{2\tilde{L}}+\kappa\left\|w_{t,k'-1}-w_{t,k'}\right\|_2\right)\\
&\le \sqrt{1+\kappa}\left(1-\frac{1}{\sqrt{\kappa}}\right)^{T_{t,k'}/2}\left(\frac{\sigma}{2\tilde{L}}+2\kappa B\right)\\
&\le \frac{\sigma}{2\tilde{L}}\ ,
\end{aligned}
$$

where the first inequality is based on Lemma 1, the second one uses triangle inequality, the third one is based on induction hypothesis and Lemma 5, the fourth one uses equation (12), and the last step use the definition of $T_{t,k}$. Therefore, equation (3) in Condition 10 can hold.

Suppose Lemma 34 and equation (3) in Condition 10 hold for any $t \le t'-1$, then we have shown that when we choose $T'_{t,k}$ as defined in equation (38), then equation (4) in Condition 10 can hold. Thus, from Lemma 12 we obtain that:

$$
\|\nabla\Phi(w_{t,k})-\hat{\nabla}\Phi(w_{t,k})\|_2 \le \sigma\ . \tag{40}
$$

For any epoch $t \le t'-1$, we have

$$
\begin{aligned}
&\|x_{t,\mathcal{K}}-x_{t,0}\|_2\\
&=\|x_{t,\mathcal{K}}-x_{t,\mathcal{K}-1}+x_{t,\mathcal{K}-1}-x_{t,0}\|_2\\
&=\left\|(1-\theta)(x_{t,\mathcal{K}-1}-x_{t,\mathcal{K}-2})-\eta\hat{\nabla}\Phi(w_{t,\mathcal{K}-1})+x_{t,\mathcal{K}-1}-x_{t,0}\right\|_2\\
&\le \|x_{t,\mathcal{K}-1}-x_{t,\mathcal{K}-2}\|_2+\|x_{t,\mathcal{K}-1}-x_{t,0}\|_2+\eta\left\|\hat{\nabla}\Phi(w_{t,\mathcal{K}-1})\right\|_2\\
&\le 2B+\eta\left\|\hat{\nabla}\Phi(w_{t,\mathcal{K}-1})-\nabla\Phi(w_{t,\mathcal{K}-1})+\nabla\Phi(w_{t,\mathcal{K}-1})\right\|_2\\
&\le 2B+\eta\left\|\hat{\nabla}\Phi(w_{t,\mathcal{K}-1})-\nabla\Phi(w_{t,\mathcal{K}-1})\right\|_2+\eta\left\|\nabla\Phi(w_{t,\mathcal{K}-1})\right\|_2\\
&\le 2B+\eta\sigma+\eta\left\|\nabla\Phi(w_{t,\mathcal{K}-1})\right\|_2\\
&\le 2B+\eta(\sigma+C)
\end{aligned}
\tag{41}
$$

for some constant C. Here we use triangle inequality in the first inequality; equation (10) in the second one; triangle inequality again in the third one; equation (40) in the fourth one and equation (33) in the last one.

Then for t'-th epoch, we have

$$
\begin{aligned}
&\left\|y^*(w_{t',-1})-y^*(x_{0,0})\right\|_2\\
&\le \kappa\left\|w_{t',-1}-x_{0,0}\right\|_2\\
&=\kappa\left\|x_{t',0}-x_{0,0}\right\|_2\\
&=\kappa\left\|x_{t'-1,\mathcal{K}}-x_{0,0}\right\|_2\\
&\le \kappa(\left\|x_{t'-1,0}-x_{0,0}\right\|_2+\left\|x_{t'-1,\mathcal{K}}-x_{t'-1,0}\right\|_2+r)\\
&\le \kappa(\left\|x_{t'-1,0}-x_{1,0}\right\|_2+(2B+B^2+\eta\sigma+\eta C))\\
&\le (2B+B^2+\eta\sigma+\eta C)\kappa t\ ,
\end{aligned}
$$

where the first inequality is based on the Lipschitz continuous of $y^*(x)$ shown in Lemma 5; the second one uses triangle inequality; the third one is based on equation (41), and the last one uses induction. Then we have

$$\left\|y^*(w_{t',-1})\right\|_2 \le \|y^*(x_{0,0})\|_2 + B\kappa t' \le \|y^*(x_{0,0})\|_2 + \frac{(2B+B^2+\eta\sigma+\eta C)\kappa\Delta\sqrt{\tilde{\rho}}}{\epsilon^{3/2}},$$

where we use Lemma 32 in the last inequality.

