Estimation and Inference on Distributed High-Dimensional Quantile Regression: Double-Smoothing and Debiasing

Caixing Wang, Ziliang Shen, *Student Member, IEEE,* Shaoli Wang, Xingdong Feng.

***Abstract*—In this paper, we focus on distributed estimation and inference for high-dimensional sparse linear quantile regression. Quantile regression is a popular alternative tool to the least squares regression for robustness against outliers and data heterogeneity. However, the non-smoothness of the check loss function poses big challenges to both computation and theory in the distributed setting. To tackle these problems, we transform the original quantile regression into a least-squares optimization. By applying a double-smoothing approach, we extend a previous Newton-type distributed approach without the restrictive independent assumption between the error term and covariates. An efficient algorithm is developed, which enjoys high computation and communication efficiency. Theoretically, the proposed distributed estimator achieves a near-oracle convergence rate and high support recovery accuracy after a constant number of iterations. Furthermore, we employ a novel communication-efficient debiasing technique based on the distributed estimator to conduct statistical inference, including hypothesis testing and interval estimation. Non-asymptotic Berry-Esseen bounds and asymptotic normality of the debiased estimator are provided to ensure the validity of the inference. Extensive experiments on synthetic examples and a real data application further demonstrate the effectiveness of the proposed method.**

***Index Terms*—Distributed estimation, distributed inference, high-dimensional quantile regression, data heterogeneity, double-smoothing, debiased method.**

I. Introduction

With the development of modern technology, the proliferation of massive data has garnered significant attention from researchers and practitioners [1], [2]. For example, financial institutions leverage big data to scrutinize customer preferences and fine-tune their marketing approaches, while manufacturers and transportation departments lean heavily on extensive datasets to streamline supply chain management and enhance delivery route optimization. However, these large-scale datasets are usually distributed across different machines due to storage and privacy concerns, making the direct application of existing statistical methods infeasible. On the other hand, another challenge arises from the high dimensionality of modern data. In the literature, a sparse assumption is often adopted [3]–[5], and support recovery is an essential problem for high-dimensional analysis. Despite its importance in practice, the support recovery in a distributed system is underexplored in theory, compared to the well-studied statistical estimation of the interested parameters [6]–[8].

Since the seminar work of [9], quantile regression has gained increasing attention across various fields, including economics, biomedicine, and environmental studies. Compared to the least squares regression that only estimates the conditional means, quantile regression models the entire conditional quantiles of the response, making it more robust against outliers in the response measurements [10], [11]. Although quantile regression can better handle data heterogeneity, computational challenges arise when both sample size and dimension are large due to the non-smooth check loss function [12], [13]. Consequently, it is natural to consider a distributed estimation procedure to address the scalability concerns.

In this paper, we focus on distributed estimation and inference for high-dimensional sparse linear quantile regression, where the number of covariates p can be much larger than the sample size N and the true parameter $\boldsymbol{\beta}^*$ is sparse. We aim to bridge the theoretical and practical gap by addressing several fundamental questions regarding distributed high-dimensional linear quantile regression. Firstly, what is the statistical limit of estimation in the presence of distributed data? And how does this limit depend on the number of local machines in the distributed system? Secondly, can distributed high-dimensional sparse linear quantile regression achieve the same convergence rate of the parameters and support recovery rate as those in a single machine setting? Thirdly, how can the high-dimensional inference problem be addressed for distributed data with high communication efficiency? Finally, can the proposed method be applied to various practical settings, such as homoscedastic and heteroscedastic data structures?

We address the aforementioned theoretical and practical questions by designing some distributed high-dimensional sparse linear quantile regression algorithms via a double-smoothing transformation and debiasing technique. To our limited knowledge, our work is one of the pioneering works in studying distributed high-dimensional sparse linear quantile regression with the least practical constraint and solid theoretical guarantees involving estimation efficiency, support recovery, as well as valid statistical inference. The specific contributions can be concluded as follows.

Caixing Wang is with the School of Statistics and Data Science, Southeast University, Nanjing, 211189, China (e-mail: wangcaixing96@gmail.com). Ziliang Shen is with the School of Statistics and Data Science, Shanghai University of Finance and Economics, Shanghai, 200433, China (e-mail: ziliangshen@stu.sufe.edu.cn). Shaoli Wang is with the School of Statistics and Data Science, Shanghai University of Finance and Economics, Shanghai, 200433, China (e-mail: swang@shufe.edu.cn). Xingdong Feng Xingdong Feng is with School of Statistics and Data Science and Institute of Big Data Research, Shanghai University of Finance and Economics, Shanghai, 200433, China(e-mail: feng.xingdong@mail.shufe.edu.cn).

1. For distributed estimation, our paper extends the idea of [14] resulting in a novel method for **D**istributed **H**igh-dimensional **S**parse **Q**uantile **R**egression (*DHSQR*) without the stringent assumption that the error term is independent of the covariates. Specifically, we start by transforming the covariate and the response, which recasts the quantile regression into a least squares framework. Next, we introduce an iterative distributed algorithm based on an approximate Newton method by using a double-smoothing approach applied to the global and local loss functions, respectively. In the distributed system, with p-dimensional covariates, the local machines only need to broadcast the p-dimensional gradient vectors (instead of the $p \times p$ Hessian matrix). This optimization problem can be efficiently addressed on the central machine due to its simplified least squares formulation.
2. For distributed inference, we transform the ℓ_1-penalized quantile regression into a least squares problem with a Lasso penalty, making it natural to apply the debiasing technique for statistical inference. Unlike previous works in linear regression [15]–[17] and quantile regression [18], [19], we propose a communication-efficient distributed debiasing estimator (*Debiased DHSQR*). This method uses a local CLIME estimator to approximate the inverse of the population Hessian matrix, and the local machines only need to broadcast the local gradients to the central machine. Based on the debiased DHSQR estimator, we construct confidence intervals and conduct hypothesis testing to ensure valid statistical inference.
3. Theoretically, we not only establish the convergence rate of our DHSQR estimator in the ℓ_2-norm (Theorems 1 and 2), but also characterize the *beta-min* condition for the exact support recovery (Theorems 3 and 4) which is novel for distributed high-dimensional sparse estimation [20]. After a constant number of iterations, the convergence rate and the *beta-min* condition of the distributed estimator align with the classical theoretical results derived for a single machine setup [4], [21]. Furthermore, we give a Bahadur representation for the debiased DHSQR estimator and provide the non-asymptotic Berry-Esseen bounds and asymptotic normality of the estimator (Theorems 5 and 6). These results ensure that the coverage probability of the confidence interval is asymptotically close to $1-\alpha$ and that the Type I error is bounded within the significance level α (Theorems 7 and 8).
4. Another contribution of this work is the comprehensive studies on the validity and effectiveness of the proposed algorithm in various synthetic and real-life examples, which further support the theoretical findings in this paper.

We remark that a shorter version of this paper has appeared as the spotlight in ICML 2024 [22]. In this extended version, we have provided a debiased approach for distributed inference (Section IV), as well as establishing the Bahadur representation, Berry-Esseen bounds, and asymptotic normality for the debiased estimator (Section V). We have also refined the theoretical results for the DHSQR estimator (Section III). Additional simulation studies of distributed inference construction have been conducted (Section VII-G).

A. Related Work

Distributed methods. Significant efforts have been dedicated to the development of distributed statistical learning methods, broadly categorized into two main streams. The first class is known as the divide-and-conquer (DC) methods [6], [23]–[25]. These one-shot methods usually compute the relevant estimates based on local samples in the first step and then send these local estimates to a central machine, where the final estimate is obtained by simply averaging the local estimates. These methods offer computational efficiency with just one round of communication, but they have the theoretical constraint on the number of local machines to guarantee the global optimal rate [2], [26]. The second class comprises multi-round distributed methods designed to improve estimation efficiency and relax restrictions on the number of local machines [7], [27], [28]. [8] and [29] proposed a communication-efficient surrogate likelihood (CSL) framework that can be applied to low-dimensional estimation, high-dimensional regularized estimation, and Bayesian inference. Notably, the CSL method eliminates the need to transfer local Hessian matrices to the central machine, resulting in significantly reduced communication costs. It is worth noting that most of the aforementioned methods only focus on homogeneous data, which can be less practical in the context of big data analysis.

Distributed linear quantile regression. In the existing literature, distributed linear quantile regression has been widely investigated using the traditional divide-and-conquer method [30]–[32]. However, when dealing with high-dimensional settings, where sparsity assumptions are commonly applied, the DC estimator is no longer sparse due to de-biasing and averaging processes, resulting in poor support recovery [14], [18]. In addition, their methods require a condition on the number of distributed machines to ensure the global convergence rate. To alleviate the restriction on the number of machines, [14] transformed the check loss to the square loss via a kernel smoothing approach and proposed a Newton-type distributed estimator. The theoretical results offered insights into estimation errors and support recovery. Nevertheless, their method and theory require the error term to be independent of the covariates, which is not very common and hard to verify in practice. Inspired by the ideas in [8] and [33], [34] studied a distributed convolution-type smoothing quantile regression whose loss function is twice continuously differentiable in both low-dimensional and high-dimensional regimes. Note that most of the previous works mainly focus on the convergence rate of their respective estimator, while we further establish the distributed support recovery and inference theory.

High-dimensional quantile regression statistical inference. In high-dimensional sparse quantile regression models, significant attention has been devoted not only to parameter estimation but also to inference properties. However, inference remains particularly challenging due to the bias introduced by traditional estimation methods. To mitigate the shrinkage bias induced by ℓ_1-regularization, various concave penalties

have been proposed, such as the SCAD penalty [35] and the MCP penalty [36]. These methods, however, rely on oracle properties, meaning valid inferences can only be made on the information of the non-sparse coefficients. To address these challenges, several studies have explored innovative inference methods. [37], [38], and [39] applied the Neyman orthogonalization method to project redundant variables onto the variables of interest, thereby improving inference accuracy. [40] studied tuning strategies and multi-quantile aggregation, and [41] discuss extensions that integrate sparsity with differential privacy. A major advancement in high-dimensional inference was the development of debiased or desparsified Lasso estimators. [15] first introduced this approach for linear models, which was later extended by [16] and [17]. They proposed debiasing techniques based on the inverse covariance matrix of predictors. Inference in high-dimensional sparse quantile regression has also been extensively studied. [30] investigated inference challenges in high-dimensional composite quantile regression, while [18] developed the debiased ℓ_1 quantile regression estimator and established its asymptotic normality. [19] extended debiasing techniques to convolution-smoothing quantile regression, providing new insights into its asymptotic properties. Despite these methodological advancements, most existing approaches remain limited to non-distributed data. For the distributed setting, [42] proposed a divide-and-conquer approach for high-dimensional quantile regression using debiased estimators. However, their method requires the number of local machines can not diverge too fast and only considers the estimation property. In this paper, we extend the debiasing technique to the proposed DHSQR estimator and provide a communication and computation efficient algorithm for valid inference.

B. Paper Organization and Notations

The rest of the paper is organized as follows. In Section II, we introduce the proposed distributed estimation method for high-dimensional sparse quantile regression, including the Newton-type transformation and the double-smoothing shifted loss function. Section III provides the theoretical guarantees for the distributed estimation, including the convergence rate and support recovery accuracy. In Section IV, we develop a debiasing method for valid distributed statistical inference and describe the distributed approach using double-smoothing and debiasing. Section V establishes the statistical guarantees for the distributed inference, including the Bahadur representation, Berry-Esseen bounds, and asymptotic normality. Section VII presents simulation studies to assess the performance of the proposed method. Finally, Section IX concludes the paper and discusses potential future research directions. Thedetailed proofs of the main theorems are provided in the supplementary material.

For two sequences $\{a_n\}$ and $\{b_n\}$, we denote $a_n \lesssim b_n$ if $a_n \leq Cb_n$, where C is a constant. And $a_n \asymp b_n$ if and only if $a_n \lesssim b_n$ and $b_n \lesssim a_n$. For a vector $\boldsymbol{u} = (u_1, \ldots, u_p)^{\mathrm{T}}$, we define $\mathrm{supp}(\boldsymbol{u}) = \{j : u_j \neq 0\}$. We use $|\cdot|_q$ to denote the ℓ_q-norm in $\mathbb{R}_p$: $|\boldsymbol{u}|_q = (\sum_{i=1}^p |u_i|^q)^{1/q}$, for $1 \leq q < \infty$ and $|\boldsymbol{u}|_\infty = \max_{1\leq i\leq p} |u_i|$. For $S \subseteq \{1, \ldots, p\}$ with length $|S|$, let $\boldsymbol{u}_S = (u_i, i \in S) \subseteq \mathbb{R}^{|S|}$. For a matrix $\boldsymbol{A} = (a_{ij}) \in \mathbb{R}^{p\times q}$, define $|\mathbf{A}|_1 = \sum_{1\leq i\leq p}\sum_{1\leq j\leq q}|a_{ij}|, |\boldsymbol{A}|_\infty = \max_{1\leq i\leq p, 1\leq j\leq q}|a_{ij}|$, $\|\boldsymbol{A}\|_\infty = \max_{1\leq i\leq p}\sum_{1\leq j\leq q}|a_{ij}|$, and $\|\boldsymbol{A}\|_{op} = \max_{|\boldsymbol{v}|_2=1}|\boldsymbol{A}\boldsymbol{v}|_2$. For two subsets $S_1 \in \{1, \ldots, p\}$ and $S_2 \in \{1, \ldots, q\}$, we define the submatrix $\boldsymbol{A}_{S_1\times S_2} = (a_{ij}, i \in S_1, j \in S_2)$, $\Lambda_{\max}(\boldsymbol{A})$ and $\Lambda_{\min}(\boldsymbol{A})$ to be the largest and smallest eigenvalues of $\boldsymbol{A}$, respectively. For two positive definite matrixes $\boldsymbol{A}$ and $\boldsymbol{B}$, $\boldsymbol{A} \succ \boldsymbol{B}$ means that $\boldsymbol{A} - \boldsymbol{B}$ is a positive definite matrix; and $\boldsymbol{A} \prec \boldsymbol{B}$ means that $\boldsymbol{B} - \boldsymbol{A}$ is a positive definite matrix. Here we denote $\boldsymbol{I}$ as the identity matrix, $\boldsymbol{e}_i$ denotes the i-th column vector of the identity matrix, and $I(\cdot)$ denotes the indicator function. For some $r \geq 0$, the unit sphere and the ℓ_1-norm ball in $\mathbb{R}^p$ are defined as $\mathbb{S}^{p-1} = \{\boldsymbol{u} \in \mathbb{R}^p : |\boldsymbol{u}|_2 = 1\}$ and $\mathbb{B}_1(r) = \{\boldsymbol{u} \in \mathbb{R}^p : |\boldsymbol{u}|_1 \leq r\}$ respectively. In this paper, we use C to denote a universal constant that may vary from line to line.

II. Methodology for Distributed Estimation

In this section, we introduce the proposed distributed estimator. Inspired by the Newton-Raphson iteration, we construct a kernel-based estimator that establishes a connection between quantile regression and ordinary least squares regression in a single machine setting. Based on this estimator, we design a distributed algorithm to minimize a double-smoothing shifted loss function.

A. The Linear Quantile Model

For a given quantile level $\tau \in (0, 1)$, we consider to construct the conditional τ-th quantile function $Q_\tau(Y|\boldsymbol{X})$ with a linear model:

$$Q_\tau(Y|\boldsymbol{X}) = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{\beta}^*(\tau) = \sum_{j=1}^p x_j\beta_j^*(\tau),$$

where $Y \in \mathbb{R}$ is a univariate response and $\boldsymbol{X} = (x_1, x_2, \ldots, x_p)^{\mathrm{T}} \in \mathbb{R}^p$ is p-dimensional covariate vector with $x_1 \equiv 1$. Here, $\boldsymbol{\beta}^* = \boldsymbol{\beta}^*(\tau) = (\beta_1^*(\tau), \beta_2^*(\tau), \ldots, \beta_p^*(\tau))$ is the true coefficient vector that can be obtained by minimizing the following stochastic optimization problem,

$$\mathcal{Q}(\boldsymbol{\beta}) = \mathbb{E}\left[\rho_\tau(Y - \boldsymbol{X}^{\mathrm{T}}\boldsymbol{\beta})\right], \tag{1}$$

where $\rho_\tau(u) = u\{\tau - I(u \leq 0)\}$ is the standard check loss function [9] with $I(\cdot)$ is the indicator function.

B. Newton-type Transformation on Quantile Regression

To solve the stochastic optimization problem in (1), we use the Newton-Raphson method. Given an initial estimator $\boldsymbol{\beta}_0$, the population form of the Newton-Raphson iteration is

$$\boldsymbol{\beta}_1 = \boldsymbol{\beta}_0 - \boldsymbol{H}^{-1}(\boldsymbol{\beta}_0)\mathbb{E}\left[\partial\mathcal{Q}(\boldsymbol{\beta}_0)\right], \tag{2}$$

where $\partial\mathcal{Q}(\boldsymbol{\beta}) = \boldsymbol{X}\{I(Y - \boldsymbol{X}^{\mathrm{T}}\boldsymbol{\beta} \leq 0) - \tau\}$ is the subgradient of the check loss function with respect to $\boldsymbol{\beta}$, and $\boldsymbol{H}(\boldsymbol{\beta}) = \partial\mathbb{E}[\partial\mathcal{Q}(\boldsymbol{\beta})]/\partial\boldsymbol{\beta} = \mathbb{E}[\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}f_{\varepsilon|\boldsymbol{X}}(\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{\beta} - \boldsymbol{\beta}^*))]$ denotes the population Hessian matrix of $\mathbb{E}[\mathcal{Q}(\boldsymbol{\beta})]$. Here, we denote the error term as $\varepsilon = Y - \boldsymbol{X}^{\mathrm{T}}\boldsymbol{\beta}^*$, and $f_{\varepsilon|\boldsymbol{X}}(\cdot)$ is the conditional density of ε given $\boldsymbol{X}$.

