Zeroth-order Optimization with Weak Dimension Dependency

Pengyun Yue YUEPY@PKU.EDU.CN
Long Yang YANGLONG001@PKU.EDU.CN
National Key Lab of General AI, School of Intelligence Science and Technology, Peking University

Cong Fang[✉] FANGCONG@PKU.EDU.CN
National Key Lab of General AI, School of Intelligence Science and Technology, Peking University
Institute for Artificial Intelligence, Peking University

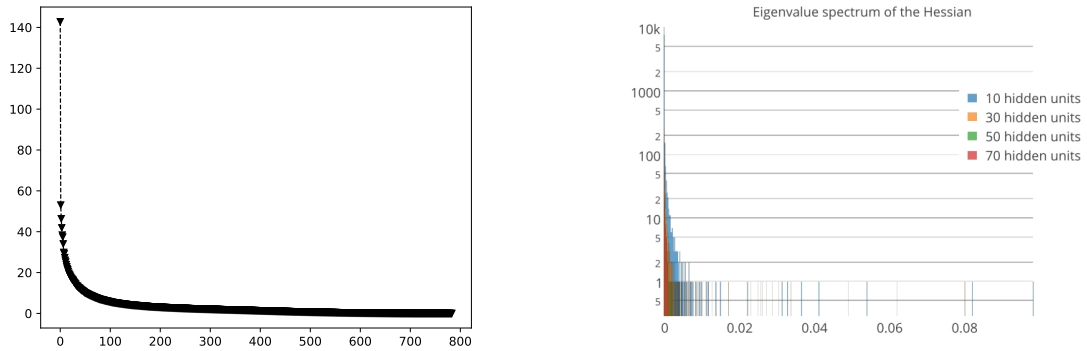
Zhouchen Lin[✉] ZLIN@PKU.EDU.CN
National Key Lab of General AI, School of Intelligence Science and Technology, Peking University
Institute for Artificial Intelligence, Peking University
Peng Cheng Laboratory

Editors: Gergely Neu and Lorenzo Rosasco

Abstract

Zeroth-order optimization is a fundamental research topic that has been a focus of various learning tasks, such as black-box adversarial attacks, bandits, and reinforcement learning. However, in theory, most complexity results assert a linear dependency on the dimension of optimization variable, which implies paralyzations of zeroth-order algorithms for high-dimensional problems and cannot explain their effectiveness in practice. In this paper, we present a novel zeroth-order optimization theory characterized by complexities that exhibit weak dependencies on dimensionality. The key contribution lies in the introduction of a new factor, denoted as $\mathrm{ED}_\alpha = \sup_{\mathbf{x}\in\mathbb{R}^d} \sum_{i=1}^d \sigma_i^\alpha(\nabla^2 f(\mathbf{x}))$ ($\alpha > 0$, $\sigma_i(\cdot)$ is the i-th singular value in non-increasing order), which effectively functions as a measure of dimensionality. The algorithms we propose demonstrate significantly reduced complexities when measured in terms of the factor ED_α. Specifically, we first study a well-known zeroth-order algorithm from Nesterov and Spokoiny (2017) on quadratic objectives and show a complexity of $\mathcal{O}\left(\frac{\mathrm{ED}_1}{\sigma_d}\log(1/\epsilon)\right)$ for the strongly convex setting. For linear regression, such a complexity is dimension-free and outperforms the traditional result by a factor of d under common conditions. Furthermore, we introduce novel algorithms that leverages the Heavy-ball mechanism to enhance the optimization process. By incorporating this acceleration scheme, our proposed algorithm exhibits a complexity of $\mathcal{O}\left(\frac{\mathrm{ED}_{1/2}}{\sqrt{\sigma_d}}\cdot\log\frac{L}{\mu}\cdot\log(1/\epsilon)\right)$. For linear regression, under some mild conditions, it is faster than state-of-the-art algorithms by $\sqrt{d}$. We further expand the scope of the method to encompass generic smooth optimization problems, while incorporating an additional Hessian-smooth condition. By considering this extended framework, our approach becomes applicable to a broader range of optimization scenarios. The resultant algorithms demonstrate remarkable complexities, with dimension-independent dominant terms that surpass existing algorithms by an order in d under appropriate conditions. Our analysis lays the foundation for investigating zeroth-order optimization methods for smooth functions within high-dimensional settings.

Keywords: zeroth-order optimization, effective dimension, convergence rate



(*a*) (*b*)

Figure 1: (a) The eigenvalues of the Gram matrix of data on MNIST (see Deng (2012)), which are also the eigenvalues of Hessian on the least square model (b) The eigenvalues of a three-layer neural network on MNIST. (b) is taken directly from Sagun et al. (2016).

1. Introduction

Consider the unconstrained optimization program:

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}). \tag{1}$$

We study solving (1) using *zeroth-order* oracles which only return the function value $f(\hat{\mathbf{x}})$ given point $\hat{\mathbf{x}}$ and improve such oracle complexities in searching suitable approximated solutions.

Zeroth-order optimization is a fundamental research topic serving as a prototype module for numerous tasks, including black-box optimization (Grill et al., 2015), adversarial attacks (Ye et al., 2019), bandits (Bubeck et al., 2017), as well as reinforcement learning (RL) (Salimans et al., 2017). From the theoretical aspect, one notable common feature among wide studies (see works in Section 2.1) is that the complexities of zeroth-order algorithms have a linear dimension dependency. For instance, consider a standard program where the objective is assumed to be μ-strongly convex and have L-Lipschitz continuous gradients. The well-known algorithm proposed by Nesterov and Spokoiny (2017) called $\mathcal{RG}_\rho$ achieves a complexity of $\mathcal{O}\left(\frac{dL}{\mu}\log\left(\frac{1}{\epsilon}\right)\right)$ to find an ϵ-approximated solution $\tilde{\mathbf{x}}$ such that $f(\tilde{\mathbf{x}}) - \min f \leq \epsilon$. The main idea of $\mathcal{RG}_\rho$ is solving a smoothed surrogate of f, whose stochastic gradients can be efficiently computed via zeroth-order oracles of f, using stochastic gradient descent. Compared with the Gradient Descent algorithm, $\mathcal{RG}_\rho$ is d-times slower. This result is reasonable and seems unimprovable in the worst case because a gradient oracle offers information that can be quantified as a d-dimensional vector in contrast to 1 of such from a function value oracle.

In practice, the dimension d can be very large in modern real-world applications. For instance, a high-resolution adversarial image has thousands of pixels. Worse still, the state numbers in contextual bandits or RL always encounter combinatorial explosions. The existing theoretical results indicate potential limitations of zeroth-order algorithms in high-dimensional problems, which seemingly contradict the observed success of these algorithms in practical applications over the past years. For example, hundreds steps of $\mathcal{RG}_\rho$ suffices to find an adversarial image (Ye et al., 2019). By estimating the objective function using (deep) neural networks, a series of RL algorithms have achieved surprising performances for decision-making (Mania et al., 2018; Choromanski et al., 2018;

Salimans et al., 2017). These phenomena appear mysterious from a theoretical optimization view and require new analysis to understand the underlying reasons.

To bridge the gap between theory and practice, this paper develops a zeroth-order optimization theory which exhibits complexities with weak dimension dependencies. The underlying intuition behind our theory revolves around the introduction of an effective dimension for zeroth-order optimization. This idea has been widely considered in the era of machine learning and statistics (see related works in Section 2.2.1), where one usually studies the required number of data for a learning task. We follow a similar philosophy and show that much fewer zeroth-order oracles and iteration complexities are inherently needed when certain effective dimension is small, because only a small amount of components contribute to most of the difficulties in zeroth-order optimization. To formalize our intuition, we introduce the concept of the effective dimension in zeroth-order optimization by $\mathrm{ED}_\alpha = \sup_{\mathbf{x}\in\mathbb{R}^d} \sum_{i=1}^d \sigma_i^\alpha(\nabla^2 f(\mathbf{x}))$ ($\alpha > 0$), where $\sigma_i(\cdot)$ is the i-th singular value in non-increasing order. If the objective has L-Lipschitz continuous gradients, one can assert $\mathrm{ED}_\alpha \le dL^\alpha$. We show that under various suitable conditions, the dependence of complexities on the factor dL^α can be enhanced to ED_α. We shall note that in practice one often has $\mathrm{ED}_\alpha \ll dL^\alpha$ because the singular values of Hessian matrices often decrease very fast. See Fig. 1 as two examples which plot eigenvalues of Hessian matrices for a convex and a non-convex function, respectively. To obtain a quantitative comparison between ED_α and dL^α, we also study some realizable cases on linear regression in Section 4.

Now we briefly introduce our complexity results under different settings. In Section 5, we first consider a basic setting where the objective is a convex quadratic function. We study the standard $\mathcal{RG}_\rho$ algorithm and show complexities of $\tilde{\mathcal{O}}\left(\frac{\mathrm{ED}_1}{\sigma_d}\right)$ and $\tilde{\mathcal{O}}\left(\frac{\mathrm{ED}_1}{\epsilon}\right)$ for strongly convex and weakly convex settings, respectively, where $\tilde{\mathcal{O}}$ hides polylogarithmic terms. For linear regression where the ℓ_2 norm of the data is normalized to a constant level, these complexities are dimension-free and outperform the traditional results by the factor d. In Section 6, we consider acceleration. We propose a new Heavy-ball based algorithm, called HB-ZGD that achieves a complexity of $\tilde{\mathcal{O}}\left(\frac{\mathrm{ED}_{1/2}}{\sqrt{\sigma_d}}\right)$ for strongly convex functions. For linear regression where the data is normalized, our algorithm is faster than the state-of-the-art algorithm (Nesterov and Spokoiny, 2017) by $\sqrt{d}$. The novelty in our complexity analysis stems from the utilization of a special Mahalanobis norm, denoted as $\|\cdot\|_{[\nabla^2 f]^2}$. This unique approach allows us to demonstrate that the function value descends more rapidly, on average. In Section 7, we extend the HB-ZGD method to encompass generic convex and non-convex optimization problems under an additional H-Hessian-smooth condition. The idea is to combine HB-ZGD with cubic-regularization tricks (Nesterov and Polyak, 2006; Monteiro and Svaiter, 2013). For generic convex optimization, we obtains a complexity of $\tilde{\mathcal{O}}\left(\mathrm{ED}_{1/2}\epsilon^{-1/2} + d\epsilon^{-2/7}\right)$ against the best-known complexity of $\mathcal{O}\left(d\epsilon^{-1/2}\right)$ from Nesterov and Spokoiny (2017). For general non-convex optimization, we consider finding a second-order stationary point and establish a complexity of $\tilde{\mathcal{O}}\left(\mathrm{ET}_{1/2}\epsilon^{-7/4} + d\epsilon^{-3/2}\right)$ against the best-known complexity of $\tilde{\mathcal{O}}\left(d\epsilon^{-7/4}\right)$ from Jin et al. (2017).

The significance of our work is two-folded. 1) By introducing an effective dimension for zeroth-order optimization, we provide a more realistic analysis for zeroth-order algorithms. Our upper bound complexities suggests that the zeroth-order optimization are usually not very hard, providing explanations for their practical successes. 2) Based on our framework, one is able to design more efficient zeroth-order algorithms under a variety of settings. We summarize the main contributions of this work in the following.

(a) We propose to use ED_α as the effective dimension to characterize the complexities in zeroth-order optimization. This optimization model is more close to practice.

(b) For quadratic objectives, we provide an improved analysis for $\mathcal{RG}_\rho$ (Nesterov and Spokoiny, 2017) and design an accelerated algorithm. We establish new dimension-independent complexities.

(c) For generic convex and non-convex optimization, we propose provable faster algorithms with weak dimension dependency using the cubic regularization tricks.

2. Related Works

2.1. Zeroth-order Optimization

In zeroth-order optimization, the algorithms only access the objective function value to find a designed solution. We review three main lines of research to design zeroth-order algorithms.

The first research line is to estimate the gradient using zeroth-order oracles and then design algorithms using the techniques from first-order optimization, which is more related to our paper. One typical algorithm is the $\mathcal{RG}_\rho$ in Nesterov and Spokoiny (2017). For objective functions that are μ-strongly convex and have L-Lipschitz continuous gradient, $\mathcal{RG}_\rho$ achieves a complexity of $\mathcal{O}\left(\frac{dL}{\mu}\log(1/\epsilon)\right)$ and can be accelerated using the momentum technique to achieve a complexity of $\mathcal{O}\left(d\sqrt{\frac{L}{\mu}}\log(1/\epsilon)\right)$. In the generic non-convex case, Nesterov and Spokoiny (2017) establish a complexity of $\mathcal{O}\left(\frac{dL}{\epsilon^2}\right)$ to find an approximated first-order stationary point. There are many works that propose variants of $\mathcal{RG}_\rho$, such as proximal (Gasnikov et al., 2016) and stochastic (Ghadimi and Lan, 2013) versions. All complexities obtained by existing works have a linear dimension dependency. And we will improve the results using the proposed effective dimension.

Another popular line of research is to consider a function approximation to the objective function (see e.g. Moulines and Bach (2011)). Specifically, the way is to estimate the objective with a white-box model and balance the exploration and exploitation. The method is closely related to Bayesian optimization (see e.g. Srinivas et al. (2009)). The complexities of these algorithms are established often in a "statistical" style: they depend on the Hypothesis capacity of the model and approximation error. From our view, our proposed algorithms can be recognized as using simple linear or quadratic functions to locally approximate the objective function. Essentially, we combine analytical methods in optimization and statistics. Specifically, the complexities are described using some geometric characterizations that are commonly used and practical to model the objective in optimization, such as the gradient Lipchistz constant, with a certain effective dimension, a common concept in statistical learning. In special, we show one can often save the oracles to inaccurately estimate the gradient (locally linear approximation). It is interesting to extend our framework to study the general function approximation and we leave such important analysis as future work.

Last but not at least, one more research line is to design algorithms for more specific tasks. Typical examples are online bandits (see e.g. Bubeck et al. (2017)), and model-free RL (see e.g. Mania et al. (2018)).There are additional challenges to deal with these problems including the varying environments and randomization from policies. Many works achieve to design more efficient algorithms in terms of low regret bounds. This paper only studies vanilla zeroth-order optimization. We also leave to apply our framework on these specific learning tasks as non-trivial future works.

2.2. Related Techniques

2.2.1. EFFECTIVE DIMENSION

The idea of an effective dimension has long been considered in the eras of machine learning and statistics. For example, in manifold learning (see e.g. Cayton (2005)), one often assumes the data is embedded in a low dimensional space. In nonparametric estimation (see e.g. Wainwright (2019, Chapter 13)), one often considers the additive structure of the target function, where only a small number of dimensions combine. More related, in linear regression, Zhang (2005) introduces effective dimensions on the data Gram matrices to characterize the difficulties for ridge regression and obtain complexities independent of d. In high-dimensional regression, one often assumes a s-sparse response between the output and input signals, under which much fewer observations ($s\log(d)$ in comparison d) are needed to determine the relations (see e.g. Zhang and Zhang (2012)). There are also effective dimension analysis on RL (see e.g. Jin et al. (2021)), whereas, our paper focuses on generic zeroth-order optimization. Freund et al. (2022) study Langevin sampling and show the convergence rate can be dimensional free. From our view, they in effect use ED_2. Our work generalizes theirs to the optimization field and considers much broader settings.

2.2.2. CUBIC REGULARIZATION ALGORITHMS

Our work follows the cubic regularization tricks to work on generic optimization frameworks. Cubic regularization algorithms can be viewed as ingenious pre-conditioned Newton methods, whose updates often involve a minimization problem with the objective composed of a quadratic function with a simple third-order regularization term that can be solved using matrix inversion and a binary search. It is shown by Nesterov (2007) that the cubic regularization algorithm achieves non-asymptotic $\mathcal{O}\left(\epsilon^{-1/3}\right)$ complexity for convex optimization when the objective has uniformly continuous Hessian matrices, which outperforms the first-order algorithm with the complexity of $\mathcal{O}\left(\epsilon^{-1/2}\right)$, whereas, the convergence of vanilla Newton method can be ensured only when the initial is close to a minimizer. Monteiro and Svaiter (2013) further studies accelerations. The best-known complexity $\mathcal{O}\left(\epsilon^{-2/7}\right)$ is obtained by tensor methods from Gasnikov et al. (2019) in the convex case, which is proved to be optimal in the worst case (Arjevani et al., 2019). In the non-convex world, Nesterov and Polyak (2006) show a complexity of $\mathcal{O}(\epsilon^{-3/2})$ to find a second-order stationary point when treating the problem-dependent parameters as constants.

3. Preliminary

3.1. Notations

We use the convention $\mathcal{O}(\cdot)$, $\Omega(\cdot)$, and $\Theta(\cdot)$, to denote lower, upper, both lower and upper bounds with a universal constant. $\tilde{\mathcal{O}}(\cdot)$ ignores the polylogarithmic terms. We use $\mathbf{I}_d$ to denote the identity matrix in d-dimensional Euclidean space, and omit the subscript when d is clear from the context. We use $\|\cdot\|$ to denote the operator norm of a matrix. Moreover, we use $\|\mathbf{x}\|$ to denote the Euclidean norm of a vector and $\|\mathbf{x}\|_{\mathbf{A}}$ to denote the Mahalanobis (semi)norm where $\mathbf{A}$ is a positive semi-definite matrix, i.e. $\|\mathbf{x}\|_{\mathbf{A}} = \sqrt{\mathbf{x}^\top \mathbf{A}\mathbf{x}}$. We use $\nabla f(\mathbf{x})$ and $\nabla^2 f(\mathbf{x})$ to denote the first- and second-order derivative of f. Moreover, let $\mathbf{x}^*$ be a minimizer of f if it exists and f^* be the minimum value.

3.2. Assumptions and Definitions

We present some basic definitions and assumptions that are commonly used to characterize the geometry of the objective in optimization.

Assumption 1 (Convexity) *We say f is convex if*

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|^2, \quad \forall \mathbf{x}, \mathbf{y},$$

where $\mu \geq 0$. Moreover, if $\mu > 0$, f is said to be μ-strongly convex.

Assumption 2 (L-gradient smoothness) *We say f is L-gradient smooth (or have L-Lipschitz continuous gradients), if*

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \quad \forall \mathbf{x}, \mathbf{y}.$$

Assumption 3 (H-Hessian smoothness) *We say f is H-Hessian smooth (or have H-Lipschitz continuous Hessian matrices), if*

$$\|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{y})\| \leq H\|\mathbf{x} - \mathbf{y}\|, \quad \forall \mathbf{x}, \mathbf{y}.$$

For an optimization algorithm starting at $\mathbf{x}^0$, we introduce the following two commonly-used quantities to describe the distance between the initial to an optimal solution.

Definition 1 (Δ-bounded function value) *Let $\Delta = f(\mathbf{x}^0) - f^*$.*

Definition 2 (D-bounded distance to the optimal solution) *Assume the minimizer of f exists. Let $\mathbf{X}^*$ be the set of all minimizers. Define $D = \inf_{\mathbf{x}^* \in \mathbf{X}^*} \sup\{\|\mathbf{x} - \mathbf{x}^*\| : f(\mathbf{x}) \leq f(\mathbf{x}^0)\}$.*

For convex problems, we consider finding an ϵ-approximated solution defined below:

Definition 3 (ϵ-optimal solution) *$\mathbf{x}$ is an ϵ-optimal solution of f if $f(\mathbf{x}) - f^* \leq \epsilon$.*

For non-convex problems, we study finding an $(\epsilon, \mathcal{O}(\sqrt{\epsilon}))$-approximated second-order stationary point with definition below:

Definition 4 ((ϵ, δ)-SSP) *$\mathbf{x}$ is said to be an (ϵ, δ)-approximated second-order stationary point (SSP) of f if it admits*

$$\|\nabla f(\mathbf{x})\| \leq \epsilon, \qquad \nabla^2 f(\mathbf{x}) \succeq -\delta \mathbf{I}.$$

It is known that a second-order stationary point is an optimal solution when the objective function satisfies the so-called strict-saddle condition Ge et al. (2015). In our complexity analysis, we will often consider the case where L, H, Δ, and D are in constant level, and focus on dependencies on μ, d, and ϵ.

4. Effective Dimension

Without any specification, we always assume the objective function f is second-order derivative. We introduce the effective dimension of zeroth-order optimization by

$$\mathrm{ED}_\alpha = \sup_{\mathbf{x}\in\mathbb{R}^d} \sum_{i=1}^{d} \sigma_i^\alpha(\nabla^2 f(\mathbf{x}))$$

where $\alpha > 0$ and $\sigma_i(\cdot)$ is the i-th singular value in non-increasing order. Note that we simply obtain ED_α by taking the supremum over $\mathbb{R}^d$, which is a global quantity to characterize the objective function. It is possible to consider a local effective dimension ED_α and then one can choose adaptive step sizes based on the local effective dimension. When the objective is convex, the singular values are the same as eigenvalues. For non-convex case, it is also possible to relax the singular values to positive eigenvalues. However, we omit its analysis in this paper.