Similarly with the case $t=0$, we use induction with respect to k again, and then we can prove that equation (3) in Condition 10 hold.

This also completes the proof for Lemma 34.

The total gradient calls from AGD in Algorithm 3 in one epoch satisfies

$$\begin{aligned}
\sum_{k=-1}^{\mathcal{K}-1} T_{t,k} &\le 2\sqrt{\kappa}\left(\frac{T_{-1}}{2\sqrt{\kappa}} + \sum_{k=0}^{\mathcal{K}-1}\log\left(\sqrt{\kappa+1} + \frac{4\tilde{L}\kappa\sqrt{\kappa+1}B}{\sigma}\right)\right) + \mathcal{K}+1 \\
&= 2\sqrt{\kappa}\left(\frac{T_{-1}}{2\sqrt{\kappa}} + \mathcal{K}\log\left(\sqrt{\kappa+1} + \frac{4\tilde{L}\kappa\sqrt{\kappa+1}B}{\sigma}\right)\right) + \mathcal{K}+1 \\
&= 2\sqrt{\kappa}\mathcal{K}\left(\frac{1}{\mathcal{K}}\log\left(\frac{2\tilde{L}\sqrt{\kappa+1}}{\sigma}\tilde{C}\right) + \log\left(\sqrt{\kappa+1} + \frac{4\tilde{L}\kappa\sqrt{\kappa+1}B}{\sigma}\right)\right) + \mathcal{K}+1.
\end{aligned}$$

This finishes our whole proof for Proposition 17. ∎

E.3. Proof of Corollary 18

Proof From Theorem 16, we have that PRAHGD in Algorithm 3 can find an $(\epsilon, \sqrt{\tilde{\rho}\epsilon})$ SOSP within at most $\mathcal{O}\left(\Delta\tilde{L}^{1/2}\tilde{\rho}^{1/4}\chi^6\epsilon^{-7/4}\right)$ iterations in the outer loop. Then we have

$$Gc(f,\epsilon) = \mathcal{O}\left(\frac{\Delta\tilde{L}^{1/2}\tilde{\rho}^{1/4}\chi^6}{\epsilon^{7/4}}\right) \quad \text{and} \quad JV(g,\epsilon) = \mathcal{O}\left(\frac{\Delta\tilde{L}^{1/2}\tilde{\rho}^{1/4}\chi^6}{\epsilon^{7/4}}\right).$$

Omitting polylogarithmic factor and pluging $\tilde{L} = \mathcal{O}(\kappa^3)$ and $\tilde{\rho} = \mathcal{O}(\kappa^5)$ into it, we have

$$Gc(f,\epsilon) = \tilde{\mathcal{O}}\left(\kappa^{11/4}\epsilon^{-7/4}\right) \quad \text{and} \quad JV(g,\epsilon) = \tilde{\mathcal{O}}\left(\kappa^{11/4}\epsilon^{-7/4}\right).$$

Lemma 32 shows that PRAHGD in Algorithm 3 will terminate in at most $\mathcal{O}\left(\frac{\Delta\sqrt{\tilde{\rho}}\chi^5}{\epsilon^{3/2}}\right)$ epochs. From Proposition 17 we can obtain that for each epoch t, we have the inner loops

$$\sum_{k=-1}^{\mathcal{K}-1} T_{t,k} \le \mathcal{O}\left(\kappa^{1/2}\mathcal{K}\log(1/\epsilon)\right) \quad \text{and} \quad \sum_{k=0}^{\mathcal{K}-1} T'_{t,k} \le \mathcal{O}\left(\kappa^{1/2}\mathcal{K}\log(1/\epsilon)\right)$$

hold. Then we have

$$Gc(g,\epsilon) = \mathcal{O}\left(\frac{\Delta\tilde{L}^{1/2}\tilde{\rho}^{1/4}\kappa^{1/2}\chi^6\log(1/\epsilon)}{\epsilon^{7/4}}\right) \text{ and } HV(g,\epsilon) = \mathcal{O}\left(\frac{\Delta\tilde{L}^{1/2}\tilde{\rho}^{1/4}\kappa^{1/2}\chi^6\log(1/\epsilon)}{\epsilon^{7/4}}\right),$$

where we use $\mathcal{K} \le K = \mathcal{O}\big(\chi\big(\tilde{L}^2/(\epsilon\tilde{\rho})\big)^{1/4}\big)$. Omit polylogarithmic factor and plug $\tilde{L} = \mathcal{O}(\kappa^3)$ and $\tilde{\rho} = \mathcal{O}(\kappa^5)$ into it, we have

$$Gc(g,\epsilon) = \tilde{\mathcal{O}}\left(\kappa^{13/4}\epsilon^{-7/4}\right) \qquad \text{and} \qquad HV(g,\epsilon) = \tilde{\mathcal{O}}\left(\kappa^{13/4}\epsilon^{-7/4}\right) .$$

■

This completes our proof of Corollary 18.

