Machine Learning (2025) 114:174
https://doi.org/10.1007/s10994-025-06813-1

Panda: partially approximate newton methods for distributed minimax optimization with unbalanced dimensions

Minheng Xiao[1] · Chengchang Liu[2] · Cheng Chen[3] · John C. S. Lui[2] · Sen Na[4]

Received: 31 January 2025 / Revised: 24 February 2025 / Accepted: 30 May 2025 /
Published online: 19 June 2025

Abstract
Unbalanced dimensions are crucial characteristics in various minimax optimization problems, such as few-shot learning (Cortes and Mohri in Adv Neural Inf Process Syst 16, 2003; Ying et al. in Adv Neural Inf Process Syst 29, 2016) and fairness-aware machine learning (Lowd and Meek, in: Proceedings of the eleventh ACM SIGKDD international conference on knowledge discovery in data mining, 2005; Zhang et al., in: Proceedings of the 2018 AAAI/ACM conference on AI, ethics, and society, 2018). In this paper, we propose a communication-efficient second-order method named PANDA (Partially Approximate Newton methods for Distributed minimAx) to solve problems with unbalanced dimensions. PANDA requires almost the same per-iteration communication cost as the first-order methods by utilizing the special problem structure in its design for data exchange between the client and server. More importantly, it exhibits a superior linear-quadratic convergence rate and significantly reduces the total number of communication rounds through the efficient use of second-order information. We also develop GIANT-PANDA based on the framework of PANDA, which further reduces the computation cost of the latter one by performing sketching operations on each client. Through comprehensive theoretical analysis and empirical evaluations, we demonstrate the superior performance of the proposed methods compared to existing state-of-the-art methods.

Keywords Distributed optimization · Minimax optimization · Fairness-aware machine learning · Few-shot learning

Editor: Lam M. Nguyen.

Minheng Xiao and Chengchang Liu have been contributed equally to this work.

✉ Chengchang Liu
ccliu22@cse.cuhk.edu.hk

[1] Ohio State University, Columbus, OH, USA

[2] The Chinese University of Hong Kong, Sha Tin, Hong Kong SAR, China

[3] East China Normal University, Shanghai, China

[4] Georgia Institute of Technology, Atlanta, GA, USA

1 Introduction

We consider a class of minimax optimization problems formulated as finite-sum expressions:

$$\min_{\mathbf{x}\in\mathbb{R}^{n_\mathbf{x}}} \max_{\mathbf{y}\in\mathbb{R}^{n_\mathbf{y}}} f(\mathbf{x},\mathbf{y}) \overset{\text{def}}{=} \frac{1}{N}\sum_{j=1}^{N} l_j(\mathbf{x},\mathbf{y}), \tag{1}$$

where N denotes the sample size, and $n_\mathbf{x}$ and $n_\mathbf{y}$ denote the dimensions of the variables $\mathbf{x}$ and $\mathbf{y}$ respectively. The smooth function $f(\mathbf{x},\mathbf{y})$ is supposed to be strongly convex in $\mathbf{x}$ and strongly concave in $\mathbf{y}$ (i.e., satisfying the SC-SC condition). For many machine learning problems with a large sample size N, distributed methods are often preferable for solving problems in a parallel fashion. To facilitate our study in a distributed setting, let us divide the N samples that the i-th client holds $|S_i|$ samples. Consequently, we have $N = \sum_{i=1}^{m} |S_i|$, leading us to the following alternative problem:

$$\min_{\mathbf{x}\in\mathbb{R}^{n_\mathbf{x}}} \max_{\mathbf{y}\in\mathbb{R}^{n_\mathbf{y}}} f(\mathbf{x},\mathbf{y}) = \frac{1}{m}\sum_{i=1}^{m} f^i(\mathbf{x},\mathbf{y}), \quad \text{where } f^i(\mathbf{x},\mathbf{y}) \overset{\text{def}}{=} \frac{1}{|S_i|}\sum_{j\in S_i} l_j(\mathbf{x},\mathbf{y}). \tag{2}$$

In this context, for each client $i = 1,\dots,m$, f^i represents its local function and S_i is the index set of local samples.

Minimax optimization has gained significant attention in the data mining and machine learning community due to its broad applications in various domains, including game theory (Basar & Olsder, 1999; Facchinei, 2003), supervised learning (Lanckriet et al., 2002), robust optimization (Ben-Tal & Nemirovski, 2002; Deng & Mahdavi, 2021; Gao & Kleywegt, 2022), and fairness-aware machine learning (Creswell et al., 2018; Liu et al., 2020). Among these applications, many of them share a critical property that the dimensions of variables $\mathbf{x}$ and $\mathbf{y}$ are *unbalanced* (Liu et al., 2022). For instance, AUC maximization (Cortes & Mohri, 2003; Ying et al., 2016) aims to train a binary classifier on imbalanced datasets $\{\mathbf{a}_j, b_j\}_{j=1}^N$, where $\mathbf{a}_j$ denotes the input with d features and $b_j \in \{-1,+1\}$ denotes the label. This problem can be formulated as a minimax problem with $n_\mathbf{x} = d+2$ and $n_\mathbf{y} = 1$. Additionally, in fairness-aware machine learning tasks (Lowd & Meek, 2005; Zhang et al., 2018), we are given a training set $\{\mathbf{a}_j, b_j, \mathbf{c}_j\}_{j=1}^N$, where $\mathbf{a}_j$ represents d-dimensional features to learn from, $\mathbf{c}_j$ contains s protected features, and b_j denotes the label. In this case, we have $n_\mathbf{x} = d \gg n_\mathbf{y} = s$. Throughout this paper, we use the term "unbalanced dimensions"to describe the above special problem structure, which can be expressed as $n_\mathbf{x} \gg n_\mathbf{y}$ or $n_\mathbf{y} \gg n_\mathbf{x}$.[1]

There are numerous first-order methods for solving minimax optimization problems in (1), including gradient descent ascent, extra gradient, and many of their variants (Chavdarova et al., 2019; Hsieh et al., 2019; Korpelevich, 1976; Lin et al., 2020; Malitsky, 2015; Mishchenko et al., 2020; Nedić & Ozdaglar, 2009; Nouiehed et al., 2019; Tseng, 2000). These first-order methods can be straightforwardly generalized to solve distributed problems in (2), by simply aggregating the gradients to the server at each iteration. Distributed first-order methods that perform multiple local iterations for each client before communication have also been proposed to solve minimax problems (Deng & Mahdavi, 2021; Sun & Wei, 2022; Zhang et al., 2024). Among these methods, Zhang et al. (2024) achieved the best-known

[1] For simplicity, we only consider the case $n_\mathbf{x} \gg n_\mathbf{y}$ in this paper, while $n_\mathbf{y} \gg n_\mathbf{x}$ can be studied trivially in the same way.

results in terms of the total communication rounds, with an order of $\mathcal{O}(\kappa_{\mathbf{g}} \ln(1/\epsilon))$ where $\kappa_{\mathbf{g}}$ is the condition number of the objective (cf. Sect. 2.1) and ϵ is the desired accuracy. It is worth noting that the per-iteration communication complexity between the client and server for first-order methods is only $\mathcal{O}(n_{\mathbf{x}} + n_{\mathbf{y}})$. However, these methods often require a substantial number of communication rounds to attain an accurate solution (e.g., their communication rounds depend heavily on the condition number). As a result, factors such as unpredictable network latency can lead to expensive total communication costs.

Second-order methods are well-known for their fast convergence rates, brought about by the utilization of the Hessian information of the objective. The Cubic regularized Newton method (Huang et al., 2022) and its restart variant (Huang & Zhang, 2022) have been proposed for solving (1) and demonstrated local superlinear convergence rates under the SC-SC condition. However, applying these methods directly in a distributed setting would require the communication of the full local Hessian matrix, resulting in a per-iteration communication complexity of $\mathcal{O}((n_{\mathbf{x}} + n_{\mathbf{y}})^2)$. This communication overhead is unacceptable due to bandwidth limitations. On the other hand, several communication-efficient[2] distributed second-order methods (Islamov et al., 2022; Liu et al., 2024; Shamir et al., 2014; Wang & Li, 2020; Ye et al., 2022) have been proposed for *convex* optimization problems, eliminating the need for full Hessian communication. However, the design and analysis of these communication-efficient second-order methods heavily depend on the convexity of the objective function, making it challenging to generalize them to the minimax setting. Building upon this, it is natural to ask: *Is it possible to develop communication-efficient distributed second-order methods for minimax optimization by leveraging the structure of "unbalanced dimensions"?*

In this paper, we provide an affirmative answer to this question by introducing PANDA (Partially Approximate Newton methods for Distributed minimAx optimization). In each iteration, PANDA avoids the need for communicating the full Hessian matrix, instead requiring only the exchange of the partial Hessian matrix associated with $\mathbf{y}$ (recall that we suppose $n_{\mathbf{x}} \gg n_{\mathbf{y}}$ in this paper), i.e. $\nabla^2_{\mathbf{yy}} f^i(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{n_{\mathbf{y}} \times n_{\mathbf{y}}}$, $\nabla^2_{\mathbf{xy}} f^i(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{n_{\mathbf{x}} \times n_{\mathbf{y}}}$, and $\nabla^2_{\mathbf{xx}} f^i(\mathbf{x}, \mathbf{y}) \nabla^2_{\mathbf{xy}} f(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{n_{\mathbf{x}} \times n_{\mathbf{y}}}$. Additionally, it exchanges vectors such as gradients and local descent directions. As a result, the per-iteration communication complexity of PANDA can be summarized as:

$$\mathcal{O}\Bigg(\underbrace{n_{\mathbf{x}} n_{\mathbf{y}} + n_{\mathbf{y}}^2}_{\nabla^2_{\mathbf{xy}} f^i,\ [\nabla^2_{\mathbf{xx}} f^i]^{-1} \nabla^2_{\mathbf{xy}} f,\ \nabla^2_{\mathbf{yy}} f^i} + \underbrace{n_{\mathbf{x}} + n_{\mathbf{y}}}_{\text{vectors}} \Bigg) = \mathcal{O}(n_{\mathbf{x}} n_{\mathbf{y}}) \approx \mathcal{O}(n_{\mathbf{x}}).$$

This complexity significantly reduces that of typical second-order methods, bringing it to the same order as first-order methods. Furthermore, the utilization of second-order information in PANDA results in improved convergence behavior compared to existing distributed first-order methods.

1.1 Contribution

The contribution of this paper is threefold.

[2] We use the term "communication-efficient" to describe second-order methods whose per-iteration communication cost is of the same order as that of first-order methods.

(a) We develop a Partially Approximate Newton (PAN) method to solve the general minimax problem (1) with unbalanced dimensions. If the approximate Hessian $\tilde{\mathbf{H}}_{\mathbf{xx}}$ satisfies$(1-\eta)\nabla^2_{\mathbf{xx}} f \preceq \tilde{\mathbf{H}}_{\mathbf{xx}} \preceq (< spanclass =' crossLinkCiteEqu' > 1 < /span > +\eta)\nabla^2_{\mathbf{xx}} f$ with $\eta \in (0,1)$, then PAN exhibits a linear-quadratic convergence rate for some measure λ_t:

$$\lambda_{t+1} \leq \frac{\eta}{1-\eta}\lambda_t + \beta\lambda_t^2.$$

This result of PAN generalizes the approximate Newton method for convex optimization in Ye et al. (2021) and relaxes the conditions in Liu et al. (2022, Lemma 4.3) for minimax optimization.

(b) We develop the PANDA method to solve the distributed minimax problem (2). If the local partial Hessian satisfies $(1-\eta)\nabla^2_{\mathbf{xx}} f \preceq \nabla^2_{\mathbf{xx}} f^i \preceq (< spanclass = 'crossLinkCiteEqu' > 1 < /span > +\eta)\nabla^2_{\mathbf{xx}} f$ with $\eta \in (0,1)$, then PANDA exhibits a linear-quadratic convergence rate:

$$\lambda_{t+1} \leq \frac{\eta^2}{1-\eta}\lambda_t + \beta_1\lambda_t^2.$$

Furthermore, we can guarantee the existence of η provided that $N \geq \mathcal{O}(mK/\mu)$, where m is the number of clients, $K = \max_j \|\nabla^2_{\mathbf{xx}} l_j\|$, and μ is the strong convexity parameter (cf. Sect. 2.1).

(c) We develop the GIANT-PANDA method, which employs matrix sketching techniques on each local client to construct the partial approximate Hessian $\tilde{\mathbf{H}}^i_{\mathbf{xx}}$. This method exhibits a linear-quadratic convergence rate:

$$\lambda_{t+1} \leq \left(\frac{\eta}{\sqrt{m}} + \frac{\eta^2}{1-\eta}\right)\lambda_t + \beta_2\lambda_t^2.$$

This result leads to a sharper analysis compared to the original GIANT method in (Wang et al., 2018), as it improves the convergence rate by a factor of $\sqrt{\kappa_{\mathbf{g}}}$ in the linear term.

Organization We introduce fundamental notation, assumptions, and preliminary results for Hessian approximation in Sect. 2. In Sects. 3 and 4, we introduce PAN for solving (1) and introduce PANDA (along with GIANT-PANDA) for solving (2), respectively. We conduct empirical studies in Sect. 5 and provide conclusions in Sect. 6. All proofs are deferred to the appendix.

2 Preliminaries

2.1 Notation and assumptions

We use $\mathbf{g}_{\mathbf{x}}(\mathbf{x},\mathbf{y})$ and $\mathbf{H}_{\mathbf{xx}}(\mathbf{x},\mathbf{y})$ to denote the gradient $\nabla_{\mathbf{x}} f(\mathbf{x},\mathbf{y})$ and Hessian $\nabla^2_{\mathbf{xx}} f(\mathbf{x},\mathbf{y})$ with respect to $\mathbf{x}$, respectively (similar for $\mathbf{g}_{\mathbf{y}}$, $\mathbf{H}_{\mathbf{xy}}$, $\mathbf{H}_{\mathbf{yy}}$). For the local gradient and Hessian associated with the i-th client, we use $\mathbf{g}^i_{\mathbf{x}}(\mathbf{x},\mathbf{y})$ and $\mathbf{H}^i_{\mathbf{xx}}(\mathbf{x},\mathbf{y})$ to denote $\nabla_{\mathbf{x}} f^i(\mathbf{x},\mathbf{y})$ and $\nabla^2_{\mathbf{xx}} f^i(\mathbf{x},\mathbf{y})$ (similar for $\mathbf{g}^i_{\mathbf{y}}$, $\mathbf{H}^i_{\mathbf{xy}}$, $\mathbf{H}^i_{\mathbf{yy}}$). We use $\|\cdot\|$ to denote the spectral norm for matrices

and the Euclidean norm for vectors. Additionally, we define the matrix row coherence as follows.

Definition 2.1 (Wang et al., 2018, Definition 1) Let $\mathbf{A} \in \mathbb{R}^{N\times d}$ be a matrix with full column rank and $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$ be its reduced singular value decomposition with $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{N\times d}$. The row coherence of $\mathbf{A}$ is defined as $\nu(\mathbf{A}) \overset{\text{def}}{=} \frac{N}{d} \max_j \|\mathbf{u}_j\|^2 \in [1, \frac{N}{d}]$, where $\mathbf{u}_j$ is the j-th row of $\mathbf{U}$.

We introduce the following assumption for the objective function in (1).

Assumption 2.2 We assume $f(\mathbf{x}, \mathbf{y})$ is twice differentiable, μ-strongly convex in $\mathbf{x}$, μ -strongly concave in $\mathbf{y}$, and has $L_\mathbf{g}$-Lipschitz continuous gradient and $L_\mathbf{H}$-Lipschitz continuous Hessian. We also assume each individual function $l_j(\mathbf{x}, \mathbf{y})$ is convex in $\mathbf{x}$. We denote $\kappa_\mathbf{g} \overset{\text{def}}{=} L_\mathbf{g}/\mu$ and $\kappa_\mathbf{H} \overset{\text{def}}{=} L_\mathbf{H}/\mu$.

The convexity of each individual function l_j in $\mathbf{x}$, along with the $L_\mathbf{g}$-Lipschitz continuity of the gradient $\mathbf{g}_\mathbf{x}$, implies that the Hessian $\nabla^2_{\mathbf{xx}} l_j$ is bounded. Let us denote $K \overset{\text{def}}{=} \max_j \|\nabla^2_{\mathbf{xx}} l_j\|$ and $\hat{\kappa} \overset{\text{def}}{=} K/\mu$.

2.2 Matrix approximation via sub-sampling and sketching

Let us introduce some preliminary results for approximating a positive definite Hessian matrix. We first consider a Hessian matrix in the form of $\mathbf{H} = \frac{1}{N}\sum_{j=1}^N \mathbf{H}_j \in \mathbb{R}^{d\times d}$, and approximate it using sub-sampling:

$$\tilde{\mathbf{H}} = \frac{1}{|\mathcal{S}|}\sum_{j\in\mathcal{S}} \mathbf{H}_j, \tag{3}$$

where elements in $\mathcal{S}$ are uniformly sampled from $\{1, \cdots, N\}$. The following lemma characterizes the error of the sub-sampling approximation.

Lemma 2.3 (Ye et al., 2021, Lemma 9) *Suppose* $\mathbf{H} \succeq \mu\mathbf{I}$ *and* $\max_{1\le j\le N} \|\mathbf{H}_j\| \le \hat{K}$ *for some constants* $\mu, \hat{K} > 0$. *For any* $\delta \in (0,1)$ *and* $\eta \in (0, 0.5)$, *if the sample size satisfies* $|\mathcal{S}| \ge \frac{3\hat{K}\log(2d/\delta)}{\mu\eta^2}$, *then with probability at least* $1-\delta$, *we have* $(1-\eta)\mathbf{H} \preceq \tilde{\mathbf{H}} \preceq (< \textit{spanclass} =' \textit{crossLinkCiteEqu}' > 1 < /\textit{span} > +\eta)\mathbf{H}$ *for* $\tilde{\mathbf{H}}$ *defined in* (3).

We then consider a special case where the Hessian matrix is expressed as $\mathbf{H} = \mathbf{A}^\top\mathbf{A} + \alpha\mathbf{I}$, with $\mathbf{A} \in \mathbb{R}^{N\times d}$ being a full column-rank matrix. This form of Hessian matrix naturally arises in classical regression problems (Ye et al., 2021; Wang et al., 2017). We construct two approximate Hessians using sketching techniques:

$$\tilde{\mathbf{H}}_i = \mathbf{A}^\top\mathbf{S}_i\mathbf{S}_i^\top\mathbf{A} + \alpha\mathbf{I}, \qquad \hat{\mathbf{H}} = \mathbf{A}^\top\mathbf{S}\mathbf{S}^\top\mathbf{A} + \alpha\mathbf{I}. \tag{4}$$

here $\{\mathbf{S}_i\}_{i=1}^m \in \mathbb{R}^{N\times s'}$ represent the sketching matrices and $\mathbf{S} \overset{\text{def}}{=} \frac{1}{\sqrt{m}}[\mathbf{S}_1, \cdots, \mathbf{S}_m] \in \mathbb{R}^{N\times ms'}$. The following lemma characterizes the errors of these two sketching approximations.

Lemma 2.4 *Adapted from* (*Wang et al.*, 2018, *Lemma* 8)

Let η and $\delta \in (0,1)$ be fixed parameters, $\nu = \nu(\mathbf{A})$, and $\mathbf{S}_1, \cdots, \mathbf{S}_m \in \mathbb{R}^{N\times s'}$ be independent uniform sampling matrices with $s' \geq \frac{3\nu d}{\eta^2}\log(\frac{dm}{\delta})$. Then, with probability at least $1-\delta$, we have $(1-\eta)\mathbf{H} \preceq \tilde{\mathbf{H}}_i \preceq (< spanclass =' crossLinkCiteEqu' > 1 < /span > +\eta)\mathbf{H}$ and $(1-\eta/\sqrt{m})\mathbf{H} \preceq \hat{\mathbf{H}} \preceq (< spanclass =' crossLinkCiteEqu' > 1 < /span > +\eta/\sqrt{m})\mathbf{H}$ for $\mathbf{H}_i$ and $\hat{\mathbf{H}}$ defined in (4).

