



A HYBRID ADAPTIVE RESTART FRAMEWORK FOR ACCELERATED INEXACT PROXIMAL GRADIENT METHODS IN NON-CONVEX OPTIMIZATION

NONTAWAT EIAMNIRAN¹, PRASIT CHOLAMJIAK^{1,*}, AND JEN-CHIH YAO²

¹*School of Science, University of Phayao, Phayao 56000, Thailand*

²*Research Center for Interneural Computing, China Medical University Hospital, China Medical University, Taichung, Taiwan*

ABSTRACT. Accelerated inexact proximal-gradient methods are effective for non-convex composite optimization, but combining adaptive step sizes with acceleration can lead to numerical instability. We propose a Hybrid Adaptive Accelerated Inexact Proximal Gradient (HA-AIPG) framework designed to balance exploratory steps with convergence safeguards. At each iteration, an adaptive spectral trial step is evaluated alongside a conservative, fixed-step reference proximal point. A relaxed combination of both updates is accepted only if it satisfies a sufficient objective decrease condition relative to the current iterate and reference update; otherwise, the algorithm defaults to the reference step and resets momentum. This structure allows trial steps to adapt to local curvature during transient phases while ensuring that global descent guarantees are preserved by the reference point. Under summable inexactness conditions, every limit point of the sequence is proved to be stationary. Under additional Kurdyka–Łojasiewicz (KL) assumptions, full sequence convergence and local rate bounds are established. Numerical experiments illustrate the practical performance of HA-AIPG, demonstrating competitive objective reduction across non-convex benchmark problems.

Keywords. Non-convex composite optimization, inexact proximal gradient method, accelerated proximal-gradient method, Barzilai-Borwein step, adaptive restart, Kurdyka–Łojasiewicz property.

© Journal of Decision Making and Healthcare

1. INTRODUCTION

Composite optimization problems arise in machine learning, signal processing, and statistics [14, 13]. Many such problems can be written as

$$\min_{x \in \mathbb{R}^N} F(x) := g(x) + h(x), \quad (1.1)$$

where $g : \mathbb{R}^N \rightarrow \mathbb{R}$ is continuously differentiable and may be non-convex, while $h : \mathbb{R}^N \rightarrow \mathbb{R} \cup \{+\infty\}$ is proper, lower semi-continuous, and convex. In learning and estimation problems, g usually represents a loss or data-fitting term, whereas h is a regularizer or constraint that promotes a desired structure, such as sparsity or low rank [23, 7].

Although much of classical theory concerns convex objectives [26, 4], many applications lead to a non-convex smooth term. One example is the correntropy-induced loss used in robust estimation [10]. In this study, the possible non-convexity is confined to g , whereas h remains convex. This assumption covers several non-convex composite models considered in recent proximal gradient studies [16, 30].

*Corresponding author.

E-mail address: nontawatskip@gmail.com (N. Eiamniran), prasit.ch@up.ac.th (P. Cholamjiak), and yaojc@mail.cmu.edu.tw (J.-C. Yao)

Received: August 28, 2026; Revised: August 30, 2026; Accepted: September 6, 2026.

2020 Mathematics Subject Classification: 90C26, 90C30, 65K05.

Without convexity, a first-order method cannot generally guarantee a global minimizer. The usual objective is to construct a sequence whose accumulation points satisfy an appropriate stationarity condition [1, 14]. This makes the control of the step size, extrapolation, and proximal accuracy important components.

Proximal-gradient (PG) methods are a natural approach to the composite problem above [4, 23]. For $\gamma > 0$, the proximal operator of h is defined by

$$\text{prox}_{\gamma h}(y) := \arg \min_{x \in \mathbb{R}^N} \left\{ h(x) + \frac{1}{2\gamma} \|x - y\|^2 \right\}.$$

The standard PG iteration is then

$$x_{k+1} = \text{prox}_{\gamma_k h}(x_k - \gamma_k \nabla g(x_k)),$$

where $\gamma_k > 0$ is the step size. Under the usual convexity and smoothness assumptions, the PG method has an objective value convergence rate of $O(1/k)$ [4].

Accelerated proximal-gradient methods introduce an extrapolated point before the proximal gradient update. A standard FISTA-type extrapolation has the form [19, 4]

$$y_k = x_k + \frac{t_{k-1} - 1}{t_k} (x_k - x_{k-1}),$$

followed by

$$x_{k+1} = \text{prox}_{\gamma_k h}(y_k - \gamma_k \nabla g(y_k)).$$

For convex composite problems, this construction improves the objective value rate to $O(1/k^2)$ under standard assumptions [4]. In the non-convex case, however, the extrapolated point y_k need not satisfy a descent relation. Consequently, accepting every accelerated update may produce non-monotonic behavior or unstable iterations [11, 14, 18].

Therefore, several non-convex accelerated proximal-gradient methods supplement extrapolation with candidate monitoring, non-monotone acceptance conditions, local curvature estimates, or restart mechanisms [14, 16, 18]. These developments motivate the monitored, inexact, and adaptive proximal gradient constructions discussed next.

One approach to controlling non-convex extrapolation is to compare the accelerated candidate with a proximal gradient point computed from the currently accepted iterate. Li and Lin introduced this strategy in their monotone APG method [14]. In the notation used above, the extrapolated point is

$$y_k = x_k + \frac{t_{k-1}}{t_k} (z_k - x_k) + \frac{t_{k-1} - 1}{t_k} (x_k - x_{k-1}).$$

The accelerated candidate and proximal gradient monitor are then computed separately as follows:

$$z_{k+1} = \text{prox}_{\alpha_y h}(y_k - \alpha_y \nabla g(y_k)),$$

and

$$v_{k+1} = \text{prox}_{\alpha_x h}(x_k - \alpha_x \nabla g(x_k)).$$

The next iterate is selected according to

$$x_{k+1} = \begin{cases} z_{k+1}, & F(z_{k+1}) \leq F(v_{k+1}), \\ v_{k+1}, & \text{otherwise.} \end{cases} \quad (1.2)$$

Point v_{k+1} provides a proximal gradient fallback when an extrapolated update is unfavorable. In particular, the selection rule ensures that the accepted objective value is not larger than that obtained from the monitor. Under the assumptions and step-size conditions in [14], this construction provides the descent properties required for the convergence to a critical point. Its main computational cost is that the monotone version evaluates two proximal mappings at each iteration.

When the proximal subproblem has no closed-form solution, computing either candidate exactly may require an iterative inner solver. Inexact proximal-gradient methods address this issue by allowing the proximal subproblem to be solved only with a prescribed accuracy [26, 28]. For $\gamma > 0$ and $\epsilon \geq 0$, an inexact proximal point can be defined by

$$\text{prox}_{\gamma h}^{\epsilon}(y) := \left\{ x : h(x) + \frac{1}{2\gamma} \|x - y\|^2 \leq \epsilon + \min_{u \in \mathbb{R}^N} \left(h(u) + \frac{1}{2\gamma} \|u - y\|^2 \right) \right\}.$$

The corresponding basic inexact proximal-gradient iteration is

$$x_{k+1} \in \text{prox}_{\gamma_k h}^{\epsilon_k}(x_k - \gamma_k \nabla g(x_k)).$$

The influence of the error sequence on convergence was studied for convex composite problems [26, 28]. Gu et al. subsequently considered inexact proximal-gradient methods for non-convex composite optimization [13]. Their results require suitable step-size conditions along with an error sequence satisfying the stated decay or summability assumptions.

Gu et al. also combined inexact proximal evaluations with the monitored accelerated construction [13]. Their AIPG method uses

$$y_k = x_k + \frac{t_{k-1}}{t_k}(z_k - x_k) + \frac{t_{k-1} - 1}{t_k}(x_k - x_{k-1}),$$

followed by the two inexact candidates

$$z_{k+1} \in \text{prox}_{\gamma h}^{\epsilon_k}(y_k - \gamma \nabla g(y_k)), \quad (1.3)$$

and

$$v_{k+1} \in \text{prox}_{\gamma h}^{\epsilon_k}(x_k - \gamma \nabla g(x_k)), \quad 0 < \gamma < \frac{1}{L}. \quad (1.4)$$

The accepted iterate is

$$x_{k+1} = \begin{cases} z_{k+1}, & F(z_{k+1}) \leq F(v_{k+1}), \\ v_{k+1}, & \text{otherwise,} \end{cases} \quad (1.5)$$

and the momentum parameter is updated by

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}.$$

This construction extends the two-candidate monitor to inexact proximal evaluations; however, its basic form still solves two proximal subproblems per iteration. Gu et al. therefore also proposed a nonmonotone variant in which the monitor is computed only when the accelerated candidate fails a sufficient-decrease test [13]. A representative test has the form

$$F(z_{k+1}) \leq F(x_k) - \frac{\delta}{2} \|z_{k+1} - y_k\|^2. \quad (1.6)$$

If this condition holds, z_{k+1} is accepted without computing v_{k+1} . Otherwise, the monitor is evaluated, and the two candidates are compared. This conditional construction reduces the average number of proximal evaluations when the extrapolated point is frequently accepted by the method.

Yao et al. developed a different strategy in niAPG by testing the extrapolated point before the proximal calculation [29]. Their method forms

$$y_k = x_k + \frac{k-1}{k+2}(x_k - x_{k-1}),$$

and defines the nonmonotone reference value

$$\Delta_k = \max_{i=\max\{1, k-q\}, \dots, k} F(x_i).$$

The point used for the proximal-gradient step is selected by

$$w_k = \begin{cases} y_k, & F(y_k) \leq \Delta_k, \\ x_k, & \text{otherwise,} \end{cases} \quad (1.7)$$

after which only one inexact proximal step is computed as follows:

$$x_{k+1} \in \text{prox}_{\eta h}^{\epsilon_k}(w_k - \eta \nabla g(w_k)), \quad 0 < \eta < \frac{1}{L}. \quad (1.8)$$

The methods above control extrapolation at different stages. Li-Lin and AIPG compare an accelerated proximal candidate with a fallback point after the proximal evaluations, whereas niAPG selects the proximal center before computing a single inexact proximal step [14, 13, 29]. The monitored construction provides an explicit fallback but may require two proximal computations. In contrast, niAPG reduces this cost but relies on a prescribed step size and a non-monotone acceptance rule. Variable-step methods provide another alternative by adapting the step size to local information [18].

A related line of research concerns the selection of the step size. Liu and Wang proposed the non-monotone variable-step method newAPG_{vs}, in which an inertial candidate is accepted only after the preceding step has supplied sufficient decrease and

$$F(\hat{x}_{k+1}) \leq F(x_k) + \min \{Q_k, \delta(F(x_{k-1}) - F(x_k))\}, \quad (1.9)$$

where $\delta \in (0, 1)$ and $\{Q_k\}$ is a prescribed summable positive sequence [18]. Separately, its variable-step rule uses a local smoothness test and a summable control sequence for step-size growth. Thus, newAPG_{vs} permits limited increases in the objective while retaining an overall decreasing mechanism.

Another approach is the average-curvature accelerated composite gradient method (AC-FISTA) [16]. It estimates local upper curvature through quantities of the form

$$C_k = \frac{2[g(y_{k+1}) - g(\tilde{x}_k) - \langle \nabla g(\tilde{x}_k), y_{k+1} - \tilde{x}_k \rangle]}{\|y_{k+1} - \tilde{x}_k\|^2}, \quad (1.10)$$

and updates its curvature estimate using information accumulated from previous iterations. Its good-bad iteration mechanism differs from an explicit comparison between an adaptive trial point and a proximal point computed with a fixed reference step.

More recent adaptive proximal-gradient methods estimate the step size directly from the local first-order information. AdaPGNC uses local gradient differences to construct a line-search-free step-size rule for non-convex composite problems [30]. In contrast, AdProxBB combines proximal updates with Barzilai–Borwein spectral step sizes and a non-monotone line search [31, 22]. The two standard spectral estimates are

$$\lambda_k^{\text{BB1}} = \frac{\langle s_k, s_k \rangle}{\langle s_k, r_k \rangle}, \quad \lambda_k^{\text{BB2}} = \frac{\langle s_k, r_k \rangle}{\langle r_k, r_k \rangle}, \quad (1.11)$$

where

$$s_k = x_k - x_{k-1}, \quad r_k = \nabla g(x_k) - \nabla g(x_{k-1}). \quad (1.12)$$

These spectral quantities approximate the local curvature without requiring prior knowledge of the global Lipschitz constant.

