OPTIMAL CONVEX OPTIMIZATION WITH INEXACT SECOND-ORDER ORACLES

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we present a novel second-order method called Accelerated Inexact Newton Extragradient (AINE) for convex optimization using δ-inexact Hessians. We show that AINE can find an ϵ-solution in the inexact second-order oracle (ISO) complexity of $\mathcal{O}((\delta/\epsilon)^{1/2} + (L_2/\epsilon)^{2/7})$ when the Hessian is L_2-Lipschitz continuous, and a better complexity of $\tilde{\mathcal{O}}((\delta/\epsilon)^{1/2} + (L_3/\epsilon)^{1/5})$ when the third-order derivative is L_3-Lipschitz continuous. Notably, each iteration of our method can be conducted in the same running time as matrix multiplication up to logarithmic factors. In addition, we also establish complexity lower bounds to demonstrate that our methods are (nearly) optimal.

1 INTRODUCTION

We consider the following optimization problem:

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

where the objective $f(\boldsymbol{x})$ is continuously differentiable. First-order methods for convex optimization have been extensively studied, and complete theories for them have been established. For the task of finding a ϵ-solution, *i.e.* a point $\hat{\boldsymbol{x}} \in \mathbb{R}^d$ that satisfies $f(\hat{\boldsymbol{x}}) - \min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}) \leq \epsilon$, Nemirovskij & Yudin (1983) established an $\Omega((L_1/\epsilon)^{1/2})$ lower bound on the first-order oracle (FO) complexity for optimizing functions with L_1-Lipschitz continuous gradients. Shortly after, Nesterov (1983) proposed the accelerated gradient descent (AGD) method and showed that it achieved the optimal FO complexity of $\mathcal{O}((L_1/\epsilon)^{1/2})$.

Over the past two decades, there has been a growing interest in studying globally convergent second-order methods. In the seminal work, Nesterov & Polyak (2006) introduced the cubic regularized Newton (CRN) method and showed that it achieves the second-order oracle (SO) complexity of $\mathcal{O}((L_2/\epsilon)^{1/2})$ when the objective has L_2-Lipschitz continuous Hessians. Later, Nesterov (2008) proposed an accelerated CRN method to achieve a better SO complexity of $\mathcal{O}((L_2/\epsilon)^{1/3})$. Monteiro & Svaiter (2013) proposed a novel accelerated Newton proximal extragradient (A-NPE) method that can achieve the SO complexity of $\tilde{\mathcal{O}}((L_2/\epsilon)^{2/7})$. When the objective has L_3-Lipschitz continuous third-order derivatives, Nesterov (2021a) proposed a superfast second-order method that achieves the SO complexity of $\tilde{\mathcal{O}}((L_3/\epsilon)^{1/4})$. Then, three independent groups (Gasnikov et al., 2019; Bubeck et al., 2019; Jiang et al., 2021) obtained a lower SO complexity of $\tilde{\mathcal{O}}((L_3/\epsilon)^{1/5})$ by extending the A-NPE method. Very recently, two independent groups (Kovalev & Gasnikov, 2022; Carmon et al., 2022) proposed novel techniques to reduce the logarithmic factors, achieving the optimal SO complexity of $\mathcal{O}(\min\{(L_2/\epsilon)^{2/7}, (L_3/\epsilon)^{1/5}\log(L_3/\epsilon)\})$ that (nearly) matches the oracle lower bounds for both settings (Arjevani et al., 2019).

Although optimal SO complexities have been established for second-order methods, implementing these methods with exact Hessian oracles could be expensive. Therefore, one may use Hessian approximations rather than the exact ones. Ghadimi et al. (2017) proposed the inexact Newton with cubic regularization (INCR) and its acceleration (AINCR) that achieves the inexact second-order oracle (ISO) complexity of $\mathcal{O}(\delta/\epsilon + (L_2/\epsilon)^{1/2})$ and $\mathcal{O}((\delta/\epsilon)^{1/2} + (L_2/\epsilon)^{1/3})$, respectively. Subsequently, Agafonov et al. (2024a;b) proposed the accelerated inexact tensor method (AITM) that further requires a δ'-inexact third-order derivative information and achieves the inexact third-order oracle complexity of $\tilde{\mathcal{O}}((\delta/\epsilon)^{1/2} + (\delta'/\epsilon)^{1/3} + (L_3/\epsilon)^{1/4})$. However, we observe several

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: Comparison of the inexact second-order oracle (ISO) complexity and per iteration costs of second-order methods with δ-inexact Hessians to find an ϵ-solution of for a convex function with L_2-Lipschitz Hessians **(top)** and with L_3-Lipschitz continuous third-order derivatives **(bottom)**.

Method	ISO Complexity	Per Iter. Cost [(a)]
INCR (Ghadimi et al., 2017)	$\mathcal{O}\left(\delta/\epsilon + (L_2/\epsilon)^{1/2}\right)$	$\mathcal{O}(d^\omega + Cd)$
AINCR (Ghadimi et al., 2017)	$\mathcal{O}\left((\delta/\epsilon)^{1/2} + (L_2/\epsilon)^{1/3}\right)$	$\mathcal{O}(d^\omega + Cd)$
AINE (This Paper)	$\mathcal{O}\left((\delta/\epsilon)^{1/2} + (L_2/\epsilon)^{2/7}\right)$	$\mathcal{O}(d^\omega + Cd)$
Lower Bound (Agafonov et al., 2024b)	$\Omega\left((\delta/\epsilon)^{1/2} + (L_2/\epsilon)^{2/7}\right)$	-
AITM (Agafonov et al., 2024a;b)	$\tilde{\mathcal{O}}\left((\delta/\epsilon)^{1/2} + (\delta'/\epsilon)^{1/3} + (L_3/\epsilon)^{1/4}\right)$ [(b)]	$\tilde{\mathcal{O}}(d^\omega + Cd)$ [(c)]
AINE (This Paper)	$\tilde{\mathcal{O}}\left((\delta/\epsilon)^{1/2} + (L_3/\epsilon)^{1/5}\right)$	$\tilde{\mathcal{O}}(d^\omega + Cd)$
Lower Bound (This Paper)	$\Omega\left((\delta/\epsilon)^{1/2} + (L_3/\epsilon)^{1/5}\right)$	-

(a) We let $d^\omega \approx d^{2.37}$ to denote the running time of multiplying two $d \times d$ matrices (Duan et al., 2023; Alman et al., 2025), and we follow the convention (Nesterov & Polyak, 2006; Nesterov, 2021a;b;c) to assume the cost of solving the one-dimensional convex sub-problem is a constant C.
(b) Agafonov et al. (2024a) requires querying δ'-inexact third-order oracle.
(c) Agafonov et al. (2024a) did not provide a rigorous analysis of the per iteration costs, but it can be obtained as we discuss in Remark 6.1.

limitations in these works: First, these inexact methods do not achieve optimal complexity in the exact setting when $\delta = 0$ (Arjevani et al., 2019). Second, the AITM method (Agafonov et al., 2024a;b) requires querying δ'-inexact third-order derivatives, and its convergence rate depends on an extra parameter δ'.

In this paper, we propose a novel inexact second-order methods called Accelerated Inexact Newton Extragradient (AINE) that generalizes the optimal exact methods (Carmon et al., 2022; Kovalev & Gasnikov, 2022) and addresses both issues above. We show that our method achieves an improved ISO complexity of $\mathcal{O}((\delta/\epsilon)^{1/2} + (L_2/\epsilon)^{2/7})$ for functions with L_2-Lipschitz continuous Hessians, and that of $\tilde{\mathcal{O}}((\delta/\epsilon)^{1/2} + (L_3/\epsilon)^{1/5})$ for functions with L_3-Lipschitz continuous third-order derivatives. Moreover, we extend the techniques by Nesterov (2021a;b;c) to propose an efficient second-order subroutine that implements the required third-order tensor step in the latter case in almost matrix multiplication time, which also removes the δ' dependency since the inexact third-order derivatives are not used at all. Finally, we also provide (nearly) matching oracle lower bounds to demonstrate the optimality of our methods. We summarize our results in Table 1 and introduce additional related works in Appendix A.

2 Preliminaries

Notations. Throughout this paper, we use $\|\cdot\|$ to denote both the Euclidean norm of a vector and the operator norm of a matrix/tensor. We let $\mathbb{B}_D^d(\boldsymbol{c})$ denote the ball $\{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x} - \boldsymbol{c}\| \le D\}$. For any set S and functions $g, h : S \to [0, \infty)$ we write $g = \mathcal{O}(h)$ or $h = \Omega(g)$ equivalently if there exists $c > 0$ such that $g(s) \le ch(s)$ for every $s \in S$. We also write $h = \Theta(g)$ if $h = \mathcal{O}(g)$ and $h = \Omega(g)$, and write $\tilde{\mathcal{O}}(\cdot)$ to hide the logarithmic factors in $\mathcal{O}(\cdot)$.

2.1 Assumptions

Throughout this paper, we assume that the objective function $f : \mathbb{R}^d \to \mathbb{R}$ is convex and has Lipschitz continuous pth-order derivatives.

Assumption 2.1. *We assume that $f : \mathbb{R}^d \to \mathbb{R}$ is convex and has at least one solution $\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x})$.*

Assumption 2.2. *We assume that the function $f : \mathbb{R}^d \to \mathbb{R}$ is p times continuous differentiable and has L_p-Lipschitz continuous pth-order derivatives, i.e.,*

$$\|\nabla^p f(\boldsymbol{x}) - \nabla^p f(\boldsymbol{y})\| \le L_p \|\boldsymbol{x} - \boldsymbol{y}\|, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d,$$

where $L_p > 0$ is the Lipschitz constant of $\nabla^p f$.

In this paper, we are mostly interested in the case that $p \in \{1, 2, 3\}$, for which we can develop globally convergent second-order algorithms (Nesterov, 2021a). For the cases $p \ge 4$, although there are conceptual tensor algorithms that could be applied (Birgin et al., 2017; Nesterov, 2021a), it is still open whether the tensor steps in these algorithms can be solved efficiently. Therefore, this work only considers the currently implementable setting that $p \in \{1, 2, 3\}$. Following Ghadimi et al. (2017); Agafonov et al. (2024a;b), we assume that the algorithms have access to a δ-Hessian estimator $\boldsymbol{H}(\boldsymbol{x})$, formally defined as follows.

