UNIFIED ACCELERATION OF HIGH-ORDER ALGORITHMS UNDER GENERAL HÖLDER CONTINUITY*

CHAOBING SONG†, YONG JIANG‡, AND YI MA§

Abstract. In this paper, through an intuitive *vanilla proximal method* perspective, we derive a concise *unified acceleration framework (UAF)* for minimizing a convex function that has Hölder continuous derivatives with respect to general (non-Euclidean) norms. The UAF reconciles the two different high-order acceleration approaches, one by Nesterov and Baes [27, 3, 31] and one by Monteiro and Svaiter [23]. As a result, the UAF unifies the high-order acceleration instances [27, 3, 31, 15, 16, 23, 18, 6, 14] of the two approaches by only two problem-related parameters and two additional parameters for framework design. Meanwhile, the UAF (and its analysis) is *the first approach* to make high-order methods applicable for high-order smoothness conditions with respect to non-Euclidean norms. Furthermore, the UAF is the first approach that can match the existing lower bound of the iteration complexity for minimizing a convex function with Hölder continuous derivatives [16]. For practical implementation, we introduce a new and effective heuristic that significantly simplifies the binary search procedure required by the framework. We use experiments to verify the effectiveness of the heuristic and demonstrate clear and consistent advantages of high-order acceleration methods over first-order ones, in terms of run-time complexity. Finally, the UAF is proposed directly in the general composite convex setting, thus show that the existing high-order algorithms [27, 3, 31, 16, 6, 14] can be naturally extended to the general composite convex setting.

Key words. High-order algorithms, Nesterov's acceleration, proximal method, non-Euclidean norm.

AMS subject classifications. 49M15, 49M37, 65K05, 68Q25, 90C25, 90C30

1. Introduction.

In optimization, people often consider the problem of minimizing a convex function:

$$\min_{x\in\mathbb{R}^d} f(x). \tag{1.1}$$

A typical assumption is that $f(x)$ has L-Lipschitz continuous gradients with respect to ($w.r.t.$) the Euclidean norm $\|\cdot\|_2$, $i.e.$,

$$\|\nabla f(x) - \nabla f(y)\|_2 \le L\|x-y\|_2, \tag{1.2}$$

where $L>0$ is the Lipschitz constant. For this problem, to find an ϵ-accurate solution x such that $f(x)-f(x^*)\le\epsilon$, the classic gradient descent method:

$$x_{k+1} = x_k - \eta\nabla f(x_k)$$

with $\eta \le 1/L$ takes $O(\epsilon^{-1})$ iterations. Nevertheless, it is known that from [28], for a convex function $f(x)$ with L-Lipschitz continuous gradients, a lower-bound for the number of iterations for any first-order algorithms is known to be

$$O\big(\epsilon^{-1/2}\big), \quad (L\text{-Lipschitz continuous gradients}). \tag{1.3}$$

In the seminal work [26], Nesterov has introduced an acceleration technique, the so-called *accelerated gradient descent (AGD)* algorithm, that achieves this optimal lower bound. This algorithm dramatically improves the convergence rate of smooth convex optimization with negligible per-iteration cost. Besides the smooth convex problem (1.1) under the Euclidean norm setting (1.2), AGD can also be generalized to solve the composite convex problem [5, 1, 13], in which the objective function may contain a second possibly non-smooth but simple convex term (see (2.6)). Meanwhile, AGD can be extended to the more general (non-Euclidean) norm settings [12, 1, 21], also achieving the optimal rate (1.3).

*Submitted to the editors on October 09, 2019.

Funding: This work was done during Chaobing Song's visit to Professor Yi Ma's group at UC Berkeley. The work is partially supported by the TBSI program and EECS Startup fund of Professor Yi Ma.

†Tsinghua-Berkeley Shenzhen Institute (TBSI), Tsinghua University (songcb16@mails.tsinghua.edu.cn).

‡Tsinghua-Berkeley Shenzhen Institute (TBSI), Tsinghua University (jiangy@sz.tsinghua.edu.cn)

§EECS Department, University of California, Berkeley (yima@eecs.berkeley.edu).

1.1. High-order Acceleration Methods with Lipschitz Continuity. To hope for a better iteration complexity beyond $O(\epsilon^{-1/2})$, $f(x)$ needs to be smooth for its high-order derivatives. A common assumption is that $f(x)$ has (p,ν,L)-Hölder continuous derivatives:

$$\frac{1}{(p-1)!}\|\nabla^p f(x) - \nabla^p f(y)\|_2 \le L\|x-y\|_2^{\nu}, \tag{1.4}$$

for some $\nu \in [0,1], p \in \mathbb{Z}_+$. Notice that for $p=1$ and $\nu=1$, this condition becomes the first order L-Lipschitz continuous gradient (1.2) above. Here, for $p \ge 2$, the $\|\cdot\|_2$ norm of a p-th order tensor denotes its operator norm [31] $w.r.t.$ the vector 2-norm $\|\cdot\|_2$. Sometimes, when $\nu=1$, the function is said to have (p,L)-Lipschitz continuous derivatives:

$$\frac{1}{(p-1)!}\|\nabla^p f(x) - \nabla^p f(y)\|_2 \le L\|x-y\|_2. \tag{1.5}$$

If a convex function $f(x)$ satisfies (1.5), the recent work [2] has given a lower-bound on the complexity: any deterministic algorithm would need at least

$$O\big(\epsilon^{-\frac{2}{3p+1}}\big), \quad ((p,L)\text{-Lipschitz continuous derivatives}) \tag{1.6}$$

iterations to find an ϵ-accurate solution. For the special case $p=2$, [27] has proposed an "*accelerated cubic regularized Newton method*" (ACNM) that achieves an iteration complexity of $O(\epsilon^{-\frac{1}{3}})$. From a different approach, after proposing an accelerated hybrid proximal extragradient (A-HPE) framework, [23] has implemented an "*accelerated Newton proximal extragradient*" (A-NPE) instance of the A-HPE framework that has achieved the optimal complexity $O(\epsilon^{-\frac{2}{7}})$[1] for $p=2$, although each iteration requires a nontrivial binary search procedure.

To achieve better complexity results and also being encouraged by the fact that third-order methods can often be implemented as efficiently as second-order methods [31], there is an increasing interest to extend ACNM and implement the A-HPE framework to even higher-order smoothness settings ($p \in \{3,4\ldots,\}$) [3, 31, 18, 14, 6]. In particular, by extending ACNM, [3] and [31] have proposed accelerated tensor methods with $O(\epsilon^{-\frac{1}{p+1}})$ iteration complexity for $p \in \{2,3,\ldots\}$. Meanwhile, by implementing A-HPE, [23, 18, 14, 6] have proposed accelerated methods that achieve the optimal $O(\epsilon^{-\frac{2}{3p+1}})$ iteration complexity, although just like A-NPE, all these methods need the nontrivial binary search procedure.

Hence the current situation seems to be: methods [27, 3, 31, 15, 16] by extending ACNM have advantages with simpler implementation, while methods [23, 18, 14, 6] by implementing A-HPE can in theory achieve the optimal rate $O(\epsilon^{-\frac{2}{3p+1}})$. However, it remains somewhat mysterious how we could reconcile the differences between these two approaches. In addition, the A-HPE framework is somewhat abstract so implementing it in the high-order setting requires rather nontrivial techniques [23, 18, 14, 6]. It remains unclear how to propose a concise but equivalently powerful alternative to the A-HPE framework and obtain these different instances of A-HPE in a unified way. Furthermore, although AGD can be generalized to general non-Euclidean norm settings, up to now, it is not known whether high-order methods can have a similar generalization. Finally, both the ACNM and A-HPE approaches do not directly address the composite convex setting (see (2.6)) at the framework level, hence obtaining high-order algorithms in this setting is highly desired and seems nontrivial [23, 15, 18].

[1] When talking about iteration complexity, we mean the complexity in terms of outer iteration without concerning about the inner implementation of subproblems.

1.2. Acceleration under Hölder Continuity and Our Results. Besides the Lipschitz continuous setting, the more general Hölder continuous setting (1.4) is also of increased interest, partly for designing universal optimization schemes [29, 37, 16, 10]. If $f(x)$ has $(1,\nu,L)$-Hölder continuous gradients, a lower bound for the iteration complexity is known to be [25]:

$$O\Big(\epsilon^{-\frac{2}{1+3\nu}}\Big), \quad ((1,\nu,L)\text{-Hölder continuous gradients}). \tag{1.7}$$

An algorithm that can achieve this lower bound has been proposed in [24].

For the more general setting of (p,ν,L)-Hölder continuous derivatives, during the preparation of this paper, [16] has given a lower bound of iteration complexity

$$O\Big(\epsilon^{-\frac{2}{3(p+\nu)-2}}\Big), \quad ((p,\nu,L)\text{-Hölder continuous derivatives}). \tag{1.8}$$

By extending Nesterov's method in [31], [16] has proposed a method that achieves the iteration complexity $O(\epsilon^{-\frac{1}{p+\nu}})$. To the best of our knowledge, methods that can achieve the lower bound $O\big(\epsilon^{-\frac{2}{3(p+\nu)-2}}\big)$ are still unknown.

In this paper, for the minimization of convex functions with (p,ν,L)-Hölder continuous derivatives, we propose a *unified acceleration framework* (UAF), see Algorithm 5.1, that achieves the iteration complexity of $O\big(\epsilon^{-\frac{2}{3(p+\nu)-2}}\big)$ ($p\in\{1,2,\ldots\}, \nu\in[0,1]$ with $p+\nu\geq 2$, $L>0$), which matches the lower bound [16]. To be more precise, if a convex function $f(x)$ has (p,ν,L)-Hölder continuous derivatives, our algorithm can find an ϵ-accurate solution with

$$O\Big(\epsilon^{-\frac{q}{(q+1)(p+\nu)-q}}\Big) \tag{1.9}$$

iterations, where q is a tunable parameter[2] such that $2\leq q\leq p+\nu$. Notice that our result and algorithm unify previously known results as (important) special cases:

- For the case of L-Lipschitz continuous gradients [28] where $p=\nu=1$ and $q=2$, the rate (1.9) of the proposed algorithm achieves the lower bound $O\big(\epsilon^{-\frac{1}{2}}\big)$ of (1.3).
- For the more general setting of (p,ν,L)-Hölder continuous derivatives: when $p\in\{2,3,\ldots,\}, q=p+\nu$, it recovers the complexity $O\big(\epsilon^{-\frac{1}{p+\nu}}\big)$ of the method in [16].

Meanwhile, by setting $q=2$, the rate (1.9) of the UAF is *the first* convergence result that matches the lower bound $O\big(\epsilon^{-\frac{2}{3(p+\nu)-2}}\big)$ of (1.8) [16] under the Hölder continuous setting.

Besides the unified convergence rate (1.9), the UAF has several significant improvements over the ACNM approach and the A-HPE framework. First, the UAF provides a continuous transition from the ACNM approach to the A-HPE framework by choosing q from $p+\nu$ to 2. Second, as we will soon see, the UAF can be conveniently instantiated by only specifying two problem-related parameters and two adjustable parameters for framework design, and thus recover the high-order acceleration algorithms [27, 3, 31, 15, 16, 23, 18, 6, 14] without extra effort. Third, we provide *the first* and also a unified convergence rate analysis for both the Euclidean and non-Euclidean norm settings, and thus opens the possibility of applying high-order methods in the non-Euclidean norm setting.[3] Fourth, the UAF is proposed and analyzed directly under the composite convex setting (see (2.6)), hence our results imply that all existing high-order algorithms [27, 3, 31, 16, 6, 14] can be naturally extended to the general composite convex setting.

[2] As we will later see, q is the order of the uniform convexity of the proxy-function for framework design. [16] has used a uniformly convex proxy-function with $q=(p+\nu)$-th order, while [18, 14, 6] have used a uniformly convex proxy-function with $q=2$-nd order.

[3] which is pertinent to many important practical problems such as logistic regression loss in machine learning, see Example 2.4.

In terms of implementation for high-order acceleration algorithms, to obtain the optimal rate that matches the lower bound [2], we must employ a binary search procedure to find a suitable coupling coefficient in each iteration, which may substantially slow down the practical performance [31]. Therefore, in addition to the above theoretical results, we introduce a simple heuristic for finding the coupling coefficient, suggested by our analysis, so that the resulting implementation does not need a binary search procedure required by the optimal acceleration method. Our experiments show that this simple heuristic is extremely effective and can easily ensure the conditions needed to achieve the optimal rate. This leads to a very practical implementation of the optimal acceleration algorithms without extra implementation cost, alleviating concerns raised by [31]. Last but not the least, with a general *restart scheme*, our analysis for the general convex setting extends to the uniformly convex setting. The resulting algorithm complexity can match the existing lower bounds [2] in most important cases[4].

1.3. Our Approach. In this paper, instead of directly designing an algorithm and then analyzing its iteration complexity, we consider a different paradigm to make our approach and algorithm more intuitive and explainable. The paradigm is inspired by the unified theory for first-order algorithms [13] and the continuous-time interpretations of Nesterov's acceleration [35, 21, 22, 36]. Our approach to the algorithmic design is based on an idealized but impractical algorithm called *vanilla proximal method (VPM)*, introduced in Section 3. The VPM aims to solve a regularized program of the original one with an arbitrary convergence rate depending on parameters of our choice. However, the VPM serves more as an ideal target and is itself computationally infeasible to realize.

We show that, in Section 4, to overcome the computational hurdle, one can instead solve a continuous-time *convex approximation* to the VPM. Then an accelerated continuous-time dynamics can be derived simply as sufficient conditions to ensure that solution to the approximate convex program achieves the same convergence rate as the original VPM. Such point of view unifies the existing continuous-time accelerated dynamics introduced in [35], [21] and [36] and serves as an arguably better guideline for the design of practical algorithms in the discrete setting.

In practice, to realize the desired accelerated dynamics, we need to know how to implement them in the discrete setting as an iterative algorithm. To this end, we consider a discrete-time *convex approximation* to the VPM. However, as we will see in Section 5, in order for the discrete-time approximation to achieve the same convergence rate as VPM, we must solve a fixed-point problem which itself is computationally infeasible (if not impossible) in practice. To circumvent this difficulty, we propose to solve the fixed-point problem *approximately* by solving a *smooth approximation* to the VPM which becomes a tractable problem. Finally, by combing the convex approximation and the smooth approximation to the VPM, we propose the implementable discrete-time unified acceleration framework which achieves the optimal iteration complexity given in (1.9) for the minimization of convex functions with (p, ν, L)-Hölder continuous derivatives (for $p \in \{1, 2 \ldots\}, \nu \in [0, 1]$ with $p + \nu \geq 2$ and $L > 0$).

2. Preliminaries. Before we proceed, we first introduce some notations. Let $:=$ denote a definition. Let $[n]$ denote the set $\{1, 2, \ldots, n\}$. For $p = \{1, 2, \ldots\}$, let $p! := 1 \times 2 \times \cdots \times p$ with $0! := 1$. let $\|\cdot\|$ denote a norm of vectors and $\|\cdot\|_*$ denote the dual norm of $\|\cdot\|$. For $x \in \mathbb{R}^d$ and $q \geq 1$, let $\|x\|_q := (\sum_{i=1}^d |x_i|^q)^{\frac{1}{q}}$. For a matrix $B \in \mathbb{R}^{d\times d}$ and $p, q \geq 1$, denote the operator norm $\|B\|_{p,q} := \max_{x\in\mathbb{R}^d}\{\|Bx\|_q : \|x\|_p \leq 1\}$. By a little abuse of notation, for a convex function $f(x)$ defined on $\mathbb{R}^d$, let $\nabla f(x)$ denote the gradient at x or one point in the subgradient set $\partial f(x)$. For a function $f(x; y)$, x denotes the variable of $f(x; y)$, y the

[4]Due to limit of space, one may refer to details in the longer arXiv version: https://arxiv.org/pdf/1906.00582.pdf.

parameter of $f(x;y)$, and $\nabla f(x;y)$ is the gradient or one point in the subgradient set $\partial f(x;y)$ $w.r.t.$ x.