These developments treat acceleration, inexactness, and local step-size information differently. The construction below considers whether a monitored spectral trial can be combined with a proximal point computed using a fixed step below $1/L$. This requires an additional proximal evaluation but allows the two points to play distinct roles in the analysis.

Taken together, the preceding methods control non-convex acceleration through candidate monitoring, non-monotone acceptance, or local curvature information [14, 13, 29, 18, 16, 30, 22]. The proposed method retains the monitored inexact setting but assigns different roles to its two candidates. A bounded spectral step generates a trial point, while a fixed step below $1/L$ generates the point used in

the descent analysis. A relaxation of the two points is accepted only when it satisfies both objective tests; otherwise, the fixed-step point is selected and the momentum is restarted.

The contribution does not assert convergence of the spectral trial sequence by itself. Instead, it studies a conditional mechanism that allows a step larger than $1/L$ only within a monitored trial. The analysis establishes descent and stationary accumulation points under summable proximal-value errors. Full-sequence convergence is obtained separately under an additional relative-residual condition and the Kurdyka–Łojasiewicz property.

2. PRELIMINARIES AND RELATED WORK

2.1. Notation and preliminaries. Throughout the paper, $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ denote the Euclidean inner product and norm, respectively. For a proper function h , its effective domain is

$$\text{dom } h := \{x \in \mathbb{R}^N : h(x) < +\infty\}.$$

If h is convex, its subdifferential at $x \in \text{dom } h$ is

$$\partial h(x) := \{\xi \in \mathbb{R}^N : h(u) \geq h(x) + \langle \xi, u - x \rangle \text{ for all } u \in \mathbb{R}^N\}.$$

We consider the composite problem introduced in (1.1), where the following basic conditions are used throughout the analysis.

Assumption 2.1. The function $g : \mathbb{R}^N \rightarrow \mathbb{R}$ is continuously differentiable, possibly non-convex, and its gradient is L -Lipschitz continuous:

$$\|\nabla g(x) - \nabla g(y)\| \leq L\|x - y\|, \quad x, y \in \mathbb{R}^N.$$

The function $h : \mathbb{R}^N \rightarrow (-\infty, +\infty]$ is proper, lower semicontinuous, and convex. Moreover,

$$F_{\inf} := \inf_{x \in \mathbb{R}^N} F(x) > -\infty.$$

For $\gamma > 0$, the proximal mapping of h is defined by

$$\text{prox}_{\gamma h}(y) := \arg \min_{u \in \mathbb{R}^N} \left\{ h(u) + \frac{1}{2\gamma} \|u - y\|^2 \right\}. \quad (2.1)$$

Under Assumption 2.1, this mapping is single-valued [4, 23].

The proximal subproblem may be solved inexactly. We use the proximal-value error model employed in inexact forward–backward and proximal-gradient methods [26, 28, 13].

Definition 2.2. Let $\gamma > 0$, $\epsilon \geq 0$, and $y \in \mathbb{R}^N$. A point u belongs to $\text{prox}_{\gamma h}^\epsilon(y)$ if

$$h(u) + \frac{1}{2\gamma} \|u - y\|^2 \leq \epsilon + \min_{w \in \mathbb{R}^N} \left\{ h(w) + \frac{1}{2\gamma} \|w - y\|^2 \right\}.$$

For $\epsilon = 0$, Definition 2.2 reduces to the exact proximal mapping in (2.1).

The Lipschitz continuity of ∇g gives the standard descent inequality [4].

Lemma 2.3. Under Assumption 2.1, for all $x, u \in \mathbb{R}^N$,

$$g(u) \leq g(x) + \langle \nabla g(x), u - x \rangle + \frac{L}{2} \|u - x\|^2.$$

A point $x^* \in \text{dom } h$ is called stationary, or critical, for F if

$$0 \in \nabla g(x^*) + \partial h(x^*).$$

We denote the set of such points by

$$\text{crit } F := \{x \in \text{dom } h : 0 \in \nabla g(x) + \partial h(x)\}.$$

For any $\gamma > 0$, the exact proximal fixed-point condition is equivalent to stationarity:

$$x^* = \text{prox}_{\gamma h}(x^* - \gamma \nabla g(x^*)) \iff x^* \in \text{crit } F.$$

This characterization is standard in proximal-gradient analysis [23, 1].

2.2. Acceleration and adaptive restart. Nesterov’s accelerated gradient method uses extrapolation to obtain an $O(1/k^2)$ objective-value rate for smooth convex problems [19]. However, in practice, the accumulated momentum can produce oscillatory behavior. O’Donoghue and Candès proposed adaptive restart criteria that reset the momentum when such behavior is detected [21]. For a smooth objective f , two commonly used tests trigger a reset when

$$f(x_k) > f(x_{k-1}) \quad \text{or} \quad \langle \nabla f(y_{k-1}), x_k - x_{k-1} \rangle > 0.$$

These tests improve practical performance without requiring prior knowledge of local curvature. Related inertial forward–backward methods for nonconvex problems include [6, 20], while the dynamical interpretation of acceleration is supported by continuous-time analyses such as [27].

Restart has subsequently been analyzed for smooth nonconvex accelerated-gradient methods [15]. These results demonstrate that restart can play a theoretical role in nonconvex optimization under smoothness assumptions different from the nonsmooth composite setting considered here. More generally, restart controls the momentum used in subsequent extrapolations but does not independently guarantee descent or convergence in an inexact proximal method. Such guarantees must follow accepted step conditions and control proximal errors.

2.3. Monitored inexact proximal-gradient methods. For structured regularizers, the proximal subproblem may require an iterative inner procedure. Inexact proximal-gradient methods replace the exact minimizer by a point satisfying a prescribed accuracy, such as the value-gap condition in Definition 2.2. Convergence under decreasing computational errors was studied for convex proximal-gradient and forward–backward methods in [26, 28]. More recent developments include relative-error linesearch and adaptive inexact accelerated schemes [5, 17, 12].

For non-convex composite problems, Li and Lin use the monitored selection in (1.2) [14]. Gu et al. extend it to the inexact candidates (1.3) and (1.4), with selection (1.5); nmAIPG computes the monitor only when (1.6) fails [13]. By contrast, niAPG selects the proximal center by (1.7) before the single update (1.8) [29]. Thus, AIPG uses a post-proximal comparison, nmAIPG makes its second proximal computation conditional, and niAPG uses pre-proximal selection. Their analyses require method-specific step-size, inexactness, and regularity conditions [13, 29].

2.4. Spectral and adaptive step sizes. Standard proximal-gradient analyses generally use a global step size bounded by $1/L$ [4, 23]. Spectral and adaptive methods instead use local first-order information to construct a trial step. The BB estimates in (1.11), formed from the differences in (1.12), approximate inverse curvature without requiring a Hessian [3]. They may exceed $1/L$, but they are not guaranteed reciprocal local Lipschitz bounds. In a non-convex problem, the curvature denominator may be non-positive or close to zero, so a raw BB value generally requires checking or safeguarding [24, 25].

Recent composite methods use local information in different ways. The newAPG_{vs} acceptance rule (1.9) is combined with a variable-step mechanism [18]; AC-FISTA uses averaged curvature information based on quantities such as (1.10) [16]; AdaPGNC employs local upper- and lower-curvature rules without line searches [30]; and AdProxBB combines proximal BB steps with a non-monotone line search [22]. These methods motivate the use of local information but have update rules and assumptions different from the present construction.

HA-AIPG uses BB information only to generate the trial point; its precise spectral choices and bounded trial rule are stated in (3.1)–(3.5). The descent analysis instead uses the point computed with $\lambda < 1/L$.

3. MAIN RESULTS: THE HA-AIPG FRAMEWORK

The Hybrid Adaptive Accelerated Inexact Proximal Gradient (HA-AIPG) method computes an adaptive spectral trial z_{k+1} and a fixed-step reference point v_{k+1} . A relaxation between these points is accepted only when it satisfies two objective tests. If no positive relaxation is accepted, the reference point is selected and the momentum is restarted [13, 21].

3.1. Adaptive spectral step-size selection. For $k \geq 1$, define

$$s_{k-1} = y_k - y_{k-1}, \quad r_{k-1} = \nabla g(y_k) - \nabla g(y_{k-1}).$$

The two classical Barzilai–Borwein estimates are

$$\gamma_{z,1}^* = \frac{\langle s_{k-1}, s_{k-1} \rangle}{\langle s_{k-1}, r_{k-1} \rangle}, \quad (3.1)$$

and

$$\gamma_{z,2}^* = \frac{\langle s_{k-1}, r_{k-1} \rangle}{\langle r_{k-1}, r_{k-1} \rangle}. \quad (3.2)$$

When $\langle s_{k-1}, r_{k-1} \rangle > 0$, both estimates are positive and

$$0 < \gamma_{z,2}^* \leq \gamma_{z,1}^*$$

by the Cauchy–Schwarz inequality [3, 24].

We also consider the reciprocal-secant estimate

$$L_k = \frac{\|\nabla g(y_k) - \nabla g(y_{k-1})\|}{\|y_k - y_{k-1}\|}, \quad \gamma_z^* = \frac{1}{L_k}. \quad (3.3)$$

Here, L_k is a secant curvature estimate and is not assumed to be a local Lipschitz constant. If $\langle s_{k-1}, r_{k-1} \rangle > 0$ and the quantities are nonzero and finite, then

$$\frac{1}{L_k} = \sqrt{\gamma_{z,1}^* \gamma_{z,2}^*}, \quad (3.4)$$

and hence $\gamma_{z,2}^* \leq 1/L_k \leq \gamma_{z,1}^*$.

The BB1 and BB2 estimates are called admissible when their curvature denominator is positive and the resulting value is positive and finite. The reciprocal-secant estimate is admissible when $\|s_{k-1}\|$ and $\|r_{k-1}\|$ are positive and finite. For the selected spectral rule, the trial step is defined by

$$\gamma_z = \begin{cases} \lambda, & k = 0 \text{ or the selected estimate is not admissible,} \\ \min\{\gamma_{\max}, \max\{\gamma_{\min}, \gamma_z^*\}\}, & \text{otherwise.} \end{cases} \quad (3.5)$$

The interval $0 < \gamma_{\min} \leq \gamma_{\max} < \infty$ excludes degenerate spectral values. It is not required that $\gamma_{\max} \leq 1/L$; a step larger than $1/L$ is used only to construct a trial point, whose acceptance is determined by the tests below.

3.2. **The HA-AIPG algorithm.** Let

$$a = \frac{1}{2\lambda} - \frac{L}{2} > 0, \quad 0 < c < a,$$

and, for $\alpha \in [0, 1]$, define

$$w_{k+1}(\alpha) = v_{k+1} + \alpha(z_{k+1} - v_{k+1}).$$

A positive relaxation is accepted only if

$$F(w_{k+1}(\alpha)) \leq F(x_k) - c\|w_{k+1}(\alpha) - x_k\|^2 \quad (3.6)$$

and

$$F(w_{k+1}(\alpha)) \leq F(v_{k+1}). \quad (3.7)$$

The first condition requires sufficient decrease from x_k , while the second compares the relaxed point with the fixed-step reference point.

Algorithm 1: Hybrid Adaptive Accelerated Inexact Proximal Gradient (HA-AIPG)

Input: Iteration limit $m \in \mathbb{N}$; initial point $x_0 \in \mathbb{R}^N$; reference step $0 < \lambda < 1/L$; descent modulus $0 < c < \frac{1}{2\lambda} - \frac{L}{2}$; spectral bounds $0 < \gamma_{\min} \leq \gamma_{\max}$; reduction factor $\rho \in (0, 1)$; minimum positive relaxation $\alpha_{\min} \in (0, 1]$; inexactness sequence $\{\epsilon_k\}$.

