ROBUST STATISTICAL LEARNING FROM STREAMING DATA: AN ADAPTIVE TRANSFER PERSPECTIVE

Anonymous authors
Paper under double-blind review

ABSTRACT

Statistical online learning is challenging when streaming batches may follow different models. This paper studies a contamination setting in which each batch follows a central model with high probability, and a batch-specific model otherwise. To address this online heterogeneity, we approach the problem from a transfer learning perspective. Starting with a point estimation example, we first develop a robust one-pass algorithm that simultaneously updates an estimate of the central parameter, and estimates the parameter of each current batch. We then extend this online method to a linear regression problem and establish time-uniform error bounds: When the current batch follows the central model, the updated central estimator becomes increasingly accurate until reaching a contamination-dependent error floor. When the current batch follows a heterogeneous model, the proposed method adaptively transfers information from previous estimates whenever beneficial, yielding a batch-specific estimator that is potentially more accurate than purely local estimation. The method also provides an online heterogeneity detection procedure. Numerical experiments demonstrate the benefits of our robust online method for learning both central and batch-specific models.

1 INTRODUCTION

This paper studies online statistical learning from a sequence of potentially heterogeneous data batches. Let Θ be the parameter space and $\{P_{\boldsymbol{\beta}} : \boldsymbol{\beta} \in \Theta\}$ be a parameterized family of distributions. At each round $b \geq 1$, a batch $D^{(b)} = \{Z_i^{(b)}\}_{1 \leq i \leq n}$ of n observations arrives. Conditional on the batch parameter $\boldsymbol{\beta}^{(b)*}$, each observation $Z_i^{(b)}$ within this batch is independent and identically distributed from $P_{\boldsymbol{\beta}^{(b)*}}$. We assume that the first batch is a trusted reference batch satisfying $\boldsymbol{\beta}^{(1)*} = \boldsymbol{\beta}^*$, where $\boldsymbol{\beta}^* \in \Theta$ is the central parameter. For every subsequent batch $b \geq 2$, let

$$\boldsymbol{\beta}^{(b)*} = (1 - A_b)\boldsymbol{\beta}^* + A_b \boldsymbol{\theta}^{(b)*}, \qquad A_b \overset{\text{iid}}{\sim} \text{Bernoulli}(\epsilon),$$

where $\{A_b\}_{b \geq 2}$ are the contamination labels and $\epsilon \in [0, 1/2)$. Here $\boldsymbol{\theta}^{(b)*} \in \Theta$ is a fixed, batch-specific parameter that differs from $\boldsymbol{\beta}^*$. Thus, a subsequent batch follows the central model with probability $1 - \epsilon$ and a heterogeneous model with probability ϵ [1]. The central parameter $\boldsymbol{\beta}^*$ characterizes the predominant behavior across batches, while the batch-specific parameters $\boldsymbol{\theta}^{(b)*}$ describe departures from this central structure. This setting provides a stylized representation of batch-level parameter heterogeneity arising in federated learning with heterogeneous clients (Li et al., 2020), distributed systems with unreliable machines (Zhang et al., 2020), sequential learning under dynamic batch-level contamination (Dorfman et al., 2024), and so on.

Without storing or revisiting historical observations, our goals are to: (i) continuously learn the central model $P_{\boldsymbol{\beta}^*}$; and (ii) adaptively learn the specific model $P_{\boldsymbol{\beta}^{(b)*}}$ in each current batch (if necessary). To this end, we develop a robust online procedure that uses the preceding central estimate as source information for the current batch, while incorporating the current batch into the update of the central estimate. This bidirectional transfer mechanism combines robust estimation with ideas from transfer learning and multi-task learning.

[1] The quantity ϵ should be interpreted as the probability, or expected fraction, of contaminated batches, rather than as a deterministic bound on the fraction of contaminated batches in some specific time horizon.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1.1 Related work

Contamination models. Robust statistics has developed bounded-influence methods and minimax benchmarks for learning under contaminated observations (Huber, 1964; Chen et al., 2018; Gao, 2020; Pensia et al., 2025; Diakonikolas et al., 2026). This line of research has also been extended to batch-level contamination in Byzantine distributed learning, where a fraction of machines or batches may return corrupted outputs (Lamport et al., 1982; Su & Vaidya, 2016; Chen et al., 2017; Qian et al., 2024; Dorfman et al., 2024). Sharp results characterize the statistical effect of such contamination and show the benefit of within-batch settings: with n observations per batch, the contamination contribution typically scales as ϵ^2/n under minimax-optimal squared error rate (Yin et al., 2018). Complementary work studies the computational aspects and develops polynomial-time, near-optimal methods for discrete and structured distribution estimation (Jain & Orlitsky, 2020; 2021). We refer readers to Loh (2025) for a comprehensive overview of contamination models.

Online and streaming learning. Stochastic gradient descent and Polyak–Ruppert averaging (Ruppert, 1988; Polyak & Juditsky, 1992) are widely used for online learning because they require only $O(p)$ memory, and their robust variants have also received considerable attention (Pesme & Flammarion, 2020; Shen et al., 2025). For batchwise streaming data, the renewable estimation framework (Luo & Song, 2020) provides a general approach that updates estimates using only the current batch and historical summary statistics. Related studies have further studied anomalous or heterogeneous batches: Luo et al. (2023) develop sequential tests and exclude batches identified as abnormal, and Chen & Yuan (2024) employ fusion penalties to learn the homogeneous structure across batches. These methods provide only asymptotic guarantees, and implicitly assume a minimum separation distance between the central model and heterogeneous models, which is difficult to verify in practice.

Transfer learning and multi-task learning. Transfer learning leverages information from related source tasks to improve estimation or prediction for a target task. When the source and target models are sufficiently similar, transfer can yield better accuracy than target-only learning (Reeve et al., 2021; Li et al., 2022; 2023; 2024; Zhao et al., 2026a). Recent statistical work has characterized the minimax limits of transfer gains and developed adaptive procedures that mitigate negative transfer from dissimilar sources (Li et al., 2022; Lin et al., 2025; Zhao et al., 2026b). Multi-task learning, by contrast, focuses on joint estimation across multiple tasks and studies how task heterogeneity affects the resulting performance (Evgeniou et al., 2005; Negahban & Wainwright, 2008; Lounici et al., 2011). Its theoretical optimality has also received substantial attention recently (Tripuraneni et al., 2020; Duan & Wang, 2023; Tian et al., 2026). Inspired by these perspectives, we consider robust online learning in a sequential batch setting: The preceding central estimate serves as source information for estimating the current batch, while the current batch can also contribute to updating the central estimate.

1.2 Main contribution

The above studies motivate the central question of this study:

> *When each incoming batch may be heterogeneous, can historical information be transferred to improve the learning accuracy of the current (heterogeneous) model? Symmetrically, can information from the current batch be incorporated into the update of the central-model learning?*

We answer these questions affirmatively through a robust one-pass online learning framework. Our main contributions are summarized as follows.

- **Bidirectional online transfer.** Based on the Gaussian location model (Section 2), we develop an online framework that simultaneously estimates the central and current-batch models. We utilize the firm-thresholding operator for adaptively learning the current model (may be heterogeneous), and the clipped-score operator for robustly updating the central-model estimation. The method requires neither knowledge of the contamination probability ϵ nor a minimum separation distance between central and heterogeneous models.

- **Finite-sample guarantees.** For the Gaussian location model and linear regression (Section 3), we provide theoretical guarantees: The central estimation improves as batches accumulate, up to a contamination-dependent error floor. Adaptive transfer can improve estimation of the current model, particularly when it is heterogeneous but close to the central model. The procedure also yields online heterogeneity detection. Numerical experiments (Section 4 and Appendix B) support these findings.

1.3 NOTATION

For sequences a_n and b_n, we write $a_n = O(b_n)$ (or $a_n \lesssim b_n$) if $a_n \leq Cb_n$ for some constant $C > 0$ and all large n, and $a_n \prec b_n$ if $a_n/b_n \to 0$ as $n \to \infty$. We write that $a_n \asymp b_n$ if $a_n = O(b_n)$ and $b_n = O(a_n)$. Let $[m] = \{1, 2, \ldots, m\}$, $\mathbf{1}(\cdot)$ be the indicator function, and $\mathbf{I}_n$ be the $n \times n$ identity matrix. For a vector $\boldsymbol{\beta}$, denote $\|\boldsymbol{\beta}\|_2$ as its Euclidean norm. For matrix $\mathbf{X}$, denote $\mathbf{X}^\dagger$ as its Moore-Penrose pseudoinverse, $\|\mathbf{X}\|_2$ as its operator norm, and $\|\mathbf{X}\|_F$ as its Frobenius norm.

2 A TOY EXAMPLE: POINT ESTIMATION

We begin with a Gaussian location model to illustrate and realize how transfer learning can be incorporated into robust online learning. At each batch $b \geq 1$, we observe a batch of n observations

$$Z_i^{(b)} = \beta^{(b)*} + \xi_i^{(b)}, \qquad \xi_i^{(b)} \sim N(0,1), \qquad \text{for every } i \in [n]. \tag{1}$$

The noise variable $\xi_i^{(b)}$ is mutually independent across observations (every $i \in [n]$) and batches (every $b \geq 1$). We designate the first batch as a trusted reference batch and assume that $\beta^{(1)*} \equiv \beta^*$, where $\beta^* \in \mathbb{R}$ denotes the central parameter. For each subsequent batch $b \geq 2$, let $A_b \overset{\text{iid}}{\sim} \text{Bernoulli}(\epsilon)$, independently of all noise variables, and define

$$\beta^{(b)*} := \begin{cases} \beta^*, & A_b = 0, \\ \theta^{(b)*}, & A_b = 1, \end{cases} \tag{2}$$

where $\theta^{(b)*} \in \mathbb{R}$ is a fixed, batch-specific parameter. This formulation describes a batch-contaminated online model: each subsequent batch follows the central model $N(\beta^*, 1)$ with probability $1 - \epsilon$ and the batch-specific "outlier" model $N(\theta^{(b)*}, 1)$ with probability ϵ, independently across batches. No restriction is imposed on the magnitude of $|\theta^{(b)*} - \beta^*|$. Upon the arrival of batch b, we use its current observations $\{Z_i^{(b)}\}_{i\in[n]}$, together with the last central estimate to: (i) learn the current-batch parameter $\beta^{(b)*}$, and (ii) update the estimate of the central parameter β^*.

2.1 A ROBUST ONLINE ALGORITHM

Suppose that the current batch is heterogeneous, with $\beta^{(b)*} = \theta^{(b)*}$, but that $\theta^{(b)*}$ is sufficiently close to the central parameter β^*. In this case, the preceding central estimate of β^*, denoted by $\hat{m}^{(b-1)}$, may provide useful source information for estimating the current $\theta^{(b)*}$ (from the perspective of transfer learning (Li et al., 2022) and adaptive multi-task learning (Duan & Wang, 2023)). Conversely, the current batch may also contribute useful information for updating the central estimate $\hat{m}^{(b-1)}$. This bidirectional information-sharing structure suggests that a mildly heterogeneous batch need not be discarded: it may benefit from the preceding central estimate and still contribute to the central update. Accordingly, our procedure does not require a preliminary hard classification of each batch as homogeneous or heterogeneous, as in Luo et al. (2023).

To implement this idea, we introduce the clipped-score and firm-thresholding operators:

$$\Psi_c(u) := \begin{cases} -c, & u < -c, \\ u, & |u| \leq c, \\ c, & u > c, \end{cases} \qquad F_a(u) := \begin{cases} 0, & |u| \leq a, \\ 2(1 - a/|u|)u, & a < |u| < 2a, \\ u, & |u| \geq 2a. \end{cases} \tag{3}$$

The clipped-score function $\Psi_c(\cdot)$ is the derivative of the Huber loss with tuning parameter c (Huber, 1973), or equivalently, the projection operator onto the interval $[-c, c]$. It is used in the central

Algorithm 1 Robust One-pass Point Estimation

Input: Batch size n, confidence level $\delta \in (0,1)$, and streaming batches $\{Z_i^{(b)}\}_{i\in[n]}^{b\geq 1}$.

Output: A central estimate $\hat{m}^{(b)}$ and a current-task estimate $\hat{\beta}^{(b)}$ after every batch b.

1: $c_\delta = 1 + 2\sqrt{2\log(6/\delta)}$
2: Observe batch 1 and set $\bar{Z}^{(1)} = n^{-1}\sum_{i=1}^n Z_i^{(1)}$
3: $\hat{m}^{(1)} = \bar{Z}^{(1)};\ \hat{\beta}^{(1)} = \bar{Z}^{(1)}$
4: $\mathcal{I} = \left[\bar{Z}^{(1)} - \sqrt{2\log(6/\delta)/n},\ \bar{Z}^{(1)} + \sqrt{2\log(6/\delta)/n}\right]$
5: **Output of the first batch:** $(\hat{m}^{(1)},\ \hat{\beta}^{(1)})$
6: **for** $b = 2, 3, \ldots,$ **do**
7: Observe batch b and set $\bar{Z}^{(b)} = n^{-1}\sum_{i=1}^n Z_i^{(b)}$
8: $a^{(b)} = \sqrt{2\log(6b^2/\delta)/n}$
9: $\hat{\beta}^{(b)} = \hat{m}^{(b-1)} + F_{a^{(b)}}(\bar{Z}^{(b)} - \hat{m}^{(b-1)})$
10: $u = \sqrt{n}(\bar{Z}^{(b)} - \hat{m}^{(b-1)})$
11: $\hat{m}^{(b)} = \Pi_{\mathcal{I}}\left(\hat{m}^{(b-1)} + \frac{2}{b\sqrt{n}}\Psi_{c_\delta}(u)\right)$
12: **Output of the b-th batch:** $(\hat{m}^{(b)},\ \hat{\beta}^{(b)})$

update to bound the influence of a batch whose mean differs substantially from the preceding central estimate. Thus, a small residual is fully incorporated, whereas a large residual contributes only through its clipped value. The firm-thresholding function $F_a(\cdot)$ (Gao & Bruce, 1997; Selesnick, 2017) suppresses weak signals while avoiding shrinkage of sufficiently strong signals (sufficiently heterogeneous model). Therefore, it is particularly suitable for updating the current-batch parameter.