Similar to the notations in [31], for $p \in \{1,2,\ldots\}$, we use $\nabla^p f(x)[y_1,y_2,\ldots,y_p]$ to denote the directional derivative of a function f at x along the directions $y_i \in \mathbb{R}^d, i = 1,2\ldots,p$. Then $\nabla^p f(x)[\cdot]$ is a symmetric p-linear form and its operator norm $w.r.t.$ a norm $\|\cdot\|$ is defined as

$$\|\nabla^p f(x)\|_* := \max_{y_1,y_2,\ldots,y_p} \{\nabla^p f(x)[y_1,\ldots,y_p] : \|y_i\| \le 1, i = 1,2,\ldots,p\}. \tag{2.1}$$

DEFINITION 2.1 (Strictly, Uniformly, or Strongly Convex). *We say a continuous function $f(x)$ is convex on $\mathbb{R}^d$, if $\forall x,y \in \mathbb{R}^d$, one has*

$$f(x) \ge f(y) + \langle \nabla f(y), x-y\rangle; \tag{2.2}$$

$f(x)$ is strictly convex on $\mathbb{R}^d$, if the equality sign in (2.2) holds if and only if $x = y$;

$f(x)$ is (s,σ)-uniformly convex on $\mathbb{R}^d$ $w.r.t.$ a norm $\|\cdot\|$, if $\forall x,y \in \mathbb{R}^d$, one has

$$f(x) \ge f(y) + \langle \nabla f(y), x-y\rangle + \frac{\sigma}{s}\|x-y\|^s, \tag{2.3}$$

where $s \ge 2$ is the order of uniform convexity and $\sigma \ge 0$ the constant of uniform convexity;

$f(x)$ is σ-strongly convex on $\mathbb{R}^d$ $w.r.t.$ $\|\cdot\|$, if $f(x)$ is $(2,\sigma)$-uniformly convex on $\mathbb{R}^d$ $w.r.t.$ $\|\cdot\|$.

In Definition 2.1, uniform convexity can be viewed as an extension to the better known concept of strong convexity. Example 2.2 gives two cases of uniform convexity.

Example 2.2 (Uniform Convexity). $\frac{1}{2}\|x\|_q^2 (1 < q \le 2)$ is $(2, q-1)$-uniformly convex on $\mathbb{R}^d$ $w.r.t.$ $\|\cdot\|_q$ [4]; $\frac{1}{q}\|x\|_2^q$ $(q \ge 2)$ is $(q, 2^{2-q})$-uniformly convex on $\mathbb{R}^d$ $w.r.t.$ $\|\cdot\|_2$ [27].

Starting from the work of [29], there is an increasing interest to replace the Lipschitz continuity assumption by the Hölder continuity assumption [37, 34, 30, 16] and to propose universal algorithms in the sense that the convergence of algorithms can optimally adapt to the Hölder parameter. [29, 37] have considered first-order algorithms with Hölder continuous gradients $w.r.t.$ $\|\cdot\|_2$; [15] has proposed cubic regularized Newton methods for minimizing functions with Hölder continuous Hessians $w.r.t.$ $\|\cdot\|_2$; [10, 16] have considered tensor methods for minimizing convex functions with p-th Hölder continuous derivatives $w.r.t.$ $\|\cdot\|_2$ $(p \in \{2,3,\ldots\})$. In this paper, we extend the definition of Hölder continuous derivatives $w.r.t.$ any norm $\|\cdot\|$, including non-Euclidean norms. Our analysis and results will be applicable to this general setting (Example 2.4 shows some important cases in machine learning.).

DEFINITION 2.3 (Hölder Continuous Derivative). *We say a function $f(x)$ on $\mathbb{R}^d$ has (p,ν,L)-Hölder continuous derivatives $w.r.t.$ $\|\cdot\|$, if $\forall x,y \in \mathbb{R}^d$, one has*

$$\frac{1}{(p-1)!}\|\nabla^p f(x) - \nabla^p f(y)\|_* \le L\|x-y\|^\nu, \tag{2.4}$$

where $p \in \{1,2,3,\ldots\}$ denotes the order of derivative, $0 \le \nu \le 1$ denotes the Hölder parameter and $L > 0$ is the constant of smoothness.

$f(x)$ is said to have (p,L)-Lipschitz continuous derivatives on $\mathbb{R}^d$ $w.r.t.$ $\|\cdot\|$ if $f(x)$ has $(p,1,L)$-Hölder continuous derivatives on $\mathbb{R}^d$ $w.r.t.$ $\|\cdot\|$.

In Definition 2.3, for $p = 1$, $\|\cdot\|_*$ denotes the dual norm of $\|\cdot\|$; for $p \in \{2,3,\ldots\}$, $\|\cdot\|_*$ denotes the operator norm of tensor of p-th order $w.r.t.$ $\|\cdot\|$, which is defined by (2.1).

Example 2.4 (Non-Euclidean High-order Smoothness). Consider the objective function $f(x) := \frac{1}{n}\sum_{j=1}^n \log(1+\exp(-\bar{b}_j \bar{a}_j^T x))$ for logistic regression, where $j \in [n], \bar{a}_j \in \mathbb{R}^d, \bar{b}_j \in$

$\{1,-1\}$. Denote $B := \frac{1}{n}\sum_{j=1}^{n} \bar{a}_j \bar{a}_j^T$. For $1 \le p \le 2$ and q satisfying $1/p + 1/q = 1$, let $\|\nabla^s f(x)\|_q$ denote the operator norm of $\nabla^s f(x)(s = 2, 3)$ in (2.1) *w.r.t.* the vector norm $\|\cdot\|_p$. Then we have

$$\|\nabla^2 f(x) - \nabla^2 f(y)\|_q \le \|B\|_{p,q} \max_{j\in[n]} \|\bar{a}_j\|_q^\nu \cdot \|x - y\|_p^\nu. \tag{2.5}$$

Proof. See Section A.1. ∎

In the paper, we consider the following composite convex optimization problem:

$$\min_{x\in\mathbb{R}^d} f(x) := g(x) + l(x), \tag{2.6}$$

where $g(x)$ is a closed proper convex function and $l(x)$ is a simple convex but maybe non-smooth function. We consider the case when $g(x)$ has (p, ν, L)-Hölder continuous derivatives for all $x \in \mathbb{R}^d$. Then we can define the following two auxiliary functions:

$$\hat{f}(x; y) := g(y) + \langle \nabla g(y), x - y\rangle + l(x), \tag{2.7}$$

$$\tilde{f}(x; y) := g(y) + \sum_{i=1}^{p} \frac{1}{i!} \nabla^i g(y)[x - y]^i + l(x), \tag{2.8}$$

where we do not linearize the term $l(x)$ which may be nonsmooth. Formally, we have:

LEMMA 2.5. *If $g(x)$ and $l(x)$ are convex, and $g(x)$ has (p, ν, L)-Hölder continuous derivatives, then we have: for all $x, y \in \mathbb{R}^d$,*

$$\hat{f}(x; y) \le f(x), \tag{2.9}$$

$$|f(x) - \tilde{f}(x; y)| \le \frac{L}{p}\|x - y\|^{p+\nu}, \tag{2.10}$$

$$\|\nabla f(x) - \nabla \tilde{f}(x; y)\|_* \le L\|x - y\|^{p+\nu-1}. \tag{2.11}$$

Proof. See Section A.2. ∎

Because of (2.9), in this paper, $\hat{f}(x; y)$ is viewed as a lower-bound convex approximation to $f(x)$ for any parameter $y \in \mathbb{R}^d$. $\tilde{f}(x; y)$ satisfies (2.10) and (2.11), and gives a high-order smooth approximation to $f(x)$ for any parameter $y \in \mathbb{R}^d$. In our analysis, the convexity and smoothness assumptions are used only by the two inequalities (2.9) and (2.11), which allow a unified treatment for the smooth and the composite convex settings (with or without the term $l(x)$). Meanwhile, because we only need the property (2.11) of high-order smoothness, it implies that in convex optimization, the high-order smoothness is mainly used to give a more accurate estimation of the "implicit gradient" $\nabla f(x)$.

Finally, we give two inequalities in Lemma 2.6 which will be used in our analysis.

LEMMA 2.6. *Given a sequence $\{b_k\}_{k\ge 0}$ with $b_0 = 0$ and $b_k > 0$ $(k \ge 1)$. One has:*

- *For $\rho \ge 1$ and $C > 0$, if $\forall k \ge 1$, $(b_k - b_{k-1})^\rho \ge C b_k^{\rho-1}$, then $b_k \ge C\left(\frac{k}{\rho}\right)^\rho$;*
- *For $\rho \ge 1, \delta > 0$ and $C > 0$, if $\forall k \ge 1$, $\sum_{i=1}^{k} \left(\frac{b_i^{\rho-1}}{(b_i - b_{i-1})^\rho}\right)^\delta \le C$, then $b_k \ge C^{-\frac{1}{\delta}}\left(\frac{k}{\rho}\right)^{\rho+\frac{1}{\delta}}$.*

Proof. See Section A.3. ∎

3. Vanilla Proximal Method. Let us start our study by considering the composite convex optimization problem in (2.6). In the following discussion, we assume that x^* is a minimizer of $f(x)$ on $\mathbb{R}^d$. To design an acceleration algorithm to minimize $f(x)$, we first introduce a

Algorithm 3.1 Vanilla Proximal Method (VPM)

1: **Input:** an initialized point $x_0 \in \mathbb{R}^d$, a scalar $A > 0$, a convex function $h(x; x_0)$.
2: Find a $z \in \mathbb{R}^d$ that satisfies

$$z := \operatorname{argmin}_{x\in\mathbb{R}^d} \left\{ \psi^{\text{vpm}}(x) := Af(x) + h(x; x_0) \right\}. \tag{3.1}$$

3: **return** z.

so-called *vanilla proximal method (VPM)*, that considers to minimize an auxiliary function $\psi^{\text{vpm}}(x)$ as in Algorithm 3.1.

In $\psi^{\text{vpm}}(x)$, the convex term $h(x; x_0)$ should satisfy the non-negative property:

Assumption 3.1. $\forall x, x_0 \in \mathbb{R}^d$, $h(x; x_0) \geq 0$ with $h(x; x_0) = 0$ if and only if $x = x_0$.

In the VPM, (3.1) is a convex program and thus there exists a minimizer z. By using only the optimality condition of (3.1) and Assumption 3.1, we can characterize the "convergence rate" of the VPM as below.

THEOREM 3.2. *The solution z generated by Algorithm 3.1 satisfies*

$$f(z) - f(x^*) \leq \frac{h(x^*; x_0)}{A}. \tag{3.2}$$

Proof. By the definition of $\psi^{\text{vpm}}(x)$ in (3.1), one has

$$\min_{x\in\mathbb{R}^d} \psi^{\text{vpm}}(x) \leq Af(x^*) + h(x^*; x_0). \tag{3.3}$$

Then by the optimality condition of z and the nonnegativity of $h(x; x_0)$, one has

$$Af(z) \leq Af(z) + h(z; x_0) = \min_{x\in\mathbb{R}^d} \psi^{\text{vpm}}(x). \tag{3.4}$$

By the upper bound of $\min_{x\in\mathbb{R}^d} \psi^{\text{vpm}}(x)$ in (3.3) and lower bound of $\min_{x\in\mathbb{R}^d} \psi^{\text{vpm}}(x)$ in (3.4), after a simple rearrangement, Theorem 3.2 is proved. ∎

By Theorem 3.2, the VPM may converge with any convergence rate if A is chosen to a large enough value. Although solving the subproblem (3.1) is impractical in general, it provides us a good starting point to design practical algorithms: by making certain assumptions on the objective function $f(x)$ and the proxy function $h(x; x_0)$, it is possible to achieve or approach the convergence rate of the VPM by solving a tractable approximation to (3.1).

4. Continuous-time Accelerated Descent Dynamics. The subproblem (3.1) in the VPM is merely conceptual as it is almost as difficult as minimizing the original function. Nevertheless, if $f(x)$ is convex, one can always seek more tractable approximations. From an acceleration perspective, the convex approximation $\hat{f}(x; y)$ in Lemma 2.5 gives a lower bound for $f(x)$ at the state y. The minimizer of $\hat{f}(x; y)$ would suggest an aggressive direction and step for the next iterate to go to. However, for such iterates not to diverge too far from the landscape of $f(x)$, we also need a good upper bound. A basic idea is that up to time t, we have already traversed a path $x_\tau, \tau \in [0, t)$ over the landscape of $f(x)$. We could potentially use all the lower-bounds $\hat{f}(x; x_\tau)$ of $f(x)$ to construct a good upper bound to guide the next step. The simplest possible form for such an upper bound we could consider is a superposition (or an integral) of these lower bounds to guide the descent trajectory as follows.

$$z_t := \operatorname{argmin}_{x\in\mathbb{R}^d} \left\{ \psi_t^{\text{cont}}(x) := \int_0^t a_\tau \hat{f}(x; x_\tau) \mathrm{d}\tau + h(x; x_0) \right\}, \tag{4.1}$$

where $\forall\ 0 < \tau \le t, a_\tau > 0$ and satisfies $\int_0^t a_\tau \mathrm{d}\tau = A_t$ with $a_0 = A_0 = 0$ and $\{x_\tau\}_{0\le\tau\le t}$ is the optimization path and its relationship with z_t will be determined soon.

In this section, our main goal is to show that the widely studied continuous-time accelerated dynamics arise from a sufficient condition that allows (4.1) to achieve the same convergence rate as the original VPM. First, the upper bound (3.3) of ψ^{vpm} is extended to ψ_t^{cont} as follows.

LEMMA 4.1. *$\forall t \ge 0$, we have* $\min_{x\in\mathbb{R}^d} \psi_t^{\mathrm{cont}}(x) \le A_t f(x^*) + h(x^*; x_0)$.

Proof. Lemma 4.1 can be easily proven by using (2.9). ∎

In other words, Lemma 4.1 provides a lower bound of $A_t f(x^*)$. Second, the lower bound (3.4) of ψ^{vpm} can be extended to ψ_t^{cont} as follows, at least approximately.

LEMMA 4.2. *$\forall t \ge 0$, we have* $A_t f(x_t) \le \min_{x\in\mathbb{R}^d} \psi_t^{\mathrm{cont}}(x) + \int_0^t \langle \nabla f(x_\tau), A_\tau \dot{x}_\tau - a_\tau(z_\tau - x_\tau)\rangle \mathrm{d}\tau$.

Proof. See Section B.1 ∎

We would like to make this approximation as close as possible and establish $\min_{x\in\mathbb{R}^d} \psi_t^{\mathrm{cont}}(x)$ as an upper bound of $A_t f(x_t)$, at least along certain path by our choice. To this end, based on Lemmas 4.1 and 4.2, we have the following theorem.