Result: Final iterate x_m .

```

1 Initialize  $x_{-1} = x_0, t_0 = 1, y_{-1} = x_0$ , and  $\nabla g(y_{-1}) = \nabla g(x_0)$ ;
2 for  $k = 0, 1, \dots, m-1$  do
3    $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}, \quad \beta_k = \frac{t_k-1}{t_{k+1}};$ 
4    $y_k = x_k + \beta_k(x_k - x_{k-1});$ 
5   if  $k = 0$  then
6      $\gamma_z = \lambda;$ 
7   else
8     Compute  $\gamma_z^*$  from (3.1), (3.2), or (3.3);
9     Set  $\gamma_z$  according to (3.5);
10  end
11  Compute  $z_{k+1} \in \text{prox}_{\gamma_z h}^{\epsilon_k}(y_k - \gamma_z \nabla g(y_k));$ 
12  Compute  $v_{k+1} \in \text{prox}_{\lambda h}^{\epsilon_k}(x_k - \lambda \nabla g(x_k));$ 
13  Set  $\alpha = 1;$ 
14  while  $\alpha \geq \alpha_{\min}$  do
15     $w_{k+1} = v_{k+1} + \alpha(z_{k+1} - v_{k+1});$ 
16    if  $w_{k+1}$  satisfies (3.6) and (3.7) then
17       $\alpha_k = \alpha, \quad x_{k+1} = w_{k+1};$ 
18      break;
19    else
20       $\alpha \leftarrow \rho\alpha;$ 
21    end
22  end
23  if  $\alpha < \alpha_{\min}$  then
24     $\alpha_k = 0, \quad x_{k+1} = v_{k+1}, \quad t_{k+1} = 1;$ 
25  end
26  Store  $y_k$  and  $\nabla g(y_k)$  for the next spectral update;
27 end
```

The finite value m specifies the computational budget. The convergence analysis considers the sequence obtained by allowing m to be arbitrary. Reducing α reuses z_{k+1} and v_{k+1} , and therefore requires only a new convex combination and an objective evaluation. Nevertheless, the method computes two proximal points at each outer iteration.

3.3. Selection and momentum restart. When a positive relaxation is accepted, the selected iterate satisfies (3.6) and (3.7) by construction. If every tested positive relaxation fails at least one condition, Algorithm 1 sets

$$\alpha_k = 0, \quad x_{k+1} = v_{k+1}, \quad t_{k+1} = 1.$$

Consequently,

$$\beta_{k+1} = \frac{t_{k+1} - 1}{t_{k+2}} = 0, \quad y_{k+1} = x_{k+1}.$$

Reducing α does not itself restart the momentum. The restart occurs only when no positive relaxation is accepted, so the rejected inertial direction is not carried into the next iteration [21, 15]. The restart is an algorithmic control; the convergence analysis uses the acceptance conditions together with the inexact descent estimate for v_{k+1} .

4. THEORETICAL ANALYSIS

This section establishes convergence properties of HA-AIPG. The analysis separates the two step-size regimes: the point v_{k+1} is computed with $\lambda < 1/L$, whereas the step used to construct z_{k+1} is only required to remain in the prescribed spectral interval. The proof uses the trial point only through the two acceptance conditions. This distinction is useful because it avoids assigning a descent property to a spectral step that may exceed $1/L$.

4.1. Algorithmic assumptions. Throughout this section, Assumption 2.1 and the inexact proximal mapping in Definition 2.2 are used. The additional conditions required by HA-AIPG are collected below.

Assumption 4.1. The following conditions hold.

- (1) The proximal value errors satisfy

$$\epsilon_k \geq 0, \quad \sum_{k=0}^{\infty} \epsilon_k < \infty. \quad (4.1)$$

- (2) The fixed reference step and initial point satisfy

$$0 < \lambda < \frac{1}{L}, \quad x_0 \in \text{dom } h. \quad (4.2)$$

- (3) The spectral trial steps generated by Algorithm 1 satisfy

$$0 < \gamma_{\min} \leq \gamma_z \leq \gamma_{\max} < \infty.$$

The summability condition implies $\epsilon_k \rightarrow 0$. The error of the reference point v_{k+1} enters the descent estimate directly, whereas the trial point z_{k+1} is used only through the two acceptance conditions. A stronger relative-residual condition is introduced later only for the KL analysis.

4.2. Descent and selection estimates. We first establish the inexact forward-backward descent inequality used by the acceptance rule. Related estimates appear in [4, 26, 13].

Lemma 4.2. Suppose that Assumption 4.1 holds and

$$v_{k+1} \in \text{prox}_{\lambda h}^{\epsilon_k}(x_k - \lambda \nabla g(x_k)).$$

Then

$$F(v_{k+1}) \leq F(x_k) - a \|v_{k+1} - x_k\|^2 + \epsilon_k, \quad a = \frac{1 - \lambda L}{2\lambda} > 0. \quad (4.3)$$

Proof. Set

$$d_k = v_{k+1} - x_k.$$

Using x_k as the comparison point in Definition 2.2 gives

$$\begin{aligned} h(v_{k+1}) + \frac{1}{2\lambda} \|v_{k+1} - (x_k - \lambda \nabla g(x_k))\|^2 &\leq h(x_k) + \frac{1}{2\lambda} \|x_k - (x_k - \lambda \nabla g(x_k))\|^2 + \epsilon_k \\ &= h(x_k) + \frac{\lambda}{2} \|\nabla g(x_k)\|^2 + \epsilon_k. \end{aligned} \quad (4.4)$$

Since

$$v_{k+1} - (x_k - \lambda \nabla g(x_k)) = d_k + \lambda \nabla g(x_k),$$

we have

$$\|d_k + \lambda \nabla g(x_k)\|^2 = \|d_k\|^2 + 2\lambda \langle \nabla g(x_k), d_k \rangle + \lambda^2 \|\nabla g(x_k)\|^2.$$

Substituting this identity into (4.4) and cancelling the common gradient term yields

$$h(v_{k+1}) \leq h(x_k) - \langle \nabla g(x_k), d_k \rangle - \frac{1}{2\lambda} \|d_k\|^2 + \epsilon_k. \quad (4.5)$$

By Lemma 2.3,

$$g(v_{k+1}) \leq g(x_k) + \langle \nabla g(x_k), d_k \rangle + \frac{L}{2} \|d_k\|^2. \quad (4.6)$$

Adding (4.5) and (4.6) gives

$$\begin{aligned} F(v_{k+1}) &\leq F(x_k) - \left(\frac{1}{2\lambda} - \frac{L}{2} \right) \|d_k\|^2 + \epsilon_k \\ &= F(x_k) - \frac{1 - \lambda L}{2\lambda} \|v_{k+1} - x_k\|^2 + \epsilon_k. \end{aligned}$$

Since $0 < \lambda < 1/L$, the coefficient $a = (1 - \lambda L)/(2\lambda)$ is positive. This proves (4.3). $\square$

The next lemma transfers the fixed-step estimate to the selected iterate. No bound on $\|z_{k+1} - v_{k+1}\|$ is required.

Lemma 4.3. *Let x_{k+1} be generated by Algorithm 1. Under Assumption 4.1,*

$$F(x_{k+1}) \leq F(v_{k+1}), \quad (4.7)$$

and

$$F(x_{k+1}) \leq F(x_k) - a \|v_{k+1} - x_k\|^2 + \epsilon_k. \quad (4.8)$$

Moreover, let

$$\bar{c} = \min\{a, c\} > 0.$$

Then

$$F(x_{k+1}) \leq F(x_k) - \bar{c} \|x_{k+1} - x_k\|^2 + \epsilon_k. \quad (4.9)$$

Proof. Suppose first that a positive relaxation is accepted. By the two acceptance conditions,

$$\begin{aligned} F(x_{k+1}) &\leq F(v_{k+1}), \\ F(x_{k+1}) &\leq F(x_k) - c \|x_{k+1} - x_k\|^2. \end{aligned}$$

Combining the first inequality with (4.3) gives

$$\begin{aligned} F(x_{k+1}) &\leq F(v_{k+1}) \\ &\leq F(x_k) - a \|v_{k+1} - x_k\|^2 + \epsilon_k, \end{aligned}$$

which proves (4.8) in the accepted case. Since $\bar{c} \leq c$ and $\epsilon_k \geq 0$,

$$\begin{aligned} F(x_{k+1}) &\leq F(x_k) - c \|x_{k+1} - x_k\|^2 \\ &\leq F(x_k) - \bar{c} \|x_{k+1} - x_k\|^2 + \epsilon_k. \end{aligned}$$

Thus, (4.9) also holds.

Suppose next that no positive relaxation is accepted. Then

$$\alpha_k = 0, \quad x_{k+1} = v_{k+1}.$$

Consequently,

$$F(x_{k+1}) = F(v_{k+1}),$$

so (4.7) holds with equality. Lemma 4.2 gives

$$\begin{aligned} F(x_{k+1}) &= F(v_{k+1}) \\ &\leq F(x_k) - a\|v_{k+1} - x_k\|^2 + \epsilon_k \\ &= F(x_k) - a\|x_{k+1} - x_k\|^2 + \epsilon_k \\ &\leq F(x_k) - \bar{c}\|x_{k+1} - x_k\|^2 + \epsilon_k, \end{aligned}$$

because $\bar{c} \leq a$. Hence (4.8) and (4.9) hold in the fallback case as well. The two cases prove the result. $\square$

Remark 4.4. Lemmas 4.2 and 4.3 impose the restriction $0 < \lambda < 1/L$ only on the fixed reference step. They do not require $\gamma_z \leq 1/L$. Therefore, if a trial with $\gamma_z > 1/L$ passes both acceptance conditions, the selected iterate still satisfies (4.8) and (4.9).

This is a conditional statement. It does not assert that trial steps larger than $1/L$ occur frequently, that they are always accepted, or that the sequence $\{z_k\}$ converges independently. Other adaptive methods, including AdaPGNC, control their step sizes through different curvature conditions [30]. Their guarantees are not consequences of this remark. HA-AIPG is also not L -free, because its fixed reference step must satisfy (4.2).

4.3. Asymptotic regularity and stationarity. The next argument follows the standard summable-error analysis used for inexact proximal-gradient methods [26, 13]. The selection estimates in Lemma 4.3 allow the same argument to be applied to the HA-AIPG iterates.

Lemma 4.5. *Suppose that Assumption 4.1 holds. Then*

$$\sum_{k=0}^{\infty} \|v_{k+1} - x_k\|^2 < \infty, \quad \sum_{k=0}^{\infty} \|x_{k+1} - x_k\|^2 < \infty. \quad (4.10)$$

Consequently,

$$\begin{aligned} \|v_{k+1} - x_k\| &\rightarrow 0, \\ \|x_{k+1} - x_k\| &\rightarrow 0, \\ \|x_{k+1} - v_{k+1}\| &\rightarrow 0. \end{aligned} \quad (4.11)$$

Moreover, there exists $F_* \in \mathbb{R}$ such that

$$F(x_k) \rightarrow F_*.$$

Proof. Rearranging (4.8) gives

$$a\|v_{k+1} - x_k\|^2 \leq F(x_k) - F(x_{k+1}) + \epsilon_k.$$

For any integer $m \geq 0$, summation from $k = 0$ to m yields

$$\begin{aligned} a \sum_{k=0}^m \|v_{k+1} - x_k\|^2 &\leq \sum_{k=0}^m (F(x_k) - F(x_{k+1})) + \sum_{k=0}^m \epsilon_k \\ &= F(x_0) - F(x_{m+1}) + \sum_{k=0}^m \epsilon_k \\ &\leq F(x_0) - F_{\inf} + \sum_{k=0}^{\infty} \epsilon_k < \infty, \end{aligned}$$

where Assumption 2.1 and (4.1) were used in the last line. Letting $m \rightarrow \infty$ proves

$$\sum_{k=0}^{\infty} \|v_{k+1} - x_k\|^2 < \infty.$$

Similarly, (4.9) gives

$$\bar{c}\|x_{k+1} - x_k\|^2 \leq F(x_k) - F(x_{k+1}) + \epsilon_k.$$

Therefore,

$$\begin{aligned} \bar{c} \sum_{k=0}^m \|x_{k+1} - x_k\|^2 &\leq F(x_0) - F(x_{m+1}) + \sum_{k=0}^m \epsilon_k \\ &\leq F(x_0) - F_{\inf} + \sum_{k=0}^{\infty} \epsilon_k. \end{aligned}$$

Letting $m \rightarrow \infty$ proves the second summability statement in (4.10).

Since both series have nonnegative terms,

$$\|v_{k+1} - x_k\| \rightarrow 0, \quad \|x_{k+1} - x_k\| \rightarrow 0.$$

The triangle inequality gives

$$\|x_{k+1} - v_{k+1}\| \leq \|x_{k+1} - x_k\| + \|v_{k+1} - x_k\| \rightarrow 0.$$

This proves (4.11).

