Alternating Iteratively Reweighted ℓ_1 and Subspace Newton Algorithms for Nonconvex Sparse Optimization

Hao Wang*[1], Xiangyu Yang[2], and Yichen Zhu[3]

[1]School of Information Science and Technology, ShanghaiTech University
[2]School of Mathematics and Statistics, Henan University, Center for Applied Mathematics of Henan Province, Henan
[3]School of Information Science and Technology, ShanghaiTech University
[1]*haw309@gmail.com*
[2]*yangxy@henu.edu.cn*
[3]*zhuych2022@shanghaitech.edu.cn*

Abstract

This paper presents a novel hybrid algorithm for minimizing the sum of a continuously differentiable loss function and a nonsmooth, possibly nonconvex, sparsity-promoting regularizer. The proposed method adaptively switches between solving a reweighted ℓ_1-regularized subproblem and performing an inexact subspace Newton step. The reweighted ℓ_1-subproblem admits an efficient closed-form solution via the soft-thresholding operator, thereby avoiding the computational overhead of proximity operators. As the algorithm approaches an optimal solution, it identifies a smooth active manifold and guarantees that nonzero components remain uniformly bounded away from zero. On this manifold, the algorithm transitions to a perturbed regularized

Newton step, accelerating local convergence. We establish global convergence to a critical point. Under the Kurdyka-Łojasiewicz property, the algorithm converges locally at a linear rate; when the subspace Newton subproblem is solved exactly, the local convergence rate improves to quadratic. Numerical experiments show that the proposed algorithm outperforms existing methods in both efficiency and solution quality across various model prediction problems.

Keywords— nonconvex nonsmooth regularization, smooth active manifold, subspace Newton method

1 Introduction

In this paper, we consider optimization problems of the form

$$\min_{x\in\mathbb{R}^n} \quad F(x) = f(x) + \lambda h(x), \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is proper and twice continuously differentiable but possibly nonconvex, $\lambda > 0$ refers to the regularization parameter, and the regularization function $h : \mathbb{R}^n \to [0, +\infty)$ is a sum of composite functions as follows

$$h(x) := \sum_{i=1}^{n} (r \circ |\cdot|)(x_i), \ \forall x \in \mathbb{R}^n. \tag{2}$$

The function r satisfies the following assumptions:

Assumption 1.1. *$r : \mathbb{R}_+ \to \mathbb{R}_+$ is $\mathcal{C}^2$-smooth on $\mathbb{R}_{++}$, satisfies $r(0) = 0$, is concave and $r'(t) \geqslant 0$ on $\mathbb{R}_{++}$.*

Under Assumption 1.1, the composite regularizer h possesses the bias-reduction properties of popular nonconvex surrogates for the ℓ_0 (quasi-)norm. Such nonconvex surrogates are widely employed in sparse learning, model compression, compressive sensing, and signal/image processing; see, e.g., (Chen and Zhou 2010, Zhao et al. 2023, Sleem et al. 2024, McCulloch et al. 2024). Table 1 lists several commonly used choices of r, together with their first and second derivatives evaluated at $|x_i|$ and in the limit as $|x_i| \to 0^+$.

1.1 Related Work

First-order methods. Numerous first-order methods have been developed for nonconvex sparse optimization problems. A prominent focus has been the $\ell_p (0 <$

Table 1: Examples of concave regularizers $r(t)$ parameterized by $p \in (0, 1)$. Primes denote derivatives with respect to the scalar argument t; in the table we then evaluate these at $t = |x_i|$.

Regularizer	$h(\|x\|)$	$r'(\|x_i\|)$	$r'(0^+)$	$r''(\|x_i\|)$	$r''(0^+)$
LPN (Fazel et al. 2003)	$\sum_{i=1}^n (\|x_i\|)^p$	$p(\|x_i\|)^{p-1}$	$+\infty$	$p(p-1)\|x_i\|^{p-2}$	$-\infty$
LOG (Lobo et al. 2007)	$\sum_{i=1}^n \log(1 + \frac{\|x_i\|}{p})$,	$\frac{1}{\|x_i\|+p}$	$\frac{1}{p}$	$-\frac{1}{(\|x_i\|+p)^2}$	$-\frac{1}{p^2}$
FRA (Fazel et al. 2003)	$\sum_{i=1}^n \frac{\|x_i\|}{\|x_i\|+p}$	$\frac{p}{(\|x_i\|+p)^2}$	$\frac{1}{p}$	$-\frac{2p}{(\|x_i\|+p)^3}$	$-\frac{2}{p^2}$
TAN (Candes et al. 2008)	$\sum_{i=1}^n \arctan(\frac{\|x_i\|}{p})$	$\frac{p}{p^2+\|x_i\|^2}$	$\frac{1}{p}$	$-\frac{2p\|x_i\|}{(p^2+\|x_i\|^2)^2}$	0
EXP (Bradley et al. 1998)	$\sum_{i=1}^n 1 - e^{-\frac{\|x_i\|}{p}}$	$\frac{1}{p}e^{-\frac{\|x_i\|}{p}}$	$\frac{1}{p}$	$-\frac{1}{p^2}e^{-\frac{\|x_i\|}{p}}$	$-\frac{1}{p^2}$

$p < 1$) regularization, recognized as a challenging and notable representative regularizer in (2). A widely-used approach for ℓ_p-regularized problems is the forward-backward (FB) algorithm, which updates iterates by solving a proximal subproblem. However, closed-form *global* solutions to the proximal subproblem are only available for specific values of p, such as $p = 1/2$ and $p = 2/3$ (Xu et al. 2012). For other values of p or alternative nonconvex penalties, one should employ iterative solvers for each proximal step (Liu and Lin 2024), thereby incurring additional computational overhead. To address the inefficiency of solving nonconvex subproblems, several first-order methods based on smoothing techniques have been proposed. For example, one replaces $|x_i|^p$ by the smooth surrogate $(x_i^2 + \epsilon)^{p/2}$ with $\epsilon > 0$ (Lai et al. 2013, Lu 2014, Wang et al. 2021a). While this yields simpler subproblems, it sacrifices accuracy and typically requires solving a sequence of smoothed problems as $\epsilon \to 0$. Another approach is the *iteratively reweighted* ℓ_1 (IRℓ_1) method (Wang et al. 2021a), which approximates the ℓ_p^p regularization by a sequence of weighted ℓ_1 norms. This method benefits from closed-form subproblem solutions via soft-thresholding and can be directly applied to various nonconvex regularization problems. In addition, it exhibits *support identification* properties, enabling it to rapidly detect zero components and preserving the sign of iterates near optimal solutions (Wang et al. 2021a, 2023, 2022). Our approach builds upon these reweighting strategies while incorporating second-order information to further enhance local convergence rate and improving solution quality.

Second-order methods. Second-order methods, such as proximal Newton methods (Lee et al. 2014, Yue et al. 2019, Mordukhovich et al. 2023), provide a more powerful framework for tackling nonconvex optimization. Yet, relatively few have been tailored to sparse, nonconvex objectives. An early contribution by Chen (Chen et al. 2013) proposed the Smoothing Trust Region Newton (STRN) method,

which applies smoothing techniques to penalty functions and solves a sequence of trust-region subproblems. While STRN guarantees global convergence to points satisfying affine-scaled second-order necessary conditions, it omits any local-rate analysis and requires careful selection of the smoothing parameter. More recent work have turned to subspace- and manifold-based Newton methods (Bareilles et al. 2023, Wu et al. 2023, Zhou et al. 2023). These methods leverage support detection or manifold identification techniques to restrict optimization to a reduced subspace, thereby reducing the computational cost of Newton steps. For example, the manifold Newton method (Bareilles et al. 2023) employs proximal gradient steps to identify the manifold containing the optimal solution, followed by Newton updates confined to the identified manifold.

Hybrid methods. Hybrid first- and second-order methods have recently attracted considerable interest. These algorithms employ inexpensive first-order updates during the early iterations and switch to second-order updates for rapid local convergence. One prominent example is the Hybrid Proximal Gradient and Subspace Regularized Newton Method (HpgSRN) (Wu et al. 2023), which alternates between proximal gradient steps and subspace Newton steps. HpgSRN therefore inherits the low per-iteration cost of first-order methods and the local superlinear convergence of second-order schemes; under the Kurdyka-Łojasiewicz (KL) property combined with a local error bound condition, it is shown to converge superlinear. The Proximal Semismooth Newton Pursuit (PCSNP) algorithm (Zhou et al. 2023) follows a similar philosophy but employs a more aggressive support-detection rule. However, it may also trigger an excessive number of quadratic subproblem solves in early iterations, leading to increased computational cost.

1.2 Our Work and Contributions

In this work, we propose a novel hybrid algorithm—the subspace **I**teratively **Re**weighted ℓ_1 and subspace **N**ewton **A**lgorithm (IReNA)—for solving problem (1). IReNA proceeds by alternating between solving a subspace reweighted ℓ_1 subproblem and a subspace Newton subproblem. At each iteration, it uses stationarity measures to differentiate between updating the zero and nonzero components of the current iterate, thereby optimizing over the most relevant subspace. Specifically, it first solves a reweighted ℓ_1 subproblem restricted to a candidate support to determine whether the support of the iterates has stabilized. Then, once stabilization is detected, the algorithm switches to an (inexact) Newton step confined to the identified subspace to accelerate local convergence. Through extensive experiments on synthetic and real-world prediction tasks, we demonstrate that IReNA consistently outperforms existing state-of-the-art methods. In particular, compared to STRN (Chen et al. 2013), HpgSRN (Wu et al. 2023) and PCSNP (Zhou et al. 2023),

IReNA offers the following key advantages:

(i) **General applicability to nonconvex regularization**: IReNA applies to a broad class of nonconvex regularized problems—not just the ℓ_p norm. Each reweighted ℓ_1 subproblem admits a closed-form solution via the soft-thresholding, thereby avoiding the computational overhead of iterative proximal-subproblem solvers. Although we assume the nonconvex regularizer is $\mathcal{C}^2$-smooth, the framework extends naturally to structured nonsmooth penalties such as SCAD and MCP (see Remark 2.7).

(ii) **Efficient adaptive subspace optimization**: At each iteration, IReNA computes component-wise optimality residuals to identify which variables most violate stationarity—both among the current nonzeros and in their complement. It then retains only those components with significant residuals for the next update, holding the rest fixed. By confining each step to this adaptively chosen low-dimensional subspace, the algorithm concentrates effort where it matters most, achieving rapid early-stage progress and substantially reducing per-iteration cost.

(iii) **Stable sign preservation**: The algorithm guarantees that once a coordinate becomes nonzero, its sign remains fixed and is uniformly bounded away from zero near the solution. After sign stabilization, IReNA smoothly transitions to an inexact Newton update—solving a perturbed quadratic programming (QP) in the identified subspace—to accelerate local convergence.

(iv) **Comprehensive convergence guarantees**: We prove global convergence of the iterate sequence to a first-order critical point and establish the local linear convergence under the KL property. Moreover, if the subspace QP is solved exactly, local quadratic convergence is achieved. Additionally, by replacing the QP with a trust-region Newton subproblem, IReNA guarantees convergence to a second-order optimal solution.