Assumption 2.3. *We assume that the algorithm has access to a δ-Hessian estimator $\boldsymbol{H}(\boldsymbol{x}) \in \mathbb{R}^{d\times d}$ satisfies that for any $\boldsymbol{x} \in \mathbb{R}^d$, we have*

$$\boldsymbol{H}(\boldsymbol{x}) = \boldsymbol{H}^\top(\boldsymbol{x}) \quad \text{and} \quad \|\boldsymbol{H}(\boldsymbol{x}) - \nabla^2 f(\boldsymbol{x})\| \le \delta.$$

This assumption unifies both first- and second-order methods, which recovers the second-order optimization with exact oracles (Nesterov & Polyak, 2006), and first-order optimization when the objective has L_1-Lipschitz continuous gradients (Nesterov, 1983) if $\delta = 0$. Under the above assumptions, we aim to find an approximate solution that has a small suboptimality gap:

Definition 2.1. *We say $\hat{\boldsymbol{x}} \in \mathbb{R}^d$ an ϵ-solution to Problem (1) if $f(\hat{\boldsymbol{x}}) - \min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}) \le \epsilon$.*

3 Accelerated Inexact Newton Extragradient

Ghadimi et al. (2017); Agafonov et al. (2024a;c) proposed inexact second- or higher-order methods by generalizing the Nesterov acceleration (Nesterov, 2008). However, in the exact case, the Nesterov acceleration has known suboptimal for high-order optimization since the ϵ dependence can be improved via the A-NPE framework (Monteiro & Svaiter, 2013). Therefore, it is natural to study applying the same technique in the exact setting.

3.1 Optimal Exact Second-Order Methods

We first review the accelerated Newton proximal extragradient (A-NPE) framework (Monteiro & Svaiter, 2013) in Algorithm 1, a powerful tool to achieve the optimal convergence rate for second-order methods. The most subtle part in Algorithm 1 is the proximal point step in Eq. (2), which requires solving an implicit and cyclical nonlinear equation in $(\boldsymbol{y}_t, \lambda_t)$. Below, we briefly recall the solutions to solve this implicit equation in existing literature.

Monteiro & Svaiter (2013) showed that this equation can be solved in $\mathcal{O}(\log(L_2/\epsilon))$ iterations by a bisection in λ_t with Newton updates $\boldsymbol{y}_t = \boldsymbol{z}_t - (\nabla^2 f(\boldsymbol{z}_t) + \lambda_t \boldsymbol{I}_d)^{-1}\nabla f(\boldsymbol{z}_t)$, leading to a near-optimal second-order method with an SO complexity of $\mathcal{O}((L_2/\epsilon)^{2/7}\log(L_2/\epsilon))$. A similar bisection strategy is also adopted in the tensor generaliza ons (Gasnikov et al., 2019; Bubeck et al., 2019; Jiang et al., 2021) that can achieve a faster rate of $\mathcal{O}((L_3/\epsilon)^{1/5}\log(L_3/\epsilon))$ under the additional L_3-smoothness assumption. However, the bisection could be costly in practice and introduces the unnecessary additional logarithmic factors in theory.

Carmon et al. (2022) removed the bisection using the following strategy. Their method maintains a guess λ_t' to calculate $\boldsymbol{z}_t$ and then obtains $\boldsymbol{y}_t$ by taking an exact tensor step from $\boldsymbol{z}_t$. After $\boldsymbol{y}_t$ is computed, it computes $\lambda_t = \Theta(L_p\|\boldsymbol{y}_t - \boldsymbol{z}_t\|^{p-1})$ and compares λ_t with the guess λ_t'. If the guess overestimates ($\lambda_t' \ge \lambda_t$), the algorithm simply performs the same updates as the original A-NPE method (Monteiro & Svaiter, 2013). If it underestimates ($\lambda_t' < \lambda_t$), the algorithm scales down the growth of A_{t+1} by a factor of λ_t'/λ_t and replaces the next iterate $\boldsymbol{x}_{t+1}$ by a convex combination of $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$. Carmon et al. (2022) applies a multiplicative update for the guess λ_t'. In a concurrent work, Kovalev & Gasnikov (2022) obtained the same optimal rate by predetermining the regularization λ_t and then solving the proximal subproblem in Eq. (2) with another Newton extragradient subroutine.

Algorithm 1 A-NPE Framework($\boldsymbol{x}_0, T$)

1: $A_0 = 0$
2: **for** $t = 0, \cdots, T-1$
3: Compute $\boldsymbol{y}_t \in \mathbb{R}^d$ and $\lambda_t \in \mathbb{R}_+$ such that

$$\boldsymbol{y}_t \approx \arg\min_{\boldsymbol{y}\in\mathbb{R}^d} f(\boldsymbol{y}) + \lambda_t \|\boldsymbol{y} - \boldsymbol{z}_t\|^2/2, \tag{2}$$

where

$$\boldsymbol{z}_t = (A_t\boldsymbol{x}_t + a_{t+1}\boldsymbol{v}_t)/A_{t+1}, \quad A_{t+1} = A_t + a_{t+1},$$
$$\lambda_t a_{t+1}^2 = A_{t+1}, \text{ and } \lambda_t = \Theta(L_p\|\boldsymbol{y}_t - \boldsymbol{z}_t\|^{p-1}).$$

4: Update $\boldsymbol{x}_{t+1} = \boldsymbol{y}_t$ and $\boldsymbol{v}_{t+1} = \boldsymbol{v}_t - a_{t+1}\nabla f(\boldsymbol{y}_t)$.
5: **end for**
6: **return** $\boldsymbol{x}_T$

3.2 Optimal Inexact Second-Order Method

As we see above, one key step in implementing A-NPE is correctly choosing the regularization coefficient λ_t. The techniques used in Monteiro & Svaiter (2013); Carmon et al. (2022); Kovalev & Gasnikov (2022) are not sufficient to handle the case when the Hessian is inexact. The analyses in Monteiro & Svaiter (2013); Carmon et al. (2022) both require lower-bounding $\|\boldsymbol{y}_t - \boldsymbol{z}_t\|^2$ using λ_t, which is unfortunately violated by inexact second-order methods where $\lambda_t = \Omega(\delta)$. Moreover, while Kovalev & Gasnikov (2022) does not use the aforementioned inequality, it remains unclear how to set the regularization λ_t in advance, since it should decrease at a different rate depending on whether δ is larger than $L_p\|\boldsymbol{y}_t - \boldsymbol{z}_t\|^{p-1}/p!$ at each iteration.

We address these issues by leveraging an epoch-wise guessing strategy to stabilize the regularization schedule. Inspired by Adil et al. (2026), our approach partitions iterations into distinct epochs and adaptively assigns a predetermined λ_t' via a Guess subroutine. This finally leads to the accelerated inexact Newton extragradient (AINE) method as presented in Algorithm 2. The algorithm utilizes the following δ-MS oracle.

Definition 3.1. *We say the mapping $\mathbb{O}_p^\delta : \mathbb{R}^d \mapsto \mathbb{R}^d \times \mathbb{R}_+$ a pth-order δ-MS oracle if for an input point $\boldsymbol{x} \in \mathbb{R}^d$, it returns the a pair $(\boldsymbol{y}, \lambda) = \mathbb{O}_p^\delta(\boldsymbol{x})$ that satisfies:*

1. *The MS condition $\|\nabla f(\boldsymbol{y}) + \lambda(\boldsymbol{y} - \boldsymbol{x})\| \le \lambda\|\boldsymbol{y} - \boldsymbol{x}\|/2$.*
2. *The pth-order δ-inexact movement bound $c(\delta + L_p\|\boldsymbol{y} - \boldsymbol{x}\|^{p-1}/p!) \le \lambda \le C(\delta + L_p\|\boldsymbol{y} - \boldsymbol{x}\|^{p-1}/p!)$ for some constants $0 < c \le C$.*

The following lemma shows that the output of a δ-inexact CRN step is a δ-MS oracle when $p = 2$.

Lemma 3.1. *Under Assumption 2.2 and 2.3, the second-order δ-MS oracle $\mathbb{O}_2^\delta : \mathbb{R}^d \mapsto \mathbb{R}^d \times \mathbb{R}_+$ can be implemented by the following inexact cubic regularized Newton (CRN) step:*

$$\begin{aligned}\boldsymbol{y} =& \arg\min_{\boldsymbol{y}\in\mathbb{R}^d} \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle + \frac{1}{2}\langle (\boldsymbol{H}(\boldsymbol{x}) + 2\delta\boldsymbol{I}_d)(\boldsymbol{y} - \boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle + \frac{L_2}{3}\|\boldsymbol{y} - \boldsymbol{x}\|^3, \\ \lambda =& 2\delta + L_2\|\boldsymbol{y} - \boldsymbol{x}\|.\end{aligned} \tag{3}$$

Remark 3.1. *Lemma 3.1 shows that $\mathbb{O}_2^\delta$ can be achieved by an inexact CRN update for functions with L_2-Lipschitz continuous Hessians. In the following, we study the convergence of the AINE by assuming access to $\mathbb{O}_p^\delta$, which applies to a broader range of settings by varying the order of p. We will also show that the proposed framework can lead to a superfast inexact second-order method for functions with L_3-Lipschitz continuous third-order derivatives in the next section.*

Since the convergence result from Carmon et al. (2022) or (Adil et al., 2026) only requires the MS condition (Condition 1 in Definition 3.1), and is independent of the other components in Algorithm 2, we can obtain the following guarantee of the AINE method.