Appendix F. Proofs for Section B

In this section, we provide the proof of Theorem 20.

Proof Lemma 19 shows that in minimax problem settings, $\tilde{L} = (\kappa+1)\ell$ and $\tilde{\rho} = 4\sqrt{2}\kappa^3\rho$. Recall that our PRAGDA evolves directly from PRAHGD —- removing the CG step in PRAHGD because we do not need to compute the Hessian-vector products when solving the minmax problem. Therefore, we can straightforwardly apply the theoretical results for PRAHGD.

Applying Theorem 16, we have that Algorithm 4 can find an $\big(\epsilon, \mathcal{O}(\kappa^{1.5}\sqrt{\epsilon}\,)\big)$-SOSP.

Now we provide the gradient oracle calls complexities. From Lemma 32, we know that Algorithm 4 will terminate in at most $\mathcal{O}\big(\Delta\sqrt{\tilde{\rho}}\chi^5\epsilon^{-3/2}\big)$ epochs. Proposition 17 shows that, for each t, the total iteration number of AGD step satisfies:

$$\sum_{k=-1}^{\mathcal{K}-1} T_{t,k} \le \mathcal{O}(\kappa^{1/2}\mathcal{K}\log(1/\epsilon)) .$$

Recall that $\mathcal{K} \le K = \mathcal{O}\left(\chi\big(\tilde{L}^2/(\epsilon\tilde{\rho})\big)^{1/4}\right)$, we have that the total gradient oracle calls is at most:

$$\mathcal{O}\left(\frac{\Delta\tilde{\rho}^{1/4}\tilde{L}^{1/2}\kappa^{1/2}\chi^6\log(1/\epsilon)}{\epsilon^{7/4}}\right) .$$

Hide polylogarithmic factor and plug $\tilde{L}$ and $\tilde{\rho}$ into it, we have the total gradient oracle calls within at most $\tilde{\mathcal{O}}(\kappa^{7/4}\epsilon^{-7/4})$. ■

Appendix G. Empirical Studies

We conducted a series of experiments to validate the theoretical contributions presented in this paper. Specifically, we evaluated the effectiveness of our proposed algorithms, RAHGD and PRAHGD, by applying them to two different tasks: data hyper-cleaning for the MNIST dataset and hyperparameter optimization of logistic regression for the 20 News Group dataset. Our experiments demonstrate that our algorithms outperform several established baseline algorithms, such as BA, AID-BiO, ITD-BiO, and PAID-BiO, with much faster convergence rates. Additionally, we conducted a synthetic minimax problem experiment, wherein our PRAGDA algorithm exhibits a faster convergence rate when compared to iMCN.

G.1. Synthetic Minimax Problem

We construct the following nonconvex-strong-concave minimax problem:

$$\min_{x\in\mathbb{R}^3}\max_{y\in\mathbb{R}^2} f(x,y) = w(x_3) - 10y_1^2 + x_1y_1 - 5y_2^2 + x_2y_2,$$

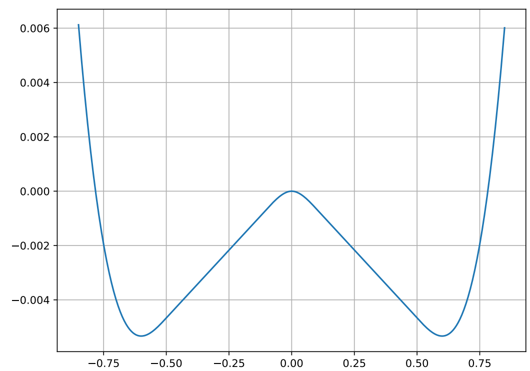
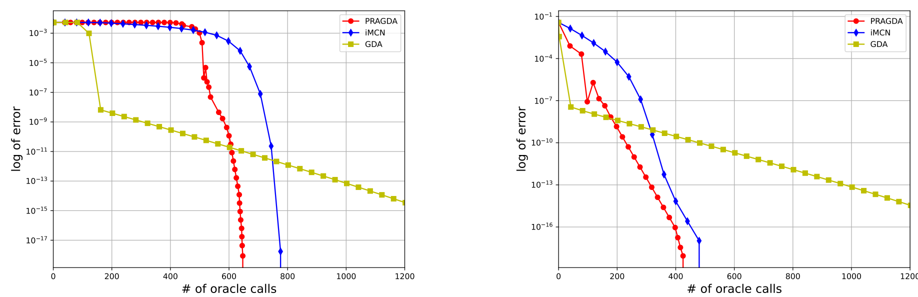


