This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2026.3719159

High-Dimensional Huber Regression over Networks

Wenfu Xia, *Student Member, IEEE*, Quan Wei, *Student Member, IEEE*, and Ziping Zhao, *Member, IEEE*

***Abstract*—In modern applications, data are often stored across multiple observation agents, each holding only a subset of the total dataset. In this paper, we investigate distributed robust regression over a network of agents, potentially without a central node, where local datasets may be contaminated by heavy-tailed noise or outliers. We consider the high-dimensional estimation setting by introducing sparsity, which allows the ambient dimension to grow faster than the sample size. The estimation problem is formulated as a Lasso-constrained Huber regression. We first analyze a projected gradient descent algorithm in the centralized setting and establish its convergence properties. Subsequently, we extend our analysis to the decentralized setting, where each agent holds only a subset of observations, and propose an efficient distributed projected gradient descent algorithm, termed Network Huber Lasso (NetHLasso). We derive statistical and computational guarantees, demonstrating that the successive iterates from NetHLasso converge linearly to an estimate with statistical precision matching that of the centralized model under mild conditions. Numerical experiments validate our theoretical results, illustrating the effectiveness of NetHLasso across various heavy-tailed distributions and demonstrating its superior performance in high-dimensional regression compared to existing methods.**

***Index Terms*—Distributed statistical estimation, high-dimensional statistics, sparse learning, robust regression, Huber, linear convergence.**

I. Introduction

Linear regression [2], [3] is a fundamental model in statistical estimation and inference, which plays a crucial role in various fields of data analysis, including biology [4], finance [5], and environmental science [6]. In the context of parameter estimation, the least-squares method is frequently used. However, with the advancement of data analysis technologies, many modern datasets are high-dimensional, where the number of variables far exceeds the number of samples, making the least-squares method ill-posed. To address this challenge, sparse regularization methods have been widely applied to least-squares regression problems, featuring well-known regularization terms such as Lasso (ℓ_1-norm) [7], group Lasso [8], and adaptive Lasso [9]. These regularization techniques promote sparsity by setting many regression coefficients to zero, thus reducing the number of effective variables for estimation. Nevertheless, the theoretical properties of least-squares regression typically rely on (sub-)Gaussian assumptions on the regression residuals, which are unrealistic in many practical scenarios where heavy-tailed noise and outliers are prevalent, thereby deteriorating estimation performance [10], [11]. For instance, the power-law nature of return distributions is a well-documented stylized fact in finance [12], and biological imaging datasets often encounter heavy-tailed noise due to limitations in measurement precision [13].

Many robust estimation methods have been developed for linear regression to address the issue of heavy-tailed errors; refer to [14] for a comprehensive review. One approach is to modify the least-squares loss to be less sensitive to large errors, leading to alternatives such as quantile loss [15], [16], [17], least absolute deviation loss [18], and Huber loss [19], [20]. Quantile loss estimates specific quantiles rather than the mean, as in the least-squares method; hence, extreme outliers do not exert additional influence once the signs of the residuals are determined. Least absolute deviation loss, also known as ℓ_1 loss, measures the error as the absolute value of residuals, which is inherently an estimate of the median. Therefore, it can be seen as a special case of quantile loss when the quantile level is set to 0.5. A limitation of quantile regression is its reliance on the strong assumption that the error distribution is symmetric around zero for it to be applicable to mean regression scenarios [20]. However, in many real-world applications, noise is often asymmetric. For example, in financial asset returns, error distributions frequently exhibit negative skewness [21]. To address this issue, the Huber loss [10] has been widely adopted for mean regression problems, offering robustness against both heavy-tailed and asymmetric noise [19], [20]. The Huber loss combines squared loss for relatively small errors and absolute loss for relatively large errors, with the degree of hybridization controlled by a tuning parameter. The asymptotic properties of the Huber regression estimator have been extensively studied in the literature [22], [23], [24], [25], [26], [27]. Recently, sparse Huber regression methods have been explored for high-dimensional data. The work of [19] proposed incorporating the adaptive Lasso penalty into the Huber regression problem. However, their theoretical analysis is still based on the symmetry assumption of the error distribution. In [20], the authors introduced the Lasso-penalized Huber regression estimator, while [1] incorporated the Lasso constraint with Huber loss to induce sparsity in linear regression. Unlike [19], both [1] and [20] relaxed the symmetry assumption by allowing the regularization parameter to diverge, thereby mitigating the bias when the error distribution is asymmetric. Furthermore, [1] and [20] demonstrated that in the high-dimensional setting, where errors may be heavy-tailed and/or asymmetric, the resulting estimators achieve the same convergence rate as in the light-tailed case, achieving the optimal statistical rate.

In many applications, data are collected independently from multiple individual agents or nodes. Due to constraints in bandwidth, restrictions on storage capacity, and privacy concerns, the data collected by each agent must remain local [28], [29], [30]. For example, in collaborative research between laboratories, agents may be unwilling to share private data

This paper was presented in part at the 13th IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM), Corvallis, Oregon, July 8–11, 2024 [1].

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2026.3719159

with others [31]. These constraints have recently driven increased interest in developing distributed statistical estimation methods that operate across multiple agents connected via a communication network [32]. In this paper, we study the distributed estimation of sparse linear regression in the high-dimensional setting, where the local data may be subject to asymmetric and/or heavy-tailed errors. Within the literature on distributed statistical estimation, two configurations have garnered significant attention: the client-server architecture and the peer-to-peer architecture. In the following, we review existing work on linear regression problems under these two architectures separately.

The client-server architecture relies on a central parameter server to aggregate information and distribute it among all agents. Over the past decade, a significant body of research has emerged on distributed statistical estimation, with one of the fundamental principles being the divide-and-conquer paradigm. This approach constructs a final estimator by aggregating local estimates, i.e., estimates computed independently by individual agents. The divide-and-conquer framework has been widely applied to various regression problems, including linear regression [33], [34], [35], [36], [37], sparse linear regression [38], [39], [40], [41], and robust linear regression [42], [43], [44]. Also referred to as one-step averaging, the divide-and-conquer approach requires only a single round of communication from clients to the server. However, for these estimators to achieve the same convergence rate as the centralized estimator, each local agent must have access to a sufficiently large number of samples, typically proportional to the number of agents [45]. This requirement imposes a constraint on the maximum permissible number of agents within the network, thereby limiting scalability. To overcome the limitations of one-step averaging, multi-round procedures that alternate between local computations and global aggregations have been proposed for distributed estimation in networks with a large number of agents. In high-dimensional settings, distributed approximate Newton-type algorithms [46], [47], [48] and the proximal point algorithm [49] have been applied to sparse regression. For robust regression problems, multi-round methods have been extensively studied for quantile regression [50] and sparse quantile regression [51], [52], [53], [54]. Additionally, [55] investigated distributed sparse Huber regression using an approximate Newton-type method, building on the algorithmic framework in [48]. In [56], a surrogate loss function was constructed to approximate the sparse Huber loss function, which was then solved using a modified proximal alternating direction method of multipliers (ADMM) [57].

The client-server architecture for distributed statistical estimation may be impractical in many real-world applications. For instance, in wireless networks, low-power devices are often constrained to communicate only with nodes in their immediate physical vicinity [58]. A natural approach to this limitation is to eliminate the central node, leading to the peer-to-peer (decentralized) architecture, where each agent can communicate only with its neighbors. The client-server model can be seen as a special case of the peer-to-peer model, while the latter operates without a central node. Due to the absence of a central coordinator, algorithms designed for client-server architectures cannot be directly implemented within peer-to-peer systems, which has spurred significant interest in developing decentralized statistical estimation methods. Algorithms based on ADMM [59], [60] and diffusion-based methods [61], [62] have been used to solve decentralized sparse regression problems. However, these methods primarily focus on optimization aspects and do not establish statistical guarantees. Recently, several statistically guaranteed decentralized algorithms have been proposed for high-dimensional sparse linear regression over networks. In [63], [45], the authors established the statistical and computational guarantees of the distributed gradient descent algorithm for sparse linear regression. This method achieves centralized statistical accuracy at a linear rate, but its communication complexity depends on the ambient dimension, meaning that both the convergence rate and estimation accuracy are affected by the dimensionality. To address this issue, [32] introduced a distributed gradient-tracking-based algorithm that attains centralized statistical accuracy at a linear rate while eliminating dimensional dependency, thereby avoiding the trade-off between speed and accuracy. Later, [64] extended this approach by developing an accelerated version of the distributed algorithm proposed in [32]. In order to obtain robustness, distributed (sub-)gradient descent [65] and ADMM-based methods [66], [67] have been applied to sparse quantile regression problems. However, for decentralized robust regression, statistical guarantees remain scarce. The study in [68] focused on the least absolute loss and demonstrated that the resulting estimator achieves a near-oracle convergence rate in low-dimensional settings, where the sample size exceeds the dimension. In [69], the authors proposed a new quadratic approximation of the quantile loss and employed gradient-tracking-based decentralized gradient descent and a Newton-type method to solve it. The methods in [69] were proven to achieve the optimal statistical rate of $O(d/N)$, where d represents the feature dimension. However, these approaches fail when the ratio d/N diverges, making them unsuitable for high-dimensional scenarios. Furthermore, the aforementioned existing decentralized robust regression models are all based on quantile loss functions, which cannot effectively handle asymmetric noise distributions.

In this paper, we investigate sparse linear regression over networks in a high-dimensional regime, without imposing the classical assumptions that the local data errors are symmetric or light-tailed. In particular, the main contributions of this paper are as follows:

- We investigate high-dimensional sparse Huber regression with potentially heavy-tailed, asymmetric, and/or heteroscedastic regression residuals over a network of agents. The estimation problem is formulated as a Lasso-constrained Huber regression, where each agent holds a subset of the total observations. Furthermore, we present the statistical guarantees for estimation error bounds and support recovery.
- We propose a projected gradient descent algorithm [70] for solving the proposed problem in the centralized setting and establish a new convergence property with

both statistical and computational guarantees.

- We propose a distributed projected gradient descent algorithm based on gradient tracking [71], [72], [73], [74] for the high-dimensional sparse Huber regression problem in a peer-to-peer system. We provide statistical and computational guarantees for the proposed distributed algorithm. Specifically, we show that when $s\log d/N = O(1)$, where s represents the number of nonzero elements, the successive iterates converge linearly at the same rate (in order) as centralized projected gradient descent, achieving an estimate within the centralized statistical precision of the model.
- Numerical experiments validate our theoretical results and demonstrate the effectiveness of the proposed method, even in scenarios where symmetry and light-tailed assumptions do not hold.

The rest of the paper is organized as follows. Section II introduces the necessary background and problem formulation. Section IV focuses on the design of the proposed optimization algorithm in the centralized setting and presents its theoretical properties. In Section V, we develop a decentralized estimation algorithm and establish its statistical and computational guarantees. Section VI provides numerical experiments to validate the theoretical findings. Finally, Section VII concludes the paper with a discussion. The proofs of all theoretical results are provided in Appendix A and in the Supplementary Material.

Notation. The following notation is adopted throughout the paper. Standard lowercase or uppercase letters represent scalars, while boldface lowercase and uppercase letters denote vectors and matrices, respectively. X_{ij} denotes the (i,j)-th entry of the matrix $\boldsymbol{X}$. $\mathbb{R}$ and $\mathbb{R}^m$ denote the sets of real numbers and real m-dimensional vectors, respectively. $\boldsymbol{I}$ stands for the identity matrix. $\boldsymbol{x}^\top$ denotes the transpose of $\boldsymbol{x}$. $\lambda_{\min}(\boldsymbol{X})$ and $\lambda_{\max}(\boldsymbol{X})$ denote the smallest and largest eigenvalues of $\boldsymbol{X}$, respectively. $\|\boldsymbol{X}\|_2$ stands for the matrix spectral norm. $\|\boldsymbol{x}\|_1$, $\|\boldsymbol{x}\|_2$, and $\|\boldsymbol{x}\|_\infty$ denote ℓ_1-norm, ℓ_2-norm, and ℓ_∞-norm, respectively. $\mathrm{P}(\cdot)$ stands for the probability of an event and $\mathrm{E}(\cdot)$ stands for expectation of a random variable. $\nabla f(\cdot)$ represents the gradient of a multivariate function f. $\langle\cdot,\cdot\rangle$ stands for the inner product. For functionals $f(x)$ and $g(x)$, we use $f(x)=O(g(x))$ to indicate that there exists a positive constant M and a constant x_0 such that $|f(x)|\le M|g(x)|$ for any $x\ge x_0$. We write $f(x)=o(g(x))$ if $\lim_{x\to\infty}\frac{f(x)}{g(x)}=0$, which implies that $f(x)$ grows strictly slower than $g(x)$ as $x\to\infty$. C and c stand for universal constants.

II. Background and Problem Setup

In this section, we introduce the data generation model, the estimation problem, and the necessary problem setup.

A. Background

We study the high-dimensional regression problem with data distributed across a network of m agents, modeled as a general connected and undirected graph. Each agent i collects independent and identically distributed (i.i.d.) observations from $(y,\boldsymbol{x})\in\mathbb{R}\times\mathbb{R}^d$, following the linear model

$$y=\boldsymbol{x}^\top\boldsymbol{\theta}^\star+\varepsilon, \tag{1}$$

where $\boldsymbol{x}=[x_1,x_2,\ldots,x_d]^\top$, $\boldsymbol{\theta}^\star\in\mathbb{R}^d$ is the true regression coefficient vector, and ε denotes the error term. For simplicity, we assume that all agents collect the same number n of observations, resulting in a total of $N=m\cdot n$ observations across the network. The combined dataset is given by $\{(y_j,\boldsymbol{x}_j)\}_{j=1}^N$. Each agent i, for $i=1,2,\ldots,m$, stores a subsample of n observations, denoted by $\{(y_j,\boldsymbol{x}_j)\}_{j\in\mathcal{J}_i}$, where $\{\mathcal{J}_i\}_{i=1}^m$ are disjoint index sets satisfying $\bigcup_{i=1}^m\mathcal{J}_i=\{1,2,\ldots,N\}$ and $|\mathcal{J}_i|=n$, with $|\mathcal{J}_i|$ denoting the cardinality of $\mathcal{J}_i$. We focus on the setting where the error variable ε follows an asymmetric heavy-tailed distribution with only a bounded second moment. Additionally, we consider the high-dimensional regime, where the number of features d exceeds the total sample size N, i.e., $d\gg N$. To enable efficient estimation, we assume that the true regression coefficient $\boldsymbol{\theta}^\star$ is s-sparse, meaning that it has at most s nonzero entries.