Algorithm 2 AINE($\boldsymbol{x}_0, T, \delta, L_p$)

Require: δ-inexact MS oracle $(\boldsymbol{y}, \lambda) = \mathbb{O}_p^\delta(\boldsymbol{z})$ for $p \in \{2, 3\}$.

function GUESS(λ_0, t, A_t)
  if $t = 0$ **then**
    return $\lambda'_t = \lambda_0$
  else if $t = 1$ **or** $A_t > 2A_s$ **then**
    Store $s = t$ and $A_s = A_t$.
    return λ'_t as Eq. (6)
  else
    return $\lambda'_t = \lambda'_s$
  end if
end function
$\boldsymbol{v}_0 = \boldsymbol{z}_0 = \boldsymbol{x}_0, \quad A_0 = 0$ ▷ Main procedure
$(\boldsymbol{y}_0, \lambda_0) = \mathbb{O}_p^\delta(\boldsymbol{z}_0)$
for $t = 0, \cdots, T-1$
  $\lambda'_t = \text{Guess}(\lambda_0, t, A_t)$
  $a'_{t+1} = (1 + \sqrt{1 + 4\lambda'_t A_t})/(2\lambda'_t)$
  $A'_{t+1} = A_t + a'_{t+1}, \quad \boldsymbol{z}_t = (A_t \boldsymbol{x}_t + a'_{t+1}\boldsymbol{v}_t)/A'_{t+1}$
  if $t > 0$ **then**
    $(\boldsymbol{y}_t, \lambda_t) = \mathbb{O}_p^\delta(\boldsymbol{z}_t)$
  end if
  $\beta_t = \min\{1, \lambda'_t/\lambda_t\}, \quad a_{t+1} = \beta_t a'_{t+1}, \quad A_{t+1} = A_t + a_{t+1}$
  $\hat{\boldsymbol{x}}_{t+1} = \big((1-\beta_t)A_t\boldsymbol{x}_t + \beta_t A'_{t+1}\boldsymbol{y}_t\big)/A_{t+1}$
  if $p \in \{1, 2\}$ **then**
    $\boldsymbol{x}_{t+1} = \hat{\boldsymbol{x}}_{t+1}$
  else ▷ Use the monotone variant of A-NPE (Carmon et al., 2022, Line 8, Algo. 4).
    $\boldsymbol{x}_{t+1} \leftarrow \arg\min_{\boldsymbol{x}}\{f(\boldsymbol{x}) : \boldsymbol{x} \in \{\boldsymbol{x}_0, \cdots, \boldsymbol{x}_t, \hat{\boldsymbol{x}}_{t+1}\}\}$
  end if
  $\boldsymbol{v}_{t+1} = \boldsymbol{v}_t - a_{t+1}\nabla f(\boldsymbol{y}_t)$
end for
return $\boldsymbol{x}_T$

Lemma 3.2 (Carmon et al. (2022, Proposition 1))**.** *Let Assumption 2.1 hold and define*

$$E_t := f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*) \;\text{ and }\; D_t := \frac{1}{2}\|\boldsymbol{v}_t - \boldsymbol{x}^*\|^2. \tag{4}$$

If in Algorithm 2, all the pair $(\boldsymbol{y}_t, \boldsymbol{z}_t, \lambda_t)$ for all $t = 0, \cdots, T-1$ satisfies the MS condition (condition 1 of Definition 3.1), then the iterates generated by the algorithm satisfy that

$$(A_t E_t + D_t) - (A_{t+1}E_{t+1} + D_{t+1}) \geq \frac{3}{8}A'_{t+1}\min\{\lambda_t, \lambda'_t\}\|\boldsymbol{y}_t - \boldsymbol{z}_t\|^2. \tag{5}$$

Based on the above lemma, we obtain the following convergence rate of AINE by appropriately selecting the guess in the subroutine.

Theorem 3.1. *Let Assumption 2.1, 2.2, and 2.3 hold for $p \in \{2, 3\}$. If we select λ'_s in the Guess subroutine of Algorithm 2 according to*

$$\lambda'_s = \Theta\left(\delta + \left(D_0 L_p^{\frac{2}{p-1}}/A_s^{\frac{3}{2}}\right)^{\frac{2(p-1)}{3p+1}}\right), \tag{6}$$

then Algorithm 2 has the following convergence rate:

$$A_T = \Omega\left(\min\left\{\frac{T^2}{\delta}, \frac{T^{(3p+1)/2}}{D_0^{(p-1)/2}L_p}\right\}\right) \quad \text{and} \quad E_T \leq \frac{D_0}{A_T} = \mathcal{O}\left(\frac{\delta D_0}{T^2} + \frac{L_p D_0^{(p+1)/2}}{T^{(3p+1)/2}}\right). \tag{7}$$

Below, we provide a proof sketch and demonstrate the difference between prior analyses for optimal exact methods (Carmon et al., 2022; Adil et al., 2026). The formal proof is deferred to the appendix.

Algorithm 3 Implement $\mathbb{O}_3^\delta(\boldsymbol{z}_t)$ via BDGM (Lu et al., 2018; Nesterov, 2021c)

1: $k = 0$, $\boldsymbol{y}_{t,0} = \boldsymbol{z}_t$
2: Define f_t as Eq. (10), ρ_t as Eq. (11), and R_t as (12).
3: **while** $(\boldsymbol{y}_{t,k}, \lambda_{t,k})$ does not satisfy the MS condition (part 1 of Definition 3.1)
4: $\quad \boldsymbol{y}_{t,k+1} = \arg\min_{\|\boldsymbol{y}-\boldsymbol{z}_t\| \le R_t} \{\langle \nabla f_t(\boldsymbol{y}_{t,k}), \boldsymbol{y} - \boldsymbol{y}_{t,k}\rangle + 2(1+1/\xi)\beta_\rho(\boldsymbol{y}_{t,k}, \boldsymbol{y})\}$
5: $\quad k \leftarrow k+1$
6: **end while**
7: **return** $(\boldsymbol{y}_t, \lambda_t) = (\boldsymbol{y}_{t,k}, \lambda_{t,k})$

Proof Sketch. Inequality (5) in Lemma 3.2 indicates that $E_T \le D_0/A_T$. Therefore, proving the convergence rate of the algorithm suffices to show the growth rate of A_T in inequality (7). Following the analysis in Adil et al. (2026), we can give a lower bound of A_t by deriving an upper bound on the number of iterations required for A_t to increase by a factor of 2. Let t_1 be given and let t_2 be the smallest value where $A_{t_2} > 2A_{t_1}$. The Guess subroutine fixes $\lambda_t' = \lambda_{t_1}'$ in the interval $t \in [t_1, t_2)$. Let $S^\downarrow$ and $S^\uparrow$ be the sets of iterations where $\lambda_t < \lambda_t'$ and its converse held respectively:

$$S^\downarrow := \{t \in [t_1, t_2) : \lambda_t < \lambda_t'\} \text{ and } S^\uparrow := [t_1, t_2) - S_j^\downarrow,$$

We can show the same upper bound $|S^\downarrow| = \mathcal{O}(\sqrt{A_{t_1}\lambda_{t_1}'})$ proved by Adil et al. (2026) still holds for our inexact algorithm. However, their analysis of upper-bounding $|S^\uparrow|$ fails under Hessian inexactness. Specifically, they summed-up inequality (5) to obtain

$$D_0 \ge \frac{3}{8} A_{t+1}' \lambda_t' \|\boldsymbol{y}_t - \boldsymbol{z}_t\|^2,$$

then lower-bound $\|y_t - z_t\|$ by λ_t. However, since $\lambda_t = \Theta(\delta + L_p\|y_t - z_t\|^{p-1}/p!)$ in our setting, it cannot lower-bound $\|y_t - z_t\|$ when δ is large, which requires a new analysis. To overcome this issue, we develop a novel separation trick to divide $S^\uparrow$ by $S_1^\uparrow$ and $S_2^\uparrow$ to differentiate the cases whether the δ-inexactness or pth-order progress dominates in each iteration:

$$S_\delta^\uparrow := \{t \in S^\uparrow : \delta > L_p\|\boldsymbol{y}_t - \boldsymbol{z}_t\|^{p-1}/p!\} \quad \text{and} \quad S_p^\uparrow := S^\uparrow - S_\delta^\uparrow.$$

We then provide different analyses for these two sets individually to obtain the bounds of their sizes:

$$|S_\delta^\uparrow| = \mathcal{O}\left(\delta\sqrt{\frac{A_{t_1}}{\lambda_{t_1}'}}\right) \quad \text{and} \quad |S_p^\uparrow| = \mathcal{O}\left(\frac{D_0 L_p^{2/(p-1)}}{A_{t_1}(\lambda_{t_1}')^{(p+1)/(p-1)}}\right).$$

Finally, we sum up the bounds of $|S^\downarrow|$, $|S_\delta^\uparrow|$, and $|S_p^\uparrow|$ to obtain the bound of $t_2 - t_1 + 1$, and select the guess λ_{t_1}' to minimize this upper bound in Eq. (6). □

Combining this theorem with Lemma 3.1, we can obtain the ISO complexity of AINE for $p = 2$.

Corollary 3.1 (AINE for $p = 2$)**.** *Let Assumption 2.1, 2.2, and 2.3 hold for any $p = 2$. By implementing the second-order δ-MS oracle via the inexact CRN step as Eq. (3), Algorithm 2 can find an ϵ-solution in the inexact second-order oracle (ISO) complexity of $\mathcal{O}\left((\delta/\epsilon)^{1/2} + (L_2/\epsilon)^{2/7}\right)$.*

4 Second-Order Implementation of the Third-Order Scheme

In this section, we show that the AINE method can be applied beyond Hessian-Lipschitz objectives (Assumption 2.2 with $p = 2$), resulting in a faster inexact second-order method under higher-order smoothness (Assumption 2.2 with $p = 3$). Our main idea is to generalize the superfast exact second-order method (Nesterov, 2021a;b;c) to inexact oracles. We follow (Nesterov, 2021a;b;c) to establish the relative non-degeneracy condition (Bauschke et al., 2017; Lu et al., 2018) of a third-order proximal function, then apply a Bregman distance gradient method (BDGM) subroutine (Lu et al., 2018; Nesterov, 2021a;c) to solve the proximal subproblem using inexact second-order oracles.

4.1 Bregman Distance Gradient Method

This section first introduces the necessary backgrounds on the relative non-degeneracy condition (Bauschke et al., 2017; Lu et al., 2018) and the Bregman distance gradient method (BDGM) subroutine (Lu et al., 2018; Nesterov, 2021a;c) for the following constrained convex minimization problem:

$$\min_{\boldsymbol{y}\in\mathcal{Y}} \varphi(\boldsymbol{y}).$$

Let $\beta_\varphi(\boldsymbol{y},\boldsymbol{y}') = \varphi(\boldsymbol{y}') - \varphi(\boldsymbol{y}) - \langle\nabla\varphi(\boldsymbol{y}), \boldsymbol{y}'-\boldsymbol{y}\rangle$ be the Bregman distance induced by φ. The relative non-degeneracy condition and the condition number are formally defined as follows.

Assumption 4.1. *Assume that $\varphi : \mathcal{Y} \to \mathbb{R}$ is non-degenerate with respect to a scaling function $\rho : \mathcal{Y} \to \mathbb{R}$:*

$$\mu_\rho(\varphi)\beta_\rho(\boldsymbol{y},\boldsymbol{y}') \le \beta_\varphi(\boldsymbol{y},\boldsymbol{y}') \le L_\rho(\varphi)\beta_\rho(\boldsymbol{y},\boldsymbol{y}'), \quad \forall \boldsymbol{y},\boldsymbol{y}' \in \mathcal{Y},$$

where $0 \le \mu_\rho(\varphi) \le L_\rho(\varphi)$. When both $\varphi(\,\cdot\,)$ and $\rho(\,\cdot\,)$ are twice differentiable, the assumption is equivalent to $\mu_\rho(\varphi)\nabla^2\rho(\boldsymbol{y}) \preceq \nabla^2\varphi(\boldsymbol{y}) \preceq L_\rho(\varphi)\nabla^2\rho(\boldsymbol{y})$ for all $\boldsymbol{y}\in\mathcal{Y}$.