THEOREM 4.3 (Continuous-Time VPM). *If the continuous-time trajectories $\{x_t\}_{t\ge0}$ and $\{z_t\}_{t\ge0}$ are evolved according to the dynamics:*

$$\begin{cases} A_t \dot{x}_t = a_t(z_t - x_t), \\ z_t = \operatorname{argmin}_{x\in\mathbb{R}^d} \left\{ \int_0^t a_\tau \hat{f}(x; x_\tau)\mathrm{d}\tau + h(x; x_0) \right\}, \end{cases} \tag{4.2}$$

where $\forall\ 0 < \tau \le t$, $a_\tau > 0$, $\int_0^t a_\tau \mathrm{d}\tau = A_t$, and $a_0 = A_0 = 0$, then for all $t > 0$, one has

$$f(x_t) - f(x^*) \le \frac{h(x^*; x_0)}{A_t}. \tag{4.3}$$

Proof. If $A_t \dot{x}_t = a_t(z_t - x_t)$, from Lemma 4.2, one has $A_t f(x_t) \le \min_{x\in\mathbb{R}^d} \psi_t^{\mathrm{cont}}(x)$. Combining Lemma 4.1, we have (4.3). ∎

In Theorem 4.3, (4.2) does not specify any concrete values or forms for a_t and A_t, except the condition $a_\tau > 0$, $\int_0^t a_\tau \mathrm{d}\tau = A_t$ $(0 < \tau \le t)$ and $a_0 = A_0 = 0$;[5] meanwhile it does not specify any concrete form for $h(x; x_0)$. As result, by instantiating the dynamical system (4.2) for different choices of a_t, A_t, and h, one may obtain all ODEs previously introduced and studied for algorithm acceleration in the literature [35, 21, 22, 36], respectively.[6]

Remark 4.4. Although we have derived the dynamics (4.2) from a different perspective, it should be noted that the dynamical system (4.2) is an extension and refinement to the ODE derived by the "approximate duality gap technique (ADGT)" [13]. The main difference is that instead of giving a upper bound of $f(x_t)$ and a lower bound of $f(x^*)$, we give a upper bound of $A_t f(x_t)$ and a lower bound of $A_t f(x^*)$. This modification allows us to set $A_0 = 0$ rather than $A_0 > 0$, and thus the initialization expression about A_0 can be removed. Such a modification simplifies future derivation and analysis greatly.

5. Unified Acceleration Framework. To achieve the same convergence rate of the VPM, the continuous-time approximation needs the extra ODE condition in (4.2), which is reasonable to assume in the continuous setting. In the discrete-time setting, if all other conditions remain

[5] Theoretically A_t should be chosen such that the differential equation has a unique solution.

[6] See examples given in the long arXiv version: https://arxiv.org/pdf/1906.00582.pdf for details.

unchanged, except that we replace the weighted continuous-time approximation (4.1) by a weighted discrete-time counterpart, one may see that the ODE will be replaced by a condition that requires us to find a solution to a *fixed-point* problem (which will be clear in Lemma 5.3). Unfortunately, directly solving this fixed-point problem is computationally infeasible in practice. To remedy this difficulty, we need stronger conditions for the proxy function $h(x;x_0)$, the associated norm $\|\cdot\|$, and the smooth component $g(x)$ of $f(x)$ as given in Assumption 5.1 below.

Assumption 5.1. $\forall x, x_0 \in \mathbb{R}^d, p \in \{1,2,\cdots\}, \nu \in [0,1]$ with $p+\nu \geq 2$, $q \in [2, p+\nu]$, $\gamma > 0$ and $L > 0$, we have

1. $h(x;x_0)$ satisfies Assumption 3.1 and is (q,γ)-uniformly convex $w.r.t.$ a norm $\|\cdot\|$.
2. $\frac{1}{q}\|x\|^q$ is (q,β)-uniformly convex $w.r.t.$ the norm $\|\cdot\|$.
3. $g(x)$ has (p,ν,L)-Hölder continuous derivatives $w.r.t.$ the norm $\|\cdot\|$.

Based on Assumption 5.1, we consider a weighted discrete-time convex approximation of (3.1): for $k \geq 0$,

$$z_k := \operatorname{argmin}_{x\in\mathbb{R}^d}\Big\{\psi_k^{\rm dis}(x) := \sum_{i=1}^{k} a_i \hat{f}(x;x_i) + h(x;x_0)\Big\}, \tag{5.1}$$

where we assume that $\forall 1 \leq i \leq k, a_i > 0$, $A_i := \sum_{j=1}^{i} a_j$ and $a_0 = A_0 = 0$, $h(x;x_0)$ satisfies Assumption 5.1, and $\hat{f}(x;x_i)$ is defined in Lemma 2.5. Meanwhile, in (5.1), when $k=0$, we let $\psi_0^{\rm dis}(x) = h(x;x_0)$ and thus $z_0 = \operatorname{argmin}_{x\in\mathbb{R}^d} h(x;x_0) = x_0$. We now derive the unified acceleration framework by analyzing the conditions needed to emulate the same rate of the VPM.

First, the upper bound (3.3) of $\psi^{\rm vpm}$ can be extended to the discrete case $\psi_k^{\rm dis}$ trivially.

LEMMA 5.2. $\forall k \geq 0$, *one has* $\min_{x\in\mathbb{R}^d} \psi_k^{\rm dis}(x) \leq A_k f(x^*) + h(x^*;x_0)$.

Proof. Lemma 5.2 can be easily proven by using (2.9). ∎

Then in Lemma 5.3 below, we show how the lower bound (3.4) of $\psi^{\rm vpm}$ can be extended to the discrete case $\psi_k^{\rm dis}$ with some extra terms.

LEMMA 5.3. $\forall i \geq 1$, *let* $E_i := A_i\left\langle \nabla f(x_i), x_i - \frac{a_i}{A_i} z_i - \frac{A_{i-1}}{A_i} x_{i-1}\right\rangle - \frac{\gamma}{q}\|z_i - z_{i-1}\|^q$. *Then* $\forall k \geq 1$, *one has* $A_k f(x_k) - \psi_k^{\rm dis}(z_k) \leq \sum_{i=1}^{k} E_i$.

Proof. See Section C.1. ∎

In Lemma 5.3, the extra negative term $-\frac{\gamma}{q}\|z_i - z_{i-1}\|^q$ in E_i is from the uniform convexity of $h(x;x_0)$. If $h(x;x_0)$ is only convex (i.e. $\gamma = 0$), this negative term does not exist and thus a sufficient condition for $E_i \leq 0$ is:

$$x_i = \frac{a_i}{A_i} z_i + \frac{A_{i-1}}{A_i} x_{i-1}, \quad \forall 1 \leq i \leq k. \tag{5.2}$$

By (5.1), z_i is a function of x_i. Therefore finding x_i to satisfy (5.2) is reduced to a fixed-point problem (so is it for z_i). It is computationally infeasible (if not impossible) to find an exact solution to this problem in general. Nevertheless, if $h(x;x_0)$ satisfies Assumption 5.1, the term E_i contains a negative term $-\frac{\gamma}{q}\|z_i - z_{i-1}\|^q$. So there is hope that an approximate solution to the fixed-point problem (5.2) can still make $E_i \leq 0$.

To approximately solve the fixed-point problem, for convenient analysis, inspired by [17, 12], we define a pair $(\hat{x}_{i-1}, \hat{z}_i)$ such that

$$\hat{x}_{i-1} := \frac{a_i}{A_i} z_{i-1} + \frac{A_{i-1}}{A_i} x_{i-1}, \qquad \hat{z}_i := \frac{A_i}{a_i} x_i - \frac{A_{i-1}}{a_i} x_{i-1}. \tag{5.3}$$

By the definition of $\hat{z}_i$ in (5.3), we have $x_i = \frac{a_i}{A_i}\hat{z}_i + \frac{A_{i-1}}{A_i}x_{i-1}$. Therefore (5.3) can be viewed as two-step fixed-point iterations for x_i based on $\hat{x}_{i-1}$ and $\hat{z}_i$. Here $\hat{z}_i$ can be viewed as the best estimate of the desired fixed point z_i based on the calculated x_i in our algorithm. It is defined for convenience and will only be used in our analysis but not in the algorithm.

Based on the definition of $(\hat{x}_{i-1}, \hat{z}_i)$, Assumption 5.1, and the definition of E_i in Lemma 5.3, we have the following result.

LEMMA 5.4. *For $i \geq 1$ and any $\gamma_i' \in (0, \gamma]$, we have*

$$E_i \leq a_i\Big\langle \nabla f(x_i) + \frac{\gamma_i' A_i^{q-1}}{a_i^q}\nabla\frac{1}{q}\|x_i - \hat{x}_{i-1}\|^q, \hat{z}_i - z_i\Big\rangle$$

$$-\gamma_i'\Big(\frac{A_i^q}{qa_i^q}\|x_i - \hat{x}_{i-1}\|^q + \frac{\beta}{q}\|\hat{z}_i - z_i\|^q\Big). \tag{5.4}$$

Proof. See Section C.2. ∎

In Lemma 5.4, we purposely introduce a new parameter γ_i', which as we will soon show, helps unify the four high-order instances [23, 18, 6, 18] of the A-HPE framework. Meanwhile, because of the uniform convexity of $\frac{1}{q}\|\cdot\|^q$, the negative term $-\frac{\gamma}{q}\|z_i - z_{i-1}\|^q$ is reduced to two negative terms and an inner product.

By Lemma 5.4, if we can find x_i such that

$$\nabla f(x_i) + \frac{\gamma_i' A_i^{q-1}}{a_i^q}\nabla\frac{1}{q}\|x_i - \hat{x}_{i-1}\|^q = 0, \tag{5.5}$$

then we can ensure $E_i \leq 0$. However the problem of finding x_i that satisfies (5.5) is equivalent to solving the VPM problem exactly in (3.1) with the settings $x_0 := \hat{x}_{i-1}, h(x; x_0) := \frac{1}{q}\|x - \hat{x}_{i-1}\|^q, A := \frac{a_i^q}{\gamma_i' A_i^{q-1}}$, which is computationally infeasible in general. Fortunately, the two negative terms in (5.4) may dominate small errors if we can solve the VPM problem (5.5) approximately. Hence we approximate the intermediate VPM problem (5.5) by a smooth approximation $\tilde{f}(x_i; \hat{x}_{i-1})$ using the fact

$$\|\nabla f(x_i) - \nabla\tilde{f}(x_i; \hat{x}_{i-1})\|_* \leq L\|x_i - \hat{x}_{i-1}\|^{p+\nu-1}, \tag{5.6}$$

from Lemma 2.5. Then by Lemmas 2.5 and 5.4, we have Lemma 5.5.

LEMMA 5.5. *Denote $c_q := \big(\beta(q-1)^{1-q}\big)^{\frac{1}{q}}$ and $\lambda_i' := \frac{a_i^q}{c_q\gamma_i' A_i^{q-1}}$. For $i \geq 1$, one has*

$$E_i \leq \Big(\big(L\lambda_i'\|x_i - \hat{x}_{i-1}\|^{p+\nu-q}\big)^{\frac{q}{q-1}} - 1\Big)\frac{\gamma_i' A_i^q}{qa_i^q}\|x_i - \hat{x}_{i-1}\|^q$$

$$+a_i\Big\langle \nabla\tilde{f}(x_i; \hat{x}_{i-1}) + \frac{\gamma_i' A_i^{q-1}}{a_i^q}\nabla\frac{1}{q}\|x_i - \hat{x}_{i-1}\|^q, \hat{z}_i - z_i\Big\rangle. \tag{5.7}$$

Proof. Sec Section C.3. ∎

From Lemma 5.5, to ensure $E_i \leq 0$, the VPM problem (5.5) can be reduced to an easier smooth approximation problem

$$\nabla\tilde{f}(x_i; \hat{x}_{i-1}) + \frac{\gamma_i' A_i^{q-1}}{a_i^q}\nabla\frac{1}{q}\|x_i - \hat{x}_{i-1}\|^q = 0, \tag{5.8}$$

and we also need the condition

$$L\lambda_i'\|x_i - \hat{x}_{i-1}\|^{p+\nu-q} \leq \theta_2 \leq 1 \tag{5.9}$$

to hold, where $\theta_2 \in (0, 1]$ is a constant.

We here discuss the role of the parameter γ_i'. So far our derivation works for any $\gamma_i' \in (0, \gamma]$. A simple choice of γ_i' would be $\gamma_i' := \gamma$. Nevertheless, under the condition (5.9), for any $\alpha \in [0, 1]$, we could choose γ_i' to satisfy:

$$\gamma_i' = \left(\frac{L\lambda_i' \|x_i - \hat{x}_{i-1}\|^{p+\nu-q}}{\theta_2} \right)^{\frac{\alpha}{1-\alpha}} \gamma, \tag{5.10}$$

where for $\alpha = 1$, we set $\frac{\alpha}{1-\alpha} = \lim_{\alpha \to 1^-} \frac{\alpha}{1-\alpha} = +\infty$. This would still ensure $\gamma_i' \in (0, \gamma]$. But notice that λ_i' in the RHS depends on γ_i'. To sort out an explicit expression for so-defined γ_i', we denote

$$\lambda_i := \frac{a_i^q}{c_q \gamma A_i^{q-1}}. \tag{5.11}$$

Then by the definition of λ_i' in Lemma 5.5, with (5.10) and (5.11), we can write γ_i' in the form:

$$\gamma_i' = \left(\frac{L\lambda_i \|x_i - \hat{x}_{i-1}\|^{p+\nu-q}}{\theta_2} \right)^{\alpha} \gamma. \tag{5.12}$$

Then by the fact for all $s \geq 0, t \geq 2, x \in \mathbb{R}^d$,

$$\|x\|^s \nabla \frac{1}{t} \|x\|^t = \nabla \frac{1}{s+t} \|x\|^{t+s}, \tag{5.13}$$

and combing (5.11) and (5.12), it follows that (5.8) is equivalent to

$$\nabla \tilde{f}(x_i; \hat{x}_{i-1}) + \frac{L^\alpha}{c_q \lambda_i^{(1-\alpha)} \theta_2^\alpha} \nabla \frac{1}{\alpha(p+\nu) + (1-\alpha)q} \|x_i - \hat{x}_{i-1}\|^{\alpha(p+\nu)+(1-\alpha)q} = 0. \tag{5.14}$$

Or equivalently, let $\varsigma := \alpha(p+\nu) + (1-\alpha)q$, and then x_i is the solution to the following minimization problem:

$$x_i := \operatorname{argmin}_{x \in \mathbb{R}^d} \left\{ \tilde{f}(x; \hat{x}_{i-1}) + \frac{L^\alpha}{c_q \lambda_i^{(1-\alpha)} \theta_2^\alpha \varsigma} \|x - \hat{x}_{i-1}\|^\varsigma \right\}. \tag{5.15}$$

In (5.15), because $\alpha \in [0, 1]$, the power of the norm $\|x - \hat{x}_{i-1}\|$ ranges from $p + \nu$ to q freely, which unifies the choice $\alpha = 0$ in [23] and $\alpha = 1$ in [6]. Meanwhile, [18, 14] has used a mixture of both $\alpha = 0$ and $\alpha = 1$ in their formulations, which are also equivalent to (5.14) by (5.13). A surprising phenomenon is that, as our analysis shows, the choice of α in (5.15) does not affect the convergence rate (except the constant).

Meanwhile, by (5.12), (5.9) is equivalent to

$$\omega_i := L\lambda_i \|x_i - \hat{x}_{i-1}\|^{p+\nu-q} \leq \theta_2 \leq 1, \tag{5.16}$$

where we call ω_i as a *convergence indicator* in the sense that if for all $1 \leq i \leq k$, $\omega_i \leq 1$, then the iterate x_k will converge according to the following theorem; otherwise, the convergence of x_k is not guaranteed. Based on the equivalence relationship between (5.8) and (5.15), (5.9) and (5.16), we have Theorem 5.6.

