Automatica 183 (2026) 112531

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief Paper

Federated Cubic Regularized Newton Learning with sparsification-amplified differential privacy ☆

Wei Huo [a], Changxin Liu [b,c], Kemi Ding [d,*], Karl Henrik Johansson [b,c], Ling Shi [a]

[a] Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong
[b] Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, SE-10044 Stockholm, Sweden
[c] Digital Futures, SE-10044 Stockholm, Sweden
[d] School of System Design and Intelligent Manufacturing, Southern University of Science and Technology, Shenzhen, 518055, China

ARTICLE INFO

Article history:
Received 8 August 2024
Received in revised form 11 March 2025
Accepted 25 June 2025
Available online 2 September 2025

Keywords:
Federated learning
Cubic regularized Newton method
Differential privacy
Communication sparsification

ABSTRACT

This paper explores the cubic-regularized Newton method within a federated learning framework while addressing two major concerns: privacy leakage and communication bottlenecks. We propose the Differentially Private Federated Cubic Regularized Newton (DP-FCRN) algorithm, which leverages second-order techniques to achieve lower iteration complexity than first-order methods. We incorporate noise perturbation during local computations to ensure privacy. Furthermore, we employ sparsification in uplink transmission, which not only reduces the communication costs but also amplifies the privacy guarantee. Specifically, this approach reduces the necessary noise intensity without compromising privacy protection. We analyze the convergence properties of our algorithm and establish the privacy guarantee. Finally, we validate the effectiveness of the proposed algorithm through experiments on a benchmark dataset.

1. Introduction

As big data grows and privacy concerns rise, conventional centralized methods for optimizing model parameters encounter substantial challenges. Federated learning (FL) has emerged as a promising approach, allowing multiple devices to collaboratively optimize a shared model under the coordination of the central server, without sharing local data. FL has found applications in fields such as robotics (Yuan et al., 2024) and autonomous driving (Nguyen et al., 2022). The prevailing FL algorithm is Fed-SGD, based on stochastic gradient descent (SGD) (Lowy & Razaviyayn, 2023). In this approach, each client trains a local model using SGD and uploads the gradient to the central server, which averages the gradients and updates the model. However, such first-order methods suffer from slow convergence, which can hinder applications requiring fast processing, such as autonomous vehicles, where timely and accurate predictions are critical. Newton's technique, a second-order method, offers faster convergence, but its integration into FL represents challenges. The key obstacle is the non-linear nature of aggregating solutions from local optimization problems for second-order approximations, in contrast to the simpler gradient aggregation used in first-order methods. This complexity is evident in recent algorithms like GIANT (Maritan et al., 2023).

While aggregating local Hessians is theoretically feasible, uploading Hessian matrices at each round incurs significant communication costs. Even without transmitting matrices, communication efficiency remains a critical bottleneck in FL. For instance, mobile devices, which are commonly used as clients, often have limited communication bandwidth. Traditional first-order optimization methods improve communication efficiency through techniques such as compression (Richtárik et al., 2021) and event trigger (Huo, Tsang, et al., 2024). Recent second-order methods, like federated Newton learning (Safaryan et al., 2021), incorporate contractive compression and partial participation to reduce communication costs. Building on this, Zhang et al. (2024) further reduced communication rounds with lazy aggregation and enhanced convergence using cubic and gradient-regularized Newton methods. Liu et al. (2023) developed a distributed Newton's method with improved communication efficiency and achieved super-linear convergence. Dal Fabbro et al. (2024) presented a Newton-type algorithm to accelerate FL while addressing communication constraints.

In addition to slow convergence and communication costs, privacy leakage is another significant concern in FL. Simply storing data locally on clients does not guarantee adequate privacy.

☆ The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Luca Schenato under the direction of Editor Christos G. Cassandras.

* Corresponding author.
E-mail addresses: whuoaa@connect.ust.hk (W. Huo), changxin@kth.se (C. Liu), dingkm@sustech.edu.cn (K. Ding), kallej@kth.se (K.H. Johansson), eesling@ust.hk (L. Shi).

https://doi.org/10.1016/j.automatica.2025.112531

Recent inference attacks (Liu et al., 2022; Zhu, Liu & Han, 2019) show that sharing local model updates or gradients between clients and the server can result in privacy breaches. Sharing second-order information can also expose sensitive client data. For instance, Yin et al. (2014) showed that eigenvalues of the Hessian matrix can leak critical information from input images. In algorithms transmitting compressed Hessians (Safaryan et al., 2021; Zhang et al., 2024), if compression does not significantly alter the eigenvalues, sensitive data may still be exposed. Therefore, privacy preservation is vital in second-order FL. Differential privacy (DP), introduced by Dwork (2006), has been the standard framework for privacy preservation due to its effectiveness in data analysis tasks. For example, Huang et al. (2024) utilized Laplacian noise for differentially private distributed gradient tracking. Wang et al. (2023) explored the relationship between differential initial-value privacy and observability in linear dynamical systems. Wang et al. (2024) proposed a distributed shuffling mechanism based on the Paillier cryptosystem to enhance the accuracy-privacy trade-off in DP-preserving average consensus algorithms. In differentially private Fed-SGD, the gradient is typically augmented with Gaussian noise to achieve DP (Lowy & Razaviyayn, 2023). Due to the composition of DP, the required noise level is influenced by the number of iterations. Recently, Ganesh et al. (2024) devised a second-order, differentially private optimization method that achieves (ε, δ)-DP with utility loss $O(d/\varepsilon^2)$ for d-dimensional model, which is optimal, i.e., the best achievable, for differentially private optimization (Bassily et al., 2014; Kairouz et al., 2021). However, this method is restricted to centralized settings. Ensuring DP for second-order optimization in FL remains a challenge, requiring the integration of noise perturbation with communication-efficient methods and addressing trade-offs between privacy, accuracy, and communication across clients.

Motivated by the above observations, we aim to investigate federated Newton learning while jointly considering DP and communication issues in the algorithm design. Prior research predominantly considers DP and communication efficiency as separate entities (Li et al., 2022; Zhang et al., 2020). While some research has explored the joint trade-off among privacy, accuracy, and communication (Chen et al., 2021; Mohammadi et al., 2021), they tackled the communication and privacy in a cascaded fashion, i.e., their communication schemes do not directly impact privacy preservation. In contrast, our study investigates the interplay between communication and privacy guarantees. Although some recent studies have employed compression in uplink transmission to improve the trade-off (Chen et al., 2024; Hu et al., 2023), these approaches are limited to first-order learning with slow convergence. Besides, Chen et al. (2024) exclusively addressed central DP, which is less robust compared to privacy mechanisms at the client level. Specifically, we propose that each local machine uses a cubic regularized Newton method for model updates, incorporating noise perturbation during local computation. In FL, where local model updates are typically sparse, we combine perturbation with random sparsification to enhance privacy. Sparsification reduces the sensitivity of updates to raw data by zeroing out some coordinates, thereby lowering privacy loss during communication. We show that the noise intensity required for DP is proportional to the number of transmitted coordinates, meaning improved communication efficiency can reduce the noise without compromising privacy. Furthermore, we illustrate that our algorithm's iteration complexity exhibits an exponential improvement compared to first-order methods, further reducing noise intensity and enhancing the trade-off between privacy and convergence. Comparison of some related works with ours is shown in Table 1.