The resulting procedure is summarized in Algorithm 1. The first batch is used both to initialize the central estimator and to construct a fixed projection interval $\mathcal{I}$ (line 4). By Gaussian concentration (Wainwright, 2019), the interval $\mathcal{I}$ contains β^* with probability at least $1 - \delta/3$. On this event, projection keeps the central updates in a controlled neighborhood of β^*.

For every batch $b \geq 2$, the operator $F_{a^{(b)}}(\cdot)$ (line 9) captures the potential transfer learning. Write $d^{(b)} = \bar{Z}^{(b)} - \hat{m}^{(b-1)}$. The estimator $\hat{\beta}^{(b)}$ uses the preceding central estimate $\hat{m}^{(b-1)}$ when $|d^{(b)}| \leq a^{(b)}$ (full transfer), interpolates between the central estimate $\hat{m}^{(b-1)}$ and local estimate $\bar{Z}^{(b)}$ when $a^{(b)} < |d^{(b)}| < 2a^{(b)}$ (partial transfer), and reduces to the local estimate $\bar{Z}^{(b)}$ when $|d^{(b)}| \geq 2a^{(b)}$. The threshold $a^{(b)}$ (line 8) includes a logarithmic dependence on b to simultaneously control current-batch estimation errors over the unbounded batch number. The operator $\Psi_{c_\delta}(\cdot)$ (line 11) captures the robust update of the central parameter by using the current batch: only when $\bar{Z}^{(b)}$ and $\hat{m}^{(b-1)}$ are sufficiently close do we fully utilize current data to update the central estimation. Here, $\Pi_{\mathcal{I}}(\cdot)$ denotes the projection operator onto the interval $\mathcal{I}$. The algorithm processes each batch once, retains no historical observations, and requires only $O(1)$ persistent memory for center parameter update.

2.2 Theoretical guarantees

This subsection establishes theoretical guarantees for the online Algorithm 1. We first present a lemma characterizing the operator F_a.

Lemma 1 (Firm transfer in any dimension)**.** *Let $a > 0$. If $|y - \theta| \leq a$, then for every proposed center m,*

$$|m + F_a(y - m) - \theta| \leq \min\{|m - \theta|,\ 2a\}. \tag{4}$$

Therefore, once we derive the central estimation bound of $|\hat{m}^{(b-1)} - \beta^*|$, we can apply Lemma 1 with $y = \bar{Z}^{(b)}$, $\theta = \beta^{(b)*}$, $m = \hat{m}^{(b-1)}$, and $a = a^{(b)}$. Then we can directly obtain the current-batch estimation accuracy $|\hat{\beta}^{(b)} - \beta^{(b)*}|$.

Theorem 2 (Online Gaussian Learning)**.** *Under the setting of equation 1–equation 2, we run the online Algorithm 1. Then, with probability at least $1 - \delta$, the following statements hold simultaneously: (i) The first batch satisfies $|\hat{m}^{(1)} - \beta^*|^2 = |\hat{\beta}^{(1)} - \beta^*|^2 \leq 2n^{-1}\log(6/\delta)$. (ii) For every*

$b \geq 2$, *under the assumption* $\epsilon \leq 1/4$, *we have*

$$|\hat{m}^{(b)} - \beta^*|^2 \leq 85c_\delta^2 \left(\frac{\log(3b/\delta)}{nb} + \frac{\epsilon^2}{n} \right), \tag{5}$$

$$|\hat{\beta}^{(b)} - \beta^{(b)*}|^2 \leq \min \left\{ \frac{8\log(6b^2/\delta)}{n}, \left(\sqrt{\frac{85c_\delta^2 \log(3b/\delta)}{n(b-1)} + \frac{85c_\delta^2\epsilon^2}{n}} + |\beta^{(b)*} - \beta^*| \right)^2 \right\}$$

$$\asymp \min \left\{ \frac{\log(b^2/\delta)}{n}, \frac{c_\delta^2 \log(b/\delta)}{nb} + \frac{c_\delta^2\epsilon^2}{n} + |\beta^{(b)*} - \beta^*|^2 \right\} \tag{6}$$

where recall $c_\delta = 1 + 2\sqrt{2\log(6/\delta)}$.

Equation 5 shows that, even without storing all cumulative samples, the online estimation error of β^* still consists of two parts (ignoring logarithmic terms): the rate $1/(nb)$ in the cumulative-sample sense, and a contamination term scaled by effective variance ($\epsilon^2 \times 1/n$). This indicates that, under the contaminated setting, our algorithm nearly attains the offline minimax optimal rate (Gao, 2020).

Equation 6 shows that (ignoring logarithmic terms), even when the current batch is heterogeneous, the estimation rate of $\beta^{(b)*}$ is no worse than that of the naive batch mean $\bar{Z}^{(b)}$, i.e., $1/n$. Furthermore, once $\beta^{(b)*}$ is sufficiently close to β^*, i.e., $|\beta^{(b)*} - \beta^*|^2 \prec 1/n$, the estimation accuracy of β^* (equation 5) can be *adaptively* transferred to learn current $\beta^{(b)*}$, yielding the estimation error in equation 6 smaller than $1/n$. When the current batch is homogeneous ($\beta^{(b)*} = \beta^*$), there is no shift cost and equation 6 directly matches equation 5. These results confirm our initial intuition: online transfer can simultaneously improve estimation of the central model and the current heterogeneous model, without knowing which batches are heterogeneous or how far their parameters are from the central parameter.

In addition, our method requires neither prior knowledge of the contamination probability ϵ nor the distance between $\beta^{(b)*}$ and β^*, making it fully adaptive. This advantage ensures its practical feasibility. The proof strategy and theoretical insights underlying Theorem 2 remain important in the linear regression setting. Proofs for this section are provided in Appendix C.

3 The linear model setting

We now extend the preceding Gaussian location model example to linear regression. Throughout this section, $C_1, C_2, \ldots$ denote fixed positive constants independent of the model parameters such as n, p, δ, ϵ. Their values are not optimized and should not be taken as practical guidance for implementation.

3.1 Model and assumptions

Fix a prespecified batch budget B. For each batch $b \in [B]$, we observe $D^{(b)} = \{(y_i^{(b)}, \mathbf{x}_i^{(b)})\}_{i\in[n]}$ from the linear model

$$y_i^{(b)} = \mathbf{x}_i^{(b)\top} \boldsymbol{\beta}^{(b)*} + \xi_i^{(b)}, \qquad i \in [n], \qquad \mathbf{x}_i^{(b)} \in \mathbb{R}^p. \tag{7}$$

Let $\mathbf{y}^{(b)} = (y_1^{(b)}, \ldots, y_n^{(b)})^\top \in \mathbb{R}^n$ and $\boldsymbol{\xi}^{(b)} = (\xi_1^{(b)}, \ldots, \xi_n^{(b)})^\top \in \mathbb{R}^n$ denote the response and noise vectors in the b-th batch, respectively.

Suppose the first batch is a trusted reference batch satisfying $\boldsymbol{\beta}^{(1)*} = \boldsymbol{\beta}^*$, where $\boldsymbol{\beta}^* \in \mathbb{R}^p$ is the central regression parameter. For each subsequent batch $b \geq 2$, let $A_b \overset{\text{iid}}{\sim} \text{Bernoulli}(\epsilon)$, where $\epsilon \in [0, 1/2)$ and $\{A_b\}_{b\geq 2}$ is independent of all design and noise variables. The batch-specific regression parameter is then defined by

$$\boldsymbol{\beta}^{(b)*} = (1 - A_b)\boldsymbol{\beta}^* + A_b\boldsymbol{\theta}^{(b)*}, \qquad 2 \leq b \leq B,$$

where $\boldsymbol{\theta}^{(b)*} \in \mathbb{R}^p$ is an arbitrary fixed parameter that may vary across batches. Thus, each subsequent batch follows the central regression model with probability $1 - \epsilon$ and a batch-specific regression model with probability ϵ. We impose the following assumptions on the designs and noises.

Assumption 1 (Online sub-Gaussian designs)**.** *For each $b \in [B]$, let $\mathbf{X}^{(b)} \in \mathbb{R}^{n\times p}$ be the design matrix whose i-th row is $\mathbf{x}_i^{(b)\top} \in \mathbb{R}^{1\times p}$. Assume that $\mathbf{X}^{(b)} \overset{d}{=} \mathbf{Z}^{(b)}\big(\mathbf{\Sigma}^{(b)}\big)^{1/2}$, where each population covariance matrix $\mathbf{\Sigma}^{(b)} \in \mathbb{R}^{p\times p}$ is positive definite. The entries $\{Z_{ij}^{(b)}\}_{i\in[n],j\in[p]}^{b\in[B]}$ are mutually independent and identically distributed, centered, and 1-sub-Gaussian with unit variance, i.e.,*

$$\mathbb{E}Z_{ij}^{(b)} = 0, \quad \mathbb{E}e^{tZ_{ij}^{(b)}} \le e^{t^2/2} \quad \textit{for all } t \in \mathbb{R}, \quad \mathbb{E}[\big(\mathbf{Z}_{i,\cdot}^{(b)}\big)^\top \mathbf{Z}_{i,\cdot}^{(b)}] = \mathbf{I}_p. \tag{8}$$

For fixed constant $\kappa_0 > 0$, assume uniformly over $b \in [B]$ that $\kappa_0 \mathbf{I}_p \preceq \mathbf{\Sigma}^{(b)}$.

Assumption 1 allows the population covariance matrices to vary across batches and does not require them to share eigenvectors, thereby accommodating batch-dependent covariate distributions.

Assumption 2 (Online sub-Gaussian noises)**.** *The noise variables $\{\xi_i^{(b)}\}_{i\in[n]}^{b\in[B]}$ are mutually independent and independent of all design matrices and contamination indicators. For a common constant $\sigma > 0$, we assume*

$$\mathbb{E}\xi_i^{(b)} = 0, \quad \mathbb{E}e^{t\xi_i^{(b)}} \le e^{\sigma^2 t^2/2} \quad \textit{for all } t \in \mathbb{R}. \tag{9}$$

Assumption 2 relies on a unify sub-Gaussian scale σ. It allows the noise distributions to vary across observations and batches, and does not require them to be symmetric.

3.2 Robust online regression algorithm

We extend Algorithm 1 to the linear model. First, define the vector version of the firm-thresholding operator in equation 3 as:

$$\mathbf{F}_a(\mathbf{z}) := F_a(\|\mathbf{z}\|_2) \cdot \frac{\mathbf{z}}{\|\mathbf{z}\|_2}, \qquad \mathbf{z} \neq \mathbf{0},$$

with $\mathbf{F}_a(\mathbf{0}) := \mathbf{0}$.

Algorithm 2 Robust Online Linear Regression

Input: Batch size n, dimension p, batch budget B, confidence level $\delta \in (0,1)$, valid scales σ, and streaming batches $\{(\mathbf{X}^{(b)}, \mathbf{y}^{(b)})\}_{b\in[B]}$

Output: A central estimate $\hat{\mathbf{m}}^{(b)}$ and a current-task estimate $\hat{\boldsymbol{\beta}}^{(b)}$ after every batch $b \le B$.

1: $\nu = \sigma\sqrt{2/(n\kappa_0)}$
2: Observe batch 1 and set $\widetilde{\boldsymbol{\beta}}^{(1)} = (\mathbf{X}^{(1)\top}\mathbf{X}^{(1)})^\dagger \mathbf{X}^{(1)\top}\mathbf{y}^{(1)}$
3: $\hat{\mathbf{m}}^{(1)} = \widetilde{\boldsymbol{\beta}}^{(1)};\ \hat{\boldsymbol{\beta}}^{(1)} = \widetilde{\boldsymbol{\beta}}^{(1)}$
4: **for** $j = 1, \dots, p$ **do**
5: $\quad \mathcal{I}_j = \Big[\widetilde{\beta}_j^{(1)} - C_1\nu\sqrt{\log(C_2 p/\delta)},\ \widetilde{\beta}_j^{(1)} + C_1\nu\sqrt{\log(C_2 p/\delta)}\Big]$
6: **Output of the first batch:** $(\hat{\mathbf{m}}^{(1)},\ \hat{\boldsymbol{\beta}}^{(1)})$
7: **for** $b = 2, \dots, B$, **do**
8: $\quad$ Observe batch b and set $\widetilde{\boldsymbol{\beta}}^{(b)} = (\mathbf{X}^{(b)\top}\mathbf{X}^{(b)})^\dagger \mathbf{X}^{(b)\top}\mathbf{y}^{(b)}$
9: $\quad r_b = C_3\nu\sqrt{p + \log(C_2 b^2/\delta)}$
10: $\quad \hat{\boldsymbol{\beta}}^{(b)} = \hat{\mathbf{m}}^{(b-1)} + \mathbf{F}_{r_b}(\widetilde{\boldsymbol{\beta}}^{(b)} - \hat{\mathbf{m}}^{(b-1)})$
11: $\quad \mathbf{u} = (\widetilde{\boldsymbol{\beta}}^{(b)} - \hat{\mathbf{m}}^{(b-1)})/\nu$
12: $\quad c_b = C_4\sqrt{\log(C_2 p b^2/\delta)}$
13: $\quad$ **for** $j = 1, \dots, p$, **do**
14: $\quad\quad \hat{m}_j^{(b)} = \Pi_{\mathcal{I}_j}\Big(\hat{m}_j^{(b-1)} + (2\nu/b)\Psi_{c_b}(u_j)\Big)$
15: $\quad$ **Output of the b-th batch:** $(\hat{\mathbf{m}}^{(b)},\ \hat{\boldsymbol{\beta}}^{(b)})$

Algorithm 2 firstly uses the batchwise OLS estimator $\widetilde{\boldsymbol{\beta}}^{(1)}$ for initialization. For each subsequent batch, the radial firm-thresholding update (line 10) adaptively interpolates between the preceding central estimate and the current batchwise OLS estimator $\widetilde{\boldsymbol{\beta}}^{(b)}$. The current-task estimate is formed before the current batch is incorporated into the central state. The central estimate is then updated coordinatewise using the clipped score $\Psi_{c_b}(\cdot)$, with each coordinate j projected onto $\mathcal{I}_j$ (lines 13–14).