Figure 1: W-shape function designed by Tripuraneni et al. [47]



(*a*) Initial point (x_1, y_1) (*b*) Initial point (x_2, y_2)

Figure 2: Comparison of various minmax algorithms at different initial points

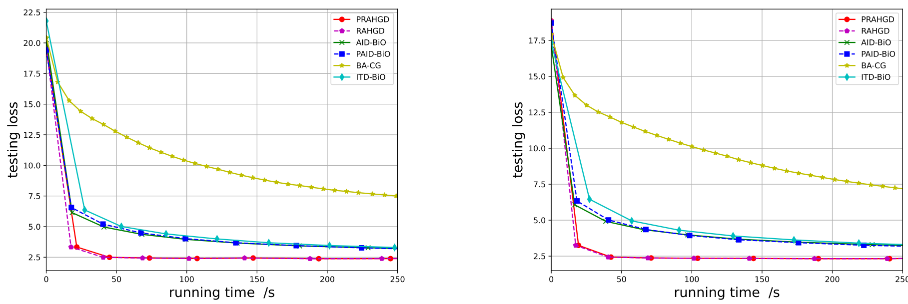
where $x = [x_1, x_2, x_3]^T$ and $y = [y_1, y_2]^T$ and

$$w(x) = \begin{cases} \sqrt{\epsilon}(x + (L+1)\sqrt{\epsilon})^2 - \frac{1}{3}(x + (L+1)\sqrt{\epsilon})^3 - \frac{1}{3}(3L+1)\epsilon^{3/2}, & x \le -L\sqrt{\epsilon}; \\ \epsilon x + \frac{\epsilon^{3/2}}{3}, & -L\sqrt{\epsilon} < x \le -\sqrt{\epsilon}; \\ -\sqrt{\epsilon}x^2 - \frac{x^3}{3}, & -\sqrt{\epsilon} < x \le 0; \\ -\sqrt{\epsilon}x^2 + \frac{x^3}{3}, & 0 < x \le \sqrt{\epsilon}; \\ -\epsilon x + \frac{\epsilon^{3/2}}{3}, & \sqrt{\epsilon} < x \le L\sqrt{\epsilon}; \\ \sqrt{\epsilon}(x - (L+1)\sqrt{\epsilon})^2 + \frac{1}{3}(x - (L+1)\sqrt{\epsilon})^3 - \frac{1}{3}(3L+1)\epsilon^{3/2}, & L\sqrt{\epsilon} < x; \end{cases}$$

is the W-shape-function [47] and we set $\epsilon = 0.01, L = 5$ in our experiment. We visualize the $w(\cdot)$ in Figure 1. It is easy to verify that $(x_0, y_0) = ([0,0,0]^\top, [0,0]^\top)$ is a saddle point of $f(x, y)$. We construct our experiment with two different initial points:

$$(x_1, y_1) = \left([10^{-3}, 10^{-3}, 10^{-16}]^\top, [0,0]^\top\right) \qquad \text{and} \qquad (x_2, y_2) = \left([0,0,1]^\top, [0,0]^\top\right).$$

Note that (x_1, y_1) is close to (x_0, y_0) while (x_2, y_2) is far from (x_0, y_0). We compare our PRAGDA with iMCN [38] algorithm and classical GDA [35] algorithm. The results are shown in Figure 2. We use a grid search to choose the learning rate of AGD steps and GDA and outer-loop learning rate of PRAGDA from $\{c \times 10^i : c \in \{1, 5\}, i \in \{1, 2, 3\}\}$ and the momentum parameter from $\{c \times 0.1 : c \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}\}$.