Our main contributions are summarized as follows:

(1) We develop the DP-FCRN algorithm (Algorithm 1), which leverages second-order Newton methods for faster convergence. We exploit noise perturbation in local computations to guarantee privacy preservation (Algorithm 2) and use sparsification to improve communication efficiency. Unlike previous studies that treat DP and communication burden separately (Chen et al., 2021; Li et al., 2022; Mohammadi et al., 2021; Zhang et al., 2020), we use the inherent characteristic of sparsification to simultaneously enhance both privacy and communication efficiency.
(2) We analyze the impact of sparsification on the privacy-accuracy trade-off. Specifically, we show that sparsification reduces the required noise intensity (Theorem 1), allowing for lower Gaussian noise while maintaining privacy. We also conduct a non-asymptotic analysis of utility loss and complexity (Theorem 2), demonstrating that the utility loss is optimal and that the iteration complexity improves over first-order methods.
(3) We evaluate our method on the benchmark dataset. Experiment results show that our algorithm improves the model accuracy, and at the same time saves communication costs compared to Fed-SGD under the same DP guarantee.

The remainder of the paper is organized as follows. Preliminaries and the problem formulation are provided in Section 2. In Section 3, a federated cubic regularized Newton learning algorithm with sparsification-amplified DP is proposed. Then, details on the DP analysis are shown in Section 4 and the convergence analysis is presented in Section 5. In Section 6, numerical simulations are presented to illustrate the obtained results. Finally, the conclusion and future research directions are discussed in Section 7.

Notations: Let $\mathbb{R}^p$ and $\mathbb{R}^{p\times q}$ represent the set of p-dimensional vectors and $p \times q$-dimensional matrices, respectively. $I_p \in \mathbb{R}^{p\times p}$ represents a $p\times p$-dimensional identity matrix. With any positive integer, we denote $[d]$ as the set of integers $\{1, 2, \ldots, d\}$. We use $[\cdot]_j$ to denote the jth coordinate of a vector and jth row of a matrix. Let c represent a set of integers, and we denote $[X]_c$ as a vector containing elements $[X]_j$ for $j \in c$ if X is a vector, and as a matrix with row vectors $[X]_j$ for $j \in c$ if X is a matrix. Let $\|\cdot\|$ be the ℓ_2-norm vector norm. For a convex and closed subset $\mathcal{X} \subseteq \mathbb{R}^d$, let $\Pi_{\mathcal{X}} : \mathbb{R}^d \to \mathcal{X}$ be the Euclidean projection operator, given by $\Pi_{\mathcal{X}}(x) = \arg\min_{y\in\mathcal{X}} \|y - x\|$. We use $\mathbb{P}\{\mathcal{A}\}$ to represent the probability of an event $\mathcal{A}$, and $\mathbb{E}[x]$ to be the expected value of a random variable x.

The notation $O(\cdot)$ is used to describe the asymptotic upper bound. Mathematically, $h(n) = O(g(n))$ if there exist positive constants C and n_0 such that $0 \le h(n) \le Cg(n)$ for all $n \ge n_0$. Similarly, the notation $\Omega(\cdot)$ provides the asymptotic lower bound, i.e., $h(n) = \Omega(g(n))$ if there exist positive constants C and n_0 such that $0 \le Cg(n) \le h(n)$ for all $n \ge n_0$. The notation $\tilde{O}(\cdot)$ is a variant of $O(\cdot)$ that ignores logarithmic factors, that is, $h(n) = \tilde{O}(g(n))$ is equivalent to $h(n) = O\left(g(n)log^k n\right)$ for some $k > 0$. The notation $\Theta(\cdot)$ is defined as the tightest bound, i.e., $h(n)$ is said to be $\Theta(g(n))$ if $h(n) = O(g(n))$ and $h(n) = \Omega(g(n))$.

2. Preliminaries and problem formulation

This section introduces the fundamental setup of FL along with key concepts on Newton's methods with cubic regularization and DP. Subsequently, we outline the considered problem.

Table 1
Comparison of some existing works with ours.

Work	Efficient communication	Optimization	DP	Impact of efficient communication on DP
Bassily et al. (2014), Kairouz et al. (2021)	×	First-order	Client-DP	×
Chen et al. (2021, 2024)	Compressed vectors	First-order	Central DP	Increased compression did not improve privacy and reduced accuracy
Safaryan et al. (2021)	Compressed matrices	Second-order	×	×
Zhang et al. (2024)	Compressed matrices	Second-order	×	×
Ours	Compressed vectors	Second-order	Client-DP	Increased compression enhances privacy and improves accuracy

2.1. Basic setup

We consider a federated setting with n clients and a central server. Each client $i \in [n]$ possesses a private local dataset $\zeta_i = \{\zeta_i^{(1)}, \ldots, \zeta_i^{(m)}\}$ containing a finite set of m data samples. Moreover, each client has a private local cost function $f_i(x) = \frac{1}{m}\sum_{j=1}^{m} l(x, \zeta_i^{(j)})$, where $l(x, \zeta_i^{(j)})$ is the loss of model x over the data instance $\zeta_i^{(j)}$ for $j \in [m]$. With the coordination of the central server, all clients aim to train a global model x by solving the following problem while maintaining their data locally:

$$\min_{x\in\mathcal{X}} f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{1}$$

where $\mathcal{X} \subseteq \mathbb{R}^d$ is a convex and closed box constraint. Specifically, the model training process takes place locally on each client, and only the updates are sent to the server for aggregation and global updates. The optimal model parameter is defined as $x^* = \arg\min_{x\in\mathcal{X}} f(x)$.

Assumption 1. The optimization problem (1) satisfies the following conditions:

(i) The parameter set $\mathcal{X}$ has finite diameter D.
(ii) The loss function $l(\cdot, \zeta)$ is L_0-Lipschitz, L_1-smooth, and has an L_2-Lipschitz Hessian for any ζ over $\mathcal{X}$.
(iii) The loss function $l(\cdot, \zeta)$ is μ-strongly convex for any ζ over $\mathcal{X}$.

From Assumption 1, we infer that also $f_i(\cdot)$ and $f(\cdot)$ are μ-strongly convex, L_0-Lipschitz, L_1-smooth, and have L_2-Lipschitz Hessian over $\mathcal{X}$.

2.2. Newton methods with cubic regularization

Newton methods (Boyd & Vandenberghe, 2004) iteratively minimize a quadratic approximation of the function $f(\cdot)$ as

$$x_{t+1} = \arg\min_{x\in\mathcal{X}} \Big\{ f(x_t) + \langle \nabla f(x_t), x - x_t\rangle + \frac{1}{2}\left\langle \nabla^2 f(x_t)(x - x_t), x - x_t\right\rangle \Big\}. \tag{2}$$

The Hessian matrix $\nabla^2 f(x_t)$ provides curvature information about $f(\cdot)$ at x_t. Newton's methods significantly improve the convergence speed of gradient descent by automatically adjusting the step size along each dimension based on the local curvature at each step.

The cubic regularized Newton method, initially introduced by Nesterov and Polyak (2006), incorporates a second-order Taylor expansion with a cubic regularization term. In particular, the update is

$$x_{t+1} = \arg\min_{x\in\mathcal{X}} \Big\{ f(x_t) + \langle \nabla f(x_t), x - x_t\rangle + \frac{1}{2}\left\langle \nabla^2 f(x_t)(x - x_t), x - x_t\right\rangle + \frac{L_2}{6}\|x - x_t\|^3 \Big\}, \tag{3}$$

where L_2 is the Lipschitz Hessian constant in Assumption 2. The cubic upper bound of $f(x_t)$ in (3) serves as a universal upper bound regardless of the specific characteristics of the objective function. However, the function to minimize in each step of (3) does not have a closed-form solution and it is limited to a centralized single node setting, which our algorithm addresses in a federated setting as discussed in Section 3. Cubic regularization ensures globally convergent second-order optimization with adaptive step control, avoiding the instability and exact Hessian inversion requirements of Newton's methods while maintaining efficiency in non-convex or distributed settings.