The algorithm retains no historical observations from previous batches; the central and current-batch estimates require $O(p)$ persistent storage.

3.3 Joint guarantees and their statistical interpretation

This subsection establishes theoretical guarantees for the simultaneous learning performance of Algorithm 2 over the first B batches. For ease of exposition, we recall and define:

$$r_b = C_3\nu\sqrt{p+\log(C_2b^2/\delta)}, \qquad \mathcal{H}_b^2 := C_5\frac{p\sigma^2}{\kappa_0 n}\left(\epsilon^2\log(C_2pb^2/\delta)+\frac{\log^2(C_2pb^2/\delta)}{b}\right). \quad (10)$$

Theorem 3 (Robust Online regression)**.** *Suppose Assumptions 1–2 hold, $\epsilon \le 1/4$, and the batch size satisfies $n \ge C_6\,[p+\log(B/\delta)]$. We take $\nu = \sigma\sqrt{2/(n\kappa_0)}$. Then with a probability at least $1-\delta$, the following bounds hold simultaneously:*

1. *The initialization bound*

$$\left\|\hat{\mathbf{m}}^{(1)}-\boldsymbol{\beta}^*\right\|_2^2 = \left\|\hat{\boldsymbol{\beta}}^{(1)}-\boldsymbol{\beta}^*\right\|_2^2 \le r_1^2 \asymp \frac{\sigma^2}{\kappa_0 n}(p+\log(1/\delta)). \quad (11)$$

2. *For every $2 \le b \le B$, we have*

$$\left\|\hat{\mathbf{m}}^{(b)}-\boldsymbol{\beta}^*\right\|_2^2 \le \mathcal{H}_b^2 \asymp \sigma^2\left(\frac{p}{nb}\log^2(pb^2/\delta)+\frac{\epsilon^2}{n}\cdot p\log(pb^2/\delta)\right), \quad (12)$$

$$\left\|\hat{\boldsymbol{\beta}}^{(b)}-\boldsymbol{\beta}^{(b)*}\right\|_2^2 \le \min\left\{4(r^{(b)})^2, \left(\mathcal{H}_{b-1}+\left\|\boldsymbol{\beta}^{(b)*}-\boldsymbol{\beta}^*\right\|_2\right)^2\right\}$$
$$\asymp \sigma^2\min\left\{\frac{p+\log(pb^2/\delta)}{n}, \frac{p\log^2(pb^2/\delta)}{nb}+\frac{\epsilon^2 p\log(pb^2/\delta)}{n}+\frac{\left\|\boldsymbol{\beta}^{(b)*}-\boldsymbol{\beta}^*\right\|_2^2}{\sigma^2}\right\}. \quad (13)$$

Similar to the result in equation 5, Theorem 3 gives a central-parameter learning bound of order $\frac{\sigma^2 p}{n}\left(\frac{1}{b}+\epsilon^2\right)$, ignoring logarithmic factors. Furthermore, if the current batch model (target) is heterogeneous, we can adaptively leverage the previous central model estimate (source) to achieve a sharper bound (equation 13) than that of local OLS, provided that $\left\|\boldsymbol{\beta}^{(b)}-\boldsymbol{\beta}^*\right\|_2^2 \prec \sigma^2 p/n$. This confirms again, from a statistical perspective, that transfer learning is beneficial for online learning under contaminated models.

Remark 1 (The dimension cost of coordinate-wise learning)**.** *In the offline setting, our contaminated model is related to certain Byzantine settings, where an ϵ-fraction of batches outputs contaminated $\widetilde{\boldsymbol{\beta}}^{(b)}$ (Lamport et al., 1982). The minimax optimal learning rate for Byzantine models is known to be $\sigma^2\big(p/(nb)+\epsilon^2/n\big)$ (Yin et al., 2018), a rate that can be traced back to a line of work on the Huber contamination model (Chen et al., 2018; Gao, 2020). Hence, up to logarithmic factors, our bound in equation 12 matches this optimal rate when $p = O(1)$ or $b\epsilon^2 = O(1)$. On the other hand, our algorithm performs robust updates of the central parameter coordinatewise, which explains the presence of p in our contamination term; by contrast, existing optimal frameworks typically require a low-dimensional projection of the parameter to circumvent the dimensionality effect (Diakonikolas et al., 2022; 2023). We leave the design of a rate-optimal online algorithm as a direction for future work.*

Remark 2 (Scale calibration in practice)**.** *The preceding guarantees use the known scale $\nu = \sigma\sqrt{2/(n\kappa_0)}$. When σ or κ_0 is unknown, we replace ν with the following practical, data-driven estimator in each batch:*

$$\hat{\nu}^{(b)} = \sqrt{\frac{2\|(\mathbf{I}_n-\mathbf{P}_{\mathbf{X}^{(b)}})\mathbf{y}^{(b)}\|^2}{n\lambda_{\min}(\mathbf{X}^{(b)\top}\mathbf{X}^{(b)})}},$$

where $\mathbf{P}_{\mathbf{X}^{(b)}} := \mathbf{X}^{(b)}(\mathbf{X}^{(b)\top}\mathbf{X}^{(b)})^{\dagger}\mathbf{X}^{(b)\top}$ denotes the orthogonal projection matrix onto the column space of $\mathbf{X}^{(b)}$.

We next examine the ability to detect heterogeneous batches. For each $2 \le b \le B$, consider the hypothesis

$$H_0^{(b)} : \boldsymbol{\beta}^{(b)*} = \boldsymbol{\beta}^*, \qquad H_1^{(b)} : \boldsymbol{\beta}^{(b)*} \neq \boldsymbol{\beta}^*.$$

We reject $H_0^{(b)}$ when $\|\widetilde{\boldsymbol{\beta}}^{(b)} - \hat{\mathbf{m}}^{(b-1)}\|_2 > r_b + 2C_1\nu\sqrt{p\log(C_2 p/\delta)}$, where recall $\nu = \sigma\sqrt{2/(n\kappa_0)}$ and $\widetilde{\boldsymbol{\beta}}^{(b)} = (\mathbf{X}^{(b)\top}\mathbf{X}^{(b)})^{\dagger}\mathbf{X}^{(b)\top}\mathbf{y}^{(b)}$ is the local OLS estimate.

Corollary 4 (Online detection)**.** *Under all assumptions of Theorem 3, with probability at least $1-\delta$, the following hold simultaneously for all $2 \le b \le B$:*

1. *Every homogeneous batch ($\boldsymbol{\beta}^{(b)*} = \boldsymbol{\beta}^*$) is accepted.*
2. *Every batch satisfying*

$$\left\|\boldsymbol{\beta}^{(b)*} - \boldsymbol{\beta}^*\right\|_2 > 2\left(r_b + 2C_1\nu\sqrt{p\log(C_2 p/\delta)}\right) \asymp \sigma\sqrt{\frac{p\log(p/\delta) + \log(b^2/\delta)}{n}} \quad (14)$$

is detected as the heterogeneous batch.

In particular, the number of false discoveries is zero with high probability. Up to logarithmic factors, the required separation rate (in equation 14) is of order $\sigma\sqrt{p/n}$. This separation rate does not decrease with b: because the batch-specific heterogeneous parameters may vary across batches, each batch constitutes a separate detecting problem based only on its own n observations.

4 Numerical experiment

We conduct linear model experiments to validate the improvement in estimation accuracy achieved by our framework. All experiments are conducted using R and executed on a personal laptop equipped with an AMD Ryzen 7 5800H processor operating at 3.20 GHz and 16.00GB of RAM.

Setup. We generate $B = 50$ batches of size $n = 100$ from $y_i^{(b)} = \mathbf{x}_i^{(b)\top}\boldsymbol{\beta}^{(b)*} + \xi_i^{(b)}$, with independent $\mathbf{x}_i^{(b)} \sim N(\mathbf{0}_p, \boldsymbol{\Sigma})$, $\Sigma_{jk} = 0.5^{|j-k|}$, and $\xi_i^{(b)} \sim N(0,1)$, i.i.d. The central coefficient is $\boldsymbol{\beta}^* = \mathbf{1}_p$. The first batch is central; each later batch is independently heterogeneous with probability $\epsilon = 0.2$, with coefficient $\boldsymbol{\theta}^{(b)*} = \boldsymbol{\beta}^* + (-1)^b \cdot h \cdot \mathbf{d}$ for a fixed unit vector $\mathbf{d}$. We consider 4 experiments in the main manuscript: E1–E2 use $(p, h) = (10, 1)$ and E3–E4 use $(30, 1.8)$. The direction vector $\mathbf{d}$ has a five-coordinate core. In E1–E2, coordinates 3–7 have equal magnitudes and alternating signs. In E3–E4, coordinates 13–17 align with the eigenvector corresponding to the smallest eigenvalue of their covariance submatrix. E1/E3 add weak shifts outside this core (*full*), while E2/E4 shift only the five fixed coordinates (*fixed5*).

Methods and metrics. We compare our method (Proposed) with renewable OLS (Luo & Song, 2020), renewable QIF (Luo et al., 2023), renewable Huber (Jiang et al., 2024), LAD SGD (Pesme & Flammarion, 2020), and online quantile regression (Shen et al., 2025). Two shaded *offline reference baselines* provide additional comparisons: the ARMUL method (Duan & Wang, 2023) jointly refits all batches offline at each checkpoint; the Oracle method knows the heterogeneous labels, pools the central batches by OLS, and fits each heterogeneous batch by local OLS. Methods without batch-specific estimates use their current center estimation. Across 50 independent Monte Carlo replications, C_b averages $\|\hat{\mathbf{m}}^{(b)} - \boldsymbol{\beta}^*\|_2^2$, and H_b averages $\|\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\theta}^{(j)*}\|_2^2$ over all truly heterogeneous batches $j \le b$.

Table 1 shows that the proposed method achieves the lowest C_b and H_b among the online methods in every setting and checkpoint. Its central estimation error C_b decreases as more batches arrive. The heterogeneous error H_b of our method decreases as the heterogeneous shift strength $h\|\mathbf{d}_p\|_2 = h$ becomes smaller (from E3/E4 to E1/E2), consistent with equation 13 in Theorem 3: heterogeneous models closer to the central model benefit more from transfer learning. Furthermore, the proposed H_b even improves on Oracle's H_b (estimating these batches by local OLS) in some cases: at $b = 50$, the proposed H_b is 0.6972 versus the Oracle 0.7164 in E3, and 0.7023 versus 0.7179 in E4. This is consistent with our adaptive transfer guarantee again: an accurate central estimate can improve heterogeneous-batch learning beyond the local estimation rate $\sigma^2 p/n$.

Table 1: Central and heterogeneous-model errors over 50 replications. Bold marks the best online method; shaded rows are offline/oracle benchmarks.