THEOREM 5.6 (Discrete-Time VPM). *If Assumption 5.1 is true, and in* (5.1)*, $\forall i \geq 1$, the sequences $\{a_i\}, \{A_i\}$ satisfy $a_i > 0, A_i = A_{i-1} + a_i$ with $a_0 = A_0 = 0$, $\{x_i\}$ satisfies* (5.15)*, and $\{\lambda_i\}$ defined in* (5.11) *satisfies* (5.16)*, then for $k \geq 1$, one has*

$$f(x_k) - f(x^*) \leq \frac{h(x^*; x_0)}{A_k}. \tag{5.17}$$

Proof. See Section C.4. ∎

As we see, Theorem 5.6 is very much like the discrete-time approximation version of Theorem 3.2. To accurately characterize the convergence rate from (5.17), we need to have a good lower-bound for A_k. In Theorem 5.6, by $A_i = A_{i-1} + a_i$, the definition of λ_i (5.11) and the condition (5.16), it follows that A_i must satisfy the condition

$$\frac{L(A_i - A_{i-1})^q}{c_q \gamma A_i^{q-1}} \|x_i - \hat{x}_{i-1}\|^{p+\nu-q} \le \theta_2 \le 1. \tag{5.18}$$

Therefore A_i cannot be chosen as an arbitrarily large value as in the continuous-time setting. Except the basic condition $A_0 = 0$ and for $i \ge 1, A_i > 0$, (5.18) is the only condition A_i needs to satisfy, therefore one may expect that the tightest bound of A_i should be obtained if $\frac{L(A_i - A_{i-1})^q}{c_q \gamma A_i^{q-1}} \|x_i - \hat{x}_{i-1}\|^{p+\nu-q} = O(1)$. In other words, we hope that

$$0 < \theta_1 \le L\lambda_i \|x_i - \hat{x}_{i-1}\|^{p+\nu-q} = \frac{L(A_i - A_{i-1})^q}{c_q \gamma A_i^{q-1}} \|x_i - \hat{x}_{i-1}\|^{p+\nu-q} \le \theta_2 \le 1, \tag{5.19}$$

where θ_1 and θ_2 are $O(1)$ constants. To verify this point of view, we discuss below the two settings $q = p + \nu$ and $q < p + \nu$, respectively.

When $q = p + \nu$, we have $\lambda_i = \frac{(A_i - A_{i-1})^q}{c_q \gamma A_i^{q-1}}$ and $\|x_i - \hat{x}_{i-1}\|^{p+\nu-q} = 1$. Taking A_i as a variable, then for all $A_i > A_{i-1}$, by the fact $q \ge 2$ and

$$\frac{\mathrm{d} \log \lambda_i}{\mathrm{d} A_i} = \frac{(q-1)A_{i-1} + A_i}{A_i(A_i - A_{i-1})} > 0, \tag{5.20}$$

we have λ_i is a strictly monotonically increasing function $w.r.t.$ A_i, which is an one to one mapping. Therefore determining the lower bound of A_i is equivalent to determining the lower bound of λ_i. To ensure $E_i \le 0$, by the condition (5.16) , when $q = p + \nu$, $L\lambda_i$ is upper bounded by the constant $\theta_2 \le 1$. Therefore the tightest lower bound for λ_i is obtained if $L\lambda_i$ is lower bounded by a constant $\theta_1 \in (0, \theta_2]$. Then by Lemma 2.6 and Theorem 5.6, we obtain Theorem 5.7.

THEOREM 5.7 (Convergence Rate for the Case $q = p + \nu$). *If Assumption 5.1 is true, c_q is defined in Lemma 5.5, and in (5.1), $\forall i \ge 1$, the sequences $\{a_i\}, \{A_i\}$ satisfy $a_i > 0, A_i = A_{i-1} + a_i$ with $a_0 = A_0 = 0$, $\{x_i\}$ satisfies (5.15), and $\{\lambda_i\}$ defined in (5.11) satisfies*

$$0 < \theta_1 \le L\lambda_i \le \theta_2 \le 1, \tag{5.21}$$

then for $k \ge 1$, we have

$$A_k \ge \frac{\theta_1 c_q \gamma}{L} \left(\frac{k}{p+\nu} \right)^{p+\nu}, \tag{5.22}$$

and

$$f(x_k) - f(x^*) \le \frac{h(x^*; x_0)}{A_k} \le \frac{L}{\theta_1 c_q \gamma} h(x^*; x_0) \left(\frac{p+\nu}{k} \right)^{p+\nu}. \tag{5.23}$$

Proof. See Section C.5. ∎

When $q < p + \nu$, because the condition of λ_i to ensure $E_i \le 0$ involves the unknown x_i, the situation seems to be more complicated. Nevertheless, under the conditions (5.15) and (5.16), and combining Lemmas 5.2 and 5.3, we can obtain a condition as in Lemma 5.8 below that leads to a good lower bound for A_k.

LEMMA 5.8. *Assume* $\{x_i\}$ *satisfies* (5.15) *and* $\{\omega_i\}$ *satisfies* (5.16). *Then if* $2 \le q < p+\nu$, *we have*

(5.24)
$$\sum_{i=1}^{k} \omega_i^{\frac{s}{p+\nu-q}} \Big(\frac{A_i^{p+\nu-1}}{(A_i - A_{i-1})^{p+\nu}} \Big)^{\frac{q}{p+\nu-q}} \le q\theta_2^{\alpha}(1-\theta_2^{\frac{q}{q-1}})^{-1}\gamma^{-\frac{p+\nu}{p+\nu-q}} \Big(\frac{L}{c_q}\Big)^{\frac{q}{p+\nu-q}} h(x^*;x_0).$$

Proof. See Section C.6. ∎

In Lemma 5.8, if $\theta_2 \in (0,1)$, then the RHS of (5.24) will be a positive constant. Therefore if ω_i on the LHS of (5.24) is lower bounded by a constant $\theta_1 \in (0, \theta_2]$, then we use Lemma 2.6 to give a lower bound about A_i. Based on the above analysis, and combining Lemma 2.6, Theorem 5.6 and Lemma 5.8, we can characterize the convergence rate of the proposed iteration when $2 \le q < p+\nu$.

THEOREM 5.9 (Convergence Rate for the Case $2 \le q < p+\nu$). *If Assumption 5.1 is true,* c_q *is defined in Lemma 5.5, and in* (5.1), $\forall i \ge 1$, *the sequences* $\{a_i\}, \{A_i\}$ *satisfy* $a_i > 0, A_i = A_{i-1} + a_i$ *with* $a_0 = A_0 = 0$, $\{x_i\}$ *satisfies* (5.15), *and* $\{\lambda_i\}$ *defined in* (5.11) *satisfies*

$$0 < \theta_1 \le \omega_i = L\lambda_i \|x_i - \hat{x}_{i-1}\|^{p+\nu-q} \le \theta_2 < 1, \tag{5.25}$$

then by defining $C_0 := \Big(q\theta_2^{\alpha}\big(1-\theta_2^{\frac{q}{q-1}}\big)^{-1}\Big)^{-\frac{p+\nu-q}{q}} \theta_1^{\frac{\alpha(p+\nu)+(1-\alpha)q}{q}} \gamma^{\frac{p+\nu}{q}} c_q$, *we have*

$$A_k \ge \frac{C_0}{L}(h(x^*;x_0))^{-\frac{p+\nu-q}{q}} \Big(\frac{k}{p+\nu}\Big)^{\frac{(q+1)(p+\nu)-q}{q}} \tag{5.26}$$

and

$$f(x_k) - f(x^*) \le \frac{h(x^*;x_0)}{A_k} \le \frac{L}{C_0}(h(x^*;x_0))^{\frac{p+\nu}{q}} \left(\frac{p+\nu}{k}\right)^{\frac{(q+1)(p+\nu)-q}{q}}. \tag{5.27}$$

Proof. See Section C.7. ∎

In Theorems 5.7 and 5.9, if we do not consider the constants, in both $q = p+\nu$ and $2 \le q < p+\nu$ settings, we can find an ϵ-accurate solution x such that $f(x) - f(x^*) \le \epsilon$ with

$$O\Big(\epsilon^{-\frac{q}{(q+1)(p+\nu)-q}}\Big)$$

iterations, where $q \in [2, p+\nu]$. It is easy to find that the rate will be the best as $O\big(\epsilon^{-\frac{2}{3(p+\nu)-2}}\big)$ if we set $q = 2$. In fact $O\big(\epsilon^{-\frac{2}{3(p+\nu)-2}}\big)$ matches the lower bound of iteration complexity [16] for all the settings of $p \in \{1, 2, \dots, \}$ and $\nu \in [0,1]$ as long as $p+\nu \ge 2$. As q becomes large, the rate $O\Big(\epsilon^{-\frac{q}{(q+1)(p+\nu)-q}}\Big)$ will become worse. However, particularly, when $q = p+\nu$, λ_i can be determined trivially and thus the setting $q = p+\nu$ is suboptimal but has the advantage of algorithmic implementation, as we will elaborate on later.

Regarding the other two parameters θ_1, θ_2, when $q = p+\nu$, based on Theorem 5.7, to minimize the bound in (5.23), the optimal choice will be $\theta_1 = 1$ and thus $\theta_2 = 1$ by $\theta_1 \le \theta_2 \le 1$. When $q < p+\nu$, based on Theorem 5.9, one can optimize the choice of θ_1, θ_2 by minimizing the bound in (5.27) under the constraint $0 < \theta_1 \le \theta_2 < 1$.

As we have noted before, by varying the parameter α from 0 to 1 in (5.15), the range of the power of $\|x - \hat{x}_{i-1}\|$ changes from q to $p+\nu$. For $q = p+\nu$, as Theorem 5.7 indicates, choice of α has no influence on the convergence rate; for $2 \le q < p+\nu$, as Theorem 5.9

TABLE 1
Algorithmic Instances of the Unified Acceleration Framework with $h(x;x_0) := \frac{1}{q}\|x - x_0\|_2^q$.

Instances	p	ν	q	α
[3, 31]	$\{2,3,\ldots\}$	1	$p+1$	1
[16]	$\{2,3,\ldots\}$	$[0,1]$	$p+\nu$	1
[23]	2	1	2	0
[6]	$\{2,3,\ldots\}$	1	2	1
[18, 14]	$\{2,3,\ldots\}$	1	2	a mixture of 0 and 1

shows, α only has a minor influence on the constant in the bound. Therefore, our result shows that α can be chosen according to implementation convenience without worrying about the convergence rate.

Compared with the existing results for high-order optimization [32, 27, 31, 16] and [23, 6, 18, 14], our convergence results are given under the Hölder continuous assumption $w.r.t.$ a general norm $\|\cdot\|$ that satisfies Assumption 5.1. Such general norms include the Euclidean norm $\|x\|_2$ and the generalized Euclidean norm $\sqrt{x^TBx}$ as special cases, where B is any positive definite matrix. To the best of our knowledge, this is the first convergence result for high-order optimization that can be applied to the high-order non-Euclidean smoothness setting. To this end, we have adopted a new proof paradigm inspired by the intuitive proof techniques for the accelerated extra-gradient descent (AXGD) algorithm [12] for first order methods.

Summarizing the above results, we obtain a *unified acceleration framework* (UAF) shown in Algorithm 5.1. In the algorithm, the parameters p, ν are from the problem setting and the parameters q, α and the proxy function $h(x;x_0)$ are for framework design. These parameters can vary in their entire feasible ranges. By specifying p, ν, q, α and $h(x;x_0)$, we obtain algorithmic instances of UAF. As results, Algorithm 5.1 recovers many existing algorithms. We give a few examples in Table 1[7].

Meanwhile, Algorithm 5.1 also includes several new interesting instances. First, if we set $p = \nu = 1, q = 2, \alpha \in [0,1]$, Algorithm 5.1 defines a new variant of AGD with an $O(1/k^2)$ convergence rate. Such variant is similar to the variant AXGD of AGD. One advantage of this variant is that Algorithm 5.1 allows $h(x;x_0)$ to be any strongly convex function $w.r.t.$ $\|\cdot\|$, while AXGD assumes that $h(x;x_0)$ is the Bregman divergence of a strongly convex function $w.r.t.$ $\|\cdot\|$. Second, if we set $p \in \{2,3,\ldots,\}, \nu \in [0,1], q = 2, \alpha \in [0,1]$, then we obtain the first kind of high-order algorithms that can attain the optimal rate $O(\epsilon^{-\frac{2}{3(p+\nu)-2}})$ for the composite minimization problem (2.6) with the smooth component $g(x)$ having (p,ν,L)-Hölder continuous derivatives $w.r.t.$ $\|\cdot\|$.

For the loop from Step 4 to 7 in Algorithm 5.1, we need solve two subproblems:

- The first one is about finding λ_i such that the minimizer x_i of the objective (5.15), together with λ_i, satisfy the conditions (5.28) and (5.29).
- The second one is about finding the solution z_i of a discrete-time convex approximation problem of the VPM in Step 6. Because in our setting the convex approximation $\hat{f}(x;y)$ defined in Lemma 2.5 is a linear function plus a simple convex function $l(x)$, the subproblem of finding z_i can be solved efficiently.

When $p = \nu = 1$ and $q = 2$, the subproblem associated with Step 5, namely (5.15), is reduced to a proximal gradient decent step [33], which can be solved efficiently. However,

[7] As we have mentioned, [18, 14] have used a mixture of 0 and 1 for α, which is also equivalent to (5.14) by (5.13). To simplify presentation and practical implementation, we usually only consider chosing a single α in Algorithm 5.1.

Algorithm 5.1 Unified Acceleration Framework (UAF)

1: **Input:** $\hat{f}(x;y)$ and $\tilde{f}(x;y)$ in (2.7) and (2.8), $h(x;x_0)$, $\|\cdot\|$ in Assumption 5.1.
2: Set the constant $\theta_2 \in (0,1]$ if $q = p+\nu$, $\theta_2 \in (0,1)$ if $2 \le q < p+\nu$; set $\theta_1 \in (0,\theta_2]$.
3: Set $\alpha \in [0,1]$, $c_q = \left(\beta(q-1)^{1-q}\right)^{\frac{1}{q}}$, $\varsigma = \alpha(p+\nu)+(1-\alpha)q$.
4: Set $A_0 = 0, x_0 = z_0 \in \mathbb{R}^d$.
5: **for** $i = 1$ **to** k **do**
6: $x_i = \operatorname{argmin}_{x\in\mathbb{R}^d}\left\{\tilde{f}(x;\hat{x}_{i-1}) + \frac{L^\alpha}{qc_q\lambda_i^{(1-\alpha)}\theta_2^\alpha\varsigma}\|x-\hat{x}_{i-1}\|^\varsigma\right\}$,
where we find a $\lambda_i > 0$ such that a_i, A_i, λ_i and $\hat{x}_{i-1} \in \mathbb{R}^d$ satisfy

$$A_i = A_{i-1} + a_i,\ \lambda_i = \frac{a_i^q}{c_q\gamma A_i^{q-1}},\ \hat{x}_{i-1} = \frac{A_{i-1}}{A_i}x_{i-1} + \frac{a_i}{A_i}z_{i-1}, \tag{5.28}$$

$$\theta_1 \le L\lambda_i\|x_i - \hat{x}_{i-1}\|^{p+\nu-q} \le \theta_2. \tag{5.29}$$

7: Update $z_i = \operatorname{argmin}_{x\in\mathbb{R}^d}\left\{\sum_{j=1}^i a_j\hat{f}(x;x_j) + h(x;x_0)\right\}$.
8: **end for**
9: **return** x_k.

in the setting of high-order optimization, *i.e.*, $p \in \{2,3,\ldots\}$, (5.15) is nontrivial and will dominate the per-iteration cost in general. Finding a general efficient procedure to solve this subproblem remains active research. Nevertheless, for some special important cases, there already exist efficient algorithms. For example, if $p=2, \nu=1, \alpha=1$ and the maybe nonsmooth part $l(x)=0$, (5.15) is reduced to an iteration of cubic regularized Newton method (CNM), which can be solved efficiently by the Lanzcos method [9]; if $p=3, \nu=1, \alpha=1$ and $l(x)=0$, (5.15) is reduced to a third-order convex multivariate polynomial and can be solved as efficiently as the iteration of CNM in many cases [31, 7].