2.3. Threat model and DP

Local datasets contain sensitive user information. If problem (1) is addressed in an insecure environment, information leakage could jeopardize both personal and property privacy. This paper considers the following adversary model:

Definition 1 (*Adversary Model*)**.** Adversaries can be

(i) an honest-but-curious central server that follows the protocol but may attempt to infer private client information from the received messages.
(ii) colluding clients or clients collaborating with the central server to deduce private information about other legal clients.
(iii) an outside eavesdropper who intercepts all transmitted messages without actively destroying communication.

Our adversary model is much stronger than some works that require a trusted third party (Chen et al., 2024; Hao et al., 2019).

DP is a widely used concept for quantifying privacy risk. It ensures that the presence or absence of any individual in a dataset cannot be inferred from the output of a randomized algorithm $\mathcal{A}$ (Dwork, 2006). Below, we present the formal definition of DP within the context of FL.

Definition 2 ((ε, δ)-DP) The algorithm $\mathcal{A}$ is called (ε, δ)-DP, if for any neighboring dataset pair $\zeta = \cup_{i\in[n]}\zeta_i$ and $\zeta' = \cup_{i\in[n]}\zeta_i'$ that differ in one data instance and every measurable $\mathcal{O} \subseteq Range(\mathcal{A})$,[1]

[1] Range($\mathcal{A}$) denotes the set of all possible observation sequences under the algorithm $\mathcal{A}$.

the output distribution satisfies

$$\mathbb{P}\{\mathcal{A}(\zeta) \in \mathcal{O}\} \leq e^{\varepsilon}\mathbb{P}\{\mathcal{A}(\zeta') \in \mathcal{O}\} + \delta, \tag{4}$$

where the probability $\mathbb{P}\{\cdot\}$ is taken over the randomness of $\mathcal{A}$.

Definition 2 states that the output distributions of neighboring datasets exhibit small variation. The factor ε in (4) represents the upper bound of privacy loss by algorithm $\mathcal{A}$, and δ denotes the probability of breaking this bound. Therefore, a smaller ε corresponds to a stronger privacy guarantee. Both Laplace and Gaussian noise can achieve DP. However, Gaussian noise, with its more concentrated distribution and superior composition properties, offers a better balance between privacy and accuracy. Therefore, we focus on the Gaussian mechanism in this work.

Lemma 1 (*Gaussian Mechanism Balle & Wang, 2018*)**.** *A Gaussian mechanism $\mathcal{G}$ for a vector-valued computation $r : \zeta \to \mathbb{R}^d$ is obtained by computing the function r on the input data $\zeta_i \in \zeta$ and then adding random Gaussian noise perturbation $v \sim \mathcal{N}(0, \sigma^2 I_d)$ to the output, i.e,*

$$\mathcal{G} = r(\zeta) + v.$$

The Gaussian mechanism $\mathcal{G}$ is $\left(\frac{\sqrt{2\log(1.25/\delta)}\Delta}{\sigma}, \delta\right)$-DP for any neighboring dataset ζ and ζ', where Δ denotes the sensitivity of r, i.e., $\Delta = \sup_{\zeta,\zeta'} \|r(\zeta) - r(\zeta')\|$.

Lemma 1 indicates that achieving (ε, δ)-DP requires adjusting the noise intensity based on the privacy guarantee ε and δ, as well as the sensitivity Δ.

2.4. Problem statement

This paper aims to answer the following questions:

(a) How can we develop a cubic regularized Newton algorithm for solving (1) in a federated setting?
(b) Can we explore the sparsification scheme to reduce communication costs while amplifying the privacy guarantee, i.e., achieving a smaller ε given σ or requiring a smaller σ given ε?
(c) What level of noise intensity, i.e., σ, is necessary to attain (ε, δ)-DP in the proposed algorithm?
(d) Is it possible to attain the best achievable utility loss under DP, i.e., $f(x_T) - f(x^*) = O(d/\varepsilon^2)$ with the output x_T? If achievable, what is the iteration complexity for achieving this optimal utility loss?

3. Main algorithm

In this section, we present Algorithms 1 and 2 to answer problems (a) and (b) in Section 2.4.

In general, there are two approaches for integrating sparsification and privacy in FL: (1) perturb first, then sparsify, and (2) sparsify first, then perturb. The first approach is direct and adaptable since sparsification preserves DP and integrates smoothly with all current privacy mechanisms. However, in the second approach, perturbation may compromise the communication savings achieved through sparsification. Furthermore, empirical observations suggest that the first approach outperforms the second one in some scenarios (Ding et al., 2021). Therefore, we adopt the first approach in this study.

As shown in Algorithm 1, during iteration t, the server broadcasts the parameter x_t to the clients. Then, client i randomly samples a data instance $\zeta_{i,t} \in \zeta_i$, estimates the local gradient $\hat{g}_{i,t} = \nabla l(x_t, \zeta_{i,t})$ and the local Hessian $\hat{H}_{i,t} = \nabla^2 l(x_t, \zeta_{i,t})$ using its local data to minimize a local cubic-regularized upper bound of its loss function, and then does the following update

$$x_{i,t+1} = \arg\min_{x\in\mathcal{X}} \left\{ f_i(x_t) + \langle \hat{g}_{i,t}, x - x_t\rangle + \frac{1}{2}\left\langle \hat{H}_{i,t}(x - x_t), x - x_t\right\rangle + \frac{L_2}{6}\|x - x_t\|^3 \right\}. \tag{5}$$

As there is no closed form for optimal solution to (5), the client instead employs the gradient descent method to compute $x_{i,t+1}$. To privately minimize the local cubic upper bound, Gaussian noise is added to perturb the gradient. This local solver utilizing the Gaussian mechanism is denoted GMSolver and is detailed in Algorithm 2.

Following the update of the local model parameter, each client uploads its model update $x_{i,t+1} - x_t$ to the server. To address the communication challenges in uplink transmissions, the random-k sparsifier is employed to reduce the size of the transmitted message by a factor of k/d (Li & Richtárik, 2021):

Definition 3 (*Random-k Sparsification*)**.** For $x \in \mathbb{R}^d$ and a parameter $k \in [d]$, the random-k sparsification operator is

$$\mathcal{S}(x) := \frac{d}{k}(\xi_k \odot x),$$

where $\xi_k \in \{0, 1\}^d$ is a uniformly random binary vector with k nonzero entries, i.e., $\|\xi_k\|_0 = k$ and $\odot$ represents the element-wise Hadamard product.

Integrating private GMSolver and random-k sparsification, the proposed algorithm simultaneously addresses privacy preservation and communication efficiency as depicted in Algorithm 1. A scaling factor $\alpha > 0$ is introduced for convergence analysis.