Method	**E1**: full, $p = 10$, $h = 1$				**E2**: fixed5, $p = 10$, $h = 1$			
	$b = 25$		$b = 50$		$b = 25$		$b = 50$	
	C_b	H_b	C_b	H_b	C_b	H_b	C_b	H_b
Proposed	**0.0104**	**0.1791**	**0.0044**	**0.1793**	**0.0102**	**0.1784**	**0.0053**	**0.1852**
Renewable OLS	0.0132	0.8413	0.0049	0.9054	0.0112	0.8567	0.0070	0.9108
Renewable QIF	0.0141	0.7583	0.0052	0.8305	0.0135	0.7703	0.0076	0.8475
Renewable Huber	0.0131	0.8594	0.0053	0.9162	0.0110	0.8615	0.0068	0.9143
LAD SGD	0.0189	1.0262	0.0082	1.0231	0.0212	1.0488	0.0105	1.0227
Online quantile	0.0167	0.9503	0.0086	0.9717	0.0147	1.3059	0.0076	1.1274
ARMUL	0.0095	0.1692	0.0040	0.1701	0.0086	0.1934	0.0049	0.1997
Oracle	0.0084	0.1791	0.0037	0.1793	0.0070	0.1784	0.0039	0.1852
Method	**E3**: full, $p = 30$, $h = 1.8$				**E4**: fixed5, $p = 30$, $h = 1.8$			
	$b = 25$		$b = 50$		$b = 25$		$b = 50$	
	C_b	H_b	C_b	H_b	C_b	H_b	C_b	H_b
Proposed	**0.0336**	**0.7045**	**0.0168**	**0.6972**	**0.0347**	**0.7037**	**0.0169**	**0.7023**
Renewable OLS	0.0450	2.7189	0.0261	2.9004	0.0429	2.7555	0.0206	2.9313
Renewable QIF	0.0450	2.7189	0.0261	2.9004	0.0429	2.7555	0.0206	2.9313
Renewable Huber	0.0409	2.7798	0.0233	2.9392	0.0390	2.8146	0.0192	2.9685
LAD SGD	0.0851	3.4210	0.0381	3.3278	0.0807	3.4650	0.0335	3.3843
Online quantile	0.0593	4.9943	0.0293	4.0607	0.0630	8.4478	0.0255	5.9081
ARMUL	0.0319	0.8011	0.0164	0.7862	0.0320	0.7955	0.0154	0.7996
Oracle	0.0253	0.7069	0.0120	0.7164	0.0247	0.7156	0.0124	0.7179

Moreover, in higher dimensions, the heterogeneity test used by Renewable QIF may become less effective at detecting heterogeneous batches. Therefore, when such batches are treated as homogeneous, its estimates coincide with those of Renewable OLS (E3 and E4). Additional simulations under alternative shift settings and in higher-dimensional settings are presented in Appendix B.

5 Conclusion

This paper develops a robust framework for online learning through a bidirectional view of transfer: Even if the current batch is not from the central model, the preceding central estimate can adaptively improve estimation of the current model, while the current batch can also contribute to the central update through a bounded-influence rule. We provide finite-sample theoretical support for this viewpoint in both the location model and linear regression. Numerical experiments also support these findings. Limitations and future directions are discussed in Appendix A.

Finally, we emphasize that our main contribution is to theoretically establish how bidirectional transfer can improve robust online learning. The Gaussian location model and the linear model serve as accessible settings for illustrating and validating this viewpoint. We expect the framework to apply more broadly, with extensions to more complex models left for future research.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

The authors used the generative AI model GPT-5.6-sol as an assistive tool during the preparation of this manuscript. Its use was limited to the following purposes:

- editing the English language, including grammar correction, sentence refinement, and improvement of clarity, conciseness, and academic style;
- assisting with the organization and presentation of the introduction, related work, model descriptions, theoretical results, and other mathematical exposition;
- checking the consistency of mathematical notation, parameter definitions, model assumptions, independence conditions, and cross-references;
- discussing the formulation and exposition of theorem statements, algorithm descriptions, and interpretations of results, including identifying possible ambiguities, unstated assumptions, or overly strong claims; and
- assisting with code debugging by suggesting possible sources of implementation errors, candidate corrections, and consistency checks between the algorithms described in the manuscript and their implementations.

Generative AI was not used to generate, select, verify, or interpret the references and citations, experimental results, tables reported in this work. These materials, as well as the corresponding scientific claims, were prepared and checked entirely by the authors. The authors take full responsibility for the final content of the paper, including all mathematical statements, proofs, algorithms, code, references, citations, experimental results, tables, and other artifacts.

ETHICS STATEMENT

(This section is **recommended** and does not count toward the page limit.)

This work develops theoretical and algorithmic methods for robust online statistical learning. It does not involve human subjects, personally identifiable information, or the collection of sensitive data. The proposed methods may be applicable to distributed and federated learning systems, but their deployment in practice should account for the privacy, fairness, and security requirements of the specific application. We are not aware of other significant ethical concerns arising directly from this work.

REPRODUCIBILITY STATEMENT

(This section is **recommended** and does not count toward the page limit.)

The numerical experiments in Section 4 and Appendix B focus on the linear regression setting and compare the proposed method with the considered baselines. These sections describes the data-generating mechanisms, parameter settings, evaluation metrics, and experimental protocols used in the simulations. Anonymous code and the corresponding configuration files for reproducing the linear-regression simulations are submitted as supplementary material. The model assumptions, algorithmic procedures, and tuning parameters are specified in Sections 2–3, while complete proofs of the theoretical results are provided in the Appendices C–D.

REFERENCES

Mengjie Chen, Chao Gao, and Zhao Ren. Robust covariance and scatter matrix estimation under Huber's contamination model. *The Annals of Statistics*, 46(5):1932 – 1960, 2018. doi: 10.1214/17-AOS1607. URL https://doi.org/10.1214/17-AOS1607.

Xuerong Chen and Senlin Yuan. Renewable quantile regression with heterogeneous streaming datasets. *Journal of Computational and Graphical Statistics*, 33(4):1185–1201, 2024. doi: 10.1080/10618600.2024.2309343. URL https://doi.org/10.1080/10618600.2024.2309343.

Yudong Chen, Lili Su, and Jiaming Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proc. ACM Meas. Anal. Comput. Syst.*, 1(2), December 2017. doi: 10.1145/3154503. URL https://doi.org/10.1145/3154503.

Ilias Diakonikolas, Daniel M. Kane, Ankit Pensia, and Thanasis Pittas. Streaming algorithms for high-dimensional robust statistics. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 5061–5117. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/diakonikolas22a.html.

Ilias Diakonikolas, Daniel Kane, Ankit Pensia, and Thanasis Pittas. Near-optimal algorithms for gaussians with huber contamination: Mean estimation and linear regression. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 43384–43422. Curran Associates, Inc., 2023. doi: 10.52202/075280-1879. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/86c283920335ed1fec3edee227e05fbf-Paper-Conference.pdf.

Ilias Diakonikolas, Chao Gao, Daniel M Kane, Ankit Pensia, and Dong Xie. Robust regression with adaptive contamination in response: Optimal rates and computational barriers. *arXiv preprint arXiv:2604.04228*, 2026.

Ron Dorfman, Naseem Amin Yehya, and Kfir Yehuda Levy. Dynamic Byzantine-robust learning: Adapting to switching Byzantine workers. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11501–11543. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/dorfman24a.html.

Yaqi Duan and Kaizheng Wang. Adaptive and robust multi-task learning. *The Annals of Statistics*, 51(5):2015 – 2039, 2023. doi: 10.1214/23-AOS2319. URL https://doi.org/10.1214/23-AOS2319.

Theodoros Evgeniou, Charles A. Micchelli, and Massimiliano Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6(21):615–637, 2005. URL http://jmlr.org/papers/v6/evgeniou05a.html.

Chao Gao. Robust regression via mutivariate regression depth. *Bernoulli*, 26(2):1139 – 1170, 2020. doi: 10.3150/19-BEJ1144. URL https://doi.org/10.3150/19-BEJ1144.

Hong-Ye Gao and Andrew G Bruce. Waveshrink with firm shrinkage. *Statistica Sinica*, pp. 855–874, 1997.

Daniel Hsu, Sham Kakade, and Tong Zhang. A tail inequality for quadratic forms of subgaussian random vectors. *Electronic Communications in Probability*, 17(none):1 – 6, 2012. doi: 10.1214/ECP.v17-2079. URL https://doi.org/10.1214/ECP.v17-2079.

Peter J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73 – 101, 1964. doi: 10.1214/aoms/1177703732. URL https://doi.org/10.1214/aoms/1177703732.

Peter J. Huber. Robust Regression: Asymptotics, Conjectures and Monte Carlo. *The Annals of Statistics*, 1(5):799 – 821, 1973. doi: 10.1214/aos/1176342503. URL https://doi.org/10.1214/aos/1176342503.

Ayush Jain and Alon Orlitsky. Optimal robust learning of discrete distributions from batches. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4651–4660. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/jain20a.html.

Ayush Jain and Alon Orlitsky. Robust density estimation from batches: The best things in life are (nearly) free. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4698–4708. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/jain21a.html.

Rong Jiang, Lei Liang, and Keming Yu. Renewable Huber estimation method for streaming datasets. *Electronic Journal of Statistics*, 18(1):674 – 705, 2024. doi: 10.1214/24-EJS2223. URL https://doi.org/10.1214/24-EJS2223.

Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

Sai Li, T. Tony Cai, and Hongzhe Li. Transfer learning for high-dimensional linear regression: Prediction, estimation and minimax optimality. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(1):149–173, 02 2022. ISSN 1369-7412. doi: 10.1111/rssb.12479. URL https://doi.org/10.1111/rssb.12479.

Sai Li, T. Tony Cai, and Hongzhe Li. Transfer learning in large-scale gaussian graphical models with false discovery rate control. *Journal of the American Statistical Association*, 118(543): 2171–2183, 2023. doi: 10.1080/01621459.2022.2044333. URL https://doi.org/10.1080/01621459.2022.2044333. PMID: 38143788.

Sai Li, Linjun Zhang, T. Tony Cai, and Hongzhe Li. Estimation and inference for high-dimensional generalized linear models with knowledge transfer. *Journal of the American Statistical Association*, 119(546):1274–1285, 2024. doi: 10.1080/01621459.2023.2184373. URL https://doi.org/10.1080/01621459.2023.2184373. PMID: 38948492.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In I. Dhillon, D. Papailiopoulos, and V. Sze (eds.), *Proceedings of Machine Learning and Systems*, volume 2, pp. 429–450, 2020. URL https://proceedings.mlsys.org/paper_files/paper/2020/file/1f5fe83998a09396ebe6477d9475ba0c-Paper.pdf.

Ziqian Lin, Yuan Gao, Feifei Wang, and Hansheng Wang. Testing sufficiency for transfer learning. *Computational Statistics & Data Analysis*, 203:108075, 2025. ISSN 0167-9473. doi: https://doi.org/10.1016/j.csda.2024.108075. URL https://www.sciencedirect.com/science/article/pii/S0167947324001592.

Po-Ling Loh. A theoretical review of modern robust statistics. *Annual Review of Statistics and Its Application*, 12:477 – 496, 2025. ISSN 2326-831X. doi: https://doi.org/10.1146/annurev-statistics-112723-034446. URL https://www.annualreviews.org/content/journals/10.1146/annurev-statistics-112723-034446.

Karim Lounici, Massimiliano Pontil, Sara van de Geer, and Alexandre B. Tsybakov. Oracle inequalities and optimal inference under group sparsity. *The Annals of Statistics*, 39(4):2164–2204, 2011.

Lan Luo and Peter X.-K. Song. Renewable estimation and incremental inference in generalized linear models with streaming data sets. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 82(1):69–97, 2020. ISSN 1369-7412. doi: 10.1111/rssb.12352. URL https://doi.org/10.1111/rssb.12352.

Lan Luo, Ling Zhou, and Peter X.-K. Song. Real-time regression analysis of streaming clustered data with possible abnormal data batches. *Journal of the American Statistical Association*, 118 (543):2029–2044, 2023. doi: 10.1080/01621459.2022.2026778. URL https://doi.org/10.1080/01621459.2022.2026778. PMID: 37771510.

Sahand Negahban and Martin J Wainwright. Joint support recovery under high-dimensional scaling: Benefits and perils of $\ell_{1,\infty}$-regularization. *Advances in Neural Information Processing Systems*, 21:1161–1168, 2008.

Roberto I. Oliveira and Lucas Resende. Trimmed sample means for robust uniform mean estimation and regression. *The Annals of Statistics*, 53(5):2153 – 2178, 2025. doi: 10.1214/25-AOS2536. URL https://doi.org/10.1214/25-AOS2536.

Ankit Pensia, Varun Jog, and Po-Ling Loh. Robust regression with covariate filtering: Heavy tails and adversarial contamination. *Journal of the American Statistical Association*, 120(550):1002 – 1013, 2025. doi: 10.1080/01621459.2024.2392906. URL https://doi.org/10.1080/01621459.2024.2392906.

Scott Pesme and Nicolas Flammarion. Online robust regression via sgd on the l1 loss. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 2540–2552. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1ae6464c6b5d51b363d7d96f97132c75-Paper.pdf.

Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992.

Chengde Qian, Mengyuan Wang, Haojie Ren, and Changliang Zou. Bymi: Byzantine machine identification with false discovery rate control. In *Forty-first International Conference on Machine Learning*, 2024.

Henry W. J. Reeve, Timothy I. Cannings, and Richard J. Samworth. Adaptive transfer learning. *The Annals of Statistics*, 49(6):3618 – 3649, 2021. doi: 10.1214/21-AOS2102. URL https://doi.org/10.1214/21-AOS2102.

David Ruppert. Efficient estimations from a slowly convergent robbins-monro process. Technical report, Cornell University Operations Research and Industrial Engineering, 1988.

Ivan Selesnick. Sparse regularization via convex analysis. *IEEE Transactions on Signal Processing*, 65(17):4481–4494, 2017. doi: 10.1109/TSP.2017.2711501.

Yinan Shen, Dong Xia, and Wen-Xin Zhou. Online quantile regression. *Journal of Machine Learning Research*, 26(231):1–55, 2025. URL http://jmlr.org/papers/v26/24-0589.html.

Lili Su and Nitin H. Vaidya. Fault-tolerant multi-agent optimization: Optimal iterative distributed algorithms. PODC '16, pp. 425–434, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450339643. doi: 10.1145/2933057.2933105. URL https://doi.org/10.1145/2933057.2933105.

Qiang Sun, Wen-Xin Zhou, and Jianqing Fan. Adaptive Huber regression. *Journal of the American Statistical Association*, 115(529):254 – 265, 2020. doi: 10.1080/01621459.2018.1543124. URL https://doi.org/10.1080/01621459.2018.1543124.