(v) **Superior empirical performance**: Extensive numerical experiments on both synthetic and real-world datasets demonstrate that IReNA consistently outperforms state-of-the-art hybrid methods—achieving faster runtimes, lower objective values, and comparable or better sparsity.

We summarize the comparison of convergence results for the related first- and second-order algorithms in Table 2. Throughout the paper, to preserve the flow of the main exposition, we have relocated all technical proofs from the main text to the appendices, which are available in the online supplementary materials.

Table 2: Comparison of convergence results for state-of-the-art algorithms for solving (1). (a) f is Lipschitz differentiable on a bounded set. (b) f is strongly smooth. (c) f is Lipschitz twice differentiable on the support set. (d) Curve ratio condition. (e) KL property. (f) Positive definiteness of generalized Hessian. (g) ∇f is strongly semismooth on a bounded set. (h) f is smooth and convex. (i) Nonsingular local minimizer.

Problems.	Algorithms	Global convergence	Local convergence
$h(x) = \|x\|_p^p$ with simple proximal operator.	HpgSRN(2023)	(c)(d)(e)	superlinear, (e)
	PCSNP(2023)	(a)	quadratic, (f)(g)
$h(x) = \|x\|_p^p$, $p \in (0,1)$	IRℓ_α(2014)	(b)	-
	EPIRℓ_1(2022)	(a),(e)	linear/superlinear, (e)
Generic $h(x)$	AIR (2021b)	(h)	linear, (h)
	STRN (2013)	(c)	-
	IReNA (Ours)	(a)	superlinear, (e) quadratic, (c)(i)

1.3 Notation and Preliminaries

Throughout let $\mathbb{R}^n$ denote the n-dimensional Euclidean space, equipped with the standard inner product $\langle \cdot, \cdot \rangle$ and its induced norm $\|\cdot\|$. Let $\mathbb{R}^n_+$ and $\mathbb{R}^n_{++}$ be the sets of nonnegative and strictly positive real vectors, respectively. Define $[n] := \{1, 2, 3, \ldots, n\}$. For any $x, y \in \mathbb{R}^n$, denote by x_i the ith component of x, by $x \circ y = (x_1 y_1, \cdots, x_n y_n)^T$ their component-wise product, and write $x \leqslant y$ to mean $x_i \leqslant y_i, \forall i \in [n]$. The index sets of nonzero and zero components of x are defined as $\mathcal{I}(x) = \{i \mid x_i \neq 0\}$ and $\mathcal{I}_0(x) = \{i \mid x_i = 0\}$, respectively. For any subset $\mathcal{S} \subseteq [n]$, the subvector of x containing the entries indexed by $\mathcal{S}$ is denoted by $x_{\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}|}$. The ℓ_p (quasi-)norm of a vector x is given by $\|x\|_p := (\sum_{i=1}^n |x_i|^p)^{1/p}$ for $p > 0$, and $\|\cdot\|$ particularly represents the ℓ_2 norm.

Consider a function $F : \mathbb{R}^n \to \mathbb{R}$. For any $x_{\mathcal{S}} \neq 0$, the partial gradient of F over $x_{\mathcal{S}}$ is denoted as $\nabla_{\mathcal{S}} F(x)$, i.e. , $\nabla_{\mathcal{S}} F(x) = \frac{\partial F}{\partial x_{\mathcal{S}}} \in \mathbb{R}^{|\mathcal{S}|}$. We avoid denoting $\nabla_i F(x) = [\nabla F(x)]_i$ for $i \in [n]$ since F may not be differentiable at x. For any vector $x \in \mathbb{R}^n$, the signum function $\mathrm{sgn} : \mathbb{R}^n \to \mathbb{R}^n$ yields a vector whose components are the signum of the individual components of x, $\mathrm{sgn}(x_i) = 1$ if $x_i > 0$, $\mathrm{sgn}(x_i) = -1$ if $x_i < 0$ and $\mathrm{sgn}(x_i) = 0$ if $x_i = 0$. The soft-thresholding operator is defined as $[\mathbb{S}_\omega(v)]_i := \mathrm{sgn}(v_i) \max\{|v_i| - \omega_i, 0\}$ for any $v \in \mathbb{R}^n$ and $\omega \in \mathbb{R}^n_{++}$. The projection operator $\mathbb{P}(y; x)$ projects y onto the subspace orthant containing x, defined as: $[\mathbb{P}(y; x)]_i := \mathrm{sgn}(x_i) \max\{0, \mathrm{sgn}(x_i) y_i\}$. Define the ball $\mathcal{B}(x, \rho) := \{y \in \mathbb{R}^n \mid \|y - x\| \leqslant \rho\}$ with $\rho > 0$. If the function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is

convex, then the subdifferential of f at $\bar{x}$ is given by $\partial f(\bar{x}) := \{z \mid f(\bar{x}) + \langle z, x - \bar{x}\rangle \leqslant f(x),\ \forall x \in \mathbb{R}^n\}$. For a set $S \subset \mathbb{R}^n$, the relative interior $\mathrm{rint}(S)$ is defined as $\mathrm{rint}(S) := \{x \in S : \text{there exists } \rho > 0 \text{ such that } \mathcal{B}(x, \rho) \cap \mathrm{aff}(S) \subseteq S\}$, where $\mathrm{aff}(S)$ is the affine hull of S.

We conclude this section by recalling the definition of a stationary point; see, e.g., (Wang et al. 2021b).

Definition 1.2 (Stationarity)**.** We say that a point $x^* \in \mathbb{R}^n$ is a first-order stationary point of the objective function F in (1) if

$$0 \in \nabla_i f(x^*) + \lambda \partial r(|x_i^*|),\ \forall i \in [n]. \tag{3}$$

2 Proposed IReNA

We present IReNA for solving (1) and derive useful properties for the subsequent convergence analysis. We begin by defining a smooth, locally Lipschitz continuous approximation of the original problem (1):

$$\min_{x \in \mathbb{R}^n} \quad F(x; \epsilon) := f(x) + \lambda \sum_{i=1}^{n} r(|x_i| + \epsilon_i), \tag{4}$$

where $\epsilon \in \mathbb{R}^n_{++}$ represents small positive perturbations. For notational clarity, note that setting $\epsilon = 0$ recovers the original objective $F(x; 0) = F(x), \forall x \in \mathbb{R}^n$. Hence $F(x; \epsilon)$ defines a family of smooth perturbations of $F(x)$, which satisfy $\lim_{\epsilon \to 0} F(x; \epsilon) = F(x), \forall x \in \mathbb{R}^n$. In the sequel, we will write $F(x, \epsilon)$ when the dependence on ϵ is essential, and simply write $F(x)$ otherwise; the intended meaning will be clear from context.

A point $x \in \mathbb{R}^n$ is first-order stationary for the perturbed problem (4) if the following conditions hold:

$$\begin{aligned} &|\nabla_i f(x)| \leqslant \omega_i, && i \in \mathcal{I}_0(x), \\ &\nabla_i f(x) + \omega_i \mathrm{sgn}(x_i) = 0, && i \in \mathcal{I}(x), \end{aligned} \tag{5}$$

where $\omega_i = \omega(x_i, \epsilon_i) = \lambda r'(|x_i| + \epsilon_i),\ \forall i \in [n]$. One readily checks that, (5) holds at x with $\epsilon_i = 0, \forall i \in \mathcal{I}(x)$ if and only if (3) holds at x. At iteration k, having current iterate x^k and perturbation ϵ^k, set $\omega_i^k = \omega(x_i^k, \epsilon_i^k), \forall i \in [n]$, and consider the following locally weighted ℓ_1 subproblem:

$$\min_{x \in \mathbb{R}^n} G_k(x) = f(x) + \sum_{i=1}^{n} \omega_i^k |x_i|. \tag{6}$$

We say $x \in \mathbb{R}^n$ is first-order stationary for problem (6) if

$$0 \in \nabla_i f(x) + \omega_i^k \partial |x_i|, \ \forall i \in [n]. \tag{7}$$

The following lemma establishes descent inequalities that link the perturbed objective and the locally weighted ℓ_1 surrogate, and shows that their notions of first-order stationarity coincide.

Lemma 2.1. *Consider* (4) *and* (6). *Let* $\{(x^k, \epsilon^k)\} \subset \mathbb{R}^n \times \mathbb{R}^n_{++}$ *be two successive iterates with* $\epsilon^{k+1} \leqslant \epsilon^k$ *component-wise. Then*

$$F(x^{k+1}; \epsilon^{k+1}) - F(x^k; \epsilon^k) \leqslant F(x^{k+1}; \epsilon^k) - F(x^k; \epsilon^k) \leqslant G_k(x^{k+1}) - G_k(x^k). \tag{8}$$

Moreover, the following statements are equivalent: (i) x^k *is a first-order stationary point of* G_k*; (ii)* x^k *is a first-order stationary point of* $F(\cdot; \epsilon^k)$*; (iii)* $x^k = \mathbb{S}_{\omega^k}(x^k - \nabla f(x^k))$.

2.1 Optimality Measure and Subspace Determination

Our algorithm adaptively constructs and solves subproblems on the zero and nonzero coordinates of a solution estimate x, using their respective optimality residuals. To that end, we decompose the support of x into $\mathcal{I}_+(x) := \{i \mid x_i > 0\}$ and $\mathcal{I}_-(x) := \{i \mid x_i < 0\}$. Motivated by (Chen et al. 2017), we define the optimality residual as $x - [\mathbb{S}_\omega(x - \nabla f(x))]$. This residual can be decomposed using Lemma 2.1 as follows:

$$\begin{aligned} [\Psi(x;\epsilon)]_i &:= \begin{cases} x_i - [\mathbb{S}_\omega(x - \nabla f(x))]_i, & \text{if } i \in \mathcal{I}_0(x) \\ 0, & \text{otherwise,} \end{cases} \\ [\Phi(x;\epsilon)]_i &:= \ x_i - [\mathbb{S}_\omega(x - \nabla f(x))]_i - [\Psi(x;\epsilon)]_i, \quad i \in [n]. \end{aligned} \tag{9}$$

Accordingly, $\Psi(x;\epsilon)$ and $\Phi(x;\epsilon)$ quantify the deviation from the perturbed optimality condition (5) on the zero and nonzero coordinates of x, respectively. For notational simplicity, define $\Phi^k = \Phi(x^k;\epsilon^k)$, $\Psi^k = \Psi(x^k;\epsilon^k)$, $\mathcal{I}^k = \mathcal{I}(x^k)$ and $\mathcal{I}_0^k = \mathcal{I}_0(x^k)$. The following proposition summarizes some useful properties of Φ^k and Ψ^k.

Proposition 2.2. *Let* $(x, \epsilon) \in \mathbb{R}^n \times \mathbb{R}^n_{++}$ *and* $\omega = \omega(x, \epsilon)$. *Then*

(i) $|[\Phi(x;\epsilon)]_i| \leqslant |\nabla_i F(x;\epsilon)|, \forall i \in \mathcal{I}(x)$.