When the initial estimator $\boldsymbol{\beta}_0$ is close to the true parameter $\boldsymbol{\beta}^*$, $\boldsymbol{H}(\boldsymbol{\beta}_0)$ will be close to $\boldsymbol{H}(\boldsymbol{\beta}^*) = \mathbb{E}[\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} f_{\varepsilon|\boldsymbol{X}}(0)]$. Motivated by this, we further approximate $\boldsymbol{H}(\boldsymbol{\beta}^*)$ with $\boldsymbol{D}_h(\boldsymbol{\beta}_0)$ such that

$$\boldsymbol{H}(\boldsymbol{\beta}_0) \approx \boldsymbol{H}(\boldsymbol{\beta}^*) \approx \boldsymbol{D}_h(\boldsymbol{\beta}_0) = \mathbb{E}(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} K_h(e_0)), \quad (3)$$

where $e_0 = Y - \boldsymbol{X}^{\mathrm{T}}\boldsymbol{\beta}_0$, and $K_h(\cdot) = K(\cdot/h)/h$ with $K(\cdot)$ denoting a symmetrix and non-negative kernel function, $h \to 0$ is the bandwidth. For simplicity of notation, we denote a pseudo covariate as $\widetilde{\boldsymbol{X}}_h = \sqrt{K_h(e_0)}\boldsymbol{X}$. Hence, we can rewrite $\boldsymbol{D}_h(\boldsymbol{\beta}_0) = \mathbb{E}(\widetilde{\boldsymbol{X}}_h\widetilde{\boldsymbol{X}}_h^{\mathrm{T}})$, which is the covariance matrix of $\widetilde{\boldsymbol{X}}_h$. Replacing $\boldsymbol{H}(\boldsymbol{\beta}_0)$ with $\boldsymbol{D}_h(\boldsymbol{\beta}_0)$ in (2) leads to the following iteration,

$$\boldsymbol{\beta}_1 = \boldsymbol{\beta}_0 - \boldsymbol{D}_h^{-1}(\boldsymbol{\beta}_0)\mathbb{E}[\partial\mathcal{Q}(\boldsymbol{\beta}_0)] \quad (4)$$

This iteration together with the Taylor expansion of $\mathbb{E}[\partial\mathcal{Q}(\boldsymbol{\beta}_0)]$ at $\boldsymbol{\beta}^*$,

$$\mathbb{E}[\partial\mathcal{Q}(\boldsymbol{\beta}_0)] = \boldsymbol{H}(\boldsymbol{\beta}^*)(\boldsymbol{\beta}_0 - \boldsymbol{\beta}^*) + \mathcal{O}(|\boldsymbol{\beta}_0 - \boldsymbol{\beta}^*|_2^2),$$

guarantee an improved convergence rate of $\boldsymbol{\beta}_1$ in ℓ_2-norm,

$$\begin{aligned}|\boldsymbol{\beta}_1 - \boldsymbol{\beta}^*|_2 = \big|\boldsymbol{\beta}_0 - \boldsymbol{D}_h^{-1}(\boldsymbol{\beta}_{0,h})(\boldsymbol{H}(\boldsymbol{\beta}^*)(\boldsymbol{\beta}_0 - \boldsymbol{\beta}^*) \\ + \mathcal{O}(|\boldsymbol{\beta}_0 - \boldsymbol{\beta}^*|_2^2)) - \boldsymbol{\beta}^*\big|_2 = \mathcal{O}\left(|\boldsymbol{\beta}_0 - \boldsymbol{\beta}^*|_2^2\right).\end{aligned}$$

Consequently, if we have a consistent estimator $\boldsymbol{\beta}_0$, we can refine it by (4).

Now we show how to transform the Newton-Raphson iteration into a least squares problem. According to (4), we have

$$\begin{aligned}\boldsymbol{\beta}_1 &= \boldsymbol{D}_h^{-1}(\boldsymbol{\beta}_0)\Big\{\boldsymbol{D}_h(\boldsymbol{\beta}_0)\boldsymbol{\beta}_0 - \mathbb{E}\big[\mathbf{X}\left(I(e_0 \le 0) - \tau\right)\big]\Big\} \\ &= \boldsymbol{D}_h^{-1}(\boldsymbol{\beta}_0)\mathbb{E}\Big\{\widetilde{\mathbf{X}}_h\Big[\widetilde{\mathbf{X}}_h^{\mathrm{T}}\boldsymbol{\beta}_0 - \frac{1}{\sqrt{K_h(e_0)}}\big(I(e_0 \le 0) - \tau\big)\Big]\Big\}.\end{aligned}$$

If we further define a new pseudo response as $\widetilde{Y}_h = \widetilde{\boldsymbol{X}}_h^{\mathrm{T}}\boldsymbol{\beta}_0 - \frac{1}{\sqrt{K_h(e_0)}}\left(I(e_0 \le 0) - \tau\right)$, then $\boldsymbol{\beta}_1 = \boldsymbol{D}_h^{-1}(\boldsymbol{\beta}_0)\mathbb{E}(\widetilde{\boldsymbol{X}}_h\widetilde{Y}_h) = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p}\mathbb{E}(\widetilde{Y}_h - \widetilde{\boldsymbol{X}}_h^{\mathrm{T}}\boldsymbol{\beta})^2$ is the least squares regression coefficient of $\widetilde{Y}_h$ on $\widetilde{\boldsymbol{X}}_h$. To further encourage the sparsity of the coefficient vector, we consider the following ℓ_1-penalized least squares problem,

$$\boldsymbol{\beta}_{1,\ell_1} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\operatorname{argmin}}\frac{1}{2}\mathbb{E}\left(\widetilde{Y}_h - \widetilde{\boldsymbol{X}}_h^{\mathrm{T}}\boldsymbol{\beta}\right)^2 + \lambda|\boldsymbol{\beta}|_1, \quad (5)$$

where $\lambda > 0$ is the regularization parameter. We can also consider other forms of penalties, including the smoothly clipped absolute deviations penalty (SCAD, [35]) and the minimax concave penalty (MCP, [36]). We refer the reader to [5] for comprehensive reviews on recent developments.

Now we are ready to define the empirical form of $\boldsymbol{\beta}_1$ in a single machine. Let $\widehat{\boldsymbol{\beta}}_0$ be an initial estimate based on random samples $\mathcal{Z}^N = \{(\boldsymbol{X}_i, Y_i)\}_{i=1}^N$, then we can transform the origin covariates and responses by

$$\begin{aligned}\widetilde{\boldsymbol{X}}_{i,h} &= \sqrt{K_h(\widehat{e}_{0,i})}\boldsymbol{X}_i \\ \widetilde{Y}_{i,h} &= \widetilde{\boldsymbol{X}}_{i,h}^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_0 - \frac{1}{\sqrt{K_h(\widehat{e}_{0,i})}}\left(I(\widehat{e}_{0,i} \le 0) - \tau\right),\end{aligned} \quad (6)$$

for $i = 1, \ldots, N$, where $\widehat{e}_{0,i} = Y_i - \boldsymbol{X}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_0$. Thus, we estimate $\boldsymbol{\beta}^*$ by the empirical version of (5):

$$\widehat{\boldsymbol{\beta}}_{pool} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\operatorname{argmin}}\frac{1}{2N}\sum_{i=1}^N(\widetilde{Y}_{i,h} - \widetilde{\boldsymbol{X}}_{i,h}^{\mathrm{T}}\boldsymbol{\beta})^2 + \lambda|\boldsymbol{\beta}|_1. \quad (7)$$

Given a consistent initial estimator $\widehat{\boldsymbol{\beta}}_0$, we introduce a reasonable estimator $\widehat{\boldsymbol{\beta}}_{pool}$ by pooling all data into a single machine. Compared to the standard ℓ_1-penalized quantile regression, the least squares problem plus a Lasso penalty is much more computationally efficient. Moreover, the smoothness and strong convexity of the quadratic loss function facilitate the development of a distributed estimator in the next section.

Remark 1. *Inspired by work in [14], we remove the stringent restriction that the error term ϵ should be independent of the covariate $\boldsymbol{X}$. Therefore, we cannot simply take the conditional density $f_{\varepsilon|\boldsymbol{X}}(0)$ from $\boldsymbol{H}(\boldsymbol{\beta}^*)$ in* (3)*. To consider such a dependence, we further define a pseudo covariate $\widetilde{\boldsymbol{X}}_h$ that can be regarded as a density-scaled surrogate of $\boldsymbol{X}$. As indicated in Remark 2 of [14], the extension to the dependent case seems relatively straightforward in a single machine setting. However, it is nontrivial for distributed implementation in both methodology and theory due to the curse of dimensionality. In this paper, we succeed in solving it by leveraging a double-smoothing approach (see details in the next section).*

C. Distributed Estimation with a Double-smoothing Shifted Loss Function

Suppose the random samples $\mathcal{Z}^N = \{(\boldsymbol{X}_i, Y_i)\}_{i=1}^N$ are randomly stored in m machines $\mathcal{M}_1, \ldots, \mathcal{M}_m$ with the equal local sample size that $n = N/m$. Without loss of generality, we assume that $\mathcal{M}_1$ is the central machine and denote those samples in the k-th machine as $\{(\boldsymbol{X}_i, Y_i)\}_{i\in\mathcal{M}_k}$ with $|\mathcal{M}_k| = n$, for $k = 1, \ldots, m$. Based on the initial estimator $\widehat{\boldsymbol{\beta}}_0$, every local machines can compute the transformed samples as $\{(\widetilde{\boldsymbol{X}}_{i,h}, \widetilde{Y}_{i,h})\}_{i\in\mathcal{M}_k}$ according to (6). For ease of notation, let

$$\begin{aligned}\widehat{\boldsymbol{D}}_{k,h} &= \frac{1}{n}\sum_{i\in\mathcal{M}_k}\widetilde{\boldsymbol{X}}_{i,h}\widetilde{\boldsymbol{X}}_{i,h}^{\mathrm{T}}, \\ \widehat{\boldsymbol{D}}_h &= \frac{1}{m}\sum_{k=1}^m\widehat{\boldsymbol{D}}_{k,h} = \frac{1}{N}\sum_{i=1}^N\widetilde{\boldsymbol{X}}_{i,h}\widetilde{\boldsymbol{X}}_{i,h}^{\mathrm{T}},\end{aligned} \quad (8)$$

as the k-th local sample covariance matrix and total sample covariance matrix, respectively. It is worth noting that our algorithm does not need the local machine to explicitly calculate and broadcast $\widehat{\boldsymbol{D}}_{k,h}$ for $k \neq 1$ (see in Algorithm 1). We further define the pseudo local and global loss functions, respectively, as

$$\begin{aligned}\mathcal{L}_k(\boldsymbol{\beta}) &= \frac{1}{2n}\sum_{i\in\mathcal{M}_k}(\widetilde{Y}_{i,h} - \widetilde{\boldsymbol{X}}_{i,h}^{\mathrm{T}}\boldsymbol{\beta})^2, \\ \mathcal{L}_N(\boldsymbol{\beta}) &= \frac{1}{2N}\sum_{i=1}^N(\widetilde{Y}_{i,h} - \widetilde{\boldsymbol{X}}_{i,h}^{\mathrm{T}}\boldsymbol{\beta})^2.\end{aligned} \quad (9)$$

According to the Taylor expansion of $\mathcal{L}_N(\boldsymbol{\beta})$ around $\widehat{\boldsymbol{\beta}}_0$, we have

$$\begin{aligned}\mathcal{L}_N(\boldsymbol{\beta}) = \mathcal{L}_N(\widehat{\boldsymbol{\beta}}_0) + \{\partial\mathcal{L}_N(\widehat{\boldsymbol{\beta}}_0)\}^{\mathrm{T}}(\boldsymbol{\beta}-\widehat{\boldsymbol{\beta}}_0) \\ + \frac{1}{2}(\boldsymbol{\beta}-\widehat{\boldsymbol{\beta}}_0)^{\mathrm{T}}\widehat{\boldsymbol{D}}_h(\boldsymbol{\beta}-\widehat{\boldsymbol{\beta}}_0).\end{aligned} \tag{10}$$

It is easy to see that $\partial\mathcal{L}_N(\widehat{\boldsymbol{\beta}}_0)$ and $\widehat{\boldsymbol{D}}_h$ can be simply calculated by averaging the local ones. However, the burden of transmitting the local $p\times p$ covariance matrix $\widehat{\boldsymbol{D}}_{k,h}$ is heavy when p is large. To save the communication cost, we replace the global Hessian $\widehat{\boldsymbol{D}}_h$ with the local Hessian $\widehat{\boldsymbol{D}}_{1,b}$. Here, h and b denote the *global bandwidth* and *local bandwidth*, respectively, and we assume $b\geq h\geq 0$. Thus we can rewrite (10) as

$$\begin{aligned}\mathcal{L}_N(\boldsymbol{\beta},\widehat{\boldsymbol{D}}_h) = \underbrace{\mathcal{L}_N(\boldsymbol{\beta},\widehat{\boldsymbol{D}}_{1,b})}_{(i)\text{ Shifted loss}} \\ + \underbrace{\mathcal{O}_{\mathbb{P}}\left\{\|\widehat{\boldsymbol{D}}_h-\widehat{\boldsymbol{D}}_{1,b}\|_{op}\cdot|\boldsymbol{\beta}-\widehat{\boldsymbol{\beta}}_0|_2^2\right\}}_{(ii)\text{ Approximation error}}.\end{aligned} \tag{11}$$

The second term in (11) is from the Cauchy–Schwarz inequality. Note that the substituted local Hessian matrix is flexibly controlled by a local bandwidth b instead of the global bandwidth h, which ensures that $\|\widehat{\boldsymbol{D}}_h-\widehat{\boldsymbol{D}}_{1,b}\|_{op}=o_{\mathbb{P}}(1)$ (Detailed proof can be referred to in the supplemental). Remove the terms that are independent of $\boldsymbol{\beta}$ in (i) and the negligible approximation error (ii) in (11), the shifted loss function can be simplified to

$$\widetilde{\mathcal{L}}(\boldsymbol{\beta}) = \frac{1}{2n}\sum_{i\in\mathcal{M}_1}(\widetilde{\boldsymbol{X}}_{i,b}^{\mathrm{T}}\boldsymbol{\beta})^2 - \boldsymbol{\beta}^{\mathrm{T}}\left\{\boldsymbol{z}_N + (\widehat{\boldsymbol{D}}_{1,b}-\widehat{\boldsymbol{D}}_h)\widehat{\boldsymbol{\beta}}_0\right\}, \tag{12}$$

where $z_N = \frac{1}{N}\sum_{i=1}^N\widetilde{\boldsymbol{X}}_{i,h}\widetilde{Y}_{i,h}$. Up to now, we only need to focus on the shifted loss function in (12) instead of the pseudo global loss function in (9) for higher communication efficiency. Specifically, we define the one-step distributed estimator as

$$\widehat{\boldsymbol{\beta}}_{1,h} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\ \widetilde{\mathcal{L}}(\boldsymbol{\beta}) + \lambda_N|\boldsymbol{\beta}|_1. \tag{13}$$

Note that the local machines only need to compute two vectors $z_{n,k} = \frac{1}{n}\sum_{i\in\mathcal{M}_k}\widetilde{\boldsymbol{X}}_{i,h}\widetilde{Y}_{i,h}$ and $\widehat{\boldsymbol{D}}_{k,h}\widehat{\boldsymbol{\beta}}_0 = \frac{1}{n}\sum_{i\in\mathcal{M}_k}\widetilde{\boldsymbol{X}}_{i,h}(\widetilde{\boldsymbol{X}}_{i,h}^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_0)$ and then broadcast them to the central machine with communication cost of $\mathcal{O}(mp)$. There is no need to communicate the $p\times p$ covariance matrix $\widehat{\boldsymbol{D}}_{k,h}$. The central machine first calculates z_N and $\widehat{\boldsymbol{D}}_h\widehat{\boldsymbol{\beta}}_0$ by simple averaging, then solves (13) via some well-learned algorithms, e.g., the PSSsp algorithm [43], the active set algorithm [44] and the coordinate descent algorithm [45].

Given $\widehat{\boldsymbol{\beta}}_{1,h}$ as the estimator from the first iteration, we can similarly construct an iterative distributed estimation procedure. Specifically, in the t-th iteration, we update the pseudo covariates and responses in (6) by substituting $\widehat{\boldsymbol{\beta}}_0$ with $\widehat{\boldsymbol{\beta}}_{t-1,h}$,

$$\begin{aligned}\widetilde{\boldsymbol{X}}_{i,h}^{(t)} &= \sqrt{K_h(\widehat{e}_{i,h}^{(t-1)})}\boldsymbol{X}_i, \\ \widetilde{Y}_{i,h}^{(t)} &= (\widetilde{\boldsymbol{X}}_{i,h}^{(t)})^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_{t-1,h} - \frac{1}{\sqrt{K_h(\widehat{e}_{i,h}^{(t-1)})}}\left(I(\widehat{e}_{i,h}^{(t-1)}\leq 0)-\tau\right),\end{aligned} \tag{14}$$

Algorithm 1 Distributed high-dimensional sparse quantile regression (DHSQR).

1: **Input:** Samples $\{(\boldsymbol{X}_i,Y_i)\}_{i\in\mathcal{M}_k}, k=1,\ldots,m$, the number of iterations T, the quantile level τ, the kernel function K, the global and local bandwidth h and b, the regularization parameters λ_0 and $\lambda_{N,t}$ for $t=1,\ldots,T$.
2: Compute the initial estimator $\widehat{\boldsymbol{\beta}}_{0,h}=\widehat{\boldsymbol{\beta}}_0$ based on $\{(\boldsymbol{X}_i,Y_i)\}_{i\in\mathcal{M}_1}$ by

$$\widehat{\boldsymbol{\beta}}_0 = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\frac{1}{n}\sum_{i\in\mathcal{M}_1}\rho_\tau(Y_i-\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\beta})+\lambda_0|\boldsymbol{\beta}|_1.$$

3: **For** $t=1,\ldots,T$ **do**:
4: Broadcast $\widehat{\boldsymbol{\beta}}_{t-1,h}$ to the local machines.
5: **for** $k=1,\ldots,m$ **do:**
6: The k-th machine update the pseudo covariates $\widetilde{\boldsymbol{X}}_{i,h}^{(t)}$ and responses $\widetilde{Y}_{i,h}^{(t)}$ based on (14), and computes $\widehat{\boldsymbol{D}}_{k,h}^{(t)}\widehat{\boldsymbol{\beta}}_{t-1,h} = \frac{1}{n}\sum_{i\in\mathcal{M}_k}\widetilde{\boldsymbol{X}}_{i,h}^{(t)}((\widetilde{\boldsymbol{X}}_{i,h}^{(t)})^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_{t-1,h})$ and $\boldsymbol{z}_{n,k}^{(t)} = \frac{1}{n}\sum_{i\in\mathcal{M}_k}\widetilde{\boldsymbol{X}}_{i,h}^{(t)}\widetilde{Y}_{i,h}^{(t)}$. Then send them back to the first machine.
7: **end for**
8: The first machine computes $(\widehat{\boldsymbol{D}}_{1,b}^{(t)}-\widehat{\boldsymbol{D}}_h^{(t)})\widehat{\boldsymbol{\beta}}_{t-1,h}$ and $\boldsymbol{z}_N^{(t)}$ based on

$$(\widehat{\boldsymbol{D}}_{1,b}^{(t)}-\widehat{\boldsymbol{D}}_h^{(t)})\widehat{\boldsymbol{\beta}}_{t-1,h} = \widehat{\boldsymbol{D}}_{1,b}^{(t)}\widehat{\boldsymbol{\beta}}_{t-1,h} - \frac{1}{m}\sum_{k=1}^m\widehat{\boldsymbol{D}}_{k,h}^{(t)}\widehat{\boldsymbol{\beta}}_{t-1,h},$$

$$\boldsymbol{z}_N^{(t)} = \frac{1}{m}\sum_{k=1}^m\boldsymbol{z}_{n,k}^{(t)}.$$

9: Compute the estimator $\widehat{\boldsymbol{\beta}}_{t,h}$ on the first machine based on (15).
10: **end for**
11: **Return:** $\widehat{\boldsymbol{\beta}}_{T,h}$.

for $i=1,\ldots,N$, where $\widehat{e}_{i,h}^{(t)} = Y_i-\boldsymbol{X}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_{t,h}$. Similar to (13), the distributed estimator in the t-th iteration is given by

$$\begin{aligned}\widehat{\boldsymbol{\beta}}_{t,h} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\frac{1}{2n}\sum_{i\in\mathcal{M}_1}\left((\widetilde{\boldsymbol{X}}_{i,b}^{(t)})^{\mathrm{T}}\boldsymbol{\beta}\right)^2 \\ -\boldsymbol{\beta}^{\mathrm{T}}\left\{\boldsymbol{z}_N^{(t)}+\left(\widehat{\boldsymbol{D}}_{1,b}^{(t)}-\widehat{\boldsymbol{D}}_h^{(t)}\right)\widehat{\boldsymbol{\beta}}_{t-1,h}\right\}+\lambda_{N,t}|\boldsymbol{\beta}|_1,\end{aligned} \tag{15}$$

where $\boldsymbol{z}_N^{(t)} = \frac{1}{N}\sum_{i=1}^N\widetilde{\boldsymbol{X}}_{i,h}^{(t)}\widetilde{Y}_{i,h}^{(t)}$, $\widehat{\boldsymbol{D}}_{1,b}^{(t)} = \frac{1}{n}\sum_{i\in\mathcal{M}_1}\widetilde{\boldsymbol{X}}_{i,b}^{(t)}(\widetilde{\boldsymbol{X}}_{i,b}^{(t)})^{\mathrm{T}}$, and $\widehat{\boldsymbol{D}}_h^{(t)} = \frac{1}{N}\sum_{i=1}^N\widetilde{\boldsymbol{X}}_{i,h}^{(t)}(\widetilde{\boldsymbol{X}}_{i,h}^{(t)})^{\mathrm{T}}$ with $\widetilde{\boldsymbol{X}}_{i,b}^{(t)} = \sqrt{K_b(\widehat{e}_{i,h}^{(t-1)})}\boldsymbol{X}_i$.