Ye Tian, Mengchu Li, and Marco Avella Medina. Contaminated multi-task learning with heterogeneity: Fundamental limits and optimal algorithms. *arXiv preprint arXiv:2607.02681*, 2026.

Nilesh Tripuraneni, Michael I. Jordan, and Chi Jin. On the theory of transfer learning: the importance of task diversity. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5650–5659. PMLR, 10–15 Jul 2018.

Xuezhou Zhang, Xiaojin Zhu, and Laurent Lessard. Online data poisoning attacks. In Alexandre M. Bayen, Ali Jadbabaie, George Pappas, Pablo A. Parrilo, Benjamin Recht, Claire Tomlin, and Melanie Zeilinger (eds.), *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pp. 201–210. PMLR, 10–11 Jun 2020. URL https://proceedings.mlr.press/v120/zhang20b.html.

Boxin Zhao, Cong Ma, and Mladen Kolar. Trans-glasso: A transfer learning approach to precision matrix estimation. *Journal of the American Statistical Association*, 0(0):1–13, 2026a. doi: 10.1080/01621459.2025.2602856. URL https://doi.org/10.1080/01621459.2025.2602856.

Junlong Zhao, Shengbin Zheng, and Chenlei Leng. Residual importance weighted transfer learning for high-dimensional linear regression. *Journal of the American Statistical Association*, 0(0): 1–14, 2026b. doi: 10.1080/01621459.2026.2623997. URL https://doi.org/10.1080/01621459.2026.2623997.

Banghua Zhu, Lun Wang, Qi Pang, Shuai Wang, Jiantao Jiao, Dawn Song, and Michael I. Jordan. Byzantine-robust federated learning with optimal statistical rates. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 3151–3178. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/zhu23b.html.

These appendices provide the future directions, numerical experiments and the technical proofs of the main manuscript.

A LIMITATIONS AND DIRECTIONS FOR FUTURE RESEARCH

Optimal projection. As discussed in Remark 1, our bounds are not fully minimax optimal: ignoring logarithmic factors, the contamination term (in equation 12) scales as $p\epsilon^2/n$, compared with the optimal term ϵ^2/n (Yin et al., 2018). Existing optimal frameworks typically use a low-dimensional projection of the parameter to circumvent the dimensional curse (Diakonikolas et al., 2022; 2023). Designing online projection methods that achieve the optimal rate, particularly in high dimensions, remains a direction for future research.

More general models. Our analysis currently covers Gaussian location models and linear regression. We expect the bidirectional transfer principle to extend to generalized linear models, single-index models, and certain nonparametric models. Such extensions require carefully designed truncation and thresholding operators that balance robustness in central-model update with effective transfer to the current-model learning.

Heavy-tailedness. The current theory relies on the sub-Gaussian noise Assumption 2, whereas many studies in robust learning allow heavy-tailed distributions under weaker moment assumptions (Sun et al., 2020; Oliveira & Resende, 2025). Existing offline results indicate that the optimal contamination term is not of the order ϵ^2/n in these settings (Zhu et al., 2023; Oliveira & Resende, 2025). Therefore, extending our framework to heavy-tailed data and characterizing the resulting rates are interesting directions for future research.

B ADDITIONAL EXPERIMENTS

This appendix provides additional experiments under a Gaussian-shift setting, together with some experiments with higher dimensions.

B.1 GAUSSIAN SHIFTS

In each heterogeneous batches, we set $\boldsymbol{\beta}^{(b)*} = \boldsymbol{\beta}^* + \boldsymbol{\Delta}^{(b)}$, with independent $\Delta_j^{(b)} \sim N(0, s^2)$ shifts across coordinates. The other settings, including the design matrices, compared methods, evaluation protocol, and C_b, H_b metrics, are all the same as in Section 4 in the main manuscript. We set $p = 10$, $n = 100$, $B = 50$, and $\epsilon = 0.2$, and examine Gaussian shifts with $s = 0.25, 0.5, 1, 2$ in experiments E5–E8, respectively.

Table 2 reports the results over 50 Monte Carlo replications. Shaded rows denote offline benchmarks. The proposed method achieves the smallest H_b among all online methods at every checkpoint, while attaining the lowest or second-lowest central-model estimation error C_b. Although the Renewable QIF method achieves lower central-model error C_b in some settings, it does not reliably identify heterogeneous batches, resulting in relatively higher heterogeneous-model estimation error H_b.

B.2 HIGHER DIMENSIONS AND VARYING CONTAMINATION PROBABILITY

We set $p = 50$, $n = 200$, $B = 100$, and $s = 0.5$. The pairs E9/E10, E11/E12, and E13/E14 have contamination probabilities 0.05, 0.15, and 0.25, respectively. Across these six experiments, heterogeneous batches in odd-numbered cases have shifts restricted to the same ten coordinates, $j = 21, \ldots, 30$ (*Gaussian fixed10*), whereas those in even-numbered cases have shifts in all coordinates (*Gaussian full*).

Table 3 reports the experimental results. The proposed method achieves the smallest heterogeneous errors H_b among online methods in all case–checkpoint combinations, and achieves the smallest central errors C_b at the later checkpoints $b = 50, 100$, with those C_b close to the offline benchmarks. Although the Renewable Huber method achieves slightly lower central errors C_b at $b = 25$ in E9 and

Table 2: Central and heterogeneous-model errors over 50 replications under Gaussian shifts: $p = 10$, $n = 100$, $B = 50$, and $\epsilon = 0.2$. Bold and underlined values indicate the lowest and second-lowest errors among online methods, respectively.

Method	**E5**: $s = 0.25$				**E6**: $s = 0.5$			
	$b = 25$		$b = 50$		$b = 25$		$b = 50$	
	C_b	H_b	C_b	H_b	C_b	H_b	C_b	H_b
Proposed	**0.0108**	**0.1845**	**0.0049**	**0.1796**	**0.0119**	**0.1698**	<u>0.0063</u>	**0.1760**
Renewable OLS	0.0124	0.5416	0.0060	0.5811	0.0272	2.0403	0.0147	2.2110
Renewable QIF	0.0117	0.4464	0.0055	0.4777	0.0135	0.5536	**0.0057**	0.5588
Renewable Huber	0.0118	0.5498	0.0055	0.5866	0.0187	2.1223	0.0097	2.2668
LAD SGD	0.0177	0.6636	0.0080	0.6552	0.0279	2.4002	0.0135	2.4498
Online quantile	0.0179	0.9943	0.0086	0.7840	0.0227	2.7108	0.0117	2.5449
ARMUL	0.0083	0.1598	0.0037	0.1557	0.0086	0.2159	0.0048	0.2279
Oracle	0.0075	0.1843	0.0037	0.1806	0.0079	0.1698	0.0042	0.1760
Method	**E7**: $s = 1$				**E8**: $s = 2$			
	$b = 25$		$b = 50$		$b = 25$		$b = 50$	
	C_b	H_b	C_b	H_b	C_b	H_b	C_b	H_b
Proposed	<u>0.0145</u>	**0.1878**	<u>0.0073</u>	**0.1827**	**0.0176**	**0.1752**	**0.0075**	**0.1796**
Renewable OLS	0.0839	7.6950	0.0500	8.8212	0.4056	33.4888	0.2038	35.7817
Renewable QIF	**0.0139**	0.6955	**0.0066**	0.5656	0.0341	1.3894	0.0129	1.1209
Renewable Huber	0.0281	8.4288	0.0152	9.2994	0.0437	38.3806	0.0208	38.7529
LAD SGD	0.0395	9.4471	0.0180	9.9164	0.0518	41.0436	0.0244	40.3213
Online quantile	0.0289	9.6366	0.0143	9.8602	0.0365	40.6551	0.0187	39.8117
ARMUL	0.0104	0.2039	0.0053	0.1949	0.0123	0.3342	0.0052	0.3249
Oracle	0.0081	0.1878	0.0041	0.1827	0.0085	0.1752	0.0038	0.1796

E12, it does not provide batch-specific estimates for heterogeneous models, resulting in substantially larger H_b.

At $b = 100$, our method reduces H_b relative to the Oracle method by approximately 9.7%, 9.9%, and 9.3% in E9, E11, and E13 cases, respectively. This result supports the adaptive transfer mechanism: in the *fixed10* settings, shifts affect only ten coordinates, leaving heterogeneous models sufficiently close to the central model. Therefore, transfer learning can yields lower estimation error than local OLS.

C Proofs for Section 2

We give the scalar proof (Proof of Theorem 2) in a form that can be reused coordinatewise in the regression model. Only two terms need to be verified again in Appendix D. This avoids repeating the induction.

C.1 Operator inequalities

For completeness, we state and prove the general vector form of Lemma 1. For vectors, recall the definition: $\mathbf{F}_a(\mathbf{z}) := F_a(\|\mathbf{z}\|_2) \cdot \mathbf{z}/\|\mathbf{z}\|_2$ when $\mathbf{z} \neq \mathbf{0}$, and $\mathbf{F}_a(\mathbf{0}) := \mathbf{0}$.

Lemma 5 (Firm transfer in any dimension)**.** *Let $a > 0$. If $\|\mathbf{y} - \boldsymbol{\theta}\|_2 \leq a$, then for every proposed center $\mathbf{m}$,*

$$\|\mathbf{m} + \mathbf{F}_a(\mathbf{y} - \mathbf{m}) - \boldsymbol{\theta}\|_2 \leq \min\{\|\mathbf{m} - \boldsymbol{\theta}\|_2, 2a\}.$$

Proof of Lemma 5. Define

$$\mathbf{R}_a(\mathbf{z}) := \mathbf{z} - \mathbf{F}_a(\mathbf{z}) = \begin{cases} \mathbf{z}, & \|\mathbf{z}\| \leq a, \\ (2a/\|\mathbf{z}\| - 1)\,\mathbf{z}, & a < \|\mathbf{z}\| < 2a, \\ \mathbf{0}, & \|\mathbf{z}\| \geq 2a. \end{cases}$$

Table 3: Higher-dimensional Gaussian shifts: $p = 50$, $n = 200$, $B = 100$, and $s = 0.5$.

Method	$b = 25$		$b = 50$		$b = 100$	
	C_b	H_b	C_b	H_b	C_b	H_b
E9: $\epsilon = 0.05$, Gaussian fixed10						
Proposed	0.0243	**0.4961**	**0.0118**	**0.5097**	**0.0059**	**0.4980**
Renewable OLS	**0.0235**	1.9794	0.0124	2.2684	0.0060	2.3676
Renewable QIF	**0.0235**	1.9794	0.0124	2.2684	0.0060	2.3676
Renewable Huber	**0.0235**	1.9798	0.0124	2.2686	0.0060	2.3678
LAD SGD	0.0489	2.4641	0.0206	2.5905	0.0091	2.5625
Online quantile	0.0437	1.9426	0.0226	2.2046	0.0116	2.3064
ARMUL	0.0179	0.6221	0.0090	0.6160	0.0045	0.6212
Oracle	0.0174	0.5693	0.0085	0.5620	0.0043	0.5515
E10: $\epsilon = 0.05$, Gaussian full						
Proposed	**0.0269**	**0.5459**	**0.0131**	**0.5477**	**0.0066**	**0.5422**
Renewable OLS	0.0516	10.1227	0.0249	11.1778	0.0126	11.6086
Renewable QIF	0.0513	10.0023	0.0249	11.1148	0.0126	11.5147
Renewable Huber	0.0498	10.1593	0.0238	11.2086	0.0121	11.6278
LAD SGD	0.0609	12.0936	0.0240	12.4985	0.0103	12.4085
Online quantile	0.0486	12.6282	0.0230	12.4744	0.0116	12.2191
ARMUL	0.0181	1.0567	0.0090	1.0926	0.0046	1.0607
Oracle	0.0170	0.5478	0.0087	0.5482	0.0044	0.5425
E11: $\epsilon = 0.15$, Gaussian fixed10						
Proposed	**0.0266**	**0.4873**	**0.0132**	**0.4935**	**0.0065**	**0.4952**
Renewable OLS	0.0370	2.0728	0.0184	2.1905	0.0092	2.3047
Renewable QIF	0.0370	2.0728	0.0184	2.1905	0.0092	2.3047
Renewable Huber	0.0284	2.2138	0.0141	2.2789	0.0070	2.3623
LAD SGD	0.0640	2.5559	0.0258	2.4859	0.0111	2.4916
Online quantile	0.0594	4.3289	0.0287	3.2225	0.0130	2.8070
ARMUL	0.0206	0.6028	0.0103	0.5871	0.0051	0.5886
Oracle	0.0195	0.5339	0.0098	0.5454	0.0048	0.5494
E12: $\epsilon = 0.15$, Gaussian full						
Proposed	0.0365	**0.5366**	**0.0185**	**0.5491**	**0.0089**	**0.5458**
Renewable OLS	0.1170	10.0246	0.0648	10.9241	0.0324	11.4682
Renewable QIF	0.1170	10.0246	0.0640	10.7686	0.0320	11.3758
Renewable Huber	**0.0360**	11.3033	0.0188	11.7070	0.0092	11.9601
LAD SGD	0.0882	11.9549	0.0365	12.1145	0.0155	12.2235
Online quantile	0.0678	10.6576	0.0365	11.1853	0.0156	11.5847
ARMUL	0.0236	1.0164	0.0118	1.0434	0.0059	1.0417
Oracle	0.0200	0.5362	0.0102	0.5493	0.0051	0.5457
E13: $\epsilon = 0.25$, Gaussian fixed10						
Proposed	**0.0298**	**0.5090**	**0.0142**	**0.5056**	**0.0070**	**0.5063**
Renewable OLS	0.0492	2.0385	0.0249	2.2578	0.0119	2.3615
Renewable QIF	0.0492	2.0385	0.0249	2.2578	0.0119	2.3615
Renewable Huber	0.0365	2.1731	0.0174	2.3441	0.0085	2.4146
LAD SGD	0.0786	2.5627	0.0325	2.5785	0.0141	2.5570
Online quantile	0.0706	2.0000	0.0349	2.1883	0.0163	2.2973
ARMUL	0.0230	0.5984	0.0115	0.5972	0.0057	0.5972
Oracle	0.0210	0.5604	0.0107	0.5549	0.0053	0.5580
E14: $\epsilon = 0.25$, Gaussian full						
Proposed	**0.0471**	**0.5492**	**0.0234**	**0.5556**	**0.0114**	**0.5568**
Renewable OLS	0.1901	10.8070	0.0972	11.4425	0.0474	11.8353
Renewable QIF	0.1916	10.7312	0.0976	11.3900	0.0475	11.7786
Renewable Huber	0.0486	11.9494	0.0248	12.1755	0.0119	12.2834
LAD SGD	0.1211	12.7415	0.0523	12.7147	0.0219	12.6187
Online quantile	0.0941	13.1990	0.0417	12.6196	0.0204	12.4075
ARMUL	0.0280	0.6313	0.0146	0.6309	0.0072	0.6285
Oracle	0.0225	0.5492	0.0116	0.5556	0.0057	0.5568