For different algorithms, we may pick different α. For $\mathcal{RG}_\rho$, we pick $\alpha = 1$. When considering acceleration, α is picked as $\frac{1}{2}$. Freund et al. (2022) studies Langevin algorithm and essentially pick $\alpha = 2$. When the objective has L-Lipschitz continuous gradients, $\mathrm{ED}_\alpha \le dL^\alpha$ for all $\alpha > 0$. And the gap of ED_α to dL^α depends on how fast the singular values for the Hessian matrices decrease. We have a simple lemma by supposing a descending order of singular values.

Proposition 5 *Assume for any* $\mathbf{x}$ *and* $\alpha > 0$*, there exists constant* $C > 0$ *and* $\beta > 0$ *such that* $\sigma_i(\nabla^2 f(\mathbf{x})) \le \frac{C}{i^\beta}$ *for* $i \in [d]$*, then we have*

$$\mathrm{ED}_\alpha \le \begin{cases} \frac{2^{\alpha\beta-1}C^\alpha}{\alpha\beta-1}, & \alpha\beta > 1, \quad \textit{dimensional free}, \\ C^\alpha \log(2d+1), & \alpha\beta = 1, \quad \textit{logarithmic growth on } d, \\ \frac{C^\alpha}{1-\alpha\beta}(d+1)^{1-\alpha\beta}, & \alpha\beta < 1, \quad \textit{improve by a } \Theta\left(d^{\alpha\beta}\right) \textit{factor}. \end{cases} \tag{2}$$

In the following, we show realizable cases where ED_α is provably smaller than dL, which generalizes the work from Freund et al. (2022). Consider the objective admits the form as:

$$f(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} q_i(\boldsymbol{\beta}_i^\top \mathbf{x}), \tag{3}$$

with assumptions below.

Assumption 4 *The function* $q_i \in \mathcal{C}^2$ *has a bounded second derivative, i.e.* $q_i'' \le L_0$ *for all* $i \in [n]$.

Assumption 5 *For all* $i \in [N]$*, then norm of* $\boldsymbol{\beta}_i$ *is bounded by* R*, i.e.* $\|\boldsymbol{\beta}_i\|_2 \le R$.

For linear regression, $\boldsymbol{\beta}_i$ is associated with the data and can achieve Assumption 5 by normalization, and σ_i is associated with the loss function and holds Assumption 4 for ℓ_2 with $L_0 = 1$. Then we have the following lemma.

Proposition 6 *For the objective in* (3) *that satisfies Assumptions 4 and 5, we have*

$$\mathrm{ED}_\alpha \le \begin{cases} (L_0 R)^\alpha, & \alpha \ge 1, \quad \textit{dimensional free}, \\ (L_0 R)^\alpha d^{1-\alpha}, & \alpha < 1 \quad \textit{improve by a } \Theta\left(d^\alpha\right) \textit{factor}. \end{cases} \tag{4}$$

For two-layer neural networks, we have the following proposition:

Proposition 7 *Define $f(\mathbf{W}, \mathbf{w}) = \mathbf{w}^\top \sigma(\mathbf{W}^\top \mathbf{x})$, where σ is the activation function. When $\|\mathbf{x}\|_1 \leq r_1$, $\|\mathbf{w}\| \leq r_2$ and $\sigma''(x) \leq \alpha$, we have $\operatorname{tr}\left(\nabla^2 f(\mathbf{W}, \mathbf{w})\right) \leq \alpha r_1 r_2$.*

The requirements in Proposition 7 can be met in most settings. For deep neural networks, a similar argument can be obtained.

Finally, we note that for lots of parameterized models, the effective dimension can be small at least when the parameter is near its optimal solution. This is due to the fact that under weak regular conditions, the fisher information $\mathcal{I}(\theta) = -\mathbb{E}\left[\frac{\partial^2}{\partial \theta^2} \log f(X;\theta) \,|\, \theta\right] = \mathbb{E}\left[\left(\frac{\partial}{\partial \theta} \log f(X;\theta)\right)^2 \,|\, \theta\right]$. So if $\frac{\partial}{\partial \theta} \log f(X;\theta)$ is bounded, the effective dimension is also bounded.

5. Improved Analysis on Quadratic Minimization

In this section, we first provide an improved analysis for zeroth-order optimization on quadratic functions. Specifically, we assume that $f(\mathbf{x})$ is a L-smooth and convex quadratic function, which is in form as

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{A}\mathbf{x} + \mathbf{b}^\top \mathbf{x}. \tag{5}$$

We study the quadratic function because it is already very representative since (1) in theory, it is known that most worst-case functions (lower-bound instances) in the convex optimization are exactly quadratic (see e.g. Nesterov (2003, Chapter 2)); (2) in practice, quadratic functions include lots of applications in machine learning, such as least-square regression (Björck, 1996). Our result can be extended to work on objective functions with varying Hessian matrices. Here the Hessian matrices are needed to have a uniformly upper bound. For the sake of simplicity, we ignore such analysis.

We focus on the standard $\mathcal{RG}_\rho$ algorithm proposed by Nesterov and Spokoiny (2017). The idea of the algorithm in Nesterov and Spokoiny (2017) is to solve a smoothed surrogate of f defined as $\hat{f} = \mathbb{E}_{\boldsymbol{\xi}} f(\mathbf{x} + \rho\boldsymbol{\xi})$, where $\rho > 0$ is picked to be small enough and $\boldsymbol{\xi} \sim N(0, \mathbf{I})$. It is shown by Nesterov and Spokoiny (2017) that the stochastic gradient of $\hat{f}$ can be obtained by

$$\hat{\nabla}_\rho f(\mathbf{x}) = \frac{f(\mathbf{x} + \rho\boldsymbol{\xi}) - f(\mathbf{x})}{\rho}\boldsymbol{\xi}, \tag{6}$$

where $\boldsymbol{\xi} \sim N(0, \mathbf{I})$. Therefore, one can perform stochastic gradient method to solve $\hat{f}$.

We directly relate $\hat{\nabla}_\rho f(\mathbf{x})$ with $f(\mathbf{x})$. In fact, by the first-order Taylor expansion on f, the limit of $\hat{\nabla}_\rho f$ defined by $\tilde{\nabla} f(\mathbf{x})$ with ρ tending to zero admits

$$\tilde{\nabla} f(\mathbf{x}) \triangleq \lim_{\rho \to 0} \hat{\nabla}_\rho f(\mathbf{x}) = \langle \nabla f(\mathbf{x}), \boldsymbol{\xi} \rangle \cdot \boldsymbol{\xi}. \tag{7}$$

Therefore when ρ is sufficiently small, (6) returns the inner product of $\nabla f(\mathbf{x})$ and a given direction $\boldsymbol{\xi}$. Moreover, by the randomness of $\boldsymbol{\xi}$, we know that $\tilde{\nabla} f(\mathbf{x})$ is an unbiased estimator of $\nabla f(\mathbf{x})$ with the sum of variance on each component bounded by $\Theta\left(d\|\nabla f(\mathbf{x})\|^2\right)$. Specifically,

Lemma 8

$$\mathbb{E}_{\boldsymbol{\xi}} \tilde{\nabla} f(\mathbf{x}) = \nabla f(\mathbf{x}) \tag{8}$$

and

$$\mathbb{E}_{\boldsymbol{\xi}} \|\tilde{\nabla} f(\mathbf{x})\|^2 = \Theta(d\|\nabla f(\mathbf{x})\|^2). \tag{9}$$

Algorithm 1 $\mathcal{RG}_\rho$ (Nesterov and Spokoiny, 2017)

Input: $\mathbf{x}_0$
while *stopping criterion is not met* **do**
  generate $\hat{\nabla}_\rho f(\mathbf{x}_k)$ by two queries to the function value and a Gaussian random vector in (6);
  $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k - h_k \hat{\nabla}_\rho f(\mathbf{x}_k)$;
  $k \leftarrow k+1$;
end

A similar result of Lemma 8 is also shown by Nesterov and Spokoiny (2017). Lemma 8 suggests that in order to offset the effect of variance, one needs $\Omega(d)$ estimates to obtain a unbiased estimation of $\nabla f(\mathbf{x})$ with small variance under ℓ_2-norm. To improve the analysis on $\mathcal{RG}_\rho$, we first generalize the bound of ℓ_2-norm variance to the case for arbitrary Mahalanobis (semi)norm.

Lemma 9 *For symmetric matrix* $\mathbf{M}$*,*

$$\mathbb{E}_{\boldsymbol{\xi}} \|\tilde{\nabla} f(\mathbf{x})\|_{\mathbf{M}}^2 \leq 3\mathrm{tr}(\mathbf{M}) \|\nabla f(\mathbf{x})\|^2. \tag{10}$$

Lemma 9 brings a new insight by bridging the connections between $\mathcal{RG}_\rho$ (Nesterov and Spokoiny, 2017) and the effective dimension. It suggests studying $\mathcal{RG}_\rho$ under a specific Mahalanobis (semi)norm to obtain an improved analysis. For the quadratic objective function in (5), we can pick $\mathbf{M}$ as $\mathbf{A}$.

Now we are ready to state the improved analysis for $\mathcal{RG}_\rho$ (Nesterov and Spokoiny, 2017). $\mathcal{RG}_\rho$ is also shown in Algorithm 1. For the convenience of later analysis in Section 7, we consider a more general zeroth-order oracle. That is, we consider a zeroth-order oracle with δ-adversarial noise. Specifically, when given the input point $\tilde{\mathbf{x}}$, such oracle returns a noisy function value $\tilde{f}(\tilde{\mathbf{x}})$ that admits

$$\left|\tilde{f}(\tilde{\mathbf{x}}) - f(\tilde{\mathbf{x}})\right| \leq \delta, \quad \forall \mathbf{x} \in \mathbb{R}^d. \tag{11}$$

Here the noise can be adversarial. We call such oracle as δ-approximated zeroth-order oracle. When $\delta = 0$, δ-approximated zeroth-order oracle reduces to the standard zeroth-order oracle. We first consider the strongly convex setting. $\mathcal{RG}_\rho$ is shown in Algorithm 1, where we allow δ-approximated zeroth-order oracle to access function value. The convergence result is shown in Theorem 10.

Theorem 10 *Suppose f is a μ-strongly convex quadratic function and has L-Lipschitz continuous gradient. The Hessian matrix of f is $\mathbf{A}$. Let $h_k = \frac{1}{12\mathrm{tr}(\mathbf{A})}$. Using an δ-approximated zeroth-order oracle, $\{\mathbf{x}_k\}_{k\in\mathbb{N}}$ generated by $\mathcal{RG}_\rho$ satisfies*

$$\begin{aligned} &\mathbb{E} f(\mathbf{x}_{k+1}) - f^* - \frac{24\mathrm{tr}(\mathbf{A})}{\mu}\left(C_1\rho^2 + C_2\frac{\delta^2}{\rho^2}\right) \\ \leq &\left(1 - \frac{\mu}{24\mathrm{tr}(\mathbf{A})}\right)\left(\mathbb{E} f(\mathbf{x}_k) - f^* - \frac{24\mathrm{tr}(\mathbf{A})}{\mu}\left(C_1\rho^2 + C_2\frac{\delta^2}{\rho^2}\right)\right), \end{aligned} \tag{12}$$

where

$$C_1 = \frac{5}{16}\mathrm{tr}(\mathbf{A})d + \frac{5}{384}\mathrm{tr}(\mathbf{A}), \quad C_2 = \frac{d}{3\mathrm{tr}(\mathbf{A})} + \frac{1}{72\mathrm{tr}(\mathbf{A})}, \tag{13}$$

and the expectation is taken for all the randomness in the algorithm.

Theorem 10 shows that $\mathcal{RG}_\rho$ converges linearly in expectation when the hyper-parameter ρ and δ are picked small enough. Moreover, in order to find an ϵ-suboptimal point of f, $\mathcal{RG}_\rho$ needs $\mathcal{O}\left(\frac{\mathrm{ED}_1}{\mu}\log\frac{1}{\epsilon}\right)$ zeroth-order oracles and iteration complexities. Here, we compare the result with the original one in Nesterov and Spokoiny (2017), who establish a complexity of $\mathcal{O}\left(\frac{dL}{\mu}\log\frac{1}{\epsilon}\right)$ in expectation. Our analysis is sharper than theirs up to constants since $\mathrm{ED}_1 \leq dL$. The rationale behind our analysis is mentioned before: in most real cases (see Fig. 1), the singular values of Hessian matrices decrease very fast. So we often have $\mathrm{ED}_1 \ll dL$. To obtain a quantitative comparison between r_1 and dL, we consider linear models in (3) and have the following corollary.

Corollary 11 *For the objective in (3) that satisfies Assumptions 4 and 5 and is μ-strongly convex, $\mathcal{RG}_\rho$ finds an ϵ-suboptimal solution in $\tilde{\mathcal{O}}\left(\frac{L_0 R}{\mu}\right)$ in expectation.*

From Corollary 11, treating R and L_0 as constants, we establish a complexity of $\tilde{\mathcal{O}}(\mu^{-1})$ in comparison to $\tilde{\mathcal{O}}(d\mu^{-1})$ in Nesterov and Spokoiny (2017). Note here L can be $\Theta(1)$. Therefore we improve the complexity by the factor of d.

Now we consider the weakly convex setting. The result is shown in Theorem 12.

Theorem 12 *Suppose f is a L-smooth quadratic function whose Hessian matrix is $\mathbf{A}$. Using a δ-approximated zeroth-order oracle, $\{\mathbf{x}_k\}_{k\in\mathbb{N}}$ generated by $\mathcal{RG}_\rho$ satisfies*

$$\begin{aligned}&(k+1)(\mathbb{E}f(\mathbf{x}_{k+1}) - f^*)\\ \leq\ & k\mathbb{E}(f(\mathbf{x}_k) - f^*) + (k+1)\left(C_1\rho^2 + C_2\frac{\delta^2}{\rho^2}\right) + \frac{12\mathrm{tr}(\mathbf{A})}{k+1}\mathbb{E}\|\mathbf{x}_k - \mathbf{x}^*\|^2,\end{aligned} \tag{14}$$

where the expectation is taken for all the randomness in the algorithm.

Theorem 12 establishes a complexity of $\mathcal{O}\left(\frac{\mathrm{ED}_1}{\epsilon}\right)$ in expectation to find an ϵ-suboptimal point of f. Again, we compare our analysis with that in Nesterov and Spokoiny (2017) which achieve a complexity of $\mathcal{O}\left(\frac{dL}{\epsilon}\right)$ in the same setting. Again our result is sharper than theirs up to constants. For linear models, when R and L_0 are treated as constants, the following corollary indicates that our analysis improves the complexity by a d factor.

Corollary 13 *For the objective in (3) that satisfies Assumptions 4 and 5, $\mathcal{RG}_\rho$ finds an ϵ-suboptimal solution in $\mathcal{O}\left(\frac{L_0 R}{\epsilon}\right)$ oracle calls in expectation.*

6. Acceleration on Quadratic Minimization

It is well-known that first-order algorithms can be accelerated using the so-called momentum in convex optimization. For example, the earlier work from Polyak (1964) shows that the Heavy-ball algorithm can achieve a faster convergence asymptotically for strongly convex functions when Hessian matrices exist. Nesterov (2003) proposes several acceleration schemes and first obtains the accelerated rate in the weakly convex setting by introducing the famous estimate sequence method. For quadratic objective functions, techniques, such as Chebyshev's acceleration and conjugate gradient (see e.g. Young (2014)) are also applicable to reach the faster rate.

Algorithm 2 ZHB: Zeroth-order Heavy-Ball algorithm.

Input: $\mathbf{x}_0$, L-smooth and μ-strongly quadratic function f
$\beta \leftarrow \sqrt{h\mu}, h \leftarrow \frac{1}{14400^2 \mathrm{ED}_{1/2}(f)^2}$;
while *stopping criterion is not met* **do**
  $\mathbf{y}_n \leftarrow \mathbf{x}_n + (1-\beta)\mathbf{v}_n$;
  generate $\hat{\nabla}_\rho f(\mathbf{y}_n)$ by two queries to the function value and a Gaussian random vector in (6);
  $\mathbf{x}_{n+1} \leftarrow \mathbf{y}_n - h\hat{\nabla}_\rho f(\mathbf{y}_n)$;
  $\mathbf{v}_{n+1} \leftarrow \mathbf{x}_{n+1} - \mathbf{x}_n$;
end

Because $\mathcal{RG}_\rho$ can be regarded as a stochastic gradient algorithm where the variance of the stochasticity can be controlled, it is possible to perform acceleration using the technique in first-order optimization. Indeed, Nesterov and Spokoiny (2017) propose an acceleration algorithm using Nesterov's scheme in Nesterov (1983). We find such a framework is not directly applicable to obtain a dimensional-independent complexity. We instead consider a Heavy-ball based acceleration with the algorithm shown in Algorithm 2. One can find that Algorithm 2 simply replaces the exact gradient in the Heavy-ball algorithm by a random approximation using (6) and adaptively choose a different step size. We still study the quadratic objective in (5) and first focus on strongly convex case. Theorem 14 below summarizes our convergence result.

Theorem 14 *Suppose f is a L-smooth μ-strongly convex quadratic function whose Hessian matrix is $\mathbf{A}$. Using an δ-approximated zeroth-order oracle, if δ and ρ is small enough such that*

$$6n\left(\frac{16\delta}{\rho}+12\rho\mathrm{tr}(\mathbf{A})\right)<80\left(1-\frac{\mu^{1/2}}{57600\mathrm{ED}_{1/2}}\right)^{n-1}\cdot\mu\cdot(f(\mathbf{x}_0)-f^*), \tag{15}$$

$\{\mathbf{x}_n\}_{n\in\mathbb{N}}$ *generated by* ZHB *satisfies*

$$\mathbb{E}f(\mathbf{y}_n)-f^*\leq 400\left(1-\frac{\mu^{1/2}}{57600\mathrm{ED}_{\frac{1}{2}}}\right)^{n}\cdot\frac{L}{\mu}\cdot(f(\mathbf{x}_0)-f^*), \tag{16}$$

where the expectation is taken for all the randomness in the algorithm.

The proof idea of Theorem 14 is to treat each update as a vector multiplying a fixed matrix with error terms caused by the variance of estimation for $\nabla f(\mathbf{x})$ and use the eigenvalues of the fixed matrix to give a convergence rate. A similar idea also appears from Jin et al. (2017) in generic non-convex optimization, whereas, our novel perspective is to use a special Mahalanobis norm $\|\cdot\|_{[\nabla^2 f]^2}$.

Theorem 14 shows Algorithm 14 converges linearly with a complexity of $\tilde{\mathcal{O}}\left(\frac{\mathrm{ED}_{1/2}}{\sqrt{\mu}}\right)$ in expectation which improves the complexity of $\tilde{\mathcal{O}}\left(d\sqrt{\frac{L}{\mu}}\right)$ in Nesterov and Spokoiny (2017). For linear models, our analysis achieves a complexity of $\tilde{\mathcal{O}}\left(\frac{L_0^{1/2}R^{1/2}d^{1/2}}{\sqrt{\mu}}\right)$, improving the result of Nesterov and Spokoiny (2017) by at least $\sqrt{d}$ when R and L_0 are treated as constants and ignores polylogarithmic factors. Moreover, a fully dimension-free complexity can be obtained when the

eigenvalues of $\mathbf{A}$ decrease very fast. In particular, this requires $\sum_{k=1}^{d} \lambda_k^{1/2}(\mathbf{A}) \leq C$, which occurs, for example, when the eigenvalues decrease in $\frac{1}{k^\alpha}$ with $\alpha > 2$ from Proposition 5.

We note that using our technique, an improved analysis of the accelerated algorithm in Nesterov and Spokoiny (2017) can only obtain a complexity of $\tilde{\mathcal{O}}\left(\sqrt{\frac{d\mathrm{ED}_1}{\mu}}\right)$, which still has a dimension dependency and is more costly than $\tilde{\mathcal{O}}\left(\frac{\mathrm{ED}_{1/2}}{\sqrt{\mu}}\right)$ by a factor of $\frac{\sqrt{d\mathrm{ED}_1}}{\mathrm{ED}_{1/2}}$. When the eigenvalues of $\mathbf{A}$ decrease in $\frac{1}{k^\alpha}$ with $\alpha > 2$, Algorithm 2 is provably faster by $\sqrt{d}$.

To extend the acceleration on the weakly convex case. We use the standard reduction technique (see e.g. Lin et al. (2015)) by optimizing a surrogate function:

$$g(\mathbf{x}) = f(\mathbf{x}) + \frac{\epsilon}{2D^2}\|\mathbf{x}\|^2, \tag{17}$$

where ϵ is a tolerant error and D is defined in Definition 2. We have the following corollary.