(ii) For any $\beta > 0$ *define* $d(\beta) := \mathbb{S}_{\beta\omega}(x - \beta\nabla f(x)) - x$. *Then*

$$d_i(\beta) = -\beta[\Psi(x;\epsilon)]_i, \qquad i \in \mathcal{I}_0(x), \tag{10a}$$

$$|d_i(\beta)| \geqslant \min\{\beta, 1\}|[\Phi(x;\epsilon)]_i|, \quad i \in \mathcal{I}(x). \tag{10b}$$

(iii) $\|\Phi(x;\epsilon)\| = 0$ *and* $\|\Psi(x;\epsilon)\| = 0$ *if and only if* x *satisfies the first-order stationary condition* (5)*. Moreover, if additionally* $\epsilon_i = 0$ *for all* $i \in \mathcal{I}(x)$*, then* x *satisfies the first-order stationary condition* (3)*.*

One reason we use separate optimality residuals for zero and nonzero components is that, at each iteration, we solve a subproblem over a subspace defined by a well-chosen *subsets* of zero and nonzero coordinates. These two residuals are thus used as the "switching sign" for choosing which subsets to update. Concretely, at the kth iterate, we select a working index set $\mathcal{W}_k$ as follows:

$$\mathcal{W}_k \subseteq \{i : [\Psi^k]_i \neq 0\} \text{ and } \|[\Psi^k]_{\mathcal{W}_k}\| \geqslant \eta_\Psi \|\Psi^k\| \text{ if } \|\Psi^k\| \geqslant \eta\|\Phi^k\|, \tag{11a}$$

$$\mathcal{W}_k \subseteq \{i : [\Phi^k]_i \neq 0\} \text{ and } \|[\Phi^k]_{\mathcal{W}_k}\| \geqslant \eta_\Phi \|\Phi^k\| \text{ if } \|\Psi^k\| < \eta\|\Phi^k\|, \tag{11b}$$

where $\{\eta_\Psi, \eta_\Phi\} \in (0,1)$ and $\eta \in (0, +\infty)$ are prescribed parameters. Since at least one of $\|\Phi^k\|$ or $\|\Psi^k\|$ is nonzero unless x^k is already stationary, this guarantees that $\mathcal{W}_k \neq \varnothing$. In the case $\|\Psi^k\| \geqslant \eta\|\Phi^k\|$, the residuals associated with zeros have relatively greater impact, so we restrict $\mathcal{W}_k \subseteq \mathcal{I}_0^k$. Otherwise, the residuals associated with nonzeros dominate, and we choose $\mathcal{W}_k \subseteq \mathcal{I}^k$. This strategy ensures that each subproblem focuses computational effort on the coordinates most responsible for the current stationarity violation.

2.2 Subspace Proximal Weighted ℓ_1 Regularized Subproblems

To correctly identify the support of a desired stationary point and ensure global convergence, we solve an approximate model of the weighted ℓ_1 model G_k at current x^k, restricted on a selected index set $\mathcal{W} \subset [n]$:

$$\min_{x_\mathcal{W} \in \mathbb{R}^{|\mathcal{W}|}} \langle \nabla_\mathcal{W} f(x^k), x_\mathcal{W} \rangle + \tfrac{1}{2\beta}\|x_\mathcal{W} - x^k_\mathcal{W}\|^2 + \sum_{i \in \mathcal{W}} \omega_i^k |x_i|. \tag{12}$$

This problem admits the closed-form solution $[\mathbb{S}_{\beta\omega^k}(x^k - \beta\nabla f(x^k))]_\mathcal{W}$.

For any sufficiently small $\beta > 0$, subproblem (12) renders a descent direction for G_k and hence for the original objective F (the convergence analysis in the sequel quantifies this descent). Then we implement a line search with backtracking on β to ensure a sufficient decrease in G_k. This procedure is implemented in subroutine 1 and is thereafter referred to as the IST (iterative soft-thresholding) step. For readability, we omit the outer-iteration index k inside that subroutine.

The following lemma proves that the line search with backtracking in Algorithm 1 terminates in a finite number of steps.

Algorithm 1 IST step: $[y^j, \beta_j, \Delta_{\text{IST}}] := \text{IST}(x, \omega, \mathcal{W}; \xi_\beta, \gamma_\beta, \bar{\beta})$

Require: $\{x, \omega\} \in \mathbb{R}^n$, $\mathcal{W} \subseteq [n]$; $\{\bar{\beta}, \xi_\beta\} \in (0, +\infty)$ and $\gamma_\beta \in (0, 1)$.
1: Set $\beta_0 \leftarrow \bar{\beta}$.
2: **for** $j = 0, 1, 2, \ldots$ **do**
3: Set $y^j_{\mathcal{W}^c} \leftarrow x_{\mathcal{W}^c}$, $y^j_{\mathcal{W}} \leftarrow [\mathbb{S}_{\beta_j \omega}(x - \beta_j \nabla f(x))]_{\mathcal{W}}$.
4: **if** $G_k(y^j) \leqslant G_k(x) - \frac{\xi_\beta}{2}\|y^j - x\|^2$ **then**
5: **return** y^j, β_j and $\Delta_{\text{IST}} \leftarrow G_k(x) - G_k(y^j)$.
6: **end if**
7: Set $\beta_{j+1} \leftarrow \gamma_\beta \beta_j$.
8: **end for**

Lemma 2.3. *Let $(x, \epsilon) \in \mathbb{R}^n \times \mathbb{R}^n_{++}$, $\omega = \omega(x, \epsilon)$ and $\mathcal{W} \subseteq [n]$. Suppose Algorithm 1 is invoked with $(\hat{y}, \hat{\beta}, \hat{\Delta}_{IST}) = \text{IST}(x, \omega, \mathcal{W}; \xi_\beta, \gamma_\beta, \bar{\beta})$. Then Algorithm 1 terminates in a finite number of iterations and satisfies*

$$F(\hat{y}; \epsilon) - F(x; \epsilon) \leqslant -\frac{\xi_\beta}{2}\|\hat{y} - x\|^2, \tag{13}$$

along with the step size bound $\min\{\bar{\beta}, \frac{\gamma_\beta}{L_1(x) + \xi_\beta}\} \leqslant \hat{\beta} \leqslant \bar{\beta}$, where $L_1(x) > 0$ is the local Lipschitz constant of ∇f in a neighborhood of x containing $\hat{y}$.

2.3 Subspace Quadratic Subproblems

To accelerate local convergence, we restrict our search to the chosen index set $\mathcal{W}_k \subseteq \mathcal{I}^k$ and solve the corresponding quadratic subproblem. In particular, we compute an approximate descent direction

$$\bar{d}^k \approx \underset{d \in \mathbb{R}^{|\mathcal{W}_k|}}{\text{argmin}} \; m_k(d) := \langle g^k, d \rangle + \frac{1}{2}\langle d, H^k d \rangle,$$

where $g^k = \nabla_{\mathcal{W}_k} F(x^k; \epsilon^k)$ denotes the subspace gradient of $F(x^k; \epsilon^k)$, and $H^k = \nabla^2_{\mathcal{W}_k \mathcal{W}_k} F(x^k; \epsilon^k) + \zeta_k I \succ 0$ denotes the modified subspace Hessian with a regularization parameter $\zeta_k > 0$. Since the local quadratic model $m_k(d)$ is defined within the subspace $\mathbb{R}^{|\mathcal{W}_k|}$, its dimension may be small. An inexact solution estimate $\bar{d}^k$ is permitted if it satisfies

$$\langle g^k, \bar{d}^k \rangle \leqslant \langle g^k, d^k_R \rangle \;\text{ and }\; m_k(\bar{d}^k) \leqslant m_k(0). \tag{14}$$

where $d^k_R := -\frac{\|g^k\|^2}{\langle g^k, H^k g^k \rangle} g^k$ is a reference direction. We mention that the subproblem (2.3) can be efficiently solved using many existing efficient quadratic programming solvers, e.g., Conjugate Gradient (CG) method (Hager and Zhang 2006). To

embed the subspace step $\bar{d}^k$ into the full space, we define the search direction $\tilde{d}^k \in \mathbb{R}^n$ by $\tilde{d}^k_{\mathcal{W}_k} = \bar{d}^k$ and $\tilde{d}^k_{\mathcal{W}_k^c} = 0$.

The following Lemma collects several useful bounds for the subspace QP (2.3). Its proof is identical to that of (Chen et al. 2017, Lemma 3).

Lemma 2.4. *Consider the subspace QP* (2.3) *with* $H^k \succ 0$. *Suppose the approximate solution* $\bar{d}^k$ *satisfies the condition* (14). *Then, the following holds:*

$$\langle g^k, \bar{d}^k \rangle \leqslant \langle g^k, d_R^k \rangle < 0, \tag{15}$$

$$|\langle g^k, \bar{d}^k \rangle| \geqslant |\langle g^k, d_R^k \rangle| = \frac{\|g^k\|^2}{\langle g^k, H^k g^k \rangle} \|g^k\|^2 \geqslant \frac{\|g^k\|^2}{\lambda_{\max}(H^k)}, \tag{16}$$

$$\frac{\|g^k\|}{\lambda_{\max}(H^k)} \leqslant \|\bar{d}^k\| \leqslant \frac{2\|g^k\|}{\lambda_{\min}(H^k)}. \tag{17}$$

2.4 Projected Line Search

Once a descent direction d is obtained by solving the subspace QP (2.3), the projected line search subroutine (see Algorithm 2) is invoked to determine a stepsize α that both preserves the current sign pattern of x and ensures a sufficient decrease in $F(x;\epsilon)$. For clarity, we omit the iteration counter superscript k in the outer loop.

Given current iterate x, perturbation ϵ and QP direction d, Algorithm 2 implements a line search with backtracking along the direction d to determine a stepsize α_j such that the projected update $\mathbb{P}(x + \alpha_j d; x)$ leads to a sufficient decrease in $F(\cdot;\epsilon)$. When the condition $\mathrm{sgn}(y^j) = \mathrm{sgn}(x)$ holds for the first time, it indicates that for any stepsize $\alpha \in (0, \alpha_j)$, $\mathrm{sgn}(\mathbb{P}(x + \alpha d; x)) = \mathrm{sgn}(x)$ holds. In this case, PLS tracks the largest possible $\alpha_B > 0$ for which $\mathrm{sgn}(x + \alpha d) = \mathrm{sgn}(x)$. This yields a trial point with the minimal support change compared to x.

Since y^j is guaranteed to encounter the same sign as x within a finite number of trials, PLS reverts to a standard line search with backtracking . Therefore, it is generally known that the overall subroutine terminates in a finite number of iterations.