Algorithm 1 DP-FCRN

Input: Clients' data $\zeta_1, \ldots, \zeta_n$, sparsification parameter k, DP parameters (ε, δ), and step size α.
Initialization: Model parameter x_0.
for $t = 0, 1, \ldots, T-1$ **do**
  ▶ **Server broadcasts**
  Broadcast x_t to all clients
  ▶ **Clients update and upload**
  for each client $i \in [n]$ in parallel **do**
    Sample $\zeta_{i,t}$ uniformly from $\{\zeta_i^{(1)}, \ldots, \zeta_i^{(m)}\}$ and compute the local estimate gradient $\hat{g}_{i,t} = \nabla l(x_t, \zeta_{i,t})$ and the local estimate Hessian $\hat{H}_{i,t} = \nabla^2 l(x_t, \zeta_{i,t})$
    $x_{i,t+1} = \text{GMSolver}(x_t, \hat{g}_{i,t}, \hat{H}_{i,t}, \tau, \sigma)$
    $y_{i,t} \leftarrow \alpha(x_{i,t+1} - x_t)$ and upload $\mathcal{S}(y_{i,t})$ to the server
  end for
  ▶ **Server updates**
  $x_{t+1} = x_t + \frac{1}{n}\sum_{i\in\mathcal{I}_t} \mathcal{S}(y_{i,t})$
end for

Algorithm 2 GMSolver

Input: Initialization θ_0, gradient g, Hessian H, the number of iterations τ, and the noise parameter σ.
for $s = 0, 1, \ldots, \tau - 1$ **do**
  $\eta_s = \frac{2}{\mu(s+2)}$
  $\text{grad}_s = g + H(\theta_s - \theta_0) + \frac{L_2}{2}\|\theta_s - \theta_0\|(\theta_s - \theta_0)$
  $\theta_{s+1} = \Pi_{\mathcal{X}}[\theta_s - \eta_s(\text{grad}_s + b_s)]$, where $b_s \sim N(0, \sigma^2 I_d)$
end for
Return $\sum_{s=0}^{\tau-1} \frac{2(s+1)}{\tau(\tau+1)}\theta_s$

Remark 1. As pointed out by Lacoste-Julien et al. (2012), the output of Algorithm 2 can be computed online. Specifically, setting $z_0 = \theta_0$, and recursively defining $z_s = \rho_s\theta_s + (1-\rho_s)z_{s-1}$ for $s \geq 1$, with $\rho_s = \frac{2}{s+1}$. It is a straightforward calculation to check that $z_\tau = \sum_{s=0}^{\tau-1} \frac{2s}{\tau(\tau+1)}\theta_s$.

Remark 2. Solving (3) directly requires significant computational resources. To improve efficiency, we use parallel cooperative solving across multiple clients. Since (5) lacks a closed-form solution, we propose a local training approach for its resolution. Privacy is preserved through noise perturbation during local training, while sparsification in uplink transmission reduces communication costs and further enhances privacy protection. Unlike existing methods that transmit compressed Hessian matrices (Safaryan et al., 2021; Zhang et al., 2024), we transmit compressed vectors to further minimize communication overhead.

4. Privacy analysis

In this section, we prove the privacy guarantee provided by Algorithm 1. To facilitate privacy analysis, we make the following assumption.

Assumption 2. For any data sample $\zeta_i^{(j)} \in \zeta_i$ and $h \in [d]$, we have

$$\left|\left[\nabla l(x, \zeta_i^{(j)})\right]_h\right| \leq \frac{L_0}{\sqrt{d}}, \quad \left\|\left[\nabla^2 l(x, \zeta_i^{(j)})\right]_h\right\| \leq \frac{L_1}{\sqrt{d}}$$

for any $x, v \in \mathcal{X}$ and $i \in [n]$.

Assumption 2 characterizes the sensitivity of each coordinate of the gradient $\nabla l(x, \zeta_i^{(j)})$ and each row of the Hessian $\nabla^2 l(x, \zeta_i^{(j)})$. This assumption is crucial for analyzing the interaction between element selection via sparsification and privacy amplification, as discussed in Chen et al. (2024), Hu et al. (2023). It implies that $\left\|\nabla l(x, \zeta_i^{(j)})\right\| \leq L_0$ and $\left\|\nabla^2 l(x, \zeta_i^{(j)})\right\|_2 \leq \left\|\nabla^2 l(x, \zeta_i^{(j)})\right\|_F \leq L_1$, which holds under the assumption of a bounded parameter set.

To analyze the interplay between the sparsification and privacy, let c_i^t denote the randomly selected coordinate set for client i at round t, i.e., $\mathcal{S}(\cdot) = \frac{d}{k}[\cdot]_{c_i^t}$.

An important observation is that only the values in c_i^t are transmitted to the central server, i.e.,

$$\mathcal{S}(y_{i,t}) = \frac{d}{k}\left[\alpha(x_{i,t+1} - x_{i,t})\right]_{c_i^t} = \frac{\alpha d}{k}\left([x_{i,t+1}]_{c_i^t} - [x_{i,t}]_{c_i^t}\right).$$

The gradient update information is contained in $[x_{i,t+1}]_{c_i^t}$ and

$$[x_{i,t+1}]_{c_i^t} = \left[\sum_{s=0}^{\tau-1} \frac{2(s+1)}{\tau(\tau+1)}\theta_i^{t,s}\right]_{c_i^t} = \sum_{s=0}^{\tau-1} \frac{2(s+1)}{\tau(\tau+1)}[\theta_i^{t,s}]_{c_i^t},$$

where $\theta_i^{t,s}$ denotes the optimization variable used by client i at iteration s in Algorithm 2 and the communication round t in Algorithm 1. Based on step 4 in the GMSolver, we have

$$[\theta_i^{t,s+1}]_{c_i^t} = \left[\Pi_{\mathcal{X}}\left[\theta_i^{t,s} - \eta_s(\text{grad}_i^{t,s} + b_i^{t,s})\right]\right]_{c_i^t},$$

where $\text{grad}_i^{t,s}$ and $b_i^{t,s}$ are the gradient and noise used by client i at iteration s in GMSolver and the communication round t in Algorithm 1, respectively. Since projection into a box constraint does not influence the set of selected coordinators c_i^t, what matters in local computation is

$$\left[\theta_i^{t,s} - \eta_s(\text{grad}_i^{t,s} + b_i^{t,s})\right]_{c_i^t} = [\theta_i^{t,s}]_{c_i^t} - \eta_s[\text{grad}_i^{t,s} + b_i^{t,s}]_{c_i^t}.$$

According to the above analysis, we conclude that the crucial aspect of privacy protection lies in the sparsified noisy gradient update, which can be expressed as

$$[\text{grad}_i^{t,s} + b_i^{t,s}]_{c_i^t} = [\text{grad}_i^{t,s}]_{c_i^t} + [b_i^{t,s}]_{c_i^t}.$$

We observe that the sparsification makes Gaussian noises only perturb the values at coordinates within c_i^t. If noise is added only at the selected coordinates, the level of privacy remains the same. In other words, we ensure the same privacy level even when incorporating a diminished amount of additional noise, thereby enhancing the optimization accuracy. Subsequently, we only need to analyze the privacy budget of $[\text{grad}_i^{t,s}]_{c_i^t}$ after adding noise $[b_i^{t,s}]_{c_i^t}$.