3 The analysis framework of partially approximate Newton method

In this section, we propose a Partially Approximate Newton (PAN) method for Problem (1). We start with the classical Newton update:

$$\begin{bmatrix}\mathbf{x}_+\\ \mathbf{y}_+\end{bmatrix} = \begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} - \begin{bmatrix}\mathbf{H}_{\mathbf{xx}}(\mathbf{x},\mathbf{y}) & \mathbf{H}_{\mathbf{xy}}(\mathbf{x},\mathbf{y})\\ (\mathbf{H}_{\mathbf{xy}}(\mathbf{x},\mathbf{y}))^\top & \mathbf{H}_{\mathbf{yy}}(\mathbf{x},\mathbf{y})\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{g}_{\mathbf{x}}(\mathbf{x},\mathbf{y})\\ \mathbf{g}_{\mathbf{y}}(\mathbf{x},\mathbf{y})\end{bmatrix}. \tag{5}$$

Using the approximate Hessian matrix $\tilde{\mathbf{H}}_{\mathbf{xx}}$ to replace the exact Hessian matrix $\mathbf{H}_{\mathbf{xx}}(\mathbf{x},\mathbf{y})$ in (5) leads to the update rule of PAN as follows:

$$\begin{bmatrix}\mathbf{x}_+\\ \mathbf{y}_+\end{bmatrix} = \begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} - \begin{bmatrix}\tilde{\mathbf{H}}_{\mathbf{xx}} & \mathbf{H}_{\mathbf{xy}}(\mathbf{x},\mathbf{y})\\ (\mathbf{H}_{\mathbf{xy}}(\mathbf{x},\mathbf{y}))^\top & \mathbf{H}_{\mathbf{yy}}(\mathbf{x},\mathbf{y})\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{g}_{\mathbf{x}}(\mathbf{x},\mathbf{y})\\ \mathbf{g}_{\mathbf{y}}(\mathbf{x},\mathbf{y})\end{bmatrix}. \tag{6}$$

We use the weighted gradient norm as the measure in our analysis (Liu et al., 2022):

$$\lambda(\mathbf{x},\mathbf{y}) \overset{\text{def}}{=} \sqrt{(\mathbf{g}_{\mathbf{x}}(\mathbf{x},\mathbf{y}))^\top(\mathbf{P}(\mathbf{x},\mathbf{y}))^{-1}\mathbf{g}_{\mathbf{x}}(\mathbf{x},\mathbf{y})} + \frac{2}{\sqrt{\mu}}\|\mathbf{g}_{\mathbf{y}}(\mathbf{x},\mathbf{y})\|,$$

where $\mathbf{P}(\mathbf{x},\mathbf{y})$ is defined as

$$\mathbf{P}(\mathbf{x},\mathbf{y}) \overset{\text{def}}{=} \mathbf{H}_{\mathbf{xx}}(\mathbf{x},\mathbf{y}) - \mathbf{H}_{\mathbf{xy}}(\mathbf{x},\mathbf{y})(\mathbf{H}_{\mathbf{yy}}(\mathbf{x},\mathbf{y}))^{-1}\mathbf{H}_{\mathbf{yx}}(\mathbf{x},\mathbf{y}). \tag{7}$$

The following lemma shows that if $\tilde{\mathbf{H}}_{\mathbf{xx}}$ is a good approximation of $\mathbf{H}_{\mathbf{xx}}(\mathbf{x},\mathbf{y})$, then $\mathbf{P}(\mathbf{x},\mathbf{y})$ can be also well approximated by $\mathbf{C}(\mathbf{x},\mathbf{y})$, defined as

$$\mathbf{C}(\mathbf{x},\mathbf{y}) \overset{\text{def}}{=} \tilde{\mathbf{H}}_{\mathbf{xx}} - \mathbf{H}_{\mathbf{xy}}(\mathbf{x},\mathbf{y})(\mathbf{H}_{\mathbf{yy}}(\mathbf{x},\mathbf{y}))^{-1}\mathbf{H}_{\mathbf{yx}}(\mathbf{x},\mathbf{y}). \tag{8}$$

Lemma 3.1 *Under Assumption* 2.2 *and suppose that $\tilde{\mathbf{H}}_{\mathbf{xx}}$ satisfies $(1-\eta)\mathbf{H}_{\mathbf{xx}} \preceq \tilde{\mathbf{H}}_{\mathbf{xx}} \preceq (< spanclass =' crossLinkCiteEqu' > 1 < /span > +\eta)\mathbf{H}_{\mathbf{xx}}$, we have*

$$\left\|\mathbf{I} - \mathbf{P}(\mathbf{x},\mathbf{y})^{1/2}(\mathbf{C}(\mathbf{x},\mathbf{y}))^{-1}\mathbf{P}(\mathbf{x},\mathbf{y})^{1/2}\right\| \leq \frac{\eta}{1-\eta}$$

We establish a linear-quadratic convergence rate for the PAN update when $\mathbf{P}(\mathbf{x},\mathbf{y})$ and $\mathbf{C}(\mathbf{x},\mathbf{y})$ are close.

Theorem 3.2 *Under the Assumption 2.2 and suppose* $\mathbf{P}(\mathbf{x},\mathbf{y})$ *and* $\mathbf{C}(\mathbf{x},\mathbf{y})$ *are close such that* $\|\mathbf{I}-\mathbf{P}(\mathbf{x},\mathbf{y})^{1/2}\mathbf{C}(\mathbf{x},\mathbf{y})^{-1}\mathbf{P}(\mathbf{x},\mathbf{y})^{1/2}\| \le \eta_1$, *the update of* PAN *in (6) exhibits the following linear-quadratic convergence rate*:

$$\lambda(\mathbf{x}_+,\mathbf{y}_+) \le \eta_1\lambda(\mathbf{x},\mathbf{y}) + \frac{12\kappa_{\mathbf{g}}^2\kappa_{\mathbf{H}}(1+\eta_1)^2}{\sqrt{\mu}}\lambda(\mathbf{x},\mathbf{y})^2.$$

When employing the sub-sampling approximation to construct $\tilde{\mathbf{H}}_{\mathbf{xx}}$, we derive the following corollary by combining the results from Lemma 3.1 and Theorem 3.2.

Corollary 3.3 *Let us construct the partial Hessian approximation by sub-sampling* $\tilde{\mathbf{H}}_{\mathbf{xx}} = \frac{1}{|S|}\sum_{j\in S}\nabla^2_{\mathbf{xx}} l_j(\mathbf{x},\mathbf{y})$. *Under Assumption 2.2 and for any* $\delta \in (0,1)$, *if the sample size satisfies* $|S| \ge 12\hat{\kappa}\log(2n_{\mathbf{x}}/\delta)$, *then with probability at least* $1-\delta$, *the update of* PAN *in (6) satisfies*

$$\lambda(\mathbf{x}_+,\mathbf{y}_+) \le \frac{\eta}{1-\eta}\lambda(\mathbf{x},\mathbf{y}) + \frac{12\kappa_{\mathbf{g}}^2\kappa_{\mathbf{H}}(1+\eta/(1-\eta))^2}{\sqrt{\mu}}\lambda(\mathbf{x},\mathbf{y})^2 \tag{9}$$

with $\eta = \eta_{\text{PAN}} \overset{\text{def}}{=} \sqrt{\frac{3\hat{\kappa}\log(2n_{\mathbf{x}}/\delta)}{|S|}}$.

Corollary 3.3 suggests that PAN requires $\mathcal{O}(\log(1/\epsilon)/\log(|S|/\hat{\kappa}))$ iterations to achieve ϵ-accuracy in terms of the measure $\lambda(\mathbf{x},\mathbf{y})$ for a quadratic objective function. Analyzing the complexity of linear-quadratic rates on quadratic functions is a common practice in the literature (Roosta-Khorasani & Mahoney, 2019; Wang et al., 2018; Ye et al., 2020, 2021), which allows us to simply ignore the quadratic term in (9) since $\kappa_{\mathbf{H}} = 0$. In comparison, state-of-the-art first-order methods such as optimistic gradient and extra gradient methods have complexities $\mathcal{O}\big(\kappa_{\mathbf{g}}\log(1/\epsilon)\big)$; methods that do not access the full Hessian at each iteration such as the quasi-Newton method (Liu & Luo, 2022) and the partial-quasi-Newton method (Liu et al., 2022) have complexities $\mathcal{O}(\kappa_{\mathbf{g}}^2 + \sqrt{n_{\mathbf{x}}\log(1/\epsilon)})$ and $\mathcal{O}(\kappa_{\mathbf{g}} + \sqrt{n_{\mathbf{x}}\log(1/\epsilon)})$, respectively. We can see that PAN exhibits a much weaker dependency on the condition number $\kappa_{\mathbf{g}}$. We present the comparisons in Table 1.

4 Partially approximate Newton methods for distributed minimax optimization

In this section, we present the PANDA method for solving Problem (2) in Sect. 4.1, establish its convergence results in Sect. 4.2, and extend PANDA to GIANT-PANDA for a special function class that commonly appears in regression problems in Sect. 4.3.

Table 1 We present the iteration complexity of proposed method (PAN) and baselines for solving quadratic minimax optimization (AUC maximization)

Methods	Iteration complexity	References
Extra gradient	$\mathcal{O}\big(\kappa_{\mathbf{g}} \log(1/\epsilon)\big)$	Korpelevich (1976)
Quasi-Newton	$\mathcal{O}(\kappa_{\mathbf{g}}^2 + \sqrt{n_{\mathbf{x}}} \log(1/\epsilon))$	Liu and Luo (2022)
Partial-quasi-Newton	$\mathcal{O}(\kappa_{\mathbf{g}} + \sqrt{n_{\mathbf{x}}} \log(1/\epsilon))$	Liu et al. (2022)
PAN	$\mathcal{O}(\log(1/\epsilon)/\log(\lvert S\rvert/\hat{\kappa}))$	Corollary 3.3

Algorithm 1 PANDA$(\mathbf{x}_0, \mathbf{y}_0, T)$

for $t = 0, \ldots, T$ **do**
  for $i = 1, \ldots, m$ **do in parallel**
    i-th client:
      compute
        $\mathbf{g}_{\mathbf{x}}^i = \mathbf{g}_{\mathbf{x}}^i(\mathbf{x}_t, \mathbf{y}_t)$, $\mathbf{g}_{\mathbf{y}}^i = \mathbf{g}_{\mathbf{y}}^i(\mathbf{x}_t, \mathbf{y}_t)$
        $\mathbf{H}_{\mathbf{xy}}^i = \mathbf{H}_{\mathbf{xy}}^i(\mathbf{x}_t, \mathbf{y}_t)$, $\mathbf{H}_{\mathbf{yy}}^i = \mathbf{H}_{\mathbf{xy}}^i(\mathbf{x}_t, \mathbf{y}_t)$
      send $\mathbf{g}_{\mathbf{x}}^i$, $\mathbf{g}_{\mathbf{y}}^i$, $\mathbf{H}_{\mathbf{xy}}^i$ and $\mathbf{H}_{\mathbf{yy}}^i$
  end for
  server:
    aggregate $\mathbf{g}_{\mathbf{x}}, \mathbf{g}_{\mathbf{y}}$, $\mathbf{H}_{\mathbf{xy}}$ and $\mathbf{H}_{\mathbf{yy}}$ from (11)
    broadcast $\mathbf{g}_{\mathbf{x}}$ and $\mathbf{H}_{\mathbf{xy}}$
  for $i = 1, \ldots, m$ **do in parallel**
    i-th client:
      compute
        $\mathbf{Q}_{\mathbf{xy}}^i = [\mathbf{H}_{\mathbf{xx}}^i(\mathbf{x}_t, \mathbf{y}_t)]^{-1}\mathbf{H}_{\mathbf{xy}}$
        $\mathbf{q}_{\mathbf{x}}^i = [\mathbf{H}_{\mathbf{xx}}^i(\mathbf{x}_t, \mathbf{y}_t)]^{-1}\mathbf{g}_{\mathbf{x}}$
      send $\mathbf{Q}_{\mathbf{xy}}^i$ and $\mathbf{q}_{\mathbf{x}}^i$
  end for
  server:
    aggregate $\mathbf{Q}_{\mathbf{xy}}, \mathbf{q}_{\mathbf{x}}$ according to (12)
    compute update directions $\tilde{\mathbf{d}}_{\mathbf{x}}$, $\tilde{\mathbf{d}}_{\mathbf{y}}$ from (13)
    update the models
      $\mathbf{x}_{t+1} = \mathbf{x}_t - \tilde{\mathbf{d}}_{\mathbf{x}}$
      $\mathbf{y}_{t+1} = \mathbf{y}_t - \tilde{\mathbf{d}}_{\mathbf{y}}$.
    broadcast $\mathbf{x}_{t+1}$ and $\mathbf{y}_{t+1}$
end for

4.1 The PANDA algorithm

For simplicity, we suppress the evaluation point and use $\mathbf{g}_\mathbf{x}$ to denote $\mathbf{g}_\mathbf{x}(\mathbf{x}, \mathbf{y})$ (similar to $\mathbf{g}_\mathbf{y}$, $\mathbf{H}_{\mathbf{xx}}$, $\mathbf{H}_{\mathbf{xy}}$, $\mathbf{H}_{\mathbf{yy}}$). We start with the standard Newton direction $\begin{bmatrix}\mathbf{d}_\mathbf{x}\\ \mathbf{d}_\mathbf{y}\end{bmatrix} \overset{\text{def}}{=} \begin{bmatrix}\mathbf{H}_{\mathbf{xx}} & \mathbf{H}_{\mathbf{xy}}\\ \mathbf{H}_{\mathbf{xy}}^\top & \mathbf{H}_{\mathbf{yy}}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{g}_\mathbf{x}\\ \mathbf{g}_\mathbf{y}\end{bmatrix}$, which can be expressed explicitly by using block matrix inversion formula:

$$\begin{aligned}\mathbf{d}_\mathbf{x} &= \mathbf{H}_{\mathbf{xx}}^{-1}\mathbf{g}_\mathbf{x} - \left(\mathbf{H}_{\mathbf{xx}}^{-1}\mathbf{H}_{\mathbf{xy}}\right)\Delta_{\mathbf{yy}}\mathbf{g}_\mathbf{y} + \left(\mathbf{H}_{\mathbf{xx}}^{-1}\mathbf{H}_{\mathbf{xy}}\right)\Delta_{\mathbf{yy}}\left(\mathbf{H}_{\mathbf{xx}}^{-1}\mathbf{H}_{\mathbf{xy}}\right)^\top\mathbf{g}_\mathbf{x}\\ \mathbf{d}_\mathbf{y} &= -\Delta_{\mathbf{yy}}\mathbf{H}_{\mathbf{xy}}^\top\mathbf{H}_{\mathbf{xx}}^{-1}\mathbf{g}_\mathbf{x} + \Delta_{\mathbf{yy}}\mathbf{g}_\mathbf{y},\end{aligned} \tag{10}$$

where $\Delta_{\mathbf{yy}} = (\mathbf{H}_{\mathbf{yy}} - \mathbf{H}_{\mathbf{xy}}^\top\mathbf{H}_{\mathbf{xx}}^{-1}\mathbf{H}_{\mathbf{xy}})^{-1}$.

Under the setup of unbalanced dimensions $n_\mathbf{x} \gg n_\mathbf{y}$, obtaining the exact Hessian $\mathbf{H}_{\mathbf{xx}}$ on the server is prohibitive due to the communication overhead associated with $\mathbf{H}_{\mathbf{xx}}^i$. However, communication costs of gradients and partial Hessians $\mathbf{H}_{\mathbf{xy}}^i$ and $\mathbf{H}_{\mathbf{yy}}^i$ are relatively low. Thus, in the first round of PANDA, the server aggregates these quantities to acquire precise gradient and partial Hessian information as follows:

$$\mathbf{g}_\mathbf{x} = \frac{1}{m}\sum_{i=1}^m \mathbf{g}_\mathbf{x}^i, \quad \mathbf{g}_\mathbf{y} = \frac{1}{m}\sum_{i=1}^m \mathbf{g}_\mathbf{y}^i, \quad \mathbf{H}_{\mathbf{xy}} = \frac{1}{m}\sum_{i=1}^m \mathbf{H}_{\mathbf{xy}}^i, \quad \mathbf{H}_{\mathbf{yy}} = \frac{1}{m}\sum_{i=1}^m \mathbf{H}_{\mathbf{yy}}^i. \tag{11}$$

The server then broadcasts the above-aggregated quantities to the clients, allowing each client to access global information of $\mathbf{g}_\mathbf{x}$, $\mathbf{g}_\mathbf{y}$, $\mathbf{H}_{\mathbf{xy}}$, and $\mathbf{H}_{\mathbf{yy}}$. Further, since the communication costs of $\mathbf{Q}_{\mathbf{xy}}^i \overset{\text{def}}{=} [\mathbf{H}_{\mathbf{xx}}^i]^{-1}\mathbf{H}_{\mathbf{xy}}$ and $\mathbf{q}_\mathbf{x}^i \overset{\text{def}}{=} [\mathbf{H}_{\mathbf{xx}}^i]^{-1}\mathbf{g}_\mathbf{x}$ are only $\mathcal{O}(n_\mathbf{x}n_\mathbf{y})$ and $\mathcal{O}(n_\mathbf{x})$, in the second round of PANDA, the server aggregates $\mathbf{Q}_{\mathbf{xy}}^i$ and $\mathbf{q}_\mathbf{x}^i$ as follows:

$$\mathbf{Q}_{\mathbf{xy}} = \frac{1}{m}\sum_{i=1}^m \mathbf{Q}_{\mathbf{xy}}^i, \qquad \mathbf{q}_\mathbf{x} = \frac{1}{m}\sum_{i=1}^m \mathbf{q}_\mathbf{x}^i. \tag{12}$$

Using $\mathbf{Q}_{\mathbf{xy}}$ and $\mathbf{q}_\mathbf{x}$ to replace $\mathbf{H}_{\mathbf{xx}}^{-1}\mathbf{H}_{\mathbf{xy}}$ and $\mathbf{H}_{\mathbf{xx}}^{-1}\mathbf{g}_\mathbf{x}$ in (10), the server finally computes the following approximate Newton direction

$$\begin{bmatrix}\tilde{\mathbf{d}}_\mathbf{x}\\ \tilde{\mathbf{d}}_\mathbf{y}\end{bmatrix} \overset{\text{def}}{=} \begin{bmatrix}\mathbf{q}_\mathbf{x} - \mathbf{Q}_{\mathbf{xy}}\tilde{\Delta}_{\mathbf{yy}}\mathbf{g}_\mathbf{y} + \mathbf{Q}_{\mathbf{xy}}\tilde{\Delta}_{\mathbf{yy}}\mathbf{Q}_{\mathbf{xy}}^\top\mathbf{g}_\mathbf{x}\\ -\tilde{\Delta}_{\mathbf{yy}}\mathbf{H}_{\mathbf{xy}}^\top\mathbf{q}_\mathbf{x} + \tilde{\Delta}_{\mathbf{yy}}\mathbf{g}_\mathbf{y}\end{bmatrix}, \tag{13}$$

with $\tilde{\Delta}_{\mathbf{yy}} \overset{\text{def}}{=} [\mathbf{H}_{\mathbf{yy}} - \mathbf{H}_{\mathbf{xy}}^\top\mathbf{Q}_{\mathbf{xy}}]^{-1}$ and updates the parameters based on $\tilde{\mathbf{d}}_\mathbf{x}$ and $\tilde{\mathbf{d}}_\mathbf{y}$.

We formally summarize the PANDA method in Algorithm 1. The following proposition indicates that the update rule of PANDA can be viewed as a partially approximate Newton method.

Proposition 4.1 *Using* PANDA *in Algorithm* 1, *the update rule on the server is equivalent to*

$$\begin{bmatrix}\mathbf{x}_{t+1}\\ \mathbf{y}_{t+1}\end{bmatrix} = \begin{bmatrix}\mathbf{x}_t\\ \mathbf{y}_t\end{bmatrix} - \begin{bmatrix}\tilde{\mathbf{H}}_{\mathbf{xx},t} & \mathbf{H}_{\mathbf{xy}}(\mathbf{x}_t, \mathbf{y}_t)\\ \mathbf{H}_{\mathbf{xy}}(\mathbf{x}_t, \mathbf{y}_t)^\top & \mathbf{H}_{\mathbf{yy}}(\mathbf{x}_t, \mathbf{y}_t)\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{g}_\mathbf{x}(\mathbf{x}_t, \mathbf{y}_t)\\ \mathbf{g}_\mathbf{y}(\mathbf{x}_t, \mathbf{y}_t)\end{bmatrix}, \tag{14}$$

where $\tilde{\mathbf{H}}_{\mathbf{xx},t} \overset{\text{def}}{=} \left[\frac{1}{m}\sum_{i=1}^m (\mathbf{H}_{\mathbf{xx}}^i(\mathbf{x}_t, \mathbf{y}_t))^{-1}\right]^{-1}$.