Corollary 15 *For convex function f, ZHB with regularization technique needs a complexity of $\tilde{\mathcal{O}}\left(\mathrm{ED}_{1/2} \cdot \epsilon^{-1/2} + d\right)$ to find an ϵ-suboptimal point in expectation.*

7. Accelerated Algorithms for Generic High-order Smooth Functions

In this section, we consider optimizing generic functions using zeroth-order oracles. To extend the analysis for quadratic minimization to a more general case, we restrict the objective to have H-continuous Hessian matrices. The main idea to design faster algorithms is to combine Algorithm 2 with the cubic regularization tricks (Monteiro and Svaiter, 2013; Nesterov and Polyak, 2006). We should mention that this paper concentrates on obtaining improved complexities. It is *not* hard to simplify the designed algorithms using techniques such as Jin et al. (2017) and Fang et al. (2019). However, since the proofs are much more involved, we leave them as future works.

7.1. Convex Case

We first present an algorithm with an improved convergence rate for convex functions. The central idea is to adopt the large-step A-NPE method in Monteiro and Svaiter (2013) but considers an inexact solution for sub-problems and a binary search for hyper-parameters. The description of the detailed algorithm is shown in Appendix A.1.

It is shown by Monteiro and Svaiter (2013) that the iteration complexity of the original Large-step A-NPE can be upper bounded by $\mathcal{O}\left(H^{2/7}D^{6/7}\epsilon^{-2/7}\right)$ for convex Hessian-smooth objective functions, where each update is associated with a complex cubic regularized optimization sub-problem. By inexactly solving these subproblems with binary search and Algorithm 2, we establish a complexity upper bound for zeroth-order algorithms. Specifically,

Theorem 16 *Assume the objective function f is convex and has L-continuous gradient and H-continuous Hessian matrices. Algorithm 4 needs*

$$\tilde{\mathcal{O}}\left(\frac{D \cdot \mathrm{ED}_{1/2}}{\epsilon^{1/2}} + d \cdot D^{6/7}H^{2/7}\epsilon^{-2/7}\right) \tag{18}$$

zeroth-order oracle calls to find an ϵ-approximated solution with high probability.

From Theorem 16, if we treat L and H as constants, Algorithm 4 obtains a complexity of $\tilde{\mathcal{O}}\left(\mathrm{ED}_{1/2}\epsilon^{-1/2} + d\epsilon^{-2/7}\right)$, which is lower than the best-known complexity of $\mathcal{O}\left(d\epsilon^{-1/2}\right)$ in Nesterov and Spokoiny (2017) since $\mathrm{ED}_{1/2} \leq dL^{1/2}$ and usually $\mathrm{ED}_{1/2} \ll dL^{1/2}$ in practice.

7.2. Non-convex Case

We consider optimizing a second-order smooth function in the general non-convex setting. For non-convex programming, it is known that finding an approximated global minimizer for a smooth objective suffers the curse of dimensionality. We consider searching an $(\epsilon, \mathcal{O}(\sqrt{\epsilon}))$-approximated second-order stationary point (see Definition 4). Such a relaxed solution can be obtained in polynomial complexities and is already a tolerant solution for many machine learning problems such as for matrix decomposition problems (Ge et al., 2015).

We consider inexactly solving the cubic regularization algorithm in Nesterov and Polyak (2006) by zeroth-order oracles. The whole algorithm is shown in Appendix A.2. We then provide a complexity analysis. Recall that the standard cubic regularization algorithm Nesterov and Polyak (2006) finds a second-order approximated solution in $\mathcal{O}\left(H^{1/2}\Delta\epsilon^{-3/2}\right)$ for a generic H-Hessian smooth function. By including the complexities to solve the subproblems, we obtain an upper bound of zeroth-order complexity for Algorithm 8 in the Theorem 17 below.

Theorem 17 *Assume the objective function f is convex and has L-continuous gradients and H-continuous Hessian matrices. Algorithm 4 finds an $(\epsilon, \sqrt{H\epsilon})$-SSP of f in*

$$\tilde{\mathcal{O}}\left(\mathrm{ED}_{1/2}H^{1/4}\Delta\epsilon^{-7/4} + dH^{1/2}\Delta\epsilon^{-3/2}\right) \tag{19}$$

zeroth-order oracle calls with high probability.

From Theorem 17, by treating L and H as constants, Algorithm 4 obtains a complexity of $\tilde{\mathcal{O}}\left(\mathrm{ED}_{1/2}\epsilon^{-7/4} + d\epsilon^{-3/2}\right)$, whereas, the best-known complexity of $\tilde{\mathcal{O}}\left(d\epsilon^{-7/4}\right)$ from Jin et al. (2017) in the same setting. Again Algorithm 8 is provably faster.

8. Conclusion

This paper proposes zeroth-order optimization theory with weak dimension dependency. We propose a new factor ED_α to characterize the complexities. Our analysis provides a new way to study zeroth-order optimization for high-dimensional problems.

Acknowledgments

C. Fang and Z. Lin were supported by National Key R&D Program of China (2022ZD0160301). Z. Lin was also supported by the NSF China (No. 62276004), the major key project of PCL, China (No. PCL2021A12) and Qualcomm.

References

Yossi Arjevani, Ohad Shamir, and Ron Shiff. Oracle complexity of second-order methods for smooth convex optimization. *Mathematical Programming*, 178(1-2):327–360, November 2019. ISSN 0025-5610, 1436-4646. doi: 10.1007/s10107-018-1293-1. URL http://link.springer.com/10.1007/s10107-018-1293-1.

Åke Björck. *Numerical methods for least squares problems*. SIAM, 1996.

Sébastien Bubeck, Yin Tat Lee, and Ronen Eldan. Kernel-based methods for bandit convex optimization. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 72–85, 2017.

Lawrence Cayton. Algorithms for manifold learning. *Univ. of California at San Diego Tech. Rep*, 12 (1-17):1, 2005.

Krzysztof Choromanski, Mark Rowland, Vikas Sindhwani, Richard Turner, and Adrian Weller. Structured evolution with compact architectures for scalable policy optimization. In *International Conference on Machine Learning*, pages 970–978. PMLR, 2018.

Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

Cong Fang, Zhouchen Lin, and Tong Zhang. Sharp analysis for nonconvex sgd escaping from saddle points. In *Conference on Learning Theory*, pages 1192–1234. PMLR, 2019.

Yoav Freund, Yi-An Ma, and Tong Zhang. When is the convergence time of langevin algorithms dimension independent? a composite optimization viewpoint. *Journal of Machine Learning Research*, 23(214):1–32, 2022.

Alexander Gasnikov, Pavel Dvurechensky, Eduard Gorbunov, Evgeniya Vorontsova, Daniil Selikhanovych, and César A. Uribe. Optimal Tensor Methods in Smooth Convex and Uniformly ConvexOptimization. In *Proceedings of the Thirty-Second Conference on Learning Theory*, pages 1374–1391. PMLR, June 2019. URL https://proceedings.mlr.press/v99/gasnikov19a.html. ISSN: 2640-3498.

Alexander V Gasnikov, Anastasia A Lagunovskaya, Ilnura N Usmanova, and Fedor A Fedorenko. Gradient-free proximal methods with inexact oracle for convex stochastic nonsmooth optimization problems on the simplex. *Automation and Remote Control*, 77:2018–2034, 2016.

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

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Jean-Bastien Grill, Michal Valko, and Rémi Munos. Black-box optimization of noisy functions with unknown smoothness. *Advances in Neural Information Processing Systems*, 28, 2015.

Chi Jin, Praneeth Netrapalli, and Michael I. Jordan. Accelerated Gradient Descent Escapes Saddle Points Faster than Gradient Descent, November 2017. URL http://arxiv.org/abs/1711.10456. arXiv:1711.10456 [cs, math, stat].

Chi Jin, Qinghua Liu, and Sobhan Miryoosefi. Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. *Advances in neural information processing systems*, 34:13406–13418, 2021.

Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. *Advances in neural information processing systems*, 28, 2015.

Horia Mania, Aurelia Guy, and Benjamin Recht. Simple random search provides a competitive approach to reinforcement learning. *arXiv preprint arXiv:1803.07055*, 2018.

Renato D. C. Monteiro and B. F. Svaiter. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. *SIAM Journal on Optimization*, 23(2):1092–1125, 2013. doi: 10.1137/110833786. URL https://doi.org/10.1137/110833786.

Eric Moulines and Francis Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. *Advances in neural information processing systems*, 24, 2011.

Yu. Nesterov. Accelerating the cubic regularization of Newton's method on convex problems. *Mathematical Programming*, 112(1):159–181, July 2007. ISSN 0025-5610, 1436-4646. doi: 10.1007/s10107-006-0089-x. URL http://link.springer.com/10.1007/s10107-006-0089-x.

Yurii Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Proceedings of the USSR Academy of Sciences*, 269:543–547, 1983.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

Yurii Nesterov and B.T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, August 2006. ISSN 0025-5610, 1436-4646. doi: 10.1007/s10107-006-0706-8. URL http://link.springer.com/10.1007/s10107-006-0706-8.

Yurii Nesterov and Vladimir Spokoiny. Random Gradient-Free Minimization of Convex Functions. *Foundations of Computational Mathematics*, 17(2):527–566, April 2017. ISSN 1615-3375, 1615-3383. doi: 10.1007/s10208-015-9296-2. URL http://link.springer.com/10.1007/s10208-015-9296-2.

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964.

Levent Sagun, Leon Bottou, and Yann LeCun. Eigenvalues of the hessian in deep learning: Singularity and beyond. *arXiv preprint arXiv:1611.07476*, 2016.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.

Haishan Ye, Zhichao Huang, Cong Fang, Chris Junchi Li, and Tong Zhang. Hessian-Aware Zeroth-Order Optimization for Black-Box Adversarial Attack. Technical Report arXiv:1812.11377, arXiv, March 2019. URL http://arxiv.org/abs/1812.11377. arXiv:1812.11377 [cs, stat] type: article.

David M Young. *Iterative solution of large linear systems*. Elsevier, 2014.

Cun-Hui Zhang and Tong Zhang. A general theory of concave regularization for high-dimensional sparse estimation problems. *Statistical Science*, 27(4):576–593, 2012.

Tong Zhang. Learning Bounds for Kernel Regression Using Effective Data Dimensionality. *Neural Computation*, 17(9):2077–2098, September 2005. ISSN 0899-7667, 1530-888X. doi: 10.1162/0899766054323008. URL https://direct.mit.edu/neco/article/17/9/2077-2098/7007.

Appendix A. Algorithms

To start with showing algorithms for generic high-order smooth function, we first define $f_{\mathbf{x}}$ to be its second-order Taylor expansion (SOE) of f at $\mathbf{x}$ as follows:

$$f_{\mathbf{x}}(\mathbf{y}) = f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{1}{2}\langle \nabla^2 f(\mathbf{x})(\mathbf{y} - \mathbf{x}), \mathbf{y} - \mathbf{x} \rangle. \tag{20}$$

We first design a simple algorithm shown in Algorithm 3 which uses $\Theta(1)$ zeroth-order oracles to compute a δ-approximated $f_{\mathbf{x}}(\mathbf{y})$ shown below.

Algorithm 3 $\mathsf{ASOE}(f, L, H, \mathbf{x}, \mathbf{y}, \delta)$: Compute δ-approximated $f_{\mathbf{x}}(\mathbf{y})$

Input: an L-gradient Lipschitz continuous and H-Hessisan Lipschitz continuous function f, a zeroth-order oracle of f

Denote $r = \|\mathbf{y} - \mathbf{x}\|$;

Query $f(\mathbf{x})$, $f\left(\mathbf{x} + \frac{\delta}{Lr^2}(\mathbf{y} - \mathbf{x})\right)$, $f\left(\mathbf{x} + \frac{\delta}{2Hr^3}(\mathbf{y} - \mathbf{x})\right)$,$f\left(\mathbf{x} - \frac{\delta}{2Hr^3}(\mathbf{y} - \mathbf{x})\right)$;

Approximate $f_{\mathbf{x}}(\mathbf{y})$ by

$$\begin{aligned}\tilde{f}_{\mathbf{x},\delta}(\mathbf{y}) = f(\mathbf{x}) &+ \frac{Lr^2}{\delta}\left(f\left(\mathbf{x} + \frac{\delta}{Lr^2}(\mathbf{y} - \mathbf{x})\right) - f(\mathbf{x})\right) \\ &+ \frac{2H^2r^6}{\delta^2}\left(f\left(\mathbf{x} + \frac{\delta}{2Hr^3}(\mathbf{y} - \mathbf{x})\right) + f\left(\mathbf{x} - \frac{\delta^2}{2Hr^3}(\mathbf{y} - \mathbf{x})\right) - 2f(\mathbf{x})\right);\end{aligned} \tag{21}$$

return $\tilde{f}_{\mathbf{x},\delta}(\mathbf{y})$

Lemma 18 *For function f that has L-continuous gradient and H-continuous Hessian matrices, given any $\delta > 0$, Algorithm 3 outputs a δ-approximated $f_{\mathbf{x}}(\mathbf{y})$ denoted by $\tilde{f}_{\mathbf{x},\delta}(\mathbf{y})$ such that $\left|\tilde{f}_{\mathbf{x},\delta}(\mathbf{y}) - f_{\mathbf{x}}(\mathbf{y})\right| \leq \delta$.*

A.1. Algorithms for Convex Optimization

The proposed algorithm is shown in Algorithm 4, where each iterate consists of inexact solving the sub-problem by Algorithm 6 and an approximated computation of the gradient using zero-order oracles by Algorithm 5. Here, Algorithm 6 solves the subproblem by a binary search with each step solving a quadratic minimization problem using our accelerated algorithm presented in Algorithm 2.

A.2. Algorithms for Non-convex Optimization

An illustration of the algorithm is shown in Algorithm 7. To solve the subproblem, we use a binary search to determine $r_k \approx \|\mathbf{x}_{k+1} - \mathbf{x}_k\|$. With a given r_k, the subproblem can be transferred to a quadratic minimization problem and is solvable by Algorithm 2. The whole algorithm is shown in Algorithm 8, where the updates use Algorithm 9.

Algorithm 4 Inexact Large-step A-NPE with Zeroth-order Oracle

Input: $\sigma_l < \sigma_u < \sigma < 1$, $\sigma_l = \frac{\sigma_u}{2}$, $A_0 = 0$, $\epsilon_A < \frac{D}{N^{3/2}}$, $\epsilon_B < \frac{(\sigma-\sigma_u)^2}{2\lambda_{k+1}(L\lambda_{k+1}+1+(\sigma-\sigma_u)^2)\left(L+\frac{1}{\lambda_{k+1}}\right)} \cdot \left(f(\tilde{\mathbf{x}}_k) - \min_{\mathbf{y}} \left\{ f_{\tilde{\mathbf{x}}_k}(\mathbf{y}) + \frac{1}{2\lambda_{k+1}}\|\mathbf{y}-\tilde{\mathbf{x}}_k\|^2 \right\}\right)$, $k = 0$, $\lambda_0 = \frac{\sigma_l(1-\sigma^2)^{1/2}}{16DH}$;

while $k < N$ **do**

  $(\mathbf{y}_{k+1}, a_{k+1}, \lambda_{k+1}) \leftarrow \mathsf{ZHPEBinarySearch}(\tilde{\mathbf{x}}_k, H, \sigma_l, \sigma_u, A_k, \lambda_k, \epsilon_B)$;

  $\mathbf{v}_{k+1} \leftarrow \mathsf{ApproximateGradient}\left(f, \mathbf{y}_{k+1}, \frac{\epsilon_A}{a_{k+1}}\right)$;

  $A_{k+1} \leftarrow A_k + a_{k+1}$;

  $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k - a_{k+1}\mathbf{v}_{k+1}$;

  $k \leftarrow k+1$;

end

Algorithm 5 $\mathsf{ApproximateGradient}(f, \mathbf{x}, \epsilon_A)$: Approximating $\nabla f(\mathbf{x})$ with precision ϵ_A for f with L-Lipschitz gradient

$\rho \leftarrow \frac{2\epsilon_A}{dL}$;

for $i \in [d]$ **do**

  $\mathbf{v}_i \leftarrow \frac{f(\mathbf{x}+\rho\mathbf{e}_i)-f(\mathbf{x})}{\rho}$, where $\mathbf{e}_i$ is a vector whose ith coordinate is 1 and other coordinates are 0;

end

return $\mathbf{v}$

Algorithm 6 $\mathsf{ZHPEBinarySearch}(\tilde{\mathbf{x}}_k, H, \sigma_l, \sigma_u, A_k, \lambda_k, \epsilon_B)$: Binary search to find λ_k

$\lambda_{k+1} \leftarrow \lambda_k$;

while *True* **do**

  $a_{k+1} \leftarrow \frac{\lambda_{k+1}+\sqrt{\lambda_{k+1}^2+4\lambda_{k+1}A_k}}{2}$;

  $\tilde{\mathbf{x}}_k \leftarrow \frac{A_k}{A_k+a_{k+1}}\mathbf{y}_k + \frac{a_{k+1}}{A_k+a_{k+1}}\mathbf{x}_k$;

  Solve (22) with Algorithm 2 using Algorithm 3 as an oracle, and find an ϵ_B-approximated solution $\mathbf{y}_{k+1}$:

$$\min_{\mathbf{y}\in\mathbb{R}^d} f_{\tilde{\mathbf{x}}_k}(\mathbf{y}) + \frac{1}{2\lambda_{k+1}}\|\mathbf{y}-\tilde{\mathbf{x}}_k\|^2. \tag{22}$$

  Require: $\frac{2\sigma_l}{H} \le \lambda_{k+1}\|\mathbf{y}_{k+1}-\tilde{\mathbf{x}}_k\| \le \frac{2\sigma_u}{H}$;

  if $\lambda_{k+1}\|\mathbf{y}_{k+1}-\tilde{\mathbf{x}}_k\| \le \frac{2\sigma_l}{H}$ **then**

    $\lambda_{k+1} \leftarrow 2\lambda_{k+1}$;

  else if $\lambda_{k+1}\|\mathbf{y}_{k+1}-\tilde{\mathbf{x}}_k\| \ge \frac{2\sigma_u}{H}$ **then**

    $\lambda_{k+1} \leftarrow \frac{1}{2}\lambda_{k+1}$;

  else

    return $(\mathbf{y}_{k+1}, a_{k+1}, \lambda_{k+1})$;

  end

end

Algorithm 7 Illustration: Inexact Cubic Regularization Algorithm

while *stopping criterion is not met* **do**

Approximately solve the following optimization problem using Binary Search and Algorithm 2:

$$\mathbf{x}_{k+1} \leftarrow \operatorname*{argmin}_{\mathbf{y}} f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{y} - \mathbf{x}_k \rangle + \frac{1}{2}\langle \nabla^2 f(\mathbf{x}_k)(\mathbf{y} - \mathbf{x}_k), \mathbf{y} - \mathbf{x}_k \rangle + \frac{H}{6}\|\mathbf{y} - \mathbf{x}_k\|^3.$$

$k \leftarrow k + 1$;

end

Algorithm 8 Inexact Cubic Regularization Algorithm with Zeroth-order Oracle

Input: Desired accuracy ϵ;

while $r_k \geq \sqrt{\frac{\epsilon}{H}}$ **do**

$(\mathbf{x}_{k+1}, r_{k+1}) \leftarrow \mathsf{ZCubicBinarySearch}(\mathbf{x}_k, H, r_k)$;

$k \leftarrow k + 1$;

end

Appendix B. Proofs of Lemmas about Estimating Gradients

In this section, we present the proof of Lemma 8 and Lemma 9, and other technical lemmas about the properties of $\hat{\nabla}_\rho \tilde{f}_\delta$.