Definition 4.1. *Under Assumption 4.1, we denote $\kappa_\rho(\varphi) \triangleq \mu_\rho(\varphi)/L_\rho(\varphi)$ as the condition number of the function $\varphi(\,\cdot\,)$ with respect to the scaling function $\rho(\,\cdot\,)$.*

When the functions satisfy Assumption 4.1 with $\mu_\rho(\varphi) > 0$, the BDGM updates guarantee a linear convergence rate to the minimizer, as stated in the following lemma.

Lemma 4.1 (Nesterov (2021c, Eq. 40))**.** *Under Assumption 4.1 with $\mu_\rho(\varphi) > 0$, iteratively applying the update*

$$\boldsymbol{y}_{k+1} = \arg\min_{\boldsymbol{y}\in\mathcal{Y}} \left\{\langle\nabla\varphi(\boldsymbol{y}_k), \boldsymbol{y}-\boldsymbol{y}_k\rangle + 2L_\rho(\varphi)\beta_\rho(\boldsymbol{y}_k,\boldsymbol{y})\right\}, \tag{8}$$

ensures a linear convergence to the minimizer of $\boldsymbol{y}^* = \arg\min_{\boldsymbol{y}\in\mathcal{Y}}\varphi(\boldsymbol{y})$ with the convergence rate:

$$\varphi(\boldsymbol{y}_K) - \varphi(\boldsymbol{y}^*) \le \frac{\mu_\rho(\varphi)\beta_\rho(\boldsymbol{y}_0,\boldsymbol{y}^*)}{(1-\kappa_\rho(\varphi)/2)^{-K}-1} \le 2L_\rho(\varphi)\left(1-\kappa_\rho(\varphi)/2\right)^K \beta_\rho(\boldsymbol{y}_0,\boldsymbol{y}^*), \quad \forall K\in\mathbb{N}_+.$$

4.2 Implementing the Third-Order Inexact MS Oracle

To implement the third-order inexact MS oracle $\mathbb{O}_3^\delta(\boldsymbol{z}_t)$, we first note that Condition 2 of Definition 3.1 can be automatically satisfied by setting

$$\lambda_t = \frac{\xi^2-1}{\xi}\delta + \frac{(1+\xi)\xi L_3}{6}\|\boldsymbol{y}_t - \boldsymbol{z}_t\|^2, \tag{9}$$

where $\xi > 1$ is a numerical constant, and solving the proximal function:

$$\min_{\boldsymbol{y}\in\mathbb{R}^d}\left\{f_t(\boldsymbol{y}) \triangleq f(\boldsymbol{y}) + \frac{(\xi^2-1)\delta}{2\xi}\|\boldsymbol{y}-\boldsymbol{z}_t\|^2 + \frac{(1+\xi)\xi L_3}{24}\|\boldsymbol{y}-\boldsymbol{z}_t\|^4\right\}. \tag{10}$$

Consequently, if we can additionally satisfy (Condition 1 of Definition 3.1), then we can return a valid MS oracle, which can guarantee the fast convergence rate as a third-order method as indicated by Theorem 3.1. To fulfill Condition 1 efficiently, we leverage the following relative smoothness.

Lemma 4.2. *Under Assumption 2.1, 2.2, and 2.3 for $p = 3$, $f_t(\boldsymbol{y})$ is non-degenerate with respect to the scaling function:*

$$\rho_t(\boldsymbol{y}) = \frac{1}{2}\langle(\boldsymbol{H}(\boldsymbol{z}_t) + \xi\delta\boldsymbol{I}_d)(\boldsymbol{y}-\boldsymbol{z}_t), \boldsymbol{y}-\boldsymbol{z}_t\rangle + \frac{(1+\xi)\xi L_3}{12}\|\boldsymbol{y}-\boldsymbol{z}_t\|^4, \tag{11}$$

and we have $(\xi-1)/(2\xi)\nabla^2\rho_t(\boldsymbol{y}) \preceq \nabla^2 f_t(\boldsymbol{y}) \preceq (1+1/\xi)\nabla^2\rho_t(\boldsymbol{y})$.

Although implementing the MS oracle is stating as minimizing the unconstrained optimization $\min_{\boldsymbol{y}\in\mathbb{R}^d} f_t(\boldsymbol{y})$, it is more convenient to work with an equivalent constraint form on a compact set $\mathcal{Y}_t$ for the stability of the process. It is possible by the following lemma.

Lemma 4.3. *Under Assumption 2.1, 2.2, and 2.3 for $p = 3$, for $\boldsymbol{y}_t^* = \arg\min_{\boldsymbol{y}\in\mathbb{R}^d} f_t(\boldsymbol{y})$, we have*

$$\|\boldsymbol{z}_t - \boldsymbol{y}_t^*\| \le 2\left(\frac{12\|\nabla f(\boldsymbol{z}_t)\|}{L_3(\xi^2-1)}\right)^{1/3} \triangleq R_t. \tag{12}$$

In view of the above lemma, we can replace Problem (10) by the following constrained one:

$$\min_{\boldsymbol{y}\in\mathcal{Y}_t}(\boldsymbol{y}), \quad \text{where} \quad \mathcal{Y}_t \triangleq \{\boldsymbol{y}\in\mathbb{R}^d : \|\boldsymbol{y}-\boldsymbol{z}_t\|\le R_t\} \tag{13}$$

We present the BDGM subroutine applied on Problem (13) in Algorithm 3. This subroutine terminates when $(\boldsymbol{y}_k,\lambda_k)$ satisfies the MS condition and then returns $(\boldsymbol{y}_t,\lambda_t)=(\boldsymbol{y}_k,\lambda_k)$ as the third-order δ-MS oracle for the query point $\boldsymbol{z}_t$. And we will present a global complexity bound for each call of the BDGM subroutine in Algorithm 2 in the next section.

4.3 Global Complexity Bound

Let $R(\boldsymbol{x}_0)=\sup_{\boldsymbol{y}\in\mathbb{R}^d}\{\|\boldsymbol{y}-\boldsymbol{x}^*\| : f(\boldsymbol{y})\le f(\boldsymbol{x}_0)\}$ be the size of the sub-level set, and assume $R(\boldsymbol{x}_0)<\infty$ throughout this subsection. The monotone MS acceleration variant we use in Algorithm 2 for $p=3$ ensures that all the generated iterates lie in the sub-level set, and then we can use the following lemma to obtain global upper bounds for their gradient and Hessian norms.

Lemma 4.4 (Nesterov (2021c, Lemma 5.1))**.** *If Assumption 2.2 holds for $p=3$, then for any $\boldsymbol{x}\in\{\boldsymbol{x}\in\mathbb{R}^d : \|\boldsymbol{x}-\boldsymbol{x}_0\|\le D\}$, we have*

$$\begin{aligned}\|\nabla f(\boldsymbol{x})\| &\le \|\nabla f(\boldsymbol{x}_0)\| + \|\nabla^2 f(\boldsymbol{x}_0)\|D + \frac{1}{2}\|\nabla^3 f(\boldsymbol{x}_0)\|D^2 + \frac{1}{6}L_3D^3,\\ \|\nabla^2 f(\boldsymbol{x})\| &\le \|\nabla^2 f(\boldsymbol{x}_0)\| + \|\nabla^3 f(\boldsymbol{x}_0)\|D + \frac{1}{2}L_3D^2.\end{aligned} \tag{14}$$

Since the size of the constraint set R_t defined in the BDGM subroutine depends on $\|\nabla f(\boldsymbol{z}_t)\|$, which is uniformly bounded by the above lemma, we can conclude the boundedness of all the iterates of our final algorithm for $p=3$.

Lemma 4.5. *Under Assumption 2.1 and 2.2 for $p=3$ and $R(\boldsymbol{x}_0)<\infty$, there exists a constant*

$$\bar{R}=\mathrm{poly}(L_3,L_3^{-1},R(\boldsymbol{x}_0),\|\nabla f(\boldsymbol{x}_0)\|,\|\nabla^2 f(\boldsymbol{x}_0)\|,\|\nabla^3 f(\boldsymbol{x}_0)\|)$$

such that all the iterates generated by the main procedure in Algorithm 2 and the BDGM subroutine (Algorithm 3) lie in the compact set $\mathcal{X}=\{\boldsymbol{x}\in\mathbb{R}^d : \|\boldsymbol{x}-\boldsymbol{x}^*\|\le\bar{R}\}$ when implementing the third-order MS oracle $\mathbb{O}_3^\delta$

Let $\mathcal{X}\subseteq\mathbb{R}^d$ be the set defined in Lemma 4.5, we further define

$$\bar{L}_0=\sup_{t\in[T],\boldsymbol{y}\in\mathcal{X}}\|\nabla f_t(\boldsymbol{y})\| \quad\text{and}\quad \bar{L}_1=\sup_{t\in[T],\boldsymbol{y}\in\mathcal{X}}\|\nabla^2 f_t(\boldsymbol{y})\|. \tag{15}$$

Then, we show that the relative error of the MS condition (Condition 1 of Definition 3.1) can be replaced by $\mathcal{O}(\delta\|\boldsymbol{z}_t-\boldsymbol{y}_t^*\|)$. This relaxed condition is easier to analyze in a subsequent analysis.

Lemma 4.6. *Let Assumption 2.2, 2.3, and 2.1 hold for $p=3$. For f_t and λ_t defined in Eqs. (10) and (9), the MS condition holds whenever*

$$\|\nabla f_t(\boldsymbol{y}_t)\|\le\max\left\{\frac{(\xi^2-1)\delta}{3\xi}\|\boldsymbol{z}_t-\boldsymbol{y}_t^*\|,\frac{(1+\xi)\xi L_3}{384}\|\boldsymbol{z}_t-\boldsymbol{y}_t^*\|^3\right\}. \tag{16}$$

Note that we can assume $\|\nabla f(\boldsymbol{z}_t)\|\ge\epsilon/R(\boldsymbol{x}_0)$ without loss of generality, since otherwise we can conclude from the convexity that $\boldsymbol{z}_t$ is already an ϵ-solution. By truncating the iterates to all these non-stationary ones, we can show that the right-hand side of the relaxed condition (16) is lower-bounded and finally arrive at the following key theorem.