It remains to prove convergence of the objective values. Let

$$E_k = \sum_{j=k}^{\infty} \epsilon_j.$$

Then $E_k \rightarrow 0$ and

$$E_k - E_{k+1} = \epsilon_k.$$

Using (4.9), we obtain

$$\begin{aligned} F(x_{k+1}) + E_{k+1} &\leq F(x_k) - \bar{c}\|x_{k+1} - x_k\|^2 + \epsilon_k + E_{k+1} \\ &= F(x_k) + E_k - \bar{c}\|x_{k+1} - x_k\|^2 \\ &\leq F(x_k) + E_k. \end{aligned}$$

Thus, $\{F(x_k) + E_k\}$ is decreasing. It is bounded below by $F_{\inf}$, so it converges to a finite value. Since $E_k \rightarrow 0$, the sequence $\{F(x_k)\}$ converges to the same value, denoted by F_* . $\square$

Theorem 4.6. *Suppose that Assumption 4.1 holds. Then every accumulation point x^* of $\{x_k\}$ is stationary for F :*

$$0 \in \nabla g(x^*) + \partial h(x^*). \quad (4.12)$$

Equivalently, $x^* \in \text{crit } F$.

Proof. Let $\{x_{k_j}\}$ be a convergent subsequence such that

$$x_{k_j} \rightarrow x^*.$$

By Lemma 4.5,

$$\|v_{k_j+1} - x_{k_j}\| \rightarrow 0.$$

Consequently,

$$v_{k_j+1} \rightarrow x^*.$$

Moreover, the nonnegativity and summability of $\{\epsilon_k\}$ imply

$$\epsilon_{k_j} \rightarrow 0.$$

Let $u \in \text{dom } h$ be arbitrary. From Definition 2.2,

$$\begin{aligned} & h(v_{k_j+1}) + \frac{1}{2\lambda} \|v_{k_j+1} - (x_{k_j} - \lambda \nabla g(x_{k_j}))\|^2 \\ & \leq h(u) + \frac{1}{2\lambda} \|u - (x_{k_j} - \lambda \nabla g(x_{k_j}))\|^2 + \epsilon_{k_j}. \end{aligned} \quad (4.13)$$

Since $x_{k_j} \rightarrow x^*$ and ∇g is continuous,

$$x_{k_j} - \lambda \nabla g(x_{k_j}) \rightarrow x^* - \lambda \nabla g(x^*).$$

Taking the lower limit on the left-hand side of (4.13), using $v_{k_j+1} \rightarrow x^*$, the lower semicontinuity of h , and $\epsilon_{k_j} \rightarrow 0$, gives

$$h(x^*) + \frac{1}{2\lambda} \|x^* - (x^* - \lambda \nabla g(x^*))\|^2 \leq h(u) + \frac{1}{2\lambda} \|u - (x^* - \lambda \nabla g(x^*))\|^2. \quad (4.14)$$

The left-hand side is finite because the right-hand side is finite for $u \in \text{dom } h$. Hence $x^* \in \text{dom } h$.

Since (4.14) holds for every $u \in \text{dom } h$, x^* minimizes the proximal subproblem:

$$x^* = \text{prox}_{\lambda h}(x^* - \lambda \nabla g(x^*)). \quad (4.15)$$

The optimality condition for (4.15) gives

$$\begin{aligned} 0 & \in \partial h(x^*) + \frac{1}{\lambda} [x^* - (x^* - \lambda \nabla g(x^*))] \\ & = \partial h(x^*) + \nabla g(x^*). \end{aligned}$$

Therefore,

$$0 \in \nabla g(x^*) + \partial h(x^*),$$

which proves (4.12). The fixed-point characterization is standard; see [23, 13]. $\square$

The limiting-proximal argument follows the usual inexact proximal-gradient pattern [26, 13]. The HA-AIPG-specific step is (4.8), which transfers the descent estimate for v_{k+1} to the selected iterate x_{k+1} . Therefore, the accepted and reference sequences have the same accumulation points, even though no descent or convergence property is assumed for the raw spectral sequence $\{z_k\}$.

4.4. Boundedness of the generated sequences.

Corollary 4.7 (Boundedness under a bounded-level-set condition). *Suppose that the level set*

$$\left\{ x \in \text{dom } h : F(x) \leq F(x_0) + \sum_{j=0}^{\infty} \epsilon_j \right\} \quad (4.16)$$

is bounded. Then both $\{x_k\}$ and $\{v_{k+1}\}$ are bounded. Consequently, $\{x_k\}$ has at least one accumulation point.

Proof. From (4.9) and $\epsilon_k \geq 0$,

$$\begin{aligned} F(x_{k+1}) & \leq F(x_k) - \bar{c} \|x_{k+1} - x_k\|^2 + \epsilon_k \\ & \leq F(x_k) + \epsilon_k. \end{aligned} \quad (4.17)$$

Applying (4.17) recursively gives

$$\begin{aligned} F(x_k) & \leq F(x_0) + \sum_{j=0}^{k-1} \epsilon_j \\ & \leq F(x_0) + \sum_{j=0}^{\infty} \epsilon_j. \end{aligned}$$

Hence every x_k belongs to the bounded level set in (4.16), and therefore $\{x_k\}$ is bounded.

Lemma 4.5 gives $\|v_{k+1} - x_k\| \rightarrow 0$. Thus,

$$\|v_{k+1}\| \leq \|x_k\| + \|v_{k+1} - x_k\|,$$

which shows that $\{v_{k+1}\}$ is also bounded. The existence of an accumulation point of $\{x_k\}$ then follows from finite dimensionality. $\square$

Under the bounded-level-set condition, the accumulation-point statement in Theorem 4.6 is non-vacuous. The same condition also provides the boundedness required in the subsequent KL analysis.

4.5. Full-sequence convergence under the KL property.

Definition 4.8. Let $\bar{x} \in \text{dom } \partial F$. The function F has the KL property at $\bar{x}$ if there exist $\delta > 0$, $\nu > 0$, and a continuous concave function $\varphi : [0, \nu) \rightarrow \mathbb{R}_+$ such that $\varphi(0) = 0$, φ is continuously differentiable on $(0, \nu)$, $\varphi'(s) > 0$, and

$$\varphi'(F(x) - F(\bar{x})) \text{dist}(0, \partial F(x)) \geq 1$$

whenever

$$\|x - \bar{x}\| < \delta, \quad F(\bar{x}) < F(x) < F(\bar{x}) + \nu.$$

A proper lower semicontinuous function satisfying this property at every point of $\text{dom } \partial F$ is called a KL function.

This is the standard nonsmooth KL property used in the convergence analysis of descent methods [1]. In particular, proper lower semicontinuous semi-algebraic functions are KL functions. The proof below uses the uniformized KL inequality on the compact set of accumulation points, following the arguments in [1, 13].

The value-gap condition in Definition 2.2 is sufficient for the preceding stationarity analysis. For the full-sequence result, we impose an additional residual condition on the fixed-step proximal point.

Assumption 4.9. There exist $\tau \geq 0$ and $\xi_{k+1} \in \partial h(v_{k+1})$ such that

$$\left\| \nabla g(x_k) + \frac{1}{\lambda}(v_{k+1} - x_k) + \xi_{k+1} \right\| \leq \tau \|v_{k+1} - x_k\|, \quad 0 \leq \tau < \frac{1}{\lambda} - \frac{L}{2}. \quad (4.18)$$

Assumption 4.9 controls the residual in the optimality condition of the fixed-step proximal subproblem. It does not follow directly from the value-gap condition and is required only for the KL analysis. The trial point z_{k+1} is not required to satisfy this condition. Related residual conditions are used in KL analyses of inexact descent and forward-backward methods [1, 28, 13].

Lemma 4.10. Suppose that Assumptions 4.1 and 4.9 hold. Then

$$F(v_{k+1}) \leq F(x_k) - \left(\frac{1}{\lambda} - \frac{L}{2} - \tau \right) \|v_{k+1} - x_k\|^2. \quad (4.19)$$

Moreover,

$$\text{dist}(0, \partial F(v_{k+1})) \leq \left(L + \frac{1}{\lambda} + \tau \right) \|v_{k+1} - x_k\|. \quad (4.20)$$

If $v_0 = x_0$, then

$$F(v_{k+1}) \leq F(v_k) - \left(\frac{1}{\lambda} - \frac{L}{2} - \tau \right) \|v_{k+1} - x_k\|^2. \quad (4.21)$$

Finally, for every $k \geq 0$,

$$\begin{aligned} F(v_{k+2}) &\leq F(v_k) - \min \left\{ c, \frac{1}{\lambda} - \frac{L}{2} - \tau \right\} \|x_{k+1} - x_k\|^2 \\ &\quad - \left(\frac{1}{\lambda} - \frac{L}{2} - \tau \right) \|v_{k+2} - x_{k+1}\|^2. \end{aligned} \quad (4.22)$$

Proof. Since h is convex and $\xi_{k+1} \in \partial h(v_{k+1})$, the subgradient inequality gives

$$h(v_{k+1}) \leq h(x_k) + \langle \xi_{k+1}, v_{k+1} - x_k \rangle. \quad (4.23)$$

The inner product in (4.23) can be written as

$$\begin{aligned} \langle \xi_{k+1}, v_{k+1} - x_k \rangle &= -\langle \nabla g(x_k), v_{k+1} - x_k \rangle - \frac{1}{\lambda} \|v_{k+1} - x_k\|^2 \\ &\quad + \left\langle \nabla g(x_k) + \frac{1}{\lambda}(v_{k+1} - x_k) + \xi_{k+1}, v_{k+1} - x_k \right\rangle. \end{aligned}$$

By Assumption 4.9 and the Cauchy–Schwarz inequality,

$$\left\langle \nabla g(x_k) + \frac{1}{\lambda}(v_{k+1} - x_k) + \xi_{k+1}, v_{k+1} - x_k \right\rangle \leq \tau \|v_{k+1} - x_k\|^2. \quad (4.24)$$

Combining (4.23)–(4.24) yields

$$\begin{aligned} h(v_{k+1}) &\leq h(x_k) - \langle \nabla g(x_k), v_{k+1} - x_k \rangle \\ &\quad - \left(\frac{1}{\lambda} - \tau \right) \|v_{k+1} - x_k\|^2. \end{aligned} \quad (4.25)$$

The Lipschitz continuity of ∇g gives

$$g(v_{k+1}) \leq g(x_k) + \langle \nabla g(x_k), v_{k+1} - x_k \rangle + \frac{L}{2} \|v_{k+1} - x_k\|^2. \quad (4.26)$$

Adding (4.25) and (4.26) proves (4.19).

The smooth sum rule gives

$$\nabla g(v_{k+1}) + \xi_{k+1} \in \partial F(v_{k+1}).$$

Furthermore,

$$\begin{aligned} \nabla g(v_{k+1}) + \xi_{k+1} &= \nabla g(v_{k+1}) - \nabla g(x_k) - \frac{1}{\lambda}(v_{k+1} - x_k) \\ &\quad + \nabla g(x_k) + \frac{1}{\lambda}(v_{k+1} - x_k) + \xi_{k+1}. \end{aligned} \quad (4.27)$$

Taking norms in (4.27), using the Lipschitz continuity of ∇g , and applying (4.18), we obtain

$$\begin{aligned} \text{dist}(0, \partial F(v_{k+1})) &\leq \|\nabla g(v_{k+1}) + \xi_{k+1}\| \\ &\leq L \|v_{k+1} - x_k\| + \frac{1}{\lambda} \|v_{k+1} - x_k\| + \tau \|v_{k+1} - x_k\| \\ &= \left(L + \frac{1}{\lambda} + \tau \right) \|v_{k+1} - x_k\|. \end{aligned}$$

This proves (4.20).

For $k = 0$, $v_0 = x_0$. For $k \geq 1$, (4.7) at iteration $k - 1$ gives

$$F(x_k) \leq F(v_k).$$

Combining this inequality with (4.19) proves (4.21).

It remains to establish (4.22). If a positive relaxation is accepted at iteration k , then

$$\begin{aligned}
 F(v_{k+2}) &\leq F(x_{k+1}) - \left(\frac{1}{\lambda} - \frac{L}{2} - \tau\right) \|v_{k+2} - x_{k+1}\|^2 \\
 &\leq F(x_k) - c\|x_{k+1} - x_k\|^2 \\
 &\quad - \left(\frac{1}{\lambda} - \frac{L}{2} - \tau\right) \|v_{k+2} - x_{k+1}\|^2 \\
 &\leq F(v_k) - c\|x_{k+1} - x_k\|^2 \\
 &\quad - \left(\frac{1}{\lambda} - \frac{L}{2} - \tau\right) \|v_{k+2} - x_{k+1}\|^2.
 \end{aligned} \tag{4.28}$$

If no positive relaxation is accepted, then $x_{k+1} = v_{k+1}$. Hence

$$\|x_{k+1} - x_k\| = \|v_{k+1} - x_k\|,$$

and two applications of (4.21) give

$$\begin{aligned}
 F(v_{k+2}) &\leq F(v_k) - \left(\frac{1}{\lambda} - \frac{L}{2} - \tau\right) \|x_{k+1} - x_k\|^2 \\
 &\quad - \left(\frac{1}{\lambda} - \frac{L}{2} - \tau\right) \|v_{k+2} - x_{k+1}\|^2.
 \end{aligned} \tag{4.29}$$

Taking the smaller of the two possible coefficients of $\|x_{k+1} - x_k\|^2$ in (4.28) and (4.29) proves (4.22). $\square$

Lemma 4.10 provides the sufficient-decrease and relative-error estimates required by the KL argument [1]. The estimates are imposed on the fixed-step points v_{k+1} , while (4.22) transfers the result to the selected HA-AIPG iterates.