In this paper, we adopt the coordinate descent algorithm to solve (15). For the choice of the initial estimator $\widehat{\boldsymbol{\beta}}_0$, we take the solution of ℓ_1-penalized QR regression using the local data on the central machine, which can be solved by the R package "*quantreg*" [10] or "*conquer*" [12]. Other types of initialization are also available as long as they satisfy Assumption 6 in Section III. We summarize the entire distributed estimation procedure in Algorithm 1.

Space complexity. In each local machine $\mathcal{M}_i$, DHSQR method necessitates storing $\{\boldsymbol{X}_i\}_{i\in\mathcal{M}_i} \in \mathbb{R}^{p\times n}$, $\{Y_i\}_{i\in\mathcal{M}_i} \in \mathbb{R}^n$, $\{\widetilde{\boldsymbol{X}}_{i,h}^{(t)}\}_{i\in\mathcal{M}_i} \in \mathbb{R}^{p\times n}$, and $\{\widetilde{Y}_{i,h}^{(t)}\}_{i\in\mathcal{M}_i} \in \mathbb{R}^n$, resulting in a space complexity of order $\mathcal{O}(np)$. Additionally, to solve the Lasso problem, we require storing a $p\times p$ matrix, resulting in a space complexity not exceeding $\mathcal{O}\left(p^2\right)$. Hence, the overall space complexity is of order $\mathcal{O}\left(np+p^2\right)$ for each local machine. The total space complexity of the total system sums up to $\mathcal{O}\left(Np+mp^2\right)$.

Remark 2. *Note that the assumption that the samples are randomly and evenly stored across the local machines is commonly required in literature [8], [29]. It is worth pointing out that the proposed algorithm is still effective if the samples in other local machines are not randomly distributed as long as the subsample on the first machine (central machine) is randomly selected from the entire sample, which is also claimed in Remark 1 in [14].*

III. Statistical Guarantees for Distributed Estimation

In this section, we establish the theoretical results of our proposed estimation method, involving the convergence rate and support recovery accuracy. Firstly, we denote

$$S = \{j : \beta_j \neq 0, j \in \mathbb{N}_+\} \subseteq \{1,\ldots,p\},$$

as the support of $\boldsymbol{\beta}^*$ and $|S| = s$. We assume the following regular conditions hold.

Assumption 1. *Assume that the kernel function $K(\cdot)$ is symmetric, non-negative, bounded, and integrates to one. In addition, the kernel function satisfies that $\int_{-\infty}^{\infty} u^2K(u)\mathrm{d}u < \infty$ and $\min_{|u|\leq 1} K(u) > 0$. We further assume $K(\cdot)$ is second-order differentiable and its derivative $K'(\cdot)$ and second derivative $K''(\cdot)$ are bounded. Moreover, denote $\kappa_k = \int_{-\infty}^{\infty} |u|^k K(u)\mathrm{d}u$ for $k \geq 1$.*

Assumption 2. *There exists $f_2 \geq f_1 > 0$ such that $f_1 \leqslant f_{\varepsilon|\boldsymbol{X}}(0) \leqslant f_2$ almost surely (for all $\boldsymbol{X}$). Moreover, there exists some l_0 such that*

$$\left|f_{\varepsilon|\boldsymbol{X}}(u) - f_{\varepsilon|\boldsymbol{X}}(v)\right| \leq l_0|u-v|,$$

for any $u, v \in \mathbb{R}$ and all $\boldsymbol{X}$, and we assume that the derivative $f'_{\varepsilon|\boldsymbol{X}}(u)$ is bounded.

Assumption 3. *The random covariate $\boldsymbol{X} \in \mathbb{R}^p$ is sub-Gaussian: there exists some $c_1 > 0$ such that*

$$\mathbb{P}\left(\left|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1/2}\delta\right| \geqslant c_1 t\right) \leq 2e^{-t^2/2},$$

for every unit vector δ and $t > 0$, where $\boldsymbol{\Sigma} = \mathbb{E}(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})$. Furthermore, $0 \leq \lambda_{\min} \leq \Lambda_{\min}(\boldsymbol{\Sigma}) \leq 1 \leq \Lambda_{\max}(\boldsymbol{\Sigma}) \leq \lambda_{\max} < \infty$ and the precision matrix $\boldsymbol{\Sigma}^{-1}$ satisfies $\|\boldsymbol{\Sigma}^{-1}\|_\infty \leq C$. Besides, $m_4 = \sup_{\boldsymbol{u}\in\mathbb{S}^{p-1}} \mathbb{E}(|\langle \boldsymbol{u}, \boldsymbol{\Sigma}^{-1/2}\boldsymbol{X}\rangle|^4) < \infty$.

Assumption 4. *Denote $\mathbf{I} = \boldsymbol{H}(\boldsymbol{\beta}^*) = \mathbb{E}\left\{f_{\varepsilon|\boldsymbol{X}}(0)\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\right\}$, then $\mathbf{I}$ satisfies that*

$$\left\|\mathbf{I}_{S^c\times S}\mathbf{I}_{S\times S}^{-1}\right\|_\infty \leq 1-\alpha,$$

for some $0 < \alpha < 1$. Moreover, we assume that $\lambda_- \leq \Lambda_{\min}(\mathbf{I}) \leq \Lambda_{\max}(\mathbf{I}) \leq \lambda_+$ for some $\lambda_-, \lambda_+ > 0$.

Assumption 5. *The dimension p satisfies $p = \mathcal{O}\left(N^\nu\right)$ for some $\nu > 0$. The local sample size n satisfies $n \geq N^c$ for some $0 < c < 1$, and the sparsity level s satisfies $s = \mathcal{O}(\sqrt{\log p})$.*

Assumption 6. *We assume the initial estimator $\widehat{\boldsymbol{\beta}}_{0,h}$ satisfies that $|\widehat{\boldsymbol{\beta}}_{0,h} - \boldsymbol{\beta}^*|_2 = \mathcal{O}_{\mathbb{P}}(a_n)$, where $a_n \asymp \sqrt{s\log p/n}$. And suppose that $\mathbb{P}(\mathrm{supp}(\widehat{\boldsymbol{\beta}}_{0,h}) \subseteq S) \to 1$.*

Assumption 1 imposes some regularity conditions on the kernel function $K(\cdot)$, which is satisfied by many popular kernels, including the Gaussian kernel. Assumption 2 is a mild condition on the smoothness of the conditional density function of the error term, which is standard in quantile regression [14], [34], [46]. Assumption 3 requires that the distribution of $\boldsymbol{X}$ have heavier tails than Gaussian to obtain standard convergence rates for the quantile regression estimates. Assumption 4 is known as the *irrepresentable condition*, which is commonly assumed in the sparse high-dimensional estimation literature for the support recovery [14], [21], [47]. Assumption 5 is also a common condition in the distributed estimation literature, see also in [8], [14], [28]. Note that $p = \mathcal{O}(N^\nu)$, we use $\log p$ instead of the commonly used $\log(\max(p, N))$ in the convergence rates. Assumption 6 assumes the convergence rate and support recovery accuracy of the initial estimator, which can be satisfied by the estimator using the local sample from a single machine under Assumption 2-5 and some regularity conditions [48]. We first show the convergence rate of the one-step DHSQR estimator $\widehat{\boldsymbol{\beta}}_{1,h}$.

Theorem 1. *Suppose that the initial estimator satisfies that $|\widehat{\boldsymbol{\beta}}_{0,h} - \boldsymbol{\beta}^*|_2 = \mathcal{O}_{\mathbb{P}}(a_n)$ and let $h \asymp (s\log p/N)^{1/3}$, $b \asymp (s\log p/n)^{1/4}$ and $a_n \asymp \sqrt{s\log p/n}$. Take*

$$\lambda_N = C\left(\sqrt{\frac{\log p}{N}} + a_n\left(\frac{s\log p}{n}\right)^{1/4}\right),$$

where C is a sufficient large constant. Then under Assumption 1-6, we have

$$\left|\widehat{\boldsymbol{\beta}}_{1,h} - \boldsymbol{\beta}^*\right|_2 = \mathcal{O}_{\mathbb{P}}\left(\sqrt{\frac{s\log p}{N}} + \sqrt{s}a_n\left(\frac{s\log p}{n}\right)^{1/4}\right). \tag{16}$$

With a proper choice of the global and local bandwidth h and b, we can refine the initial estimator by one iteration of our algorithm. Specifically, the convergence rate reduces from $\mathcal{O}_{\mathbb{P}}(a_n)$ to $\mathcal{O}_{\mathbb{P}}(\max\{\sqrt{s\log p/N}, s^{3/4}(\log p/n)^{1/4}a_n\})$ with $s^{3/4}(\log p/n)^{1/4} = o(1)$ by Assumption 5. Now, we can recursively apply Theorem 1 to get the convergence rate of the iterative DHSQR estimator.

Theorem 2. *Suppose that the initial estimator satisfies that $|\widehat{\boldsymbol{\beta}}_{0,h} - \boldsymbol{\beta}^*|_2 = \mathcal{O}_{\mathbb{P}}(\sqrt{s\log p/n})$ and let $h \asymp (s\log p/N)^{1/3}$, $b \asymp (s\log p/n)^{1/4}$. For $1 \leq g \leq t$, take*

$$\lambda_{N,g} = = C\left(\sqrt{\frac{\log p}{N}} + s^{3g/4}\left(\frac{\log p}{n}\right)^{(t+2)/4}\right),$$

where C is a sufficiently large constant. Then under Assumption 1-6, we have

$$\left|\widehat{\boldsymbol{\beta}}_{t,h}-\boldsymbol{\beta}^*\right|_2=\mathcal{O}_{\mathbb{P}}\left(\sqrt{\frac{s\log p}{N}}+s^{(3t+2)/4}\left(\frac{\log p}{n}\right)^{(t+2)/4}\right). \tag{17}$$

When the number of iterations t satisfies that

$$t\geq t_{max}=\frac{2\log(N/n)}{\log(c_0 n/(s^3\log p))},\quad \text{for some constant } c_0>0, \tag{18}$$

the second term in (17) will be dominated by the first term, therefore, we have $|\widehat{\boldsymbol{\beta}}_{t,h}-\boldsymbol{\beta}^*|_2=\mathcal{O}_{\mathbb{P}}(\sqrt{s\log p/N})$. Under Assumption 5, we can easily verify that the right side of (18) is bounded by a constant, which indicates that after a constant number of iterations, the DHSQR estimator can reach the same convergence rate as the traditional ℓ_1-penalized QR estimators in a single machine [21]. Interestingly, our algorithm needs the number of iterations to increase logarithmically with the number of machines m to achieve the oracle rate $\sqrt{s/N}$ (up to a logarithmic factor). However, most existing distributed first-order algorithms require the number of iterations to increase polynomially with m [49].

Remark 3. *It is worth noting that the shrinkage rate of the second term in* (17) *is of order* $(s^3\log p/n)^{1/4}$*, whereas in [14], it is of order* $\sqrt{s^2\log p/n}$*. The shrinkage rate in our algorithm is slightly slower than that in [14], indicating that our algorithm requires more iterations to achieve the global convergence rate. However, our method is capable of handling not only the homogeneous case, where the error term is independent of the covariates, but also the heterogeneous case, which is more common in practice. This is largely due to the novel design of double smoothing, and the choice of the local bandwidth b in our algorithm is crucial to the shrinkage rate. The experiments in Section VII further demonstrate this phenomenon. It is important to emphasize that the choices of the two bandwidths h and b in Theorem 1 and 2 play determining roles in the distributed inference discussed in Section IV.*

Next, we provide the support recovery of the one-step and t-th iteration DHSQR estimators in the following two theorems. Let $\widehat{\boldsymbol{\beta}}_{t,h}=(\widehat{\beta}^1_{t,h},\widehat{\beta}^2_{t,h},\ldots,\widehat{\beta}^p_{t,h})$ and

$$\widehat{S}_t=\{j:\widehat{\beta}^p_{t,h}\neq 0, j\in\mathbb{N}_+\},$$

be the support of $\widehat{\boldsymbol{\beta}}_{t,h}$, where $t\geq 1$.

Theorem 3. *Under the same conditions of Theorem 1, we have* $\mathbb{P}(\widehat{S}_1\subseteq S)\to 1$*. Furthermore, if there exists a sufficiently large constant $C>0$ such that*

$$\min_{j\in S}\left|\beta^*_j\right|\geq C\left\|\mathbf{I}^{-1}_{S\times S}\right\|_\infty\left(\sqrt{\frac{\log p}{N}}+a_n\left(\frac{s\log p}{n}\right)^{1/4}\right). \tag{19}$$

Then we have $\mathbb{P}(\widehat{S}_1=S)\to 1$.

Based on Theorem 3, we can show a weaker *beta-min* condition for the support recovery result of the t-th iteration DHSQR estimator.

Theorem 4. *Under the same conditions of Theorem 2, we have* $\mathbb{P}(\widehat{S}_t\subseteq S)\to 1$*. Furthermore, if there exists a sufficiently large constant $C>0$ such that*

$$\min_{j\in S}\left|\beta^*_j\right|\geq C\left\|\mathbf{I}^{-1}_{S\times S}\right\|_\infty\left(\sqrt{\frac{\log p}{N}}+s^{3t/4}\left(\frac{\log p}{n}\right)^{\frac{t+2}{4}}\right). \tag{20}$$

Then we have $\mathbb{P}(\widehat{S}_t=S)\to 1$.

Theorems 3 and 4 establish the support recovery results of our one-step and iterative DHSQR estimators by the *beta-min* condition $\min_{j\in S}\left|\beta^*_j\right|$, which is wildly used in the sparse high-dimensional estimation literature. When the number of iterations t satisfies (18), the *beta-min* condition will reduce to $\min_{j\in S}\left|\beta^*_j\right|\geq C\left\|\mathbf{I}^{-1}_{S\times S}\right\|_\infty\sqrt{\log p/N}$, which matches the rate of the lower bound for the *beta-min* condition in a single machine [4].

IV. Methodology for Distributed Debiasing

In this section, we develop a debiasing method for valid distributed statistical inference for the proposed DHSQR estimator. We first introduce the debiasing technique for the transformed Lasso estimator and then extend it to the distributed setting using a double-smoothing approach. Specifically, to balance communication efficiency and statistical accuracy, we adopt the local CLIME estimator [50] with a local bandwidth b to estimate the sparse precision matrix. The local machines only need to compute and broadcast the gradients with a global bandwidth h.

A. Debiasing Transformed Lasso Estimator

Recall that we transform the standard ℓ_1-penalized quantile regression into a least squares regression with a Lasso penalty in (7). In the literature, substantial work has discussed that the Lasso estimator is asymptotically biased due to the ℓ_1-penalty, thus lacking a tractable limiting distribution. This has been explored in linear regression [7], [15]–[17] and quantile regression [18], [19], [37], [51]. Following the debiasing approach proposed by [15] and [16], we first invert the optimality condition of (7). The estimator $\widehat{\boldsymbol{\beta}}_{pool}$ satisfies that

$$\frac{1}{N}\sum_{i=1}^{N}(\widetilde{\boldsymbol{X}}^{\mathrm{T}}_{i,h}\widehat{\boldsymbol{\beta}}_{pool}-\widetilde{Y}_{i,h})\widetilde{\boldsymbol{X}}_{i,h}+\lambda\boldsymbol{g}=0, \tag{21}$$

where $\boldsymbol{g}=(g_0,g_1,\ldots,g_p)^{\mathrm{T}}$ is a sub-gradient of $\|\cdot\|_1$ at $\widehat{\boldsymbol{\beta}}_{pool}$, satisfying $g_j=\operatorname{sign}\left(\widehat{\boldsymbol{\beta}}_{pool;j}\right)$ if $\widehat{\beta}_{pool;j}\neq 0$ and otherwise $g_j\in[-1,1]$. Here, $\widehat{\beta}_{pool;j}$ denotes the j-th coordinate of $\widehat{\boldsymbol{\beta}}_{pool}$.