Proof We ignore the subscript t in the following proof such that $\tilde{\mathbf{H}}_{\mathbf{xx}} = \tilde{\mathbf{H}}_{\mathbf{xx},t}$, $\mathbf{H}_{\mathbf{xy}} = \mathbf{H}_{\mathbf{xy}}(\mathbf{x}_t, \mathbf{y}_t)$ (similar for $\mathbf{H}_{\mathbf{xx}}$, $\mathbf{H}^i_{\mathbf{xx}}$, $\mathbf{H}_{\mathbf{yy}}$, $\mathbf{g}_{\mathbf{x}}$, $\mathbf{g}_{\mathbf{y}}$). We denote

$$\hat{\Delta}_{\mathbf{yy}} \overset{\text{def}}{=} [\mathbf{H}_{\mathbf{yy}} - \mathbf{H}_{\mathbf{xy}}^\top \tilde{\mathbf{H}}_{\mathbf{xx}}^{-1} \mathbf{H}_{\mathbf{xy}}]^{-1} = \left[\mathbf{H}_{\mathbf{yy}} - \mathbf{H}_{\mathbf{xy}}^\top \frac{1}{m} \sum_{i=1}^{m} (\mathbf{H}^i_{\mathbf{xx}})^{-1} \mathbf{H}_{\mathbf{xy}}\right]^{-1}$$
$$= \left[\mathbf{H}_{\mathbf{yy}} - \mathbf{H}_{\mathbf{xy}}^\top \frac{1}{m} \sum_{i=1}^{m} \mathbf{Q}^i_{\mathbf{xy}}\right]^{-1} = \tilde{\Delta}_{\mathbf{yy}}.$$

Then, it holds that

$$\begin{bmatrix} \tilde{\mathbf{H}}_{\mathbf{xx}} & \mathbf{H}_{\mathbf{xy}} \\ \mathbf{H}_{\mathbf{xy}}^\top & \mathbf{H}_{\mathbf{yy}} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{g}_{\mathbf{x}} \\ \mathbf{g}_{\mathbf{y}} \end{bmatrix}$$
$$= \begin{bmatrix} \tilde{\mathbf{H}}_{\mathbf{xx}}^{-1} \mathbf{g}_{\mathbf{x}} + \left(\tilde{\mathbf{H}}_{\mathbf{xx}}^{-1} \mathbf{H}_{\mathbf{xy}}\right) \hat{\Delta}_{\mathbf{yy}} \left(\tilde{\mathbf{H}}_{\mathbf{xx}}^{-1} \mathbf{H}_{\mathbf{xy}}\right)^\top \mathbf{g}_{\mathbf{x}} - \left(\tilde{\mathbf{H}}_{\mathbf{xx}}^{-1} \mathbf{H}_{\mathbf{xy}}\right) \hat{\Delta}_{\mathbf{yy}} \mathbf{g}_{\mathbf{y}} \\ -\hat{\Delta}_{\mathbf{yy}} \mathbf{H}_{\mathbf{xy}}^\top \tilde{\mathbf{H}}_{\mathbf{xx}}^{-1} \mathbf{g}_{\mathbf{x}} + \hat{\Delta}_{\mathbf{yy}} \mathbf{g}_{\mathbf{y}} \end{bmatrix}$$
$$\overset{(12)}{=} \begin{bmatrix} \frac{1}{m} \sum_{i=1}^{m} \left(\mathbf{q}^i_x + \mathbf{Q}^i_{\mathbf{xy}} \tilde{\Delta}_{\mathbf{yy}} \left(\frac{1}{m} \sum_{i=1}^{m} \mathbf{Q}^i_{\mathbf{xy}}\right)^\top \mathbf{g}_{\mathbf{x}} - \mathbf{Q}^i_{\mathbf{xy}} \tilde{\Delta}_{\mathbf{yy}} \mathbf{g}_{\mathbf{y}} \right) \\ -\frac{1}{m} \sum_{i=1}^{m} \tilde{\Delta}_{\mathbf{yy}} \mathbf{H}_{\mathbf{xy}}^\top \mathbf{q}^i_{\mathbf{x}} + \tilde{\Delta}_{\mathbf{yy}} \mathbf{g}_{\mathbf{y}} \end{bmatrix} \overset{(13)}{=} \begin{bmatrix} \tilde{\mathbf{d}}_{\mathbf{x}} \\ \tilde{\mathbf{d}}_{\mathbf{y}} \end{bmatrix}.$$

□

4.2 Convergence analysis of PANDA

We suppose the N samples are i.i.d drawn from some distribution and each sample is associated with a local loss function $l_j(\cdot)$. We also assume each client holds s samples drawn from $\{l_j(\cdot)\}_{j=1}^N$, such that $N = ms$ and $|S_i| \equiv s$. According to Lemma 2.3, each local partial Hessian, $\mathbf{H}^i_{\mathbf{xx}}(\mathbf{x}_t, \mathbf{y}_t) = \frac{1}{s} \sum_{j \in S_i} \nabla^2 l_j(\mathbf{x}_t, \mathbf{y}_t)$, can be viewed as an sub-sampling approximation of $\mathbf{H}_{\mathbf{xx}}(\mathbf{x}_t, \mathbf{y}_t)$ when s is large. The following Lemma indicates that

$$\mathbf{C}_t \overset{\text{def}}{=} \tilde{\mathbf{H}}_{\mathbf{xx},t} - \mathbf{H}_{\mathbf{xy}}(\mathbf{x}_t, \mathbf{y}_t)(\mathbf{H}_{\mathbf{yy}}(\mathbf{x}_t, \mathbf{y}_t))^{-1}(\mathbf{H}_{\mathbf{xy}}(\mathbf{x}_t, \mathbf{y}_t))^\top$$

with $\tilde{\mathbf{H}}_{\mathbf{xx},t}$ defined in (14) is a good estimation of $\mathbf{P}(\mathbf{x}_t, \mathbf{y}_t)$.

Lemma 4.2 *Under Assumption 2.2 and suppose that for all $i \in [m]$, $\mathbf{H}^i_{\mathbf{xx}}(\mathbf{x}, \mathbf{y})$ satisfies that*

$$(1-\eta)\mathbf{H}_{\mathbf{xx}}(\mathbf{x}_t, \mathbf{y}_t) \preceq \mathbf{H}^i_{\mathbf{xx}}(\mathbf{x}_t, \mathbf{y}_t) \preceq (< spanclass =' crossLinkCiteEqu' > 1 < /span > +\eta)\mathbf{H}_{\mathbf{xx}}(\mathbf{x}_t, \mathbf{y}_t),$$

then we have $\left\| \mathbf{I} - \mathbf{P}(\mathbf{x}_t, \mathbf{y}_t)^{1/2} \mathbf{C}_t^{-1} \mathbf{P}(\mathbf{x}_t, \mathbf{y}_t)^{1/2} \right\| \leq \frac{\eta^2}{1-\eta}$.

Incorporating the linear-quadratic rates established by the PAN framework, we can obtain the improved linear-quadratic rates for PANDA.

Theorem 4.3 *Under Assumption 2.2 and suppose that*

Table 2 We present the communication complexity of proposed method (PANDA) and baselines for solving quadratic distributed minimax optimization (AUC maximization)

Methods	Communication complexity	References
Distributed extra gradient	$\mathcal{O}\big(\kappa_{\mathbf{g}}\log(1/\epsilon)\big)$	Korpelevich (1976)
FedGDA-GT	$\mathcal{O}(\text{poly}(\kappa_{\mathbf{g}})\log(1/\epsilon))$	Sun and Wei (2022)
Proxskip-VI-FL	$\mathcal{O}(\kappa_{\mathbf{g}}\log(1/\epsilon))$	Zhang et al. (2024)
PANDA	$\mathcal{O}\Big(\frac{\log(1/\epsilon)}{\log(N/(m\hat{\kappa}))}\Big)$	Corollary 4.4

$$(1-\eta)\mathbf{H}_{\mathbf{xx}}(\mathbf{x}_t,\mathbf{y}_t) \preceq \mathbf{H}^i_{\mathbf{xx}}(\mathbf{x}_t,\mathbf{y}_t) \preceq (< spanclass =' crossLinkCiteEqu' > 1 < /span > +\eta)\mathbf{H}_{\mathbf{xx}}(\mathbf{x}_t,\mathbf{y}_t)$$

holds for all $i \in [m]$, *the update rule of* PANDA (*Algorithm 1*) *in* (14) *satisfies that*

$$\lambda(\mathbf{x}_{t+1},\mathbf{y}_{t+1}) \le \frac{\eta^2}{1-\eta}\lambda(\mathbf{x}_t,\mathbf{y}_t) + \frac{12\kappa_{\mathbf{g}}^2\kappa_{\mathbf{H}}(1-\eta+\eta^2)^2}{\sqrt{\mu}(1-\eta)^2}\lambda(\mathbf{x}_t,\mathbf{y}_t)^2. \tag{15}$$

Similar to Corollary 3.3, we can guarantee a small $\eta \in (0, 0.5)$ for Theorem 4.3.

Corollary 4.4 *Under Assumption 2.2, for any* $\delta \in (0,1)$ *and* $\eta \in (0, 0.5)$, *if each client holds* $s \ge \frac{3\hat{\kappa}\log(2n_{\mathbf{x}}m/\delta)}{\eta^2}$ *samples, then with probability at least* $1-\delta$, *the update rule of* PANDA (*Algorithm 1*) *in* (14) *satisfies* (15).

Remark 4.5 The Corollary 4.4 can be interpreted in this way: if N is at least $12m\hat{\kappa}\log(2n_{\mathbf{x}}m/\delta)$, then (15) holds with probability at least $1-\delta$ where $\eta = \eta_{\text{PANDA}} \overset{\text{def}}{=} \sqrt{\frac{3\hat{\kappa}m\log(2n_{\mathbf{x}}m/\delta)}{N}}$.

We highlight the advancements of the PANDA method in the following two aspects:

(a) We compare PANDA with its single-agent version, which corresponds to using N/m samples to construct the approximated Hessian in PAN. According to Corollary 3.3 and Corollary 4.4, we observe that $\eta_{\text{PAN}} = \sqrt{\frac{3\hat{\kappa}m\log(2n_{\mathbf{x}}/\delta)}{N}} \approx \eta_{\text{PANDA}}$. This indicates that the linear-quadratic rate (15) of PANDA significantly improves upon its single-agent version (9), which demonstrates the superiority of using the distributed framework.

(b) We compare PANDA with state-of-the-art first-order methods in Table 2. Both distributed EG and Proxskip-VI-FL (Zhang et al., 2024) require the communication rounds of $\mathcal{O}(\kappa_{\mathbf{g}}\log(1/\epsilon))$, whereas PANDA only requires $\mathcal{O}\Big(\frac{\log(1/\epsilon)}{\log(N/(m\hat{\kappa}))}\Big)$ communication rounds. This highlights the advantage of using second-order information.

4.3 Extension to the GIANT-PANDA Algorithm

PANDA exhibits provably faster convergence rates than the first-order methods for minimax distributed optimization, however, each client is required to access the full local Hessian at

each iteration. In this section, we develop a communication-efficient algorithm that allows using inexact Hessian instead of the exact one during the local computation.

We focus on a specific function class that $l_j(\cdot)$ in (2) can be expressed as $l_j(\mathbf{x}, \mathbf{y}) \overset{\text{def}}{=} h_j(\mathbf{w}^\top \mathbf{x}, \mathbf{y}) + \frac{\mu}{2}\|\mathbf{x}\|^2$, where $h_j(\cdot,\cdot)$ is convex in $\mathbf{x}$ and μ-strongly concave in $\mathbf{y}$. This function class generalizes the objective considered in convex optimization as discussed in GIANT (Wang et al., 2018), which has important applications in regression-type models.

The partial Hessian of the objective at $(\mathbf{x}_t, \mathbf{y}_t)$ can be written as

$$\mathbf{H}_{\mathbf{xx}}(\mathbf{x}_t, \mathbf{y}_t) = \frac{1}{N}\sum_{j=1}^{N} \nabla^2_{\mathbf{xx}} h_j\big(\mathbf{w}^\top \mathbf{x}_t, \mathbf{y}_t\big)\mathbf{w}\mathbf{w}^\top + \mu \mathbf{I} = \frac{1}{m}\sum_{i=1}^{m} \big\{\mathbf{A}_t^\top \mathbf{S}^i (\mathbf{S}^i)^\top \mathbf{A}_t + \mu \mathbf{I}\big\},$$

where $\mathbf{A}_t \overset{\text{def}}{=} \big[\mathbf{a}_1^\top, \cdots, \mathbf{a}_N^\top\big] \in \mathbb{R}^{N \times n_{\mathbf{x}}}$ is a full column-rank matrix with $n_{\mathbf{x}} \le N$, $\mathbf{a}_j = \sqrt{\nabla^2_{\mathbf{xx}} h_j(\mathbf{w}^\top \mathbf{x}_t, \mathbf{y}_t)}\mathbf{w}/\sqrt{N}$, $\mathbf{S}^i$ is some sketching matrix such that $(\mathbf{S}^i)^\top \mathbf{A}_t$ contains the rows of $\mathbf{A}_t$ indexed by $\mathcal{S}_i$. The local partial Hessian of the i-th client can be indicated by $\mathbf{H}^i_{\mathbf{xx}}(\mathbf{x}_t, \mathbf{y}_t) \overset{\text{def}}{=} \big\{\mathbf{A}_t^\top \mathbf{S}^i (\mathbf{S}^i)^\top \mathbf{A}_t + \mu \mathbf{I}\big\}$

Taking advantage of such a structure, we perform a sketch operation on $\mathbf{H}^i_{\mathbf{xx}}(\mathbf{x}_t, \mathbf{y}_t)$ to reduce the computation cost on the client such that:

$$\tilde{\mathbf{H}}^i_{\mathbf{xx},t} \overset{\text{def}}{=} \mathbf{A}_t^\top \tilde{\mathbf{S}}^i_t (\tilde{\mathbf{S}}^i_t)^\top \mathbf{A}_t + \mu \mathbf{I}, \tag{16}$$

where $\tilde{\mathbf{S}}^i_t \in \mathbb{R}^{n_{\mathbf{x}} \times s_t}$ is chosen randomly from the columns of $\mathbf{S}_i$ so that $s_t \le s$. We replace $\mathbf{H}^i_{\mathbf{xx}}(\mathbf{x}_t, \mathbf{y}_t)$ by its sketched approximation $\tilde{\mathbf{H}}^i_{\mathbf{xx}}(\mathbf{x}_t, \mathbf{y}_t)$ in line 15 and line 16 of PANDA (Algorithm 1), naturally resulting the modified algorithm GIANT-PANDA. The routine of GIANT-PANDA is formally presented in Algorithm 2 in A.

Now, we start to characterize the convergence behavior of GIANT-PANDA. Since GIANT-PANDA inherits the framework of PANDA, the update rule of GIANT-PANDA can be viewed as

$$\begin{bmatrix} \mathbf{x}_{t+1} \\ \mathbf{y}_{t+1} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_t \\ \mathbf{y}_t \end{bmatrix} - \begin{bmatrix} \tilde{\mathbf{H}}^{\text{gp}}_{\mathbf{xx},t} & \mathbf{H}_{\mathbf{xy}}(\mathbf{x}_t, \mathbf{y}_t) \\ \mathbf{H}_{\mathbf{xy}}(\mathbf{x}_t, \mathbf{y}_t)^\top & \mathbf{H}_{\mathbf{yy}}(\mathbf{x}_t, \mathbf{y}_t) \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{g}_{\mathbf{x}}(\mathbf{x}_t, \mathbf{y}_t) \\ \mathbf{g}_{\mathbf{y}}(\mathbf{x}_t, \mathbf{y}_t) \end{bmatrix}, \tag{17}$$

where $\tilde{\mathbf{H}}^{\text{gp}}_{\mathbf{xx},t} \overset{\text{def}}{=} \Big[\frac{1}{m}\sum_{i=1}^{m} [\tilde{\mathbf{H}}^i_{\mathbf{xx},t}]^{-1}\Big]^{-1}$.

Let $\mathbf{C}^{\text{gp}}_t \overset{\text{def}}{=} \tilde{\mathbf{H}}^{\text{gp}}_{\mathbf{xx},t} - \mathbf{H}_{\mathbf{xy}}(\mathbf{x}_t, \mathbf{y}_t)[\mathbf{H}_{\mathbf{yy}}(\mathbf{x}_t, \mathbf{y}_t)]^{-1}(\mathbf{H}_{\mathbf{xy}}(\mathbf{x}_t, \mathbf{y}_t))^\top$, the following lemma shows $\mathbf{C}^{\text{gp}}_t$ is still a good approximation to $\mathbf{P}(\mathbf{x}_t, \mathbf{y}_t)$.

Lemma 4.6 *Let $\eta, \delta \in (0,1)$ be fixed parameters, $\nu_t = \nu(\mathbf{A}_t)$, and $\{\tilde{\mathbf{S}}^i_t\}$ is independent uniform sampling matrices with $s_t \ge \frac{3\nu_t n_{\mathbf{x}}}{\eta^2}\log\left(\frac{mn_{\mathbf{x}}}{\delta}\right)$. Under Assumption 2.2, we have*

$$\Big\|\mathbf{I} - \mathbf{P}(\mathbf{x}_t, \mathbf{y}_t)^{1/2}(\mathbf{C}^{\text{gp}}_t)^{-1}\mathbf{P}(\mathbf{x}_t, \mathbf{y}_t)^{1/2}\Big\| \le \frac{\eta}{\sqrt{m}} + \frac{\eta^2}{1-\eta}$$

holds with probability at least $1-\delta$.

Remark 4.7 The condition of Lemma 4.6 requires $\{\tilde{\mathbf{S}}^i_t\}$ to be uniform sampling matrices, which means we perform uniform sketch to obtain the local approximate Hessian $\tilde{\mathbf{H}}^i(\mathbf{x}_t, \mathbf{y}_t)$

in GIANT-PANDA. GIANT-PANDA also allows using other sketching techniques like count sketch (Clarkson & Woodruff, 2017; Meng & Mahoney, 2013) or Gaussian sketch (Johnson & Lindenstrauss, 1984) to obtain $\tilde{\mathbf{S}}_t^i$. These sketching methods can improve the dependence of s_t on ν_t, but will be more expensive to implement than the simple uniform sketching matrix (Wang et al., 2018).

Using the analysis framework of PAN, we establish the linear-quadratic rate of GIANT-PANDA.

Theorem 4.8 *Under the same condition of Lemma 4.6, the update of* GIANT-PANDA *(Algorithm 2) in (17) satisfies*

$$\lambda(\mathbf{x}_{t+1}, \mathbf{y}_{t+1}) \leq \left(\frac{\eta}{\sqrt{m}} + \frac{\eta^2}{1-\eta} \right) \lambda(\mathbf{x}_t, \mathbf{y}_t) + \frac{c\kappa_{\mathbf{g}}^2 \kappa_{\mathbf{H}}}{\sqrt{\mu}} \lambda(\mathbf{x}_t, \mathbf{y}_t)^2. \tag{18}$$

with probability at least $1-\delta$, *where* $c = \frac{12((\sqrt{m}-1)(\eta^2-\eta+1)+1)^2}{(1-\eta)^2 m}$

Remark 4.9 $\left(\frac{\eta}{\sqrt{m}} + \frac{\eta^2}{1-\eta} \right)$ in the linear term $\lambda(\mathbf{x}_t, \mathbf{y}_t)$ in (18) for GIANT-PANDA is slightly worse than $\left(\frac{\eta^2}{1-\eta} \right)$ in (15) for PANDA. This is because GIANT-PANDA uses the approximate local partial Hessian instead of the full local partial Hessian. However, it is still better than $\left(\frac{\eta}{1-\eta} \right)$ in (9) for PAN by a factor of $\frac{1}{\sqrt{m}}$. This demonstrates the advantage of utilizing m clients in the parallel training process.

Improved Results for GIANT. GIANT (Wang et al., 2018) (Algorithm 3 in Appendix A) can be regarded as a special case of GIANT-PANDA for *convex* optimization when taking $n_{\mathbf{y}} = 0$. Using the analysis techniques developed for GIANT-PANDA, we also improve the convergence results for GIANT.

In the following corollary, we present a sharper linear-quadratic rate for GIANT under the same assumption as in (Wang et al., 2018), which improves the previous result by a factor of $\sqrt{\kappa_{\mathbf{g}}}$ in the linear term.

Corollary 4.10 *To solve the minimization problem* $(\mu > 0)$ $\min_{\mathbf{x}\in\mathbb{R}^{n_{\mathbf{x}}}} f(\mathbf{x}) = \frac{1}{N}\sum_{j=1}^{N} h_j(\mathbf{w}^\top \mathbf{x}) + \frac{\mu}{2}\|\mathbf{x}\|^2$ *on m clients and each client holds s samples, if* $h_j(\cdot)$ *is a convex loss function,* $f(\cdot)$ *has* L_2*-Lipschitz continuous Hessian, and* s_t *satisfies that* $s \geq s_t \geq \frac{3\nu_t n_{\mathbf{x}}}{\eta^2} \log\left(\frac{mn_{\mathbf{x}}}{\delta} \right)$ *for some fixed parameters* $\eta, \delta \in (0,1)$, *then with probability at least* $1-\delta$, *the update rule of* GIANT *(Algorithm 3) satisfies that*

$$\hat{\lambda}(\mathbf{x}_{t+1}) \leq \left(\frac{\eta}{\sqrt{m}} + \frac{\eta^2}{1-\eta} \right) \hat{\lambda}(\mathbf{x}_t) + \frac{2L_2}{\mu^{3/2}} \hat{\lambda}(\mathbf{x}_t)^2,$$

where $\hat{\lambda}(\mathbf{x}) \overset{\text{def}}{=} \sqrt{\nabla f(\mathbf{x})[\nabla^2 f(\mathbf{x})]^{-1} \nabla f(\mathbf{x})}$.

5 Experiments

We validate the proposed methods on the following important data mining tasks, which enjoy the structure of "unbalanced dimension" and have been well studied in previous literature (Liu et al., 2022; Liu & Luo, 2022). The experiments are conducted on a workstation with an Intel(R) Core(TM) i7-10870 H CPU @ 2.20GHz. The code was executed using Python 3.8.