Theorem 4.11. *Suppose that Assumptions 4.1 and 4.9 hold. Assume further that*

- (1) F is a KL function;
- (2) the sequence $\{v_k\}$ is bounded; and
- (3) for every convergent subsequence $v_{k_j} \rightarrow \bar{x}$, one has $F(v_{k_j}) \rightarrow F(\bar{x})$.

Then

$$\sum_{k=0}^{\infty} \|x_{k+1} - x_k\| < \infty.$$

Consequently, $\{x_k\}$ converges to a stationary point x^* of F , and $v_k \rightarrow x^*$.

Proof. Lemma 4.5 gives $F(x_k) \rightarrow F_*$. For $k \geq 1$, (4.7) and (4.19) give

$$F(x_k) \leq F(v_k) \leq F(x_{k-1}).$$

Hence, by the squeeze theorem, $F(v_k) \rightarrow F_*$. Moreover, (4.21) shows that $\{F(v_k)\}$ is decreasing. Summing (4.21) gives

$$\sum_{k=0}^{\infty} \|v_{k+1} - x_k\|^2 < \infty, \quad \|v_{k+1} - x_k\| \rightarrow 0. \tag{4.30}$$

Let Ω be the set of accumulation points of $\{v_k\}$. Boundedness makes Ω nonempty and compact. If $v_{k_j} \rightarrow \bar{x}$, then (4.30) gives $x_{k_j-1} \rightarrow \bar{x}$. Theorem 4.6 and the assumed convergence of the function values therefore give

$$\Omega \subset \text{crit } F, \quad F(\bar{x}) = F_* \quad \text{for every } \bar{x} \in \Omega.$$

If $F(v_k) = F_*$ at some sufficiently large index, then (4.21) and (4.22) force all subsequent steps to vanish. Otherwise, the uniformized KL property on Ω provides an index K and a common desingularizing function φ such that

$$\varphi'(F(v_k) - F_*) \operatorname{dist}(0, \partial F(v_k)) \geq 1, \quad k \geq K. \quad (4.31)$$

By concavity of φ , together with (4.21), (4.20), and (4.31),

$$\begin{aligned} & \varphi(F(v_k) - F_*) - \varphi(F(v_{k+1}) - F_*) \\ & \geq \frac{\left(\frac{1}{\lambda} - \frac{L}{2} - \tau\right) \|v_{k+1} - x_k\|^2}{\left(L + \frac{1}{\lambda} + \tau\right) \|v_k - x_{k-1}\|}. \end{aligned} \quad (4.32)$$

If $\|v_k - x_{k-1}\| = 0$, then (4.20) gives $0 \in \partial F(v_k)$, and the stationary case has already been reached. Otherwise, applying $2\sqrt{ab} \leq a + b$ to (4.32) gives

$$\begin{aligned} 2\|v_{k+1} - x_k\| & \leq \|v_k - x_{k-1}\| \\ & + \frac{L + \frac{1}{\lambda} + \tau}{\frac{1}{\lambda} - \frac{L}{2} - \tau} \left[\varphi(F(v_k) - F_*) - \varphi(F(v_{k+1}) - F_*) \right]. \end{aligned} \quad (4.33)$$

Summing and telescoping (4.33) yields

$$\sum_{k=0}^{\infty} \|v_{k+1} - x_k\| < \infty. \quad (4.34)$$

It remains to control the selected steps. Applying the same concavity argument to (4.22) gives

$$\begin{aligned} \|x_{k+1} - x_k\|^2 & \leq \frac{L + \frac{1}{\lambda} + \tau}{\min\left\{c, \frac{1}{\lambda} - \frac{L}{2} - \tau\right\}} \|v_k - x_{k-1}\| \\ & \times \left[\varphi(F(v_k) - F_*) - \varphi(F(v_{k+2}) - F_*) \right]. \end{aligned}$$

Consequently, the Cauchy-Schwarz inequality and telescoping give

$$\begin{aligned} \left(\sum_{k=K}^N \|x_{k+1} - x_k\| \right)^2 & \leq \frac{L + \frac{1}{\lambda} + \tau}{\min\left\{c, \frac{1}{\lambda} - \frac{L}{2} - \tau\right\}} \\ & \times \left(\sum_{k=K}^N \|v_k - x_{k-1}\| \right) \\ & \times [\varphi(F(v_K) - F_*) + \varphi(F(v_{K+1}) - F_*)]. \end{aligned}$$

The right-hand side remains bounded as $N \rightarrow \infty$ by (4.34). Therefore,

$$\sum_{k=0}^{\infty} \|x_{k+1} - x_k\| < \infty.$$

Thus, $\{x_k\}$ is Cauchy and converges to some x^* . Moreover,

$$\|v_{k+1} - x^*\| \leq \|v_{k+1} - x_k\| + \|x_k - x^*\| \longrightarrow 0,$$

so $v_k \rightarrow x^*$. Finally, Theorem 4.6 gives

$$0 \in \partial F(x^*).$$

□

Corollary 4.12. *Under the assumptions of Theorem 4.11, suppose that the desingularizing function has the form*

$$\varphi(s) = \frac{C}{1-\theta} s^{1-\theta}, \quad C > 0, \quad \theta \in [0, 1). \quad (4.35)$$

Then, for all sufficiently large k :

- (1) if $\theta = 0$, the sequence becomes stationary after finitely many iterations;
- (2) if $\theta \in (0, 1/2]$, $F(v_k) - F_*$ converges at least linearly;
- (3) if $\theta \in (1/2, 1)$, then

$$F(v_k) - F_* = O\left(k^{-\frac{1}{2\theta-1}}\right)$$

Proof. Since $\varphi'(s) = Cs^{-\theta}$, the KL inequality and (4.20) give

$$(F(v_{k+1}) - F_*)^\theta \leq C \left(L + \frac{1}{\lambda} + \tau \right) \|v_{k+1} - x_k\|. \quad (4.36)$$

Combining (4.36) with (4.21) yields

$$F(v_k) - F(v_{k+1}) \geq \frac{\frac{1}{\lambda} - \frac{L}{2} - \tau}{C^2 \left(L + \frac{1}{\lambda} + \tau \right)^2} (F(v_{k+1}) - F_*)^{2\theta}. \quad (4.37)$$

The finite, linear, and sublinear objective-value rates follow from the standard scalar analysis of (4.37) [1, 14, 13]. $\square$

4.6. Summary of the convergence argument. The fixed step $\lambda < 1/L$ provides the descent estimate, while the spectral step used to compute z_{k+1} may be larger than $1/L$. The two acceptance conditions transfer the fixed-step estimate to the selected iterate and control its displacement. Under summable proximal-value errors, the generated sequence is asymptotically regular, and every accumulation point is stationary.

Under the additional relative-residual and KL assumptions, the reference descent and two-step selected-iterate estimate yield finite length and full-sequence convergence. The objective-value rates in Corollary 4.12 follow when the KL desingularizing function has the power form (4.35).

5. NUMERICAL EXPERIMENTS

We evaluate HA-AIPG on deterministic MNIST autoencoder training, MNIST classification, and convex logistic regression using the `mushrooms` and `w8a` datasets. The comparison includes IPG, AIPG, and nmAIPG [13], niAPG [29], newAPG_{vs} [18], and the AdaPGNC variants [30].

All methods were implemented in MATLAB R2025b and executed sequentially without GPU acceleration. Within each experiment, they used the same data, initial point, and iteration budget. Because their per-iteration costs differ, we also report gradient, objective, and proximal evaluation counts.

For a reference step $\lambda > 0$, define the proximal-gradient mapping

$$G_\lambda(x) = \frac{1}{\lambda} [x - \text{prox}_{\lambda h}(x - \lambda \nabla g(x))].$$

Under the assumptions of the convergence analysis, $G_\lambda(x) = 0$ implies $0 \in \nabla g(x) + \partial h(x)$. When $h = 0$, as in the experiments below, $G_\lambda(x) = \nabla g(x)$.

We report the relative stationarity measure

$$\mathcal{R}_k = \frac{\|G_\lambda(x_k)\|}{\max\{\|G_\lambda(x_0)\|, \epsilon_{\text{mach}}\}}, \quad (5.1)$$

where $\epsilon_{\text{mach}} \approx 2.22 \times 10^{-16}$ is the machine precision for double-precision arithmetic. A smaller $\mathcal{R}_k$ indicates a greater reduction in the stationarity residual. It is reported together with the objective value because different methods may approach different stationary points.

5.1. Autoencoder for MNIST. We consider deterministic MNIST autoencoder training on the MNIST dataset [8], following the neural-network setting of [30] and the implementation pattern in [2, 9]. Each image is flattened as $u_i \in [0, 1]^{784}$, and the network architecture is

$$784-32-16-32-784.$$

The hidden layers use either

$$\phi_{\text{sig}}(s) = \frac{1}{1 + \exp(-s)} \quad \text{or} \quad \phi_{\text{ReLU}}(s) = \max\{0, s\},$$

while the output layer uses a sigmoid activation. The network contains 52,064 trainable parameters.

Let θ collect the network parameters and let $D_\theta(u_i)$ denote the reconstruction of u_i . For $N = 10,000$ training images, the objective is

$$\min_{\theta} F(\theta) = \frac{1}{2N(784)} \sum_{i=1}^N \|D_\theta(u_i) - u_i\|^2.$$

The sigmoid configuration is smooth. The ReLU configuration is only piecewise differentiable and therefore lies outside Assumption 2.1; it is retained as an empirical comparison, and its gradient-based residual is used only as a numerical diagnostic. No artificial proximal error is added.

All methods use the same data split, LeCun-normal initialization, and budget of 500 iterations. A disjoint set of 2,000 images is reserved for reconstruction and is not used for training or parameter selection.

We use empirical working curvature estimates obtained from 1,024 training samples and checked by an initial majorization test:

$$\widehat{L}_{\text{ReLU}} = 3.247 \times 10^{-1}, \quad \widehat{L}_{\text{sig}} = 6.149 \times 10^{-3}.$$

These values are numerical estimates rather than certified global bounds. The empirical reference step is

$$\lambda = \frac{0.9}{\widehat{L}}, \tag{5.2}$$

giving $\lambda_{\text{ReLU}} = 2.772$ and $\lambda_{\text{sig}} = 1.464 \times 10^2$. Table 1 lists the remaining parameters. For HA-AIPG, the geometric mean (3.4) is used.

TABLE 1. Algorithmic parameters for the MNIST autoencoder experiment

Method	Parameter setting
IPG	Fixed step λ given by (5.2).
AIPG	Fixed step λ given by (5.2), with the two-candidate selection rule.
nmAIPG	Fixed step λ given by (5.2), with $\delta = 0.6$.
niAPG	Fixed step λ given by (5.2), with nonmonotone window $q = 5$.
newAPG _{vs}	$\delta = 0.8$, $c_{\text{new}} = 10^4$, $Q_{k+1} = 0.99Q_k$, error exponent 1.1, $\mu_0 = 0.99$, and $\mu_1 = 0.95$.
AdaPGNC-1	Initial step $\lambda_0 = 1$, initial control parameter $\rho_0 = 10^{10}$, and curvature floor 10^{-14} .
AdaPGNC-2	Initial step $\lambda_0 = 1$, initial control parameter $\rho_0 = 10^{10}$, and curvature floor 10^{-14} .
HA-AIPG	Geometric BB trial step; $\gamma_k \in [0.05/\widehat{L}, 1.6 \times 10^4/\widehat{L}]$; $\rho_\alpha = 0.8$; $\alpha_{\min} = 10^{-4}$; and $c = 0.01 \left(\frac{1}{2\lambda} - \frac{\widehat{L}}{2} \right)$.

For each activation, let

$$F_{\text{best}} = \min_{\mathcal{M}} \min_{0 \leq k \leq 500} F_{\mathcal{M}}(\theta_k),$$

where $\mathcal{M}$ ranges over all compared methods. The best-observed objective gap is

$$\text{Gap}_{\mathcal{M}} = F_{\mathcal{M}}(\theta_{500}) - F_{\text{best}}. \quad (5.3)$$

This is a gap from the best value observed among the compared methods, not from a known global optimum.