Since $\widehat{\boldsymbol{\beta}}_{pool}$ is close to $\boldsymbol{\beta}^*$ when N is large, according to concentration inequalities and Taylor expansion, it holds that:

$$\begin{aligned}\frac{1}{N}\sum_{i=1}^{N}(\widetilde{\boldsymbol{X}}_{i,h}^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_{pool}-\widetilde{Y}_{i,h})\widetilde{\boldsymbol{X}}_{i,h}&=\frac{1}{N}\sum_{i=1}^{N}(I(\widehat{e}_{pool,i}\le 0)-\tau)\boldsymbol{X}_i\\ &\approx\frac{1}{N}\sum_{i=1}^{N}(I(\varepsilon_i\le 0)-\tau)\boldsymbol{X}_i\\ &\quad+\mathbb{E}\big[(I(\widehat{e}_{pool,i}\le 0)-\tau)\boldsymbol{X}\big]-\mathbb{E}\big[(I(\varepsilon\le 0)-\tau)\boldsymbol{X}\big]\\ &\approx\frac{1}{N}\sum_{i=1}^{N}(I(\varepsilon_i\le 0)-\tau)\boldsymbol{X}_i+\boldsymbol{H}(\boldsymbol{\beta}^*)(\widehat{\boldsymbol{\beta}}_{pool}-\boldsymbol{\beta}^*).\end{aligned}\tag{22}$$

where $\varepsilon_i=Y_i-\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\beta}^*$ and $\widehat{e}_{pool,i}=Y_i-\boldsymbol{X}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_{pool}$. Resorting (22), we can get

$$\begin{aligned}\widehat{\boldsymbol{\beta}}_{pool}\approx\boldsymbol{\beta}^*+\boldsymbol{H}^{-1}(\boldsymbol{\beta}^*)\frac{1}{N}\sum_{i=1}^{N}(\widetilde{\boldsymbol{X}}_{i,h}^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_{pool}-\widetilde{Y}_{i,h})\widetilde{\boldsymbol{X}}_{i,h}\\ -\boldsymbol{H}^{-1}(\boldsymbol{\beta}^*)\frac{1}{N}\sum_{i=1}^{N}(I(\varepsilon_i\le 0)-\tau)\boldsymbol{X}_i.\end{aligned}$$

According to (21), the non-negligible bias term is $\boldsymbol{H}^{-1}(\boldsymbol{\beta}^*)\frac{1}{N}\sum_{i=1}^{N}(\widetilde{\boldsymbol{X}}_{i,h}^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_{pool}-\widetilde{Y}_{i,h})\widetilde{\boldsymbol{X}}_{i,h}$ which needs to be removed from $\widehat{\boldsymbol{\beta}}_{pool}$. And the term $\boldsymbol{H}^{-1}(\boldsymbol{\beta}^*)\frac{1}{N}\sum_{i=1}^{N}(I(\varepsilon_i\le 0)-\tau)\boldsymbol{X}_i$ is asympotic negligible. Therefore, the debiased pooled DHSQR estimator can be defined as

$$\widetilde{\boldsymbol{\beta}}_{pool}=\widehat{\boldsymbol{\beta}}_{pool}+\widehat{\mathbf{W}}\frac{1}{N}\sum_{i=1}^{N}(\widetilde{\boldsymbol{X}}_{i,h}^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_{pool}-\widetilde{Y}_{i,h})\widetilde{\boldsymbol{X}}_{i,h},\tag{23}$$

where $\widehat{\mathbf{W}}$ is an approximate inverse to $\boldsymbol{H}(\boldsymbol{\beta}^*)$. Note that $\boldsymbol{H}(\boldsymbol{\beta}^*)$ is identical to $\mathbf{I}$ in Assumption 4; we choose one of them to denote the population Hessian matrix for ease of notation. Intuitively, the correcting term $\frac{1}{N}\sum_{i=1}^{N}\widetilde{\boldsymbol{X}}_{i,h}(\widetilde{\boldsymbol{X}}_{i,h}^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_{pool}-\widetilde{Y}_{i,h})$ is a subgradient of $\lambda\|\cdot\|_1$ at $\widehat{\boldsymbol{\beta}}_{pool}$. By adding a term proportional to the subgradient of the penalty, the debiased pooled DHSQR estimator compensates for the bias induced by regularization.

B. General CLIME Estimator

Under Assumptions 1-3, if the initial estimator $\widehat{\boldsymbol{\beta}}_0$ is consistent, we can show that $\widehat{\boldsymbol{D}}_h(\widehat{\boldsymbol{\beta}}_{pool})=\frac{1}{N}\sum_{i=1}^{N}K_h(Y_i-\boldsymbol{X}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_{pool})\boldsymbol{X}_i\boldsymbol{X}_i^{\mathrm{T}}$ is consistent estimator of $\boldsymbol{H}(\boldsymbol{\beta}^*)$. As a consequence, we can inverse $\widehat{\boldsymbol{D}}_h(\widehat{\boldsymbol{\beta}}_{pool})$ to approximate the precision matrix $\boldsymbol{H}^{-1}(\boldsymbol{\beta}^*)$. To avoid the case when $\widehat{\boldsymbol{D}}_h(\widehat{\boldsymbol{\beta}}_{pool})$ is singular, we use the sparse precision matrix estimators proposed in [52], also named as the CLIME estimator. Specifically, $\widehat{\mathbf{W}}_h$ is the solution to the following optimization problem:

$$\widehat{\mathbf{W}}_h=\underset{\mathbf{W}\in\mathbb{R}^{p\times p}}{\operatorname{argmin}}\|\mathbf{W}\|_\infty,\quad\text{s.t. }|\mathbf{W}\widehat{\boldsymbol{D}}_h(\widehat{\boldsymbol{\beta}}_{pool})-\boldsymbol{I}|_\infty\le\gamma_{N,h},\tag{24}$$

where $\gamma_{N,h}$ is a predetermined tuning parameter. The optimization can be solved by the R package *"falre"* [53]. Since the obtained result $\widehat{\mathbf{W}}_h$ is not symmetric in general, the final CLIME estimator is obtained by symmetrizing $\widehat{\mathbf{W}}_h$ as follows, if $\widehat{\mathbf{W}}_h=(\widehat{w}_{i,j})_{1\le i,j\le p}$, then

$$\widehat{\mathbf{W}}_h'=\left(\widehat{w}_{i,j}'\right)_{1\le i,j\le p},$$

where $\widehat{w}_{i,j}'=\widehat{w}_{j,i}'=\widehat{w}_{i,j}I\{|\widehat{w}_{i,j}|\le|\widehat{w}_{j,i}|\}+\widehat{w}_{j,i}I\{|\widehat{w}_{i,j}|>|\widehat{w}_{j,i}|\}$. Without loss of generality, we assume $\widehat{\mathbf{W}}_h$ is symmetric in the rest of the paper. For simple implementation and technical analysis, the optimization problem can be decomposed into p-vector minimization problem. For more details, we refer to Lemma 1 in [52].

C. Distributed Approach: Double-smoothing and Debiasing

Recall the distributed setting in Section II-C, we assume the entire data is randomly and evenly stored in m local machines with sample size $n=N/m$. A naive approach is to construct the averaging debiased Lasso estimator. Specifically, in each machine $\mathcal{M}_k$, we first use the local data to calculate the transformed Lasso estimator $\widehat{\boldsymbol{\beta}}_{k,naive}$ and local CLIME estimator $\widehat{\mathbf{W}}_h^{(k)}$ as that in (7) and (24). The final estimator is then defined as

$$\begin{aligned}\bar{\boldsymbol{\beta}}_{DC}^{d}=\frac{1}{m}\sum_{k=1}^{m}\Bigg(&\widehat{\boldsymbol{\beta}}_{k,naive}\\ &+\widehat{\mathbf{W}}_h^{(k)}\frac{1}{n}\sum_{i\in\mathcal{M}_k}(\widetilde{\boldsymbol{X}}_{i,h}^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_{k,naive}-\widetilde{Y}_{i,h})\widetilde{\boldsymbol{X}}_{i,h}\Bigg).\end{aligned}\tag{25}$$

However, the averaged debiased estimator needs each machine to estimate the $p\times p$ precision matrix, thus the whole system requires solving mp optimization problems. Along with the expensive computation cost, the theoretical results also need a stringent condition that the number of machines m is not too large to retain the global convergence rate [30]. To address these issues, motivated by the iterative double-smoothing distributed estimator proposed in (15), we define its debiased version in T_0-th iteration as

$$\widetilde{\boldsymbol{\beta}}_{T_0,h}=\widehat{\boldsymbol{\beta}}_{T_0,h}-\widehat{\mathbf{W}}_b^{(1)}\frac{1}{N}\sum_{i=1}^{N}\left((\widetilde{\boldsymbol{X}}_{i,h}^{(T_0)})^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_{T_0,h}-\widetilde{Y}_{i,h}^{(T_0)}\right)\widetilde{\boldsymbol{X}}_{i,h}^{(T_0)},\tag{26}$$

where $\widehat{\mathbf{W}}_b^{(1)}$ represents the solution in (24) based on $\widehat{\boldsymbol{D}}_{1,b}^{(T_0)}=\frac{1}{n}\sum_{i\in\mathcal{M}_1}\widetilde{\boldsymbol{X}}_{i,b}^{(T_0)}(\widetilde{\boldsymbol{X}}_{i,b}^{(T_0)})^{\mathrm{T}}$ and tuning parameter $\gamma_{N,n,b}$, and $\widehat{\boldsymbol{\beta}}_{T_0,h}$ is T_0-th estimator from algorithm 1 with $T_0\ge t_{max}+1$ where t_{max} is defined in inequality (18). To achieve a trade-off between communication efficiency and statistical accuracy, we only use a local CLIME estimator $\widehat{\mathbf{W}}_b^{(1)}$ instead of the averaged one $\sum_{k=1}^{m}\widehat{\mathbf{W}}_h^{(k)}/m$, thus the local machines need not to compute and communicate the $p\times p$ matrix.

Remark 4. *Note that* $\widehat{\boldsymbol{D}}_{1,b}^{(T_0)}=\frac{1}{n}\sum_{i\in\mathcal{M}_1}K_b(Y_i-\boldsymbol{X}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_{T_0-1,h})\boldsymbol{X}_i\boldsymbol{X}_i^{\mathrm{T}}$. *Under the assumptions in Theorem 2 and with* T_0-1 *satisfying inequality* (18)*, the* (T_0-1)*-th DHSQR estimator can achieve the global convergence rate, i.e.,* $|\widehat{\boldsymbol{\beta}}_{T_0-1,h}-\boldsymbol{\beta}^*|_2=\mathcal{O}_{\mathbb{P}}(\sqrt{s\log p/N})$. *This is a key condition to derive the non-asymptotic bound of* $\widehat{\mathbf{W}}_b^{(1)}$ *as an approximation of the inverse of* $\widehat{\boldsymbol{D}}_{1,b}^{(T_0)}$ *and thus* $\boldsymbol{H}(\boldsymbol{\beta}^*)$*, as shown in Lemma 1 in the next section. We also want to emphasize that the global bandwidth* h *and local bandwidth* b *should*

Algorithm 2 Inference of distributed high-dimensional debiasing quantile regression

1: **Input:** Samples $\{(\boldsymbol{X}_i, Y_i)\}_{i\in\mathcal{M}_k}, k = 1, \ldots, m$, the quantile level τ, the kernel function K, the global and local bandwidth h and b, significance level ρ, and $\boldsymbol{\nu} \in \mathbb{B}_1(r)$.
2: Run Algorithm 1 for T_0 iteration and obtain the final estimator $\widehat{\boldsymbol{\beta}}_{T_0,h}$ and the pseudo covariates and responses $\{(\widetilde{\boldsymbol{X}}_{i,h}^{(T_0)}, \widetilde{Y}_{i,h}^{(T_0)})\}_{i=1}^N$.
3: The central machine broadcast $\widehat{\boldsymbol{\beta}}_{T_0,h}$ to each local machine.
4: **for** $k = 1, \ldots, m$ **do:**
5: The k-th local machine compute $\boldsymbol{g}_{T_0,h}^{(k)} = \frac{1}{n}\sum_{i\in\mathcal{M}_k}((\widetilde{\boldsymbol{X}}_{i,h}^{(T_0)})^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_{T_0,h} - \widetilde{Y}_{i,h}^{(T_0)})\widetilde{\boldsymbol{X}}_{i,h}^{(T_0)}$, and then send them back to the central machine.
6: **end for**
7: The central machine solve the CLIME optimization in (24) and get $\widehat{\mathbf{W}}_b^{(1)}$ based on the local data. Then compute the debiased estimator as follows:

$$\widetilde{\boldsymbol{\beta}}_{T_0,h} = \widehat{\boldsymbol{\beta}}_{T_0,h} - \widehat{\mathbf{W}}_b^{(1)}\frac{1}{m}\sum_{k=1}^{m}\boldsymbol{g}_{T_0,h}^{(k)}.$$

8: **Output:** The confidence interval for $\boldsymbol{\nu}^{\mathrm{T}}\boldsymbol{\beta}^*$ is $\widehat{C}_N(\alpha)$ defined in (33) and the p-value of the hypothesis test for $\beta_j^* = 0$ is P_j defined in (35).

satisfy the conditions in Theorem 2, i.e., $h \asymp (s\log p/N)^{1/3}$ *and* $b \asymp (s\log p/n)^{1/4}$.

To formulate (26), we first run Algorithm 1 for T_0 iterations, and the central machine broadcasts $\widehat{\boldsymbol{\beta}}_{T_0,h}$ to each local machine. Each local machine then calculates the local gradient $\frac{1}{n}\sum_{i\in\mathcal{M}_k}(\widetilde{\boldsymbol{X}}_{i,h}^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_{T_0,h} - \widetilde{Y}_{i,h})\widetilde{\boldsymbol{X}}_{i,h}$ and sends it to the central machine. The central machine averages the local gradients and performs the CLIME algorithm based on its local data to get $\widehat{\mathbf{W}}_b^{(1)}$. Finally, we obtain the debiased DHSQR estimator in (26). Algorithm 2 outlines the steps for distributed inference.

V. Statistical Guarantees for Distributed Inference

In this section, we provide the theoretical guarantees for the debiased DHSQR estimator, including its Bahadur representation, Berry-Esseen bounds, and asymptotic normality. Finally, we demonstrate how to construct confidence intervals and perform hypothesis testing based on it.

A. Non-asymptotic Bound of the CLIME Estimator

Before presenting the main results, we first show the non-asymptotic bound of $\widehat{\mathbf{W}}_b^{(1)}$, which is an approximation of the inverse of $\widehat{\boldsymbol{D}}_{1,b}^{(T_0)}$ and $\boldsymbol{H}(\boldsymbol{\beta}^*)$. Note that $\widehat{\mathbf{W}}_b^{(1)}$ is a specifical case of the CLIME estimator for sparse precision matrix. To ensure the efficiency of the CLIME estimator and extend the related theory in [52], we need an additional assumption on the inverse of the population Hessian $\boldsymbol{H}^{-1}(\boldsymbol{\beta}^*)$.

Assumption 7. *There exists a constant* $M' > 0$ *such that* $\|\boldsymbol{H}^{-1}(\boldsymbol{\beta}^*)\|_\infty \le M'$. *Moreover,* $\boldsymbol{H}^{-1}(\boldsymbol{\beta}^*) := (\widetilde{\boldsymbol{h}}_1, \ldots, \widetilde{\boldsymbol{h}}_p)^{\mathrm{T}} = (\widetilde{h}_{i,j})_{1\le i,j\le p}$ *is sparse row-wise, i.e.,* $\max_{0\le i\le p}\sum_{j=0}^{p} I(\widetilde{h}_{i,j} \neq 0) \le c_{N,p}, c_{N,p} \ge p$, *where* $c_{N,p}$ *is positive and bounded away from 0 and allowed to increase as* N *and* p *grow.*

Assumption 7 requires $\boldsymbol{H}^{-1}(\boldsymbol{\beta}^*)$ to be sparse both in terms of ℓ_1-norm and matrix row space. This assumption is quite standard in the literature of precision matrix estimation and more general inverse Hessian matrix estimation [16], [52], [54]. Note that a similar assumption is also used in [19] for the inference of convolution-smoothing quantile regression in a single machine setting. However, their assumption relies on the sparsity of the inverse of the population kernel matrix $\mathbb{E}(K_h(\varepsilon)\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})$, which is intrinsically tied to a certain bandwidth h. In contrast, our assumption is directly related to the population Hessian matrix of the quantile loss function $\boldsymbol{H}(\boldsymbol{\beta}^*)$, which is independent of the bandwidth h. This makes our assumption more reliable and applicable to a wider range of quantile regression problems.

Lemma 1. *Suppose the conditions in Theorem 2 and Assumptions 1-7 hold, the iteration satisfies* $T_0 \ge t_{max} + 1$, *then with probability near to 1, we have*

$$\|\widehat{\mathbf{W}}_b^{(1)}\|_\infty \le \|\boldsymbol{H}^{-1}(\boldsymbol{\beta}^*)\|_\infty,\ |\widehat{\mathbf{W}}_b^{(1)}\widehat{\boldsymbol{D}}_{1,b}^{(T_0)} - \boldsymbol{I}|_\infty \lesssim \gamma_{N,n,b},$$
$$|\widehat{\mathbf{W}}_b^{(1)}\boldsymbol{H}(\boldsymbol{\beta}^*) - \boldsymbol{I}|_\infty \lesssim \gamma_{N,n,b}, \tag{27}$$

where $\gamma_{N,n,b} = \sqrt{\frac{\log p}{nb}} + \frac{\log p}{nb} + \frac{s^2(\log p)^3}{Nb^3} + s\sqrt{\frac{(\log p)^2}{N}}(\frac{1}{b} + \sqrt{\frac{\log p}{nb^3}} + \frac{\log p}{nb^2}) + b^2$. *Thus we have*

$$\|\widehat{\mathbf{W}}_b^{(1)} - \boldsymbol{H}^{-1}(\boldsymbol{\beta}^*)\|_\infty \lesssim 8c_{N,p}\gamma_{N,n,b}\|\boldsymbol{H}^{-1}(\boldsymbol{\beta}^*)\|_\infty \asymp \gamma_{N,n,b}. \tag{28}$$

Lemma 1 shows several upper-bound properties of the CLIME estimator of the Hessian. We need to emphasize the importance of $\gamma_{N,n,b}$ in the upper bound, which plays a key role in deriving the asymptotic results. When the local bandwidth b satisfies $b \asymp (s\log p/n)^{1/4}$ and the sparsity $s = \mathcal{O}(\sqrt{\log p})$, $\gamma_{N,n,b}$ will be dominated by the first term $\sqrt{\frac{\log p}{nb}}$, and we can obtain that $\gamma_{N,n,b} \lesssim \max(\frac{(\log p)^{5/16}}{n^{3/8}}, \frac{(\log p)^{9/8}}{m^{1/2}n^{1/4}})$. We will frequently use this lemma in the following theorems.