A standard approach to estimating $\boldsymbol{\theta}^\star$ is to solve the constrained optimization problem

$$\widehat{\boldsymbol{\theta}}\in\underset{\|\boldsymbol{\theta}\|_1\le R}{\arg\min}\left\{\mathcal{L}(\boldsymbol{\theta})=\frac{1}{m}\sum_{i=1}^m\mathcal{L}_i(\boldsymbol{\theta})\right\}, \tag{2}$$

where the ℓ_1-norm constraint promotes sparsity in the solution $\widehat{\boldsymbol{\theta}}$, with R being a predefined constant, and $\mathcal{L}_i$ is the empirical loss for agent i based on its local dataset $\{(y_j,\boldsymbol{x}_j)\}_{j\in\mathcal{J}_i}$. A natural choice for $\mathcal{L}_i$ in (2) is the least-squares (quadratic) loss, which is widely used for fitting high-dimensional sparse linear regression models. However, least-squares estimation is not robust and is highly sensitive to the tails of the error distribution [22]. In the presence of heavy-tailed noise, outliers become more prevalent and can significantly degrade the performance of least-squares estimation. A well-established approach to mitigate the adverse effects of outliers is to employ a robust loss. One commonly used alternative to the least-squares loss is the Huber loss [10], formally defined as follows.

Definition 1. The Huber loss function, denoted as $\ell_\alpha(u)$, is defined as

$$\ell_\alpha(u)=\begin{cases}\alpha|u|-\frac{1}{2}\alpha^2, & \text{if } |u|>\alpha\\ \frac{1}{2}u^2, & \text{if } |u|\le\alpha,\end{cases}$$

where $\alpha>0$ is the robustification parameter, which controls the trade-off between bias and robustness.

The loss function $\ell_\alpha(x)$ is quadratic for small values of x and transitions to a linear form when x exceeds α in magnitude. Compared to the least-squares loss, the Huber loss down-weights outliers, making it more robust to heavy-tailed noise. The parameter α governs the trade-off between the squared and absolute loss components, effectively blending these two extremes. Specifically, as $\alpha\to\infty$, the Huber loss converges to the least-squares loss, whereas as $\alpha\to 0$, it approaches the absolute loss.

Motivated by these properties, we propose using the Huber loss at each agent to study high-dimensional sparse robust regression over networks. More specifically, at agent i, the empirical loss $\mathcal{L}_i$ in (2) is defined as

$$\mathcal{L}_i(\boldsymbol{\theta})=\frac{1}{n}\sum_{j\in\mathcal{J}_i}\ell_\alpha\left(y_j-\boldsymbol{x}_j^\top\boldsymbol{\theta}\right).$$

B. Problem setup

a) Data generation: We first impose a set of mild conditions on the data-generating process (1). Specifically, we assume that the covariate vector $\boldsymbol{x}$ is mean-zero and sub-Gaussian. The regression error ε is required only to have zero conditional mean and a finite conditional second moment. Beyond these conditions, its conditional distribution may be asymmetric, heavy-tailed, and heteroscedastic.

Assumption 1 (Data generation)**.** *In the linear model (1), we assume the following conditions:*

(i) *the covariate vector $\boldsymbol{x}$ is sub-Gaussian:* $\mathrm{P}(|\boldsymbol{u}^\top \widetilde{\boldsymbol{x}}| \geq t) \leq 2\exp(-t^2\|\boldsymbol{u}\|_2^2/A_0^2)$ *for any $t \geq 0$ and $\boldsymbol{u} \in \mathbb{R}^d$, where $\widetilde{\boldsymbol{x}} = \boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{x}$ with $\boldsymbol{\Sigma} = \mathrm{E}(\boldsymbol{x}\boldsymbol{x}^\top)$ satisfying $0 < \beta_l \leq \lambda_{\min}(\boldsymbol{\Sigma}) \leq \lambda_{\max}(\boldsymbol{\Sigma}) \leq \beta_u < \infty$ for some constants β_l and β_u, and $A_0 > 0$ is a constant;*

(ii) *the regression error ε satisfies $\mathrm{E}(\varepsilon \mid \boldsymbol{x}) = 0$, and $\mathrm{E}(\mathrm{E}(\varepsilon^2 \mid \boldsymbol{x})^2) \leq v^2 < \infty$ almost surely.*

Assumption 1 is relatively standard [20], which ensures that the observations $\{\boldsymbol{x}_j\}_{j=1}^N$ are not overly ill-conditioned and that the tail of the noise distribution is not excessively heavy, so that the landscape of the objective exhibits benign properties.

b) Global landscape: We require the global loss function $\mathcal{L}$ to satisfy the following restricted smoothness (RSM) and localized restricted strong convexity (RSC) conditions.

Assumption 2 (RSM for the global loss)**.** *The global loss function $\mathcal{L}$ satisfies the RSM condition with positive parameters (L, τ_L) for all $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathbb{R}^d$:*

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}_1) - \mathcal{L}(\boldsymbol{\theta}_2) - \langle \nabla\mathcal{L}(\boldsymbol{\theta}_2), \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\rangle \\ \leq \frac{L}{2}\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2 + \tau_L\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_1^2.\end{aligned} \tag{3}$$

Assumption 3 (RSC for the global loss)**.** *Let $\mathcal{B}_r = \{\boldsymbol{\theta} \in \mathbb{R}^d \mid \|\boldsymbol{\theta}\|_2 \leq r\}$ denote the ℓ_2-norm ball of radius r centered at $\mathbf{0}$. The global loss function $\mathcal{L}$ satisfies the RSC condition with positive parameters (μ, τ_μ, r) for all $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathcal{B}_r$:*

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}_1) - \mathcal{L}(\boldsymbol{\theta}_2) - \langle \nabla\mathcal{L}(\boldsymbol{\theta}_2), \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\rangle \\ \geq \frac{\mu}{2}\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2 - \tau_\mu\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_1^2.\end{aligned} \tag{4}$$

RSM and RSC are standard assumptions in convergence analyses for high-dimensional estimators, especially for high-dimensional regression [75]. They directly characterize the landscape properties of the objective that are key to obtaining computational and statistical guarantees. Note that in Assumption 3, we impose the RSC condition only within the ℓ_2-norm ball $\mathcal{B}_r$, due to the use of the Huber loss. This differs from the classical RSC condition assumed over the entire domain for the least-squares loss [70], [32]. Based on Assumptions 2 and 3, we define the restricted condition number of the loss function $\mathcal{L}$ as $\kappa = L/\mu \geq 1$.

c) Local landscape: In addition to global landscape assumptions, we also introduce the RSM condition for the local loss function $\mathcal{L}_i$.

Assumption 4 (RSM for the local loss)**.** *Each local loss function $\mathcal{L}_i$ satisfies the RSM condition with positive parameters (ℓ, τ_ℓ) for all $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathbb{R}^d$:*

$$\begin{aligned}\mathcal{L}_i(\boldsymbol{\theta}_1) - \mathcal{L}_i(\boldsymbol{\theta}_2) - \langle \nabla\mathcal{L}_i(\boldsymbol{\theta}_2), \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\rangle \\ \leq \frac{\ell}{2}\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2 + \tau_\ell\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_1^2.\end{aligned} \tag{5}$$

Assumption 4 is commonly adopted in distributed optimization, and it ensures that the local objective landscape is well-behaved [30].

d) Network setups: Finally, we impose conditions on the network structure. The agents are connected through a communication network, which is modeled as an undirected time-invariant graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where the vertex set $\mathcal{V} = \{1, 2, \ldots, m\}$ represents the m agents, and the edge set $\mathcal{E}$ defines the communication links, where $(i, j) \in \mathcal{E}$ if and only if agents i and j can directly communicate. For each agent i, the neighborhood set is denoted as $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\} \cup \{i\}$, which includes all directly connected agents as well as agent i itself. Given an undirected communication graph $\mathcal{G}$, the interactions between agents are encoded in a weight matrix $\boldsymbol{W} \in \mathbb{R}^{m\times m}$, which is a nonnegative matrix that governs the information exchange across the network.

Assumption 5 (Network setup)**.** *The weight matrix $\boldsymbol{W} = (w_{ij})_{i,j=1}^m$ satisfies the following conditions:*

(i) *$w_{ij} > 0$, if $(i, j) \in \mathcal{E}$, otherwise $w_{ij} = 0$; $w_{ii} > 0$ for $i = 1, 2, \ldots, m$;*

(ii) *$\boldsymbol{W} = \boldsymbol{W}^\top$ and $\boldsymbol{W}\mathbf{1} = \mathbf{1}$;*

(iii) *there holds $\rho = \|\boldsymbol{W} - \boldsymbol{J}\|_2 < 1$, where $\boldsymbol{J} = \frac{1}{m}\mathbf{1}\mathbf{1}^\top$ denotes the consensus matrix.*

Assumption 5 is widely used in the development of distributed algorithms [30]. Assumption 5(i) ensures that $\boldsymbol{W}$ is consistent with the graph $\mathcal{G}$, i.e., each agent can exchange information only with its neighbors. Since the network is time-invariant, $\boldsymbol{W}$ is fixed. Assumption 5(ii) guarantees that $\boldsymbol{W}$ is doubly stochastic, which ensures the convergence of the algorithm and consensus among agents. Assumption 5(iii) ensures the graph $\mathcal{G}$ is connected. Intuitively, ρ quantifies the rate at which information propagates across the network; a smaller ρ indicates faster information mixing. In the extreme cases, $\rho = 0$ corresponds to a fully connected network, while $\rho \to 1$ suggests that the network tends to be disconnected. There are many standard designs of $\boldsymbol{W}$ that satisfy Assumption 5, for instance, Metropolis weights [76], [77]

$$W_{ij} = \begin{cases} \frac{1}{\max(d_i, d_j)+1}, & i \neq j \text{ and } (i, j) \in \mathcal{E}, \\ 0, & i \neq j \text{ and } (i, j) \notin \mathcal{E}, \\ 1 - \sum_{l\neq i} W_{il}, & i = j, \end{cases} \tag{6}$$

where d_i is the degree of agent i.

C. Theoretical roadmap

We next proceed to the theoretical and algorithmic development, which is organized in three steps. First, in Section III, we study the global estimator $\widehat{\boldsymbol{\theta}}$ in (2). We establish its estimation error bound and support recovery guarantee, which characterize the statistical precision of the target estimator.

Second, in Section IV, we analyze the projected gradient descent algorithm for solving the global estimation problem. This section serves as both a computational baseline and an analytical warm-up for the decentralized method. Theorem 3 is a deterministic convergence result under the RSM and RSC conditions: it shows that the optimization error of the centralized iterates decreases linearly up to the statistical error level of $\widehat{\boldsymbol{\theta}}$. Corollary 1 then specializes this deterministic theorem to the sparse Huber regression model by using the high-probability verification of the required landscape conditions.

Finally, in Section V, we extend the centralized computation and analysis to the decentralized setting. Theorem 4 establishes a deterministic linear convergence bound for the gradient-tracking-based algorithm toward the same global estimator $\widehat{\boldsymbol{\theta}}$, with an additional network-dependent error term caused by decentralization. Corollary 2 further specializes this result to the sparse Huber regression model. It shows that, under mild conditions, the iterates converge linearly at the same order of rate as the centralized projected gradient descent method, while the limiting error matches the centralized statistical error.

III. Statistical Guarantees

We first analyze the estimation error and support recovery performance of the estimator in (2). The Karush–Kuhn–Tucker (KKT) conditions of problem (2) are

$$\nabla\mathcal{L}(\widehat{\boldsymbol{\theta}}) + \lambda\boldsymbol{\xi} = \mathbf{0}, \quad \lambda\big(\|\widehat{\boldsymbol{\theta}}\|_1 - R\big) = 0, \quad \lambda \geq 0, \quad \|\widehat{\boldsymbol{\theta}}\|_1 \leq R, \tag{7}$$

where λ is the dual variable and $\boldsymbol{\xi} \in \partial\|\widehat{\boldsymbol{\theta}}\|_1$. Here, the dual variable λ and the constraint level R are in one-to-one correspondence via Lagrangian duality [78].

Theorem 1 (Estimation error)**.** *Suppose that Assumptions 1, 2, and 3 hold. Assume that the solution $\widehat{\boldsymbol{\theta}}$ to (2) satisfies $\|\widehat{\boldsymbol{\theta}}\|_1 = R$ and the total sample size N satisfies*

$$N \geq \max\left\{64c_\lambda^4 s, \frac{c_u^2 \max\{R, \|\boldsymbol{\theta}^\star\|_2\}^2}{c_\alpha^2}\right\}\log d.$$

Then, when

$$\lambda = c_\lambda\sqrt{\frac{\log d}{N}}, \quad \alpha = c_\alpha\sqrt{\frac{N}{\log d}},$$

where $c_\lambda^2 > 32c_{\text{grad}}^2 c_v v^2 A_0^2$, $c_\alpha \leq 4c_v v^2 A_0/(c_l c_\lambda)$, $c_u, c_v, c_l, c_\alpha > 0$, and $c_{\text{grad}} \geq 1$, with high probability we have

$$\big\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\big\|_2 = O\left(\sqrt{\frac{s\log d}{N}}\right).$$

Theorem 2 (Support recovery)**.** *Assume the conditions in Theorem 1 are satisfied. Let $\mathcal{S} = \operatorname{supp}(\boldsymbol{\theta}^\star) = \{j \mid |\theta_j^\star| > 0\}$ with $|\mathcal{S}| = s$. Then when $\min_{j\in\mathcal{S}}|\theta_j^\star| \geq 2c_\infty\sqrt{\frac{s\log d}{N}}$ and*

$$c_{\text{grad}} \geq \left(1 + \frac{c_{\rho_+}\beta_u\sqrt{s}}{c_{\rho_-}}\right)(1 - c_p)^{-1},$$

where $c_\infty, c_{\rho_-}, c_{\rho_+} > 0$ and $c_p \in (0,1)$ are constants, with high probability we have $\operatorname{supp}(\widehat{\boldsymbol{\theta}}) = \mathcal{S}$.

Theorem 1 shows that, even with asymmetric and heavy-tailed noise, our estimator can converge to the ground truth $\boldsymbol{\theta}^\star$ with a rate of order $\sqrt{s\log d/N}$ in ℓ_2 norm with high probability, which is the same as the minimax rate for sparse models under light tails [79]. In addition, Theorem 2 shows that our estimator can recover the support of $\boldsymbol{\theta}^\star$ with high probability.