Table 2 reports the final training objective, the objective gap in (5.3), and the relative stationarity measure $\mathcal{R}_{500}$ defined in (5.1). The final column reports the numbers of gradient, additional objective-only, and proximal evaluations, respectively.

TABLE 2. Results for the deterministic MNIST autoencoder after 500 iterations

Method	Objective value	Objective gap	Stationarity ($\mathcal{R}_{500}$)	Grad./Obj./Prox. evals.
<i>ReLU hidden activation</i>				
IPG	0.03791	0.02048	0.08625	501/0/500
AIPG	0.01813	0.00070	0.07656	999/1000/1000
nmAIPG	0.01813	0.00070	0.07656	999/704/704
niAPG	0.02117	0.00374	0.10790	808/500/500
newAPG _{vs}	0.02300	0.00557	0.05041	1000/998/501
AdaPGNC-1	0.02656	0.00914	0.06296	501/0/500
AdaPGNC-2	0.02298	0.00555	0.68850	501/0/500
HA-AIPG	0.01743	$\leq 10^{-5}$	0.15210	999/1064/1000
<i>Sigmoid hidden activation</i>				
IPG	0.03367	0.01915	0.00241	501/0/500
AIPG	0.01920	0.00468	0.14560	999/1000/1000
nmAIPG	0.01920	0.00468	0.14560	999/1000/1000
niAPG	0.02964	0.01512	0.10270	653/500/500
newAPG _{vs}	0.02077	0.00625	0.01869	998/996/502
AdaPGNC-1	0.02693	0.01241	0.03938	501/0/500
AdaPGNC-2	0.02515	0.01063	0.01691	501/0/500
HA-AIPG	0.01452	$\leq 10^{-5}$	0.01505	997/1174/1000

Gaps smaller than the displayed objective precision are reported as $\leq 10^{-5}$. HA-AIPG attained the lowest final objective for both activations, although it did not produce the smallest stationarity residual. This suggests that the methods reached different regions of the nonconvex objective within 500 iterations. For ReLU, 460 trials were accepted with $\alpha_k = 1$, 40 required relaxation, and no hard restart occurred. For sigmoid, 463 full trials and 34 relaxed trials were accepted, while three iterations used the reference point and restarted the momentum. The mean positive relaxation was 0.9775 and 0.9818, respectively.

The trial step exceeded $1/\hat{L}$ in 444 ReLU and 363 sigmoid iterations. Since $\hat{L}$ is empirical, these counts describe only the observed behavior. The final column of Table 2 reports gradient, additional objective, and proximal evaluations. HA-AIPG required more evaluations than IPG and AdaPGNC, with a cost comparable to the two-candidate AIPG scheme. Because $h = 0$, the reported proximal evaluations are identity-map calls.

Figures 1 and 2 show the objective and adaptive step trajectories for the two activation functions. In both experiments, HA-AIPG decreased the objective rapidly and attained the lowest final value among

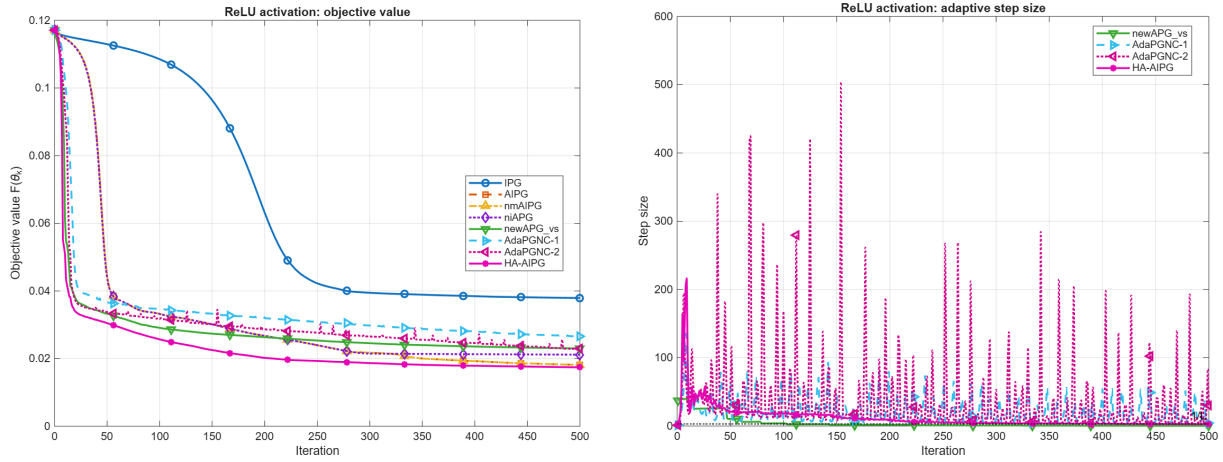


FIGURE 1. Deterministic MNIST autoencoder with ReLU hidden activation: objective values and adaptive trial steps over 500 iterations

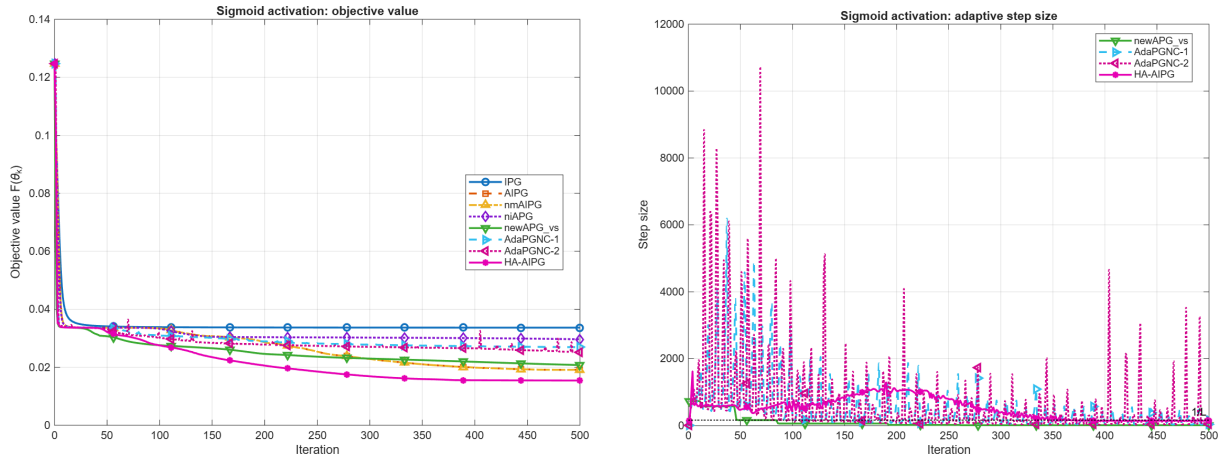


FIGURE 2. Deterministic MNIST autoencoder with sigmoid hidden activation: objective values and adaptive trial steps over 500 iterations

the compared methods. Its trial step varied substantially and frequently exceeded the empirical reference $1/\hat{L}$. These results illustrate the use of the adaptive trial step, but the comparison is based on iterations and should be considered together with the evaluation counts in Table 2.

5.2. MNIST classification. We next consider deterministic multiclass classification on the MNIST dataset. The data selection and initialization follow the protocol of Section 5.1. We used 10,000 training images, 2,000 disjoint validation images, and the official 10,000-image MNIST test set. The same random seed was used for all methods.

The classifier has the architecture

$$784-32-16-10,$$

with 25,818 trainable weights and biases. The two hidden layers use the sigmoid activation, while the output layer uses the softmax function. This architecture follows the dimensions used in the classification experiment of [30].

Let $u_i \in [0, 1]^{784}$ denote an input image, $y_i \in \{0, 1\}^{10}$ be its one-hot label, and $p_\theta(u_i) \in (0, 1)^{10}$ be the softmax output of the network. The smooth training problem is to minimize the full-batch softmax

cross-entropy loss:

$$\min_{\theta} F(\theta) = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^{10} y_{ij} \log p_{\theta,j}(u_i).$$

The local Hessian-norm estimation and initial majorization check were performed as in Section 5.1. They produced the empirical working value

$$\hat{L}_{\text{cls}} = 3.2539, \quad \lambda_{\text{cls}} = \frac{0.9}{\hat{L}_{\text{cls}}} = 0.2766.$$

As in the autoencoder experiment, $\hat{L}_{\text{cls}}$ is an empirically calibrated value and is not claimed to be a certified global Lipschitz constant; consequently, $0.9/\hat{L}_{\text{cls}}$ is an empirical reference step rather than a theoretically safe step.

The parameters of IPG, AIPG, nmAIPG, niAPG, newAPG_{vs}, AdaPGNC-1, and AdaPGNC-2 were identical to those listed in Table 1, but with $\hat{L}$ replaced by $\hat{L}_{\text{cls}}$. For HA-AIPG, we used the BB1 spectral step bounded by

$$\gamma_k \in \left[\frac{0.05}{\hat{L}_{\text{cls}}}, \frac{100}{\hat{L}_{\text{cls}}} \right].$$

The remaining HA-AIPG parameters were set to

$$\rho_{\alpha} = 0.8, \quad \alpha_{\min} = 0.05, \quad c = 0.001 \left(\frac{1}{2\lambda_{\text{cls}}} - \frac{\hat{L}_{\text{cls}}}{2} \right).$$

5.2.1. *Optimization results.* All methods were first run for a fixed budget of 500 iterations. The objective gap is defined as in (5.3), using the smallest training objective observed in the classification experiment. The final column reports gradient, additional objective-only, and proximal evaluations, respectively.

TABLE 3. Optimization results for deterministic MNIST classification after 500 iterations

Method	Objective value	Objective gap	Stationarity ($\mathcal{R}_{500}$)	Grad./Obj./Prox. evals.
IPG	0.719410	0.719400	0.102200	501/0/500
AIPG	0.002824	0.002814	0.001782	999/1000/1000
nmAIPG	0.002824	0.002814	0.001782	999/500/500
niAPG	0.002803	0.002794	0.001760	1000/500/500
newAPG _{vs}	0.043024	0.043014	0.007122	998/995/500
AdaPGNC-1	0.056403	0.056393	0.019420	501/0/500
AdaPGNC-2	0.008612	0.008602	0.003928	501/0/500
HA-AIPG	0.000010	$\leq 10^{-6}$	0.000004	1000/1073/1000

Within the 500-iteration budget, HA-AIPG attained the lowest final training objective and the smallest relative stationarity residual in Table 3. It first satisfied $\mathcal{R}_k \leq 10^{-5}$ at iteration 360. This result required 1,000 gradient, 1,073 additional objective, and 1,000 proximal evaluations. The improvement is therefore measured with respect to iterations and does not imply a lower per-iteration cost than the single-step methods.

Figure 3 shows the objective progress over 500 iterations and the adaptive steps during the first 100 iterations. HA-AIPG decreased the training objective rapidly and continued to a lower final value. Its trial step varied around and above the empirical reference $1/\hat{L}_{\text{cls}}$. These curves are consistent with Table 3, but they are not normalized by the number of evaluations.

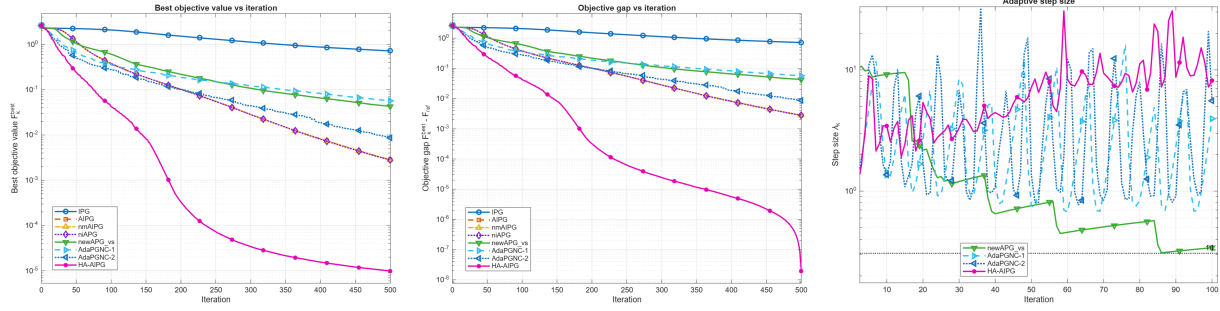


FIGURE 3. Optimization performance for deterministic MNIST classification. The objective curves use 500 iterations, whereas the adaptive-step panel shows the first 100 iterations

5.2.2. *Validation-selected prediction results.* A low training cross-entropy does not necessarily imply improved test accuracy. To separate optimization performance from prediction performance, classification accuracy and cross-entropy were evaluated every ten iterations. For each method, the checkpoint k^* was selected by the smallest validation cross-entropy. The test set was evaluated only after this checkpoint had been fixed and was not used for model selection.