For client i, considering any two neighboring dataset ζ_i and ζ_i' of the same size m but with only one data sample different (e.g., $\zeta_i^{j_0}$ and $\zeta_i^{j_0\prime}$). Denote Δ as the ℓ_2-sensitivity of $[\text{grad}_i^{t,s}]_{c_i^t}$, and we have

$$\begin{aligned}
&\Delta^2 \\
=&\max_{\zeta,\zeta'}\Big\| \left[\hat{g}_{i,t}\right]_{c_i^t} - \left[\hat{g}'_{i,t}\right]_{c_i^t} + \left[\hat{H}_{i,t}(\theta_i^{t,s} - x_t)\right]_{c_i^t} \\
&\qquad - \left[\hat{H}'_{i,t}(\theta_i^{t,s} - x_t)\right]_{c_i^t}\Big\|^2 \\
=&\max_{\zeta,\zeta'}\Big\| \left[\nabla l(x_t, \zeta_i^{j_0}) - \nabla l(x_t, \zeta_i^{j_0\prime})\right]_{c_i^t} \\
&\qquad + \left[(\nabla^2 l(x_t, \zeta_i^{j_0}) - \nabla^2 l(x_t, \zeta_i^{j_0\prime}))(\theta_i^{t,s} - x_t)\right]_{c_i^t}\Big\|^2 \\
\leq&\frac{4k(L_0 + L_1 D)^2}{d},
\end{aligned} \tag{6}$$

where the last inequality holds from

$$\begin{aligned}
&\Big\| \left[\nabla l(x_t, \zeta_i^{j_0}) - \nabla l(x_t, \zeta_i^{j_0\prime})\right]_{c_i^t} \\
&\qquad + \left[(\nabla^2 l(x_t, \zeta_i^{j_0}) - \nabla^2 l(x_t, \zeta_i^{j_0\prime}))(\theta_i^{t,s} - x_t)\right]_{c_i^t}\Big\| \\
\leq&\Big\| \left[\nabla l(x_t, \zeta_i^{j_0}) - \nabla l(x_t, \zeta_i^{j_0\prime})\right]_{c_i^t}\Big\| \\
&+ \Big\| \left[\nabla^2 l(x_t, \zeta_i^{j_0}) - \nabla^2 l(x_t, \zeta_i^{j_0\prime})\right]_{c_i^t}\left[\theta_i^{t,s} - x_t\right]_{c_i^t}\Big\| \\
\leq&\frac{2\sqrt{k}L_0}{\sqrt{d}} + \frac{2\sqrt{k}L_1 D}{\sqrt{d}}.
\end{aligned}$$

Lemma 1 indicates that the noise intensity required to achieve (ε, δ)-DP relies on the sensitivity. From (6), sparsification reduces the conventional sensitivity $2(L_0 + L_1 D)$ by a factor of $\sqrt{k/d}$, thereby decreasing sensitivity and reducing the required noise intensity. For each client's sensitive local dataset ζ_i, $\forall i \in [n]$, if we treat DP-FCRN as the algorithm $\mathcal{A}$ defined in Definition 2, the worst-case observation by the attacker $\mathcal{A}(\zeta_i) = \{\mathcal{S}(y_{i,t}) | 0 \leq t \leq T\}$. Theorem 1 states a sufficient condition for achieving (ε, δ)-DP based on the reduced sensitivity resulting from sparsification.

Theorem 1. *Suppose Assumption 1 holds, and the random-k sparsifier with $k \leq d$ is used in Algorithm 1. Given m, τ, $\varepsilon \in (0, 1]$, and $\delta_0 \in (0, 1]$, if the noise variance*

$$\sigma^2 \geq \frac{160\tau T k \log(1.25/\delta_0)(L_0 + L_1 D)^2}{\varepsilon^2 m^2 d} \tag{7}$$

and $T \geq \frac{\varepsilon^2}{4\tau}$, then DP-FCRN is (ε, δ)-DP for ζ_i, $\forall i \in [n]$, with some constant $\delta \in (0, 1]$. Specifically, for any output set of DP-FCRN, $\mathcal{A}(\zeta_i)$, we have

$$\mathbb{P}\{\mathcal{A}(\zeta_i) \in \mathcal{O}\} \leq e^{\varepsilon}\mathbb{P}\{\mathcal{A}(\zeta_i') \in \mathcal{O}\} + \delta. \tag{8}$$

Proof. The proof is provided in Appendix B (Huo, Liu, et al., 2024).

Remark 3. The required noise intensity is proportional to the sparsification ratio, k/d. Therefore, to achieve the same level of DP, the required noise under our algorithm can be reduced by decreasing k. In other words, the fewer transmitted bits, the less noise required for (ε, δ)-DP. The assumption that $\varepsilon \in (0, 1]$ is motivated by the need to ensure the validity of theoretical results, such as composition theorems and privacy amplification, which often require ε to be small. Additionally, small ε aligns with the goal of providing strong privacy guarantees, making this range both theoretically and practically relevant for differential privacy research.

Remark 4. Common compression methods include quantizers and sparsifiers. However, quantization can increase the sensitivity of gradient updates and disrupt the distribution of Gaussian perturbations, making both algorithm design and analysis more difficult. In contrast, sparsifiers simply set some elements to zero, reducing the sensitivity of the messages and making privacy amplification more tractable. Moreover, compared to the Top-k sparsifier, the random-k sparsifier introduces additional randomness, further enhancing the privacy guarantee. In the future, it will be interesting to study the potential privacy amplification under other compression schemes.

5. Convergence analysis

This section presents the convergence analysis of Algorithm 1. In each step of the algorithm, a global cubic upper bound function $\phi : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ for $f(w)$ is constructed as

$$\begin{aligned}&\phi(v; w)\\ \triangleq& f(w) + \langle \nabla f(w), v - w\rangle + \frac{1}{2}\left\langle \nabla^2 f(w)(v - w), v - w\right\rangle\\ &+ \frac{M}{6}\|v - w\|^3, \quad \forall v \in \mathcal{X},\end{aligned}$$

and local cubic upper bound functions $\phi_i : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ for $f_i(w)$, $i \in \{1, 2, \ldots, n\}$, as

$$\begin{aligned}&\phi_i(v; w)\\ \triangleq& f_i(w) + \langle \nabla f_i(w), v - w\rangle + \frac{1}{2}\left\langle \nabla^2 f_i(w)(v - w), v - w\right\rangle\\ &+ \frac{M}{6}\|v - w\|^3, \quad \forall v \in \mathcal{X}.\end{aligned} \tag{9}$$

Algorithm 2 uses the standard SGD to solve (5) and the local cubic upper bound $\phi_i(x; x_t)$ is a strongly convex function. Since each sample $\zeta_i^{(j)}$ is chosen with equal probability, i.e., $\mathbb{P}(\zeta_i^{(j)}) = 1/m$, $\mathbb{E}[\hat{g}_{i,t}] = \sum_{j=1}^m \nabla l(x, \zeta_i^{(j)})\mathbb{P}(\zeta_i^{(j)}) = \nabla f_i(x)$ and $\mathbb{E}[\hat{H}_{i,t}] = \sum_{j=1}^m \nabla^2 l(x, \zeta_i^{(j)})\mathbb{P}(\zeta_i^{(j)}) = \nabla^2 f_i(x)$ are unbiased estimates. Therefore, we can obtain the suboptimality gap based on the typical SGD analysis.

Lemma 2. *Suppose that Assumptions 2–1 hold. Given parameters $\varepsilon \in (0, 1]$, $\delta_0 \in (0, 1]$, and $w \in \mathcal{X}$ the output of Algorithm 2, if we set the number of local iterations as*

$$\tau = \frac{(L_0 + L_1 D + MD^2/2)^2 \varepsilon^2 m^2}{kT \log(1/\delta_0)(L_0 + L_1 D)^2}, \tag{10}$$

and the noise as (7), then $\hat{v}$ satisfies

$$\begin{aligned}&\mathbb{E}[\phi_{i,t}(\hat{v}; w)] - \min_{v \in \mathcal{X}} \phi_{i,t}(v; w)\\ =& O\left(\frac{k \log(1/\delta_0)(L_0 + L_1 D)^2 T}{\varepsilon^2 m^2 \mu}\right).\end{aligned} \tag{11}$$

Proof. The proof is provided in Appendix C (Huo, Liu, et al., 2024).