Algorithm 1 Euclidean Projection onto the ℓ_1-Ball [80]

Input: vector $\boldsymbol{u} \in \mathbb{R}^d$, radius $R > 0$, $\mathcal{U} = \{1, \ldots, d\}$, $\zeta = 0$, and $\varsigma = 0$.
While $\mathcal{U} \neq \emptyset$ **do**
  Pick a pivot index $k \in \mathcal{U}$ uniformly at random.
  Partition $\mathcal{U}$ into $\mathcal{U}^+ = \{\, j \in \mathcal{U} :\ |u_j| \geq |u_k| \,\}$ and $\mathcal{U}^- = \{\, j \in \mathcal{U} :\ |u_j| < |u_k| \,\}$.
  Let $\Delta\varsigma = |\mathcal{U}^+|$ and $\Delta\zeta = \sum_{j\in\mathcal{U}^+}|u_j|$.
  If $(\zeta + \Delta\zeta) - (\varsigma + \Delta\varsigma)\,|u_k| < R$ **then**
    $\zeta = \zeta + \Delta\zeta$, $\varsigma = \varsigma + \Delta\varsigma$, $\mathcal{U} = \mathcal{U}^-$;
  else $\mathcal{U} = \mathcal{U}^+ \setminus \{k\}$.
End while
Output:

$$[\Pi_{\mathcal{C}}(\boldsymbol{u})]_i = \operatorname{sign}(u_i)\cdot\max\left\{|u_i| - \frac{\zeta - R}{\varsigma},\ 0\right\},\ i = 1, \ldots, d.$$

IV. Warm-Up: High-Dimensional Sparse Robust Regression

In this section, we introduce the centralized algorithm for high-dimensional sparse robust regression and present its theoretical properties [1]. Consider problem (2) as a centralized estimation problem. The projected gradient descent algorithm [70] is commonly used to solve this convex optimization problem. Given an initial point $\boldsymbol{\theta}^0$, the update rule of the projected gradient descent algorithm at each iteration $t = 0, 1, \ldots$ is given by:

$$\begin{aligned}\boldsymbol{\theta}^{t+1} &\in \arg\min_{\boldsymbol{\theta}\in\mathcal{C}}\left\{\mathcal{L}(\boldsymbol{\theta}^t) + \langle\nabla\mathcal{L}(\boldsymbol{\theta}^t), \boldsymbol{\theta} - \boldsymbol{\theta}^t\rangle + \frac{\gamma}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}^t\|_2^2\right\} \\ &= \Pi_{\mathcal{C}}\left(\boldsymbol{\theta}^t - \frac{1}{\gamma}\nabla\mathcal{L}(\boldsymbol{\theta}^t)\right),\end{aligned} \tag{8}$$

where γ is a proper quadratic coefficient, and $\Pi_{\mathcal{C}}$ denotes the Euclidean projection onto the convex set $\mathcal{C} = \{\boldsymbol{\theta} \in \mathbb{R}^d \mid \|\boldsymbol{\theta}\|_1 \leq R\}$. Notably, this projection has a finite step solution [80], as summarized in Algorithm 1. The iterates $\boldsymbol{\theta}^t$ will converge to the solution $\widehat{\boldsymbol{\theta}}$. We now present a theoretical result on the convergence properties of the algorithm.

Theorem 3. *Suppose that Assumptions 2 and 3 hold with $r \geq \max\{R, \|\boldsymbol{\theta}^\star\|_2\}$. Assume that any solution $\widehat{\boldsymbol{\theta}}$ to (2) satisfies $\|\widehat{\boldsymbol{\theta}}\|_1 = R$ and $\mu \geq 96\max\{\tau_L, \tau_\mu\}s$. Let $\gamma = c_\gamma L$ with $c_\gamma \geq 1$ and $\{\boldsymbol{\theta}^t\}_{t\geq 0}$ be the sequence generated by (8). Then, it holds*

$$\|\boldsymbol{\theta}^t - \widehat{\boldsymbol{\theta}}\|_2^2 \leq \underbrace{C_1\left(1 - \frac{1}{C_2\kappa}\right)^t}_{\text{optimization error}} + \underbrace{O\left((\tau_L + \tau_\mu)\Delta^2\right)}_{\text{statistical error}},$$

where C_1 is a positive constant depending on the initial point $\boldsymbol{\theta}^0$, $C_2 > 1$ is a constant, and

$$\Delta = 2\sqrt{s}\|\boldsymbol{\theta}^\star - \widehat{\boldsymbol{\theta}}\|_2 + 2\|\boldsymbol{\theta}^\star - \widehat{\boldsymbol{\theta}}\|_1. \tag{9}$$

Theorem 3 establishes an optimization–statistical error decomposition for the projected gradient descent algorithm in (8). Specifically, the iterates converge linearly up to a fixed statistical error. This result is stated under the deterministic landscape conditions in Assumptions 2 and 3, and therefore applies to a broad class of high-dimensional sparse estimation problems satisfying these conditions. We next specialize Theorem 3 to the sparse Huber regression model in (2). Under Assumption 1 and a proper choice of the robustification parameter α, Assumptions 2 and 3 hold with high probability for the empirical Huber loss; see Lemma 6 in the Appendix. Combining this high-probability landscape verification with Theorem 3 yields the following corollary.

Corollary 1. *Suppose that Assumption 1 and the conditions in Theorem 3 hold. Assume that $\alpha \geq c_u \max\{R, \|\boldsymbol{\theta}^\star\|_2\}$, where c_u is a positive constant. Then, it holds*

$$\|\boldsymbol{\theta}^t - \widehat{\boldsymbol{\theta}}\|_2^2 \leq C_1 \left(1 - \frac{1}{C_2\kappa}\right)^t + O\left(\frac{s\log d}{N}\|\boldsymbol{\theta}^\star - \widehat{\boldsymbol{\theta}}\|_2^2\right),$$

with high probability.

Note that, by the triangle inequality, the statistical estimation error is given by

$$\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^\star\|_2 \leq \|\boldsymbol{\theta}^t - \widehat{\boldsymbol{\theta}}\|_2 + \|\boldsymbol{\theta}^\star - \widehat{\boldsymbol{\theta}}\|_2.$$

Therefore, Corollary 1 indicates that when $s\log d/N = o(1)$, the projected gradient descent algorithm requires $O\left(\frac{\log(1/\varepsilon)}{\log(1-1/\kappa)}\right)$ iterations to reach an ε-neighborhood of a statistically optimal estimate, with an error of $\|\boldsymbol{\theta}^\star - \widehat{\boldsymbol{\theta}}\|_2^2$.

V. Decentralized High-Dimensional Sparse Robust Regression

In this section, we consider high-dimensional sparse robust regression in a generic peer-to-peer architecture. We develop an efficient distributed algorithm to solve the estimation problem (2) in a decentralized estimation setting. We then present the statistical and computational guarantees of the proposed distributed algorithm. Notably, the proposed algorithm is naturally applicable to the client-server architecture, where the network includes a central node.

A. Optimization algorithm

The proposed estimation algorithm, formally presented in Algorithm 2, is referred to as Network Huber Lasso (NetHLasso), which integrates two communication steps and a local optimization step. Unlike the centralized setting in (8), where the update rule utilizes the global gradient $\nabla\mathcal{L}$, in the decentralized setting, each agent relies only on its local gradient. This is necessary because $\nabla\mathcal{L}$ is not directly accessible to any single agent. Instead, we introduce a local estimate $\boldsymbol{y}_i$ in (12) to approximate $\nabla\mathcal{L}$. The tracking of $\nabla\mathcal{L}$ via $\boldsymbol{y}_i$ is achieved using a dynamic consensus mechanism, commonly referred to as gradient tracking [71], [72], [73], [74]. This technique has been widely incorporated into distributed optimization algorithms to improve convergence rates. At each iteration t, for every agent $i = 1, \ldots, m$ we form the local aggregate

$$\boldsymbol{z}_i^t = \sum_{j\in\mathcal{N}_i} w_{ij}\,\boldsymbol{\theta}_j^t,$$

which aggregates the local estimates of agent i and its neighbors. A key property of this approach is that the average of the local gradient estimates $\boldsymbol{y}_i^t$ approximates the average of the true local gradients $\mathcal{L}_i(\boldsymbol{z}_i^t)$, i.e.,

$$\frac{1}{m}\sum_{i=1}^m \boldsymbol{y}_i^t = \frac{1}{m}\sum_{i=1}^m \nabla\mathcal{L}_i(\boldsymbol{z}_i^t). \tag{10}$$

If asymptotic consensus among the variables $\boldsymbol{z}_i^t$ and $\boldsymbol{y}_i^t$ is achieved, i.e., $\boldsymbol{z}_i^t \to \boldsymbol{z}_j^t$ and $\boldsymbol{y}_i^t \to \boldsymbol{y}_j^t$ as $t \to \infty$ for all $i, j = 1, \ldots, m$, then (10) ensures that the gradient estimates converge to the true global gradient, i.e., $\boldsymbol{y}_i^t \to \nabla\mathcal{L}(\boldsymbol{z}_i^t)$ as $t \to \infty$ for all $i = 1, \ldots, m$. This property ensures that the distributed updates in NetHLasso effectively approximate the centralized projected gradient descent updates, thereby enabling efficient optimization in a decentralized setting.

At each iteration t, we replace the original objective (6) by a strongly convex surrogate around $\boldsymbol{z}_i^t$:

$$\frac{1}{m}\sum_{j=1}^m \mathcal{L}_j(\boldsymbol{z}_i^t) + \left\langle \frac{1}{m}\sum_{j=1}^m \nabla\mathcal{L}_j(\boldsymbol{z}_i^t),\ \boldsymbol{z}_i - \boldsymbol{z}_i^t \right\rangle + \frac{\gamma}{2}\|\boldsymbol{z}_i - \boldsymbol{z}_i^t\|_2^2. \tag{11}$$

After computing $\{\boldsymbol{z}_i^t\}_{i=1}^m$, each agent i forms

$$\tilde{\boldsymbol{y}}_i^t = \boldsymbol{y}_i^{t-1} + \nabla\mathcal{L}_i(\boldsymbol{z}_i^t) - \nabla\mathcal{L}_i(\boldsymbol{z}_i^{t-1}),$$

and aggregates neighbor information to update

$$\boldsymbol{y}_i^t = \sum_{j\in\mathcal{N}_i} w_{ij}\,\tilde{\boldsymbol{y}}_j^t.$$

Since $\boldsymbol{y}_i^t$ tracks the global gradient $\frac{1}{m}\sum_{j=1}^m \nabla\mathcal{L}_j(\boldsymbol{z}_i^t)$, we substitute it into the surrogate subproblem (11) and obtain the local update

$$\begin{aligned}\boldsymbol{\theta}_i^{t+1} &\in \arg\min_{\boldsymbol{z}_i\in\mathcal{C}} \left\{\langle \boldsymbol{y}_i^t,\ \boldsymbol{z}_i - \boldsymbol{z}_i^t\rangle + \frac{\gamma}{2}\|\boldsymbol{z}_i - \boldsymbol{z}_i^t\|_2^2\right\} \\ &= \Pi_{\mathcal{C}}\left(\boldsymbol{z}_i^t - \frac{1}{\gamma}\boldsymbol{y}_i^t\right).\end{aligned}$$

These steps are iteratively performed until convergence.

B. Statistical and computational guarantees

In this section, we establish the convergence properties of Algorithm 2, which serves as the decentralized counterpart of Theorem 3.

Theorem 4. *Suppose that Assumptions 2, 3, 4, and 5 hold with $r \geq \max\{R, \|\theta^\star\|_2\}$. Assume that any solution $\widehat{\boldsymbol{\theta}}$ to (2) satisfies $\|\widehat{\boldsymbol{\theta}}\|_1 = R$, $\mu \geq 240\max\{\tau_L, \tau_\mu\}s$, and*

$$\rho = O\left(\min\left\{\frac{1}{\kappa}, \frac{\mu^2}{(\ell + s\tau_\ell)\mu + ms\tau_\ell\ell + m\ell^2}\right\}\right).$$

Algorithm 2 Network Huber Lasso (NetHLasso)

Input: $\boldsymbol{\theta}_i^1 = \boldsymbol{z}_i^0 = \mathbf{0}$, $\boldsymbol{y}_i^0 = \nabla \mathcal{L}_i(\boldsymbol{\theta}_i^1)$ for $i = 1, 2, \ldots, m$, γ, and $t = 1$;

While not converged, each agent i **do**

#Communication step

pass $\boldsymbol{\theta}_j^t$ to neighbors and update $\boldsymbol{z}_i^t = \sum_{j \in \mathcal{N}_i} w_{ij} \boldsymbol{\theta}_j^t$;

compute $\tilde{\boldsymbol{y}}_i^t = \boldsymbol{y}_i^{t-1} + \nabla \mathcal{L}_i(\boldsymbol{z}_i^t) - \nabla \mathcal{L}_i(\boldsymbol{z}_i^{t-1})$;

pass $\tilde{\boldsymbol{y}}_i^t$ to neighbors and update $\boldsymbol{y}_i^t = \sum_{j \in \mathcal{N}_i} w_{ij} \tilde{\boldsymbol{y}}_j^t$;

#Optimization step

compute $\boldsymbol{\theta}_i^{t+1} = \Pi_{\mathcal{C}} \left(\boldsymbol{z}_i^t - \frac{1}{\gamma} \boldsymbol{y}_i^t \right)$; (12)

$t = t + 1$.

End while

Output: $\{\boldsymbol{\theta}_i^t\}_{i=1}^m$

Let $\gamma = c_\gamma L$ with $c_\gamma \geq 1$, and $\{\{\boldsymbol{\theta}_i^t\}_{i=1}^m\}_{t \geq 1}$ be the sequence generated by Algorithm 2. Then, it holds

$$\frac{1}{m} \sum_{i=1}^m \|\boldsymbol{\theta}_i^t - \widehat{\boldsymbol{\theta}}\|_2^2 \leq \underbrace{C_3 \left(1 - \frac{1}{C_4 \kappa}\right)^t}_{\text{optimization error}} + \underbrace{O\left((\tau_\mu + \tau_L)\Delta^2\right) + O\left(\frac{\rho m \tau_\ell}{(1-\rho)^2} \Delta^2\right)}_{\text{statistical error}},$$

where C_3 is a positive constant depending on the initial points $\{\boldsymbol{\theta}_i^1\}_{i=1}^m$, $C_4 > 1$ is a constant, and Δ is defined in (9).

Theorem 4 provides the distributed counterpart of Theorem 3. The first two error terms match the centralized optimization and statistical errors, while the additional term involving the network parameter ρ captures the effect of decentralization. This network-induced term vanishes when $\rho = 0$, which corresponds to the fully connected case. We next specialize Theorem 4 to the distributed sparse Huber regression model in (1). As in the centralized case, Assumptions 2 and 3 hold with high probability under Assumption 1 and a proper choice of α. Moreover, Lemma 7 in the Appendix shows that Assumption 4 holds with high probability. Combining these high-probability landscape guarantees with Theorem 4 yields the following corollary.

Corollary 2. *Suppose that Assumption 1 and the conditions in Theorem 4 hold. Assume that $\alpha \geq c_u \max\{R, \|\boldsymbol{\theta}^\star\|_2\}$, where c_u is a positive constant. Then, there exists a small ρ such that it holds*

$$\frac{1}{m} \sum_{i=1}^m \|\boldsymbol{\theta}_i^t - \widehat{\boldsymbol{\theta}}\|_2^2 \leq C_3 \left(1 - \frac{1}{C_4 \kappa}\right)^t + O\left(\frac{s \log d}{N} \|\boldsymbol{\theta}^\star - \widehat{\boldsymbol{\theta}}\|_2^2\right),$$

with high probability.