B. Bahadur Representation and Berry-Esseen bound

In this section, we establish a Bahadur representation for the debiased DHSQR estimator $\widetilde{\boldsymbol{\beta}}_{T_0,h}$, ensuring the theoretical foundation of the statistical inference. Denote $\widehat{\boldsymbol{\delta}}_{T_0,h} = \widehat{\boldsymbol{\beta}}_{T_0,h} - \boldsymbol{\beta}^*$ and $\widehat{e}_{i,h}^{(T_0)} = Y_i - \boldsymbol{X}_i{}^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_{T_0-1,h}$, we can get the following equality after simple calculation (details can be seen in Appendix):

$$\sqrt{N}\boldsymbol{\nu}^{\mathrm{T}}(\widetilde{\boldsymbol{\beta}}_{T_0,h} - \boldsymbol{\beta}^*) = \boldsymbol{\nu}^{\mathrm{T}}\boldsymbol{H}^{-1}(\boldsymbol{\beta}^*)\frac{1}{\sqrt{N}}\sum_{i=1}^{N}(\tau - I(\varepsilon_i \le 0))\boldsymbol{X}_i$$
$$-\sqrt{N}\boldsymbol{\nu}^{\mathrm{T}}\left(\Gamma_1 + \Gamma_2 + \Gamma_3 + \Gamma_4\right),$$

where $\boldsymbol{\nu}$ lies in the ℓ_1-ball $\mathbb{B}_1(r) = \{\boldsymbol{a} \in \mathbb{R}^p : |\boldsymbol{a}|_1 \le r\}$, $\frac{1}{\sqrt{N}}\sum_{i=1}^{N}(I(\varepsilon_i \le 0) - \tau)\boldsymbol{X}_i$ is a zero-mean random vector,

and let $\theta \in (0,1)$. The remainder is given by the expression $\Gamma_1 - \Gamma_4$, which is detailed in the supplement. Note that by the De Moivre-Laplace Central Limit Theorem, as $N \to \infty$, we have

$$\frac{1}{\sqrt{N}} \sum_{i=1}^{N} (I(\varepsilon_i \le 0) - \mathbb{E}I(\varepsilon_i \le 0)) \xrightarrow{D} N(0, \tau(1-\tau)), \tag{29}$$

where $\xrightarrow{D}$ represents for convergence in distribution. With the derived convergence rates of $\widehat{\boldsymbol{\beta}}_{T_0,h}$ and $\widehat{\mathbf{W}}_b^{(1)}$ in Theorem 2 and Lemma 1, we can proof that $\sqrt{N}\boldsymbol{\nu}^{\mathrm{T}}(\Gamma_1+\Gamma_2+\Gamma_3+\Gamma_4) = o_{\mathbb{P}}(1)$ when $N, n \to \infty$. Informally, by Slutsky theorem, we can get the following asymptotic distribution of $\sqrt{N}\boldsymbol{\nu}^{\mathrm{T}}(\widetilde{\boldsymbol{\beta}}_{T_0,h} - \boldsymbol{\beta}^*)$ as

$$\sqrt{N}\boldsymbol{\nu}^{\mathrm{T}}(\widetilde{\boldsymbol{\beta}}_{T_0,h} - \boldsymbol{\beta}^*) \xrightarrow{D} N(0, \tau(1-\tau)\boldsymbol{\nu}^{\mathrm{T}}\boldsymbol{H}^{-1}(\boldsymbol{\beta}^*)\boldsymbol{\Sigma}\boldsymbol{H}^{-1}(\boldsymbol{\beta}^*)\boldsymbol{\nu}).$$

Theorem 5. *Suppose the conditions in Theorem 2 and Assumptions 1-7 hold, the iteration satisfies $T_0 \ge t_{max}+1$, the global and local bandwidths satisfy that $h \asymp (s\log p/N)^{1/3}$, $b \asymp (s\log p/n)^{1/4}$ and the sparsity $s = \mathcal{O}(\sqrt{\log p})$, then for any $\boldsymbol{\nu} \in \mathbb{B}_1(r)$, the debiased DHSQR estimator $\widetilde{\boldsymbol{\beta}}_{T_0,h}$ in* (23) *satisfies*

$$\begin{aligned}&\Big|\sqrt{N}\,\boldsymbol{\nu}^{\top}(\widetilde{\boldsymbol{\beta}}_{T_0,h} - \boldsymbol{\beta}^*) - \tfrac{1}{\sqrt{N}}\,\boldsymbol{\nu}^{\top}\boldsymbol{H}^{-1}(\boldsymbol{\beta}^*)\sum_{i=1}^{N}(I(\varepsilon_i \le 0) - \tau)\mathbf{X}_i\Big| \\ &\lesssim \frac{r\,s^{5/4}\log^{3/2}p}{N^{1/4}} + r\,c_{N,p}\max\Big(\frac{(\log p)^{21/16}}{n^{3/8}}, \frac{(\log p)^{17/8}}{m^{1/2}n^{1/4}}\Big).\end{aligned} \tag{30}$$

with probability near to 1.

Theorem 5 establishes a distributional approximation in the form of a Berry-Esseen bound for the debiased DHSQR estimator. The explicit error bound of the normal approximation is based on the selection of the global and local bandwidths, h and b, and the sparsity level s. When the sample size $N, n \to \infty$, we can derive $\sqrt{N}\sigma_\tau^{-1}\boldsymbol{\nu}^{\mathrm{T}}\left(\widetilde{\boldsymbol{\beta}}_{T_0,h} - \boldsymbol{\beta}^*\right) \xrightarrow{D} N(0,1)$ uniformly over $\boldsymbol{\nu} \in \mathbb{B}_1(r)$, where $\sigma_\tau^2 = \tau(1-\tau)\boldsymbol{\nu}^{\mathrm{T}}\boldsymbol{H}^{-1}(\boldsymbol{\beta}^*)\boldsymbol{\Sigma}\boldsymbol{H}^{-1}(\boldsymbol{\beta}^*)\boldsymbol{\nu}$. This enables us to construct confidence intervals and perform hypothesis testing with a consistent estimator of the asymptotic variance. In particular, to reduce the computation and communication burden, we choose the local CLIME estimator and local sample covariance matrix to build the estimated variance. The following theorem provides the theoretical guarantee for our choice.

Theorem 6. *Suppose the conditions in Theorem 2 and Assumptions 1-7 hold, when the global and local bandwidths satisfy that $h \asymp (s\log p/N)^{1/3}$, $b \asymp (s\log p/n)^{1/4}$ and the sparsity $s = \mathcal{O}(\sqrt{\log p})$, the iteration satisfies $T_0 \ge t_{max}+1$, then for any $\boldsymbol{\nu} \in \mathbb{B}_1(r)$, the debiased DHSQR estimator $\widetilde{\boldsymbol{\beta}}_{T_0,h}$ in* (23) *satisfies*

$$\begin{aligned}&\sup_{x\in\mathbb{R}}\left|\mathbb{P}\left(\frac{\sqrt{N}\,\boldsymbol{\nu}^{\top}(\widetilde{\boldsymbol{\beta}}_{T_0,h} - \boldsymbol{\beta}^*)}{\sqrt{\tau(1-\tau)\,\boldsymbol{\nu}^{\top}\widehat{\mathbf{W}}_b^{(1)}\widehat{\boldsymbol{\Sigma}}^{(1)}\widehat{\mathbf{W}}_b^{(1)}\boldsymbol{\nu}}} \le x\right) - \Phi(x)\right| \\ &\quad\lesssim \frac{r\,s^{5/4}\,\log^{3/2}p}{N^{1/4}} + r\,c_{N,p}\,\max\left(\frac{(\log p)^{21/16}}{n^{3/8}}, \frac{(\log p)^{17/8}}{m^{1/2}n^{1/4}}\right) \\ &\quad + r^2\frac{(\log p)^{1/2}}{n^{1/2}}.\end{aligned} \tag{31}$$

with probability near to 1, where $\Phi(\cdot)$ denotes the cumulative distribution function of the standard normal random variable and $\widehat{\boldsymbol{\Sigma}}^{(1)} = \frac{1}{n}\sum_{i\in\mathcal{M}_1}\boldsymbol{X}_i\boldsymbol{X}_i^{\mathrm{T}}$ is the sample covariance matrix. Thereby, there holds

$$\frac{\sqrt{N}\boldsymbol{\nu}^{\mathrm{T}}(\widetilde{\boldsymbol{\beta}}_{T_0,h} - \boldsymbol{\beta}^*)}{\sqrt{\tau(1-\tau)\boldsymbol{\nu}^{\mathrm{T}}\widehat{\mathbf{W}}_b^{(1)}\widehat{\boldsymbol{\Sigma}}^{(1)}\widehat{\mathbf{W}}_b^{(1)}\boldsymbol{\nu}}} \xrightarrow{D} N(0,1) \tag{32}$$

as $N, n \to \infty$, where $\xrightarrow{D}$ represents for convergence in distribution.

Under some certain choice of the global and local bandwidths, and the iteration, Theorem 6 shows that the linear functionals of $\widetilde{\boldsymbol{\beta}}_{T_0,h}$ are asymptotically normal as $N, n \to \infty$ with the local-based estimated variance. To the best of our knowledge, this is the first result with explicit error bounds and asymptotic normality for high-dimensional debiased quantile estimator in a distributed setting. In addition, with a sparse vector $\boldsymbol{\nu}$, we can construct the asymptotic normality that for every single parameters, i.e., let $\boldsymbol{\nu} = (0,\ldots,1,\ldots,0)^{\mathrm{T}}$, and $\sqrt{N}\left(\widetilde{\beta}_{T_0,h,j} - \beta_j^*\right) \xrightarrow{D} N(0, \tau(1-\tau)[\widehat{\mathbf{W}}_b^{(1)}\widehat{\boldsymbol{\Sigma}}^{(1)}\widehat{\mathbf{W}}_b^{(1)}]_{j,j})$ with $[\cdot]_{j,j}$ denotes the j-th diagonal elements.

C. Confidence Interval and Hypothesis Test

In this section, we consider the distributed inference for the proposed debiased DHSQR estimator $\widetilde{\boldsymbol{\beta}}_{T_0,h}$. First, we construct the confidence interval of the linear functionals of $\widetilde{\boldsymbol{\beta}}_{T_0,h}$. For significance level $\alpha \in (0,1)$, from Theorem 6, it is straightforward to derive an asymptotically valid confidence interval for $\boldsymbol{\nu}^{\mathrm{T}}\widetilde{\boldsymbol{\beta}}^*$ as

$$\widehat{C}_N(\alpha) = \left[\boldsymbol{\nu}^{\mathrm{T}}\widetilde{\boldsymbol{\beta}}_{T_0,h} \pm \frac{\zeta(\tau,\alpha,\boldsymbol{\nu},\widehat{\mathbf{W}}_b^{(1)},\widehat{\boldsymbol{\Sigma}}^{(1)})}{\sqrt{N}}\right], \tag{33}$$

where $\zeta(\tau,\alpha,\boldsymbol{\nu},\widehat{\mathbf{W}}_b^{(1)},\widehat{\boldsymbol{\Sigma}}^{(1)}) = \Phi^{-1}(1-\alpha/2)\sqrt{\tau(1-\tau)\boldsymbol{\nu}^{\mathrm{T}}\widehat{\mathbf{W}}_b^{(1)}\widehat{\boldsymbol{\Sigma}}^{(1)}\widehat{\mathbf{W}}_b^{(1)}\boldsymbol{\nu}}$ and $\Phi^{-1}(1-\alpha/2)$ is the $1-\alpha/2$ quantile of the standard normal distribution.

Theorem 7. *Suppose the conditions and assumptions in Theorem 6 hold, then for any $\boldsymbol{\nu} \in \mathbb{B}_1(r)$, the confidence interval $\widehat{J}_N(\alpha)$ is asymptotically valid, namely*

$$\lim_{N,n\to\infty}\sup_{\boldsymbol{\beta}^*:|\boldsymbol{\beta}^*|_0\le s}\mathbb{P}(\boldsymbol{\nu}^{\mathrm{T}}\boldsymbol{\beta}^* \in \widehat{C}_N(\alpha)) = 1-\alpha. \tag{34}$$

Next, we consider a hypothesis test for one single variable and assess the statistical significance of the non-zero coefficient. Specifically, let the null hypothesis $H_{0,j}: \beta_j^* = 0$ versus the alternative hypothesis $H_{1,j}: \beta_j^* \ne 0$. We construct a p-value P_j for the test $H_{0,j}$ as follows:

$$P_j = 2\left(1 - \Phi\left(\frac{\sqrt{N}|\widetilde{\beta}_{T_0,h,j}|}{\sqrt{\tau(1-\tau)[\widehat{\mathbf{W}}_b^{(1)}\widehat{\boldsymbol{\Sigma}}^{(1)}\widehat{\mathbf{W}}_b^{(1)}]_{j,j}^{1/2}}}\right)\right). \tag{35}$$

Consequently, the decision rule based on the p-value P_j is

$$\widehat{T}_j(\alpha) = \begin{cases}1 & \text{if } P_j \le \alpha \quad (\text{reject } H_{0,j}),\\ 0 & \text{otherwise} \quad (\text{accept } H_{0,j}),\end{cases} \tag{36}$$

where α is the fixed target Type I error probability. Note that choosing $\beta_j \neq 0$ arbitrarily close to zero, it makes no difference to distinguish $H_{0,j}$ and $H_{0,j}$. Consequently, we assume that $|\beta_j| \geq \psi$ if $\beta_j \neq 0$. Follow the framework of high-dimensional linear regression in [17], we give a family of tests for all s-sparse vectors. Specifically, let $T_{j,\mathbf{X}}(\mathbf{Y}) : \mathbb{R}^N \to \{0,1\}$, where $\mathbf{X} \in \mathbb{R}^{N\times p}$ and $\mathbf{Y} \in \mathbb{R}^N$ is the global design matrix and response vector. For $\psi > 0$, we define

$$\alpha_j(T) = \sup_{\boldsymbol{\beta}^*}\Big\{\mathbb{P}_{\beta^*,X,Y}\big(T_{j,X}(Y)=1\big) : \beta^* \in \mathbb{R}^p,\ |\beta^*|_0 \leq s,\ \beta_j^* = 0\Big\},$$

$$\varrho_j(T,\psi) = \sup_{\boldsymbol{\beta}^*}\Big\{\mathbb{P}_{\beta^*,X,Y}\big(T_{j,X}(Y)=0\big) : \beta^* \in \mathbb{R}^p,\ |\beta^*|_0 \leq s,\ |\beta_j^*| \geq \psi\Big\}.$$

where $\mathbb{P}_{\boldsymbol{\beta}^*,\mathbf{X},\mathbf{Y}}(\cdot)$ denotes the probability measure induced by $(\mathbf{X},\mathbf{Y})$ and the true parameters $\boldsymbol{\beta}^*$.

Theorem 8. *Suppose the conditions and assumptions in 6 hold, considering a sequence of design matrix $\mathbf{X} \in \mathbb{R}^{N\times p}$, for any $j \in [p]$ and $\alpha \in [0,1]$, then for test defined in* (36)*, we have*

$$\lim_{N,n\to\infty} \alpha_j(\widehat{T}_j) \leq \alpha, \tag{37}$$

$$\liminf_{N,n\to\infty} \frac{1-\varrho_j(\widehat{T}_j,\psi)}{1-\varrho_j(\psi)} \geq 1, \tag{38}$$

$$1-\varrho_j(\psi) = R\left(\alpha, \frac{\sqrt{N\mathbb{E}[f_{\varepsilon|\boldsymbol{X}}(0)]}\psi}{\sqrt{\tau(1-\tau)[\boldsymbol{H}^{-1}(\boldsymbol{\beta}^*)]_{j,j}}}\right), \tag{39}$$

where $R(\alpha,x) = \mathbb{P}(z_{1-\alpha/2} \leq |Z+x|) = 2-\Phi(z_{1-\alpha/2}+x)-\Phi(z_{1-\alpha/2}-x)$ for $Z \sim N(0,1)$,$z_{1-\alpha/2} = \Phi^{-1}(1-\alpha/2)$,and $x \in (0,\infty)$.

Theorem 8 establishes that the Type I error, $\alpha_j(\widehat{T}_j)$, is uniformly bounded above by the significance level α, while the statistical power, $1-\varrho_j(\widehat{T}_j,\psi)$, is bounded below by $1-\varrho_j(\psi)$. Similar theoretical guarantees for high-dimensional linear regression and quantile regression in a single-machine setting have been derived in [17] and [19].

VI. Comparison to the Competitors

We compare our DHSQR method with other three competitors, DREL, DPQR, and Avg-DC methods, whose definitions are provided in the simulation part of Section VII.

Space Aspects. The space complexity of the DHSQR method is provided in Section 2.3 of the main text. Similarly, all other algorithms exhibit the same spatial complexity of $\mathcal{O}\left(np+p^2\right)$. While our algorithm demonstrates strong performance, it remains comparable to other methods in terms of space complexity. We prioritize efficient storage utilization, avoiding unnecessary memory overhead compared to competing algorithms.

Computational Aspects. Compared to the Avg-DC method, our approach eliminates the need to solve non-smoothness quantile optimization on each machine. Instead, each local machine only calculates and communicates a p-dimensional vector (rather than a $p\times p$ matrix), while the central machine performs linear optimization with a Lasso penalty using a coordinate descent algorithm. Our method incurs lower communication costs. For instance, DREL [14] requires an additional round of communication for calculating and broadcasting the density function $\widehat{f}^{g,k}(0)$. Simulation results demonstrate that DHSQR outperforms DREL in terms of speed. Leveraging a Newton-type iteration, our method is a second-order algorithm, requiring fewer iterations to achieve convergence compared to gradient-based first-order algorithms like DPQR [34].

Theoretical Aspects. In addition, we wish to reiterate the theoretical innovations of our method, DHSQR. Unlike the DREL method, we relax the homoscedasticity assumption of the error term. In contrast to the Avg-DC and DPQR methods, we provide theoretical guarantees to support recovery and statistical inference.

TABLE I: Comparison of different methods.

Method	Convergence rate	Support recovery	Heterogeneity	Inference
DHSQR	✓	✓	✓	✓
DREL	✓	✓	×	×
DPQR	✓	×	✓	✓ (only low-dimension)
Avg-DC	✓	×	×	×

VII. Simulation Studies

A. Simulation Setup

In this section, we provide simulation studies to assess the performance of our DHSQR estimator. We generate synthetic data from the following linear models, corresponding to the homoscedastic error case (Model 1) and the heteroscedastic error case (Model 2):

- Model 1: $Y_i = \boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \varepsilon_i$;
- Model 2: $Y_i = \boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\beta}^* + (1+0.4x_{i1})\varepsilon_i$,

where $\boldsymbol{X}_i = (1, x_{i1}, \ldots, x_{ip})^{\mathrm{T}}$ is a p-dimensional vector and $(x_{i1},\ldots,x_{ip})$ is drawn from a multivariate normal distribution $N(\mathbf{0},\boldsymbol{\Sigma})$ with covariance matrix $\boldsymbol{\Sigma}_{ij} = 0.5^{|i-j|}$ for $1 \leq i,j \leq p$, the true parameter $\boldsymbol{\beta}^* = (1,1,2,3,4,5,\mathbf{0}_{p-5})^{\mathrm{T}}$. We fix the dimension $p = 500$, and consider three different values of τ, i.e., $\tau \in \{0.3, 0.5, 0.7\}$. We consider the following three noise distributions:

1) Normal distribution: the noise $\varepsilon_i \sim N(0,1)$;
2) t_3 distribution: the noise $\varepsilon_i \sim t(3)$;
3) Cauchy distribution: the noise $\varepsilon_i \sim Cauchy(0,1)$.