Lemma 8

$$\mathbb{E}_{\boldsymbol{\xi}} \tilde{\nabla} f(\mathbf{x}) = \nabla f(\mathbf{x}) \tag{24}$$

and

$$\mathbb{E}_{\boldsymbol{\xi}} \|\tilde{\nabla} f(\mathbf{x})\|^2 = \Theta(d\|\nabla f(\mathbf{x})\|^2). \tag{25}$$

Proof For the expectation, we have

$$\begin{aligned} \mathbb{E}_{\boldsymbol{\xi}} \tilde{\nabla} f(\mathbf{x}) &= \mathbb{E}_{\boldsymbol{\xi}} \langle \tilde{\nabla} f(\mathbf{x}), \boldsymbol{\xi} \rangle \cdot \boldsymbol{\xi} = \mathbb{E}_{\boldsymbol{\xi}} \boldsymbol{\xi}\boldsymbol{\xi}^\top \nabla f(\mathbf{x}) \\ &= \mathbf{I} \nabla f(\mathbf{x}) = \nabla f(\mathbf{x}). \end{aligned} \tag{26}$$

For the sum of second-order moment, for an arbitrary symmetric matrix $\mathbf{M}$, we have

$$\begin{aligned} \mathbb{E}_{\boldsymbol{\xi}} \|\tilde{\nabla} f(\mathbf{x})\|_{\mathbf{M}}^2 &= \mathbb{E}_{\boldsymbol{\xi}} \|\langle \nabla f(\mathbf{x}), \boldsymbol{\xi} \rangle \cdot \boldsymbol{\xi}\|_{\mathbf{M}}^2 \\ &= \mathbb{E}_{\boldsymbol{\xi}} \nabla f(\mathbf{x})^\top \boldsymbol{\xi}\boldsymbol{\xi}^\top \mathbf{M} \boldsymbol{\xi}\boldsymbol{\xi}^\top \nabla f(\mathbf{x}) \\ &= \nabla f(\mathbf{x})^\top \mathbb{E}_{\boldsymbol{\xi}} \left[\boldsymbol{\xi}\boldsymbol{\xi}^\top \mathbf{M} \boldsymbol{\xi}\boldsymbol{\xi}^\top\right] \nabla f(\mathbf{x}). \end{aligned} \tag{27}$$

Let $\mathbf{M} = \mathbf{U}^\top \mathbf{D} \mathbf{U}$ be the eigenvalue decomposition of $\mathbf{M}$ where $\mathbf{D} = \operatorname{diag}\{b_1, \cdots, b_d\}$ is a diagonal matrix, and $\boldsymbol{\zeta} = \mathbf{U}\boldsymbol{\xi}$ be a random variable. We have $\boldsymbol{\zeta} \sim N(0, \mathbf{I})$ because $\mathbf{U}$ is a orthogonal matrix, and

$$\begin{aligned} \mathbb{E}_{\boldsymbol{\xi}} \left[\boldsymbol{\xi}\boldsymbol{\xi}^\top \mathbf{M} \boldsymbol{\xi}\boldsymbol{\xi}^\top\right] &\overset{a}{=} \mathbb{E}_{\boldsymbol{\zeta}} \left[\mathbf{U}^\top \boldsymbol{\zeta}\boldsymbol{\zeta}^\top \mathbf{D} \boldsymbol{\zeta}\boldsymbol{\zeta}^\top \mathbf{U}\right] = \mathbf{U}^\top \mathbb{E}_{\boldsymbol{\zeta}} \left[\sum_{i=1}^d b_i \zeta_i^2 \cdot \boldsymbol{\zeta}\boldsymbol{\zeta}^\top\right] \mathbf{U} \\ &\overset{b}{=} \mathbf{U}^\top \left(\sum_{i=1}^d b_i \cdot \mathbf{I} + 2\mathbf{D}\right) \mathbf{U} \overset{c}{=} \operatorname{tr}(\mathbf{M}) \cdot \mathbf{I} + 2\mathbf{M}, \end{aligned} \tag{28}$$

Algorithm 9 ZCubicBinarySearch$(\mathbf{x}_k, H, r, \epsilon_C, \epsilon_D)$: Binary search to find r_k

$r_{k+1} \leftarrow r$;
$r_l \leftarrow 0, r_u \leftarrow \infty$;
while *True* **do**
  Solve (23) with Algorithm 2 using Algorithm 3 as an oracle, and find an ϵ_C-approximated solution $\mathbf{y}_{k+1}$:

$$\min_{\mathbf{y}\in\mathbb{R}^d} f_{\mathbf{x}_k}(\mathbf{y}) + \frac{r_{k+1}H}{2}\|\mathbf{y}-\mathbf{x}_k\|^2. \tag{23}$$

  if $\|\mathbf{y}_{k+1}-\mathbf{x}_k\| \le r_{k+1}$ **then**
    $r_u \leftarrow r_{k+1}$;
    $r_{k+1} \leftarrow \frac{r_{k+1}}{2}$;
  else if $\|\mathbf{y}_{k+1}-\mathbf{x}_k\| > r_{k+1}$ **then**
    $r_l \leftarrow r_{k+1}$;
    $r_{k+1} \leftarrow 2r_{k+1}$;
  end
  if *($r_l > 0$ and $r_u < \infty$) or $r_u < \epsilon_D$* **then**
    break;
  end
end
while $r_u - r_l \ge \epsilon_D$ **do**
  $r_{k+1} \leftarrow \frac{r_u+r_l}{2}$;
  Solve (23) with Algorithm 2 using Algorithm 3 as an oracle, and find an ϵ_C-approximated solution $\mathbf{y}_{k+1}$;
  if $\|\mathbf{y}_{k+1}-\mathbf{x}_k\| \le r_{k+1}$ **then**
    $r_u \leftarrow r_{k+1}$;
  else if $\|\mathbf{y}_{k+1}-\mathbf{x}_k\| > r_{k+1}$ **then**
    $r_l \leftarrow r_{k+1}$;
  end
end
$r_{k+1} \leftarrow r_u$;
Solve (23) with Algorithm 2 using Algorithm 3 as an oracle, and find an ϵ_C-approximated solution $\mathbf{y}_{k+1}$;
return $(\mathbf{y}_{k+1}, r_{k+1})$;

where in $\overset{a}{=}$, we introduce $\boldsymbol{\zeta} = \mathbf{U}\boldsymbol{\xi}$, then $\boldsymbol{\zeta}$ also follows from standard Gaussian distribution by the rotational invariance, in $\overset{b}{=}$, we use the second and fourth order moment of standard Gaussian variables: $\mathbb{E}\boldsymbol{\zeta}_i^2 = 1$, $\mathbb{E}\boldsymbol{\zeta}_i^4 = 3$, and in $\overset{c}{=}$, we use $\mathrm{tr}(\mathbf{M}) = \mathrm{tr}(\mathbf{U}^\top\mathbf{D}\mathbf{U}) = \mathrm{tr}(\mathbf{D}\mathbf{U}\mathbf{U}^\top) = \mathrm{tr}(\mathbf{D})$.

When $\mathbf{M} = \mathbf{I}$, we have

$$\mathbb{E}_{\boldsymbol{\xi}}\|\tilde{\nabla} f(\mathbf{x})\|^2 = (d+2)\|\nabla f(\mathbf{x})\|^2 = \Theta(d\|\nabla f(\mathbf{x})\|^2). \tag{29}$$

■

Lemma 9 *For symmetric matrix* $\mathbf{M}$*,*

$$\mathbb{E}_{\boldsymbol{\xi}}\|\tilde{\nabla} f(\mathbf{x})\|_{\mathbf{M}}^2 \le 3\mathrm{tr}(\mathbf{M})\|\nabla f(\mathbf{x})\|^2. \tag{30}$$

Proof By (28), (27), and the fact that $\mathrm{tr}(\mathbf{M}) \cdot \mathbf{I} \succeq L\mathbf{I} \succeq \mathbf{M}$ we have (30). ■

Lemma 19 *Let $\tilde{f}_\delta$ be a δ-approximated estimate of f. If $\hat{\nabla}_\rho \tilde{f}_\delta(\mathbf{x})$ and $\tilde{\nabla} f(\mathbf{x})$ are generated by the same Gaussian random variable,*

$$\mathbb{E}_{k+1}\|\hat{\nabla}_\rho \tilde{f}_\delta(\mathbf{x}) - \tilde{\nabla} f(\mathbf{x})\|_{\mathbf{B}}^2 \leq \frac{8\delta^2}{\rho^2}\mathrm{tr}(\mathbf{B}) + \frac{15\rho^2}{2}\mathrm{tr}(\mathbf{A})^2\mathrm{tr}(\mathbf{B}), \tag{31}$$

where $\mathbf{B}$ is an arbitrary positive semi-definite symmetric matrix.

Proof By the definition of $\hat{\nabla}$ in (6), $\tilde{\nabla}$ in (7), and $\tilde{f}_\delta$, we have

$$\begin{aligned}
&\hat{\nabla}_\rho \tilde{f}_\delta(\mathbf{x}) - \tilde{\nabla} f(\mathbf{x}) \qquad (32)\\
&= \left(\frac{\tilde{f}_\delta(\mathbf{x}+\rho\boldsymbol{\xi}) - \tilde{f}_\delta(\mathbf{x})}{\rho} - \langle \nabla f(\mathbf{x}), \boldsymbol{\xi}\rangle\right)\cdot\boldsymbol{\xi}\\
&= \left(\left[\frac{\tilde{f}_\delta(\mathbf{x}+\rho\boldsymbol{\xi}) - \tilde{f}_\delta(\mathbf{x})}{\rho} - \frac{f(\mathbf{x}+\rho\boldsymbol{\xi}) - f(\mathbf{x})}{\rho}\right] + \left[\frac{f(\mathbf{x}+\rho\boldsymbol{\xi}) - f(\mathbf{x})}{\rho} - \langle \nabla f(\mathbf{x}), \boldsymbol{\xi}\rangle\right]\right)\cdot\boldsymbol{\xi}.
\end{aligned}$$

By (32), we have

$$\begin{aligned}
&\mathbb{E}_{k+1}\left\|\hat{\nabla}_\rho \tilde{f}_\delta(\mathbf{x}) - \tilde{\nabla} f(\mathbf{x})\right\|_{\mathbf{B}}^2\\
&\leq 2\mathbb{E}_{k+1}\left(\frac{\tilde{f}_\delta(\mathbf{x}+\rho\boldsymbol{\xi}) - \tilde{f}_\delta(\mathbf{x})}{\rho} - \frac{f(\mathbf{x}+\rho\boldsymbol{\xi}) - f(\mathbf{x})}{\rho}\right)^2\cdot\|\boldsymbol{\xi}\|_{\mathbf{B}}^2\\
&\quad + 2\mathbb{E}_{k+1}\left(\frac{f(\mathbf{x}+\rho\boldsymbol{\xi}) - f(\mathbf{x})}{\rho} - \langle \nabla f(\mathbf{x}), \boldsymbol{\xi}\rangle\right)^2\cdot\|\boldsymbol{\xi}\|_{\mathbf{B}}^2 \qquad (33)\\
&\leq \frac{8\delta^2}{\rho^2}\mathbb{E}_{k+1}\|\boldsymbol{\xi}\|_{\mathbf{B}}^2 + \frac{1}{2}\mathbb{E}_{k+1}\rho^2\|\boldsymbol{\xi}\|_{\mathbf{A}}^4\|\boldsymbol{\xi}\|_{\mathbf{B}}^2\\
&\leq \frac{8\delta^2}{\rho^2}\mathrm{tr}(\mathbf{B}) + \frac{15\rho^2}{2}\mathrm{tr}(\mathbf{A})^2\mathrm{tr}(\mathbf{B}).
\end{aligned}$$

■

Appendix C. Proofs of Theorems 10 and 12

Theorem 10 *Suppose f is a μ-strongly convex quadratic function and has L-Lipschitz continuous gradient. The Hessian matrix of f is $\mathbf{A}$. Let $h_k = \frac{1}{12\mathrm{tr}(\mathbf{A})}$. Using an δ-approximated zeroth-order oracle, $\{\mathbf{x}_k\}_{k\in\mathbb{N}}$ generated by $\mathcal{RG}_\rho$ satisfies*

$$\begin{aligned}
&\mathbb{E}f(\mathbf{x}_{k+1}) - f^* - \frac{24\mathrm{tr}(\mathbf{A})}{\mu}\left(C_1\rho^2 + C_2\frac{\delta^2}{\rho^2}\right)\\
&\leq \left(1 - \frac{\mu}{24\mathrm{tr}(\mathbf{A})}\right)\left(\mathbb{E}f(\mathbf{x}_k) - f^* - \frac{24\mathrm{tr}(\mathbf{A})}{\mu}\left(C_1\rho^2 + C_2\frac{\delta^2}{\rho^2}\right)\right),
\end{aligned} \tag{34}$$

where

$$C_1 = \frac{5}{16}\mathrm{tr}(\mathbf{A})d + \frac{5}{384}\mathrm{tr}(\mathbf{A}), \quad C_2 = \frac{d}{3\mathrm{tr}(\mathbf{A})} + \frac{1}{72\mathrm{tr}(\mathbf{A})}, \tag{35}$$

and the expectation is taken for all the randomness in the algorithm.

Proof Because f is quadratic, we have

$$f(\mathbf{x}_{k+1}) = f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{x}_{k+1} - \mathbf{x}_k \rangle + \frac{h_k^2}{2}\|\hat{\nabla}_\rho \tilde{f}_\delta(\mathbf{x}_k)\|_{\mathbf{A}}^2. \tag{36}$$

Taking expectation with respect to the randomization of $\hat{\nabla}_\rho \tilde{f}_\delta(\mathbf{x}_k)$ on both sides of (36), we get

$$\begin{aligned}
&\mathbb{E}_{k+1} f(\mathbf{x}_{k+1}) \\
\leq & f(\mathbf{x}_k) - h_k \langle \nabla f(\mathbf{x}_k), \mathbb{E}_{k+1}\hat{\nabla}_\rho \tilde{f}_\delta(\mathbf{x}_k)\rangle + \frac{h_k^2}{2}\mathbb{E}_{k+1}\|\hat{\nabla}_\rho \tilde{f}_\delta(\mathbf{x}_k)\|_{\mathbf{A}}^2 \\
\leq & f(\mathbf{x}_k) - h_k \langle \nabla f(\mathbf{x}_k), \mathbb{E}_{k+1}\tilde{\nabla} f(\mathbf{x}_k)\rangle - h_k \left\langle \nabla f(\mathbf{x}_k), \mathbb{E}_{k+1}\left[\hat{\nabla}_\rho \tilde{f}_\delta(\mathbf{x}_k) - \tilde{\nabla} f(\mathbf{x}_k)\right]\right\rangle \\
& + h_k^2 \mathbb{E}_{k+1}\|\tilde{\nabla} f(\mathbf{x}_k)\|_{\mathbf{A}}^2 + h_k^2 \mathbb{E}_{k+1}\|\hat{\nabla}_\rho \tilde{f}_\delta(\mathbf{x}_k) - \tilde{\nabla} f(\mathbf{x}_k)\|_{\mathbf{A}}^2.
\end{aligned} \tag{37}$$

Using Lemma 8 and Lemma 9 in (37), we have

$$\begin{aligned}
&\mathbb{E}_{k+1} f(\mathbf{x}_{k+1}) \\
\leq & f(\mathbf{x}_k) - h_k\|\nabla f(\mathbf{x}_k)\|^2 + h_k^2 \cdot 3\mathrm{tr}(\mathbf{A})\mathbb{E}_{k+1}\|\nabla f(\mathbf{x}_k)\|^2 \\
& - h_k \left\langle \nabla f(\mathbf{x}_k), \mathbb{E}_{k+1}\left[\hat{\nabla}_\rho \tilde{f}_\delta(\mathbf{x}_k) - \tilde{\nabla} f(\mathbf{x}_k)\right]\right\rangle + h_k^2 \mathbb{E}_{k+1}\|\hat{\nabla}_\rho \tilde{f}_\delta(\mathbf{x}_k) - \tilde{\nabla} f(\mathbf{x}_k)\|_{\mathbf{A}}^2 \\
\leq & f(\mathbf{x}_k) - h_k\|\nabla f(\mathbf{x}_k)\|^2 + h_k^2 \cdot 3\mathrm{tr}(\mathbf{A})\mathbb{E}_{k+1}\|\nabla f(\mathbf{x}_k)\|^2 \\
& + \frac{h_k}{2}\|\nabla f(\mathbf{x}_k)\|^2 + \frac{h_k}{2}\mathbb{E}_{k+1}\|\hat{\nabla}_\rho \tilde{f}_\delta(\mathbf{x}_k) - \tilde{\nabla} f(\mathbf{x}_k)\|^2 \\
& + h_k^2 \mathbb{E}_{k+1}\|\hat{\nabla}_\rho \tilde{f}_\delta(\mathbf{x}_k) - \tilde{\nabla} f(\mathbf{x}_k)\|_{\mathbf{A}}^2.
\end{aligned}$$

By Lemma 19, we have

$$\begin{aligned}
\mathbb{E}_{k+1} f(\mathbf{x}_{k+1}) \leq & f(\mathbf{x}_k) - \frac{h_k}{2}\|\nabla f(\mathbf{x}_k)\|^2 + 3h_k^2 \mathrm{tr}(\mathbf{A})\mathbb{E}_{k+1}\|\nabla f(\mathbf{x}_k)\|^2 \\
& + \frac{h_k}{2} \cdot \left(\frac{8\delta^2}{\rho^2}d + \frac{15\rho^2}{2}\mathrm{tr}(\mathbf{A})^2 d\right) + h_k^2 \cdot \left(\frac{8\delta^2}{\rho^2}\mathrm{tr}(\mathbf{A}) + \frac{15\rho^2}{2}\mathrm{tr}(\mathbf{A})^3\right).
\end{aligned} \tag{38}$$

By the definition of h_k, we have

$$\begin{aligned}
\mathbb{E}_{k+1} f(\mathbf{x}_{k+1}) \leq & f(\mathbf{x}_k) - \frac{1}{48\mathrm{tr}(\mathbf{A})}\|\nabla f(\mathbf{x}_k)\|^2 \\
& + \frac{1}{24}\left(\frac{8\sigma^2 d}{\rho^2 \mathrm{tr}(\mathbf{A})} + \frac{15\rho^2 \mathrm{tr}(\mathbf{A})d}{2}\right) + \frac{1}{576}\left(\frac{8\delta^2}{\rho^2\mathrm{tr}(\mathbf{A})} + \frac{15\rho^2\mathrm{tr}(\mathbf{A})}{2}\right) \\
= & f(\mathbf{x}_k) - \frac{1}{48\mathrm{tr}(\mathbf{A})}\|\nabla f(\mathbf{x}_k)\|^2 \\
& + \rho^2\left(\frac{5}{16}\mathrm{tr}(\mathbf{A})d + \frac{5}{384}\mathrm{tr}(\mathbf{A})\right) + \frac{\sigma^2}{\rho^2}\left(\frac{d}{3\mathrm{tr}(\mathbf{A})} + \frac{1}{72\mathrm{tr}(\mathbf{A})}\right).
\end{aligned} \tag{39}$$

By (39), we have

$$\begin{aligned}\mathbb{E}_{k+1}f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) - \frac{1}{48\mathrm{tr}(\mathbf{A})}\|\nabla f(\mathbf{x}_k)\|^2 \\ + C_1\rho^2 + C_2\frac{\delta^2}{\rho^2},\end{aligned} \tag{40}$$

where C_1 and C_2 are constants depending only on $\mathrm{tr}(\mathbf{A})$ and d, defined in (35). By the strong convexity of f, we have

$$f(\mathbf{x}_k) \leq f^* + \langle \nabla f(\mathbf{x}_k), \mathbf{x}_k - \mathbf{x}^* \rangle - \frac{\mu}{2}\|\mathbf{x}_k - \mathbf{x}^*\|^2. \tag{41}$$

By Cauchy-Schwartz inequality, we have

$$\langle \nabla f(\mathbf{x}_k), \mathbf{x}_k - \mathbf{x}^* \rangle \leq \frac{1}{2\mu}\|\nabla f(\mathbf{x}_k)\|^2 + \frac{\mu}{2}\|\mathbf{x}_k - \mathbf{x}^*\|^2. \tag{42}$$

Therefore,

$$f(\mathbf{x}_k) \leq f^* + \frac{1}{2\mu}\|\nabla f(\mathbf{x}_k)\|^2. \tag{43}$$

Plugging (40) into (43), we have

$$\begin{aligned}\mathbb{E}_{k+1}f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) - \frac{\mu}{24\mathrm{tr}(\mathbf{A})}\left(f(\mathbf{x}_k) - f^*\right) \\ + C_1\rho^2 + C_2\frac{\delta^2}{\rho^2}.\end{aligned} \tag{44}$$

Taking full expectation to (44), we have

$$\mathbb{E}f(\mathbf{x}_{k+1}) - f^* \leq \left(1 - \frac{\mu}{24\mathrm{tr}(\mathbf{A})}\right)\left(\mathbb{E}f(\mathbf{x}_k) - f^*\right) + C_1\rho^2 + C_2\frac{\delta^2}{\rho^2}. \tag{45}$$

By (45), we have

$$\begin{aligned}&\mathbb{E}f(\mathbf{x}_{k+1}) - f^* - \frac{24\mathrm{tr}(\mathbf{A})}{\mu}\left(C_1\rho^2 + C_2\frac{\delta^2}{\rho^2}\right) \\ \leq &\left(1 - \frac{\mu}{24\mathrm{tr}(\mathbf{A})}\right)\left(\mathbb{E}f(\mathbf{x}_k) - f^* - \frac{24\mathrm{tr}(\mathbf{A})}{\mu}\left(C_1\rho^2 + C_2\frac{\delta^2}{\rho^2}\right)\right).\end{aligned} \tag{46}$$

■

Theorem 12 *Suppose f is a L-smooth quadratic function whose Hessian matrix is $\mathbf{A}$. Using a δ-approximated zeroth-order oracle, $\{\mathbf{x}_k\}_{k\in\mathbb{N}}$ generated by $\mathcal{RG}_\rho$ satisfies*

$$\begin{aligned}&(k+1)(\mathbb{E}f(\mathbf{x}_{k+1}) - f^*) \\ \leq\ &k\mathbb{E}(f(\mathbf{x}_k) - f^*) + (k+1)\left(C_1\rho^2 + C_2\frac{\delta^2}{\rho^2}\right) + \frac{12\mathrm{tr}(\mathbf{A})}{k+1}\mathbb{E}\|\mathbf{x}_k - \mathbf{x}^*\|^2,\end{aligned} \tag{47}$$

where the expectation is taken for all the randomness in the algorithm.