Notice that, in Step 5, for the setting $q = p+\nu$, λ_i can be determined easily as it does not depend on x_i and thus A_i, a_i can be solved efficiently by solving a simple one-dimensional equation with Newton method. However, for the setting $2 \le q < p+\nu$, the condition (5.29) depends on the solution x_i and can not be determined so trivially. In fact, as of now, when $2 \le q < p+\nu$, we do not even know whether we can find such a pair (x_i, λ_i) that satisfies all the conditions simultaneously. However, as nearly a trivial extension to [6], the following Proposition 5.10 ensures such a pair always exists until we attain the minimizer.

PROPOSITION 5.10. *Let $A \ge 0, \lambda \ge 0, x, y \in \mathbb{R}^d$ such that $f(x) \ne f(x^*)$. Assume that $a(\lambda)$ is implicitly defined by*

$$\lambda = \frac{(a(\lambda))^q}{c_q\gamma(A+a(\lambda))^{q-1}}, \quad \text{and} \quad x(\lambda) = \frac{a(\lambda)}{A+a(\lambda)}x + \frac{A}{A+a(\lambda)}y, \tag{5.30}$$

$$w(v) = \operatorname{argmin}_{z\in\mathbb{R}^d}\left\{\tilde{f}(z;v) + \frac{L^\alpha}{c_q\lambda^{(1-\alpha)}\theta_2^\alpha\varsigma}\|z-v\|^\varsigma\right\}, \tag{5.31}$$

$$\chi(\lambda) = L\lambda\|w(x(\lambda)) - x(\lambda)\|^{p+\nu-q}, \tag{5.32}$$

where the constants $p, q, \nu, \alpha, c_q, \gamma, L, \varsigma$ and θ_2 are given in Algorithm 5.1. Then $\chi(\lambda)$ is a continuous function with $\chi(0) = 0$ and $\chi(+\infty) = +\infty$.

Proof. See Section C.8 ∎

By Proposition 5.10, with the setting $A := A_{i-1}, x := z_{i-1}, y := x_{i-1}$, we can always use a binary search procedure to find a pair (x_i, λ_i) such that $\chi(\lambda_i) = L\lambda_i\|x_i - \hat{x}_{i-1}\|^{p+\nu-q}$

satisfies the condition (5.29). For the case with $\alpha = 0, q = 2$ and $\|\cdot\| := \|\cdot\|_2$, a complexity analysis for a binary search procedure can be found in [18]; for the case with $\alpha = 1, p \in \{2, 3, \ldots\}, \nu = 1$ and $\|\cdot\| := \|\cdot\|_2$, a complexity analysis for a binary search procedure can be found in [6]. Although it is possible to give a complexity analysis of binary search for the general setting in (5.31), in this paper we consider another perspective.

In the Discussion section of [31], Nesterov claims that from the view of practical efficiency, the Algorithm 5.1 with the suboptimal setting $q = p + \nu$ may be better than the Algorithm 5.1 with the optimal setting $q = 2$, where "optimal" is in the sense of iteration complexity. If we do not consider the implementation cost in the Step 5 of Algorithm 5.1 and ignore the difference of constants in the bound of Theorems 5.7 and 5.9, to attain an ϵ-accurate solution such that $f(x) - f(x^*) \leq \epsilon$, the ratio from the number of iterations of the suboptimal algorithm with $q = p + \nu$ to that of the optimal algorithm with $q = 2$ is

$$O\Big(\Big(\frac{1}{\epsilon}\Big)^{\frac{1}{p+\nu} - \frac{2}{3(p+\nu)-2)}}\Big) = O\Big(\Big(\frac{1}{\epsilon}\Big)^{\frac{p+\nu-2}{(p+\nu)(3(p+\nu)-2)}}\Big).$$

If $p = 2, \nu = 1$, *i.e.*, the commonly second-order setting, the ratio will be $O\left(\left(\frac{1}{\epsilon}\right)^{\frac{1}{21}}\right)$, which implies that when we pursue an accuracy $\epsilon = 2^{-21} \approx 10^{-6}$, if the per-iteration cost of the optimal setting $q = 2$ (or the settings $2 \leq q < p + \nu$) is twice larger than the suboptimal setting $q = p + \nu$, then the small advantage of the optimal setting will be removed by the additional implementation complexity. Because of this effect, a binary search procedure which involves $O(\log \frac{1}{\epsilon})$ calls to the subprocedure for finding x_i may be rather unrealistic in practice. Therefore in this paper, instead of binary search, we propose a simple heuristic to find a pair (x_i, λ_i) that satisfy the condition (5.29). The proposed heuristic only needs call the subprocedure once for finding x_i, which will be explained in Section 6.

Remark 5.11. The idea that two-step fixed-point iterations lead to acceleration is first introduced in [12], which has proposed the variant AXGD of AGD. In this paper, such point of view motivates us to simplify the analysis by defining an intermediate variable $\hat{z}_i$ in (5.3), whereas the main strategy leading to acceleration in this paper is to use a combination of a convex approximation (5.1) to the original VPM problem (3.1) and a smooth approximation (5.15) to the intermediate VPM problem (5.5).

Remark 5.12. Similar to [16], it is also possible to give a universal version of Algorithm 5.1 in the sense that, by modifying Algorithm 5.1 according to the paradigm of [16], we can obtain a near-optimal rate even if the Hölder parameter ν is unknown. Such an improvement is interesting, however it goes beyond the scope of this paper and will be left for further research.

6. Implementation Details and Experimental Validation. High-order optimization is a very different situation from first-order optimization in that the optimal acceleration method (*e.g.*, the UAF with $q = 2$) requires certain conditions to be met (in each iteration). Those conditions sometimes are not so trivial to be satisfied. In fact, in the UAF Algorithm 5.1, for $2 \leq q < p + \nu$, it is not trivial to find a pair (λ_i, x_i) that satisfies the condition (5.29). In (5.16), we have defined $\omega_i = L\lambda_i \|x_i - \hat{x}_{i-1}\|^{p+\nu-q}$ as a convergence indicator in the sense that $\forall 2 \leq q \leq p + \nu$, if

$$\omega_i = L\lambda_i \|x_i - \hat{x}_{i-1}\|^{p+\nu-q} \leq \theta_2 \leq 1,$$

Algorithm 5.1 will converge according to Theorem 5.6; otherwise, the convergence behavior of Algorithm 5.1 cannot be guaranteed. More specifically, when $q = p + \nu$ if ω_i satisfies (5.21), then Algorithm 5.1 converges according to Theorem 5.7; when $2 \leq q < p + \nu$, if ω_i satisfies (5.25), then Algorithm 5.1 converges according to Theorem 5.9. When $q = p + \nu$, we

can easily find $0 < \theta_1 \leq \omega_i \leq \theta_2 \leq 1$ to satisfy (5.21); while when $2 \leq q < p + \nu$, because ω_i involves the variable x_i, it is nontrivial to find a $0 < \theta_1 \leq \omega_i \leq \theta_2 < 1$ to satisfy (5.25). A standard technique to ensure that the value of the convergence indicator ω_i stays in $[\theta_1, \theta_2]$ is through a binary search procedure [23, 18, 6]. However, as per our discussion at the end of Section 5, the cost of the binary search procedure could substantially reduce the advantage of convergence rate of the optimal method in practice.

6.1. A Good Heuristic for Practical Implementation. In this section, inspired by the analysis of Theorem 5.9, for the Algorithm 5.1 with $2 \leq q < p + \nu$, instead of using a binary search, we introduce a simple heuristic: in the i-th iteration of Algorithm 5.1, A_i is set as its lower bound such that

$$A_i = \frac{C_0}{L}\big(h(x^*; x_0)\big)^{-\frac{p+\nu-q}{q}} \left(\frac{i}{p+\nu}\right)^{\frac{(q+1)(p+\nu)-q}{q}}, \tag{6.1}$$

where all the constants are from Theorem 5.9. With so assigned A_i, λ_i and a_i can be easily determined by (5.28). Therefore the per-iteration cost under the setting $2 \leq q < p + \nu$ will remain the same as the setting $q = p + \nu$.

However, if we use the heuristic (6.1) of A_i for $2 \leq q < p + \nu$, there is no theoretical guarantee for convergence of the algorithm. In this section, we conduct experiments to show that this heuristic (6.1) is surprisingly effective: the values of the convergence indicator (5.16) will always remain within the range $(0, 1)$, hence Algorithm 5.1 converges according to Theorem 5.9.

To be more precise, we consider the commonly second-order ($i.e., p = 2$) setting with Euclidean Lipschitz smoothness Hessians ($i.e, \nu = 1$), and set $h(x; x_0) := \frac{1}{q}\|x - x_0\|_2^q$, where q is chosen as $q \in \{2, 2.5, 3\} \subset [2, p + \nu]$. Meanwhile, as shown in Theorems 5.7 and 5.9, the parameter α of Algorithm 5.1 has only a minor influence on performance. Therefore to simplify our implementation, we always set $\alpha = 1$. By setting $\alpha = 1$, when $p = 2, \nu = 1$, given $\hat{x}_{i-1}$ and λ_i, the subproblem of finding x_i in the Step 5 of UAF is a standard cubic regularized Newton step [8]. We solve this subproblem to high accuracy by an implementaion [20] [8] of the Lanzcos method [9]. Furthermore, in the heuristic (6.1) for A_i, C_0 is determined by the parameters $p, \nu, q, \theta_1, \theta_2, \beta$ and γ, while we already set the values of p, ν. By the uniformly convexity of $h(x; x_0) = \frac{1}{q}\|x - x_0\|_2^q (q \geq 2)$ [27], we have $\gamma = \beta = 2^{2-q}$. We simply choose $\theta_1 = 0.5, \theta_2 = 0.67$. The Lipschitz smoothness constant L is tuned in $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$ to optimize the convergence speed in terms of run time, while the value of $h(x^*; x_0) = \frac{1}{q}\|x^* - x_0\|_2^q$ is determined by setting $x_0 = 0$ and using an approximation of x^* to replace x^*.

Under the above setting, three instances of the UAF Algorithm 5.1 with $q = 2, 2.5, 3$ respectively will be tested. The instance with $q = 3$ is equivalent to the accelerated cubic regularized Newton method (ACNM) [27]. For the instance with $q = 2$ or 2.5, we always use the heuristic (6.1) to determine the values of A_i, a_i and λ_i in each iteration.

6.2. Experiments on Large-Scale Classification Datasets. To verify the performance of the proposed UAF and the effectiveness of the heuristic (6.1) in all three instances, we consider large-scale optimization associated with the logistic regression problem as follows

$$\min_{x \in \mathbb{R}^d} \Big\{ f(x) := \frac{1}{n} \sum_{j=1}^{n} \log(1 + \exp(-\bar{b}_j \bar{a}_j^T x)) \Big\}, \tag{6.2}$$

[8] The GitHub URL: https://github.com/dalab/subsampled_cubic_regularization

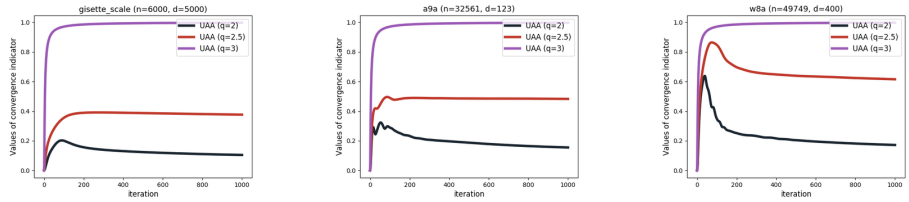

FIG. 1. *The values of the convergence indicator* (5.16) *versus the number of iterations, for the* 3 *datasets "gisette_scale", "a9a", "w8a," respectively.*

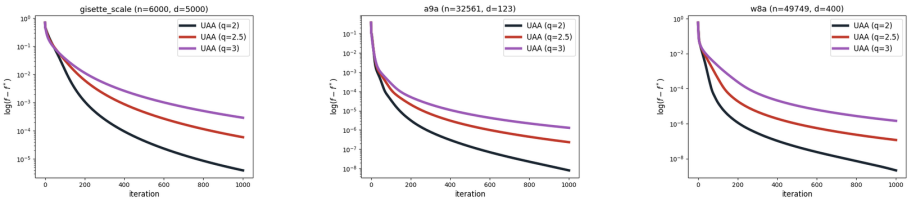

FIG. 2. *Accuracy of the objective function* (6.2) *versus the number of iterations, for the* 3 *datasets "gisette_scale", "a9a", "w8a," respectively.*

where $\{(\bar{a}_j, \bar{b}_j)\}_{j=1}^n$ denotes a dataset. (For $j \in [n], \bar{a}_j \in \mathbb{R}^d$ denotes the j-th sample and $\bar{b}_j \in \{1, -1\}$ denotes the corresponding label of $\bar{a}_j$.) In our experiments, we choose the three datasets "gisette_scale", "a9a" and "w8a" from the LIBSVM library [11] to validate the performance of our algorithm.

In Figure 1, we show the values of the convergence indicator (5.16) of UAF along the iterations. It is interesting (and somewhat surprising) to see that after several initial steps, the convergence indicator will approach to a constant in $[0, 1]$. For the case with $q = 3$, *i.e.*, the ACNM [27], the value of the indicator will approach to 1, which matches the condition (5.21) with the optimal choice $\theta_1 = \theta_2 = 1$. For the cases with $q = 2$ and 2.5, the values of the indicator will stay stable around a constant in $(0, 1)$.

Because the values of the indicators satisfy the condition (5.21) when $q = 3$ and the condition (5.25) when $q = 2$ and 2.5, the UAF algorithm will converge according to the rates in Theorems 5.7 and 5.9 respectively, which is shown in Figure 2. In Figure 2, with the heuristic (6.1), then the UAF with $q = 2$ has the fastest convergence speed, which matches the theoretical result that the setting $q = 2$ gives us the best possible iteration complexity $O\left(k^{-\frac{3(p+\nu)-2}{2}}\right)$.