- **AUC Maximization**. To train a classifier $\mathbf{w}$ on imbalanced datasets $\{\mathbf{a}_j, b_j\}_{j=1}^N$ such that $p = \frac{N^+}{N} \approx 1$ or 0 where N^+ is the number of positive instances, AUC maximization can be reformulated into minimax problems, where $l_j(\mathbf{x}, \mathbf{y})$ of (1) takes the following quadratic form

$$l_j(\mathbf{x}, y) = (1-p)\big(\big(\mathbf{w}^\top \mathbf{a}_j - u\big)^2 - 2(1+y)\mathbf{w}^\top \mathbf{a}_j\big)\mathbb{I}_{b_j=1} + \frac{\lambda}{2}\|\mathbf{x}\|^2 + p\big(\big(\mathbf{w}^\top \mathbf{a}_j - v\big)^2 + 2(1+y)\mathbf{w}^\top \mathbf{a}_j\big)\mathbb{I}_{b_j=-1} - p(1-p)y^2,$$

  where $\mathbf{x} = [\mathbf{w}; u; v] \in \mathbb{R}^{d+2}, \mathbf{w} \in \mathbb{R}^d, u \in \mathbb{R}, v \in \mathbb{R}$ and $y \in \mathbb{R}$. We set $\lambda = 0.5$. We perform experiments on "a9a" ($N = 32,651$, $n_{\mathbf{x}} = 125$, $n_{\mathbf{y}} = 1$, $p = 0.241$), "w8a" ($N = 45,546, n_{\mathbf{x}} = 302, n_{\mathbf{y}} = 1, p = 0.029$), and "sido0" ($N = 12,678$, $n_{\mathbf{x}} = 4,932$, $n_{\mathbf{y}} = 1$) which can be downloaded from Libsvm (Chang & Lin, 2011). We choose the regularized parameter $\lambda = 0.5$ and the number of the clients $m = 8$. We tune the learning rates of all methods (include the baselines) from $\{1.0, 0.9, \ldots, 0.1\}$.
- **Fairness-Aware Machine Learning.** Given the training set $\{\mathbf{a}_j, b_j, c_j\}_{j=1}^N$ where $\mathbf{a}_j \in \mathbb{R}^d$ and $c_j \in \mathbb{R}$, we can use the following adversarial training model to train a binary classifier $\mathbf{x}$ (Zhang et al., 2018) and make it unbiased to the feature c_j that we want to protect:

$$l_j(\mathbf{x}, y) = \log\left(1+\exp\left(-b_j(\mathbf{a}_j)^\top \mathbf{x}\right)\right) + \lambda\|\mathbf{x}\|^2 - \gamma y^2 - \beta \log\left(1+\exp\left(-c_j(\mathbf{a}_j)^\top \mathbf{x} y\right)\right).$$

  We choose $\lambda = 0.5$ and $\beta = \gamma = 0.0001$. We conduct experiments on "adult" ($N = 32,651, n_{\mathbf{x}} = 122, n_{\mathbf{y}} = 1$) and "law school" ($N = 20,427, n_{\mathbf{x}} = 379, n_{\mathbf{y}} = 1$) datasets (Le Quy et al., 2022; Liu & Luo, 2022). We set regularization parameters $\lambda = 0.5$ and $\beta = \gamma = 0.0001$.

5.1 Comparison with the baselines

We compare `PANDA` and `GIANT-PANDA` with existing state-of-the-art communication-efficient methods. Specifically, we adopt distributed version of extra gradient (Korpelevich, 1976; Tseng, 2000) (`EG`), federated gradient descent ascent with gradient tracking (Sun & Wei, 2022) (`FedGDA`), and proximal skip method for variational inequalities (Zhang et al., 2024) (`ProxSkip`) as the baselines. Both `EG` and `ProxSkip` achieve the optimal communication complexity for first-order methods. We tune the learning rates of all methods from $\{1.0, 0.9, \ldots, 0.1\}$.

For all experiments, we use 70% percent local data in `GIANT-PANDA`. The results for AUC maximization under different client numbers $m = 8$ and $m = 128$ are presented in

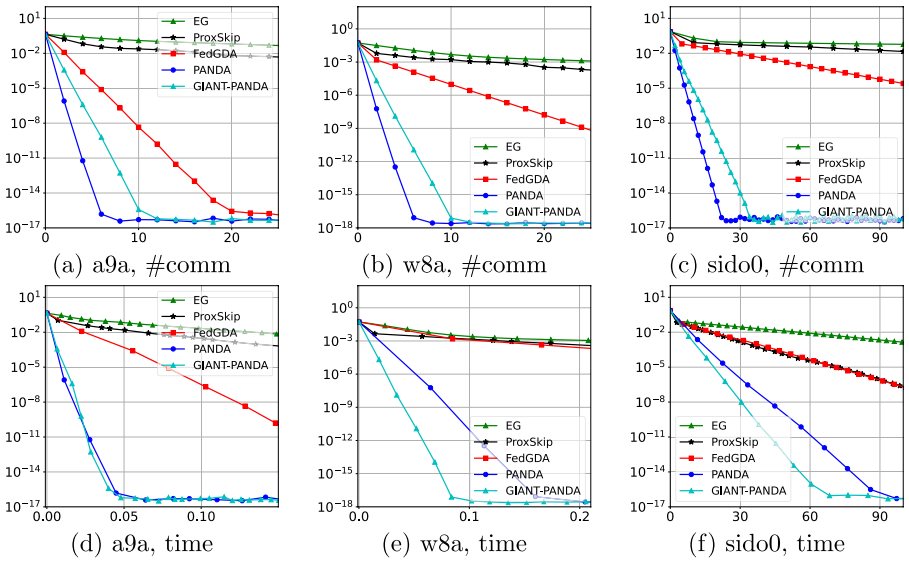


Fig. 1 We demonstrate the communication rounds (#comm) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ and running time (seconds) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ for AUC maximization under the case $m = 8$ on datasets "a9a", "w8a", and "sido0"

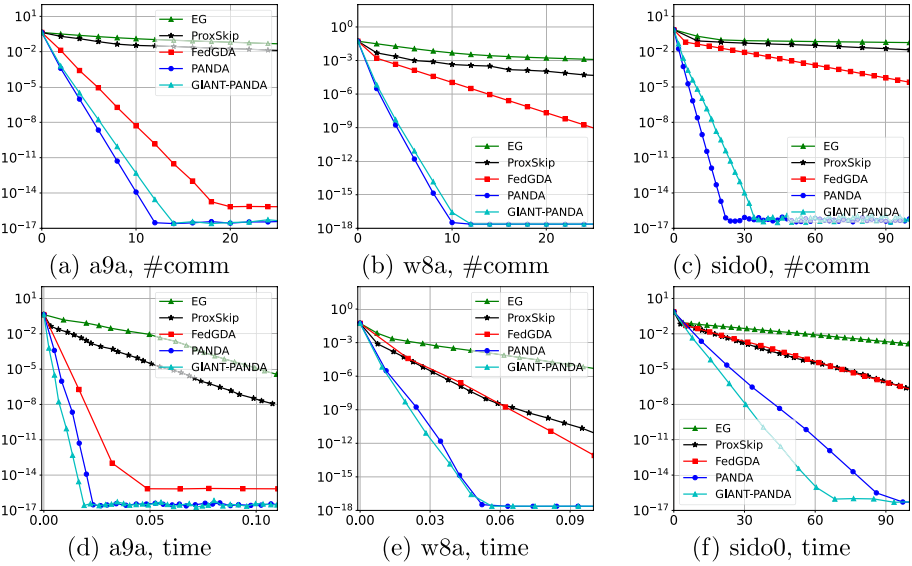


Fig. 2 We demonstrate the communication rounds (#comm) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ and running time (seconds) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ for AUC maximization under the case $m = 128$ on datasets "a9a", "w8a", and "sido0"

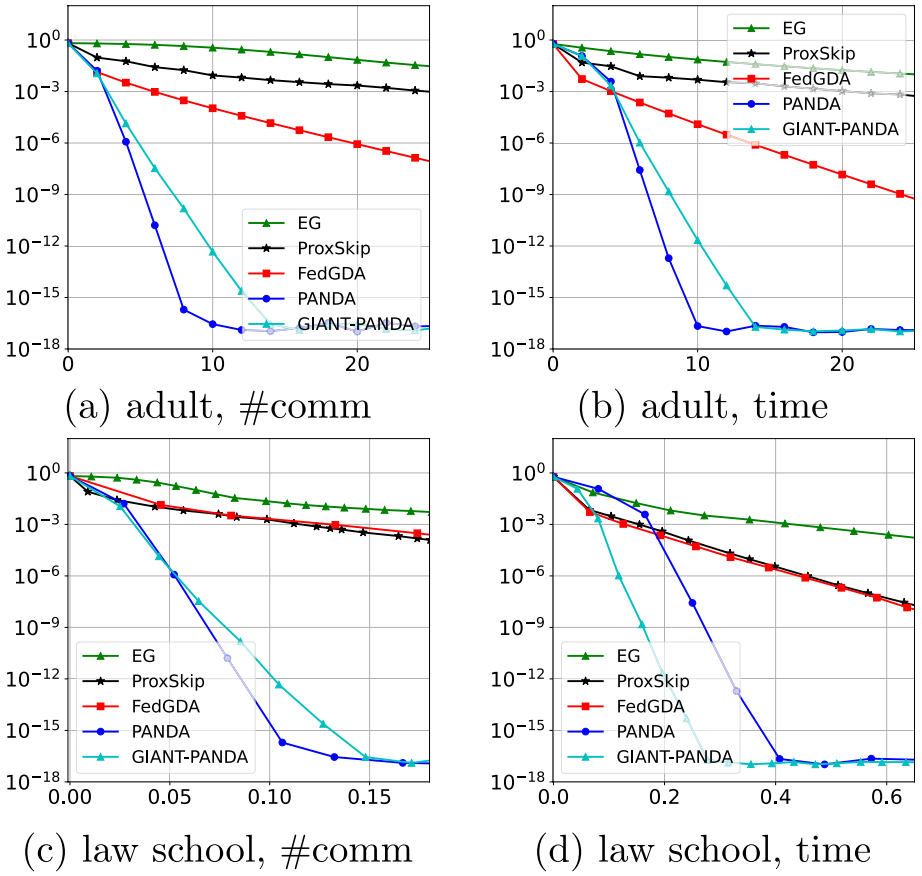


Fig. 3 We demonstrate the communication rounds (#comm) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ and running time (second) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ for Fairness-aware machine learning under the case $m = 8$ on datasets "adult" and "law school"

Figs. 1 and 2. We also demonstrate the results for Fairness-aware machine learning under different client numbers $m = 8$ and $m = 128$ in Figs. 3 and 4.

We observe that our newly proposed PANDA and GIANT-PANDA outperform the baselines in terms of both communication rounds and the running time for all cases. This indicates that our methods indeed not only significantly reduce the communication rounds as compared to the optimal first-order methods, but also maintain communication efficiency which makes the optimization procedure fast.

We also observe that the communication complexity of PANDA can be affected by the number of clients (m). This is because η_{PANDA} is proportional to $\sqrt{m}$ according to Remark 4.5. On the other hand, the increase of m makes the training time per iteration smaller due to the distributed framework, thus, larger m always leads to a faster training process. Take (a), (b) of Figs. 1 and 2 for example, PANDA requires less communication round when $m = 8$, but takes less running time when $m = 128$.

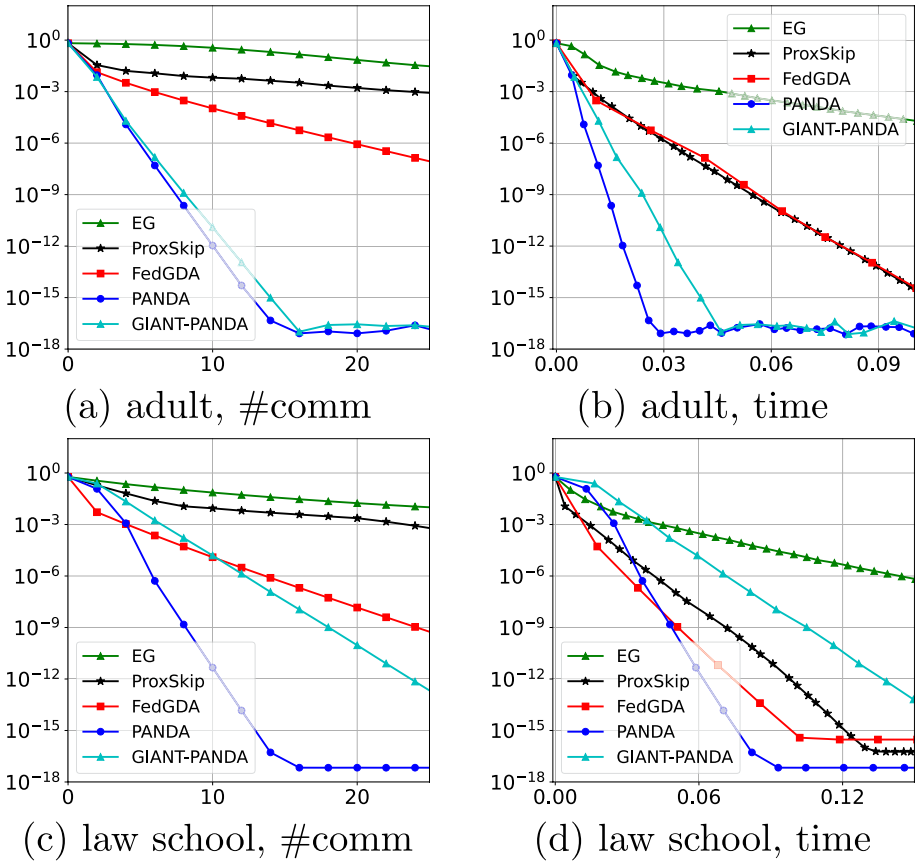


Fig. 4 We demonstrate the communication rounds (#comm) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ and running time (second) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ for Fairness-aware machine learning under the case $m = 128$ on datasets "adult" and "law school"

5.2 Comparison of different sketch ratios for GIANT-PANDA

We investigate the impact of the sketch ratio ($p = s_t/s$) on GIANT-PANDA. We choose different sketch ratios p from $\{10\%, 30\%, 50\%, 70\%, 100\%\}$ for GIANT-PANDA. For the case $p = 100\%$, GIANT-PANDA reduces to its full version PANDA. We set the number of clients as $m = 8$.

We present the results for AUC maximization and Fairness-aware machine learning in Figs. 5 and 6 respectively. The numerical results show that larger sketch ratios lead to fewer communication rounds for the training process, which is because one can obtain a better approximation to the local exact partial Hessian, and thus get a smaller η. GIANT-PANDA with $p = 100\%$ (PANDA) outperforms other cases in terms of the communication rounds. On the other hand, GIANT-PANDA shows its advantage in terms of the running time. We find that GIANT-PANDA with $p = 30\%$ for "a9a", $p = 10\%$ for "w8a" in AUC maximization and with $p = 10\%$ for "law school" in Fairness-aware machine learning achieves the best behavior in terms of the running time ((b), (d) of

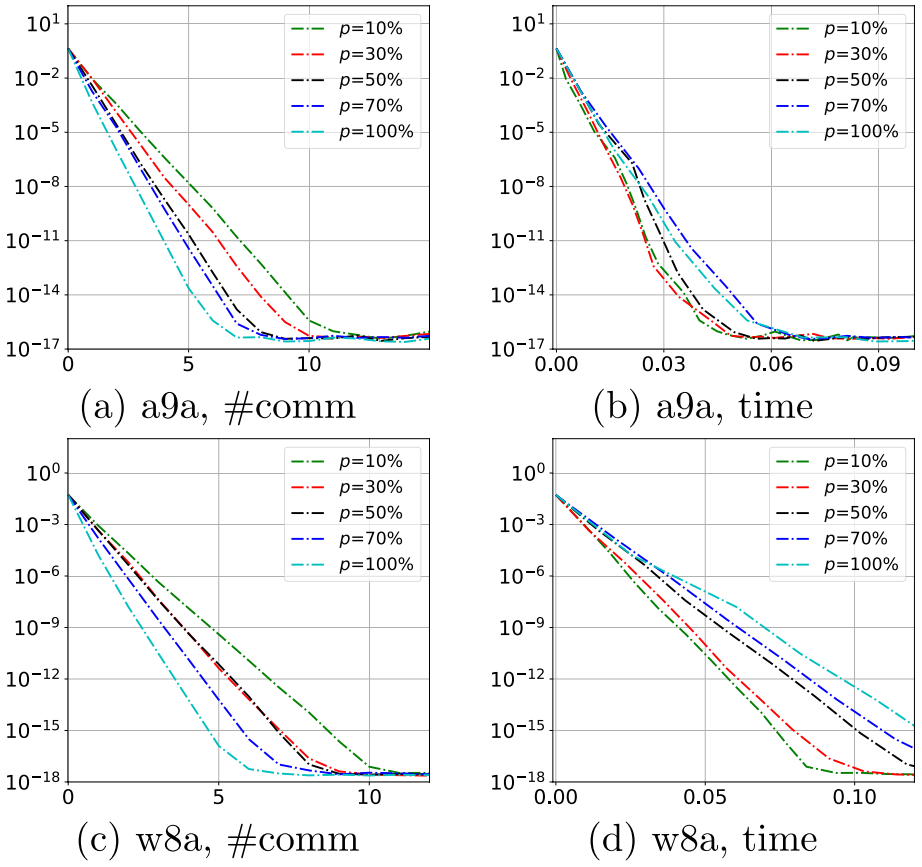


Fig. 5 We demonstrate the communication rounds (#comm) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ and running time (second) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ for AUC maximization on datasets "a9a" and "w8a" with different sketch ratio p under the case $m = 8$

Figs. 5 and 6). This is because the sketch operation in GIANT-PANDA reduces the computation time for each client.

We also provide additional experiments to study the impact of sketch ratio under $m = 128$ and the impact of using different sketch methods in G.

6 Conclusion

In this paper, we have proposed PANDA and GIANT-PANDA to solve the distributed minimax problems with unbalanced dimensions. PANDA eliminates the requirement of communicating the full Hessian and substantially reduces the communication rounds compared to the optimal first-order methods. GIANT-PANDA further reduces the computation cost by performing sketch operations to compute the local partial Hessian on each client.

For future work, it is interesting to generalize PANDA and GIANT-PANDA to more general minimax optimization problems (Adil et al., 2022; Lin & Jordan, 2022; Liu & Luo, 2022; Luo et al., 2022). It is also possible to leverage the idea of Hessian average (Na et al.,

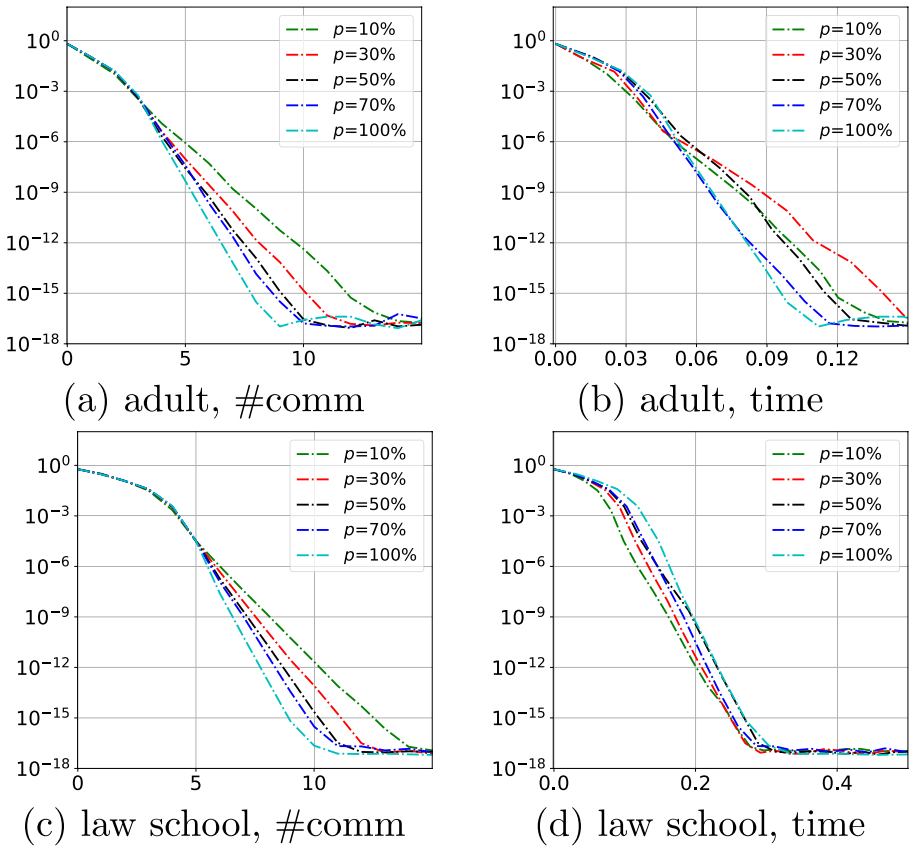


Fig. 6 We demonstrate the communication rounds (#comm) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ and running time (second) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ for Fairness-aware machine learning on datasets "adult" and "law school" with different sketch ratio p under the case $m = 8$

2023) to further enhance the behavior of GIANT-PANDA and design the decentralized scenario of PANDA and GIANT-PANDA.