Proof By (39), we have

$$\begin{aligned}\mathbb{E}_{k+1}f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) - \frac{1}{48\mathrm{tr}(\mathbf{A})}\|\nabla f(\mathbf{x}_k)\|^2 \\ + C_1\rho^2 + C_2\frac{\delta^2}{\rho^2},\end{aligned} \tag{48}$$

where C_1 and C_2 are constants depending only on $\|\mathbf{A}\|_*$ and d, defined in (35). By the convexity of f, we have

$$f(\mathbf{x}_k) \leq f^* + \langle \nabla f(\mathbf{x}_k), \mathbf{x}_k - \mathbf{x}^* \rangle. \tag{49}$$

By Cauchy-Schwartz inequality, we have

$$\langle \nabla f(\mathbf{x}_k), \mathbf{x}_k - \mathbf{x}^* \rangle \leq \frac{1}{48\mathrm{tr}(\mathbf{A})(k+1)}\|\nabla f(\mathbf{x}_k)\|^2 + 12\mathrm{tr}(\mathbf{A})(k+1)\|\mathbf{x}_k - \mathbf{x}^*\|^2. \tag{50}$$

Using (49) and (50), we have

$$\begin{aligned}f(\mathbf{x}_k) &\leq f^* + \frac{k+1}{48\mathrm{tr}(\mathbf{A})}\|\nabla f(\mathbf{x}_k)\|^2 + \frac{12\mathrm{tr}(\mathbf{A})}{k+1}\|\mathbf{x}_k - \mathbf{x}^*\|^2 \\ &\overset{(48)}{\leq} f^* + (k+1)\left(f(\mathbf{x}_k) - \mathbb{E}_{k+1}f(\mathbf{x}_{k+1}) + C_1\rho^2 + C_2\frac{\delta^2}{\rho^2}\right) \\ &\quad + \frac{12\mathrm{tr}(\mathbf{A})}{k+1}\|\mathbf{x}_k - \mathbf{x}^*\|^2.\end{aligned} \tag{51}$$

Therefore, we have

$$\begin{aligned}&(k+1)(\mathbb{E}_{k+1}f(\mathbf{x}_{k+1}) - f^*) \\ &\leq k(f(\mathbf{x}_k) - f^*) + (k+1)\left(C_1\rho^2 + C_2\frac{\delta^2}{\rho^2}\right) + \frac{12\mathrm{tr}(\mathbf{A})}{k+1}\|\mathbf{x}_k - \mathbf{x}^*\|^2.\end{aligned} \tag{52}$$

■

Appendix D. Proof of Theorem 14

Theorem 14 *Suppose f is a L-smooth μ-strongly convex quadratic function whose Hessian matrix is $\mathbf{A}$. Using an δ-approximated zeroth-order oracle, if δ and ρ is small enough such that*

$$6n\left(\frac{16\delta}{\rho} + 12\rho\mathrm{tr}(\mathbf{A})\right) < 80\left(1 - \frac{\mu^{1/2}}{57600\mathrm{ED}_{1/2}}\right)^{n-1} \cdot \mu \cdot (f(\mathbf{x}_0) - f^*), \tag{53}$$

$\{\mathbf{x}_n\}_{n\in\mathbb{N}}$ generated by ZHB *satisfies*

$$\mathbb{E}f(\mathbf{y}_n) - f^* \leq 400\left(1 - \frac{\mu^{1/2}}{57600\mathrm{ED}_{\frac{1}{2}}}\right)^n \cdot \frac{L}{\mu} \cdot (f(\mathbf{x}_0) - f^*), \tag{54}$$

where the expectation is taken for all the randomness in the algorithm.

Proof Let $\mathbf{z}_{k+1} = \begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{x}_k \end{bmatrix}$. The iterations of ZHB can be written as

$$\mathbf{z}_{k+1} = \begin{bmatrix} (2-\beta)(\mathbf{I}-h\mathbf{A}) & -(1-\beta)(\mathbf{I}-h\mathbf{A}) \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \mathbf{z}_k + h\boldsymbol{\epsilon}_k^1 + h\boldsymbol{\epsilon}_k^2 \triangleq \mathbf{B}\mathbf{z}_k + h\boldsymbol{\epsilon}_k^1 + h\boldsymbol{\epsilon}_k^2, \quad (55)$$

where $\boldsymbol{\epsilon}_k^1 = \begin{bmatrix} (\mathbf{I}-\boldsymbol{\xi}\boldsymbol{\xi}^\top)\mathbf{A}\mathbf{y}_k \\ \mathbf{0} \end{bmatrix}$, and $\boldsymbol{\epsilon}_k^2 = \begin{bmatrix} \tilde{\nabla} f(\mathbf{x}_k) - \hat{\nabla}_\rho \tilde{f}_\delta(\mathbf{x}_k) \\ \mathbf{0} \end{bmatrix}$. $\boldsymbol{\epsilon}_k^1$ represents the error of estimating $\nabla f(\mathbf{x}_k)$ with $\tilde{\nabla}(\mathbf{x}_k)$, and $\boldsymbol{\epsilon}_k^2$ represents the error of estimating ZHB with $\hat{\nabla}_\rho \tilde{f}_\delta(\mathbf{x}_k)$.

By induction on k, we have

$$\mathbf{z}_n = \mathbf{B}^n \mathbf{z}_0 + h\sum_{k=0}^{n-1} \mathbf{B}^{n-k-1}\boldsymbol{\epsilon}_k^1 + h\sum_{k=0}^{n-1} \mathbf{B}^{n-k-1}\boldsymbol{\epsilon}_k^2. \quad (56)$$

Without loss of generality, we assume that $\mathbf{x}^* = \mathbf{0}$. We estimate the distance to the optimal solution by the $\mathbf{A}^2$ norm of $\mathbf{x}_k$. To compute $\|\mathbf{x}_k\|_{\mathbf{A}^2}$, we decompose $\mathbf{x}_k$ into eigen-directions of $\mathbf{A}$, and $\mathbf{B}$ can be decomposed into 2×2 matrices. For an eigen-direction with eigenvalue λ, the update of AGD can be written as follows:

$$\begin{aligned} \begin{bmatrix} x_{k+1} \\ x_k \end{bmatrix} &= \begin{bmatrix} (2-\beta)(1-h\lambda) & -(1-\beta)(1-h\lambda) \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ x_{k-1} \end{bmatrix} + h\begin{bmatrix} \epsilon \\ 0 \end{bmatrix} \\ &\triangleq \mathbf{B}_\lambda \begin{bmatrix} x_k \\ x_{k-1} \end{bmatrix} + h\begin{bmatrix} \epsilon \\ 0 \end{bmatrix}. \end{aligned} \quad (57)$$

Let μ_1 and μ_2 be the eigenvalues of $\mathbf{B}_\lambda$. By the Lemma 19 of Jin et al. (2017), we can write the eigen-decomposition of $\mathbf{B}_\lambda$ as

$$\mathbf{B}_\lambda = \frac{1}{\mu_1-\mu_2}\begin{bmatrix} \mu_1 & \mu_2 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{bmatrix}\begin{bmatrix} 1 & -\mu_2 \\ -1 & \mu_1 \end{bmatrix}. \quad (58)$$

Let $\mathbf{C} = \begin{bmatrix} \mathbf{A}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^2 \end{bmatrix}$. By (56), We have

$$\mathbb{E}\|\mathbf{z}_n\|_{\mathbf{C}}^2 \le 3\|\mathbf{B}^n\mathbf{z}_0\|_{\mathbf{C}}^2 + 3\mathbb{E}\left\|\sum_{k=0}^{n-1}\mathbf{B}^{n-k-1}\boldsymbol{\epsilon}_k^1\right\|_{\mathbf{C}}^2 + 3\mathbb{E}\left\|\sum_{k=0}^{n-1}\mathbf{B}^{n-k-1}\boldsymbol{\epsilon}_k^2\right\|_{\mathbf{C}}^2 \quad (59)$$

First we tackle the $\boldsymbol{\epsilon}_k^1$ terms.

$$\begin{aligned} &\mathbb{E}\left\|\sum_{k=0}^{n-1}\mathbf{B}^{n-k-1}\boldsymbol{\epsilon}_k^1\right\|_{\mathbf{C}}^2 \\ &= \sum_{k=0}^{n-1}\mathbb{E}_k\left\|\mathbf{B}^{n-k-1}\boldsymbol{\epsilon}_k^1\right\|_{\mathbf{C}}^2 \\ &= \sum_{k=0}^{n-1}\mathbb{E}_i\left[\mathbf{y}_k^\top\mathbf{A}^\top(\mathbf{I}-\boldsymbol{\xi}_k\boldsymbol{\xi}_k^\top) \quad \mathbf{0}\right]\mathbf{B}^{(n-k-1)T}\mathbf{C}\mathbf{B}^{n-k-1}\begin{bmatrix}(\mathbf{I}-\boldsymbol{\xi}_k\boldsymbol{\xi}_k^\top)\mathbf{A}\mathbf{y}_k \\ \mathbf{0}\end{bmatrix} \\ &\overset{\text{Lemma 9}}{\le} 3\sum_{k=1}^{n}\mathrm{tr}\left(\mathbf{B}^{(n-k-1)T}\mathbf{C}\mathbf{B}^{n-k-1}\right)\cdot\|\mathbf{y}_k\|_{\mathbf{A}^2}^2. \end{aligned} \quad (60)$$

In order to estimate $\mathrm{tr}\left(\mathbf{B}^{(n-k-1)T}\mathbf{C}\mathbf{B}^{n-k-1}\right)$, we consider blocks of $\mathbf{B}$ with respect to eigen-directions of $\mathbf{A}$. The contribution of an eigen-direction with eigenvalue λ in the trace is

$$\begin{aligned}&\mathrm{tr}\left(\mathbf{B}_\lambda^{(n-k-1)\top}\cdot\begin{bmatrix}\lambda^2 & 0\\ 0 & \lambda^2\end{bmatrix}\mathbf{B}^{(n-k-1)}\right)\\ &=\lambda^2\left(\left\|\begin{bmatrix}1 & 0\end{bmatrix}\mathbf{B}_\lambda^{n-k-1}\right\|^2+\left\|\begin{bmatrix}0 & 1\end{bmatrix}\mathbf{B}_\lambda^{n-k-1}\right\|^2\right)\end{aligned}\tag{61}$$

By Lemma 19 of Jin et al. (2017), the last line in (61) equals to

$$\begin{aligned}&\lambda^2\left\|\begin{bmatrix}\sum_{i=0}^{n-k-1}\mu_{\lambda,1}^i\mu_{\lambda,2}^{n-k-1-i} & -\mu_{\lambda,1}\mu_{\lambda,2}\sum_{i=0}^{n-k-2}\mu_{\lambda,1}^i\mu_{\lambda,2}^{n-k-2-i}\end{bmatrix}\right\|^2\\ &+\lambda^2\left\|\begin{bmatrix}\sum_{i=0}^{n-k-2}\mu_{\lambda,1}^i\mu_{\lambda,2}^{n-k-2-i} & -\mu_{\lambda,1}\mu_{\lambda,2}\sum_{i=0}^{n-k-3}\mu_{\lambda,1}^i\mu_{\lambda,2}^{n-k-3-i}\end{bmatrix}\right\|^2.\end{aligned}\tag{62}$$

Define $a_\lambda=|\mu_{\lambda,1}|=\sqrt{(1-\beta)(1-h\lambda)}$. By the choice of β, we have $a_\lambda\le 1-\frac{\sqrt{h\mu}}{2}$. We have the following equation:

$$\lambda^2\left\|\begin{bmatrix}\sum_{i=0}^{n-k}\mu_{\lambda,1}^i\mu_{\lambda,2}^{n-k-i} & -\mu_{\lambda,1}\mu_{\lambda,2}\sum_{i=0}^{n-k-1}\mu_{\lambda,1}^i\mu_{\lambda,2}^{n-k-1-i}\end{bmatrix}\right\|^2\le 4\lambda^2(n-k)^2a_\lambda^{n-k}.\tag{63}$$

From the definition of $\mathbf{y}_i$ and Cauchy-Schwartz inequality, we have

$$\|\mathbf{y}_i\|_{\mathbf{A}^2}^2\le 8\|\mathbf{x}_i\|_{\mathbf{A}^2}^2+2\|\mathbf{x}_{i-1}\|_{\mathbf{A}^2}^2\le 8\|\mathbf{z}_i\|_{\mathbf{C}}^2+2\|\mathbf{z}_{i-1}\|_{\mathbf{C}}^2.\tag{64}$$

Therefore,

$$\begin{aligned}&\mathbb{E}\left\|\sum_{k=0}^{n-1}\mathbf{B}^{k-i}\boldsymbol{\epsilon}_i^1\right\|_{\mathbf{C}}^2\\ &\le 3\sum_{k=0}^{n-1}\sum_{i=1}^{d}8\lambda_i^2(n-k)^2a_{\lambda_i}^{n-k}\cdot\|\mathbf{y}_k\|_{\mathbf{A}^2}^2\\ &=24\sum_{i=1}^{d}\sum_{k=0}^{n-1}\lambda_i^2(n-k)^2a_{\lambda_i}^{n-k}\cdot\|\mathbf{y}_k\|_{\mathbf{A}^2}^2\end{aligned}\tag{65}$$

Then we calculate $\|\mathbf{B}^n\mathbf{z}_0\|_{\mathbf{C}}^2$. As $\mathbf{x}_{-1}=\mathbf{x}_0$, the contribution of an eigen-directions of $\mathbf{A}$ to the norm is

$$\lambda^2x_\lambda^2\left\|\mathbf{B}_\lambda^n\begin{bmatrix}1\\1\end{bmatrix}\right\|^2,\tag{66}$$

where λ is the eigenvalue, and x_λ is the coefficient of the eigen-decomposition of $\mathbf{x}_0$. By Lemma 19 of Jin et al. (2017), we have

$$\begin{aligned}
\mathbf{B}_\lambda^n \begin{bmatrix}1\\1\end{bmatrix} &= \begin{bmatrix} \sum_{i=0}^{n} \mu_{\lambda,1}^i \mu_{\lambda,2}^{n-i} - \mu_{\lambda,1}\mu_{\lambda,2} \sum_{i=0}^{n-1} \mu_{\lambda,1}^i \mu_{\lambda,2}^{n-1-i} \\ \sum_{i=0}^{n-1} \mu_{\lambda,1}^i \mu_{\lambda,2}^{n-1-i} - \mu_{\lambda,1}\mu_{\lambda,2} \sum_{i=0}^{n-2} \mu_{\lambda,1}^i \mu_{\lambda,2}^{n-2-i} \end{bmatrix} \\
&= \frac{1}{2} \begin{bmatrix} \mu_{\lambda,1}^n + \mu_{\lambda,2}^n + (2-\mu_{\lambda,1}-\mu_{\lambda,2}) \sum_{i=0}^{n} \mu_{\lambda,1}^i \mu_{\lambda,2}^{n-i} \\ \mu_{\lambda,1}^{n-1} + \mu_{\lambda,2}^{n-1} + (2-\mu_{\lambda,1}-\mu_{\lambda,2}) \sum_{i=0}^{n-1} \mu_{\lambda,1}^i \mu_{\lambda,2}^{n-1-i} \end{bmatrix} \\
&= \frac{1}{2} \begin{bmatrix} \mu_{\lambda,1}^n + \mu_{\lambda,2}^n + (2-\mu_{\lambda,1}-\mu_{\lambda,2}) \frac{\mu_{\lambda,1}^{n+1} - \mu_{\lambda,2}^{n+1}}{\mu_{\lambda,1}-\mu_{\lambda,2}} \\ \\ \mu_{\lambda,1}^{n-1} + \mu_{\lambda,2}^{n-1} + (2-\mu_{\lambda,1}-\mu_{\lambda,2}) \frac{\mu_{\lambda,1}^{n} - \mu_{\lambda,2}^{n}}{\mu_{\lambda,1}-\mu_{\lambda,2}} \end{bmatrix}
\end{aligned} \tag{67}$$

The $\frac{2-\mu_{\lambda,1}-\mu_{\lambda,2}}{\mu_{\lambda,1}-\mu_{\lambda,2}}$ term in (67) can be bounded as follows:

$$\begin{aligned}
\frac{2-\mu_{\lambda,1}-\mu_{\lambda,2}}{\mu_{\lambda,1}-\mu_{\lambda,2}} &= \frac{2-(2-\beta)(1-h\lambda)}{\sqrt{(1-h\lambda)(h\lambda(2-\beta)^2-\beta^2)}} \\
&\le \frac{\beta+h\lambda}{\sqrt{\frac{1}{4}\cdot h\lambda}} \\
&\le 2+\sqrt{h\lambda} \\
&\le 3.
\end{aligned} \tag{68}$$

Therefore,

$$\begin{aligned}
&\left\| \mathbf{B}_\lambda^n \begin{bmatrix}1\\1\end{bmatrix} \right\|^2 \\
&\le \frac{1}{4} \cdot 4 \left(|\mu_{\lambda,1}^{2n}| + |\mu_{\lambda,2}^{2n}| + 9|\mu_{\lambda,1}^{2n+2}| + 9|\mu_{\lambda,2}^{2n+2}| + |\mu_{\lambda,1}^{2n-2}| + |\mu_{\lambda,2}^{2n-2}| + 9|\mu_{\lambda,1}^{2n}| + 9|\mu_{\lambda,2}^{2n}| \right) \\
&\le 40 \left(1 - \frac{\sqrt{h\mu}}{2}\right)^{2n-2},
\end{aligned} \tag{69}$$

and we have

$$\|\mathbf{B}^n \mathbf{z}_0\|_{\mathbf{C}}^2 \le 40 \left(1 - \frac{\sqrt{h\mu}}{2}\right)^{2n-2} \|\mathbf{z}_0\|_{\mathbf{C}}^2. \tag{70}$$

Finally we tackle the $\boldsymbol{\epsilon}_k^2$ terms. We have

$$\begin{aligned}
&\hat{\nabla}_\rho \tilde{f}_\delta(\mathbf{x}) - \tilde{\nabla} f(\mathbf{x}) \\
&= \left(\frac{\tilde{f}_\delta(\mathbf{x}+\rho\boldsymbol{\xi}) - \tilde{f}_\delta(\mathbf{x})}{\rho} - \langle \nabla f(\mathbf{x}), \boldsymbol{\xi} \rangle \right) \cdot \boldsymbol{\xi} \\
&= \left(\left[\frac{\tilde{f}_\delta(\mathbf{x}+\rho\boldsymbol{\xi}) - \tilde{f}_\delta(\mathbf{x})}{\rho} - \frac{f(\mathbf{x}+\rho\boldsymbol{\xi}) - f(\mathbf{x})}{\rho} \right] + \left[\frac{f(\mathbf{x}+\rho\boldsymbol{\xi}) - f(\mathbf{x})}{\rho} - \langle \nabla f(\mathbf{x}), \boldsymbol{\xi} \rangle \right] \right) \cdot \boldsymbol{\xi}. \\
&= \left(\left[\frac{\tilde{f}_\delta(\mathbf{x}+\rho\boldsymbol{\xi}) - \tilde{f}_\delta(\mathbf{x})}{\rho} - \frac{f(\mathbf{x}+\rho\boldsymbol{\xi}) - f(\mathbf{x})}{\rho} \right] + \frac{\rho}{2} \|\boldsymbol{\xi}\|_{\mathbf{A}}^2 \right) \cdot \boldsymbol{\xi}.
\end{aligned} \tag{71}$$