Corollary 2 demonstrates that when $s \log d / N = O(1)$, the successive iterates generated by Algorithm 2 converge linearly to an estimate with the same order of convergence rate as the centralized model.

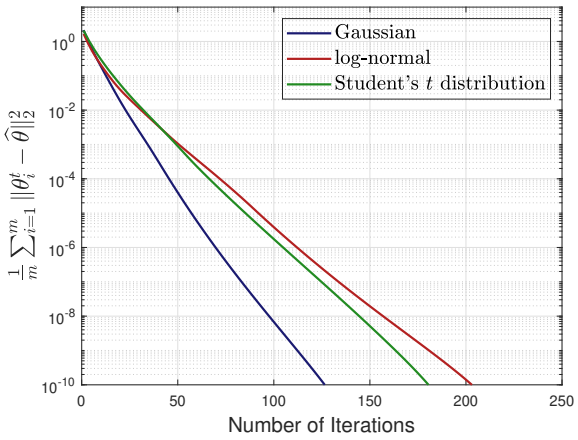


Fig. 1: Linear convergence of NetHLasso.

C. Discussion of heterogeneous sample sizes

In the preceding discussion, we have assumed for simplicity that each agent possesses an equal number of n samples. In practice, however, sample sizes are often heterogeneous across agents; in extreme cases, a small subset of agents may even hold the majority of the samples. In this case, let agent i hold n_i samples, and let $N := \sum_{i=1}^m n_i$. Define the local empirical loss

$$\mathcal{L}_i(\boldsymbol{\theta}) = \frac{1}{n_i} \sum_{j \in \mathcal{J}_i} \ell_\alpha\big(y_j - \boldsymbol{x}_j^\top \boldsymbol{\theta}\big),$$

and replace the unweighted average in (2) by the sample-size–weighted average:

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^m \frac{n_i}{N} \mathcal{L}_i(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^m \sum_{j \in \mathcal{J}_i} \ell_\alpha\big(y_j - \boldsymbol{x}_j^\top \boldsymbol{\theta}\big).$$

For this problem, our algorithm and all theoretical guarantees remain valid with some minor modifications. However, when some agents have very few samples, the local landscape (RSM) becomes worse. According to Lemma 7, one has RSM parameter for agent i's local loss

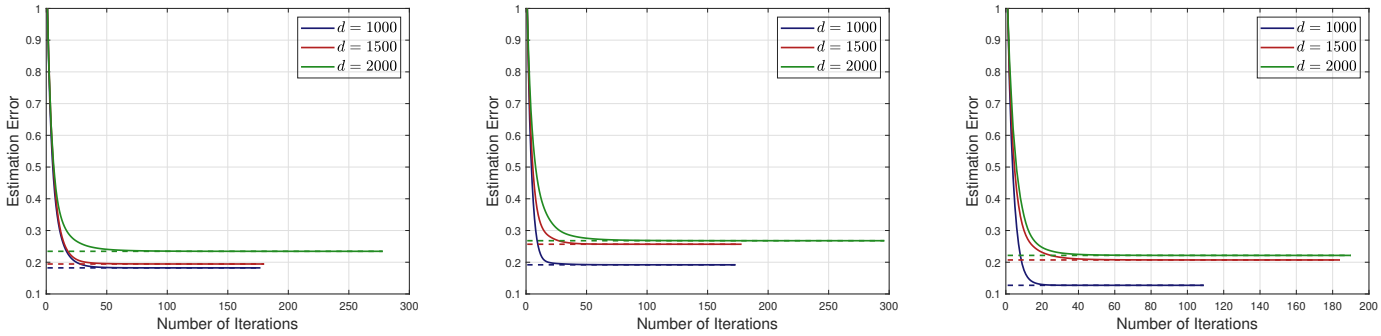
$$\tau_{\ell,i} = O\Big(\frac{\log d}{n_i}\Big),$$

so smaller n_i leads to larger $\tau_{\ell,i}$. By Theorem 4, the admissible upper bound on the network connectivity parameter ρ decreases as τ_ℓ grows. Hence, to maintain the same linear convergence rate when some n_i are small, one may use a better-connected graph (smaller ρ), or perform multiple consensus rounds per local step to shrink the effective connectivity. For K consensus rounds per local step, we have

$$\big\|(\boldsymbol{W} - \boldsymbol{J})^K\big\|_2 = \big\|\boldsymbol{W} - \boldsymbol{J}\big\|_2^K = \rho^K,$$

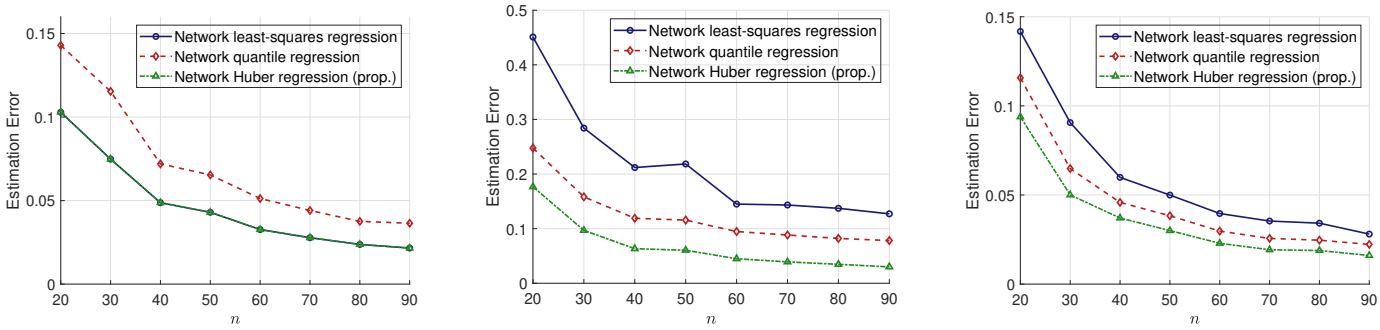
so the effective connectivity $\rho_{\text{eff}} = \rho^K$, and hence choosing $K \geq \log(\bar{\rho}) / \log(\rho)$ yields $\rho_{\text{eff}} \leq \bar{\rho}$ for any target $\bar{\rho} \in (0, 1)$.

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2026.3719159



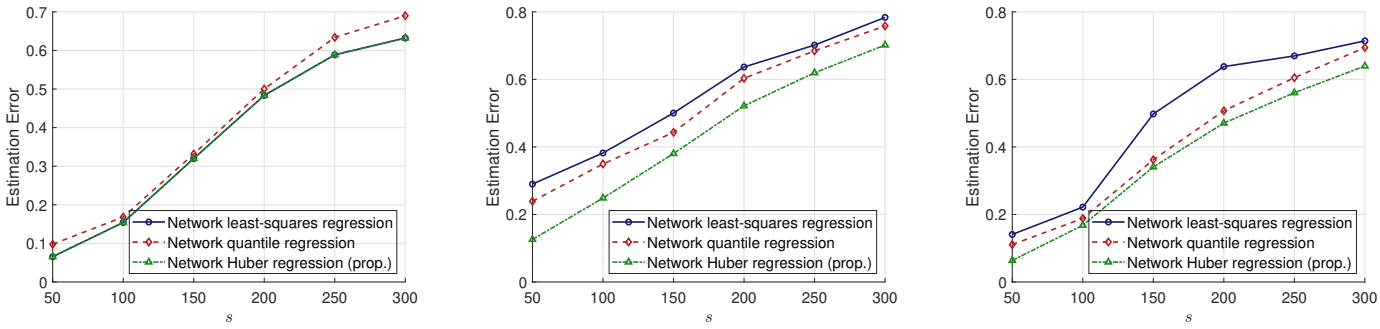
(a) Gaussian distribution (b) log-normal distribution (c) Student's t distribution

Fig. 2: Estimation error versus number of iterations for different feature dimensions.



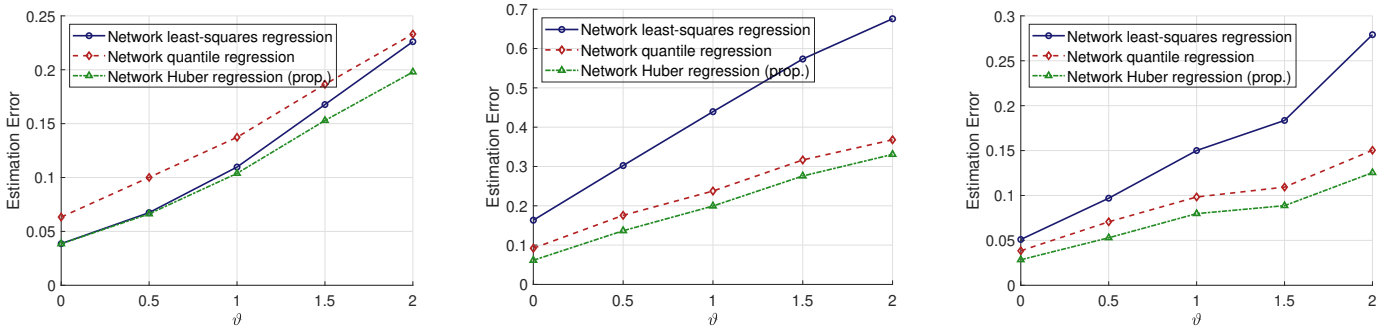
(a) Gaussian distribution (b) log-normal distribution (c) Student's t distribution

Fig. 3: Estimation error versus sample size n for different regression methods.



(a) Gaussian distribution (b) log-normal distribution (c) Student's t distribution

Fig. 4: Estimation error versus sparsity level s for different regression methods.



(a) Gaussian distribution (b) log-normal distribution (c) Student's t distribution

Fig. 5: Estimation error versus heteroscedasticity ϑ for different regression methods.

VI. Numerical Experiments

In this section, we verify our theoretical findings and compare the numerical performance of the proposed method with several state-of-the-art distributed linear regression methods on both synthetic data and a real-world mRNA–protein expression dataset from the Cancer Cell Line Encyclopedia [81], [82].

A. Synthetic data

We consider the following experimental setup. Given the linear regression model in (1), the unknown s-sparse vector $\boldsymbol{\theta}^\star$ is generated such that its first s elements are i.i.d. and follow the standard Gaussian distribution, while the remaining coordinates are set to zero.[1] In each trial, the covariates $\boldsymbol{x}_i$ are first drawn from the standard multivariate Gaussian distribution. To evaluate estimation performance under different noise distributions, we consider the following three scenarios:

- Gaussian distribution with location 0 and scale 1.5;
- Logarithmic normal (log-normal) distribution with log-location 0 and log-scale 1.2;
- Student's t distribution with degrees of freedom 3.

This setup allows us to examine the robustness of the proposed method across various noise distributions. Furthermore, we generate the communication network among m agents using a random Erdős–Rényi graph, where each edge between any two agents is independently included with probability p. We set the weight matrix $\boldsymbol{W}$ as Metropolis weight (6). In our simulation, we set the parameters as follows: $s = 10$, $m = 10$, and $p = 0.8$. The step size γ is tuned to ensure the convergence of the algorithm. Specifically, we set $\gamma = 1.5^k$, where $k \in \{0, 1, 2, \ldots\}$ is the smallest integer ensuring the convergence. NetHLasso is terminated when the maximum number of iterations 20000 is reached or when the maximum relative change of the distributed estimates is below a prescribed tolerance:

$$\max_{i=1,\ldots,m} \frac{\left\|\boldsymbol{\theta}_i^{t+1} - \boldsymbol{\theta}_i^t\right\|_\infty}{1 + \left\|\boldsymbol{\theta}_i^t\right\|_\infty} \leq 10^{-6}.$$

We first examine the numerical convergence of the NetHLasso algorithm. We set $\alpha = 1$, $R = \|\boldsymbol{\theta}^\star\|_1$, $n = 20$, and $d = 2000$. As a benchmark, we employ the projected gradient descent algorithm in (8) to solve problem (2) in a centralized setting, using the resulting estimate as the optimal solution $\widehat{\boldsymbol{\theta}}$. Fig. 1 illustrates the convergence behavior of NetHLasso under three different error distributions, confirming that the iterates converge to $\widehat{\boldsymbol{\theta}}$ at a linear rate, as established in Theorem 4.

The estimation performance is then evaluated, as shown in Fig. 2, where we fix $\alpha = 1$, $R = \|\boldsymbol{\theta}^\star\|_1$, and $n = 20$. Fig. 2 presents the average relative estimation error $\frac{1}{m}\sum_{i=1}^m \|\boldsymbol{\theta}_i^t - \boldsymbol{\theta}^\star\|_2^2 / \|\boldsymbol{\theta}^\star\|_2^2$ against the number of iterations for different feature dimensions, $d = 1000, 1500, 2000$. The dashed-line curves indicate the centralized statistical precision. As observed, in all cases, the estimation error reduces to the centralized statistical precision, which is consistent with our theoretical findings.

[1] According to Corollaries 1 and 2, our method depends only on the number of nonzero entries s, and is not affected by their locations.

Finally, we evaluate the estimation performance of the proposed Huber-based network robust regression model in comparison to existing methods. The compared approaches include network least-squares regression [32] and network quantile regression [66], where for quantile regression we specifically consider the median loss. The robustification parameter α and the constraint parameter R are selected via cross-validation, while the stepsize parameter γ is manually tuned. All results are averaged over 100 Monte Carlo realizations.

In Fig. 3, we fix $d = 2000$ and vary the sample size n from 20 to 90. As expected, the estimation error decreases as n increases for all methods. The results indicate that under heavy-tailed noise settings, our method outperforms both network least-squares regression and network quantile regression in terms of estimation accuracy. In particular, our approach exhibits a notable advantage over network quantile regression in scenarios involving log-normal and other asymmetric error distributions. In the light-tailed case (i.e., Gaussian distribution), our method achieves performance comparable to network least-squares regression, whereas network quantile regression performs the worst.

In Fig. 4, we fix $d = 1000$ and $n = 30$, and vary the sparsity level s from 50 to 300. The estimation error naturally increases with larger sparsity levels. Meanwhile, the proposed method consistently achieves lower estimation error than both baselines across all distributions and sparsity levels.