The Appendix A GIANT-PANDA algorithm

We present the detailed implementation of GIANT-PANDA and GIANT in Algorithms 2 and 3 respectively.

Algorithm 2 GIANT-PANDA($\mathbf{x}_0, \mathbf{y}_0, T$)

for $t = 0, \ldots, T$ **do**
  for $i = 1, \ldots, m$ **do in parallel**
    i-th client:
      compute
        $\mathbf{g}_{\mathbf{x}}^i = \mathbf{g}_{\mathbf{x}}^i(\mathbf{x}_t, \mathbf{y}_t), \mathbf{g}_{\mathbf{y}}^i = \mathbf{g}_{\mathbf{y}}^i(\mathbf{x}_t, \mathbf{y}_t)$
        $\mathbf{H}_{\mathbf{xy}}^i = \mathbf{H}_{\mathbf{xy}}^i(\mathbf{x}_t, \mathbf{y}_t), \mathbf{H}_{\mathbf{yy}}^i = \mathbf{H}_{\mathbf{xy}}^i(\mathbf{x}_t, \mathbf{y}_t)$
      send $\mathbf{g}_{\mathbf{x}}^i$, $\mathbf{g}_{\mathbf{y}}^i$, $\mathbf{H}_{\mathbf{xy}}^i$ and $\mathbf{H}_{\mathbf{yy}}^i$
  end for
  server:
    aggregate $\mathbf{g}_{\mathbf{x}}, \mathbf{g}_{\mathbf{y}}$, $\mathbf{H}_{\mathbf{xy}}$ and $\mathbf{H}_{\mathbf{yy}}$ from (11)
    broadcast $\mathbf{g}_{\mathbf{x}}$ and $\mathbf{H}_{\mathbf{xy}}$
  for $i = 1, \ldots, n$ **do in parallel**
    i-th client:
      compute $\tilde{\mathbf{H}}_{\mathbf{xx}}^i$ according to (16)
      compute
        $\mathbf{Q}_{\mathbf{xy}}^i = [\tilde{\mathbf{H}}_{\mathbf{xx}}^i]^{-1}\mathbf{H}_{\mathbf{xy}}$
        $\mathbf{q}_{\mathbf{x}}^i = [\tilde{\mathbf{H}}_{\mathbf{xx}}^i]^{-1}\mathbf{g}_{\mathbf{x}}$
      send $\mathbf{Q}_{\mathbf{xy}}^i$ and $\mathbf{q}_{\mathbf{x}}^i$
  end for
  server:
    aggregate $\mathbf{Q}_{\mathbf{xy}}, \mathbf{q}_{\mathbf{x}}$ from (12)
    compute update directions $\tilde{\mathbf{d}}_{\mathbf{x}}$, $\tilde{\mathbf{d}}_{\mathbf{y}}$ from (13)
    update the models
      $\mathbf{x}_{t+1} = \mathbf{x}_t - \tilde{\mathbf{d}}_{\mathbf{x}}$
      $\mathbf{y}_{t+1} = \mathbf{y}_t - \tilde{\mathbf{d}}_{\mathbf{y}}$.
    broadcast $\mathbf{x}_{t+1}$ and $\mathbf{y}_{t+1}$
end for

Algorithm 3 GIANT($\mathbf{x}_0, T$)

for $t = 0, \ldots, T$ **do**
  for $i = 1, \ldots, m$ **do in parallel**
    i-th client:
      compute and send $\mathbf{g}_{\mathbf{x}}^i = \nabla^i f(\mathbf{x}_t)$
  end for
  server:
    aggregate and broadcast $\mathbf{g}_{\mathbf{x}} = \frac{1}{m} \sum \mathbf{g}_{\mathbf{x}}^i$
  for $i = 1, \ldots, n$ **do in parallel**
    i-th client:
      compute and send $\mathbf{d}_{\mathbf{x}}^i = [\tilde{\mathbf{H}}_{\mathbf{xx}}^i(\mathbf{x}_t)]^{-1} \mathbf{g}_{\mathbf{x}}$
  end for
  server:
    aggregate $\mathbf{d}_{\mathbf{x}} = \frac{1}{m} \sum_{i=1}^m \mathbf{d}_{\mathbf{x}}^i$
    update and broadcast the models $\mathbf{x}_{t+1} = \mathbf{x}_t - \mathbf{d}_{\mathbf{x}}$
end for

Appendix B Auxiliary Lemmas for positive definite matrices

We first provide some useful lemmas for positive definite matrices.

Lemma B.1 *For two positive definite matrices* $\mathbf{A}, \mathbf{B}$, *if*

$$(1 - \eta)\mathbf{A} \preceq \mathbf{B} \preceq (1 + \eta)\mathbf{A},$$

for some $\eta \in (0, 1)$, *then it holds that*

$$\left\| \mathbf{I} - \mathbf{A}^{1/2} \mathbf{B}^{-1} \mathbf{A}^{1/2} \right\| \leq \frac{\eta}{1 - \eta}.$$

Proof We have

$$\frac{1}{1 + \eta} \mathbf{A}^{-1} \preceq \mathbf{B}^{-1} \preceq \frac{1}{1 - \eta} \mathbf{A}^{-1},$$

so that

$$\frac{1}{1 + \eta} \mathbf{I} \preceq \mathbf{A}^{1/2} \mathbf{B}^{-1} \mathbf{A}^{1/2} \preceq \frac{1}{1 - \eta} \mathbf{I},$$

and

$$\frac{\eta}{1 - \eta} \mathbf{I} \preceq \mathbf{I} - \mathbf{A}^{1/2} \mathbf{B}^{-1} \mathbf{A}^{1/2} \preceq \frac{\eta}{1 + \eta} \mathbf{I}.$$

So, we have

$$\left\|\mathbf{I}-\mathbf{A}^{1/2}\mathbf{B}^{-1}\mathbf{A}^{1/2}\right\| \leq \max\left\{\frac{\eta}{1-\eta}, \frac{\eta}{1+\eta}\right\} = \frac{\eta}{1-\eta}.$$

□

Lemma B.2 *For two positive definite matrices* $\mathbf{A}, \mathbf{B}$, *if*

$$\|\mathbf{I}-\mathbf{A}^{1/2}\mathbf{B}^{-1}\mathbf{A}^{1/2}\| \leq \eta, \tag{B1}$$

for some $\eta \in (0,1)$, *then it holds that*

$$\|\mathbf{I}-(\mathbf{A}+\Delta)^{1/2}(\mathbf{B}+\Delta)^{-1}(\mathbf{A}+\Delta)^{1/2}\| \leq \eta,$$

for any $\Delta \succeq \mathbf{0}$.

Proof According to (B1), we have

$$(1-\eta)\mathbf{I} \preceq \mathbf{A}^{1/2}\mathbf{B}^{-1}\mathbf{A}^{1/2} \preceq (1+\eta)\mathbf{I},$$

which means

$$(1-\eta)\mathbf{A}^{-1} \preceq \mathbf{B}^{-1} \preceq (1+\eta)\mathbf{A}^{-1},$$

so that

$$\frac{1}{1+\eta}\mathbf{A} \preceq \mathbf{B} \preceq \frac{1}{1-\eta}\mathbf{A}.$$

Since $\Delta \succeq \mathbf{0}$, we have

$$\frac{1}{1+\eta}(\mathbf{A}+\Delta) \preceq \frac{1}{1+\eta}\mathbf{A}+\Delta \preceq \mathbf{B}+\Delta \preceq \frac{1}{1-\eta}\mathbf{A}+\Delta \preceq \frac{1}{1-\eta}(\mathbf{A}+\Delta),$$

which means

$$(1-\eta)(\mathbf{A}+\Delta)^{-1} \preceq (\mathbf{B}+\Delta)^{-1} \preceq (1+\eta)(\mathbf{A}+\Delta)^{-1}.$$

So that we have

$$(1-\eta)\mathbf{I} \preceq (\mathbf{A}+\Delta)^{1/2}(\mathbf{B}+\Delta)^{-1}(\mathbf{A}+\Delta)^{1/2} \preceq (1+\eta)\mathbf{I}.$$

Finally, we have

$$-\eta\mathbf{I} \preceq \mathbf{I}-(\mathbf{A}+\Delta)^{1/2}(\mathbf{B}+\Delta)^{-1}(\mathbf{A}+\Delta)^{1/2} \preceq \eta\mathbf{I}.$$

□

Appendix C The Proof of Sect. 2.2

C.1 The Proof of Lemma 2.4

Proof Recall the singular value decomposition of $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}$ in Definition 2.1 where $\mathbf{U} \in \mathbb{R}^{N\times d}$, $\Sigma \in \mathbb{R}^{d\times d}$ and $\mathbf{V} \in \mathbb{R}^{d\times N}$. We can directly obtain the following results by taking $\rho = d$ of Wang et al. (2018, Lemma 8) that

$$\left\|\mathbf{U}^\top \mathbf{S}_i\mathbf{S}_i^\top \mathbf{U} - \mathbf{I}\right\| \le \eta \quad \text{and} \quad \left\|\mathbf{U}^\top \mathbf{S}\mathbf{S}^\top \mathbf{U} - \mathbf{I}\right\| \le \frac{\eta}{\sqrt{m}},$$

holds for all $i \in [m]$ with probability at least $1 - \delta$. Then we have

$$(1-\eta)\mathbf{I} \preceq \mathbf{U}^\top \mathbf{S}_i\mathbf{S}_i^\top \mathbf{U} \preceq (1+\eta)\mathbf{I},$$

and

$$\left(1-\eta/\sqrt{m}\right)\mathbf{I} \preceq \mathbf{U}^\top \mathbf{S}\mathbf{S}^\top \mathbf{U} \preceq \left(1+\eta/\sqrt{m}\right)\mathbf{I}.$$

Recall the definition of $\mathbf{H}_i = \mathbf{A}^\top \mathbf{S}_i\mathbf{S}_i^\top \mathbf{A} + \alpha$, $\mathbf{H} = \mathbf{A}^\top \mathbf{A} + \alpha\mathbf{I}$, we have

$$\begin{aligned}\mathbf{H}_i &= \mathbf{V}^\top \Sigma \mathbf{U}^\top \mathbf{S}_i\mathbf{S}_i^\top \mathbf{U}\Sigma\mathbf{V} + \alpha\mathbf{I}\\ &\preceq (1+\eta)\mathbf{V}^\top \Sigma^2\mathbf{V} + \alpha\mathbf{I}\\ &\preceq (1+\eta)\left(\mathbf{A}^\top \mathbf{A} + \alpha\mathbf{I}\right) = (1+\eta)\mathbf{H},\end{aligned}$$

and

$$\begin{aligned}\mathbf{H}_i &= \mathbf{V}^\top \Sigma \mathbf{U}^\top \mathbf{S}_i\mathbf{S}_i^\top \mathbf{U}\Sigma\mathbf{V} + \alpha\mathbf{I}\\ &\succeq (1-\eta)\mathbf{V}^\top \Sigma^2\mathbf{V} + \alpha\mathbf{I}\\ &\succeq (1-\eta)(\mathbf{A}^\top \mathbf{A} + \alpha\mathbf{I}) = (1-\eta)\mathbf{H}.\end{aligned}$$

Similarly, recall the definition of $\hat{\mathbf{H}} = \mathbf{A}^\top \mathbf{S}\mathbf{S}^\top \mathbf{A} + \alpha$, we have

$$\begin{aligned}\hat{\mathbf{H}} &= \mathbf{V}^\top \Sigma \mathbf{U}^\top \mathbf{S}\mathbf{S}^\top \mathbf{U}\Sigma\mathbf{V} + \alpha\mathbf{I}\\ &\preceq (1+\eta/\sqrt{m})\mathbf{V}^\top \Sigma^2\mathbf{V} + \alpha\mathbf{I}\\ &\preceq (1+\eta/\sqrt{m})(\mathbf{A}^\top \mathbf{A} + \alpha\mathbf{I}) = (1+\eta/\sqrt{m})\mathbf{H},\end{aligned}$$

and

$$\begin{aligned}\hat{\mathbf{H}} &= \mathbf{V}^\top \Sigma \mathbf{U}^\top \mathbf{S}\mathbf{S}^\top \mathbf{U}\Sigma\mathbf{V} + \alpha\mathbf{I}\\ &\succeq (1-\eta/\sqrt{m})\mathbf{V}^\top \Sigma^2\mathbf{V} + \alpha\mathbf{I}\\ &\succeq (1-\eta/\sqrt{m})(\mathbf{A}^\top \mathbf{A} + \alpha\mathbf{I}) = (1-\eta/\sqrt{m})\mathbf{H}.\end{aligned}$$

□

Appendix D The Proof of Sect. 3

D.1 The Proof of Lemma 3.1

Proof According to the condition, it holds that

$$(1-\eta)\mathbf{H}_{\mathbf{xx}}(\mathbf{x},\mathbf{y}) \preceq \tilde{\mathbf{H}}_{\mathbf{xx}} \preceq (1+\eta)\mathbf{H}_{\mathbf{xx}}(\mathbf{x},\mathbf{y}).$$

Using Lemma B.1, we have

$$\left\|\mathbf{I} - [\mathbf{H}_{\mathbf{xx}}(\mathbf{x},\mathbf{y})]^{1/2}[\tilde{\mathbf{H}}_{\mathbf{xx}}]^{-1}[\mathbf{H}_{\mathbf{xx}}(\mathbf{x},\mathbf{y})]^{1/2}\right\| \leq \frac{\eta}{1-\eta}.$$

Denote $\Delta = -\mathbf{H}_{\mathbf{xy}}(\mathbf{x},\mathbf{y})\mathbf{H}_{\mathbf{yy}}(\mathbf{x},\mathbf{y})^{-1}\mathbf{H}_{\mathbf{xy}}^{\top}(\mathbf{x},\mathbf{y})$. According to Assumption 2.2, we have $\mathbf{H}_{\mathbf{yy}}(\mathbf{x},\mathbf{y}) \preceq -\mu]\mathbf{I}$, which means that $\Delta \succeq \mathbf{0}$. Using Lemma B.2 on $\mathbf{A} = \mathbf{H}_{\mathbf{xx}}(\mathbf{x},\mathbf{y})$ and $\mathbf{B} = \tilde{\mathbf{H}}_{\mathbf{xx}}$, we have

$$\left\|\mathbf{I} - \mathbf{P}(\mathbf{x},\mathbf{y})^{1/2}\mathbf{C}(\mathbf{x},\mathbf{y})^{-1}\mathbf{P}(\mathbf{x},\mathbf{y})^{1/2}\right\| \leq \frac{\eta}{1-\eta}.$$

□

D.2 The Proof of Theorem 3.2

Denote $\mathbf{J} = \begin{bmatrix} \mathbf{P}(\mathbf{x},\mathbf{y})^{1/2} & \mathbf{0} \\ \mathbf{0} & \frac{\sqrt{\mu}}{2}\mathbf{I}_{n_{\mathbf{y}}} \end{bmatrix}$ and $r = \left\| \begin{bmatrix} \mathbf{P}(\mathbf{x},\mathbf{y})^{1/2}(\mathbf{x}_+ - \mathbf{x}) \\ \frac{\sqrt{\mu}}{2}(\mathbf{y}_+ - \mathbf{y}) \end{bmatrix} \right\|$. The following lemma illustrates the relation between $\mathbf{P}(\mathbf{x},\mathbf{y})$ and $\mathbf{P}(\mathbf{x}_+,\mathbf{y}_+)$.

Lemma D.1 (Liu et al., 2022, Lemma 4.3) *Under Assumptions 2.2, we have*

- $\frac{1}{1+\frac{3\kappa^2\kappa_2}{\sqrt{\mu}}r}\mathbf{P}(\mathbf{x},\mathbf{y}) \preceq \mathbf{P}(\mathbf{x}_+,\mathbf{y}_+) \preceq \left(1+\frac{3\kappa^2\kappa_2}{\sqrt{\mu}}r\right)\mathbf{P}(\mathbf{x},\mathbf{y})$.
- $\|\mathbf{J}\| \geq \frac{\sqrt{\mu}}{2}$.

For simplicity, we use $\mathbf{g}_{\mathbf{x}}$, $\mathbf{g}_{\mathbf{y}}$, $\mathbf{H}_{\mathbf{xx}}$, $\mathbf{H}_{\mathbf{xy}}$, $\mathbf{H}_{\mathbf{yy}}$, $\mathbf{P}$, $\mathbf{P}_+$, $\mathbf{C}$ to represent the $\mathbf{g}_{\mathbf{x}}(\mathbf{x},\mathbf{y})$, $\mathbf{g}_{\mathbf{y}}(\mathbf{x},\mathbf{y})$, $\mathbf{H}_{\mathbf{xx}}(\mathbf{x},\mathbf{y})$, $\mathbf{H}_{\mathbf{xy}}(\mathbf{x},\mathbf{y})$, $\mathbf{H}_{\mathbf{yy}}(\mathbf{x},\mathbf{y})$, $\mathbf{P}_{\mathbf{xx}}(\mathbf{x},\mathbf{y})$, $\mathbf{P}(\mathbf{x}_+,\mathbf{y}_+)$ and $\mathbf{C}(\mathbf{x},\mathbf{y})$. We also represent the full accurate Hessian matrix $\mathbf{H} = \mathbf{H}(\mathbf{x},\mathbf{y}) = \begin{bmatrix} \mathbf{H}_{\mathbf{xx}}(\mathbf{x},\mathbf{y}) & \mathbf{H}_{\mathbf{xy}}(\mathbf{x},\mathbf{y}) \\ \mathbf{H}_{\mathbf{yx}}(\mathbf{x},\mathbf{y}) & \mathbf{H}_{\mathbf{yy}}(\mathbf{x},\mathbf{y}) \end{bmatrix}$ and approximated full Hessian matrix $\tilde{\mathbf{H}} = \tilde{\mathbf{H}}(\mathbf{x},\mathbf{y}) = \begin{bmatrix} \tilde{\mathbf{H}}_{\mathbf{xx}}(\mathbf{x},\mathbf{y}) & \mathbf{H}_{\mathbf{xy}}(\mathbf{x},\mathbf{y}) \\ \mathbf{H}_{\mathbf{yx}}(\mathbf{x},\mathbf{y}) & \mathbf{H}_{\mathbf{yy}}(\mathbf{x},\mathbf{y}) \end{bmatrix}$.