Theorem 4.1. *Let Assumption 2.2, 2.3, and 2.1 hold for $p=3$ and $R(\boldsymbol{x}_0)<\infty$. If $\|\nabla f(\boldsymbol{z}_t)\|\ge\epsilon/R(\boldsymbol{x}_0)$, then, for any fixed $\xi>1$, the BDGM subroutine (Algorithm 3) terminates after at most K iterations, where*

$$K=\mathcal{O}\left(\log\left(2+\epsilon^{-1}+\delta+L_3+L_3^{-1}+R(\boldsymbol{x}_0)+\sum_{i=1}^{3}\|\nabla^i f(\boldsymbol{x}_0)\|\right)\right). \tag{17}$$

Corollary 4.1 (AINE for $p=3$)**.** *Let Assumption 2.2, 2.3, and 2.1 hold for $p=3$ and $R(\boldsymbol{x}_0)<\infty$. By implementing the third-order δ-MS oracle via Algorithm 3, Algorithm 2 finds an ϵ-solution using $\mathcal{O}((\delta/\epsilon)^{1/2}+(L_3/\epsilon)^{1/5})$ Hessian estimates and $\tilde{\mathcal{O}}((\delta/\epsilon)^{1/2}+(L_3/\epsilon)^{1/5})$ gradient evaluations; each inner call takes at most K iterations as in Eq. (17).*

5 Lower Complexity Bound

We next show that the upper bounds established by Theorem 3.1 are indeed optimal up to logarithmic factors by proving (nearly) matching lower bounds for the following algorithm class.

Definition 5.1. *Let $\mathbb{O}^{\mathrm{ISO}}(\boldsymbol{x}) = (f(\boldsymbol{x}), \nabla f(\boldsymbol{x}), \boldsymbol{H}(\boldsymbol{x}))$ be the inexact second-order oracle (ISO). A deterministic ISO algorithm $\mathtt{A}$ consists of a sequence of measurable mappings $\{\mathtt{A}_t\}_{t\in\mathbb{N}}$ such that $\mathtt{A}_t$ takes in the first $t-1$ oracle responses to produce the tth query, i.e., $\mathtt{A}$ generates a sequence $\{\boldsymbol{x}_t\}_{t\in\mathbb{N}}$ such that*

$$\boldsymbol{x}_t = \mathtt{A}_t(\mathbb{O}^{\mathrm{ISO}}(\boldsymbol{x}_0), \mathbb{O}^{\mathrm{ISO}}(\boldsymbol{x}_1), \cdots, \mathbb{O}^{\mathrm{ISO}}(\boldsymbol{x}_{t-1})),$$

where $\boldsymbol{H}(\boldsymbol{x})$ is the inexact Hessian at the query point $\boldsymbol{x}$ and satisfies Assumption 2.3.

For the above algorithm class, we establish lower bounds by using the results in Arjevani et al. (2019), where the $\Omega(L_p/T^{(3p+1)/2})$ lower bound comes from Arjevani et al. (2019, Theorem 3), and the $\Omega(\delta/T^2)$ lower bound comes from the same theorem with $p=1$ and $L_1=\delta$.

Proposition 5.1. *For every $p \in \{2,3\}$ and every deterministic algorithm in Definition 5.1, there exist a sufficiently large dimension d, a function $f:\mathbb{R}^d\to\mathbb{R}$, and an estimator $\boldsymbol{H}$ satisfying Assumptions 2.2, 2.1, and 2.3, such that*

$$f(\boldsymbol{x}_T) - f(\boldsymbol{x}^*) = \Omega\left(\delta D_0/T^2 + L_p D_0^{(p+1)/2}/T^{(3p+1)/2}\right).$$

Note that Agafonov et al. (2024b, Theorem 4.2) gives the corresponding $p=2$ result for linear-span algorithms. Proposition 5.1 extends their results bound to arbitrary deterministic algorithms and to $p=-3$. Thus, AINE is optimal for $p=2$ and near-optimal for $p=3$.

6 Solving the Auxiliary Subproblems in Nearly Matmul Time

In this section, we demonstrate that all the auxiliary subproblems in AINE for $p=2$ or $p=3$ can be solved in nearly matrix multiplication time (Alman et al., 2025; Duan et al., 2023). Recall that for $p=2$ we need to solve the inexact CRN step in Eq. (3) and for $p=3$ we need to conduct the BDGM update in Eq. (8). We can observe that they can be unified in the following form:

$$\min_{\|\boldsymbol{h}\|\le R} q(\boldsymbol{h}) \triangleq \langle \boldsymbol{b}, \boldsymbol{h}\rangle + \langle \boldsymbol{A}\boldsymbol{h}, \boldsymbol{h}\rangle/2 + c\|\boldsymbol{h}\|^{p+1}/(p+1). \tag{18}$$

This subproblem is closely related to the CRN subproblem (Nesterov & Polyak, 2006) and can be solved in a similar way. By leveraging the eigenvalue decomposition for matrix $\boldsymbol{A}$ and then transforming Problem (18) to a one-dimensional root-finding problem, we can show that the subproblem is almost as cheap as a Newton step.

Proposition 6.1. *Let C be the cost of solving the one-dimensional sub-problem and $\omega \approx 2.37$ be the matrix multiplication constant. For any $p \in \mathbb{N}_+$, subproblem (18) can be solved in running time of $\mathcal{O}(d^\omega + Cd)$.*

The above result not only applies to our method, but a similar idea can be used to characterize the running time of the auxiliary problem in Agafonov et al. (2024a), as remarked below.

Remark 6.1. *The auxiliary problem when $p=3$ in Agafonov et al. (2024a) takes the form of $\tilde{q}(\boldsymbol{h}) \triangleq q(\boldsymbol{h}) + \lambda\|\boldsymbol{h}\|^3$, which contains a third-order regularization term. However, one can follow the same idea as Proposition 6.1 to obtain the same running time.*

7 Conclusion and Future Works

In this paper, we propose a novel δ-inexact second-order method AINE for convex optimization. We show that the proposed method achieve an optimal ISO complexity of $\mathcal{O}((\delta/\epsilon)^{1/2} + (L_2/\epsilon)^{2/7})$ for functions with L_2-Lipschitz continuous Hessians and that of $\tilde{\mathcal{O}}((\delta/\epsilon)^{1/2} + (L_3/\epsilon)^{1/5})$ with L_3-Lipschitz third-order derivatives. Our method is (nearly) tight for both setups, and we hope that it can be a starting point to study many practical inexact/approximate second-order methods in the future. We discuss these potential future directions in Appendix B.

REFERENCES

Natalie Abreu, Nikhil Vyas, Sham Kakade, and Depen Morwani. The potential of second-order optimization for LLMs: A study with full Gauss-Newton. In *ICLR*, 2026.

Deeksha Adil, Brian Bullins, Aaron Sidford, and Chenyi Zhang. Balancing gradient and hessian queries in non-convex optimization. In *NeurIPS*, 2025.

Deeksha Adil, Brian Bullins, Arun Jambulapati, and Aaron Sidford. Convex optimization with p-norm oracles. In *ALT*, 2026.

Artem Agafonov, Dmitry Kamzolov, Pavel Dvurechensky, Alexander Gasnikov, and Martin Takáč. Inexact tensor methods and their application to stochastic convex optimization. *Optimization Methods and Software*, 39(1):42–83, 2024a.

Artem Agafonov, Dmitry Kamzolov, Alexander Gasnikov, Ali Kavis, Kimon Antonakopoulos, Volkan Cevher, and Martin Takáč. Advancing the lower bounds: an accelerated, stochastic, second-order method with optimal adaptation to inexactness. In *ICLR*, 2024b.

Artem Agafonov, Petr Ostroukhov, Roman Mozhaev, Konstantin Yakovlev, Eduard Gorbunov, Martin Takác, Alexander Gasnikov, and Dmitry Kamzolov. Exploring jacobian inexactness in second-order methods for variational inequalities: lower bounds, optimal algorithms and quasi-newton approximations. In *NeurIPS*, 2024c.

Josh Alman, Ran Duan, Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. More asymmetry yields faster matrix multiplication. In *SODA*, 2025.

Kimon Antonakopoulos, Ali Kavis, and Volkan Cevher. Extra-Newton: A first approach to noise-adaptive accelerated second-order methods. In *NeurIPS*, 2022.

Yossi Arjevani, Ohad Shamir, and Ron Shiff. Oracle complexity of second-order methods for smooth convex optimization. *Mathematical Programming*, 178(1):327–360, 2019.

Yossi Arjevani, Yair Carmon, John C. Duchi, Dylan J. Foster, Ayush Sekhari, and Karthik Sridharan. Second-order information in non-convex stochastic optimization: Power and limitations. In *COLT*, 2020.

Heinz H Bauschke, Jérôme Bolte, and Marc Teboulle. A descent lemma beyond lipschitz gradient continuity: first-order methods revisited and applications. *Mathematics of Operations Research*, 42(2):330–348, 2017.

Ernesto G. Birgin, J.L. Gardenghi, José Mario Martínez, Sandra Augusta Santos, and Ph L. Toint. Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. *Mathematical Programming*, 163(1):359–368, 2017.

Charles George Broyden. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, 6(1):76–90, 1970.

Sébastien Bubeck, Qijia Jiang, Yin Tat Lee, Yuanzhi Li, and Aaron Sidford. Near-optimal method for highly smooth convex optimization. In *COLT*, 2019.

Yair Carmon, Danielle Hausler, Arun Jambulapati, Yujia Jin, and Aaron Sidford. Optimal and adaptive Monteiro-Svaiter acceleration. In *NeurIPS*, 2022.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM transactions on intelligent systems and technology*, 2(3):1–27, 2011.

El Mahdi Chayti, Nikita Doikov, and Martin Jaggi. Unified convergence theory of stochastic and variance-reduced cubic newton methods. *TMLR*, 2024.

Andrew R. Conn, Nicholas I.M. Gould, and Ph. L. Toint. Convergence of quasi-Newton matrices generated by the symmetric rank one update. *Mathematical Programming*, 50(1):177–195, 1991.

Nikita Doikov, Martin Jaggi, et al. Second-order optimization with lazy hessians. In *ICML*, 2023.

Ran Duan, Hongxun Wu, and Renfei Zhou. Faster matrix multiplication via asymmetric hashing. In *FOCS*, 2023.

Nicolas Emmenegger, Rasmus Kyng, and Ahad N Zehmakan. On the oracle complexity of higher-order smooth non-convex finite-sum optimization. In *AISTATS*, 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 *COLT*, 2019.

Saeed Ghadimi, Han Liu, and Tong Zhang. Second-order methods with cubic regularization under inexact information. *arXiv preprint arXiv:1710.05782*, 2017.

Roger Grosse and James Martens. A kronecker-factored approximate fisher matrix for convolution layers. In *ICML*, 2016.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *ICML*, 2018.