(a) Corruption rate p = 0.2 (b) Corruption rate p = 0.4

Figure 3: Comparison of various bilevel algorithms for data hypercleaning at different corruption rate

G.2. Data Hypercleaning

Data hypercleaning [15, 44] is an application example of bilevel optimization. In practice, we may have a dataset with label noise and we could only offer some time or cost to clean-up a subset of the noise data. To train a model in such a setting, we can treat the cleaned data as the validation set and the rest data as the training set. Then it can be transferred into a bilevel optimization:

$$\begin{aligned} &\min_{\lambda \in \mathbb{R}^{N_{\text{tr}}}} f(W^*(\lambda), \lambda) \triangleq \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{(x_i, y_i) \in \mathcal{D}_{\text{val}}} - \log(y_i^\top W^*(\lambda) x_i) \\ &\text{s.t.} \quad W^*(\lambda) = \operatorname*{arg\,min}_{W \in \mathbb{R}^{d_y \times d_x}} g(W, \lambda) \triangleq \frac{1}{|\mathcal{D}_{\text{tr}}|} \sum_{(x_i, y_i) \in \mathcal{D}_{\text{tr}}} -\sigma(\lambda_i) \log(y_i^\top W x_i) + C_r ||W||^2 \,, \end{aligned} \tag{42}$$

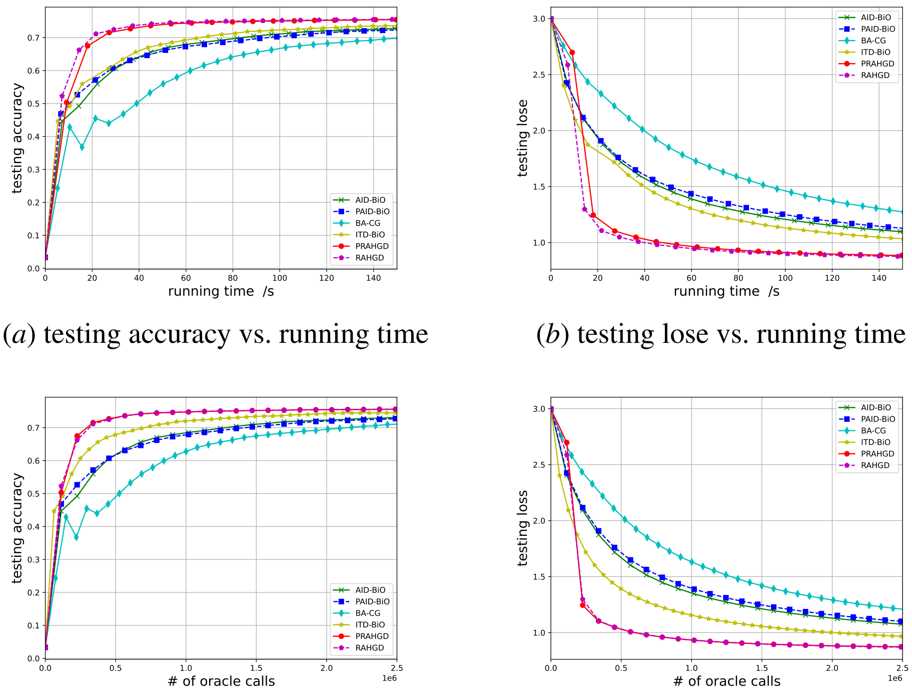
where $\mathcal{D}_{\text{tr}} = \{(x_i, y_i)\}$ is the training dataset, $\mathcal{D}_{\text{val}} = \{(x_i, y_i)\}$ is the validation dataset, W is the weight of the classifier, $\lambda_i \in \mathbb{R}$, σ is the sigmoid function and C_r is a regularization parameter. Following Shaban et al. [44] and Ji et al. [25], we choose $C_r = 0.001$.

We conducted the experiment on MNIST[31]. We have $x \in \mathbb{R}^{785}$, $y \in \mathbb{R}^{10}$ and $W \in \mathbb{R}^{10 \times 785}$ in equation (42). Training set contains 20,000 images, some of which have their labels randomly disrupted. We called the ratio of images with disrupted labels as corruption rate p. Validation set contains 5,000 images with correct labels. The testing set consists 10,000 images. The results are shown in Figure 3.