Proof According to the condition $\|\mathbf{I} - \mathbf{P}^{1/2}\mathbf{C}^{-1}\mathbf{P}^{1/2}\| \leq \eta_1$, we can obtain that

$$\left\|\mathbf{P}^{1/2}\mathbf{C}^{-1}\mathbf{P}^{1/2}\right\| \leq \|\mathbf{I}\| + \left\|\mathbf{I} - \mathbf{P}^{1/2}\mathbf{C}^{-1}\mathbf{P}^{1/2}\right\| \leq 1 + \eta_1, \tag{D2}$$

Under assumption 2.2, we have $\mathbf{H}_{\mathbf{xx}} \succeq \mu\mathbf{I}$ and $\mathbf{H}_{\mathbf{yy}} \preceq -\mu\mathbf{I}$, then

$$\mu \mathbf{I} \preceq -\mathbf{H}_{\mathbf{yy}} \preceq -\mathbf{H}_{\mathbf{yy}} + \mathbf{H}_{\mathbf{xy}}^{\top} \mathbf{H}_{\mathbf{xx}}^{-1} \mathbf{H}_{\mathbf{xy}},$$

and

$$\mu \mathbf{I} \preceq -\mathbf{H}_{\mathbf{yy}} \preceq -\mathbf{H}_{\mathbf{yy}} + \mathbf{H}_{\mathbf{xy}}^{\top} \tilde{\mathbf{H}}_{\mathbf{xx}}^{-1} \mathbf{H}_{\mathbf{xy}}.$$

If follows that

$$\left\| \left[\mathbf{H}_{\mathbf{yy}} - \mathbf{H}_{\mathbf{xy}}^{\top} \mathbf{H}_{\mathbf{xx}}^{-1} \mathbf{H}_{\mathbf{xy}}(\mathbf{x}, \mathbf{y}) \right]^{-1} \right\| \leq \frac{1}{\mu}, \tag{D3}$$

and

$$\left\| \left[\mathbf{H}_{\mathbf{yy}} - \mathbf{H}_{\mathbf{xy}}^{\top} \tilde{\mathbf{H}}_{\mathbf{xx}}^{-1} \mathbf{H}_{\mathbf{xy}} \right]^{-1} \right\| \leq \frac{1}{\mu}. \tag{D4}$$

According to Woodbury identity, we have

$$\left[\mathbf{H}_{\mathbf{yy}} - \mathbf{H}_{\mathbf{xy}}^{\top} \mathbf{H}_{\mathbf{xx}}^{-1} \mathbf{H}_{\mathbf{xy}} \right]^{-1} = \mathbf{H}_{\mathbf{yy}}^{-1} + \mathbf{H}_{\mathbf{yy}}^{-1} \mathbf{H}_{\mathbf{xy}}^{\top} \mathbf{P}^{-1} \mathbf{H}_{\mathbf{xy}} \mathbf{H}_{\mathbf{yy}}^{-1}.$$

Hence, we have

$$\begin{aligned} \left\| \mathbf{P}^{-1/2} \mathbf{H}_{\mathbf{xy}} \mathbf{H}_{\mathbf{yy}}^{-1} \right\| &= \sqrt{\lambda_{\max} \left(\mathbf{H}_{\mathbf{yy}}^{-1} \mathbf{H}_{\mathbf{xy}}^{\top} \mathbf{P}^{-1} \mathbf{H}_{\mathbf{xy}} \mathbf{H}_{\mathbf{yy}}^{-1} \right)} \\ &= \sqrt{\left\| \mathbf{H}_{\mathbf{yy}}^{-1} \mathbf{H}_{\mathbf{xy}}^{\top} \mathbf{P}^{-1} \mathbf{H}_{\mathbf{xy}} \mathbf{H}_{\mathbf{yy}}^{-1} \right\|} \\ &= \sqrt{\left\| \left[\mathbf{H}_{\mathbf{yy}} - \mathbf{H}_{\mathbf{xy}}^{\top} \mathbf{H}_{\mathbf{xx}}^{-1} \mathbf{H}_{\mathbf{xy}} \right]^{-1} - \mathbf{H}_{\mathbf{yy}}^{-1} \right\|} \\ &\leq \sqrt{\left\| \left[\mathbf{H}_{\mathbf{yy}} - \mathbf{H}_{\mathbf{xy}}^{\top} \mathbf{H}_{\mathbf{xx}}^{-1} \mathbf{H}_{\mathbf{xy}} \right]^{-1} \right\| + \left\| \mathbf{H}_{\mathbf{yy}}^{-1} \right\|} \\ &\leq \frac{2}{\sqrt{\mu}}. \end{aligned} \tag{D5}$$

According to the update rule, we have

$$\begin{aligned} \begin{bmatrix} \mathbf{g}_{\mathbf{x}}(\mathbf{x}_+, \mathbf{y}_+) \\ \mathbf{g}_{\mathbf{y}}(\mathbf{x}_+, \mathbf{y}_+) \end{bmatrix} &= \underbrace{\begin{bmatrix} \tilde{\mathbf{H}}_{\mathbf{xx}} - \mathbf{H}_{\mathbf{xx}} & 0 \\ 0 & 0 \end{bmatrix} \tilde{\mathbf{H}}^{-1} \begin{bmatrix} \mathbf{g}_{\mathbf{x}} \\ \mathbf{g}_{\mathbf{y}} \end{bmatrix}}_{\boldsymbol{\Upsilon}} \\ &+ \underbrace{\int_0^1 \left(\left[\mathbf{H}\big(\mathbf{x} + s(\mathbf{x}_+ - \mathbf{x}), \mathbf{y} + s(\mathbf{y}_+ - \mathbf{y})\big) - \mathbf{H}(\mathbf{x}, \mathbf{y}) \right] \begin{bmatrix} \mathbf{x}_+ - \mathbf{x} \\ \mathbf{y}_+ - \mathbf{y} \end{bmatrix} \right) \mathrm{d}s}_{\boldsymbol{\Gamma}}. \end{aligned} \tag{D6}$$

Using block inverse formula, we can write $\boldsymbol{\Upsilon}$ as

$$\begin{bmatrix}\boldsymbol{\Upsilon}_{\mathbf{x}}\\ \boldsymbol{\Upsilon}_{\mathbf{y}}\end{bmatrix} = \begin{bmatrix}(\tilde{\mathbf{H}}_{\mathbf{xx}} - \mathbf{H}_{\mathbf{xx}})\mathbf{C}^{-1}\mathbf{g}_{\mathbf{x}} - (\tilde{\mathbf{H}}_{\mathbf{xx}} - \mathbf{H}_{\mathbf{xx}})\mathbf{C}^{-1}\mathbf{H}_{\mathbf{xy}}\mathbf{H}_{\mathbf{yy}}^{-1}\mathbf{g}_{\mathbf{y}}\\ 0\end{bmatrix}.$$

Let $\begin{bmatrix}\boldsymbol{\zeta}_{\mathbf{x}}\\ \boldsymbol{\zeta}_{\mathbf{y}}\end{bmatrix} = \mathbf{J}^{-1}\begin{bmatrix}\mathbf{g}_{\mathbf{x}}\\ \mathbf{g}_{\mathbf{y}}\end{bmatrix} = \begin{bmatrix}\mathbf{P}^{-1/2} & 0\\ 0 & \frac{2}{\sqrt{\mu}}\mathbf{I}_{n_{\mathbf{y}}}\end{bmatrix}\begin{bmatrix}\mathbf{g}_{\mathbf{x}}\\ \mathbf{g}_{\mathbf{y}}\end{bmatrix} = \begin{bmatrix}\mathbf{P}^{-1/2}\mathbf{g}_{\mathbf{x}}\\ \frac{2}{\sqrt{\mu}}\mathbf{g}_{\mathbf{y}}\end{bmatrix}$, then the weighted gradient norm can be written as

$$\lambda(\mathbf{x},\mathbf{y}) = \langle \mathbf{g}_{\mathbf{x}}, \mathbf{P}^{-1}\mathbf{g}_{\mathbf{x}}\rangle^{1/2} + \frac{2}{\sqrt{\mu}}\|\mathbf{g}_{\mathbf{y}}\| = \|\boldsymbol{\zeta}_x\| + \|\boldsymbol{\zeta}_y\|.$$

From (D6) we can build the following relationship

$$\begin{aligned}\mathbf{g}_{\mathbf{x}}(\mathbf{x}_+,\mathbf{y}_+) &= \boldsymbol{\Upsilon}_{\mathbf{x}} + \boldsymbol{\Gamma}_{\mathbf{x}},\\ \mathbf{g}_{\mathbf{y}}(\mathbf{x}_+,\mathbf{y}_+) &= \boldsymbol{\Upsilon}_{\mathbf{y}} + \boldsymbol{\Gamma}_{\mathbf{y}} = \boldsymbol{\Gamma}_{\mathbf{y}}.\end{aligned}$$

Hence, in order to build the relationship between λ_+ and λ, we need to bound $\left\|\mathbf{P}_+^{-1/2}\boldsymbol{\Upsilon}_{\mathbf{x}}\right\|$, $\left\|\mathbf{P}_+^{-1/2}\boldsymbol{\Gamma}\right\|_{\mathbf{x}}$ and $\|\boldsymbol{\Gamma}\|_y$. Since

$$\begin{aligned}\left\|\mathbf{P}^{-1/2}(\tilde{\mathbf{H}}_{\mathbf{xx}} - \mathbf{H}_{\mathbf{xx}})\mathbf{C}^{-1}\mathbf{g}_{\mathbf{x}}\right\| &\overset{(7),(8)}{=} \left\|\mathbf{P}^{-1/2}(\mathbf{C}-\mathbf{P})\mathbf{C}^{-1}\mathbf{g}_{\mathbf{x}}\right\|\\ &= \left\|\left(\mathbf{I} - \mathbf{P}^{1/2}\mathbf{C}^{-1}\mathbf{P}^{1/2}\right)\mathbf{P}^{-1/2}\mathbf{g}_{\mathbf{x}}\right\|\\ &\le \left\|\mathbf{I} - \mathbf{P}^{1/2}\mathbf{C}^{-1}\mathbf{P}^{1/2}\right\|\left\|\mathbf{P}^{-1/2}\mathbf{g}_{\mathbf{x}}\right\|\\ &\le \eta_1\left\|\mathbf{P}^{-1/2}\mathbf{g}_{\mathbf{x}}\right\|,\end{aligned}$$

and

$$\begin{aligned}&\left\|\mathbf{P}^{-1/2}(\tilde{\mathbf{H}}_{\mathbf{xx}} - \mathbf{H}_{\mathbf{xx}})\mathbf{C}^{-1}\mathbf{H}_{\mathbf{xy}}\mathbf{H}_{\mathbf{yy}}^{-1}\mathbf{g}_{\mathbf{y}}\right\|\\ &\quad= \left\|\left(\mathbf{I} - \mathbf{P}^{1/2}\mathbf{C}^{-1}\mathbf{P}^{1/2}\right)\mathbf{P}^{-1/2}\mathbf{H}_{\mathbf{xy}}\mathbf{H}_{\mathbf{yy}}\mathbf{g}_{\mathbf{y}}\right\|\\ &\quad\overset{(D5)}{\le} \eta_1\frac{2}{\sqrt{\mu}}\left\|\mathbf{g}_{\mathbf{y}}\right\|,\end{aligned}$$

we have

$$\begin{aligned}\left\|\mathbf{P}^{-1/2}\boldsymbol{\Upsilon}_{\mathbf{x}}\right\| &\le \left\|\mathbf{P}^{-1/2}(\tilde{\mathbf{H}}_{\mathbf{xx}} - \mathbf{H}_{\mathbf{xx}})\mathbf{C}^{-1}\mathbf{g}_{\mathbf{x}}\right\|\\ &\quad+ \left\|\mathbf{P}^{-1/2}(\tilde{\mathbf{H}}_{\mathbf{xx}} - \mathbf{H}_{\mathbf{xx}})\mathbf{C}^{-1}\mathbf{H}_{\mathbf{xy}}\mathbf{H}_{\mathbf{yy}}^{-1}\mathbf{g}_{\mathbf{y}}\right\|\\ &\le \eta_1\left\|\mathbf{P}^{-1/2}\mathbf{g}_{\mathbf{x}}\right\| + \eta_1\frac{2}{\sqrt{\mu}}\left\|\mathbf{g}_{\mathbf{y}}\right\|.\end{aligned}$$

Using Lemma D.1, wehave

$$\begin{aligned}\left\|\mathbf{P}_+^{-1/2}\boldsymbol{\Upsilon}_{\mathbf{x}}\right\| &\le \sqrt{1+\frac{3\kappa^2\kappa_2}{\sqrt{\mu}}r}\left\|\mathbf{P}^{-1/2}\boldsymbol{\Upsilon}_{\mathbf{x}}\right\| \\ &\le \sqrt{1+\frac{3\kappa^2\kappa_2}{\sqrt{\mu}}r}\left(\eta_1\left\|\mathbf{P}^{-1/2}\mathbf{g}_{\mathbf{x}}\right\|+\eta_1\frac{2}{\sqrt{\mu}}\left\|\mathbf{g}_{\mathbf{y}}\right\|\right).\end{aligned}$$

Then, the term $\|\boldsymbol{\Gamma}\|$ can be bounded under assumption 2.2, that is

$$\begin{aligned}\|\boldsymbol{\Gamma}\| &\le \int_0^1 \left\|\mathbf{H}\big(\mathbf{x}+s(\mathbf{x}_+-\mathbf{x}),\mathbf{y}+s(\mathbf{y}_+-\mathbf{y})-\mathbf{H}(\mathbf{x},\mathbf{y})\right\|\left\|\begin{bmatrix}\mathbf{x}_+-\mathbf{x}\\ \mathbf{y}_+-\mathbf{y}\end{bmatrix}\right\|\mathrm{d}s \\ &\le \frac{L_2}{2}\left\|\begin{bmatrix}\mathbf{x}_+-\mathbf{x}\\ \mathbf{y}_+-\mathbf{y}\end{bmatrix}\right\|^2 \le \frac{L_2}{2\mu}\left\|\mathbf{J}\begin{bmatrix}\mathbf{x}_+-\mathbf{x}\\ \mathbf{y}_+-\mathbf{y}\end{bmatrix}\right\|^2 \le \frac{L_2}{2\mu}r^2.\end{aligned}$$

We can further bound $\left\|\mathbf{P}_+^{-1/2}\boldsymbol{\Gamma}_{\mathbf{x}}\right\|$ as follows:

$$\begin{aligned}\left\|\mathbf{P}_+^{-1/2}\boldsymbol{\Gamma}_{\mathbf{x}}\right\| &\le \left\|\mathbf{P}_+^{-1/2}\right\|\|\boldsymbol{\Gamma}_{\mathbf{x}}\| \\ &\le \frac{1}{\sqrt{\mu}}\frac{L_2}{2\mu}r^2 = \frac{L_2}{2\mu\sqrt{\mu}}r^2.\end{aligned}$$

With all bounds we have obtained, we are able to construct an incomplete relationship as follows:

$$\begin{aligned}\lambda(\mathbf{x}_+,\mathbf{y}_+) &= \left\|\mathbf{P}_+^{-1/2}\mathbf{g}_x(\mathbf{x}_+,\mathbf{y}_+)\right\|+\left\|\frac{2}{\sqrt{\mu}}\mathbf{g}_{\mathbf{y}}(\mathbf{x}_+,\mathbf{y}_+)\right\| \\ &= \left\|\mathbf{P}_+^{-1/2}(\boldsymbol{\Upsilon}_{\mathbf{x}}+\boldsymbol{\Gamma}_{\mathbf{x}})\right\|+\left\|\frac{2}{\sqrt{\mu}}\boldsymbol{\Gamma}_{\mathbf{y}}\right\| \\ &\le \left\|\mathbf{P}_+^{-1/2}\boldsymbol{\Upsilon}_{\mathbf{x}}\right\|+\left\|\mathbf{P}_+^{-1/2}\boldsymbol{\Gamma}_{\mathbf{x}}\right\|+\frac{2}{\sqrt{\mu}}\|\boldsymbol{\Gamma}\| \\ &\le \eta_1\sqrt{1+\frac{3\kappa^2\kappa_2}{\sqrt{\mu}}r}\lambda+\frac{L_2}{2\mu\sqrt{\mu}}r^2+\frac{2}{\sqrt{\mu}}\frac{L_2}{2\mu}r^2 \\ &= \eta_1\sqrt{1+\frac{3\kappa^2\kappa_2}{\sqrt{\mu}}r}\lambda+\frac{3L_2}{2\mu\sqrt{\mu}}r^2.\end{aligned} \tag{D7}$$

Then we need to bound r by λ. By the update rule, we have

$$\mathbf{J}\begin{bmatrix}\mathbf{x}_+-\mathbf{x}\\ \mathbf{y}_+-\mathbf{y}\end{bmatrix} = -\mathbf{J}\tilde{\mathbf{H}}^{-1}\begin{bmatrix}\mathbf{g}_{\mathbf{x}}\\ \mathbf{g}_{\mathbf{y}}\end{bmatrix} = -\mathbf{J}\tilde{\mathbf{H}}^{-1}\mathbf{J}\begin{bmatrix}\boldsymbol{\zeta}_{\mathbf{x}}\\ \boldsymbol{\zeta}_{\mathbf{y}}\end{bmatrix},$$

which is equivalent to

$$
\begin{aligned}
-\begin{bmatrix} \mathbf{P}^{1/2}(\mathbf{x}_+ - \mathbf{x}) \\ \frac{\sqrt{\mu}}{2}(\mathbf{y}_+ - \mathbf{y}) \end{bmatrix} &= \begin{bmatrix} \mathbf{P}^{1/2}\mathbf{C}^{-1}\mathbf{P}^{1/2}\zeta_{\mathbf{x}} \\ -\frac{\sqrt{\mu}}{2}(\mathbf{P}^{1/2}\mathbf{C}^{-1}\mathbf{P}^{1/2}\mathbf{P}^{-1/2}\mathbf{H}_{\mathbf{xy}}\mathbf{H}_{\mathbf{yy}}^{-1})^\top \zeta_{\mathbf{x}} \end{bmatrix} \\
&\quad + \begin{bmatrix} -\frac{\sqrt{\mu}}{2}\mathbf{P}^{1/2}\mathbf{C}^{-1}\mathbf{H}_{\mathbf{xy}}\mathbf{H}_{\mathbf{yy}}^{-1}\zeta_{\mathbf{y}} \\ \frac{\mu}{4}(\mathbf{H}_{\mathbf{yy}} - \mathbf{H}_{\mathbf{yx}}\tilde{\mathbf{H}}^{-1}\mathbf{H}_{\mathbf{xy}})^{-1}\zeta_{\mathbf{y}} \end{bmatrix}
\end{aligned}
$$

Then we can bound $\left\|\mathbf{P}^{1/2}(\mathbf{x}_+ - \mathbf{x})\right\|$ and $\left\|\frac{\sqrt{\mu}}{2}(\mathbf{y}_+ - \mathbf{y})\right\|$ as follows:

$$
\begin{aligned}
\left\|\mathbf{P}^{1/2}(\mathbf{x}_+ - \mathbf{x})\right\| &\le \left\|\mathbf{P}^{1/2}\mathbf{C}^{-1}\mathbf{P}^{1/2}\right\|\|\zeta_{\mathbf{x}}\| \\
&\quad + \frac{\sqrt{\mu}}{2}\left\|\mathbf{P}^{1/2}\mathbf{C}^{-1}\mathbf{P}^{1/2}\right\|\left\|\mathbf{P}^{-1/2}\mathbf{H}_{\mathbf{xy}}\mathbf{H}_{\mathbf{yy}}^{-1}\right\|\left\|\zeta_{\mathbf{y}}\right\| \\
&\overset{(D2),\,(D5)}{\le} (1+\eta_1)\|\zeta_{\mathbf{x}}\| + \frac{\sqrt{\mu}}{2}(1+\eta_1)\frac{2}{\sqrt{\mu}}\left\|\zeta_{\mathbf{y}}\right\| \\
&= (1+\eta_1)\lambda,
\end{aligned}
$$

and

$$
\begin{aligned}
\left\|\frac{\sqrt{\mu}}{2}(\mathbf{y}_+ - \mathbf{y})\right\| &\le \frac{\sqrt{\mu}}{2}\left\|\mathbf{P}^{1/2}\mathbf{C}^{-1}\mathbf{P}^{1/2}\right\|\left\|\mathbf{P}^{-1/2}\mathbf{H}_{\mathbf{xy}}\mathbf{H}_{\mathbf{yy}}^{-1}\right\|\|\zeta_{\mathbf{x}}\| \\
&\quad + \frac{\mu}{4}\left\|\left(\mathbf{H}_{\mathbf{yy}} - \mathbf{H}_{\mathbf{yx}}\tilde{\mathbf{H}}^{-1}\mathbf{H}_{\mathbf{xy}}\right)^{-1}\right\|\left\|\zeta_{\mathbf{y}}\right\| \\
&\le \frac{\sqrt{\mu}}{2}(1+\eta_1)\frac{2}{\sqrt{\mu}}\|\zeta_{\mathbf{x}}\| + \frac{\mu}{4}\cdot\frac{1}{\mu}\left\|\zeta_{\mathbf{y}}\right\| \\
&\le (1+\eta_1)\lambda.
\end{aligned}
$$

Combine these two bounds, we have

$$
\begin{aligned}
r &= \left\|\mathbf{P}^{1/2}(\mathbf{x}_+ - \mathbf{x})\right\| + \left\|\frac{\sqrt{\mu}}{2}(\mathbf{y}_+ - \mathbf{y})\right\| \\
&\le 2(1+\eta_1)\lambda.
\end{aligned}
$$

Plug this back to (D7), we have

$$
\begin{aligned}
\lambda(\mathbf{x}_+, \mathbf{y}_+) &\le \eta_1\sqrt{1 + \frac{3\kappa^2\kappa_2}{\sqrt{\mu}}r\lambda} + \frac{3L_2}{2\mu\sqrt{\mu}}r^2 \\
&\le \eta_1\left(1 + \frac{3\kappa^2\kappa_2}{\sqrt{\mu}}r\right)\lambda + \frac{3L_2}{2\mu\sqrt{\mu}}r^2 \\
&\le \eta_1\lambda + \frac{6\eta_1\kappa^2\kappa_2}{\sqrt{\mu}}(1+\eta_1)^2\lambda^2 + \frac{6\kappa_2}{\sqrt{\mu}}(1+\eta_1)^2\lambda^2 \\
&\le \eta_1\lambda + \frac{12(1+\eta_1)^2\kappa^2\kappa_2}{\sqrt{\mu}}\lambda^2.
\end{aligned} \tag{D8}
$$

□

D.3 The Proof of Corollary 3.3

Proof According Lemma 2.3 and take $\eta = \eta_{\text{PAN}} = \sqrt{\frac{3K\log(2n_{\mathbf{x}}/\delta)}{|\mathcal{S}|\mu}}$, it holds with probability at least $1-\delta$ that

$$(1-\eta)\mathbf{H}_{\mathbf{xx}}(\mathbf{x},\mathbf{y}) \preceq \tilde{\mathbf{H}}_{\mathbf{xx}} \preceq (1+\eta)\mathbf{H}_{\mathbf{xx}}(\mathbf{x},\mathbf{y}).$$

Using Lemma 3.1, we have

$$\left\|\mathbf{I} - \mathbf{P}(\mathbf{x},\mathbf{y})^{1/2}[\mathbf{C}(\mathbf{x},\mathbf{y})]^{-1}\mathbf{P}(\mathbf{x},\mathbf{y})^{1/2}\right\| \leq \frac{\eta}{1-\eta},$$

holds with probability at least $1-\delta$. Using Theorem 3.2, we can directly conclude (9).