Lemma 2.5. *Let* $(x, \epsilon, d) \in \mathbb{R}^n \times \mathbb{R}^n_{++} \times \mathbb{R}^n$ *such that* $\langle \nabla F(x;\epsilon), d \rangle < 0$. *Suppose Algorithm 2 is invoked with* $(\tilde{y}, \tilde{\alpha}, \tilde{\Delta}_{QP}) = \mathrm{PLS}(x, \epsilon, d; \xi_\alpha, \gamma_\alpha)$. *Then Algorithm 2 terminates finitely satisfying*

$$F(\tilde{y};\epsilon) - F(x;\epsilon) \leqslant -\frac{\xi_\alpha}{2}\|\tilde{y} - x\|^2. \tag{18}$$

Moreover, if $sgn(\tilde{y}) \neq sgn(x)$, *then*

$$1 \geqslant \tilde{\alpha} \geqslant \min\{\alpha_B, 1\} \quad \textit{and} \quad \mathcal{I}(\tilde{y}) \subset \mathcal{I}(x). \tag{19}$$

Algorithm 2 Projected line search: $[y^j, \alpha_j, \Delta_{\text{QP}}] := \text{PLS}(x, \epsilon, d; \xi_\alpha, \gamma_\alpha)$

Require: $\{x, d, \epsilon\} \in \mathbb{R}^n$; $\xi_\alpha \in (0, +\infty)$ and $\gamma_\alpha \in (0, 1)$.

1: Set $\alpha_0 \leftarrow 1$ and Flag = `True`.
2: **for** $j = 0, 1, 2, \ldots$ **do**
3: Set $y^j \leftarrow \mathbb{P}(x + \alpha_j d; x)$.
4: **if** $\text{sgn}(y^j) = \text{sgn}(x)$ and Flag **then**
5: Set $\alpha_B \leftarrow \arg\sup\{\alpha > 0 : \text{sgn}(x + \alpha d) = \text{sgn}(x)\}$ and Flag = `False`. ▷ "sign-lock".
6: Set $\alpha_j \leftarrow \min\{1, \alpha_B\}$ and $y^j \leftarrow \mathbb{P}(x + \alpha_j d; x)$.
7: **end if**
8: **if** $F(y^j; \epsilon) \leqslant F(x; \epsilon) - \frac{\xi_\alpha}{2}\|y^j - x\|^2$ **then**
9: **return** y^j, α_j and $\Delta_{\text{QP}} \leftarrow F(x; \epsilon) - F(y^j; \epsilon)$.
10: **end if**
11: Set $\alpha_{j+1} \leftarrow \gamma_\alpha \alpha_j$.
12: **end for**

Otherwise, $F(x; \epsilon)$ is $L_2(x; \epsilon)$-Lipschitz differentiable on $[x, x + \alpha_B d)$, and

$$\min\{1, \alpha_B\} \geqslant \tilde{\alpha} \geqslant \min\left\{1, \frac{-2\gamma_\alpha \langle \nabla F(x; \epsilon), d\rangle}{(L_2(x; \epsilon) + \xi_\alpha)\|d\|^2}\right\} \quad \textit{and} \quad \mathcal{I}(\tilde{y}) = \mathcal{I}(x). \tag{20}$$

2.5 Main Algorithm

With the two building-block subproblems now in place, we summarize the complete IReNA method in Algorithm 3. At each outer iteration k, IReNA proceeds by first using an IST step, restricted to the index set $\mathcal{W}_k \subset [n]$, to detect the optimal support; only when that support appears to have stabilized does it invoke a second-order QP on the corresponding reduced subspace to accelerate local convergence.

By construction, the selected index set $\mathcal{W}_k$ satisfies exactly one of two alternatives: either $\mathcal{W}_k \subseteq \mathcal{I}_0^k$, as in (11a), or $\mathcal{W}_k \subseteq \mathcal{I}^k$, as in (11b). In the former case, the IST update yields $\hat{x}^k = x^k + d(\hat{\beta}_k) = d(\hat{\beta}_k) \neq 0$ by (10a), which necessarily flips at least one sign and thus enlarges the active set, i.e., $\mathcal{I}^k \subset \mathcal{I}(\hat{x}^k)$. Consequently, the stabilization test in line 8 fails and the QP subproblem is skipped. Conversely, if $\mathcal{W}_k \subseteq \mathcal{I}^k$ then $\mathcal{I}(\hat{x}^k) \subseteq \mathcal{I}^k$, ensuring that, when invoked, the QP subproblem indeed operates only over a subset of currently nonzero components.

To guard against unproductive QP steps, line 11 compares the predicted objective decrease $\tilde{\Delta}_{\text{QP}}$ from the QP subproblem with the decrease $\hat{\Delta}_{\text{IST}}$ achieved by the IST update. If $\tilde{\Delta}_{\text{QP}} < \nu \hat{\Delta}_{\text{IST}}$ for a given $\nu > 0$, the algorithm discards the QP

solution in favor of the IST iterate. This safeguard ensures that the QP correction is accepted only when it yields a sufficiently large reduction in the objective and the support remains locally stable.

Algorithm 3 Proposed IReNA for solving (1)

Require: $(x^0, \epsilon^0) \in \mathbb{R}^n \times \mathbb{R}^n_{++}$, $\{\eta_\Phi, \eta_\Psi\} \in (0, 1]$, $\{\gamma_\epsilon, \gamma_\beta, \gamma_\alpha\} \in (0, 1)$, $\{\tau, \eta, \xi_\beta, \xi_\alpha, \bar{\beta}, \zeta_k, \nu\} \in (0, \infty)$.

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: (**Optimality measure**) Compute Ψ^k, Φ^k.
3: **if** $\max\{\|\Psi^k\|, \|\Phi^k\|\} \leqslant \tau$ and $\epsilon_i^k \leqslant \tau, i \in \mathcal{I}^k$ **then**
4: **return** the (approximate) solution x^k.
5: **end if**
6: (**Subspace determination**) Choose $\mathcal{W}_k$ based on Ψ^k, Φ^k by (11).
7: (**IST step 1**) Compute $(\hat{x}^k, \hat{\beta}_k, \hat{\Delta}^k_{\mathrm{IST}}) \leftarrow \mathrm{IST}(x^k, \omega^k, \mathcal{W}_k; \xi_\beta, \gamma_\beta, \bar{\beta})$.
8: **if** $\mathrm{sgn}(\hat{x}^k) = \mathrm{sgn}(x^k)$ **then**
9: (**QP subproblem**) Find an approximate solution $\tilde{d}^k$ to (2.3) satisfying (14).
10: (**Projected line search 2**) Compute $(\tilde{x}^k, \tilde{\alpha}_k, \tilde{\Delta}^k_{\mathrm{QP}}) \leftarrow \mathrm{PLS}(x^k, \epsilon^k, \tilde{d}^k; \xi_\alpha, \gamma_\alpha)$.
11: **if** $\mathrm{sgn}(\tilde{x}^k) \neq \mathrm{sgn}(x^k)$ and $\tilde{\Delta}^k_{\mathrm{QP}} < \nu \hat{\Delta}^k_{\mathrm{IST}}$ **then**
12: Set $x^{k+1} \leftarrow \hat{x}^k$.
13: **else**
14: Set $x^{k+1} \leftarrow \tilde{x}^k$.
15: **end if**
16: **else**
17: Set $x^{k+1} \leftarrow \hat{x}^k$.
18: **end if**
19: Update $\epsilon_i^{k+1} \in (0, \gamma_\epsilon \epsilon_i^k)$, $i \in \mathcal{I}_0^{k+1}$ and $\epsilon_i^{k+1} = \epsilon_i^k$, $i \in \mathcal{I}_0^{k+1}$.
20: Update $\omega_i^{k+1} = \lambda r'(|x_i^{k+1}| + \epsilon_i^{k+1})$, $i \in [n]$.
21: **end for**

Proposition 2.2 implies that IReNA converges to an optimal solution of (1) provided that $\max\{\|\Psi^k\|, \|\Phi^k\|\} \to 0$ and $\epsilon_i^k \to 0$ for all $i \in \mathcal{I}^k$. This condition motivates our termination criterion (see lines 3–4). On the other hand, the update of the smoothing parameter ϵ should be carefully designed to ensure global convergence and practical performance. Once the true support $\mathcal{I}(x^*)$ is identified, we drive each $\epsilon_i^k \to 0$ for $i \in \mathcal{I}(x^*)$ while keeping ϵ_i^k fixed for $i \in \mathcal{I}_0(x^*)$. Therefore, all zero components and their associated smoothing terms are effectively removed

from $F(x;\epsilon)$, and the remaining problem reduces to a smooth optimization—a crucial aspect of the convergence analysis. Concretely, in line 19 we decrease ϵ_i^k only for indices in the current support $\mathcal{I}^k$. As shown in (Wang et al. 2021a), this strategy ensures that, after sufficiently large iterations, the algorithm ceases to update ϵ_i for inactive indices while continuously shrinking ϵ_i on the true support.

Remark 2.6. The criterion for switching between first-order and second-order algorithm plays a pivotal role in hybrid optimization methods. Early algorithms—such as those in (Shi et al. 2010)—employed simple heuristics (e.g. checking whether successive iterates are sufficiently close). More recent schemes, including PCSNP (Zhou et al. 2023) and HpgSRN (Wu et al. 2023), invoke a second-order update whenever the signs of two consecutive iterates coincide. While aggressive, it can lead to premature selection of costly Newton updates and yield limited net computational benefit. Our approach mitigates this issue by incorporating a rollback safeguard: if a candidate second-order update fails to achieve a sufficient objective reduction, it is rejected and the algorithm reverts to a safeguarded first-order iterate. This safeguard is especially important when Hessian approximations are inexact, as it prevents the propagation of misleading search directions.

Remark 2.7. Our framework extends directly to any penalty r that is $\mathcal{C}^1$-smooth on $\mathbb{R}_+$ yet admits only finitely many nondifferentiable "kinks" (as in Smoothly Clipped Absolute Deviation (SCAD) (Fan and Li 2001), with two breakpoints, or Minimax Concave Penalty (MCP) (Zhang 2010), with one). On $\mathbb{R}_{++}$, r' is piecewise affine and continuous, hence globally Lipschitz; by *Rademacher's* theorem, r' is therefore differentiable almost everywhere on $\mathbb{R}_{++}$. Consequently, in the the IST step only gradient information is required, so no modification is needed. In the QP subproblem, the penalty contributes a diagonal term $\mathrm{Diag}(r''(|x_i^k| + \epsilon_i^k))$; to ensure each evaluation avoids the finitely many kink points of r' (where r'' fails to exist), we introduce a small perturbation $\epsilon_i^k > 0$. Since the set of all such breakpoints has Lebesgue measure zero, one can always choose ϵ_i^k arbitrarily small so that $|x_i^k| + \epsilon_i^k$ lies in a $\mathcal{C}^2$-smooth region of r. Hence every $r''(|x_i^k| + \epsilon_i^k)$ is well-defined. All of our global convergence arguments remain intact under these mild perturbations. For local quadratic convergence, we merely assume the eventual limit point x^* does not coincide exactly with one of the breakpoint—an event of probability zero under any small random perturbation.