Filip Hanzely, Nikita Doikov, Yurii Nesterov, and Peter Richtarik. Stochastic subspace cubic newton method. In *ICML*, 2020.

Bo Jiang, Haoyue Wang, and Shuzhong Zhang. An optimal high-order tensor method for convex optimization. *Mathematics of Operations Research*, 46(4):1390–1412, 2021.

Ruichen Jiang and Aryan Mokhtari. Accelerated quasi-newton proximal extragradient: Faster rate for smooth convex optimization. In *NeurIPS*, 2023.

Ruichen Jiang, Qiujiang Jin, and Aryan Mokhtari. Online learning guided curvature approximation: A quasi-newton method with global non-asymptotic superlinear convergence. In *COLT*, pp. 1962–1992, 2023.

Ruichen Jiang, Parameswaran Raman, Shoham Sabach, Aryan Mokhtari, Mingyi Hong, and Volkan Cevher. Krylov cubic regularized newton: A subspace second-order method with dimension-free convergence rate. In *AISTATS*, 2024.

Ruichen Jiang, Aryan Mokhtari, and Francisco Patitucci. Improved complexity for smooth nonconvex optimization: a two-level online learning approach with quasi-newton methods. In *STOC*, 2025.

Qiujiang Jin and Aryan Mokhtari. Non-asymptotic superlinear convergence of standard quasi-newton methods. *Mathematical Programming*, 200(1):425–473, 2023.

Dmitry Kovalev and Alexander Gasnikov. The first optimal acceleration of high-order methods in smooth convex optimization. In *NeurIPS*, 2022.

Hong Liu, Zhiyuan Li, David Leo Wright Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. In *ICLR*, 2024.

Haihao Lu, Robert M Freund, and Yurii Nesterov. Relatively smooth convex optimization by first-order methods, and applications. *SIAM Journal on Optimization*, 28(1):333–354, 2018.

Renato D.C. Monteiro and B. F. Svaiter. Iteration-complexity of a newton proximal extragradient method for monotone variational inequalities and inclusion problems. *SIAM Journal on Optimization*, 22(3):914–935, 2012.

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.

Arkadij Semenovič Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

Yurii Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. In *Dokl akad nauk Sssr*, volume 269, pp. 543, 1983.

Yurii Nesterov. Accelerating the cubic regularization of newton's method on convex problems. *Mathematical Programming*, 112(1):159–181, 2008.

Yurii Nesterov. Implementable tensor methods in unconstrained convex optimization. *Mathematical Programming*, 186(1):157–183, 2021a.

Yurii Nesterov. Inexact high-order proximal-point methods with auxiliary search procedure. *SIAM Journal on Optimization*, 31(4):2807–2828, 2021b.

Yurii Nesterov. Superfast second-order methods for unconstrained convex optimization. *Journal of Optimization Theory and Applications*, 191(1):1–30, 2021c.

Yurii Nesterov and Boris T. Polyak. Cubic regularization of newton method and its global performance. *Mathematical programming*, 108(1):177–205, 2006.

Anton Rodomanov and Yurii Nesterov. Rates of superlinear convergence for classical quasi-Newton methods. *Mathematical Programming*, pp. 1–32, 2022.

Nilesh Tripuraneni, Mitchell Stern, Chi Jin, Jeffrey Regier, and Michael I. Jordan. Stochastic cubic regularization for fast nonconvex optimization. In *NeurIPS*, 2018.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham M Kakade. SOAP: Improving and stabilizing Shampoo using Adam for language modeling. In *ICLR*, 2025.

Zhe Wang, Yi Zhou, Yingbin Liang, and Guanghui Lan. Stochastic variance-reduced cubic regularization for nonconvex optimization. In *AISTATS*, 2019.

Dongruo Zhou and Quanquan Gu. Stochastic recursive variance-reduced cubic regularization methods. In *AISTATS*, 2020.

Dongruo Zhou and Quanquan Gu. Dimension-free complexity bounds for high-order nonconvex finite-sum optimization. In *ICML*, 2022.

Dongruo Zhou, Pan Xu, and Quanquan Gu. Stochastic variance-reduced cubic regularization methods. *JMLR*, 20(134):1–47, 2019.

A Additional Related Works

Efficient Second-Order Methods. Second-order methods are known for fast convergence rates at a cost of higher computation overhead compared to first-order methods. It is a highly active research area for developing efficient second-order methods. Many studies focuses on different aspects from this work, including but no limited to: quasi-Newton methods (Broyden, 1970; Conn et al., 1991; Rodomanov & Nesterov, 2022; Jin & Mokhtari, 2023; Jiang & Mokhtari, 2023; Jiang et al., 2023; 2025), stochastic Newton methods (Zhou et al., 2019; Arjevani et al., 2020; Antonakopoulos et al., 2022), subspace Newton methods (Hanzely et al., 2020; Jiang et al., 2024), lazy Hessian updates (Doikov et al., 2023; Adil et al., 2025), and efficient implementations in neural networks (Grosse & Martens, 2016; Gupta et al., 2018; Liu et al., 2024; Vyas et al., 2025; Abreu et al., 2026). These works are complementary to ours with orthogonal techniques that can be jointly applied.

Accelerated Inexact Second-Order Methods. Although extra-gradient acceleration (Monteiro & Svaiter, 2012; 2013; Carmon et al., 2022; Kovalev & Gasnikov, 2022) is widely recognized as a standard technique to obtain optimal oracle complexities in exact second-order or higher-order optimization, it is challenging to generalize these optimal exact methods to the inexact setting. We briefly introduce the results in closely related works to demonstrate the technical challenges to balance the convergence in δ-inexact term and the exact one: Antonakopoulos et al. (2022) proposed a stochastic and adaptive second-order method Extra-Newton by combining Newton updates in extra-gradient methods, but their $\mathcal{O}((\delta/\epsilon)^{2/3} + (L_2/\epsilon)^{1/3})$ ISO complexity has a high dependency on the Hessian inexactness δ even in the deterministic case. Jiang & Mokhtari (2023) studied a quasi-Newton generalization of A-NPE, but directly applying their analysis to our setting can only lead to a suboptimal ISO complexity of $\mathcal{O}((\delta/\epsilon)^{1/2} + (L_2/\epsilon)^{2/5})$. Compared with them, this work aims at establishing much lower and optimal complexity bounds on both convergent terms by the new analysis in Section 3.2.

B Future Works

There are several future directions to extend our work. First, this work only considers a uniform inexactness δ in Hessian estimator. It is interesting to consider iteration-dependent inexactness δ_t and making our framework covers broader Hessian estimators, including quasi-Newton estimators (Broyden, 1970; Conn et al., 1991), lazy Hessian estimators (Doikov et al., 2023; Chayti et al., 2024; Adil et al., 2025), and variance-reduced estimators (Zhou et al., 2019; Zhou & Gu, 2020; 2022; Wang et al., 2019; Emmenegger et al., 2022). Second, it is also important to study stochastic (Antonakopoulos et al., 2022) and nonconvex problems (Nesterov & Polyak, 2006; Arjevani et al., 2020; Tripuraneni et al., 2018). Third, our methods may be extended to solve variational inequalities (Agafonov et al., 2024c) using δ-inexact Jacobians or minimax optimization.

C Proof of Theorem 3.1

Proof. Lemma 3.2 gives $E_t \le D_0/A_t$, so it remains to lower-bound A_T. Partition the iterations into epochs as in Algorithm 2: an epoch starts with weight $A_s > 0$ and ends when the weight exceeds $2A_s$. The guess λ'_s is fixed within the epoch.

If $\lambda_t < \lambda'_s$, then the damping is inactive and

$$\sqrt{A_{t+1}} - \sqrt{A_t} \ge \frac{1}{2\sqrt{\lambda'_s}},$$

so there are $\mathcal{O}(1+\sqrt{A_s\lambda'_s})$ such iterations before the weight doubles. Otherwise $\lambda_t \ge \lambda'_s$. Choosing the constant hidden in Eq. (6) large enough gives $\lambda'_s \ge 2C\delta$, where C is the upper-movement constant in Definition 3.1. Hence

$$\|\boldsymbol{y}_t - \boldsymbol{z}_t\| \ge \left(\frac{p!\lambda'_s}{2CL_p}\right)^{1/(p-1)}.$$

By Lemma 3.2, every such damped iteration decreases the potential by at least a constant multiple of

$$A_s L_p^{-2/(p-1)} (\lambda'_s)^{(p+1)/(p-1)}.$$

Since the total available potential is at most D_0, the number of damped iterations is

$$\mathcal{O}\left(\frac{D_0 L_p^{2/(p-1)}}{A_s(\lambda'_s)^{(p+1)/(p-1)}}\right).$$

Thus the epoch length is

$$\mathcal{O}\left(1+\sqrt{A_s\lambda'_s}+\frac{D_0 L_p^{2/(p-1)}}{A_s(\lambda'_s)^{(p+1)/(p-1)}}\right).$$

With the guess in Eq. (6), this becomes

$$\mathcal{O}\left(1+\sqrt{\delta A_s}+D_0^{(p-1)/(3p+1)}(L_p A_s)^{2/(3p+1)}\right).$$

The initialization uses $\lambda'_0 = \lambda_0$, hence $A_1 = 1/\lambda_0$. The first-step potential bound gives $\|\boldsymbol{y}_0 - \boldsymbol{x}_0\|^2 \le 8D_0/3$, and the upper-movement bound controls λ_0 by the original problem parameters. Therefore the additive epoch costs are absorbed by the displayed terms. Summing over the geometrically growing epoch weights yields

$$T = \mathcal{O}\left(\sqrt{\delta A_T}+D_0^{(p-1)/(3p+1)}(L_p A_T)^{2/(3p+1)}\right).$$

Inverting this inequality and using $E_T \le D_0/A_T$ proves Eq. (7). □

D Proof of Lemma 3.1

Proof. Recall that $\lambda = 2\delta + L_2\|\boldsymbol{y}-\boldsymbol{x}\|$. From the first-order optimality condition of Eq. (3), we have

$$\mathbf{0} = \nabla f(\boldsymbol{x}) + (\boldsymbol{H}(\boldsymbol{x}) + 2\delta \boldsymbol{I}_d)(\boldsymbol{y}-\boldsymbol{x}) + L_2\|\boldsymbol{y}-\boldsymbol{x}\|(\boldsymbol{y}-\boldsymbol{x}).$$