□

Appendix E The Proof of Sect. 4.2

To simplify the presentation, we use $\mathbf{H}_{\mathbf{xx},t}$, $\mathbf{H}^i_{\mathbf{xx},t}$ to represent $\mathbf{H}_{\mathbf{xx}}(\mathbf{x}_t,\mathbf{y}_t)$ and $\mathbf{H}^i_{\mathbf{xx}}(\mathbf{x}_t,\mathbf{y}_t)$.

E.1 The Proof of Lemma 4.2

Proof We denote

$$\mathbf{H}^{1/2}_{\mathbf{xx},t}\left[\mathbf{H}^i_{\mathbf{xx},t}\right]^{-1}\mathbf{H}^{1/2}_{\mathbf{xx},t} \overset{\text{def}}{=} \mathbf{I} + \mathbf{E}^i. \tag{E9}$$

According to the condition for $\mathbf{H}^i_{\mathbf{xx},t}$, we have

$$\frac{1}{1+\eta}\mathbf{I} \preceq \mathbf{H}^{1/2}_{\mathbf{xx}}\left[\mathbf{H}^i_{\mathbf{xx},t}\right]^{-1}\mathbf{H}^{1/2}_{\mathbf{xx}} \preceq \frac{1}{1-\eta}\mathbf{I},$$

so that $\mathbf{E}^i$ satisfies that

$$-\frac{\eta}{1+\eta}\mathbf{I} \preceq \mathbf{E}^i \preceq \frac{\eta}{1-\eta}\mathbf{I}, \tag{E10}$$

and

$$\left\|\mathbf{H}^{-1/2}_{\mathbf{xx},t}(\mathbf{H}_{\mathbf{xx},t} - \mathbf{H}^i_{\mathbf{xx},t})\mathbf{H}^{-1/2}_{\mathbf{xx},t}\right\| \leq \eta. \tag{E11}$$

We rewrite $\mathbf{H}^{1/2}_{\mathbf{xx},t}[\mathbf{H}^i_{\mathbf{xx},t}]^{-1}\mathbf{H}^{1/2}_{\mathbf{xx},t} - \mathbf{I}$ as follows

$$\begin{aligned}&\mathbf{H}^{1/2}_{\mathbf{xx},t}\left[\mathbf{H}^i_{\mathbf{xx},t}\right]^{-1}\mathbf{H}^{1/2}_{\mathbf{xx},t} - \mathbf{I}\\ &\quad= \mathbf{H}^{-1/2}_{\mathbf{xx},t}\left(\mathbf{H}_{\mathbf{xx},t} - \mathbf{H}^i_{\mathbf{xx},t}\right)\mathbf{H}^{-1/2}_{\mathbf{xx},t}\left(\mathbf{H}^{1/2}_{\mathbf{xx},t}\left[\mathbf{H}^i_{\mathbf{xx},t}\right]^{-1}\mathbf{H}^{1/2}_{\mathbf{xx},t}\right).\end{aligned}$$

Then, $\tilde{\mathbf{H}}_{\mathbf{xx},t}$ satisfies that

$$\begin{aligned}\mathbf{H}_{\mathbf{xx},t}^{1/2}[\tilde{\mathbf{H}}_{\mathbf{xx},t}]^{-1}\mathbf{H}_{\mathbf{xx},t}^{1/2}-\mathbf{I}&=\mathbf{H}_{\mathbf{xx},t}^{1/2}\left[\frac{1}{m}\sum_{i=1}^{m}[\mathbf{H}_{\mathbf{xx},t}^{i}]^{-1}\right]\mathbf{H}_{\mathbf{xx},t}^{1/2}-\mathbf{I}\\&=\frac{1}{m}\sum_{i=1}^{m}\mathbf{H}_{\mathbf{xx},t}^{-1/2}(\mathbf{H}_{\mathbf{xx},t}-\mathbf{H}_{\mathbf{xx},t}^{i})\mathbf{H}_{\mathbf{xx},t}^{-1/2}(\mathbf{H}_{\mathbf{xx},t}^{1/2}[\mathbf{H}_{\mathbf{xx},t}^{i}]^{-1}\mathbf{H}_{\mathbf{xx},t}^{1/2}).\end{aligned}$$

Thus, we have

$$\begin{aligned}&\left\|\mathbf{H}_{\mathbf{xx},t}^{1/2}\tilde{\mathbf{H}}_{\mathbf{xx},t}^{-1}\mathbf{H}_{\mathbf{xx},t}^{1/2}-\mathbf{I}\right\|\\&=\left\|\frac{1}{m}\sum_{i=1}^{m}\mathbf{H}_{\mathbf{xx},t}^{-1/2}(\mathbf{H}_{\mathbf{xx},t}-\mathbf{H}_{\mathbf{xx},t}^{i})\mathbf{H}_{\mathbf{xx},t}^{-1/2}(\mathbf{H}_{\mathbf{xx},t}^{1/2}[\mathbf{H}_{\mathbf{xx},t}^{i}]^{-1}\mathbf{H}_{\mathbf{xx},t}^{1/2}))\right\|\\&\overset{(E9)}{=}\left\|\frac{1}{m}\sum_{i=1}^{m}\mathbf{H}_{\mathbf{xx},t}^{-1/2}(\mathbf{H}_{\mathbf{xx},t}-\mathbf{H}_{\mathbf{xx},t}^{i})\mathbf{H}_{\mathbf{xx},t}^{-1/2}(\mathbf{I}+\mathbf{E}^{i})\right\|\\&\leq\left\|\frac{1}{m}\sum_{i=1}^{m}\mathbf{H}_{\mathbf{xx},t}^{-1/2}(\mathbf{H}_{\mathbf{xx},t}-\mathbf{H}_{\mathbf{xx},t}^{i})\mathbf{H}_{\mathbf{xx},t}^{-1/2}\right\|\\&\quad+\left\|\frac{1}{m}\sum_{i=1}^{m}\mathbf{H}_{\mathbf{xx},t}^{-1/2}(\mathbf{H}_{\mathbf{xx},t}-\mathbf{H}_{\mathbf{xx},t}^{i})\mathbf{H}_{\mathbf{xx},t}^{-1/2}\mathbf{E}^{i}\right\|\\&\leq\left\|\mathbf{H}_{\mathbf{xx},t}^{-1/2}\underbrace{\left(\mathbf{H}_{\mathbf{xx},t}-\frac{1}{m}\sum_{i=1}^{m}\mathbf{H}_{\mathbf{xx},t}^{i}\right)}_{=\mathbf{0}}\mathbf{H}_{\mathbf{xx},t}^{-1/2}\right\|\\&\quad+\frac{1}{m}\sum_{i=1}^{m}\left\|\mathbf{H}_{\mathbf{xx},t}^{-1/2}(\mathbf{H}_{\mathbf{xx},t}-\mathbf{H}_{\mathbf{xx},t}^{i})\mathbf{H}_{\mathbf{xx},t}^{-1/2}\right\|\left\|\mathbf{E}^{i}\right\|\\&\overset{(E11),(E10)}{\leq}\frac{\eta^{2}}{1-\eta}.\end{aligned}$$

Applying Lemma B.2 on $\mathbf{A}=\mathbf{H}_{\mathbf{xx},t}$, $\mathbf{B}=\tilde{\mathbf{H}}_{\mathbf{xx},t}$ and

$$\Delta=-\mathbf{H}_{\mathbf{xy}}(\mathbf{x}_t,\mathbf{y}_t)[\mathbf{H}_{\mathbf{yy}}(\mathbf{x}_t,\mathbf{y}_t)^{-1}][\mathbf{H}_{\mathbf{xy}}(\mathbf{x}_t,\mathbf{y}_t)]^{\top},$$

we obtain

$$\left\|\mathbf{P}(\mathbf{x}_t,\mathbf{y}_t)^{1/2}\mathbf{C}_t^{-1}\mathbf{P}(\mathbf{x}_t,\mathbf{y}_t)^{1/2}-\mathbf{I}\right\|\leq\frac{\eta^{2}}{1-\eta}.$$

□

E.2 The Proof of Theorem 4.3

Proof Using Lemma 4.2, we have

$$\left\|\mathbf{P}(\mathbf{x}_t,\mathbf{y}_t)^{1/2}\mathbf{C}_t^{-1}\mathbf{P}(\mathbf{x}_t,\mathbf{y}_t)^{1/2}-\mathbf{I}\right\| \leq \frac{\eta^2}{1-\eta}.$$

According to Proposition 4.1 and using Theorem 3.2 by taking $\eta_1 = \frac{\eta^2}{1-\eta}$, we obtain (15). □

E.3 The Proof of Corollary 4.4

Proof Using Lemma 2.3, with sample size on each client $s \geq \frac{3K/\mu \log(2n_{\mathbf{x}}m/\delta)}{\eta^2}$, we have

$$(1-\eta)\mathbf{H}_{\mathbf{xx}}(\mathbf{x}_t,\mathbf{y}_t) \preceq \mathbf{H}^i_{\mathbf{xx}}(\mathbf{x}_t,\mathbf{y}_t) \preceq (1+\eta)\mathbf{H}_{\mathbf{xx}}(\mathbf{x}_t,\mathbf{y}_t),$$

holds with probability at least $1-\delta/m$. Then we have

$$(1-\eta)\mathbf{H}_{\mathbf{xx}}(\mathbf{x}_t,\mathbf{y}_t) \preceq \mathbf{H}^i_{\mathbf{xx}}(\mathbf{x}_t,\mathbf{y}_t) \preceq (1+\eta)\mathbf{H}_{\mathbf{xx}}(\mathbf{x}_t,\mathbf{y}_t),$$

holds for all $i \in [m]$ with probability $1-\delta$, remaining proof directly follows the proof of Theorem 4.3. □

Appendix F The Proof of Sect. 4.3

To simplify the presentation, we use $\mathbf{H}_{\mathbf{xx},t}$, $\mathbf{H}^i_{\mathbf{xx},t}$ to represent $\mathbf{H}_{\mathbf{xx}}(\mathbf{x}_t,\mathbf{y}_t)$ and $\mathbf{H}^i_{\mathbf{xx}}(\mathbf{x}_t,\mathbf{y}_t)$.

F.1 The Proof of Lemma 4.6

Proof Recall that we do matrix sketching on each client such that as

$$\tilde{\mathbf{H}}^i_{\mathbf{xx},t} = \mathbf{A}_t^\top \tilde{\mathbf{S}}_i \tilde{\mathbf{S}}_i^\top \mathbf{A}_t + \alpha\mathbf{I},$$

we denote $\mathbf{S} = \frac{1}{\sqrt{m}}[\tilde{\mathbf{S}}_1,\cdots,\tilde{\mathbf{S}}_m] \in \mathbb{R}^{N\times ms_t}$, then the aggregation of $\tilde{\mathbf{H}}^i_{\mathbf{xx},t}$ can be defined by

$$\hat{\mathbf{H}}_{\mathbf{xx},t} \overset{\text{def}}{=} \frac{1}{m}\sum_{i=1}^m \tilde{\mathbf{H}}^i_{\mathbf{xx},t} = \mathbf{A}_t^\top \mathbf{S}\mathbf{S}^\top \mathbf{A}_t + \alpha\mathbf{I}.$$

Using the results of Lemma 2.4, we have

$$(1-\eta)\mathbf{H}_{\mathbf{xx},t} \preceq \tilde{\mathbf{H}}^i_{\mathbf{xx},t} \preceq (1+\eta)\mathbf{H}_{\mathbf{xx},t}, \tag{F12}$$

and

$$(1-\eta/\sqrt{m})\mathbf{H}_{\mathbf{xx},t} \preceq \hat{\mathbf{H}}_{\mathbf{xx},t} \preceq (1+\eta/\sqrt{m})\mathbf{H}_{\mathbf{xx},t}, \tag{F13}$$

hold with probability at least $1-\delta$. Following the proof of Lemma 4.2, we also denote

$$\mathbf{H}_{\mathbf{xx},t}^{1/2}\left[\tilde{\mathbf{H}}_{\mathbf{xx},t}^{i}\right]^{-1}\mathbf{H}_{\mathbf{xx},t}^{1/2} \overset{\text{def}}{=} \mathbf{I} + \tilde{\mathbf{E}}^{i}. \tag{F14}$$

According to (F12), we have

$$-\frac{\eta}{1+\eta}\mathbf{I} \preceq \tilde{\mathbf{E}}^{i} \preceq \frac{\eta}{1-\eta}\mathbf{I}, \tag{F15}$$

$$\left\|\mathbf{H}_{\mathbf{xx},t}^{-1/2}\left(\mathbf{H}_{\mathbf{xx},t} - \tilde{\mathbf{H}}_{\mathbf{xx},t}^{i}\right)\mathbf{H}_{\mathbf{xx},t}^{-1/2}\right\| \leq \eta. \tag{F16}$$

and

$$\left\|\mathbf{H}_{\mathbf{xx},t}^{-1/2}\left(\mathbf{H}_{\mathbf{xx},t} - \hat{\mathbf{H}}_{\mathbf{xx},t}\right)\mathbf{H}_{\mathbf{xx},t}^{-1/2}\right\| \leq \eta/\sqrt{m}, \tag{F17}$$

hold with probability at least $1-\delta$. It also satisfies that

$$\begin{aligned}\mathbf{H}_{\mathbf{xx},t}^{1/2}[\tilde{\mathbf{H}}_{\mathbf{xx},t}^{\text{gp}}]^{-1}\mathbf{H}_{\mathbf{xx},t}^{1/2} - \mathbf{I} &= \mathbf{H}_{\mathbf{xx},t}^{1/2}\left[\frac{1}{m}\sum_{i=1}^{m}[\tilde{\mathbf{H}}_{\mathbf{xx},t}^{i}]^{-1}\right]\mathbf{H}_{\mathbf{xx},t}^{1/2} - \mathbf{I} \\ &= \frac{1}{m}\sum_{i=1}^{m}\mathbf{H}_{\mathbf{xx},t}^{-1/2}(\mathbf{H}_{\mathbf{xx},t} - \tilde{\mathbf{H}}_{\mathbf{xx},t}^{i})\mathbf{H}_{\mathbf{xx},t}^{-1/2}(\mathbf{H}_{\mathbf{xx},t}^{1/2}[\tilde{\mathbf{H}}_{\mathbf{xx},t}^{i}]^{-1}\mathbf{H}_{\mathbf{xx},t}^{1/2}).\end{aligned}$$

Thus, we have

$$\begin{aligned}&\left\|\mathbf{H}_{\mathbf{xx},t}^{1/2}[\tilde{\mathbf{H}}_{\mathbf{xx},t}^{\text{gp}}]^{-1}\mathbf{H}_{\mathbf{xx},t}^{1/2} - \mathbf{I}\right\| \\ &= \left\|\frac{1}{m}\sum_{i=1}^{m}\mathbf{H}_{\mathbf{xx},t}^{-1/2}(\mathbf{H}_{\mathbf{xx},t} - \tilde{\mathbf{H}}_{\mathbf{xx},t}^{i})\mathbf{H}_{\mathbf{xx},t}^{-1/2}(\mathbf{H}_{\mathbf{xx},t}^{1/2}[\tilde{\mathbf{H}}_{\mathbf{xx},t}^{i}]^{-1}\mathbf{H}_{\mathbf{xx},t}^{1/2})\right\| \\ &= \left\|\frac{1}{m}\sum_{i=1}^{m}\mathbf{H}_{\mathbf{xx},t}^{-1/2}(\mathbf{H}_{\mathbf{xx},t} - \tilde{\mathbf{H}}_{\mathbf{xx},t}^{i})\mathbf{H}_{\mathbf{xx},t}^{-1/2}(\mathbf{I} + \tilde{\mathbf{E}}^{i})\right\| \\ &\leq \left\|\frac{1}{m}\sum_{i=1}^{m}\mathbf{H}_{\mathbf{xx},t}^{-1/2}(\mathbf{H}_{\mathbf{xx},t} - \tilde{\mathbf{H}}_{\mathbf{xx},t}^{i})\mathbf{H}_{\mathbf{xx},t}^{-1/2}\right\| \\ &\quad + \left\|\frac{1}{m}\sum_{i=1}^{m}\mathbf{H}_{\mathbf{xx},t}^{-1/2}(\mathbf{H}_{\mathbf{xx},t} - \mathbf{H}_{\mathbf{xx},t}^{i})\mathbf{H}_{\mathbf{xx},t}^{-1/2}\tilde{\mathbf{E}}^{i}\right\| \\ &\leq \left\|\mathbf{H}_{\mathbf{xx},t}^{-1/2}\left(\mathbf{H}_{\mathbf{xx},t} - \hat{\mathbf{H}}_{\mathbf{xx},t}\right)\mathbf{H}_{\mathbf{xx},t}^{-1/2}\right\| \\ &\quad + \frac{1}{m}\sum_{i=1}^{m}\left\|\mathbf{H}_{\mathbf{xx},t}^{-1/2}(\mathbf{H}_{\mathbf{xx},t} - \mathbf{H}_{\mathbf{xx},t}^{i})\mathbf{H}_{\mathbf{xx},t}^{-1/2}\right\|\left\|\mathbf{E}^{i}\right\| \\ &\overset{(F17),(F15)}{\leq} \frac{\eta}{\sqrt{m}} + \frac{\eta^2}{1-\eta},\end{aligned}$$

holds with probability at least $1-\delta$.

Applying Lemma B.2 on $\mathbf{A} = \mathbf{H}_{\mathbf{xx},t}$, $\mathbf{B} = \tilde{\mathbf{H}}_{\mathbf{xx},t}^{\text{gp}}$ and

$$\Delta = -\mathbf{H}_{\mathbf{xy}}(\mathbf{x}_t, \mathbf{y}_t)[\mathbf{H}_{\mathbf{yy}}(\mathbf{x}_t, \mathbf{y}_t)^{-1}][\mathbf{H}_{\mathbf{xy}}(\mathbf{x}_t, \mathbf{y}_t)]^\top,$$

we conclude that

$$\left\|\mathbf{P}(\mathbf{x}_t, \mathbf{y}_t)^{1/2}[\mathbf{C}_t^{\text{gp}}]^{-1}\mathbf{P}(\mathbf{x}_t, \mathbf{y}_t)^{1/2} - \mathbf{I}\right\| \leq \frac{\eta}{\sqrt{m}} + \frac{\eta^2}{1-\eta}$$

holds with probability at least $1-\delta$. □

F.2 The Proof of Theorem 4.8

Using Lemma 4.6, we have

$$\left\|\mathbf{P}(\mathbf{x}_t, \mathbf{y}_t)^{1/2}[\mathbf{C}_t^{\text{gp}}]^{-1}\mathbf{P}(\mathbf{x}_t, \mathbf{y}_t)^{1/2} - \mathbf{I}\right\| \leq \frac{\eta}{\sqrt{m}} + \frac{\eta^2}{1-\eta},$$

holds with probability at least $1-\delta$. Since the update rule of **GIANT-PANDA** on the server can be written as

$$\begin{bmatrix}\mathbf{x}_{t+1}\\ \mathbf{y}_{t+1}\end{bmatrix} = \begin{bmatrix}\mathbf{x}_t\\ \mathbf{y}_t\end{bmatrix} - \begin{bmatrix}\tilde{\mathbf{H}}_{\mathbf{xx},t}^{\text{gp}} & \mathbf{H}_{\mathbf{xy}}(\mathbf{x}_t, \mathbf{y}_t)\\ \mathbf{H}_{\mathbf{xy}}(\mathbf{x}_t, \mathbf{y}_t)^\top & \mathbf{H}_{\mathbf{yy}}(\mathbf{x}_t, \mathbf{y}_t)\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{g}_{\mathbf{x}}(\mathbf{x}_t, \mathbf{y}_t)\\ \mathbf{g}_{\mathbf{y}}(\mathbf{x}_t, \mathbf{y}_t)\end{bmatrix},$$

Using the convergence results of PAN (Theorem 3.2) by taking $\eta_1 = \frac{\eta}{\sqrt{m}} + \frac{\eta^2}{1-\eta}$, we can directly obtain (18).

F.3 The Proof of Corollary 4.10

When the variable $\mathbf{y}$ vanishes, we have

$$\hat{\mathbf{P}}(\mathbf{x}) \overset{\text{def}}{=} \nabla^2 f(\mathbf{x}),$$

the corresponding measure can be written as

$$\hat{\gamma}(\mathbf{x}) \overset{\text{def}}{=} \left\|[\hat{\mathbf{P}}(\mathbf{x})]^{-1}\nabla f(\mathbf{x})\right\|,$$

which recovers the measure of Rodomanov and Nesterov (2021). Under the condition of Corollary 4.10, f is μ-strongly convex, L_2-Lipschitz continuous. We define $\hat{r} \overset{\text{def}}{=} \|\hat{\mathbf{P}}(\mathbf{x})^{1/2}(\mathbf{x}_+ - \mathbf{x})\|$, using the results in Rodomanov and Nesterov (2021), it holds that

- $\frac{1}{1+\frac{L_2\hat{r}}{2\mu^{3/2}}}\hat{\mathbf{P}}(\mathbf{x}) \preceq \hat{\mathbf{P}}(\mathbf{x}_+) \preceq \left(1 + \frac{L_2\hat{r}}{2\mu^{3/2}}\right)\hat{\mathbf{P}}(\mathbf{x})$,
- $\hat{r} \leq \hat{\lambda}(\mathbf{x})$.