3 Convergence Analysis

We have shown that each quadratic subproblem is well-defined (Lemma 2.4), and that the line search routines in the IST step 1 (Lemma 2.3) and in the projected line search 2 (Lemma 2.5) both terminate in finitely many iterations. We next

show that the perturbed objective values $\{F(x^k;\epsilon^k)\}$ converge and that all iterates remain in the level set $\mathrm{Lev}_F := \{x \in \mathbb{R}^n \mid F(x) \leqslant F(x^0;\epsilon^0)\}$.

The following assumption is assumed in the subsequent convergence analysis.

Assumption 3.1. *The tolerance is set as $\tau = 0$ and the level set Lev_F is contained in a closed ball $\{x \in \mathbb{R}^n \mid \|x\| \leqslant R\}$ for some $R > 0$, so that F is bounded below and its smooth term f is continuously differentiable with Lipschitz constant $L_f > 0$ over Lev_F.*

Theorem 3.2. *Let Assumptions 1.1 and 3.1 hold. Then Algorithm 3 generates an infinite sequence $\{(x^k,\epsilon^k)\}$ for which both $\{\epsilon_i^k\}, i \in [n]$ and $\{F(x^k;\epsilon^k)\}$ are monotonically nonincreasing and converge, and $\{x^k\} \subset Lev_F$.*

Theorem 3.2 implies that the local Lipschitz constant of f in Lemma 2.3 satisfies $L_1(x^k) \leqslant L_f$, so that

$$\underline{\beta} \leqslant \hat{\beta}_k \leqslant \bar{\beta} \tag{21}$$

with $\underline{\beta} := \min\{\bar{\beta}, \frac{\gamma_\beta}{L_f+\xi_\beta}\}$. Next, we show similar results for $\lambda_{\min}(H^k)$, $\lambda_{\max}(H^k)$ in Lemma 2.4 and $L_2(x^k;\epsilon^k)$ in Lemma 2.5.

Lemma 3.3. *Let Assumptions 1.1 and 3.1 hold and let $\{x^k,\epsilon^k\}$ be the sequence generated by Algorithm 3. Consider the Hessian H^k involved in subproblem (2.3). Then there exist constants $0 < \lambda_{\min} < \lambda_{\max} < +\infty$ such that, for each $k \in \mathbb{N}$, $\lambda_{\min} I \preceq H^k \preceq \lambda_{\max} I$. Moreover, there exists $L_F > 0$ such that $L_2(x^k;\epsilon^k) \leqslant L_F, \forall k \in \mathbb{N}$.*

As a consequence of Lemma 3.3, the step-size lower bound in (20) can be sharpened to

$$\underline{\alpha} \leqslant \tilde{\alpha}_k \leqslant 1 \tag{22}$$

with $\underline{\alpha} := \frac{\gamma_\alpha \lambda_{\min}^2}{2\lambda_{\max} L_F + 2\xi_\alpha \lambda_{\max}}$. Indeed, by (16) and (17), one obtains

$$\tilde{\alpha}_k \geqslant \frac{-2\gamma_\alpha \langle \nabla_{\mathcal{W}} F(x^k;\epsilon^k), d^k\rangle}{(L_2(x^k;\epsilon^k)+\xi_\alpha)\|d^k\|^2} \geqslant 2\frac{\gamma_\alpha \frac{\|g^k\|^2}{\lambda_{\max}}}{(L_2(x^k;\epsilon^k)+\xi_\alpha)\frac{4\|g^k\|^2}{\lambda_{\min}^2}} \geqslant \frac{\gamma_\alpha \lambda_{\min}^2}{2\lambda_{\max} L_F + 2\xi_\alpha \lambda_{\max}},$$

where the third inequality uses Lemma 3.3.

3.1 Global Convergence

The global convergence of IReNA is established in this subsection. For ease of presentation, we first define the following sets of iterations for our analysis.

$$\mathcal{S}_{\mathrm{IST}} := \{k \in \mathbb{N} : x^{k+1} = \hat{x}^{k+1} \text{ by line 12 or 17 at the } k\text{th iteration}\},$$
$$\mathcal{S}_{\mathrm{QP}} := \{k \in \mathbb{N} : x^{k+1} = \tilde{x}^{k+1} \text{ by line 14 at the } k\text{th iteration}\}.$$

The set $\mathcal{S}_{\mathrm{QP}}$ is further spitted into two subsets:

$$\mathcal{S}_{\mathrm{QP}}^{\mathrm{N}} := \{k \in \mathcal{S}_{\mathrm{QP}} : \mathrm{sgn}(x^{k+1}) \neq \mathrm{sgn}(x^k)\}, \quad \mathcal{S}_{\mathrm{QP}}^{\mathrm{Y}} := \mathcal{S}_{\mathrm{QP}} \backslash \mathcal{S}_{\mathrm{QP}}^{\mathrm{N}}.$$

We are now ready to prove the global convergence of the proposed algorithm.

Theorem 3.4 (Global subsequential convergence)**.** *Let Assumptions 1.1 and 3.1 hold and let $\{(x^k, \epsilon^k)\}$ be the sequence generated by Algorithm 3. Then*

$$\lim_{k\to+\infty} \|x^{k+1}-x^k\| = 0, \quad \lim_{k\to+\infty} \|\epsilon^{k+1}-\epsilon^k\| = 0 \quad \textit{and} \quad \lim_{k\to+\infty} \|\Phi^k\| = 0, \quad \lim_{k\to+\infty} \|\Psi^k\| = 0.$$

Moreover, every cluster point x^* of $\{x^k\}$ satisfies the first-order stationary condition (3)*.*

We next show that, after finitely many iterations, the iterates generated by IReNA exhibit locally sign-stabilization: there exists $K \in \mathbb{N}$ such that for all $k \geqslant K$, $\mathrm{sgn}(x^{k+1}) = \mathrm{sgn}(x^k)$. Achieving sign stability of the iterates requires that the algorithm correctly identifies the nonzero support in a finite number of steps. In the nonsmooth optimization literature, this property is typically known as active manifold identification, and is usually established under a nondegeneracy condition at the limit point, namely $0 \in \mathrm{rint}\, \partial F(x)$ (Liang et al. 2017, Lewis and Wylie 2021). We also impose this assumption throughout the remainder analysis, stated equivalently as follows:

Assumption 3.5. *Let x^* be any first-order stationary point of* (1)*. we assume that $\nabla_i f(x^*) \in (-\lambda r'(0^+), \lambda r'(0^+)),\ \forall i \in \mathcal{I}_0(x^*)$.*

This assumption naturally holds when $r'(0^+) = +\infty$. With this, we are now ready to establish the local support stability of the iterates.

Proposition 3.6 (Locally stable sign)**.** *Let $\{x^k\}$ be the sequence generated by Algorithm 3. The following statements hold.*

(i) There exists $\delta > 0$ such that $|x_i^k| > \delta, i \in \mathcal{I}^k$ holds for all $k \in \mathbb{N}$. Consequently, $\omega_i^k < \hat{\omega} := \lambda r'(\delta), i \in \mathcal{I}^k$ holds for all $k \in \mathbb{N}$.

(ii) There exist index sets $\mathcal{I}_0^*$ and $\mathcal{I}^*$ such that $\mathcal{I}_0^k \equiv \mathcal{I}_0^*$ and $\mathcal{I}^k \equiv \mathcal{I}^*$ for sufficiently large $k \in \mathbb{N}$.

(iii) $k \in \mathcal{S}_{QP}^Y$ and $sgn(x^{k+1}) = sgn(x^k)$ for sufficiently large $k \in \mathbb{N}$.

Proposition 3.6(ii) implies that there exists an index $\check{k} \in \mathbb{N}$ such that for all $k \geqslant \check{k}$, (x^k, ϵ^k) always lies in the reduced subspace $\mathcal{M}(x^*, \epsilon^*) := \{(x, \epsilon) \mid x_{\mathcal{I}_0^*} =$

$0, \epsilon_{\mathcal{I}_0^*} \equiv \epsilon_{\mathcal{I}_0^*}^{\check{k}}\}$ and in particular the iterates x^k satisfy $x^k \in \overline{\mathcal{M}}(x^*) := \{x \mid x_{\mathcal{I}_0^*} = 0\}$. As a direct consequence, Proposition 3.6 also guarantees the local equivalence between Φ and the subspace gradient of $F(x;\epsilon)$, which we now state formally in the following corollary.

Corollary 3.7. *Let $\{(x^k, \epsilon^k)\}$ be the sequence generated by Algorithm 3. Then, for sufficiently large $k \in \mathbb{N}$ and each $i \in \mathcal{I}^*$, $[\Phi(x^k;\epsilon^k)]_i = \nabla_i F(x^k;\epsilon^k)$. Consequently, $\lim_{k\to+\infty} \nabla_{\mathcal{I}^*} F(x^k;\epsilon^k) = 0$.*

3.2 Convergence Rate Under the KL Property

In this subsection, we establish the local convergence properties by using the well-known KL property. For clarity, we adopt the definitions of the KL exponent from (Li and Pong 2018, Definitions 2.3).

Definition 3.8 (KL exponent)**.** For a proper closed function f satisfying the KL property at $\bar{x} \in \operatorname{dom} \partial f$, if the corresponding function ϕ can be chosen as $\phi(s) = cs^{1-\theta}$ for some $c > 0$ and $\theta \in [0, 1)$, i.e., there exist $c, \rho > 0$ and $\upsilon \in (0, \infty]$ such that $\operatorname{dist}(0, \partial f(x)) \geqslant c(f(x) - f(\bar{x}))^\theta$ whenever $x \in \mathcal{B}(\bar{x}, \rho)$ and $f(\bar{x}) < f(x) < f(\bar{x}) + \upsilon$, then we say that f has the KL property at $\bar{x}$ with an exponent of θ. If f is a KL function and has the same exponent θ at any $\bar{x} \in \operatorname{dom} \partial f$, then we say that f is a KL function with an exponent of θ.

We now restrict our discussion to the reduced subspace $\mathbb{R}^{|\mathcal{I}^*|}$. To simplify notation, write $\epsilon = \varepsilon \circ \varepsilon$ with the vector $\varepsilon \geqslant 0$ regarded as the new variable. As noted in (Luo et al. 1996, Page 63, Section 2.1), determining or estimating the KL exponent of a given function can be extremely challenging. A particularly relevant and useful result of (Zeng et al. 2016), with a proof in (Wang et al. 2022, Theorem 7), shows that any twice continuously differentiable function f has KL exponent $\theta = 1/2$ at a non-degenerate critical point—that is, at any point where $\nabla f = 0$ and the Hessian $\nabla^2 f$ is nonsingular. We therefore invoke this theorem to obtain the following result.

Proposition 3.9. *Consider the following statements.*

(i) The KL exponent of $F(x, \varepsilon)$ restricted on $\mathcal{M}(x^*, \varepsilon^*)$ at $(x^*_{\mathcal{I}^*}, 0)$ is θ.

(ii) The KL exponent of $F(x, \varepsilon)$ at $(x^*, 0)$ is θ.

(iii) The KL exponent of $F(x)$ restricted on $\overline{\mathcal{M}}(x^*)$ at $x^*_{\mathcal{I}^*}$ is θ.