With the above equality, we have

$$\begin{aligned}
&\mathbb{E} \|\hat{\nabla}_\rho \tilde{f}_\delta(\mathbf{x}) - \tilde{\nabla} f(\mathbf{x})\|_{\mathbf{B}^{k\top}\mathbf{C}\mathbf{B}^k}^2 \\
&\leq 2\mathbb{E} \left[\frac{\tilde{f}_\delta(\mathbf{x}+\rho\boldsymbol{\xi}) - \tilde{f}_\delta(\mathbf{x})}{\rho} - \frac{f(\mathbf{x}+\rho\boldsymbol{\xi}) - f(\mathbf{x})}{\rho} \right] \cdot \left\| \begin{bmatrix} \boldsymbol{\xi} \\ \mathbf{0} \end{bmatrix} \right\|_{\mathbf{B}^{k\top}\mathbf{C}\mathbf{B}^k}^2 \\
&\quad + 2\mathbb{E} \frac{\rho}{2} \|\boldsymbol{\xi}\|_{\mathbf{A}}^2 \cdot \left\| \begin{bmatrix} \boldsymbol{\xi} \\ \mathbf{0} \end{bmatrix} \right\|_{\mathbf{B}^{k\top}\mathbf{C}\mathbf{B}^k}^2 \\
&\leq \frac{4\delta}{\rho} \mathbb{E} \left\| \begin{bmatrix} \boldsymbol{\xi} \\ \mathbf{0} \end{bmatrix} \right\|_{\mathbf{B}^{kT}\mathbf{C}\mathbf{B}^k}^2 + \rho \mathbb{E} \|\boldsymbol{\xi}\|_{\mathbf{A}}^2 \cdot \left\| \begin{bmatrix} \boldsymbol{\xi} \\ \mathbf{0} \end{bmatrix} \right\|_{\mathbf{B}^{kT}\mathbf{C}\mathbf{B}^k}^2 \\
&\leq \frac{4\delta}{\rho} \mathrm{tr} \left(\mathbf{B}^{k\top}\mathbf{C}\mathbf{B}^k \right) + 3\rho \mathrm{tr}(\mathbf{A}) \cdot \mathrm{tr}(\mathbf{B}^{k\top}\mathbf{C}\mathbf{B}^k) \\
&\leq \left(\frac{4\delta}{\rho} + 3\rho \mathrm{tr}(\mathbf{A}) \right) \cdot 4 \sum_{i=1}^d \lambda_i^2 (k+1)^2 a_{\lambda_i}^k.
\end{aligned} \tag{72}$$

Therefore,

$$\begin{aligned}
\mathbb{E} \left\| \sum_{k=0}^{n-1} \mathbf{B}^{n-k-1} \boldsymbol{\epsilon}_k^2 \right\|_{\mathbf{C}}^2 &\leq n \sum_{k=0}^{n-1} \mathbb{E}_k \|\mathbf{B}^{n-k-1} \boldsymbol{\epsilon}_k^2\|_{\mathbf{C}}^2 \\
&\leq n \sum_{k=1}^{n} \left(\frac{4\delta}{\rho} + 3\rho \mathrm{tr}(\mathbf{A}) \right) \cdot 4 \sum_{i=1}^d \lambda_i^2 (n-k)^2 a_{\lambda_i}^{n-k} \\
&= n \left(\frac{16\delta}{\rho} + 12\rho \mathrm{tr}(\mathbf{A}) \right) \cdot \sum_{i=1}^d \sum_{k=1}^{n} \lambda_i^2 (n-k)^2 a_{\lambda_i}^{n-k}.
\end{aligned} \tag{73}$$

Finally, we use induction to prove that $\mathbb{E}\|\mathbf{z}_n\|_{\mathbf{C}}^2 < 200(1-b)^n \|\mathbf{z}_0\|_{\mathbf{C}}^2$ where $b = 1 - \frac{\sqrt{h\mu}}{4}$ when $\frac{16\delta}{\rho} + 12\rho \mathrm{tr}(\mathbf{A})$ is small. Suppose that for $k < n$, we have $\mathbb{E}\|\mathbf{z}_k\|_{\mathbf{C}}^2 < 200(1-b)^n \|\mathbf{z}_0\|_{\mathbf{C}}^2$. By (59),

we have

$$\begin{aligned}\mathbb{E}\|\mathbf{z}_n\|_{\mathbf{C}}^2 \leq 120\left(1-\frac{\sqrt{h\mu}}{2}\right)^{2n-2}\|\mathbf{z}_0\|_{\mathbf{C}}^2 \\ + 72h^2\sum_{i=1}^{d}\sum_{k=0}^{n-1}\lambda_i^2(n-k)^2 a_{\lambda_i}^{n-k}\cdot\|\mathbf{y}_k\|_{\mathbf{A}^2}^2 \\ + 3nh^2\left(\frac{16\delta}{\rho}+12\rho\mathrm{tr}(\mathbf{A})\right)\cdot\sum_{i=1}^{d}\sum_{k=1}^{n}\lambda_i^2(n-k)^2 a_{\lambda_i}^{n-k}.\end{aligned} \tag{74}$$

By the definition of $\mathbf{y}_k$ and the assumption for induction, we have

$$\mathbb{E}\|\mathbf{y}_k\|_{\mathbf{A}^2}^2 \leq 2000(1-b)^{n-1}\|\mathbf{z}_0\|_{\mathbf{C}}^2. \tag{75}$$

Using the summation result:

$$\sum_{k=1}^{n}k^2a^k < \frac{1}{(1-a)^3}, \tag{76}$$

we have

$$\begin{aligned}\mathbb{E}\|\mathbf{z}_n\|_{\mathbf{C}}^2 &\leq 120\left(1-\frac{\sqrt{h\mu}}{2}\right)^{2n-2}\|\mathbf{z}_0\|_{\mathbf{C}}^2 \\ &\quad + 144000(1-b)^{n-1}\sum_{i=1}^{d}\frac{h^2\lambda_i^2}{\left(1-\frac{a_{\lambda_i}}{b}\right)^3}\|\mathbf{z}_0\|_{\mathbf{C}}^2 \\ &\quad + 3n\left(\frac{16\delta}{\rho}+12\rho\mathrm{tr}(\mathbf{A})\right)\cdot\sum_{i=1}^{d}\frac{h^2\lambda_i^2}{(1-a_{\lambda_i})^3} \\ &\leq 120\left(1-\frac{\sqrt{h\mu}}{2}\right)^{2n-2}\|\mathbf{z}_0\|_{\mathbf{C}}^2 \\ &\quad + 576000(1-b)^{n-1}\sum_{i=1}^{d}\sqrt{h\lambda_i}\|\mathbf{z}_0\|_{\mathbf{C}}^2 \\ &\quad + 6n\left(\frac{16\delta}{\rho}+12\rho\mathrm{tr}(\mathbf{A})\right)\cdot\sum_{i=1}^{d}\sqrt{h\lambda_i}.\end{aligned} \tag{77}$$

By $h=\frac{1}{14400^2(\sum_i\lambda_i^{1/2})^2}$, we have

$$\mathbb{E}\|\mathbf{z}_n\|_{\mathbf{C}}^2 \leq 160(1-b)^{n-1}\|\mathbf{z}_0\|_{\mathbf{C}}^2 + 6n\left(\frac{16\delta}{\rho}+12\rho\mathrm{tr}(\mathbf{A})\right)\cdot\sum_{i=1}^{d}\sqrt{h\lambda_i}. \tag{78}$$

Therefore, if $6n\left(\frac{16\delta}{\rho}+12\rho\mathrm{tr}(\mathbf{A})\right) < 40(1-b)^{n-1}\|\mathbf{z}_0\|_{\mathbf{C}}^2$, we have $\mathbb{E}\|\mathbf{z}_n\|_{\mathbf{C}}^2 < 200(1-b)^n\|\mathbf{z}_0\|_{\mathbf{C}}^2$.

Finally, we have

$$\begin{aligned}\|\mathbf{z}_n\|_{\mathbf{C}}^2 &= {\mathbf{x}_n}^\top\mathbf{A}^2\mathbf{x}_n + {\mathbf{x}_{n-1}}^\top\mathbf{A}^2\mathbf{x}_{n-1} \\ &\geq \mu\left({\mathbf{x}_n}^\top\mathbf{A}\mathbf{x}_n + {\mathbf{x}_{n-1}}^\top\mathbf{A}\mathbf{x}_{n-1}\right) \\ &= 2\mu(f(\mathbf{x}_n)+f(\mathbf{x}_{n-1})),\end{aligned} \tag{79}$$

and

$$\begin{aligned} \|\mathbf{z}_0\|_{\mathbf{C}}^2 &= 2\mathbf{x}_0{}^\top \mathbf{A}^2 \mathbf{x}_n \\ &\leq 2L\mathbf{x}_0{}^\top \mathbf{A}\mathbf{x}^0 \\ &= 4Lf(\mathbf{x}_0). \end{aligned} \tag{80}$$

Therefore,

$$\begin{aligned} \mathbb{E}f(\mathbf{x}_n) &\leq \frac{1}{2\mu} \cdot 200(1-b)^n \cdot (4Lf(\mathbf{x}_0)) \\ &= 400 \cdot \frac{L}{\mu} \cdot \left(1 - \frac{\mu}{57600 \sum_i \lambda_i^{1/2}}\right)^n \cdot f(\mathbf{x}_0). \end{aligned} \tag{81}$$

■

Appendix E. Proof of Theorem 16

In this section, we give the proof of Theorem 16.

E.1. Proof of Main Results

We first present a theorem on the number of iterations of Algorithm 4, whose proof can be found in Monteiro and Svaiter (2013):

Theorem 20 (Theorem 4.1 in Monteiro and Svaiter (2013)) *If all the parameters satisfy the requirements of Algorithm 4, then for every integer $1 \leq k \leq n$, the following statements hold:*

$$A_k \geq \left(\frac{2}{3}\right)^{7/2} \cdot \left(\frac{\sigma_l(1-\sigma^2)^{1/2}}{16DH}\right) \cdot k^{7/2}, \tag{82}$$

and

$$f(\mathbf{y}_k) - f^* \leq \frac{3^{7/2}}{\sqrt{2}} \frac{HD^3}{\sigma_l \sqrt{1-\sigma^2}} \frac{1}{k^{7/2}}. \tag{83}$$

Now we give the proof of Theorem 16 below.

Theorem 16 *Assume the objective function f is convex and has L-continuous gradient and H-continuous Hessian matrices. Algorithm 4 needs*

$$\tilde{\mathcal{O}}\left(\frac{D \cdot \mathrm{ED}_{1/2}}{\epsilon^{1/2}} + d \cdot D^{6/7} H^{2/7} \epsilon^{-2/7}\right) \tag{84}$$

zeroth-order oracle calls to find an ϵ-approximated solution with high probability.

Proof Theorem 20 analyzes the outer loop of Algorithm 4, so we only need to analyze the inner loop of Algorithm 4, namely Algorithm 6 and Algorithm 5. For Algorithm 5, the zeroth-order oracle is called $\Theta(d)$ times. For Algorithm 6, the problem (22) is solved $\mathcal{O}\left(\left|\log \frac{\lambda_{j+1}}{\lambda_j}\right|\right)$ times. Note that Theorem 14 shows that solving (22) needs

$$\mathcal{O}\left(\left(\left(\frac{1}{\lambda_{\text{temp}}}\right)^{-1/2} \text{ED}_{1/2} + d\right) \cdot \log \frac{1}{\epsilon_B} \cdot \log L\lambda_{\text{temp}}\right) \tag{85}$$

zeroth-order oracle calls in expectation. By Markov's inequality, using same order of such oracles, we can find an approximated solution with a constant probability. So by repeating Algorithm 2 for logarithm times and taking the minimum solution, we can obtain a high probability result (see e.g. Ghadimi and Lan (2013)). Moreover, we have $\lambda_{\text{temp}} \leq \max\{\lambda_j, \lambda_{j+1}\}$. In order to find an ϵ-approximated solution, we need to find the first k such that $A_k \geq \frac{D^2}{\epsilon}$. Suppose that $A_k = \Theta\left(\frac{D^2}{\epsilon}\right)$, and in this case $k = \mathcal{O}\left(D^{6/7}H^{2/7}\epsilon^{-2/7}\right)$. Therefore, ignoring all logarithmic factors, the zeroth-order oracle is called at most

$$\begin{aligned}
&\sum_{j=1}^{k}\left(\tilde{\mathcal{O}}\left(\left(\frac{1}{\lambda_{\text{temp}}}\right)^{-1/2} \text{ED}_{1/2} + d\right) + \Theta(d)\right) \\
\leq &\sum_{j=1}^{k}\tilde{\mathcal{O}}\left(\left(\frac{1}{\max\{\lambda_j, \lambda_{j+1}\}}\right)^{-1/2} \text{ED}_{1/2} + d\right) \\
= &\tilde{\mathcal{O}}\left(\text{ED}_{1/2} \cdot \sum_{j=1}^{k}\sqrt{\lambda_j} + kd\right) \\
\overset{\text{Lemma 28}}{\leq} &\tilde{\mathcal{O}}\left(\text{ED}_{1/2} \cdot \sqrt{A_k} + kd\right) \\
= &\tilde{\mathcal{O}}\left(\frac{D \cdot \text{ED}_{1/2}}{\epsilon^{1/2}} + d \cdot D^{6/7}H^{2/7}\epsilon^{-2/7}\right).
\end{aligned} \tag{86}$$

■

E.2. Properties of Approximate Solutions

In this subsection, we present a new framework for considering errors from inexactly solving solutions. With Lemmas 21 and 22, we show that if ϵ_A and ϵ_B are small enough, the results in Monteiro and Svaiter (2013) still hold with a different numerical constant.

Lemma 21 *If*

$$\epsilon_B < \frac{(\sigma - \sigma_u)^2}{2\lambda_{k+1}(L\lambda_{k+1} + 1 + (\sigma - \sigma_u)^2)\left(L + \frac{1}{\lambda_{k+1}}\right)} \cdot \left(f(\tilde{\mathbf{x}}_k) - \min_y \left\{f_{\tilde{\mathbf{x}}_k}(\mathbf{y}) + \frac{1}{2\lambda_{k+1}}\|\mathbf{y} - \tilde{\mathbf{x}}_k\|^2\right\}\right),$$

$\mathbf{y}_{k+1}$ *satisfies*

$$\|\lambda_{k+1}\nabla f(\mathbf{y}_{k+1}) + \mathbf{y}_{k+1} - \tilde{\mathbf{x}}_k\|^2 \leq \sigma^2\|\mathbf{y}_{k+1} - \tilde{\mathbf{x}}_k^2\|. \tag{87}$$

Proof [Proof of Lemma 21] Denote

$$g(\mathbf{y}) = f_{\tilde{\mathbf{x}}_k}(\mathbf{y}) + \frac{1}{2\lambda_{k+1}}\|\mathbf{y} - \tilde{\mathbf{x}}_k\|^2. \tag{88}$$

By the $L + \frac{1}{\lambda_{k+1}}$-Lipschitz contiouity of ∇g, we have

$$g(\mathbf{y}) - g^* \geq \frac{1}{2\left(L + \frac{1}{\lambda_{k+1}}\right)}\|\nabla g(\mathbf{y})\|^2. \tag{89}$$

Let $\mathbf{y} = \mathbf{y}_{k+1}$ in (89). We have

$$\begin{aligned}\|\lambda_{k+1}\nabla f_{\tilde{\mathbf{x}}_k}(\mathbf{y}_{k+1}) + \mathbf{y}_{k+1} - \tilde{\mathbf{x}}_k\|^2 &\overset{(88)}{=} \lambda_{k+1}^2\|\nabla g(\mathbf{y})\|^2 \\ &\overset{(89)}{\leq} \left(2L\lambda_{k+1}^2 + 2\lambda_{k+1}\right)\left(g(\mathbf{y}_{k+1}) - g^*\right) \\ &\leq \left(2L\lambda_{k+1}^2 + 2\lambda_{k+1}\right)\epsilon_B.\end{aligned} \tag{90}$$

The optimal solution to (22) is

$$\mathbf{y}^* = \tilde{\mathbf{x}}_k - \left(\nabla^2 f(\tilde{\mathbf{x}}_k) + \frac{1}{\lambda_{k+1}}\mathbf{I}\right)^{-1}\nabla f(\tilde{\mathbf{x}}_k) \tag{91}$$

and

$$\begin{aligned}g^* &= f(\tilde{\mathbf{x}}_k) - \frac{1}{2}\left\langle\left(\nabla^2 f(\tilde{\mathbf{x}}_k) + \frac{1}{\lambda_{k+1}}\mathbf{I}\right)^{-1}\nabla f(\tilde{\mathbf{x}}_k), \nabla f(\tilde{\mathbf{x}}_k)\right\rangle \\ &\geq f(\tilde{\mathbf{x}}_k) - \frac{1}{2}\left(L + \frac{1}{\lambda_{k+1}}\right)\|\tilde{\mathbf{x}}_k - \mathbf{y}^*\|^2 \\ &\geq f(\tilde{\mathbf{x}}_k) - \left(L + \frac{1}{\lambda_{k+1}}\right)\left(\|\tilde{\mathbf{x}}_k - \mathbf{y}_{k+1}\|^2 + \|\mathbf{y}_{k+1} - \mathbf{y}^*\|^2\right) \\ &\overset{a}{\geq} f(\tilde{\mathbf{x}}_k) - \left(L + \frac{1}{\lambda_{k+1}}\right)\left(\|\tilde{\mathbf{x}}_k - \mathbf{y}_{k+1}\|^2 + 2\lambda_{k+1}\epsilon_B\right),\end{aligned} \tag{92}$$

where $\overset{a}{\geq}$ uses the λ_{k+1}-strong convexity of g. Therefore, if $\epsilon_B < \frac{(\sigma-\sigma_u)^2}{2\lambda_{k+1}(L\lambda_{k+1}+1+(\sigma-\sigma_u)^2)\left(L+\frac{1}{\lambda_{k+1}}\right)}\cdot(f(\tilde{\mathbf{x}}_k) - g^*)$, we have

$$\begin{aligned}&\|\lambda_{k+1}\nabla f_{\tilde{\mathbf{x}}_k} + \mathbf{y}_{k+1} - \tilde{\mathbf{x}}_k\|^2 \\ \overset{(90)}{\leq}& \left(2L\lambda_{k+1}^2 + 2\lambda_{k+1}\right)\epsilon_B \\ \leq& \frac{(\sigma-\sigma_u)^2}{L + \frac{1}{\lambda_{k+1}}}(f(\tilde{\mathbf{x}}_k) - g^*) \\ &+ \left(2L\lambda_{k+1}^2 + 2\lambda_{k+1} - (2L\lambda_{k+1}^2 + 2\lambda_{k+1} + 2(\sigma-\sigma_u)^2\lambda_{k+1})\right)\epsilon_B \\ \overset{(92)}{=}& (\sigma-\sigma_u)^2\|\mathbf{y}_{k+1} - \tilde{\mathbf{x}}_k\|^2,\end{aligned} \tag{93}$$

and we have

$$
\begin{aligned}
&\|\lambda_{k+1}\nabla f(\mathbf{y}_{k+1}) + \mathbf{y}_{k+1} - \tilde{\mathbf{x}}_k\|^2 \qquad (94)\\
&= \|(\lambda_{k+1}\nabla f_{\tilde{\mathbf{x}}_k}(\mathbf{y}_{k+1}) + \mathbf{y}_{k+1} - \tilde{\mathbf{x}}_k) + (\lambda_{k+1}\nabla f_{\tilde{\mathbf{x}}_k}(\mathbf{y}_{k+1}) - \lambda_{k+1}\nabla f(\mathbf{y}_{k+1}))\|^2\\
&\le \|\lambda_{k+1}\nabla f_{\tilde{\mathbf{x}}_k}(\mathbf{y}_{k+1}) + \mathbf{y}_{k+1} - \tilde{\mathbf{x}}_k\|^2\\
&\quad + 2\|\lambda_{k+1}\nabla f(\mathbf{y}_{k+1}) + \mathbf{y}_{k+1} - \tilde{\mathbf{x}}_k\| \cdot \|\lambda_{k+1}\nabla f(\tilde{\mathbf{x}}_k)\nabla^2 f(\tilde{\mathbf{x}}_k)(\mathbf{y}_{k+1} - \tilde{\mathbf{x}}_k) - \lambda_{k+1}\nabla f(\mathbf{y}_{k+1})\|\\
&\quad + \|\lambda_{k+1}\nabla f(\tilde{\mathbf{x}}_k) + \nabla^2 f(\tilde{\mathbf{x}}_k)(\mathbf{y}_{k+1} - \tilde{\mathbf{x}}_k) - \lambda_{k+1}\nabla f(\mathbf{y}_{k+1})\|^2\\
&\overset{(93)}{\le} (\sigma - \sigma_u)^2\|\mathbf{y}_{k+1} - \tilde{\mathbf{x}}_k\|^2 + 2(\sigma - \sigma_u)\|\mathbf{y}_{k+1} - \tilde{\mathbf{x}}_k\| \cdot \frac{H\lambda_{k+1}\|\mathbf{y}_{k+1} - \tilde{\mathbf{x}}_k\|}{2}\\
&\quad + \left(\frac{H\lambda_{k+1}\|\mathbf{y}_{k+1} - \tilde{\mathbf{x}}_k\|}{2}\right)^2\\
&\le \left(\sigma - \sigma_u + \frac{H}{2} \cdot \frac{2\sigma_u}{H}\right)^2 = \sigma^2.
\end{aligned}
$$

■

Lemma 22 *If $\epsilon_A < \frac{D}{N^{3/2}}$, then we have*

$$\sum_{j=1}^{k} \|\mathbf{x}_j - \mathbf{x}_j^*\|^2 \le D^2. \qquad (95)$$

for $k \le N$.