An interesting phenomenon is that the speed edge for the cases $q = 2$ and 2.5 is beyond our expectation based on the bound (5.27). In the k-th iteration, from the theoretical bound in Theorems 5.7 and 5.9, the error ratio from the setting $q = 3$ to the setting $q \in [2, p + \nu)$ should be

$$O\Big(\frac{C_0}{\theta_1 c_q \gamma}\Big(\frac{k}{(p+\nu)h(x^*; x_0)}\Big)^{\frac{p+\nu-q}{q}}\Big). \tag{6.3}$$

In the experiments on all the 3 datasets, we found empirically that $h(x^*; x_0) > 1$. Meanwhile, by simple calculation, we also know that $\frac{\theta_1 c_q \gamma}{C_0} > 1$. Therefore in the 1000-th iteration, by the theoretical bound (6.3), the error ratio from $q = 3$ to 2 should not go beyond $(\frac{1000}{3})^{\frac{1}{2}} < 20$. However, in practice the ratio is beyond 100. A possible explanation for this phenomenon is that even we do not add any strongly convex regularizer in (6.2), the problem itself may have some kind of local strong convexity around the minimum point (also known as implicit regularization). Such implicit strong convexity makes the algorithms converges faster as the

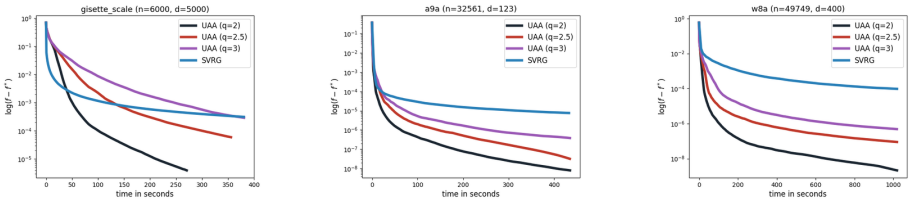

FIG. 3. *Accuracy of the objective function* (6.2) *versus algorithm run time, for the* 3 *datasets "gisette_scale", "a9a", "w8a," respectively.*

iterate approaches the minimizer.

In Figure 3, we show the performance comparison measured by error versus run time. Here we add a *stochastic variance reduction gradient* (SVRG) [19] method to show the practical efficiency of the proposed UAF algorithm. SVRG is a representative first-order algorithm for finite-sum stochastic convex optimization. The implementation of SVRG is also from the GitHub project of [20] and the learning rate of SVRG is tuned in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10, 10^2\}$.

As shown in Figure 3, SVRG can effectively exploit the finite-sum structure of the objective (6.2) and shows advantage in obtaining a low-accurate solution quickly. However, when further pursuing a high-accuracy solution, the high-order UAFs demonstrate clear edges of their faster convergence rates. In particular, with the effective heuristic (6.1), the UAF with $q = 2$ demonstrates consistent and superior performance in terms of run time behaviors.

7. Conclusions. In this paper, inspired by recent work on high-order acceleration methods, we have introduced a rather unified framework towards developing and understanding high-order acceleration algorithms for convex optimization. We show how various ideas, techniques, results, and algorithms can be derived from a simple vanilla proximal method (VPM). Based on this framework, through careful analysis, we are able to derive a unified acceleration framework (UAF) that achieves the optimal lower bounds for functions that have Hölder continuous derivatives. Our analysis and results also seem to unify many results known for the first order and high order methods, as well as results previously obtained through two separate approaches, namely the ACNM [27] and A-HPE [23] approaches. Meanwhile, the UAF is the first high-order acceleration approach that can be used in general (non-Euclidean) norm settings. Furthermore, for practical implementation of the proposed algorithm, through a new heuristic inspired from our analysis, our experiments show how the binary search procedure required by the optimal acceleration methods can be significantly simplified or forgone. This helps alleviate concerns about practical efficiency of optimal high-order acceleration methods versus suboptimal ones [31]. Finally, combined with a general *restart scheme* similar to that in [27], our analysis for the general convex setting can be easily extended to the uniformly convex setting. The resulted complexity results can match the existing lower bounds [2] in most important cases. Particularly, we shaved off the logarithmic factor of the upper bound in [2] so that matching the lower bound [2] for the σ-strongly convex minimization problem with $(2, L)$-Lipschitz continuous derivatives.[9]

Appendix A. Proofs for Section 2.

A.1. Proof of Example 2.4. By direct computation, for $x, h \in \mathbb{R}^d$, we have

$$\langle \nabla^2 f(x)h, h\rangle = \frac{1}{n}\sum_{j=1}^{n} \frac{\exp(-\bar{b}_j \bar{a}_j^T x)}{(1+\exp(-\bar{b}_j \bar{a}_j^T x))^2}\langle \bar{a}_j, h\rangle^2 \le \frac{1}{n}\sum_{j=1}^{n}\langle \bar{a}_j, h\rangle^2 = h^T B h. \tag{A.1}$$

[9] See details in the longer arXiv version: https://arxiv.org/pdf/1906.00582.pdf.

Meanwhile, we have

$$\nabla^3 f(x)[h,h,h] = \frac{1}{n}\sum_{j=1}^{n} -\frac{\exp(-\bar{b}_j\bar{a}_j^T x)(1-\exp(-\bar{b}_j\bar{a}_j^T x))}{(1+\exp(-\bar{b}_j\bar{a}_j^T x))^3}\langle \bar{b}_j\bar{a}_j, h\rangle^3 \;\le\; \frac{1}{n}\sum_{j=1}^{n}|\langle \bar{a}_j, h\rangle|^3$$

$$\le \frac{1}{n}(\sum_{j=1}^{n}\langle \bar{a}_j, h\rangle^2)\max_{j\in[n]}|\langle \bar{a}_j, h\rangle| = h^T Bh \cdot \max_{j\in[n]}|\langle \bar{a}_j, h\rangle|. \tag{A.2}$$

Therefore,

$$\|\nabla^2 f(x)\|_q = \max_{h\in\mathbb{R}^d:\|h\|_p\le 1}\langle \nabla^2 f(x)h, h\rangle \le \max_{h\in\mathbb{R}^d:\|h\|_p\le 1}\|Bh\|_q \le \|B\|_{p,q},$$

$$\|\nabla^3 f(x)\|_q = \max_{h\in\mathbb{R}^d:\|h\|_p\le 1}\nabla^3 f(x)[h,h,h] \le \|B\|_{p,q}\max_{j\in[n]}\|\bar{a}_j\|_q.$$

Let $L(\nu) := \sup_{x,y\in\mathbb{R}^d, x\neq y}\frac{\|\nabla^2 f(x)-\nabla^2 f(y)\|_q}{\|x-y\|_p^\nu}$, $\nu\in[0,1]$. Then $L(0) = \|\nabla^2 f(x)\|_q$, $L(1) = \|\nabla^3 f(x)\|_q$. Note that $L(\nu)$ is log-convex, therefore we have

$$L(\nu) \le L^{1-\nu}(0)L^{\nu}(1) \le \|B\|_{p,q}\max_{j\in[n]}\|\bar{a}_j\|_q^\nu. \tag{A.3}$$

Example 2.4 is proved.

A.2. Proof of Lemma 2.5 . By the convexity of $g(x)$, (2.9) holds trivially.

If $g(x)$ has p-th derivatives, for $i\in\{0,1,2,\ldots,p-1\}$, we define a sequence

$$C_i := \frac{1}{i!}\int_0^1 (1-\tau)^i \nabla^{i+1} g(y+\tau(x-y))[x-y]^{i+1}\mathrm{d}\tau. \tag{A.4}$$

Then one has

$$C_0 = \int_0^1 \nabla g(y+\tau(x-y))[x-y]\mathrm{d}\tau = \int_0^1 \langle \nabla g(y+\tau(x-y)), x-y\rangle \mathrm{d}\tau$$

$$= g(y+\tau(x-y))|_{\tau=0}^1 = g(x)-g(y). \tag{A.5}$$

Meanwhile,

$$C_i = \frac{1}{i!}\int_0^1 (1-\tau)^i \mathrm{d}\left(\nabla^i g(y+\tau(x-y))[x-y]^i\right)$$

$$= \frac{1}{i!}\left(\nabla^i g(y+\tau(x-y))[x-y]^i\right)(1-\tau)^i|_{\tau=0}^1$$

$$-\frac{1}{i!}\int_0^1 \left(\nabla^i g(y+\tau(x-y))[x-y]^i\right)\mathrm{d}(1-\tau)^i$$

$$= -\frac{1}{i!}\nabla^i g(y)[x-y]^i + \frac{1}{(i-1)!}\int_0^1 (1-\tau)^{i-1}\left(\nabla^i g(y+\tau(x-y))[x-y]^i\right)\mathrm{d}\tau$$

$$= -\frac{1}{i!}\nabla^i g(y)[x-y]^i + C_{i-1}. \tag{A.6}$$

Therefore by (A.5) and (A.6), one has

$$C_{p-1} = \sum_{i=1}^{p-1}(C_i - C_{i-1}) + C_0 = \sum_{i=1}^{p-1} -\frac{1}{i!}\nabla^i g(y)[x-y]^i + g(x) - g(y)$$

$$= f(x) - \tilde{f}(x;y) + \frac{1}{p!}\nabla^p g(y)[x-y]^p$$

$$= f(x) - \tilde{f}(x;y) + \frac{1}{(p-1)!}\nabla^p g(y)[x-y]^p\int_0^1 (1-\tau)^{p-1}\mathrm{d}\tau. \tag{A.7}$$

Then by (A.7), it follows that

$$
\begin{aligned}
|f(x)-\tilde{f}(x;y)| &= \Big|C_{p-1}-\frac{1}{(p-1)!}\nabla^p g(y)[x-y]^p\int_0^1(1-\tau)^{p-1}\mathrm{d}\tau\Big| \\
&= \frac{1}{(p-1)!}\Big|\int_0^1(1-\tau)^{p-1}\left(\nabla^p g(y+\tau(x-y))-\nabla^p g(y)\right)[x-y]^p\mathrm{d}\tau\Big| \\
&\le \frac{1}{(p-1)!}\int_0^1(1-\tau)^{p-1}\mathrm{d}\tau\max_{\tau\in[0,1]}\Big|\left(\nabla^p g(y+\tau(x-y))-\nabla^p g(y)\right)[x-y]^p\Big| \\
&\le \frac{1}{(p-1)!}\int_0^1(1-\tau)^{p-1}\mathrm{d}\tau\max_{\tau\in[0,1]}\|\nabla^p g(y+\tau(x-y))-\nabla^p g(y)\|_*\,\|x-y\|^p \\
&\le \frac{1}{(p-1)!}\frac{1}{p}\max_{\tau\in[0,1]}\left((p-1)!L\|\tau(x-y)\|^\nu\right)\|x-y\|^p \\
&\le \frac{L}{p}\|x-y\|^{p+\nu},
\end{aligned}
$$

Therefore (2.10) holds. Then by (A.7), by taking gradient $w.r.t.$ x, one has

$$
\begin{aligned}
\nabla C_{p-1} &= \nabla f(x)-\nabla\tilde{f}(x;y)+\frac{1}{(p-1)!}\nabla^p g(y)[x-y]^{p-1} \\
&= \nabla f(x)-\nabla\tilde{f}(x;y)+\frac{p}{(p-1)!}\nabla^p g(y)[x-y]^{p-1}\int_0^1(1-\tau)^{p-1}\mathrm{d}\tau, \qquad \text{(A.8)}
\end{aligned}
$$

while by (A.4), one also has

$$
\nabla C_{p-1}=\frac{p}{(p-1)!}\int_0^1(1-\tau)^{p-1}\nabla^p g(y+\tau(x-y))[x-y]^{p-1}\mathrm{d}\tau. \qquad \text{(A.9)}
$$

By (A.8) and (A.9), it follows that

$$
\begin{aligned}
\|\nabla f(x)-\nabla\tilde{f}(x;y)\|_* &= \Big\|\nabla C_{p-1}-\frac{p}{(p-1)!}\nabla^p g(y)[x-y]^{p-1}\int_0^1(1-\tau)^{p-1}\mathrm{d}\tau\Big\|_* \\
&= \Big\|\frac{p}{(p-1)!}\int_0^1(\nabla^p g(y+\tau(x-y))-\nabla^p g(y))[x-y]^{p-1}(1-\tau)^{p-1}\mathrm{d}\tau\Big\|_* \\
&= \max_{v:\|v\|\le1}\frac{p}{(p-1)!}\int_0^1(\nabla^p g(y+\tau(x-y))-\nabla^p g(y))[v][x-y]^{p-1}(1-\tau)^{p-1}\mathrm{d}\tau \\
&\le \frac{p}{(p-1)!}\int_0^1\max_{v:\|v\|\le1}(\nabla^p g(y+\tau(x-y))-\nabla^p g(y))[v][x-y]^{p-1}(1-\tau)^{p-1}\mathrm{d}\tau \\
&\le \frac{p}{(p-1)!}\int_0^1(1-\tau)^{p-1}\mathrm{d}\tau\max_{\tau\in[0,1]}\max_{v:\|v\|\le1}(\nabla^p g(y+\tau(x-y))-\nabla^p g(y))[v][x-y]^{p-1} \\
&\le \frac{p}{(p-1)!}\cdot\frac{1}{p}\cdot\max_{\tau\in[0,1]}\max_{v:\|v\|\le1}\|\nabla^p g(y+\tau(x-y))-\nabla^p g(y)\|_*\cdot\|v\|\cdot\|x-y\|^{p-1} \\
&= \frac{p}{(p-1)!}\cdot\frac{1}{p}\cdot\max_{\tau\in[0,1]}\|\nabla^p g(y+\tau(x-y))-\nabla^p g(y)\|_*\cdot\|x-y\|^{p-1} \\
&\le \frac{p}{(p-1)!}\cdot\frac{1}{p}\cdot\max_{\tau\in[0,1]}(p-1)!L\|\tau(x-y)\|^\nu\cdot\|x-y\|^{p-1}
\end{aligned}
$$

Then $\|\nabla f(x)-\nabla\tilde{f}(x;y)\|_*\le L\|x-y\|^{p+\nu-1}$, *i.e.*, (2.11) holds. Lemma 2.5 is proved.

A.3. Proof of Lemma 2.6. For the first statement, by the condition we have $b_k - b_{k-1} \geq C^{\frac{1}{\rho}} b_k^{\frac{\rho-1}{\rho}}$. Then by $b_0 = 0$,

$$b_k = \sum_{i=1}^{k} (b_i - b_{i-1}) \geq C^{\frac{1}{\rho}} \sum_{i=1}^{k} b_i^{\frac{\rho-1}{\rho}}.$$

Then in [6, Lemma 12], for $i \geq 1$, by setting $B_i := b_i^{\frac{\rho-1}{\rho}}$, $\alpha := \frac{\rho}{\rho-1}$, $c := C^{\frac{1}{\rho}}$, then one has

$$b_k^{\frac{\rho-1}{\rho}} = B_k \geq \Big(\frac{1}{\rho} C^{\frac{1}{\rho}} k\Big)^{\rho-1}.$$

Then after a simple rearrangement, we obtain the first statement.