(iv) The KL exponent of $F(x)$ at x^* is θ.

Then we have (i) $\iff$ (ii), (iii) $\iff$ (iv), and (i) $\Longrightarrow$ (iii). Moreover, we have $\theta \in (0,1)$ and $\theta = 1/2$ if $\nabla^2_{\mathcal{I}^*\mathcal{I}^*} F(x^*)$ is nonsingular in (iii).

The convergence rate analysis of IRℓ_1-type algorithms for nonconvex ℓ_p-regularized problem (1) under the KL property has been studied in (Wang et al. 2023, 2022). For a more comprehensive treatment of convergence rate estimates across a wider class of descent methods, see (Li and Pong 2018) as well as (Attouch et al. 2013). We proceed to demonstrate that IReNA satisfies both the *sufficient decrease condition* and the *relative error condition* as outlined in (Attouch et al. 2013, Wen et al. 2018).

Lemma 3.10. *Let $\{(x^k, \epsilon^k)\}$ be the sequence generated by Algorithm 3. The following holds.*

(i) There exists $C_1 > 0$ such that $F(x^{k+1}, \varepsilon^{k+1}) + C_1\|x^{k+1} - x^k\|^2 \leqslant F(x^k, \varepsilon^k)$.

(ii) There exists $C_2 > 0$ such that $\|\nabla F(x^{k+1}, \varepsilon^{k+1})\| \leqslant C_2(\|x^{k+1} - x^k\| + \|\varepsilon^k\|_1 - \|\varepsilon^{k+1}\|_1)$.

With the results established above, we now present the convergence properties of the iterates under KL property in the following theorem.

Theorem 3.11. *Let $\{(x^k, \epsilon^k)\}$ be the sequence generated by Algorithm 3. Suppose that $F(x, \varepsilon)$ restricted to $\mathcal{M}(x^*, \varepsilon^*)$ is a KL function at all stationary points $(x^*, 0)$. Then it holds that*

$$\sum_{k=0}^{+\infty} \|x^{k+1} - x^k\| < +\infty. \tag{23}$$

Therefore the entire sequence $\{x^k\}$ converges to a stationary point x^*. In addition, if F is a local KL function of some exponent $\theta \in (0,1)$ at all stationary points. Then for sufficiently large k, it holds that

(i) If $\theta \in (0, \frac{1}{2}]$, then there exist $\vartheta \in (0,1)$ and $C_3 > 0$ such that $\|x^k - x^*\| < C_3\vartheta^k$.

(ii) If $\theta \in (\frac{1}{2}, 1)$, then there exists $C_4 > 0$ such that $\|x^k - x^*\| < C_4 k^{-\frac{1-\theta}{2\theta-1}}$.

3.3 Local Convergence Analysis with Exact QP Solution

In this subsection, we establish the local convergence properties of Algorithm 3. By Proposition 3.6, after a finite number of iterations, the iterates generated by IReNA are obtained by solving only a reduced-space QP subproblem, combined with a backtracking line search. Let χ^∞ be the set of all stationary points of $\{x^k\}$ generated by IReNA. We now impose the following assumptions.

Assumption 3.12. *Let $\{x^k\}$ be the sequence generated by Algorithm 3 with $\{x^k\} \to x^* \in \chi^\infty$. Suppose there exists $M > 0$ such that the reduced-space Hessian of $F(x)$ at x^* satisfies $\|\nabla^2_{\mathcal{I}^*\mathcal{I}^*} F(x^*)^{-1}\| \leqslant M$. For all sufficiently large $k \in \mathbb{N}$, we assume*

(i) The exact Hessian $H^k = \nabla^2_{\mathcal{I}^k\mathcal{I}^k} F(x^k; \epsilon^k)$ is used in $m_k(d)$ and $\mathcal{W}_k \equiv \mathcal{I}^k = \mathcal{I}(x^*)$.

(ii) The reduced-space QP subproblem (2.3) is solved exactly i.e., $\tilde{d}^k = (H^k)^{-1} g^k$.

(iii) For all sufficiently large k, the unit stepsize $\alpha_k \equiv 1$ is accepted.

As shown in Proposition 3.9 , the non-singularity of $\nabla^2_{\mathcal{I}^*\mathcal{I}^*} F(x)$ implies that the KL exponent of F is $1/2$ at x^*. Therefore, a linear convergence rate is achieved by Theorem 3.11. In the following, we show that under Assumption 3.12, a quadratic convergence rate can be attained when $\epsilon^k_{\mathcal{I}^*} \to 0$ at a quadratic rate.

Theorem 3.13. *Let Assumption 3.12 hold, and let $\{(x^k, \epsilon^k)\}$ be the sequence generated by Algorithm 3, with $\{x^k\} \to x^* \in \chi^\infty$. Then there exist $\check{k} \in \mathbb{N}$ and $\rho > 0$ such that for all $x^k \in \mathcal{B}(x^*, \rho)$, the following holds:*

$$\|x^{k+1} - x^*\| \leqslant \frac{3ML_H}{2}\|x^k - x^*\|^2 + \mathcal{O}(\|\epsilon^k\|), \quad \forall k \geqslant \check{k},$$

where $\nabla^2_{\mathcal{I}^*\mathcal{I}^*} F(x)$ is locally Lipschitz continuous with constant $L_H > 0$ on $\mathcal{B}(x^*, \rho)$.

3.4 Local Second-order Complexity Grantees with Trust-region Subproblem

The QP subproblem solved in line 9 of Algorithm 3 seeks an inexact Newton direction restricted to a reduced space using subspace regularized Newton method. Proposition 3.6 indicate that the QP subproblem will be always triggered after some iterations. Therefore, we can consider replacing (2.3) with the following tailored regularized trust-region subproblem:

$$\tilde{d}^k_{\mathcal{W}_k} = \underset{d \in \mathbb{R}^{|\mathcal{W}_k|}}{\arg\min} \quad \langle g^k, d\rangle + \frac{1}{2}\langle d, H^k d\rangle + \frac{1}{2}\zeta_H \|d\|^2, \quad \text{s.t. } \|d\| \leqslant \rho_k, \tag{24}$$

where $\zeta_H > 0$ and $\rho_k > 0$ represents the trust-region radius and H^k can be indefinite. The trust-region Newton method presented in (Curtis et al. 2021, Algorithm 2.1) can be directly applied to solve (24) with a slight modification. Specifically, we solve the regularized trust-region subproblem (24) and set $\tilde{d}^k_{[\mathcal{W}_k]^c} = 0$ to obtain an intermediate direction $\tilde{d}^k$. We then compute the trial step $d^k = \mathbb{P}(x^k + \tilde{d}^k; x^k) - x^k$. If d^k leads to a non-increasing objective, it is accepted; otherwise, the radius of

the trust region is reduced. In addition, it can be straightforwardly verified that Theorem 3.4 also holds. Therefore, a similar local second-order complexity can be derived, which is stated as follows.

Theorem 3.14. *(Curtis et al. 2021, Theorem 2.6) Suppose the termination condition line 3 in Algorithm 3 is set to* $\|\nabla F(x^k;\epsilon^k)\| \leqslant \tau$ *and* $\lambda_{\min}(\nabla^2 F(x^k;\epsilon^k)) \geqslant -\sqrt{\tau}$. *Let* $\check{k} \in \mathbb{N}$ *such that* $\mathcal{I}(x^k) = \mathcal{I}^*$ *for all* $k > \check{k}$. *If the trust-region subproblem is implemented, then it holds that* $|\mathcal{K}| = \mathcal{O}(\tau^{-3/2}\log_{1/\varpi}(\tau^{-1/2}))$ *for some* $\varpi \in (0,1)$, *where* $\mathcal{K} := \{k|\ k > \check{k}\}$.

4 Numerical Experiments

In this section, we deliver a set of numerical experiments on both synthetic and real-world datasets across diverse model prediction problems to demonstrate the effectiveness of the proposed IReNA. We begin with the nonconvex ℓ_p-regularized logistic regression problem:

$$\min_{x\in\mathbb{R}^n} \quad \sum_{i=1}^{m} \log(1 + e^{-a_i x^T b_i}) + \lambda\|x\|_p^p, \tag{25}$$

where $\lambda > 0$, $a_i \in \{-1, 1\}$ are class labels, and $b_i \in \mathbb{R}^n$ are feature vectors for $i \in [m]$. To further demonstrate IReNA's flexibility, we also incorporate additional nonconvex regularizers in place of the ℓ_p^p term.

We compare IReNA with several state-of-the-art algorithms for solving (25), including HpgSRN (Wu et al. 2023)[1], PCSNP (Zhou et al. 2023)[2] and EPIRℓ_1 (Wang et al. 2022). All implementations were conducted in MATLAB, except for the feedforward neural network task in Section 4.4.1, which was implemented in Python. The code was executed on a PC with an Intel i9-13900K (3.00 GHz) CPU, 64 GB of RAM and an NVIDIA RTX 2080 GPU.

4.1 Experimental Setup

In our experiments, we set $\lambda = 1$ and obtain labels a and features b from different datasets. The synthetic dataset is generated following the methods in (Chen et al. 2017, Keskar et al. 2016). Specifically, for $i \in [m]$, labels a_i are drawn from a Bernoulli distribution and feature vectors b_i are sampled from a standard Gaussian distribution. The real-world datasets— *w8a*, *a9a*, *real-sim*, *gisette*, *news20*, and

[1]The source code is available at https://github.com/YuqiaWU/HpgSRN

[2]The source code is available at https://github.com/ShenglongZhou/PSNP

rcv1.train—are binary classification examples obtained from the LIBSVM repository[3].

For IReNA, we set $\eta = 1, \eta_\Phi = 1, \eta_\Psi = 1, \gamma_\alpha = 0.5, \gamma_\beta = 0.5, \xi_\beta = 10^{-8}, \xi_\alpha = 10^{-8}, \tau = 10^{-6}, \nu = 1$. For the subspace quadratic subproblem (2.3), we adopt the truncated conjugate gradient method described in (Chen et al. 2017, Section 4.2). In this subproblem, we set $\zeta_k = 10^{-8} + 10^{-4}\|g^k\|^{0.5} + \min\{-\lambda r''(x_i^k) \mid i \in \mathcal{I}(x^k)\}$ to ensure $H^k \succ 0$. For the tailored trust-region Newton subproblem discussed in Section 3.4, we employ the solver from (Curtis et al. 2021, Algorithm 2). In line 7 of Algorithm 3, the initial stepsize $\bar{\beta}$ is determined using the classic Barzilai-Borwein rule (Barzilai and Borwein 1988). We initialize the perturbation values as $\epsilon_i^0 = 1$ for all $i \in [n]$ and update the perturbation ϵ in line 20 according to:

$$\epsilon_i^{k+1} = \begin{cases} 0.9\epsilon_i^k, & k \in \mathcal{S}_{\text{IST}} \text{ and } \mathcal{W}^k \text{ is chosen from (11a)}, i \in \mathcal{I}^{k+1}, \\ 0.9(\epsilon_i^k)^{1.1}, & k \in \mathcal{S}_{\text{IST}} \text{ and } \mathcal{W}^k \text{ is chosen from (11b)}, i \in \mathcal{I}^{k+1}, \\ \min\{0.9\epsilon_i^k, (\epsilon_i^k)^2\}, & k \in \mathcal{S}_{\text{QP}}, i \in \mathcal{I}^{k+1}, \\ \epsilon_i^k, & \text{otherwise.} \end{cases}$$

In addition, before entering the QP phase ($k \in \mathcal{S}_{\text{QP}}$), we enforce a lower bound on each perturbation copmponent during IST updates ($k \in \mathcal{S}_{\text{IST}}$) by setting $\epsilon_i^k \leftarrow \max\{\epsilon_i^k, 10^{-8}\}, \forall i \in [n]$. This precaution ensures that the perturbation value does not become excessively small.