TABLE 4. MNIST prediction results at the checkpoint selected by validation cross-entropy

Method	k^*	Training CE	Validation CE	Test CE	Test accuracy (%)
IPG	500	0.71940	0.73160	0.71210	81.62
AIPG	250	0.05494	0.24830	0.22980	93.91
nmAIPG	250	0.05494	0.24830	0.22980	93.91
niAIPG	250	0.05499	0.24840	0.22940	93.91
newAPG _{vs}	200	0.21870	0.49530	0.44140	89.47
AdaPGNC-1	490	0.05880	0.23100	0.19600	94.38
AdaPGNC-2	270	0.05904	0.22780	0.19590	94.28
HA-AIPG	90	0.05786	0.24190	0.23440	93.96

Table 4 shows that validation-based selection chose the HA-AIPG checkpoint at $k^* = 90$, earlier than the checkpoints selected for the other methods. Its test accuracy was 93.96%, compared with the highest value of 94.38% for AdaPGNC-1 and 94.28% for AdaPGNC-2. Thus, HA-AIPG reached a comparable predictive result at an earlier checkpoint, although its lower training objective did not yield the highest test accuracy. The common 500-iteration budget allows the same validation-based selection rule to be applied to methods with different rates of progress.

Figure 4 shows that HA-AIPG reduced the training cross-entropy rapidly, while the training and validation accuracies stabilized earlier. This illustrates the distinction between minimizing the training objective and improving predictive accuracy. HA-AIPG accepted all 500 candidates, including 461 full spectral trials and 39 relaxed trials, with a mean relaxation of 0.9751 and no hard restart. Hence, this experiment mainly reflects the accepted spectral-step behavior rather than the restart mechanism.

5.3. **Convex logistic regression.** We finally consider the regularized logistic-regression problem used in [30]:

$$\min_{x \in \mathbb{R}^n} F(x) = -\frac{1}{m} \sum_{i=1}^m \left[y_i \log \sigma(a_i^\top x) + (1 - y_i) \log(1 - \sigma(a_i^\top x)) \right] + \frac{\gamma}{2} \|x\|^2, \quad (5.4)$$

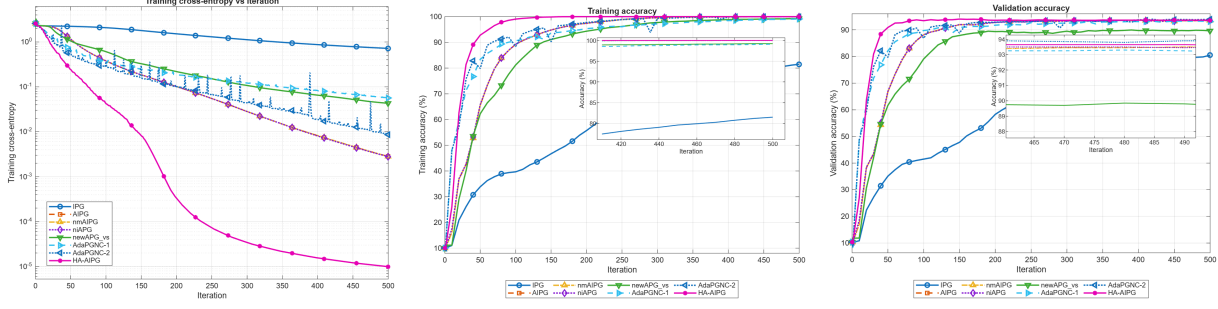


FIGURE 4. Performance trajectories for deterministic MNIST classification. The insets in the two accuracy panels show the final 100 iterations

where $\sigma(t) = 1/(1 + e^{-t})$, $a_i \in \mathbb{R}^n$, and $y_i \in \{0, 1\}$. Problem (5.4) is smooth and convex, so $h = 0$ and the proximal mapping is the identity. We use the mushrooms and w8a datasets with the settings in Table 5.

TABLE 5. Dataset settings for the logistic-regression experiment

Dataset	m	n	L_{tab}	γ	Iteration budget
mushrooms	8,124	112	2.59	L_{tab}/m	1,000
w8a	49,749	300	0.66	L_{tab}/m	3,000

Following the experimental scaling in [30], Table 5 lists L_{tab} and uses $\gamma = L_{\text{tab}}/m$. In our implementation, the same tabulated values are used as empirical working curvature parameters, $\hat{L} = L_{\text{tab}}$. Thus, the fixed-step methods use the empirical reference step $\lambda = 0.9/\hat{L}$, giving $\lambda = 0.3474$ for mushrooms and $\lambda = 1.3611$ for w8a. These numerical parameters are not presented as new certifications of a global Lipschitz constant. All methods start from $x_0 = 0$.

Table 6 summarizes the method-specific parameters used in these tests.

TABLE 6. Algorithmic parameters for the logistic-regression experiment

Method	Parameter setting
IPG	Fixed step $\lambda = 0.9/\hat{L}$ as described above.
AIPG	Fixed step λ , with the two-candidate selection rule.
nmAIPG	Fixed step λ , with $\delta = 0.6$.
niAPG	Fixed step λ , with nonmonotone window $q = 5$.
newAPG _{vs}	$\delta = 0.8$, $c_{\text{new}} = 10^4$, $Q_{k+1} = 0.99Q_k$, error exponent 1.1, $\mu_0 = 0.99$, and $\mu_1 = 0.95$.
AdaPGNC-BB-1	Short BB step of [30]; $\lambda_0 = 10^{-10}$, $\rho_0 = 10^{10}$, and the $\rho_k^{(1)}$ growth rule.
AdaPGNC-BB-2	Short BB step of [30]; $\lambda_0 = 10^{-10}$, $\rho_0 = 10^{10}$, and the $\rho_k^{(2)}$ growth rule.
HA-AIPG	Geometric BB trial step; $\gamma_k \in [10^{-4}/\hat{L}, 10^4/\hat{L}]$; $\rho_\alpha = 0.5$; $\alpha_{\min} = 10^{-3}$; and $c = 10^{-3} \left(\frac{1}{2\lambda} - \frac{\hat{L}}{2} \right)$.

The reference objective is the smallest value observed over all runs, including the extended AdaPGNC-BB runs. Its displayed values are 0.02623884 for mushrooms and 0.12320374 for w8a; the gaps were

computed using the full recorded precision. The objective value, objective gap, and gradient norm are the smallest values observed along each trajectory, while the evaluation counts cover the complete run. Gaps below 10^{-14} are reported as being at numerical working precision.

TABLE 7. Results for logistic regression on the mushrooms dataset (1,000 iterations). The last column reports gradient, additional objective-only, and proximal evaluations

Method	Objective value	Objective gap	Gradient norm	Grad./Obj./Prox. evals.
IPG	0.03717367	0.01093483	0.0065363	1001 / 0 / 1000
AIPG	0.02624068	1.8428×10^{-6}	6.0143×10^{-5}	1999 / 2000 / 2000
nmAIPG	0.02624065	1.8171×10^{-6}	5.9842×10^{-5}	1999 / 1026 / 1026
niAPG	0.02623884	2.1018×10^{-11}	2.3164×10^{-7}	1993 / 1000 / 1000
newAPG _{vs}	0.02623884	$\leq 10^{-14}$	8.7938×10^{-11}	1682 / 1680 / 1074
AdaPGNC-BB-1	0.02623884	$\leq 10^{-14}$	1.0384×10^{-12}	1001 / 0 / 1000
AdaPGNC-BB-2	0.02623884	$\leq 10^{-14}$	3.5951×10^{-9}	1001 / 0 / 1000
HA-AIPG	0.02623884	$\leq 10^{-14}$	1.9692×10^{-13}	1471 / 8397 / 2402

Table 7 shows that, on mushrooms, newAPG_{vs}, AdaPGNC-BB-1, and HA-AIPG reached the reference objective to the reported precision. HA-AIPG also attained the smallest reported gradient norm. The last three columns report the objective gap, gradient norm, and numbers of gradient, additional objective, and proximal evaluations. HA-AIPG required more objective and proximal evaluations because of its candidate-selection tests; therefore, its residual improvement does not imply a lower computational cost.

Figure 5 shows that the adaptive methods reduced the objective and gradient norm substantially faster than IPG. AdaPGNC-BB-2 produced the fastest early reduction in the objective gap, whereas HA-AIPG continued to a smaller gradient norm. The step-size panel shows that all adaptive methods used variable steps, frequently above the reference $1/\hat{L}$.

TABLE 8. Results for logistic regression on the w8a dataset (3,000 iterations). The last column reports gradient, additional objective-only, and proximal evaluations

Method	Objective value	Objective gap	Gradient norm	Grad./Obj./Prox. evals.
IPG	0.13422566	0.01102192	0.0014869	3001 / 0 / 3000
AIPG	0.12320480	1.0620×10^{-6}	9.1241×10^{-6}	5999 / 6000 / 6000
nmAIPG	0.12320480	1.0596×10^{-6}	9.1128×10^{-6}	5999 / 3018 / 3018
niAPG	0.12320374	4.1681×10^{-10}	2.1593×10^{-7}	5994 / 3000 / 3000
newAPG _{vs}	0.12320374	1.1116×10^{-14}	7.0540×10^{-10}	5251 / 5249 / 3314
AdaPGNC-BB-1	0.12320374	$\leq 10^{-14}$	1.6630×10^{-14}	3001 / 0 / 3000
AdaPGNC-BB-2	0.12320374	$\leq 10^{-14}$	4.2583×10^{-10}	3001 / 0 / 3000
HA-AIPG	0.12320374	$\leq 10^{-14}$	1.1212×10^{-12}	4846 / 34358 / 7876

Table 8 shows that, on w8a, HA-AIPG and AdaPGNC-BB-1 reached the reference objective to numerical precision. AdaPGNC-BB-1 attained the smallest gradient norm, while HA-AIPG obtained a

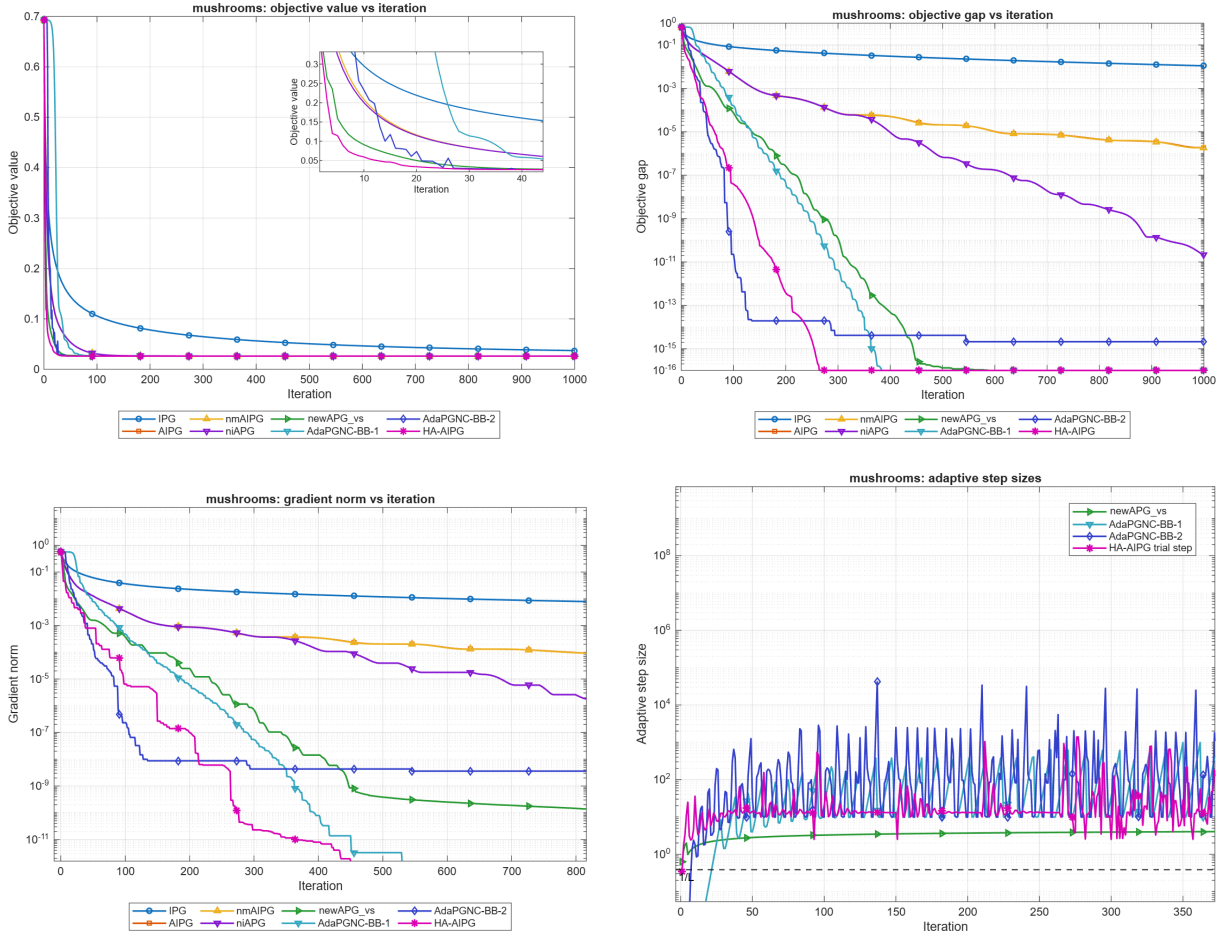


FIGURE 5. Convergence trajectories and adaptive step sizes for the mushrooms dataset

gradient norm of 1.1212×10^{-12} . HA-AIPG again required additional objective and proximal evaluations for candidate selection, so the table shows a trade-off between objective reduction and evaluation cost rather than uniform dominance.