For the second statement, by the reverse Hölder inequality, $\|fg\|_1 \geq \|f\|_{\frac{1}{t}} \|g\|_{-\frac{1}{t-1}}$ for $t \geq 1$ and invoking this with $t = \rho\delta + 1$ and by $b_0 = 0$, then

$$\begin{aligned}
&\sum_{i=1}^{k} \Big(\frac{b_i^{\rho-1}}{(b_i - b_{i-1})^{\rho}}\Big)^{\delta} = \sum_{i=1}^{k} b_i^{(\rho-1)\delta} (b_i - b_{i-1})^{-\rho\delta} \\
&\geq \Big(\sum_{i=1}^{k} b_i^{(\rho-1)\delta \cdot \frac{1}{t}}\Big)^{t} \Big(\sum_{i=1}^{k} (b_i - b_{i-1})^{-\rho\delta \cdot \frac{-1}{t-1}}\Big)^{-(t-1)} \\
&= \Big(\sum_{i=1}^{k} b_i^{\frac{(\rho-1)\delta}{\rho\delta+1}}\Big)^{\rho\delta+1} \Big(\sum_{i=1}^{k} (b_i - b_{i-1})\Big)^{-\rho\delta} = \Big(\sum_{i=1}^{k} b_i^{\frac{(\rho-1)\delta}{\rho\delta+1}}\Big)^{\rho\delta+1} b_k^{-\rho\delta}.
\end{aligned}$$

Then by the corresponding condition, we have $b_k^{\frac{\rho\delta}{\rho\delta+1}} \geq C^{-\frac{1}{\rho\delta+1}} \Big(\sum_{i=1}^{k} b_i^{\frac{(\rho-1)\delta}{\rho\delta+1}}\Big)$. Then in [6, Lemma 12], for $i \geq 1$, by setting $B_i := b_i^{\frac{(\rho-1)\delta}{\rho\delta+1}}$, $\alpha := \frac{\rho}{\rho-1}$, $c := C^{-\frac{1}{\rho\delta+1}}$, one has

$$b_k^{\frac{(\rho-1)\delta}{\rho\delta+1}} = B_k \geq \Big(\frac{1}{\rho} C^{-\frac{1}{\rho\delta+1}} k\Big)^{\rho-1}.$$

Then after a simple rearrangement, we obtain the second statement. Lemma 2.6 is proved.

Appendix B. Proofs for Section 4.

B.1. Proof of Lemma 4.2. By the optimality condition of $z_t := \operatorname{argmin}_{x \in \mathbb{R}^d} \psi_t^{\mathrm{cont}}(x)$, one has $\Big\langle \int_0^t a_\tau \nabla \hat{f}(z_t; x_\tau) \mathrm{d}\tau + \nabla h(z_t; x_0), \dot{z}_t \Big\rangle \geq 0$. It follows that

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}t}\Big(\min_{x \in \mathbb{R}^d} \psi_t^{\mathrm{cont}}(x)\Big) = \frac{\mathrm{d}}{\mathrm{d}t} \psi_t^{\mathrm{cont}}(z_t) = \frac{\mathrm{d}}{\mathrm{d}t}\Big(\int_0^t a_\tau \hat{f}(z_t; x_\tau) \mathrm{d}\tau + h(z_t; x_0)\Big) \\
&= a_t \hat{f}(z_t; x_t) + \Big\langle \int_0^t a_\tau \nabla \hat{f}(z_t; x_\tau) \mathrm{d}\tau + \nabla h(z_t; x_0), \dot{z}_t \Big\rangle \\
&\geq a_t (\hat{f}(x_t; x_t) + \langle \nabla \hat{f}(x_t; x_t), z_t - x_t \rangle) + \Big\langle \int_0^t a_\tau \nabla \hat{f}(z_t; x_\tau) \mathrm{d}\tau + \nabla h(z_t; x_0), \dot{z}_t \Big\rangle \\
&\overset{①}{=} a_t (f(x_t) + \langle \nabla f(x_t), z_t - x_t \rangle) + \Big\langle \int_0^t a_\tau \nabla \hat{f}(z_t; x_\tau) \mathrm{d}\tau + \nabla h(z_t; x_0), \dot{z}_t \Big\rangle \\
&\geq a_t (f(x_t) + \langle \nabla f(x_t), z_t - x_t \rangle), \qquad \text{(B.1)}
\end{aligned}$$

where ① is by the definition of $\hat{f}(x; y)$ in (2.7). Furthermore, one has

$$\frac{\mathrm{d}(A_t f(x_t))}{\mathrm{d}t} = a_t f(x_t) + A_t \langle \nabla f(x_t), \dot{x}_t \rangle. \qquad \text{(B.2)}$$

By Combing (B.1) and (B.2), one has

$$\frac{\mathrm{d}}{\mathrm{d}t}\Big(A_t f(x_t) - \min_{x\in\mathbb{R}^d}\psi_t^{\mathrm{cont}}(x)\Big) \le \langle \nabla f(x_t), A_t\dot{x}_t - a_t(z_t - x_t)\rangle. \tag{B.3}$$

Meanwhile by $A_0 = 0$ and $\min_{x\in\mathbb{R}^d}\psi_0(x) = 0$, one has $A_0 f(x_0) - \min_{x\in\mathbb{R}^d}\psi_0^{\mathrm{cont}}(x) = 0$.

Taking integral from $\tau = 0$ to t for (B.3), then Lemma 4.2 is proved.

Appendix C. Proofs for Section 5.

C.1. Proof of Lemma 5.3.

First, in (5.1), by $A_0 = 0$ and $z_0 = x_0$, we have

$$A_0 f(x_0) - \psi_0^{\mathrm{dis}}(z_0) = 0. \tag{C.1}$$

Then by our assumption, $\hat{f}(x; x_i)$ is convex $w.r.t.\ \|\cdot\|$ and $h(x; x_0)$ is (q,γ)-uniformly convex $w.r.t.\ \|\cdot\|$. Therefore for all $x, y \in \mathbb{R}^d$, it follows that

$$\psi_i^{\mathrm{dis}}(x) \ge \psi_i^{\mathrm{dis}}(y) + \langle \nabla\psi_i^{\mathrm{dis}}(y), x - y\rangle + \frac{\gamma}{q}\|x - y\|^q. \tag{C.2}$$

Then by the optimality condition of z_i, it follows that for all $x \in \mathbb{R}^d$, $\langle \nabla\psi_i^{\mathrm{dis}}(z_i), x - z_i\rangle \ge 0$. Therefore, it follows that $\psi_i^{\mathrm{dis}}(x) \ge \psi_i^{\mathrm{dis}}(z_i) + \frac{\gamma}{q}\|x - z_i\|^q$. Therefore we have,

$$\psi_i^{\mathrm{dis}}(x) = \psi_{i-1}^{\mathrm{dis}}(x) + a_i\hat{f}(x; x_i) \ge \psi_{i-1}^{\mathrm{dis}}(z_{i-1}) + \frac{\gamma}{q}\|x - z_{i-1}\|^q + a_i\hat{f}(x; x_i). \tag{C.3}$$

Meanwhile, we can lower bound the last term of RHS of (C.3).

$$\begin{aligned}
a_i\hat{f}(x; x_i) &\overset{①}{\ge} a_i(\hat{f}(x_i; x_i) + \langle \nabla\hat{f}(x_i; x_i), x - x_i\rangle) \overset{②}{=} a_i(f(x_i) + \langle \nabla f(x_i), x - x_i\rangle)\\
&\overset{③}{=} A_i(f(x_i) + \langle \nabla f(x_i), x - x_i\rangle) - A_{i-1}(f(x_i) + \langle \nabla f(x_i), x - x_i\rangle)\\
&= A_i\left(f(x_i) + \left\langle \nabla f(x_i), \frac{a_i}{A_i}x + \frac{A_{i-1}}{A_i}x_{i-1} - x_i\right\rangle\right)\\
&\quad -A_{i-1}(f(x_i) + \langle \nabla f(x_i), x_{i-1} - x_i\rangle)\\
&\overset{④}{\ge} A_i\left(f(x_i) + \left\langle \nabla f(x_i), \frac{a_i}{A_i}x + \frac{A_{i-1}}{A_i}x_{i-1} - x_i\right\rangle\right) - A_{i-1}f(x_{i-1})\\
&= A_i f(x_i) - A_{i-1}f(x_{i-1}) + A_i\left\langle \nabla f(x_i), \frac{a_i}{A_i}x + \frac{A_{i-1}}{A_i}x_{i-1} - x_i\right\rangle,
\end{aligned}$$

where ① is by the convexity of $\hat{f}(x; y)$ $w.r.t.$ x, ② is by the definition of $\hat{f}(x; y)$ in (2.7), ③ is by the identity $a_i = A_i - A_{i-1}$, and ④ is by the convexity of $f(x)$.

Therefore it follows that

$$\begin{aligned}
\psi_i^{\mathrm{dis}}(x) \ge \psi_{i-1}^{\mathrm{dis}}(z_{i-1}) + \frac{\gamma}{q}\|x - z_{i-1}\|^q + A_i f(x_i) - A_{i-1}f(x_{i-1})\\
+A_i\left\langle \nabla f(x_i), \frac{a_i}{A_i}x + \frac{A_{i-1}}{A_i}x_{i-1} - x_i\right\rangle.
\end{aligned} \tag{C.4}$$

By setting $x := z_i$ and a simple arrangement of (C.4), we have

$$\begin{aligned}
&(A_i f(x_i) - \psi_i^{\mathrm{dis}}(z_i)) - (A_{i-1}f(x_{i-1}) - \psi_{i-1}^{\mathrm{dis}}(z_{i-1}))\\
&\le A_i\left\langle \nabla f(x_i), x_i - \frac{a_i}{A_i}z_i - \frac{A_{i-1}}{A_i}x_{i-1}\right\rangle - \frac{\gamma}{q}\|z_i - z_{i-1}\|^q
\end{aligned} \tag{C.5}$$

Summing (C.5) from $i=0$ to $k-1$ and by (C.1), it follows that

$$\begin{aligned} A_k f(x_k) - \psi_k^{\mathrm{dis}}(z_k) &\le A_0 f(x_0) - \psi_0^{\mathrm{dis}}(z_0) \\ &\quad + \sum_{i=1}^{k} \left(A_i \left\langle \nabla f(x_i), x_i - \frac{a_i}{A_i} z_i - \frac{A_{i-1}}{A_i} x_{i-1} \right\rangle - \frac{\gamma}{q} \|z_i - z_{i-1}\|^q \right) \\ &= \sum_{i=1}^{k} \left(A_i \left\langle \nabla f(x_i), x_i - \frac{a_i}{A_i} z_i - \frac{A_{i-1}}{A_i} x_{i-1} \right\rangle - \frac{\gamma}{q} \|z_i - z_{i-1}\|^q \right). \end{aligned}$$

Then by the definition of E_i, Lemma 5.3 is proved.

C.2. Proof of Lemma 5.4. By the definition of E_i, one has

$$\begin{aligned} E_i &\overset{①}{\le} a_i \langle \nabla f(x_i), \hat{z}_i - z_i \rangle - \frac{\gamma}{q} \|z_i - z_{i-1}\|^q \overset{②}{\le} a_i \langle \nabla f(x_i), \hat{z}_i - z_i \rangle - \frac{\gamma_i'}{q} \|z_i - z_{i-1}\|^q \\ &\overset{③}{\le} \left\langle a_i \nabla f(x_i) + \gamma_i' \nabla \frac{1}{q} \|\hat{z}_i - z_{i-1}\|^q, \hat{z}_i - z_i \right\rangle - \gamma_i' \left(\frac{1}{q} \|\hat{z}_i - z_{i-1}\|^q + \frac{\beta}{q} \|\hat{z}_i - z_i\|^q \right) \\ &\overset{④}{=} \left\langle a_i \nabla f(x_i) + \frac{\gamma_i' A_i^{q-1}}{a_i^{q-1}} \nabla \frac{1}{q} \|x_i - \hat{x}_{i-1}\|^q, \hat{z}_i - z_i \right\rangle - \gamma_i' \Big(\frac{A_i^q}{q a_i^q} \|x_i - \hat{x}_{i-1}\|^q + \frac{\beta}{q} \|\hat{z}_i - z_i\|^q \Big) \end{aligned}$$

where ① is (5.3), ② is by $\gamma \ge \gamma_i'$, ③ is by Assumption 5.1, ④ is (5.3) such that $\nabla \|\hat{z}_i - z_i\|^q = \frac{A_i^{q-1}}{a_i^{q-1}} \nabla \|x_i - \hat{x}_{i-1}\|^q$. Therefore Lemma 5.4 is proved.

C.3. Proof of Lemma 5.5. By Lemma 5.4, one has

$$\begin{aligned} E_i &\le a_i \Big\langle \nabla f(x_i) + \frac{\gamma_i' A_i^{q-1}}{a_i^q} \nabla \frac{1}{q} \|x_i - \hat{x}_{i-1}\|^q, \hat{z}_i - z_i \Big\rangle - \gamma_i' \Big(\frac{A_i^q}{q a_i^q} \|x_i - \hat{x}_{i-1}\|^q + \frac{\beta}{q} \|\hat{z}_i - z_i\|^q \Big) \\ &\le a_i \langle \nabla f(x_i) - \nabla \tilde{f}(x_i; \hat{x}_{i-1}), \hat{z}_i - z_i \rangle + a_i \Big\langle \nabla \tilde{f}(x_i; \hat{x}_{i-1}) + \frac{\gamma_i' A_i^{q-1}}{a_i^q} \nabla \frac{1}{q} \|x_i - \hat{x}_{i-1}\|^q, \hat{z}_i - z_i \Big\rangle \\ &\quad - \gamma_i' \left(\frac{A_i^q}{q a_i^q} \|x_i - \hat{x}_{i-1}\|^q + \frac{\beta}{q} \|\hat{z}_i - z_i\|^q \right). \end{aligned} \tag{C.6}$$

Meanwhile, it follows that

$$\begin{aligned} &a_i \langle \nabla f(x_i) - \nabla \tilde{f}(x_i; \hat{x}_{i-1}), \hat{z}_i - z_i \rangle - \gamma_i' \Big(\frac{A_i^q}{q a_i^q} \|x_i - \hat{x}_{i-1}\|^q + \frac{\beta}{q} \|\hat{z}_i - z_i\|^q \Big) \\ &\le a_i \|\nabla f(x_i) - \nabla \tilde{f}(x_i; \hat{x}_{i-1})\|_* \|\hat{z}_i - z_i\| - \gamma_i' \Big(\frac{A_i^q}{q a_i^q} \|x_i - \hat{x}_{i-1}\|^q + \frac{\beta}{q} \|\hat{z}_i - z_i\|^q \Big) \\ &\overset{①}{\le} a_i L \|x_i - \hat{x}_{i-1}\|^{p+\nu-1} \|\hat{z}_i - z_i\| - \gamma_i' \Big(\frac{A_i^q}{q a_i^q} \|x_i - \hat{x}_{i-1}\|^q + \frac{\beta}{q} \|\hat{z}_i - z_i\|^q \Big) \\ &\overset{②}{\le} \frac{q-1}{q} (\beta \gamma_i')^{-\frac{1}{q-1}} (a_i L)^{\frac{q}{q-1}} \|x_i - \hat{x}_{i-1}\|^{\frac{q(p+\nu-1)}{q-1}} - \frac{\gamma_i' A_i^q}{q a_i^q} \|x_i - \hat{x}_{i-1}\|^q \\ &\overset{③}{=} \left(\Big(L \frac{a_i^q}{c_q \gamma_i' A_i^{q-1}} \|x_i - \hat{x}_{i-1}\|^{p+\nu-q} \Big)^{\frac{q}{q-1}} - 1 \right) \frac{\gamma_i' A_i^q}{q a_i^q} \|x_i - \hat{x}_{i-1}\|^q \\ &\overset{④}{=} \left(\big(L \lambda_i' \|x_i - \hat{x}_{i-1}\|^{p+\nu-q} \big)^{\frac{q}{q-1}} - 1 \right) \frac{\gamma_i' A_i^q}{q a_i^q} \|x_i - \hat{x}_{i-1}\|^q, \end{aligned} \tag{C.7}$$

where ① is by (5.6), ② is by [27, Lemma 1.3], ③ is by a simple rearrangement and the definition of c_q in Lemma 5.5, and ④ is by the definition of λ_i'.