Lemma F.1 *Under the same condition of Corollary 4.10, when we use the update rule*

$$\mathbf{x}_+ = \mathbf{x} - \mathbf{H}^{-1}\nabla f(\mathbf{x}),$$

where $\mathbf{H} \in \mathbb{R}^{n_\mathbf{x} \times n_\mathbf{x}}$ *is a positive definite matrix satisfies*

$$\left\| \mathbf{I} - \nabla^2 f(\mathbf{x})^{1/2} \mathbf{H}^{-1} \nabla^2 f(\mathbf{x})^{1/2} \right\| \leq \eta_1,$$

then it holds that

$$\hat{\lambda}(\mathbf{x}_+) \leq \eta_1 \hat{\lambda}(\mathbf{x}) + \frac{2L_2}{\mu^{3/2}} \hat{\lambda}(\mathbf{x})^2.$$

Now, we formally present the proof of Corollary 4.10.

Proof Replacing the factor $\sqrt{1 + \frac{3\kappa^2\kappa_2}{\mu} r}$ by $\sqrt{1 + \frac{L_2}{2\mu^{3/2}} \hat{r}}$ of (D7) in Theorem 3.2, we have

$$\hat{\lambda}(\mathbf{x}_+) \leq \eta_1 \sqrt{1 + \frac{L_2}{2\mu^{3/2}} \hat{r}} \hat{\lambda}(\mathbf{x}) + \frac{3L_2}{2\mu^{3/2}} \hat{r}^2, \tag{F18}$$

Using $\hat{r} \leq \hat{\lambda}(\mathbf{x})$, we have

$$\hat{\lambda}(\mathbf{x}_+) \leq \eta_1 \hat{\lambda}(\mathbf{x}) + \frac{L_2}{\mu^{3/2}} \hat{\lambda}(\mathbf{x})^2 + \frac{3L_2}{2\mu^{3/2}} \hat{\lambda}(\mathbf{x})^2 = \eta_1 \hat{\lambda}(\mathbf{x}) + \frac{2L_2}{\mu^{3/2}} \hat{\lambda}(\mathbf{x})^2.$$

□

Now we present the proof of Corollary 4.10 which is very similar to the proof of Theorem 4.8.

Proof Using the results of Lemma 4.6, we have

$$\left\| \mathbf{I} - [\nabla^2 f(\mathbf{x}_t)]^{1/2} \tilde{\mathbf{H}}_t^{-1} [\nabla^2 f(\mathbf{x}_t)]^{1/2} \right\| \leq \frac{\eta}{\sqrt{m}} + \frac{\eta^2}{1-\eta}, \tag{F19}$$

holds with probability at least $1 - \delta$ where $\tilde{\mathbf{H}}_t$ is defined by

$$\tilde{\mathbf{H}}_t \overset{\text{def}}{=} \left[\frac{1}{m} \sum_{i=1}^{m} [\tilde{\mathbf{H}}_{\mathbf{xx},t}^i]^{-1} \right]^{-1}.$$

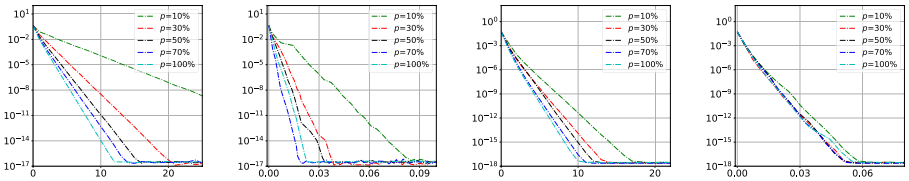
The update rule of GIANT can be written as

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \tilde{\mathbf{H}}_t^{-1} \nabla f(\mathbf{x}_t).$$

Using Lemma F.1 by taking $\eta_1 = \frac{\eta}{\sqrt{m}} + \frac{\eta^2}{1-\eta}$, we have

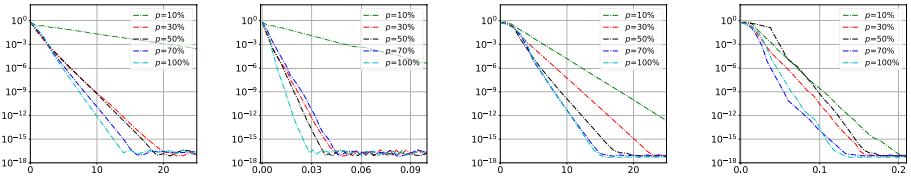
$$\hat{\lambda}(\mathbf{x}_{t+1}) \leq \left(\frac{\eta}{\sqrt{m}} + \frac{\eta^2}{1-\eta} \right) \hat{\lambda}(\mathbf{x}_t) + \frac{2L_2}{\mu^{3/2}} \hat{\lambda}(\mathbf{x}_t)$$

holds with probability at least $1 - \delta$. □



(a) a9a, #comm (b) a9a, time (c) w8a, #comm (d) w8a, time

Fig. 7 We demonstrate the communication rounds (#comm) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ and running time (second) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ for AUC maximization on datasets "a9a" and "w8a" with different sketch ratio p under the case $m = 128$



(a) adult, #comm (b) adult, time (c) law school, #comm (d) law school, time

Fig. 8 We demonstrate the communication rounds (#comm) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ and running time (second) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ for Fairness-aware machine learning on datasets "adult" and "law school" with different sketch ratio p under the case $m = 128$

Appendix G Additional experiments

This section provides additional experiments to validate our new proposed methods. In G.1, we present additional numerical results of different sketch ratios for `GIANT-PANDA`. In G.2, we study the impact of the sketch methods on the convergence behavior for `GIANT-PANDA`. In G.3, we compare `PAN` with existing state-of-the-art methods for single-agent optimization ($m = 1$).

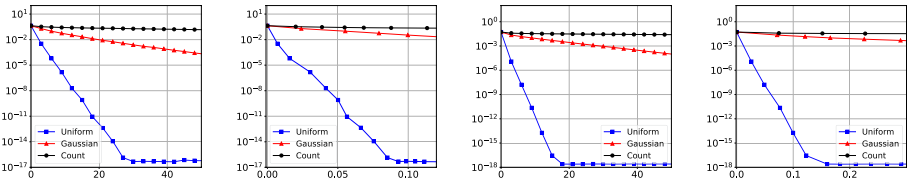
G.1 More study on the impact of the sketch ratio

We choose the sketch ratio from $p \in \{10\%, 30\%, 50\%, 70\%, 100\%\}$. We set the number of clients $m = 128$. We present the results for AUC maximization and Fairness-aware machine learning in Figs. 7 and 8 respectively. We observe similar behaviors as in Sect. 5.2.

G.2 Comparison of different sketch methods for `GIANT-PANDA`

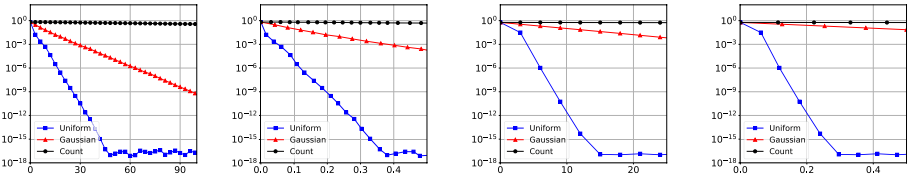
We validate the impact of using different sketch methods in `GIANT-PANDA`. Specifically, we choose uniform sketch, Gauss sketch (Johnson & Lindenstrauss, 1984), and count sketch (Clarkson & Woodruff, 2017) to construct the local approximate partial Hessian $\tilde{\mathbf{H}}^i_{\mathbf{xx},t}$ in (16) in `GIANT-PANDA`.

We present the behavior of `GIANT-PANDA` under $m = 8$ for AUC maximization and Fairness-aware machine learning in Figs. 9 and 10 respectively.



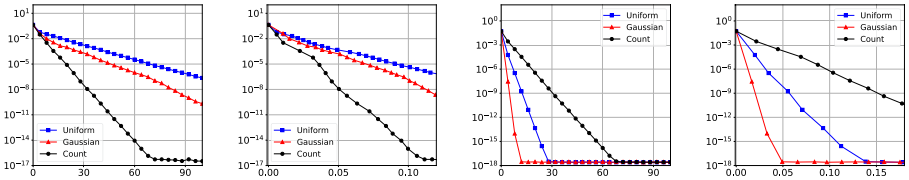
(a) a9a, #comm (b) a9a, time (c) w8a, #comm (d) w8a, time s

Fig. 9 We demonstrate the communication rounds (#comm) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ and running time (second) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ for AUC maximization under $m = 8$ on datasets "a9a" and "w8a" with different sketch methods



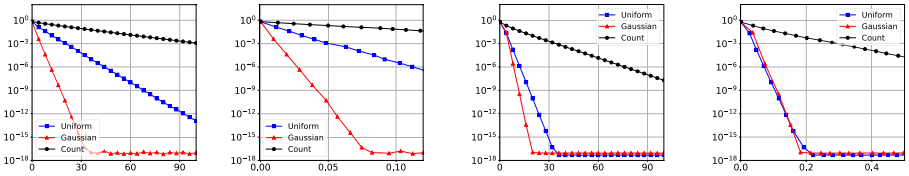
(a) adult, #comm (b) adult, time (c) law school, #comm (d) law school, time

Fig. 10 We demonstrate the communication rounds (#comm) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ and running time (second) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ for Fairness-aware machine learning under $m = 8$ on datasets "adult" and "law school" with different sketch methods



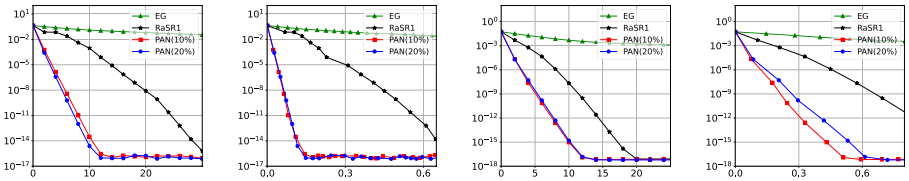
(a) a9a, #comm (b) a9a, time (c) w8a, #comm (d) w8a, time

Fig. 11 We demonstrate the communication rounds (#comm) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ and running time (second) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ for AUC maximization under $m = 128$ on datasets "a9a" and "w8a" with different sketch methods



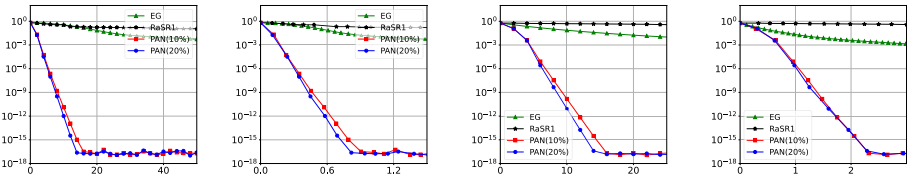
(a) adult, #comm (b) adult, time (c) law school, #comm (d) law school, time

Fig. 12 We demonstrate the communication rounds (#comm) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ and running time (second) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ for AUC maximization under $m = 128$ on datasets "adult" and "law school" with different sketch ratio p



(a) a9a, #comm (b) a9a, time (c) w8a, #comm (d) w8a, time

Fig. 13 We demonstrate the iteration rounds (#iter) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ and running time (second) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ for AUC maximization on datasets "a9a" and "w8a"



(a) adult, #comm (b) adult, time (c) law school, #comm (d) law school, time

Fig. 14 We demonstrate the iteration rounds (#iter) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ and running time (second) against $\|\nabla f(\mathbf{x}, \mathbf{y})\|_2$ for Fairness-aware machine learning on datasets "adult" and "law school"

We also present the behavior of GIANT-PANDA under $m = 128$ for AUC maximization and Fairness-aware machine learning in Figs. 11 and 12 respectively.

We observe that when $m = 8$, GIANT-PANDA with uniform sketch achieves the best behavior in terms of both communication rounds and running time (Figs. 9 and 10). This means when the local sample size s is relatively large, employing uniform sketch in GIANT-PANDA is good enough.

However, when the number of clients is as large as $m = 128$, the count sketch and Gauss sketch behave better than the uniform sketch (Figs. 11 and 12). This encourages us to choose more complicated sketch methods to improve the behavior of GIANT-PANDA when the local sample size is small.

G.3 Comparison of baselines on single-agent optimization

We compare partially approximate Newton (PAN) with existing state-of-the-art methods for single-agent minimax optimization. We adopt extra gradient (EG) and partial-quasi-Newton methods with SR1 update (Liu et al., 2022) (RaSR1) algorithms for comparison. We present the numerical results for AUC maximization and Fairness-aware machine learning in Figs. 13 and 14. The experiment results indicate that PAN with the sample ratio ($p' = |\mathcal{S}|/N$) at $p' = 10\%$ or $p' = 20\%$ outperforms the baselines significantly.

Author Contributions M.X. and C.L. wrote the main manuscript text, including numerical experiments and proofs. C.C. provided proofs of a key lemma. J.L and S.N. reviewed the manuscript and offered valuable suggestions and assistance. All authors reviewed and approved the final version of the manuscript.

Data availibility No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

Adil, D., Bullins, B., Jambulapati, A., & Sachdeva, S. (2022). Line search-free methods for higher-order smooth monotone variational inequalities. *arXiv preprint* arXiv:2205.06167

Basar, T., & Olsder, G. J. (1999). *Dynamic noncooperative game theory*. ser. Classics in Applied MathematicsSIAM.

Ben-Tal, A., & Nemirovski, A. (2002). Robust optimization-methodology and applications. *Mathematical Programming, 92*, 453–480.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology, 2*, 27–12727.

Chavdarova, T., Gidel, G., Fleuret, F., & Lacoste-Julien, S. (2019). Reducing noise in gan training with variance reduced extragradient. Advances in Neural Information Processing Systems, *32*.

Clarkson, K. L., & Woodruff, D. P. (2017). Low-rank approximation and regression in input sparsity time. *Journal of the ACM (JACM), 63*(6), 1–45.

Cortes, C., & Mohri, M. (2003). Auc optimization vs. error rate minimization. Advances in neural information processing systems, *16*.

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. (2018). Generative adversarial networks: An overview. *IEEE Signal Processing Magazine, 35*(1), 53–65.

Deng, Y., & Mahdavi, M. (2021). Local stochastic gradient descent ascent: Convergence analysis and communication efficiency. *International conference on artificial intelligence and statistics* (pp. 1387–1395). PMLR.

Facchinei, F. (2003). *Finite-dimensional variational inequalities and complementarity problems*. Springer.

Gao, R., & Kleywegt, A. (2022). Distributionally robust stochastic optimization with wasserstein distance. *Mathematics of Operations Research, 48*(2), 603–655.

Hsieh, Y.-G., Iutzeler, F., Malick, J., & Mertikopoulos, P. (2019). On the convergence of single-call stochastic extra-gradient methods. *Advances in Neural Information Processing Systems*, *32*.

Huang, K., & Zhang, S. (2022). An approximation-based regularized extra-gradient method for monotone variational inequalities. arXiv preprint arXiv:2210.04440

Huang, K., Zhang, J., & Zhang, S. (2022). Cubic regularized newton method for the saddle point models: A global and local convergence analysis. *Journal of Scientific Computing, 91*(2), 60.

Islamov, R., Qian, X., Hanzely, S., Safaryan, M., & Richtárik, P. (2022). Distributed Newton-type methods with communication compression and bernoulli aggregation. *Transactions on Machine Learning Research*

Johnson, W. B., & Lindenstrauss, J. (1984). Extensions of Lipschitz maps into a Hilbert space. *Contemporary Mathematics, 26*, 189–206.

Korpelevich, G. M. (1976). The extragradient method for finding saddle points and other problems. *Matecon, 12*, 747–756.

Lanckriet, G. R., Ghaoui, L. E., Bhattacharyya, C., & Jordan, M. I. (2002). A robust minimax approach to classification. *Journal of Machine Learning Research, 3*, 555–582.

Le Quy, T., Roy, A., Iosifidis, V., Zhang, W., & Ntoutsi, E. (2022). A survey on datasets for fairness-aware machine learning. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 12*(3), 1452.

Lin, T., & Jordan, M.I. (2022). Perseus: A simple high-order regularization method for variational inequalities. *arXiv preprint* arXiv:2205.03202

Lin, T., Jin, C., & Jordan, M. (2020). On gradient descent ascent for nonconvex-concave minimax problems. *International conference on machine learning* (pp. 6083–6093). PMLR.

Liu, C., & Luo, L. (2022). Regularized newton methods for monotone variational inequalities with holders continuous jacobians. *arXiv preprint* arXiv:2212.07824

Liu, C., Bi, S., Luo, L., & Lui, J. C. (2022). Partial-quasi-newton methods: Efficient algorithms for minimax optimization problems with unbalanced dimensionality. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 1031–1041).

Liu, C., Chen, L., Luo, L., & Lui, J. (2024). Communication efficient distributed newton method with fast convergence rates. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*.

Liu, C., & Luo, L. (2022). Quasi-newton methods for saddle point problems. *Advances in Neural Information Processing Systems, 35*, 3975–3987.

Liu, M., Zhang, W., Mroueh, Y., Cui, X., Ross, J., Yang, T., & Das, P. (2020). A decentralized parallel algorithm for training generative adversarial nets. *Advances in Neural Information Processing Systems, 33*, 11056–11070.

Lowd, D., & Meek, C. (2005). Adversarial learning. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining* (pp. 641–647).

Luo, L., Li, Y., & Chen, C. (2022). Finding second-order stationary points in nonconvex-strongly-concave minimax optimization. *Advances in Neural Information Processing Systems, 35*, 36667–36679.

Malitsky, Y. (2015). Projected reflected gradient methods for monotone variational inequalities. *SIAM Journal on Optimization, 25*(1), 502–520.

Meng, X., & Mahoney, M. W. (2013). Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *Proceedings of the forty-fifth annual ACM symposium on theory of computing* (pp. 91–100).

Mishchenko, K., Kovalev, D., Shulgin, E., Richtárik, P., & Malitsky, Y. (2020). Revisiting stochastic extragradient. In *International conference on artificial intelligence and statistics* (pp. 4573–4582). PMLR

Na, S., Dereziński, M., & Mahoney, M. W. (2023). Hessian averaging in stochastic newton methods achieves superlinear convergence. *Mathematical Programming, 201*(1–2), 473–520.

Nedić, A., & Ozdaglar, A. (2009). Subgradient methods for saddle-point problems. *Journal of Optimization Theory and Applications, 142*, 205–228.

, Nouiehed, M., Sanjabi, M., Huang, T., Lee, J. D., & Razaviyayn, M. (2019). Solving a class of non-convex min-max games using iterative first order methods. *Advances in Neural Information Processing Systems*, *32*.

Rodomanov, A., & Nesterov, Y. (2021). Greedy quasi-newton methods with explicit superlinear convergence. *SIAM Journal on Optimization, 31*(1), 785–811.

Roosta-Khorasani, F., & Mahoney, M. W. (2019). Sub-sampled newton methods. *Mathematical Programming, 174*, 293–326.

Shamir, O., Srebro, N., & Zhang, T. (2014). Communication-efficient distributed optimization using an approximate newton-type method. *International conference on machine learning* (pp. 1000–1008). PMLR.

Sun, Z., & Wei, E. (2022). A communication-efficient algorithm with linear convergence for federated minimax learning. *Advances in Neural Information Processing Systems, 35*, 6060–6073.

Tseng, P. (2000). A modified forward-backward splitting method for maximal monotone mappings. *SIAM Journal on Control and Optimization, 38*(2), 431–446.

Wang, S., Gittens, A., & Mahoney, M. W. (2017). Sketched ridge regression: Optimization perspective, statistical perspective, and model averaging. *International conference on machine learning* (pp. 3608–3616). PMLR.

Wang, S., Roosta, F., Xu, P., & Mahoney, M. W. (2018). Giant: Globally improved approximate newton method for distributed optimization. *Advances in Neural Information Processing Systems*, *31*.

Wang, Y., & Li, J. (2020). Improved algorithms for convex-concave minimax optimization. *Advances in Neural Information Processing Systems, 33*, 4800–4810.

Ye, H., He, C., & Chang, X. (2022). Accelerated distributed approximate newton method. *IEEE Transactions on Neural Networks and Learning Systems*

Ye, H., Luo, L., & Zhang, Z. (2020). Nesterov's acceleration for approximate newton. *The Journal of Machine Learning Research, 21*(1), 5627–5663.

Ye, H., Luo, L., & Zhang, Z. (2021). Approximate newton methods. *The Journal of Machine Learning Research, 22*(1), 3067–3107.

Ying, Y., Wen, L., & Lyu, S. (2016). *Stochastic online AUC maximization*. NIPS.

Zhang, S., Choudhury, S., Stich, S. U., & Loizou, N. (2024). Communication-efficient gradient descent-accent methods for distributed variational inequalities: Unified analysis and local updates. In *The twelfth international conference on learning representations*

Zhang, B. H., Lemoine, B., & Mitchell, M. (2018). Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM conference on AI, ethics, and society* (pp. 335–340).