Figure 6 shows the rapid initial objective reduction of HA-AIPG and the subsequent progress of the adaptive methods toward the reference value. AdaPGNC-BB-1 eventually attained the smallest gradient norm, while HA-AIPG reached the reference objective with a small residual. The adaptive-step trajectories also show that the methods used different step-size patterns, with HA-AIPG testing steps above $1/\hat{L}$ throughout much of the run.

Summary of the numerical results. Across the three experiments, HA-AIPG provided strong objective reduction with respect to outer iterations. It attained the lowest final autoencoder objectives for both activations, the lowest training objective in MNIST classification, and the reference objective in both logistic-regression tests. Other adaptive methods nevertheless attained better stationarity, predictive accuracy, or evaluation cost in some settings. The results therefore show a trade-off between objective progress per outer iteration and the additional work required by the two-candidate selection mechanism.

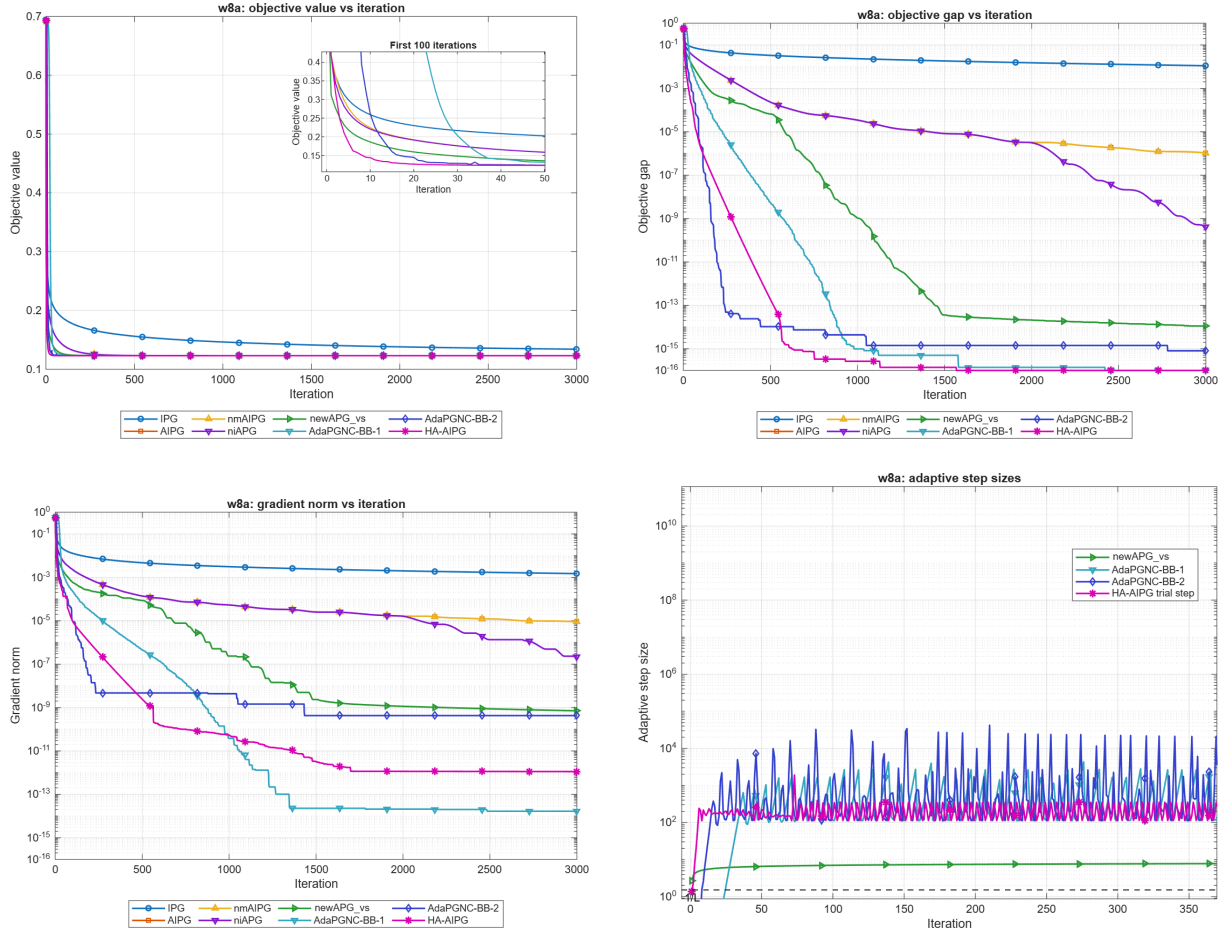


FIGURE 6. Convergence trajectories and adaptive step sizes for the w8a dataset

6. CONCLUSION

This paper introduced HA-AIPG, which combines an accelerated spectral trial with a reference proximal step, adaptive relaxation, and momentum restart. The spectral candidate may use a step larger than $1/L$, whereas the reference candidate supplies the descent estimate used in the convergence analysis. Summable inexactness yields stationary accumulation points. Full-sequence convergence additionally requires the stated relative-residual, boundedness, continuity, and KL assumptions.

The numerical results indicate that HA-AIPG can be useful when objective progress within a limited number of iterations is the main consideration. In these tests, its selection mechanism produced lower objective values than the AIPG-family baselines and was competitive with single-step adaptive methods. This behavior required additional gradient, objective, and proximal evaluations, and other methods gave better stationarity or predictive accuracy in some settings. HA-AIPG should therefore be viewed as a monitored alternative when candidate evaluations are affordable, rather than as a uniformly faster method. Reducing the cost of the reference candidate is a natural direction for future work.

STATEMENTS AND DECLARATIONS

The authors declare that they have no conflicts of interest.

Declaration of generative AI and AI-assisted technologies. During the preparation of this work, the authors used OpenAI ChatGPT/Codex for language editing, LaTeX preparation, and assistance with

MATLAB code. The authors reviewed and verified all mathematical content, code, results, and interpretations and take full responsibility for the final manuscript.

REFERENCES

- [1] H. Attouch, J. Bolte, and B. F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods. *Mathematical Programming*, 137(1-2):91-129, 2013.
- [2] M. Bakrey. MNIST by using autoencoder. Kaggle Notebook, 2026.
- [3] J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1):141-148, 1988.
- [4] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183-202, 2009.
- [5] Y. Bello-Cruz, M. L. N. Gonçalves, J. G. Melo, and C. Mohr. A relative inexact proximal gradient method with an explicit linesearch. *Journal of Optimization Theory and Applications*, 206(1):Article ID 9, 2025.
- [6] R. I. Boţ, E. R. Csetnek, and S. C. László. An inertial forward-backward algorithm for the minimization of the sum of two nonconvex functions. *EURO Journal on Computational Optimization*, 4(1):3-25, 2016.
- [7] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717-772, 2009.
- [8] L. Deng. The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141-142, 2012.
- [9] A. Fellah. Autoencoders with MNIST dataset. Kaggle Notebook, 2026.
- [10] Y. Feng, X. Huang, L. Shi, Y. Yang, and J. A. K. Suykens. Learning with the maximum correntropy criterion induced losses for regression. *Journal of Machine Learning Research*, 16(30):993-1034, 2015.
- [11] S. Ghadimi and G. Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming*, 156(1-2):59-99, 2016.
- [12] G. R. S. Grillo, E. S. Helou, and S. A. Santos. Inexact FISTA-like methods with adaptive backtracking. *Computational Optimization and Applications*, 93(3):1259-1304, 2026.
- [13] B. Gu, D. Wang, Z. Huo, and H. Huang. Inexact proximal gradient methods for non-convex and non-smooth optimization. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, volume 32, New Orleans, Louisiana, USA, 2018. AAAI Press, California, USA.
- [14] H. Li and Z. Lin. Accelerated proximal gradient methods for nonconvex programming. In: C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 379-387. Curran Associates Inc., 2015.
- [15] H. Li and Z. Lin. Restarted nonconvex accelerated gradient descent: No more polylogarithmic factor in the $\mathcal{O}(\epsilon^{-7/4})$ complexity. *Journal of Machine Learning Research*, 24(157):1-37, 2023.
- [16] J. Liang and R. D. C. Monteiro. Average curvature FISTA for non-convex smooth composite optimization problems. *Computational Optimization and Applications*, 86(1):275-302, 2023.
- [17] Q. Lin and Y. Xu. Reducing the complexity of two classes of optimization problems by an inexact accelerated proximal gradient method. *SIAM Journal on Optimization*, 33(1):1-35, 2023.
- [18] H. Liu, T. Wang, and Z. Liu. A nonmonotone accelerated proximal gradient method with variable stepsize strategy for nonsmooth and nonconvex minimization problems. *Journal of Global Optimization*, 89(4):863-897, 2024.
- [19] Y. E. Nesterov. A method for solving the convex programming problem with convergence rate $\mathcal{O}(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372-376, 1983.
- [20] P. Ochs, Y. Chen, T. Brox, and T. Pock. iPiano: Inertial proximal algorithm for nonconvex optimization. *SIAM Journal on Imaging Sciences*, 7(2):1388-1419, 2014.
- [21] B. O'Donoghue and E. Candès. Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, 15(3):715-732, 2015.
- [22] M. Pandey, H. Zhang, and Z. Zheng. Adaptive proximal Barzilai-Borwein methods for nonlinear composite optimization. *Numerical Algorithms*, 2026.
- [23] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127-239, 2014.
- [24] M. Raydan. On the Barzilai and Borwein choice of steplength for the gradient method. *IMA Journal of Numerical Analysis*, 13(3):321-326, 1993.
- [25] M. Raydan. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM Journal on Optimization*, 7(1):26-33, 1997.

- [26] M. Schmidt, N. Le Roux, and F. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1458-1466, 2011. Curran Associates Inc., New York.
- [27] W. Su, S. Boyd, and E. J. Candès. A differential equation for modeling Nesterov’s accelerated gradient method: Theory and insights. *Journal of Machine Learning Research*, 17(153):1-43, 2016.
- [28] S. Villa, S. Salzo, L. Baldassarre, and A. Verri. Accelerated and inexact forward-backward algorithms. *SIAM Journal on Optimization*, 23(3):1607-1633, 2013.
- [29] Q. Yao, J. T. Kwok, F. Gao, W. Chen, and T.-Y. Liu. Efficient inexact proximal gradient algorithm for non-convex problems. In: C. Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 3308-3314. IJCAI Organization, Melbourne, Australia, 2017. AAAI Press, USA.
- [30] Z. Ye, S. Ma, J. Yang, and D. Zhou. A simple adaptive proximal gradient method for nonconvex optimization. ArXiv preprint ArXiv:2510.06079, 2025.
- [31] D. Zhou, S. Ma, and J. Yang. AdaBB: Adaptive Barzilai–Borwein method for convex optimization. *Mathematics of Operations Research*, 51(1):715-745, 2025.