Its radial and tangential derivatives have absolute values at most one; hence $\|\mathbf{R}_a(\mathbf{z}) - \mathbf{R}_a(\mathbf{z}')\|_2 \le \|\mathbf{z} - \mathbf{z}'\|_2$ uniformly. Therefore we have

$$\begin{aligned} &\|\mathbf{m} + \mathbf{F}_a(\mathbf{y} - \mathbf{m}) - \boldsymbol{\theta}\|_2 \\ =& \|[\mathbf{y} - \boldsymbol{\theta} - \mathbf{F}_a(\mathbf{y} - \boldsymbol{\theta})] - [\mathbf{y} - \mathbf{m} - \mathbf{F}_a(\mathbf{y} - \mathbf{m})]\|_2 \\ =& \|\mathbf{R}_a(\mathbf{y} - \boldsymbol{\theta}) - \mathbf{R}_a(\mathbf{y} - \boldsymbol{m})\|_2 \\ \le& \|\mathbf{m} - \boldsymbol{\theta}\|_2 . \end{aligned}$$

Moreover we have $\|\mathbf{R}_a(\mathbf{z})\|_2 \le a$ uniformly. Then, by the fact $\mathbf{m} + \mathbf{F}_a(\mathbf{y} - \mathbf{m}) = \mathbf{y} - \mathbf{R}_a(\mathbf{y} - \mathbf{m})$, we have

$$\|\mathbf{y} - R_a(\mathbf{y} - \mathbf{m}) - \boldsymbol{\theta}\|_2 \le \|\mathbf{y} - \boldsymbol{\theta}\|_2 + \|R_a(\mathbf{y} - \mathbf{m})\|_2 \le 2a,$$

which proves the second bound and completes the proof of Lemma 5. □

Lemma 6 (Projection bound)**.** *For any closed interval I containing zero and any scalars x, z,*

$$|\Pi_I(x) - z| \le \max\{|x - z|, |z|\}. \tag{15}$$

Proof of Lemma 6. Assume $I = [\ell, u],\ \ell \le 0 \le u$. Then

$$|\Pi_I(x) - z| = \begin{cases} |x - z|, & \ell \le x \le u, \\ |u - z| \le \begin{cases} |x - z|, & z \le u, \\ |z|, & z > u, \end{cases} & x > u, \\ |\ell - z| \le \begin{cases} |x - z|, & z \ge \ell, \\ |z|, & z < \ell, \end{cases} & x < \ell. \end{cases}$$

Therefore we complete the proof of Lemma 6. □

C.2 A SCALAR RECURSION

This subsection develops the scalar results used for the Gaussian batch means (Section 2) and, coordinatewise, for the regression OLS summaries (Section 3).

Preliminary. Consider the summary statsitic of the b-th batch is $Y^{(b)}$, which is a local estimation of parameter $\beta^{(b)*} \in \mathbb{R}$. For point estimation, we have $Y^{(b)} = \overline{Z}^{(b)}$. Write $\mathcal{F}^{(b-1)}$ for the history available before batch b, including the first-batch summary $Y^{(1)}$. Given the history of previous $b-1$ batches, the next summary belongs to the central model with probability $1 - \epsilon$, where $0 \le \epsilon \le 1/4$; the heterogeneous model (with probability ϵ) is unrestricted. Fix a calibration scale $\nu > 0$, a reference radius $L > 0$, and a deterministic *nondecreasing* sequence of clipping levels $\tau_b > 0$. Let Π project onto $[Y^{(1)} - \nu L, Y^{(1)} + \nu L]$ and set

$$m^{(1)} = Y^{(1)}, \qquad m^{(b)} = \Pi\left[m^{(b-1)} + \frac{2\nu}{b}\Psi_{\tau_b}((Y^{(b)} - m^{(b-1)})/\nu)\right], \quad b \ge 2. \tag{16}$$

On a central-component batch (where $\beta^{(b)*} = \beta^*$), let $W^{(b)} := (Y^{(b)} - \beta^*)/\nu$. Its expected clipped score, conditional on the history, is

$$g^{(b)}(u) = \mathbb{E}[\Psi_{\tau_b}(W^{(b)} - u) \mid \mathcal{F}^{(b-1)},\ \beta^{(b)*} = \beta^*].$$

Assume that there exists a deterministic $d_b \ge 0$ bound the clipping bias and that:

$$|g^{(b)}(0)| \le d_b, \qquad \frac{2}{3} \le \frac{g^{(b)}(0) - g^{(b)}(u)}{u} \le 1, \qquad 0 < |u| \le 2L, \tag{17}$$

where the bounds of the slope will be proved in Gaussian location model and linear regression respectively. For the Gaussian point estimation, symmetry gives $d_b = 0$ first. Though linear regression allows asymmetric noise, but equation 17 still holds and the sum $\sum_{b \ge 2} d_b$ is finite by the sub-Gaussianity (proved in Appendix C.3). The cumulative centered score is $M^{(1)} := 0$ and

$$M^{(b)} := \sum_{k=2}^{b} \left\{\Psi_{\tau_k}((Y^{(k)} - m^{(k-1)})/\nu) - \mathbb{E}[\Psi_{\tau_k}((Y^{(k)} - m^{(k-1)})/\nu) \mid \mathcal{F}^{(k-1)}]\right\}. \tag{18}$$

Lemma 7 (Error decomposition for the projected recursion)**.** *Under equation 16–equation 17, if $|Y^{(1)} - \beta^*| \le \nu L$, then simultaneously for every $b \ge 2$, we have*

$$|m^{(b)} - \beta^*| \le 2\nu\epsilon\tau_b + \frac{2\nu}{b}\sum_{k=2}^{b} d_k + \frac{2\nu L}{b} + \frac{4\nu}{b}\max_{2\le k\le b}|M^{(k)}|. \tag{19}$$

The four terms respectively represent the contamination contribution, accumulated clipping bias, initialization error, and centered random fluctuation.

Proof of Lemma 7. The initialization event makes the projection interval contain β^*, so $|(m^{(b)} - \beta^*)/\nu| \le 2L$ at every batch-learning. For $u \in \mathbb{R}$, define the conditional mixture score

$$\mu_b(u) = \mathbb{E}[\Psi_{\tau_b}((Y^{(b)} - \beta^*)/\nu - u) \mid \mathcal{F}^{(b-1)}].$$

By equation 17, we have $g^{(b)}(u) = g^{(b)}(0) - u \cdot q_b(u)$ with some function $q_b(u) \in [2/3, 1]$ for every $-2L \le u \le 2L$. Define $\widetilde{g}^{(b)}(u) := \mathbb{E}[\Psi_{\tau_b}((Y^{(b)} - \beta^*)/\nu - u) \mid \mathcal{F}^{(b-1)}, \beta^{(b)*} \ne \beta^*]$ as the conditional expected score on the contaminating component. Clipping gives $|\widetilde{g}^{(b)}(u)| \le \tau_b$. Additionally, define

$$\zeta_b(u) := (1-\epsilon)g^{(b)}(0) + \epsilon\widetilde{g}^{(b)}(u), \tag{20}$$

we have $|\zeta_b(u)| \le (1-\epsilon)d_b + \epsilon\tau_b \le d_b + \epsilon\tau_b$, and we then write

$$\mu_b(u) = (1-\epsilon)g^{(b)}(u) + \epsilon\widetilde{g}^{(b)}(u) = -(1-\epsilon)u \cdot q_b(u) + \zeta_b(u). \tag{21}$$

Set $\lambda_b = b - 2(1-\epsilon)\cdot q_b(u)$. Since $b \ge 2$, $q_b \in [2/3, 1]$, and $\epsilon \le 1/4$, we have:

$$0 \le b - 2 \le \lambda_b, \qquad \lambda_b \le b - 2\cdot\frac{3}{4}\cdot\frac{2}{3} = b - 1.$$

Thus $\alpha_b := \lambda_b/(b-1) \in [0, 1]$. For every $z \in \mathbb{R}$, we have the following contraction:

$$\begin{aligned}
&\left|bu + 2\mathbb{E}[\Psi_{\tau_b}((Y^{(b)} - \beta^*)/\nu - u) \mid \mathcal{F}^{(b-1)}] - z\right| \\
&= |bu + 2\mu_b(u) - z| \\
&= |\lambda_b u + 2\zeta_b(u) - z| \\
&\le |\lambda_b u - z| + 2|\zeta_b(u)| \\
&= |\alpha_b\{(b-1)u - z\} - (1-\alpha_b)z| + 2|\zeta_b(u)| \\
&\le \alpha_b|(b-1)u - z| + (1-\alpha_b)|z| + 2|\zeta_b(u)| \\
&\le \max\{|(b-1)u - z|, |z|\} + 2(\tau_b\epsilon + d_b),
\end{aligned} \tag{22}$$

where the first two equalities follows from equation 20 and equation 21. The last inequality follows from $|\zeta_b(u)| \le d_b + \epsilon\tau_b$.

To pass from the conditional drift to the current-batch update, we take

$$u = \frac{m^{(b-1)} - \beta^*}{\nu}, \qquad \psi_b = \Psi_{\tau_b}\left(\frac{Y^{(b)} - m^{(b-1)}}{\nu}\right), \qquad \bar{\psi}_b = \mathbb{E}(\psi_b \mid \mathcal{F}^{(b-1)}). \tag{23}$$

Then equation 18 lead that $M^{(b)} = M^{(b-1)} + \psi_b - \bar{\psi}_b$. Let $\mathcal{I} = [Y^{(1)} - \nu L, Y^{(1)} + \nu L]$ be the initial projection interval and let $\mathcal{J}_b := (b/\nu)(\mathcal{I} - \beta^*)$. Successful initialization implies $0 \in \mathcal{J}_b$, and by the update in equation 16, we have

$$\frac{b}{\nu}(m^{(b)} - \beta^*) = \Pi_{\mathcal{J}_b}(bu + 2\psi_b).$$

Consequently, for each batch index $b \geq 2$, we have

$$\begin{aligned}
&\left|\frac{b}{\nu}(m^{(b)} - \beta^*) - 2M^{(b)}\right| \\
&= \left|\Pi_{\mathcal{J}_b}(bu + 2\psi_b) - 2M^{(b)}\right| \\
&\leq \max\left\{|bu + 2\psi_b - 2M^{(b)}|,\ 2|M^{(b)}|\right\} \\
&= \max\left\{|bu + 2\bar{\psi}_b - 2M^{(b-1)}|,\ 2|M^{(b)}|\right\} \\
&\overset{(i)}{\leq} \max\Big\{\max\{|(b-1)u - 2M^{(b-1)}|,\ 2|M^{(b-1)}|\} + 2(\tau_b\epsilon + d_b),\ 2|M^{(b)}|\Big\} \\
&\leq 2(\tau_b\epsilon + d_b) + \max\left\{\left|(b-1)\frac{m^{(b-1)} - \beta^*}{\nu} - 2M^{(b-1)}\right|,\ 2|M^{(b-1)}|,\ 2|M^{(b)}|\right\}, \qquad (24)
\end{aligned}$$

where the first inequality follows from Lemma 6, the following equality uses $\psi_b - M^{(b)} = \bar{\psi}_b - M^{(b-1)}$. Inequality (i) follows from taking equation 23 in to equation 22, with $z = 2M^{(b-1)}$.

By iteratively using equation 24, with $M^{(1)} = 0$, we have

$$\left|b\frac{m^{(b)} - \beta^*}{\nu} - 2M^{(b)}\right| \leq 2\sum_{k=2}^{b}(\tau_k\epsilon + d_k) + 2\max\left\{\frac{|Y^{(1)} - \beta^*|}{\nu},\ \max_{1\leq k\leq b}|M^{(k)}|\right\}, \qquad (25)$$

for every $b \geq 2$. Finally, by adding the $2|M^{(b)}|$ term, using the initialization $|Y^{(1)} - \beta^*|/\nu \leq L$, and bounding $\sum_{k=2}^{b}\tau_k \leq b\tau_b$, we complete the proof of equation 19. □

The last term in equation 19 is a running maximum of the centered-score sum in equation 18. The next lemma makes this term explicit (with a high probability).