We further evaluate the performance of the proposed method under heteroscedastic noise. Specifically, we generate the noise

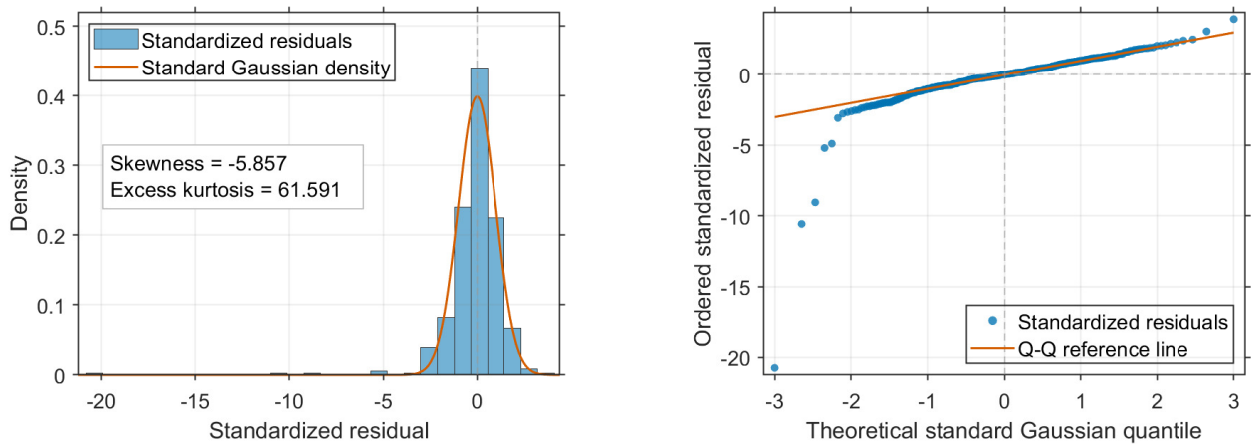
$$\varepsilon = \sigma_0\big(1 + \vartheta|\boldsymbol{a}^\top \boldsymbol{x}|\big)\varepsilon_0,$$

where $\boldsymbol{a}$ is a given random direction, $\sigma_0 > 0$ is a base noise scale, and ε_0 follows one of the three zero-mean distributions considered above (Gaussian, log-normal, or Student's t) [83]. The parameter $\vartheta \geq 0$ controls the degree of heteroscedasticity. When $\vartheta = 0$, the model reduces to the homoscedastic case and ε is independent of $\boldsymbol{x}$; as ϑ increases, the level of heteroscedasticity becomes more severe. In this setting, Assumption 1 is still satisfied. We set the problem dimension to $d = 1000$, the number of samples per agent to $n = 50$, and the sparsity level to $s = 10$. Fig. 5 shows that the estimation error increases as the heteroscedasticity parameter ϑ becomes larger. In addition, our proposed method consistently achieves the lowest estimation error across all cases, and this advantage becomes more pronounced as the heteroscedasticity grows.

B. Real data

We further assess the proposed method on a real-world multi-omics dataset from the Cancer Cell Line Encyclopedia (CCLE) project [81], [82], which provides paired transcriptome-wide mRNA expression and deep quantitative proteomic profiles for a large panel of human cancer cell lines. We select the top 1000 mRNA features with the largest variances and standardize these selected features. We then augment the standardized covariate vector with an additional entry $x_0 = 1$ to incorporate an intercept term in the regression model, thereby accounting for a possible nonzero baseline level of the response. The resulting predictor vector

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2026.3719159



(a) Distribution of standardized residuals

(b) Standard Gaussian Q-Q plot

Fig. 6: Residual diagnostics based on cross-fitted standardized residuals for the CCLE regression task.

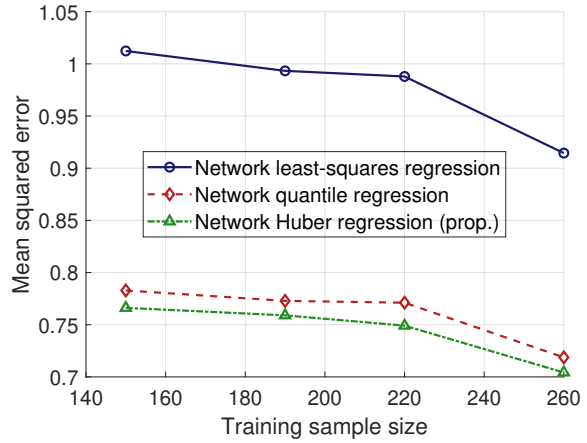


Fig. 7: Estimation error versus training sample size.

is $\boldsymbol{x} \in \mathbb{R}^{d=1001}$, and the protein abundance of the ribosomal protein RPL22 is used as the response y. After preprocessing and aligning cell line identifiers, the final dataset consists of $N = 369$ samples.

We further conduct a residual diagnostic analysis on this dataset. Specifically, we compute out-of-sample residuals from a five-fold cross-fitted Lasso regression. In each outer fold, a Lasso model is trained on four folds, with its regularization parameter selected by an inner five-fold cross-validation, and then used to predict the held-out fold. As shown in Figure 6(a), the distribution of the standardized residuals is clearly asymmetric and shows a pronounced left tail compared with the standard Gaussian density. Figure 6(b) further compares the ordered standardized residuals with the theoretical quantiles of the standard Gaussian distribution through a quantile-quantile (Q-Q) plot, where the substantial deviation in the left tail indicates a departure from Gaussianity. Quantitatively, the standardized residuals have skewness -5.857 and excess kurtosis 61.591, and the Jarque-Bera test [84] rejects Gaussianity with $p < 10^{-16}$. These diagnostic results suggest heavy-tailed and asymmetric residual behavior, thereby motivating the use of the Huber loss rather than the squared loss for this real-world regression task. Moreover, biomedical prediction tasks often arise in multi-site settings where data are collected across different hospitals, laboratories, or research centers and direct data pooling may be restricted by privacy or institutional constraints; see, e.g., the multi-site biomedical learning study in [85]. This further motivates a robust decentralized regression framework that enables collaborative model fitting without centralizing raw data. We note that the main purpose of our experiment is to evaluate the estimation performance of the proposed decentralized algorithm. Therefore, for simplicity, we first standardize the selected mRNA features using the full dataset and then distribute the standardized samples across the network to construct the decentralized learning scenario. This centralized preprocessing step is not essential in a genuinely decentralized implementation. When the data are naturally stored across different nodes, the same standardization can be performed directly over the network before running the decentralized learning algorithm. Specifically, each node first computes the local sample mean of its covariates, and averaging consensus is then applied over the network to obtain the global empirical mean. Given this global mean, each node computes the local sum of squared deviations from the global mean. A second round of averaging consensus is used to obtain the global empirical standard deviation. Each node can subsequently standardize its local covariates using the resulting global mean and standard deviation.

We conduct the experiments over 100 Monte Carlo trials. In each trial, we randomly split the dataset into a training set of 150, 190, 220, or 260 samples and a test set comprising the remaining samples. The training data are further evenly partitioned and assigned to $m = 10$ agents, and the test samples are randomly distributed. The communication network among agents is generated as a random Erdős–Rényi graph with connection probability 0.5, and the consensus weight matrix $\boldsymbol{W}$ is constructed using the Metropolis rule (6). This yields a distributed high-dimensional linear regression problem, in which transcriptomic features are used to predict the protein

abundance of the cancer-relevant ribosomal protein RPL22. Hyperparameters, like α and R, are selected via five-fold cross-validation, where each agent divides its local data into training and validation subsets for parameter tuning. In each fold, all agents first run Algorithm 2 on their local training subsets to obtain $\widehat{\boldsymbol{\theta}}$, compute local validation predictions via $\hat{y}_j = \boldsymbol{x}_j^\top \widehat{\boldsymbol{\theta}}$, and then use averaging consensus to aggregate the local validation squared errors and evaluate the global mean squared error $\frac{1}{N_{\text{val}}} \sum_{j=1}^{N_{\text{val}}} (y_j - \hat{y}_j)^2$, where N_{val} denotes the total number of validation samples and y_j are their true responses. Finally, we select the hyperparameters that minimize this global validation mean squared error.

Fig. 7 illustrates the estimation performance of different methods in terms of mean squared error for test sets. The prediction errors of all three methods decrease as the number of training samples increases. Moreover, the two robust regression approaches, namely network quantile regression and the proposed network Huber regression, consistently outperform the conventional and non-robust network least-squares regression. In particular, our network Huber regression achieves the lowest regression error in all settings. These results suggest that the dataset may contain heavy-tailed noise or outliers, and that Huber regression can achieve accurate estimation under such conditions.

VII. Conclusion

In this paper, we have studied high-dimensional sparse robust regression in a networked setting, where data are distributed across multiple agents and the regression error may follow a heavy-tailed and/or asymmetric distribution. We have first analyzed a projected gradient descent algorithm in the centralized setting and have established its convergence properties. Building on these results, we have proposed a distributed algorithm, NetHLasso, for high-dimensional sparse Huber regression over networks. We have provided both statistical and computational guarantees, showing that the successive iterates of NetHLasso converge linearly to an estimator that achieves the statistical precision of its centralized counterpart. Numerical experiments have further demonstrated the effectiveness of NetHLasso, and the results indicate that it is robust to heavy-tailed noise and achieves higher statistical accuracy than existing distributed regression methods.

Appendix A
Proofs of Theoretical Results in Section II

A. Proof of Theorem 1

Since $\|\widehat{\boldsymbol{\theta}}\|_1 = R$, the KKT conditions of problem (2) given in (7) are consistent with the KKT conditions of the estimation problem defined in [20]. Then, based on Assumptions 1, 2, and 3, Theorem 1 can be proved with similar procedures as the proof of Theorem 3 in [20].

B. Proof of Theorem 2

Let $\mathcal{S} = \operatorname{supp}(\boldsymbol{\theta}^\star)$ with $|\mathcal{S}| = s$. Define the oracle estimator

$$\widehat{\boldsymbol{\theta}}^{\,\mathrm{O}} \in \underset{\boldsymbol{\theta}:\ \operatorname{supp}(\boldsymbol{\theta}) \subseteq \mathcal{S}, \|\boldsymbol{\theta}\|_1 \le R}{\arg\min} \mathcal{L}(\boldsymbol{\theta}). \tag{13}$$

We prove Theorem 2 by two steps: 1). showing the uniqueness, the ℓ_∞-norm error bound, and the oracle support recovery of $\widehat{\boldsymbol{\theta}}^{\,\mathrm{O}}$; 2). proving that under the conditions given in Theorem 2, $\widehat{\boldsymbol{\theta}}^{\,\mathrm{O}}$ is also the unique solution of the original problem (2), and hence recover the exact support of $\boldsymbol{\theta}^\star$.

a) Technical lemmas: We first present some required technical lemmas for Theorem 2.

Lemma 1. *For any sparsity level $\tilde{s} \ge 1$ and radius $R^\star \ge \|\boldsymbol{\theta}^\star\|_2$, define the sparse eigenvalues of the Hessian restricted to the ℓ_2-ball by*

$$\rho_-\big(\nabla^2\mathcal{L}, \tilde{s}, R^\star\big) = \inf_{\substack{\|\boldsymbol{v}\|_2=1,\ \|\boldsymbol{v}\|_0 \le \tilde{s} \\ \|\boldsymbol{\theta}\|_2 \le R^\star}} \left\{\boldsymbol{v}^\top \nabla^2\mathcal{L}(\boldsymbol{\theta})\, \boldsymbol{v}\right\},$$
$$\rho_+\big(\nabla^2\mathcal{L}, \tilde{s}, R^\star\big) = \sup_{\substack{\|\boldsymbol{v}\|_2=1,\ \|\boldsymbol{v}\|_0 \le \tilde{s} \\ \|\boldsymbol{\theta}\|_2 \le R^\star}} \left\{\boldsymbol{v}^\top \nabla^2\mathcal{L}(\boldsymbol{\theta})\, \boldsymbol{v}\right\}.$$

Suppose Assumption 1 holds. Then, under the sample size condition $N \gtrsim s \log d$ and the scaling condition $\mu \gtrsim s \cdot \max\{\tau_L, \tau_\mu\}$, there exist absolute constants $c_{\rho_-}, c_{\rho_+} > 0$ and $\tilde{s}$ satisfying

$$\tilde{s} \ge \big(144\kappa_\rho^2 + 250\kappa_\rho\big)s, \quad \kappa_\rho = \frac{\rho_+\big(\nabla^2\mathcal{L}, s+2\tilde{s}, R^\star\big)}{\rho_-\big(\nabla^2\mathcal{L}, s+2\tilde{s}, R^\star\big)},$$

such that we have the restricted eigenvalue condition

$$c_{\rho_-} \le \rho_-\big(\nabla^2\mathcal{L}, s+2\tilde{s}, R^\star\big) \le \rho_+\big(\nabla^2\mathcal{L}, s+2\tilde{s}, R^\star\big) \le c_{\rho_+}\beta_u$$

with high probability.

Proof. The proof follows from the arguments in Appendix C of [86] with minor modification, and hence is omitted. □

Lemma 2. *Suppose Assumptions 1, 2, and 3 hold. Assume*

$$\lambda = c_\lambda \sqrt{\frac{\log d}{N}}, \quad \alpha = c_\alpha \sqrt{\frac{N}{\log d}}, \tag{14}$$

where $c_\lambda^2 > 32 c_{\text{grad}}^2 c_v v^2 A_0^2$, $c_\alpha \le 4 c_v v^2 A_0/(c_l c_\lambda)$, $c_v, c_l, c_\alpha > 0$, and $c_{\text{grad}} \ge 1$. Then with probability at least $1 - 2e^{-(c_\lambda^2/(32 c_{\text{grad}}^2 c_v v^2 A_0^2)-1)N}$, we have

$$\big\|\nabla\mathcal{L}(\boldsymbol{\theta}_\alpha^\star)\big\|_\infty \le \frac{\lambda}{c_{\text{grad}}},$$

where $\boldsymbol{\theta}_\alpha^\star \in \arg\min_{\boldsymbol{\theta}} \mathrm{E}\left\{\ell_\alpha\left(y_j - \boldsymbol{x}_j^\top\boldsymbol{\theta}\right)\right\}$.

Proof. The proof follows from the arguments in Appendix A.1 of [20] with minor modification, and hence is omitted. □

Lemma 3 ([20, Theorem 1])**.** *Assume Assumptions 1, 2, and 3 hold. Then there exists a universal constant $c_e > 0$ such that*

$$\big\|\boldsymbol{\theta}_\alpha^\star - \boldsymbol{\theta}^\star\big\|_2 \le \frac{c_e}{\alpha}\frac{\sqrt{\beta_u}}{\beta_l}\Big(A_0 + v\Big). \tag{15}$$

Lemma 4. *Suppose Assumptions 1, 2, and 3 hold and the conditions in Lemma 1 and Lemma 2 are satisfied. Fix any integer $\tilde{s} \ge 1$ that fulfills the requirement in Lemma 1, and choose a radius*

$$R^\star \ge \sup_{t\in[0,1]} \Big\|\boldsymbol{\theta}_\alpha^\star + t\big(\boldsymbol{\theta}^\star - \boldsymbol{\theta}_\alpha^\star\big)\Big\|_2.$$

Then, with probability at least

$$1-2\exp\left(-\left(\frac{c_\lambda^2}{32c_{\text{grad}}^2\, c_v v^2 A_0^2}-1\right)N\right),$$

we have

$$\left\|\nabla\mathcal{L}(\boldsymbol{\theta}^\star)\right\|_\infty \le \frac{c_\lambda}{c_{\text{grad}}}\sqrt{\frac{\log d}{N}}+\frac{c_e c_{\rho_+}\beta_u}{\alpha}\frac{\sqrt{\beta_u}}{\beta_l}(A_0+v), \tag{16}$$

with high probability.