For the choice of the kernel function, we use the standard Gaussian kernel function that satisfies Assumption 1. For the global and local bandwidth, we set $h = 5(s\log p/n)^{1/3}$ and $b = 0.53(s\log p/n)^{1/3}$, respectively, according to the theoretical results in Theorem 1 and 2. The regularization parameters $\lambda_{N,g}$ are selected by validation. Specifically, we choose C_0 to minimize the check loss on the validation set. All the simulation results are the average of 100 independent experiments.

To evaluate the performance of our proposed method, we report the ℓ_2-error between the estimate and the true parameter. In addition, we calculate precision and recall defined as the proportion of correctly estimated positive (TP) in estimated positive (TP+FP) and the proportion of correctly estimated positive (TP) in true positive (TP+FN), which is used to show the support recovery accuracy (i.e., precision = 1 and recall = 0

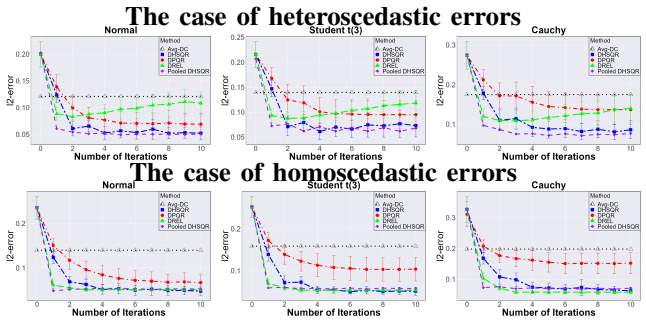

Fig. 1: The ℓ_2-error with an error bound between the true parameter and the estimated parameter versus the number of iterations with a fixed quantile level $\tau = 0.5$.

implies perfect support recovery). We also report the F_1-score defined as

$$F_1 = \frac{2}{\text{Recall}^{-1} + \text{Precision}^{-1}},$$

which is commonly used as an evaluation of support recovery. Note that F_1-score$= 1$ implies perfect support recovery. We compare the finite sample performance of the *DHSQR* estimator and the other four estimators:

(a) Averaged DC (*Avg-DC*) estimator which computes the ℓ_1-penalized QR estimators on the local machine and then combines the local estimators by taking the average;
(b) The distributed high-dimensional sparse quantile regression estimator on a single machine with pooled data defined in (7), which is denoted by *Pooled DHSQR*;
(c) Distributed robust estimator with Lasso (*DREL*), see in [14];
(d) Distributed penalty quantile regression estimator (*DPQR*) with convolution smoothing, see in [34], [55].

B. Effect of the number of iterations Under Heavy-Tailed Noise

We first show the effect of the number of iterations in our proposed method. We fix the total sample size $N = 20000$ and local sample size $n = 500$. We plot the ℓ_2-error from the true QR coefficients versus the number of iterations. Since the Avg-DC only requires one-shot communication, we use a horizontal line to show its performance. The results are shown in Figure 1.

From the result, our DHSQR estimator outperforms the Avg-DC and DPQR estimators in all the cases after a few iterations, and the estimated ℓ_2-error is very close to that of the pooled DHSQR estimator. Our method is robust for different noise settings, while DPQR performs poorly for heavy-tailed noise. Interestingly, for the heteroscedastic error case, the DREL estimator becomes unstable and fails to converge since it ignores the dependence between the error term and covariates. These results confirm our theoretical findings in Section II. For the rest of the experiments in this section, we fix the number of iterations $T = 10$.

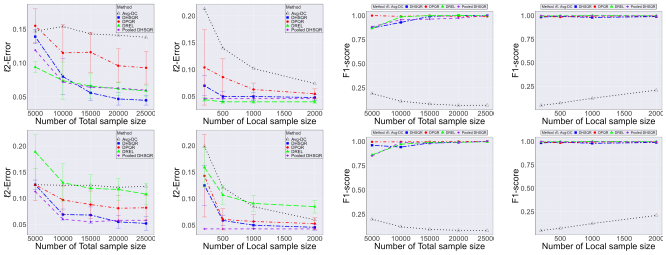

Fig. 2: The ℓ_2-error and F_1 score from the true parameter versus the number of total and local sample size with a fixed quantile level $\tau = 0.5$ and Normal error distribution.

C. Effect of Total Sample Size and Local Sample Size

In this section, we investigate the performance of our proposed estimator under varying total and local sample sizes. We only consider the setting with normal error case and quantile level $\tau = 0.5$. For the effect of total sample size, we fix the local sample size $n = 500$, and vary the total sample size $N \in \{5000, 10000, 15000, 20000, 25000\}$. For the effect of local sample size, we fix the total sample size $N = 20000$, and vary the local sample size $n \in \{200, 500, 1000, 2000\}$. We plot the ℓ_2-error and F_1-score of the five estimators under different models in Figure 2.

The ℓ_2-error of all methods decreases with an increase in the total sample size N and local sample size n, as shown in Figure 2. For DHSQR, DPQR, and DREL estimators, as N increases or n increases, the ℓ_2-error tends to decrease, and both of them outperform the Avg-DC estimator. We can see that the ℓ_2-error of DHSQR is very close to that of the Pooled DHSQR estimator in both homoscedastic and heteroscedastic error cases. The DHSQR estimator is better than other three distributed estimators. In the heteroscedastic error case, the DREL estimator's ℓ_2-error is significantly worse compared to DHSQR and DPQR. The Pooled DHSQR estimator's performance is not significantly affected by variations in n, while Avg-DC shows minimal sensitivity to the changes in N. In terms of support recovery, as Figure 2 is shown, both DHSQR, Pooled DHSQR, DPQR, and DREL estimators outperform the Avg-DC estimator in all settings, and their F_1 score are nearly equal to 1. It is also noteworthy that the Avg-DC estimator fails in support recovery since it is usually dense after averaging, especially when N is large and n is small. We also provide some additional experiment results using quantile level $\tau = \{0.3, 0.5, 0.7\}$. The results are reported in Tables II-VII(* indicates failure to converge).

D. Sensitivity Analysis for the Bandwidth

In this section, we study the sensitivity of the scaling constant in the bandwidth of the DHSQR estimator. Recall that the global bandwidth is $h = c_h(s \log p/n)^{1/3}$ and the local bandwidth is $b = c_b(s \log p/n)^{1/3}$ with constant $c_h, c_b > 0$ being the scaling constant. We fix the quantile level $\tau = 0.5$, the number of local sample size $n = 500$ and the number total sample size $N = 20000$, We vary the constant c_h, c_b from 1 to 10 and compute the F_1-score and the ℓ_2-error of the DHSQR estimator under the heteroscedastic error case. Due to space

TABLE II: The ℓ_2-error, precision, and recall of the DHSQR, Pooled DHSQR, DPQR, DREL, and Avg-DC estimator under different sample size N and local sample size n. Noises are generated from a normal distribution for the homoscedastic error case. The quantile level is fixed $\tau = 0.3$ and the iteration is fixed $T = 10$. (The standard deviation is given in parentheses.)

n		200			500			1000		
N		5000	10000	20000	5000	10000	20000	5000	10000	20000
DHSQR	Precision	0.951(0.100)	0.961(0.072)	0.967(0.082)	0.968(0.100)	0.966(0.066)	0.995(0.035)	0.941(0.141)	0.961(0.108)	1.000(0.001)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.126(0.077)	0.135(0.128)	0.089(0.067)	0.092(0.017)	0.077(0.025)	0.059(0.028)	0.097(0.022)	0.073(0.022)	0.054(0.013)
Pooled DHSQR	Precision	0.952(0.102)	0.974(0.055)	0.997(0.020)	0.947(0.101)	0.955(0.083)	0.997(0.020)	0.939(0.112)	0.964(0.071)	0.997(0.020)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.092(0.017)	0.072(0.016)	0.049(0.012)	0.091(0.016)	0.073(0.017)	0.049(0.012)	0.091(0.017)	0.073(0.022)	0.049(0.012)
DPQR	Precision	0.967(0.069)	0.949(0.097)	0.978(0.057)	0.917(0.132)	0.970(0.067)	0.986(0.048)	0.960(0.087)	0.986(0.048)	0.987(0.053)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.137(0.092)	0.141(0.119)	0.094(0.068)	0.101(0.020)	0.078(0.027)	0.061(0.030)	0.098(0.017)	0.076(0.011)	0.054(0.014)
DREL	Precision	0.970(0.081)	0.991(0.034)	0.994(0.028)	0.934(0.103)	0.986(0.043)	0.994(0.028)	0.925(0.105)	0.986(0.043)	1.000(0.001)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.095(0.017)	0.065(0.016)	0.049(0.009)	0.091(0.015)	0.064(0.014)	0.044(0.008)	0.088(0.016)	0.064(0.014)	0.043(0.008)
Avg-DC	Precision	0.046(0.006)	0.031(0.002)	0.025(0.001)	0.117(0.028)	0.061(0.011)	0.039(0.004)	0.235(0.086)	0.118(0.036)	0.065(0.012)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.232(0.016)	0.230(0.011)	0.227(0.008)	0.156(0.013)	0.153(0.010)	0.148(0.007)	0.117(0.014)	0.114(0.009)	0.107(0.007)

TABLE III: The ℓ_2-error, precision, and recall of the DHSQR, Pooled DHSQR, DPQR, DREL, and Avg-DC estimator under different sample size N and local sample size n. Noises are generated from a normal distribution for the heteroscedastic error case. The quantile level is fixed $\tau = 0.3$ and the iteration is fixed $T = 10$. (The standard deviation is given in parentheses.)

n		200			500			1000		
N		5000	10000	20000	5000	10000	20000	5000	10000	20000
DHSQR	Precision	0.966(0.072)	0.954(0.119)	0.971(0.084)	0.976(0.079)	0.971(0.058)	1.000(0.001)	0.931(0.119)	0.993(0.032)	0.993(0.032)
	Recall	1.000(0.030)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.193(0.254)	0.143(0.167)	0.128(0.131)	0.075(0.032)	0.067(0.026)	0.051(0.025)	0.070(0.024)	0.053(0.017)	0.052(0.015)
Pooled DHSQR	Precision	0.979(0.057)	0.996(0.023)	1.000(0.001)	0.973(0.068)	0.996(0.023)	1.000(0.001)	0.979(0.057)	0.996(0.023)	1.000(0.001)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.065(0.014)	0.057(0.021)	0.050(0.011)	0.062(0.016)	0.057(0.021)	0.049(0.011)	0.061(0.015)	0.057(0.021)	0.048(0.011)
DPQR	Precision	0.943(0.158)	0.948(0.109)	0.977(0.098)	0.937(0.086)	0.997(0.020)	1.000(0.001)	0.943(0.138)	0.978(0.057)	1.000(0.001)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.278(0.055)	0.196(0.048)	0.135(0.045)	0.201(0.012)	0.139(0.009)	0.094(0.008)	0.114(0.016)	0.081(0.033)	0.121(0.007)
DREL	Precision	*	0.977(0.086)	0.974(0.082)	0.982(0.048)	1.000(0.001)	0.996(0.023)	0.996(0.023)	0.996(0.023)	1.000(0.001)
	Recall	*	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	*	0.201(0.173)	0.195(0.168)	0.147(0.069)	0.124(0.065)	0.108(0.046)	0.115(0.043)	0.110(0.037)	0.090(0.028)
Avg-DC	Precision	0.053(0.009)	0.034(0.003)	0.026(0.002)	0.130(0.033)	0.068(0.012)	0.042(0.004)	0.266(0.105)	0.133(0.038)	0.072(0.013)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.175(0.012)	0.174(0.009)	0.170(0.008)	0.111(0.010)	0.108(0.008)	0.103(0.005)	0.086(0.011)	0.083(0.009)	0.077(0.005)

TABLE IV: The ℓ_2-error, precision, and recall of the DHSQR, Pooled DHSQR, DPQR, DREL, and Avg-DC estimator under different sample size N and local sample size n. Noises are generated from a normal distribution for the homoscedastic error case. The quantile level is fixed $\tau = 0.5$ and the iteration is fixed $T = 10$. (The standard deviation is given in parentheses.)

n		200			500			1000		
N		5000	10000	20000	5000	10000	20000	5000	10000	20000
DHSQR	Precision	0.901(0.147)	0.947(0.119)	0.969(0.077)	0.935(0.136)	0.901(0.159)	0.984(0.079)	0.913(0.177)	0.940(0.086)	0.961(0.101)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.131(0.064)	0.114(0.074)	0.070(0.038)	0.090(0.025)	0.080(0.023)	0.050(0.018)	0.088(0.025)	0.070(0.016)	0.050(0.016)
Pooled DHSQR	Precision	0.874(0.18)	0.943(0.095)	0.997(0.02)	0.903(0.158)	0.949(0.095)	0.991(0.034)	0.903(0.164)	0.944(0.099)	0.994(0.028)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.086(0.019)	0.066(0.016)	0.046(0.009)	0.086(0.019)	0.066(0.016)	0.046(0.009)	0.087(0.019)	0.066(0.016)	0.045(0.009)
DPQR	Precision	0.929(0.141)	0.956(0.103)	0.985(0.057)	0.935(0.139)	0.923(0.131)	0.970(0.080)	0.957(0.085)	0.918(0.118)	0.988(0.058)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.149(0.182)	0.119(0.118)	0.104(0.14)	0.098(0.045)	0.092(0.040)	0.086(0.068)	0.101(0.044)	0.077(0.029)	0.063(0.024)
DREL	Precision	0.922(0.144)	0.991(0.034)	0.991(0.034)	0.853(0.141)	0.991(0.034)	0.994(0.028)	0.842(0.122)	0.994(0.028)	1.000(0.001)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.094(0.017)	0.060(0.011)	0.044(0.008)	0.085(0.015)	0.061(0.011)	0.040(0.007)	0.084(0.015)	0.060(0.011)	0.040(0.007)
Avg-DC	Precision	0.043(0.005)	0.029(0.002)	0.025(0.001)	0.106(0.028)	0.058(0.009)	0.038(0.006)	0.196(0.075)	0.115(0.025)	0.066(0.011)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.219(0.014)	0.216(0.011)	0.213(0.007)	0.148(0.012)	0.144(0.01)	0.140(0.008)	0.112(0.013)	0.108(0.009)	0.102(0.007)

TABLE V: The ℓ_2-error, precision, and recall of the DHSQR, Pooled DHSQR, DPQR, DREL, and Avg-DC estimator under different sample size N and local sample size n. Noises are generated from a normal distribution for the heteroscedastic error case. The quantile level is fixed $\tau = 0.5$ and the iteration is fixed $T = 10$. (The standard deviation is given in parentheses.)

n		200			500			1000		
N		5000	10000	20000	5000	10000	20000	5000	10000	20000
DHSQR	Precision	0.901(0.160)	0.941(0.118)	0.979(0.061)	0.943(0.10)	0.936(0.129)	0.981(0.055)	0.961(0.100)	0.947(0.116)	0.978(0.072)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.180(0.133)	0.157(0.107)	0.125(0.075)	0.092(0.025)	0.081(0.027)	0.059(0.018)	0.088(0.022)	0.065(0.02)	0.050(0.014)
Pooled DHSQR	Precision	0.937(0.191)	0.983(0.047)	1.000(0.001)	0.933(0.110)	0.984(0.052)	1.000(0.001)	0.920(0.130)	0.980(0.050)	1.000(0.001)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.083(0.018)	0.058(0.015)	0.043(0.011)	0.083(0.018)	0.058(0.015)	0.043(0.011)	0.083(0.018)	0.057(0.016)	0.043(0.011)
DPQR	Precision	0.969(0.060)	0.936(0.128)	0.969(0.084)	0.970(0.060)	0.946(0.087)	0.978(0.073)	0.950(0.110)	0.943(0.105)	0.994(0.028)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.173(0.185)	0.165(0.172)	0.143(0.155)	0.155(0.112)	0.093(0.046)	0.061(0.04)	0.135(0.087)	0.079(0.036)	0.057(0.025)
DREL	Precision	0.939(0.120)	0.920(0.107)	0.967(0.080)	0.950(0.100)	0.968(0.073)	0.997(0.02)	0.921(0.110)	0.984(0.052)	0.997(0.020)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.192(0.110)	0.184(0.100)	0.159(0.074)	0.146(0.037)	0.127(0.06)	0.107(0.036)	0.119(0.034)	0.103(0.027)	0.091(0.031)
Avg-DC	Precision	0.049(0.010)	0.032(0.002)	0.025(0.001)	0.112(0.030)	0.062(0.011)	0.04(0.004)	0.221(0.080)	0.125(0.028)	0.071(0.013)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.203(0.017)	0.201(0.011)	0.199(0.008)	0.126(0.015)	0.125(0.011)	0.121(0.007)	0.092(0.016)	0.090(0.010)	0.085(0.008)

TABLE VI: The ℓ_2-error, precision, and recall of the DHSQR, Pooled DHSQR, DPQR, DREL, and Avg-DC estimator under different sample size N and local sample size n. Noises are generated from a normal distribution for the homoscedastic error case. The quantile level is fixed $\tau = 0.7$ and the iteration is fixed $T = 10$. (The standard deviation is given in parentheses.)

n		200			500			1000		
N		5000	10000	20000	5000	10000	20000	5000	10000	20000
DHSQR	Precision	0.943(0.105)	0.951(0.118)	0.978(0.072)	0.981(0.055)	0.962(0.096)	0.994(0.028)	0.956(0.11)	0.977(0.088)	0.994(0.028)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.140(0.092)	0.107(0.053)	0.084(0.063)	0.095(0.021)	0.072(0.026)	0.056(0.014)	0.095(0.020)	0.067(0.022)	0.053(0.011)
Pooled DHSQR	Precision	0.949(0.110)	0.955(0.102)	1.000(0.001)	0.956(0.096)	0.958(0.101)	1.000(0.001)	0.957(0.098)	0.959(0.102)	1.000(0.001)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.094(0.021)	0.067(0.018)	0.050(0.012)	0.094(0.021)	0.066(0.019)	0.049(0.011)	0.095(0.02)	0.065(0.018)	0.050(0.012)
DPQR	Precision	0.997(0.020)	1.000(0.001)	1.000(0.001)	0.994(0.028)	0.997(0.020)	1.000(0.001)	0.985(0.057)	0.986(0.048)	1.000(0.001)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.201(0.018)	0.144(0.011)	0.099(0.009)	0.140(0.015)	0.100(0.010)	0.068(0.007)	0.101(0.015)	0.074(0.013)	0.051(0.010)
DREL	Precision	0.951(0.083)	0.997(0.020)	0.997(0.020)	0.896(0.129)	0.997(0.020)	1.000(0.001)	0.901(0.107)	0.997(0.020)	1.000(0.001)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.098(0.021)	0.063(0.015)	0.050(0.011)	0.094(0.019)	0.062(0.014)	0.044(0.008)	0.09(0.018)	0.061(0.014)	0.044(0.008)
Avg-DC	Precision	0.048(0.006)	0.031(0.002)	0.025(0.001)	0.105(0.028)	0.063(0.011)	0.038(0.005)	0.218(0.069)	0.116(0.029)	0.065(0.013)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.234(0.015)	0.231(0.011)	0.226(0.008)	0.155(0.014)	0.151(0.011)	0.147(0.008)	0.119(0.015)	0.112(0.01)	0.107(0.007)