Lemma 2 quantifies the suboptimal gap when solving (5) with Algorithm 2 for each client in every communication round. Based on this result, we are in a position to provide the convergence of DP-FCRN.

Theorem 2. *Suppose that Assumptions 2–1 hold and the random-k sparsifier with $k \le d$ is used in Algorithm 1. Given parameters m and $\varepsilon \in (0, 1]$, $\delta_0 \in (0, 1]$, by setting the number of local iterations as (10), the step size as $\alpha > 1$ and*

$$\alpha = O\left(\frac{k \log(1/\delta_0)(L_0 + L_1 D)^2 T}{\varepsilon^2 m^2 \mu (L_0 + L_1 D + MD^2/2) D}\right),$$

and the number of iterations in DP-FCRN to

$$\begin{aligned}T = \Theta\Bigg(&\frac{\sqrt{L_2}(f(x_0) - f(x^*))^{\frac{1}{4}}}{\mu^{\frac{3}{4}}}\\ &+ \log\log\left(\frac{\varepsilon m}{\sqrt{k \log(1/\delta_0)}}\right)\Bigg),\end{aligned}$$

then the output of DP-FCRN, that is, x_T, preserves (ε, δ)-DP and

$$\begin{aligned}&\mathbb{E}[f(x_T)] - f(x^*)\\ \le& \tilde{O}\left(\frac{k \log(1/\delta_0)(L_0 + L_1 D)^2}{\varepsilon^2 m^2 \mu} \cdot \frac{\sqrt{L_2}(f(x_0) - f(x^*))^{\frac{1}{4}}}{\mu^{\frac{3}{4}}}\right).\end{aligned}$$

Proof. The proof is provided in Appendix D (Huo, Liu, et al., 2024).

Remark 5. With the boundedness established in Assumption 2, existing DP algorithms for strongly convex functions achieve the best bound for optimization error, $O\left(\frac{d}{\varepsilon^2}\right)$ (Bassily et al., 2014; Kairouz et al., 2021). This indicates that the error bound derived in Theorem 2 is optimal w.r.t. the privacy loss ε. Furthermore, our result $O\left(\frac{k}{\varepsilon^2}\right)$ reduces the error bound by a factor k/d, attributed to sparsification. This result underscores how efficient communication better balances the trade-off between privacy and utility. Unlike the recent algorithm in Chen et al. (2024), which assumes a trusted central server, we adopt a client-level differential privacy (DP) approach that offers stronger and more robust privacy protection. Furthermore, the error bound in Chen et al. (2024) increases when fewer coordinates are transmitted, implying that higher communication efficiency leads to worse convergence accuracy. In contrast, the error bound of our algorithm shows that more efficient communication reduces convergence error. Thus, in the context of federated second-order learning, we are the first to improve the trade-off between privacy and accuracy through efficient communication.

Remark 6. While DP-FCRN does not explicitly include a switching step, the proof of Theorem 2 indicates that DP-FCRN operates in two distinct phases. Initially, when x_t is distant from x^*, the convergence rate is $1/T^4$. Subsequently, as x_t approaches x^*, the algorithm transitions to the second phase with a convergence rate of $\exp(\exp(-T))$. In summary, leveraging second-order techniques in our algorithm significantly improves the oracle complexity compared to first-order methods (Liu et al., 2024).

Remark 7. The privacy analysis is independent of convexity assumptions. While many non-convex algorithms focus on first-order stationary points, which may be poor local minima or saddle points, future work will explore convergence to second-order stationary points using cubic regularization. This can reduce the risk of saddle points and improve local minima. Additionally, time-varying step sizes will be essential for optimizing the achievable bounds.

6. Numerical evaluation

In this section, we evaluate the effectiveness of DP-FCRN with different sparsification ratios and compare them to the first-order Fed-SGD with DP (Lowy & Razaviyayn, 2023).

6.1. Experimental setup

We test our algorithm on the benchmark datasets *epsilon* (Sonnenburg et al., 2006), which include 400,000 samples and 2,000 features for each sample. The data samples are evenly and randomly allocated among the $n = 40$ clients. The clients cooperatively solve the following logistic regression problem:

$$\min_{x\in\mathcal{X}} f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x),$$

where

$$f_i(x) = \frac{1}{m}\sum_{j=1}^{m} \log(1+\exp(-b_j a_j^\top x)) + \frac{1}{2m}\|x\|^2,$$

$\mathcal{X} \subseteq [-0.5, 0.5]^d$, m is the number of samples in the local dataset, and $a_j \in \mathbb{R}^d$ and $b_j \in \{-1, 1\}$ are the data samples.

As DP parameters, we consider $\varepsilon \in \{0.4, 0.6, 0.8, 1\}$ and $\delta_0 = 0.01$. The random noise is generated according to (7). We choose $\alpha = 1$. As for Fed-SGD, we set the learning rate as one. Moreover, $L_0 = 0.1$, $L_1 = 1$, $M = 1$, $D = 0.1$, $\delta_0 = 0.01$, and we calculate the value of τ using (10). In iteration t, client i processes one data point from ζ_i and the server updates x_t accordingly. Upon finishing processing the entire dataset, one epoch is completed. We conduct the algorithm for four epochs and repeat each experiment five times. We show the mean curve along with the region representing one standard deviation. The convergence performance of the algorithm is evaluated by training suboptimality and testing accuracy over iterations. Training suboptimality is calculated by $f(x_t) - f(x^*)$, where $f(x^*)$ is obtained using the LogisticSGD optimizer from scikit-learn (Pedregosa et al., 2011). Testing accuracy is determined by applying the logistic function to the entire dataset. It is calculated as the percentage of correct predictions out of the total number of predictions.

6.2. Performance and comparison with Fed-SGD

By setting the privacy budget as $\varepsilon = 0.8$, we compare the convergence performance between first-order Fed-SGD with DP and Algorithm 1 with different choices of sparsification ratio $k/d \in \{0.08, 0.1, 0.2, 1\}$. Fig. 1 implies that DP-FCRN outperforms Fed-SGD with DP in terms of optimization accuracy and convergence speed. Moreover, employing a larger sparsification ratio k/d in DP-FCRN results in worse training suboptimality, verifying Theorem 2. We find that keeping more coordinates in sparsification leads to more complete information transmission together with increased noise. The results shown in Fig. 1 indicate that, in certain settings, the benefit of noise reduction for convergence performance may outweigh the negative impacts arising from information completeness. On the other hand, there is no obvious difference in testing accuracy with different sparsification ratios, which indicates that the performance under the proposed DP-FCRN does not deteriorate much while reducing the communication burden.

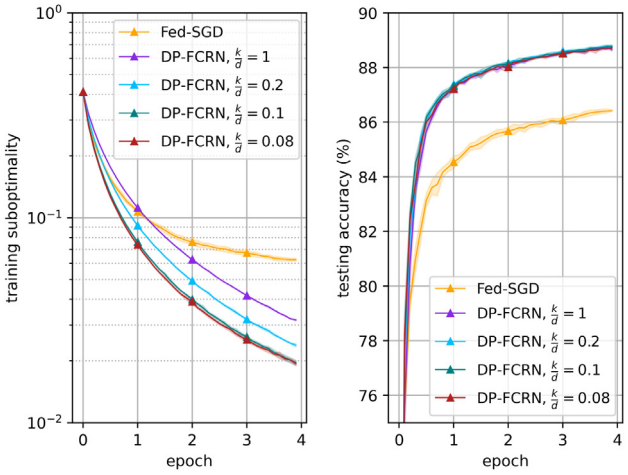


Fig. 1. Performance comparison between Fed-SGD with DP and DP-FCRN with $\varepsilon = 0.8$.