Proof. By the triangle inequality, we have

$$\left\|\nabla\mathcal{L}(\boldsymbol{\theta}^\star)\right\|_\infty \le \left\|\nabla\mathcal{L}(\boldsymbol{\theta}_\alpha^\star)\right\|_\infty + \left\|\nabla\mathcal{L}(\boldsymbol{\theta}^\star)-\nabla\mathcal{L}(\boldsymbol{\theta}_\alpha^\star)\right\|_\infty.$$

By Lemma 2, with probability at least $1-2\exp\big(-(\frac{c_\lambda^2}{32c_{\text{grad}}^2\, c_v v^2 A_0^2}-1)N\big)$, we have $\|\nabla\mathcal{L}(\boldsymbol{\theta}_\alpha^\star)\|_\infty \le c_\lambda\sqrt{\log d/N}/c_{\text{grad}}$.

For the second term, given any $j\in\mathcal{S}$, there exists $t\in[0,1]$ such that

$$\begin{aligned}&\left[\nabla\mathcal{L}(\boldsymbol{\theta}^\star)-\nabla\mathcal{L}(\boldsymbol{\theta}_\alpha^\star)\right]_j\\ &\quad=\int_0^1 \boldsymbol{e}_j^\top\nabla^2\mathcal{L}\big(\boldsymbol{\theta}_\alpha^\star+t(\boldsymbol{\theta}^\star-\boldsymbol{\theta}_\alpha^\star)\big)\,(\boldsymbol{\theta}^\star-\boldsymbol{\theta}_\alpha^\star)\,\mathrm{d}t,\end{aligned} \tag{17}$$

where $\boldsymbol{e}_j$ is the j-th canonical basis vector. By the Cauchy–Schwarz inequality for quadratic forms, we have

$$\begin{aligned}&\left|\boldsymbol{e}_j^\top\nabla^2\mathcal{L}(\boldsymbol{\theta}_\alpha^\star+t(\boldsymbol{\theta}^\star-\boldsymbol{\theta}_\alpha^\star))(\boldsymbol{\theta}^\star-\boldsymbol{\theta}_\alpha^\star)\right|\\ \le&\sqrt{\boldsymbol{e}_j^\top\nabla^2\mathcal{L}(\boldsymbol{\theta}_\alpha^\star+t(\boldsymbol{\theta}^\star-\boldsymbol{\theta}_\alpha^\star))\,\boldsymbol{e}_j}\\ &\cdot\sqrt{(\boldsymbol{\theta}^\star-\boldsymbol{\theta}_\alpha^\star)^\top\nabla^2\mathcal{L}(\boldsymbol{\theta}_\alpha^\star+t(\boldsymbol{\theta}^\star-\boldsymbol{\theta}_\alpha^\star))(\boldsymbol{\theta}^\star-\boldsymbol{\theta}_\alpha^\star)}.\end{aligned}$$

Because the entire path lies in $\{\|\boldsymbol{\theta}\|_2\le R^\star\}$, Lemma 1 yields

$$\boldsymbol{e}_j^\top\nabla^2\mathcal{L}(\boldsymbol{\theta}_\alpha^\star+t(\boldsymbol{\theta}^\star-\boldsymbol{\theta}_\alpha^\star))\boldsymbol{e}_j \le \rho_+\big(\nabla^2\mathcal{L},1,R^\star\big),$$

and

$$\begin{aligned}&(\boldsymbol{\theta}^\star-\boldsymbol{\theta}_\alpha^\star)^\top\nabla^2\mathcal{L}(\boldsymbol{\theta}_\alpha^\star+t(\boldsymbol{\theta}^\star-\boldsymbol{\theta}_\alpha^\star))\,(\boldsymbol{\theta}^\star-\boldsymbol{\theta}_\alpha^\star)\\ \le&\rho_+\big(\nabla^2\mathcal{L},s+2\tilde{s},R^\star\big)\,\|\boldsymbol{\theta}^\star-\boldsymbol{\theta}_\alpha^\star\|_2^2.\end{aligned}$$

In addition, according to the definition of ρ_+ in Lemma 1, we have

$$\begin{aligned}\rho_+\big(\nabla^2\mathcal{L},1,R^\star\big)=&\sup_{\substack{\|\boldsymbol{v}\|_2=1,\,\|\boldsymbol{v}\|_0\le1\\ \|\boldsymbol{\theta}\|_2\le R^\star}}\left\{\boldsymbol{v}^\top\nabla^2\mathcal{L}(\boldsymbol{\theta})\,\boldsymbol{v}\right\}\\ \le&\sup_{\substack{\|\boldsymbol{v}\|_2=1,\,\|\boldsymbol{v}\|_0\le s+2\tilde{s}\\ \|\boldsymbol{\theta}\|_2\le R^\star}}\left\{\boldsymbol{v}^\top\nabla^2\mathcal{L}(\boldsymbol{\theta})\,\boldsymbol{v}\right\}\\ =&\rho_+\big(\nabla^2\mathcal{L},s+2\tilde{s},R^\star\big)\end{aligned}$$

Therefore, since $\rho_+\big(\nabla^2\mathcal{L},s+2\tilde{s},R^\star\big)\le c_{\rho_+}\beta_u$, we have

$$\left|[\nabla\mathcal{L}(\boldsymbol{\theta}^\star)-\nabla\mathcal{L}(\boldsymbol{\theta}_\alpha^\star)]_j\right| \le c_{\rho_+}\beta_u\,\|\boldsymbol{\theta}^\star-\boldsymbol{\theta}_\alpha^\star\|_2.$$

Taking the supremum gives

$$\left\|\nabla\mathcal{L}(\boldsymbol{\theta}^\star)-\nabla\mathcal{L}(\boldsymbol{\theta}_\alpha^\star)\right\|_\infty \le c_{\rho_+}\beta_u\|\boldsymbol{\theta}^\star-\boldsymbol{\theta}_\alpha^\star\|_2.$$

Finally, applying Lemma 3 yields $\|\boldsymbol{\theta}^\star-\boldsymbol{\theta}_\alpha^\star\|_2\le\frac{c_e}{\alpha}\frac{\sqrt{\beta_u}}{\beta_l}(A_0+v)$, which proves (16). □

b) Step 1. Oracle support recovery: Then, we show the uniqueness, the ℓ_∞-norm error bound, and the oracle support recovery of $\widehat{\boldsymbol{\theta}}^{\mathrm{O}}$ in the following lemma.

Lemma 5 (Oracle support recovery)**.** *Suppose Assumptions 1, 2, and 3 hold and the conditions in Lemma 1, Lemma 2, and Lemma 4 are satisfied. Assume that $R^\star\ge R$. Then, with high probability, the following statements hold:*

(a) ***Uniqueness.*** *$\widehat{\boldsymbol{\theta}}^{\mathrm{O}}$ is the unique minimizer of the oracle estimation problem.*

(b) ***ℓ_∞-norm error bound.***

$$\|\widehat{\boldsymbol{\theta}}^{\mathrm{O}}-\boldsymbol{\theta}^\star\|_\infty \le c_\infty\sqrt{\frac{s\log d}{N}},$$

where

$$c_\infty=\frac{1}{c_{\rho_-}}\left(\frac{1}{c_{\text{grad}}}+\frac{c_e c_{\rho_+}\beta_u}{c_\lambda c_\alpha}\frac{\sqrt{\beta_u}}{\beta_l}(A_0+v)\right)c_\lambda.$$

(c) ***Oracle support recovery.*** *If $\min_{j\in\mathcal{S}}|\theta_j^\star|\ge 2c_\infty\sqrt{\frac{s\log d}{N}}$, then $\operatorname{supp}(\widehat{\boldsymbol{\theta}}^{\mathrm{O}})=\mathcal{S}$.*

Proof. **Uniqueness.** Take arbitrary feasible $\boldsymbol{\theta},\boldsymbol{\theta}'$ with $\operatorname{supp}(\boldsymbol{\theta})\subseteq\mathcal{S}$, $\operatorname{supp}(\boldsymbol{\theta}')\subseteq\mathcal{S}$ and $\|\boldsymbol{\theta}\|_2,\|\boldsymbol{\theta}'\|_2\le R^\star$. By Taylor's theorem, there exists a constant $\eta\in[0,1]$ such that

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}')=&\mathcal{L}(\boldsymbol{\theta})+\big\langle\nabla\mathcal{L}(\boldsymbol{\theta}),\boldsymbol{\theta}'-\boldsymbol{\theta}\big\rangle\\ &+\frac{1}{2}(\boldsymbol{\theta}'-\boldsymbol{\theta})^\top\nabla^2\mathcal{L}(\eta\boldsymbol{\theta}'+(1-\eta)\boldsymbol{\theta})(\boldsymbol{\theta}'-\boldsymbol{\theta}).\end{aligned}$$

Since $\|\boldsymbol{\theta}'-\boldsymbol{\theta}\|_0\le s\le s+2\tilde{s}$, the segment lies in $\mathcal{B}_{R^\star}$, and Lemma 1, we have

$$(\boldsymbol{\theta}'-\boldsymbol{\theta})^\top\nabla^2\mathcal{L}(\eta\boldsymbol{\theta}'+(1-\eta)\boldsymbol{\theta})(\boldsymbol{\theta}'-\boldsymbol{\theta})\ge c_{\rho_-}\|\boldsymbol{\theta}'-\boldsymbol{\theta}\|_2^2.$$

Hence

$$\mathcal{L}(\boldsymbol{\theta}')\ge\mathcal{L}(\boldsymbol{\theta})+\big\langle\nabla\mathcal{L}(\boldsymbol{\theta}),\,\boldsymbol{\theta}'-\boldsymbol{\theta}\big\rangle+\frac{c_{\rho_-}}{2}\|\boldsymbol{\theta}'-\boldsymbol{\theta}\|_2^2,$$

i.e., $\mathcal{L}$ is strongly convex on the feasible set, which implies the uniqueness of $\widehat{\boldsymbol{\theta}}^{\mathrm{O}}$.

ℓ_∞ norm error bound. Restricting to coordinates $\mathcal{S}$, the feasible region is the convex set $\mathcal{B}_{R^\star}\cap\mathbb{R}^s$. According to the first-order optimality condition of the constrained problem, for any $\boldsymbol{\theta}_{\mathcal{S}}\in\mathcal{B}_{R^\star}\cap\mathbb{R}^s$, we have

$$\big\langle\nabla_{\mathcal{S}}\mathcal{L}(\widehat{\boldsymbol{\theta}}^{\mathrm{O}}),\ \boldsymbol{\theta}_{\mathcal{S}}-\widehat{\boldsymbol{\theta}}^{\mathrm{O}}_{\mathcal{S}}\big\rangle\ge 0.$$

Since $\|\boldsymbol{\theta}^\star_{\mathcal{S}}\|_2\le R^\star$ and plugging $\boldsymbol{\theta}_{\mathcal{S}}=\boldsymbol{\theta}^\star_{\mathcal{S}}$, we have

$$\big\langle\nabla_{\mathcal{S}}\mathcal{L}(\widehat{\boldsymbol{\theta}}^{\mathrm{O}})-\nabla_{\mathcal{S}}\mathcal{L}(\boldsymbol{\theta}^\star),\boldsymbol{\theta}^\star_{\mathcal{S}}-\widehat{\boldsymbol{\theta}}^{\mathrm{O}}_{\mathcal{S}}\big\rangle+\big\langle\nabla_{\mathcal{S}}\mathcal{L}(\boldsymbol{\theta}^\star),\boldsymbol{\theta}^\star_{\mathcal{S}}-\widehat{\boldsymbol{\theta}}^{\mathrm{O}}_{\mathcal{S}}\big\rangle\ge 0. \tag{18}$$

By the integral mean-value form of Taylor's theorem along the segment $\{\boldsymbol{\theta}^\star+t(\widehat{\boldsymbol{\theta}}^{\mathrm{O}}-\boldsymbol{\theta}^\star):t\in[0,1]\}\subset\mathcal{B}_{R^\star}$, we have

$$\begin{aligned}&\nabla_{\mathcal{S}}\mathcal{L}(\widehat{\boldsymbol{\theta}}^{\mathrm{O}})-\nabla_{\mathcal{S}}\mathcal{L}(\boldsymbol{\theta}^\star)\\ =&\int_0^1\nabla^2_{\mathcal{S}\mathcal{S}}\mathcal{L}\big(\boldsymbol{\theta}^\star+t(\widehat{\boldsymbol{\theta}}^{\mathrm{O}}-\boldsymbol{\theta}^\star)\big)(\widehat{\boldsymbol{\theta}}^{\mathrm{O}}_{\mathcal{S}}-\boldsymbol{\theta}^\star_{\mathcal{S}})\mathrm{d}t.\end{aligned}$$

Taking inner product with $\boldsymbol{\theta}^\star_{\mathcal{S}}-\widehat{\boldsymbol{\theta}}^{\mathrm{O}}_{\mathcal{S}}$, we have

$$\big\langle\nabla_{\mathcal{S}}\mathcal{L}(\widehat{\boldsymbol{\theta}}^{\mathrm{O}})-\nabla_{\mathcal{S}}\mathcal{L}(\boldsymbol{\theta}^\star_{\mathcal{S}}\boldsymbol{\theta}^\star),\boldsymbol{\theta}^\star_{\mathcal{S}}-\widehat{\boldsymbol{\theta}}^{\mathrm{O}}_{\mathcal{S}}\big\rangle \tag{19}$$

$$= -\int_0^1 (\widehat{\boldsymbol{\theta}}^{\mathsf{O}}_{\mathcal{S}} - \boldsymbol{\theta}^\star_{\mathcal{S}})^\top \nabla^2_{\mathcal{S}\mathcal{S}} \mathcal{L}\big(\boldsymbol{\theta}^\star + t(\widehat{\boldsymbol{\theta}}^{\mathsf{O}} - \boldsymbol{\theta}^\star)\big)(\widehat{\boldsymbol{\theta}}^{\mathsf{O}}_{\mathcal{S}} - \boldsymbol{\theta}^\star_{\mathcal{S}})\mathrm{d}t.$$