Combing (C.6) and (C.7), Lemma 5.5 is proved.

C.4. Proof of Theorem 5.6. First, for $i \geq 1$, if the conditions (5.15) and (5.16) are true, then one can know that (5.8) and (5.9) are true and thus for $i \geq 1$, $E_i \leq 0$. Then by Lemma 5.3, one has $A_k f(x_k) - \psi_k^{\mathrm{dis}}(z_k) \leq \sum_{i=1}^k E_i \leq 0$. Then combing Lemma 5.2, one has

$$A_k f(x_k) \leq \psi_k^{\mathrm{dis}}(z_k) \leq A_k f(x^*) + h(x^*; x_0). \tag{C.8}$$

Theorem 5.6 is proved.

C.5. Proof of Theorem 5.7. First, by our assumption, $\{\lambda_i\}$ defined in (5.11) satisfies (5.21), therefore $\{\lambda_i\}$ satisfies (5.16); meanwhile $\{x_i\}$ satisfies (5.15). Therefore Theorem 5.6 holds, *i.e.*, $f(x_k) - f(x^*) \leq \frac{h(x^*;x_0)}{A_k}$. Then by (5.21), because $L\lambda_i = \frac{La_i^q}{c_q \gamma A_i^{q-1}} = \frac{L(A_i - A_{i-1})^q}{c_q \gamma A_i^{q-1}} \geq \theta_1$, in Lemma 2.6, by setting $b_i := A_i, \rho := p + \nu, C := \frac{\theta_1 c_q \gamma}{L}$, we can obtain the lower bound $A_k \geq \frac{\theta_1 c_q \gamma}{L} \left(\frac{k}{p+\nu}\right)^{p+\nu}$. Finally, (5.23) is obtained and Theorem 5.7 is proved.

C.6. Proof of Lemma 5.8. When (5.15) and (5.16) are satisfied, by Lemma 5.5, we have

$$\begin{aligned}
E_i &\overset{①}{\leq} (\theta_2^{\frac{q}{q-1}} - 1)\frac{\gamma_i' A_i^q}{q a_i^q}\|x_i - \hat{x}_{i-1}\|^q \overset{②}{\leq} (\theta_2^{\frac{q}{q-1}} - 1)\frac{A_i^q}{q a_i^q}\left(\frac{\omega_i}{\theta_2}\right)^{\alpha} \gamma \|x_i - \hat{x}_{i-1}\|^q \\
&\overset{③}{=} (\theta_2^{\frac{q}{q-1}} - 1)\frac{A_i^q}{q a_i^q}\left(\frac{\omega_i}{\theta_2}\right)^{\alpha} \gamma (L\lambda_i \|x_i - \hat{x}_{i-1}\|^{p+\nu-q})^{\frac{q}{p+\nu-q}} (L\lambda_i)^{-\frac{q}{p+\nu-q}} \\
&\overset{④}{=} \frac{1}{q\theta_2^{\alpha}}(\theta_2^{\frac{q}{q-1}} - 1)\frac{A_i^q}{a_i^q}\omega_i^{\frac{\varsigma}{p+\nu-q}} \gamma (L\lambda_i)^{-\frac{q}{p+\nu-q}} \\
&\overset{⑤}{=} \frac{1}{q\theta_2^{\alpha}}(\theta_2^{\frac{q}{q-1}} - 1)\omega_i^{\frac{\varsigma}{p+\nu-q}} \gamma \left(\frac{c_q \gamma A_i^{p+\nu-1}}{L(A_i - A_{i-1})^{p+\nu}}\right)^{\frac{q}{p+\nu-q}},
\end{aligned} \tag{C.9}$$

where ① is by (5.15) and (5.16), ② is by the value of γ_i' in (5.12) and the definition of ω_i in Lemma 5.8, ③ is by a simple rearrangement, ④ is by definition of ω_i and $\varsigma = \alpha(p+\nu) + (1-\alpha)q$, ⑤ is by definition of λ_i in (5.11) and the fact $a_i = A_i - A_{i-1}$.

Then by combing Lemmas 5.2 and 5.3, it follows that

$$A_k f(x_k) \leq \psi_k^{\mathrm{dis}}(z_k) + \sum_{i=1}^k E_i \leq A_k f(x^*) + h(x^*; x_0) + \sum_{i=1}^k E_i. \tag{C.10}$$

Then by combing (C.9) and (C.10), and $f(x_k) \geq f(x^*)$, $A_k \geq 0$, one has

$$\frac{1}{q\theta_2^{\alpha}}(1 - \theta_2^{\frac{q}{q-1}})\gamma \sum_{i=1}^k \omega_i^{\frac{\varsigma}{p+\nu-q}} \left(\frac{c_q \gamma A_i^{p+\nu-1}}{L(A_i - A_{i-1})^{p+\nu}}\right)^{\frac{q}{p+\nu-q}} \leq \sum_{i=1}^k -E_i \leq h(x^*; x_0).$$

Then after a simple rearrangement, we have Lemma 5.8.

C.7. Proof of Theorem 5.9. First, by our assumption, $\{\lambda_i\}$ defined in (5.11) satisfies (5.25), therefore $\{\lambda_i\}$ satisfies (5.16); meanwhile $\{x_i\}$ satisfies (5.15). Therefore Theorem 5.6 holds, *i.e.*,

$$f(x_k) - f(x^*) \leq \frac{h(x^*; x_0)}{A_k}. \tag{C.11}$$

Then by Lemma 5.8 and the assumption that $\omega_i \geq \theta_1$, we have

$$\sum_{i=1}^{k} \Big(\frac{A_i^{p+\nu-1}}{(A_i - A_{i-1})^{p+\nu}} \Big)^{\frac{q}{p+\nu-q}} \leq (C_0^{-1} L)^{\frac{q}{p+\nu-q}} h(x^*; x_0), \tag{C.12}$$

where C_0 is defined in Theorem 5.9.

In Lemma 2.6, for $1 \leq i \leq k$, by setting $b_i := A_i, \rho := p + \nu, \delta := \frac{q}{p+\nu-q}, C := (C_0^{-1} L)^{\frac{q}{p+\nu-q}} h(x^*; x_0)$, then we obtain the lower bound

$$A_k \geq \frac{C_0}{L} \left(h(x^*; x_0)\right)^{-\frac{p+\nu-q}{q}} \left(\frac{k}{p+\nu} \right)^{\frac{(q+1)(p+\nu)-q}{q}}. \tag{C.13}$$

Then combing (C.11), we obtain (5.27).

C.8. Proof of Proposition 5.10. First by our assumption about $\|\cdot\|$ and $\tilde{f}(x; y)$, (5.31) is a strictly convex function, therefore $w(\upsilon)$ is a continuous function of υ. Meanwhile $x(\lambda)$ is continuous about λ. Therefore $\chi(\lambda)$ is continuous $w.r.t.\ \lambda$.

Next by the fact $\varsigma = \alpha(p + \nu) + (1 - \alpha)q \in [q, p + \nu]$ and

$$\tilde{f}(z; \upsilon) + \frac{L^\alpha}{c_q \lambda^{(1-\alpha)} \theta_2^\alpha \varsigma} \|z - \upsilon\|^\varsigma \leq \tilde{f}(\upsilon; \upsilon) = f(\upsilon) < +\infty, \tag{C.14}$$

as $\lambda \to 0$, $\|z - \upsilon\| \to 0$ if $\varsigma \in [q, p + \nu)$ or $\|z - \upsilon\|$ is a finite value if $\varsigma = p + \nu$. In both cases, we have $\chi(0) = 0$. Then since $f(\upsilon) \neq f(x^*)$, we will also have as $\lambda \to +\infty$, it is easy to find that $\frac{a(\lambda)}{A+a(\lambda)} \to 1$ and thus $x(\lambda) = x$. Since $f(x) \neq f(x^*)$, we have $\omega(x) \neq x$. Therefore $\chi(+\infty) = +\infty$.

REFERENCES

[1] Z. Allen-Zhu, *Katyusha: The first direct acceleration of stochastic gradient methods*, The Journal of Machine Learning Research, 18 (2017), pp. 8194–8244.

[2] Y. Arjevani, O. Shamir, and R. Shiff, *Oracle complexity of second-order methods for smooth convex optimization*, Math. Program., (2017), pp. 1–34.

[3] M. Baes, *Estimate sequence methods: extensions and approximations*, manuscript, (2009).

[4] K. Ball, E. A. Carlen, and E. H. Lieb, *Sharp uniform convexity and smoothness inequalities for trace norms*, Inventiones mathematicae, 115 (1994), pp. 463–482.

[5] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM journal on imaging sciences, 2 (2009), pp. 183–202.

[6] S. Bubeck, Q. Jiang, Y. T. Lee, Y. Li, and A. Sidford, *Near-optimal method for highly smooth convex optimization*, in Proceedings of the Thirty-Second Conference on Learning Theory, vol. 99 of Proc. Mach. Learn. Res, Phoenix, USA, 2019, pp. 492–507, http://proceedings.mlr.press/v99/bubeck19a.html.

[7] B. Bullins, *Fast minimization of structured convex quartics*, arXiv preprint arXiv:1812.10349, (2018).

[8] Y. Carmon and J. Duchi, *Gradient descent finds the cubic-regularized nonconvex newton step*, SIAM J. Optim., 29 (2019), pp. 2146–2178, https://doi.org/10.1137/17M1113898.

[9] Y. Carmon and J. C. Duchi, *Analysis of krylov subspace solutions of regularized nonconvex quadratic problems*, in Proceedings of the 32Nd International Conference on Neural Information Processing Systems (NeurIPS), Montrèal, Canada, 2018, pp. 10728–10738, http://dl.acm.org/citation.cfm?id=3327546.3327730.

[10] C. Cartis, N. I. Gould, and P. L. Toint, *Universal regularization methods: varying the power, the smoothness and the accuracy*, SIAM J. Optim., 29 (2019), pp. 595–615.

[11] C.-C. Chang and C.-J. Lin, *LIBSVM: A library for support vector machines*, ACM T. Intel. Syst. Tec., 2 (2011), pp. 27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[12] J. Diakonikolas and L. Orecchia, *Accelerated extra-gradient descent: A novel accelerated first-order method*, in 9th Innovations in Theoretical Computer Science Conference (ITCS 2018), Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[13] J. DIAKONIKOLAS AND L. ORECCHIA, *The approximate duality gap technique: A unified theory of first-order methods*, SIAM J. Optim., 29 (2019), pp. 660–689.

[14] A. GASNIKOV, P. DVURECHENSKY, E. GORBUNOV, E. VORONTSOVA, D. SELIKHANOVYCH, AND C. A. URIBE, *Optimal tensor methods in smooth convex and uniformly convexoptimization*, in Proceedings of the Thirty-Second Conference on Learning Theory, vol. 99 of Proc. Mach. Learn. Res., Phoenix, USA, 2019, PMLR, pp. 1374–1391.

[15] G. N. GRAPIGLIA AND Y. NESTEROV, *Accelerated regularized newton methods for minimizing composite convex functions*, SIAM J. Optim., 29 (2019), pp. 77–99.

[16] G. N. GRAPIGLIA AND Y. NESTEROV, *Tensor methods for minimizing functions with Hölder continuous higher-order derivatives*, arXiv preprint arXiv:1904.12559, (2019).

[17] HAIRER AND PETERS, *Solving ordinary differential equations I*, Springer Berlin Heidelberg, 1987.

[18] B. JIANG, H. WANG, AND S. ZHANG, *An optimal high-order tensor method for convex optimization*, in Proceedings of the Thirty-Second Conference on Learning Theory, vol. 99 of Proc. Mach. Learn. Res., Phoenix, USA, 2019, PMLR, pp. 1799–1801, http://proceedings.mlr.press/v99/jiang19a.html.

[19] R. JOHNSON AND T. ZHANG, *Accelerating stochastic gradient descent using predictive variance reduction*, in Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1, Lake Tahoe, USA, 2013, pp. 315–323, http://dl.acm.org/citation.cfm?id=2999611.2999647.

[20] J. M. KOHLER AND A. LUCCHI, *Sub-sampled cubic regularization for non-convex optimization*, in Proceedings of the 34th International Conference on Machine Learning (ICML), vol. 70, 2017, pp. 1895–1904.

[21] W. KRICHENE, A. BAYEN, AND P. L. BARTLETT, *Accelerated mirror descent in continuous and discrete time*, in Advances in neural information processing systems, 2015, pp. 2845–2853.

[22] W. KRICHENE, A. BAYEN, AND P. L. BARTLETT, *Adaptive averaging in accelerated descent dynamics*, in Advances in Neural Information Processing Systems (NeurIPS), 2016, pp. 2991–2999.

[23] R. D. MONTEIRO AND B. F. SVAITER, *An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods*, SIAM J. Optim., 23 (2013), pp. 1092–1125.

[24] A. S. NEMIROVSKII AND Y. NESTEROV, *Optimal methods of smooth convex minimization*, USSR Computational Mathematics and Mathematical Physics, 25 (1985), pp. 21–30.

[25] A. S. NEMIROVSKY AND D. B. YUDIN, *Problem complexity and method efficiency in optimization.*, (1983).

[26] Y. NESTEROV, *A method for unconstrained convex minimization problem with the rate of convergence o(1/kˆ2)*, in Doklady AN USSR, vol. 269, 1983, pp. 543–547.

[27] Y. NESTEROV, *Accelerating the cubic regularization of Newton's method on convex problems*, Math. Program., 112 (2008), pp. 159–181.

[28] Y. NESTEROV, *Introductory Lectures on Convex Optimization: A Basic Course*, Springer Publishing Company, Incorporated, 1 ed., 2014.

[29] Y. NESTEROV, *Universal gradient methods for convex optimization problems*, Math. Program., 152 (2015), pp. 381–404.

[30] Y. NESTEROV, *Complexity bounds for primal-dual methods minimizing the model of objective function*, Math. Program., 171 (2018), pp. 311–330.

[31] Y. NESTEROV, *Implementable tensor methods in unconstrained convex optimization*, Universite catholique de Louvain, Center for Operations Research and Econometrics (CORE), (2018).

[32] Y. NESTEROV AND B. T. POLYAK, *Cubic regularization of Newton method and its global performance*, Math. Program., 108 (2006), pp. 177–205.

[33] N. PARIKH, S. BOYD, ET AL., *Proximal algorithms*, Foundations and Trends® in Optimization, 1 (2014), pp. 127–239.

[34] V. ROULET AND A. D'ASPREMONT, *Sharpness, restart and acceleration*, in Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS), Long Beach, USA, 2017, pp. 1119–1129, http://dl.acm.org/citation.cfm?id=3294771.3294878.

[35] W. SU, S. BOYD, AND E. J. CANDÈS, *A differential equation for modeling nesterov's accelerated gradient method: Theory and insights*, in Proceedings of the 27th International Conference on Neural Information Processing Systems (NeurIPS), Montreal, Canada, 2014, pp. 2510–2518, http://dl.acm.org/citation.cfm?id=2969033.2969107.

[36] A. WIBISONO, A. C. WILSON, AND M. I. JORDAN, *A variational perspective on accelerated methods in optimization*, Proc. Natl. Acad. Sci., 113 (2016), pp. E7351–E7358.

[37] A. YURTSEVER, Q. TRAN-DINH, AND V. CEVHER, *A universal primal-dual convex optimization framework*, in Proceedings of the 28th International Conference on Neural Information Processing Systems (NeurIPS), Montreal, Canada, 2015, pp. 3150–3158, http://dl.acm.org/citation.cfm?id=2969442.2969591.