For BA algorithm proposed by Ghadimi and Wang [17], we also use conjugate gradient descent method to compute the Hessian vector since they didn't specify it and we called it BA-CG in Figure 3. For all algorithms, We choose the inner-loop learning rate and outer-loop learning rate from $\{0.001, 0.01, 0.1, 1, 10\}$ and the iteration number of CG step from $\{3, 6, 12, 24\}$. For all algorithms except BA, we choose the iteration number of GD or AGD steps from $\{50, 100, 200, 500, 1000\}$ and for BA algorithm, as adopted by Ghadimi and Wang [17], we choose the iteration number of GD steps from $\{\lceil c(k+1)^{1/4} \rceil : c \in \{0.5, 1, 2, 4\}\}$. We observe that both `RAHGD` and `PRAHGD` converge faster than other algorithms.

G.3. Hyperparameter Optimization

Hyperparameter optimization is a classical bilevel problem. The goal of hyperparameter optimization is to find the optimal hyperparameter to minimize the loss on the validation dataset. We compare the performance of our algorithm `RAHGD` and `PRAHGD` with the baseline algorithms listed in Table 1 and Table 2 over a logistic regression problem on 20 News group dataset[20]. This dataset consists



(a) testing accuracy vs. running time (b) testing lose vs. running time

(c) testing accuracy vs. # of oracle calls (d) testing lose vs. # of oracle calls

Figure 4: Comparison of various bilevel algorithms on logistic regression on 20 Newsgroup dataset. Figure (a) and (b) show the results of testing accuracy and testing lose vs.running time respectively. Figure (c) and (d) show the results of testing accuracy and testing lose vs. # of oracles calls respectively.

of 18,846 news items divided into 20 topics, and features include 130,170 tf-idf sparse vectors. We divided the data into three parts: 5,657 for training, 5,657 for validation, and 7,532 for testing. Then the objective function of this problem can be written in the following form.

$$\min_{\lambda \in \mathbb{R}^p} \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{(x_i,y_i)\in\mathcal{D}_{\text{val}}} L(w^*(\lambda); x_i, y_i)$$

$$s.t. \;\; w^*(\lambda) = \arg\min_{w \in \mathbb{R}^{c\times p}} \frac{1}{|\mathcal{D}_{\text{tr}}|} \sum_{(x_i,y_i)\in\mathcal{D}_{\text{tr}}} L(w; x_i, y_i) + \frac{1}{2cp} \sum_{j=1}^{c} \sum_{k=1}^{p} \exp(\lambda_k) w_{jk}^2,$$

where $\mathcal{D}_{\text{tr}} = \{(x_i, y_i)\}$ is the training dataset, $\mathcal{D}_{\text{val}} = \{(x_i, y_i)\}$ is the validation dataset, L is the cross-entropy loss function, $c = 20$ is the number of topics and $p = 130,170$ is the dimension of features. Same as that in section G.2, we use the conjugate gradient descent method to approximate the Hessian vector.

For all algorithms listed in Figure 4, we choose the inner-loop learning rate and out-loop learning rate from $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$, the iteration number of GD or AGD step from $\{5, 10, 30, 50\}$, and the iteration number of CG step from $\{5, 10, 30, 50\}$. For BA-CG, we choose the iteration number of GD steps from $\{\lceil c(k+1)^{1/4} \rceil : c \in \{0.5, 1, 2, 4\}\}$ as adopted by Ghadimi and Wang [17]. The results are shown in Figure 4. We observe that our `RAHGD` and `PRAHGD` converge faster than other algorithms.

Table 1: Comparison of complexities for nonconvex bilevel optimization algorithms of finding approximate FOSPs.

Algorithm	$Gc(f,\epsilon)$	$Gc(g,\epsilon)$	$JV(g,\epsilon)$	$HV(g,\epsilon)$
BA [17]	$\mathcal{O}(\kappa^4\epsilon^{-2})$	$\mathcal{O}(\kappa^5\epsilon^{-2.5})$	$\mathcal{O}(\kappa^4\epsilon^{-2})$	$\tilde{\mathcal{O}}(\kappa^{4.5}\epsilon^{-2})$
AID-BiO [25]	$\mathcal{O}(\kappa^3\epsilon^{-2})$	$\mathcal{O}(\kappa^4\epsilon^{-2})$	$\mathcal{O}(\kappa^3\epsilon^{-2})$	$\mathcal{O}(\kappa^{3.5}\epsilon^{-2})$
ITD-BiO [25]	$\mathcal{O}(\kappa^3\epsilon^{-2})$	$\tilde{\mathcal{O}}(\kappa^4\epsilon^{-2})$	$\tilde{\mathcal{O}}(\kappa^4\epsilon^{-2})$	$\tilde{\mathcal{O}}(\kappa^4\epsilon^{-2})$
`RAHGD` (this work)	$\tilde{\mathcal{O}}(\kappa^{2.75}\epsilon^{-1.75})$	$\tilde{\mathcal{O}}(\kappa^{3.25}\epsilon^{-1.75})$	$\tilde{\mathcal{O}}(\kappa^{2.75}\epsilon^{-1.75})$	$\tilde{\mathcal{O}}(\kappa^{3.25}\epsilon^{-1.75})$