All algorithmic parameters for HpgSRN, PCSNP, and EPIRℓ_1 follow the suggestions in their original publications. In particular, since PCSNP addresses an ℓ_2-norm regularized logistic regression problem, we disable its ℓ_2-norm penalty by setting the regularization coefficient to zero for ensuring a fair comparison. For all algorithms, the initial point was set as $x^0 = 0$.

4.2 Convergence behavior

To visualize the algorithm's convergence behavior, we uniformly discretize the iteration timeline and color-code each update type at each step in Figure 1 for $p = 0.5$. Each row corresponds to one dataset—the first six are synthetic (sizes from 10000×10000 to 100×100), and the last six are real-world datasets, arranged in descending order of size. Each column denotes a time segment, and each cell is color-coded to indicate one of three update types: **IST step 1** (Line 12), **QP step** (Line 9), or **IST step 2** (Line 17).

This grid-based visualization reveals a clear three-phase pattern. In the initial phase, IST step 1 predominates to rapidly identify a low-dimensional active

[3]Refer to https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/

subspace. In the transition phase, QP steps emerge to accelerate progress, while occasional IST step 2 updates serve as corrective refinements before the active set stabilizes. In the final phase, QP steps dominate, yielding rapid convergence within the stabilized subspace. This progression exemplifies the algorithm's core design: use first-order IST updates to identify the active subspace, then switch to the second-order QP updates for local acceleration.

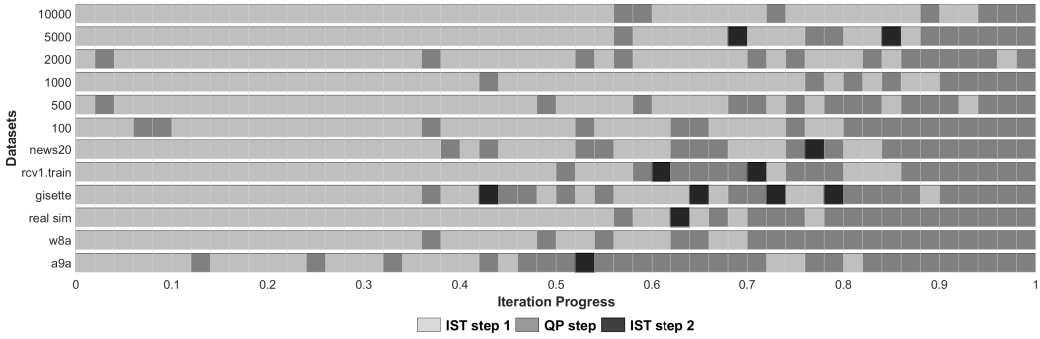


Figure 1: Visualization of algorithmic step types across iterations for multiple datasets.

4.2.1 Local Quadratic Convergence

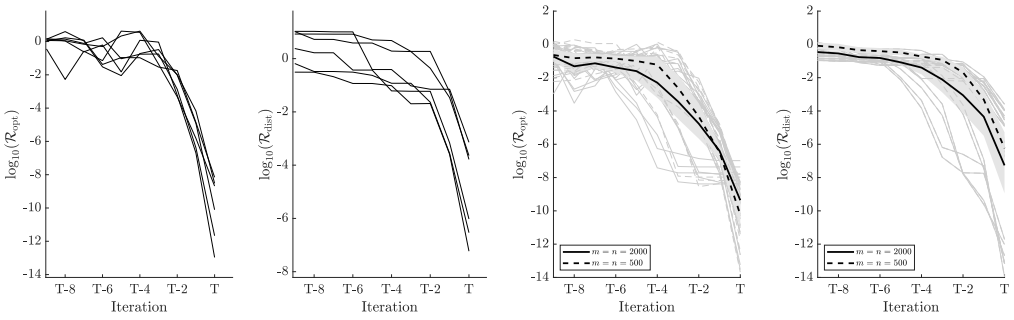
We further demonstrate that IReNA exhibits local quadratic convergence on both the synthetic and real-world datasets. Following (Burke et al. 2014), our demonstration is based on the following two metrics:

$$\mathcal{R}_{\text{opt}}(x) = \|x \circ \nabla f(x) + \lambda p|x|^p\|_\infty, \quad \text{and} \quad \mathcal{R}_{\text{dist}}(x) = \|x - x^*\|_\infty,$$

where x^* is the solution returned by IReNA for $p = 0.5$. Figure 2 displays $\log_{10}(\mathcal{R}_{\text{opt}})$ and $\log_{10}(\mathcal{R}_{\text{dist}})$ over the last ten iterations of Algorithm 3 on six real-world datasets (see panels (a) and (b)) and synthetic datasets (see panels (c) and (d)) of various sizes. For the synthetic datasets, each case was tested over 20 random trials, and the average results are reported. As illustrated in Figure 2, the slopes of these curves in the final iterations generally fall below -2, which indicates local quadratic convergence.

4.3 Numerical results

In this subsection, we compare the performance of IReNA with HpgSRN, PC-SNP and EPIRℓ_1 on real-world datasets. Note that IReNA-RQP refers to the



(a) (b) (c) (d)

Figure 2: Illustration of the local quadratic convergence of IReNA.

variant of IReNA that solves a regularized quadratic subproblem (2.3), whereas IReNA-TR refers to the variant that solves a trust-region subproblem (24). For all methods, we adopt the termination criterion $\mathcal{R}_{\text{opt}}(x) < 10^{-6}$. For the first-order method EPIRℓ_1, we additionally maintain its original termination condition of $\|x^k - x^{k-1}\|/\|x^k\| < 10^{-4}$ to accommodate its slow local convergence.

We evaluate the performance of these methods in terms of CPU time, objective value, and the sparsity level of the obtained solution (e.g., the percentage of zeros). All presented results are averaged over 10 independent runs. In this test, we set $p = 0.5$.

The comparison results are presented in Table 3. Both IReNA-RQP and IReNA-TR consistently deliver faster runtimes and achieve the lowest (or nearly lowest) objective values, and they maintain a sparsity level comparable to that of the benchmark algorithms. During our numerical experiments, we observed that Newton steps in PCSNP are triggered frequently in its early iterations, which appears to account for its comparatively higher CPU time.

4.4 Broader Classes of Nonconvex Regularization and Objectives

To evaluate the performance of IReNA on various nonconvex regularization problems, we incorporate different nonconvex regularizers, as summarized in Table 1, into logistic regression using the *a9a*, *gisette*, and *rcv1.train* datasets. In these evaluations, we set $q = 10^{-5}$ for the LOG regularizer and $q = 0.1$ for all others. Figure 3 presents the logarithmic residuals over the last 10 iterations, where the residual is defined as $\mathcal{R}_{\text{opt}} = \|x \circ (\nabla f(x) + \omega(x; 0) \circ \text{sgn}(x))\|_\infty$.

Table 3: Performance comparison on real-world datasets ($p = 0.5$). The feature matrix size is shown in parentheses. The two best objective values and CPU times are highlighted in bold.

Dataset	Algorithm	Time (s)	Obj. Values	% of zeros
a9a (32561×123)	IReNA-RQP	**0.31**	10577.79	46.34%
	IReNA-TR	**0.33**	**10576.84**	44.72%
	HpgSRN	2.16	10588.76	53.66%
	PCSNP	1.00	10586.02	49.59%
	EPIRℓ_1	5.99	**10570.49**	39.02%
w8a (49749×300)	IReNA-RQP	**0.57**	5879.25	37.00%
	IReNA-TR	**1.00**	5878.75	38.00%
	HpgSRN	2.27	**5856.52**	37.00%
	PCSNP	1.91	5865.53	35.33%
	EPIRℓ_1	13.05	**5865.06**	38.00%
gisette (6000×5000)	IReNA-RQP	40.74	**176.17**	97.02%
	IReNA-TR	**39.80**	**176.96**	96.94%
	HpgSRN	**35.96**	177.64	96.98%
	PCSNP	105.83	182.62	97.12%
	EPIRℓ_1	326.48	178.29	97.02%
real sim (72309×20958)	IReNA-RQP	**4.57**	7157.17	93.59%
	IReNA-TR	8.82	**7141.33**	93.69%
	HpgSRN	**6.02**	7262.25	94.47%
	PCSNP	11.22	7338.50	94.45%
	EPIRℓ_1	33.48	**7152.72**	93.78%
rcv1.train (20242×47236)	IReNA-RQP	**1.29**	**2552.32**	99.06%
	IReNA-TR	2.80	**2549.31**	99.07%
	HpgSRN	**1.50**	2577.47	99.21%
	PCSNP	2.60	2567.11	99.18%
	EPIRℓ_1	14.10	2562.90	99.13%
news20 (19996×1355191)	IReNA-RQP	**19.64**	4014.90	99.97%
	IReNA-TR	89.02	**4012.78**	99.97%
	HpgSRN	**23.89**	4106.68	99.97%
	PCSNP	191.38	4241.60	99.97%
	EPIRℓ_1	207.05	**3983.64**	99.97%

4.4.1 Feedforward neural network training problem

We test our algorithm on sparse image classification using the MNIST dataset, where each 28×28 grayscale digit is vectorized into a 784-dimensional input. The model $h(x; w_i)$ is a fully connected neural network with two hidden layers of 300 and 100 units (with ReLU activation), and a 10-dimensional output layer passed to a softmax classifier. All weights and biases are concatenated into the vector $x \in \mathbb{R}^n$. The training objective is defined as

$$\min_{x\in\mathbb{R}^n} \; F(x) := \frac{1}{m}\sum_{i=1}^{m} \ell(h(x; w_i), y_i) + \lambda h(x),$$

where ℓ is the cross-entropy loss and $\lambda = 1e-4$. For $h(x)$, we consider two choices: ℓ_p regularization with $p = 2/3$, and exponential regularization $R(x) = \sum_{i=1}^{n}(1 - e^{-|x_i|/p})$ with $p = 0.1$.