Rearranging this inequality gives

$$\begin{aligned}
&\nabla f(\boldsymbol{y}) + \lambda(\boldsymbol{y}-\boldsymbol{x}) \\
=&\nabla f(\boldsymbol{y}) - \nabla f(\boldsymbol{x}) - \boldsymbol{H}(\boldsymbol{x})(\boldsymbol{y}-\boldsymbol{x}) \\
=&\nabla f(\boldsymbol{y}) - \nabla f(\boldsymbol{x}) - \nabla^2 f(\boldsymbol{x})(\boldsymbol{y}-\boldsymbol{x}) + (\nabla^2 f(\boldsymbol{x}) - \boldsymbol{H}(\boldsymbol{x}))(\boldsymbol{y}-\boldsymbol{x}).
\end{aligned}$$

Hence, taking the norm on both sides and using the triangle inequality, we obtain

$$\begin{aligned}
&\|\nabla f(\boldsymbol{y}) + \lambda(\boldsymbol{y}-\boldsymbol{x})\| \\
\le&\|\nabla f(\boldsymbol{y}) - \nabla f(\boldsymbol{x}) - \nabla^2 f(\boldsymbol{x})(\boldsymbol{y}-\boldsymbol{x})\| + \|\nabla^2 f(\boldsymbol{x}) - \boldsymbol{H}(\boldsymbol{x})\|\|\boldsymbol{y}-\boldsymbol{x}\| \\
\le&\frac{L_2}{2}\|\boldsymbol{y}-\boldsymbol{x}\|^2 + \delta\|\boldsymbol{y}-\boldsymbol{x}\| = \frac{\lambda}{2}\|\boldsymbol{y}-\boldsymbol{x}\|,
\end{aligned}$$

where the last inequality uses Assumption 2.2 and 2.3. □

E Proof of Lemma 4.2

Proof. Write $\boldsymbol{s} = \boldsymbol{y} - \boldsymbol{z}_t$. Nesterov's derivative comparison (Nesterov, 2021a, Lemma 3) gives, for every $\tau > 0$,

$$-\frac{1}{\tau}\nabla^2 f(\boldsymbol{x}) - \frac{\tau L_3}{2}\|\boldsymbol{h}\|^2 \boldsymbol{I}_d \preceq \nabla^3 f(\boldsymbol{x})[\boldsymbol{h}] \preceq \frac{1}{\tau}\nabla^2 f(\boldsymbol{x}) + \frac{\tau L_3}{2}\|\boldsymbol{h}\|^2 \boldsymbol{I}_d. \tag{19}$$

We also use

$$\left\|\nabla^2 f(\boldsymbol{x}+\boldsymbol{h}) - \nabla^2 f(\boldsymbol{x}) - \nabla^3 f(\boldsymbol{x})[\boldsymbol{h}]\right\| \le \frac{L_3}{2}\|\boldsymbol{h}\|^2. \tag{20}$$

Taking $\boldsymbol{x} = \boldsymbol{z}_t$, $\boldsymbol{h} = \boldsymbol{s}$, and $\tau = \xi$ in these two inequalities yields

$$\nabla^2 f(\boldsymbol{y}) \preceq \left(1+\frac{1}{\xi}\right)\nabla^2 f(\boldsymbol{z}_t) + \frac{(1+\xi)L_3}{2}\|\boldsymbol{s}\|^2 \boldsymbol{I}_d.$$

Since $\nabla^2(\|\boldsymbol{s}\|^4/4) = \|\boldsymbol{s}\|^2\boldsymbol{I}_d + 2\boldsymbol{s}\boldsymbol{s}^\top$, Assumption 2.3 and direct substitution of Eqs. (10)–(11) give

$$\nabla^2 f_t(\boldsymbol{y}) \preceq \left(1 + \frac{1}{\xi}\right) \nabla^2 \rho_t(\boldsymbol{y}).$$

Indeed, the remaining quartic coefficient equals $(\xi^2 - 1)L_3/6 \geq 0$.

For the reverse comparison, apply Eqs. (19)–(20) at $\boldsymbol{x} = \boldsymbol{y}$, $\boldsymbol{h} = -\boldsymbol{s}$, and $\tau = 1$ to obtain

$$\nabla^2 f(\boldsymbol{z}_t) \preceq 2\nabla^2 f(\boldsymbol{y}) + L_3\|\boldsymbol{s}\|^2\boldsymbol{I}_d.$$

Using $\boldsymbol{H}(\boldsymbol{z}_t) \preceq \nabla^2 f(\boldsymbol{z}_t) + \delta\boldsymbol{I}_d$ and substituting the same Hessian formulas, we get

$$\begin{aligned}\nabla^2 f_t(\boldsymbol{y}) - \frac{\xi - 1}{2\xi}\nabla^2 \rho_t(\boldsymbol{y}) \succeq \frac{1}{\xi}\nabla^2 f(\boldsymbol{y}) + \frac{\xi^2 - 1}{2\xi}\delta\boldsymbol{I}_d \\ + \frac{\xi^2 - 2\xi + 3}{6\xi}L_3\|\boldsymbol{s}\|^2\boldsymbol{I}_d + \frac{1 + \xi}{3}L_3\boldsymbol{s}\boldsymbol{s}^\top \succeq 0.\end{aligned}$$

Here $\xi^2 - 2\xi + 3 = (\xi - 1)^2 + 2 > 0$. The two bounds prove the claim. □

F Proof of Lemma 4.3

Proof. The proof of this lemma follows the same idea as Nesterov (2021c, Lemma 4.3). Let $d_4(\boldsymbol{y}) = \|\boldsymbol{y}\|^4/4$ be the prox-function. We know that this function satisfies the following inequality (Nesterov, 2021c, inequality 2.3):

$$d_4(\boldsymbol{y}') \geq d_4(\boldsymbol{y}) + \langle \nabla d_4(\boldsymbol{y}), \boldsymbol{y}' - \boldsymbol{y}\rangle + \frac{1}{16}\|\boldsymbol{y}' - \boldsymbol{y}\|^4, \quad \forall \boldsymbol{y}, \boldsymbol{y}' \in \mathbb{R}^d. \tag{21}$$

Convexity of f, the optimality condition $\nabla f_t(\boldsymbol{y}_t^*) = \boldsymbol{0}$, and $\delta \geq 0$ give

$$\frac{(1 + \xi)\xi L_3}{6}\|\boldsymbol{y}_t^* - \boldsymbol{z}_t\|^4 \leq -\langle \nabla f(\boldsymbol{z}_t), \boldsymbol{y}_t^* - \boldsymbol{z}_t\rangle \leq \|\nabla f(\boldsymbol{z}_t)\|\,\|\boldsymbol{y}_t^* - \boldsymbol{z}_t\|.$$

If $\boldsymbol{y}_t^* = \boldsymbol{z}_t$, the claim is immediate. Otherwise, division yields

$$\|\boldsymbol{y}_t^* - \boldsymbol{z}_t\| \leq \left(\frac{6\|\nabla f(\boldsymbol{z}_t)\|}{(1 + \xi)\xi L_3}\right)^{1/3} \leq 2\left(\frac{12\|\nabla f(\boldsymbol{z}_t)\|}{L_3(\xi^2 - 1)}\right)^{1/3} = R_t.$$

□

G Proof of Lemma 4.5

Proof. For the monotone variant of AINE we use in Algorithm 2 for $p = 3$, we have $f(\boldsymbol{x}_t) \leq f(\boldsymbol{x}_0)$ and hence $\|\boldsymbol{x}_t - \boldsymbol{x}^*\| \leq R(\boldsymbol{x}_0)$. Then, Lemma 3.2 indicates that $\|\boldsymbol{v}_t - \boldsymbol{x}^*\| \leq \|\boldsymbol{x}_0 - \boldsymbol{x}^*\| \leq R(\boldsymbol{x}_0)$. Therefore, we have $\|\boldsymbol{z}_t - \boldsymbol{x}^*\| \leq \max\{\|\boldsymbol{x}_t - \boldsymbol{x}^*\|, \|\boldsymbol{v}_t - \boldsymbol{x}^*\|\|\} \leq R(\boldsymbol{x}_0)$. Therefore, all the iterates in the outer loop of Algorithm 2 lies in the compact set $\{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x} - \boldsymbol{x}^*\| \leq R(\boldsymbol{x}_0)\}$.

Now, we show that all the iterates generated by the BDGM subroutine in the inner loop to implement the MS oracle also lies in a compact set. By Lemma 4.3, it remains to upper-bound R_t defined in the lemma, which requires an upper bound of $\|\nabla f(\boldsymbol{z}_t)\|$. Since $\|\boldsymbol{z}_t - \boldsymbol{x}_0\| \leq \|\boldsymbol{z}_t - \boldsymbol{x}^*\| + \|\boldsymbol{x}_0 - \boldsymbol{x}^*\| \leq 2R(\boldsymbol{x}_0)$, we can apply Lemma 4.4 to upper-bound $\|\nabla f(\boldsymbol{z}_t)\|$ by

$$\|\nabla f(\boldsymbol{z}_t)\| \leq \|\nabla f(\boldsymbol{x}_0)\| + 2R(\boldsymbol{x}_0)\|\nabla^2 f(\boldsymbol{x}_0)\| + 2\|\nabla^3 f(\boldsymbol{x}_0)\|R^2(\boldsymbol{x}_0) + \frac{4L_3R^3(\boldsymbol{x}_0)}{3}.$$

Hence, R_t and $\bar{R}$ are bounded above by polynomials in $L_3, L_3^{-1}, R(\boldsymbol{x}_0), \|\nabla f(\boldsymbol{x}_0)\|, \|\nabla^2 f(\boldsymbol{x}_0)\|$, and $\|\nabla^3 f(\boldsymbol{x}_0)\|$. □

H Proof of Lemma 4.6

Proof. Set $M = (1+\xi)\xi L_3/6$. If the first term in Eq. (16) is active and $\delta > 0$, the $(\xi^2-1)\delta/\xi$-strong convexity of f_t gives

$$\begin{aligned}\|\nabla f_t(\boldsymbol{y}_t)\| &\le \frac{(\xi^2-1)\delta}{3\xi}\big(\|\boldsymbol{y}_t - \boldsymbol{z}_t\| + \|\boldsymbol{y}_t - \boldsymbol{y}_t^*\|\big)\\ &\le \frac{(\xi^2-1)\delta}{3\xi}\|\boldsymbol{y}_t - \boldsymbol{z}_t\| + \frac{1}{3}\|\nabla f_t(\boldsymbol{y}_t)\|,\end{aligned}$$

which rearranges to the MS condition.