Table 2: Comparison of complexities for nonconvex bilevel optimization algorithms of finding approximate SOSPs.

Algorithm	$Gc(f,\epsilon)$	$Gc(g,\epsilon)$	$JV(g,\epsilon)$	$HV(g,\epsilon)$
Perturbed AID [23]	$\tilde{\mathcal{O}}(\kappa^3\epsilon^{-2})$	$\tilde{\mathcal{O}}(\kappa^4\epsilon^{-2})$	$\tilde{\mathcal{O}}(\kappa^3\epsilon^{-2})$	$\tilde{\mathcal{O}}(\kappa^{3.5}\epsilon^{-2})$
`PRAHGD` (this work)	$\tilde{\mathcal{O}}(\kappa^{2.75}\epsilon^{-1.75})$	$\tilde{\mathcal{O}}(\kappa^{3.25}\epsilon^{-1.75})$	$\tilde{\mathcal{O}}(\kappa^{2.75}\epsilon^{-1.75})$	$\tilde{\mathcal{O}}(\kappa^{3.25}\epsilon^{-1.75})$

— $Gc(f,\epsilon)$ and $Gc(g,\epsilon)$: number of gradient evaluations w.r.t. f and g. κ: condition number of the lower-level objective.

— $JV(g,\epsilon)$: number of Jacobian-vector products $\nabla^2_{xy}g(x,y)v$. Notation $\tilde{\mathcal{O}}$ omits a polylogarithmic factor in relevant parameters.

— $HV(g,\epsilon)$: number of Hessian-vector products $\nabla^2_{yy}g(x,y)v$.

Algorithm 1 AGD$(h, z_0, T, \alpha, \beta)$

1: **Input:** objective $h(\cdot)$; initialization z_0; iteration number $T \geq 1$; step-size $\alpha > 0$; momentum param. $\beta \in (0,1)$
2: $\tilde{z}_0 \leftarrow z_0$
3: **for** $t \leftarrow 0, \ldots, T-1$ **do**
4: $\quad z_{t+1} \leftarrow \tilde{z}_t - \alpha \nabla h(\tilde{z}_t)$
5: $\quad \tilde{z}_{t+1} \leftarrow z_{t+1} + \beta(z_{t+1} - z_t)$
6: **end for**
7: **Output:** z_T

Algorithm 2 $\mathrm{CG}(A, b, T, q_0)$

Input: quadratic objective (as in Eq. (2)); initialization q_0; iteration number $T \geq 1$
$r_0 \leftarrow Aq_0 - b$, $p_0 \leftarrow -r_0$
for $t \leftarrow 0, \ldots, T-1$ **do**
  $\alpha_t \leftarrow \dfrac{r_t^T r_t}{p_t^T A p_t}$
  $q_{t+1} \leftarrow q_t + \alpha_t p_t$
  $r_{t+1} \leftarrow r_t + \alpha_t A p_t$
  $\beta_{t+1} \leftarrow \dfrac{r_{t+1}^T r_{t+1}}{r_t^T r_t}$
  $p_{t+1} \leftarrow -r_{t+1} + \beta_{t+1} p_t$
end for
Output: q_T