We compare our method with IRℓ_1(Wang et al. 2023), proximal gradient (PG) method(Tang et al. 2019), and HpgSRN(Wu et al. 2023). For exponential regularization, only IRℓ_1 is included as a baseline mainly due to the absence of a closed-form proximal operator. Each experiment is repeated five times and trained for 100 epochs with a warm start. The average objective values and floating-point operations (FLOPs) are reported at each epoch. Notably, FLOPs directly reflect model sparsity: fewer nonzero weights yield lower computational costs during inference.

To assess convergence behavior, we plot the relative error $\frac{|F(x^k)-F(x^\star)|}{|F(x^\star)|}$, where $x^\star$ is the best solution obtained among all methods, and include a shaded region indicating one standard deviation to reflect variability across runs. As shown in Figure 4, our method consistently attains lower objective values and higher sparsity. This illustrates the advantage of incorporating small perturbations ϵ and curvature information into the subproblem, tending to yield more effective descent directions and better avoidance of poor local minima.

5 Conclusion

In this paper, we have proposed IReNA, a novel hybrid algorithm designed for a class of nonconvex and nonsmooth sparsity-promoting regularization problems. IReNA adaptively switches between subspace iteratively reweighted ℓ_1 updates and subspace regularized Newton steps, using sign changes in consecutive iterates and a rollback safeguard to determine the switch. We proved the global convergence of the entire sequence of iterates and demonstrated local linear convergence under the KL property. Moreover, when the subspace quadratic subproblem is solved exactly, IReNA achieves local quadratic convergence. Extensive numerical experiments on

various model prediction tasks validated the efficiency and robustness of IReNA, demonstrating its promise for large-scale nonconvex sparse optimization.

References

Attouch H, Bolte J, Svaiter BF (2013) 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.

Bareilles G, Iutzeler F, Malick J (2023) Newton acceleration on manifolds identified by proximal gradient methods. *Mathematical Programming* 200(1):37–70.

Barzilai J, Borwein JM (1988) Two-point step size gradient methods. *IMA Journal of Numerical Analysis* 8(1):141–148.

Bradley PS, Mangasarian OL, Street WN (1998) Feature selection via mathematical programming. *INFORMS Journal on Computing* 10(2):209–217.

Burke JV, Curtis FE, Wang H (2014) A sequential quadratic optimization algorithm with rapid infeasibility detection. *SIAM Journal on Optimization* 24(2):839–872.

Candes EJ, Wakin MB, Boyd SP (2008) Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications* 14:877–905.

Chen T, Curtis FE, Robinson DP (2017) A reduced-space algorithm for minimizing ℓ_1-regularized convex functions. *SIAM Journal on Optimization* 27(3):1583–1610.

Chen X, Niu L, Yuan Y (2013) Optimality conditions and a smoothing trust region newton method for non-Lipschitz optimization. *SIAM Journal on Optimization* 23(3):1528–1552.

Chen X, Zhou W (2010) Smoothing nonlinear conjugate gradient method for image restoration using nonsmooth nonconvex minimization. *SIAM Journal on Imaging Sciences* 3(4):765–790.

Curtis FE, Robinson DP, Royer CW, Wright SJ (2021) Trust-region Newton-CG with strong second-order complexity guarantees for nonconvex optimization. *SIAM Journal on Optimization* 31(1):518–544.

Fan J, Li R (2001) Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* 96(456):1348–1360.

Fazel M, Hindi H, Boyd SP (2003) Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices. *Proceedings of the 2003 American Control Conference*, volume 3, 2156–2162 (IEEE).

Hager WW, Zhang H (2006) A survey of nonlinear conjugate gradient methods. *Pacific Journal of Optimization* 2(1):35–58.

Keskar N, Nocedal J, Öztoprak F, Waechter A (2016) A second-order method for convex ℓ_1-regularized optimization with active-set prediction. *Optimization Methods and Software* 31(3):605–621.

Lai MJ, Xu Y, Yin W (2013) Improved iteratively reweighted least squares for unconstrained smoothed ℓ_q minimization. *SIAM Journal on Numerical Analysis* 51(2):927–957.

Lee JD, Sun Y, Saunders MA (2014) Proximal Newton-type methods for minimizing composite functions. *SIAM Journal on Optimization* 24(3):1420–1443.

Lewis AS, Wylie C (2021) Active-set newton methods and partial smoothness. *Mathematics of Operations Research* 46(2):712–725.

Li G, Pong TK (2018) Calculus of the exponent of Kurdyka–Łojasiewicz inequality and its applications to linear convergence of first-order methods. *Foundations of Computational Mathematics* 18(5):1199–1232.

Liang J, Fadili J, Peyré G (2017) Activity identification and local linear convergence of forward–backward-type methods. *SIAM Journal on Optimization* 27(1):408–437.

Liu Y, Lin R (2024) A bisection method for computing the proximal operator of the ℓ_p-norm for any $0 < p < 1$ with application to Schatten p-norms. *Journal of Computational and Applied Mathematics* 447:115897.

Lobo MS, Fazel M, Boyd S (2007) Portfolio optimization with linear and fixed transaction costs. *Annals of Operations Research* 152:341–365.

Lu Z (2014) Iterative reweighted minimization methods for ℓ_p regularized unconstrained nonlinear programming. *Mathematical Programming* 147(1-2):277–307.

Luo ZQ, Pang JS, Ralph D (1996) *Mathematical programs with equilibrium constraints* (Cambridge: Cambridge University Press).

McCulloch JA, St Pierre SR, Linka K, Kuhl E (2024) On sparse regression, ℓ_p-regularization, and automated model discovery. *International Journal for Numerical Methods in Engineering* 125(14):e7481.

Mordukhovich BS, Yuan X, Zeng S, Zhang J (2023) A globally convergent proximal Newton-type method in nonsmooth convex optimization. *Mathematical Programming* 198(1):899–936.

Shi J, Yin W, Osher S, Sajda P (2010) A fast hybrid algorithm for large-scale ℓ_1-regularized logistic regression. *The Journal of Machine Learning Research* 11:713–741.

Sleem OM, Ashour ME, Aybat NS, Lagoa CM (2024) Lp quasi-norm minimization: algorithm and applications. *EURASIP Journal on Advances in Signal Processing* 2024(1):22.

Tang A, Ma R, Miao J, Niu L (2019) Sparse optimization based on non-convex regularization for deep neural networks. *International Conference on Data Service*, 158–166 (Springer).

Wang H, Zeng H, Wang J (2022) An extrapolated iteratively reweighted ℓ_1 method with complexity analysis. *Computational Optimization and Applications* 83(3):967–997.

Wang H, Zeng H, Wang J (2023) Convergence rate analysis of proximal iteratively reweighted ℓ_1 methods for ℓ_p regularization problems. *Optimization Letters* 17(2):413–435.

Wang H, Zeng H, Wang J, Wu Q (2021a) Relating ℓ_p regularization and reweighted ℓ_1 regularization. *Optimization Letters* 15(8):2639–2660.

Wang H, Zhang F, Shi Y, Hu Y (2021b) Nonconvex and nonsmooth sparse optimization via adaptively iterative reweighted methods. *Journal of Global Optimization* 81:717–748.

Wen B, Chen X, Pong TK (2018) A proximal difference-of-convex algorithm with extrapolation. *Computational Optimization and Applications* 69(2):297–324.

Wu Y, Pan S, Yang X (2023) A regularized Newton method for ℓ_q-norm composite optimization problems. *SIAM Journal on Optimization* 33(3):1676–1706.

Xu Z, Chang X, Xu F, Zhang H (2012) $\ell_{1/2}$ regularization: A thresholding representation theory and a fast solver. *IEEE Transactions on Neural Networks and Learning Systems* 23(7):1013–1027.

Yue MC, Zhou Z, So AMC (2019) A family of inexact SQA methods for non-smooth convex minimization with provable convergence guarantees based on the Luo–Tseng error bound property. *Mathematical Programming* 174(1):327–358.

Zeng J, Lin S, Xu Z (2016) Sparse regularization: Convergence of iterative jumping thresholding algorithm. *IEEE Transactions on Signal Processing* 64(19):5106–5118.

Zhang CH (2010) Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics* 38(2):894–942.

Zhao X, Li J, Guo Q (2023) Robust target localization in distributed mimo radar with nonconvex ℓ_p minimization and iterative reweighting. *IEEE Communications Letters* 27(12):3230–3234.

Zhou S, Xiu X, Wang Y, Peng D (2023) Revisiting $\ell_q(0 \leqslant q < 1)$ norm regularized optimization. *arXiv preprint arXiv:2306.14394* .

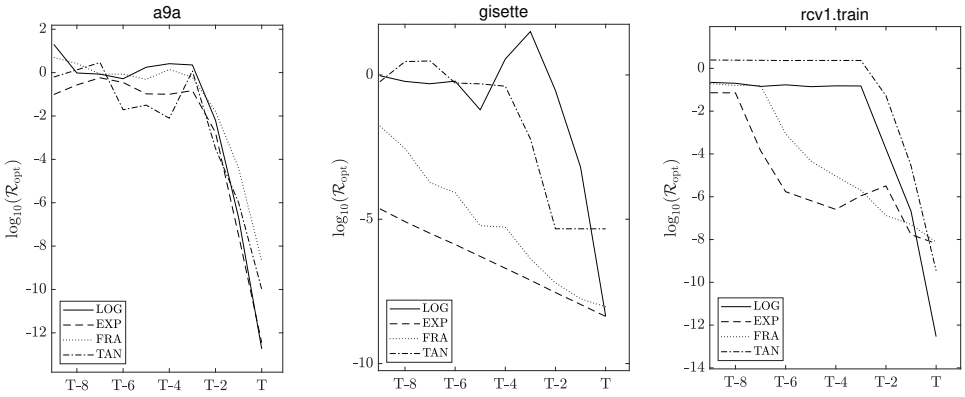


Figure 3: Convergence behavior for other nonconvex regularizers on real-world datasets.

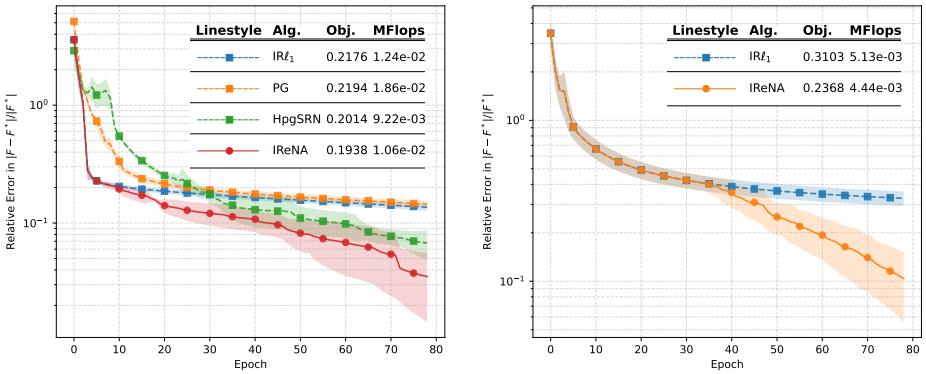


Figure 4: Relative objective error versus epoch for sparse image classification on MNIST, using ℓ_p-norm regularization (left) and exponential regularization (right).