Lemma 8 (A joint bound for centered scores)**.** *For the centered score in equation 18, a nondecreasing clipping sequence $\{\tau_b\}_{b\geq 2}$, and any $\eta \in (0,1)$, with probability at least $1-\eta$, simultaneously for every $b \geq 2$,*

$$\max_{1\leq k\leq b}|M^{(k)}| \leq \tau_b\sqrt{2(b-1)\log(2b^2/\eta)}. \qquad (26)$$

Proof of Lemma 8. The score at batch b, i.e., $\Psi_{\tau_k}((Y^{(k)} - m^{(k-1)})/\nu)$, lies in $[-\tau_b, \tau_b]$, so the conditional support of its centered increment has length at most $2\tau_b$. By the Hoeffding exponential-moment inequality (Wainwright, 2019) and the tower property, we have

$$\mathbb{E}[e^{t(M^{(b)} - M^{(b-1)})} \mid \mathcal{F}^{(b-1)}] \leq e^{t^2\tau_b^2/2}, \qquad \mathbb{E}e^{tM^{(b)}} \leq e^{t^2\sum_{k=2}^{b}\tau_k^2/2} \leq e^{(b-1)t^2\tau_b^2/2},$$

for every $t \in \mathbb{R}$. Then by using the Chernoff bound with the probability union bound, we have

$$\begin{aligned}
&\mathbb{P}\left(\exists k \geq 2 : |M^{(k)}| \geq \tau_k\sqrt{2(k-1)\log(2k^2/\eta)}\right) \\
\leq &\sum_{k\geq 2}\mathbb{P}\left(|M^{(k)}| > \tau_k\sqrt{2(k-1)\log(2k^2/\eta)}\right) \\
\leq &\sum_{k\geq 2} 2\exp\left(-\frac{2\tau_k^2(k-1)\log(2k^2/\eta)}{2\tau_k^2(k-1)}\right) \ \leq \ \sum_{k\geq 2}\eta/k^2 \ \leq \ \eta.
\end{aligned}$$

Therefore, we complete the proof of Lemma 8. □

C.3 Proof of Theorem 2

Here we only need to verify the conditions required in Lemma 7. Apply the scalar setup with $Y^{(b)} = \bar{Z}^{(b)}$, $\nu = n^{-1/2}$, $L = \sqrt{2\log(6/\delta)}$, and the fixed clipping level $\tau_b = c_\delta = 1 + 2\sqrt{2\log(6/\delta)}$.

Verify Assumption in equation 17. For $Z \sim N(0,1)$, symmetry leads $\mathbb{E}\Psi_{c_\delta}(Z) = 0$ and we can take $d_b = 0$ for each $b \geq 2$. We next verify the restoring slope: For $u \neq 0$, define the pointwise secant

$$D_u(z) = \frac{\Psi_{c_\delta}(z) - \Psi_{c_\delta}(z-u)}{u}.$$

Monotonicity and the 1-Lipschitz property of Ψ_{c_δ} imply $0 \leq D_u(z) \leq 1$. If $|u| \leq c_\delta - 1 = 2\sqrt{2\log(6/\delta)} = 2L$ and $|z| \leq 1$, then both z and $z-u$ belong to $[-c_\delta, c_\delta]$, so $R_u(z) = 1$ and thus $\mathbf{1}\{|z| \leq 1\} \leq R_u(z) \leq 1$. Taking expectations, by the definition $g^{(b)}(u) = \mathbb{E}\Psi_{c_\delta}(Z-u)$, we have

$$2/3 < 0.6827 = \mathbb{P}(|Z| \leq 1) \leq \frac{g^{(b)}(0) - g^{(b)}(u)}{u} \leq 1, \text{ for every } |u| \leq 2L = 2\sqrt{2\log(6/\delta)}. \tag{27}$$

Therefore, in the Gaussian point estimation example, the assumption equation 17 holds with $\tau_b = c_\delta$, $d_b = 0$, $\nu = n^{-1/2}$, and $L = \sqrt{2\log(6/\delta)}$.

High-probability good event. Allocate a probability of $\delta/3$ each to initialization, cumulative centered scores, and current-batch noise: The Gaussian tail bound gives initialization failure at most $\delta/3$:

$$\mathbb{P}\left(\beta^* \notin \left[\bar{Z}^{(1)} - \sqrt{2\log(6/\delta)/n},\ \bar{Z}^{(1)} + \sqrt{2\log(6/\delta)/n}\right]\right) \leq \delta/3. \tag{28}$$

Lemma 8 with $\eta = \delta/3$ controls all centered-score maxima:

$$\mathbb{P}\left(\bigcup_{b\geq 2}\left\{\max_{1\leq k\leq b}|M^{(k)}| \geq \tau_b\sqrt{2(b-1)\log(6b^2/\delta)}\right\}\right) \leq \delta/3. \tag{29}$$

The standard Gaussian tail inequality gives

$$\mathbb{P}\left(\bigcup_{b\geq 2}\left\{|\bar{Z}^{(b)} - \beta^{(b)*}| > \sqrt{\frac{2}{n}\log(6b^2/\delta)}\right\}\right) \leq \sum_{b\geq 2}\delta/(3b^2) \leq \delta/3, \tag{30}$$

and for simplicity we take $a^{(b)} = \sqrt{\frac{2}{n}\log(6b^2/\delta)} < 2\sqrt{\frac{\log(3b/\delta)}{n}}$. These three events equation 28–equation 30 need not be independent. Their complements hold simultaneously with probability at least $1-\delta$.

Main proof. Here we prove the bounds in Theorem equation 2. The following analysis is carried out on the intersection of the complements of the events in equation 28–equation 30. Recall $L = \sqrt{2\log(6/\delta)}$ and $c_\delta = 1 + 2L$. Apply equation 19 directly with every $d_k = 0$, and use Lemma 8 with $\eta = \delta/3$. Since $\log(6k^2/\delta) < 2\log(3k/\delta)$, $\sqrt{\log(3k/\delta)/k} \geq 1/k$, and $c_\delta < 4\sqrt{\log(6/\delta)}$, for every $k \geq 2$, we obtain

$$\begin{aligned}
\sqrt{n}|m^{(k)} - \beta^*| &\leq 2c_\delta\epsilon + \frac{2L}{k} + \frac{4c_\delta}{k}\sqrt{2(k-1)\log(6k^2/\delta)} \\
&\leq 2c_\delta\epsilon + \frac{c_\delta}{k} + 8c_\delta\sqrt{\frac{\log(3k/\delta)}{k}} \\
&< c_\delta\left(2\epsilon + 9\sqrt{\frac{\log(3k/\delta)}{k}}\right) \\
&< c_\delta\sqrt{85}\sqrt{\epsilon^2 + \frac{\log(3k/\delta)}{k}}.
\end{aligned} \tag{31}$$

Squaring equation 31 proves the second inequality in equation 5. By using Lemma 1, for every $b \geq 2$, we have

$$\begin{aligned}
|\hat{\beta}^{(b)} - \beta^{(b)*}|^2 &= \left|\hat{m}^{(b-1)} + F_{a^{(b)}}(\bar{Z}^{(b)} - \hat{m}^{(b-1)}) - \beta^{(b)*}\right|^2 \\
&\leq \min\left\{4(a^{(b)})^2,\ \left(|\hat{m}^{(b-1)} - \beta^*| + |\beta^{(b)*} - \beta^*|\right)^2\right\} \\
&\asymp \min\left\{\frac{\log(b^2/\delta)}{n},\ |\hat{m}^{(b-1)} - \beta^*|^2 + |\beta^{(b)*} - \beta^*|^2\right\}.
\end{aligned} \tag{32}$$

Then, by applying the inequality equation 31 (with $k = b - 1$) into equation 32, we complete the proof of equation 6.

D PROOFS FOR THE LINEAR MODEL

We now prove results in Section 3. The constants used in Section 3 are defined at their first occurrence, with explicit choices specified in the proof. To simplify notation, we write $\mathbf{m}^{(b)} = \hat{\mathbf{m}}^{(b)}$ and $r^{(b)} = r_b$ throughout.

D.1 THE DESIGN EVENT AND LOCAL OLS ERRORS

Set $\hat{\mathbf{\Sigma}}^{(b)} = \mathbf{X}^{(b)\top}\mathbf{X}^{(b)}/n$ and $\nu := \sigma\sqrt{2/(\kappa_0 n)}$. The sub-Gaussian Assumption 1 gives the identity

$$(\mathbf{\Sigma}^{(b)})^{-1/2}\hat{\mathbf{\Sigma}}^{(b)}(\mathbf{\Sigma}^{(b)})^{-1/2} = \frac{1}{n}\mathbf{Z}^{(b)\top}\mathbf{Z}^{(b)}. \tag{33}$$

The standard sample-covariance inequality for independent 1-sub-Gaussian rows (e.g., Remark 5.40 in Vershynin (2010)) gives universal constants $c, C > 0$ such that

$$\mathbb{P}\left(\left\|\frac{1}{n}\mathbf{Z}^{(b)\top}\mathbf{Z}^{(b)} - \mathbf{I}_p\right\|_{\mathrm{op}} > \frac{1}{2}\right) \le 2e^{-cn}, \qquad n \ge Cp. \tag{34}$$

Take $C_6 := \max\{C, (\log 8)/c\}$. Define $\mathcal{E}_X^{(b)} := \bigcap_{k=1}^{b}\left\{\frac{1}{2}\mathbf{\Sigma}^{(k)} \preceq \hat{\mathbf{\Sigma}}^{(k)} \preceq \frac{3}{2}\mathbf{\Sigma}^{(k)}\right\}$, and we have $\mathcal{E}_X^{(B)} = \bigcap_{b=1}^{B}\mathcal{E}_X^{(b)}$. Under the condition $n \ge C_6[p+\log(B/\delta)]$ (sample size assumption in Theorem 3), by equation 33 and equation 34, we have

$$\mathbb{P}((\mathcal{E}_X^{(B)})^c) \le 2Be^{-cn} \le \delta/4. \tag{35}$$

σ algebraic flow. Define the design history and the observable history by

$$\mathsf{X}^{(k)} := (\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(k)}), \qquad \mathcal{F}_k := \sigma(\mathbf{X}^{(s)},\ \mathbf{y}^{(s)} :\ 1 \le s \le k). \tag{36}$$

For batch $b \ge 2$, let

$$\mathcal{K}_b := \sigma(\mathcal{F}_{b-1},\ \mathbf{X}^{(b)}), \qquad \mathcal{F}_{b-1} \subseteq \mathcal{K}_b \subseteq \mathcal{F}_b. \tag{37}$$

Thus $\hat{\mathbf{m}}^{(b-1)}$ is $\mathcal{F}_{b-1}$-measurable, whereas the local OLS $\widetilde{\boldsymbol{\beta}}^{(b)}$ and updated outputs $\hat{m}^{(b)}$ are $\mathcal{F}_b$-measurable.

OLS errors. Let $\mathbf{e}^{(b)} := \widetilde{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^{(b)*} = (\mathbf{X}^{(b)\top}\mathbf{X}^{(b)})^{-1}\mathbf{X}^{(b)\top}\boldsymbol{\xi}^{(b)} \in \mathbb{R}^p$. For deterministic $\mathbf{v} \in \mathbb{R}^p$, the vector $\mathbf{a} := \mathbf{X}^{(b)}(\mathbf{X}^{(b)\top}\mathbf{X}^{(b)})^{-1}\mathbf{v}$ is fixed when conditioning on $\mathcal{K}_b$. For all $t \in \mathbb{R}$, we have

$$\begin{aligned}\mathbb{E}[e^{t\mathbf{v}^\top\mathbf{e}^{(b)}} \mid \mathcal{K}_b, A_b, \mathcal{E}_X^{(b)}] &= \mathbb{E}[e^{t\mathbf{a}^\top\boldsymbol{\xi}^{(b)}} \mid \mathcal{K}_b, A_b, \mathcal{E}_X^{(b)}] \le \exp\left(\frac{\sigma^2 t^2}{2}\|\mathbf{a}\|_2^2\right) \\ &= \exp\left(\frac{\sigma^2 t^2}{2}\mathbf{v}^\top(\mathbf{X}^{(b)\top}\mathbf{X}^{(b)})^{-1}\mathbf{v}\right) \le \exp\left(\frac{\nu^2 t^2}{2}\|\mathbf{v}\|_2^2\right).\end{aligned} \tag{38}$$

The conditional mean of $\mathbf{e}^{(b)}$ is zero. Thus $W_j^{(b)} := e_j^{(b)}/\nu$ is a conditional centered-and-1-sub-Gaussian random variable for every $i \in [p]$.