Proof [Proof of Lemma 22] By the definition of $\mathbf{x}_j$, $\mathbf{x}_j^*$ and A_k, we have

$$
\begin{aligned}
\sum_{j=1}^{k} \|\mathbf{x}_j - \mathbf{x}_j^*\|^2 &= \sum_{j=1}^{k} \left\|\sum_{i=1}^{j} a_i(v_i - \nabla f(\mathbf{y}_i))\right\|^2\\
&\le \sum_{j=1}^{k} \left(\sum_{i=1}^{j} a_i \cdot \frac{\epsilon_A}{a_i}\right)^2 \qquad (96)\\
&= \sum_{j=1}^{k} j^2\epsilon_A^2 \le k^3\epsilon_A^2.
\end{aligned}
$$

Therefore, if $\epsilon_A \le \frac{D}{N^{3/2}}$, we have (95). ■

In order to analyze $f(\mathbf{x}_k)$, we define the affine maps γ_k as

$$\gamma_k(\mathbf{x}) = f(\mathbf{y}_k) + \langle \nabla f(\mathbf{y}_k), \mathbf{x} - \mathbf{y}_k\rangle. \qquad (97)$$

and the aggregate affine maps Γ_k recursively as:

$$\Gamma_0 \equiv 0, \qquad \Gamma_{k+1} = \frac{A_k}{A_{k+1}}\Gamma_k + \frac{a_{k+1}}{A_{k+1}}\gamma_{k+1}. \qquad (98)$$

We define

$$\mathbf{x}_k^* = \mathbf{x}_0 - \sum_{j=1}^{k} a_j \nabla f(\mathbf{y}_{k+1}), \tag{99}$$

Lemma 23 (Lemma 3.2 of Monteiro and Svaiter (2013)) *For every integer $k \geq 0$, there hold:*

1. *γ_{k+1} is affine and $\gamma_{k+1} \leq f$.*
2. *Γ_k is affine and $A_k \Gamma_k \leq A_k f$.*
3. *$\mathbf{x}_k^* = \operatorname{argmin}_{\mathbf{x}} A_k \Gamma_k(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \mathbf{x}_0\|^2$.*

Lemma 24 (Inspired by Lemma 3.4 of Monteiro and Svaiter (2013)) *For integer $k \geq 0$, define*

$$\beta_k = \left(\inf_{\mathbf{x} \in \mathbb{R}^n} A_k \Gamma_k(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \mathbf{x}_0\|^2 \right) - A_k f(\mathbf{y}_k). \tag{100}$$

If $\|\lambda \mathbf{v} + \mathbf{y} - \mathbf{x}\|^2 \leq \sigma^2 \|\mathbf{y} - \mathbf{x}\|^2$, we have $\beta_0 = 0$, and

$$\beta_{k+1} \geq \beta_k + \frac{(1-\sigma^2) A_{k+1}}{2\lambda_{k+1}} \|\mathbf{y}_{k+1} - \tilde{\mathbf{x}}_k\|^2 - \frac{1}{2}\|\mathbf{x}_k - \mathbf{x}_k^*\|^2. \tag{101}$$

Proof [Proof of Lemma 24] We have $\beta_0 = 0$ since $A_0 = 0$. For $\mathbf{x} \in \mathbb{R}^n$, define

$$\tilde{\mathbf{x}} = \frac{A_k}{A_{k+1}} \mathbf{y}_k + \frac{a_{k+1}}{A_{k+1}} \mathbf{x}. \tag{102}$$

By the definition of $\tilde{\mathbf{x}}_k$ in Algorithm 4 and the affinity of γ, we have

$$\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_k = \frac{a_{k+1}}{A_{k+1}} (\mathbf{x} - \mathbf{x}_k), \tag{103}$$

$$\gamma_{k+1}(\tilde{\mathbf{x}}) = \frac{A_k}{A_{k+1}} \gamma_{k+1}(\mathbf{y}_k) + \frac{a_{k+1}}{A_{k+1}} \gamma_{k+1}(\mathbf{x}). \tag{104}$$

We have the following equality:

$$\begin{aligned}
&A_{k+1}\Gamma_{k+1}(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \mathbf{x}_0\|^2 \\
&\overset{(98)}{=} a_{k+1}\gamma_{k+1}(\mathbf{x}) + A_k \Gamma_k(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \mathbf{x}_0\|^2 \\
&\overset{\text{Lemma 23 and (100)}}{=} a_{k+1}\gamma_{k+1}(\mathbf{x}) + A_k f(\mathbf{y}_k) + \beta_k + \frac{1}{2}\|\mathbf{x} - \mathbf{x}_k^*\|^2 \\
&\geq a_{k+1}\gamma_{k+1}(\mathbf{x}) + A_k f(\mathbf{y}_k) + \beta_k + \frac{1}{4}\|\mathbf{x} - \mathbf{x}_k\|^2 - \frac{1}{2}\|\mathbf{x}_k - \mathbf{x}_k^*\|^2 \\
&\overset{\text{Lemma 23}}{\geq} a_{k+1}\gamma_{k+1}(\mathbf{x}) + A_k \gamma_{k+1}(\mathbf{y}_k) + \beta_k + \frac{1}{4}\|\mathbf{x} - \mathbf{x}_k\|^2 - \frac{1}{2}\|\mathbf{x}_k^2 - \mathbf{x}_k^*\|^2 \\
&\overset{(102)}{\geq} A_{k+1}\gamma_{k+1}(\tilde{\mathbf{x}}) + \beta_k + \frac{1}{4}\|\mathbf{x} - \mathbf{x}_k\|^2 - \frac{1}{2}\|\mathbf{x}_k - \mathbf{x}_k^*\|^2 \\
&\overset{(102)}{\geq} A_{k+1}\gamma_{k+1}(\tilde{\mathbf{x}}) + \beta_k + \frac{A_{k+1}^2}{4a_{k+1}^2}\|\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_k\|^2 - \frac{1}{2}\|\mathbf{x}_k - \mathbf{x}_k^*\|^2 \\
&\overset{\text{Lemma 26}}{\geq} \beta_k + A_{k+1}\left(\gamma_{k+1}(\tilde{\mathbf{x}}) + \frac{1}{4\lambda_{k+1}}\|\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_k\|^2\right) - \frac{1}{2}\|\mathbf{x}_k - \mathbf{x}_k^*\|^2.
\end{aligned} \tag{105}$$

With (105), we have

$$\begin{aligned}
&\beta_{k+1} + A_{k+1} f(\mathbf{y}_{k+1}) \\
&\overset{(100)}{=} \inf_{\mathbf{x}} \left\{ A_{k+1}\Gamma_{k+1} + \frac{1}{2}\|\mathbf{x} - \mathbf{x}_0\|^2 \right\} \\
&\overset{(105)}{\geq} \beta_k + A_{k+1} \inf_{\tilde{\mathbf{x}}} \left\{ \gamma_{k+1}(\tilde{\mathbf{x}}) + \frac{1}{2\lambda_{k+1}}\|\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_k\|^2 \right\} - \frac{1}{2}\|\mathbf{x}_k - \mathbf{x}_k^*\|^2 \\
&\overset{(97)}{\geq} \beta_k + A_{k+1} f(\mathbf{y}_{k+1}) + A_{k+1} \inf_{\tilde{\mathbf{x}}} \left\{ \langle \nabla f(\mathbf{y}_k), \tilde{\mathbf{x}} - \mathbf{y}_{k+1} \rangle + \frac{1}{4\lambda_{k+1}}\|\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_k\|^2 \right\} - \frac{1}{2}\|\mathbf{x}_k - \mathbf{x}_k^*\|^2 \\
&\overset{\text{Lemma } 27}{\geq} \beta_k + A_{k+1} f(\mathbf{y}_{k+1}) + \frac{(1-\sigma^2)A_{k+1}}{4\lambda_{k+1}}\|\mathbf{y}_j - \tilde{\mathbf{x}}_{j-1}\|^2 - \frac{1}{2}\|\mathbf{x}_k - \mathbf{x}_k^*\|^2
\end{aligned} \tag{106}$$

Therefore, we have (101). ■

Lemma 25 *Let* $D = \|\mathbf{x}_0 - \mathbf{x}^*\|$. *If*

$$\sum_{j=1}^{k} \|\mathbf{x}_j - \mathbf{x}_j^*\|^2 \leq D^2, \tag{107}$$

then for every integer $k \geq 1$,

$$\frac{1}{4}\|\mathbf{x}_k - \mathbf{x}^*\|^2 + A_k[f(\mathbf{y}_k) - f^*] + \frac{1-\sigma^2}{4}\sum_{j=1}^{k} \frac{A_j}{\lambda_j}\|\mathbf{y}_j - \tilde{\mathbf{x}}_{j-1}\|^2 \leq D^2. \tag{108}$$

As a consequence,

$$f(\mathbf{y}_k) - f^* \leq \frac{D^2}{A_k}, \qquad \|\mathbf{x}_k - \mathbf{x}^*\| \leq 2D, \tag{109}$$

and if $\sigma^2 \leq 1$,

$$\sum_{j=1}^{k} \frac{A_k}{\lambda_j}\|\mathbf{y}_j - \mathbf{x}_{j-1}\|^2 \leq \frac{4D^2}{1-\sigma^2}. \tag{110}$$

Proof [Proof of Lemma 25] Summing (101) from $k = 0$ to $k - 1$, we have

$$\beta_k \geq \frac{1-\sigma^2}{2}\sum_{j=1}^{k} \frac{A_{k+1}}{\lambda_{k+1}}\|\mathbf{y}_{k+1} - \tilde{\mathbf{x}}_k\|^2 - \frac{1}{2}\sum_{j=1}^{k-1}\|\mathbf{x}_k - \mathbf{x}_k^*\|^2. \tag{111}$$

Using the definition of β_k in (100), we have

$$\begin{aligned}
&A_k f(\mathbf{y}_k) + \frac{1-\sigma^2}{4}\sum_{j=1}^{k} \frac{A_j}{\lambda_j}\|\mathbf{y}_j - \tilde{\mathbf{x}}_{j-1}\|^2 \\
&\leq \inf_{\mathbf{x}\in\mathbb{R}^n} \left(A_k\Gamma_k + \frac{1}{2}\|\mathbf{x} - \mathbf{x}_0\|^2 \right) + \frac{1}{2}\sum_{j=1}^{k-1}\|\mathbf{x}_j - \mathbf{x}_j^*\|^2.
\end{aligned} \tag{112}$$

With Lemma 23, we have

$$\inf_{\mathbf{x}\in\mathbb{R}^n}\left(A_k\Gamma_k(\mathbf{x})+\frac{1}{2}\|\mathbf{x}-\mathbf{x}_0\|^2\right)+\frac{1}{2}\|\mathbf{x}-\mathbf{x}_k^*\|^2=A_k\Gamma_k(\mathbf{x})+\frac{1}{2}\|\mathbf{x}-\mathbf{x}_0\|^2. \tag{113}$$

Plugging (113) into (112), we have

$$\begin{aligned}
&A_kf(\mathbf{y}_k)+\frac{1-\sigma^2}{4}\sum_{j=1}^{k}\frac{A_j}{\lambda_j}\|\mathbf{y}_j-\tilde{\mathbf{x}}_{j-1}\|^2+\frac{1}{4}\|\mathbf{x}-\mathbf{x}_k\|^2\\
\le\ &A_k\Gamma_k(\mathbf{x})+\frac{1}{4}\|\mathbf{x}-\mathbf{x}_k\|^2-\frac{1}{2}\|\mathbf{x}-\mathbf{x}_k^*\|^2+\frac{1}{2}\|\mathbf{x}-\mathbf{x}_0\|^2+\frac{1}{2}\sum_{j=1}^{k-1}\|\mathbf{x}_j-\mathbf{x}_j^*\|^2\\
\le\ &A_k\Gamma_k(\mathbf{x})+\frac{1}{2}\|\mathbf{x}-\mathbf{x}_0\|^2+\frac{1}{2}\sum_{j=1}^{k}\|\mathbf{x}_j-\mathbf{x}_j^*\|^2
\end{aligned} \tag{114}$$

Letting $\mathbf{x}=\mathbf{x}^*$ in (114), we have

$$\begin{aligned}
&A_kf(\mathbf{y}_k)+\frac{1-\sigma^2}{4}\sum_{j=1}^{k}\frac{A_j}{\lambda_j}\|\mathbf{y}_j-\tilde{\mathbf{x}}_{j-1}\|^2+\frac{1}{4}\|\mathbf{x}^*-\mathbf{x}_k\|^2\\
\le\ &A_k\Gamma_k(\mathbf{x}^*)+\frac{1}{2}\|\mathbf{x}^*-\mathbf{x}_0\|^2+\frac{1}{2}\sum_{j=1}^{k}\|\mathbf{x}_j-\mathbf{x}_j^*\|^2\\
\le\ &A_kf^*+\frac{1}{2}\|\mathbf{x}^*-\mathbf{x}_0\|^2+\frac{1}{2}\sum_{j=1}^{k}\|\mathbf{x}_j-\mathbf{x}_j^*\|^2.
\end{aligned} \tag{115}$$

Therefore, using Lemma 23 and (107), we have

$$A_k(f(\mathbf{y}_k)-f^*)+\frac{1-\sigma^2}{4}\sum_{j=1}^{k}\frac{A_j}{\lambda_j}\|\mathbf{y}_j-\tilde{\mathbf{x}}_{j-1}\|^2+\frac{1}{4}\|\mathbf{x}^*-\mathbf{x}_k\|^2\le D^2 \tag{116}$$

■

E.3. Useful Lemmas in Monteiro and Svaiter (2013)

In this subsection, we list some results leading to Theorem 20 in Monteiro and Svaiter (2013).

Lemma 26 (Lemma 3.1 of Monteiro and Svaiter (2013))

$$\lambda_{k+1}A_{k+1}=a_{k+1}^2. \tag{117}$$

Lemma 27 (Lemma 3.3 of Monteiro and Svaiter (2013)) *The inequality*

$$\|\lambda\mathbf{v}+\mathbf{y}-\mathbf{x}\|^2\le\sigma^2\|\mathbf{y}-\mathbf{x}\|^2 \tag{118}$$

is equivalent to inequality

$$\min_{\mathbf{z}\in\mathbb{R}^n}\left\{\langle \mathbf{v}, \mathbf{z}-\mathbf{y}\rangle + \frac{1}{2\lambda}\|\mathbf{z}-\mathbf{x}\|^2\right\} \geq \frac{1-\sigma^2}{2\lambda}\|\mathbf{y}-\mathbf{x}\|^2. \tag{119}$$

Lemma 28 (Lemma 3.7 of Monteiro and Svaiter (2013)) *For every integer $k \geq 0$,*

$$\sqrt{A_{k+1}} \geq \sqrt{A_k} + \frac{1}{2}\sqrt{\lambda_{k+1}}. \tag{120}$$

Lemma 29 (Lemma 4.2 of Monteiro and Svaiter (2013)) *If all the parameters satisfy the requirements of Algorithm 4, then for every integer $1 \leq k \leq N$,*

$$\sum_{j=1}^{k} \frac{A_j}{\lambda_j^3} \leq \frac{H^2D^2}{\sigma_l^2(1-\sigma^2)}. \tag{121}$$

Lemma 30 (Lemma 4.4 of Monteiro and Svaiter (2013)) *If all the parameters satisfy the requirements of Algorithm 4, then for $1 \leq k \leq N$,*

$$A_k \geq \frac{1}{4}w\left(\sum_{j=1}^{k} A_j^{1/3}\right)^{7/3}, \tag{122}$$

where

$$w = \frac{\sigma_l^2(1-\sigma^2)}{4H^2D^2}. \tag{123}$$

Appendix F. Proof of Theorem 17

In this section, we provide the proof of Theorem 17.

F.1. Properties of Approximated Solutions

We define

$$\tilde{\mathbf{x}}_{k+1} = \operatorname*{argmin}_{\mathbf{y}} f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{y}-\mathbf{x}_k\rangle + \frac{1}{2}\langle \nabla^2 f(\mathbf{x}_k)(\mathbf{y}-\mathbf{x}_k), \mathbf{y}-\mathbf{x}_k\rangle + \frac{H}{6}\|\mathbf{y}-\mathbf{x}_k\|^3, \tag{124}$$

which is the exact solution of the cubic regularization subproblem, and

$$\tilde{r}_{k+1} = \|\tilde{\mathbf{x}}_{k+1} - \mathbf{x}_k\|. \tag{125}$$

We first present some results which considers the error of inexact solutions.

Lemma 31 *If* $\epsilon_C < \min\left\{\frac{\epsilon}{800\left(\frac{16\cdot(24\Delta/H)^{1/3}}{\sqrt{\epsilon H}}+1\right)}, \frac{\epsilon}{800}, \frac{1}{2000}\left(\frac{\epsilon}{H}\right)^{3/2}\right\}$ *and* $\epsilon_D < \min\left\{\frac{\sqrt{\frac{\epsilon}{H}}}{200\left(\frac{16\cdot(24\Delta/H)^{1/3}}{\sqrt{\epsilon H}}+1\right)}, \sqrt{\frac{\epsilon}{40000H}}\right\}$, *then we have*

$$\|\tilde{\mathbf{x}}_k - \mathbf{x}_k\| < \sqrt{\frac{\epsilon}{10000H}}. \tag{126}$$

Proof [Proof of Lemma 31] For any $r \geq 0$, Define

$$g_r(\mathbf{y}) = f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{y} - \mathbf{x}_k\rangle + \frac{1}{2}\langle \nabla^2 f(\mathbf{x}_k)(\mathbf{y} - \mathbf{x}_k), \mathbf{y} - \mathbf{x}_k\rangle + \frac{rH}{2}\|\mathbf{y} - \mathbf{x}_k\|^2. \tag{127}$$

We first show that

$$\tilde{\mathbf{x}}_{k+1} = \operatorname*{argmin}_{\mathbf{y}} g_{\tilde{r}_{k+1}}(\mathbf{y}). \tag{128}$$

Indeed, according to Lemma 36, $g_{\tilde{r}_{k+1}}$ is $\frac{\tilde{r}_{k+1}H}{2}$-strongly convex, and according to (124), we have $\nabla g_{\tilde{r}_{k+1}}(\tilde{\mathbf{x}}_{k+1}) = \mathbf{0}$. Thus we have (128).

By the definition of r_{k+1} and r_u, we have

$$\tilde{r}_{k+1} + 4\frac{\epsilon_C}{\sqrt{\epsilon H}} + \epsilon_D \geq r_l + \epsilon_D \geq r_u \geq \tilde{r}_{k+1} - 4\frac{\epsilon_C}{r_{k+1}H} \geq \tilde{r}_{k+1} - 4\frac{\epsilon_C}{\sqrt{\epsilon H}}, \tag{129}$$

and

$$\tilde{r}_{k+1} + 4\frac{\epsilon_C}{\sqrt{\epsilon H}} \geq r_l \geq \tilde{r}_u - \epsilon_D \geq \tilde{r}_{k+1} - 4\frac{\epsilon_C}{\sqrt{\epsilon H}} - \epsilon_D. \tag{130}$$

Therefore,

$$\begin{aligned}\|\mathbf{x}_{k+1} - \tilde{\mathbf{x}}_{k+1}\| &\leq \frac{8\left(4\frac{\epsilon_C}{\sqrt{\epsilon H}} + \epsilon_D\right)\|\tilde{\mathbf{x}}_{k+1} - \mathbf{x}_k\|}{\tilde{r}_{k+1}H} + \left(4\frac{\epsilon_C}{\sqrt{\epsilon H}} + \epsilon_D\right) \\ &\leq \frac{16\left(4\frac{\epsilon_C}{\sqrt{\epsilon H}} + \epsilon_D\right)(24\Delta/H)^{1/3}}{\sqrt{\epsilon H}} + \left(4\frac{\epsilon_C}{\sqrt{\epsilon H}} + \epsilon_D\right)\end{aligned} \tag{131}$$

Therefore, it can be verified that if the assumptions of Lemma 31 are satisfied, then $\|\tilde{\mathbf{x}}_k - \mathbf{x}_k\| < \sqrt{\frac{\epsilon}{H}}$. ■

Lemma 32 *Define*

$$\tilde{f}_{\mathbf{x}}(\mathbf{y}) = f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle + \langle \nabla^2 f(\mathbf{x})(\mathbf{y} - \mathbf{x}), \mathbf{y} - \mathbf{x}\rangle + \frac{H}{6}\|\mathbf{y} - \mathbf{x}\|^3. \tag{132}$$

Then we have

$$\tilde{f}_{\mathbf{x}_k}(\mathbf{x}_{k+1}) - \tilde{f}_{\mathbf{x}_k}(\tilde{\mathbf{x}}_{k+1}) \leq \frac{H\tilde{r}_{k+1}^3}{500}. \tag{133}$$

Proof [Proof of Lemma 32]

$$\begin{aligned}\tilde{f}_{x_k}(\mathbf{x}_{k+1}) &= g_{r_{k+1}}(\tilde{\mathbf{x}}_{k+1}) + \frac{H}{6}\|\mathbf{x}_{k+1}-\mathbf{x}_k\|^3 - \frac{Hr_{k+1}}{2}\|\mathbf{x}_{k+1}-\mathbf{x}_k\|^2 \\ &\le g_{r_{k+1}}(\mathbf{x}^*_{k+1}) + \epsilon_C + \frac{H}{6}\|\mathbf{x}_{k+1}-\mathbf{x}_k\|^3 - \frac{Hr_{k+1}}{2}\|\mathbf{x}_{k+1}-\mathbf{x}_k\|^2 \\ &\le g_{\tilde{r}_{k+1}}(\tilde{\mathbf{x}}_{k+1}) + \frac{H(r_{k+1}-\tilde{r}_{k+1})}{2}\|\tilde{\mathbf{x}}_{k+1}-\mathbf{x}_k\|^2 + \epsilon_C \\ &\quad + \frac{H}{6}\|\mathbf{x}_{k+1}-\mathbf{x}_k\|^3 - \frac{Hr_{k+1}}{2}\|\mathbf{x}_{k+1}-\mathbf{x}_k\|^2 \\ &= \tilde{f}_{\mathbf{x}_k}(\tilde{\mathbf{x}}_{k+1}) + \frac{H\tilde{r}^3_{k+1}}{3} + \frac{H(r_{k+1}-\tilde{r}_{k+1})\tilde{r}^2_{k+1}}{2} + \epsilon_C \\ &\quad + \frac{H}{6}\|\mathbf{x}_{k+1}-\mathbf{x}_k\|^3 - \frac{Hr_{k+1}}{2}\|\mathbf{x}_{k+1}-\mathbf{x}_k\|^2.\end{aligned} \tag{134}$$

By (129) and (130) and the definition of ϵ_C and ϵ_D, we have

$$\frac{100}{101}\tilde{r}_{k+1} \le \|\mathbf{x}_{k+1}-\mathbf{x}_k\| < \frac{100}{99}\tilde{r}_{k+1}, \quad r_{k+1}-\tilde{r}_{k+1} \le \frac{1}{99}\tilde{r}_{k+1}. \tag{135}$$

Therefore,

$$\tilde{f}_{\mathbf{x}_k}(\mathbf{x}_{k+1}) - \tilde{f}_{\mathbf{x}_k}(\tilde{\mathbf{x}}_{k+1}) \le \frac{H\tilde{r}^3_{k+1}}{500}. \tag{136}$$

■

Theorem 33 *We have*

$$\|\nabla f(\mathbf{x}_k)\| \le H\tilde{r}_k^2 + \frac{\epsilon}{100}, \quad \nabla^2 f(\mathbf{x}_k) \succeq -\left(\frac{H\tilde{r}_k}{2} + \frac{\sqrt{H\epsilon}}{100}\right)\mathbf{I} \tag{137}$$

Proof [Proof of Theorem 33] The results of $\nabla^2 f(\tilde{\mathbf{x}}_k)$ follows from Lemma 36, Lemma 38, and Lemma 31.