TABLE VII: The ℓ_2-error, precision, and recall of the DHSQR, Pooled DHSQR, DPQR, DREL, and Avg-DC estimator under different sample size N and local sample size n. Noises are generated from a normal distribution for the heteroscedastic error case. The quantile level is fixed $\tau = 0.7$ and the iteration is fixed $T = 10$. (The standard deviation is given in parentheses.)

n		200			500			1000		
N		5000	10000	20000	5000	10000	20000	5000	10000	20000
DHSQR	Precision	0.947(0.116)	0.948(0.100)	0.974(0.069)	0.965(0.098)	0.974(0.069)	0.997(0.020)	0.926(0.15)	0.989(0.081)	0.997(0.020)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.239(0.144)	0.174(0.054)	0.177(0.102)	0.169(0.033)	0.134(0.032)	0.117(0.021)	0.159(0.028)	0.122(0.022)	0.113(0.019)
Pooled DHSQR	Precision	0.964(0.102)	0.985(0.057)	1.000(0.001)	0.959(0.107)	0.983(0.064)	1.000(0.001)	0.974(0.069)	0.985(0.057)	1.000(0.001)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.152(0.024)	0.118(0.017)	0.107(0.017)	0.152(0.025)	0.118(0.017)	0.106(0.017)	0.153(0.024)	0.118(0.018)	0.107(0.017)
DPQR	Precision	0.994(0.028)	1.000(0.001)	1.000(0.001)	0.991(0.034)	1.000(0.001)	1.000(0.001)	0.980(0.050)	1.000(0.001)	1.000(0.001)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.329(0.013)	0.238(0.007)	0.171(0.007)	0.291(0.014)	0.210(0.007)	0.152(0.009)	0.222(0.012)	0.159(0.008)	0.118(0.011)
DREL	Precision	0.923(0.125)	0.976(0.067)	0.983(0.047)	0.934(0.083)	0.977(0.053)	1.000(0.001)	0.962(0.073)	0.977(0.053)	1.000(0.001)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.229(0.066)	0.207(0.068)	0.198(0.052)	0.184(0.032)	0.166(0.04)	0.153(0.036)	0.159(0.023)	0.15(0.024)	0.141(0.025)
Avg-DC	Precision	0.05(0.006)	0.032(0.002)	0.025(0.001)	0.114(0.025)	0.065(0.012)	0.040(0.005)	0.207(0.07)	0.117(0.029)	0.066(0.011)
	Recall	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
	ℓ_2-error	0.278(0.02)	0.275(0.015)	0.271(0.008)	0.187(0.019)	0.183(0.016)	0.181(0.008)	0.150(0.020)	0.144(0.016)	0.141(0.008)

TABLE IX: The ℓ_2-error, precision, and recall of the DHSQR, pooled DHSQR, DPQR, DREL, and Avg-DC estimator under different sample size N and local sample size n. Noises are generated from a normal distribution for the heteroscedastic error case. The quantile level is fixed $\tau = 0.7$ and the iteration is fixed $T = 10$. (The standard deviation is given in parentheses.)

N		5000	10000	20000
DHSQR	F_1-score	0.881(0.082)	0.893(0.037)	0.904(0.029)
	ℓ_2-error	0.159(0.021)	0.136(0.032)	0.125(0.031)
Pooled DHSQR	F_1-score	0.897(0.034)	0.901(0.023)	0.909(0.001)
	ℓ_2-error	0.144(0.020)	0.120(0.017)	0.102(0.016)
DREL	F_1-score	0.892(0.045)	0.906(0.015)	0.906(0.020)
	ℓ_2-error	0.181(0.035)	0.158(0.034)	0.152(0.039)
DPQR	F_1-score	0.878(0.071)	0.909(0.009)	0.909(0.001)
	ℓ_2-error	0.176(0.011)	0.146(0.002)	0.137(0.007)
Avg-DC	F_1-score	0.179(0.047)	0.105(0.023)	0.066(0.007)
	ℓ_2-error	0.184(0.018)	0.182(0.013)	0.180(0.009)

limitations, we report the Normal noise case as an example. The results are shown in Table VIII.

TABLE VIII: F_1-score and ℓ_2-error of DHSQR under different bandwidth constants c_h, c_b. (The standard deviation is given in parentheses.)

	$c_h = 1$		$c_h = 2$		$c_h = 5$		$c_h = 10$	
	F_1-score	ℓ_2-error	F_1-score	ℓ_2-error	F_1-score	ℓ_2-error	F_1-score	ℓ_2-error
$c_b = 1$	0.991 (0.032)	0.077 (0.032)	0.991 (0.032)	0.077 (0.032)	0.991 (0.032)	0.077 (0.032)	0.991 (0.032)	0.077 (0.032)
$c_b = 2$	0.951 (0.068)	0.260 (0.081)	0.951 (0.068)	0.260 (0.081)	0.951 (0.068)	0.260 (0.081)	0.951 (0.068)	0.260 (0.081)
$c_b = 5$	0.996 (0.023)	1.856 (1.084)	0.996 (0.023)	1.856 (1.084)	0.996 (0.023)	1.856 (1.084)	0.996 (0.023)	1.856 (1.084)
$c_b = 10$	1.000 (0.020)	5.666 (1.399)	1.000 (0.020)	5.666 (1.399)	1.000 (0.020)	5.666 (1.399)	1.000 (0.020)	5.666 (1.399)

Table VIII shows that our estimator DHSQR is sensitive only to the local bandwidth b and not to the global bandwidth. Therefore, even under a suboptimal choice of the global bandwidth constant c_h, the distributed REL still achieves a small ℓ_2-error and high support recovery accuracy. These results further suggest selecting a smaller local bandwidth constant c_b.

E. Experiments for the Decaying Sequence Setting of Nonzero Parameters

In this section, we provide some additional experiment results using the decaying sequence setting of nonzero parameters. We consider the heteroscedastic error case with normal distribution, and $\tau = 0.7, n = 500, T = 10$. Here, we set the true parameter as

$$\boldsymbol{\beta}^* = (1, 2^1, 2^0, 2^{-1}, 2^{-2}, 2^{-3}, \mathbf{0}_{p-5})^{\mathrm{T}}.$$

Other settings align with those in Section VII. The average results from 100 replicates are summarized in Table IX.

As depicted in Table IX, our DHSQR method consistently outperforms other methods across all sample sizes under the decaying sequence setting. DHSQR demonstrates superior performance in terms of ℓ_2-error, indicating its capability to provide more accurate estimations compared to alternative methods. Additionally, the DHSQR method achieves an indistinguishable F_1-score when compared to other iteration methods, reaffirming its strength and reliability.

TABLE X: The F_1 score, ℓ_2 error, and Time computation of five estimators under different total sample sizes N. (The standard deviation is given in parentheses.)

N	DHSQR			Pooled DHSQR		
	F_1-score	ℓ_2-error	Time	F_1-score	ℓ_2-error	Time
5000	0.961(0.121)	0.126(0.018)	0.122(0.029)	0.851(0.132)	0.113(0.032)	0.877(0.115)
10000	0.941(0.074)	0.069(0.021)	0.130(0.024)	0.991(0.032)	0.060(0.014)	1.671(0.080)
15000	0.984(0.043)	0.068(0.030)	0.143(0.017)	0.981(0.072)	0.055(0.015)	2.500(0.094)
20000	0.991(0.031)	0.055(0.024)	0.151(0.020)	0.991(0.017)	0.058(0.016)	3.393(0.140)
N	DPQR			DREL		
	F_1-score	ℓ_2-error	Time	F_1-score	ℓ_2-error	Time
5000	0.996(0.001)	0.126(0.061)	0.193(0.033)	0.862(0.152)	0.189(0.065)	1.006(0.095)
10000	0.996(0.002)	0.097(0.056)	0.189(0.010)	0.972(0.621)	0.131(0.072)	1.768(0.034)
15000	0.997(0.002)	0.088(0.052)	0.208(0.020)	0.991(0.052)	0.119(0.053)	2.622(0.166)
20000	1.000(0.001)	0.081(0.050)	0.223(0.030)	0.992(0.053)	0.117(0.042)	3.471(0.126)
N	Avg-DC					
	F_1-score	ℓ_2-error	Time			
5000	0.199(0.052)	0.126(0.015)	31.780(2.884)			
10000	0.122(0.022)	0.125(0.011)	59.576(5.125)			
15000	0.094(0.011)	0.126(0.008)	93.553(7.594)			
20000	0.084(0.011)	0.121(0.007)	117.129(11.709)			

F. Computation Time Comparison

We further study the computation efficiency of our proposed estimator. We fix the local sample size $n = 500$ and vary the total sample size N. In Table X, we report the F_1 score, ℓ_2 error, and computation time (per iteration) with different the total sample sizes N under the heteroscedastic error case. As is shown in Table X, We can see that our DHSQR has the fastest single iteration time, followed by DPQP, which maintains the same order of magnitude, then DREL and Pooled DHSQR, REL, and Pooled DHSQR maintain the same order of magnitude, and Avg-DC is the slowest.

G. Distribued Inference Performance

In this section, we evaluate the effectiveness of the debiased DHSQR estimator across different scenarios. We present the distribution of test statistics under the null hypothesis, quantile-quantile (Q-Q) plots, empirical cumulative distribution functions (CDFs) of p-values, and power curves for hypothesis testing. Data is generated from both homoscedastic (Model 1) and heteroscedastic (Model 2) cases with various error distributions. For simplicity, we fix the total sample size $N = 20000$, the number of machines $m = 20$ and vary the quantile level $\tau = \{0.3, 0.5, 0.7\}$,. For DHSQR point estimation, we implement Algorithm 1 with $T = 10$ iterations.

In Figure 3, we present the distributions of $z_j = \frac{\sqrt{N}(\widetilde{\beta}_{T_0,h,j} - \beta_j^*)}{\sqrt{\tau(1-\tau)[\widehat{\mathbf{W}}_b^{(1)}\widehat{\boldsymbol{\Sigma}}^{(1)}\widehat{\mathbf{W}}_b^{(1)}]_{j,j}^{1/2}}}$ for $j \in \{2, 5, 500\}$ under the null hypothesis across various scenarios. These histograms reveal that z_j closely approximates a standard normal distribution, regardless of noise type, even when the stochastic error follows an extremely heavy-tailed distribution. Our test statistics

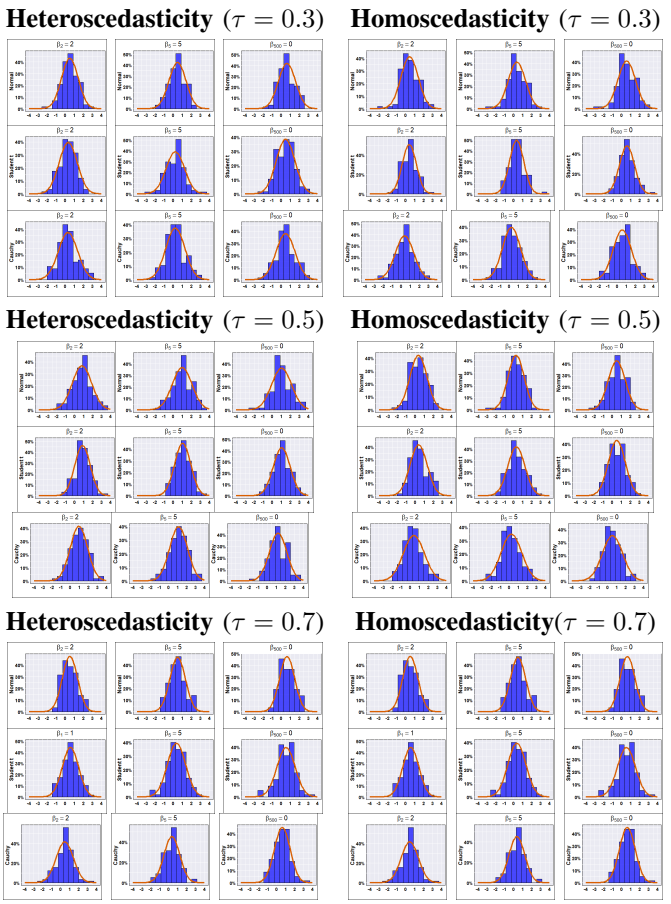

Fig. 3: Histograms of the standardized test estimator $\frac{\sqrt{n}(\widetilde{\beta}_j - \beta_j^*)}{\sqrt{\tau(1-\tau)}[\widehat{\mathbf{W}}_b^{(1)}\widehat{\mathbf{\Sigma}}^{(1)}\widehat{\mathbf{W}}_b^{(1)}]_{j,j}^{1/2}}$ at $\tau = \{0.3, 0.5, 0.7\}$, under different noises ($m = 20$, $N = 20000$). Rows correspond to Normal, t, and Cauchy noises, columns correspond to $\beta_2 = 2, \beta_5 = 5, \beta_{500} = 0$.

,

exhibit consistent performance for both large and small signal strengths. The Q-Q plots in Figure 4 further illustrate the relationship between the sample quantiles of z_j and the quantiles of the standard normal distribution. The close alignment of points with the line $y = x$ confirms the asymptotic normality of our test statistics, supporting the theoretical results established in Section V. It is evident that both the sparse coefficient and non-coefficient coordinate debiased estimates exhibit normal-like properties across different quantile levels.

Figure 5 illustrates the empirical cumulative distribution functions (CDFs) of the p-values for z_{20}, focusing specifically on variables outside the support set. As theoretically anticipated, these p-values follow a nearly uniform distribution, confirming the validity of our hypothesis testing procedure. When examining different error distributions, we observe that while the empirical CDFs closely track the theoretical diagonal line under normal errors, there are slight deviations under heavy-tailed conditions such as t_3 and Cauchy distributions. Nevertheless, these empirical CDFs remain reasonably close to the expected uniform distribution, demonstrating the robustness of our inference method even in challenging scenarios with extreme noise.

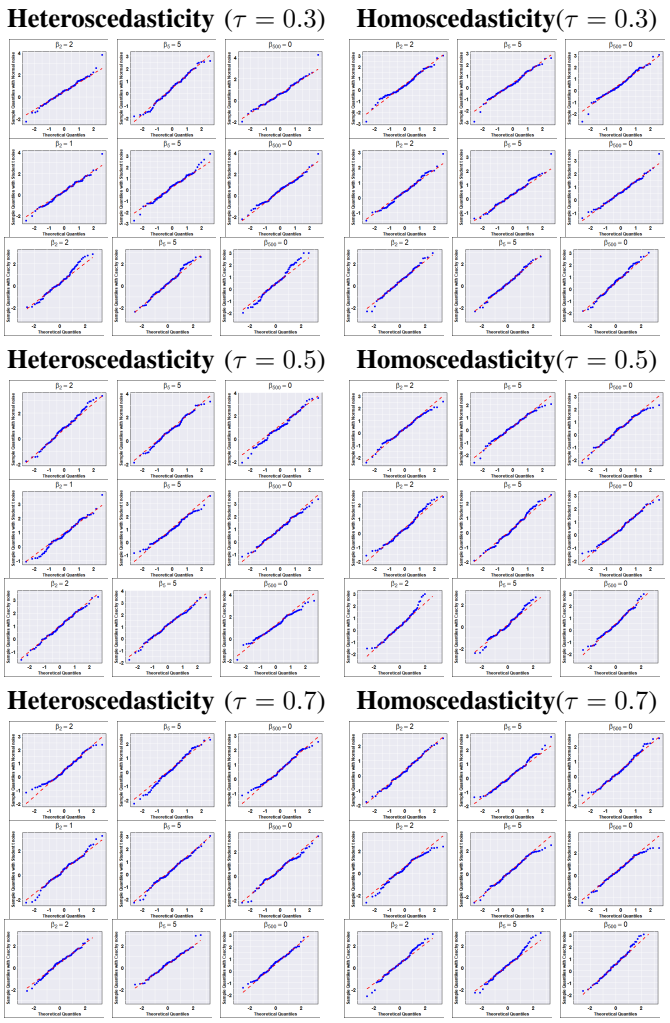

Fig. 4: QQ plots of the standardized test estimator $\frac{\sqrt{n}\left(\widetilde{\beta}_j - \beta_j^*\right)}{\sqrt{\tau(1-\tau)}\left[\widehat{\mathbf{W}}_b^{(1)}\widehat{\mathbf{\Sigma}}^{(1)}\widehat{\mathbf{W}}_b^{(1)}\right]_{j,j}^{1/2}}$ under at $\tau = \{0.3, 0.5, 0.7\}$, under different noises ($m = 20$, $N = 20000$). Rows correspond to Normal, t, and Cauchy noises, columns correspond to $\beta_2 = 2, \beta_5 = 5, \beta_{500} = 0$.

We also present power curves for testing the null hypotheses $H_{0,1} : \beta_1 = 1$ in Figure 6 and $H_{0,100} : \beta_{100} = 0$ in Figure 7 across different quantile levels $\tau = 0.5$ and under both homoscedastic and heteroscedastic error settings. As shown in the figures, the tests achieve high power within a narrow range of the parameter values. While the power curves rise rapidly under $N(0, 1)$ noise, those under heavy-tailed distributions exhibit a noticeable delay, reflecting the additional challenges posed by extreme noise conditions.

VIII. Real Data Analysis

In this section, we employ the proposed DHSQR algorithm to analyze the drug sensitivity data of the Human Immunodeficiency Virus (HIV) [56], [57]. This data is sourced from the Stanford University HIV Drug Resistance Database (http://hivdb.stanford.edu). Efavirenz (EFV) is the preferred first-line antiretroviral drug for HIV and belongs to the category of selective non-nucleoside reverse transcriptase inhibitors

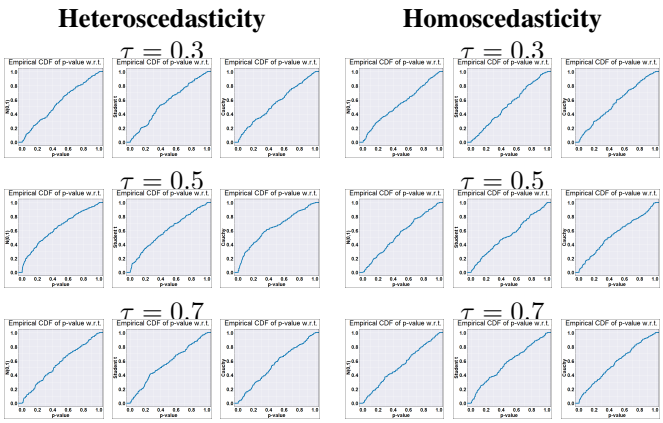

Fig. 5: Empirical CDF of p-values w.r.t. z_{20} (restricted to entries out of the support set) under Normal, t, and Cauchy noises for Heteroscedastic and Homoscedastic cases with $(N, m, p) = (20000, 40, 500)$ at $\tau = \{0.3, 0.5, 0.7\}$.