Set $C_1 = \sqrt{2}$, $C_2 = 16$ (appeared in Algorithm 2), and $L = \sqrt{2\log(16p/\delta)}$. For the initialization event $\mathcal{E}_0 := \{\max_{j\le p}|e_j^{(1)}| \le \nu L\}$,

$$\mathbb{P}(\mathcal{E}_0^c \mid \mathcal{E}_X^{(1)}) \le 2pe^{-L^2/2} = \delta/8, \qquad \mathbb{P}(\mathcal{E}_0^c \cup (\mathcal{E}_X^{(1)})^c) = \mathbb{P}((\mathcal{E}_X^{(1)})^c) + \mathbb{P}(\mathcal{E}_0^c \cap \mathcal{E}_X^{(1)}) \le 3\delta/8. \tag{39}$$

On $\mathcal{E}_0 \cap \mathcal{E}_X^{(1)}$, every reference interval contains β_j^*, so

$$|\hat{m}_j^{(b)} - \beta_j^*| \le 2\nu L, \qquad \text{for every } 1 \le j \le p,\ 1 \le b \le B. \tag{40}$$

For a given $\mathsf{X}^{(b)}$, from Theorem 2.1 of Hsu et al. (2012), we learn that

$$\mathbb{P}\Bigg\{\frac{1}{\sigma^2}\left\|\mathbf{e}^{(b)}\right\|_2^2 \geq tr\left((\mathbf{X}^{(b)\top}\mathbf{X}^{(b)})^{-1}\right) + 2\left\|(\mathbf{X}^{(b)\top}\mathbf{X}^{(b)})^{-1}\right\|_F\sqrt{u} + 2\left\|(\mathbf{X}^{(b)\top}\mathbf{X}^{(b)})^{-1}\right\|_2 u \;\Big|\; \mathsf{X}^{(b)}\Bigg\} \leq e^{-u}. \tag{41}$$

Under the event $\mathcal{E}_X^{(B)}$, for each $b \in [B]$, we further have

$$\begin{aligned}\left\|(\mathbf{X}^{(b)\top}\mathbf{X}^{(b)})^{-1}\right\|_2 &\leq 2/(\kappa_0 n),\\ tr\left((\mathbf{X}^{(b)\top}\mathbf{X}^{(b)})^{-1}\right) &\leq 2p/(\kappa_0 n),\\ \left\|(\mathbf{X}^{(b)\top}\mathbf{X}^{(b)})^{-1}\right\|_F^2 &\leq p\left\|(\mathbf{X}^{(b)\top}\mathbf{X}^{(b)})^{-1}\right\|_2^2 \leq 4p/(\kappa_0^2 n^2).\end{aligned}$$

Therefore, by taking $u = \log(8b^2/\delta)$ into equation 41, we have

$$\mathbb{P}\left(\bigcup_{b\in[B]}\left\{\left\|\mathbf{e}^{(b)}\right\|_2^2 \geq \frac{6\sigma^2}{\kappa_0 n}(p + \log(8b^2/\delta))\right\} \;\Big|\; \mathcal{E}_X^{(B)}\right) \leq \sum_{b\in[B]} \delta/(8b^2) \;\leq\; \delta/4. \tag{42}$$

We set $C_3 = \sqrt{3}$ and therefore $r_b = \sqrt{3}\nu\sqrt{p + \log(16b^2/\delta)}$ (in Algorithm 2 and equation 10). For $\mathcal{E}_{\text{loc}} := \bigcap_{b=1}^{B}\{\left\|\mathbf{e}^{(b)}\right\|_2 \leq r_b\}$, by equation 42 we have

$$\mathbb{P}(\mathcal{E}_{\text{loc}}^c \mid \mathcal{E}_X^{(B)}) < \delta/4. \tag{43}$$

which proves equation 11.

D.2 Conditional clipping bias and the slope bound

Set $C_4 := 4\sqrt{2}$ and $c_b := 4\sqrt{2}\sqrt{\log(16pb^2/\delta)}$ (in Algorithm 2). For $b \geq 2$, define the random function

$$g_{b,j}(u) = \mathbb{E}\left[\Psi_{c_b}\left(\frac{\widetilde{\beta}_j^{(b)} - \beta_j^*}{\nu} - u\right) \;\middle|\; \mathcal{K}_b, A_b = 0\right], \tag{44}$$

All assertions in this subsection hold almost surely on $\mathcal{E}_X^{(b)}$. Recall $W_j^{(b)} = e_j^{(b)}/\nu$, conditional on $\mathcal{K}_b, A_b = 0$, equation 38 gives $W_j^{(b)}$ a centering and a 1-sub-Gaussian tail. Hence

$$\begin{aligned}|g_{b,j}(0)| &\leq \mathbb{E}[(|W_j^{(b)}| - c_b)_+ \mid \mathcal{K}_b, A_b = 0]\\ &= \int_{c_b}^{\infty}\mathbb{P}(|W_j^{(b)}| > t \mid \mathcal{K}_b, A_b = 0)\,dt \leq 2\int_{c_b}^{\infty} e^{-t^2/2}\,dt \leq \frac{2e^{-c_b^2/2}}{c_b} =: d_b.\end{aligned} \tag{45}$$

Since $c_b \geq 2$, we have $d_b \leq b^{-2}/2$ and thus

$$\sum_{b=2}^{B} d_b < \sum_{b=2}^{B} e^{-16\log(16pb^2)} < 1. \tag{46}$$

For $0 < |u| \leq 2L$, define $Q_u(w) := \{\Psi_{c_b}(w) - \Psi_{c_b}(w-u)\}/u$. Monotonicity and the 1-Lipschitz property imply $0 \leq Q_u(w) \leq 1$. Since $2L \leq c_b/2$, we have $Q_u(w) = 1$ whenever $|w| \leq c_b/2$. Similar to equation 27, for every $0 < |u| \leq 2L$, by taking conditional expectations, almost surely on event $\mathcal{E}_X^{(b)}$ we have

$$\frac{2}{3} \leq 1 - 2e^{-c_b^2/8} \leq \mathbb{P}(|W| \leq c_b/2 \mid \mathcal{K}_b, A_b = 0) \leq \frac{g_{b,j}(0) - g_{b,j}(u)}{u} \leq 1. \tag{47}$$

Here $2e^{-c_b^2/8} \leq 2e^{-4\log(16)} < 1/3$. Finally, equation 39–equation 40 verify the initialization and interval range conditions based on the event $\mathcal{E}_0$. Thus all hypotheses needed for Lemma 7 hold on $\mathcal{E}_0 \cap \mathcal{E}_X^{(B)}$.

D.3 Proof of Theorem 3

On the original probability space, set

$$\psi_{b,j} = \Psi_{c_b}\left(\frac{\widetilde{\beta}_j^{(b)} - \hat{m}_j^{(b-1)}}{\nu}\right), \quad \bar{\psi}_{b,j} = \mathbb{E}[\psi_{b,j} \mid \mathcal{K}_b], \quad M_j^{(k)} = \sum_{s=2}^{k} \psi_{s,j} - \bar{\psi}_{s,j}, \quad M_j^{(1)} = 0.$$

Define

$$\mathcal{E}_M := \bigcap_{\substack{j\le p\\ 2\le b\le B}} \left\{ \max_{1\le k\le b} |M_j^{(k)}| \le c_b\sqrt{2(b-1)\log(8pb^2/\delta)} \right\}. \tag{48}$$

Applying Lemma 8 with $\eta = \delta/(4p)$, $\tau_b = c_b$, and taking a union bound over the p yield that

$$\mathbb{P}(\mathcal{E}_M^c) \le \mathbb{P}(\mathcal{E}_M^c|\mathcal{E}_X^{(B)}) + \mathbb{P}((\mathcal{E}_X^{(B)})^c) \le \sum_{j\in[p]}\sum_{k=2}^{B} \frac{\delta}{4pk^2} + \delta/4 \ \le \ \delta/2. \tag{49}$$

By taking $\tau_b = c_b$ into Lemma 7 and using equation 39, equation 46, and equation 48, on $\mathcal{E}_X^{(B)} \cap \mathcal{E}_0 \cap \mathcal{E}_M$, we have

$$\begin{aligned} |\hat{m}_j^{(b)} - \beta_j^*| &\le \nu\left[2\epsilon c_b + \frac{2+2L}{b} + \frac{4c_b\sqrt{2(b-1)\log(16pb^2/\delta)}}{b}\right] \\ &\le \nu c_b\left[2\epsilon + 7\sqrt{\frac{\log(16pb^2/\delta)}{b}}\right], \end{aligned} \tag{50}$$

where the last inequality follows from

$$2 + 2L = 2 + 2\sqrt{2\log(16p/\delta)} \le 4\sqrt{2}\sqrt{\log(16pb^2/\delta)} = c_b.$$

Then for every $b \ge 2$, applying the Cauchy-Schwarz inequality to equation 50 and summing over $j \in [p]$ yields

$$\begin{aligned} \left\|\hat{\mathbf{m}}^{(b)} - \boldsymbol{\beta}^*\right\|_2^2 &\le 53p\nu^2 c_b^2\left(\epsilon^2 + \frac{\log(16pb^2/\delta)}{b}\right) \\ &= 3392\frac{\sigma^2 p}{\kappa_0 n}\left(\epsilon^2\log(16pb^2/\delta) + \frac{\log^2(16pb^2/\delta)}{b}\right) =: \mathcal{H}_b^2. \end{aligned} \tag{51}$$

Therefore, we show that the constnat $C_5 = 3392$ (in equation 10), and complete the proof of equation 12 in Theorem 3.

On $\mathcal{E}_X^{(B)} \cap \mathcal{E}_0 \cap \mathcal{E}_{\mathrm{loc}} \cap \mathcal{E}_M$ (where recall the definition of $\mathcal{E}_{\mathrm{loc}}$ before equation 43), applying Lemma 5 gives

$$\begin{aligned} \left\|\hat{\boldsymbol{\beta}}^{(b)} - \boldsymbol{\beta}^{(b)*}\right\|_2^2 &\le \min\left\{4r_b^2,\ \left\|\hat{\mathbf{m}}^{(b-1)} - \boldsymbol{\beta}^{(b)*}\right\|_2^2\right\} \\ &\le \min\left\{4r_b^2,\ \left(\mathcal{H}_{b-1} + \left\|\boldsymbol{\beta}^{(b)*} - \boldsymbol{\beta}^*\right\|_2\right)^2\right\}, \qquad 2 \le b \le B, \end{aligned} \tag{52}$$

which proves equation 13 in Theorem 3.

Let $\mathcal{G} := \mathcal{E}_X^{(B)} \cap \mathcal{E}_0 \cap \mathcal{E}_{\mathrm{loc}} \cap \mathcal{E}_M$. Combining equation 35, equation 39, equation 43, and equation 49, we have

$$\begin{aligned} \mathbb{P}(\mathcal{G}^c) &\le \mathbb{P}((\mathcal{E}_X^{(B)})^c) + \mathbb{P}(\mathcal{E}_0^c \mid \mathcal{E}_X^{(B)}) + \mathbb{P}(\mathcal{E}_{\mathrm{loc}}^c \mid \mathcal{E}_X^{(B)}) + \mathbb{P}(\mathcal{E}_M^c \mid \mathcal{E}_X^{(B)}) \\ &\le \delta/4 + \delta/8 + \delta/4 + \delta/4 \ < \ \delta. \end{aligned} \tag{53}$$

Thus all results of Theorem 3 hold simultaneously for $1 \le b \le B$ with probability at least $1 - \delta$.

D.4 Proof of Corollary 4

Define

$$t_b := r_b + 2\sqrt{p}\,\nu L = \nu\left(\sqrt{3p + 3\log(16b^2/\delta)} + 2\sqrt{2p\log(16p/\delta)}\right),$$

we reject $H_0^{(b)}$ if and only if $\left\|\widetilde{\boldsymbol{\beta}}^{(b)} - \hat{\mathbf{m}}^{(b-1)}\right\|_2 > t_b$. This threshold does not require the knowledge of ϵ. We next work on the same event $\mathcal{G} = \mathcal{E}_X^{(B)} \cap \mathcal{E}_0 \cap \mathcal{E}_{\mathrm{loc}} \cap \mathcal{E}_M$ from equation 53.

Homogeneous batches. The projection operator $\Pi_{\mathcal{I}_j}$ implies $\left\|\hat{\mathbf{m}}^{(b-1)} - \boldsymbol{\beta}^*\right\|_2 \le 2\sqrt{p}\,\nu L$; equation 43 gives $\left\|\mathbf{e}^{(b)}\right\|_2 \le r_b$. Therefore, for every homogeneous batch b, we simultaneously have

$$\left\|\widetilde{\boldsymbol{\beta}}^{(b)} - \hat{\mathbf{m}}^{(b-1)}\right\|_2 \le \left\|\mathbf{e}^{(b)}\right\|_2 + \left\|\hat{\mathbf{m}}^{(b-1)} - \boldsymbol{\beta}^*\right\|_2 \le t_b,$$

and it is accepted.

Heterogeneous batches. For any heterogeneous batch with parameter $\boldsymbol{\theta}^{(b)*}$, the triangle inequality gives

$$\left\|\widetilde{\boldsymbol{\beta}}^{(b)} - \hat{\mathbf{m}}^{(b-1)}\right\|_2 \ge \left\|\boldsymbol{\theta}^{(b)*} - \boldsymbol{\beta}^*\right\|_2 - \left\|\mathbf{e}^{(b)}\right\|_2 - \left\|\hat{\mathbf{m}}^{(b-1)} - \boldsymbol{\beta}^*\right\|_2 \ge \left\|\boldsymbol{\theta}^{(b)*} - \boldsymbol{\beta}^*\right\|_2 - t_b.$$

Therefore, once $\left\|\boldsymbol{\theta}^{(b)*} - \boldsymbol{\beta}^*\right\|_2 > 2t_b$, we have $\left\|\widetilde{\boldsymbol{\beta}}^{(b)} - \hat{\mathbf{m}}^{(b-1)}\right\|_2 > t_b$, proving this batch will be detected.

Combining these two cases, we complete the proof of Corollary 4.