Combining (18) and (19), we have

$$\int_0^1 (\widehat{\boldsymbol{\theta}}^{\mathsf{O}}_{\mathcal{S}} - \boldsymbol{\theta}^\star_{\mathcal{S}})^\top \nabla^2_{\mathcal{S}\mathcal{S}} \mathcal{L}\big(\boldsymbol{\theta}^\star + t(\widehat{\boldsymbol{\theta}}^{\mathsf{O}} - \boldsymbol{\theta}^\star)\big)(\widehat{\boldsymbol{\theta}}^{\mathsf{O}}_{\mathcal{S}} - \boldsymbol{\theta}^\star_{\mathcal{S}})\mathrm{d}t \leq \big\langle \nabla_{\mathcal{S}} \mathcal{L}(\boldsymbol{\theta}^\star),\ \boldsymbol{\theta}^\star_{\mathcal{S}} - \widehat{\boldsymbol{\theta}}^{\mathsf{O}}_{\mathcal{S}} \big\rangle. \tag{20}$$

Since $\mathrm{supp}(\boldsymbol{\theta}^\star_{\mathcal{S}} - \widehat{\boldsymbol{\theta}}^{\mathsf{O}}_{\mathcal{S}}) \subseteq \mathcal{S}$ with $|\mathcal{S}| = s \leq s + 2\tilde{s}$ and the path lies in $\mathcal{B}_{R^\star}$, Lemma 1 yields

$$\frac{(\widehat{\boldsymbol{\theta}}^{\mathsf{O}}_{\mathcal{S}} - \boldsymbol{\theta}^\star_{\mathcal{S}})^\top}{\|\widehat{\boldsymbol{\theta}}^{\mathsf{O}}_{\mathcal{S}} - \boldsymbol{\theta}^\star_{\mathcal{S}}\|_2} \nabla^2_{\mathcal{S}\mathcal{S}} \mathcal{L}\big(\boldsymbol{\theta}^\star + t(\widehat{\boldsymbol{\theta}}^{\mathsf{O}} - \boldsymbol{\theta}^\star)\big) \frac{(\widehat{\boldsymbol{\theta}}^{\mathsf{O}}_{\mathcal{S}} - \boldsymbol{\theta}^\star_{\mathcal{S}})}{\|\widehat{\boldsymbol{\theta}}^{\mathsf{O}}_{\mathcal{S}} - \boldsymbol{\theta}^\star_{\mathcal{S}}\|_2} \geq c_{\rho_-}.$$

for all points on the segment. Thus (20) implies that

$$c_{\rho_-} \|\widehat{\boldsymbol{\theta}}^{\mathsf{O}}_{\mathcal{S}} - \boldsymbol{\theta}^\star_{\mathcal{S}}\|_2^2 \leq \big\langle \nabla_{\mathcal{S}} \mathcal{L}(\boldsymbol{\theta}^\star),\ \boldsymbol{\theta}^\star_{\mathcal{S}} - \widehat{\boldsymbol{\theta}}^{\mathsf{O}}_{\mathcal{S}} \big\rangle. \tag{21}$$

By Hölder's inequality and $|\mathrm{supp}(\boldsymbol{\theta}^\star_{\mathcal{S}} - \widehat{\boldsymbol{\theta}}^{\mathsf{O}}_{\mathcal{S}})| \leq s$, we have

$$\begin{aligned}\big\langle \nabla_{\mathcal{S}} \mathcal{L}(\boldsymbol{\theta}^\star),\ \boldsymbol{\theta}^\star_{\mathcal{S}} - \widehat{\boldsymbol{\theta}}^{\mathsf{O}}_{\mathcal{S}} \big\rangle &\leq \|\nabla_{\mathcal{S}} \mathcal{L}(\boldsymbol{\theta}^\star)\|_\infty \|\boldsymbol{\theta}^\star_{\mathcal{S}} - \widehat{\boldsymbol{\theta}}^{\mathsf{O}}_{\mathcal{S}}\|_1 \\ &\leq \sqrt{s}\, \|\nabla_{\mathcal{S}} \mathcal{L}(\boldsymbol{\theta}^\star)\|_\infty \|\boldsymbol{\theta}^\star_{\mathcal{S}} - \widehat{\boldsymbol{\theta}}^{\mathsf{O}}_{\mathcal{S}}\|_2.\end{aligned}$$

Combining with (21) and dividing by $c_{\rho_-}\|\widehat{\boldsymbol{\theta}}^{\mathsf{O}}_{\mathcal{S}} - \boldsymbol{\theta}^\star_{\mathcal{S}}\|_2$ yields

$$\|\boldsymbol{\theta}^\star_{\mathcal{S}} - \widehat{\boldsymbol{\theta}}^{\mathsf{O}}_{\mathcal{S}}\|_2 \leq \frac{\sqrt{s}}{c_{\rho_-}} \|\nabla_{\mathcal{S}} \mathcal{L}(\boldsymbol{\theta}^\star)\|_\infty.$$

Using Lemma 4, we get

$$\begin{aligned}\|\widehat{\boldsymbol{\theta}}^{\mathsf{O}} - \boldsymbol{\theta}^\star\|_\infty = \|\boldsymbol{\theta}^\star_{\mathcal{S}} - \widehat{\boldsymbol{\theta}}^{\mathsf{O}}_{\mathcal{S}}\|_\infty \leq \|\boldsymbol{\theta}^\star_{\mathcal{S}} - \widehat{\boldsymbol{\theta}}^{\mathsf{O}}_{\mathcal{S}}\|_2 \\ \leq \frac{\sqrt{s}}{c_{\rho_-}} \left(\frac{c_\lambda}{c_{\mathrm{grad}}} \sqrt{\frac{\log d}{N}} + \frac{c_e c_{\rho_+} \beta_u}{\alpha} \frac{\sqrt{\beta_u}}{\beta_l} (A_0 + v) \right).\end{aligned}$$

Since the conditions $\lambda = c_\lambda \sqrt{\frac{\log d}{N}}$ and $\alpha = c_\alpha \sqrt{\frac{N}{\log d}}$ are satisfied in Lemma 2, we have

$$\|\widehat{\boldsymbol{\theta}}^{\mathsf{O}} - \boldsymbol{\theta}^\star\|_\infty \leq c_\infty \sqrt{\frac{s \log d}{N}},$$

where

$$c_\infty = \frac{1}{c_{\rho_-}} \left(\frac{1}{c_{\mathrm{grad}}} + \frac{c_e c_{\rho_+} \beta_u}{c_\lambda c_\alpha} \frac{\sqrt{\beta_u}}{\beta_l} (A_0 + v) \right) c_\lambda.$$

Oracle support recovery. For any $j \in \mathcal{S}$, the triangle inequality gives

$$\big|[\widehat{\boldsymbol{\theta}}^{\mathsf{O}}]_j\big| \geq |\theta^\star_j| - \big|[\widehat{\boldsymbol{\theta}}^{\mathsf{O}} - \boldsymbol{\theta}^\star]_j\big| \geq \min_{k \in \mathcal{S}} |\theta^\star_k| - c_\infty \sqrt{\frac{s \log d}{N}}.$$

If $\min_{j \in \mathcal{S}} |\theta^\star_j| \geq 2c_\infty \sqrt{\frac{s \log d}{N}}$, then we have $\min_{j \in \mathcal{S}} |[\widehat{\boldsymbol{\theta}}^{\mathsf{O}}]_j| > 0$. Since $\mathrm{supp}(\widehat{\boldsymbol{\theta}}^{\mathsf{O}}) \subseteq \mathcal{S}$ by construction, it follows that $\mathrm{supp}(\widehat{\boldsymbol{\theta}}^{\mathsf{O}}) = \mathcal{S}$. □

c) Step 2: Support recovery: We finally prove Theorem 2 by showing that $\widehat{\boldsymbol{\theta}}^{\mathsf{O}}$ is the unique solution of problem (2). By feasibility of the oracle solution (13), we have $\|\widehat{\boldsymbol{\theta}}^{\mathsf{O}}\|_1 \leq R$. On $\mathcal{S}$, since $\widehat{\boldsymbol{\theta}}^{\mathsf{O}}$ solves the oracle problem (13), its restricted KKT conditions yield $\nabla_{\mathcal{S}} \mathcal{L}(\widehat{\boldsymbol{\theta}}^{\mathsf{O}}) + \lambda \boldsymbol{\xi}_{\mathcal{S}} = \mathbf{0}$. We then show that there exists $\boldsymbol{\xi}_{\mathcal{S}^c} \in [-1,1]^{|\mathcal{S}^c|}$ such that $\nabla_{\mathcal{S}^c} \mathcal{L}(\widehat{\boldsymbol{\theta}}^{\mathsf{O}}) + \lambda \boldsymbol{\xi}_{\mathcal{S}^c} = \mathbf{0}$. By the triangle inequality, we have

$$\begin{aligned}\big\|\nabla_{\mathcal{S}^c} \mathcal{L}(\widehat{\boldsymbol{\theta}}^{\mathsf{O}})\big\|_\infty \leq &\big\|\nabla_{\mathcal{S}^c} \mathcal{L}(\widehat{\boldsymbol{\theta}}^{\mathsf{O}}) - \nabla_{\mathcal{S}^c} \mathcal{L}(\boldsymbol{\theta}^\star)\big\|_\infty \\ &+ \big\|\nabla_{\mathcal{S}^c} \mathcal{L}(\boldsymbol{\theta}^\star)\big\|_\infty.\end{aligned} \tag{22}$$

Using Lemma 5 and following similar steps in the proof of Lemma 4, we have

$$\begin{aligned}\big\|\nabla_{\mathcal{S}^c} \mathcal{L}(\widehat{\boldsymbol{\theta}}^{\mathsf{O}})\big\|_\infty &\leq \frac{\lambda}{c_{\mathrm{grad}}} + c_{\rho_+} \beta_u \|\widehat{\boldsymbol{\theta}}^{\mathsf{O}} - \boldsymbol{\theta}^\star\|_\infty \\ &\leq \frac{c_{\rho_+} \beta_u \sqrt{s}}{c_{\rho_-} c_{\mathrm{grad}}} \left(1 + \frac{c_e c_{\rho_+} \beta_u}{4\sqrt{2c_v} v A_0 c_\alpha} \frac{\sqrt{\beta_u}}{\beta_l} (A_0 + v) \right) \lambda + \frac{\lambda}{c_{\mathrm{grad}}}.\end{aligned}$$

Since equation (14) and

$$c_{\mathrm{grad}} \geq \left(1 + \frac{c_{\rho_+} \beta_u \sqrt{s}}{c_{\rho_-}} \right) (1 - c_p)^{-1},$$

when

$$c_p = 1 - \frac{c_{\rho_+}\, \beta_u\, \sqrt{s}}{c_{\rho_-}} \frac{c_e c_{\rho_+} \beta_u c_l \sqrt{\beta_u} (A_0 + v)}{4 c_v v^2 A_0 \beta_l},$$

we have

$$c_{\mathrm{grad}} \geq 1 + \frac{c_{\rho_+} \beta_u \sqrt{s}}{c_{\rho_-}} \left(1 + \frac{c_e\, c_{\rho_+} \beta_u}{4\sqrt{2c_v} v A_0 c_\alpha} \frac{\sqrt{\beta_u}}{\beta_l} (A_0 + v) \right),$$

and hence

$$\big\|\nabla_{\mathcal{S}^c} \mathcal{L}(\widehat{\boldsymbol{\theta}}^{\mathsf{O}})\big\|_\infty \leq \lambda.$$

Therefore, there exists $\boldsymbol{\xi}_{\mathcal{S}^c} \in [-1,1]^{|\mathcal{S}^c|}$ such that $\nabla_{\mathcal{S}^c} \mathcal{L}(\widehat{\boldsymbol{\theta}}^{\mathsf{O}}) + \lambda \boldsymbol{\xi}_{\mathcal{S}^c} = \mathbf{0}$.

In addition, if $\|\widehat{\boldsymbol{\theta}}^{\mathsf{O}}\|_1 < R$, take $\lambda = 0$ and $\lambda\big(\|\widehat{\boldsymbol{\theta}}\|_1 - R\big) = 0$ holds. If $\|\widehat{\boldsymbol{\theta}}^{\mathsf{O}}\|_1 = R$, take $\lambda > 0$ and $\lambda\big(\|\widehat{\boldsymbol{\theta}}\|_1 - R\big) = 0$ still holds. Therefore $(\widehat{\boldsymbol{\theta}}^{\mathsf{O}}, \lambda, \boldsymbol{\xi})$ satisfies all KKT conditions of problem (2), and hence $\widehat{\boldsymbol{\theta}}^{\mathsf{O}}$ is a solution of problem (2). By Assumption 3, the minimizer of the problem is unique, hence $\widehat{\boldsymbol{\theta}} = \widehat{\boldsymbol{\theta}}^{\mathsf{O}}$ is the unique solution. Finally, using Lemma 5 and $\widehat{\boldsymbol{\theta}} = \widehat{\boldsymbol{\theta}}^{\mathsf{O}}$ we conclude $\mathrm{supp}(\widehat{\boldsymbol{\theta}}) = \mathcal{S}$.

C. Technical Lemmas

Lemma 6 (Global RSM and RSC for the empirical Huber loss [20, Lemma 4])**.** *Under Assumption 1, for any $r \geq \max\{R, \|\boldsymbol{\theta}^\star\|_2\}$ and $\alpha \geq c_u r$, with high probability, Assumptions 2 and 3 hold with $\mu = O(\beta_l)$, $\tau_\mu = O\left(\frac{\log d}{N}\right)$, $L = O(\beta_u)$, and $\tau_L = O\left(\frac{\log d}{N}\right)$, where c_u is a positive constant depending on A_0, β_l, β_u, and v.*

Lemma 7 (Local RSM for the empirical Huber loss)**.** *Under Assumption 1, for $\alpha \geq c_u \max\{r, \|\boldsymbol{\theta}^\star\|_2\}$, with high probability, Assumption 4 holds with $\ell = O(\beta_u)$, and $\tau_\ell = O\left(\frac{\log d}{n}\right)$.*

Proof: The proof follows the same restricted-smoothness argument as in Lemma 6; see also Lemma 4 of [20]. For each fixed agent i, the samples $\{(y_j, \boldsymbol{x}_j)\}_{j\in\mathcal{J}_i}$ are i.i.d. and satisfy Assumption 1. Hence, the argument for the empirical Huber loss can be applied with the local sample size n in place of the total sample size N. This yields the same smoothness order $\ell = O(\beta_u)$, while the tolerance term becomes $\tau_\ell = O(\log d/n)$. A union bound over $i = 1, \ldots, m$ gives the result for all local losses with high probability. The detailed steps are therefore omitted. ■

REFERENCES

[1] Q. Wei and Z. Zhao, "High-dimensional constrained Huber regression," in *Proc. IEEE 13th Sensor Array Multichannel Signal Process. Workshop (SAM)*. IEEE, 2024, pp. 1–5.

[2] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to linear regression analysis*. John Wiley & Sons, 2021.

[3] G. A. Seber and A. J. Lee, *Linear regression analysis*. John Wiley & Sons, 2012.

[4] J. Yang, B. Benyamin, B. P. McEvoy, S. Gordon, A. K. Henders, D. R. Nyholt, P. A. Madden, A. C. Heath, N. G. Martin, G. W. Montgomery *et al.*, "Common SNPs explain a large proportion of the heritability for human height," *Nature Genet.*, vol. 42, no. 7, pp. 565–569, 2010.