By the $(L + Hr_{k+1})$-Lipschitz continuity of ∇f

$$\|\nabla f(\mathbf{x}_k) + \nabla^2 f(\mathbf{x}_k)(\mathbf{x}_{k+1}-\mathbf{x}_k) + Hr_{k+1}(\mathbf{x}_{k+1}-\mathbf{x})\| \le \sqrt{2(L+Hr_{k+1})\epsilon_C}, \tag{138}$$

and

$$\|\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k) - \nabla^2 f(\mathbf{x}_k)(\mathbf{x}_{k+1}-\mathbf{x}_k)\| \le \frac{H}{2}\|\mathbf{x}_{k+1}-\mathbf{x}_k\|^2. \tag{139}$$

Therefore, we have

$$\begin{aligned}\|\nabla f(\mathbf{x}_{k+1})\| &\le Hr_{k+1}\|\mathbf{x}_{k+1}-\mathbf{x}_k\| + \frac{H}{2}\|\mathbf{x}_{k+1}-\mathbf{x}_k\|^2 + \sqrt{2(L+Hr_{k+1})\epsilon_C} \\ &\le 2H\tilde{r}^2_{k+1} + \frac{\epsilon}{100}.\end{aligned} \tag{140}$$

■

Theorem 34 (Theorem 1 of Nesterov (2007)) *Define* $\Delta = f(\mathbf{x}^0) - f^*$. *If*

$$\epsilon_C < \min\left\{\frac{\epsilon}{800\left(\frac{16\cdot(24\Delta/H)^{1/3}}{\sqrt{\epsilon H}}+1\right)}, \frac{\epsilon}{800}, \frac{1}{2000}\left(\frac{\epsilon}{H}\right)^{3/2}\right\}$$

and $\epsilon_D < \min\left\{\frac{\sqrt{\frac{\epsilon}{H}}}{200\left(\frac{16\cdot(24\Delta/H)^{1/3}}{\sqrt{\epsilon H}}+1\right)}, \sqrt{\frac{\epsilon}{40000H}}\right\}$,

$$\sum_{i=0}^{\infty} \|\tilde{\mathbf{x}}_{i+1} - \mathbf{x}_i\|^3 \leq \frac{24\Delta}{H}. \tag{141}$$

Proof [Proof of Theorem 34] By Lemma 39 and Lemma 32, we have

$$\begin{aligned} f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) &\geq f(\tilde{\mathbf{x}}_k) - \tilde{f}_{\mathbf{x}_k}(\mathbf{x}_{k+1}) \\ &= f(\tilde{\mathbf{x}}_k) - \tilde{f}_{\mathbf{x}_k}(\tilde{\mathbf{x}}_{k+1}) + \tilde{f}_{\mathbf{x}_k}(\tilde{\mathbf{x}}_{k+1}) - \tilde{f}_{\mathbf{x}_k}(\mathbf{x}_{k+1}) \\ &\geq \frac{H\tilde{r}_{k+1}^3}{12} - \frac{H\tilde{r}_{k+1}^3}{500} \geq \frac{H\tilde{r}_{k+1}^3}{24}. \end{aligned} \tag{142}$$

Summing (142) from $k = 0$ to ∞ yields the desired result. ■

F.2. Proof of Main Results

By Theorems 34 and 33, we have the following theorem:

Theorem 35 *Algorithm 8 finds an* $(\epsilon, \sqrt{H\epsilon})$*-stationary point in* $\frac{48\sqrt{2}H^{1/2}\Delta}{\epsilon^{3/2}}$ *rounds.*

Theorem 17 *Assume the objective function* f *is convex and has* L*-continuous gradients and* H*-continuous Hessian matrices. Algorithm 4 finds an* $(\epsilon, \sqrt{H\epsilon})$*-SSP of* f *in*

$$\tilde{\mathcal{O}}\left(\mathrm{ET}_{1/2}H^{1/4}\Delta\epsilon^{-7/4} + dH^{1/2}\Delta\epsilon^{-3/2}\right) \tag{143}$$

zeroth-order oracle calls with high probability.

Proof We analyze the inner loop of Algorithm 8, namely Algorithm 9. In Algorithm 8, problem (23) is solved

$$\mathcal{O}\left(\left|\log\frac{r_k}{r_{k+1}}\right| + \max\left\{1, \log\frac{r_{k+1}}{\epsilon_D}\right\}\right) \tag{144}$$

times. By Theorem 14, solving subproblem (23) needs

$$\mathcal{O}\left(\left((Hr_{\mathrm{temp}})^{-1/2}\mathrm{ET}_{1/2}(f) + d\right)\cdot\log\frac{1}{\epsilon_C}\cdot\log\frac{L}{Hr_{\mathrm{temp}}}\right) \tag{145}$$

calls to the zeroth-order oracle in average. By Markov's inequality, using same order of such oracles, we can find an approximated solution with a constant probability. So by repeating Algorithm 2 for

logarithm times and taking the minimum solution. So by repeating Algorithm 2 for logarithm times, we can obtain a high probability result (see e.g. Ghadimi and Lan (2013)). The maximum calls in solving one problem depends on the smallest possible value of r_{temp}. It can be verified that $r_{\text{temp}} \geq \min\left\{r_k, \frac{r_{k+1}}{2}\right\}$ in Algorithm 9. Assume without loss of generality that $r_k \geq \sqrt{\frac{\epsilon}{2H}}$ for $k < N$, and $r_N < \sqrt{\frac{\epsilon}{2H}}$ where $N = \mathcal{O}\left(H^{1/2}\Delta\epsilon^{-3/2}\right)$. By the definition of r_N, we have $r_N \geq \epsilon_D = \Omega\left(\sqrt{\frac{\epsilon}{H}}\right)$. The logarithm factors in (144) and (145) can be bounded by linear combinations of $\log\frac{1}{\epsilon}$, $\log\Delta$, $\log L$ and $\log H$. Therefore, ignoring all logarithmic factors, the zeroth-order oracle is called at most

$$\begin{aligned}
&\sum_{j=1}^{k} \tilde{\mathcal{O}}\left((H\min\{r_j, r_{j-1}\})^{-1/2}\mathrm{ET}_{1/2}(f) + d\right) \\
\leq &\sum_{j=1}^{k} \tilde{\mathcal{O}}\left((\sqrt{H\epsilon})^{-1/2}\mathrm{ET}_{1/2}(f) + d\right) \\
= &\tilde{\mathcal{O}}\left(\mathrm{ET}_{1/2}(f)H^{1/4}\Delta\epsilon^{-7/4} + dH^{1/2}\Delta\epsilon^{-3/2}\right).
\end{aligned} \tag{146}$$

■

F.3. Useful Results in Nesterov and Polyak (2006)

In this subsection, we present some results in Nesterov and Polyak (2006), which we is used in our analysis.

Lemma 36 (Proposition 1 of Nesterov and Polyak (2006))

$$\nabla^2 f(\mathbf{x}) + \frac{1}{2}H\|\tilde{\mathbf{x}}_{k+1} - \mathbf{x}_k\|\mathbf{I} \succeq \mathbf{0}. \tag{147}$$

Lemma 37 (Lemma 2 of Nesterov and Polyak (2006)) *For any $k \geq 0$, we have*

$$\langle \nabla f(\mathbf{x}_k), \mathbf{x}_k - \tilde{\mathbf{x}}_{k+1}\rangle \geq 0. \tag{148}$$

Lemma 38 (Lemma 3 of Nesterov and Polyak (2006)) *For any $k \geq 0$, we have*

$$\|\nabla f(\tilde{\mathbf{x}}_{k+1})\| \leq H\|\tilde{\mathbf{x}}_{k+1} - \mathbf{x}_k\|^2. \tag{149}$$

Lemma 39 (Lemma 4 of Nesterov and Polyak (2006))

$$f(\mathbf{x}_k) - f(\tilde{\mathbf{x}}_{k+1}) \geq \frac{H}{12}\|\tilde{\mathbf{x}}_{k+1} - \mathbf{x}_k\|^3. \tag{150}$$

Appendix G. Proof of Additional Lemmas

Proposition 5 *Assume for any* $\mathbf{x}$ *and* $\alpha > 0$*, there exists constant* $C > 0$ *and* $\beta > 0$ *such that* $\sigma_i(\nabla^2 f(\mathbf{x})) \leq \frac{C}{i^\beta}$ *for* $i \in [d]$*, then we have*

$$\mathrm{ED}_\alpha \leq \begin{cases} \frac{2^{\alpha\beta-1}C^\alpha}{\alpha\beta-1}, & \alpha\beta > 1, \quad \textit{dimensional free}, \\ C^\alpha \log(2d+1), & \alpha\beta = 1, \quad \textit{logarithmic growth on } d, \\ \frac{C^\alpha}{1-\alpha\beta}(d+1)^{1-\alpha\beta}, & \alpha\beta < 1, \quad \textit{improve by a } \Theta\left(d^{\alpha\beta}\right) \textit{factor}. \end{cases} \tag{151}$$

Proof

$$\begin{aligned} \mathrm{ED}_\alpha &= \sup_{\mathbf{x}\in\mathbb{R}^d} \sum_{i=1}^d \sigma_i^\alpha(\nabla^2 f(\mathbf{x})) \leq \sum_{i=1}^d \left(\frac{C}{i^\beta}\right)^\alpha = C^\alpha \sum_{i=1}^d i^{-\alpha\beta} \leq C^\alpha \int_{\frac{1}{2}}^{d+\frac{1}{2}} x^{-\alpha\beta} \mathrm{d}x \\ &= \begin{cases} \frac{C^\alpha}{1-\alpha\beta}\left(\left(d+\frac{1}{2}\right)^{1-\alpha\beta} - \left(\frac{1}{2}\right)^{1-\alpha\beta}\right), & \alpha\beta \neq 1, \\ C^\alpha \log(2d+1), & \alpha\beta = 1. \end{cases} \end{aligned} \tag{152}$$

■

Proposition 6 *For the objective in* (3) *that satisfies Assumptions 4 and 5, we have*

$$\mathrm{ED}_\alpha \leq \begin{cases} (L_0 R)^\alpha, & \alpha \geq 1, \quad \textit{dimensional free}, \\ (L_0 R)^\alpha d^{1-\alpha}, & \alpha < 1 \quad \textit{improve by a } \Theta\left(d^\alpha\right) \textit{factor}. \end{cases} \tag{153}$$

Proof First, we compute ED_1 as follows:

$$\begin{aligned} \mathrm{ED}_1 &= \sum_{i=1}^d \sigma_i(\nabla^2 f) \\ &= \left\| \sum_{i=1}^N \frac{1}{N} q''(\boldsymbol{\beta}_i^\top \mathbf{x}) \boldsymbol{\beta}_i \boldsymbol{\beta}_i^\top \right\|_* \\ &\leq \frac{L_0}{N} \left\| \sum_{i=1}^N \boldsymbol{\beta}_i \boldsymbol{\beta}_i^\top \right\|_* \leq L_0 R. \end{aligned} \tag{154}$$

For $\alpha \geq 1$, by the convexity of $g(x) = x^\alpha$, we have

$$\sum_{i=1}^d \sigma_i^\alpha(\nabla^2 f(\mathbf{x})) \leq \left(\sum_{i=1}^d \sigma_i(\nabla^2 f(\mathbf{x}))\right)^\alpha. \tag{155}$$

For $\alpha < 1$, by Hölder's inequality, we have

$$\left(\sum_{i=1}^d \sigma_i^\alpha(\nabla^2 f(\mathbf{x}))\right) \leq \left(\sum_{i=1}^d \sigma_i(\nabla^2 f(\mathbf{x}))\right)^\alpha \cdot d^{1-\alpha}. \tag{156}$$

Taking supremum to both sides of (155) and (156) on $\mathbf{x}$ yields the result. ■

Proposition 7 *Define $f(\mathbf{W}, \mathbf{w}) = \mathbf{w}^\top \sigma(\mathbf{W}^\top \mathbf{x})$, where σ is the activation function. When $\|\mathbf{x}\|_1 \leq r_1$, $\|\mathbf{w}\| \leq r_2$ and $\sigma''(x) \leq \alpha$, we have $\operatorname{tr}\left(\nabla^2 f(\mathbf{W}, \mathbf{w})\right) \leq \alpha r_1 r_2$.*

Proof [Proof of Proposition 7] By direct computation, we have

$$\begin{aligned}
\frac{\partial f}{\partial \mathbf{w}} &= \sigma(\mathbf{W}^\top \mathbf{x}), \\
\frac{\partial f}{\partial \mathbf{W}} &= \left(\sigma'(\mathbf{W}^\top \mathbf{x}) \odot \mathbf{w}\right) \otimes \mathbf{x}, \\
\frac{\partial^2 f}{\partial \mathbf{w}^2} &= \mathbf{0}, \\
\frac{\partial^2 f}{\partial \mathbf{W}^2} &= \operatorname{Diag}(\sigma''(\mathbf{W}^\top \mathbf{x}) \odot \mathbf{w}) \otimes \mathbf{x} \otimes \mathbf{x}.
\end{aligned} \tag{157}$$

Therefore,

$$\begin{aligned}
\operatorname{tr}\left(\nabla^2 f(\mathbf{W}, \mathbf{w})\right)) &= \|\mathbf{x}\|^2 \cdot \operatorname{tr}\left(\operatorname{Diag}(\sigma''(\mathbf{W}^\top \mathbf{x}) \odot \mathbf{w})\right) \\
&\leq r_1^2 \cdot \langle \sigma''(\mathbf{W}^\top \mathbf{x}), \mathbf{x} \rangle \\
&\leq \alpha r_1 r_2.
\end{aligned} \tag{158}$$

■

Lemma 18 *For function f that has L-continuous gradient and M-continuous Hessian matrices, given any $\delta > 0$, Algorithm 3 outputs a δ-approximated $f_{\mathbf{x}}(\mathbf{y})$ such that $\left|\tilde{f}_{\mathbf{x},\delta}(\mathbf{y}) - f_{\mathbf{x}}(\mathbf{y})\right| \leq \delta$.*

Proof [Proof of Lemma 18] We only need to prove that $|\tilde{f}_{\mathbf{x},\delta}(\mathbf{y}) - f_{\mathbf{x}}(\mathbf{y})| \leq \delta$. We have the following inequality:

$$\begin{aligned}
&\left|\frac{Lr^2}{\delta}\left(f\left(\mathbf{x} + \frac{\delta}{Lr^2}(\mathbf{y} - \mathbf{x})\right) - f(\mathbf{x})\right) - \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle\right| \\
&= \frac{Lr^2}{\delta}\left|f\left(\mathbf{x} + \frac{\delta}{Lr^2}(\mathbf{y} - \mathbf{x})\right) - \left(f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \frac{\delta}{Lr^2}(\mathbf{y} - \mathbf{x}) \rangle\right)\right| \\
&\overset{a}{\leq} \frac{Lr^2}{\delta} \cdot \frac{L}{2}\left\|\frac{\delta}{Lr^2}(\mathbf{y} - \mathbf{x})\right\|^2 \\
&= \frac{Lr^2}{\delta} \cdot \frac{L}{2}\left(\frac{\delta}{Lr}\right)^2 = \frac{\delta}{2},
\end{aligned} \tag{159}$$

where $\overset{a}{\leq}$ uses the L-Lipschitz continuity of $\nabla f(\mathbf{x})$. We also have

$$\begin{aligned}
&\left|\frac{2H^2 r^6}{\delta^2}\left(f\left(\mathbf{x} + \frac{\delta}{2Hr^3}(\mathbf{y} - \mathbf{x})\right) + f\left(\mathbf{x} - \frac{\delta^2}{2Hr^3}(\mathbf{y} - \mathbf{x})\right) - 2f(\mathbf{x})\right) - \langle \nabla^2 f(\mathbf{x})(\mathbf{y} - \mathbf{x}), \mathbf{y} - \mathbf{x} \rangle\right| \\
&\overset{a}{=} \left|\frac{1}{2}(\langle \nabla^2 f(\mathbf{x}_1)(\mathbf{y} - \mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \langle \nabla^2 f(\mathbf{x}_2)(\mathbf{y} - \mathbf{x}), \mathbf{y} - \mathbf{x} \rangle) - \langle \nabla^2 f(\mathbf{x})(\mathbf{y} - \mathbf{x}), \mathbf{y} - \mathbf{x} \rangle\right|,
\end{aligned} \tag{160}$$

where $\stackrel{a}{=}$ uses the second-order Taylor expansion of f at $\mathbf{x}$. We have $\|\mathbf{x}_1 - \mathbf{x}\| \leq \frac{\delta}{2Hr^2}$ and $\|\mathbf{x}_2 - \mathbf{x}\| \leq \frac{\delta}{2Hr^2}$. By the H-Lipschitz continuity of $\nabla^2 f(\mathbf{x})$, we have

$$\begin{aligned}&\left|\frac{1}{2}(\langle\nabla^2 f(\mathbf{x}_1)(\mathbf{y}-\mathbf{x}),\mathbf{y}-\mathbf{x}\rangle + \langle\nabla^2 f(\mathbf{x}_2)(\mathbf{y}-\mathbf{x}),\mathbf{y}-\mathbf{x}\rangle) - \langle\nabla^2 f(\mathbf{x})(\mathbf{y}-\mathbf{x}),\mathbf{y}-\mathbf{x}\rangle\right| \\ &\leq \frac{\delta}{2Hr^2}\cdot Hr^2 = \frac{\delta}{2}. \end{aligned} \tag{161}$$

By (159), (160), and (161) we have $|\tilde{f}_{\mathbf{x},\delta}(\mathbf{y}) - f_{\mathbf{x}}(\mathbf{y})| \leq \delta$, hence the lemma is proved. ■

Proof [Proof of Corollary 11] By (154) and Theorem 10, we know that Algorithm 1 needs $\tilde{\mathcal{O}}\left(\frac{L_0 R}{\mu}\right)$ to find an ϵ-approximated solution with high probability. ■

Proof [Proof of Corollary 13] By (154) and Theorem 12, we know that Algorithm 1 needs $\tilde{\mathcal{O}}\left(\frac{L_0 R}{\epsilon}\right)$ to find an ϵ-approximated solution with high probability. ■