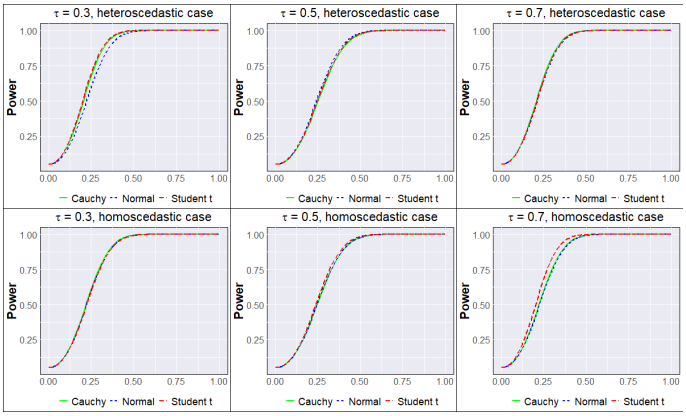

Fig. 6: Power curves of testing the null $H_{0,1} : \beta_1^* = 1$ under Normal, t, and Cauchy noises with $(N, m, p) = (20000, 40, 500)$.

(NNRTIs) for subtype 1 HIV. We investigate the impact of mutations at different positions on EFV drug sensitivity. After excluding some missing records, our initial dataset comprises $N = 2046$ HIV isolates and $p = 201$ viral mutation positions. We define the response variable y for regression as the drug sensitivity of the samples, which quantifies the fold reduction in susceptibility of a single virus isolate compared to a control isolate [57]. The covariate x indicates whether the virus has mutated at different positions ($x = 1$ for mutation, $x = 0$ for no mutation). Due to the heavy-tailed distribution of the response variable, we apply a logarithmic transformation, transforming it into $\log_{10} y$. Histograms of both the initial response variable and the transformed response variable are shown in Figure 8. Even after the transformation, the response variable remains non-normally distributed. Therefore, we employ quantile regression for data analysis, which is more robust than linear regression. Additionally, it's important to note that while the dataset is not large in size, sensitivity data are typically distributed across different hospitals in practical scenarios, making data aggregation challenging. In such cases, distributed quantile regression algorithms become a suitable

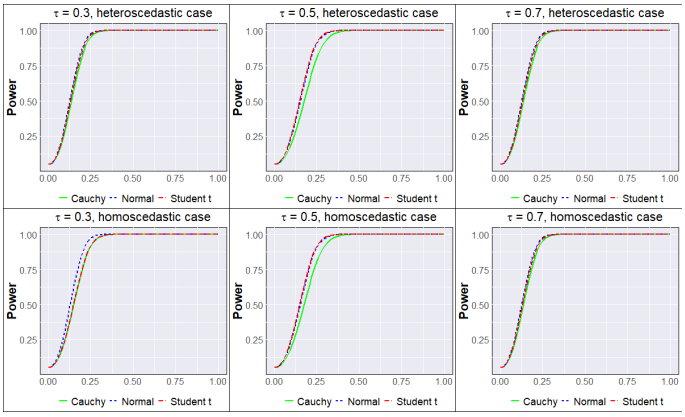

Fig. 7: Power curves of testing the null $H_{0,100} : \beta_{100}^* = 0$ under Normal, t, and Cauchy noises with $(N, m, p) = (20000, 40, 500)$.

choice.

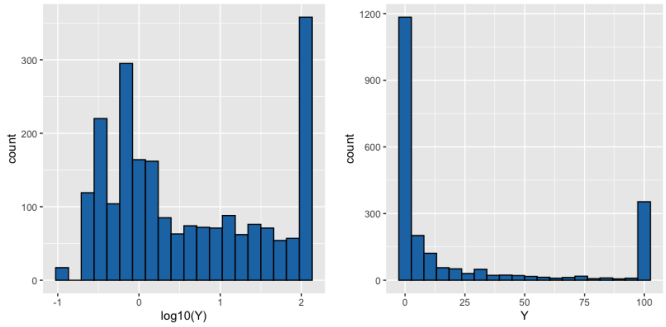

Fig. 8: The left figure represents the histogram of the initial drug sensitivity variable distribution, while the right figure represents the histogram of the drug sensitivity variable distribution after undergoing a logarithmic transformation.

In the experimental setup, we randomly selected a training dataset with a sample size of $N_{tr} = 1500$, a validation dataset with a sample size of $N_{va} = 300$ to select the optimal penalty parameter λ, and the remaining data served as the test dataset, which had a sample size of $N_{te} = 246$. Typically, our interest lies in drug sensitivity at higher quantile levels since it is associated with stronger resistance [57], enabling the development of better treatment strategies. Therefore, in the experiments, we chose quantile levels $\tau \in 0.5, 0.75, 0.9$. We employed the same comparative methods as in Section VII to evaluate our algorithm. To assess the performance of each estimator, we computed the Predicted Quantile Error (PQE) on the prediction dataset, defined as

$$\text{PQE} = \sum_{i=1}^{n_{te}} \rho_\tau(y_i - \widehat{y}_i), \quad i = 1, \ldots, 246,$$

where $\widehat{y}_i$ represents the predictions made by each model. The results of the five different methods at quantile levels $\tau \in \{0.5, 0.75, 0.9\}$ are presented in Table XI. From the results, it can be observed that our DHSQR estimator exhibits smaller prediction errors compared to other distributed estimators and is very close to the performance of the single ma-

chine estimator, Pooled DHSQR. This further underscores the method's excellent performance on HIV data. It is noteworthy that the performance of the DREL estimator is even worse than the one-step Avg-DC estimator, indicating the presence of heteroscedasticity in the data. Clearly, the DREL algorithm is not suitable for analyzing this type of data.

TABLE XI: The results of the Predicted Quantile Error (PQE) for different estimators at quantile levels $\tau \in \{0.5, 0.75, 0.9\}$.

Quantile Level	DHSQR	Pooled DHSQR	DREL	DPQR	Avg-DC
$\tau = 0.5$	0.205	0.201	0.241	0.226	0.225
$\tau = 0.75$	0.193	0.189	0.278	0.213	0.211
$\tau = 0.9$	0.176	0.172	0.329	0.186	0.232

Furthermore, to analyze the performance differences of various distributed algorithms in variable selection, we present the results of their predictions of non-zero coefficients at quantile levels $\tau \in \{0.5, 0.75, 0.9\}$ in Table XII. We observed that, apart from the Avg-DC algorithm, all the multi-round communication-based distributed algorithms were able to select fewer variables than the total number, $p = 201$. As the quantile level increases, the DHSQR algorithm selects an increasing number of non-zero coefficients, going from 24 to 35. This suggests that more virus positions have an impact on resistance at higher quantile levels.

TABLE XII: The results of the number of selected non-zero coefficients for different estimators at quantile levels $\tau \in \{0.5, 0.75, 0.9\}$.

Quantile Level	DHSQR	Pooled DHSQR	DREL	DPQR	Avg-DC
$\tau = 0.5$	24	24	26	24	41
$\tau = 0.75$	29	29	31	29	63
$\tau = 0.9$	35	35	36	35	81

IX. Conclusion

In this paper, we propose an efficient distributed quantile regression method for dealing with heterogeneous data. By constructing a kernel-based pseudo covariate and response, we transform the non-smooth quantile regression problem into a smooth least squares problem. Based on this procedure, we establish a double-smoothing surrogate likelihood framework to facilitate distributed learning. An efficient algorithm is developed, which enjoys computation and communication efficiency. We also investigate the inference problem in distributed high-dimensional quantile regression based on the debiased procedure. Confidence intervals and hypothesis tests are constructed for the quantile regression coefficients. Theoretically, for the distributed estimation, we provide the convergence rate and support recovery of the proposed DHSQR estimator. For the distributed inference, we establish the Bahadur representation, the non-asymptotic Berry-Esseen bound, and the asymptotic normality of the debiased DHSQR estimator, which guarantee the validity of confidence interval construction and hypothesis testing. The empirical studies demonstrate the effectiveness of the proposed method in terms of estimation and inference. Future work includes developing decentralized algorithms for estimation and inference in quantile regression over networked data, extending prior approaches to online quantile estimation and sensor-network frameworks [58], [59].

References

[1] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[2] Y. Gao, W. Liu, H. Wang, X. Wang, Y. Yan, and R. Zhang, "A review of distributed statistical inference," *Statistical Theory and Related Fields*, vol. 6, no. 2, pp. 89–99, 2022.

[3] P. Zhao and B. Yu, "On model selection consistency of lasso," *Journal of Machine Learning Research*, vol. 7, pp. 2541–2563, 2006.

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

[5] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical learning with sparsity: the lasso and generalizations*. CRC press, 2015.

[6] Y. Zhang, J. C. Duchi, and M. J. Wainwright, "Communication-efficient algorithms for statistical optimization," *Journal of Machine Learning Research*, vol. 14, no. 68, pp. 3321–3363, 2013.

[7] J. D. Lee, Q. Liu, Y. Sun, and J. E. Taylor, "Communication-efficient sparse regression," *Journal of Machine Learning Research*, vol. 18, no. 1, pp. 115–144, 2017.

[8] M. I. Jordan, J. D. Lee, and Y. Yang, "Communication-efficient distributed statistical inference," *Journal of the American Statistical Association*, vol. 114, no. 526, pp. 668–681, 2019.

[9] R. Koenker and G. Bassett Jr, "Regression quantiles," *Econometrica*, vol. 46, no. 1, pp. 33–50, 1978.

[10] R. Koenker, *Quantile regression*. Cambridge University Press, 2005, vol. 38.

[11] X. Feng, Q. Liu, and C. Wang, "A lack-of-fit test for quantile regression process models," *Statistics & Probability Letters*, vol. 192, p. 109680, 2023.

[12] X. He, X. Pan, K. M. Tan, and W.-X. Zhou, "Smoothed quantile regression with large-scale inference," *Journal of Econometrics*, vol. 232, no. 2, pp. 367–388, 2023.

[13] C. Wang, T. Li, X. Zhang, X. Feng, and X. He, "Communication-efficient nonparametric quantile regression via random features," *Journal of Computational and Graphical Statistics*, vol. 33, no. 4, pp. 1175–1184, 2024.

[14] X. Chen, W. Liu, X. Mao, and Z. Yang, "Distributed high-dimensional regression under a quantile loss function," *Journal of Machine Learning Research*, vol. 21, no. 1, pp. 7432–7474, 2020.

[15] C.-H. Zhang and S. S. Zhang, "Confidence intervals for low dimensional parameters in high dimensional linear models," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 76, no. 1, pp. 217–242, 2014.

[16] S. A. van de Geer, P. Buhlmann, Y. Ritov, and R. Dezeure, "On asymptotically optimal confidence regions and tests for high-dimensional models," *Annals of Statistics*, vol. 42, pp. 1166–1202, 2013.

[17] A. Javanmard and A. Montanari, "Confidence intervals and hypothesis testing for high-dimensional regression," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 2869–2909, 2014.

[18] J. Bradic and M. Kolar, "Uniform inference for high-dimensional quantile regression: linear functionals and regression rank scores," *arXiv preprint arXiv:1702.06209*, 2017.

[19] Y. Yan, X. Wang, and R. Zhang, "Confidence intervals and hypothesis testing for high-dimensional quantile regression: Convolution smoothing and debiasing," *Journal of Machine Learning Research*, vol. 24, no. 245, pp. 1–49, 2023.

[20] M. Neykov, J. S. Liu, and T. Cai, "ℓ_1-regularized least squares for support recovery of high dimensional single index models with Gaussian designs," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2976–3012, 2016.

[21] A. Belloni and V. Chernozhukov, "ℓ_1-penalized quantile regression in high-dimensional sparse models," *The Annals of Statistics*, vol. 39, no. 1, pp. 82–130, 2011.

[22] C. Wang and Z. Shen, "Distributed high-dimensional quantile regression: Estimation efficiency and support recovery," in *International Conference on Machine Learning*. PMLR, 2024, pp. 51 415–51 441.

[23] R. Li, D. K. Lin, and B. Li, "Statistical inference in massive data sets," *Applied Stochastic Models in Business and Industry*, vol. 29, no. 5, pp. 399–409, 2013.

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

[25] Y. Zhang, J. Duchi, and M. Wainwright, "Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 3299–3340, 2015.

[26] C. Huang and X. Huo, "A distributed one-step estimator," *Mathematical Programming*, vol. 174, pp. 41–76, 2019.

[27] O. Shamir, N. Srebro, and T. Zhang, "Communication-efficient distributed optimization using an approximate newton-type method," in *International Conference on Machine Learning*. PMLR, 2014, pp. 1000–1008.

[28] J. Wang, M. Kolar, N. Srebro, and T. Zhang, "Efficient distributed learning with sparsity," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3636–3645.

[29] J. Fan, Y. Guo, and K. Wang, "Communication-efficient accurate statistical estimation," *Journal of the American Statistical Association*, vol. 116, pp. 1–11, 2021.

[30] T. Zhao, M. Kolar, and H. Liu, "A general framework for robust testing and confidence regions in high-dimensional quantile regression," *arXiv preprint arXiv:1412.8724*, 2014.

[31] Q. Xu, C. Cai, C. Jiang, F. Sun, and X. Huang, "Block average quantile regression for massive dataset," *Statistical Papers*, vol. 61, no. 1, pp. 141–165, 2020.

[32] L. Chen and Y. Zhou, "Quantile regression in big data: A divide and conquer based strategy," *Computational Statistics & Data Analysis*, vol. 144, p. 106892, 2020.

[33] M. Fernandes, E. Guerre, and E. Horta, "Smoothing quantile regressions," *Journal of Business & Economic Statistics*, vol. 39, no. 1, pp. 338–357, 2021.

[34] K. M. Tan, H. Battey, and W.-X. Zhou, "Communication-constrained distributed quantile regression with optimal statistical guarantees," *Journal of Machine Learning Research*, vol. 23, pp. 1–61, 2022.

[35] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.

[36] C.-H. Zhang, "Nearly unbiased variable selection under minimax concave penalty," *The Annals of Statistics*, vol. 38, no. 2, pp. 894–942, 2010.

[37] A. Belloni, V. Chernozhukov, and K. Kato, "Uniform post-selection inference for least absolute deviation regression and other z-estimation problems," *Biometrika*, vol. 102, no. 1, pp. 77–94, 2015.

[38] A. Belloni, V. Chernozhukov, D. Chetverikov, and I. Fernández-Val, "Conditional quantile processes based on series or many regressors," *Journal of Econometrics*, vol. 213, no. 1, pp. 4–29, 2019.

[39] C. Cheng, X. Feng, J. Huang, and X. Liu, "Regularized projection score estimation of treatment effects in high-dimensional quantile regression," *Statistica Sinica*, vol. 32, no. 1, pp. 23–41, 2022.

[40] Y. Gu and H. Zou, "Sparse composite quantile regression in ultrahigh dimensions with tuning parameter calibration," *IEEE Transactions on Information Theory*, vol. 66, no. 11, pp. 7132–7154, 2020.

[41] W. Liu, X. Mao, X. Zhang, and X. Zhang, "Efficient sparse least absolute deviation regression with differential privacy," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 2328–2339, 2024.

[42] W. Zhao, F. Zhang, and H. Lian, "Debiasing and distributed estimation for high-dimensional quantile regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 7, pp. 2569–2577, 2019.

[43] M. Schmidt, "Graphical model structure learning with l1-regularization," *University of British Columbia*, 2010.

[44] S. Solntsev, J. Nocedal, and R. H. Byrd, "An algorithm for quadratic ℓ_1-regularized optimization with a flexible active-set strategy," *Optimization Methods and Software*, vol. 30, no. 6, pp. 1213–1237, 2015.

[45] S. J. Wright, "Coordinate descent algorithms," *Mathematical Programming*, vol. 151, no. 1, pp. 3–34, 2015.

[46] X. Chen, W. Liu, and Y. Zhang, "Quantile regression under memory constraint," *The Annals of Statistics*, vol. 47, no. 6, pp. 3244–3273, 2019.

[47] Y. Bao and W. Xiong, "One-round communication efficient distributed M-estimation," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 46–54.

[48] J. Fan, Y. Fan, and E. Barut, "Adaptive robust variable selection," *The Annals of Statistics*, vol. 42, no. 1, pp. 324–351, 2014.

[49] Y. Zhang and L. Xiao, "Communication-efficient distributed optimization of self-concordant empirical loss," in *Large-Scale and Distributed Optimization*. Springer, 2018, pp. 289–341.

[50] T. Cai and W. Liu, "Adaptive thresholding for sparse covariance matrix estimation," *Journal of the American Statistical Association*, vol. 106, no. 494, pp. 672–684, 2011.

[51] A. Belloni, V. Chernozhukov, and K. Kato, "Valid post-selection inference in high-dimensional approximately sparse quantile regression models," *Journal of the American Statistical Association*, vol. 114, no. 526, pp. 749–758, 2019.

[52] T. Cai, W. Liu, and X. Luo, "A constrained ℓ_1 minimization approach to sparse precision matrix estimation," *Journal of the American Statistical Association*, vol. 106, no. 494, pp. 594–607, 2011.

[53] X. Li, T. Zhao, X. Yuan, and H. Liu, "The flare package for high dimensional linear regression and precision matrix estimation in r," *Journal of Machine Learning Research*, 2015.

[54] A. Belloni, V. Chernozhukov, and Y. Wei, "Post-selection inference for generalized linear models with many controls," *Journal of Business & Economic Statistics*, vol. 34, no. 4, pp. 606–619, 2016.

[55] R. Jiang and K. Yu, "Smoothing quantile regression for a distributed system," *Neurocomputing*, vol. 466, pp. 311–326, 2021.

[56] S.-Y. Rhee, M. J. Gonzales, R. Kantor, B. J. Betts, J. Ravela, and R. W. Shafer, "Human immunodeficiency virus reverse transcriptase and protease sequence database." *Nucleic Acids Research*, vol. 31, no. 1, pp. 298–303, 2003.

[57] A. Hu, C. Li, and J. Wu, "Communication-efficient modeling with penalized quantile regression for distributed data," *Complexity*, vol. 2021, pp. 1–16, 1 2021.

[58] H. Wang and C. Li, "Distributed quantile regression over sensor networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 2, pp. 338–348, 2017.

[59] D. Yuan, A. Proutiere, and G. Shi, "Distributed online linear regressions," *IEEE Transactions on Information Theory*, vol. 67, no. 1, pp. 616–639, 2020.