[5] E. F. Fama and K. R. French, "Common risk factors in the returns on stocks and bonds," *J. Financial Econ.*, vol. 33, no. 1, pp. 3–56, 1993.

[6] G. M. Grossman and A. B. Krueger, "Economic growth and the environment," *Quart. J. Econ.*, vol. 110, no. 2, pp. 353–377, 1995.

[7] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc. B, Statist. Methodol.*, vol. 58, no. 1, pp. 267–288, 1996.

[8] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Roy. Statist. Soc. B, Statist. Methodol.*, vol. 68, no. 1, pp. 49–67, 2006.

[9] H. Zou, "The adaptive lasso and its oracle properties," *J. Amer. Statist. Assoc.*, vol. 101, no. 476, pp. 1418–1429, 2006.

[10] P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Statist.*, vol. 35, no. 1, pp. 73–101, 1964.

[11] P. J. Huber and E. M. Ronchetti, *Robust statistics*. John Wiley & Sons, 2011.

[12] R. Cont, "Empirical properties of asset returns: stylized facts and statistical issues," *Quant. Finance*, vol. 1, no. 2, p. 223, 2001.

[13] J. Fan, W. Wang, and Z. Zhu, "A shrinkage principle for heavy-tailed data: High-dimensional robust low-rank matrix recovery," *Ann. Statist.*, vol. 49, no. 3, p. 1239, 2021.

[14] G. Lugosi and S. Mendelson, "Mean estimation and regression under heavy-tailed distributions: A survey," *Found. Comput. Math.*, vol. 19, no. 5, pp. 1145–1190, 2019.

[15] H. Zou and M. Yuan, "Composite quantile regression and the oracle model selection theory," *Ann. Statist.*, pp. 1108–1126, 2008.

[16] A. Belloni and V. Chernozhukov, "ℓ_1-penalized quantile regression in high-dimensional sparse models," *Ann. Statist.*, vol. 39, no. 1, pp. 82–130, 2011.

[17] J. Fan, Y. Fan, and E. Barut, "Adaptive robust variable selection," *Ann. Statist.*, vol. 42, no. 1, p. 324, 2014.

[18] L. Wang, "The L1 penalized LAD estimator for high dimensional linear regression," *J. Multivariate Anal.*, vol. 120, pp. 135–151, 2013.

[19] S. Lambert-Lacroix and L. Zwald, "Robust regression through the Huber's criterion and adaptive lasso penalty," *Electron. J. Statist.*, vol. 5, pp. 1015–1053, 2011.

[20] J. Fan, Q. Li, and Y. Wang, "Estimation of high dimensional mean regression in the absence of symmetry and light tail assumptions," *J. Roy. Statist. Soc. B, Statist. Methodol.*, vol. 79, no. 1, pp. 247–265, 2017.

[21] C. Adcock, M. Eling, and N. Loperfido, "Skewed distributions in finance and actuarial science: a review," *Eur. J. Finance*, vol. 21, no. 13-14, pp. 1253–1281, 2015.

[22] P. J. Huber, "Robust regression: Asymptotics, conjectures and Monte Carlo," *Ann. Statist.*, vol. 1, no. 5, pp. 799–821, 1973.

[23] V. J. Yohai and R. A. Maronna, "Asymptotic behavior of M-estimators for the linear model," *Ann. Statist.*, vol. 7, no. 2, pp. 258–268, 1979.

[24] S. Portnoy, "Asymptotic behavior of M estimators of p regression parameters when p^2/n is large; ii. normal approximation," *Ann. Statist.*, vol. 13, no. 4, pp. 1403–1417, 1985.

[25] E. Mammen, "Asymptotics with increasing dimension for robust regression with applications to the bootstrap," *Ann. Statist.*, vol. 17, no. 1, pp. 382–400, 1989.

[26] X. He and Q.-M. Shao, "A general bahadur representation of M-estimators and its application to linear regression with nonstochastic designs," *Ann. Statist.*, vol. 24, no. 6, pp. 2608–2630, 1996.

[27] ——, "On parameters of increasing dimensions," *J. Multivariate Anal.*, vol. 73, no. 1, pp. 120–135, 2000.

[28] D. Bertsekas and J. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Athena Scientific, 2015.

[29] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[30] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proc. IEEE*, vol. 106, no. 5, pp. 953–976, 2018.

[31] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *J. Mach. Learn. Res.*, vol. 11, pp. 1663–1707, 2010.

[32] M. Maros, G. Scutari, Y. Sun, and G. Cheng, "Decentralized sparse linear regression via gradient-tracking," *J. Mach. Learn. Res.*, vol. 26, no. 197, pp. 1–56, 2025.

[33] Y. Zhang, J. C. Duchi, and M. J. Wainwright, "Communication-efficient algorithms for statistical optimization," *J. Mach. Learn. Res.*, vol. 14, pp. 3321–3363, 2013.

[34] Y. Zhang, J. Duchi, and M. Wainwright, "Divide and conquer kernel ridge regression," in *Proc. Conf. Learn. Theory (COLT)*. PMLR, 2013, pp. 592–617.

[35] T. Zhao, G. Cheng, and H. Liu, "A partially linear framework for massive heterogeneous data," *Ann. Statist.*, vol. 44, no. 4, pp. 1400–1437, 2016.

[36] J. D. Rosenblatt and B. Nadler, "On the optimality of averaging in distributed statistical learning," *Inf. Inference, J. IMA*, vol. 5, no. 4, pp. 379–404, 2016.

[37] Z. Shang and G. Cheng, "Computational limits of a distributed algorithm for smoothing spline," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 3809–3845, 2017.

[38] X. Chen and M.-G. Xie, "A split-and-conquer approach for analysis of extraordinarily large data," *Statist. Sinica*, vol. 24, pp. 1655–1684, 2014.

[39] J. D. Lee, Q. Liu, Y. Sun, and J. E. Taylor, "Communication-efficient sparse regression," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 115–144, 2017.

[40] H. Battey, J. Fan, H. Liu, J. Lu, and Z. Zhu, "Distributed testing and estimation under sparse high dimensional models," *Ann. Statist.*, vol. 46, no. 3, pp. 1352–1382, 2018.

[41] Y. Bao and W. Xiong, "One-round communication efficient distributed M-estimation," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*. PMLR, 2021, pp. 46–54.

[42] X. Chen, W. Liu, and Y. Zhang, "Quantile regression under memory constraint," *Ann. Statist.*, vol. 47, no. 6, pp. 3244–3273, 2019.

[43] S. Volgushev, S.-K. Chao, and G. Cheng, "Distributed inference for quantile regression processes," *Ann. Statist.*, vol. 47, no. 3, pp. 1634–1662, 2019.

[44] P. Xiao, X. Liu, A. Li, and G. Pan, "Distributed inference for the quantile regression model based on the random weighted bootstrap," *Inf. Sci.*, vol. 680, p. 121172, 2024.

[45] Y. Ji, G. Scutari, Y. Sun, and H. Honnappa, "Distributed sparse regression via penalization," *J. Mach. Learn. Res.*, vol. 24, no. 272, pp. 1–62, 2023.

[46] O. Shamir, N. Srebro, and T. Zhang, "Communication-efficient distributed optimization using an approximate Newton-type method," in *Proc. Int. Conf. Mach. Learn. (ICML)*. PMLR, 2014, pp. 1000–1008.

[47] J. Wang, M. Kolar, N. Srebro, and T. Zhang, "Efficient distributed learning with sparsity," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*. PMLR, 2017, pp. 3636–3645.

[48] M. I. Jordan, J. D. Lee, and Y. Yang, "Communication-efficient distributed statistical inference," *J. Amer. Statist. Assoc.*, vol. 114, no. 526, pp. 668–681, 2019.

[49] J. Fan, Y. Guo, and K. Wang, "Communication-efficient accurate statistical estimation," *J. Amer. Statist. Assoc.*, vol. 118, no. 542, pp. 1000–1010, 2023.

[50] A. Hu, Y. Jiao, Y. Liu, Y. Shi, and Y. Wu, "Distributed quantile regression for massive heterogeneous data," *Neurocomputing*, vol. 448, pp. 249–262, 2021.

[51] X. Chen, W. Liu, X. Mao, and Z. Yang, "Distributed high-dimensional regression under a quantile loss function," *J. Mach. Learn. Res.*, vol. 21, no. 182, pp. 1–43, 2020.

[52] K. M. Tan, H. Battey, and W.-X. Zhou, "Communication-constrained distributed quantile regression with optimal statistical guarantees," *J. Mach. Learn. Res.*, vol. 23, no. 272, pp. 1–61, 2022.

[53] R. Mirzaeifard, V. C. Gogineni, N. K. Venkategowda, and S. Werner, "Distributed quantile regression with non-convex sparse penalties," in *Proc. IEEE Statist. Signal Process. Workshop (SSP)*. IEEE, 2023, pp. 250–254.

[54] C. Wang and Z. Shen, "Distributed high-dimensional quantile regression: Estimation efficiency and support recovery," *arXiv preprint arXiv:2405.07552*, 2024.

[55] J. Luo, Q. Sun, and W.-X. Zhou, "Distributed adaptive Huber regression," *Comput. Statist. Data Anal.*, vol. 169, p. 107419, 2022.

[56] Y. Pan, K. Xu, S. Wei, X. Wang, and Z. Liu, "Efficient distributed optimization for large-scale high-dimensional sparse penalized huber regression," *Commun. Statist. Simul. Comput.*, vol. 53, no. 7, pp. 3106–3125, 2024.

[57] B. He, L.-Z. Liao, D. Han, and H. Yang, "A new inexact alternating directions method for monotone variational inequalities," *Math. Program.*, vol. 92, pp. 103–118, 2002.

[58] J. B. Predd, S. R. Kulkarni, and H. V. Poor, "A collaborative training algorithm for distributed learning," *IEEE Trans. Inf. Theory*, vol. 55, no. 4, pp. 1856–1871, 2009.

[59] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5262–5276, 2010.

[60] T.-H. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus ADMM," *IEEE Trans. Signal Process.*, vol. 63, no. 2, pp. 482–497, 2014.

[61] Y. Liu, C. Li, and Z. Zhang, "Diffusion sparse least-mean squares over networks," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4480–4485, 2012.

[62] P. Di Lorenzo and A. H. Sayed, "Sparse distributed learning based on diffusion adaptation," *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1419–1433, 2012.

[63] Y. Ji, G. Scutari, Y. Sun, and H. Honnappa, "Distributed (ATC) gradient descent for high dimension sparse regression," *IEEE Trans. Inf. Theory*, 2023.

[64] M. Maros and G. Scutari, "Acceleration in distributed sparse regression," in *Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 36 832–36 844.

[65] H. Wang and C. Li, "Distributed quantile regression over sensor networks," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 2, pp. 338–348, 2018.

[66] Y. Wang and H. Lian, "On linear convergence of ADMM for decentralized quantile regression," *IEEE Trans. Signal Process.*, vol. 71, pp. 3945–3955, 2023.

[67] L. Shen, Y. Chao, and X. Ma, "Distributed quantile regression in decentralized optimization," *Inf. Sci.*, vol. 643, p. 119259, 2023.

[68] W. Liu, X. Mao, and X. Zhang, "Fast and robust sparsity learning over networks: A decentralized surrogate median regression approach," *IEEE Trans. Signal Process.*, vol. 70, pp. 797–809, 2022.

[69] J. Shi, Y. Wang, Z. Zhu, and H. Lian, "Decentralized learning of quantile regression: a smoothing approach," *J. Comput. Graph. Statist.*, pp. 1–11, 2025.

[70] A. Agarwal, S. Negahban, and M. J. Wainwright, "Fast global convergence rates of gradient methods for high-dimensional statistical recovery," *Ann. Statist.*, vol. 40, no. 5, pp. 2452–2482, 2012.

[71] P. Di Lorenzo and G. Scutari, "NEXT: In-network nonconvex optimization," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 2, pp. 120–136, 2016.

[72] A. Nedic, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM J. Optim.*, vol. 27, no. 4, pp. 2597–2633, 2017.

[73] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Convergence of asynchronous distributed gradient methods over stochastic networks," *IEEE Trans. Autom. Control*, vol. 63, no. 2, pp. 434–448, 2017.

[74] Y. Sun, G. Scutari, and A. Daneshmand, "Distributed optimization based on gradient tracking revisited: Enhancing convergence rate via surrogation," *SIAM J. Optim.*, vol. 32, no. 2, pp. 354–385, 2022.

[75] S. N. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu, "A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers," *Statist. Sci.*, vol. 27, no. 4, pp. 538–557, 2012.

[76] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. 4th Int. Symp. Inf. Process. Sensor Netw. (IPSN)*. IEEE, 2005, pp. 63–70.

[77] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, 2009.

[78] M. J. Wainwright, "Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (lasso)," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2183–2202, 2009.

[79] G. Raskutti, M. J. Wainwright, and B. Yu, "Minimax rates of estimation for high-dimensional linear regression over ell_q-balls," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6976–6994, 2011.

[80] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the ℓ_1-ball for learning in high dimensions," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 272–279.

[81] DepMap, Broad, "DepMap Public 25Q2," https://depmap.org, 2025, dataset.

[82] D. P. Nusinow, J. Szpyt, M. Ghandi, C. M. Rose, I. McDonald, E. Robert, M. Kalocsay, J. Jané-Valbuena, E. Gelfand, D. K. Schweppe, M. Jedrychowski, J. Golji, D. A. Porter, T. Rejtar, Y. K. Wang, G. V. Kryukov, F. Stegmeier, B. K. Erickson, L. A. Garraway, W. R. Sellers, and S. P. Gygi, "Quantitative proteomics of the cancer cell line encyclopedia," *Cell*, vol. 180, no. 2, pp. 387–402.e16, 2020.

[83] A. C. Harvey, "Estimating regression models with multiplicative heteroscedasticity," *Econometrica*, pp. 461–465, 1976.

[84] C. M. Jarque and A. K. Bera, "Efficient tests for normality, homoscedasticity and serial independence of regression residuals," *Econ. Lett.*, vol. 6, no. 3, pp. 255–259, 1980.

[85] G. J. Katuwal, N. D. Cahill, S. A. Baum, and A. M. Michael, "The predictive power of structural mri in autism diagnosis," in *Proc. 37th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*. IEEE, 2015, pp. 4270–4273.

[86] Z. Wang, H. Liu, and T. Zhang, "Optimal computational and statistical rates of convergence for sparse nonconvex learning problems," *Ann. Statist.*, vol. 42, no. 6, p. 2164, 2014.

[87] P.-L. Loh and M. J. Wainwright, "Regularized M-estimators with nonconvexity: Statistical and algorithmic theory for local optima," *J. Mach. Learn. Res.*, vol. 16, pp. 559–616, 2015.