For the second term, the quartic Bregman inequality in both directions implies

$$\|\nabla f_t(\boldsymbol{y}_t)\| \ge \frac{(\xi^2-1)\delta}{\xi}\|\boldsymbol{y}_t - \boldsymbol{y}_t^*\| + \frac{M}{8}\|\boldsymbol{y}_t - \boldsymbol{y}_t^*\|^3.$$

Thus $\|\nabla f_t(\boldsymbol{y}_t)\| \le M\|\boldsymbol{z}_t - \boldsymbol{y}_t^*\|^3/64$ implies $\|\boldsymbol{y}_t - \boldsymbol{y}_t^*\| \le \|\boldsymbol{z}_t - \boldsymbol{y}_t^*\|/2$ and $\|\boldsymbol{y}_t - \boldsymbol{z}_t\| \ge \|\boldsymbol{z}_t - \boldsymbol{y}_t^*\|/2$. Consequently,

$$\frac{1}{2}\lambda_t\|\boldsymbol{y}_t - \boldsymbol{z}_t\| \ge \frac{M}{2}\|\boldsymbol{y}_t - \boldsymbol{z}_t\|^3 \ge \frac{M}{16}\|\boldsymbol{z}_t - \boldsymbol{y}_t^*\|^3 \ge \|\nabla f_t(\boldsymbol{y}_t)\|.$$

This argument also covers $\delta = 0$. □

I Proof of Theorem 4.1

Proof. Set $M = (1+\xi)\xi L_3/6$ and $q = 1 - (\xi-1)/(4(\xi+1))$. At every nonstationary center, $\|\nabla f(\boldsymbol{z}_t)\| \ge \bar{\epsilon}$ and therefore

$$\|\boldsymbol{z}_t - \boldsymbol{y}_t^*\| \ge \frac{\bar{\epsilon}}{\bar{L}_1}.$$

By Lemma 4.6, it suffices to make the residual no larger than

$$\tau := \max\left\{\frac{(\xi^2-1)\delta\bar{\epsilon}}{3\xi\bar{L}_1}, \frac{M\bar{\epsilon}^3}{64\bar{L}_1^3}\right\} > 0.$$

The corrected Lemma 4.1 gives

$$f_t(\boldsymbol{y}_{t,K}) - f_t(\boldsymbol{y}_t^*) \le \frac{(\xi-1)\beta_{\rho_t}(\boldsymbol{z}_t, \boldsymbol{y}_t^*)}{2\xi(q^{-K}-1)}.$$

Moreover, quartic uniform convexity and the $\bar{L}_1$-Lipschitz continuity of ∇f_t on $\mathcal{X}$ imply

$$\frac{M}{16\bar{L}_1^4}\|\nabla f_t(\boldsymbol{y}_{t,K})\|^4 \le f_t(\boldsymbol{y}_{t,K}) - f_t(\boldsymbol{y}_t^*).$$

Since $\|\boldsymbol{z}_t - \boldsymbol{y}_t^*\| \le R_t$, we also have

$$\beta_{\rho_t}(\boldsymbol{z}_t, \boldsymbol{y}_t^*) \le \frac{1}{2}\|\boldsymbol{H}(\boldsymbol{z}_t) + \xi\delta\boldsymbol{I}_d\|R_t^2 + \frac{M}{2}R_t^4 =: P_t.$$

Hence it is sufficient to take

$$K = \left\lceil \frac{\log\left(1 + \dfrac{8(\xi-1)P_t\bar{L}_1^4}{\xi M\tau^4}\right)}{\log(1/q)} \right\rceil.$$

Since $\bar{\epsilon} = \epsilon/R(\boldsymbol{x}_0)$, the quartic term in τ gives

$$\frac{P_t\bar{L}_1^4}{M\tau^4} \le \frac{64^4 P_t R(\boldsymbol{x}_0)^{12}\bar{L}_1^{16}}{M^5\epsilon^{12}}.$$

By the bounded-norm lemma and the Hessian bound in Lemma 4.4, the right-hand side is bounded above by a polynomial in $\epsilon^{-1}, \delta, L_3, L_3^{-1}, R(\boldsymbol{x}_0)$ and the initial derivative norms. Since $\xi > 1$ is fixed, $q \in (0,1)$ is constant, proving Eq. (17). Because $\tau > 0$ even when $\delta = 0$, this also proves finite termination for exact Hessians. □

J Proof of Proposition 5.1

Proof. Fix a deterministic algorithm and a query budget T. The exact pth-order lower bound of Arjevani et al. (2019, Theorem 3) supplies an algorithm-dependent instance with error $\Omega(L_p D_0^{(p+1)/2}/T^{(3p+1)/2})$. For the noise term, apply the same theorem with $p = 1$ and $L_1 = \delta$. Its hard instance is a convex quadratic with $0 \preceq \nabla^2 f \preceq \delta \boldsymbol{I}_d$; choosing $\boldsymbol{H} \equiv \boldsymbol{O}_d$ satisfies Assumption 2.3 and reveals no information beyond the bundled first-order response. Choosing the harder of the two algorithm-dependent instances gives the sum of the two lower bounds up to a factor of two. This proves the stated $\forall$-algorithm, $\exists$-instance quantifiers in the bundled-oracle model. □

K Proof of Proposition 6.1

Proof. For subproblem (18) of the form:

$$\min_{\|\boldsymbol{h}\| \le R} q(\boldsymbol{h}) \triangleq \langle \boldsymbol{b}, \boldsymbol{h} \rangle + \frac{1}{2}\langle \boldsymbol{A}\boldsymbol{h}, \boldsymbol{h} \rangle + \frac{c}{p+1}\|\boldsymbol{h}\|^{p+1},$$

we consider two cases depending on whether the optimal solution $\boldsymbol{h}^*$ lies on the boundary such that $\|\boldsymbol{h}^*\| = R$. If not, then $\boldsymbol{h}^*$ lies in the interior of the feasible set ($\|\boldsymbol{h}\| < R$), it must satisfy the first-order optimality condition:

$$\boldsymbol{0} = \boldsymbol{b} + \boldsymbol{A}\boldsymbol{h} + c\|\boldsymbol{h}\|^{p-1}\boldsymbol{h}.$$

Let $\lambda = c\|\boldsymbol{h}\|^{p-1}$ be a non-negative scalar. The equation becomes $(\boldsymbol{A} + \lambda \boldsymbol{I})\boldsymbol{h} = -\boldsymbol{b}$. For any fixed λ, the solution is $\boldsymbol{h}(\lambda) = -(\boldsymbol{A} + \lambda \boldsymbol{I})^{-1}\boldsymbol{b}$. The problem reduces to finding λ such that:

$$\lambda = c\|\boldsymbol{h}(\lambda)\|^{p-1}.$$

Otherwise, if the solution $\boldsymbol{h}^*$ lies on the boundary $\|\boldsymbol{h}\| = R$. The KKT conditions imply the existence of a Lagrange multiplier $\sigma > 0$ such that:

$$\boldsymbol{0} = \boldsymbol{b} + \boldsymbol{A}\boldsymbol{h} + cR^{p-1}\boldsymbol{h} + \sigma \boldsymbol{h}.$$

By defining a combined multiplier $\gamma = cR^{p-1} + \sigma$, we obtain the same functional form: $\boldsymbol{h}(\gamma) = -(\boldsymbol{A} + \gamma \boldsymbol{I})^{-1}\boldsymbol{b}$. Now, we must find γ such that the norm constraint is exactly satisfied:

$$\|\boldsymbol{h}(\gamma)\| = R.$$

To solve for λ or γ efficiently, we compute the eigenvalue decomposition of the symmetric matrix $\boldsymbol{A}$:

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top, \quad \text{where } \boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_d).$$

This decomposition requires $O(d^\omega)$ running time. Letting $\tilde{\boldsymbol{b}} = \boldsymbol{Q}^\top \boldsymbol{b}$, the squared norm of the solution as a function of the multiplier α (representing either λ or γ) becomes:

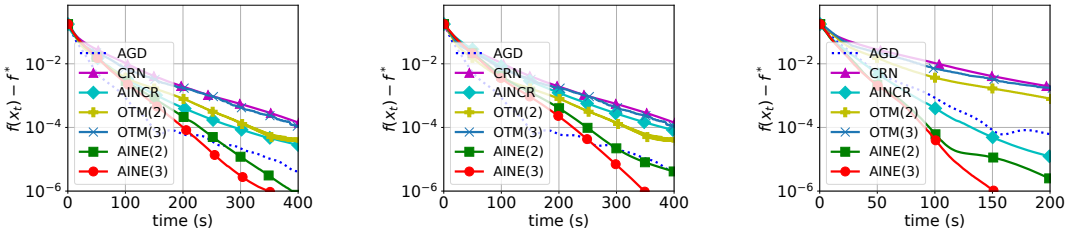
$$\phi(\alpha) = \|\boldsymbol{h}(\alpha)\|^2 = \sum_{i=1}^{d} \frac{\tilde{b}_i^2}{(\lambda_i + \alpha)^2}.$$

The convex function $\phi(\alpha)$ is strictly decreasing for $\alpha > -\lambda_1$. Consequently, the optimal multiplier can be found using the one-dimensional bisection search with C iterations, and each iteration requires $\mathcal{O}(d)$ running time. Thus, the overall running time is $\mathcal{O}(d^\omega + Cd)$. □

L Numerical Experiments

In this section, we conduct numerical experiments on the regularized logistic regression problem of the following form:

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} f(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^{n} \log(1 + \exp(-b_i \boldsymbol{a}_i^\top \boldsymbol{x})) + \frac{1}{2n}\|\boldsymbol{x}\|^2,$$



(a) $B = 100$ (b) $B = 1000$ (c) $B = 10000$

Table 2: The results on logistic regression with different Hessian batch sizes B. We use AINE(p) and OTM(p) to denote the corresponding methods with order $p \in \{2, 3\}$.

where $\{(\boldsymbol{a}_i, b_i\}_{i=1}^n$ is a dataset with n samples. We set $\boldsymbol{H}(\boldsymbol{x})$ as a mini-batch Hessian estimator with batch size $B \in \{100, 1000, 10000\}$, which provably satisfies Assumption 2.3 with high probability due to the matrix Hoeffding inequality (Agafonov et al., 2024a, Section 7). We run experiments on the `covtype` dataset ($n = 581012, d = 54$) from LIBSVM (Chang & Lin, 2011) by comparing our proposed methods for $p \in \{2, 3\}$ with baselines AGD (Nesterov, 1983), CRN (Nesterov & Polyak, 2006), AITM (Agafonov et al., 2024a;b), and optimal tensor method (OTM) (Carmon et al., 2022) for $p \in \{2, 3\}$. The results are presented in Figure 2, showcasing the superiority of our method.

M AI USAGE

All the main theoretical results are proved by the authors. AI tools are used to write the code in the numerical experiments, polish writing, and check the correctness of the proof.