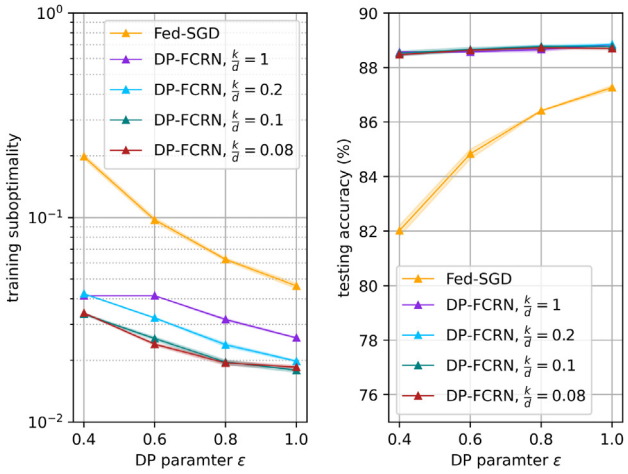


Fig. 2. Performance comparison between Fed-SGD and DP-FCRN under different DP parameters.

6.3. Trade-off between privacy and utility

Fig. 2 illustrates the trade-off between privacy and utility. It shows that when we increase the value of ε, i.e., relax the privacy requirement, the suboptimality will decrease across all the methods.

Additionally, under a tighter DP requirement, i.e., smaller ε, the performance between DP-FCRN and Fed-SGD is more significant.

7. Conclusion and future work

This paper explores communication efficiency and differential privacy within federated second-order methods. We demonstrate that the inherent sparsification characteristic can bolster privacy protection. Moreover, employing second-order methods in a privacy setting can achieve the worst-case convergence guarantees and a faster convergence rate. Experiment results illustrate that our algorithm substantially outperforms first-order Fed-SGD in terms of utility loss.

There are several promising directions for future research. Firstly, investigating methods to reduce the computational complexity of federated second-order learning approaches is valuable. Additionally, exploring communication-efficient and privacy-

preserving variants of advanced federated second-order algorithms, such as GIANT and SHED, presents promising research directions.

Acknowledgments

The work by W. Huo and L. Shi is supported by the Hong Kong RGC General Research Fund 16203723. The work by K. Ding is supported by NSFC under Grant No.62303212, Shenzhen Science and Technology Program, under Grant No. 20231120102304001, NSF of Guangdong Province under Grant No. 2024A1515011633. The work by K. H. Johansson was supported in part by Swedish Research Council Distinguished Professor Grant 2017-01078, Knut and Alice Wallenberg Foundation Wallenberg Scholar Grant, and Swedish Strategic Research Foundation FUSS SUCCESS Grant.

References

Balle, B., & Wang, Y.-X. (2018). Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International conference on machine learning* (pp. 394–403). PMLR.

Bassily, R., Smith, A., & Thakurta, A. (2014). Private empirical risk minimization: Efficient algorithms and tight error bounds. In *IEEE 55th annual symposium on foundations of computer science* (pp. 464–473).

Boyd, S. P., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Chen, W.-N., Choquette-Choo, C. A., & Kairouz, P. (2021). Communication efficient federated learning with secure aggregation and differential privacy. In *NeurIPS workshop privacy in machine learning*.

Chen, W.-N., Song, D., Ozgur, A., & Kairouz, P. (2024). Privacy amplification via compression: Achieving the optimal privacy-accuracy-communication trade-off in distributed mean estimation. *Advances in Neural Information Processing Systems*, *36*.

Dal Fabbro, N., Dey, S., Rossi, M., & Schenato, L. (2024). SHED: A Newton-type algorithm for federated learning based on incremental Hessian eigenvector sharing. *Automatica*, *160*, Article 111460.

Ding, J., Liang, G., Bi, J., & Pan, M. (2021). Differentially private and communication efficient collaborative learning. In *Proceedings of the AAAI conference on artificial intelligence*: *vol. 35*, (8), (pp. 7219–7227).

Dwork, C. (2006). Differential privacy. In *International colloquium on automata, languages, and programming* (pp. 1–12). Springer.

Ganesh, A., Haghifam, M., Steinke, T., & Guha Thakurta, A. (2024). Faster differentially private convex optimization via second-order methods. *Advances in Neural Information Processing Systems*, *36*.

Hao, M., Li, H., Luo, X., Xu, G., Yang, H., & Liu, S. (2019). Efficient and privacy-enhanced federated learning for industrial artificial intelligence. *IEEE Transactions on Industrial Informatics*, *16*(10), 6532–6542.

Hu, R., Guo, Y., & Gong, Y. (2023). Federated learning with sparsified model perturbation: Improving accuracy under client-level differential privacy. *IEEE Transactions on Mobile Computing*.

Huang, L., Wu, J., Shi, D., Dey, S., & Shi, L. (2024). Differential privacy in distributed optimization with gradient tracking. *IEEE Transactions on Automatic Control*.

Huo, W., Liu, C., Ding, K., Johansson, K. H., & Shi, L. (2024). Federated cubic regularized Newton learning with sparsification-amplified differential privacy. arXiv preprint arXiv:2408.04315.

Huo, W., Tsang, K. F. E., Yan, Y., Johansson, K. H., & Shi, L. (2024). Distributed Nash equilibrium seeking with stochastic event-triggered mechanism. *Automatica*, *162*, Article 111486.

Kairouz, P., Liu, Z., & Steinke, T. (2021). The distributed discrete gaussian mechanism for federated learning with secure aggregation. In *International conference on machine learning* (pp. 5201–5212). PMLR.

Lacoste-Julien, S., Schmidt, M., & Bach, F. (2012). A simpler approach to obtaining an $O(1/t)$ convergence rate for the projected stochastic subgradient method. arXiv preprint arXiv:1212.2002.

Li, Z., & Richtárik, P. (2021). CANITA: Faster rates for distributed convex optimization with communication compression. *Advances in Neural Information Processing Systems*, *34*, 13770–13781.

Li, Z., Zhao, H., Li, B., & Chi, Y. (2022). SoteriaFL: A unified framework for private federated learning with communication compression. *Advances in Neural Information Processing Systems*, *35*, 4285–4300.

Liu, C., Johansson, K. H., & Shi, Y. (2024). Distributed empirical risk minimization with differential privacy. *Automatica*, *162*, Article 111514.

Liu, L., Wang, Y., Liu, G., Peng, K., & Wang, C. (2022). Membership inference attacks against machine learning models via prediction sensitivity. *IEEE Transactions on Dependable and Secure Computing*.

Liu, H., Zhang, J., So, A. M.-C., & Ling, Q. (2023). A communication-efficient decentralized Newton's method with provably faster convergence. *IEEE Transactions on Signal and Information Processing over Networks*, *9*, 427–441.

Lowy, A., & Razaviyayn, M. (2023). Private federated learning without a trusted server: Optimal algorithms for convex losses.

Maritan, A., Sharma, G., Schenato, L., & Dey, S. (2023). Network-GIANT: Fully distributed Newton-type optimization via harmonic hessian consensus. In *IEEE globecom workshops* (pp. 902–907).