Algorithm 3 (Perturbed) Restarted Accelerated HyperGradient Descent, (P)RAHGD

Input: initial vector $x_{0,0}$; step-size $\eta > 0$; momentum parameter $\theta \in (0,1)$; parameters $\alpha > 0, \beta \in (0,1), \{T_{t,k}\}$ of AGD; parameter $\{T'_{t,k}\}$ of CG; iteration threshold $K \geq 1$; parameter B for triggering restarting; perturbation radius $r > 0$; option $\texttt{Perturbation} \in \{0,1\}$
$k \leftarrow 0,\ t \leftarrow 0,\ x_{0,-1} \leftarrow x_{0,0},\ y_{0,-1} \leftarrow \mathrm{AGD}(g(x_{1,-1},\ \cdot\ ), 0, T_{0,-1}, \alpha, \beta),\ v_{0,-1} \leftarrow y_{0,-1}$
while $k < K$
  $w_{t,k} \leftarrow x_{t,k} + (1-\theta)(x_{t,k} - x_{t,k-1})$
  $y_{t,k} \leftarrow \mathrm{AGD}(g(w_{t,k},\ \cdot\ ), y_{t,k-1}, T_{t,k}, \alpha, \beta)$
  $v_{t,k} \leftarrow \mathrm{CG}(\nabla^2_{yy} g(w_{t,k}, y_{t,k}), \nabla_y f(w_{t,k}, y_{t,k}), T'_{t,k}, v_{t,k-1})$
  $u_{t,k} \leftarrow \nabla_x f(w_{t,k}, y_{t,k}) - \nabla^2_{xy} g(w_{t,k}, y_{t,k}) v_{t,k}$
  $x_{t,k+1} \leftarrow w_{t,k} - \eta u_{t,k}$
  $k \leftarrow k+1$
  if $k \sum_{i=0}^{k-1} \|x_{t,i+1} - x_{t,i}\|^2 > B^2$
    $v_{t+1,-1} \leftarrow v_{t,k}$
    $x_{t+1,0} \leftarrow \begin{cases} x_{t,k}, & \text{if } \texttt{Perturbation} = 0 \\ x_{t,k} + \xi \ \text{ with } \ \xi \sim \mathrm{Unif}(\mathbb{B}(r)), & \text{if } \texttt{Perturbation} = 1 \end{cases}$
    $x_{t+1,-1} \leftarrow x_{t+1,0}$
    $k \leftarrow 0, t \leftarrow t+1$
    $y_{t,-1} \leftarrow \mathrm{AGD}(g(x_{t,-1},\ \cdot\ ), 0, T_{t,-1}, \alpha, \beta)$
  end if
end while
$K_0 \leftarrow \arg\min_{\lfloor \frac{K}{2} \rfloor \leq k \leq K-1} \|x_{t,k+1} - x_{t,k}\|_2$
Output: $\hat{w} \leftarrow \frac{1}{K_0+1} \sum_{k=0}^{K_0} w_{t,k}$

Algorithm 4 PRAGDA

Input: initial vector $x_{0,0}$; step-size $\eta > 0$; momentum param. $\theta \in (0,1)$; params. $\alpha > 0, \beta \in (0,1), \{T_{t,k}\}$ of AGD; iteration threshold $K \geq 1$; param. B for triggering restarting; perturbation radius $r > 0$
$k \leftarrow 0,\ t \leftarrow 0,\ x_{0,-1} \leftarrow x_{0,0}$
$y_{0,-1} \leftarrow \mathrm{AGD}(-\bar{f}(x_{0,-1},\,\cdot\,), 0, T_{0,-1}, \alpha, \beta)$
while $k < K$
    $w_{t,k} \leftarrow x_{t,k} + (1-\theta)(x_{t,k} - x_{t,k-1})$
    $y_{t,k} \leftarrow \mathrm{AGD}(-\bar{f}(w_{t,k},\,\cdot\,), y_{t,k-1}, T_{t,k}, \alpha, \beta)$
    $x_{t,k+1} \leftarrow w_{t,k} - \eta \nabla_x \bar{f}(w_{t,k}, y_{t,k})$
    $k \leftarrow k+1$
    if $k \sum_{i=0}^{k-1} \|x_{t,i+1} - x_{t,i}\|^2 > B^2$
        $x_{t+1,0} \leftarrow x_{t,k} + \xi,\ \xi \sim \mathrm{Unif}(\mathbb{B}(r))$
        $x_{t+1,-1} \leftarrow x_{t+1,0}$
        $k \leftarrow 0,\ t \leftarrow t+1$
        $y_{t,-1} \leftarrow \mathrm{AGD}(-\bar{f}(x_{t,-1},\,\cdot\,), 0, T_{t,-1}, \alpha, \beta)$
    end if
end while
$K_0 \leftarrow \arg\min_{\lfloor \frac{K}{2} \rfloor \leq k \leq K-1} \|x_{t,k+1} - x_{t,k}\|_2$
Output: $\hat{w} \leftarrow \frac{1}{K_0+1} \sum_{k=0}^{K_0} w_{t,k}$