Mohammadi, N., Bai, J., Fan, Q., Song, Y., Yi, Y., & Liu, L. (2021). Differential privacy meets federated learning under communication constraints. *IEEE Internet of Things Journal*, *9*(22), 22204–22219.

Nesterov, Y., & Polyak, B. T. (2006). Cubic regularization of Newton method and its global performance. *Mathematical Programming*, *108*(1), 177–205.

Nguyen, A., Do, T., Tran, M., Nguyen, B. X., Duong, C., Phan, T., Tjiputra, E., & Tran, Q. D. (2022). Deep federated learning for autonomous driving. In *IEEE intelligent vehicles symposium* (pp. 1824–1830).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Richtárik, P., Sokolov, I., & Fatkhullin, I. (2021). EF21: A new, simpler, theoretically better, and practically faster error feedback. *Advances in Neural Information Processing Systems*, *34*, 4384–4396.

Safaryan, M., Islamov, R., Qian, X., & Richtárik, P. (2021). FedNL: Making Newton-type methods applicable to federated learning. arXiv preprint arXiv:2106.02969.

Sonnenburg, S., Rätsch, G., Schäfer, C., & Schölkopf, B. (2006). Large scale multiple kernel learning. *Journal of Machine Learning Research*, *7*, 1531–1565.

Wang, L., Liu, W., Guo, F., Qiao, Z., & Wu, Z. (2024). Differentially private average consensus with improved accuracy-privacy trade-off. *Automatica*, *167*, Article 111769.

Wang, L., Manchester, I. R., Trumpf, J., & Shi, G. (2023). Differential initial-value privacy and observability of linear dynamical systems. *Automatica*, *148*, Article 110722.

Yin, X., Ng, B. W., He, J., Zhang, Y., & Abbott, D. (2014). Accurate image analysis of the retina using hessian matrix and binarisation of thresholded entropy with application of texture mapping. *PloS One*, *9*(4), Article e95943.

Yuan, Z., Xu, S., & Zhu, M. (2024). Federated reinforcement learning for robot motion planning with zero-shot generalization. *Automatica*, *166*, Article 111709.

Zhang, Z., Che, K., Yang, S., & Xu, W. (2024). Communication-efficient distributed cubic Newton with compressed lazy hessian. *Neural Networks*, *174*, Article 106212.

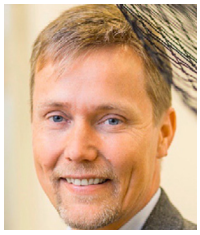
Zhang, X., Fang, M., Liu, J., & Zhu, Z. (2020). Private and communication-efficient edge learning: a sparse differential gaussian-masking distributed SGD approach. In *Proceedings of the 21st international symposium on theory, algorithmic foundations, and protocol design for mobile networks and mobile computing* (pp. 261–270).

Zhu, L., Liu, Z., & Han, S. (2019). Deep leakage from gradients. In *Advances in neural information processing systems*: *vol. 32*.

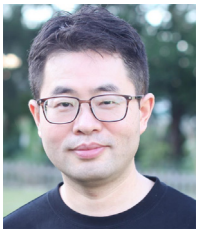
Wei Huo received her B.S. degree in Electronic and Information Engineering from Huazhong University of Science and Technology, Wuhan, China, in 2020, and Ph.D. degree in Electrical and Computer Engineering from Hong Kong University of Science and Technology, Hong Kong, in 2024. From August 2023 to December 2023, she was a visiting student in the School of Electrical Engineering and Computer Science in KTH Royal Institute of Technology. In December 2024, she joined Wireless Technology Lab, 2012, at Huawei, Shanghai, China, where she is currently a Senior Engineer. Her research interests include multi-agent systems, distributed optimization, privacy protection, and agentic AI for 6G.

Changxin Liu received his Ph.D. in mechanical engineering from the University of Victoria, Victoria, BC, Canada, in 2021. From 2021 to 2024, he worked as a postdoctoral researcher at the School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden. In October 2024, he joined the Department of Automation at East China University of Science and Technology, Shanghai, China, where he is currently a Professor. His research interests include distributed optimization and control, machine learning, and combinatorial optimization, with applications to the process industry.

Kemi Ding received the B.S. degree in Electronic and Information Engineering from Huazhong University of Science and Technology, China, in 2014 and the Ph.D. degree in the Department of Electronic and Computer Engineering from Hong Kong University of Science and Technology in 2018. Currently she is an Associate Professor in the school of Automation and Intelligent Manufacturing. Prior to this, she was a postdoctoral researcher in the School of Electrical, Computer and Energy Engineering, Arizona State University from September 2018 to August 2019, and a postdoctoral researcher at the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, from August 2019 to October 2021. Her current research interests include security and privacy in Cyber-physical systems/Internet-of-things, intelligent control and decision, networked state estimation, and game theory.



Karl Henrik Johansson is Professor with the School of Electrical Engineering and Computer Science at KTH Royal Institute of Technology in Sweden and Director of Digital Futures. He received M.Sc. degree in Electrical Engineering and Ph.D. in Automatic Control from Lund University. He has held visiting positions at UC Berkeley, Caltech, NTU, HKUST Institute of Advanced Studies, and NTNU. His research interests are in networked control systems and cyber–physical systems with applications in transportation, energy, and automation networks. He is President of the European Control Association and member of the IFAC Council, and has served on the IEEE Control Systems Society Board of Governors and the Swedish Scientific Council for Natural Sciences and Engineering Sciences. He has received several best paper awards and other distinctions from IEEE, IFAC, and ACM. He has been awarded Swedish Research Council Distinguished Professor, Wallenberg Scholar with the Knut and Alice Wallenberg Foundation, Future Research Leader Award from the Swedish Foundation for Strategic Research, the triennial IFAC Young Author Prize, and IEEE Control Systems Society Distinguished Lecturer. He is Fellow of the IEEE and the Royal Swedish Academy of Engineering Sciences.



Ling Shi received his B.E. degree in Electrical and Electronic Engineering from The Hong Kong University of Science and Technology (HKUST) in 2002 and Ph.D. degree in Control and Dynamical Systems from The California Institute of Technology (Caltech) in 2008. He is currently a Professor in the Department of Electronic and Computer Engineering at HKUST with a joint appointment in the Department of Chemical and Biological Engineering (2025–2028). His research interests include cyber–physical systems security, networked control systems, sensor scheduling, event-based state estimation, and multi-agent robotic systems (UAVs and UGVs). He served as an editorial board member for the European Control Conference 2013–2016. He was a subject editor for International Journal of Robust and Nonlinear Control (2015–2017), an associate editor for IEEE Transactions on Control of Network Systems (2016–2020), an associate editor for IEEE Control Systems Letters (2017–2020), and an associate editor for a special issue on Secure Control of Cyber Physical Systems in IEEE Transactions on Control of Network Systems (2015–2017). He also served as the General Chair of the 23rd International Symposium on Mathematical Theory of Networks and Systems (MTNS 2018). He is currently serving as a member of the Engineering Panel (Joint Research Schemes) of the Hong Kong Research Grants Council (RGC) (2023–2026). He received the 2024 Chen Fan-Fu Award given by the Technical Committee on Control Theory, Chinese Association of Automation (TCCT, CAA). He is a member of the Young Scientists Class 2020 of the World Economic Forum (WEF), a member of The Hong Kong Young Academy of Sciences (YASHK), and he is an IEEE Fellow.